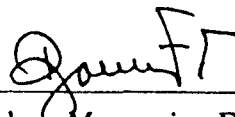


# Algoritmos para a Simulação da Dinâmica de Sistemas de Energia Elétrica em Computadores Paralelos

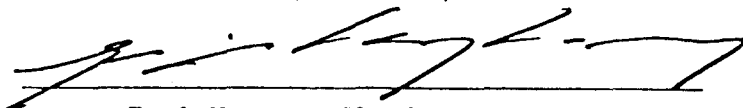
*Ildemar Cassana Decker*

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

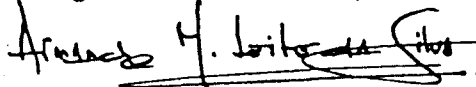
Aprovada por:



Prof. Djalma Mosqueira Falcão, Ph.D.  
(Presidente)



Prof. Eugenius Kaszkurewicz, D.Sc.



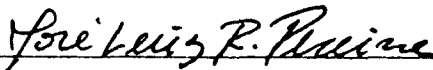
Prof. Armando M. Leite da Silva, Ph.D.



Eng. Nelson Martins, Ph.D.



Prof. Alcir José Monticelli, D.Sc.



Prof. José Luiz R. Pereira, Ph.D.

RIO DE JANEIRO, RJ — BRASIL

SETEMBRO DE 1993

DECKER, ILDEMAR CASSANA

Algoritmos para a Simulação da Dinâmica de Sistemas de Energia Elétrica em  
Computadores Paralelos

[Rio de Janeiro] 1993

xvi, 215 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Elétrica, 1993)

Tese — Universidade Federal do Rio de Janeiro, COPPE

1. Estabilidade Transitória
2. Processamento Paralelo
3. Simulação da Dinâmica

I. COPPE/UFRJ II. Título (série).

*À minha esposa Isabel Cristina  
Aos meus filhos Vinícius e Brígida*

# Agradecimentos

Aos professores Djalma M. Falcão e Eugenius Kaszkurewicz pela orientação segura, amizade e dedicação na implantação do Laboratório de Computação Paralela da COPPE/UFRJ, elementos fundamentais para a realização deste trabalho.

Ao professor José Luiz R. Pereira pelo apoio, amizade, valiosas contribuições e pelo seu trabalho na coordenação do Laboratório de Sistemas de Potência (LASPOT).

Ao Dr. Nelson Martins pelas interessantes discussões e valiosas contribuições.

À colega Doutoranda Maria Helena M. Vale pelas discussões e contribuições, especialmente pela realização da maioria das decomposições das redes elétricas utilizadas nos testes computacionais.

Ao colega Doutorando Jorge Mario Campagnolo pela amizade e contribuições nas importantes discussões técnicas que realizamos ao longo do desenvolvimento deste trabalho.

Aos colegas do Laboratório de Computação Paralela da COPPE/UFRJ, em particular ao Engo Francisco C. Mota pela ajuda na utilização do computador paralelo iPSC/860.

À Enga Carmen L. T. Borges pelo apoio na utilização do LASPOT.

A UFSC e a CAPES/PICD pelo suporte financeiro.

À minha esposa Enga Isabel Cristina U. Decker pelo grande apoio, carinho e incentivo demonstrado nesses muitos anos que temos vividos juntos.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências (D.Sc.).

# Algoritmos para a Simulação da Dinâmica de Sistemas de Energia Elétrica em Computadores Paralelos

Ildemar Cassana Decker

SETEMBRO, 1993

Orientadores: Djalma Mosqueira Falcão e Eugenius Kaszkurewicz

Programa: Engenharia Elétrica

Na presente tese investiga-se a utilização de computadores paralelos na simulação da dinâmica de sistemas de energia elétrica. São apresentados três novos algoritmos paralelos desenvolvidos para a resolução das equações diferenciais e algébricas não lineares que representam a dinâmica desses sistemas. Nos dois primeiros, baseados no esquema *Alternado Entrelaçado Implícito*, explora-se o paralelismo no espaço. No terceiro, baseado no esquema *Simultâneo Implícito*, explora-se o paralelismo no tempo. Uma das principais novidades é a utilização de métodos iterativos do tipo *Gradiente Conjugado* na resolução paralela de sistemas algébricos lineares simétricos e assimétricos, que surgem no âmbito desses algoritmos. São também desenvolvidas técnicas de pré-condicionamento, para os sistemas lineares, e de inicialização das variáveis de todos os passos de integração tomados simultaneamente, no caso do algoritmo que explora o paralelismo no tempo.

Foram feitas implementações computacionais paralelas e testes de avaliação de desempenho dos três algoritmos desenvolvidos, usando-se sistemas elétricos com 118 e 616 barras e os computadores paralelos NCP I e iPSC/860 da COPPE/UFRJ. São também realizadas comparações de desempenho em relação a simulações processadas em máquinas do tipo "mainframe". Os resultados mostram uma considerável redução no tempo computacional quando utiliza-se o processamento paralelo e a efetividade da utilização dos métodos tipo Gradiente Conjugado pré-condicionados.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.).

## Algorithms for Power Systems Dynamics Simulation with Parallel Computers

Ildemar Cassana Decker

1993, SEPTEMBER

Thesis Supervisor: Djalma Mosqueira Falcão and Eugenius Kaszkurewicz

Department: Electrical Engineering

In this thesis the use of parallel computers in power systems dynamics simulation is investigated. Three new parallel algorithms developed for the solution of a set of nonlinear algebraic and ordinary differential equations that model the dynamic of these systems are presented. The first two algorithms, based on implicit interlaced alternate scheme, explore the parallelism in space. In the third one, based on implicit simultaneous scheme, the parallelism in time is explored. One of the main novelty is the use of the iterative gradient conjugate type methods, in the parallel solution of the symmetric and non-symmetric linear algebraic system of equations, which appears in these algorithms. Preconditioning techniques for the linear systems and initialization techniques for all steps that are taken simultaneously, in the algorithms that explore the parallelism in time, are also developed.

Computational implementations were made in parallel machines NCP I and Intel iPSC/860, in order to evaluate the performance of the algorithms, when applied to the electric power systems with 118 and 616 buses. The results are also compared with the performance of the simulations processed in the mainframe computers. The results show considerable reduction in the CPU time when parallel processing is used in conjunction with preconditioned conjugate gradient type methods.

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	O Processamento de Alto Desempenho . . . . .	2
1.1.1	Aplicabilidade em Análise de Redes Elétricas . . . . .	4
1.2	Simulação da Dinâmica de SEE em Ambiente Paralelo . . . . .	5
1.3	Estrutura do Trabalho . . . . .	7
<b>2</b>	<b>Modelagem Matemática Básica</b>	<b>9</b>
2.1	Aspectos Gerais . . . . .	9
2.2	Esquemas Básicos de Solução . . . . .	14
2.2.1	Esquema Alternado Implícito . . . . .	15
2.2.2	Esquema Simultâneo Implícito . . . . .	18
2.3	Comentários Finais . . . . .	24
<b>3</b>	<b>Simulação da Dinâmica de Sistemas de Energia Elétrica em Ambiente Paralelo - Revisão da Literatura</b>	<b>26</b>
3.1	Paralelização no Espaço . . . . .	27
3.1.1	O Método VDHN Paralelo . . . . .	27
3.1.2	O Método “Newton-W Matrix” . . . . .	28
3.1.3	O Simulador Digital em Tempo Real . . . . .	29
3.2	Paralelização no Tempo . . . . .	31
3.2.1	Formulação Matemática . . . . .	31
3.2.2	Comentários . . . . .	34
3.3	Paralelização Simultânea no Espaço e Tempo . . . . .	35
3.3.1	Solução por Gauss-Jacobi (Substituições Sucessivas) . . . . .	36
3.3.2	Solução pelo Método Gauss-Jacobi-Block-Newton . . . . .	38
3.3.3	Comentários . . . . .	40

3.4	Métodos do Tipo Newton Relaxado . . . . .	40
3.4.1	O Método SOR-Newton . . . . .	41
3.4.2	O Método Maclaurin-Newton . . . . .	41
3.4.3	Discussão . . . . .	41
3.5	Relaxação da Forma de Onda (RFO) . . . . .	42
3.5.1	O Método RFO para Sistemas de Equações Diferenciais Ordinárias . . . . .	43
3.5.2	O Método da RFO para Sistemas de Equações Diferenciais e Algébricas . . . . .	44
3.5.3	Aplicações a SDSEE/Decomposição . . . . .	46
3.6	A Sinergia do Processamento Paralelo-Vetorial . . . . .	48
3.7	Métodos Iterativos do Tipo Gradiente Conjugado . . . . .	49
3.8	Conclusões . . . . .	51
<b>4</b>	<b>Esquema Alternado Entrelaçado Implícito - Paralelização no Espaço</b>	<b>55</b>
4.1	Formulação Matemática . . . . .	56
4.2	Métodos de Solução . . . . .	56
4.3	O Método Gradiente Conjugado Pré-condicionado . . . . .	58
4.4	O Algoritmo AEIGC . . . . .	59
4.4.1	Solução da Rede Elétrica pelo Método GC-Fatoração LU . . . . .	59
4.4.2	Pré-condicionamento . . . . .	61
4.4.3	Paralelização do Algoritmo AEIGC . . . . .	62
4.5	O Algoritmo AEIGCRD . . . . .	69
4.5.1	Pré-condicionamento . . . . .	69
4.5.2	Paralelização do Algoritmo AEIGCRD . . . . .	70
4.6	Conclusões . . . . .	76
<b>5</b>	<b>Esquema Simultâneo Implícito - Paralelização no Tempo</b>	<b>77</b>
5.1	Modelagem Matemática . . . . .	77
5.1.1	Equações para um Único Passo de Integração . . . . .	78
5.1.2	Equações para $p$ Passos de Integração . . . . .	79
5.2	Métodos Numéricos de Solução . . . . .	84
5.2.1	Métodos do Tipo Gradiente Conjugado . . . . .	85
5.2.2	Pré-condicionamento . . . . .	87



5.2.3	O Pós-condicionamento Explícito . . . . .	88
5.2.4	Inicialização . . . . .	90
5.3	A Paralelização do Algoritmo SIMPT . . . . .	91
5.4	Conclusões . . . . .	92
<b>6</b>	<b>Avaliação de Desempenho em Ambientes Computacionais Paralelos</b>	<b>97</b>
6.1	Computadores Paralelos Utilizados . . . . .	98
6.2	Sistemas Elétricos de Testes . . . . .	99
6.3	Programas Computacionais . . . . .	99
6.4	Parâmetros das Simulações . . . . .	101
6.5	Avaliação de Desempenho e Figuras de Mérito . . . . .	104
6.5.1	O Padrão de Comparação . . . . .	104
6.5.2	Eficiências e “Speed ups” . . . . .	104
6.6	Resultados para o Algoritmo AEIGC . . . . .	107
6.6.1	Parâmetros das Decomposições do Sistema . . . . .	107
6.6.2	Resultados Computacionais . . . . .	110
6.7	Resultados para o Algoritmo AEIGCRD . . . . .	117
6.7.1	Parâmetros das Decomposições do Sistema . . . . .	117
6.7.2	Resultados Computacionais . . . . .	118
6.8	Análise Comparativa dos Resultados - Paralelização no Espaço . . . . .	124
6.9	Resultados para o Algoritmo SIMPT . . . . .	127
6.9.1	Avaliação Comparativa de Desempenho - Paralelização no Tempo	134
6.10	Conclusões . . . . .	136
<b>7</b>	<b>Conclusões</b>	<b>139</b>
7.1	Considerações Gerais . . . . .	139
7.2	Sugestões de Estudos Futuros . . . . .	142
<b>A</b>	<b>Modelagem dos Elementos do Sistema</b>	<b>144</b>
A.1	Máquinas Síncronas . . . . .	144
A.1.1	Equações Diferenciais Eletromecânicas . . . . .	145
A.1.2	Equações Diferenciais Elétricas . . . . .	146
A.1.3	Equações Algébricas do Estator . . . . .	147
A.2	Sistemas de Excitação . . . . .	147

A.2.1	Sistema de Excitação IEEE Tipo 1 . . . . .	147
A.2.2	Sistema de Excitação Tipo IEEE AC4 . . . . .	149
A.2.3	Sistema de Transmissão . . . . .	150
A.3	Acoplamento da Máquina Síncrona à Rede de Transmissão . . . . .	151
A.3.1	Cargas . . . . .	153
<b>B</b>	<b>Esquemas Tradicionais para a SDSEE</b>	<b>155</b>
B.1	Algebrização das Equações Diferenciais Usando o Método Trapezoidal Implícito . . . . .	155
B.1.1	Equações das Máquinas Síncronas . . . . .	156
B.1.2	Sistema de Excitação IEEE Tipo 1 . . . . .	158
B.2	Esquema Alternado Entrelaçado Implícito . . . . .	160
B.2.1	Resumo da Modelagem . . . . .	161
B.2.2	Equivalente Norton para o Estator . . . . .	162
B.2.3	Cargas . . . . .	164
B.2.4	Interface . . . . .	165
B.2.5	Síntese do Modelo Global . . . . .	166
B.2.6	Algoritmo Alternado Entrelaçado Implícito (AEI) . . . . .	167
B.3	Esquema Simultâneo Implícito (SI) . . . . .	171
B.3.1	Resumo da Modelagem . . . . .	171
B.3.2	Equações de Interface . . . . .	172
B.3.3	Equações das Cargas . . . . .	173
B.3.4	Modelo Global para o Intervalo $t - \Delta t$ a $t$ . . . . .	173
B.3.5	Algoritmo Simultâneo Implícito (SI) . . . . .	176
<b>C</b>	<b>Esquema Simultâneo Implícito Paralelo no Tempo (SIMPT) - Modelagem Matemática</b>	<b>178</b>
C.1	Equações dos Elementos do Sistema . . . . .	178
C.1.1	Sistema de Excitação IEEE Tipo 1 . . . . .	178
C.1.2	Sistema de Excitação Tipo AC4 . . . . .	179
C.1.3	Equações Elétricas do Rotor . . . . .	179
C.1.4	Equações Eletromecânicas do Rotor . . . . .	180
C.1.5	Equivalente Norton para o Estator . . . . .	181
C.1.6	Síntese das Equações Algébricas da Rede, Estatores e Cargas .	183

C.2	Modelagem Global do Sistema . . . . .	184
C.2.1	Modelo Global por Barra . . . . .	184
C.2.2	Modelo Global para o Sistema no Intervalo de Tempo ( $t - \Delta t$ ) . . . . .	185
C.2.3	Modelo para uma Janela de Integração . . . . .	186
C.3	Resolução por Newton-Raphson . . . . .	188
C.3.1	Equação Linear Jacobiana . . . . .	188
C.3.2	Estrutura das Sub-matrizes da Matriz Jacobiana . . . . .	190
<b>D</b>	<b>O Método Gradiente Conjugado</b>	<b>197</b>
D.1	O Método Gradiente Conjugado . . . . .	197
D.1.1	Conceitos Básicos . . . . .	197
D.1.2	Pré-condicionamento . . . . .	200
<b>E</b>	<b>Solução Inicial Aproximada por Série de Taylor</b>	<b>203</b>
E.1	Modelo Matemático . . . . .	203
E.2	Aplicação da Expansão em Série de Taylor . . . . .	204
E.3	Determinação das Derivadas de Alta Ordem para $\delta_i$ e $\omega_i$ . . . . .	204
E.4	Potência Elétrica e suas Derivadas . . . . .	205
E.5	Determinação das Tensões e suas Derivadas . . . . .	205
E.6	Determinação das Injeções de Correntes e suas Derivadas . . . . .	206

# Lista das Figuras

2.1	Principais malhas de controle de um SEE . . . . .	10
2.2	Estrutura usual das equações envolvidas no modelo matemático para estudos de estabilidade transitória . . . . .	12
4.1	Fluxograma do algoritmo AEIGC paralelo para um passo de integração	68
4.2	Fluxograma do algoritmo AEIGCRD paralelo para um passo de integração . . . . .	74
4.3	Fluxograma do algoritmo GC paralelo para resolver $I(E^k, V^k) = Y.V^{k+1}$ AEIGCRD . . . . .	75
5.1	Fluxograma do algoritmo SIMPT paralelo no tempo para $p$ passos de integração . . . . .	93
5.2	Fluxograma do algoritmo biconjugado paralelo para resolver a equação linear Jacobiana . . . . .	94
5.3	Fluxograma do algoritmo Bi-CGSTAB paralelo para resolver a equação linear jacobiana . . . . .	95
6.1	Estrutura constitucional do sistema computacional PPSDS - (“Parallel Power System Dynamic Simulation”) . . . . .	102
6.2	Sistema IEEE118 : Decomposição em 4 subredes (barras de interconexão: 5, 17, 20, 38, 46, 49, 68, 69, 77, 100) . . . . .	108
6.3	Sistema IEEE118 : Decomposição em 8 subredes (barras de interconexão: 5, 17, 20, 23, 30, 33, 34, 49, 62, 65, 69, 77, 82, 94, 100) . . . . .	108
6.4	“Speed ups” $S_{a1}$ e $S_{a2}$ nos computadores NCP I e iPSC/860 . . . . .	115
6.5	“Speed up” obtido desprezando-se a comunicação (NCP I) . . . . .	115
6.6	Tempo de CPU para o cálculo do resíduo e de $\hat{Y}, d^k$ (NCP I) . . . . .	116
6.7	Número total de iterações do método GC (NCP I) . . . . .	116

6.8	Tempos de CPU para diversas parcelas do algoritmo AEIGCRD (seqüencial) . . . . .	121
6.9	Tempos de CPU e comunicação, em processamentos seqüenciais e paralelos, para decomposições automatizadas (linha cheia) e não automatizadas (linha tracejada) . . . . .	122
6.10	Melhores tempos de CPU e comunicação, em processamentos seqüenciais e paralelos no iPSC/860, para os algoritmos AEIC, AEIGC e AEIGCRD . . . . .	125
6.11	Número médio de iterações nos algoritmos tipo GC para a inicialização por série de Taylor em $x$ (esquerda) e em $x$ e $V^e$ (direita) . . . . .	130
6.12	Número total de iterações para os algoritmos de Newton e Bi-CGSTAB, com a inicialização por série de Taylor em $x$ (linha tracejada) e em $x$ e $V^e$ (linha cheia) . . . . .	132
6.13	Tempos de CPU para diversas parcelas do algoritmo SIMPT, em processamentos seqüenciais com níveis de otimização 0 (linha tracejada) e 4 (linha cheia) na fase de compilação . . . . .	133
6.14	Tempos de CPU e Comunicação para o algoritmo SIMPT seqüencial e paralelo . . . . .	136
A.1	Sistema de excitação IEEE tipo 1 ( <i>ação contínua</i> ) . . . . .	148
A.2	Sistema de excitação IEEE tipo AC4 ( <i>Alternator supplied controlled rectifier excitation system</i> ) . . . . .	150
A.3	Diagrama composto de transformação de coordenadas $d - q \rightarrow \mathcal{R} - \mathcal{S}$ e vice-versa . . . . .	152
B.1	Circuito equivalente Norton do estator . . . . .	163
E.1	Solução paralela aproximada por série de Taylor para uma janela, de integração com $p$ passos . . . . .	208

# Lista das Tabelas

2.1	Esquemas de resolução das EDAs para a SDSEE . . . . .	14
6.1	Número de iterações para o algoritmo AEIC Seqüencial . . . . .	104
6.2	Tempos de CPU (s) para o algoritmo AEIC Seqüencial . . . . .	106
6.3	Parâmetros das decomposições - IEEE 118 e SSUD 616 . . . . .	109
6.4	Tempos de CPU, “speed ups” e eficiências (no NCP I) . . . . .	111
6.5	Tempos de CPU, “speed ups” e eficiências (no iPSC/860) . . . . .	112
6.6	Tempos de CPU, “speed ups” e eficiências desprezando-se a comunicação (no NCP I) . . . . .	113
6.7	Parâmetros das decomposições - IEEE 118 e SSUD 616 . . . . .	119
6.8	Tempos de CPU, “speed ups” e eficiências (no iPSC/860) . . . . .	120
6.9	Tempos de CPU, “speed ups” e eficiências - paralelização no espaço .	124
6.10	Tempos de CPU (s) em processamentos paralelos . . . . .	129
6.11	Tempos de CPU (s) em processamentos seqüenciais . . . . .	131
6.12	“Speed ups” relativos ao processamento seqüencial com $p$ passos tomados simultaneamente . . . . .	134
6.13	“Speed ups” relativos ao processamento seqüencial com 1 passo de integração . . . . .	135
6.14	Parâmetros de alguns computadores paralelos (valores de pico) . . . .	138

## Lista de Abreviações

**AEI** - Alternado Entrelaçado Implícito;

**AEIC** - Alternado Entrelaçado Implícito Convencional;

**AEIGC** - Esquema Alternado Entrelaçado Implícito em que a resolução das equações da rede elétrica é feita por um método híbrido (matriz admitância nodal na forma BBDF);

**AEIGCRD** - Esquema Alternado Entrelaçado Implícito em que a resolução das equações da rede elétrica é feita pelo método Gradiente Conjugado pré-condicionado;

**AEIPS** - Programa computacional seqüencial que utiliza o esquema AEIC;

**BBDF** - “Block Bordered Diagonal Form”;

**CGS** - “Conjugate Gradient Squared”;

**CSC** - Centro de Supervisão e Controle;

**EAs** - Equações Algébricas;

**EDAs** - Equações Diferenciais e Algébricas;

**EDOs** - Equações Diferenciais Ordinárias;

**BCG** - Gradiente Biconjugado;

**GC** - Gradiente Conjugado;

**GJBN** - Gauss-Jacobi-Bolock-Newton;

**IEEE 118** - Sistema padrão de testes do IEEE, constituído de 34 geradores (20 representados dinamicamente), 118 barras e 179 linhas de transmissão e transformadores;

**NBDF** - “Near Block Diagonal Form”;

**NCP I** - Computador paralelo desenvolvido pela COPPE/UFRJ, baseado em processadores do tipo Transputers T800, com topologia hipercúbica e memória distribuída;

**PPSDS** - “Parallel Power System Dynamic Simulation”;

**RFO** - Relaxação da Forma de Onda;

**SEE** - Sistema de Energia Elétrica;

**SDSEE** - Simulação da Dinâmica de Sistemas de Energia Elétrica

**SI** - Simultâneo Implícito;

**SIMPT** - Simultâneo Implícito Paralelo no Tempo;

**SOR-Newton** - Sobrerelaxação-Newton;

**SSUD 616** - Configuração do sistema de energia elétrica das regiões Sul e Sudeste do Brasil, para o ano de 1987, com algumas subareas representadas por equivalentes estáticos (88 geradores representados dinamicamente, 616 barras e 995 linhas de transmissão e transformadores);

**VDHN** - “Very Dishonest Newton”;



# Capítulo 1

## Introdução

O objetivo geral desta tese é investigar a utilização de computadores paralelos na Simulação da Dinâmica de Sistemas de Energia Elétrica (SDSEE). A descrição matemática da dinâmica de um Sistema de Energia Elétrica (SEE) é realizada por um conjunto de equações diferenciais ordinárias não-lineares e um conjunto de equações algébricas também não-lineares [1]. São comuns estudos em sistemas com cerca de 3000 equações diferenciais ordinárias não-lineares e cerca de 4000 equações algébricas não-lineares [2]. Em uma discussão da referência [3], Kundur e Rogers citam que, na época (1987), já estavam realizando simulações, para fins de estudos de estabilidade transitória, de SEE constituídos de 1000 máquinas síncronas e 10.000 barras (cerca de 10.000 a 20.000 equações diferenciais ordinárias e 10.000 a 20.000 equações algébricas não-lineares, dependendo do modelo utilizado) que demandavam diversas horas de CPU em computadores do porte de um VAX 8600 ( capacidade aproximada de 1 Mflop). É importante ainda acrescentar que um estudo de estabilidade transitória requer, em adição, um elevado número de simulações relativas a diferentes configurações e pontos de operação do sistema.

Nos estudos de estabilidade transitória feitos desta forma, o diagnóstico é dado a partir da análise de curvas da evolução no tempo de variáveis eletromecânicas [1]. Este é um diagnóstico do tipo pontual, no sentido de que é válido para um determinado ponto de operação e seqüência de perturbações ( contingência ).

A crescente dimensão, complexidade e a integração dos Centros de Supervisão e Controle (CSC) na operação de SEE, têm demandado grande interesse pelo desenvolvimento de meios mais expeditos na avaliação da estabilidade transitória . Especificamente destacam-se os métodos diretos, baseados no segundo método de Liapunov, e o processamento paralelo de alto desempenho. De acordo com o objetivo geral acima

definido, o enfoque deste trabalho será na área do processamento paralelo.

## 1.1 O Processamento de Alto Desempenho

Uma das principais aplicações da computação científica é no problema da simulação do comportamento de sistemas físicos, descritos matematicamente por milhares ou centenas de milhares de equações algébricas, diferenciais ordinárias e algébricas ou diferenciais parciais. O tipo particular das equações depende do sistema físico e da natureza dos fenômenos que se quer descrever. Adicionalmente, os resultados dessas simulações devem ser apresentados em uma forma suficientemente sintética de modo a facilitar a análise. Esses requisitos demandam elevada capacidade computacional, podendo atingir o nível de Teraflops em certas áreas do conhecimento [4].

A demanda por capacidades de processamento cada vez maiores e as inerentes limitações relativas à velocidade de transmissão de sinais elétricos através de condutores, levaram os fabricantes de computadores a explorarem o paralelismo [5]. O processamento paralelo é definido por Hwang [6], como uma forma eficiente de processar informações que enfatiza a exploração de eventos concorrentes no processo de computação. Neste contexto, as máquinas apresentam uma grande diversidade de arquiteturas computacionais que, para fins de classificação, ainda podem ser organizadas nas três classes definidas por Hwang [7]: “pipeline computers”, “array processors” e sistemas multiprocessadores.

Em “pipeline computers”, entre os quais se enquadram a maioria dos supercomputadores, instruções sucessivas são executadas em um esquema de sobreposição, operando tal como uma linha de montagem, com cada estágio sendo responsável por uma microtarefa antes de enviar o resultado para o próximo estágio. Estas máquinas têm custo elevado e atrativos para o processamento de grandes vetores, com potencialidades moderadas para acelerar as simulações da dinâmica de SEE, usando-se as metodologias tradicionais, devido ao elevado grau da esparsidade e a falta de uma estrutura padrão das matrizes envolvidas [8].

Os “array processors”, também conhecidos como máquinas SIMD (“Single Instruction stream and Multiple Data Streams”), são máquinas constituídas por uma unidade central de controle, diversos elementos de processamento (EPs) com memória individual, e uma rede de interconexão. Os EPs são sincronizados para realizarem a mesma função ao mesmo tempo, sendo necessário se estabelecer um mecanismo

adequado de fluxo de dados entre os EPs.

Os sistemas multiprocessadores, também conhecidos como máquinas MIMD (“Multiple Instruction streams and Multiple Data Streams”), constituem-se de diversos processadores que operam independentemente, podendo compartilhar memória, capacidade de entrada/saída e equipamentos periféricos. A diferença fundamental entre um “array processor” e um sistema multiprocessador é que os EPs, em um “array processor”, operam sincronamente, e um processador de um sistema multiprocessador pode operar assincronamente, pois tem unidade de controle própria. Os sistemas multiprocessadores são ainda classificados em três tipos : memória compartilhada (“shared memory”), memória distribuída (“distributed memory”) e híbridos. Nos sistemas de memória compartilhada, todos os processadores têm acesso a uma memória comum que representa também o meio de comunicação de dados entre os processadores. O desempenho em aplicações específicas é fortemente dependente das características dessa memória e da intensidade da concorrência de acesso dos processadores, exigidas por um dado algoritmo. Nos sistemas de memória distribuída não existe memória comum e a comunicação de dados, entre os processadores, é feita por “message-passing” através de uma rede de interconexão, cuja configuração define o tipo de arquitetura. O volume, frequência e a velocidade dessas transmissões de dados, têm influência significativa no desempenho global. Outro aspecto importante no desempenho de sistemas multiprocessadores é o balanceamento da carga entre os processadores.

Atualmente estão comercialmente disponíveis máquinas paralelas constituídas de vários processadores de baixo custo e que tem capacidade de processamento comparável, até mesmo, a um supercomputador [7]. No contexto dos Sistemas de Energia Elétrica, estas máquinas podem ser usadas como estações de trabalho para estudos interativos de planejamento ou ainda incorporadas aos Centros de Supervisão e Controle em tempo real (CSC), como elementos de cálculo de alto desempenho visando, por exemplo, a avaliação e correção da segurança dinâmica. Entretanto, a difusão dessas máquinas entre as empresas de Energia Elétrica ainda depende da superação de alguns aspectos, tais como:

- *Novos Algoritmos e Programas* : Para uma utilização eficiente dessas máquinas é necessário adaptar os algoritmos usados em máquinas seqüenciais ou, até mesmo, desenvolver novos algoritmos e programas. Isso exige estudos cuidadosos de revalidação dos programas de aplicação, em relação a sua escala de

aplicabilidade e precisão dos resultados obtidos;

- *Diversidade de Arquiteturas* : Existe um elevado número de arquiteturas paralelas entre as máquinas comercialmente disponíveis e outras que estão em desenvolvimento, dificultando a elaboração de “software” de uso abrangente. Além disso, a evolução tecnológica na área tem sido extremamente rápida levando os equipamentos a obsolescência poucos anos após o seu desenvolvimento. Este aspecto faz da tomada de decisão pelo investimento em um determinado equipamento, um ato de elevado risco. Isto pode ser atribuído ao fato de que essa tecnologia é relativamente recente e ainda sem padrões bem estabelecidos;
- *A Carência de “Software”* : A maioria das máquinas ainda têm sistemas de “software” básico bastante primitivos, no sentido de que o programador deve se envolver com detalhes do sistema de “hardware”. Isso tem implicações na portabilidade dos aplicativos, aspecto fundamental para a larga aplicação industrial de uma metodologia.

Por outro lado, há dois aspectos que tornam inevitável o interesse de pesquisadores e engenheiros de SEE pela computação paralela [2]: a potencialidade de alcançar grande desempenho computacional a custo relativamente baixo e à necessidade de resposta rápida no caso de aplicações em tempo real.

### 1.1.1 Aplicabilidade em Análise de Redes Elétricas

A análise de redes elétricas está, em geral, associada a estudos de planejamento da expansão, planejamento da operação, ou a operação em tempo real. No planejamento da expansão do sistema são comuns horizontes de estudos de 20 a 40 anos e, necessariamente, para diversos cenários; demandando enorme esforço computacional pela dimensão dos problemas e pelo elevado número de alternativas a serem avaliadas. Por outro lado, na operação em tempo real, vencer o curto espaço de tempo para a tomada e execução de decisões é um dos principais desafios dos instrumentos de análise de redes elétricas. Especificamente, destaca-se que instrumentos, tais como, *Análise de Segurança e Fluxo de Carga Ótimo* [9] têm sido desenvolvidos sem a incorporação dos aspectos dinâmicos dos SEE.

Várias metodologias de análise de redes elétricas tem recebido a atenção de pesquisadores no sentido de sua utilização em computadores paralelos [2]. Alguns exemplos

são: Fluxo de Carga [10, 11]; Estabilidade Transitória [8, 12, 13, 14, 15, 16, 17, 18]; Transitórios Eletromagnéticos [19]; a determinação de autovalores e autovetores no âmbito da análise dinâmica para pequenas perturbações [20], etc.

A estabilidade transitória tem apresentado a maior demanda, talvez pela expectativa do desenvolvimento de meios que viabilizem a sua incorporação no conjunto dos instrumentos de monitoração e controle da operação em tempo real.

## 1.2 Simulação da Dinâmica de SEE em Ambiente Paralelo

As equações que descrevem a dinâmica de um sistema de energia elétrica constituem um conjunto de equações diferenciais ordinárias não-lineares, associadas aos rotores das máquinas síncronas e seus controladores, e um conjunto de equações algébricas não-lineares, associadas a rede de transmissão, estatores das máquinas síncronas e cargas representadas por modelos estáticos. Essas equações podem ser expressas na forma seguinte [1]:

$$\dot{x} = f(x, z) \quad (1.1)$$

$$0 = g(x, z) \quad (1.2)$$

onde  $f$  é uma função vetorial que define as equações diferenciais;  $g$  é uma função vetorial que define as equações algébricas;  $x$  é o vetor das variáveis de estado; e  $z$  é o vetor de variáveis das equações algébricas.

Tradicionalmente, os algoritmos seqüenciais usam um dentre os dois esquemas básicos de resolução das equações (1.1) e (1.2): *Alternado* e *Simultâneo*. O esquema *Alternado* consiste basicamente em transformar as equações diferenciais (1.1) em equações algébricas a diferenças, através de um método de integração, em geral implícito, e resolvê-las alternadamente com as equações originalmente algébricas (1.2). No esquema *Simultâneo* as equações (1.1) são discretizadas por um método de integração numérica implícito e reunidas as equações (1.2), formando um único sistema de equações que é resolvido, para cada passo de integração, por um método do tipo Newton. A fim de diminuir o tempo gasto em fatorações LU, tem sido comum manter os mesmos elementos do Jacobiano por várias iterações e até mesmo passos de

integração. Essa versão do método é denominada “Very Dishonest Newton” (VDHN) [15].

A efetiva utilização de computadores paralelos, e em especial na SDSEE, pressupõe um adequado mapeamento do algoritmo de solução entre os processadores. Isso significa decompor o problema em subtarefas que possam ser processadas simultaneamente. Três linhas básicas de investigação são identificadas na literatura:

- **Paralelização no Espaço** : baseia-se na utilização de esquemas tradicionais de solução do problema com a decomposição do espaço de variáveis em subespaços, sendo o processo de paralelização mais natural [8, 12, 13, 17, 18, 21, 22];
- **Relaxação da Forma de Onda (RFO)** : consiste essencialmente em decompor o sistema de equações diferenciais e algébricas em subsistemas fracamente acoplados entre si, e realizar a solução de cada subsistema independentemente por um número determinado de passos de integração, denominados de janela de integração, obtendo-se uma primeira estimativa da forma de onda. Os resultados são, então, intercambiados entre os subsistemas, repetindo-se o processo de integração numérica de cada janela até a convergência final das formas de onda. Concluída a solução para uma janela, passa-se à janela seguinte [3, 14, 23];
- **Paralelização no Espaço e Tempo (PET)** : Visa explorar simultaneamente o paralelismo existente em cada passo de integração (paralelização no espaço) e entre os passos de integração (paralelização no tempo), resolvendo-se vários passos de integração simultaneamente [15, 24, 25].

Os principais apelos da RFO são: a possibilidade de resolver um sistema de grande porte através da solução de subsistemas de pequeno porte; a utilização de passos de integração diferentes em diferentes subsistemas; e o fato de evitar a solução, por métodos diretos, de sistemas algébricos lineares de porte muito grande. Contudo, a efetividade do método depende fortemente da fase de decomposição que representa atualmente um problema crucial desta metodologia. A idéia inicial de se explorar o paralelismo existente entre os passos de integração numérica, em SEE, está descrita em [24] e teve continuidade em [15], com a proposição de um algoritmo denominado de Gauss-Jacobi-Block-Newton (GJBN). Este algoritmo explora o PET, porém, os resultados de testes mostraram uma rápida saturação da curva de “speed up” com

o crescimento do número de processadores [17]. No capítulo 3 desta tese apresenta-se uma análise de diversos trabalhos da literatura relacionados com a utilização de técnicas de processamento paralelo na SDSEE.

No presente trabalho desenvolvem-se três algoritmos objetivando viabilizar uma efetiva utilização de computadores paralelos na SDSEE. Nos dois primeiros investiga-se a paralelização no espaço e no terceiro a paralelização no tempo. Uma das principais novidades é a utilização de métodos iterativos do tipo Gradiente Conjugado (GC) na resolução dos sistemas lineares simétricos e assimétricos que surgem no âmbito das soluções propostas.

### 1.3 Estrutura do Trabalho

No capítulo 2 são discutidos os aspectos gerais da formulação matemática do problema da SDSEE e os esquemas básicos de solução numérica seqüencial normalmente utilizados. São abordados os aspectos gerais relativos aos esquemas *Alternado e Simultâneo*, associados com as formas de integração numérica *explícitas e implícitas*.

No capítulo 3 apresenta-se uma revisão da literatura relacionada com a aplicação de técnicas de processamento paralelo ao problema da SDSEE. São analisados os seguintes temas : paralelização no espaço; paralelização no tempo; paralelização simultânea no espaço e tempo; métodos do tipo Newton relaxado; o método de relaxação da forma de onda; a sinergia do processamento paralelo-vetorial; e a aplicação de métodos iterativos do tipo Gradiente Conjugado.

No capítulo 4 descrevem-se dois algoritmos paralelos desenvolvidos tendo como base o esquema *Alternado Entrelaçado Implícito* [1]. No primeiro, denominado AEIGC, as equações da rede elétrica são ordenadas tal que a matriz admitância nodal complexa assuma a forma bloco diagonal com blocos nas bordas (“Block Bordered Diagonal Form” - BBDF [26, 27, 28]) e resolvidas, para cada iteração nas variáveis de estado, como se fossem lineares, através de um algoritmo que combina o método iterativo Gradiente Conjugado e o método direto que faz uso da fatoração LU de matrizes esparsas. No segundo algoritmo, denominado AEIGCRD, as equações da rede elétrica também são resolvidas como se fossem lineares, para cada iteração nas variáveis de estado, contudo, emprega-se exclusivamente o método Gradiente Conjugado pré-condicionado na resolução dessas equações.

No capítulo 5 decreve-se um algoritmo paralelo desenvolvido visando explorar o

paralelismo no tempo, através da resolução simultânea de vários passos de integração. O algoritmo baseia-se no esquema *Simultâneo Implícito* e emprega: o método de Newton na resolução das equações algébricas não-lineares; e os métodos iterativos Gradiente Biconjugado (BCG) e Bi-CGSTAB para resolver o sistema linear Jacobiano. São também desenvolvidos métodos de pré-condicionamento e de inicialização para as variáveis de cada janela de integração.

No capítulo 6 descreve-se a aplicação e a avaliação de desempenho dos algoritmos desenvolvidos nos capítulos 4 (AEIGC e AEIGCRD) e 5 (SIMPT), em simulações seqüenciais e paralelas. Nestas simulações foram utilizados os computadores paralelos NCP I e iPSC/860, do Laboratório de Computação Paralela da COPPE/UFRJ, e os sistemas elétricos IEEE 118 barras e uma configuração do sistema elétrico interligado das regiões sul e sudeste do Brasil com 616 barras.

No capítulo 7 são apresentadas as conclusões sobre as investigações e algoritmos desenvolvidos neste trabalho e algumas sugestões de estudos futuros.

Finalmente, são apresentados os apêndices A, B, C, D e E, onde descrevem-se, respectivamente pela ordem : a modelagem dos elementos do sistema; os esquemas *Alternado Entrelaçado Implícito* (AEI) e *Simultâneo Implícito* (SI) de resolução das equações diferenciais e algébrica (EDAs) que descrevem a dinâmica de SEE; a modelagem detalhada para o esquema *Simultâneo Implícito Paralelo no Tempo* (SIMPT); o método Gradiente Conjugado pré-condicionado; e a utilização da série de Taylor na inicialização das variáveis da janela de integração.



# Capítulo 2

## Modelagem Matemática Básica

Neste capítulo trata-se da modelagem matemática básica para o estudo de algoritmos empregados na resolução dos sistemas de equações diferenciais ordinárias e algébricas (EDAs), ambos não-lineares, que representam a dinâmica de Sistemas de Energia Elétrica (SEE). O objetivo principal é se estabelecer um referencial, em termos de modelagem e nomenclatura, para os demais capítulos que constituem o presente documento. Especificamente, são abordados alguns aspectos gerais da formulação matemática e os esquemas básicos tradicionalmente empregados em algoritmos seqüenciais de Simulação da Dinâmica de Sistemas de Energia Elétrica (SDSEE) em computadores digitais.

### 2.1 Aspectos Gerais

A operação dos SEE é executada através de cinco malhas de controle básicas: o controle de tensão (sistema de excitação ou RAT), o controle de velocidade (governador ou RV.), o controle de estabilização suplementar (PSS), o controle automático de geração (CAG) (ou controle suplementar, responsável pelo controle da frequência e fluxos de intercâmbio entre áreas do sistema) e o despacho de carga ou econômico, em geral não automático, porém, servindo de base para fatores de distribuição de carga no CAG. A figura 2.1 apresenta uma visão global destas malhas a partir do detalhamento para um gerador particular conectado ao SEE [29]. As malhas de controle suplementar e despacho econômico são de resposta bastante lenta e, normalmente, não consideradas em estudos de estabilidade transitória.

A formulação matemática do problema da SDSEE conta, principalmente, com as seguintes características [1]:

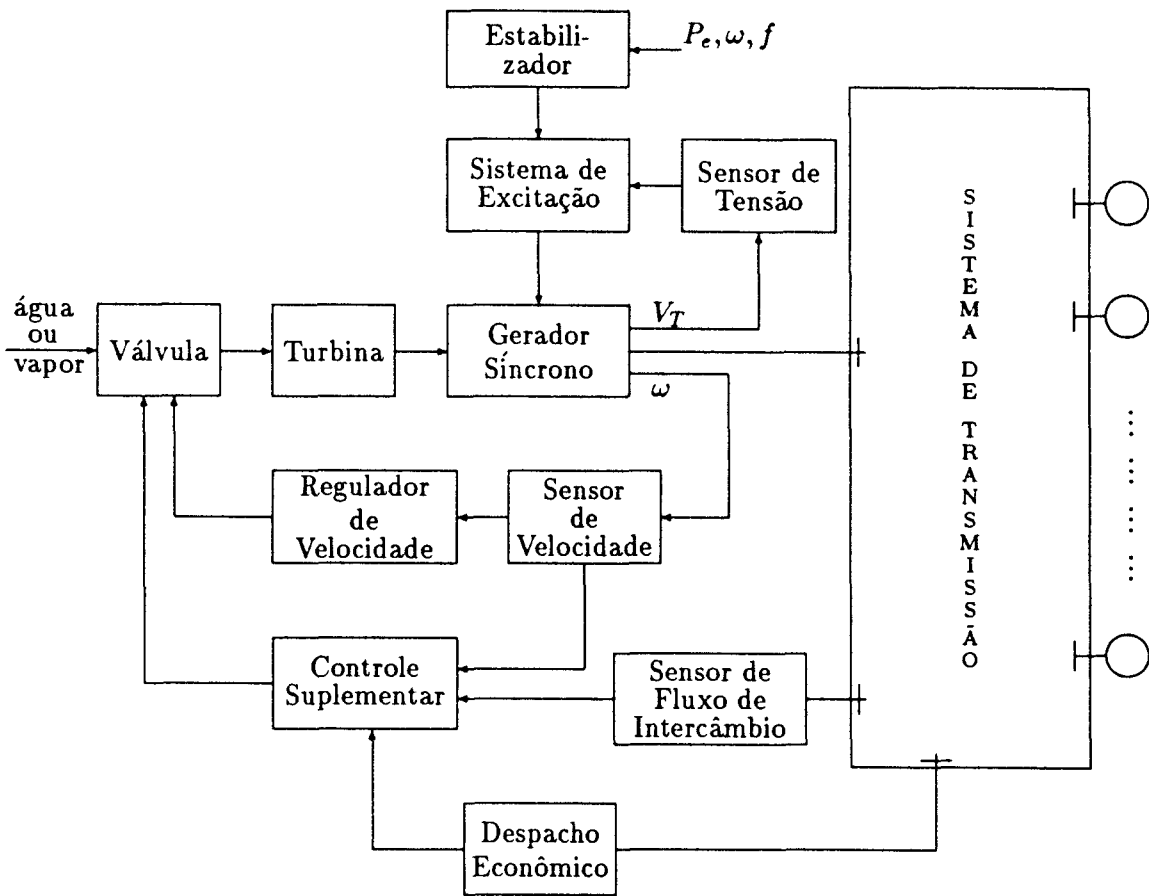


Figura 2.1: Principais malhas de controle de um SEE

- Existência de uma grande diversidade de modelos com diferentes níveis de complexidade;
- É desejável se encontrar modelos aceitáveis para os vários tipos de estudos com o mínimo grau de complexidade;
- É um problema algébrico-diferencial de valor inicial;
- Existência de grandes descontinuidades nas equações algébricas que descrevem a rede elétrica, resultantes de defeitos e chaveamentos em linhas de transmissão;
- Existência de pequenas descontinuidades que ocorrem nas equações diferenciais, devido aos limites dos equipamentos de controle automático.

Genericamente, as equações que descrevem o comportamento dinâmico de um SEE tem uma formulação bem definida e são válidas para uma grande escala de modelos. Estas equações constituem um conjunto de equações diferenciais ordinárias não-lineares, associadas aos rotores das máquinas síncronas e seus controladores, e um conjunto de equações algébricas não-lineares, associadas a rede de transmissão, estatores das máquinas síncronas (geradores) e cargas representadas por modelos estáticos. Elas podem ser expressas na forma seguinte [1]:

$$\dot{x} = f(x, z) \quad (2.1)$$

$$0 = g(x, z) \quad (2.2)$$

onde:

$f$  - função vetorial que define as equações diferenciais;

$g$  - função vetorial que define as equações algébricas;

$x$  - vetor das variáveis de estado;

$z$  - vetor de variáveis das equações algébricas.

A figura 2.2 apresenta a estrutura típica do modelo matemático global para estudos de estabilidade transitória, a partir do detalhamento das equações associadas a um único gerador síncrono [1, 29].

As variáveis que aparecem na figura 2.2 têm o seguinte significado:

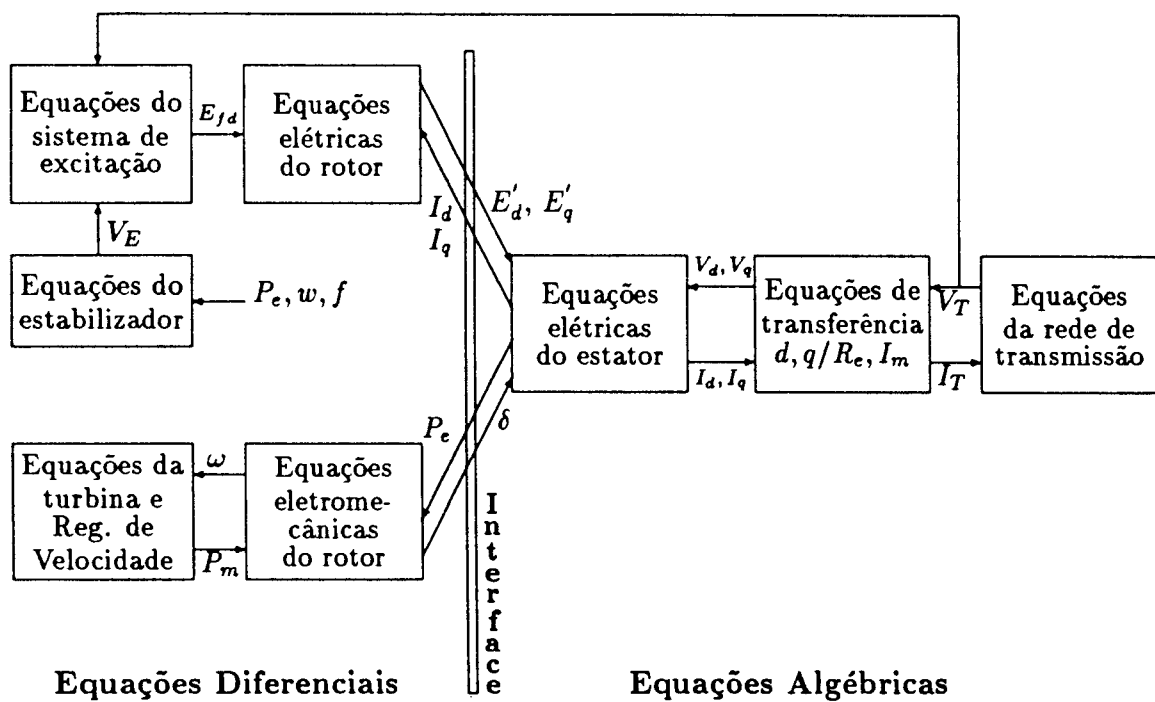


Figura 2.2: Estrutura usual das equações envolvidas no modelo matemático para estudos de estabilidade transitória

$P_m$  - potência mecânica aplicada ao eixo do gerador;

$\omega$  - velocidade angular do eixo do gerador;

$V_E$  - tensão na saída do estabilizador;

$P_e$  - potência elétrica de entreferro;

$\delta$  - posição angular do rotor em relação a uma referência síncrona;

$E_{fd}$  - tensão na saída do sistema de excitação aplicada ao enrolamento de campo do gerador;

$V_d, V_q$  - componentes de eixo direto e quadratura da tensão terminal do gerador;

$E'_d, E'_q$  - componentes de eixo direto e quadratura da tensão transitória interna do gerador;

$I_d, I_q$  - componentes de eixo direto e quadratura da corrente de estator do gerador;

$I_T$  - corrente complexa de estator do gerador;

$V_T$  - tensão complexa terminal do gerador.

As variáveis que aparecem tanto nas equações diferenciais como nas equações algébricas, são denominadas variáveis de interface e assim definidas:

$u$  - subconjunto do vetor de variáveis algébricas que aparecem nas equações diferenciais ( $I_d, I_q, P_e, V$ );

$E$  - subconjunto do vetor de variáveis de estado que aparecem nas equações algébricas ( $E'_d, E'_q, \delta$ ).

A simulação da dinâmica de sistemas de energia elétrica, em computadores digitais, consiste em resolver numericamente as (EDAs) (2.1) e (2.2) para um determinado período de tempo, a partir de condições iniciais correspondentes a um ponto de operação em regime permanente.

## 2.2 Esquemas Básicos de Solução

Historicamente diversos esquemas tem sido utilizados na resolução das EDAs (2.1) e (2.2) ([1, 29, 30, 31, 32, 33]; e outras). A caracterização destes esquemas baseia-se no método de integração numérica aplicado às equações diferenciais, no método de solução das equações algébricas e no modo de interação entre os sistemas algébrico e diferencial ao longo do processo de resolução. Métodos de integração numérica explícitos definem esquemas explícitos e métodos de integração numérica implícitos definem esquemas implícitos. Dois esquemas básicos, *alternado* e *simultâneo*, têm sido suficientes para classificar os modos de interação utilizados entre as soluções dos sistemas algébrico e diferencial ([1, 32] e outros). O esquema *alternado* consiste em resolver separadamente, em cada passo de integração, os sistemas diferencial e algébrico, alternando-se as respectivas soluções. No esquema *simultâneo*, as equações diferenciais são transformadas em equações algébricas a diferenças, através da aplicação de um determinado método de integração numérica, e resolvidas simultaneamente com as equações algébricas como um único sistema de equações. Este sistema de equações algébricas e algebrizadas é resolvido, em geral, pelo método de Newton ou um método do tipo Newton. A combinação desses esquemas básicos, *alternado* e *simultâneo*, com os métodos de integração numérica *explícitos* e *implícitos*, define quatro esquemas de solução ilustrados pela tabela 2.1 [32].

Tabela 2.1: Esquemas de resolução das EDAs para a SDSEE

Métodos de integração	Esquemas básicos	
	Alternado	Simultâneo
Explícito (E)	AE	SE
Implícito (I)	AI	SI

As fórmulas implícitas de integração numérica tem apresentado melhores resultados [1, 30, 31, 32], sendo que a maioria dos trabalhos empregam o método trapezoidal implícito pelas suas características de estabilidade e precisão. Este é também o método de integração adotado nos algoritmos desenvolvidos neste trabalho e descritos

nos capítulos 4 e 5. Nos dois próximos sub-ítem, realiza-se uma breve abordagem dos esquemas *Alternado Implícito* (AI) e *Simultâneo Implícito* (SI).

### 2.2.1 Esquema Alternado Implícito

Para facilitar a descrição deste esquema e suas variantes, as equações (2.1) e (2.2) são reescritas na forma seguinte [1]:

$$\dot{x} = Ax + Bu \quad (2.3)$$

$$I(E, V) = YV \quad (2.4)$$

$$u = h(E, V) \quad (2.5)$$

A equação (2.3) é apenas uma forma mais conveniente de se expressar a equação (2.1); onde  $A$  é uma matriz quadrada, esparsa e bloco-diagonal;  $B$  é uma matriz retangular esparsa e formada por blocos;  $x$  é o vetor de variáveis de estado associadas as máquinas elétricas, máquinas primárias e a dinâmica de seus controladores; e  $u$  é um vetor de variáveis algébricas que aparecem nas equações diferenciais. Desprezando-se os efeitos da saturação magnética, as matrizes  $A$  e  $B$  são constantes para os modelos mais comumente empregados. A equação (2.4) representa a descrição nodal da rede elétrica, onde  $Y$  é a matriz admitância nodal, esparsa, complexa e simétrica;  $I$  é o vetor de injeções de correntes nodais complexas associadas as cargas e aos estatores das máquinas síncronas;  $V$  é o vetor de tensões nodais complexas; e  $E$  é um subvetor de  $x$ , constituído das variáveis de estado necessárias para o cálculo das injeções de correntes dos estatores das máquinas síncronas. Para uma carga, a injeção de corrente é função da tensão na barra; e para um gerador, ela é a corrente de estator, relacionada a referência síncrona (complexa) e função das tensões interna e terminal da máquina. A equação (2.5) serve exclusivamente como interface para o cálculo dos componentes do vetor  $u$  e  $h$  é uma função vetorial não-linear.

O esquema AI consiste basicamente em transformar as equações diferenciais em equações algébricas a diferenças, através de um método de integração implícito, e resolvê-las iterativamente e alternadamente com as equações originalmente algébricas. Desse modo, o comportamento dinâmico de um SEE, supostamente constituído de  $m$  geradores e  $n$  barras, no intervalo de tempo  $t - \Delta t$  a  $t$ , pode ser genericamente representado pelas seguintes equações:

$$\begin{cases} x_1(t) = H_1(x_1(t), x_1(t - \Delta t), u_1(t), u_1(t - \Delta t)) \\ x_2(t) = H_2(x_2(t), x_2(t - \Delta t), u_2(t), u_2(t - \Delta t)) \\ \vdots \\ x_m(t) = H_m(x_m(t), x_m(t - \Delta t), u_m(t), u_m(t - \Delta t)) \end{cases} \quad (2.6)$$

$$\begin{bmatrix} I_1(E_1(t), V_1(t)) \\ I_2(E_2(t), V_2(t)) \\ \vdots \\ I_n(E_n(t), V_n(t)) \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1(t) \\ V_2(t) \\ \vdots \\ V_n(t) \end{bmatrix} \quad (2.7)$$

$$\begin{cases} u_1(t) = h_1(E_1(t), V_1(t)) \\ u_2(t) = h_2(E_2(t), V_2(t)) \\ \vdots \\ u_m(t) = h_m(E_m(t), V_m(t)) \end{cases} \quad (2.8)$$

onde:

$\Delta t$  - passo de integração;

$x_i = [\omega_i, \delta_i, E'_{qi}, E'_{di}, E'_{fdi}]^T$  - vetor de variáveis de estado do  $i$ -ésimo gerador;

$E_i = [E'_{qi}, E'_{di}, \delta_i]^T$  - subvetor de variáveis de interface do  $i$ -ésimo gerador;

$u_i = [I_{di}, I_{qi}, P_{ei}]^T$  - vetor de variáveis das equações de interface do  $i$ -ésimo gerador.

Definindo-se:

$H = [H_1, H_2, \dots, H_m]^T$  - vetor de funções não-lineares das equações algébricas a diferenças;



$x = [x_1, x_2, \dots, x_m]^T$  - vetor de variáveis de estado;

$u = [u_1, u_2, \dots, u_m]^T$  - vetor de variáveis das equações de interface;

$I = [I_1, I_2, \dots, I_n]^T$  - vetor de injeções de correntes nodais complexas;

$V = [V_1, V_2, \dots, V_n]^T$  - vetor de tensões nodais complexas;

$h = [h_1, h_2, \dots, h_m]^T$  - vetor de funções não-lineares de interface;

pode-se escrever as equações (2.6), (2.7) e (2.8) na seguinte forma sintética:

$$\begin{cases} x(t) = H(x(t), x(t - \Delta t), u(t), u(t - \Delta t)) \\ I(E(t), V(t)) = YV(t) \\ u(t) = h(E(t), V(t)) \end{cases} \quad (2.9)$$

A partir das equações (2.9) e definindo-se  $k$  como o contador de iterações e  $\epsilon$  como uma tolerância de convergência para as variáveis de estado, o esquema AI segue basicamente o seguinte algoritmo:

### Algoritmo AI

#### Inicialização

Para  $t = 1, 2, \dots, T$

Calcule  $u^*(t)$ , por extrapolação, e faça  $k \leftarrow 0$

Calcule  $x^0(t) = H[x^*(t), u^*(t), x(t-1), u(t-1)]$  (\*- valor extrapolado)

Enquanto  $\|\Delta x^k(t)\|_2^2 > \epsilon$ , faça

Resolva  $I(E(t), V(t)) = YV(t)$

Calcule  $u^k(t) = h(E(t), V(t))$

Calcule  $x^{k+1}(t) = H[x^k(t), u^k(t), x(t-1), u(t-1)]$

Calcule  $\Delta x^{k+1}(t) = |x^{k+1}(t) - x^k(t)|$

faça  $k \leftarrow k + 1$

Fim-Enquanto

Fim-Para

O processo de cálculo de  $x(t)$  é iterativo devido a adoção de um método de integração implícito. As injeções de correntes são, em geral, funções não-lineares de  $V(t)$  e, portanto, a equação da rede elétrica também deve ser resolvida iterativamente. Os valores extrapolados de  $u^*(t)$ , para a primeira iteração, representam, em geral, melhores condições iniciais que os valores do instante  $t - \Delta t$ . Somente nos instantes de descontinuidades, em que não é possível realizar a extrapolação, é que são utilizados, como condição inicial, os valores do instante de tempo  $t - \Delta t$ .

O algoritmo AI representa a forma clássica do esquema alternado implícito de resolução das EDAs que descrevem a dinâmica de SEE. Uma alternativa a este esquema é obtida relaxando-se os requisitos de convergência na solução das equações da rede elétrica, por exemplo, realizando-se somente uma iteração. Neste caso os processos iterativos diferencial e algébrico são entrelaçados, ora realizando-se uma iteração em um, ora em outro [1, 34]. O teste de convergência é feito nas variáveis de estado, que determinam a precisão da solução final. Este esquema é denominado *Alternado Entrelaçado Implícito* (AEI) e tem apresentado desempenho superior ao AI clássico [34]. No capítulo 4 são desenvolvidos dois novos algoritmos baseados no esquema AEI, visando uma eficiente utilização de computadores paralelos na SDSEE.

### 2.2.2 Esquema Simultâneo Implícito

No esquema *Simultâneo Implícito* (SI), as equações diferenciais são transformadas em equações algébricas a diferenças através de um método implícito de integração numérica. Estas equações são reunidas com as equações originalmente algébricas, para constituírem um único sistemas de equações algébricas não-lineares, representado matematicamente pelas seguintes equações [1, 30, 31, 32, 33]:

$$\begin{cases} F = x(t) - x(t - \Delta t) - [f(x(t), z(t)) + f(x(t - \Delta t), z(t - \Delta t))] \frac{\Delta t}{2} = 0 \\ G = g(x(t), z(t)) = 0 \end{cases} \quad (2.10)$$

O sistema de equações (2.10) descreve o comportamento dinâmico do SEE no intervalo de tempo  $t - \Delta t$  a  $t$ , podendo ser reescrito na seguinte forma compacta.

$$H(y(t)) = 0, \quad (2.11)$$

onde  $y = [x, z]^T$  e  $H = [F, G]^T$ .

A resolução de (2.11), para  $y(t)$ , é comumente realizada pelo método de Newton [1, 29]. Definindo-se  $y^0$  como condição inicial, a constante  $\epsilon$  como a tolerância para a convergência e  $k$  como o contador de iterações, o método de Newton aplicado as equações (2.11), e conseqüentemente a solução para um passo de integração, segue o seguinte algoritmo:

### Algoritmo de Newton

*Faça  $k = 0$  e especifique  $y^0$*

*Enquanto  $\|H(y^k)\|_2^2 > \epsilon$ , faça:*

*Calcule  $H(y^k)$*

*Resolva:*

$$H(y^k) + \frac{\partial H}{\partial y} \Big|_{y=y^k} \Delta y^{k+1} = 0 \quad (2.12)$$

*Calcule:*

$$y^{k+1} = y^k + \Delta y^{k+1} \quad (2.13)$$

*Faça  $k \leftarrow k + 1$*

*Fim-Enquanto*

A equação matricial (2.12) é linear e deve ser resolvida, em geral, várias vezes para cada passo de integração, pois os elementos do Jacobiano ( $\partial H/\partial y$ ) devem ser recalculados para cada  $y = y^k$  ( $k = 1, 2, \dots$ , até a convergência). Note que o maior esforço computacional no algoritmo de Newton e conseqüentemente para o esquema SI, concentra-se na resolução de (2.12). A matriz de coeficientes de (2.12) é, para SEE reais, muito esparsa e essa característica deve ser explorada para uma eficiente implementação computacional. A fim de minimizar o esforço dispendido na fatoração dessa matriz, muitos trabalhos ([29, 32, 35] e outros) usam a matriz Jacobiana calculado na primeira iteração, para as iterações subsequentes, no mesmo passo de integração, ou até mesmo para vários passos de integração. Desse modo a equação matricial (2.12) assume a forma seguinte:

$$H(y^k) + \frac{\partial H}{\partial y} \Big|_{y=y^k} \Delta y^{k+1} = 0 \quad (2.14)$$

Essa versão do método de Newton é denominada “Very Dishonest Newton” (VDHN).

### Resolução do Sistema Linear Jacobiano

A matriz Jacobiana da equação (2.14) ou (2.12) apresenta uma estrutura que permanece válido para uma diversidade grande de modelos dos elementos que compõe um SEE. Desse modo, o objetivo desse item é estudar esta estrutura de forma genérica, afim de destacar as características que propiciam uma eficiente implementação computacional.

Um SEE com  $m$  geradores e  $n$  barras, após a transformação das equações diferenciais em equações algébricas a diferenças, pode ser descrito, genericamente, pelas seguintes equações [35]:

$$\left\{ \begin{array}{l} F_1(x_1(t), x_1(t-1), V^e(t), V^e(t-1)) = 0 \\ F_2(x_2(t), x_2(t-1), V^e(t), V^e(t-1)) = 0 \\ \vdots \\ F_m(x_m(t), x_m(t-1), V^e(t), V^e(t-1)) = 0 \\ G'(x(t), V^e(t)) - I^e(x(t), V^e(t)) = 0 \\ Y^e \cdot V^e(t) - I^e(x(t), V^e(t)) = 0 \end{array} \right. \quad (2.15)$$

onde:

$$t-1 = t - \Delta t;$$

$x = [x_1, x_2, \dots, x_m]^T$  - vetor de variáveis de estado (geradores e controladores);

$G' = [G'_1, \dots, G'_m, G'_{m+1}, \dots, G'_n]^T$  - vetor de funções não-lineares associadas as equações algébricas (cargas, estatores e interfaces);

$F = [F_1, F_2, \dots, F_m]^T$  - vetor de funções não-lineares associadas as equações algébricas a diferenças;

$V^e = [V_1^e, V_2^e, \dots, V_n^e]^T$  - vetor de tensões nodais expandidas nas parcelas real e imaginária

$I^e = [I_1^e, I_2^e, \dots, I_n^e]^T$  - vetor de injeções de correntes nodais expandidas nas parcelas real e imaginária (gerações e cargas);

$V_i^e = [V_{mi}, V_{ri}]^T$  - vetor das parcelas imaginária e real da tensão nodal (barra  $i$ )

$I_i^e = [I_{ri}, I_{mi}]^T$  - vetor das parcelas real e imaginária da corrente nodal (barra  $i$ )

$Y_{2n,2n}^e$  - matriz admitância nodal incluindo-se as admitâncias das cargas.

Note que cada componente dos vetores  $x$ ,  $G'$  e  $F$  representa um vetor. Então, para a barra  $i$ ,  $x_i$  representa todas as variáveis de estado do gerador  $i$ ;  $F_i$  e  $G'_i$  são as funções algebrizadas e algébricas, respectivamente.

Eliminando-se  $I^e$  nas duas últimas equações do sistema (2.15), tem-se:

$$\left\{ \begin{array}{l} F_1(x_1(t), x_1(t-1), V^e(t), V^e(t-1)) = 0 \\ F_2(x_2(t), x_2(t-1), V^e(t), V^e(t-1)) = 0 \\ \vdots \\ F_m(x_m(t), x_m(t-1), V^e(t), V^e(t-1)) = 0 \\ Y^e \cdot V^e(t) - G'(x(t), V^e(t)) = G(x(t), V^e(t)) = 0 \end{array} \right. \quad (2.16)$$

As equações principais do algoritmo do método VDHN (equações (2.14) e (2.13)), para o sistema (2.16), são:

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \\ Y^e V^e - G' \end{bmatrix}_{x^k, V^{ek}} = - \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & 0 & \dots & 0 & \frac{\partial F_1}{\partial V^e} \\ 0 & \frac{\partial F_2}{\partial x_2} & \dots & 0 & \frac{\partial F_2}{\partial V^e} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \frac{\partial F_m}{\partial x_m} & \frac{\partial F_m}{\partial V^e} \\ -\frac{\partial G'}{\partial x_1} & -\frac{\partial G'}{\partial x_2} & \dots & -\frac{\partial G'}{\partial x_m} & Y^e - \frac{\partial G'}{\partial V^e} \end{bmatrix}_{x^0, V^{e0}} \begin{bmatrix} \Delta x_1^{k+1} \\ \Delta x_2^{k+1} \\ \vdots \\ \Delta x_m^{k+1} \\ \Delta V^{ek+1} \end{bmatrix} \quad (2.17)$$

$$\begin{cases} x^{k+1} = x^k + \Delta x^{k+1} \\ V^{ek+1} = V^{ek} + \Delta V^{ek+1} \end{cases} \quad (2.18)$$

Reescrevendo-se o sistema (2.17), em termos dos incrementos  $\Delta x^{k+1}$  e  $V^{ek+1}$ , tem-se:

$$\begin{aligned} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \\ Y^e V^e - G' \end{bmatrix}_{x^k, V^{ek}} + \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & 0 & \dots & 0 & \frac{\partial F_1}{\partial V^e} \\ 0 & \frac{\partial F_2}{\partial x_2} & \dots & 0 & \frac{\partial F_2}{\partial V^e} \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & \frac{\partial F_m}{\partial x_m} & \frac{\partial F_m}{\partial V^e} \\ -\frac{\partial G'}{\partial x_1} & -\frac{\partial G'}{\partial x_2} & \dots & -\frac{\partial G'}{\partial x_m} & Y^e - \frac{\partial G'}{\partial V^e} \end{bmatrix}_{x^0, V^{e0}} \begin{bmatrix} \Delta x_1^{k+1} \\ \Delta x_2^{k+1} \\ \vdots \\ \Delta x_m^{k+1} \\ V^{ek+1} \end{bmatrix} \\ = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & 0 & \dots & 0 & \frac{\partial F_1}{\partial V^e} \\ 0 & \frac{\partial F_2}{\partial x_2} & \dots & 0 & \frac{\partial F_2}{\partial V^e} \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & \frac{\partial F_m}{\partial x_m} & \frac{\partial F_m}{\partial V^e} \\ -\frac{\partial G'}{\partial x_1} & -\frac{\partial G'}{\partial x_2} & \dots & -\frac{\partial G'}{\partial x_m} & Y^e - \frac{\partial G'}{\partial V^e} \end{bmatrix}_{x^0, V^{e0}} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ V^{ek} \end{bmatrix} \end{aligned} \quad (2.19)$$

ou

$$\begin{aligned}
& \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \\ Y^e V^e - G' \end{bmatrix}_{x^k, V^{ek}} + \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & 0 & \dots & 0 & \frac{\partial F_1}{\partial V^e} \\ 0 & \frac{\partial F_2}{\partial x_2} & \dots & 0 & \frac{\partial F_2}{\partial V^e} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \frac{\partial F_m}{\partial x_m} & \frac{\partial F_m}{\partial V^e} \\ -\frac{\partial G'}{\partial x_1} & -\frac{\partial G'}{\partial x_2} & \dots & -\frac{\partial G'}{\partial x_m} & Y^e - \frac{\partial G'}{\partial V^e} \end{bmatrix}_{x^0, V^{e0}} \begin{bmatrix} \Delta x_1^{k+1} \\ \Delta x_2^{k+1} \\ \vdots \\ \Delta x_m^{k+1} \\ V^{ek+1} \end{bmatrix} \\
& = \begin{bmatrix} \frac{\partial F_1}{\partial V^e} V^{ek} \\ \frac{\partial F_2}{\partial V^e} V^{ek} \\ \vdots \\ \frac{\partial F_m}{\partial V^e} V^{ek} \\ Y^e V^{ek} - \frac{\partial G'}{\partial V^e} V^{ek} \end{bmatrix}_{x^0, V^{e0}}
\end{aligned} \tag{2.20}$$

Note que os termos  $Y^e V^{ek}$ , no primeiro e último vetor da equação (2.20), cancelam-se, eliminando a necessidade de realização desses produtos.

A fim de simplificar o desenvolvimento na seqüência deste trabalho, escreve-se a equação (2.17) na seguinte forma simbólica :

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ G \end{bmatrix}^k = - \begin{bmatrix} A_{M1} & & & B_1 \\ & A_{M2} & & B_2 \\ & & A_{M3} & B_3 \\ & & & \ddots \\ C_1 & C_2 & C_3 & \dots & Y^e + Y_L \end{bmatrix}^0 \cdot \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \vdots \\ \Delta V^e \end{bmatrix}^{k+1} \tag{2.21}$$

onde  $A_{Mi} = \frac{\partial F_i}{\partial x_i}$ ,  $B_i = \frac{\partial F_i}{\partial V^e}$  e  $C_i = \frac{\partial G}{\partial x_i}$ ,  $i = 1, 2, \dots, m$ .  $\frac{\partial G}{\partial V^e} = Y^e + Y_L$  é a matriz admitância nodal, acrescida da admitância das cargas.  $F_i$ ,  $i = 1, 2, \dots, m$  e  $G$ , representam os resíduos para as equações algebrizadas dos geradores e para as equações da rede elétrica, respectivamente.

A equação (2.21) pode ser expressa em uma forma ainda mais compacta como a seguir:

$$\begin{bmatrix} F \\ G \end{bmatrix}^k = - \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}^0 \cdot \begin{bmatrix} \Delta x \\ \Delta V^e \end{bmatrix}^{k+1} \tag{2.22}$$

A matriz de coeficientes da equação (2.22) tem a forma bloco diagonal com blocos nas bordas (BBDF) e pode ser resolvida em duas fases. Inicialmente para  $\Delta V^e$ , realizando-se uma eliminação Gaussiana por blocos, dado pela equação :

$$-(J_4 + Y_M) \cdot \Delta V^e = -\bar{J}_4 \cdot \Delta V^e = \bar{G} \quad (2.23)$$

onde  $Y_M = -J_3 J_1^{-1} J_2$  representa o somatório dos efeitos dos geradores na matriz admitância nodal e  $\bar{G} = G - J_3 J_1^{-1} F$ .

Na segunda fase, resolve-se para cada  $\Delta x_i$  ( $i = 1, 2, \dots, m$ ) a seguinte equação :

$$\Delta x_i = -A_{M_i}^{-1} \cdot (F_i + B_i \cdot \Delta V^e) \quad (2.24)$$

Note que os efeitos dos geradores na matriz admitância nodal estão restritos aos blocos diagonais  $2 \times 2$  e, portanto,  $\bar{J}_4$  manterá a mesma estrutura de  $J_4$ , ou seja, de  $Y^e$ . As equações lineares (2.23) e (2.24) são resolvidas pelo método direto baseado na fatoração LU. As matrizes  $A_{M_i}$ ,  $i = 1, 2, \dots, m$ , tem dimensão igual ao número de equações usadas para representar o  $i$ -ésimo gerador e seus controladores. A dimensão dessas matrizes varia de 2, para o modelo clássico, a cerca de 20 para modelos mais sofisticados. A inversão dessas matrizes não implica em expressivo esforço computacional. Adicionalmente, destaca-se que no método VDHN somente os elementos de  $\bar{J}_4$  são mantidos constantes por diversas iterações e até passos de integração. As matrizes  $A_{M_i}$  e  $B_i$  -  $i = 1, 2, \dots, m$  - são recalculadas a cada iteração.

## 2.3 Comentários Finais

Neste capítulo foram discutidos vários aspectos relacionados com a representação e simulação da dinâmica de Sistemas de Energia Elétrica. Considerou-se, em detalhes, os esquemas *Alternado Entrelaçado Implícito* (AEI) e *Simultâneo Implícito* (SI) de resolução das equações diferenciais e algébricas que descrevem esse comportamento. O esquema AEI é sensivelmente mais simples e oferece maiores facilidades para a incorporação e tratamento de novos e sofisticados modelos de elementos dos SEE. Além disso, ele utiliza a matriz admitância nodal ( $Y$ ) na forma compacta (complexa), cujos elementos são constante entre chaveamentos na rede elétrica de transmissão. Portanto, os fatores LU de  $Y$  somente devem ser recalculados nos instantes em que ocorrem alterações estruturais na rede elétrica.



O esquema SI é mais atrativo sob o ponto de vista da formalização matemática, já que está baseado no método de Newton. Entretanto, a matriz admitância nodal deve ser utilizada em sua forma expandida, nas parcelas real e imaginária de seus elementos, e, rigorosamente, a matriz Jacobiana e seus fatores LU devem ser recalculados a cada iteração. Por outro lado, um significativo ganho é obtido mantendo-se o Jacobiano constante por várias iterações, ou até mesmo vários passos de integração. Essa metodologia é conhecida como “Very Dishonest Newton” (VDHN).

No próximo capítulo serão abordadas algumas metodologias apresentadas na literatura técnica, relacionados com a aplicação de técnicas de processamento paralelo ao problema da SDSEE.

## Capítulo 3

# Simulação da Dinâmica de Sistemas de Energia Elétrica em Ambiente Paralelo - Revisão da Literatura

O objetivo deste capítulo é apresentar uma revisão de publicações recentes e relacionadas com a aplicação de técnicas de processamento paralelo à Simulação da Dinâmica de Sistemas de Energia Elétrica (SDSEE). Os principais temas abordados são :

- Paralelização no espaço;
- Paralelização no tempo;
- Paralelização simultânea no espaço e tempo;
- Aplicação de métodos do tipo Newton Relaxado;
- Relaxação da Forma de Onda;
- Sinergia do processamento Paralelo-Vetorial;
- Aplicação de métodos iterativos do tipo Gradiente Conjugado.

Procurar-se-á destacar as potencialidades e limitações de cada metodologia, de modo a facilitar a compreensão da linha de investigação do presente trabalho.

## 3.1 Paralelização no Espaço

Caracteriza-se pela utilização de esquemas tradicionais de resolução das equações que descrevem a dinâmica do sistema, com a decomposição do espaço de variáveis em subespaços. Este é considerado o processo mais natural de paralelização. Três metodologias encontradas na literatura serão analisadas com algum detalhe, a saber:

- O método VDHN paralelo;
- O método “Newton-W matrix”;
- O simulador digital em tempo real.

### 3.1.1 O Método VDHN Paralelo

Caracteriza-se pela paralelização do algoritmo VDHN, simplesmente identificando tarefas que possam ser realizadas simultaneamente e distribuindo-as entres os processadores [21]. A essência do método VDHN é o cálculo dos acréscimos  $\Delta x$  e  $\Delta V^e$ , a cada iteração, de acordo com as seguintes equações, obtidas a partir das equações (2.23) e (2.24):

$$\Delta V^{ek+1}(t) = -\bar{J}_4^{-1} \cdot \bar{G}(x^k(t), x^k(t-1), V^{ek}(t), V^{ek}(t-1)) \quad (3.1)$$

$$\Delta x^{k+1}(t) = -(A_M^k)^{-1} \cdot (F^k(t) + B^k(t) \cdot \Delta V^{ek+1}(t)) \quad (3.2)$$

onde  $k$  é o contador de iterações,  $t$  é o instante de tempo correspondente ao  $t$ -ésimo passo de integração e  $t-1$  é o passo de integração imediatamente anterior ao passo  $t$ .

O cálculo dos elementos de  $\bar{G}, \bar{J}_4, F, B, A_M$  e a solução da equação (3.2), foram realizados em paralelo. Por outro lado, a resolução da equação (3.1) foi feita em um único processador, através da fatoração LU de  $\bar{J}_4$  e das substituições direta e inversa. Segundo os autores desta proposta de paralelização [17], as substituições direta e inversa representam cerca de 7 % do tempo total de cada iteração no algoritmo seqüencial do método VDHN. Entretanto, pela lei de Amdahl [36], isto impõe um limite máximo de ganho de velocidade igual a 14,3 para o método VDHN paralelo.

O algoritmo foi implementado nos computadores iPSC/2 (hipercúbico com memória distribuída) e Alliant FX/8 (com memória compartilhada), e testado em simulações da dinâmica dos sistemas IEEE 118 e US. Midwestern com 662 barras. Os

resultados mostram “speed ups” levemente superiores para o iPSC/2, porém, com forte saturação conforme cresce o número de processadores. O máximo “speed up” obtido foi 5,61 para 32 processadores (eficiência = 17,5 %), isto é, muito inferior ao limite imposto pela lei de Amdahl.

### 3.1.2 O Método “Newton-W Matrix”

O objetivo principal desta proposta é substituir a solução seqüencial da equação (3.1), no método VDHN, por uma versão paralelizada do método dos fatores  $W$  [17] (“Sparse matrix inverse factors” [37]). Para uma matriz fatorada na forma  $LDL^T$ , as matrizes  $W$  são assim definidas :

$$W = L^{-1} = (L_1 \dots L_n)^{-1} = L_n^{-1} \dots L_1^{-1} = W_n \dots W_1 \quad (3.3)$$

Adicionalmente, esta metodologia requer o cálculo dos fatores inversos após a fatoração e provoca o aparecimento de novos “fill-ins”, cujo volume depende do esquema de ordenação utilizado.

As matrizes elementares  $W_i$  ( $i = 1, 2, \dots, n$ ) são agrupáveis em subgrupos, denominados partições, que podem ser calculados concorrentemente, como por exemplo  $W_a$  e  $W_b$ . Assim e omitindo-se a representação da dependência de  $x$ ,  $V^e$ , e de  $t$ , a equação (3.1) pode ser reescrita na seguinte forma [17] :

$$\Delta V^{ek+1} = -W_a^T W_b^T D^{-1} W_b W_a \bar{G} \quad (3.4)$$

O paralelismo na equação (3.4), é explorado dentro de cada partição, ou seja, nas matrizes  $W_a$  e  $W_b$ . Por outro lado, note que a equação (3.4) deve ser processada da direita para a esquerda em 5 passos seqüenciais. Além disso, a fatoração LDU da matriz Jacobiano ainda permanece sendo realizada seqüencialmente. A metodologia foi testada no computador Symmetry, de memória compartilhada, em simulações da dinâmica dos sistemas IEEE 118 e US. Midwestern com 662 barras. Os resultados mostram que, ao contrário da expectativa inicial, o método “Newton-W matrix” teve desempenho inferior ao VDHN paralelo, com um “slowdown” de 10 a 30 % dependendo das partições de  $W$  escolhidas e da esparsidade do sistema. É importante notar que o método “Newton-W matrix” não foi implementado no iPSC/2, onde o custo de comunicação deverá implicar em desempenho ainda inferior. Os autores atribuem esse insucesso a dois fatores principais, a saber :

- A fatoração LDU que continuou sendo feita seqüencialmente e aos excessivos cálculos do método dos fatores  $W$ . É citado que foi usado o esquema de ordenação “Tinney 2” tendo sido gerados uma enorme quantidade de “fill-ins” adicionais na formação das partições de  $W$ ;
- As dificuldades para equacionar o desbalanceamento da carga entre os processadores no método dos fatores  $W$ .

Por outro lado, em [38] é descrita uma versão do método dos fatores  $W$ , cuja implementação no computador paralelo PP/286, desenvolvido pelo CPQd/Telebrás, apresentou elevados valores de “speed up”. Contudo, é importante observar que essa máquina é do tipo híbrida e tem uma razão capacidade de processamento por nó pela capacidade de comunicação muito baixa (um processador i286 em cada nó). Ver a tabela 6.14 para fins de comparação com os parâmetros de outras máquinas paralelas. Assim, os índices de desempenho apresentados em [38] não deverão se repetir em máquinas paralelas mais modernas e do tipo memória distribuída, como por exemplo o Intel iPSC/860 e Intel Delta.

### 3.1.3 O Simulador Digital em Tempo Real

A idéia básica desta investigação foi desenvolver um simulador digital em tempo real, da dinâmica de SEE, baseado em uma configuração massivamente paralela do computador Ncube 2 [18]. O objetivo deste instrumento, que contém conversores analógicos/digitais e interface homem/máquina, é viabilizar a análise do efeito da adição de novos equipamentos em SEE, antes de realizar a instalação final. Na seqüência, a análise concentrar-se-á na parte referente ao algoritmo paralelo, sua implementação e resultados.

O algoritmo de solução utilizado baseia-se no esquema *Alternado*, com a filosofia de usar um processador para cada barra da rede elétrica. As equações diferenciais correspondentes a um determinado gerador e seus controladores, são resolvidas no processador atribuído a barra a qual estão conectados, a fim de minimizar o custo de comunicação. O método de integração numérica usado é o trapezoidal implícito. As equações da rede elétrica são resolvidas por um método semelhante ao Gauss-Seidel e segundo o seguinte algoritmo, onde  $V_G$  e  $V_i$  representam, respectivamente, as tensões interna e externa da  $i$ -ésima barra de geração.

Algoritmo para resolver  $I = YV$  [18]

1. *Calcule*

$$V_i^{k+1} = -(Y_G \cdot V_G + \sum_{j \neq i} Y_{ij} \cdot V_j^k) / Y_{ii}$$

2. *Calcule*

$$I_i^{k+1} = Y_G \cdot V_G + \sum_{j \neq i} Y_{ij} \cdot V_j^{k+1} + Y_{ii} \cdot V_i^{k+1}$$

3. *Se  $|I_i^{k+1}| > \epsilon$  vá ao passo 1*

*Senão, retorne as equações diferenciais.*

Faz-se então os seguintes comentários :

- Os resultados apresentados em simulações com sistemas de até 261 barras, em uma configuração do Ncube 2 com 512 processadores, mostram que o tempo de CPU não é afetado pela dimensão do sistema. Entretanto, tem sido mostrado na literatura [1] que o número de iterações do método Gauss-Seidel, quando aplicado ao problema da simulação da dinâmica de SEE, varia da faixa de 2-3 a centenas de iterações, podendo crescer muito com a dimensão do sistema. Assim, essa independência do tempo de CPU em relação a dimensão do sistema pode não ocorrer em sistemas de médio e grande porte;
- O simulador desenvolvido parece ter uma faixa de aplicações bem definida, ou seja, como instrumento de treinamento e de testes de novos equipamentos. No âmbito dos Centros de Supervisão e Controle da operação em tempo real e especificamente na realização do controle preventivo [39, 9], a incorporação dos aspectos dinâmicos requer a simulação de um número elevado de contingências. Assim, usando-se a filosofia adotada no referido simulador, o número de processadores seria proibitivo. Parece que para essas aplicações um outro caminho deve ser seguido;
- Em termos de índices de desempenho, a única preocupação dos autores foi vencer a barreira do tempo real. O desbalanceamento da carga e o custo de comunicação fazem com que os processadores sejam utilizados com índices de eficiência muito baixos. Este aspecto pode ser ilustrado destacando-se que a simulação de 10 segundos da dinâmica do sistema IEEE 118 barras, foi feita em pouco menos de 10 segundos usando-se 118 processadores do Ncube 2. Por outro

lado, em [21] é citado que foram simulados 1,12 segundos do comportamento dinâmico do sistema IEEE 118 em 1,0 segundo de CPU, usando-se o método SOR-Newton no computador Alliant FX/8 com somente 8 processadores.

## 3.2 Paralelização no Tempo

A idéia inicial de explorar o paralelismo no tempo, em SEE, é devida a Alvarado [24], que fez um estudo genérico para a resolução de equações diferenciais ordinárias lineares e não-lineares. No caso não-linear, as equações diferenciais são transformadas em equações algébricas a diferenças, pela aplicação do método de integração trapezoidal implícito a diversos passos de integração simultaneamente, denominados de janela de integração. O sistema de equações algébricas a diferenças é resolvido pelo método de Newton; e a equação linear Jacobiana, associada a todos os passos da janela de integração, é resolvida pelo método direto baseado na fatoração LU. Foi utilizado um critério de reordenação sucessiva dos pivôs da matriz Jacobiana de modo a se obter algum desacoplamento entre as equações. O número de passos seqüenciais resultantes é  $\tau = \log_2 p$ , onde  $p$  é o número total de passos de integração considerados simultaneamente. Supondo-se que o sistema possua  $n$  equações para cada passo de integração, a proposta de Alvarado [24] é resolver  $p \cdot n$  equações, em  $\tau$  passos seqüenciais de processamento, com um máximo de  $p/2$  processadores.

### 3.2.1 Formulação Matemática

Considere o seguinte sistema de equações diferenciais ordinárias não-lineares de ordem  $n$ .

$$\dot{x} = f(x, t), \quad x(0) = x_0 \quad (3.5)$$

onde:  $x \in \mathbb{R}^n$  e  $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$  é contínua.

Aplicando-se a regra trapezoidal a equação (3.5), no intervalo de tempo  $t$  e  $t + \Delta t$  (ver apêndice B), resulta:

$$x(t + \Delta t) = x(t) + \frac{\Delta t}{2} [f(x(t), t) + f(x(t + \Delta t), t + \Delta t)] \quad (3.6)$$

Definindo-se um subscrito  $t$  para o vetor  $x$ , de modo que  $t$  represente o instante de tempo  $t$ , ou seja,  $x_t = x(t)$ ,  $x_{t+1} = x(t + \Delta t)$ , ..., a equação (3.6) pode ser rearranjada e reescrita na seguinte forma:

$$g_{t+1}(x_{t+1}) = b_t(x_t) \quad (3.7)$$

onde:

$$\begin{cases} g_{t+1}(x_{t+1}) = x(t + \Delta t) - \frac{\Delta t}{2} f(x(t + \Delta t), t + \Delta t) \\ b_t(x_t) = x(t) + \frac{\Delta t}{2} f(x(t), t) \end{cases} \quad (3.8)$$

Se  $x_t$  for conhecido, a equação (3.7) representa uma equação algébrica não-linear em  $x_{t+1}$ , que pode ser resolvida pelo método de Newton a partir de uma condição inicial  $x_{t+1}^0$ . Consequentemente, a descrição do sistema (3.5) para  $p$  passos de integração tomados simultaneamente ( $p = 2^\tau$ ,  $\tau$  inteiro), será dada pelo seguinte sistema de equações algébricas a diferenças.

$$\begin{cases} g_1(x_1) = b_1(x_0) \\ g_2(x_1, x_2) = b_2 \\ g_3(x_2, x_3) = b_3 \\ \vdots \\ g_p(x_{p-1}, x_p) = b_p \end{cases} \quad (3.9)$$

ou, na seguinte forma compacta:

$$\Gamma = g(x) - b = 0 \quad (3.10)$$

O sistema (3.10) representa  $p.n$  equações algébricas não lineares em  $p.n$  incógnitas.

O método de Newton também pode ser aplicado a equação (3.10), de modo que uma estimativa para  $x$  pode ser sucessivamente melhorada resolvendo-se as seguintes equações:

$$\begin{cases} \Gamma(x^k) = J\Delta x^{k+1} \\ e \\ x^{k+1} = x^k - \Delta x^{k+1} \end{cases} \quad (3.11)$$

onde:

$$J = \frac{\partial g(x)}{\partial x} \Big|_{x=x^k} \quad (3.12)$$

A matriz Jacobiana da primeira das equações (3.11) tem estrutura de blocos, conforme ilustra-se a seguir:

$$\begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \vdots \\ \Gamma_p \end{bmatrix}^k = \begin{bmatrix} d_1 & & & & & \\ h_1 & d_2 & & & & \\ & h_2 & d_3 & & & \\ & & \ddots & \ddots & & \\ & & & h_{p-1} & d_p & \end{bmatrix}^k \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \vdots \\ \Delta x_p \end{bmatrix}^{k+1} \quad (3.13)$$





a fase de fatoração apresenta as seguintes peculiaridades:

- Os primeiros  $p/2$  blocos diagonais ( $d_i; i = 1, 3, \dots, p-1$ ), podem ser fatorados independentemente;
- Posteriormente, a fatoração deve incluir também os elementos dos blocos  $h_i$ . Mesmo assim, tem-se, sucessivamente,  $p/4, p/8, \dots$ , processos independentes de fatoração.

Reescrevendo-se a equação (3.14), já fatorada, na seguinte forma compacta

$$J\Delta x = LU\Delta x = \Gamma, \quad (3.15)$$

o processo de resolução resume-se em resolver, seqüencialmente, as seguintes equações matriciais triangulares.

$$\begin{cases} Lz = \Gamma \\ U\Delta x = z \end{cases} \quad (3.16)$$

### 3.2.2 Comentários

A análise do modelo proposto permitiu destacar os seguintes aspectos considerados mais relevantes:

- O algoritmo paralelo de solução apresentado envolve a construção, reordenação e fatoração de uma matriz Jacobiana muito grande. Rigorosamente, esse Jacobiano deve ser recalculado a cada iteração, porém, o tempo de uma fatoração tal como a descrita é muito significativo;
- O método de Newton apresenta convergência quadrática, com o número de iterações praticamente independente da dimensão do sistema. Esta característica sugere que o método de Alvarado [24] é potencialmente promissor, porém, a condição inicial deve ser cuidadosamente fornecida para evitar dificuldades de convergência ou até mesmo divergência da solução;
- As soluções paralelas exigem a inicialização dos vetores  $x_1, x_2, \dots, x_p$ , em todos os  $p$  passos de integração;

- O autor [24] propôs dois critérios para a inicialização: No primeiro, denominado “flat start”, as variáveis são inicializadas com o valor correspondente ao instante de tempo  $t = 0$ , ou seja,  $x_t = x_0$ ,  $t = 1, 2, \dots, p$ . No segundo, foi proposta a realização de uma solução preliminar, no período  $0 - p$ , com o intervalo de integração  $\Delta t' = 2^{\tau'} \Delta t$ , onde  $1 < \tau' < \tau$ . Os resultados dessa solução são interpolados de modo a gerarem condições iniciais para a resolução do problema com o passo de integração igual a  $\Delta t$ . É também citada a possibilidade de utilização de um processo recursivo de inicialização, de modo a gerar uma seqüência de problemas com  $2, 2^2, 2^4, \dots, 2^{\tau/2}, 2^\tau$  passos de integração;
- A aplicação sucessiva do critério de ordenação proposto, implica na necessidade de fatorar matrizes resultantes do produto de duas ou mais matrizes esparsas, dependendo do número de aplicações sucessivas do critério, e que, portanto, serão menos esparsas que as matrizes originais. Este problema acentua-se com o crescimento de  $p$ ;
- O tratamento de descontinuidades tais como: chaveamentos na rede elétrica, limites dos controladores e outros é também uma questão importante a ser considerada;
- O sistema pode ter constantes de tempo muito menores que o intervalo de integração adotado no processo de inicialização sugerido. Isto deve ser tratado convenientemente;
- Em sistemas de grande porte pode ser necessário, e até conveniente, dividir os cálculos envolvidos, a cada passo de integração, em tarefas que possam ser realizadas simultaneamente. Neste caso, o método proposto por Alvarado [24] ficaria na dependência do desenvolvimento de uma forma efetiva de paralelização do método direto de solução de sistemas lineares baseado na fatoração LU de matrizes esparsas.

### 3.3 Paralelização Simultânea no Espaço e Tempo

Neste caso, em trabalhos realizados por LaScala et alii [15, 25], são explorados tanto a paralelização no tempo quanto a paralelização no espaço. Diferentemente do trabalho de Alvarado [24], os trabalhos de LaScala et alii [15, 25] usam o conceito de

relaxação da forma de onda, embora com uma formulação diferente da formulação clássica usada em [14, 23, 40]. Foi realizada a discretização para todos os passos de integração simultaneamente, resultando em um sistema de equações algébricas não-lineares com um número muito grande de equações. No primeiro trabalho [25], a resolução desse sistema foi feita com a aplicação do método Gauss-Jacobi (substituições sucessivas), que apresentou desempenho relativamente fraco. No segundo [15], foi proposto o método denominado Gauss-Jacobi-Block-Newton que consiste essencialmente em resolver, através do método Very Dishonest Newton (VDHN), as equações associadas a cada passo de integração (bloco de equações) e aplicar, globalmente, o método iterativo Gauss-Jacobi.

A seguir apresentam-se uma síntese da formulação matemática empregada e análise.

### 3.3.1 Solução por Gauss-Jacobi (Substituições Sucessivas)

Algebrizando-se as equações diferenciais (2.1) através da aplicação do método de integração trapezoidal implícito, o SEE pode ser descrito pelo seguinte sistema de  $2n + 6m$  equações:

$$\begin{cases} x(t) - x(t - \Delta t) = \frac{\Delta t}{2} \{A(x(t) + x(t - \Delta t)) + B[u(x(t), V^e(t)) + u(x(t - \Delta t), V^e(t - \Delta t))]\} \\ I^e(x(t), V^e(t)) = Y^e(x(t)) V^e(t), \end{cases} \quad (3.17)$$

onde  $\Delta t$  é o tamanho do passo de integração e  $t$  representa o  $t$ -ésimo passo de integração. Essas equações devem ser resolvidas, simultaneamente, para cada passo de integração.

Separando-se a matriz  $Y^e(x(t))$  como

$$Y^e(x(t)) = Y_D(x(t)) + Y_0, \quad (3.18)$$

onde  $Y_D$  é uma matriz diagonal formada pelos elementos diagonais de  $Y^e(x(t))$ , e considerando-se  $p$  passos de integração simultaneamente (uma janela de integração), a aplicação do método iterativo Gauss-Jacobi, ao sistema (3.17), implica em calcular sucessivamente as seguintes  $p \cdot (2n + 6m)$  equações:

$$\begin{cases} \hat{x}_t^{k+1} = \hat{x}_{t-1}^k + \frac{\Delta t}{2} \left\{ A(\hat{x}_t^k + \hat{x}_{t-1}^k) + B[u(\hat{x}_t^k, \hat{V}_t^{ek}) + u(\hat{x}_{t-1}^k, \hat{V}_{t-1}^{ek})] \right\} \\ \hat{V}_t^{ek+1} = -Y_D^{-1}(\hat{x}_t^k) Y_0 \hat{V}_t^{ek} + Y_D^{-1}(\hat{x}_t^k) I(\hat{x}_t^k, \hat{V}_t^{ek}), t = 1, 2, \dots, p \end{cases} \quad (3.19)$$

onde:

$k$  -  $k$ -ésima iteração do algoritmo de relaxação;

$\hat{x} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_p]^T$  - vetor de variáveis de estado da janela de integração;

$\hat{V}^e = [\hat{V}_1^e, \hat{V}_2^e, \dots, \hat{V}_p^e]^T$  - vetor de tensões nodais expandidas da janela de integração.

Para acelerar a convergência do processo iterativo baseado em (3.19), foi utilizado o método “nested iteration”, uma das versões da técnica denominada “multigrid integration”, empregada com sucesso na resolução discretizada de equações diferenciais parciais. Essa técnica é essencialmente um processo recursivo de inicialização do processo de solução, a partir de  $x_t = x(0)$ , para  $t = 1, 2, \dots, p$  (“flat start”). Com essa condição inicial, aplica-se o algoritmo baseado nas equações (3.19), em seqüência, para passos de integração  $\Delta t, \Delta t/2, \Delta t/4, \dots$ , até obter-se uma solução com o passo de integração inferior a menor constante de tempo do sistema. A transferência de dados de uma grade grossa, com  $p$  passos de integração, para uma grade mais fina, com  $2p$  passos de integração, é feita com o uso de interpolação e da forma como descrito a seguir:

$$\begin{cases} x_{2t}^{\Delta t/2} = x_t^{\Delta t} \\ x_{2t+1}^{\Delta t/2} = \frac{1}{2} [x_t^{\Delta t} + x_{t+1}^{\Delta t}] \\ V_{2t}^{e\Delta t/2} = V_t^{e\Delta t} \\ V_{2t+1}^{e\Delta t/2} = \frac{1}{2} [V_t^{e\Delta t} + V_{t+1}^{e\Delta t}] \end{cases} \quad (3.20)$$

onde  $t = 0, 1, 2, \dots, p$ , e os sobrescritos  $\Delta t/2$  e  $\Delta t$  indicam, respectivamente as grades com  $p$  e  $2p$  passos de integração.

LaScala [25] menciona que a dificuldade na implementação da técnica “nested iteration”, reside no requisito de que o esquema de relaxação não deve apresentar instabilidade numérica nas grades mais grossas. Para contornar esse problema foi utilizada

a idéia de que, para as grades mais grossas, deve ser utilizado um modelo que represente apenas a dinâmica mais lenta. O efeito dos componentes com dinâmica mais rápida é considerado somente nas grades mais finas. Essa idéia foi implementada anulando-se a primeira derivada das variáveis associadas a constantes de tempo mais rápidas, para o caso das grades mais grossas. Outra alternativa proposta, é aumentar artificialmente as constantes de tempo dos reguladores nas grades mais grossas.

### 3.3.2 Solução pelo Método Gauss-Jacobi-Block-Newton

Esta proposta, também feita por LaScala et alii [25], é descrita sucintamente a seguir. Considere a equação (2.16) que descreve o comportamento dinâmico do sistema para o intervalo de tempo  $t - \Delta t$  a  $t$ . Definindo-se  $y(t) = [x(t), V^e(t)]^T$  e  $H = [F, G]^T$ , (2.16) pode ser representado por:

$$H(y(t), y(t - 1)) = 0 \quad (3.21)$$

Nos métodos de Newton ou quase-Newton, a equação (3.21) é resolvida através das seguintes equações principais:

$$\frac{\partial H}{\partial y} \Big|_{y(t)=y^k(t)} \Delta y(t)^{k+1} = -H(y^k(t), y(t - 1)) \quad (3.22)$$

e

$$y^{k+1}(t) = y^k(t) + \Delta y^{k+1}(t) \quad (3.23)$$

onde  $k = 1, 2, \dots$ , representa a  $k$ -ésima iteração e  $y(t - 1)$  é a solução do mesmo problema para o instante de tempo  $t - \Delta t$ .

#### Paralelização no Tempo

LaScala [15] propõe reescrever a equação (3.21) na seguinte forma:

$$\hat{H}(\hat{y}) = 0 \quad (3.24)$$

onde:

$p$  - é o número de passos de integração tomados simultaneamente;

$\hat{y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_i, \dots, \hat{y}_p]^T$  - vetor de variáveis do problema para todos os  $p$  passos de integração;

$\hat{H} = [\hat{H}_1, \hat{H}_2, \dots, \hat{H}_t, \dots, \hat{H}_p]^T$  - vetor de funções algébricas não-lineares para todos os  $p$  passos de integração.

A equação (3.24) descreve o comportamento do sistema para  $p$  passos de integração e constitui-se de  $p \cdot (6m + 2n)$  equações algébricas não-lineares.

O algoritmo Gauss-Jacobi-Block-Newton, proposto para resolver o sistema (3.24), baseia-se em aplicar um método tipo Newton às equações associadas a cada passo de integração e usar uma relaxação do tipo Gauss-Jacobi entre os passos de integração.

Essencialmente, o método proposto consiste em resolver as  $p$  seguintes equações simultaneamente:

$$\hat{y}_t^{k+1} = \hat{y}_t^k - \left. \frac{\partial \hat{H}_t}{\partial \hat{y}_t} \right|_{\hat{y}_t = \hat{y}_t^k}^{-1} \hat{H}_t(\hat{y}_t^k, \hat{y}_{t-1}^k); \quad t = 1, 2, \dots, p \quad (3.25)$$

Note que na equação (3.25), os resultados finais do passo anterior, usados pelos métodos passo a passo convencionais, são substituídos pelos valores do passo anterior correspondentes a iteração  $k$ , caracterizando a relaxação do tipo Gauss-Jacobi. Considerando-se somente a paralelização no tempo, essa formulação gera  $p$  processos independentes para cada iteração.

LaScala et alii [15] propuseram a exploração simultaneamente da paralelização no tempo e no espaço, através de uma arquitetura computacional formada por um conjunto de  $p$  hipercubos de dimensão  $d = \log_2(n)$ . As equações associadas ao  $t$ -ésimo passo de integração são atribuídas ao  $t$ -ésimo hipercubo (paralelização no tempo), e cada processador do hipercubo é responsável pela avaliação das variáveis associadas a um determinado nó da rede elétrica (paralelização no espaço), inclusive as variáveis de estado dos geradores, no caso de nós que possuam geradores. Parece óbvio, nesta proposta, que alguns processadores serão utilizados com baixa eficiência, pelos seguintes motivos:

- A carga computacional associada a nós com geradores é muito superior a dos nós sem geração;
- A resolução da equação Jacobiana reduzida é realizada pelo método da fatoração LU de matrizes esparsas, naturalmente seqüencial.

Os trabalhos de LaScala et alii [15, 25] apresentam resultados de simulações realizadas em um computador seqüencial, com sistemas testes de 6, 118 e 662 barras.

Nenhuma implementação paralela real é relatada no corpo dos trabalhos. Entretanto em [21] são descritos uma implementação e testes desta metodologia no computador iPSC/2. Os resultados mostram “speed ups” maiores para o sistema de 662 barras comparativamente ao sistema de 118 barras. Além disso, a paralelização no tempo propiciou ganhos para no máximo 8 passos de integração tomados simultaneamente.

### 3.3.3 Comentários

Ao encerrar a análise dos trabalhos de LaScala et alii [15, 25] e discussões associadas, pode-se destacar o que segue:

- A paralelização no tempo permite auferir ganhos razoáveis, mas somente para um número limitado de passos de integração considerados simultaneamente, conforme antecipado por Alvarado [24];
- Em termos de “speed up” e eficiência os resultados parecem pobres, entretanto, houve um ganho de velocidade (“speed up”) de 8,79 quando foram tomados 8 passos simultaneamente e empregados 32 processadores do iPSC/2 na simulação do sistema de 662 barras [21];
- Esforços devem ser concentrados no sentido do desenvolvimento de métodos mais eficientes que os empregados por LaScala [25, 15], para resolver o sistema de equações algébricas não lineares resultantes da discretização de vários passos de integração simultaneamente.

## 3.4 Métodos do Tipo Newton Relaxado

O desenvolvimento dos métodos tipo Newton relaxado, analisados brevemente a seguir, foi motivado pela busca de alternativas aos baixos índices de desempenho do método VDHN em propósitos de processamento paralelo e vetorial. A principal dificuldade está na resolução da equação linear Jacobiana pelo método direto baseado na fatoração LU. São analisados a seguir os métodos SOR-Newton [21] e Maclaurin-Newton [17], cuja filosofia base é realizar aproximações na matriz Jacobiana de modo que o sistema linear resultante possa ser resolvido eficientemente em computadores paralelos.



### 3.4.1 O Método SOR-Newton

Neste caso, a matriz Jacobiano modificada é diagonal e tem seus elementos assim definidos:

$$\frac{\partial h_q(x_t^k, V_t^{ek})}{\partial y_{p,t}} = \begin{cases} \partial_q h_q(x_t^k, V_t^{ek}) & p = q \\ 0 & p \neq q \end{cases} \quad (3.26)$$

onde  $y_t = [x_t^k, V_t^{ek}]^T$  é o vetor de variáveis do problema para o  $t$  éximo passo de integração e  $h_q$  é a  $q$ -ésima função não-linear do sistema de equações.

A renovação dos elementos de  $y_t$ , a cada iteração, é feita usando-se fatores de relaxação  $W_q$  e conforme a seguinte expressão:

$$y_{q,t}^{k+1} = y_{q,t}^k - W_q \frac{h_q(x_t^k, V_t^{ek})}{\partial_q h_q(x_t^k, V_t^{ek})} \quad (3.27)$$

Foram utilizados fatores de relaxação diferentes para as variáveis de estado ( $W_d = 1,9$ ) e para as variáveis algébricas ( $W_s = 0,9$ ). Esses fatores de relaxação podem ter influência significativa no comportamento do método e precisam ser ajustados com a prática, pois, não há meios de se determinar fatores ótimos em sistemas lineares tão complexos [21].

### 3.4.2 O Método Maclaurin-Newton

A idéia básica dessa formulação é aproximar a inversa da matriz Jacobiano reduzida  $\bar{J}_4$  pelos dois primeiros termos de sua expansão em série de Maclaurin. Assim a solução do sistema linear Jacobiano é dada por [17]:

$$\Delta V_t^{ek+1} \approx \omega_v (I - (\bar{J}_{4D}^k)^{-1} \bar{J}_{4off}) \cdot (\bar{J}_{4D}^k)^{-1} \bar{G}_t^k \quad (3.28)$$

$$\Delta x_t^{k+1} \approx -\omega_x (A_{Mt}^k)^{-1} (\hat{F}_t^k + B_t^k \cdot \Delta V_t^{ek+1}) \quad (3.29)$$

onde  $t$  corresponde ao instante de tempo do  $t$ -ésimo passo de integração;  $\omega_v$  e  $\omega_x$  são fatores de aceleração;  $\bar{J}_{4D}$  e  $\bar{J}_{4off}$  são, respectivamente, as matrizes formadas por elementos diagonais e elementos não diagonais, extraídos do Jacobiano.

### 3.4.3 Discussão

Os algoritmos foram implementados e testados no computador paralelo hipercúbico de memória distribuída iPSC/2 e nos computadores de memória compartilhada Al-

liant FX/8 e Symmetry. Os testes foram realizados com os sistemas IEEE 118 e US. Midwestern com 662 barras. Dentre os resultados apresentados destacam-se os seguintes aspectos:

- As curvas de “speed up” apresentam forte saturação com o aumento do número de processadores, contudo os valores são um pouco superiores aos do algoritmo VDHN paralelo. Os melhores índices de desempenho foram obtidos com o computador Symmetry de memória compartilhada, em simulações do sistema de 662 barras com o algoritmo Maclaurin-Newton. Para ilustrar, destacam-se os “speed ups” máximos de 8,48 no iPSC/2 com 32 processadores e 11,8 no Symmetry com 26 processadores;
- Os algoritmos tipo Newton relaxado são sensíveis a severidade das perturbações e ao ajuste dos coeficientes de aceleração;
- Em relação ao paralelismo no tempo, os métodos do tipo Newton relaxado apresentaram desempenho muito inferior ao VDHN, com ganhos inexpressivos. Os autores [17] afirmam que a principal razão disso é que os fatores de aceleração causam grandes oscilações nos resíduos quando são utilizados vários passos de integração simultaneamente.

### 3.5 Relaxação da Forma de Onda (RFO)

A aplicação do método de relaxação da forma de onda, em sua forma clássica, ao problema da SDSEE, tem sido explorada por Crow e Ilić em vários trabalhos [3, 14, 23, 41, 42].

A premissa básica do método de RFO é que um sistema de grande porte pode ser decomposto em diversos subsistemas menores, fracamente acoplados entre si, e que cada um deles pode exibir comportamento muito distinto. Os principais apelos da RFO são a possibilidade de resolver subsistemas diferentes com passos de integração diferentes (“Multirate integration”) e o fato de evitar a resolução, por métodos diretos, de sistemas algébricos lineares muito grandes.

Trabalhos na área de simulação de circuitos VLSI tem mostrado que o tempo necessário para resolver um sistema linear e esparso, por métodos diretos, cresce *superlinearmente* com a dimensão do sistema. Além disso, é difícil se obter um alto grau

de paralelização para os métodos diretos [40]. A razão de convergência do método de RFO é fortemente influenciada pelas características de acoplamento dos subsistemas, ou seja, pela decomposição obtida. Entretanto, é citado em [23] que o algoritmo de RFO não tem sido tão efetivo na simulação de circuitos VLSI como inicialmente esperado. Isto é atribuído a problemas relacionados com as características desses circuitos, tais como a presença de “floating capacitor” que não são encontrados em SEE.

### 3.5.1 O Método RFO para Sistemas de Equações Diferenciais Ordinárias

Considere o sistema dinâmico descrito por:

$$\dot{x}(t) = f(x, t), \quad x(0) = x_0 \quad (3.30)$$

onde  $x \in \mathbb{R}^n$  em  $t \in [0, T]$  e  $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$  é contínua.

A decomposição do sistema (3.30), em  $r$  subsistemas fracamente acoplados entre si, de modo que cada subsistema reúna variáveis de estado fortemente acopladas, é representada matematicamente por:

$$\begin{cases} \dot{x}_1(t) = f_1(x_1, x_2, \dots, x_r, t), & x_1(0) = x_{10} \\ \dot{x}_2(t) = f_2(x_1, x_2, \dots, x_r, t), & x_2(0) = x_{20} \\ \vdots \\ \dot{x}_r(t) = f_r(x_1, x_2, \dots, x_r, t), & x_r(0) = x_{r0} \end{cases} \quad (3.31)$$

onde  $x_i \in \mathbb{R}^{n_i}$ ,  $\sum_{i=1}^r n_i = n$ , e  $f_i : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n_i}$ .

A resolução das equações (3.31) baseia-se nos métodos de relaxação Gauss-Seidel e Gauss-Jacobi, usados para resolver sistemas de equações algébricas lineares e não-lineares. A título de ilustração, apresenta-se a seguir o algoritmo de RFO do tipo Gauss-Jacobi aplicado a equação (3.31). Mais detalhes podem ser encontrados em [23] ou em [43].

#### Algoritmo de RFO do tipo Gauss-Jacobi

*Inicialize  $x^0(t)$  tal que  $x^1(0) = x(0)$ ;*

Para  $k = 0, 1, 2, \dots$

Para  $i = 1, 2, \dots, r$  e no intervalo  $[0, T]$

Resolva:

$$\dot{x}_i^{k+1} = f_i(x_1^k, x_2^k, \dots, x_{i-1}^k, x_i^{k+1}, x_{i+1}^k, \dots, x_r^k, t), \quad x_i^{k+1}(0) = x_i(0)$$

Fim-Para

Calcule  $\Delta x^{k+1} = \|x^{k+1} - x^k\|$

Se  $\Delta x^{k+1} \geq \epsilon$ , continue; ( $\epsilon$  pequeno e positivo)

Fim-Para

No algoritmo Gauss-Jacobi, cada subsistema é resolvido independentemente com as formas de onda da iteração anterior como dados. Os subsistemas podem ser resolvidos em qualquer ordem ou em paralelo. Por outro lado, o algoritmo de Gauss-Seidel tem natureza seqüencial.

### 3.5.2 O Método da RFO para Sistemas de Equações Diferenciais e Algébricas

Considere um sistema dinâmico descrito pelas seguintes equações:

$$\begin{cases} \dot{x}(t) = F(x, z, t); & x(0) = x_0 \\ 0 = G(x, z, t); & z(0) = z_0 \end{cases} \quad (3.32)$$

onde  $x \in \mathbb{R}^n$ ,  $z \in \mathbb{R}^m$  em  $t \in [0, T]$ ; e  $F : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$ ,  $G : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^m$ , são contínuas.

Crow e Ilić [14, 23], propuseram que a decomposição das equações (3.32), em subsistemas fracamente acoplados entre si, seja feita de forma similar a adotada para sistemas completamente descritos por equações diferenciais. As variáveis de estado e algébricas fortemente acopladas, são atribuídas a um mesmo subsistema. Matematicamente, isto é representado através da decomposição das equações (3.32)

em  $r$  subsistemas, conforme as seguintes equações:

$$\left\{ \begin{array}{l} \dot{x}_1(t) = F_1(x_1, x_2, \dots, x_r, z_1, z_2, \dots, z_r, t); \quad x_1(0) = x_{10} \\ 0 = G_1(x_1, x_2, \dots, x_r, z_1, z_2, \dots, z_r, t); \quad z_1(0) = z_{10} \\ \dot{x}_2(t) = F_2(x_1, x_2, \dots, x_r, z_1, z_2, \dots, z_r, t); \quad x_2(0) = x_{20} \\ 0 = G_2(x_1, x_2, \dots, x_r, z_1, z_2, \dots, z_r, t); \quad z_2(0) = z_{20} \\ \vdots \\ \vdots \\ \dot{x}_r(t) = F_r(x_1, x_2, \dots, x_r, z_1, z_2, \dots, z_r, t); \quad x_r(0) = x_{r0} \\ 0 = G_r(x_1, x_2, \dots, x_r, z_1, z_2, \dots, z_r, t); \quad z_r(0) = z_{r0} \end{array} \right. \quad (3.33)$$

onde:  $x_i \in \mathbb{R}^{n_i}$ ,  $z_i \in \mathbb{R}^{m_i}$ ,  $\sum_{i=1}^r n_i = n$ ,  $\sum_{i=1}^r m_i = m$ ,  $F_i : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^{n_i}$  e  $G_i : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^{m_i}$ .

Nestas equações são aplicados os algoritmos de RFO do tipo Gauss-Jacobi ou Gauss-Seidel. Os dados para uma determinada iteração são as formas de onda das variáveis de estado e algébricas da iteração anterior. Também a título de ilustração, apresenta-se a seguir o algoritmo de RFO do tipo Gauss-Jacobi para as equações (3.33). Outros detalhes podem ser encontrados em [23] ou em [43].

### Algoritmo de RFO do tipo Gauss-Jacobi para EDAs

Inicialize  $x^0(t)$  e  $z^0(t)$  tal que

$$x^1(0) = x(0) \text{ e } z^1(0) = z(0);$$

Para  $k = 0, 1, 2, \dots$

Para  $i = 1, 2, \dots, r$  e no intervalo  $[0, T]$

Resolva:

$$\dot{x}_i^{k+1} = F_i(x_1^k, \dots, x_i^{k+1}, \dots, x_r^k, z_1^k, \dots, z_i^{k+1}, \dots, z_r^k, t), \quad x_i^{k+1}(0) = x_i(0)$$

$$0 = G_i(x_1^k, \dots, x_i^{k+1}, \dots, x_r^k, z_1^k, \dots, z_i^{k+1}, \dots, z_r^k, t); \quad z_i^{k+1}(0) = z_i(0)$$

Fim-Para

Calcule  $\Delta x^{k+1} = \|x^{k+1} - x^k\|$  e  $\Delta z^{k+1} = \|z^{k+1} - z^k\|$

Se  $\Delta x^{k+1} \geq \epsilon_x$  ou  $\Delta z^{k+1} \geq \epsilon_z$ , continue

Fim-Para

### 3.5.3 Aplicações a SDSEE/Decomposição

O trabalho de Crow e Ilić, relativo a aplicação do método de RFO ao problema da SDSEE, descrito nas publicações [3, 14, 23, 41], apresenta uma larga abrangência. Inicialmente a idéia foi implementada para o modelo clássico de SEE [3]. Os resultados de experimentos apresentados referem-se a simulações seqüenciais realizadas com sistemas de 3, 10 e 20 máquinas. São apresentados: os resultados de comparações do desempenho do método de RFO, em relação a metodologia clássica com o método de integração numérica Runge-Kutta (RK); os resultados relativos ao efeito da divisão do intervalo de simulação em janelas; e os resultados relativos a utilização das propriedades de coerência das oscilações na decomposição do SEE. Analisando-se esses resultados pode-se concluir o que segue:

- A RFO sem o uso de janelas e propriedades de coerência, na decomposição, foi 3 vezes mais lenta que a metodologia convencional;
- A utilização de janelas reduziu em cerca de 50 % o tempo de CPU;
- A decomposição baseada nas propriedades de coerência contribuiu para uma significativa redução no tempo de CPU.

O método de RFO é uma contração em uma norma uniforme para pequenos intervalos de tempo e uma contração em uma norma exponencialmente ponderada para grandes intervalos de tempo [40]. Em razão dessas características, Crow e Ilić [41] tentaram estabelecer alguns limites teóricos para o máximo tamanho de janela em que o acoplamento entre partições se mantenha relativamente fraco, usando as propriedades de coerência. Contudo, esses autores também destacam que uma decomposição ótima do ponto de vista da convergência do algoritmo de RFO, pode não ser adequada do ponto de vista do processamento paralelo, e uma solução de compromisso deve ser encontrada.

Várias técnicas tais como: decomposição adaptativa; decomposição baseada no conceito de resposta localizada; “multirate integration”, baseada no conceito de resposta localizada; e divisão do intervalo de integração em janelas, foram estudadas com o objetivo de obter decomposições que favoreçam o desempenho do algoritmo de RFO. Em [23] são apresentados resultados de simulações nos sistemas de 3, 10 e 20 máquinas síncronas, onde todas essas técnicas foram utilizadas simultaneamente. A

comparação de desempenho foi feita em relação a metodologia convencional com o método de integração trapezoidal implícito. As principais conclusões são: o método de RFO apresentou desempenho crescente com o crescimento da dimensão do sistema; e o desempenho relativo variou de cerca de 1,5 vezes para o sistema de 3 máquinas a cerca de 3,5 vezes para o sistema de 20 máquinas. Embora essa reversão nos índices de desempenho, em relação a comparação anterior, a implementação efetiva de todas essas técnicas parece ser uma tarefa difícil.

Em [14, 23], a aplicação do método de RFO a SEE é estendida para um modelo que preserva a estrutura original da rede elétrica, ou seja, para um sistema de equações diferenciais e algébrica (EDAs). Neste caso uma das dificuldades é a extensão dos critérios de decomposição usados anteriormente e baseados nas propriedades de coerência, para sistemas constituídos de EDAs. Em termos físicos, trata-se de identificar quais as barras da rede elétrica que devem ser atribuídas a cada grupo coerente de máquinas síncronas. Lembre-se que isto deve ser feito de modo que os subsistemas de EDAs resultantes tenham fraco acoplamento entre si. Além disso, para propósitos de processamento paralelo, esses sistemas devem ter dimensões não muito díspares, a fim de propiciarem um razoável balanceamento da carga entre os processadores. Foi, então, proposta a realização da decomposição do sistema de EDAs baseado nas propriedades de coerência para as EDOs, e baseado em um critério de distâncias elétricas para associar as barras da rede elétrica a determinados grupos de máquinas síncronas. Os resultados de simulações nos sistemas de 3, 10 e 20 máquinas, considerando todas as técnicas que contribuem para acelerar a convergência do método de RFO, em relação a metodologia convencional com o método de integração Euler Implícito, mostraram uma superioridade do método de RFO da ordem de 8 a 9 vezes. Contudo, deve ser destacado que Chai et alii [44], nas discussões do artigo de Crow e Ilić [14], afirmam que em suas simulações não tem encontrado qualquer método de relaxação que não seja diversas vezes mais lento que o método VDHN, em computadores seqüenciais.

No sentido de tentar equacionar o problema da decomposição para propósitos de processamento paralelo, Crow e Ilić [14, 23] realizaram uma adaptação do modelo textural de um SEE [45], ao problema da SDSEE com o método de RFO. Contudo, não está claro como as decomposições, baseadas nas propriedades de coerência e no modelo textural, foram relacionadas e como elas são combinadas. Por outro lado,

das discussões do artigo [14], é possível inferir que é usada uma combinação das duas técnicas.

Analisando-se todos os aspectos abordados nesses comentários, parece claro que os trabalhos de Crow e Ilić oferecem importantes contribuições para se aprofundar os conhecimentos relativos a aplicação do método da RFO ao problema da SDSEE. Contudo, destacam-se as dificuldades em se obter uma decomposição prática e de baixo custo computacional, que torne o método efetivo.

### 3.6 A Sinergia do Processamento Paralelo-Vetorial

Este assunto foi tratado por Tylavsky [46], motivado pela tendência futura do processamento de alto desempenho fixar-se no processamento distribuído e no processamento paralelo-vetorial. O referido autor faz uma avaliação dos ganhos de velocidade, registrados na literatura, em relação a utilização de computadores vetoriais ou paralelos em problemas de SEE. Ênfases são dadas na resolução de sistemas lineares do tipo  $LUx = b$  e na simulação da dinâmica. A principal conclusão é que tanto a vetorização quanto a paralelização da resolução de  $LUx = b$  têm resultado em ganhos de velocidade inexpressivos.

Tylavsky [46], então, propõe uma nova metodologia, denominada “Vectorization Across Equations Sets” (VAES), para resolver sistemas do tipo

$$LUx_i = b_i; \quad i = 1, 2, \dots, M \quad (3.34)$$

A metodologia consiste em realizar todas as operações com, por exemplo, o elemento  $l_{21}$  para todos os sistemas lineares ( $i = 1, 2, \dots, M$ ), antes de operar com o próximo elemento da matriz  $L$ . Desta forma, as operações são realizadas com vetores densos de dimensão igual a  $M$ .

São apresentados resultados de testes realizados em um computador Alliant FX/8, operando em modo vetorial e com 8 processadores. Os ganhos máximos de velocidade obtidos são cerca de 25 para  $M$  igual a 128, ou seja, na resolução simultânea de 128 sistemas lineares. São também citados ganhos dessa ordem em análise de contingências, usando-se uma metodologia semelhante a VAES e o computador Cray Y/MP operando com 4 processadores.



Tylavsky então sugere a utilização desta metodologia (VAES) na resolução das equações que descrevem a dinâmica de SEE. Especificamente nos algoritmos que exploram o paralelismo no tempo, com a resolução do sistema não-linear por Newton. Contudo, a metodologia VAES ficará restrita ao sistema linear Jacobiano. Além disso, vários trabalhos tem mostrado ganhos com o paralelismo no tempo para, no máximo, oito passos de integração tomados simultaneamente; e esta dimensão de vetores não resulta em ganhos no processamento vetorial.

### 3.7 Métodos Iterativos do Tipo Gradiente Conjugado

Os inexpressivos ganhos que tem sido obtidos com a paralelização ou vetorização dos métodos diretos de resolução dos sistemas lineares, que surgem no âmbito dos problemas de SEE, conforme descrito nos itens anteriores, tem motivado a investigação no sentido da utilização de métodos iterativos que sejam potencialmente promissores para soluções paralelas e/ou vetoriais. Na literatura técnica [36] encontram-se vários métodos iterativos para a resolução de sistemas algébricos lineares. Os métodos de Jacobi, Gauss-Seidel, Sobrerelaxação e Gradiente Conjugado são os mais conhecidos.

O método de Jacobi é inerentemente paralelo, contudo apresenta convergência muito lenta [36], características que praticamente inviabilizam a aplicação em problemas de SEE [47]. Os métodos Gauss-Seidel e Sobrerelaxação apresentam condições de convergência superiores às do método de Jacobi, porém, não apresentam o mesmo potencial para a paralelização. Adicionalmente, os métodos de Sobrerelaxação utilizam coeficientes para acelerar a convergência, cujos valores devem ser ajustados com a experiência de utilização da metodologia. Valores ótimos são muito difíceis de serem determinados em sistemas de grande porte, caso dos SEE.

Em outras áreas, muitos trabalhos [5, 36, 48, 49, 50, 51, 52] têm aplicado com sucesso os métodos do tipo Gradiente Conjugado, na resolução paralela e vetorial de sistemas lineares esparsos. Os aspectos acima citados contribuíram de forma decisiva para a investigação da aplicabilidade de métodos do tipo Gradiente Conjugado no âmbito da SDSEE, objeto do presente trabalho.

Os métodos do tipo Gradiente Conjugado derivam da família de métodos de solução de sistemas lineares denominada de métodos do subespaço de Krylov ou

métodos tipo projetados [5, 36]. O mais popular é o Gradiente Conjugado (GC), utilizável somente em sistemas com a matriz de coeficientes simétrica [36]. Para matrizes assimétricas, têm sido empregados o GMRES (Generalized Minimal Residual) [5], o Gradiente Biconjugado (BCG - Biconjugate Gradient) [5], o CGS (Conjugate Gradiente-Squared) [5] e, mais recentemente, o Bi-CGSTAB [53]. O maior esforço computacional requerido pelos algoritmos associados a esses métodos está no cálculo de produtos internos e multiplicações de matriz por vetor. Essa peculiaridade é muito atrativa para o processamento em computadores paralelos-vetoriais. Por outro lado, há a necessidade de se encontrar uma efetiva matriz de pré-condicionamento  $M$  [36].

A utilização de métodos do tipo GC em ambientes paralelos para resolver problemas de SEE, foi feita inicialmente por Decker et alii [12, 13]. O método GC foi aplicado no âmbito do esquema *Alternado Entrelaçado Implícito* de resolução das equações que descrevem o comportamento dinâmico. Especificamente, na resolução de um subsistema linearizado das equações que descrevem a rede elétrica. Os resultados apresentam ganhos de velocidade de aproximadamente 4, em um computador paralelo baseado em transputers T800, com 8 processadores em topologia hipercúbica. Esse trabalho evoluiu para a utilização do método GC pré-condicionado na resolução de todas as equações associadas a rede elétrica [54], ainda no âmbito do esquema *Alternado Entrelaçado Implícito*; e para a utilização dos métodos Gradiente Biconjugado e Bi-CGSTAB pré-condicionados, no âmbito do esquema *Simultâneo Implícito Paralelo no Tempo* [55]. Estes trabalhos constituem a parte principal da presente tese e são descritos em detalhes nos capítulos 4, 5 e 6.

Recentemente, Pai et alii [16] descrevem o emprego do método GMRES para resolver a equação Jacobiana do método VDHN. O pré-condicionamento utilizado é do tipo fatoração LU incompleta, baseada na manutenção somente dos maiores "fill-ins". Foram realizados testes com um sistema de 10 máquinas e 39 barras, em processamentos feitos em um único processador do computador Cray-Y-MP. Os resultados mostram ganhos de 3,53 na vetorização sem o pré-condicionamento, e de 2,65 com o pré-condicionamento. Essa queda de desempenho é devida a dificuldades em vetorizar eficientemente o método de pré-condicionamento utilizado. Os tempos de CPU foram menores nos processamentos com pré-condicionamento. É importante ressaltar que para sistemas de grande porte, o pré-condicionamento é imprescindível, senão o número de iterações poderá ser muito elevado. Note também que o método

GMRES necessita de um adequado ajuste, dependente da intervenção e até mesmo da experiência do analista.

Fora do âmbito da SDSEE, vários trabalhos tem descrito a aplicação com sucesso, em computadores paralelos ou vetoriais, de métodos do tipo Gradiente Conjugado pré-condicionado na resolução de sistemas lineares esparsos de grande porte. Aykanat et alii [48], apresentam, para o método GC, índices de eficiências superiores a 80 %, no computador paralelo iPSC/2 com 16 processadores, para soluções paralelas de sistemas do tipo banda e dimensões da ordem de 2000 a 5000 variáveis. Cabral [56] realizou uma avaliação de desempenho do método GC no simulador do iPSC/2. A aplicação refere-se a resolução de sistemas de equações lineares associadas ao modelo desacoplado rápido de fluxo de carga. Foram utilizados dois SEE, respectivamente, com 118 e 340 barras. Os índices de eficiências obtidos são de 75% e 53%, respectivamente, para as configurações paralelas de 8 e 16 processadores. Borges [11] investigou o emprego do método Gradiente Biconjugado pré-condicionado (BCG), na resolução seqüencial e escalar do fluxo de carga pelo método de Newton-Raphson. A metodologia mostrou-se competitiva em relação à forma tradicional de resolução do problema. Dongarra et alii [5] descrevem diversos métodos iterativos do tipo GC, visando a resolução paralela e vetorial de sistemas lineares simétricos e não simétricos. Muitos outros trabalhos têm mostrado a competitividade de métodos do tipo GC, em relação aos métodos diretos, na resolução de sistemas lineares esparsos em áreas de aplicação dos métodos dos elementos finitos e dos elementos de fronteira [49, 50, 51, 57].

### 3.8 Conclusões

Neste capítulo foi descrita uma análise de metodologias recentemente publicadas, que fazem uso de técnicas de processamento paralelo para a simulação da dinâmica de SEE. Algumas dessas metodologias foram testadas em computadores paralelos reais (multiprocessadores com memória distribuída ou com memória compartilhada), outras foram apenas simuladas seqüencialmente para validação e identificação de potencialidades. Os temas abordados abrangem a paralelização no espaço, a paralelização no tempo, a paralelização simultânea no espaço e tempo, a aplicação de métodos do tipo Newton relaxado, o método de relaxação da forma de onda (RFO), a sinergia do processamento paralelo-vetorial; e a aplicação de métodos iterativos do tipo Gradiente Conjugado.

Em relação a paralelização no espaço, foram analisados os métodos VDHN paralelo, “Newton-W matrix” e o simulador digital em tempo real. O método “Newton-W matrix” surgiu de uma sugestão de substituir a solução da equação Jacobiana, feita seqüencialmente no método VDHN paralelo, por uma versão paralelizada do método dos fatores  $W$ . Ambos os métodos apresentaram forte saturação da curva de “speed up” com o crescimento do número de processadores. O método “Newton-W matrix”, contrariamente à expectativa que motivou a sugestão de sua aplicação, apresentou desempenho inferior ao VDHN, com “slowdown” de 10 a 30 %, dependendo das partições de  $W$  escolhidas e da esparsidade do sistema. Isto foi atribuído a dificuldades no balanceamento da carga e aos excessivos cálculos resultantes dos “fill-ins” adicionais na formação das partições da matriz  $W$ .

No que se refere a paralelização no tempo, analisou-se o modelo de Alvarado [24], que requer a solução de um sistema de equações algébricas não-lineares de muito grande porte. Esse sistema é resolvido pelo método de Newton, a matriz Jacobiana é assimétrica e constituída de elementos matriciais não nulos na diagonal principal e na primeira subdiagonal. A equação linear Jacobiana é então resolvida pelo método direto baseado na fatoração LU, após a reordenação dos elementos matriciais de modo a gerarem subtarefas que possam ser processadas simultaneamente. Nenhuma implementação envolvendo máquinas paralelas reais e SEE multimaquinas é relatada. O modelo é potencialmente promissor, porém, as dificuldades atuais residem no estabelecimento de condições iniciais adequadas para todos os passos de integração, no tratamento de discontinuidades, no balanceamento da carga entre os processadores e na forma de solução do sistema linear Jacobiano. É possível que outras formas de solução do sistema linear Jacobiano, como por exemplo os métodos tipo Gradiente Conjugado, possam contribuir para uma eficiente implementação paralela desse modelo.

Na paralelização no espaço e tempo as equações diferenciais são discretizadas para todos os passos de integração do intervalo de simulação simultaneamente. A resolução das equações algébricas e algebrizadas foi feita por Gauss-Jacobi [25] ou pelo método denominado Gauss-Jacobi-Block-Newton [15]. O conceito de relaxação da forma de onda é usado implicitamente. A avaliação dessas metodologias foi feita através da simulação seqüencial de um computador paralelo hipotético, cujo número de processadores é função do número de equações do sistema. Desse modo eles concluíram

que a exploração do paralelismo no espaço e tempo traz vantagens em relação ao paralelismo somente no espaço. Além disso, o método Gauss-Jacobi apresentou desempenho muitas vezes inferior ao método Gauss-Jacobi-Block-Newton. Chai et alii [21] apresentam resultados de uma implementação paralela real do algoritmo Gauss-Jacobi-Block-Newton no computador paralelo iPSC/2. As principais conclusões são: a curva de “speed up” satura rapidamente com o crescimento do número de processadores; os índices de eficiência são todos inferiores a 30 %; a exploração do paralelismo no tempo propicia ganhos para um máximo de 8 passos de integração tomados simultaneamente. As principais causas desses resultados foram identificadas como a lenta convergência do método Gauss-Jacobi e a resolução seqüencial, baseada na fatoração LU, dos sistemas lineares Jacobianos de cada passo de integração.

Os métodos do tipo Newton relaxado foram desenvolvidos como alternativas a solução da equação Jacobiana pelo método baseado na fatoração LU. A idéia básica é obter uma matriz Jacobiana aproximada tal que o sistema linear possa ser resolvido eficientemente em paralelo. Contudo, esses métodos dependem de fatores de aceleração, nem sempre fáceis de serem ajustados adequadamente com a necessidade de um método efetivo. Os “speed ups” obtidos foram superiores aos dos métodos VDHN e “Newton-W matrix”, em computadores com memória compartilhada. No iPSC/2, de memória distribuída, o elevado número de iterações desses métodos implica em elevado volume de comunicação de dados, deteriorando a performance. Além disso, esses métodos são bastante sensíveis a severidade das perturbações sob simulação e apresentam ganhos inexpressivos quando usados em modelos que exploram a paralelização no tempo.

No método de relaxação da forma de onda (RFO), encontram-se aplicações ao modelo clássico, descrito por EDOs, e a um modelo que preserva a estrutura original da rede elétrica, descrito por EDAs. Nenhuma implementação paralela real foi realizada. Os principais apelos do método de RFO são a sua concepção algorítmica, a possibilidade da utilização de técnicas de “multirate integration” e o fato de evitar a resolução de sistemas lineares muito grandes. Por outro lado, conclui-se dos trabalhos analisados que a efetividade do método da RFO está fortemente vinculada a decomposição adotada, ao emprego de técnicas de “multirate integration” e a definição de tamanhos adequados de janelas de integração. Mesmo assim, uma decomposição ótima do ponto de vista da convergência do método, dificilmente será

ótima do ponto de vista do processamento paralelo. Da análise do processo evolutivo da aplicação da RFO a SEE, parece claro que o aspecto crucial dessa metodologia é a fase de decomposição e, portanto, nesse aspecto é que devem ser concentrados os futuros esforços de desenvolvimento.

A tendência futura do processamento de alto desempenho fixar-se no processamento distribuído e no processamento paralelo-vetorial, motivou Tylavsky [46] a propor uma nova forma, baseada na fatoração LU, de resolver grandes conjuntos de sistemas lineares esparsos que podem surgir em problemas como, por exemplo, análise de contingências e simulação dinâmica. A metodologia foi denominada “Vectorization Across Equations Sets” (VAES) e busca operar com vetores longos e densos, de modo a tirar proveito do efeito sinérgico dos ganhos com o paralelismo e vetorização em computadores paralelos-vetoriais. São apresentados resultados de ganhos da ordem de 25; na solução simultânea de 128 sistemas lineares hipotéticos usando o computador Alliant FX/8 com 8 processadores; e na análise de contingências feita no computador Cray Y/MP com 4 processadores.

O último tema abordado neste capítulo de revisão da literatura, foi a aplicação de métodos iterativos do tipo Gradiente Conjugado ao problema da SDSEE, feita inicialmente por Decker et alii [12, 13]; trabalhos que são partes constituintes desta tese. O maior esforço computacional requerido pelos algoritmos associados a esses métodos está no cálculo de produtos internos e multiplicações de matriz por vetor, características que os tornam potencialmente promissores na exploração do efeito sinérgico nos ganhos de velocidade, com o paralelismo e vetorização, em computadores paralelos-vetoriais. Além disso, trabalhos nas áreas de aplicação dos métodos dos elementos finitos e dos elementos de fronteira têm mostrado competitividade dos métodos tipo Gradiente Conjugado em relação aos métodos diretos, com vantagens em sistemas de porte elevado e processados em computadores vetoriais.

## Capítulo 4

# Esquema Alternado Entrelaçado Implícito - Paralelização no Espaço

No presente capítulo descrevem-se dois novos algoritmos paralelos, baseados no esquema *Alternado Entrelaçado Implícito*, desenvolvidos para a resolução das equações que representam a dinâmica de SEE. Nestes algoritmos, as equações diferenciais são resolvidas alternadamente com as equações algébricas, em um processo com relaxação na resolução das equações que representam a rede elétrica. A principal novidade é a utilização de métodos iterativos do tipo Gradiente Conjugado no âmbito da resolução das equações da rede elétrica. Essas propostas resultaram da investigação de novos algoritmos objetivando viabilizar uma eficiente utilização de computadores paralelos na simulação da dinâmica de sistemas de energia elétrica.

No primeiro algoritmo, denominado AEIGC, as equações da rede elétrica são ordenadas tal que a matriz admitância nodal complexa assuma a forma bloco diagonal com blocos nas bordas (“Block Bordered Diagonal Form” - BBDF [26, 27, 28]) e resolvidas, para cada iteração nas variáveis de estado, como se fossem lineares, através de um algoritmo que combina o método iterativo Gradiente Conjugado e o método direto que faz uso da fatoração LU de matrizes esparsas. No segundo algoritmo, denominado AEIGCRD, as equações da rede elétrica também são resolvidas como se fossem lineares, para cada iteração nas variáveis de estado, contudo, emprega-se exclusivamente o método Gradiente Conjugado pré-condicionado na resolução dessas equações.

O capítulo está estruturado de modo que apresenta-se, pela ordem, uma síntese da modelagem empregada, cujos detalhamentos encontram-se no capítulo 2 e apêndices

A e B; a descrição do algoritmo AEIGC; a descrição do algoritmo AEIGCRD; e, finalmente, breves comentários e conclusões.

## 4.1 Formulação Matemática

O modelo matemático para a SDSEE foi descrito no capítulo 2 e pode ser sinteticamente representado pelas seguintes equações:

$$\dot{x} = f(x, u) = Ax + Bu \quad (4.1)$$

$$I(E, V) = YV \quad (4.2)$$

$$u = h(E, V), \quad (4.3)$$

onde a equação (4.1) descreve o comportamento dinâmico dos rotores das máquinas síncronas e seus controladores; a equação (4.2) descreve o comportamento, em regime permanente, da rede elétrica de transmissão, dos estatores das máquinas síncronas e das cargas; e a equação (4.3) serve como interface entre as variáveis da equação (4.1), relacionadas à referência de Park  $d - q$ , e as variáveis da equação (4.2), relacionadas à referência síncrona ou complexa.

No desenvolvimento deste capítulo consideram-se somente os efeitos transitórios para as máquinas síncronas, com a representação de um único enrolamento amortecedor localizado no eixo  $q$ ; desprezam-se os torques de amortecimento; representam-se as cargas por modelos polinomiais em função da tensão; e descreve-se a rede elétrica de transmissão pela formulação nodal, com a matriz admitância nodal na forma complexa.

## 4.2 Métodos de Solução

Nos algoritmos desenvolvidos neste capítulo, a solução numérica das equações (4.1-4.3) está baseada no esquema *Alternado Entrelaçado Implícito* (AEI) com a utilização do método iterativo *Gradiente Conjugado* (GC) *pré-condicionado* no âmbito do processo de resolução das equações da rede elétrica (4.2). Estes algoritmos constituem-se basicamente do seguinte:



- As equações diferenciais são transformadas em equações algébricas a diferenças, através do método de integração trapezoidal implícito, e resolvidas iterativamente e alternadamente com as equações originalmente algébricas;
- Para cada iteração nas equações diferenciais, a equação da rede elétrica (4.2) é resolvida como se fosse linear, a partir da determinação das injeções de correntes para valores das tensões nodais da iteração anterior;
- As equações da rede elétrica são resolvidas por um método híbrido que combina o método iterativo GC e o método direto baseado na fatoração LU e substituições direta e inversa (algoritmo AEIGC), ou totalmente pelo método iterativo GC pré-condicionado (algoritmo AEIGCRD);
- O teste de convergência é feito nas variáveis de estado, calculando-se um vetor de diferenças entre os valores das iterações atual e anterior e comparando-se a sua norma Euclidiana a uma tolerância especificada.

Portanto, o comportamento dinâmico do SEE no intervalo de tempo  $t - \Delta t$  a  $t$ , passa a ser representado pelas seguintes equações:

$$x(t) = H(x(t), x(t - \Delta t), u(t), u(t - \Delta t)) \quad (4.4)$$

$$I(E(t), V(t)) = YV(t) \quad (4.5)$$

$$u(t) = h(E(t), V(t)) \quad (4.6)$$

onde  $\Delta t$  é o passo de integração e  $H$  é uma função vetorial não-linear que surge da aplicação do método de integração numérica as equações diferenciais.

O esquema AEI acima descrito pode ser sinteticamente representado pelo seguinte algoritmo:

### Algoritmo AEI seqüencial proposto

#### Inicialização

Para  $t = 1, 2, \dots, T$

Calcule  $u^*(t)$  e  $V^*(t)$ , por extrapolação, e faça  $k \leftarrow 0$

Calcule  $x^0(t) = H[x^*(t), u^*(t), x(t - 1), u(t - 1)]$

Enquanto  $\|\Delta x^k(t)\|_2^2 > \epsilon$ , faça:

Obtenha  $E^k(t), I^k(t)$

Resolva  $I(E^k(t), V^k(t)) = YV^{k+1}(t)$  (usando GC)

Obtenha  $u^{k+1}(t) = h(E^k(t), V^{k+1}(t))$

Resolva  $x^{k+1}(t) = H[x^k(t), u^{k+1}(t), x(t-1), u(t-1)]$

Calcule  $\Delta x^{k+1}(t) = |x^{k+1}(t) - x^k(t)|$

Faça  $k \leftarrow k + 1$

Fim-Enquanto

Fim-Para

### 4.3 O Método Gradiente Conjugado Pré-condicionado

O Gradiente Conjugado (GC) é um método iterativo de solução de sistemas lineares simétricos, derivado da família de métodos do subespaço de Krylov, também denominada de métodos projetados [5, 36]. Uma versão básica e seqüencial deste algoritmo, aplicado a equação do tipo  $Ax = b$  e usando-se uma matriz de pré-condicionamento  $M$ , é dada a seguir :

#### Algoritmo GC Pré-condicionado

Escolha  $x^0$

Calcule  $r^0 = b - Ax^0$

Resolva  $M\tilde{r}^0 = r^0$  e faça  $d^0 = \tilde{r}^0$

Para  $k = 0, 1, \dots$ , faça:

$$\alpha^k = -[\tilde{r}^k]^t \cdot [r^k] / [d^k]^t \cdot [A \cdot d^k]$$

$$x^{k+1} = x^k - \alpha^k d^k$$

$$r^{k+1} = r^k + \alpha^k A \cdot d^k$$

$$\text{Resolva } M\tilde{r}^{k+1} = r^{k+1}$$

Se  $\|r^{k+1}\|_2^2 \leq \epsilon$ , então, Pare

Senão, Calcule :

$$\beta^k = [\tilde{r}^{k+1}]^t \cdot [r^{k+1}] / [\tilde{r}^k]^t \cdot [r^k]$$

$$d^{k+1} = \tilde{r}^{k+1} + \beta^k d^k$$

Fim-Se

Fim-Para

Note que o maior esforço computacional neste algoritmo é devido ao cálculo de produtos internos e multiplicações de matriz por vetor. Essa peculiaridade é muito atrativa para o processamento em computadores paralelos-vetoriais. Por outro lado, há a necessidade de se encontrar uma efetiva matriz de pré-condicionamento  $M$ .

A definição de uma matriz de pré-condicionamento  $M$  visando melhorar a razão de convergência de métodos iterativos, deve ser tal que  $M \approx A$  e  $\text{cond}(M^{-1}A) \leq \text{cond}(A)$  [36].

Nos algoritmos GC, o pré-condicionamento é efetuado, em geral, através de modificações no vetor de resíduos  $r$ , a cada iteração, obtidas resolvendo-se um sistema linear algébrico do tipo:

$$M\tilde{r} = r \quad (4.7)$$

Assim, a escolha de  $M$  deve ser uma solução de compromisso tal que  $M$  seja o mais próximo possível de  $A$ , porém, de modo que o sistema auxiliar (4.7) tenha solução relativamente simples. Sob a ótica dos computadores paralelos-vetoriais, é desejável que  $M$  seja do tipo bloco-diagonal, para que a solução de (4.7) possa ser obtida com paralelismo perfeito, e se utilize o máximo possível a capacidade vetorial da máquina na resolução de cada subsistema.

## 4.4 O Algoritmo AEIGC

Neste item descreve-se em detalhes o algoritmo AEIGC desenvolvido para a resolução das equações que representam a dinâmica de SEE. As principais novidades são a utilização de um método híbrido que combina os métodos GC e fatoração LU na resolução das equações da rede elétrica e a paralelização, ou seja, o mapeamento do algoritmo em sistemas computacionais do tipo multiprocessadores

### 4.4.1 Solução da Rede Elétrica pelo Método GC-Fatoração LU

As equações da rede elétrica podem ser ordenadas de modo que a matriz admitância nodal assuma a forma BBDF. Isto é equivalente a decompor a rede elétrica em um conjunto de subredes interligadas somente através de uma única subrede, denominada subrede de interligação ou interconexão.

Supondo a existência de  $q$  subredes, interligadas através da subrede de interligação  $s$ , a equação matricial da rede elétrica assume a forma seguinte:

$$\begin{bmatrix} I_1(E_1, V_1) \\ I_2(E_2, V_2) \\ \vdots \\ I_p(E_q, V_q) \\ I_s(E_s, V_s) \end{bmatrix} = \begin{bmatrix} Y_1 & & & & & \\ & Y_2 & & & & \\ & & \ddots & & & \\ & & & Y_q & & \\ \bar{Y}_1^T & \bar{Y}_2^T & \dots & \bar{Y}_q^T & Y_s & \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_q \\ V_s \end{bmatrix} \quad (4.8)$$

onde  $V_i$  e  $I_i \in C^{n_i}$ ;  $V_s$  e  $I_s \in C^{n_s}$ ;  $n = n_s + \sum_{i=1}^q n_i$ ;

$I_i$  - vetor de injeções de correntes nodais da  $i$ -ésima subrede;

$I_s$  - vetor de injeções de correntes nodais da subrede de interconexão;

$V_i$  - vetor de tensões nodais da  $i$ -ésima subrede;

$V_s$  - vetor de tensões nodais da subrede de interconexão;

$Y_i$  - matriz admitância nodal da  $i$ -ésima subrede;

$Y_s$  - matriz admitância nodal da subrede de interconexão;

$\bar{Y}_i$  - submatriz da matriz admitância nodal associada aos ramos de interconexão entre a  $i$ -ésima subrede e a subrede de interconexão.

A equação (4.8) é resolvida em duas fases: Na primeira, para  $V_s$ , aplicando-se o método de eliminação Gaussiana por blocos e, na segunda, para cada  $V_i$ ,  $i = 1, 2, \dots, q$ , conforme dado a seguir:

**Fase 1: Resolva**

$$\hat{Y}_s V_s = \hat{I}_s \quad (4.9)$$

onde:

$$\hat{Y}_s = Y_s - \sum_{i=1}^q \bar{Y}_i^T Y_i^{-1} \bar{Y}_i \quad (4.10)$$

$$\hat{I}_s = I_s - \sum_{i=1}^q \bar{Y}_i^T Y_i^{-1} I_i \quad (4.11)$$

**Fase 2:** Para  $i = 1, 2, \dots, q$ ; resolva

$$Y_i V_i = I_i - \bar{Y}_i V_i \quad (4.12)$$

Note que na **Fase 2** devem ser resolvidos  $q$  sistemas lineares completamente desacoplados. Então, usando-se  $q$  processadores, pode-se atribuir um sistema desses para cada processador e resolvê-los pelo método direto baseado na fatoração LU de  $Y_i$ . A matriz  $Y_i$  é esparsa e necessita ser refatorada, ao longo da simulação, somente nos instantes de tempo em que ocorrem alterações na configuração da rede elétrica.

A paralelização do processo de resolução da equação (4.8) pode ser melhor explorado resolvendo-se a equação 4.9, **Fase 1**, pelo método GC, conforme descreve-se no item 4.4.3.

#### 4.4.2 Pré-condicionamento

A resolução da equação (4.9) pelo método GC, tem apresentado um comportamento satisfatório do ponto de vista de convergência, embora a matriz  $\hat{Y}_s$  não seja diagonal dominante para a maioria das decomposições estudadas. Contudo, não se pode descartar a possibilidade de um aumento significativo do número de iterações, para uma determinada decomposição. Conforme citado anteriormente e apresentado com maiores detalhes no apêndice D, o número de condição de  $\hat{Y}_s$  pode ser melhorado com o uso de técnicas de pré-condicionamento. Além disso, o efeito do pré-condicionamento no algoritmo GC pode ser representado por modificações no vetor de resíduos, resolvendo-se um sistema linear auxiliar.

Realiza-se, então, uma exploração preliminar do uso dessas técnicas para pré-condicionar a matriz  $\hat{Y}_s$ . O método usado é o do pré-condicionamento por séries truncadas [36], em que o vetor resíduo modificado  $\tilde{r}$ , para cada iteração do método GC, é obtido através da realização de um número fixo de iterações do tipo Gauss-Jacobi, no sistema auxiliar dado por:

$$\hat{Y}_s \tilde{r} = r, \quad (4.13)$$

a partir da condição inicial  $\tilde{r}^0 = 0$ . O número de iterações define a ordem do último termo da série truncada.

### 4.4.3 Paralelização do Algoritmo AEIGC

O algoritmo AEIGC constitui-se dos seguintes passos principais:

1. Extrapolação das tensões nas barras e variáveis de interface ( $u$  e  $V$ );
2. Cálculos na fase de integração numérica implícita;
3. Cálculo das injeções de correntes  $I(E, V)$ ;
4. Resolução das equações de rede elétrica (Fase 1 + Fase 2);
5. Cálculo das variáveis de interface ( $u$ ).

Todos esses passos principais possuem um elevado potencial para propósitos de processamento paralelo, a saber:

- As componentes de  $u$  e  $V$  são extrapoladas independentemente;
- O acoplamento entre as variáveis de estado de máquinas síncronas diferentes, se dá somente através de variáveis da rede elétrica, conforme a modelagem apresentada no capítulo 2. Portanto, as equações diferenciais de máquinas diferentes podem ser integradas simultaneamente e independentemente;
- As componentes de  $u$ ,  $E$  e  $I$ , são funções somente de variáveis da barra à qual estão associadas;
- A **Fase 2** do método de resolução das equações da rede elétrica consiste em resolver  $q$  subsistemas algébricos lineares independentes;
- A **Fase 1** do método de resolução das equações da rede elétrica é a que apresenta maiores dificuldades de paralelização. Contudo ela pode ser realizada com elevado grau de paralelismo através do método GC.

Então, aproveitando-se as peculiaridades da matriz admitância nodal na forma BBDF, as equações (4.1) (4.2) e (4.3) são reescritas na seguinte forma:

$$\left\{ \begin{array}{l} \dot{x}_i = f_i(x_i, u_i) \\ I_i(E_i, V_i) = Y_i V_i - \bar{Y}_i V_s \\ \hat{I}_s(E, V) = \hat{Y}_s V_s \\ u_i = h_i(E_i, V_i) \\ i = 1, 2, \dots, q \end{array} \right. \quad (4.14)$$

onde:

$x_i$  - vetor de variáveis de estado do  $i$ -ésimo subsistema de equações diferenciais;

$f_i$  - função vetorial que define o  $i$ -ésimo subsistema de equações diferenciais;

$u_i$  - vetor de variáveis do  $i$ -ésimo subsistema de equações de interface;

$h_i$  - vetor de funções não-lineares do  $i$ -ésimo subsistema de equações de interface;

$q$  - número de subsistemas;

as demais variáveis já foram definidas no item anterior.

A aplicação do esquema AEI as equações (4.14), faz com que o comportamento dinâmico do SEE passe a ser descrito, no interval  $t - \Delta t$  a  $t$ , pelas seguintes equações:

$$x_i(t) = H_i(x_i(t), u_i(t), x_i(t - \Delta t), u_i(t - \Delta t)) \quad (4.15)$$

$$I_i(E_i(t), V_i(t)) = Y_i V_i(t) - \bar{Y}_i V_s(t) \quad (4.16)$$

$$\hat{I}_s(E(t), V(t)) = \hat{Y}_s V_s(t) \quad (4.17)$$

$$u_i(t) = h_i(E_i(t), V_i(t)) \quad (4.18)$$

$$i = 1, 2, \dots, q$$

Note que no esquema AEI, para cada iteração nas equações (4.15) realiza-se uma e somente uma iteração nas equações (4.16) e, portanto, as equações algébricas não-lineares (4.16) e (4.17), transformam-se em equações algébricas lineares. Sob essas hipóteses, as equações (4.15), (4.16) e (4.18) definem  $q$  subsistemas de equações independentes, porém, dependentes da equação (4.17).

A seguir apresentam-se os detalhes da proposta de paralelização da solução da equação (4.17), através do método GC.

### O Método Gradiente Conjugado Paralelo

O algoritmo GC aplicado a equação (4.9) não requer a formação explícita de  $\hat{Y}_s$ , mas somente o cálculo de  $\hat{Y}_s d^k$ , em cada iteração, e de  $\hat{Y}_s V_s^0$  no cálculo do resíduo  $r^0$ . Esse e outros aspectos associados às características da rede elétrica, podem ser explorados de modo a se obter um algoritmo com elevado grau de paralelismo.

#### Determinação de $\hat{Y}_s d^k$

O cálculo desse produto pode ser feito, a partir de 4.10, conforme os seguintes passos:

$$\text{Calcule } \eta_i = \bar{Y}_i d^k, \quad i = 1, 2, \dots, q \quad (4.19)$$

$$\text{Resolva } Y_i \delta_i = \eta_i, \quad i = 1, 2, \dots, q \quad (4.20)$$

$$\text{Calcule } \omega_i = \bar{Y}_i^T \delta_i, \quad i = 1, 2, \dots, q \quad (4.21)$$

$$\text{Calcule } \hat{Y}_s d^k = Y_s d^k - \sum_{i=1}^q \omega_i \quad (4.22)$$

#### Determinação de $\hat{I}_s$

Esse vetor pode ser calculado de modo similar ao do produto  $\hat{Y}_s d^k$ , a partir de 4.11, a saber:

$$\text{Resolva } Y_i \gamma_i = I_i, \quad i = 1, 2, \dots, q \quad (4.23)$$

$$\text{Calcule } \xi_i = \bar{Y}_i^T \gamma_i, \quad i = 1, 2, \dots, q \quad (4.24)$$

$$\text{Calcule } \hat{I}_s = I_s - \sum_{i=1}^q \xi_i \quad (4.25)$$

Somente os passos definidos pelas equações (4.22) e (4.25) necessitam de comunicação de dados; os demais podem ser feitos com paralelismo perfeito em  $q$  processadores. Note que os fatores da decomposição LU de  $Y_i$ ,  $i = 1, 2, \dots, q$ , requeridos na Fase 2, equação (4.12), podem ser usados para resolver os sistemas lineares definidos nas equações (4.20) e (4.23)



## Matrizes $\Omega_i$ e $\Phi_i$

Outra alternativa para explorar o paralelismo do método GC aplicado a equação (4.9), é através da definição das matrizes  $\Omega_i$  e  $\Phi_i$ ,  $i = 1, 2, \dots, q$ , apresentado a seguir.

As matrizes  $\Omega_i$  são definidas de (4.10) como

$$\Omega_i = \bar{Y}_i^T Y_i^{-1} \bar{Y}_i, \quad i = 1, 2, \dots, q \quad (4.26)$$

Essas matrizes são normalmente esparsas e seus elementos dependem dos parâmetros da rede e da decomposição. Contudo, eles permanecem constantes nos períodos de simulação compreendidos entre cada dois instantes de chaveamento. Os elementos não nulos de cada matriz  $\Omega_i$ ,  $i = 1, 2, \dots, q$ , formam uma matriz quadrada, cheia e de dimensão igual ao número de barras da subrede de interconexão diretamente conectadas a  $i$ -ésima subrede. Essas matrizes são denominadas, neste trabalho, de matrizes  $\Phi_i$ ,  $i = 1, 2, \dots, q$ . Os elementos de  $\Omega_i$  e  $\Phi_i$ ,  $i = 1, 2, \dots, q$ , são determinados, basicamente, em dois passos, como segue:

$$\text{Calcule } G_i = \bar{Y}_i^T Z_i, \quad i = 1, 2, \dots, q \quad (4.27)$$

$$\text{Calcule } \Omega_i = G_i \bar{Y}_i, \quad i = 1, 2, \dots, q \quad (4.28)$$

onde  $Z_i$  é a  $i$ -ésima coluna da matriz inversa de  $Y_i$

Note que as operações indicadas em (4.27) e (4.28), podem ser realizadas com paralelismo perfeito. A equação (4.28) é trivial, pois requer o produto de duas matrizes, sendo  $\bar{Y}_i$  muito esparsa. Em (4.27), devido ao elevado grau de esparsidade de  $\bar{Y}_i^T$ , somente um pequeno número de colunas das matrizes  $Z_i$  necessitam ser determinadas. Essa colunas são obtidas resolvendo-se os seguintes sistemas lineares

$$Y_i^{-1} Z_i^k = e^k, \quad k = 1, 2, \dots, ns \quad (4.29)$$

onde  $Z_i^k$  é a  $k$ -ésima coluna da matriz  $Z_i$ ;  $e^k$  é um vetor com elementos nulos, exceto o  $k$ -ésimo elemento que é igual a 1; e  $ns$  é o número de barras da  $i$ -ésima subrede diretamente conectadas a subrede de interconexão. Em geral, para uma cuidadosa decomposição,  $ns$  é muito menor que o número de barras da  $i$ -ésima subrede.

O produto  $\hat{Y}_i d^k$  é, então, dado por:

$$\hat{Y}_s d^k = Y_s d^k - \sum_{i=1}^q \Omega_i d^k \quad (4.30)$$

Todas as operações envolvendo a determinação dos elementos de  $\Omega_i$  e os produtos  $\Omega_i d^k$ ,  $i = 1, 2, \dots, q$ , são realizados com paralelismo perfeito.

### O Algoritmo AEIGC Paralelo

O esquema de paralelização proposto pode ser sintetizado no algoritmo apresentado a seguir, onde a variável  $i$  refere-se à  $i$ -ésima tarefa ( $i = 1, 2, \dots, q$ ).

#### Inicialização

Para  $t = 1, 2, \dots, T$

Para  $i = 1, 2, \dots, q$

Calcule  $u_i^*(t)$  e  $V_s^*(t)$ , por extrapolação

Calcule  $x_i^0(t) = H_i[x_i^*(t), u_i^*(t), x_i(t-1), u_i(t-1)]$

Faça:  $k \leftarrow 0$

Enquanto  $\Delta x_i^k(t) \geq \epsilon$ , para algum  $i$ , faça:

Calcule  $E_i^k(t)$ ,  $I_i^k(t)$ ,  $E_s^k(t)$ , e  $I_s^k(t)$

Resolva  $\hat{I}_s^k(t) = \hat{Y}_s^k V_s^{k+1}(t)$  (pelo GC)

Resolva  $I_i^k(t) = Y_i V_i^{k+1}(t) - \bar{Y}_i V_s^{k+1}(t)$

Calcule  $u_i^{k+1}(t) = h_i(E_i^k(t), V^{k+1}(t))$

Calcule  $x_i^{k+1}(t) = H_i[x_i^k(t), u_i^{k+1}(t), x_i(t-1), u_i(t-1)]$

Calcule  $\Delta x_i^{k+1}(t) = \|x_i^{k+1}(t) - x_i^k(t)\|$

Faça :  $k \leftarrow k + 1$

Fim-Enquanto

Fim-Para

Fim-Para

Os passos mais relevantes desse algoritmo, em termos de esforço computacional, são constituídos de  $q$  tarefas que podem ser realizadas simultaneamente. Assim, é possível organizá-lo como  $q$  algoritmos seqüenciais síncronos que podem ser processados concorrentemente em  $q$  processadores, trocando dados em passos pré-determinados. Uma visualização global e sintética deste algoritmo paralelo pode ser obtida através do fluxograma da figura 4.1. Este fluxograma descreve os cálculos necessários para a realização de um passo de integração numérica na SDSEE. Cada ramo desse fluxograma

é um algoritmo seqüencial que será associado a um determinado processador do computador paralelo. O número de ramos deve ser igual ao número de processadores que serão utilizados, ou seja, igual ao número de subsistema em que é decomposto o SEE. Note que em determinados pontos, os ramos seqüenciais se interconectam. Esses são pontos em que é necessária a troca de dados entre os processadores, para a continuidade do processamento, e são denominados instantes ou pontos de comunicação. Obsevando-se a figura 4.1, verifica-se a existência de cinco pontos de comunicação, sendo o último destinado a troca de sinalizadores para o controle da convergência global do algoritmo paralelo. Nos outros quatro pontos de comunicação, realizam-se a troca de vetores de dados, descritos a seguir:

$I_{gj}$  : subvetores do vetor de injeções de correntes dos geradores  $I_g$ , necessário devido à desvinculação proposital das decomposições da rede elétrica e do sistema de equações diferenciais, a fim de minimizar o custo de comunicação. A dimensão de  $I_g$  é igual ao número de geradores representados dinamicamente e independe do número de variáveis de estado usadas para representar as máquinas síncronas e seus controladores;

$\omega_j^0, \xi_j^0$  : vetores de resultados parciais no cálculo do resíduo para o algoritmo GC. Ambos têm dimensão igual ao número de barras da subrede de interconexão;

$\omega_j^k$  : vetores de resultados parciais para o algoritmo GC e com dimensão igual a dos vetores  $\omega_j^0$ ;

$V_{gj}$  : subvetor do vetor das tensões terminais dos geradores  $V_g$ , com dimensões iguais as de  $I_{gj}$ .

### Características do Algoritmo AEIGC Paralelo

A escolha das  $(q + 1)$  subredes que definem a forma BBDF para a matriz  $Y$ , tem influência significativa no algoritmo AEIGC paralelo. Ela afeta a sua eficiência em muitos aspectos, a saber:

**Balanceamento de Carga** : As primeiras  $q$  subredes devem ser definidas tal que, nos passos do algoritmo em que são processados simultaneamente, impliquem em esforço computacional razoavelmente equilibrado entre os processadores.

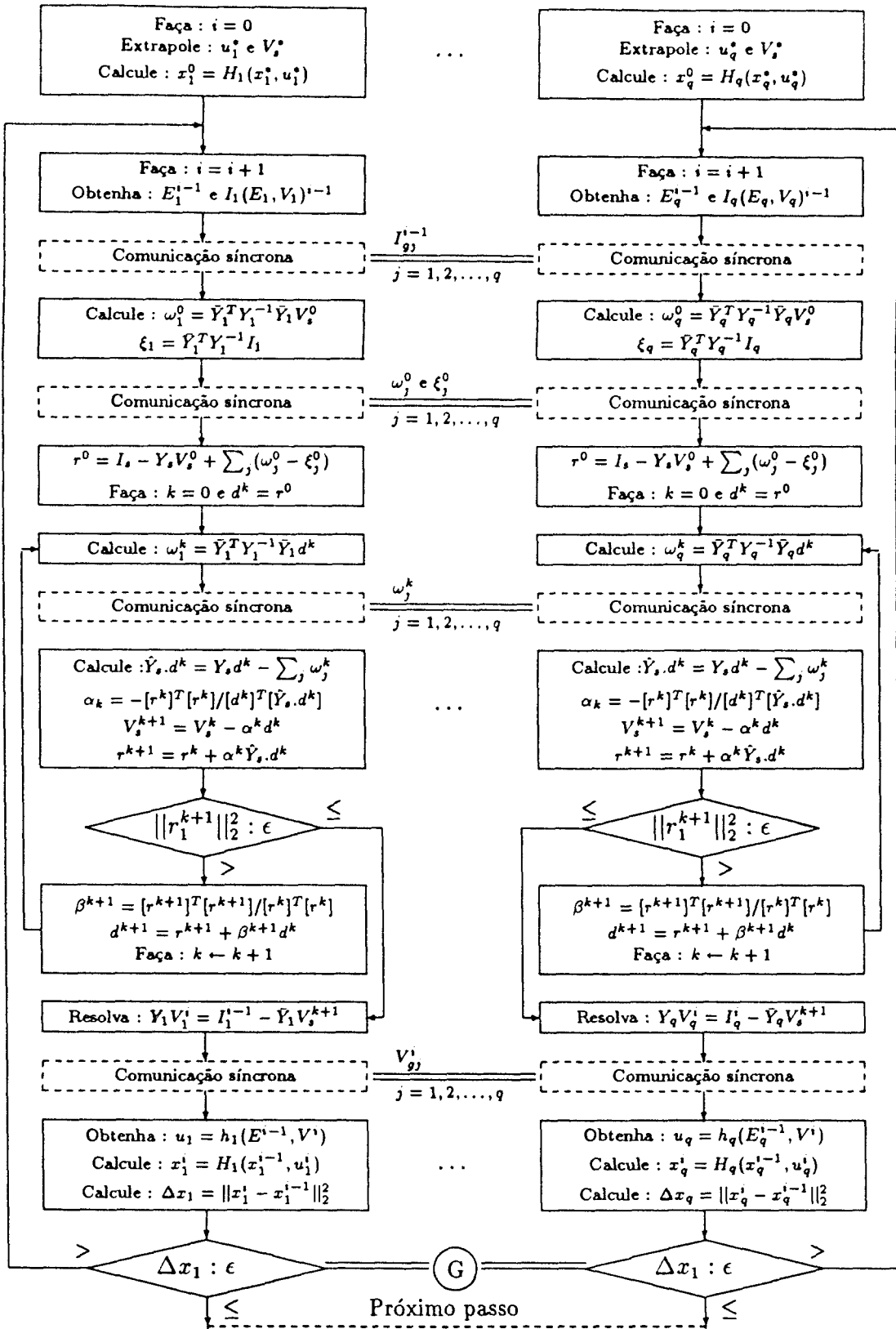


Figura 4.1: Fluxograma do algoritmo AEIGC paralelo para um passo de integração

Caso contrário, alguns processadores ficarão bloqueados por um longo tempo. É importante observar que o esforço computacional para cada subrede depende de suas características, tais como: número de nós, número de ramos, etc.;

**Dimensão de  $\hat{Y}_s$  :** A Fase 1 do algoritmo de solução das equações da rede elétrica é o passo crucial do algoritmo AEIGC e, então, manter a dimensão de  $\hat{Y}_s$  tão pequena quanto possível, é um dos objetivos principais no processo de decomposição da rede elétrica;

**Razão de Convergência :** A decomposição da rede elétrica define o número de condição de  $\hat{Y}_s$ , que influi na razão de convergência do método GC. Contudo, esse número de condição pode ser melhorado através de técnicas de pré-condicionamento, conforme discutido anteriormente.

## 4.5 O Algoritmo AEIGCRD

O tema deste item representa uma continuação da investigação sobre a aplicabilidade do método *Gradiente Conjugado* (GC), no âmbito do esquema AEI para a SDSEE em computadores paralelos. Especificamente, propõe-se o emprego do método Gradiente Conjugado pré-condicionado na resolução de todas as equações algébricas associadas a rede elétrica. Os apelos principais para essa proposta são a independência de uma decomposição da rede elétrica para a forma BBDF e a aplicação do método GC em sistemas com dimensões maiores que as propiciadas no algoritmo AIEGC, de modo a permitir a exploração do elevado potencial de vetorização desse método. Por outro lado, os principais desafios são a definição de um efetivo método de pré-condicionamento e o mapeamento do algoritmo em computadores paralelos.

### 4.5.1 Pré-condicionamento

Inicialmente, foram utilizadas algumas técnicas de pré-condicionamento encontradas na literatura, tais como: a) pré-condicionamento diagonal [5, 36] e b) séries truncadas de Neumann [5, 36] e Maclaurin [17]. Embora a simplicidade e as potencialidades para o paralelismo e vetorização, essas técnicas não foram efetivas em sistemas de grande porte. Então, optou-se pela proposição de uma nova versão de pré-condicionamento do tipo fatoração LU incompleta, decrita a seguir, que permite a resolução do sistema auxiliar (4.7) com paralelismo perfeito.

Supondo que a rede elétrica constitua-se de  $q$  subredes interligadas, a equação (4.2) linearizada pode ser reescrita, simbolicamente, como segue:

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_q \end{bmatrix}^k = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1q} \\ Y_{21} & Y_{22} & \dots & Y_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{q1} & Y_{q2} & \dots & Y_{qq} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_q \end{bmatrix}^{k+1} \quad (4.31)$$

Propõe-se, então, que a matriz de pré-condicionamento  $M$  seja constituída somente dos blocos diagonais de  $Y$ . Assim, o sistema auxiliar (4.7) assume a forma seguinte:

$$\begin{bmatrix} Y_{11} & & & \\ & Y_{22} & & \\ & & \ddots & \\ & & & Y_{qq} \end{bmatrix} \begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_2 \\ \vdots \\ \tilde{r}_q \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_q \end{bmatrix} \quad (4.32)$$

Note que este sistema auxiliar pode ser resolvido através das técnicas de fatoração LU de matrizes esparsas e com paralelismo perfeito. Entre elas destaca-se a metodologia que utiliza a matriz de fatores inversos  $W$  [37], que propicia significativos ganhos quando empregam-se técnicas de vetorização [10].

Faz-se, ainda, os seguintes comentários:

1. Uma parcela significativa dos cálculos poderá ser realizada vetorialmente;
2. A matriz  $Y$  é constante entre chaveamentos e, então, somente os blocos diagonais de  $M$  afetados pelas alterações na configuração, necessitam ser redefinidos nos instantes de chaveamento, assim como também os correspondentes fatores LU. Além disso, poderá ser estudada a efetividade da definição de  $M$  uma única vez ao longo de toda a simulação, ou seja, para a configuração do período pré-perturbação;
3. É mantido o desacoplamento na resolução das equações diferenciais associadas às máquinas elétricas e seus controladores.

### 4.5.2 Paralelização do Algoritmo AEIGCRD

O algoritmo AEIGCRD, semelhantemente ao algoritmo AEIGC, constitui-se dos seguintes passos principais:

1. Extrapolação das tensões nas barras e variáveis de interface ( $u^*(t)$  e  $V^*(t)$ );
2. Cálculos de  $x^0(t)$  e  $x^{k+1}(t)$  usando as funções algébricas a diferenças, oriundas das equações diferenciais;
3. Cálculo das injeções de correntes  $I(E^k, V^k)$ ;
4. Resolução das equações linearizadas da rede elétrica  $I(E^k, V^k) = Y.V^{k+1}$ ;
5. Cálculo das variáveis de interface  $u^k(t)$ .

Assim, a paralelização do algoritmo AEIGCRD para o processamento, por exemplo em  $q$  processadores, significa essencialmente decompor cada um desses passos em  $q$  subtarefas.

### Decomposição do Sistema

Inicialmente, faz-se as seguintes observações em relação aos passos principais do algoritmo:

- As componentes de  $u(t)$  e  $V(t)$  são extrapoladas independentemente;
- O acoplamento entre as variáveis de estado de máquinas síncronas diferentes, se dá somente através de variáveis da rede elétrica, conforme a modelagem apresentada anteriormente. Portanto, as equações diferenciais de máquinas diferentes podem ser integradas simultaneamente e independentemente;
- As componentes de  $u(t)$ ,  $E(t)$  e  $I(t)$ , são funções somente de variáveis da barra à qual estão associadas;
- No algoritmo GC, a predominância do esforço computacional está restrita a três tipos de operações básicas: a) somas de vetor por vetor multiplicado por um escalar; b) produtos internos; c) produtos de matriz por vetor. Em princípio, a decomposição desses vetores e matrizes pode ser feita trivialmente;
- Os sistemas auxiliares de pré-condicionamento, usando-se a matriz  $M$  bloco diagonal proposta, tem decomposição trivial e a resolução dos subsistemas é completamente desacoplada.

Diante dessas observações não é difícil concluir que o algoritmo AEIGCRD tem elevado potencial para a paralelização, contudo a matriz  $M$  é extraída de  $Y$  (matriz admitância nodal) e neste processo deve-se atender ao seguintes requisitos:

- É necessário que os blocos diagonais de  $Y$  sejam fracamente acoplados entre si (bloco diagonal dominância) para que  $M$  seja efetiva como pré-condicionador. Esta forma de  $Y$  é conhecida na literatura como NBDF (“Near Block Diagonal Form”) e é equivalente a decompor a rede elétrica em subredes fracamente acopladas entre si.
- O número de subredes é fixado a priori em valor igual ao número de processadores que se quer utilizar;
- O número de interligações entre as subredes deve ser o mínimo possível, a fim de minimizar o volume da comunicação de dados entre os processadores na realização dos produtos do tipo matriz por vetor, no algoritmo GC;
- As características das subredes, em termos de dimensão e complexidade, devem ser tais que a solução do sistema auxiliar de pré-condicionamento e os produtos do tipo matriz por vetor, possam ser feitos com um aceitável balanceamento da carga entre os processadores.

## O Algoritmo AEIGCRD Paralelo

O esquema de paralelização proposto pode ser sintetizado no algoritmo apresentado a seguir, onde a variável  $i$  refere-se a  $i$ -ésima tarefa ( $i = 1, 2, \dots, q$ ):

### Inicialização

Para  $t = 1, 2, \dots, T$

Para  $i = 1, 2, \dots, p$

Calcule  $u_i^*(t)$  e  $V_i^*(t)$ , por extrapolação

Calcule  $x_i^0(t) = H_i[x_i^*(t), u_i^*(t), x_i(t-1), u_i(t-1)]$

Faça:  $k \leftarrow 0$

Enquanto  $\Delta x_i^k(t) \geq \epsilon$ , para algum  $i$ , faça:

Calcule  $E_i^k(t)$  e  $I_i^k(t)$

Resolva  $I^k(t) = YV^{k+1}(t)$  (pelo GC)



Calcule  $u_i^{k+1}(t) = h_i(E_i^k(t), V^{k+1}(t))$

Calcule  $x_i^{k+1}(t) = H_i[x_i^k(t), u_i^{k+1}(t), x_i(t-1), u_i(t-1)]$

Calcule  $\Delta x_i^{k+1}(t) = \|x_i^{k+1}(t) - x_i^k(t)\|$

Faça:  $k \leftarrow k + 1$

*Fim-Enquanto*

*Fim-Para*

*Fim-Para*

O fluxograma da figura 4.2 permite uma visualização global do algoritmo AEIG-CRD paralelo, principalmente no que se refere aos requisitos de comunicação de dados entre os processadores. A fase de resolução paralela das equações da rede elétrica, representada em um retângulo tracejado ( $I(E^k, V^k) = Y.V^{k+1}$ ), é detalhada no fluxograma da figura 4.3. Os demais retângulos tracejados são usados para indicar a forma de comunicação de dados entre os processadores.

Observam-se nestes dois fluxogramas, oito pontos de comunicação de dados e sincronização entre os processadores, a saber:

$I_g$  e  $V_g$  : Respectivamente, injeções de correntes e tensões nas barras que contém geradores representados dinamicamente. A dimensão desses vetores independe do número de variáveis de estado usadas para representar as máquinas síncronas e seus controladores;

$V_f$  : Tensões nas barras de fronteira dos subsistemas da rede elétrica;

$G$  : Testes de convergência global;

$RR^0$ ,  $RR^{k+1}$  e  $DYD^k$  : Escalares resultantes dos subprodutos internos, realizados em cada processador, e cujo somatório para todos os processadores, formará o resultado global dos produtos internos do algoritmo GC;

$d_f^k$  : Subvetor do vetor de direções global, contendo somente os valores associados as barras de fronteira dos subsistemas da rede elétrica e com dimensão igual a de  $V_f$ ;

Os vetores  $I_g$ ,  $V_g$ ,  $V_f$  e  $d_f$ , têm dimensões, em geral, da ordem de 10% da dimensão total do sistema de equações da rede elétrica. Por outro lado,  $G$ ,  $RR^0$ ,  $RR^{k+1}$  e  $DYD^k$  são escalares e, conseqüentemente, o custo para a comunicação dos mesmos independe

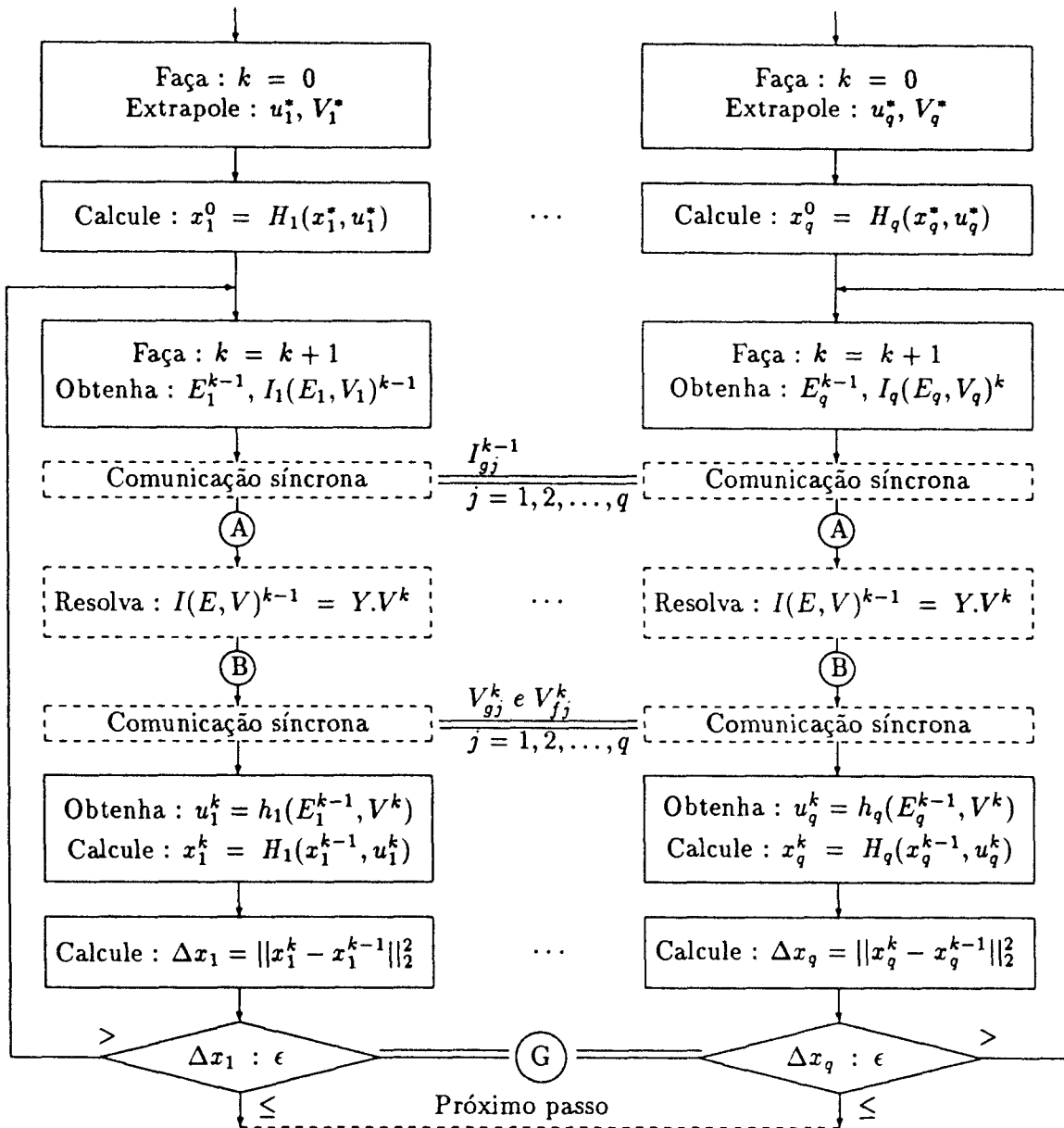


Figura 4.2: Fluxograma do algoritmo AEIGCRD paralelo para um passo de integração

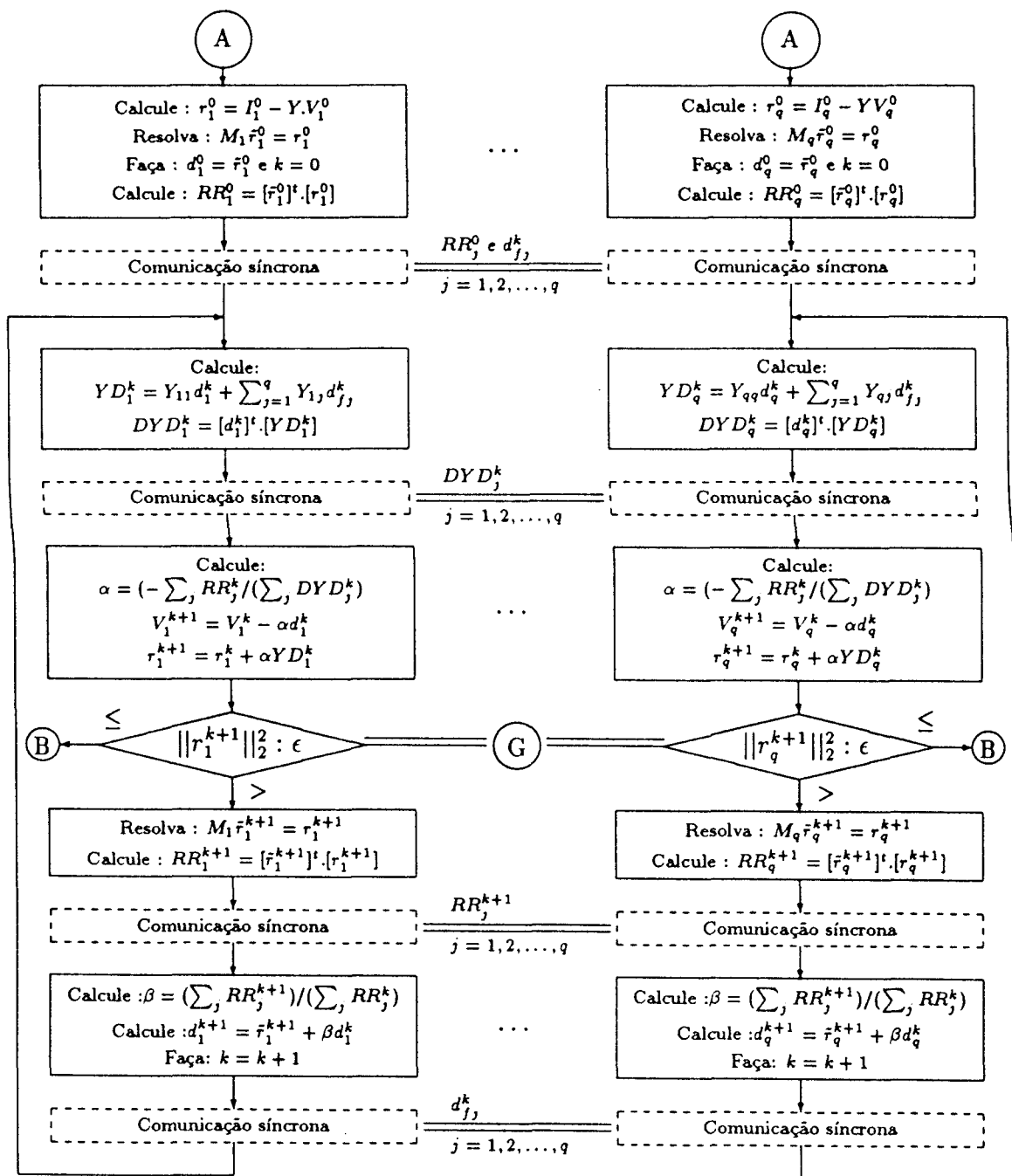


Figura 4.3: Fluxograma do algoritmo GC paralelo para resolver  $I(E^k, V^k) = Y.V^{k+1}$  AEIGCRD

da dimensão do sistema a ser resolvido. Outro detalhe importante é que a sofisticação na representação dinâmica das máquinas elétricas e seus controladores, através de um maior número de equações diferenciais, não requer aumento algum no volume de comunicação de dados e contribui para uma maior eficiência no processamento paralelo. Em função dessas características, espera-se que o algoritmo AEIGCRD apresente melhores índices de “speed up” e eficiências em sistemas elétricos com dimensões elevadas e representados por modelos sofisticados.

## 4.6 Conclusões

Neste capítulo foram descritos dois novos algoritmos paralelos, baseados no esquema *Alternado Entrelaçado Implícito* de resolução das EDAs que representam a dinâmica de um SEE. Os algoritmos foram denominados, respectivamente, AEIGC e AEIGCRD, e visam a servir como meio de utilização eficiente de computadores paralelos, do tipo multiprocessadores, na SDSEE. No caso do algoritmo AEIGC, a principal novidade é a resolução das equações da rede elétrica por um método híbrido, a partir da matriz admitância nodal na forma BBDF. O algoritmo tal como foi proposto, permite que a maioria dos cálculos seja feita em subtarefas processáveis simultaneamente e com pequeno volume de comunicação de dados entre essas subtarefas. Propôs-se também uma forma de otimizar os cálculos envolvidos no algoritmo GC, através da definição de novas matrizes denominadas  $\Omega_i$  e  $\Phi_i$ , cujos elementos permanecem constantes entre dois instantes subsequentes de chaveamento na rede elétrica. As matrizes  $\Omega_i$  são muito esparsas e as matrizes  $\Phi_i$  têm dimensões muito pequenas. Foram ainda discutidos alguns aspectos da decomposição do sistema, relacionados com o balanceamento da carga, dimensão dos subsistemas de interconexão e pré-condicionamento; este último diretamente relacionado com a razão de convergência do algoritmo GC.

No que se refere ao algoritmo AEIGCRD, as principais contribuições são a extensão da aplicação do método GC para resolver todas as equações da rede elétrica e a proposição de um pré-condicionador do tipo fatoração LU incompleta. A matriz de pré-condicionamento  $M$  é formada por blocos diagonais esparsos. Dessa forma contribui-se para a exploração do elevado potencial de vetorização do método GC e o sistema auxiliar de pré-condicionamento pode ser resolvido com paralelismo perfeito.

# Capítulo 5

## Esquema Simultâneo Implícito - Paralelização no Tempo

Neste capítulo descreve-se um novo algoritmo paralelo para a resolução das equações que representam o comportamento dinâmico de SEE, usando-se o esquema *Simultâneo Implícito* aplicado a vários passos de integração tomados simultaneamente. Especificamente, empregam-se: o método trapezoidal implícito para discretizar as equações diferenciais, transformando-as em equações algébricas a diferenças; o método de Newton-Raphson para resolver as equações não-lineares algébricas a diferenças e algébricas; os métodos iterativos Gradiente Biconjugado (BCG) e Bi-CGSTAB para resolver as equações lineares Jacobianas; e o método da série de Taylor para inicializar as variáveis da janela de integração (passos de integração tomados simultaneamente). Para o algoritmo resultante, denominado SIMPT, é proposta uma forma de mapeamento (paralelização) em sistemas computacionais do tipo multiprocessadores. Também são desenvolvidos métodos de pré-condicionamento e pós-condicionamento explícito do tipo fatoração LU incompleta.

### 5.1 Modelagem Matemática

Conforme descrito em detalhes no capítulo 2, a dinâmica de um SEE é representada matematicamente por um conjunto de equações diferenciais ordinárias não-lineares, associadas aos rotores das máquinas síncronas e seus controladores, e um conjunto de equações algébricas não-lineares, associadas a rede de transmissão, estatores das máquinas síncronas (geradores) e cargas representadas por modelos estáticos. Elas podem ser expressas na forma seguinte [1]:

$$\dot{x} = f(x, z) \quad (5.1)$$

$$0 = g(x, z) \quad (5.2)$$

onde  $f$  é uma função vetorial que define as equações diferenciais;  $g$  é uma função vetorial que define as equações algébricas;  $x$  é o vetor das variáveis de estado; e  $z$  é o vetor de variáveis das equações algébricas.

No esquema *Simultâneo Implícito* (SI), as equações diferenciais (5.1) são transformadas em equações algébricas a diferenças, através do método de integração trapezoidal implícito, e reunidas às equações originalmente algébricas (5.2), para constituírem um único sistema de equações algébricas não-lineares. Estas equações passam a representar o comportamento dinâmico do SEE no intervalo de tempo de um passo de integração e sua resolução é comumente realizada por um método tipo Newton [1, 32, 35]. Recentemente, com o advento dos computadores paralelos, têm crescido o interesse por esquemas de resolução que realizam a discretização das equações diferenciais para vários passos de integração simultaneamente [15, 24, 25]. O objetivo é explorar o paralelismo existente entre os passos de integração de modo a reduzir os tempos computacionais envolvidos nas simulações.

Nos dois próximos subitens serão apresentados, pela seqüência, a modelagem matemática do esquema simultâneo para a discretização de um único passo de integração e para  $p$  passos de integração tomados simultaneamente.

### 5.1.1 Equações para um Único Passo de Integração

Conforme descrito no capítulo 2, no intervalo de tempo de um passo de integração,  $t - \Delta t$  a  $t$ , o SEE é representado pela equação (2.10), reescrita a seguir:

$$\begin{cases} F = x(t) - x(t - \Delta t) - [f(x(t), z(t)) + f(x(t - \Delta t), z(t - \Delta t))] \frac{\Delta t}{2} = 0 \\ G = g(x(t), z(t)) = 0 \end{cases} \quad (5.3)$$

O sistema de equações (5.3) pode ser representado na seguinte forma compacta:

$$H(y(t)) = 0, \quad (5.4)$$

onde  $y = [x, z]^T$  e  $H = [F, G]^T$ .

A resolução de (5.4) pelo método de Newton, para o instante de tempo  $t$ , requer sucessivas soluções de equações lineares do tipo:

$$H(y^k) = J(y^k) \cdot \Delta y^{k+1} \quad (5.5)$$

$$y^{k+1} = y^k + \Delta y^{k+1} \quad (5.6)$$

onde o sobrescrito  $k$  indica a iteração de Newton Raphson e  $J$  é a matriz Jacobiana definida a seguir.

$$J = - \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \quad (5.7)$$

com  $J_1 = \frac{\partial F_1}{\partial x}$ ,  $J_2 = \frac{\partial F_1}{\partial z}$ ,  $J_3 = \frac{\partial F_2}{\partial x}$ ,  $J_4 = \frac{\partial F_2}{\partial z}$ .

### 5.1.2 Equações para $p$ Passos de Integração

O processo de algebrização das equações diferenciais (5.1), para  $p$  passos de integração é semelhante ao apresentado no item anterior. Assim, o comportamento dinâmico do SEE, no intervalo de tempo coberto por esses  $p$  passos, pode ser representado sucintamente pela seguinte equação matricial [15, 24, 25].

$$\hat{H}(\hat{y}) = 0 \quad (5.8)$$

onde:

$\hat{y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_p]^T$  : vetor de variáveis de estado e algébricas;

$\hat{H} = [\hat{H}_1, \hat{H}_2, \dots, \hat{H}_p]^T$  : vetor de funções algébricas e algébricas a diferenças não-lineares.

Neste trabalho, propõe-se que a resolução de (5.8) seja feita através do método de Newton-Raphson, pelas razões de suas características de convergência e da experiência acumulada em vários anos de sua aplicação ao esquema *Simultâneo Implícito* tradicional. Conseqüentemente, a solução de (5.8) será obtida a partir de sucessivas soluções de sistemas lineares do tipo:

$$\hat{H}(\hat{y}^k) = \hat{J}(\hat{y}^k) \cdot \Delta \hat{y}^{k+1} \quad (5.9)$$

$$\hat{y}^{k+1} = \hat{y}^k + \Delta \hat{y}^{k+1} \quad (5.10)$$

onde  $\hat{J} = -\frac{\partial \hat{H}}{\partial \hat{y}}$

Para  $p$  elevado e SEE reais, a matriz Jacobiana  $\hat{J}$  será de muito grande porte. É, então, de fundamental importância discutir em detalhes a sua estrutura, antes de se escolher um método numérico de solução para as equações (5.9).

### Estrutura Matricial Detalhada da Equação Jacobiana

Define-se:

**Para o  $t$ -ésimo passo de integração**

$\hat{x}_t = [x_1, x_2, \dots, x_m]^T$  : vetor de variáveis de estado;

$\hat{V}_t^e = [V_1^e, V_2^e, \dots, V_n^e]^T$  : vetor de tensões nodais;

$\hat{y}_t = [\hat{x}_t, \hat{V}_t^e]^T$  : vetor de variáveis do sistema;

$\hat{H}_t = [\hat{F}_t, \hat{G}_t]^T$  : vetor de funções algébricas não-lineares;

$\hat{F}_t = [F_1, F_2, \dots, F_m]^T$  : vetor de funções algébricas a diferenças não-lineares;

$\hat{G}_t = [G_1, G_2, \dots, G_n]^T$  : vetor de funções algébricas não-lineares;

**Para  $p$  passos de integração tomados simultaneamente**

$\hat{x} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_p]^T$  : vetor de variáveis de estado;

$\hat{V}^e = [\hat{V}_1^e, \hat{V}_2^e, \dots, \hat{V}_p^e]^T$  : vetor de tensões nodais;

$\hat{F} = [\hat{F}_1, \hat{F}_2, \dots, \hat{F}_p]^T$  : vetor de funções algébricas de diferenças não-lineares;

$\hat{G} = [\hat{G}_1, \hat{G}_2, \dots, \hat{G}_p]^T$  : vetor de funções algébricas não-lineares;

Usando-se essas definições, o sistema (5.8) pode ser reescrito na seguinte forma:

$$\begin{cases} \hat{F}(\hat{x}, \hat{V}^e) = 0 \\ \hat{G}(\hat{x}, \hat{V}^e) = 0 \end{cases} \quad (5.11)$$



ou, mais explicitamente, tal como:

$$\begin{cases} \hat{F}_t(\hat{x}_t, \hat{x}_{t-1}, \hat{V}_t^e, \hat{V}_{t-1}^e) = 0 \\ \hat{G}_t(\hat{x}_t, \hat{V}_t^e) = 0 \quad t = 1, 2, \dots, p \end{cases} \quad (5.12)$$

Supondo-se que cada máquina síncrona e seus respectivos controladores sejam representados por  $\kappa$  equações diferenciais, o sistema linear Jacobiano para (5.12) será constituído por  $(\kappa m + 2n).p$  equações. A representação simbólica desse sistema linear, bem como a de suas submatrizes, é dada a seguir.

$\hat{F}_1$	$\frac{\partial \hat{F}_1}{\partial \hat{x}_1}$	$\frac{\partial \hat{F}_1}{\partial \hat{V}_1^e}$			$\Delta \hat{x}_1$
	$\frac{\partial \hat{G}_1}{\partial \hat{x}_1}$	$Y^e - \frac{\partial \hat{G}'_1}{\partial \hat{V}_1^e}$			
$\hat{F}_2$	$\frac{\partial \hat{F}_2}{\partial \hat{x}_1}$	$\frac{\partial \hat{F}_2}{\partial \hat{V}_1^e}$	$\frac{\partial \hat{F}_2}{\partial \hat{x}_2}$	$\frac{\partial \hat{F}_2}{\partial \hat{V}_2^e}$	$\Delta \hat{x}_2$
$\hat{G}_2$	0	0	$\frac{\partial \hat{G}_2}{\partial \hat{x}_2}$	$Y^e - \frac{\partial \hat{G}'_2}{\partial \hat{V}_2^e}$	
$\vdots$	$\dots$				$\vdots$
$\vdots$	$\dots$				
$\hat{F}_p$	$\frac{\partial \hat{F}_p}{\partial \hat{x}_{p-1}}$	$\frac{\partial \hat{F}_p}{\partial \hat{V}_{p-1}^e}$	$\frac{\partial \hat{F}_p}{\partial \hat{x}_p}$	$\frac{\partial \hat{F}_p}{\partial \hat{V}_p^e}$	$\Delta \hat{x}_p$
$\hat{G}_p$	0	0	$\frac{\partial \hat{G}'_p}{\partial \hat{x}_p}$	$Y^e - \frac{\partial \hat{G}_p}{\partial \hat{V}_p^e}$	

(5.13)

onde:

$$\frac{\partial \hat{F}_t}{\partial \hat{x}_t} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & & & \\ & \frac{\partial F_2}{\partial x_2} & & \\ & & \dots & \\ & & & \frac{\partial F_m}{\partial x_m} \end{bmatrix}_{(\kappa m, \kappa m)} \quad (5.14)$$

$$\frac{\partial \hat{F}_t}{\partial \hat{V}_t^e} = \left[ \frac{\partial F_1}{\partial V^e} \quad \frac{\partial F_2}{\partial V^e} \quad \dots \quad \frac{\partial F_m}{\partial V^e} \right]_{(\kappa m, 2n)}^T \quad (5.15)$$

$$\frac{\partial \hat{G}_t}{\partial \hat{x}_t} = \left[ -\frac{\partial G'}{\partial x_1} \quad -\frac{\partial G'}{\partial x_2} \quad \dots \quad -\frac{\partial G'}{\partial x_m} \right]_{(2n, \kappa m)} \quad (5.16)$$

$$\frac{\partial \hat{F}_t}{\partial \hat{x}_{t-1}} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & & & \\ & \frac{\partial F_2}{\partial x_2} & & \\ & & \dots & \\ & & & \frac{\partial F_m}{\partial x_m} \end{bmatrix}_{(\kappa m, \kappa m)} \quad (5.17)$$

$$\frac{\partial \hat{F}_t}{\partial \hat{V}_{t-1}^e} = \left[ \frac{\partial F_1}{\partial V^e} \quad \frac{\partial F_2}{\partial V^e} \quad \dots \quad \frac{\partial F_m}{\partial V^e} \right]_{(\kappa m, 2n)}^T \quad (5.18)$$

$$Y^e = \begin{array}{|c|c|c|c|} \hline \begin{array}{cc} -b_{11} & g_{11} \\ g_{11} & b_{11} \end{array} & \begin{array}{cc} -b_{12} & g_{12} \\ g_{12} & b_{12} \end{array} & \begin{array}{cc} \dots & \dots \end{array} & \begin{array}{cc} -b_{1n} & g_{1n} \\ g_{1n} & b_{1n} \end{array} \\ \hline \begin{array}{cc} -b_{21} & g_{21} \\ g_{21} & b_{21} \end{array} & \begin{array}{cc} -b_{22} & g_{22} \\ g_{22} & b_{22} \end{array} & \begin{array}{cc} \dots & \dots \end{array} & \begin{array}{cc} -b_{2n} & g_{2n} \\ g_{2n} & b_{2n} \end{array} \\ \hline \begin{array}{cc} \vdots & \vdots \\ \vdots & \vdots \end{array} & \begin{array}{cc} \vdots & \vdots \\ \vdots & \vdots \end{array} & \begin{array}{cc} \ddots & \ddots \end{array} & \begin{array}{cc} \vdots & \vdots \\ \vdots & \vdots \end{array} \\ \hline \begin{array}{cc} -b_{n1} & g_{n1} \\ g_{n1} & b_{n1} \end{array} & \begin{array}{cc} -b_{n2} & g_{n2} \\ g_{n2} & b_{n2} \end{array} & \begin{array}{cc} \dots & \dots \end{array} & \begin{array}{cc} -b_{nn} & g_{nn} \\ g_{nn} & b_{nn} \end{array} \\ \hline \end{array} \quad (2n, 2n) \tag{5.19}$$

$$\frac{\partial \hat{G}'_i}{\partial V_i^e} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline \frac{\partial g_{11}}{\partial V_{m1}^e} & \frac{\partial g_{11}}{\partial V_{r1}^e} & & & & & & & & & \\ \frac{\partial g_{21}}{\partial V_{m1}^e} & \frac{\partial g_{21}}{\partial V_{r1}^e} & & & & & & & & & \\ \hline & & \ddots & \ddots & & & & & & & \\ \hline & & \ddots & \ddots & & & & & & & \\ \hline & & & & \frac{\partial g_{1i}}{\partial V_{mi}^e} & \frac{\partial g_{1i}}{\partial V_{ri}^e} & & & & & \\ & & & & \frac{\partial g_{2i}}{\partial V_{mi}^e} & \frac{\partial g_{2i}}{\partial V_{ri}^e} & & & & & \\ \hline & & & & & & \ddots & \ddots & & & \\ & & & & & & \ddots & \ddots & & & \\ \hline & & & & & & & & \frac{\partial g_{in}}{\partial V_{mn}^e} & \frac{\partial g_{in}}{\partial V_{rn}^e} & \\ & & & & & & & & \frac{\partial g_{2n}}{\partial V_{mn}^e} & \frac{\partial g_{2n}}{\partial V_{rn}^e} & \\ \hline \end{array} \quad (2n, 2n) \tag{5.20}$$

Note que a matriz Jacobiana em (5.13) é assimétrica e muito esparsa. A submatriz  $\frac{\partial \hat{F}_i}{\partial \hat{x}_i}$ ;  $i = 1, 2, \dots, p$ ; é bloco diagonal, onde cada bloco é, também, muito esparsa e está associado à uma determinada máquina síncrona e seus controladores. As submatrizes  $\frac{\partial \hat{G}'_i}{\partial V_i^e}$ ;  $i = 1, 2, \dots, p$ ; alteram a matriz admitância nodal somente para os elementos bloco diagonais  $2 \times 2$ . As demais submatrizes  $\frac{\partial \hat{F}_i}{\partial V_i^e}$ ;  $i = 1, 2, \dots, p$ ; e  $\frac{\partial \hat{G}'_i}{\partial \hat{x}_i}$ ;  $i = 1, 2, \dots, p$ ; são também esparsas. Todas essas características devem ser consideradas no momento da escolha de um método de solução.

Em [24] encontra-se uma tentativa de resolver um sistema do tipo (5.9) ou (5.13), através do método direto baseado na fatoração LU e de um critério de ordenação entre os blocos da matriz  $\hat{J}$ . Rigorosamente, esse Jacobiano deve ser recalculado a cada iteração no método de Newton, tornando muito elevado o tempo computacional gasto com estas fatorações e ordenações. Além disso, o critério de ordenação citado implica na necessidade de fatorar matrizes resultantes do produto de duas ou mais matrizes esparsas, dependendo do número de aplicações sucessivas do critério. Conseqüentemente, essas matrizes serão menos esparsas que as matrizes originais, acentuando-se esse problema.

Uma alternativa a ser avaliada é a aplicação de métodos iterativos, tais como os métodos do tipo Gradiente Conjugado pré-condicionados, objeto do próximo item. Estes métodos têm sido usados com sucesso, em várias áreas do conhecimento, na resolução de sistemas lineares tanto em máquinas vetoriais quanto em máquinas paralelas [11, 12, 13, 48, 49, 56].

## 5.2 Métodos Numéricos de Solução

Conforme discutido no item anterior, a solução das equações algébricas não-lineares que descrevem o comportamento dinâmico de um SEE para  $p$  passos de integração tomados simultaneamente (janela de integração), será realizada pelo método de Newton-Raphson com a utilização de um método iterativo do tipo Gradiente Conjugado para a equação linear Jacobiana. O algoritmo resultante foi denominado SIMPT e é sucintamente apresentado a seguir, onde  $\Gamma$  representa o número total de janelas de integração do período de simulação:

### Algoritmo SIMPT Seqüencial

Calcule  $x^0$  e  $V^{e0}$  em  $t = 0^-$

Para  $N_\gamma = 1, 2, \dots, \Gamma$  ( $\gamma = \Delta t.p$ ), faça:

Calcule :  $\hat{y}_i^0 = [\hat{x}_i^0, \hat{V}_i^{e0}]^T$ ;  $i = 1, 2, \dots, p$

Faça :  $k \leftarrow 0$

Enquanto  $\|\hat{H}\|_2^2 > \epsilon$ ,  $\hat{H} = [\hat{H}_1, \dots, \hat{H}_p]^T$ , faça:

Calcule  $\hat{H}_i = [\hat{F}_i^t, \hat{G}_i^t]^T$ ;  $i = 1, 2, \dots, p$

Calcule  $\hat{J} = -\frac{\partial \hat{H}}{\partial \hat{y}}$

Resolva  $\hat{H}^k = \hat{J}^k \cdot \Delta \hat{y}^{k+1}$  (método tipo GC)

Calcule  $\hat{y}_i^{k+1} = \hat{y}_i^k + \Delta \hat{y}_i^{k+1}; i = 1, 2, \dots, p$

Faça :  $k \leftarrow k + 1$

*Fim-Enquanto*

*Fim-Para*

Neste algoritmo, três áreas de investigação são consideradas principais:

- O mapeamento do algoritmo em uma arquitetura paralela, de modo a minimizar o desbalanceamento da carga e o volume de dados que precisam ser transferidos entre os processadores;
- A definição do método tipo Gradiente Conjugado e dos pré-condicionadores para o sistema linear Jacobiano;
- O desenvolvimento de um efetivo método de inicialização para a resolução do sistema algébrico não-linear associado aos  $p$  passos de integração.

## 5.2.1 Métodos do Tipo Gradiente Conjugado

Os métodos do tipo Gradiente Conjugado derivam da família de métodos de solução de sistemas lineares denominada de métodos do subespaço de Krylov ou métodos tipo projetados [5, 36]. O mais popular é o Gradiente Conjugado (CG), utilizável somente em sistemas com a matriz de coeficiente simétrica [5, 36]. Para matrizes assimétricas, têm sido empregados o GMRES (Generalized Minimal Residual) [5, 52], o Gradiente Biconjugado (BCG - Biconjugate Gradient) [5], o CGS (Conjugate Gradient-Squared) [5] e, mais recentemente, o Bi-CGSTAB [53].

Neste trabalho optou-se pela utilização dos métodos BCG e Bi-CGSTAB, brevemente descritos nos próximos dois subitens.

### O Método BCG Pré-condicionado

Para sistemas lineares assimétricos, em geral, não é possível obter-se a ortogonalidade entre os vetores de resíduo  $r$ , usando-se uma simples relação de recorrência, tal como empregado no método Gradiente Conjugado [5]. Uma das alternativas propostas [5], é gerar, semelhantemente ao método GC, duas seqüências de vetores de resíduos  $r$  e  $\bar{r}$  e de direções  $p$  e  $\bar{p}$ , satisfazendo a condição de mútua ortogonalidade

$$\begin{aligned}
(r_i, \bar{r}_j) &= 0 \\
(\bar{p}_i, A p_j) &= 0
\end{aligned}
\quad \text{para } i \neq j \quad (5.21)$$

o que explica o nome do algoritmo.

O algoritmo do método Gradiente Biconjugado pré-condicionado, aplicado a equação Jacobiana (5.9), é apresentado a seguir ( $M$  é a matriz de pré-condicionamento):

### Algoritmo BCG

*Escolha*  $\Delta \hat{y}^0$

*Calcule*  $r^0 = \hat{H} - \hat{J} \Delta \hat{y}^0$

*Resolva*  $M \bar{r}^0 = r^0$  e faça  $p^0 = \bar{p}^0 = \bar{r}^0 = \bar{r}^0$

*Para*  $k = 0, 1, \dots$ , faça:

$$\alpha^k = [\bar{r}^k]^T \cdot [\bar{r}^k] / [\bar{p}^k]^T \cdot [M^{-1} \hat{J} \cdot p^k]$$

$$\Delta \hat{y}^{k+1} = \Delta \hat{y}^k + \alpha^k p^k$$

$$\bar{r}^{k+1} = \bar{r}^k - \alpha^k M^{-1} \hat{J} \cdot p^k$$

$$\bar{r}^{k+1} = \bar{r}^k - \alpha^k \hat{J}^T (M^{-1})^T \cdot \bar{p}^k$$

*Se*  $\|\bar{r}^{k+1}\|_2^2 \leq \epsilon$

$$\text{Calcule } r^{k+1} = M \bar{r}^{k+1}$$

*Se*  $\|r^{k+1}\|_2^2 \leq \epsilon$ , então, *Pare*

*Fim-Se*

*Senão, Calcule :*

$$\beta^k = [\bar{r}^{k+1}]^T \cdot [\bar{r}^{k+1}] / [\bar{r}^k]^T \cdot [\bar{r}^k]$$

$$p^{k+1} = \bar{r}^{k+1} + \beta^k p^k$$

$$\bar{p}^{k+1} = \bar{r}^{k+1} + \beta^k \bar{p}^k$$

*Fim-Se*

*Fim-Para*

A principal desvantagem desse método é a necessidade de se calcular dois produtos do tipo matriz-vetor para cada iteração ( $\hat{J} \cdot p^k$  e  $\hat{J}^T \cdot \bar{p}^k$ , supondo  $M = I$ ).

### O Método Bi-CGSTAB Pré-condicionado

O método Bi-CGSTAB é uma variante do método BCG, para sistemas assimétricos e não necessita trabalhar com a matriz  $\hat{J}^T$  [53]. O seu algoritmo, aplicado a equação (5.9) e usando-se uma matriz de pré-condicionamento  $M$ , é apresentado a seguir:

## Algoritmo Bi-CGSTAB

Escolha  $\Delta\hat{y}^0$

Calcule  $r^0 = \hat{H} - \hat{J}\Delta\hat{y}^0$

Faça  $\bar{r}^0 = r^0$  e  $p^0 = r^0$

Para  $k = 0, 1, 2, \dots$ , faça:

$$M\hat{p} = p^k$$

$$\alpha^k = [\bar{r}^0]^T \cdot [r^k] / [\bar{r}^0]^T \cdot [\hat{J} \cdot \hat{p}]$$

$$q = r^k - \alpha^k \hat{J} \cdot \hat{p}$$

$$M\hat{q} = q$$

$$\omega^k = [\hat{J} \cdot \hat{q}]^T \cdot [q] / [\hat{J} \cdot \hat{q}]^T \cdot [\hat{J} \cdot \hat{q}]$$

$$\Delta\hat{y}^{k+1} = \Delta\hat{y}^k + \alpha^k \hat{p} + \omega^k \hat{q}$$

$$r^{k+1} = q - \omega^k \hat{J} \hat{q}$$

Se  $\|r^{k+1}\|_2^2 \leq \epsilon$ , então, Pare

Senão, Calcule :

$$\beta^k = [\bar{r}^0]^T \cdot [r^{k+1}] / [\bar{r}^0]^T \cdot [r^k] \cdot \alpha^k / \omega^k$$

$$p^{k+1} = r^{k+1} + \beta^k [p^k - \omega^k \hat{J} \cdot \hat{p}]$$

Fim-Se

Fim-Para

### 5.2.2 Pré-condicionamento

A escolha de uma matriz de pré-condicionamento  $M = M_1 M_2$  para um sistema linear, por exemplo  $\hat{H} = \hat{J} \Delta\hat{y}$ , visando melhorar a razão de convergência de métodos iterativos, deve ser tal que  $M \approx \hat{J}$  e  $\text{cond}(M^{-1} \hat{J}) \leq \text{cond}(\hat{J})$  [36]. O sistema linear explicitamente pré-condicionado é representado por:

$$\tilde{H} = \tilde{J} \Delta\tilde{y} \quad (5.22)$$

onde  $\tilde{J} = M_1^{-1} \hat{J} M_2^{-1}$ ,  $\Delta\tilde{y} = M_2^{-1} \Delta\hat{y}$  e  $\tilde{H} = M_1^{-1} \hat{H}$ .

Por exemplo, para  $M_1 = I$  tem-se o pré-condicionamento a direita ( ou pós-condicionamento), para  $M_2 = I$  tem-se o pré-condicionamento a esquerda e para  $M_1 = L$ ,  $M_2 = U$ , tem-se o pré-condicionamento de ambos os lados.

Conforme já discutido no capítulo anterior, nos métodos do tipo GC, o uso do pré-condicionamento é representado, em geral, através de modificações no vetor de

resíduos  $r$ , em determinados passos do algoritmo, obtidas resolvendo-se um sistema linear algébrico do tipo:

$$M\bar{r} = r \quad (5.23)$$

Assim, a escolha de  $M$  deve ser uma solução de compromisso tal que  $M$  seja o mais próximo possível de  $\hat{J}$ , porém, de modo que o sistema auxiliar (5.23) tenha solução relativamente simples. Sob a ótica dos computadores paralelos/vetoriais, é desejável que  $M$  seja do tipo bloco-diagonal, para que a solução de (5.23) possa ser obtida com paralelismo perfeito, e se utilize o máximo possível a capacidade vetorial da máquina na resolução de cada subsistema.

A proposta inicial para o algoritmo SIMPT, é escolher  $M$  uma matriz bloco diagonal a seguir definida, de modo a priorizar a resolução do sistema auxiliar com paralelismo perfeito.

$$M = \begin{array}{|c|c|c|c|} \hline Q_1 & Y_1 & & \\ \hline 0 & 0 & Q_2 & Y_2 \\ \hline & \dots & & \\ \hline & & & \dots \\ \hline & & & & 0 & 0 & Q_p & Y_p \\ \hline \end{array} \quad (5.24)$$

Na equação (5.24) têm-se  $Q_i = \frac{\partial \hat{F}_i}{\partial \hat{x}_i}$  e  $Y_i = Y^e - \frac{\partial \hat{G}_i}{\partial V_i^e}$  ( $i = 1, 2, \dots, p$ ). Todas essas derivadas parciais já foram definidas anteriormente.

### 5.2.3 O Pós-condicionamento Explícito

Usando-se o descrito no item anterior e definindo-se  $M = M_2$  com  $M_1 = I$ , o sistema linear (5.22) explicitamente pós-condicionado pode ser escrito na forma seguinte:

$$\hat{H} = \hat{J}M_2^{-1}\Delta\hat{y} \quad (5.25)$$

onde o vetor de incógnitas originais  $\Delta\hat{y}$  pode ser encontrado resolvendo-se

$$\Delta\hat{y} = M_2^{-1}\Delta\hat{y} \quad (5.26)$$



Para facilitar o entendimento na seqüência desta exposição, a equação matricial Jacobiana, já representada em (5.13), será reescrita na seguinte forma simbólica:

$$\begin{array}{c} \hat{F}_1 \\ \hat{G}_1 \\ \hat{F}_2 \\ \hat{G}_2 \\ \vdots \\ \vdots \\ \hat{F}_p \\ \hat{G}_p \end{array} = - \begin{array}{|c|c|c|c|} \hline Q_1 & R_1 & & \\ \hline S_1 & Y_1 & & \\ \hline Q_{21} & R_{21} & Q_2 & R_2 \\ \hline 0 & 0 & S_2 & Y_2 \\ \hline & \ddots & \ddots & \\ \hline & & Q_{pp-1} & R_{pp-1} \\ \hline & & 0 & 0 \\ \hline & & S_p & Y_p \\ \hline \end{array} \cdot \begin{array}{c} \Delta \hat{x}_1 \\ \Delta \hat{V}_1^e \\ \Delta \hat{x}_2 \\ \Delta \hat{V}_2^e \\ \vdots \\ \vdots \\ \Delta \hat{x}_p \\ \Delta \hat{V}_p^e \end{array} \quad (5.27)$$

Adotando-se para  $M_2$ , a mesma matriz  $M$  definida em (5.24), e usando a equação (5.27), o sistema linear (5.25) assume a seguinte forma:

$$\begin{array}{c} \hat{F}_1 \\ \hat{G}_1 \\ \hat{F}_2 \\ \hat{G}_2 \\ \vdots \\ \vdots \\ \hat{F}_p \\ \hat{G}_p \end{array} = - \begin{array}{|c|c|c|c|} \hline I & R_1 Y_1^{-1} & & \\ \hline S_1 Q_1^{-1} & I & & \\ \hline Q_{21} Q_1^{-1} & R_{21} Y_1^{-1} & I & R_2 Y_2^{-1} \\ \hline 0 & 0 & S_2 Q_2^{-1} & I \\ \hline & \ddots & \ddots & \\ \hline & & Q_{pp-1} Q_{p-1}^{-1} & R_{pp-1} Y_{p-1}^{-1} \\ \hline & & 0 & 0 \\ \hline & & S_p Q_p^{-1} & I \\ \hline \end{array} \begin{array}{c} \Delta \tilde{x}_1 \\ \Delta \tilde{V}_1^e \\ \Delta \tilde{x}_2 \\ \Delta \tilde{V}_2^e \\ \vdots \\ \vdots \\ \Delta \tilde{x}_p \\ \Delta \tilde{V}_p^e \end{array} \quad (5.28)$$

Em relação a equação (5.28), faz-se os seguintes comentários:

- Os produtos  $R_i Y_i^{-1}$ ,  $i = 1, 2, \dots, p$  exigem valores somente das colunas de  $Y_i^{-1}$  associados a barras de geração;
- Se as submatrizes  $R_i$  e  $R_{ij}$  puderem ser desprezadas, a resolução do sistema linear Jacobiano ficaria bastante simplificada;

- A solução deste sistema, nas variáveis originais  $\Delta\hat{y}$ , é obtida a partir de  $\Delta\tilde{y}$  resolvendo-se  $M_2\Delta\hat{y} = \Delta\tilde{y}$ , onde a matriz  $M_2$  é bloco diagonal.

### 5.2.4 Inicialização

A ideia é se obter um algoritmo simples e com elevado grau de paralelismo para gerar uma solução aproximada, no intervalo de tempo de uma janela de integração. A primeira tentativa de se encontrar um algoritmo com tais características, será feita com a utilização de um modelo simplificado para o SEE, sendo a sua solução obtida pelo método da Série de Taylor, empregada com sucesso em [58] e [59] no contexto dos métodos diretos de análise da estabilidade transitória de SEE.

#### Modelo Matemático

Desprezando-se os efeitos dos controladores e representando-se as máquinas síncronas por um modelo de segunda ordem, o SEE pode ser descrito pelas seguintes equações:

$$\left\{ \begin{array}{l} \frac{d\omega_i}{dt} = \frac{P_{mi} - P_{ei}}{M_i} \\ \frac{d\delta_i}{dt} = \omega_i \\ I = YV \\ i = 1, 2, \dots, m \end{array} \right. \quad (5.29)$$

onde:

$$\left\{ \begin{array}{l} M_i = 2H_i/\omega_s \\ P_{ei} = \frac{E'_{qi}}{x'_{di}} (V_{ri} \sin \delta_i - V_{mi} \cos \delta_i) \end{array} \right. \quad (5.30)$$

e  $V_{ri}$ ,  $V_{mi}$  : são, respectivamente, as parcelas real e imaginária da tensão na  $i$ -ésima barra e  $\delta_i$  é o ângulo atrás da reatância transitória.

#### Aplicação da Expansão em Série de Taylor

A solução do sistema de equações (5.29) dada por :  $\delta_i(t)$  e  $\omega_i(t)$ ,  $i = 1, 2, \dots, m$ , pode ser obtida através de uma expansão em série de Taylor, realizada para o instante de tempo  $t = t_0$ , conforme descrito a seguir:

$$\delta_i(t) = \delta_i(t_0) + \delta_i^{(1)}(t_0) \frac{(t-t_0)}{1!} + \delta_i^{(2)}(t_0) \frac{(t-t_0)^2}{2!} + \dots \quad (5.31)$$

$$\omega_i(t) = \omega_i(t_0) + \omega_i^{(1)}(t_0) \frac{(t-t_0)}{1!} + \omega_i^{(2)}(t_0) \frac{(t-t_0)^2}{2!} + \dots \quad (5.32)$$

ou

$$\omega_i(t) = \delta_i^{(1)}(t_0) + \delta_i^{(2)}(t_0) \frac{(t-t_0)}{1!} + \delta_i^{(3)}(t_0) \frac{(t-t_0)^2}{2!} + \dots \quad (5.33)$$

As soluções de (5.31) e (5.32) exigem a determinação de derivadas de alta ordem para  $\delta_i$  e  $\omega_i$ ,  $i = 1, 2, \dots, m$ , para o instante de tempo  $t = t_0$ . Essas derivadas podem ser obtidas muito eficientemente em computadores paralelos, conforme descrito no apêndice E.

### 5.3 A Paralelização do Algoritmo SIMPT

O algoritmo SIMPT, para cada janela de integração  $\gamma = \Delta t.p$ , constitui-se dos seguintes passos principais:

1. Inicialização das variáveis do problema para cada passo de integração ( $\hat{y}_i^0 = [\hat{x}_i^0, \hat{V}_i^{e0}]^T$ ;  $i = 1, 2, \dots, p$ );
2. Cálculo dos elementos dos vetores de funções  $\hat{H}_i$ ;  $i = 1, 2, \dots, p$ ;
3. Cálculo dos elementos da matriz Jacobiana  $\hat{J} = -\frac{\partial \hat{H}}{\partial \hat{y}}$ ;
4. Resolução do sistema linear Jacobiano  $\hat{H}^k = \hat{J}^k \cdot \Delta \hat{y}^{k+1}$ ;
5. Adição em  $\hat{y}^k$  dos acréscimos  $\Delta \hat{y}^{k+1}$  determinados no passo anterior ( $\hat{y}_i^{k+1} = \hat{y}_i^k + \Delta \hat{y}_i^{k+1}$ ;  $i = 1, 2, \dots, p$ )

Observa-se desta síntese que todos os passos principais do algoritmo SIMPT, exceto o passo 4, podem ser decompostos trivialmente em  $p$  subtarefas. Assim, a proposta deste trabalho é alocar todas as equações de cada passo da janela de integração em um único processador. A resolução paralela da equação matricial Jacobiana será tratada posteriormente. Os principais apelos dessa proposta são:

- O perfeito balanceamento da carga entre os processadores;

- O baixo custo para a comunicação de dados, já que, exceto para o teste de convergência, cada processador necessita comunicar-se somente com os processadores alocados aos passos de integração imediatamente anterior e posterior na escala dos tempos. Isto é devido a utilização do método de integração de passo único trapezoidal implícito;
- Cada processador poderá ser interpretado como sendo um “cluster” constituído de  $q$  processadores e assim, as equações de cada passo de integração poderiam ser adicionalmente mapeadas nestes  $q$  processadores. O resultado final seria o mapeamento das equações da janela de integração em  $q \times p$  processadores, caracterizando-se a chamada paralelização no tempo e no espaço.

A figura 5.1 sintetiza em um fluxograma, o mapeamento do algoritmo SIMPT em um computador paralelo com  $p$  processadores e memória distribuída.

A resolução paralela da equação matricial Jacobiana é ilustrada através dos fluxogramas da figura 5.2, para a aplicação do método Gradiente Biconjugado pré-condicionado (BCG), e da figura 5.3 para a aplicação do método Bi-CGSTAB pré-condicionado. Observa-se nestes fluxogramas que todas as comunicações de dados de natureza global são feitas com simples escalares e, portanto, o seu custo independe da dimensão do problema a ser resolvido. Por outro lado, em função das características estruturais da matriz Jacobiana, as demais comunicações de dados são de vizinhança no sentido de que cada processador necessita comunicar-se somente com os processadores alocados aos passos de integração imediatamente anterior e posterior.

## 5.4 Conclusões

Neste capítulo foi descrito um novo algoritmo paralelo para a resolução das EDAs que representam a dinâmica de um SEE. Empregam-se o esquema *Simultâneo Implícito* e o conceito de paralelismo no tempo, ou seja, a exploração do paralelismo existente entre os passos de integração. O algoritmo desenvolvido foi denominado SIMPT. Da mesma forma que para o esquema Simultâneo Implícito convencional, é aplicado o método de Newton as equações algébricas e algebrizadas não-lineares, resultantes da aplicação do método de integração trapezoidal implícito as equações diferenciais.

As principais contribuições estão na utilização dos métodos iterativos Gradiente Biconjugado pré-condicionado e Bi-CGSTAB pré-condicionado, na resolução da e-

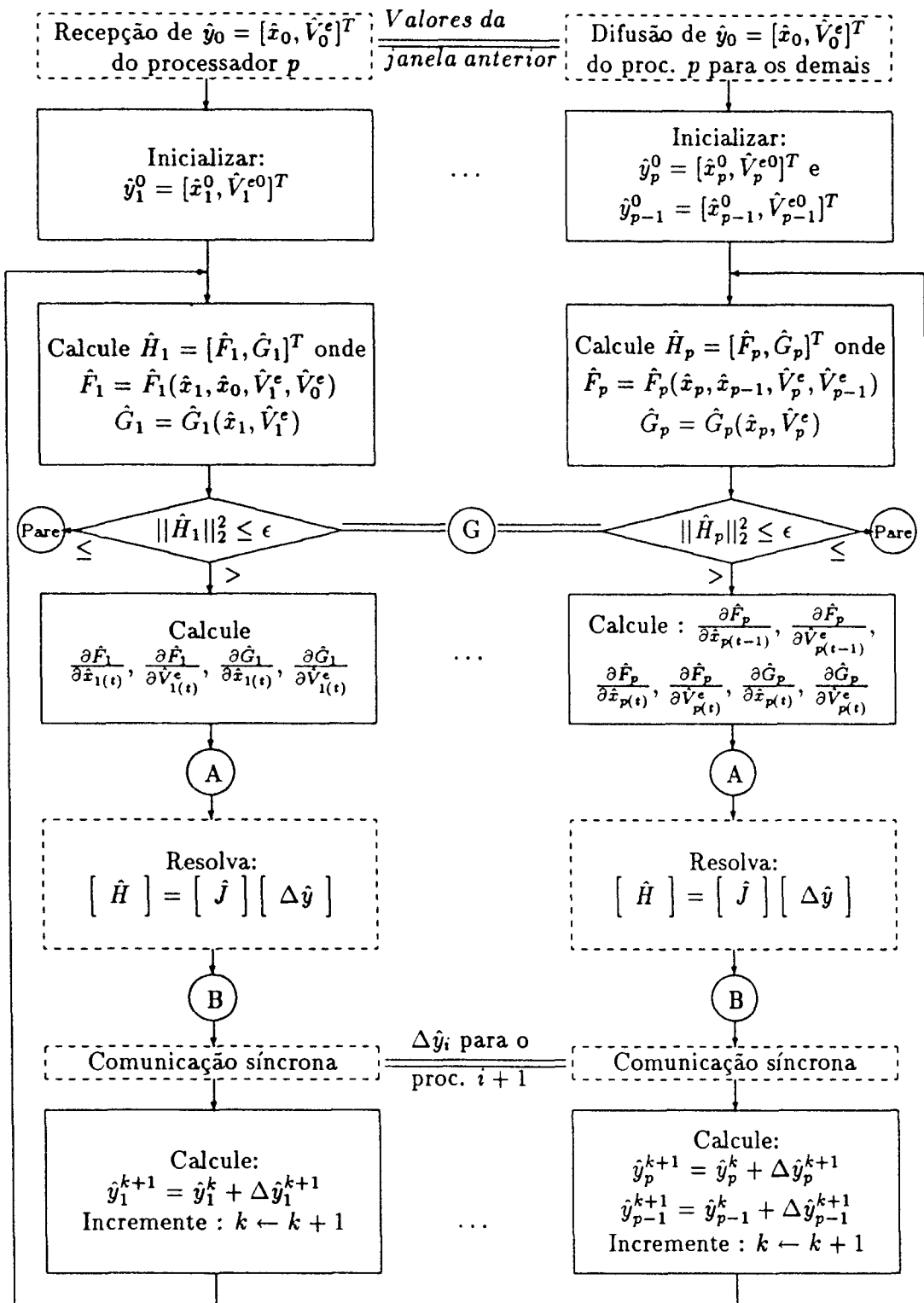


Figura 5.1: Fluxograma do algoritmo SIMPT paralelo no tempo para  $p$  passos de integração

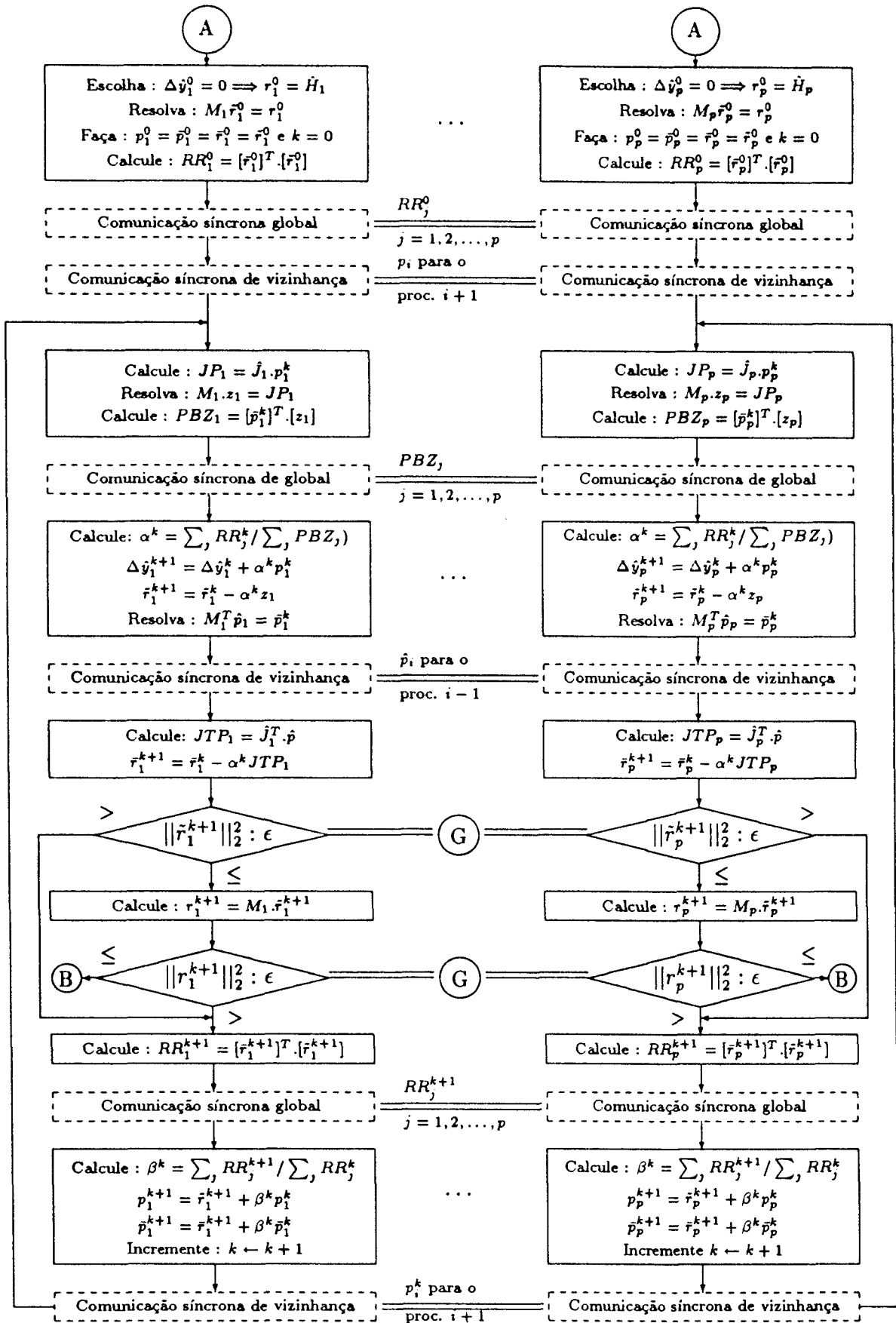


Figura 5.2: Fluxograma do algoritmo biconjugado paralelo para resolver a equação linear Jacobiana

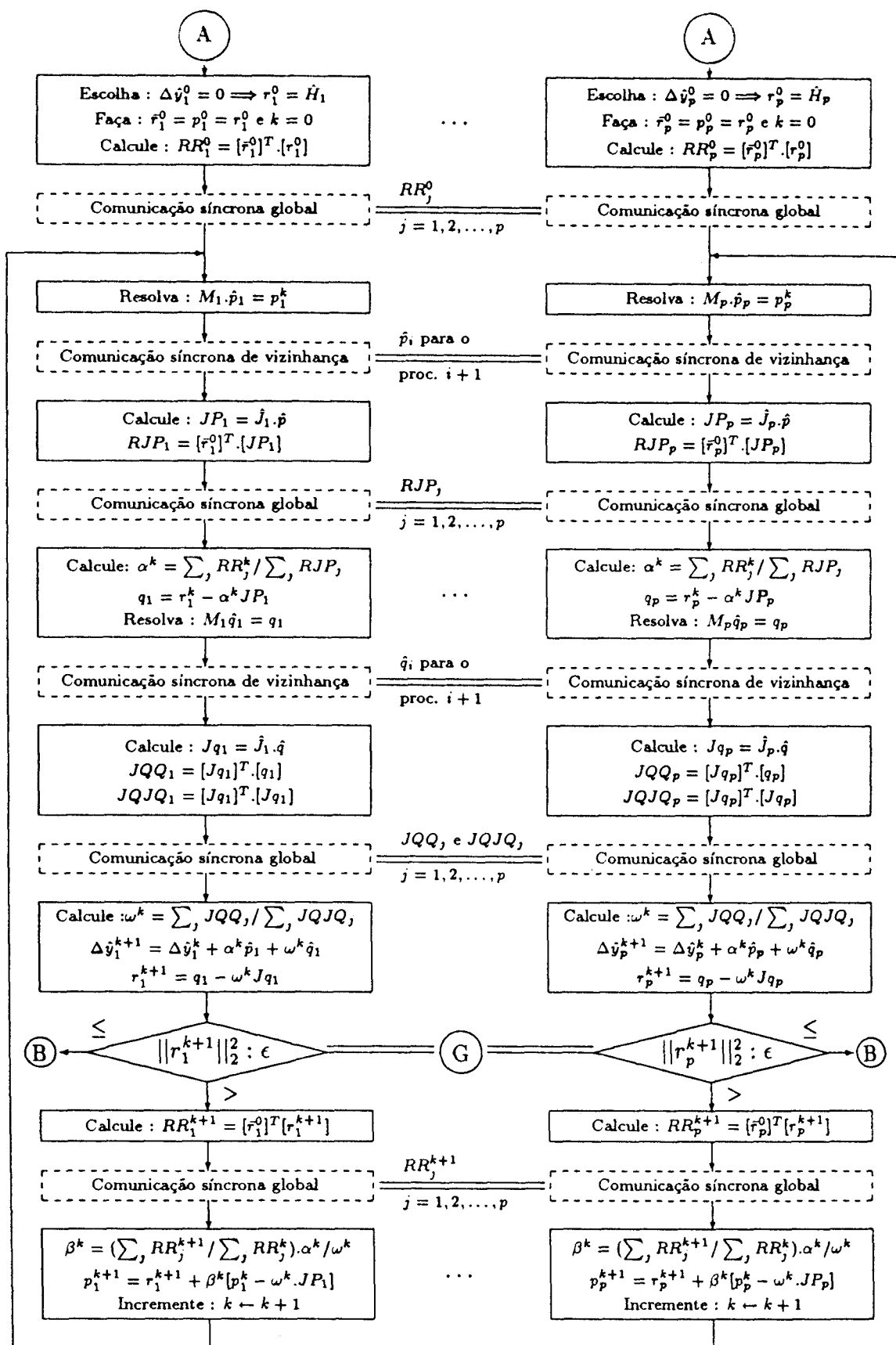


Figura 5.3: Fluxograma do algoritmo Bi-CGSTAB paralelo para resolver a equação linear jacobiana

quação matricial Jacobiana de muito grande porte, resultante da consideração de vários passos de integração simultaneamente. Outras contribuições são a proposta de inicialização das variáveis da janela de integração por série de Taylor e de métodos de pré-condicionamento e pós-condicionamento explícito do sistema linear Jacobiano.



## Capítulo 6

# Avaliação de Desempenho em Ambientes Computacionais Paralelos

Neste capítulo descreve-se a aplicação e a avaliação de desempenho dos algoritmos desenvolvidos nos capítulos 4 (AEIGC e AEIGCRD) e 5 (SIMPT), em simulações seqüenciais e paralelas do comportamento dinâmico de dois SEE. O sistema padrão de testes IEEE 118 barras e uma configuração do sistema interligado das regiões Sul e Sudeste do Brasil com 616 barras. Os experimentos consideraram perturbações do tipo curto-circuito trifásico eliminado pelo chaveamento de uma linha de transmissão. Utilizou-se os computadores paralelos NCP I, desenvolvido pela COPPE/UFRJ, e o Intel iPSC/860. Também foram feitas simulações em máquinas seqüenciais, para propósitos de comparação. Foram realizados os seguintes tipos de processamentos:

- Paralelo real em configurações de 2, 4 e 8 processadores;
- Paralelo simulado seqüencialmente para as configurações de 2, 4, 8, 16 e 32 processadores;
- Seqüencial usando o esquema *Alternado Entrelaçado Implícito Convencional* (AEIC), nos computadores iPSC/860, NCP I, IBM 4381 e IBM 3090.

O processamento paralelo simulado seqüencialmente teve o objetivo de estimar o desempenho dos algoritmos desenvolvidos, em configurações paralelas com o número de processadores superior ao disponível atualmente na COPPE/UFRJ. Por outro lado, os processamentos seqüenciais tiveram o objetivo de comparar os algoritmos

desenvolvidos, com o esquema AEIC, empregado em programas comerciais de simulação do comportamento dinâmico dos SEE.

As avaliações de desempenho são feitas a partir da definição de figuras de mérito, tais como: número de iterações, tempos de CPU e comunicação, eficiências e “speed ups”.

Este capítulo está estruturado de modo que nos cinco primeiros itens descrevem-se, pela ordem: os computadores paralelos utilizados; os sistemas elétricos de testes; o sistema computacional implementado; os parâmetros das simulações; e as definições das figuras de mérito utilizadas nas avaliações de desempenho. Nos três itens subsequentes apresentam-se os resultados numéricos obtidos com a paralelização no espaço, através dos algoritmos AEIGC e AEIGCRD, e uma análise comparativa de desempenho. No item subsequente apresentam-se os resultados numéricos obtidos com a paralelização no tempo, algoritmo SIMPT, e faz-se uma avaliação. Finalmente apresentam-se as conclusões e comentários.

## 6.1 Computadores Paralelos Utilizados

Os processamentos paralelos reais foram realizados no NCP I, somente para o algoritmo AEIGC, e no Intel iPSC/860, para todos os algoritmos desenvolvidos. Os dois computadores citados pertencem ao Laboratório de Computação Paralela da COPPE/UFRJ.

O NCP I, desenvolvido na COPPE/UFRJ, possui arquitetura hipercúbica com 8 processadores atualmente, e está conectado a um microcomputador do tipo IBM PC/386. Cada nó desse computador paralelo possui um processador de 32 bits com tecnologia RISC, do tipo Transputer T800, e memória local de 2Mb. A capacidade de pico de cada processador, operando com clock de 20 MHz, é de aproximadamente 10 Mips e 1,5 Megaflops. O sistema de comunicação é do tipo “message-passing” e tem capacidade de 20 Megabits por segundo. Estes elementos definem uma capacidade de pico de 12 Megaflops e 80 Mips, para a versão atual do NCP I.

O Intel iPSC/860 é do tipo multiprocessador com memória distribuída, topologia hipercúbica, possui 8 nós de processamento atualmente, e está conectado a um microcomputador do tipo IBM PC/386. Cada nó contém um microprocessador i860 com tecnologia RISC de 64 bits e 8 Megabytes de memória. O sistema de comunicação entre os nós é do tipo “message-passing” e tem capacidade de pico de 2,8 Megabytes

por segundo. Adicionalmente, cada mensagem partindo de um nó toma cerca de 65 microsegundos de tempo de CPU para estabelecer seu caminho. A capacidade de pico de cada processador é de até 80 Megaflops em operações com precisão simples e de até 60 Megaflops em precisão dupla.

## 6.2 Sistemas Elétricos de Testes

Foram utilizados os sistemas de energia elétrica IEEE 118 e SSUD 616, descritos a seguir:

**IEEE 118** : Sistema padrão de testes do IEEE, constituído de 34 geradores (20 representados dinamicamente), 118 barras e 179 linhas de transmissão e transformadores;

**SSUD 616** : Configuração do sistema interligado das regiões Sul e Sudeste do Brasil, para o ano de 1987, com algumas subareas representadas por equivalentes estáticos de rede. Este sistema constitui-se de 88 geradores representados dinamicamente, 616 barras e 995 linhas de transmissão e transformadores.

O modelo dinâmico global foi limitado a representação das máquinas síncronas por modelos de até quarta ordem e os sistemas de excitação pelo modelo IEEE tipo 1. As cargas são representadas por impedâncias constantes. O emprego do mesmo tipo de modelo para todas as máquinas e sistemas de excitação, exigiu a adoção de valores típicos para alguns parâmetros.

## 6.3 Programas Computacionais

As implementações computacionais, seqüenciais e paralelas, dos algoritmos desenvolvidos estão organizadas em uma estrutura de sistema computacional denominada PPSDS (“Parallel Power System Dynamic Simulation”) e foi construída ao longo deste trabalho. Este sistema contempla os esquemas de solução *Alternado Entrelaçado Implícito* e *Simultâneo Implícito*, e os algoritmos AEIGC, AEIGCRD e SIMPT. Neste contexto encontram-se diversas opções de modelos dos elementos de SEE, diversos métodos de solução de sistemas lineares simétricos e assimétricos e diversos métodos de pré-condicionamento; a seguir descritos.

- **Modelos para os elementos de SEE**

- *Máquinas Síncronas* : Modelo Clássico e modelo transitório de terceira e quarta ordem;
- *Sistemas de Excitação* : IEEE tipo 1 e IEEE tipo AC4;
- *Cargas* : Modelo polinomial com parcelas de impedância constante, corrente constante e potência constante.

- **Métodos de Resolução de Sistemas Lineares**

- Direto usando a fatoração LU e substituições direta e inversa;
- *GC* : Gradiente Conjugado pré-condicionado (método iterativo, para sistemas simétricos);
- *BCG* : Gradiente Biconjugado pré-condicionado (método iterativo para sistemas assimétricos);
- *Bi-CGSTAB* : (método iterativo, para sistemas assimétricos);

- **Métodos de Pré-condicionamento**

- Nenhum;
- Diagonal
- Fatoração LU incompleta;
- Série de Neumann;
- Série de Maclaurin.

A estrutura constitucional do sistema PPSDS contém 3 bibliotecas de sub-rotinas, 4 programas seqüenciais e 3 programas paralelos; descritos a seguir:

- **Bibliotecas de sub-rotinas**

**LIBSPSDS** : sub-rotinas exclusivamente seqüenciais e utilizadas pelos programas seqüenciais e paralelos;

**LIBPPSDS** : sub-rotinas utilizadas exclusivamente pelos programas paralelos;

**COMUNICAÇÃO** : sub-rotinas destinadas a realização da comunicação de dados entre os processadores;

- **Programas Computacionais**

**GCPP** : versão paralela do algoritmo AEIGC;

**GCPS** : versão seqüencial do algoritmo AEIGC;

**GCRDPP** : versão paralela do algoritmo AEIGCRD;

**GCRDPS** : versão seqüencial do algoritmo AEIGCRD;

**SMPTPP** : versão paralela do algoritmo SIMPT;

**SMPTPS** : versão seqüencial do algoritmo SIMPT;

**AEIPS** : programa que utiliza o esquema Alternado Entrelaçado Implícito Convencional (AEIC), implementado seqüencialmente.

Na figura 6.1 apresenta-se uma síntese dessa estrutura.

As implementações seqüenciais foram feitas em linguagem Fortran 77. Nos programas paralelos utilizou-se o Fortran Paralelo 3L [60], desenvolvido para sistemas computacionais com transputers, e o Fortran 77 adaptado para o iPSC/860. Essas linguagens são basicamente o Fortran 77, acrescido de sub-rotinas para tornar possível a comunicação entre processadores e outros instrumentos necessários as implementações paralelas. Todos os programas e sub-rotinas estão dimensionados, atualmente, para tratar sistemas com até 1500 linhas de transmissão, 1000 barras e 99 geradores representados dinamicamente.

A implementação do programa AEIPS teve o objetivo de viabilizar a comparação de desempenho dos algoritmos paralelos com um dos esquemas mais empregados em programas comerciais de SDSEE. Por outro lado, as implementações seqüenciais GCPS, GCRDPS e SMPTPS foram utilizadas para estimar o desempenho dos algoritmos paralelos em computadores com mais de 8 processadores, número atual dos computadores da COPPE/UFRJ.

## 6.4 Parâmetros das Simulações

Os experimentos computacionais constituíram-se de simulações da dinâmica de ambos os SEE de testes, citados anteriormente, frente a grandes perturbações. Os

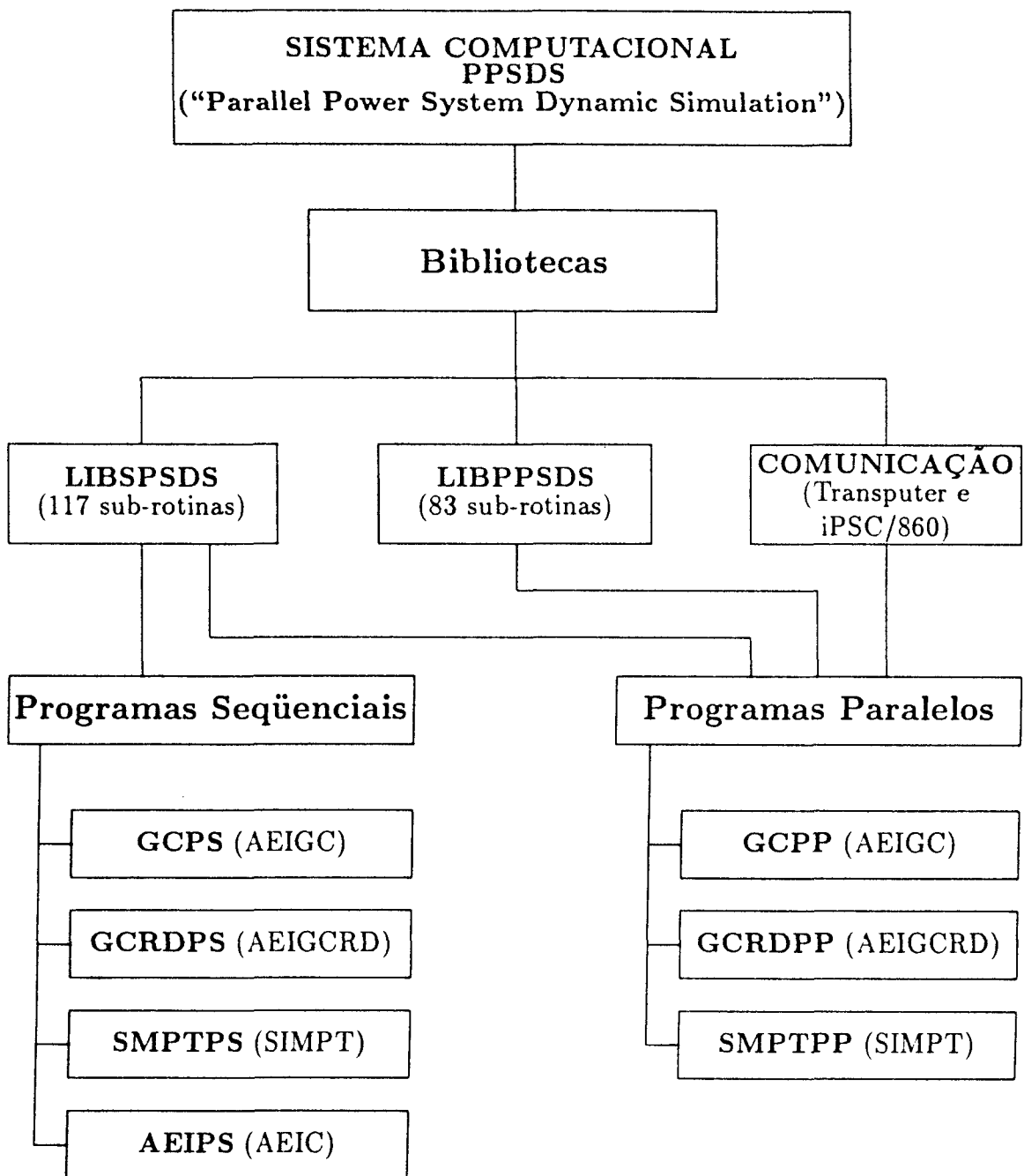


Figura 6.1: Estrutura constitucional do sistema computacional PPSDS - ("Parallel Power System Dynamic Simulation")

parâmetros dessas simulações são os seguintes:

- **Perturbações** : Curto-circuito trifásico eliminado pelo chaveamento de uma linha de transmissão;
- **Tempo de permanência do curto-circuito** : 0,10 s.
- **Período total de simulação** : 5,0 s.
- **Passo de integração** : 0,01 s.
- **Tolerância de convergência** :
  - *Esquema Alternado*
    - \* Para o GC :  $10^{-2}$
    - \* Para a rede elétrica :  $10^{-3}$
    - \* Para as variáveis diferenciais (AEI) :  $10^{-4}$
  - *Esquema Simultâneo*
    - \* Sistema linear (met. tipo GC.) :  $10^{-4}$
    - \* Sistema não-linear (met. tipo Newton) :  $10^{-4}$
- **Tipos de processamento** :
  - Seqüencial
  - Paralelo
- **Nível de otimização na compilação** :
  - Sem otimização ( -O0 )
  - Otimização máxima ( -O4 )

O exame do desempenho para níveis diferentes de otimização na compilação, visam destacar as potencialidades de vetorização dos algoritmos desenvolvidos, pois empregam métodos tipo GC na resolução de sistemas lineares.

## 6.5 Avaliação de Desempenho e Figuras de Mérito

A avaliação de desempenho dos algoritmos paralelos desenvolvidos é feita em termos do número de iterações, tempos de CPU e comparativamente em relação a SDSEE, em processamentos seqüenciais, usando-se o esquema *Alternado Entrelaçado Implícito Convencional*. Além disso, são empregadas figuras de mérito do tipo eficiência e “speed up”, definidas em um subitem a seguir.

### 6.5.1 O Padrão de Comparação

Neste subitem são apresentados os resultados de simulações do comportamento dinâmico dos sistemas padrão de testes IEEE 118 barras e SSUD 616 barras. Eles foram obtidos em simulações usando-se o algoritmo AEIC seqüencial em um único nó dos computadores paralelos NCP I e Intel iPSC/860.

A tabela 6.1 contém os resultados em termos do número de iterações e a tabela 6.2 em termos dos tempos de CPU, em segundos, respectivamente, para a solução global e para a resolução das equações associadas a rede elétrica.

Nas simulações feitas no iPSC/860, foram utilizadas as opções de compilação 0 (sem qualquer otimização de código) e 4 (nível máximo de otimização de código da versão atual do compilador). Contudo, as diferenças nos tempos de CPU são desprezíveis para esse algoritmo.

Tabela 6.1: Número de iterações para o algoritmo AEIC Seqüencial

Sistema Elétrico	Número de Iterações			
	Total		Média	
	AEI	Rede	AEI	Rede
IEEE 118	1540	1048	2,96	2,02
SSUD 616	1501	1017	2,89	1,96

### 6.5.2 Eficiências e “Speed ups”

Definem-se dois conjuntos de “speed ups” e eficiências: um para o esquema alterado, algoritmos AEIGC e AEIGCRD, e outro para o esquema simultâneo, algoritmo SIMPT.



### Para o Esquema Alternado

$$S_{a1} = \frac{t_{CPU} \text{ para o algoritmo AEIC (seqüencial)}}{t_{CPU} \text{ para o algoritmo AEI com GC (paralelo)}} , \quad E_{a1} = S_{a1}/q$$

$$S_{a2} = \frac{t_{CPU} \text{ para o algoritmo AEI com GC (seqüencial)}}{t_{CPU} \text{ para o algoritmo AEI com GC (paralelo)}} , \quad E_{a2} = S_{a2}/q$$

onde  $t_{CPU}$  representa o tempo de CPU em segundos e  $q$  é o número de processadores utilizados em paralelo. O “speed up”  $S_{a2}$  e a eficiência  $E_{a2}$  representam uma medida da paralelização alcançada pelo algoritmo que utiliza o método GC (AEIGCRD ou AEIGC). Por outro lado,  $S_{a1}$  e  $E_{a1}$  representam, respectivamente, o “speed up” e a eficiência alcançados pelo algoritmo AEIGCRD ou AEIGC paralelos, comparativamente ao algoritmo AEIC seqüencial. Conseqüentemente, eles refletem o ganho efetivo no emprego de computadores paralelos.

### Para o Esquema Simultâneo

$$S_{s1} = \frac{t_{CPU} \text{ para } p \text{ passos (seqüencial)}}{t_{CPU} \text{ para } p \text{ passos (paralelo)}} , \quad E_{s1} = S_{s1}/p$$

$$S_{s2} = \frac{t_{CPU} \text{ para } 1 \text{ passo (seqüencial)}}{t_{CPU} \text{ para } p \text{ passos (paralelo)}} , \quad E_{s2} = S_{s2}/p$$

onde  $t_{CPU}$  representa o tempo de CPU em segundos e  $p$  é o número de passos de integração tomados simultaneamente, assim como também o número de processadores nos processamentos paralelos. O “speed up”  $S_{s1}$  e a eficiência  $E_{s1}$ , representam uma medida da paralelização alcançada com o algoritmo SIMPT, pois consideram, como referência, o processamento seqüencial com o mesmo número de passos de integração dos processamentos paralelos. Por outro lado, o “speed up”  $S_{s2}$  e a eficiência  $E_{s2}$ , representam, respectivamente, o “speed up” e a eficiência alcançados pelo algoritmo SIMPT paralelo, em relação ao esquema Simultâneo Implícito seqüencial para 1 (um) passo de integração e com a resolução da equação Jacobiana pelo mesmo método tipo GC do algoritmo SIMPT paralelo.

Tabela 6.2: Tempos de CPU (s) para o algoritmo AEIC Seqüencial

Compu- tador	Sistema Elétrico	Opção de Compil.	Tempos de CPU (s)	
			Total	Rede
NCP I	IEEE 118	-	109,8	50,4
	SSUD 616	-	436,0	161,0
iPSC/860	IEEE 118	0	13,5	5,1
		4	13,8	5,5
	SSUD 616	0	62,3	26,6
		4	63,5	28,4

## 6.6 Resultados para o Algoritmo AEIGC

Neste item são apresentados e discutidos os resultados obtidos com a simulação da dinâmica dos sistemas IEEE 118 e SSUD 616, usando-se o algoritmo AEIGC e os computadores paralelos NCP I e iPSC/860.

### 6.6.1 Parâmetros das Decomposições do Sistema

Conforme descrito no capítulo 4, a utilização do algoritmo AEIGC requer a decomposição do sistema de equações em  $q$  subsistemas, onde  $q$  é o número de processadores da configuração paralela que se quer utilizar. No caso específico da rede elétrica, deve-se definir  $q$  subredes interligadas somente através da  $(q + 1)$ -ésima subrede. A decomposição das equações diferenciais é feita independentemente da rede elétrica e tal que sejam atribuídas a um mesmo processador, todas as equações correspondentes a uma máquina síncrona e seus controladores. O objetivo é minimizar os requisitos de comunicação de dados entre os processadores. Assim, a decomposição do sistema de equações diferenciais pode ser expressa em termos do número de máquinas atribuídas a cada processador.

Nos experimentos realizados, foram feitas decomposições em 4 e 8 subsistemas, para o sistema elétrico IEEE 118, e em 2, 4, 8, 16 e 32 subsistemas, para o sistema elétrico SSUD 616. A metodologia adotada baseou-se na análise do diagrama unifilar sem nenhum procedimento automatizado. Contudo, existem trabalhos [61] no sentido de se obter decomposições da rede elétrica por meios automatizados, de modo que a matriz admitância nodal assumia uma forma BBDF que atenda satisfatoriamente aos requisitos da rapidez de convergência do algoritmo e do balanceamento da carga entre os processadores. A tabela 6.3 apresenta o número de barras de cada subrede, o número de máquinas alocadas a cada processador e as dimensões das matrizes  $\Phi_i$  associadas a cada subrede, para todas as decomposições utilizadas. A fim de ilustrar a decomposição da rede elétrica, as figuras 6.2 e 6.3 apresentam as decomposições do sistema IEEE 118 em 4 e 8 subredes, respectivamente.

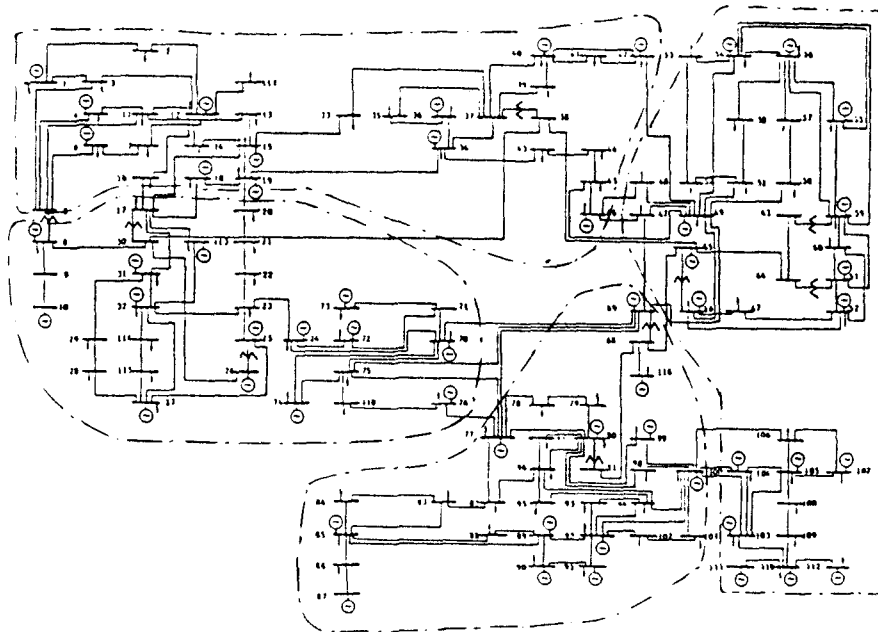


Figura 6.2: Sistema IEEE118 : Decomposição em 4 subredes (barras de interconexão: 5, 17, 20, 38, 46, 49, 68, 69, 77, 100)

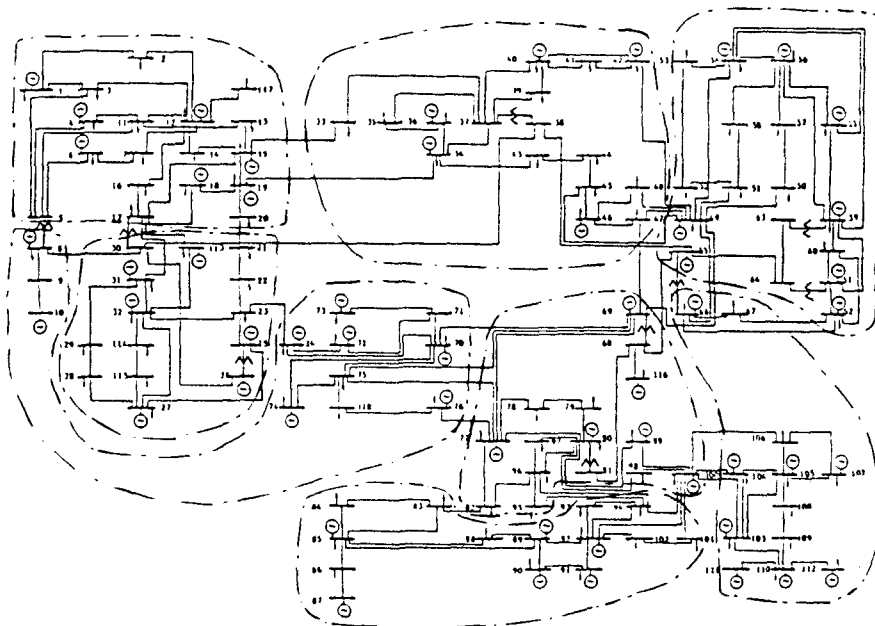


Figura 6.3: Sistema IEEE118 : Decomposição em 8 subredes (barras de interconexão: 5, 17, 20, 23, 30, 33, 34, 49, 62, 65, 69, 77, 82, 94, 100)

Tabela 6.3: Parâmetros das decomposições - IEEE 118 e SSUD 616

	Sistema Elétrico	Nsbs.	Número total de barras, máquinas e dimensões das matrizes $\Phi_i$								I	
			Subsistema No. $k = 1, 2, \dots, Nsbs$									
R e d e	IEEE 118	4	27	26	25	30					10	
		8	15	12	13	12	11	12	14	14	15	
	SSUD 616	2	294	310							12	
		4	155	132	156	153					20	
		8	77	68	75	72	76	78	76	66	28	
		16	34	42	37	35	31	33	37	36		43
			41	32	33	35	36	36	39	36		
		32	19	19	20	19	18	17	15	18		64
			16	18	16	18	18	16	14	19		
			19	18	18	18	15	21	18	14		
21	18		16	15	17	15	12	17				
M á q u i n a s	IEEE 118	4	5	5	5	5						
		8	2	3	2	3	2	3	2	3		
	SSUD 616	2	44	44								
		4	22	22	22	22						
		8	11	11	11	11	11	11	11	11		
		16	5	6	5	6	5	6	5	6		
			5	6	5	6	5	6	5	6		
		32	2	2	3	3	3	3	3	3	3	
			2	2	3	3	3	3	3	3	3	
			2	2	3	3	3	3	3	3	3	
2	2		3	3	3	3	3	3	3			
D i m . $\Phi_i$	IEEE 118	4	6	6	3	6						
		8	5	5	3	4	6	4	6	3		
	SSUD 616	2	12	12								
		4	13	12	5	11						
		8	10	10	8	7	3	6	6	9		
		16	8	7	8	8	8	8	4	3		
			5	8	4	6	5	4	6	4		
		32	1	3	3	3	2	3	6	3		
			6	4	4	4	2	5	5	3		
			4	4	4	5	5	8	8	8		
8	6		7	6	4	6	7	4				
Nsbs. : Número de Subsistemas												
Dim. $\Phi_i$ : Dimensões das matrizes $\Phi_i$												
I : Dimensão do subsistema de Interconexão												

## 6.6.2 Resultados Computacionais

Os tempos de CPU, “speed ups” e eficiências, obtidos nos processamentos paralelos reais, são apresentados na tabela 6.4, para o NCP I, e na tabela 6.5 para o iPSC/860. A fim de estudar o desempenho da metodologia proposta em configurações paralelas de 16 e 32 processadores, não disponíveis atualmente na COPPE/UFRJ, foram realizadas simulações para as decomposições do sistema SSUD 616 em 2, 4, 8, 16 e 32 subsistemas, simulando-se o processamento do algoritmo paralelo AEIGC em um único processador do NCP I. Os resultados obtidos, desprezando-se os tempos de comunicação de dados, são apresentados na tabela 6.6 e na figura 6.5. Nas tabelas 6.4, 6.5 e 6.6 segue-se a seguinte convenção: a) a opção *Sim* refere-se à versão do algoritmo que emprega as matrizes  $\Phi_i$ ; b) para cada par de linhas de resultados, a primeira refere-se à solução da rede elétrica e a segunda ao algoritmo AEIGC global, descontando-se o tempo associado às operações de entrada/saída de dados. Os resultados em termos de “speed up”, podem ser melhor visualizados nas figuras 6.4 e 6.5. Note que os resultados apresentados até então, foram obtidos sem o pré-condicionamento de  $\hat{Y}_s$ .

A figura 6.6 mostra os tempos de CPU dispendidos para o cálculo do resíduo e do produto  $\hat{Y}_s d^k$  no algoritmo Gradiente Conjugado (GC), com e sem o uso das matrizes  $\Phi_i$ . A figura 6.7 mostra o desempenho do método GC em sua formulação convencional e com pré-condicionamento. O pré-condicionamento empregado é do tipo séries truncadas [36] e foram utilizados somente os dois primeiros termos da série.

Tabela 6.4: Tempos de CPU, "speed ups" e eficiências (no NCP I)

SEE	Nsbs.	$\Phi_i$	Tempo de CPU (s)		"Speed up" e Eficiência			
			Sequencial	Paralelo	$S_{a1}$	$E_{a1}$ %	$S_{a2}$	$E_{a2}$ %
I E E E 1 1 8	4	Não	76,7	24,8	2,03	50,8	3,09	77,3
			134,5	43,3	2,53	63,3	3,10	77,6
		Sim	66,6	21,8	2,30	57,7	3,04	76,2
			124,7	40,6	2,70	67,6	3,07	76,8
	8	Não	80,8	17,9	2,81	35,1	4,50	56,2
			138,7	32,2	3,41	42,6	4,30	53,8
		Sim	71,1	17,2	2,93	36,6	4,14	51,7
			129,6	31,7	3,46	43,3	4,09	51,1
S S U D 6 1 6	2	Não	442	210	0.8	38	2.1	105
			715	352	1.2	62	2.0	101
		Sim	379	188	0.8	43	2.0	100
			662	334	1.3	65	2.0	99
	4	Não	463	127	1.3	31	3.6	91
			737	209	2.1	52	3.5	88
		Sim	352	96	1.7	41	3.7	91
			631	175	2.5	62	3.6	90
	8	Não	455	81	2.0	24	5.6	70
			730	128	3.4	42	5.7	71
		Sim	353	68	2.3	29	5.2	64
			629	112	3.9	48	5.6	70

Nsbs. : Número de subsistemas (=  $q$  nos processamentos paralelos reais)  
 $q$  : Número de processadores  
 $\Phi_i$  : Uso das matrizes  $\Phi_i$

Tabela 6.5: Tempos de CPU, "speed ups" e eficiências (no iPSC/860)

SEE	Nsbs.	$\Phi_i$	Tempo de CPU (s)		"Speed up" e Eficiência				
			Sequencial	Paralelo	$S_{a1}$	$E_{a1}$ %	$S_{a2}$	$E_{a2}$ %	
I E E E	4	Não	9,29	5,72	0,89	22,3	1,62	40,6	
			17,4	8,83	1,53	38,2	1,97	49,3	
		Sim	7,86	5,38	0,95	23,7	1,46	36,5	
			16,0	8,50	1,59	39,7	1,88	47,0	
	1 1 8	8	Não	9,55	7,89	0,65	8,08	1,21	15,1
				17,7	10,5	1,24	16,1	1,69	21,0
		Sim	8,10	7,77	0,66	8,20	1,04	13,0	
			16,2	10,4	1,30	16,2	1,56	19,5	
S S U D  6 1 6	2	Não	47,7	24,2	1,10	54,9	1,97	98,5	
			84,1	43,5	1,43	71,6	1,93	96,7	
		Sim	38,3	19,5	1,36	68,2	1,96	98,2	
			75,1	38,7	1,61	80,5	1,94	97,0	
	4	Não	48,5	18,4	1,44	36,1	2,64	65,9	
			85,2	29,7	2,10	52,4	2,87	71,7	
		Sim	40,3	15,4	1,73	43,2	2,62	65,4	
			76,3	26,7	2,33	58,3	2,86	71,4	
	8	Não	48,3	17,9	1,49	18,5	2,70	33,7	
			84,9	25,4	2,45	30,6	3,34	41,8	
		Sim	40,2	16,4	1,62	20,2	2,45	30,6	
			76,1	23,9	2,61	32,6	3,18	39,8	

Nsbs. : Número de subsistemas (=  $q$  nos processamentos paralelos reais)  
 $q$  : Número de processadores  
 $\Phi_i$  : Uso das matrizes  $\Phi_i$



Tabela 6.6: Tempos de CPU, “speed ups” e eficiências desprezando-se a comunicação (no NCP I)

SEE	Nsbs	$\Phi_i$	Tempo de CPU (s)		“Speed up”	Eficiência
			Algoritmo AEIC seqüencial	Algoritmo AEIGC paralelo		
			$S_{a1}$	$E_{a1}$ (%)		
S S U D  6 1 6	2	Não	161	204	0,79	39,3
			436	336	1,30	64,8
	2	Sim	161	175	0,92	45,8
			436	308	1,41	70,6
	4	Não	161	111	1,44	36,1
			436	180	2,41	60,3
	4	Sim	161	85	1,88	47,0
			436	155	2,80	70,0
	8	Não	161	61	2,64	33,0
			436	95	4,60	57,3
	8	Sim	161	48	3,36	42,0
			436	82	5,30	66,1
	16	Não	161	35	4,52	28,3
			436	53	8,24	51,5
	16	Sim	161	28	5,70	35,6
			436	46	9,50	59,4
	32	Não	161	25	6,34	19,8
			436	34	12,8	40,1
	32	Sim	161	22	7,22	22,6
			436	31	14,1	43,8

Nsbs : Número de subsistemas  
 $\Phi_i$  : Uso das matrizes  $\Phi_i$   
 AEIC : Alternado Entrelaçado Implícito Convencional

## Comentários

- O algoritmo AEIGC paralelo propicia uma redução substancial no tempo de processamento, comparativamente a um dos melhores esquemas de resolução seqüencial das EDAs que descrevem o comportamento dinâmico de um SEE. A última linha de dados da tabela 6.4 mostra que o tempo de processamento obtido, quando empregam-se 8 processadores, é reduzido de aproximadamente 4 vezes. As estimativas de desempenho para configurações computacionais paralelas com 16 e 32 processadores, tabela 6.6 e figura 6.5, indicam potencialidades ainda maiores de redução no tempo de processamento.
- O uso das matrizes  $\Phi_i$  resultou em desempenho consistentemente melhor, relativamente à versão do algoritmo AEIGC que não as emprega. Isto pode ser visualizado nas figuras 6.5 e 6.6;
- A introdução de técnicas de pré-condicionamento também produz uma significativa redução no número de iterações requeridas pelo método GC, conforme mostrado na figura 6.7, embora nem sempre acompanhado por uma correspondente redução no tempo de processamento;
- A decomposição da rede elétrica tem influência significativa no desempenho do algoritmo AEIGC paralelo, devido principalmente às suas implicações no balanceamento de carga entre os processadores, conforme ilustrado na figura 6.6;
- Os tempos de CPU no iPSC/860 são cerca de uma ordem de grandeza inferiores aos do NCP I (ver tabelas 6.4 e 6.5). Contudo, os “speed ups” são superiores para o NCP I ( ver figura 6.4).

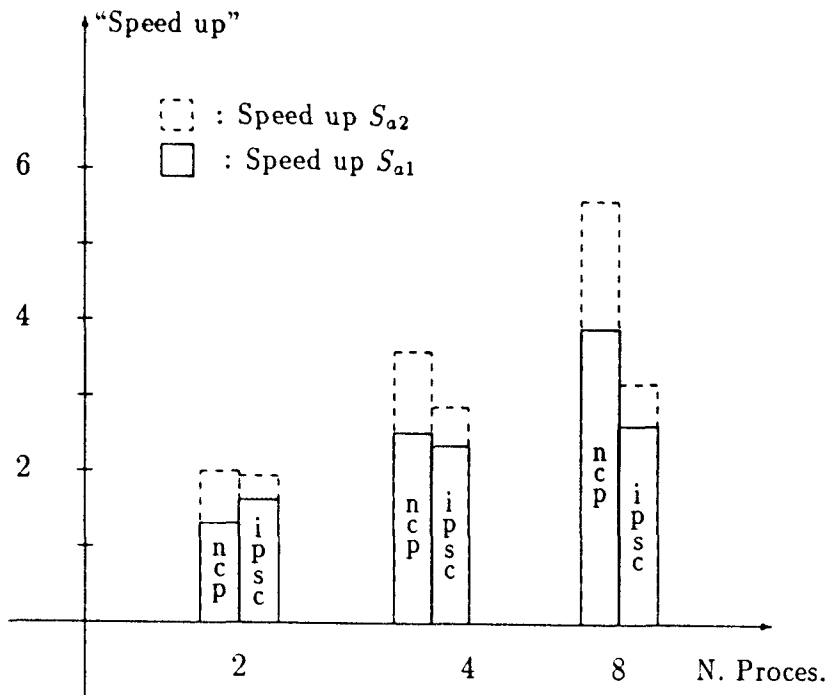


Figura 6.4: "Speed ups"  $S_{a1}$  e  $S_{a2}$  nos computadores NCP I e iPSC/860

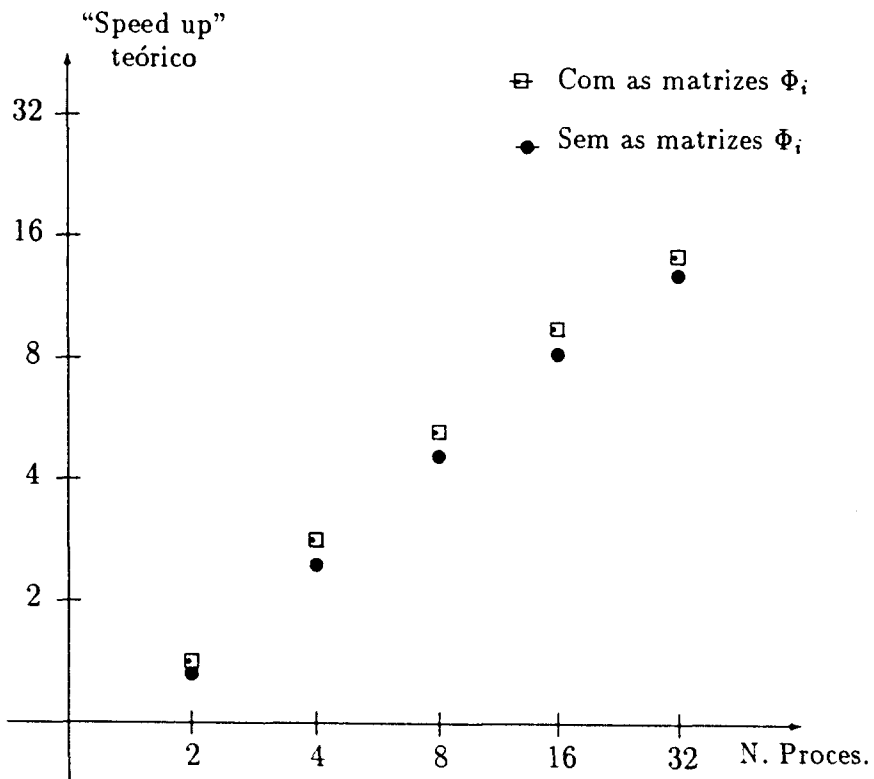


Figura 6.5: "Speed up" obtido desprezando-se a comunicação (NCP I)

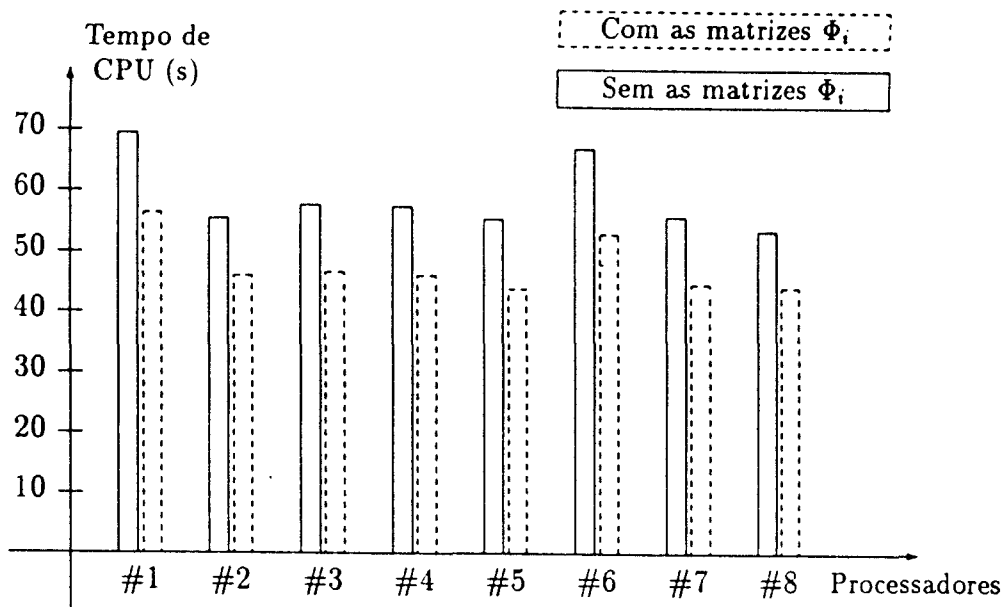


Figura 6.6: Tempo de CPU para o cálculo do resíduo e de  $\hat{Y}, d^k$  (NCP I)

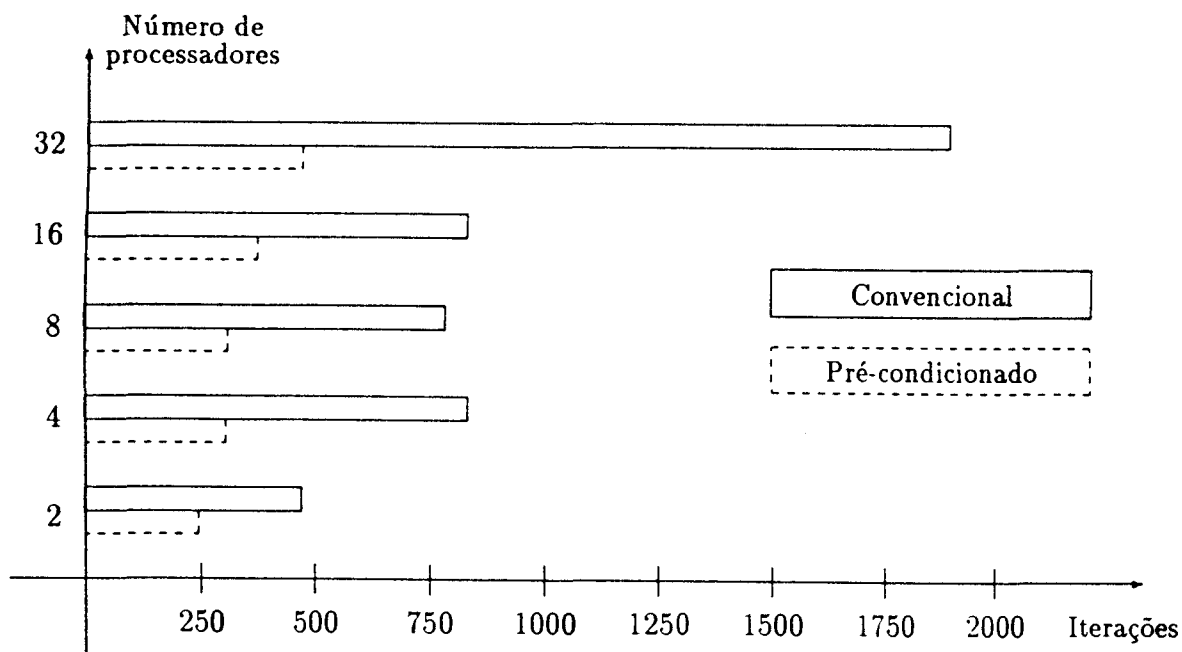


Figura 6.7: Número total de iterações do método GC (NCP I)

## 6.7 Resultados para o Algoritmo AEIGCRD

Neste item são apresentados e discutidos os resultados obtidos com a simulação da dinâmica dos sistemas IEEE 118 e SSUD 616, usando-se o algoritmo AEIGCRD e o computador paralelo iPSC/860 somente. Este algoritmo paralelo também requer a decomposição do sistema de equações em  $q$  subsistemas, onde  $q$  é o número de processadores.

### 6.7.1 Parâmetros das Decomposições do Sistema

A decomposição do sistema de equações diferenciais é feita da mesma forma que para o algoritmo AEIGC, ou seja, independentemente da decomposição das equações da rede elétrica e expressa em termos do número de máquinas síncronas atribuídas a cada processador. A rede elétrica necessita ser decomposta em  $q$  subredes tal que a matriz admitância nodal  $Y$  assuma a forma NBDF. Note que essa forma específica é exclusivamente em função da utilização de um pré-condicionamento do tipo fatoração LU incompleta, conforme definido no capítulo 4.

As decomposições utilizadas foram classificadas em duas categorias, a saber:

**Decomposições não Automatizadas :** Obtidas a partir da análise do diagrama unifilar, tal como foi feito nos experimentos realizados com o algoritmo AEIGC, apresentados no item anterior, sem nenhum procedimento automatizado;

**Decomposições Automatizadas :** Obtidas através de procedimentos automatizados que tentam encontrar subredes fracamente acopladas entre si, com um mínimo de linhas de interligação e com dimensões tais que propiciem um razoável balanceamento da carga entre os processadores [61].

Os procedimentos automatizados foram utilizados somente em decomposições do sistema SSUD 616. Para o sistema IEEE 118 foram feitas decomposições em 4 e 8 subsistemas somente, em função de suas reduzidas dimensões para propósitos de processamento paralelo. Por outro lado, no sistema SSUD 616 foram feitas decomposições em 2, 4, 8, 16 e 32 subsistemas. Foram gerados um número elevado de decomposições obtidas por meios automatizados, com o objetivo de explorar a influência da decomposição na razão de convergência do algoritmo GC, no balanceamento da carga entre

os processadores e no tempo gasto com a comunicação de dados entre os processadores. Neste trabalho apresenta-se somente uma síntese desses resultados; os dados completos estão na referência [54].

A tabela 6.7 apresenta o número de barras de cada subrede e o número de máquinas alocadas a cada processador, para as decomposições não automatizadas (N. Aut.) e para as decomposições automatizadas que apresentaram os melhores resultados (Aut.).

## 6.7.2 Resultados Computacionais

Os resultados apresentados neste subitem referem-se a processamentos realizados usando-se as versões paralela e seqüencial do algoritmo AEIGCRD e o computador paralelo Intel iPSC/860 da COPPE/UFRJ. Em face do número limitado de processadores, os processamentos paralelos reais limitaram-se a configurações de 4 e 8 processadores, para o sistema IEEE 118, e 2, 4 e 8 processadores para o sistema SSUD 616. Por outro lado, nos processamentos seqüenciais e para o sistema SSUD 616, foram feitas simulações com decomposições do sistema em 2, 4, 8, 16 e 32 subsistemas.

A tabela 6.8 resume os resultados obtidos em termos de tempos de CPU, “speed ups” e eficiências; nos processamentos paralelos reais e configurações de até 8 processadores. Para cada par de linhas de resultados, a primeira refere-se a solução da rede elétrica e a segunda ao algoritmo AEIGCRD completo, descontando-se o tempo associado às operações de entrada/saída de dados.

Na figura 6.8 apresentam-se, para o sistema SSUD616 e decomposições não automatizadas, os tempos de CPU de diversas parcelas do algoritmo AEIGCRD em processamentos seqüenciais. O objetivo é ilustrar a participação de cada parcela na formação do tempo total de CPU do algoritmo.

Na figura 6.9 apresenta-se uma síntese comparativa dos resultados, para o sistema elétrico SSUD 616, em termos dos tempos de CPU e comunicação, em processamentos paralelos e seqüenciais, para decomposições automatizadas (retângulos em linha cheia) e não automatizadas (retângulos em linha tracejada). Os resultados completos encontram-se em [54].

Tabela 6.7: Parâmetros das decomposições - IEEE 118 e SSUD 616

De-compo-sição		SEE	Nsbs.	Número total de barras e máquinas para cada subsistema												
				Subsistema No. $k = 1, 2, \dots, N_{sbs}$												
Máquinas		IEEE 118	4	5	5	5	5									
			8	2	3	2	3	2	3	2	3	2	3			
		SSUD 616	2	44	44											
			4	22	22	22	22									
			8	11	11	11	11	11	11	11	11	11	11	11		
			16	5	6	5	6	5	6	5	6	5	6	5	6	
				5	6	5	6	5	6	5	6	5	6	5	6	
			32	2	2	3	3	3	3	3	3	3	3	3	3	
				2	2	3	3	3	3	3	3	3	3	3	3	
				2	2	3	3	3	3	3	3	3	3	3	3	
2	2	3		3	3	3	3	3	3	3	3	3				
Rede	N. Aut.	IEEE 118	4	30	28	29	31									
			8	18	12	14	14	15	12	16	17					
		SSUD 616	2	300	316											
			4	163	137	156	160									
			8	84	71	79	78	78	78	78	70					
			16	38	47	37	41	36	35	39	36					
				41	36	37	37	39	38	39	40					
			32	20	19	20	20	19	19	19	20					
				18	20	17	20	18	17	15	20					
				21	21	20	20	19	23	23	17					
				23	21	19	17	19	18	15	19					
			Rede	Aut.	SSUD 616	2	283	333								
						4	153	149	154	160						
						8	74	72	76	74	88	77	77	78		
16	39	50				36	40	48	37	39	42					
	47	32				40	50	19	35	35	27					
32	20	12				21	16	32	10	5	6					
	22	18				16	16	1	4	2	16					
	16	10				22	8	37	86	8	6					
	8	38				38	12	32	53	2	23					
Nsbs. : Número de Subsistemas																
N. Aut. : Decomposições não automatizadas																
Aut. : Decomposições automatizadas																

Tabela 6.8: Tempos de CPU, "speed ups" e eficiências (no iPSC/860)

	SEE	Nsbs.	O.C.	Tempo de CPU (s)		"Speed up" e Eficiência			
				Sequencial	Paralelo	$S_{a1}$	$E_{a1}$ %	$S_{a2}$	$E_{a2}$ %
N . A u t . 1 1 8	I E E	4	0	7,52	6,11	0,83	20,8	1,23	30,7
				15,9	9,58	1,41	35,2	1,66	41,5
			4	7,52	5,97	0,85	21,3	1,26	31,5
				15,9	9,39	1,43	35,9	1,69	42,3
		8	0	7,61	7,21	0,71	8,84	1,05	13,2
			16,1	10,8	1,25	15,6	1,49	18,6	
		4	7,63	7,14	0,71	8,90	1,07	13,3	
			16,0	9,96	1,36	16,9	1,61	20,0	
N . A u t . 6 1 6	S S U	2	0	66,6	36,6	0,72	36,3	1,82	90,1
				102,7	56,6	1,10	55,0	1,81	90,1
			4	60,9	31,1	0,86	42,7	1,96	97,9
				95,3	50,5	1,23	61,7	1,89	94,3
	D	4	0	64,9	28,2	0,94	23,6	2,30	57,5
				101,0	40,0	1,56	38,9	2,52	63,1
			4	59,8	25,9	1,03	25,7	2,31	57,7
				95,5	37,4	1,67	41,6	2,55	63,8
	6	8	0	105,3	38,0	0,70	8,75	2,77	34,6
				141,7	46,1	1,35	16,9	3,07	38,4
		4	92,8	35,9	0,74	9,26	2,58	32,3	
			128,8	43,6	1,43	17,8	2,95	36,9	
A u t . 6 1 6	S S U	2	0	36,3	21,9	1,21	60,7	1,66	82,8
				73,1	41,6	1,50	74,9	1,76	87,8
			4	36,2	18,6	1,43	71,5	1,95	97,3
				71,3	37,6	1,66	82,8	1,90	94,8
	D	4	0	38,1	17,7	1,50	37,6	2,23	55,7
				75,2	29,4	2,12	53,0	2,56	63,9
			4	38,0	14,8	1,80	45,0	2,57	64,2
				73,3	26,0	2,40	60,0	2,82	70,5
	6	8	0	45,6	17,6	1,51	18,9	2,59	32,4
				83,7	25,8	2,42	30,2	3,24	40,5
		4	44,6	15,7	1,69	21,2	2,84	35,5	
			80,9	23,5	2,65	33,1	3,44	43,0	
<p>Nsbs. : Número de subsistemas (= <math>q</math> nos processamentos paralelos reais)  <math>q</math> : Número de processadores  O.C. : Opção de compilação  N. Aut. : Decomposição não automatizada  Aut. : Decomposição automatizada</p>									



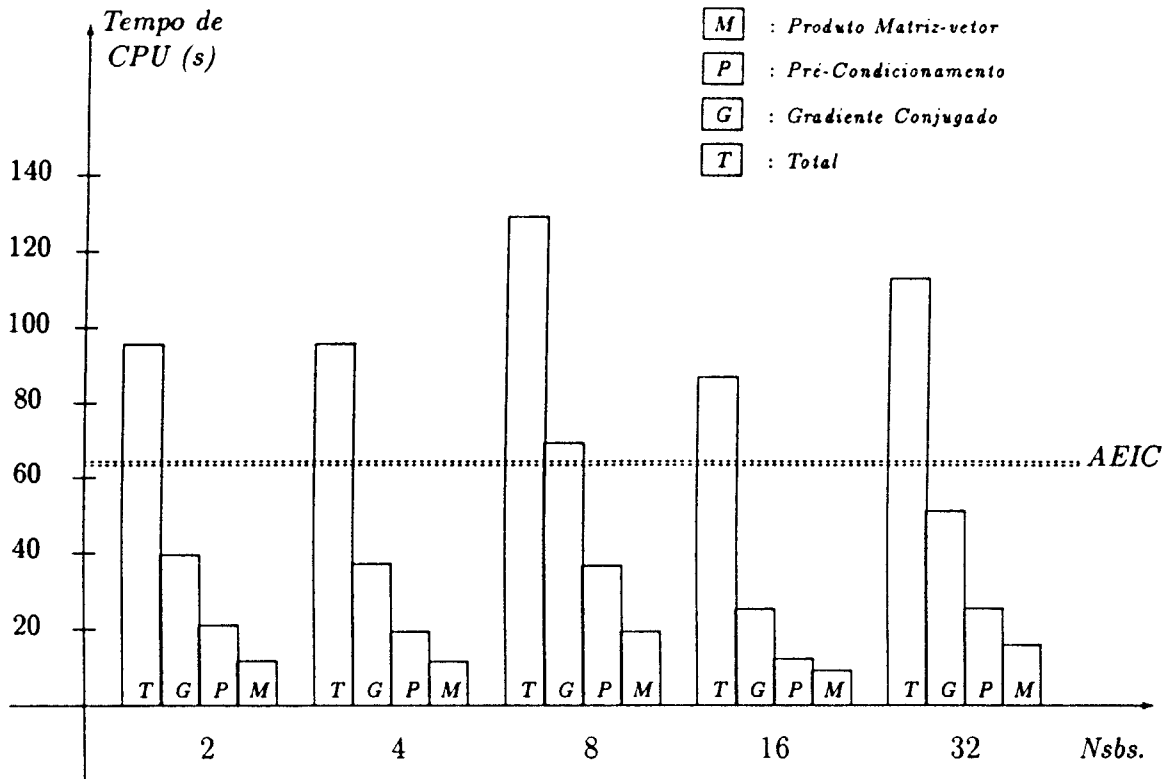


Figura 6.8: Tempos de CPU para diversas parcelas do algoritmo AEIGCRD (seqüencial)

### Comentários

- O algoritmo AEIGCRD, em processamentos paralelos reais, apresentou uma significativa redução nos tempos de CPU. A dimensão desses ganhos mostrou-se sensível a decomposição, conforme pode-se observar na tabela 6.8 e na figura 6.9. Observa-se também que o tempo de comunicação cresce com o número de processadores, atingindo cerca de 50 % do tempo total de CPU para 8 processadores. Isso é devido principalmente a características peculiares do iPSC/860, mais favoráveis em outras máquinas paralelas tais como: NCP I, Intel Delta, Intel Paragon, etc., e será tratado no próximo item;
- O desempenho foi significativamente superior na maioria dos casos de decomposições automatizadas ( ver figura 6.9). Os instrumentos de decomposição automática utilizados encontram-se ainda em fase de desenvolvimento com a perspectiva da obtenção de melhores resultados;

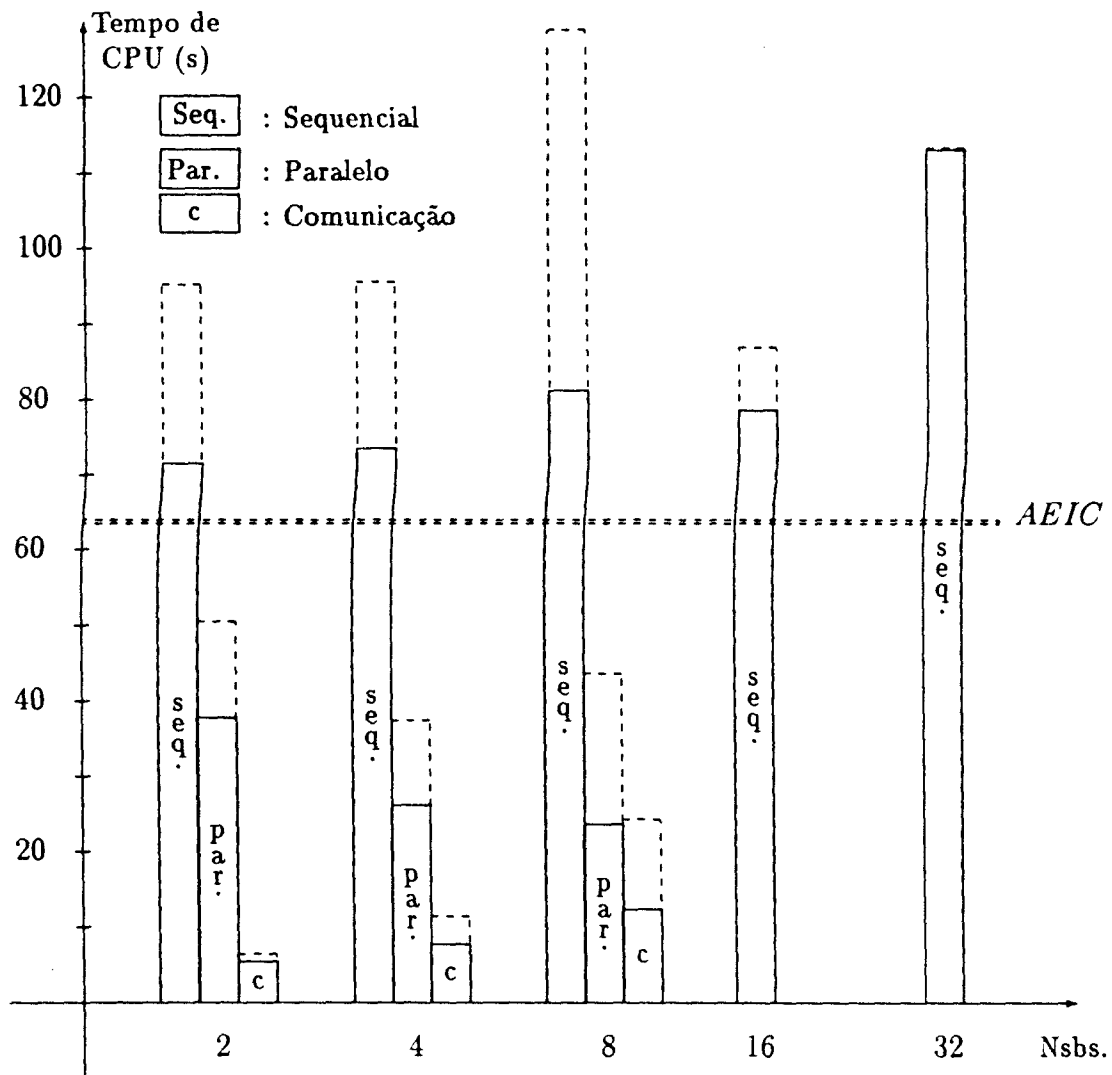


Figura 6.9: Tempos de CPU e comunicação, em processamentos seqüenciais e paralelos, para decomposições automatizadas (linha cheia) e não automatizadas (linha tracejada)

- O nível de otimização 4, na compilação dos programas, propiciou uma redução nos tempos de CPU na faixa de 5 a 10 %, para o tempo total de simulação, e na faixa de 15 a 20 % para o algoritmo Gradiente Conjugado pré-condicionado. Isto confirma a maior potencialidade para a vetorização no algoritmo GC;
- O número de iterações no algoritmo GC pode variar significativamente com a decomposição, todavia, isto não acontece nos laços iterativos mais externos do algoritmo AEIGCRD (AEI e Rede). Esta sensibilidade se reflete nos tempos de processamentos. Por exemplo, no caso de decomposições não automatizadas e processamento seqüencial, o pior e o melhor desempenhos ocorreram, respectivamente, para as decomposições em 8 e 16 subsistemas. Nestes dois casos extremos, a variação total no tempo de CPU foi cerca de 50 %, tomando-se o menor deles como referência;
- Nos processamentos seqüenciais, o pré-condicionamento consome cerca de 50 % do tempo de CPU gasto pelo algoritmo GC (ver figura 6.8). Todavia, isso não se verifica nos processamentos paralelos em razão da realização do pré-condicionamento com paralelismo perfeito [54];
- Na figura 6.9 também observa-se que, para processamentos seqüenciais, o algoritmo AEIGCRD foi, nas decomposições que apresentaram os melhores resultados, somente 20 a 30 % inferior ao algoritmo AEIC, diferenças essas que poderão ser significativamente reduzidas com o desenvolvimento de melhores pré-condicionadores e com a exploração das potencialidades de vetorização existentes no AEIGCRD.

## 6.8 Análise Comparativa dos Resultados - Paralelização no Espaço

Neste item faz-se uma análise comparativa de desempenho para os algoritmos AEIGC e AEIGCRD, ou seja, das paralelizações no espaço realizadas usando-se o esquema *Alternado Entrelaçado Implícito*.

Na tabela 6.9 apresentam-se os tempos de CPU, "speed ups" e eficiências, obtidos em simulações seqüenciais e paralelas usando-se o sistema SSUD 616, os algoritmos AEIGC e AEIGCRD e os computadores NCP I e iPSC/860. Também nesta tabela e para fins de comparação, encontram-se os tempos de CPU em processamentos seqüenciais do algoritmo AEIC em diversas máquinas. Na figura 6.10 apresenta-se uma outra forma de visualização dos melhores resultados obtidos exclusivamente nos processamentos realizados no iPSC/860, com a apresentação dos tempos gastos na comunicação de dados entre os processadores. Os retângulos tracejados referem-se ao algoritmo AEIGCRD e os em linha cheia, ao algoritmo AEIGC. A linha horizontal tracejada indica o tempo de CPU para o algoritmo AEIC seqüencial, tomado como padrão.

Tabela 6.9: Tempos de CPU, "speed ups" e eficiências - paralelização no espaço

Algoritmo	Computador	q	CPU (s)	"Speed up" e Eficiência			
				$S_{a1}$	$E_{a1}$ %	$S_{a2}$	$E_{a2}$ %
AEIGC	NCP I	2	334	1,30	65	2,00	100
		4	175	2,50	62	3,60	90
		8	112	3,90	48	5,60	70
AEIGC	iPSC/860	2	38,7	1,61	80	1,95	98
		4	26,7	2,33	58	2,85	71
		8	23,9	2,60	33	3,18	40
AEIGCRD	iPSC/860	2	37,6	1,66	83	1,90	95
		4	26,0	2,40	60	2,82	71
		8	23,5	2,65	33	3,44	43
AEIC	NCP I	1	436	q : Num. de processadores CPU : Tempos de CPU (s)			
	IBM 4381	1	194				
	IBM 3090	1	20				
	iPSC/860	1	62,3				

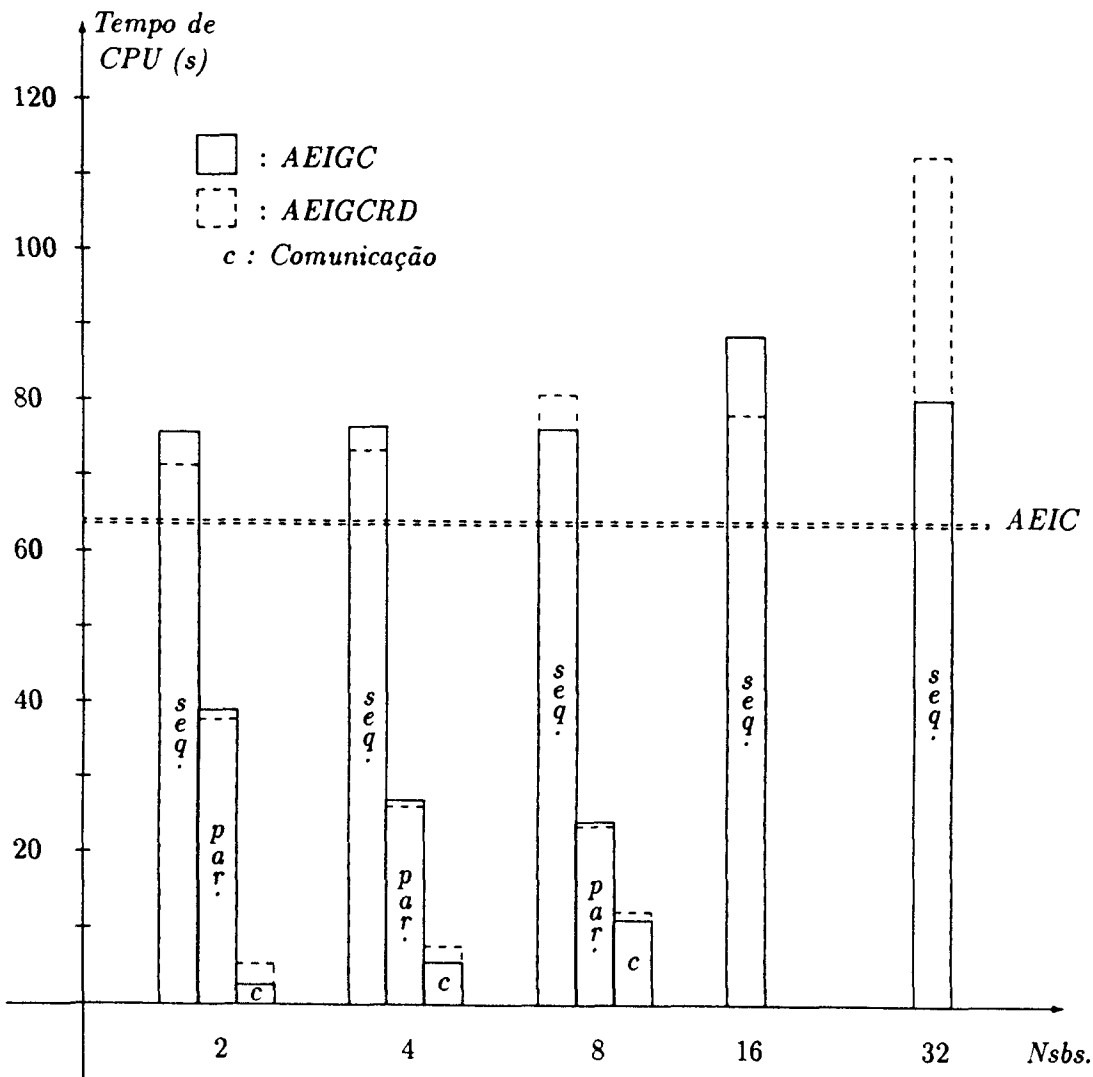


Figura 6.10: Melhores tempos de CPU e comunicação, em processamentos seqüenciais e paralelos no iPSC/860, para os algoritmos AEIC, AEIGC e AEIGCRD

## Comentários

- Observa-se na tabela 6.9 que o processamento paralelo reduziu significativamente os tempos das simulações dinâmicas realizadas. Além disso obteve-se tempos comparáveis aos de “mainframes” que são muitas vezes mais caros. Ilustra-se este fato destacando-se dois resultados: 1) Usando-se 4 Transputers, o tempo de CPU foi inferior ao de um IBM 4381. Observa-se que atualmente é possível adquirir placas com 4 transputers T800, acopláveis a um microcomputador pessoal, a um custo inferior a US \$ 10,000; 2) os tempos de CPU no iPSC/860 são comparáveis aos de um IBM 3090.
- O algoritmo AEIGCRD foi levemente superior, mesmo tendo tempos de comunicação maiores e sendo mais sensível a decomposição que o AEIGC;
- Na configuração de 8 processadores, os valores de “speed up” e eficiência, para o NCP I, são significativamente superiores. Isto é devido a razão capacidade de processamento pela capacidade de comunicação, muito superior no iPSC/860.
- Uma parcela significativa da comunicação de dados entre os procesadores independe da dimensão do sistema, indicando que os índices de desempenho devem ser ainda melhores em sistemas de maior dimensão. Além disso, a literatura fora da área de sistema de energia elétrica registra, em estudos realizados com processamentos seqüenciais, maior competitividade dos métodos tipo GC, em relação ao métodos diretos, para sistemas com dimensões muito superiores as utilizadas nos testes realizados neste trabalho.

## 6.9 Resultados para o Algoritmo SIMPT

Neste item apresentam-se os resultados obtidos nas simulações da dinâmica dos sistemas IEEE 118 e SSUD 616, usando-se o algoritmo SIMPT desenvolvido, em processamentos paralelos e seqüenciais no computador Intel iPSC/860. O número de passos de integração tomados simultaneamente (janela de integração) foram de 1, 2, 4 e 8, para os processamentos seqüenciais, e 2, 4 e 8 para os processamentos paralelos. A apresentação dos resultados é feita através de tabelas e figuras, elaboradas de modo a permitir a visualização de diversos aspectos do algoritmo, a saber:

- Método de resolução da equação linear Jacobiana ( BCG ou Bi-CGSTAB);
- Técnica de pré-condicionamento;
- Técnica de inicialização das variáveis da janela de integração (“flat start”, série de Taylor em  $x$  ou em  $x$  e  $V^e$ );
- Parcelas do algoritmo com maior demanda computacional ( total, resolução da equação Jacobiana, pré-condicionamento e produtos do tipo matriz-vetor);
- Índices de desempenho ( tempos de CPU, “speed ups” e eficiências);
- Potencialidades para a vetorização;
- Tempos gastos na comunicação versus o tamanho da janela de integração;

Na tabela 6.10 encontram-se os tempos de CPU, totais e para a resolução da equação Jacobiana, pelos métodos BCG e Bi-CGSTAB pré-condicionados, em processamentos paralelos reais da simulação da dinâmica dos dois sistemas elétricos de testes já citados. A tabela 6.11 apresenta o mesmo tipo de resultado da tabela 6.10, porém, para os processamentos seqüenciais realizados.

Na figura 6.11 ilustra-se, para esses mesmos processamentos, o efeito do crescimento do tamanho da janela de integração no número médio de iterações do algoritmo de resolução da equação linear Jacobiana. Essa média é feita em relação a cada iteração no algoritmo de Newton. Os retângulos em linha cheia referem-se a utilização do método Bi-CGSTAB pré-condicionado e os retângulos em linha tracejada, referem-se a utilização do método BCG pré-condicionado. Ainda nesta mesma figura, mostra-se o efeito do método de inicialização das variáveis da janela de integração.

Para cada conjunto de retângulos associados a um determinado número de passos de integração tomados simultaneamente, o mais da esquerda refere-se a inicialização das variáveis de estado ( $x$ ) por série de Taylor com “flat start” para as tensões ( $V^e$ ); e os retângulos da direita referem-se a inicialização de todas as variáveis ( $x$  e  $V^e$ ) por série de Taylor.

Na figura 6.12 o objetivo é ilustrar o comportamento do número total de iterações com o crescimento do tamanho da janela de integração, na solução do sistema não-linear ( algoritmo de Newton ) e na solução do sistema linear ( algoritmo Bi-CGSTAB ). Os retângulos tracejados referem-se a inicialização por série de Taylor somente nas variáveis de estado ( $x$ ) e os em linha cheia a inicialização por série de Taylor em todas as variáveis do problema ( $x$  e  $V^e$ ). No eixo das abscissas também apresenta-se a dimensão da equação Jacobina, para o sistema SSUD 616, associadas ao tamanho da janela de integração.

Na figura 6.13 apresenta-se a participação de diversas parcelas do algoritmo SIMPT na formação do tempo total de CPU, em processamentos seqüenciais. Os retângulos tracejados estão associados a processamentos realizados sem nenhuma otimização na compilação dos programas (nível 0); e os retângulos em linha cheia estão associados a processamentos realizados a partir da utilização da opção de otimização máxima, na compilação (nível 4), ativando o software que explora o uso eficiente dos “pipelines” do processador i860. Esta última opção permite que se tenha uma perspectiva das potencialidades de vetorização do algoritmo SIMPT.

### Comentários

- Os tempos de CPU, em processamentos seqüenciais, crescem com o aumento do número de passos de integração  $p$  tomados simultaneamente, porém, a uma taxa inferior a 50 % da taxa de crescimento de  $p$ . Este detalhe permite uma expectativa de ganhos em computadores paralelos;
- Os índices de desempenho foram significativamente melhores para o sistema de 616 barras, comparativamente aos do sistema de 118 barras, sinalizando que maiores ganhos poderão ser obtidos em SEE de maior dimensão.
- A série de Taylor mostrou-se eficaz como meio de inicialização em uma janela de integração, tendo apresentado melhor desempenho quando empregada em



Tabela 6.10: Tempos de CPU (s) em processamentos paralelos

SEE	Ini- ciali- zação	Opção de Compil.	Num. de passos Simult.	Tempo de CPU (s)			
				C/ BCG		C/ Bi-CGSTAB	
				Total	E. Jab.	Total	E. Jab.
IEEE 118	Série de Taylor em $x$	0	2	12,4	8,27	10,6	6,51
			4	10,6	8,13	9,07	6,67
			8	8,92	7,43	8,51	7,02
	4	2	10,3	6,67	9,05	5,44	
		4	8,98	6,82	7,84	5,72	
		8	7,64	6,28	7,45	6,10	
SSUD 616	Série de Taylor em $x$	0	2	53,1	34,2	44,4	25,5
			4	41,9	31,1	34,4	23,5
			8	nc.	nc.	30,1	23,5
	4	2	42,5	26,3	35,9	19,9	
		4	33,6	24,3	28,1	18,8	
		8	nc.	nc.	24,6	18,8	

nc. : O algoritmo BCG não convergiu

todas as variáveis do problema;

- Os tempos de comunicação de dados mantém-se praticamente constante com o crescimento do número de processadores;
- o método Bi-CGSTAB foi cerca de 20 % superior ao BCG, em termos de tempos de CPU e número de iterações, na resolução da equação linear Jacobiana;
- Há um significativo ganho com a utilização do nível de otimização 4 na compilação dos programas, ou seja, com a utilização dos recursos dos “pipelines” do processador Intel i860;
- O número médio de iterações na resolução do sistema linear Jacobiano, manteve-se em patamares bastante baixos, porém, os valores são crescentes com  $p$  e da mesma forma que os tempos de CPU;
- Observa-se mais claramente na figura 6.13, que o tempo gasto com o pré-condicionamento é elevado e cerca da metade do tempo consumido na resolução global seqüencial do sistema linear Jacobiano.

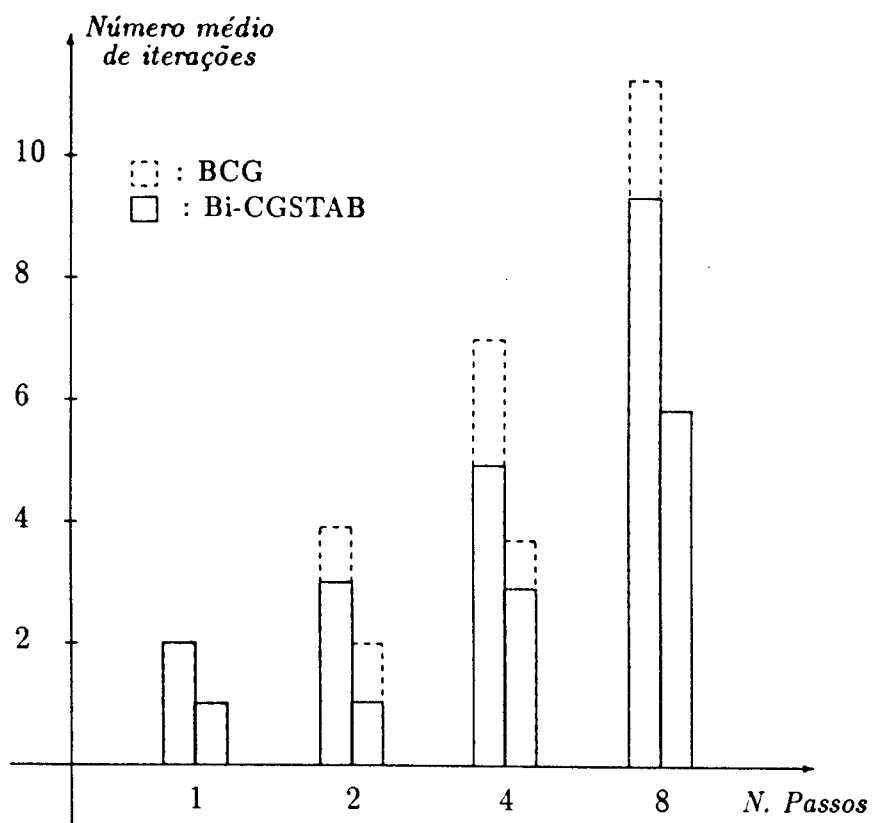


Figura 6.11: Número médio de iterações nos algoritmos tipo GC para a inicialização por série de Taylor em  $x$  (esquerda) e em  $x$  e  $V^e$  (direita)

Tabela 6.11: Tempos de CPU (s) em processamentos seqüenciais

SEE	Ini- ciali- zação	Opção de Compil.	Num. de passos Simult.	Tempo de CPU (s)			
				C/ BCG		C/ Bi-CGSTAB	
				Total	E. Jab.	Total	E. Jab.
IEEE 118	Série de Taylor em $x$	0	1	13,5	7,22	12,7	6,45
			2	19,3	13,4	15,7	9,65
			4	28,7	22,7	21,9	15,9
			8	nc.	nc.	36,5	30,4
	4	4	1	10,8	5,41	10,2	4,79
			2	14,9	9,79	12,2	7,01
			4	21,3	16,2	16,4	11,4
			8	nc.	nc.	26,5	21,4
	Série de Taylor em $x$ e $V^e$	0	1	9,86	3,48	8,95	2,69
			2	12,5	5,79	9,87	3,23
			4	17,5	10,8	14,7	8,23
			8	29,3	22,6	30,7	24,1
4		4	1	8,38	2,67	7,38	2,02
			2	10,3	4,29	8,02	2,37
			4	13,6	7,77	11,4	5,88
			8	21,8	15,9	22,5	16,9
SSUD 616	Série de Taylor em $x$	0	1	65,5	35,5	61,6	31,6
			2	97,7	67,5	77,2	47,8
			4	143,6	114,5	109,0	79,9
			8	228,4	197,6	186,7	157,0
	4	4	1	51,1	25,8	47,3	22,4
			2	72,8	48,0	57,7	33,6
			4	105,0	80,9	79,7	55,8
			8	165,1	139,1	134,9	110,2
	Série de Taylor em $x$ e $V^e$	0	1	53,8	20,7	49,1	15,9
			2	68,1	35,6	48,7	16,2
			4	92,8	60,8	78,2	46,1
			8	nc.	nc.	129,4	96,8
4		4	1	44,7	15,5	39,7	11,5
			2	54,3	25,8	39,0	11,6
			4	71,2	43,2	59,5	32,6
			8	nc.	nc.	95,2	68,0

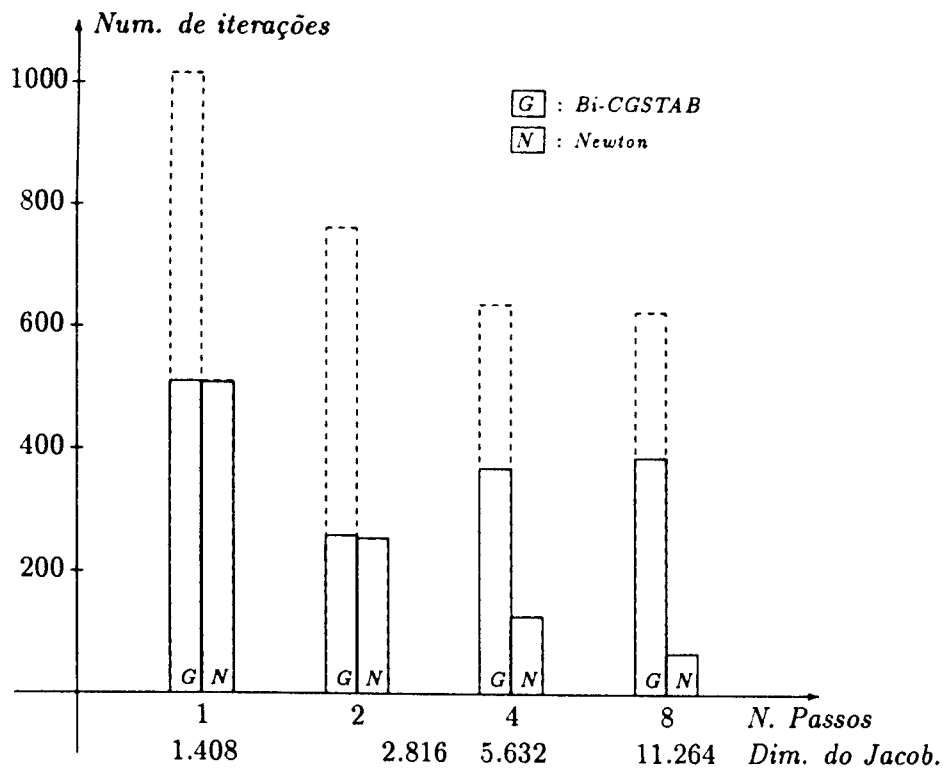


Figura 6.12: Número total de iterações para os algoritmos de Newton e Bi-CGSTAB, com a inicialização por série de Taylor em  $x$  (linha tracejada) e em  $x$  e  $V^e$  (linha cheia)

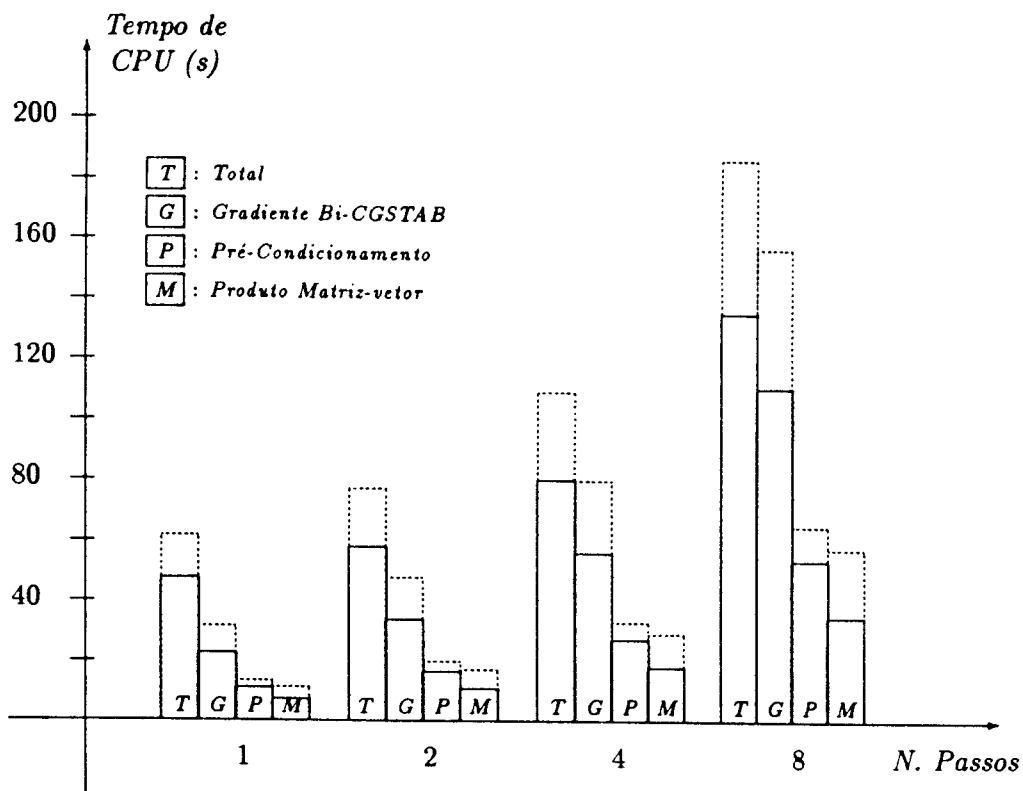


Figura 6.13: Tempos de CPU para diversas parcelas do algoritmo SIMPT, em processamentos seqüenciais com níveis de otimização 0 (linha tracejada) e 4 (linha cheia) na fase de compilação

### 6.9.1 Avaliação Comparativa de Desempenho - Paralelização no Tempo

Neste subitem procura-se quantificar e analisar, em termos das figuras de mérito do tipo “speed up” e eficiência, definidas anteriormente, os resultados dos processamentos, paralelos e seqüenciais, realizados com o algoritmo SIMPT.

Nas tabelas 6.12 e 6.13, são apresentados os resultados, respectivamente, para os “speed ups”  $S_{,1}$  e  $S_{,2}$ . Nestas tabelas, *E. Jab.* é o “speed up” obtido na solução da equação linear Jacobiana; *Total* é o “speed up” global obtido para o algoritmo SIMPT; *BCG* refere-se a solução da equação linear Jacobiana pelo método Gradiente Biconjugado pré-condicionado; e *Bi-CGSTAB* refere-se a solução da equação linear Jacobiana pelo método Bi-CGSTAB pré-condicionado.

Tabela 6.12: “Speed ups” relativos ao processamento seqüencial com  $p$  passos tomados simultaneamente

SEE	Inici- ciami- zação	Opção de Compil .	Num. de passos Simult.	“Speed up” $S_{,1}$			
				C/ BCG		C/ Bi-CGSTAB	
				Total	E. Jab.	Total	E. Jab.
IEEE 118	Série de Taylor em $x$	0	2	1,55	1,62	1,48	1,48
			4	2,70	2,79	2,41	2,38
			8	-	-	4,29	4,33
	4	2	1,45	1,47	1,35	1,29	
		4	2,37	2,37	2,09	1,99	
		8	-	-	3,56	3,51	
SSUD 616	Série de Taylor em $x$	0	2	1,84	1,97	1,74	1,87
			4	3,43	3,68	3,17	3,40
			8	-	-	6,20	6,68
	4	2	1,71	1,82	1,61	1,69	
		4	3,12	3,33	2,83	2,97	
		8	-	-	5,48	5,86	

Na figura 6.14 mostram-se tempos de CPU e comunicação ( $c$ ) em processamentos, paralelos (*Par.*) e seqüenciais (*Seq.*), realizados com o algoritmo SIMPT no computador iPSC/860. Os retângulos tracejados referem-se a utilização do método BCG pré-condicionado e os em linha cheia a utilização do método Bi-CGSTAB pré-condicionado, na resolução da equação matricial linear Jacobiana.  $p$  é o número de passos de integração tomados simultaneamente e também o número de processadores

Tabela 6.13: “Speed ups” relativos ao processamento seqüencial com 1 passo de integração

SEE	Ini- ciali- zação	Opção de Compil .	Num. de passos Simult.	“Speed up” $S_{s2}$			
				C/ BCG		C/ Bi-CGSTAB	
				Total	E. Jab.	Total	E. Jab.
IEEE 118	Série de Taylor em $x$	0	2	1,09	0,87	1,20	0,99
			4	1,27	0,89	1,40	0,97
			8	1,51	0,97	1,49	0,92
	4	2	1,05	0,81	1,13	0,88	
		4	1,20	0,79	1,30	0,83	
		8	1,41	0,86	1,37	0,78	
SSUD 616	Série de Taylor em $x$	0	2	1,23	1,04	1,38	1,24
			4	1,56	1,14	1,79	1,34
			8	-	-	2,05	1,34
	4	2	1,20	0,98	1,32	1,12	
		4	1,52	1,06	1,68	1,19	
		8	-	-	1,92	1,19	

nos processamentos paralelos.

#### Comentários

A figura 6.14 ilustra com clareza os principais méritos e problemas da paralelização no tempo tal como foi proposta. Eles são:

- O algoritmo SIMPT, em sua versão seqüencial, apresenta um significativo “slow-down” com o crescimento do número de passos de integração tomados simultaneamente. Há evidências que esse “slowdown” está diretamente associado ao método de pré-condicionamento utilizado e que maiores investigações neste tema poderão reduzi-lo consideravelmente;
- Os “speed ups” são elevados na comparação direta com a versão seqüencial para o mesmo número de passos tomados simultaneamente (“speed up”  $S_{s1}$ ), mostrando a efetividade da paralelização proposta;
- A paralelização no tempo tal como foi proposta apresenta ganhos significativos, mesmo com o “slowdown” seqüencial do algoritmo (“speed up”  $S_{s2}$ );
- O método Bi-CGSTAB mostrou-se superior ao BCG na resolução do sistema linear Jacobiano;

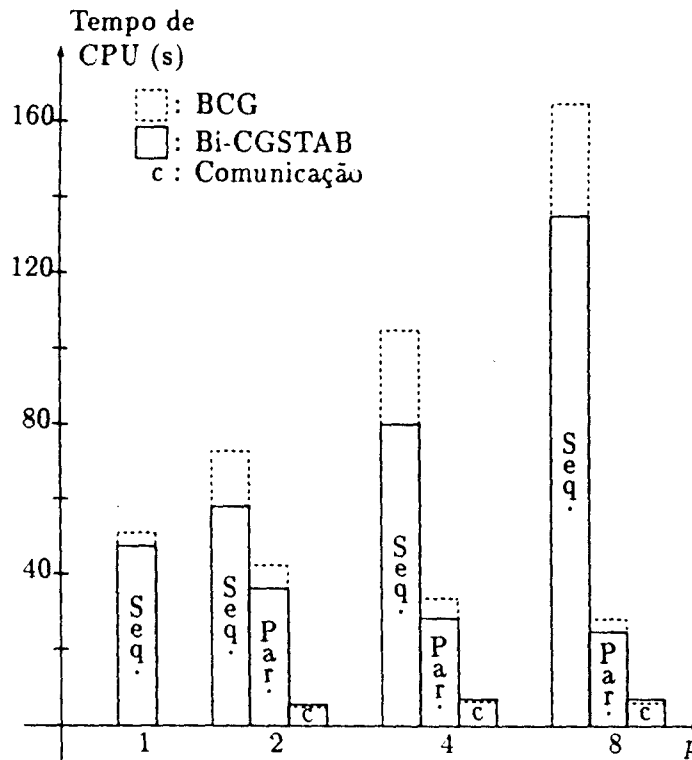


Figura 6.14: Tempos de CPU e Comunicação para o algoritmo SIMPT sequencial e paralelo

- Os tempos de comunicação permaneceram praticamente constantes com o crescimento do número de processadores.

## 6.10 Conclusões

Neste capítulo foram apresentados e analisados os resultados de aplicações dos três algoritmos paralelos desenvolvidos na presente tese, em simulações da dinâmica dos sistemas de energia elétrica IEEE 118 barras e SSUD 616 barras. Os dois primeiros algoritmos ( AEIGC e AEIGCRD ) exploram o paralelismo no espaço e o terceiro ( SIMPT ) explora o paralelismo no tempo. As simulações referem-se ao comportamento dinâmico durante 5 segundos após a ocorrência de perturbações do tipo curto-circuito trifásico sólido, mantido por 100 milisegundos, e eliminado pelo desligamento de uma linha de transmissão ou transformador. Foram feitas simulações paralelas reais e seqüenciais, utilizando-se os computadores NCP I, com 8 processadores e 12 Megaflops de capacidade de pico, e iPSC/860, com 8 processadores e 640 Megaflops de capacidade de pico. As simulações paralelas reais foram feitas em con-



figurações de 2, 4 e 8 processadores. As simulações seqüenciais serviram para estimar o desempenho dos algoritmos paralelos em configurações com 16 e 32 processadores. Adicionalmente, serviram para propósitos de comparação de desempenho dos algoritmos paralelos em relação a suas versões seqüenciais e em relação a versão seqüencial do algoritmo Alternado Entrelaçado Implícito Convencional, que emprega o método direto baseado na fatoração LU e substituições direta e inversa para a resolução das equações da rede elétrica.

As principais conclusões são :

- Os índices de “speed ups” e eficiências obtidos com o algoritmo AEIGC (paralelismo no espaço), em processamentos no NCP I com até 8 processadores, estão muito próximos daqueles encontrados na revisão de literatura. Para o iPSC/860 os tempos de CPU são cerca de uma ordem de grandeza inferiores, contudo, os índices de “speed ups” e eficiências são também inferiores. Isto é devido a razão capacidade de processamento por nó pela capacidade de comunicação entre os nós, ser muito superior no iPSC/860. Outras máquinas, tais como, o iPSC/2 (anterior ao iPSC/860), Intel Delta e Intel Paragon, têm esta razão em valor significativamente inferior a do iPSC/860, conforme mostra a tabela 6.14. Conseqüentemente os valores de “speed ups” e eficiências deverão ser melhores nestas máquinas;
- Nas simulações da dinâmica do sistema SSUD com 662 barras, comparativamente as do sistema IEEE 118 barras, os índices de “speed ups” e eficiências foram significativamente superiores em todos os algoritmos desenvolvidos neste trabalho. Isto é devido principalmente ao fato de que uma parcela significativa do volume de comunicação destes algoritmos independe da dimensão do sistema; com a expectativa de índices de desempenho superiores em SEE com modelos dinâmicos mais sofisticados e dimensões maiores;
- No que se refere a paralelização no espaço, o algoritmo AEIGCRD apresentou índices de desempenho levemente superiores aos do algoritmo AEIGC e, também, mostrou-se potencialmente mais promissor para a exploração da vetorização. Além disso, na maioria dos casos de decomposições obtidas por meios automatizados, os índices de desempenho foram significativamente superiores.

Estes instrumentos de decomposição encontram-se ainda em fase de desenvolvimento com a perspectiva da obtenção de melhores resultados;

- A paralelização no tempo tal como foi proposta, apresentou ganhos significativos, mesmo com o “slowdown” seqüencial apresentado pelo algoritmo SIMPT. Há evidências que esse “slowdown” está diretamente associado ao método de pré-condicionamento utilizado e que maiores investigações neste tema poderão reduzi-lo consideravelmente;
- O método Bi-CGSTAB mostrou-se superior ao BCG, para a resolução do sistema linear Jacobiano. Adicionalmente, o algoritmo SIMPT paralelo mostrou elevado potencial para a vetorização e apresenta tempos de comunicação praticamente constantes com o crescimento do número de processadores.

Tabela 6.14: Parâmetros de alguns computadores paralelos (valores de pico)

Computador	C.P./nó (Mflops)	C.C. (Mb/s)	C.P./C.C.
NCP I	1.5	2.5	0.60
iPSC/2	1.5	2.8	0.53
iPSC/860	80	2.8	28.5
Intel Delta	75	65	1.15
Paragon	75	200	0.375
CM-5	128	20	6.40
SP1/IBM	125	40	3.125
PP/Telebrás	0.5	10	0.05

C.P. : Capacidade de Processamento  
C.C. : Capacidade de Comunicação

# Capítulo 7

## Conclusões

O objetivo desta tese foi investigar a utilização de computadores paralelos na simulação da dinâmica de sistemas de energia elétrica. As principais contribuições são o desenvolvimento, implementação e avaliação de desempenho, em ambiente computacional paralelo, de três algoritmos que têm como novidade mais importante, a utilização de métodos iterativos do tipo gradiente conjugado pré-condicionados.

### 7.1 Considerações Gerais

O interesse dos profissionais de sistemas de energia elétrica pela computação paralela advem da dimensão dos problemas envolvidos, associados a necessidade de alta velocidade de cálculo, principalmente nos centros de supervisão e controle em tempo real, e das condições que esta tecnologia oferece para se dispor de estações de trabalho de grande capacidade computacional a custo relativamente baixo. Contudo, trata-se de uma tecnologia recente e que apresenta, na diversidade de arquiteturas computacionais, uma variável adicional aos problemas de sistemas de energia elétrica. São necessárias investigações a nível de “hardware”, “software” e no desenvolvimento de algoritmos, a fim de transformar a elevada capacidade computacional e as alternativas de arquiteturas oferecidas por estas máquinas, em ganhos de tempo e qualidade na realização dos estudos.

Na literatura são encontradas várias metodologias desenvolvidas visando uma efetiva utilização de computadores paralelos na simulação da dinâmica de sistemas de energia elétrica. Elas contemplam a exploração do paralelismo no espaço, no tempo e simultaneamente no espaço e tempo, além do emprego do conceito de relaxação da forma de onda. Algumas foram implementadas e testadas em computadores paralelos

reais, outras foram apenas validadas e avaliadas em simulações seqüenciais de computadores paralelos hipotéticos. No capítulo 3 deste trabalho, foi feita uma análise destas metodologias com as seguintes conclusões principais:

- As paralelizações no espaço e simultaneamente no espaço e tempo, baseadas no método “Very Dishonest Newton” (VDHN), apresentam forte saturação da curva de “speed up” com o crescimento do número de processadores;
- A paralelização no tempo apresentou ganhos de velocidade para no máximo oito passos de integração tomados simultaneamente;
- Os métodos baseados na relaxação da forma de onda, não implementados em computadores paralelos reais, têm a sua efetividade simultaneamente dependente de um acoplamento relativamente fraco entre os subsistemas (essa característica poderá variar ao longo da simulação), da utilização de técnicas de “multirate integration” e da definição de tamanhos adequados da janela de integração. Além disso, em implementações paralelas, as decomposições deverão atender aos requisitos conflitantes das condições de convergência rápida e do balanceamento da carga entre os processadores;
- Os trabalhos de investigação devem ser dirigidos no sentido do desenvolvimento de metodologias que explorem a sinergia que pode ocorrer com a utilização de computadores paralelos-vetoriais.

Nos algoritmos desenvolvidos neste trabalho, os dois primeiros, denominados respectivamente AEIGC e AEIGCRD, exploraram o paralelismo no espaço, baseado no esquema *Alternado Entrelaçado Implícito*. O terceiro, denominado SIMPT, explorou o paralelismo no tempo e está baseado no esquema *Simultâneo Implícito*, com a solução das equações algébricas e algebrizadas pelo método de Newton.

No algoritmo AEIGC obteve-se um elevado grau de paralelismo com a aplicação do método Gradiente Conjugado (GC) pré-condicionado, na resolução de um subsistema linearizado associado as equações que descrevem a rede elétrica, tomando-se a matriz admitância nodal na forma BBDF.

No algoritmo AEIGCRD foi utilizado o método GC pré-condicionado na resolução de todas as equações associadas a rede elétrica. O desenvolvimento de técnicas de

pré-condicionamento para a matriz admitância nodal, também são contribuições do trabalho.

No algoritmo SIMPT, diferentemente dos trabalhos da literatura, a equação Jacobiana de muito grande porte é resolvida por um método tipo GC. A matriz Jacobiana não é simétrica e, então, foram utilizados os métodos Gradiente Biconjugado pré-condicionado e Bi-CGSTAB pré-condicionado na resolução do sistema linear Jacobiano. O desenvolvimento de técnicas de pré-condicionamento para o sistema linear Jacobiano, novamente representam contribuições do trabalho.

Os algoritmos AEIGC, AEIGCRD e SIMPT foram implementados em uma estrutura de sistema computacional construída ao longo de seus desenvolvimentos. Neste sistema encontram-se as versões seqüenciais e paralelas desses algoritmos com diversas opções de modelos dos elementos de SEE, diversos métodos de solução de sistemas lineares simétricos e assimétricos e diversos métodos de pré-condicionamento.

Foram realizados testes para avaliação de desempenho, em simulações da dinâmica de dois sistemas elétricos frente a ocorrência de grandes perturbações. Os sistemas são o IEEE 118 barras e uma configuração do sistema sul-sudeste do Brasil com 616 barras. Utilizou-se os computadores paralelos NCP I e iPSC/860 da COPPE/UFRJ.

As principais conclusões são:

- Obteve-se uma efetiva redução nos tempos de processamento da simulação da dinâmica de sistemas de energia elétrica, usando computadores paralelos e os algoritmos desenvolvidos;
- A utilização de métodos iterativos do tipo Gradiente Conjugado, no âmbito dos algoritmos desenvolvidos, mostrou-se efetiva e com potencialidades de maiores ganhos, principalmente, se usada em computadores paralelos-vetoriais e em simulações de sistemas representados por modelos dinâmicos mais sofisticados e com dimensões superiores a dos utilizados neste trabalho. Isto é devido ao fato de que uma parcela significativa dos custos de comunicação dos algoritmos desenvolvidos, é praticamente independente da dimensão e dos modelos do sistema de energia elétrica a ser simulado;
- Na exploração do paralelismo no espaço, o algoritmo AEIGCRD apresentou índices de desempenho levemente superiores aos do algoritmo AEIGC e maiores potencialidades para a vetorização. Os valores de “speed ups” e eficiências

foram significativamente superiores para os processamentos realizados no NCP I, embora os tempos de CPU tenham sido cerca de uma ordem de grandeza inferiores no iPSC/860. Isto é devido a razão capacidade de processamento pela capacidade de comunicação, muito superior no iPSC/860. Os valores nominais são 0,6 Mflops/Mbytes/s para o NCP I e 28 Mflops/Mbytes/s para o iPSC/860. No Intel Paragon, por exemplo, a máquina mais moderna da Intel, o valor nominal desta razão é igual a 0,375 Mflops/Mbytes/s, gerando a expectativa de “speed ups” superiores aos obtidos no NCP I;

- Na exploração do paralelismo no tempo, o algoritmo desenvolvido (SIMPT) apresentou elevado ganho na paralelização, porém, os “speed ups” relativos a solução seqüencial tomando-se um único passo de integração de cada vez, são de valores apenas moderados, devido ao “slowdown” seqüencial apresentado pelo algoritmo. Esta característica poderá ser revertida com maiores investigações nas áreas de pré-condicionamento e inicialização das janelas de integração. Por outro lado, o algoritmo apresentou elevado potencial para a vetorização. Adicionalmente, o método Bi-CGSTAB foi significativamente superior em relação ao método Gradiente Biconjugado, na resolução do sistema linear Jacobiano.

Finalmente, destaca-se que o presente trabalho representa um primeiro passo na investigação da aplicabilidade de métodos iterativos do tipo Gradiente Conjugado, no âmbito do problema da simulação da dinâmica de sistemas de energia elétrica. Foram feitas a identificação de potencialidades e de pontos de estrangulamento. Além disso, as implementações realizadas estão restritas a arquiteturas paralelas do tipo memória distribuída. Maiores investigações deverão concentrar-se em aspectos relacionados com o pré-condicionamento e decomposição, para os algoritmos que exploram o paralelismo no espaço, e com o pré-condicionamento, inicialização e aproximações na matriz Jacobiana, para os algoritmos que exploram o paralelismo no tempo. Implementações e avaliações de desempenho em outras máquinas paralelas também serão importantes.

## 7.2 Sugestões de Estudos Futuros

Neste item apresentam-se algumas sugestões de estudos futuros no sentido da continuação das investigações iniciadas no presente trabalho de tese. As principais su-

gestões são:

- Realizar a implementação e avaliações de desempenho dos algoritmos AEIGC, AEIGCRD e SIMPT, em computadores paralelos-vetoriais e/ou com memória compartilhada, tais como por exemplo: Alliant FX/8, Cray Y/MP, etc.;
- Desenvolver métodos de pré-condicionamento mais efetivos que os utilizados neste trabalho, tanto do ponto de vista da aceleração da convergência quanto do ponto de vista da vetorização;
- Investigar técnicas de relaxação do método de Newton aplicado ao esquema simultâneo implícito paralelo no tempo, através de aproximações na matriz Jacobiano;
- Investigar a exploração simultânea do paralelismo no espaço e tempo para o esquema *Simultâneo Implícito*;
- Implementar, no sistema computacional desenvolvido neste trabalho (PPSDS), modelos mais sofisticados para as máquinas síncronas e seus controladores. Avaliar o desempenho dos três algoritmos paralelos desenvolvidos (AEIGC, AEIGCRD e SIMPT), usando-se esses novos modelos para simulações em sistemas tendo dimensões maiores que a dos utilizados no presente trabalho;
- Aprofundar as investigações em métodos de inicialização para as variáveis das janelas de integração, na exploração do paralelismo no tempo.

# Apêndice A

## Modelagem dos Elementos do Sistema

Neste apêndice são apresentados os modelos matemáticos dos elementos dos SEE utilizados no presente trabalho. O modelo dinâmico global foi limitado à representação das máquinas síncronas por modelos de até quarta ordem e seus respectivos sistemas de excitação. Empregam-se os modelos de sistemas de excitação IEEE tipo 1 [62], com excitatriz rotativa, e IEEE tipo AC4 [63], com excitatriz estática. A opção por dois modelos visa permitir a avaliação da influência de excitatrizes com tempos de resposta distintos, no comportamento dos algoritmos desenvolvidos.

A rede elétrica é representada pela formulação nodal compacta (complexa), tradicionalmente empregada nos esquemas alternados, ou pela formulação nodal expandida, nas partes real e imaginária para cada um de seus elementos, exigida pelos esquemas simultâneos baseados no método de Newton. As cargas são representadas por modelos polinomiais em função da tensão.

### A.1 Máquinas Síncronas

Os modelos de máquinas síncronas, normalmente usados em estudos de estabilidade transitória, estão baseados nas transformações de Park e no conceito de reatâncias operacionais. Em geral, as seguintes hipóteses simplificadoras são adotadas [1]:

- Representa-se somente a componente de frequência fundamental das correntes e tensões nos estatores das máquinas síncronas e sistema de transmissão. Isto permite o uso de fasores e implica em desprezar-se as componentes de corrente contínua dos estatores;



- Usam-se componentes simétricas para representar as condições desequilibradas;
- Desprezam-se os efeitos das variações de velocidade das máquinas sobre as tensões de seus estatores, sob a hipótese de que o desvio de velocidade em relação a velocidade síncrona é, normalmente, pequeno.

O comportamento dos rotores é descrito por equações diferenciais eletromecânicas e equações diferenciais elétricas. Os estatores são representados por equações algébricas.

### A.1.1 Equações Diferenciais Eletromecânicas

As equações diferenciais eletromecânicas que descrevem o movimento do rotor de uma máquina síncrona, relacionando seus torques inercial e de amortecimento com a resultante entre o torque mecânico, fornecido pela máquina primária, e o torque elétrico, proporcional a potência elétrica fornecida a rede, já em grandezas por unidade (pu.), são [30]:

$$\begin{cases} \frac{d\omega}{dt} = \frac{\omega_s}{2H} [P_m - P_e - D(\omega - \omega_s)] \\ \frac{d\delta}{dt} = \omega - \omega_s \end{cases} \quad (\text{A.1})$$

onde:

$t$  - tempo (em segundos);

$H$  - constante de inércia (em segundos);

$\omega$  - velocidade angular (em radianos/segundos);

$\omega_s$  - velocidade angular síncrona (em radianos/segundos);

$P_e$  - potência elétrica de entreferro (em p.u.);

$P_m$  - potência mecânica de entrada (em p.u.);

$\delta$  - ângulo entre o eixo do rotor e uma referência que gira a velocidade síncrona (em radianos);

$D$  - coeficiente de amortecimento devido ao atrito mecânico, variações das cargas com a frequência, etc. (em segundos/ radianos elétricos).

Neste trabalho despreza-se a ação do regulador de velocidade e, portanto, a potência mecânica  $P_m$  permanecerá constante ao longo do período de simulação.

### A.1.2 Equações Diferenciais Elétricas

As equações diferenciais elétricas podem representar um variado número de circuitos equivalentes de eixo direto ( $d$ ) e quadratura ( $q$ ) para o rotor. Entretanto, o modelo adotado neste trabalho considera apenas os efeitos transitórios (os efeitos subtransitórios são desprezados), com a representação do enrolamento de campo principal e, no caso de máquinas de rotor cilíndrico, mais um enrolamento amortecedor, localizado no eixo  $q$ . Com estas hipóteses e convencionando-se o eixo  $d$  adiantado em relação ao eixo  $q$ , as equações diferenciais elétricas do rotor são dadas por [30]:

$$\begin{cases} \frac{dE'_q}{dt} = \frac{-1}{T'_{d0}} [E'_q - (x_d - x'_d)I_d - E_{fd}] \\ \frac{dE'_d}{dt} = \frac{-1}{T'_{q0}} [E'_d + (x_q - x'_q)I_q] \end{cases} \quad (\text{A.2})$$

onde:

$E'_d$  - componente de eixo direto da tensão transitória interna da máquina (em pu.);

$E'_q$  - componente de eixo quadratura da tensão transitória interna da máquina (em pu.);

$T'_{d0}$  - constante de tempo transitória de eixo direto com estator em aberto (em segundos);

$T'_{q0}$  - constante de tempo transitória de eixo quadratura com estator em aberto (em segundos);

$x_d, x'_d$  - reatâncias em regime permanente e transitória de eixo direto (em pu.);

$x_q, x'_q$  - reatâncias em regime permanente e transitória de eixo quadratura (em pu.);

$E_{fd}$  - tensão de excitação refletida no estator (em pu.);

$I_d, I_q$  - Componentes de eixo direto e quadratura da corrente de estator (em pu.).

No caso de rotor de pólos salientes, representa-se apenas o enrolamento de campo principal, cujo comportamento é descrito pela seguinte equação diferencial:

$$\frac{dE'_q}{dt} = \frac{-1}{T'_{d0}} [E'_q - (x_d - x'_d)I_d - E_{fd}] \quad (\text{A.3})$$

### A.1.3 Equações Algébricas do Estator

Com a hipótese de considerar apenas os efeitos transitórios do rotor e desprezando-se a resistência dos enrolamentos de estator, por ter valor pouco significativo em máquinas de grande porte, as equações que representam o estator, em pu., no sistema de referência  $d, q$  (modelo de Park), são dadas por [30]:

$$\begin{cases} E'_d - V_d = x'_q I_q \\ E'_q - V_q = -x'_d I_d \\ P_e = V_d I_d + V_q I_q = E'_d I_d + E'_q I_q - (x'_q - x'_d) I_d I_q \end{cases} \quad (\text{A.4})$$

onde:

$V_d$  - tensão terminal de eixo direto (em pu.);

$V_q$  - tensão terminal de eixo quadratura (em pu.).

Note que a tensão  $E'_d$  anula-se para o caso de máquinas de pólos salientes.

## A.2 Sistemas de Excitação

Atualmente existem em uso uma grande variedade de sistemas de excitação. Nos mais antigos, a excitatriz é um gerador de corrente contínua acionado pela turbina (mesmo eixo do gerador) ou por um motor de indução. Nos sistemas modernos as excitatrizes são estáticas e constituem-se de elementos retificadores controlados.

O tempo de resposta das excitatrizes estáticas são, em geral, muito menores que o das excitatrizes rotativas, permitindo uma ação mais efetiva de controle nos períodos transitórios, pela rápida variação da corrente de campo do rotor.

Neste documento, empregam-se os modelos IEEE tipo 1 [62] com excitatriz rotativa, e o modelo IEEE AC4 [63] com excitatriz estática.

### A.2.1 Sistema de Excitação IEEE Tipo 1

Este sistema de excitação é representado pelo diagrama de blocos da figura A.1 e descrito pelas seguintes equações diferenciais [30]:

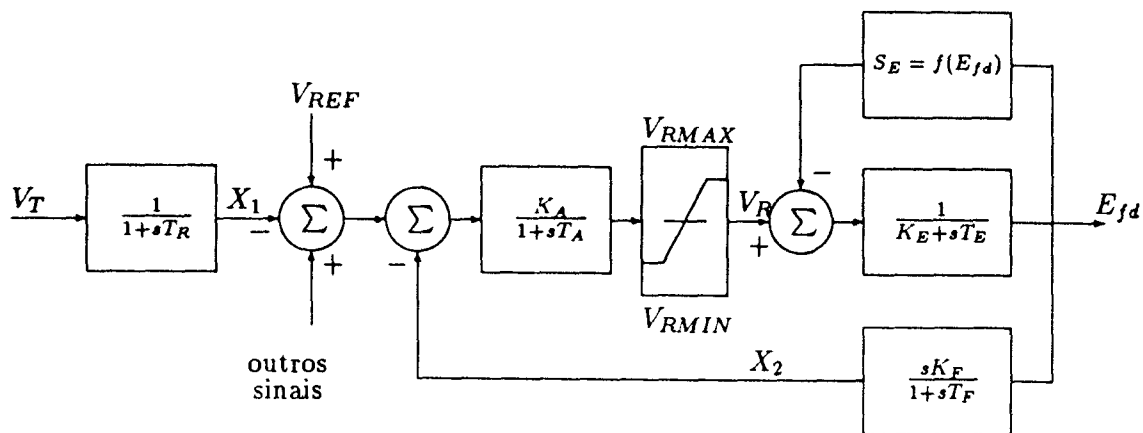


Figura A.1: Sistema de excitação IEEE tipo 1 (ação contínua)

$$\left\{ \begin{array}{l} \frac{dX_1}{dt} = \frac{1}{T_R} [V_T - X_1] \\ \frac{dV_R}{dt} = \frac{K_A}{T_A} [V_{REF} - X_1 - X_2] - \frac{1}{T_A} V_R \\ \frac{dE_{fd}}{dt} = \frac{1}{T_E} [V_R - (K_E + S_E) E_{fd}] \\ \frac{dX_2}{dt} = \frac{K_F}{T_F} \frac{dE_{fd}}{dt} - \frac{K_F X_2}{T_F} \end{array} \right. \quad (A.5)$$

onde:

$V_T$  - tensão terminal do gerador;

$V_{REF}$  - tensão de referência do regulador (definida para satisfazer as condições iniciais);

$V_R$  - tensão na saída do regulador;

$V_{RMIN}$  - valor mínimo de  $V_R$ ;

$V_{RMAX}$  - valor máximo de  $V_R$ ;

$E_{fd}$  - tensão na saída da excitatriz (tensão de campo do gerador);

$K_A$  - ganho do regulador;

$K_E$  - constante da excitatriz relacionada ao campo de auto-excitação;

$K_F$  - ganho do circuito de estabilização do regulador;

$S_E$  - função saturação da excitatriz;

$T_A$  - constante de tempo do amplificador;

$T_E$  - constante de tempo da excitatriz;

$T_F$  - constante de tempo do circuito de estabilização do regulador;

$T_R$  - constante de tempo do filtro de entrada do regulador.

### A.2.2 Sistema de Excitação Tipo IEEE AC4

Este sistema de excitação utiliza uma ponte completa de tiristores na saída da excitatriz, oferecendo resposta inicial muito rápida [63]. O regulador de tensão controla os disparos da ponte de tiristores. O alternador excitatriz usa um regulador de tensão independente para manter constante a sua tensão terminal. Estes efeitos não são modelados, contudo, os efeitos do carregamento transitório do alternador excitatriz são incluídos através de modificações nos limites de excitação. O diagrama de blocos é apresentado na figura A.2 e as equações diferenciais a seguir.

$$\begin{cases} \frac{dV_e}{dt} + \frac{T_C}{T_B} \frac{dV_T}{dt} = \frac{1}{T_B} [V_{REF} - V_T - V_e] \\ \frac{dE_{fd}}{dt} = \frac{1}{T_A} [K_A V_e - E_{fd}] \end{cases} \quad (A.6)$$

onde:

$V_T$  - tensão terminal do gerador;

$V_{REF}$  - tensão de referência do regulador (definida para satisfazer as condições iniciais);

$K_C$  - fator de carga do retificador relacionado a reatância de comutação;

$I_{fd}$  - corrente de campo do gerador;

$E_{fd}$  - tensão na saída da excitatriz (tensão de campo do gerador);

$V_e$  - tensão na saída do regulador;

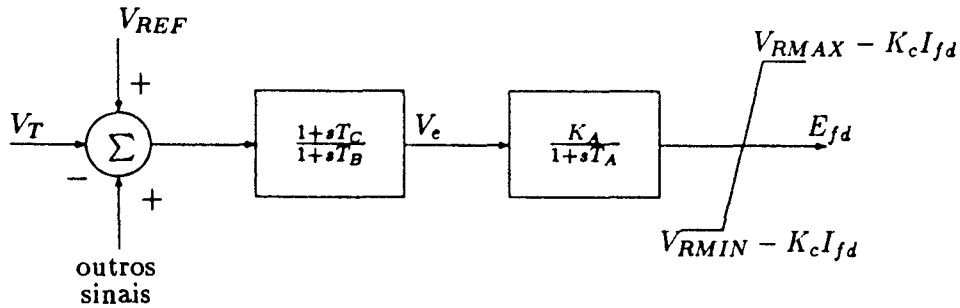


Figura A.2: Sistema de excitação IEEE tipo AC4 (*Alternator supplied controlled rectifier excitation system*)

$V_{RMIN}$  - valor mínimo de  $V_R$ ;

$V_{RMAX}$  - valor máximo de  $V_R$ ;

$T_A, T_B, T_C$  - constantes de tempo do regulador de tensão.

### A.2.3 Sistema de Transmissão

O sistema de transmissão é descrito usando-se a matriz admitância nodal na formulação de injeções de correntes, conforme a equação abaixo [1]:

$$I(E, V) = YV \tag{A.7}$$

onde:

$I$  - vetor de injeções de correntes nodais;

$Y$  - matriz admitância nodal;

$V$  - vetor de tensões nodais.

Este sistema de equações, normalmente não-linear, poderá aparecer na forma compacta (complexa) ou na forma expandida, nas parcelas real e imaginária de cada

elemento. Na forma compacta ele é resolvido iterativamente para  $V$ , através da fatoração LU de  $Y$ .

A forma expandida é necessária quando realiza-se a solução de (A.7) através do método de Newton, e é dada por:

$$I^e(E, V^e) = Y^e V^e \quad (\text{A.8})$$

onde  $I^e$  é o vetor de injeções de correntes nodais e  $V^e$  é o vetor de tensões nodais, ambos expandidos nas parcelas real e imaginária de seus elementos.

Neste caso pode-se escrever:

$$F^e = I^e - Y^e V^e = 0 \quad (\text{A.9})$$

sendo que  $F^e$  é zero na solução.

Nos esquemas simultâneos de solução do problema da SDSEE, emprega-se a forma expandida da matriz admitância nodal,  $Y^e$ , devido a utilização do método de Newton para resolver o sistema constituído de equações algébricas e algébricas a diferenças; essas últimas resultantes da aplicação de um método de integração implícito às equações diferenciais. No caso dos esquemas alternados, a fatoração LU de  $Y$ , na forma compacta, tem sido mais utilizada. Isto se deve ao fato de que  $Y$ , nesta forma, precisa ser refatorada somente quando ocorrem alterações na rede de transmissão, enquanto que a matriz jacobiana deve, a rigor, ser recalculada e fatorada a cada iteração.

### A.3 Acoplamento da Máquina Síncrona à Rede de Transmissão

As grandezas  $E'_d, E'_q, E'_{fd}, V_d, V_q, I_d$  e  $I_q$ , utilizadas na modelagem da máquina síncrona, são dadas em termos de valores eficazes equivalentes de estator e em relação a referência  $d, q$ . Por outro lado, as grandezas do sistema de transmissão estão relacionadas a um sistema de referência complexo (eixos real e imaginário). Ambos os sistemas de referência giram a velocidade síncrona, porém, estão defasados de um ângulo  $\delta$ . Então, para realizar o acoplamento das equações da máquina síncrona com as equações da rede de transmissão, é necessário definir-se uma transformação de coordenadas por rotação de eixos, conforme ilustra a figura A.3 para o caso de tensões.

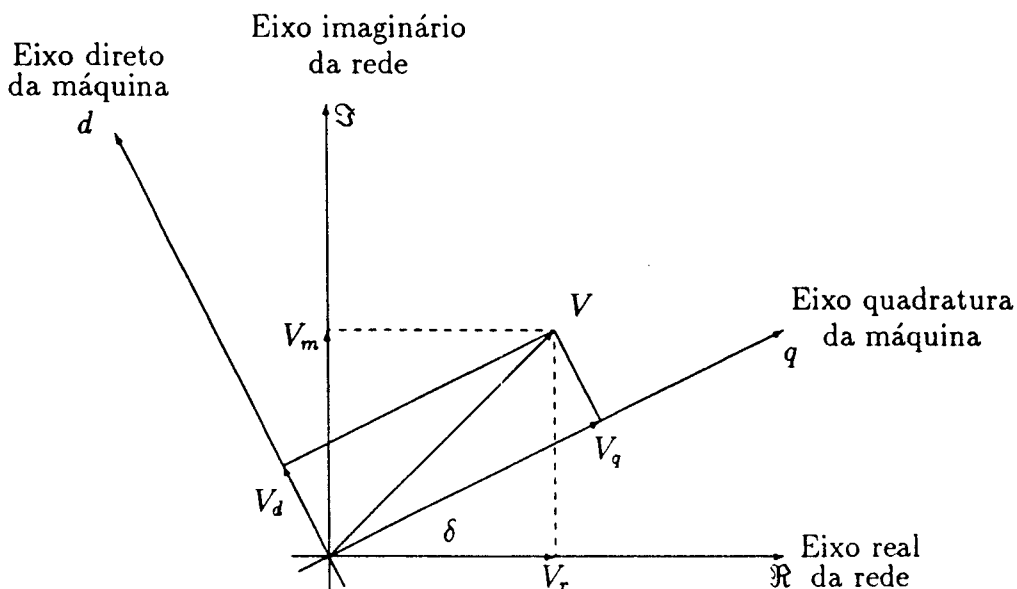


Figura A.3: Diagrama composto de transformação de coordenadas  $d - q \rightarrow \Re - \Im$  e vice-versa

Da figura A.3, obtém-se:

$$\begin{cases} V_d = -V_r \operatorname{sen} \delta + V_m \cos \delta \\ V_q = V_r \cos \delta + V_m \operatorname{sen} \delta \end{cases} \quad (\text{A.10})$$

As equações (A.10) podem ser reescritas na seguinte forma matricial:

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \begin{bmatrix} -\operatorname{sen} \delta & \cos \delta \\ \cos \delta & \operatorname{sen} \delta \end{bmatrix} \begin{bmatrix} V_r \\ V_m \end{bmatrix} \quad (\text{A.11})$$

Note que essa transformação é igualmente válida para correntes e sua inversa é dada por:

$$\begin{bmatrix} V_r \\ V_m \end{bmatrix} = \begin{bmatrix} -\operatorname{sen} \delta & \cos \delta \\ \cos \delta & \operatorname{sen} \delta \end{bmatrix} \begin{bmatrix} V_d \\ V_q \end{bmatrix} \quad (\text{A.12})$$



### A.3.1 Cargas

Neste trabalho, as cargas são representadas pelo modelo polinomial constituído de parcelas que refletem o comportamento de impedância constante, corrente constante e potência constante [29]. Estas parcelas estão baseados nas hipóteses seguintes:

- *Potência constante* - a potência da carga é constante independentemente da tensão nodal;
- *Corrente constante* - o módulo da corrente solicitada pela carga não varia durante o transitório, podendo ser calculado a partir da potência complexa  $S$ , que varia com o módulo da tensão ( $S = S_0V/V_0$ ), e da tensão complexa;
- *Impedância constante* - a carga é representada por uma impedância constante para a terra, calculada em função das condições nominais de operação.

Matematicamente, as três parcelas do modelo de carga são dadas por:

- Parcela de potência constante

$$P_s + jQ_s = a_1P_0 + ja_2Q_0 \quad (\text{A.13})$$

- Parcela de corrente constante

$$P_I + jQ_I = \frac{[b_1P_0 + jb_2Q_0] \cdot V}{V_0} \quad (\text{A.14})$$

- Parcela de impedância constante

$$P_z + jQ_z = \left[ \frac{c_1P_0}{|V_0^2|} - j \frac{c_2Q_0}{|V_0^2|} \right] |V|^2 \quad (\text{A.15})$$

onde:

$P_0$  - potência ativa de regime permanente pré-perturbação (em pu.);

$Q_0$  - potência reativa no regime permanente pré-perturbação (em pu.);

$V_0$  - tensão na barra de carga no regime permanente pré-perturbação (em pu.);

$V$  - tensão na barra de carga ao longo da simulação (em pu.);

$a_1, b_1, c_1$  - constantes que especificam a percentagem de potência constante, corrente constante e impedância constante, respectivamente, para a carga ativa;

$a_2, b_2, c_2$  - constantes que especificam a percentagem de potência constante, corrente constante e impedância constante, respectivamente, para a carga reativa;

$P_s, P_I, P_z$  - potências ativas associadas, respectivamente as parcelas de carga do tipo potência constante, corrente constante e impedância constante;

$Q_s, Q_I, Q_z$  - potências reativas associadas, respectivamente as parcelas de carga do tipo potência constante, corrente constante e impedância constante.

# Apêndice B

## Esquemas Tradicionais para a SDSEE

Neste apêndice descreve-se, em detalhes, os esquemas Alternado Entrelaçado Implícito (AEI) e Simultâneo Implícito (SI) de resolução das EDAs que descrevem o comportamento dinâmico de um SEE, especificamente para os modelos apresentados no apêndice A. O texto foi dividido em três partes principais: a) a algebrização das equações diferenciais usando o método de integração trapezoidal implícito; b) o esquema AEI; c) o esquema SI.

### B.1 Algebrização das Equações Diferenciais Usando o Método Trapezoidal Implícito

Suponha a seguinte equação diferencial

$$\dot{x} = f(x) \tag{B.1}$$

A aplicação do método trapezoidal implícito à equação (B.1), resulta:

$$x(t) - x(t - \Delta t) = [f(x(t - \Delta t)) + f(x(t))] \frac{\Delta t}{2} \tag{B.2}$$

ou:

$$x(t) = x(t - \Delta t) + [f(x(t - \Delta t)) + f(x(t))] \frac{\Delta t}{2} \tag{B.3}$$

As equações (B.2) e (B.3) são equações algébricas a diferenças e representam a equação diferencial (B.1) no intervalo de tempo  $t - \Delta t$  a  $t$ .

### B.1.1 Equações das Máquinas Síncronas

Neste item descreve-se a algebrização das equações diferenciais de uma máquina síncrona, segundo a metodologia apresentada nas equações (B.1) a (B.3).

#### Equações Eletromecânicas do Rotor

Desprezando-se os torques de amortecimento nas equações (A.1), têm-se:

$$\begin{cases} \frac{d\omega}{dt} = \frac{\omega_s}{2H} [P_m - P_e] \\ \frac{d\delta}{dt} = \omega - \omega_s \end{cases} \quad (\text{B.4})$$

Aplicando-se o método trapezoidal as equações (B.4), resulta:

$$\omega(t) - \omega(t - \Delta t) = \frac{\omega_s \Delta t}{4H} [2P_m - P_e(t - \Delta t) - P_e(t)] \quad (\text{B.5})$$

$$\delta(t) - \delta(t - \Delta t) = \{[\omega(t - \Delta t) - \omega_s] + [\omega(t) - \omega_s]\} \frac{\Delta t}{2} \quad (\text{B.6})$$

Definindo-se:

$$\begin{cases} \sigma(t - \Delta t) = \omega(t - \Delta t) + \frac{\omega_s \Delta t}{4H} [2P_m - P_e(t - \Delta t)] \\ \beta(t - \Delta t) = \delta(t - \Delta t) + [\omega(t - \Delta t) - 2\omega_s] \frac{\Delta t}{2} \end{cases} \quad (\text{B.7})$$

as equações para  $\omega(t)$  e  $\delta(t)$  assumem a forma seguinte:

$$\begin{cases} \omega(t) = -\frac{\omega_s \Delta t}{4H} P_e(t) + \sigma(t - \Delta t) \\ \delta(t) = \frac{\omega(t) \Delta t}{2} + \beta(t - \Delta t) \end{cases} \quad (\text{B.8})$$

Para os esquemas alternados é conveniente expressar-se  $\delta(t)$  em função de  $P_e(t)$  somente. Assim, substituindo-se (B.5) em (B.6), tem-se:

$$\begin{aligned} \delta(t) - \delta(t - \Delta t) &= [\omega(t - \Delta t) - \omega_s] \frac{\Delta t}{2} + [\omega(t) - \omega_s] \frac{\Delta t}{2} \\ &\quad + \frac{\omega_s \Delta t^2}{8H} [2P_m - P_e(t - \Delta t) - P_e(t)] \end{aligned} \quad (\text{B.9})$$

Definindo-se,

$$\alpha(t - \Delta t) = \delta(t - \Delta t) + [\omega(t - \Delta t) - \omega_s] \Delta t + \frac{\omega_s(\Delta t)^2}{8H} [2P_m - P_c(t - \Delta t)], \quad (\text{B.10})$$

para a equação (B.9) e isolando-se  $\omega(t)$  na equação (B.6), obtém-se:

$$\begin{cases} \delta(t) = \alpha(t - \Delta t) - \frac{\omega_s(\Delta t)^2}{8H} P_c(t) \\ \omega(t) = \frac{2}{\Delta t} [\delta(t) - \delta(t - \Delta t)] - [\omega(t - \Delta t) - 2\omega_s] \end{cases} \quad (\text{B.11})$$

### Equações Elétricas do Rotor (*efeitos transitórios*)

Aplicando-se a regra trapezoidal as equações (A.2), resulta:

$$\begin{cases} E'_q(t) - E'_q(t - \Delta t) = \frac{-\Delta t}{2T'_{d0}} [E'_q(t - \Delta t) - (x_d - x'_d)I_d(t - \Delta t) - E_{fd}(t - \Delta t) + E'_q(t) - (x_d - x'_d)I_d(t) - E_{fd}(t)] \\ E'_d(t) - E'_d(t - \Delta t) = \frac{-\Delta t}{2T'_{q0}} [E'_d(t - \Delta t) + (x_q - x'_q)I_q(t - \Delta t) + E'_d(t) + (x_q - x'_q)I_q(t)] \end{cases} \quad (\text{B.12})$$

Rearranjando-se as equações (B.12), obtém-se:

$$\begin{cases} E'_q(t) = E'_q(t - \Delta t) + \frac{\Delta t}{2T'_{d0} + \Delta t} [E_{fd}(t - \Delta t) - 2E'_q(t - \Delta t) + (x_d - x'_d)I_d(t - \Delta t)] + \frac{\Delta t}{2T'_{d0} + \Delta t} [(x_d - x'_d)I_d(t) + E_{fd}(t)] \\ E'_d(t) = E'_d(t - \Delta t) - \frac{\Delta t}{2T'_{q0} + \Delta t} [2E'_d(t - \Delta t) + (x_q - x'_q)I_q(t - \Delta t)] - \frac{\Delta t}{2T'_{q0} + \Delta t} [(x_q - x'_q)I_q(t)] \end{cases} \quad (\text{B.13})$$

Definindo-se,

$$\begin{cases} a_d = \frac{\Delta t}{2T'_{d0} + \Delta t} \\ F_q = a_d [E_{fd}(t - \Delta t) - 2E'_q(t - \Delta t) + (x_d - x'_d)I_d(t - \Delta t)] + E'_q(t - \Delta t) \\ a_q = \frac{\Delta t}{2T'_{q0} + \Delta t} \\ F_d = E'_d(t - \Delta t) - a_q [2E'_d(t - \Delta t) + (x_q - x'_q)I_q(t - \Delta t)] \end{cases} \quad (\text{B.14})$$

e substituindo-se as equações (B.14) em (B.13), resulta:

$$\begin{cases} E'_q(t) = a_d(x_d - x'_d)I_d(t) + a_d E_{fd}(t) + F_q \\ E'_d(t) = -a_q(x_q - x'_q)I_q(t) + F_d \end{cases} \quad (\text{B.15})$$

Note que para o cálculo de  $E'_q(t)$  e  $E'_d(t)$  necessita-se dos valores de  $I_d(t)$  e  $I_q(t)$ .

### B.1.2 Sistema de Excitação IEEE Tipo 1

Para este sistema de excitação, o efeito da saturação é considerado linearizando-se a realimentação  $S_E E_{fd}$  em torno do ponto de solução no passo anterior ( $t - \Delta t$ ). O resultado desta linearização é uma equação na forma:

$$S_E E_{fd} = K_1 E_{fd} - K_2 \quad (\text{B.16})$$

com  $K_1$  e  $K_2$  conhecidos. Assim, a equação diferencial da excitatriz é dada por:

$$\frac{dE_{fd}}{dt} = \frac{1}{T_E} [V_R - (K_E + K_1)E_{fd} + K_2] \quad (\text{B.17})$$

Aplicando-se o método trapezoidal implícito a equação (B.17), tem-se:

$$E_{fd}(t) - E_{fd}(t - \Delta t) = \frac{1}{T_E} \{V_R(t) + V_R(t - \Delta t) - [K_E + K_1][E_{fd}(t) + E_{fd}(t - \Delta t)] + 2K_2\} \frac{\Delta t}{2} \quad (\text{B.18})$$

Realizando-se algumas manipulações matemáticas, pode-se escrever a equação (B.18) na forma seguinte:

$$\left[ \frac{2T_E}{\Delta t} + K_E + K_1 \right] E_{fd}(t) = \left[ \frac{2T_E}{\Delta t} - K_E - K_1 \right] E_{fd}(t - \Delta t) + V_R(t) + V_R(t - \Delta t) + 2K_2 \quad (\text{B.19})$$

Definindo-se, em (B.19),

$$\begin{cases} a_E = \frac{2T_E}{\Delta t} + K_E + K_1 \\ b_E = \left[ \frac{2T_E}{\Delta t} - K_E - K_1 \right] E_{fd}(t - \Delta t) + V_R(t - \Delta t) + 2K_2 \end{cases} \quad (\text{B.20})$$

obtém-se:

$$E_{fd}(t) = [b_E + V_R(t)] / a_E \quad (\text{B.21})$$

As equações diferenciais para o filtro, malha de estabilização e regulador são dadas por:

$$\begin{cases} \frac{dX_1}{dt} = \frac{1}{T_R} [V_T - X_1] \\ \frac{dV_R}{dt} = \frac{K_A}{T_A} [V_{REF} - X_1 - X_2] - \frac{1}{T_A} V_R \\ \frac{dX_2}{dt} = \frac{K_F}{T_F} \frac{dE_{fd}}{dt} - \frac{X_2}{T_F} \end{cases} \quad (\text{B.22})$$

Aplicando-se o método trapezoidal implícito as equações (B.22), obtém-se:

$$\begin{cases} X_1(t) - X_1(t - \Delta t) = \frac{\Delta t}{2T_R} [V_T(t) - X_1(t) + V_T(t - \Delta t) - X_1(t - \Delta t)] \\ V_R(t) - V_R(t - \Delta t) = \frac{K_A}{T_A} [V_{REF} - X_1(t) - X_2(t) + V_{REF} - X_1(t - \Delta t) - X_2(t - \Delta t)] \frac{\Delta t}{2} - \frac{1}{T_A} [V_R(t) + V_R(t - \Delta t)] \frac{\Delta t}{2} \\ X_2(t) - X_2(t - \Delta t) = \frac{K_F}{T_F} [E_{fd}(t) - E_{fd}(t - \Delta t)] - \frac{1}{T_F} [X_2(t) + X_2(t - \Delta t)] \frac{\Delta t}{2} \end{cases} \quad (\text{B.23})$$

Definindo-se,

$$\begin{cases} a_R = \frac{2T_R}{\Delta t} + 1 \\ a_A = \frac{2T_A}{\Delta t} + 1 \end{cases} \quad (\text{B.24})$$

e substituindo-se (B.24) em (B.23) e rearranjando, obtém-se:

$$\begin{cases} X_1(t) = \left[ \frac{2T_R - \Delta t}{2T_R + \Delta t} \right] [X_1(t - \Delta t)] + \frac{1}{a_R} [V_T(t) + V_T(t - \Delta t)] \\ V_R(t) = \frac{K_A}{a_A} [2V_{REF} - X_1(t) - X_1(t - \Delta t) - X_2(t - \Delta t)] + \left[ \frac{2T_A - \Delta t}{2T_A + \Delta t} \right] V_R(t - \Delta t) \\ X_2(t) = \left[ \frac{2K_F}{2T_F + \Delta t} \right] [E_{fd}(t) - E_{fd}(t - \Delta t)] + \left[ \frac{2T_F - \Delta t}{2T_F + \Delta t} \right] X_2(t - \Delta t) \end{cases} \quad (\text{B.25})$$

Substituindo-se, no sistema de equações (B.25), a primeira e a última na segunda equação, resulta:

$$V_R(t) = \frac{K_A}{a_A} \left[ 2V_{REF} - \frac{1}{a_R} V_T(t - \Delta t) - \frac{4T_R}{2T_R + \Delta t} X_1(t - \Delta t) + \right.$$

$$\begin{aligned} & \left. \frac{2K_F}{2T_F + \Delta t} E_{fd}(t - \Delta t) - \frac{4T_F}{2T_F + \Delta t} X_2(t - \Delta t) \right] - \frac{K_A}{a_A a_R} V_T(t) - \\ & \frac{2K_A K_F}{a_A (2T_F + \Delta t)} E_{fd}(t) + \left[ \frac{2T_A - \Delta t}{2T_A + \Delta t} \right] V_R(t - \Delta t) \end{aligned} \quad (\text{B.26})$$

Substituindo-se a equação (B.26) na equação (B.21), resulta:

$$\begin{aligned} E_{fd} = & \frac{a_A}{a_A a_E + \frac{2K_A K_F}{2T_F + \Delta t}} \left\{ b_E + \frac{K_A}{a_A} \left[ 2V_{REF} - \frac{1}{a_R} V_T(t - \Delta t) - \right. \right. \\ & \frac{4T_R}{2T_R + \Delta t} X_1(t - \Delta t) + \frac{2K_F}{2T_F + \Delta t} E_{fd}(t - \Delta t) - \\ & \left. \left. \frac{4T_F}{2T_F + \Delta t} X_2(t - \Delta t) \right] + \left[ \frac{2T_A - \Delta t}{2T_A + \Delta t} \right] V_R(t - \Delta t) \right\} \\ & - \left[ \frac{K_A/a_R}{a_A a_E + K_A K_F / (T_F + \Delta t / 2)} \right] V_T(t) \end{aligned} \quad (\text{B.27})$$

Definindo-se na equação (B.27),

$$\begin{aligned} E_0 = & \frac{a_A}{a_A a_E + \frac{2K_A K_F}{2T_F + \Delta t}} \left\{ b_E + \frac{K_A}{a_A} \left[ 2V_{REF} - \frac{1}{a_R} V_T(t - \Delta t) - \frac{4T_R}{2T_R + \Delta t} X_1(t - \Delta t) + \right. \right. \\ & \left. \left. \frac{2K_F}{2T_F + \Delta t} E_{fd}(t - \Delta t) - \frac{4T_F}{2T_F + \Delta t} X_2(t - \Delta t) \right] + \right. \\ & \left. \left[ \frac{2T_A - \Delta t}{2T_A + \Delta t} \right] V_R(t - \Delta t) \right\} \end{aligned} \quad (\text{B.28})$$

e

$$b = \frac{K_A/a_R}{a_A a_E + K_A K_F / (T_F + \Delta t / 2)} \quad (\text{B.29})$$

resulta:

$$E_{fd}(t) = E_0 - b|V_T(t)| \quad (\text{B.30})$$

## B.2 Esquema Alternado Entrelaçado Implícito

Neste item descreve-se o esquema Alternado Entrelaçado Implícito de resolução das EDAs, especificamente para os modelos dos elementos de SEE apresentados no apêndice A.



## B.2.1 Resumo da Modelagem

Neste item faz-se um resumo dos modelos apresentados no apêndice A, em que o comportamento dinâmico é representado por equações algébricas a diferenças, deduzidas anteriormente.

**Para o rotor de uma máquina síncrona:**

$$\left\{ \begin{array}{l} \delta(t) = \alpha(t - \Delta t) - \frac{\omega_s(\Delta t)^2}{8H} P_e(t) \\ \omega(t) = \frac{2}{\Delta t} [\delta(t) - \delta(t - \Delta t)] - [\omega(t - \Delta t) - 2\omega_s] \\ E'_q(t) = a_d(x_d - x'_d)I_d(t) + a_d E_{fd}(t) + F_q \\ E'_d(t) = -a_q(x_q - x'_q)I_q(t) + F_d \end{array} \right. \quad (\text{B.31})$$

onde  $a_d$ ,  $F_q$ ,  $a_q$ , e  $F_d$  são dados pelas equações (B.14).

**Para o estator de uma máquina síncrona:**

$$\left\{ \begin{array}{l} E'_d - V_d = x'_q I_q \\ E'_q - V_q = -x'_d I_d \\ P_e = E'_d I_d + E'_q I_q - (x'_q - x'_d) I_d I_q \end{array} \right. \quad (\text{B.32})$$

**Para o sistema de excitação:**

$$E_{fd}(t) = E_0(t - \Delta t) - b|V_T(t)|, \quad (\text{B.33})$$

onde as constantes  $E_0(t - \Delta t)$  e  $b$  diferem para modelos diferentes de sistemas de excitação. Essas constantes foram deduzidas anteriormente para o sistema de excitação IEEE tipo 1.

**Para o sistema de transmissão:**

O sistema de transmissão é descrito através da matriz admitância nodal na forma complexa e representado pela seguinte equação matricial:

$$I(E, V) = YV \quad (\text{B.34})$$

onde:

$I$  - vetor de injeções de correntes nodais de geração e carga;

$Y$  - matriz admitância nodal;

$V$  - vetor de tensões nodais.

$E = [E'_d, E'_q, \delta]^T$  - vetor de variáveis de interface.

Note que, conforme indica a equação (B.34), as cargas e os estatores das máquinas síncronas devem ser representados por injeções de correntes. Desse modo as variáveis de estator são convertidas para a referência complexa a fim de representá-lo por um equivalente Norton, conforme descrito no próximo item.

### B.2.2 Equivalente Norton para o Estator

O equivalente Norton do estator de uma máquina síncrona é obtido aplicando-se uma transformação de coordenadas  $d - q \rightarrow \Re - \Im$  (ver apêndice A), às variáveis  $E'_d, E'_q, V_d, V_q, I_d$  e  $I_q$ , referindo-as ao sistema complexo e obtendo-se a seguinte equação, em termos de injeção de corrente [31]:

$$I_g = -j\frac{1}{2} \left[ \frac{x'_d + x'_q}{x'_d x'_q} \right] [E' - V] + j\frac{1}{2} \left[ \frac{x'_d - x'_q}{x'_d x'_q} \right] [E'^* - V^*] \exp j2\delta \quad (\text{B.35})$$

Note que todas as quantidades da equação (B.35) são fasores referidos a referência síncrona e o símbolo “\*” indica complexo conjugado. A diferença  $(x'_d - x'_q)$  é conhecida como saliência transitória.

Assim, a injeção de corrente que representa o comportamento do estator de uma máquina síncrona, no sistema de referência complexo, constitui-se de uma parcela dependente exclusivamente da tensão interna  $E'$ , uma parcela dependente exclusivamente da tensão terminal  $V$  e uma terceira parcela que depende da saliência transitória, das tensões  $E'$  e  $V$ , e do ângulo  $\delta$ . No caso de saliência transitória desprezível, o terceiro termo anula-se.

Simbolicamente pode-se escrever a equação (B.35) na forma seguinte:

$$I_g = I_{eq}(E') - y_{eq}V + I_{dep}(\text{saliência}) \quad (\text{B.36})$$

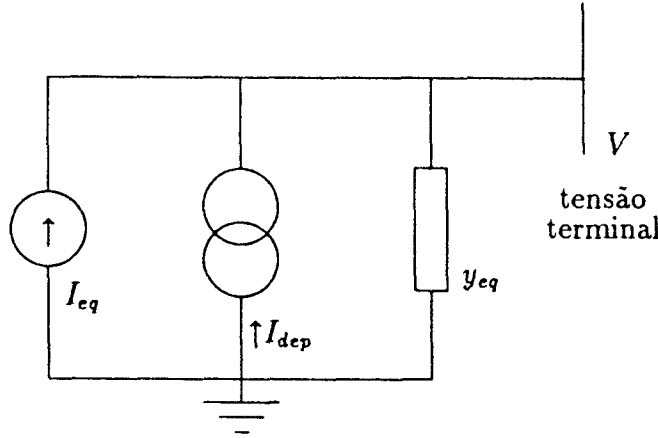


Figura B.1: Circuito equivalente Norton do estator

onde:

$$I_{eq} = -j\frac{1}{2} \left[ \frac{x'_d + x'_q}{x'_d x'_q} \right] E' \quad (\text{B.37})$$

$$y_{eq} = -j\frac{1}{2} \left[ \frac{x'_d + x'_q}{x'_d x'_q} \right] \quad (\text{B.38})$$

$$I_{dep} = j\frac{1}{2} \left[ \frac{x'_d - x'_q}{x'_d x'_q} \right] [E'^* - V^*] \exp j2\delta \quad (\text{B.39})$$

A figura B.1 representa o circuito elétrico equivalente para a equação (B.36) e conseqüentemente para o estator da máquina síncrona.

Observa-se que para os modelos que levam em conta os efeitos subtransitórios nos circuitos do rotor, também pode ser obtido um equivalente Norton para o estator. Porém, na equação (B.35), deve-se substituir a tensão transitória  $E'$  pela sua correspondente tensão subtransitória ( $E''$ ), as reatâncias transitórias ( $x'_d$  e  $x'_q$ ) por reatâncias subtransitórias ( $x''_d$  e  $x''_q$ ), e conseqüentemente a saliência transitória pela saliência subtransitória, de efeito muito menos significativo e, em geral, desprezível.

No que se refere ao acoplamento da máquina síncrona ao sistema de transmissão, as parcelas de corrente do equivalente Norton,  $I_{eq}$  e  $I_{dep}$ , são tratadas como injeções de correntes e a admitância,  $y_{eq}$ , é incorporada a matriz admitância nodal  $Y$ .

### B.2.3 Cargas

Neste trabalho as cargas são representadas por um modelo polinomial, constituído de parcelas que refletem o comportamento de impedância constante, corrente constante e potência constante, conforme definido no apêndice A. Semelhantemente ao caso dos estatores das máquinas síncronas, cada carga é transformada em um equivalente Norton de modo a compatibilizá-las com a representação nodal da rede elétrica. A admitância do equivalente é incluída diretamente na matriz admitância nodal. Nas barras em que a carga se comporta como impedância constante, toda a carga é transformada em admitância e a injeção de corrente é nula. Por outro lado, nas barras em que a carga tem parcelas que se comportam diferentemente de impedância constante, obtém-se uma admitância equivalente ( $y_0$ ) para as condições de operação em regime permanente pré-perturbação, que é incluída na matriz admitância nodal, conforme a equação (B.40).

$$y_0 = g_0 + jb_0 = \frac{P_0}{|V_0|^2} - j \frac{Q_0}{|V_0|^2} \quad (\text{B.40})$$

Ao longo da simulação e em função da tensão na barra e do modelo de carga especificado, calcula-se o valor atual da admitância da carga, isto é:

$$y = \frac{S^*}{|V|^2} \quad (\text{B.41})$$

Assim, a fonte de corrente do equivalente Norton representa o desvio do comportamento da carga em relação ao de impedância constante; e é dada por:

$$I_{eq} = (y_0 - y)V \quad (\text{B.42})$$

As expressões gerais para  $I_{eq}$ , no caso de um modelo de carga polinomial composto, são apresentadas a seguir.

Tomando-se as equações de potência para os três tipos de modelos de carga, equações (A.13), (A.14) e (A.15), e transformando-as em correntes, resulta:

- Parcela de potência constante:

$$I_s = (a_1 P_0 - j a_2 Q_0) \frac{1}{V^*}; \quad (\text{B.43})$$

- Parcela de corrente constante:

$$I_I = (b_1 P_0 - j b_2 Q_0) \frac{1}{V_0^*}; \quad (\text{B.44})$$

- Parcela de impedância constante:

$$y_c = g_c + j b_c = \frac{c_1 P_0}{|V_0|^2} - j \frac{c_2 Q_0}{|V_0|^2}. \quad (\text{B.45})$$

Assim, a injeção de corrente equivalente da carga,  $I_{eq}$  será dada por:

$$I_{eq} = y_0 V - I_s - I_I - I_e \quad (\text{B.46})$$

ou ainda

$$I_{eq} = \left[ 1 - a_1 \frac{|V_0|^2}{|V|^2} - b_1 \frac{V_0}{V} - c_1 \right] g_0 V + j \left[ 1 - a_2 \frac{|V_0|^2}{|V|^2} - b_2 \frac{V_0}{V} - c_2 \right] b_0 V \quad (\text{B.47})$$

## B.2.4 Interface

Na modelagem matemática apresentada até então, a rede de transmissão, as cargas e os estatores estão incluídos na equação (B.34). Os rotores das máquinas síncronas e os sistemas de excitação são representados pelas equações (B.31) e (B.33). Contudo, deve ser realizada uma interação entre as soluções das equações (B.31) e (B.33) com a solução da equação (B.34). Neste sentido deve-se estabelecer uma relação entre a tensão  $V$ , da barra onde a máquina estiver conectada, e as correntes de estator,  $I_d$  e  $I_q$ , além de uma expressão para a potência elétrica,  $P_e$ . Esta relação é obtida aplicando-se a transformação de coordenadas  $d - q \rightarrow \Re - \Im$ , às variáveis  $V_d$  e  $V_q$  das equações do estator. Assim as equações resultantes para  $I_d$ ,  $I_q$  e  $P_e$  são conhecidas como equações de interface, e dadas por:

$$\begin{cases} I_d = [-E'_q + V_r \cos \delta + V_m \sin \delta] / x'_d \\ I_q = [E'_d + V_r \sin \delta - V_m \cos \delta] / x'_q \\ P_e = E'_d I_d + E'_q I_q - (x'_q - x'_d) I_d I_q \end{cases} \quad (\text{B.48})$$

## B.2.5 Síntese do Modelo Global

Tomando-se a modelagem matemática definida anteriormente, o comportamento dinâmico de um SEE, no intervalo de tempo  $t - \Delta t$  a  $t$  e supondo o sistema constituído de  $m$  geradores e  $n$  barras, pode ser genericamente representado pelas seguintes equações:

$$\left\{ \begin{array}{l} x_1(t) = H_1(x_1(t), x_1(t - \Delta t), u_1(t), u_1(t - \Delta t)) \\ x_2(t) = H_2(x_2(t), x_2(t - \Delta t), u_2(t), u_2(t - \Delta t)) \\ \vdots \\ x_m(t) = H_m(x_m(t), x_m(t - \Delta t), u_m(t), u_m(t - \Delta t)) \end{array} \right. \quad (\text{B.49})$$

$$\begin{bmatrix} I_1(E_1(t), V_1(t)) \\ I_2(E_2(t), V_2(t)) \\ \vdots \\ I_n(E_n(t), V_n(t)) \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1(t) \\ V_2(t) \\ \vdots \\ V_n(t) \end{bmatrix} \quad (\text{B.50})$$

$$\left\{ \begin{array}{l} u_1(t) = h_1(E_1(t), V_1(t)) \\ u_2(t) = h_2(E_2(t), V_2(t)) \\ \vdots \\ u_m(t) = h_m(E_m(t), V_m(t)) \end{array} \right. \quad (\text{B.51})$$

onde:

$x_i = [\omega_i, \delta_i, E'_{qi}, E'_{di}, E_{fdi}]^T$  - vetor de variáveis de estado do  $i$ -ésimo gerador;

$E_i = [E'_{qi}, E'_{di}, \delta_i]^T$  - subvetor de variáveis de interface do  $i$ -ésimo gerador;

$u_i = [I_{di}, I_{qi}, P_{ei}]^T$  - vetor de variáveis das equações de interface do  $i$ -ésimo gerador.

Definindo-se:

$H = [H_1, H_2, \dots, H_m]^T$  - vetor de funções não-lineares das equações algébricas de diferenças;

$x = [x_1, x_2, \dots, x_m]$  - vetor de variáveis de estado;

$u = [u_1, u_2, \dots, u_m]^T$  - vetor de variáveis das equações de interface;

$I = [I_1, I_2, \dots, I_n]^T$  - vetor de injeções de correntes nodais;

$V = [V_1, V_2, \dots, V_n]^T$  - vetor de tensões nodais;

$h = [h_1, h_2, \dots, h_m]^T$  - vetor de funções não-lineares de interface;

pode-se escrever as equações (B.49), (B.50) e (B.51) na seguinte forma sintética:

$$\begin{cases} x(t) = H(x(t), x(t - \Delta t), u(t), u(t - \Delta t)) \\ I(E, V) = YV \\ u = h(E, V) \end{cases} \quad (\text{B.52})$$

## B.2.6 Algoritmo Alternado Entrelaçado Implícito (AEI)

A partir da modelagem sintética representada pelas equações (B.52), o esquema AEI segue basicamente o seguinte algoritmo:

### Algoritmo AEI (sucinto)

1. *Determine as condições iniciais  $x_0, V_0$  e  $u_0$ ;*
2. *Monte a matriz  $Y$  incluindo a admitância das cargas e as reatâncias dos geradores;*
3. *Verifique: Se houve alteração na rede, atualize  $Y$ , resolva  $I(E, V) = YV$  e calcule  $u(t)$ . Senão, vá para 4;*
4. *Verifique: Se  $t \geq t_{final}$ , FIM. Senão, vá para 5;*
5. *Incremente o contador de tempo  $t$ ;  $t \leftarrow t + \Delta t$  e inicialize o contador de iterações  $l$ :  $l \leftarrow 0$ ;*
6. *Extrapole  $u^*(t)$  ou, para os instantes de chaveamento, use o valor do passo de integração anterior,  $u^*(t) = u(t - \Delta t)$ ;*

7. Calcule  $x^0(t) = H(x^*(t), u^*(t), x(t - \Delta t), u(t - \Delta t))$ ; onde o sobrescrito “\*” indica condição inicial correspondente a valor extrapolado ou do passo anterior;
8. Obtenha  $E^l(t)$  e  $I(E^l(t), V^l(t))$ ;
9. Resolva  $I(E^l(t), V^l(t)) = YV^{l+1}(t)$ ;
10. Obtenha  $u^{l+1}(t) = h(E^l(t), V^{l+1}(t))$
11. Resolva:  $x^{l+1}(t) = H(x^l(t), u^{l+1}(t), x(t - \Delta t), u(t - \Delta t))$
12. Teste: Se  $\|x^{l+1} - x^l(t)\|_2^2 \leq \epsilon$ , vá para 3; Senão, vá para 13;
13. Incremente o contador de iterações  $l$ :  $l \leftarrow l + 1$  e vá para 8;

A forma detalhada do algoritmo AEI, em termos das variáveis empregadas na modelagem de cada componente do SEE, é apresentada a seguir:

### Algoritmo AEI (detalhado)

1. Determinar as condições iniciais  $x_0, V_0$  e  $u_0$ ;

Valores fornecidos pelo fluxo de carga

$$\begin{aligned} S_{0i} &= P_{0i} + jQ_{0i} \\ e \quad V_{0i} \\ i &= 1, 2, \dots, n \end{aligned}$$

Variáveis  $x_0$  e  $u_0$

$$\left\{ \begin{array}{l} I_i = (P_i - jQ_i)/V_i^* \\ E_{dqi} = V_i + jx_{qi}I_i \\ \delta_i = \arctg \frac{E_{dqmi}}{E_{dqrs}} \\ I_{di} = -I_{ri} \sin \delta_i + I_{mi} \cos \delta_i \\ I_{qi} = I_{ri} \cos \delta_i + I_{mi} \sin \delta_i \\ E'_{qi} = -x'_{di}I_{di} + V_{rei} \cos \delta_i + V_{imi} \sin \delta_i \\ E_{di} = x'_{qi}I_{qi} - V_{rei} \sin \delta_i + V_{imi} \cos \delta_i \\ E_{fdi} = |E_{qdi}| - (x_{di} - x_{qi})I_{di} \\ \omega_i = 0 \\ P_{ei} = E'_{di}I_{di} + E'_{qi}I_{qi} - (x'_{qi} - x'_{di})I_{di}I_{qi} \\ i = 1, 2, \dots, m \end{array} \right.$$



2. Monte a matriz  $Y$  incluindo a admitância das cargas e as reatâncias dos geradores;
3. Verifique: Se houve alteração na rede, atualize  $Y$ , resolva  $I(E, V) = YV$  e calcule  $u(t)$ . Senão, vá para 4;
4. Verifique: Se  $t \geq t_{final}$ , FIM. Senão, vá para 5;
5. Incremente o contador de tempo  $t$ ;  $t \leftarrow t + \Delta t$  e inicialize o contador de iterações  $l$ :  $l \leftarrow 0$ ;
6. Extrapole  $u^*(t)$  ou, para os instantes de chaveamento, use o valor do passo de integração anterior;

$$\begin{cases} P_{ei}^*(t) = 2P_{ei}(t - \Delta t) - P_{ei}(t - 2\Delta t) \\ I_{di}^*(t) = (-F_{qi} + V_{qi}^*(t) - a_{di}E_{fdi})/x'_{dimod} \\ I_{qi}^*(t) = (F_{di} - V_{di}^*(t))/x'_{qimod} \\ i = 1, 2, \dots, m \end{cases}$$

onde  $x'_{dimod} = x'_{di} + a_{di}(x_{di} - x'_{di})$  e  $x'_{qimod} = x'_{qi} + a_{qi}(x_{qi} - x'_{qi})$ .

7. Calcule  $x^0(t) = H(x^*(t), u^*(t), x(t - \Delta t), u(t - \Delta t))$ , onde o sobrescrito “\*” indica a condição inicial correspondente ao valor extrapolado ou do passo anterior. Use as seguintes expressões:

$$\begin{cases} \delta_i(t) = \alpha_i(t - \Delta t) - \frac{\omega_s(\Delta t)^2}{8H_i} P_{ei}(t) \\ \omega_i(t) = \frac{2}{\Delta t} [\delta_i(t) - \delta_i(t - \Delta t)] - [\omega_i(t - \Delta t) - 2\omega_s] \\ E'_{qi}(t) = a_{di}(x_{di} - x'_{di})I_{di}(t) + a_{di}E_{fdi}(t) + F_{qi} \\ E'_{di}(t) = -a_{qi}(x_{qi} - x'_{qi})I_{qi}(t) + F_{di} \end{cases}$$

onde:

$$\left\{ \begin{array}{l} \alpha_i(t - \Delta t) = \delta_i(t - \Delta t) + [\omega_i(t - \Delta t) - \omega_s] \Delta t + \frac{\omega_s(\Delta t)^2}{8H_i} [2P_{mi} - P_{ei}(t - \Delta t)] \\ a_{di} = \frac{\Delta t}{2T'_{do} + \Delta t} \\ F_{qi} = a_{di} [E_{fdi}(t - \Delta t) - 2E'_{qi}(t - \Delta t) + (x_{di} - x'_{di})I_{di}(t - \Delta t)] + E'_{qi}(t - \Delta t) \\ a_{qi} = \frac{\Delta t}{2T'_{qo} + \Delta t} \\ F_{di} = E'_{di}(t - \Delta t) - a_{qi} [2E'_{di}(t - \Delta t) + (x_{qi} - x'_{qi})I_{qi}(t - \Delta t)] \\ i = 1, 2, \dots, m \end{array} \right.$$

8. Obtenha  $E^l(t) = [E'_{qi}(t), E'_{di}(t), \delta_i(t)]^T$  e calcule as injeções de correntes  $I(E^l(t), V^l(t))$ , nas barras de geração e carga, como segue:

Para barras que contém geração

$$I_{gi}(t) = I_{eqi}(E'_i) + I_{depi}(\text{saliência})$$

onde:

$$\begin{cases} I_{eqi}(t) = -j\frac{1}{2} \left[ \frac{x'_{di} + x'_{qi}}{x_{di} x_{qi}} \right] E'_i(t) \\ I_{depi}(t) = j\frac{1}{2} \left[ \frac{x'_{di} - x'_{qi}}{x_{di} x_{qi}} \right] [E'_i(t) - V_i(t)] \exp j2\delta_i(t) \\ i = 1, 2, \dots, m \end{cases}$$

Para barras que contém carga

$$I_{eqi} = \left[ 1 - a_{1i} \frac{|V_{0i}|^2}{|V_i|^2} - b_{1i} \frac{V_{0i}}{V_i} - c_{1i} \right] g_{0i} V_i + j \left[ 1 - a_{2i} \frac{|V_{0i}|^2}{|V_i|^2} - b_{2i} \frac{V_{0i}}{V_i} - c_{2i} \right] b_{0i} V_i$$

$$i = 1, 2, \dots, n$$

onde:

$$\begin{aligned} g_{0i} &= \frac{P_{0i}}{|V_{0i}|^2} \\ e \\ b_{0i} &= \frac{-Q_{0i}}{|V_{0i}|^2} \end{aligned}$$

Note que para as barras que contém geração e carga, deve-se somar as injeções de corrente correspondentes;

9. Resolva  $I(E^l(t), V^l(t)) = YV^{l+1}(t)$ ;

10. Obtenha  $u^{l+1}(t) = h(E^l(t), V^{l+1}(t))$ , usando as seguintes expressões:

$$\begin{cases} I_{di}(t) = \left[ -E'_{qi}(t) + V_{ri}(t) \cos \delta_i(t) - V_{mi}(t) \sin \delta_i(t) \right] / x'_{di} \\ I_{qi}(t) = \left[ E'_{di}(t) + V_{ri}(t) \sin \delta_i(t) - V_{mi}(t) \cos \delta_i(t) \right] / x'_{qi} \\ P_{ei}(t) = E'_{di}(t)I_{di}(t) + E'_{qi}(t)I_{qi}(t) + (x'_{qi} - x'_{di})I_{di}(t)I_{qi}(t) \\ i = 1, 2, \dots, m \end{cases}$$

11. Resolva:  $x^{l+1}(t) = H(x^l(t), u^{l+1}(t), x(t - \Delta t), u(t - \Delta t))$ , usando as seguintes expressões:

$$\begin{cases} \delta_i(t) = \alpha_i(t - \Delta t) - \frac{\omega_s(\Delta t)^2}{8H_i} P_{ei}(t) \\ \omega_i(t) = \frac{2}{\Delta t} [\delta_i(t) - \delta_i(t - \Delta t)] - [\omega_i(t - \Delta t) - 2\omega_s] \\ E'_{qi}(t) = a_{di}(x_{di} - x'_{di})I_{di}(t) + a_{di}E_{fdi}(t) + F_{qi} \\ E'_{di}(t) = -a_{qi}(x_{qi} - x'_{qi})I_{qi}(t) + F_{di} \end{cases}$$

Note que as grandezas que dependem do instante de tempo  $t - \Delta t$ , já foram calculadas no passo 7

12. Teste: Se  $\|x^{l+1} - x^l(t)\|_2 \leq \epsilon$ , vá para 3. Senão, vá para 13;

13. Incremente o contador de iterações  $l$ :  $l \leftarrow l + 1$  e vá para 8;

## B.3 Esquema Simultâneo Implícito (SI)

Neste item descreve-se o esquema SI de resolução de EDAs no contexto da SDSEE, especificamente para os modelos de SEE apresentados no apêndice A. As equações dos diversos elementos que compõe o sistema, já na forma algebrizada para o caso das equações diferenciais, são apresentadas no subitens seguintes:

### B.3.1 Resumo da Modelagem

Neste item apresenta-se um resumo dos modelos, onde o comportamento dinâmico é representado por equações algébricas a diferenças.

**Para o rotor de uma máquina síncrona**

$$\left\{ \begin{array}{l} \omega(t) = -\frac{\omega_s \Delta t}{4H} P_e(t) + \sigma(t - \Delta t) \\ \delta(t) = \frac{\Delta t}{2} \omega(t) + \beta(t - \Delta t) \\ E'_q(t) = a_d(x_d - x'_d)I_d(t) + a_d E_{fd}(t) + F_q \\ E'_d(t) = -a_q(x_q - x'_q)I_q(t) + F_d \end{array} \right. \quad (\text{B.53})$$

**Para o estator de uma máquina síncrona**

$$\left\{ \begin{array}{l} E'_q(t) - V_q(t) + x'_d I_d(t) = 0 \\ E'_d(t) - V_d(t) - x'_q I_q(t) = 0 \\ P_e(t) - E'_q(t)I_q(t) - E'_d(t)I_d(t) + (x'_q - x'_d)I_d(t)I_q(t) = 0 \end{array} \right. \quad (\text{B.54})$$

**Para o sistema de excitação**

$$E_{fd}(t) = E_0(t - \Delta t) - b|V_T| \quad (\text{B.55})$$

**Para o sistema de Transmissão**

$$I^e(E, V^e) = Y^e V^e \quad (\text{B.56})$$

onde o sobrescrito *e* indica a forma expandida dos vetores (parcelas real e imaginária).

### B.3.2 Equações de Interface

No equacionamento do esquema SI que se está desenvolvendo, as tensões e correntes de eixo direto,  $V_d$  e  $I_d$ , e de eixo quadratura,  $V_q$  e  $I_q$ , do estator, serão expressas em relação ao sistema de referência síncrona, conforme as seguintes equações:

$$\left\{ \begin{array}{l} V_d(t) = -V_r(t) \text{sen } \delta(t) + V_m(t) \text{cos } \delta(t) \\ V_q(t) = V_r(t) \text{cos } \delta(t) + V_m(t) \text{sen } \delta(t) \end{array} \right. \quad (\text{B.57})$$

$$\left\{ \begin{array}{l} I_d(t) = -I_r(t) \text{sen } \delta(t) + I_m(t) \text{cos } \delta(t) \\ I_q(t) = I_r(t) \text{cos } \delta(t) + I_m(t) \text{sen } \delta(t) \end{array} \right. \quad (\text{B.58})$$

### B.3.3 Equações das Cargas

As cargas são representadas por um modelo polinomial e da mesma forma como foi feito para o caso do esquema AEI, ou seja, através da definição de um equivalente Norton para cada carga. A única diferença é que, no caso do esquema SI, as admitâncias do equivalente e a injeção de corrente equivalente são tratadas na forma expandida de suas parcelas real e imaginária. Assim, as partes real ( $g_0$ ) e imaginária ( $b_0$ ) da admitância equivalente da carga são dadas pelas equações:

$$g_0 = \frac{P_0}{|V_0|^2} \quad (\text{B.59})$$

$$b_0 = \frac{-Q_0}{|V_0|^2}; \quad (\text{B.60})$$

e a partes real  $I_{req}$  e imaginária  $I_{meq}$  das correntes equivalentes, pelas equações:

$$\begin{cases} I_{req} = \left[1 - a_1 \frac{|V_0|^2}{|V|^2} - b_1 \frac{V_{0r}}{V_r} - c_1\right] g_0 V_r - \left[1 - a_2 \frac{|V_0|^2}{|V|^2} - b_2 \frac{V_{0m}}{V_m} - c_2\right] b_0 V_m \\ I_{meq} = \left[1 - a_1 \frac{|V_0|^2}{|V|^2} - b_1 \frac{V_{0m}}{V_m} - c_1\right] g_0 V_m + \left[1 - a_2 \frac{|V_0|^2}{|V|^2} - b_2 \frac{V_{0r}}{V_r} - c_2\right] b_0 V_r \end{cases} \quad (\text{B.61})$$

### B.3.4 Modelo Global para o Intervalo $t - \Delta t$ a $t$

Escolhendo-se o vetor  $[x, V^e] = [V_T, E_{fd}, E'_d, E'_q, \delta, \omega, I_d, I_q, V_m, V_r]^T$  como o vetor de variáveis do problema, para cada máquina, e de acordo com os modelos desenvolvidos anteriormente, pode-se escrever as seguintes funções para representar o comportamento de uma máquina síncrona, no intervalo  $t - \Delta t$  a  $t$ .

$$\left\{ \begin{array}{l}
f_1 = V_T^2(t) - V_r^2(t) - V_m^2(t) \\
f_2 = E_{fd}(t) - E_0(t - \Delta t) + b|V_T(t)| \\
f_3 = E'_d(t) + a_q(x_q - x'_q)I_q(t) - F_d \\
f_4 = E'_q(t) - a_d(x_d - x'_d)I_d(t) - a_d E_{fd}(t) - F_q \\
f_5 = \delta(t) - \frac{\Delta t}{2}\omega(t) - \beta(t - \Delta t) \\
f_6 = \omega(t) + \frac{\omega_i \Delta t}{4H} \left[ E'_q(t)I_q(t) + E'_d(t)I_d(t) - (x'_q - x'_d)I_d(t)I_q(t) \right] - \sigma(t - \Delta t) \\
f_7 = E'_q(t) - V_r(t) \cos \delta(t) - V_m(t) \sin \delta(t) + x'_d I_d(t) \\
f_8 = E'_d(t) + V_r(t) \sin \delta(t) - V_m(t) \cos \delta(t) - x'_q I_q(t) \\
g_1 = -I_d(t) \sin \delta(t) + I_q(t) \cos \delta(t) \\
g_2 = I_d(t) \cos \delta(t) + I_q(t) \sin \delta(t)
\end{array} \right. \quad (B.62)$$

Note que para as equações (B.62), as cargas foram representadas por impedâncias constantes e incluídas na matriz  $Y^e$ .

Definindo-se, para a  $i$ -ésima máquina síncrona:

$F_i = [f_1, f_2, \dots, f_8]^T$  - vetor de funções das equações do rotor e estator;

$G_i = [g_1, g_2]^T$  - vetor de funções das correntes do estator;

$x_i = [V_T, E_{fd}, E'_d, E'_q, \delta, \omega, I_d, I_q]^T$  - vetor de variáveis das equações do rotor e estator;

$V_i^e = [V_{mi}, V_{ri}]^T$  - vetor das partes real e imaginária da tensão terminal;

$I_i^e = [I_{ri}, I_{mi}]^T$  - vetor das partes real e imaginária da corrente de estator;

o modelo de todo o SEE pode ser representado pelo sistema de equações genéricas (2.16), reescrito a seguir:

$$\left\{ \begin{array}{l} F_1(x_1, V^e) = 0 \\ F_2(x_2, V^e) = 0 \\ \vdots \\ F_m(x_m, V^e) = 0 \\ Y^e V^e - G'(x, V^e) = 0 \end{array} \right. \quad (\text{B.63})$$

Desse modo as sub-matrizes jacobianas  $\frac{\partial F_i}{\partial x_i}$ ,  $\frac{\partial G'_i}{\partial x_i}$  e  $\frac{\partial F_i}{\partial V}$ , definidas nas equações (2.17) a (2.20), têm, para esse modelo particular de SEE, a forma seguinte:

$$\frac{\partial F_i}{\partial x_i} =$$

$V_T$	$E_{fd}$	$E'_d$	$E'_q$	$\delta$	$\omega$	$I_d$	$I_q$
$2V_T$							
$b$	1						
		1					$a_q(x_q - x'_q)$
	$-a_d$		1			$-a_d(x_d - x'_d)$	
				1	$-\frac{\Delta t}{2}$		
		$\frac{\omega_s \Delta t}{4H} I_d$	$\frac{\omega_s \Delta t}{4H} I_q$		1	$\frac{\omega_s \Delta t}{4H} [E'_d - (x_q - x'_d) I_q]$	$\frac{\omega_s \Delta t}{4H} [E'_q - (x_q - x'_d) I_d]$
			1	$V_r \text{ sen } \delta - V_m \text{ cos } \delta$		$x'_d$	
		1		$V_r \text{ cos } \delta + V_m \text{ sen } \delta$			$-x'_q$

(B.64)

$$\frac{\partial G'_i}{\partial x_i} =$$

$V_T$	$E_{fd}$	$E'_d$	$E'_q$	$\delta$	$\omega$	$I_d$	$I_q$
				$-I_d \text{ cos } \delta - I_q \text{ sen } \delta$		$-\text{sen } \delta$	$\text{cos } \delta$
				$-I_d \text{ sen } \delta + I_q \text{ cos } \delta$		$\text{cos } \delta$	$\text{sen } \delta$

(B.65)

$$\frac{\partial F_i}{\partial V_i^e} = \begin{array}{|c|c|} \hline V_m & V_r \\ \hline -2V_m & -2V_r \\ \hline & \\ \hline & \\ \hline & \\ \hline & \\ \hline -\text{sen } \delta & -\text{cos } \delta \\ \hline -\text{cos } \delta & \text{sen } \delta \\ \hline \end{array} \quad (\text{B.66})$$

$$\frac{\partial G_i'}{\partial V_i^e} = 0 \quad (\text{B.67})$$

### B.3.5 Algoritmo Simultâneo Implícito (SI)

De acordo com a modelagem definida anteriormente, o esquema SI para a SDSEE segue, basicamente, o seguinte algoritmo:

#### Algoritmo SI

1. *Determine as condições iniciais  $x^0$  e  $V^{e0}$ ;*
2. *Monte a matriz  $Y^e$  incluindo a admitância das cargas;*
3. *Verifique: Se houve alteração na rede, atualize  $Y^e$  e vá para 4. Senão, vá para 4;*
4. *Verifique: Se  $t \geq t_{final}$ . FIM. Senão, vá para 5;*
5. *Incremente o contador de tempo  $t$ :  $t \leftarrow t + \Delta t$ ;*
6. *Faça  $k = 0$ ,  $V^{ek} = V^{e0}$  e  $x^k = x^0$ ;*
7. *Calcule:  $H(x^k, V^{ek}) = [F(x^k, V^{ek}), G(x^k, V^{ek})]^T$ , conforme as equações (B.62) e (B.63);*
8. *Verifique a convergência : Se  $\|H(x^k, V^{ek})\|_2^2 \leq \epsilon$ , vá para 3. Senão, vá para 9;*



9. Calcule:  $\bar{G}$  e  $\bar{J}_4$ , definidos na equação (2.23);
10. Resolva:  $-\bar{J}_4 \cdot \Delta V^{ek+1} = \bar{G}$  e calcule  $V^{ek+1} = V^{ek} + \Delta V^{ek+1}$ ;
11. Resolva  $\Delta x_i = -A_{M_i}^{-1}(F_i + B_i \cdot \Delta V^e)$  e calcule  $x_i^{k+1} = x_i^k + \Delta x_i^{k+1}$ , para  $i = 1, 2, \dots, m$
12. Incremente o contador de iterações  $k$ :  $k \leftarrow k + 1$ , e vá para 7.

# Apêndice C

## Esquema Simultâneo Implícito Paralelo no Tempo (SIMPT) - Modelagem Matemática

### C.1 Equações dos Elementos do Sistema

#### C.1.1 Sistema de Excitação IEEE Tipo 1

Equações Diferenciais

$$\left\{ \begin{array}{l} \frac{dE_{fd}}{dt} = \frac{1}{T_E} [V_R - (K_E + K_1)E_{fd} + K_2] \\ \frac{dX_1}{dt} = \frac{1}{T_R} [V_T - X_1] \\ \frac{dV_R}{dt} = \frac{K_A}{T_A} [V_{REF} - X_1 - X_2] - \frac{1}{T_A} V_R \\ \frac{dX_2}{dt} = \frac{K_F}{T_F} \frac{dE_{fd}}{dt} - \frac{1}{T_F} X_2 \end{array} \right. \quad (C.1)$$

Equações Algebrizadas

Definindo-se:

$$\left\{ \begin{array}{ll} a_R = \frac{2T_R}{\Delta t} + 1 & ; \quad a_A = \frac{2T_A}{\Delta t} + 1 ; \\ a_E = \frac{2T_E}{\Delta t} + K_E + K_1 & ; \quad b_E = \frac{2T_E}{\Delta t} - K_E - K_1 ; \\ b_R = \frac{2T_R - \Delta t}{2T_R + \Delta t} & ; \quad b_A = \frac{2T_A - \Delta t}{2T_A + \Delta t} ; \\ b_F = \frac{2K_F}{2T_F + \Delta t} & ; \quad c_F = \frac{2T_F - \Delta t}{2T_F + \Delta t} ; \end{array} \right. \quad (C.2)$$

resulta:

$$\left\{ \begin{array}{l} E_{fd}(t) = \frac{b_E}{a_E} E_{fd}(t - \Delta t) + \frac{1}{a_E} [V_R(t) + V_R(t - \Delta t) + 2K_2] \\ X_1(t) = b_R X_1(t - \Delta t) + \frac{1}{a_R} [V_T(t) + V_T(t - \Delta t)] \\ V_R(t) = \frac{K_A}{a_A} [2V_{REF} - X_1(t) - X_1(t - \Delta t) - X_2(t) - X_2(t - \Delta t)] \\ \quad + b_A V_R(t - \Delta t) \\ X_2(t) = b_F [E_{fd}(t) - E_{fd}(t - \Delta t)] + c_F X_2(t - \Delta t) \end{array} \right. \quad (C.3)$$

onde  $V_T(\cdot) = \sqrt{V_r^2(\cdot) + V_m^2(\cdot)}$

### C.1.2 Sistema de Excitação Tipo AC4

Equações Diferenciais

$$\left\{ \begin{array}{l} \frac{dV_e}{dt} + \frac{T_C}{T_B} \frac{dV_T}{dt} = \frac{1}{T_B} [V_{REF} - V_T - V_e] \\ \frac{dE_{fd}}{dt} = \frac{1}{T_A} [K_A V_e - E_{fd}] \end{array} \right. \quad (C.4)$$

Equações Algebrizadas

Definindo-se;

$$\left\{ \begin{array}{l} a_A = \frac{\Delta t}{(2T_A + \Delta t)} \quad ; \quad a_B = \frac{\Delta t}{(2T_B + \Delta t)} \quad ; \\ a_C = \frac{2T_C + \Delta t}{2T_B + \Delta t} \quad ; \quad b_C = \frac{2T_C - \Delta t}{2T_B + \Delta t} \quad ; \end{array} \right. \quad (C.5)$$

resulta:

$$\left\{ \begin{array}{l} E_{fd}(t) = E_{fd}(t - \Delta t) + a_A \{K_A [V_e(t) + V_e(t - \Delta t)] - 2E_{fd}(t - \Delta t)\} \\ V_e(t) = V_e(t - \Delta t) + a_B [2V_{REF} - 2V_e(t - \Delta t)] - a_C V_T(t) + b_C V_T(t - \Delta t) \end{array} \right. \quad (C.6)$$

### C.1.3 Equações Elétricas do Rotor

Equações Diferenciais

$$\left\{ \begin{array}{l} \frac{dE'_q}{dt} = -\frac{1}{T'_{do}} [E'_q - (x_d - x'_d)I_d - E_{fd}] \\ \frac{dE'_d}{dt} = -\frac{1}{T'_{q0}} [E'_d + (x_q - x'_q)I_q] \end{array} \right. \quad (C.7)$$

onde:

$$\begin{cases} I_d = [-E'_q + V_r \cos \delta + V_m \sin \delta] / x'_d \\ I_q = [E'_d + V_r \sin \delta - V_m \cos \delta] / x'_q \end{cases} \quad (\text{C.8})$$

## Equações Algebrizadas

Definindo-se:

$$\begin{cases} a_d = \frac{\Delta t}{2T'_{d0} + \Delta t} ; & a_q = \frac{\Delta t}{2T'_{q0} + \Delta t} ; \\ Q_x = \frac{x_q - x'_q}{x'_q} ; & D_x = \frac{x_d - x'_d}{x'_d} ; \end{cases} \quad (\text{C.9})$$

resulta:

$$\begin{cases} (1 + a_q Q_x) E'_d(t) = (1 - 2a_q - a_q Q_x) E'_d(t - \Delta t) - \\ \quad a_q Q_x [V_r(t) \sin \delta(t) - V_m(t) \cos \delta(t)] - \\ \quad a_q Q_x [V_r(t - \Delta t) \sin \delta(t - \Delta t) - V_m(t - \Delta t) \cos \delta(t - \Delta t)] \\ (1 + a_d D_x) E'_q(t) = (1 - 2a_d - a_d D_x) E'_q(t - \Delta t) + a_d [E_{fd}(t) + E_{fd}(t - \Delta t)] + \\ \quad a_d D_x [V_r(t) \cos \delta(t) + V_m(t) \sin \delta(t)] + \\ \quad a_d D_x [V_r(t - \Delta t) \cos \delta(t - \Delta t) + V_m(t - \Delta t) \sin \delta(t - \Delta t)] \end{cases} \quad (\text{C.10})$$

## C.1.4 Equações Eletromecânicas do Rotor

### Equações Diferenciais

$$\begin{cases} \frac{d\omega}{dt} = \frac{\omega_s}{2H} (P_m - P_e) \\ \frac{d\delta}{dt} = \omega - \omega_s \end{cases} \quad (\text{C.11})$$

onde  $P_e = \frac{E'_d V_q}{x'_q} - \frac{E'_q V_d}{x'_d} + \frac{(x'_q - x'_d)}{x'_d x'_q} V_d V_q$

Lembrando que

$$\begin{cases} V_d = -V_r \sin \delta + V_m \cos \delta \\ V_q = V_r \cos \delta + V_m \sin \delta \end{cases} \quad (\text{C.12})$$

resulta:

$$\begin{aligned}
P_e &= \frac{E'_d}{x'_q} [V_r \cos \delta + V_m \sin \delta] + \frac{E'_q}{x'_d} [V_r \sin \delta - V_m \cos \delta] \\
&\quad - \left( \frac{x'_q - x'_d}{2x'_d x'_q} \right) [(V_r^2 - V_m^2) \sin 2\delta - 2V_r V_m \cos 2\delta]
\end{aligned} \tag{C.13}$$

### Equações Algebrizadas

$$\begin{cases} \omega(t) = \omega(t - \Delta t) + \frac{\omega_s \Delta t}{4H} [2P_m - P_e(t) - P_e(t - \Delta t)] \\ \delta(t) = \delta(t - \Delta t) + \frac{\Delta t}{2} [\omega(t) + \omega(t - \Delta t) - 2\omega_s] \end{cases} \tag{C.14}$$

onde:

$$\begin{aligned}
P_e(\cdot) &= \frac{E'_d(\cdot)}{x'_q} [V_r(\cdot) \cos \delta(\cdot) + V_m(\cdot) \sin \delta(\cdot)] + \frac{E'_q(\cdot)}{x'_d} [V_r(\cdot) \sin \delta(\cdot) - V_m(\cdot) \cos \delta(\cdot)] \\
&\quad - \left( \frac{x'_q - x'_d}{2x'_d x'_q} \right) [(V_r^2(\cdot) - V_m^2(\cdot)) \sin 2\delta(\cdot) - 2V_r(\cdot)V_m(\cdot) \cos 2\delta(\cdot)]
\end{aligned} \tag{C.15}$$

### C.1.5 Equivalente Norton para o Estator

As equações do estator nas variáveis de Park são:

$$\begin{bmatrix} E'_d \\ E'_q \end{bmatrix} - \begin{bmatrix} V_d \\ V_q \end{bmatrix} = \begin{bmatrix} 0 & x'_q \\ -x'_d & 0 \end{bmatrix} \cdot \begin{bmatrix} I_d \\ I_q \end{bmatrix} \tag{C.16}$$

Aplicando-se a transformação de variáveis  $T$  as tensões e correntes do estator, pode-se escrever:

$$\begin{bmatrix} E'_d \\ E'_q \end{bmatrix} - T \begin{bmatrix} V_r \\ V_m \end{bmatrix} = \begin{bmatrix} 0 & x'_q \\ -x'_d & 0 \end{bmatrix} T \begin{bmatrix} I_r \\ I_m \end{bmatrix} \tag{C.17}$$

Pré-multiplicando-se por  $T^{-1}$ , resulta:

$$T^{-1} \begin{bmatrix} E'_d \\ E'_q \end{bmatrix} - \begin{bmatrix} V_r \\ V_m \end{bmatrix} = T^{-1} \begin{bmatrix} 0 & x'_q \\ -x'_d & 0 \end{bmatrix} T \begin{bmatrix} I_r \\ I_m \end{bmatrix} \tag{C.18}$$

ou

$$T^{-1} \begin{bmatrix} E'_d \\ E'_q \end{bmatrix} - \begin{bmatrix} V_r \\ V_m \end{bmatrix} = T^{-1} [Z_{dq}] \cdot T \begin{bmatrix} I_r \\ I_m \end{bmatrix} \tag{C.19}$$

Definindo-se:

$$Z_s = T^{-1} Z_{dq} T = \begin{bmatrix} -\left(\frac{x'_q - x'_d}{2}\right) \text{sen } 2\delta & -x'_q + (x'_q - x'_d) \cos^2 \delta \\ x'_d + (x'_q - x'_d) \cos^2 \delta & \left(\frac{x'_q - x'_d}{2}\right) \text{sen } 2\delta \end{bmatrix}; \quad (\text{C.20})$$

pode-se escrever:

$$\begin{bmatrix} I_r \\ I_m \end{bmatrix} = Z_s^{-1} \left\{ T^{-1} \begin{bmatrix} E'_d \\ E'_q \end{bmatrix} - \begin{bmatrix} V_r \\ V_m \end{bmatrix} \right\} \quad (\text{C.21})$$

onde:

$$Z_s^{-1} = \frac{1}{x'_q x'_d} \begin{bmatrix} \left(\frac{x'_q - x'_d}{2}\right) \text{sen } 2\delta & x'_q - (x'_q - x'_d) \cos^2 \delta \\ -x'_d - (x'_q - x'_d) \cos^2 \delta & -\left(\frac{x'_q - x'_d}{2}\right) \text{sen } 2\delta \end{bmatrix} \quad (\text{C.22})$$

$$Z_s^{-1} T^{-1} = \frac{1}{x'_q x'_d} \begin{bmatrix} x'_d \cos \delta & x'_q \text{sen } \delta \\ x'_d \text{sen } \delta & -x'_q \cos \delta \end{bmatrix} \quad (\text{C.23})$$

Portanto, pode-se escrever:

$$\begin{aligned} \begin{bmatrix} I_r \\ I_m \end{bmatrix} &= \frac{1}{x'_q x'_d} \begin{bmatrix} x'_d \cos \delta & x'_q \text{sen } \delta \\ x'_d \text{sen } \delta & -x'_q \cos \delta \end{bmatrix} \begin{bmatrix} E'_d \\ E'_q \end{bmatrix} \\ &\quad - \frac{1}{x'_q x'_d} \begin{bmatrix} \left(\frac{x'_q - x'_d}{2}\right) \text{sen } 2\delta & -\left(\frac{x'_q - x'_d}{2}\right) \cos 2\delta \\ -\left(\frac{x'_q - x'_d}{2}\right) \cos 2\delta & -\left(\frac{x'_q - x'_d}{2}\right) \text{sen } 2\delta \end{bmatrix} \begin{bmatrix} V_r \\ V_m \end{bmatrix} \\ &\quad - \frac{1}{x'_q x'_d} \begin{bmatrix} 0 & \left(\frac{x'_q + x'_d}{2}\right) \\ -\left(\frac{x'_q + x'_d}{2}\right) & 0 \end{bmatrix} \begin{bmatrix} V_r \\ V_m \end{bmatrix} \end{aligned} \quad (\text{C.24})$$

ou, simbolicamente, conforme a seguinte equação:

$$\begin{bmatrix} I_r \\ I_m \end{bmatrix} = \begin{bmatrix} I_{req}(E') \\ I_{meq}(E') \end{bmatrix} + \begin{bmatrix} I_{rdcp}(\text{saliência}) \\ I_{mdcp}(\text{saliência}) \end{bmatrix} - y_{eq}^e(\delta) \begin{bmatrix} V_r \\ V_m \end{bmatrix} \quad (\text{C.25})$$

### C.1.6 Síntese das Equações Algébricas da Rede, Estatores e Cargas

Rede

$$I^e = Y^e V^e \quad (C.26)$$

Estator

$$\left\{ \begin{array}{l} I_r = \frac{E'_d}{x'_q} \cos \delta + \frac{E'_q}{x'_d} \sin \delta - \frac{(x'_q - x'_d)}{2x'_q x'_d} V_r \sin 2\delta \\ \quad + \left( \frac{x'_q - x'_d}{2x'_q x'_d} \right) V_m \cos 2\delta - \left( \frac{x'_q + x'_d}{2x'_q x'_d} \right) V_m \\ I_m = \frac{E'_d}{x'_q} \sin \delta - \frac{E'_q}{x'_d} \cos \delta + \frac{(x'_q - x'_d)}{2x'_q x'_d} V_m \sin 2\delta \\ \quad + \left( \frac{x'_q - x'_d}{2x'_q x'_d} \right) V_r \cos 2\delta + \left( \frac{x'_q + x'_d}{2x'_q x'_d} \right) V_r \end{array} \right. \quad (C.27)$$

Cargas

$$\left\{ \begin{array}{l} I_{req} = \left[ 1 - a_1 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_1 \frac{V_{r0}}{V_r} - c_1 \right] g_0 V_r - \\ \quad \left[ 1 - a_2 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_2 \frac{V_{m0}}{V_m} - c_2 \right] b_0 V_m \\ I_{meq} = \left[ 1 - a_1 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_1 \frac{V_{m0}}{V_m} - c_1 \right] g_0 V_m + \\ \quad \left[ 1 - a_2 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_2 \frac{V_{r0}}{V_r} - c_2 \right] b_0 V_r \end{array} \right. \quad (C.28)$$

Note que na equação (C.28),  $I_{req}$  e  $I_{meq}$  representam, respectivamente, as partes real e imaginária da fonte de corrente do equivalente Norton das cargas, ou seja, o desvio do comportamento da carga em relação ao de impedância constante (para maiores detalhes ver apêndice B).

## C.2 Modelagem Global do Sistema

### C.2.1 Modelo Global por Barra

Para Barras de Geração

$$\left\{ \begin{aligned}
 f_1 = 0 &= V_e(t) - V_e(t - \Delta t) - a_B [2V_{REF} - 2V_e(t - \Delta t)] + a_C V_T(t) - \\
 &\quad b_C V_T(t - \Delta t) \\
 f_2 = 0 &= E_{fd}(t) - E_{fd}(t - \Delta t) - a_A K_A [V_e(t) + V_e(t - \Delta t)] + 2a_A E_{fd}(t - \Delta t) \\
 f_3 = 0 &= (1 + a_q Q_x) E'_d(t) - (1 - 2a_q - a_q Q_x) E'_d(t - \Delta t) + \\
 &\quad a_q Q_x [V_r(t) \sin \delta(t) - V_m(t) \cos \delta(t)] + \\
 &\quad a_q Q_x [V_r(t - \Delta t) \sin \delta(t - \Delta t) - V_m(t - \Delta t) \cos \delta(t - \Delta t)] \\
 f_4 = 0 &= (1 + a_d D_x) E'_q(t) - (1 - 2a_d - a_d D_x) E'_q(t - \Delta t) - \\
 &\quad a_d [E_{fd}(t) + E_{fd}(t - \Delta t)] - \\
 &\quad a_d D_x [V_r(t) \cos \delta(t) + V_m(t) \sin \delta(t)] - \\
 &\quad a_d D_x [V_r(t - \Delta t) \cos \delta(t - \Delta t) + V_m(t - \Delta t) \sin \delta(t - \Delta t)] \\
 f_5 = 0 &= \omega(t) - \omega(t - \Delta t) - \frac{\omega_s \Delta t}{4H} [2P_m - P_e(t) - P_e(t - \Delta t)] \\
 f_6 = 0 &= \delta(t) - \delta(t - \Delta t) - \frac{\Delta t}{2} [\omega(t) + \omega(t - \Delta t) - 2\omega_s] \\
 g_1 &= \frac{E'_d}{x'_q} \cos \delta + \frac{E'_q}{x'_d} \sin \delta - \frac{(x'_q - x'_d)}{2x'_q x'_d} V_r \sin 2\delta \\
 &\quad + \left( \frac{x'_q - x'_d}{2x'_q x'_d} \right) V_m \cos 2\delta \\
 g_2 &= \frac{E'_d}{x'_q} \sin \delta - \frac{E'_q}{x'_d} \cos \delta + \frac{(x'_q - x'_d)}{2x'_q x'_d} V_m \sin 2\delta \\
 &\quad + \left( \frac{x'_q - x'_d}{2x'_q x'_d} \right) V_r \cos 2\delta
 \end{aligned} \right.$$

(C.29)



Para Barras de Carga

$$\left\{ \begin{array}{l} g_1 = \left[ 1 - a_1 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_1 \frac{V_{r0}}{V_r} - c_1 \right] g_0 V_r - \\ \quad \left[ 1 - a_2 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_2 \frac{V_{m0}}{V_m} - c_2 \right] b_0 V_m \\ g_2 = \left[ 1 - a_1 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_1 \frac{V_{m0}}{V_m} - c_1 \right] g_0 V_m + \\ \quad \left[ 1 - a_2 \frac{|V_0|^2}{(V_r^2 + V_m^2)} - b_2 \frac{V_{r0}}{V_r} - c_2 \right] b_0 V_r \end{array} \right. \quad (C.30)$$

### C.2.2 Modelo Global para o Sistema no Intervalo de Tempo $(t - \Delta t)$

Define-se:

$x_i = [V_e, E_{fd}, E'_d, E'_q, \delta, \omega]^T$  : vetor de variáveis de estado para a  $i$ -ésima máquina síncrona;

$x = [x_1, x_2, \dots, x_m]^T$  vetor de variáveis de estado do sistema ( $m$  máquinas síncronas);

$V_i^e = [V_{mi}, V_{ri}]^T$  : vetor das parcelas imaginária e real da  $i$ -ésima tensão nodal do sistema;

$V^e = [V_1^e, V_2^e, \dots, V_n^e]^T$  vetor de tensões nodais do sistema;

$F_i(x_i, V_i^e) = [f_1, f_2, \dots, f_6]^T$  : vetor de funções algébricas a diferenças não-lineares para a  $i$ -ésima máquina síncrona;

$G'_i(x_i, V_i^e) = [g_1, g_2]^T$  : vetor de funções algébricas não-lineares para o  $i$ -ésimo nó;

$F = [F_1, F_2, \dots, F_m]^T$  vetor de funções algébricas a diferenças não-lineares para o sistema;

$G' = [G'_1, G'_2, \dots, G'_n]^T$  vetor de funções algébricas não-lineares para o sistema;

Usando-se essas definições e tomando  $G(x, V^e) = Y^e \cdot V^e - G'(x, V^e)$ , o SEE pode ser descrito pelo seguinte sistema de equações:

$$\left\{ \begin{array}{l} F(x, V^e) = 0 \\ G(x, V^e) = 0 \end{array} \right. \quad (C.31)$$

ou, ainda, por:

$$H(y) = 0 \quad (\text{C.32})$$

onde:

$y = [x, V^e]^T$  : vetor de variáveis do sistema;

$H = [F, G]^T$  : vetor de funções algébricas não lineares.

### C.2.3 Modelo para uma Janela de Integração

Neste caso, o sistema de equações (C.32) deve ser alterado para representar todos os  $t = 1, 2, \dots, p$  passos de integração que constituem a janela de integração.

Definindo-se:

**Para o t-ésimo passo de integração**

$\hat{x}_t = [x_1, x_2, \dots, x_m]^T$  : vetor de variáveis de estado do sistema;

$\hat{V}_t^e = [V_1^e, V_2^e, \dots, V_n^e]^T$  : vetor de tensões nodais do sistema;

$\hat{F}_t = [F_1, F_2, \dots, F_m]^T$  : vetor de funções algébricas a diferenças não-lineares;

$\hat{G}_t = [G_1, G_2, \dots, G_n]^T$  : vetor de funções algébricas não-lineares;

$\hat{y}_t = [\hat{x}_t, \hat{V}_t^e]^T$  : vetor de variáveis do sistema;

$\hat{H}_t = [\hat{F}_t, \hat{G}_t]^T$  : vetor de funções algébricas não-lineares.

**Para todos os  $p$  passos de integração tomados simultaneamente**

$\hat{x} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_p]^T$  : vetor de variáveis de estado;

$\hat{V}^e = [\hat{V}_1^e, \hat{V}_2^e, \dots, \hat{V}_p^e]^T$  : vetor de tensões nodais;

$\hat{F} = [\hat{F}_1, \hat{F}_2, \dots, \hat{F}_p]^T$  : vetor de funções algébricas de diferenças não-lineares;

$\hat{G} = [\hat{G}_1, \hat{G}_2, \dots, \hat{G}_p]^T$  : vetor de funções algébricas não-lineares;

$\hat{y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_p]^T$  : vetor de variáveis do sistema;

$\hat{H} = [\hat{H}_1, \hat{H}_2, \dots, \hat{H}_p]^T$  : vetor de funções algébricas não lineares.

Assim, a generalização do sistema de equações (C.32) para uma janela de integração, pode ser representada na seguinte forma sintética:

$$\hat{H}(\hat{y}) = 0 \quad (\text{C.33})$$

Consequentemente, o sistema de equações (C.31) para  $p$  passos de integração, assumirá a seguinte forma:

$$\begin{cases} \hat{F}(\hat{x}, \hat{V}^e) = 0 \\ \hat{G}(\hat{x}, \hat{V}^e) = 0 \end{cases} \quad (\text{C.34})$$

ou, mais explicitamente, tal como a equação (C.35)

$$\begin{cases} \hat{F}_t(\hat{x}_t, \hat{x}_{t-1}, \hat{V}_t^e, \hat{V}_{t-1}^e) = 0 \\ \hat{G}_t(\hat{x}_t, \hat{V}_t^e) = 0 \quad t = 1, 2, \dots, p \end{cases} \quad (\text{C.35})$$

## C.3 Resolução por Newton-Raphson

### C.3.1 Equação Linear Jacobiana

A equação linear jacobiana para o sistema de equações (C.35), possui a seguinte estrutura genérica:

$$\begin{array}{c} \hat{F}_1 \\ \hat{G}_1 \\ \hat{F}_2 \\ \hat{G}_2 \\ \vdots \\ \vdots \\ \hat{F}_p \\ \hat{G}_p \end{array} = - \begin{array}{|c|c|c|c|} \hline \frac{\partial \hat{F}_1}{\partial \hat{x}_1} & \frac{\partial \hat{F}_1}{\partial \hat{V}_1^e} & & \\ \hline \frac{\partial \hat{G}_1}{\partial \hat{x}_1} & Y^e - \frac{\partial \hat{G}'_1}{\partial \hat{V}_1^e} & & \\ \hline \frac{\partial \hat{F}_2}{\partial \hat{x}_1} & \frac{\partial \hat{F}_2}{\partial \hat{V}_1^e} & \frac{\partial \hat{F}_2}{\partial \hat{x}_2} & \frac{\partial \hat{F}_2}{\partial \hat{V}_2^e} \\ \hline 0 & 0 & \frac{\partial \hat{G}_2}{\partial \hat{x}_2} & Y^e - \frac{\partial \hat{G}'_2}{\partial \hat{V}_2^e} \\ \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \frac{\partial \hat{F}_p}{\partial \hat{x}_{p-1}} & \frac{\partial \hat{F}_p}{\partial \hat{V}_{p-1}^e} & \frac{\partial \hat{F}_p}{\partial \hat{x}_p} & \frac{\partial \hat{F}_p}{\partial \hat{V}_p^e} \\ \hline 0 & 0 & \frac{\partial \hat{G}_p}{\partial \hat{x}_p} & Y^e - \frac{\partial \hat{G}'_p}{\partial \hat{V}_p^e} \\ \hline \end{array} \cdot \begin{array}{c} \Delta \hat{x}_1 \\ \Delta \hat{V}_1^e \\ \Delta \hat{x}_2 \\ \Delta \hat{V}_2^e \\ \vdots \\ \vdots \\ \Delta \hat{x}_p \\ \Delta \hat{V}_p^e \end{array} \quad (C.36)$$

onde:

$$\frac{\partial \hat{F}_t}{\partial \hat{x}_t} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & & & \\ & \frac{\partial F_2}{\partial x_2} & & \\ & & \dots & \\ & & & \frac{\partial F_m}{\partial x_m} \end{bmatrix} \quad (C.37)$$

$$\frac{\partial \hat{F}_t}{\partial \hat{V}_t^e} = \left[ \frac{\partial F_1}{\partial V^e} \quad \frac{\partial F_2}{\partial V^e} \quad \dots \quad \frac{\partial F_m}{\partial V^e} \right]^T \quad (C.38)$$

$$\frac{\partial \hat{G}_t}{\partial \hat{x}_t} = \left[ -\frac{\partial G'}{\partial x_1} \quad -\frac{\partial G'}{\partial x_2} \quad \dots \quad -\frac{\partial G'}{\partial x_m} \right] \quad (\text{C.39})$$

$$\frac{\partial \hat{F}_t}{\partial \hat{x}_{t-1}} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & & & \\ & \frac{\partial F_2}{\partial x_2} & & \\ & & \ddots & \\ & & & \frac{\partial F_m}{\partial x_m} \end{bmatrix} \quad (\text{C.40})$$

$$\frac{\partial \hat{F}_t}{\partial \hat{V}_{t-1}^e} = \left[ \frac{\partial F_1}{\partial V^e} \quad \frac{\partial F_2}{\partial V^e} \quad \dots \quad \frac{\partial F_m}{\partial V^e} \right]^T \quad (\text{C.41})$$

$$Y^e = \begin{array}{|c|} \hline \begin{array}{|c|c|c|c|c|c|} \hline -b_{11} & g_{11} & -b_{12} & g_{12} & \dots & \dots & -b_{1n} & g_{1n} \\ \hline g_{11} & b_{11} & g_{12} & b_{12} & \dots & \dots & g_{1n} & b_{1n} \\ \hline -b_{21} & g_{21} & -b_{22} & g_{22} & \dots & \dots & -b_{2n} & g_{2n} \\ \hline g_{21} & b_{21} & g_{22} & b_{22} & \dots & \dots & g_{2n} & b_{2n} \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \hline -b_{n1} & g_{n1} & -b_{n2} & g_{n2} & \dots & \dots & -b_{nn} & g_{nn} \\ \hline g_{n1} & b_{n1} & g_{n2} & b_{n2} & \dots & \dots & g_{nn} & b_{nn} \\ \hline \end{array} \\ \hline \end{array} \quad (\text{C.42})$$

$$\frac{\partial \hat{G}'_t}{\partial \hat{V}_t^e} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline \frac{\partial q_{11}}{\partial V_{m1}} & \frac{\partial q_{11}}{\partial V_{r1}} & & & & & & & & \\ \frac{\partial q_{21}}{\partial V_{m1}} & \frac{\partial q_{21}}{\partial V_{r1}} & & & & & & & & \\ \hline & & \ddots & \ddots & & & & & & \\ \hline & & \ddots & \ddots & & & & & & \\ \hline & & & & \frac{\partial q_{1i}}{\partial V_{mi}} & \frac{\partial q_{1i}}{\partial V_{ri}} & & & & \\ \frac{\partial q_{2i}}{\partial V_{mi}} & \frac{\partial q_{2i}}{\partial V_{ri}} & & & & & & & & \\ \hline & & & & & & \ddots & \ddots & & \\ \hline & & & & & & \ddots & \ddots & & \\ \hline & & & & & & & & \frac{\partial q_{1n}}{\partial V_{mn}} & \frac{\partial q_{1n}}{\partial V_{rn}} \\ \frac{\partial q_{2n}}{\partial V_{mn}} & \frac{\partial q_{2n}}{\partial V_{rn}} & & & & & & & & \\ \hline \end{array} \quad (C.43)$$

### C.3.2 Estrutura das Sub-matrizes da Matriz Jacobiana

Neste item são apresentadas a estrutura genérica de cada uma das sub-matrizes que aparecem nas equações (C.36) a (C.43), bem como as expressões que definem os seus elementos.

$$\frac{\partial F_{i(t)}}{\partial x_{i(t)}} = \begin{array}{|c|c|c|c|c|c|c|} \hline & & V_e & E_{fd} & E'_d & E'_q & \delta & \omega \\ \hline \hline f_1 & \frac{\partial f_1}{\partial V_e} & & & & & & \\ f_2 & \frac{\partial f_2}{\partial V_e} & \frac{\partial f_2}{\partial E_{fd}} & & & & & \\ f_3 & & & \frac{\partial f_3}{\partial E'_d} & & \frac{\partial f_3}{\partial \delta} & & \\ f_4 & & \frac{\partial f_4}{\partial E_{fd}} & & \frac{\partial f_4}{\partial E'_q} & \frac{\partial f_4}{\partial \delta} & & \\ f_5 & & & \frac{\partial f_5}{\partial E'_d} & \frac{\partial f_5}{\partial E'_q} & \frac{\partial f_5}{\partial \delta} & \frac{\partial f_5}{\partial \omega} & \\ f_6 & & & & & \frac{\partial f_6}{\partial \delta} & \frac{\partial f_6}{\partial \omega} & \\ \hline \end{array} \quad (C.44)$$

onde:

$$\begin{aligned}
\frac{\partial f_1}{\partial V_e} &= 1 \\
\frac{\partial f_2}{\partial V_e} &= -a_A K_A \\
\frac{\partial f_2}{\partial E_{fd}} &= 1 \\
\frac{\partial f_3}{\partial E'_d} &= 1 + a_q Q_x \\
\frac{\partial f_3}{\partial \delta} &= a_q Q_x [V_r \cos \delta + V_m \sin \delta] \\
\frac{\partial f_4}{\partial E_{fd}} &= -a_d \\
\frac{\partial f_4}{\partial E'_q} &= 1 + a_d D_x \\
\frac{\partial f_4}{\partial \delta} &= a_d D_x [V_r \sin \delta - V_m \cos \delta] \\
\frac{\partial f_5}{\partial E'_d} &= \frac{\omega_s \Delta t}{4Hx'_q} [V_r \cos \delta + V_m \sin \delta] \\
\frac{\partial f_5}{\partial E'_q} &= \frac{\omega_s \Delta t}{4Hx'_d} [V_r \sin \delta - V_m \cos \delta] \\
\frac{\partial f_5}{\partial \delta} &= \frac{\omega_s \Delta t}{4H} \left\{ -\frac{E'_d}{x'_q} [V_r \sin \delta - V_m \cos \delta] + \frac{E'_q}{x'_d} [V_r \cos \delta + V_m \sin \delta] - \right. \\
&\quad \left. \left( \frac{x'_q - x'_d}{x'_q x'_d} \right) [(V_r^2 - V_m^2) \cos 2\delta + 2V_r V_m \sin 2\delta] \right\} \\
\frac{\partial f_6}{\partial \omega} &= 1 \\
\frac{\partial f_6}{\partial \delta} &= 1 \\
\frac{\partial f_6}{\partial \omega} &= -\frac{\Delta t}{2}
\end{aligned} \tag{C.45}$$

	$V_{m1}$	$V_{r1}$	...	...	$V_{mi}$	$V_{ri}$	$V_{mi+1}$	$V_{ri+1}$	...	...	$V_{mn}$	$V_{rn}$
--	----------	----------	-----	-----	----------	----------	------------	------------	-----	-----	----------	----------

$\frac{\partial F_{i(t)}}{\partial V^e(t)} =$	$f_1$	0	0	...	...	$\frac{\partial f_1}{\partial V_{mi}}$	$\frac{\partial f_1}{\partial V_{ri}}$	0	0	...	...	0	0
	$f_2$	0	0	...	...	0	0	0	0	...	...	0	0
	$f_3$	0	0	...	...	$\frac{\partial f_3}{\partial V_{mi}}$	$\frac{\partial f_3}{\partial V_{ri}}$	0	0	...	...	0	0
	$f_4$	0	0	...	...	$\frac{\partial f_4}{\partial V_{mi}}$	$\frac{\partial f_4}{\partial V_{ri}}$	0	0	...	...	0	0
	$f_5$	0	0	...	...	$\frac{\partial f_5}{\partial V_{mi}}$	$\frac{\partial f_5}{\partial V_{ri}}$	0	0	...	...	0	0
	$f_6$	0	0	...	...	0	0	0	0	...	...	0	0

(C.46)

onde:

$$\frac{\partial f_1}{\partial V_r} = a_c V_r (V_r^2 + V_m^2)^{-1/2}$$

$$\frac{\partial f_1}{\partial V_m} = a_c V_m (V_r^2 + V_m^2)^{-1/2}$$

$$\frac{\partial f_3}{\partial V_r} = a_q Q_x \sin \delta$$

$$\frac{\partial f_3}{\partial V_m} = -a_q Q_x \cos \delta$$

$$\frac{\partial f_4}{\partial V_r} = -a_d D_x \cos \delta$$

$$\frac{\partial f_4}{\partial V_m} = -a_d D_x \sin \delta$$

$$\frac{\partial f_5}{\partial V_r} = \frac{\omega_s \Delta t}{4H} \left\{ \frac{E'_d}{x'_q} \cos \delta + \frac{E'_q}{x'_d} \sin \delta - \left( \frac{x'_q - x'_d}{x'_d x'_q} \right) [V_r \sin 2\delta - V_m \cos 2\delta] \right\}$$

$$\frac{\partial f_5}{\partial V_m} = \frac{\omega_s \Delta t}{4H} \left\{ \frac{E'_d}{x'_q} \sin \delta - \frac{E'_q}{x'_d} \cos \delta + \left( \frac{x'_q - x'_d}{x'_d x'_q} \right) [V_m \sin 2\delta + V_r \cos 2\delta] \right\}$$

(C.47)



		$V_e$	$E_{fd}$	$E'_d$	$E'_q$	$\delta$	$\omega$
--	--	-------	----------	--------	--------	----------	----------

$$\frac{\partial F_{i(t)}}{\partial x_{i(t-1)}} = \begin{array}{c|cccccc} f_1 & \frac{\partial f_1}{\partial V_e} & & & & & & \\ f_2 & \frac{\partial f_2}{\partial V_e} & \frac{\partial f_2}{\partial E_{fd}} & & & & & \\ f_3 & & & \frac{\partial f_3}{\partial E'_d} & & \frac{\partial f_3}{\partial \delta} & & \\ f_4 & & \frac{\partial f_4}{\partial E_{fd}} & & \frac{\partial f_4}{\partial E'_q} & \frac{\partial f_4}{\partial \delta} & & \\ f_5 & & & \frac{\partial f_5}{\partial E'_d} & \frac{\partial f_5}{\partial E'_q} & \frac{\partial f_5}{\partial \delta} & \frac{\partial f_5}{\partial \omega} & \\ f_6 & & & & & \frac{\partial f_6}{\partial \delta} & \frac{\partial f_6}{\partial \omega} & \end{array} \quad (C.48)$$

onde:

$$\begin{aligned} \frac{\partial f_1}{\partial V_e} &= -1 + 2a_B \\ \frac{\partial f_2}{\partial V_e} &= -a_A K_A \\ \frac{\partial f_2}{\partial E_{fd}} &= -1 + 2a_A \\ \frac{\partial f_3}{\partial E'_d} &= -(1 - 2a_q - a_q Q_x) \\ \frac{\partial f_3}{\partial \delta} &= a_q Q_x [V_r \cos \delta + V_m \sin \delta] \\ \frac{\partial f_4}{\partial E_{fd}} &= -a_d \\ \frac{\partial f_4}{\partial E'_q} &= -(1 - 2a_d - a_d D_x) \\ \frac{\partial f_4}{\partial \delta} &= a_d D_x [V_r \sin \delta - V_m \cos \delta] \\ \frac{\partial f_5}{\partial E'_d} &= \frac{\omega_s \Delta t}{4H x'_q} [V_r \cos \delta + V_m \sin \delta] \\ \frac{\partial f_5}{\partial E'_q} &= \frac{\omega_s \Delta t}{4H x'_d} [V_r \sin \delta - V_m \cos \delta] \\ \frac{\partial f_5}{\partial \delta} &= \frac{\omega_s \Delta t}{4H} \left\{ -\frac{E'_d}{x'_q} [V_r \sin \delta - V_m \cos \delta] + \frac{E'_q}{x'_d} [V_r \cos \delta + V_m \sin \delta] - \right. \\ &\quad \left. \left( \frac{x'_q - x'_d}{x'_q x'_d} \right) [(V_r^2 - V_m^2) \cos 2\delta + 2V_r V_m \sin 2\delta] \right\} \\ \frac{\partial f_5}{\partial \omega} &= -1 \\ \frac{\partial f_6}{\partial \delta} &= -1 \\ \frac{\partial f_6}{\partial \omega} &= -\frac{\Delta t}{2} \end{aligned} \quad (C.49)$$

		$V_{m1}$	$V_{r1}$	...	...	$V_{mi}$	$V_{ri}$	$V_{mi+1}$	$V_{ri+1}$	...	...	$V_{mn}$	$V_{rn}$
--	--	----------	----------	-----	-----	----------	----------	------------	------------	-----	-----	----------	----------

$\frac{\partial F_{i(t)}}{\partial V_{(t-1)}^e} =$	$f_1$	0	0	...	...	$\frac{\partial f_1}{\partial V_{m_i}}$	$\frac{\partial f_1}{\partial V_{r_i}}$	0	0	...	...	0	0
	$f_2$	0	0	...	...	0	0	0	0	...	...	0	0
	$f_3$	0	0	...	...	$\frac{\partial f_3}{\partial V_{m_i}}$	$\frac{\partial f_3}{\partial V_{r_i}}$	0	0	...	...	0	0
	$f_4$	0	0	...	...	$\frac{\partial f_4}{\partial V_{m_i}}$	$\frac{\partial f_4}{\partial V_{r_i}}$	0	0	...	...	0	0
	$f_5$	0	0	...	...	$\frac{\partial f_5}{\partial V_{m_i}}$	$\frac{\partial f_5}{\partial V_{r_i}}$	0	0	...	...	0	0
	$f_6$	0	0	...	...	0	0	0	0	...	...	0	0

(C.50)

onde:

$$\frac{\partial f_1}{\partial V_r} = -b_C V_r (V_r^2 + V_m^2)^{-1/2}$$

$$\frac{\partial f_1}{\partial V_m} = -b_C V_m (V_r^2 + V_m^2)^{-1/2}$$

$$\frac{\partial f_3}{\partial V_r} = a_q Q_x \sin \delta$$

$$\frac{\partial f_3}{\partial V_m} = -a_q Q_x \cos \delta$$

$$\frac{\partial f_4}{\partial V_r} = -a_d D_x \cos \delta$$

$$\frac{\partial f_4}{\partial V_m} = -a_d D_x \sin \delta$$

$$\frac{\partial f_5}{\partial V_r} = \frac{\omega_s \Delta t}{4H} \left\{ \frac{E'_d}{x'_q} \cos \delta + \frac{E'_q}{x'_d} \sin \delta - \left( \frac{x'_q - x'_d}{x'_d x'_q} \right) [V_r \sin 2\delta - V_m \cos 2\delta] \right\}$$

$$\frac{\partial f_5}{\partial V_m} = \frac{\omega_s \Delta t}{4H} \left\{ \frac{E'_d}{x'_q} \sin \delta - \frac{E'_q}{x'_d} \cos \delta + \left( \frac{x'_q - x'_d}{x'_d x'_q} \right) [V_m \sin 2\delta + V_r \cos 2\delta] \right\}$$

(C.51)

	$V_c$	$E_{fd}$	$E'_d$	$E'_q$	$\delta$	$\omega$
--	-------	----------	--------	--------	----------	----------

$$\frac{\partial G(t)}{\partial x_{i(t)}} = \begin{matrix} g_{11} \\ g_{21} \\ \vdots \\ \vdots \\ g_{1i} \\ g_{2i} \\ \vdots \\ \vdots \\ g_{1n} \\ g_{2n} \end{matrix} = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \frac{\partial g_{11}}{\partial E'_d} & \frac{\partial g_{11}}{\partial E'_q} & \frac{\partial g_{11}}{\partial \delta} & 0 \\ 0 & 0 & \frac{\partial g_{21}}{\partial E'_d} & \frac{\partial g_{21}}{\partial E'_q} & \frac{\partial g_{21}}{\partial \delta} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \quad (C.52)$$

onde:

$$\begin{aligned} \frac{\partial g_1}{\partial E'_d} &= \frac{1}{x'_q} \cos \delta \\ \frac{\partial g_1}{\partial E'_q} &= \frac{1}{x'_d} \sin \delta \\ \frac{\partial g_2}{\partial E'_d} &= \frac{1}{x'_q} \sin \delta \\ \frac{\partial g_2}{\partial E'_q} &= -\frac{1}{x'_d} \cos \delta \end{aligned} \quad (C.53)$$

$$\frac{\partial g_1}{\partial \delta} = -\frac{E'_d}{x'_q} \sin \delta + \frac{E'_q}{x'_d} \cos \delta - \left( \frac{x'_q - x'_d}{x'_q x'_d} \right) [V_r \cos 2\delta + V_m \sin 2\delta]$$

$$\frac{\partial g_2}{\partial \delta} = \frac{E'_d}{x'_q} \cos \delta + \frac{E'_q}{x'_d} \sin \delta + \left( \frac{x'_q - x'_d}{x'_q x'_d} \right) [V_m \cos 2\delta - V_r \sin 2\delta]$$

$$\frac{\partial \hat{G}'_t}{\partial \hat{V}'_t} =$$

		$V_{m1}$	$V_{r1}$	...	...	$V_{mi}$	$V_{ri}$	...	...	$V_{mn}$	$V_{rn}$
$g_{11}$	$\frac{\partial g_{11}}{\partial V_{m1}}$	$\frac{\partial g_{11}}{\partial V_{r1}}$									
$g_{21}$	$\frac{\partial g_{21}}{\partial V_{m1}}$	$\frac{\partial g_{21}}{\partial V_{r1}}$									
$\vdots$			$\ddots$	$\ddots$							
$\vdots$			$\ddots$	$\ddots$							
$g_{1i}$					$\frac{\partial g_{1i}}{\partial V_{mi}}$	$\frac{\partial g_{1i}}{\partial V_{ri}}$					
$g_{2i}$					$\frac{\partial g_{2i}}{\partial V_{mi}}$	$\frac{\partial g_{2i}}{\partial V_{ri}}$					
$\vdots$							$\ddots$	$\ddots$			
$\vdots$							$\ddots$	$\ddots$			
$g_{1n}$									$\frac{\partial g_{1n}}{\partial V_{mn}}$	$\frac{\partial g_{1n}}{\partial V_{rn}}$	
$g_{2n}$									$\frac{\partial g_{2n}}{\partial V_{mn}}$	$\frac{\partial g_{2n}}{\partial V_{rn}}$	

(C.54)

onde:

*Para barras de geração*

$$\begin{aligned} \frac{\partial g_1}{\partial V_r} &= - \left( \frac{x'_q - x'_d}{2x'_q x'_q} \right) \text{sen } 2\delta \\ \frac{\partial g_1}{\partial V_m} &= \left( \frac{x'_q - x'_d}{2x'_q x'_q} \right) \cos 2\delta \\ \frac{\partial g_2}{\partial V_r} &= \left( \frac{x'_q - x'_d}{2x'_q x'_q} \right) \cos 2\delta \\ \frac{\partial g_2}{\partial V_m} &= \left( \frac{x'_q - x'_d}{2x'_q x'_q} \right) \text{sen } 2\delta \end{aligned}$$
(C.55)

*Para barras de carga*

$$\begin{aligned} \frac{\partial g_1}{\partial V_r} &= g_0 + a_1 |V_0|^2 g_0 \left[ \frac{V_r^2 - V_m^2}{(V_r^2 + V_m^2)^2} \right] - c_1 g_0 \\ \frac{\partial g_1}{\partial V_m} &= -b_0 + a_2 |V_0|^2 b_0 \left[ \frac{V_r^2 - V_m^2}{(V_r^2 + V_m^2)^2} \right] + c_2 b_0 \\ \frac{\partial g_2}{\partial V_r} &= b_0 + a_2 |V_0|^2 b_0 \left[ \frac{V_r^2 - V_m^2}{(V_r^2 + V_m^2)^2} \right] - c_2 b_0 \\ \frac{\partial g_2}{\partial V_m} &= g_0 - a_1 |V_0|^2 g_0 \left[ \frac{V_r^2 - V_m^2}{(V_r^2 + V_m^2)^2} \right] - c_1 g_0 \end{aligned}$$
(C.56)

# Apêndice D

## O Método Gradiente Conjugado

Neste apêndice descrevem-se os elementos básicos do método iterativo Gradiente Conjugado (GC) pré-condicionado [36, 5], empregado no âmbito dos algoritmos desenvolvidos no presente trabalho.

### D.1 O Método Gradiente Conjugado

#### D.1.1 Conceitos Básicos

Para uma matriz  $\hat{Y}_s(n, n)$ , simétrica e definida positiva, a solução do sistema linear

$$\hat{Y}_s V_s = \hat{I}_s \quad (\text{D.1})$$

é equivalente ao seguinte problema de otimização sem restrições:

$$\min Q(V_s) = \frac{1}{2} V_s^T \hat{Y}_s V_s - \hat{I}_s^T V_s \quad (\text{D.2})$$

A função  $Q(V_s)$  tem um mínimo global onde o seu gradiente

$$\nabla Q(V_s) = \hat{Y}_s V_s - \hat{I}_s \quad (\text{D.3})$$

que é a solução de

$$\hat{Y}_s V_s = \hat{I}_s \quad (\text{D.4})$$

anula-se.

A forma geral dos métodos iterativos de minimização é dada por [36]:

$$V_s^{k+1} = V_s^k - \alpha^k d^k \quad (\text{D.5})$$

$$k = 0, 1, 2, \dots \quad (\text{D.6})$$

onde:

$d^k$  direção vetorial de minimização;

$\alpha^k$  escalar que define a distância de movimento na direção  $d^k$

A forma de escolha de  $\alpha^k$  e  $d^k$  define uma grande variedade de métodos (Ortega, 1988), entre eles o método gradiente conjugado, objeto deste apêndice.

Desse modo o problema de minimizar a função  $Q(V_s)$  é transformado em sucessivos problemas do tipo minimizar  $Q(V_s)$  na direção  $d^k$  de modo que  $\alpha^k$  satisfaça a seguinte equação:

$$Q(V_s^k - \alpha^k d^k) = \min_{\alpha} Q(V_s^k - \alpha d^k) \quad (\text{D.7})$$

Para  $V_s^k$  e  $d^k$  especificados e suprimindo o sobrescrito  $k$ , pode-se escrever a seguinte função em  $\alpha$ :

$$q(\alpha) = Q(V_s - \alpha d) = \frac{1}{2}(V_s - \alpha d)^T \hat{Y}_s (V_s - \alpha d) - \hat{I}_s^T (V_s - \alpha d) \quad (\text{D.8})$$

ou

$$q(\alpha) = \frac{1}{2} d^T \hat{Y}_s d \alpha^2 - d^T (\hat{Y}_s V_s - \hat{I}_s) \alpha + \frac{1}{2} V_s^T (\hat{Y}_s V_s - 2 \hat{I}_s) \quad (\text{D.9})$$

A função quadrática  $q(\alpha)$  é minimizada quando sua primeira derivada for nula, ou seja:

$$\frac{dq(\alpha)}{d\alpha} = 0 = d^T \hat{Y}_s d \alpha - d^T (\hat{Y}_s V_s - \hat{I}_s) \quad (\text{D.10})$$

Resolvendo-se a equação (D.10) para  $\alpha$ , resulta:

$$\alpha = \frac{d^T(\hat{Y}_s V_s - \hat{I}_s)}{d^T \hat{Y}_s d} \quad (\text{D.11})$$

No método gradiente conjugado as direções de minimização unidirecionais são geradas a partir da direção do gradiente, sucedidas por direções conjugadas de modo que satisfaçam a seguinte equação [36]:

$$(d^i)^T \hat{Y}_s d^j = 0 \text{ para } i \neq j \quad (\text{D.12})$$

A direção do gradiente é obtida da seguinte forma:

Tome a função objetivo

$$Q(V_s) = \frac{1}{2} V_s^T \hat{Y}_s V_s - \hat{I}_s^T V_s \quad (\text{D.13})$$

e defina  $d^k$  tal que:

$$d^k = -\nabla Q(V_s^k) = -(\hat{Y}_s V_s^k - \hat{I}_s) \quad (\text{D.14})$$

Então, o algoritmo gradiente conjugado quando aplicado na resolução de um sistema algébrico linear, como o da equação (D.1), segue os seguintes passos.

### Algoritmo GC

1. *Especifique a solução inicial  $V_s^0$ ;*
2. *Calcule a direção negativa do gradiente*

$$r^0 = \hat{I}_s - \hat{Y}_s V_s^0$$
3. *Faça  $d^0 = r^0$  e inicialize o contador de iterações  $k \leftarrow 0$ ;*
4. *Calcule:*

$$\alpha^k = -\frac{[r^k]^T \cdot [r^k]}{[d^k]^T \cdot [\hat{Y}_s d^k]}$$

$$V_s^{k+1} = V_s^k - \alpha^k d^k$$

$$r^{k+1} = r^k + \alpha^k \hat{Y}_s d^k$$

5. Teste: Se  $\|r^{k+1}\|_2^2 \leq \epsilon$  PARE. Se não vá para 6;

6. Calcule a direção conjugada  $d^{k+1}$ , usando as seguintes equações:

$$\beta^k = \frac{[r^{k+1}]^T \cdot [r^{k+1}]}{[r^k]^T \cdot [r^k]}$$

$$d^{k+1} = r^{k+1} + \beta^k d^k$$

7. Incremente o contador de iterações  $k \leftarrow k + 1$  e vá ao passo 4.

### D.1.2 Pré-condicionamento

A razão de convergência do método GC é função do número de condição da matriz de coeficientes  $\hat{Y}_s$  ( $cond(\hat{Y}_s) = \|\hat{Y}_s\|_2 \|\hat{Y}_s^{-1}\|_2$ ). Contudo, esta razão pode ser melhorada se o número condição de  $\hat{Y}_s$  for alterado no sentido de aproximá-lo de 1 [36]. Isto pode ser feito pré-condicionado  $\bar{Y}_s$  pela seguinte transformação de congruência:

$$\bar{Y}_s = S \hat{Y}_s S^T \quad (D.15)$$

onde  $S$  é escolhida tal que  $cond(\bar{Y}_s) \leq cond(\hat{Y}_s)$

No algoritmo GC, o uso de pré-condicionamento é representado através de modificações no vetor de resíduos  $r$ , em determinados passos do algoritmo, obtidos resolvendo-se o seguinte sistema linear algébrico auxiliar:

$$M \tilde{r} = r \quad (D.16)$$

onde  $M = (S^T S)^{-1}$  e  $\tilde{r}$  é o vetor de resíduos modificado.

A equação (D.16) deve ser resolvida a cada iteração e, então, é importante que  $S$ , ou equivalentemente  $M$ , sejam escolhidos de modo que seja fácil resolvê-la.

Da equação (D.15) e da definição da matriz  $M = (S^T S)^{-1}$ , pode-se escrever o seguinte [36]:

$$S^T \bar{Y}_s S^{-T} = S^T S \hat{Y}_s = M^{-1} \hat{Y}_s \quad (D.17)$$

Note que a matriz  $M^{-1} \hat{Y}_s$  é similar a  $\bar{Y}_s$  e assim,

$$cond(\bar{Y}_s) = \lambda_{max}(M^{-1} \hat{Y}_s) / \lambda_{min}(M^{-1} \hat{Y}_s) \quad (D.18)$$



Então, o ideal é ter  $M = \hat{Y}_s$  ( $cond(\bar{Y}_s) = 1$ ), mas o sistema auxiliar seria tão difícil de resolver quanto o sistema original.

Portanto, a escolha de  $M$  deve ser uma solução de compromisso tal que  $M$  seja o mais próximo possível de  $\hat{Y}_s$  ( $cond(M^{-1}\hat{Y}_s) < cond(\hat{Y}_s)$ ), porém, de modo que o sistema auxiliar  $M\tilde{r} = r$  tenha resolução relativamente simples.

A seguir apresenta-se a versão básica do algoritmo Gradiente Conjugado pré-condicionado aplicado a equação (D.1).

### Algoritmo GC pré-condicionado

1. *Especifique a solução inicial  $V_s^0$ ;*
2. *Calcule a direção negativa do gradiente*

$$r^0 = \hat{I}_s - \hat{Y}_s V_s^0$$

*e resolva o sistema linear*

$$M\tilde{r}^0 = r^0$$

3. *Faça  $d^0 = \tilde{r}^0$  e inicialize o contador de iterações  $k \leftarrow 0$ ;*
4. *Calcule:*

$$\alpha^k = -\frac{[\tilde{r}^k]^T \cdot [r^k]}{[d^k]^T \cdot [\hat{Y}_s d^k]}$$

$$V_s^{k+1} = V_s^k - \alpha^k d^k$$

$$r^{k+1} = r^k + \alpha^k \hat{Y}_s d^k$$

*e resolva*

$$M\tilde{r}^{k+1} = r^{k+1}$$

5. *Teste: Se  $\|r^{k+1}\|_2^2 \leq \epsilon$  PARE. Se não vá para 6;*

6. *Calcule a direção conjugada  $d^{k+1}$ , usando as seguintes equações:*

$$\beta^k = \frac{[\tilde{r}^{k+1}]^T \cdot [r^{k+1}]}{[\tilde{r}^k]^T \cdot [r^k]}$$

$$d^{k+1} = \tilde{r}^{k+1} + \beta^k d^k$$

7. *Incremente o contador de iterações  $k \leftarrow k + 1$  e vá ao passo 4.*

Note que, neste caso, o produto  $[\tilde{r}^{k+1}]^T [r^{k+1}]$ , utilizado nos passos 4 e 6, é diferente do produto  $[r^{k+1}]^T [r^{k+1}]$ , empregado no teste de convergência e, portanto, o algoritmo GC necessita de mais um produto de vetores que o algoritmo GC. Ortega [36] sugere que se verifique, inicialmente, as condições de convergência através do produto  $[\tilde{r}^{k+1}]^T [r^{k+1}]$ , e somente quando este atendê-las é que se deve calcular e utilizar  $[r^{k+1}]^T [r^{k+1}]$ .

# Apêndice E

## Solução Inicial Aproximada por Série de Taylor

A ideia é se obter um algoritmo simples e com elevado grau de paralelismo para gerar uma solução aproximada, no intervalo de tempo de uma janela de integração, que sirva como uma efetiva condição inicial para os algoritmos que exploram o paralelismo no tempo. A primeira tentativa de se encontrar um algoritmo com tais características, será feita com a utilização de um modelo simplificado para o SEE, sendo a sua solução obtida pelo método da série de Taylor. Ribbens-Pavella [58] e Decker [59] empregaram, com sucesso, o método da série de Taylor no contexto dos métodos diretos de análise da estabilidade transitória de SEE.

### E.1 Modelo Matemático

Desprezando-se os efeitos dos controladores e representando-se as máquinas síncronas por um modelo de segunda ordem, o SEE pode ser descrito pelas seguintes equações:

$$\left\{ \begin{array}{l} \frac{d\omega_i}{dt} = \frac{P_{mi} - P_{ei}}{M_i} \\ \frac{d\delta_i}{dt} = \omega_i \\ I = YV \\ i = 1, 2, \dots, m \end{array} \right. \quad (\text{E.1})$$

onde:

$$\begin{cases} M_i = 2H_i/\omega_s \\ P_{ei} = \frac{E'_g}{x'_d} (V_{ri} \sin \delta_i - V_{mi} \cos \delta_i) \end{cases} \quad (\text{E.2})$$

e  $V_{ri}$ ,  $V_{mi}$  : são, respectivamente, as parcelas real e imaginária, da tensão na  $i$ -ésima barra, e  $\delta_i$  é o ângulo atrás da reatância transitória.

## E.2 Aplicação da Expansão em Série de Taylor

A solução do sistema de equações (E.1) para  $\delta_i(t)$  e  $\omega_i(t)$ ,  $i = 1, 2, \dots, m$ , pode ser obtida através de uma expansão em série de Taylor, realizada para o instante de tempo  $t = t_0$ , conforme descrito a seguir:

$$\delta_i(t) = \delta_i(t_0) + \delta_i^{(1)}(t_0) \frac{(t-t_0)}{1!} + \delta_i^{(2)}(t_0) \frac{(t-t_0)^2}{2!} + \dots \quad (\text{E.3})$$

$$\omega_i(t) = \omega_i(t_0) + \omega_i^{(1)}(t_0) \frac{(t-t_0)}{1!} + \omega_i^{(2)}(t_0) \frac{(t-t_0)^2}{2!} + \dots \quad (\text{E.4})$$

ou

$$\omega_i(t) = \delta_i^{(1)}(t_0) + \delta_i^{(2)}(t_0) \frac{(t-t_0)}{1!} + \delta_i^{(3)}(t_0) \frac{(t-t_0)^2}{2!} + \dots \quad (\text{E.5})$$

As soluções de (E.3) e (E.4) exigem a determinação de derivadas de alta ordem para  $\delta_i$  e  $\omega_i$ ,  $i = 1, 2, \dots, m$ , para o instante de tempo  $t = t_0$

## E.3 Determinação das Derivadas de Alta Ordem para $\delta_i$ e $\omega_i$

Os valores das  $\nu$ -ésimas derivadas de  $\delta_i(t_0)$  e  $\omega_i(t_0)$ ,  $i = 1, 2, \dots, m$ , são dadas por:

$$\begin{cases} \delta_i^{(1)}(t_0) = \omega_i(t_0) \\ \delta_i^{(2)}(t_0) = \omega_i^{(1)}(t_0) = \frac{P_{m_i} - P_{ei}(t_0)}{M_i} \\ \delta_i^{(3)}(t_0) = \omega_i^{(2)}(t_0) = \frac{-P_{ei}^{(1)}(t_0)}{M_i} \\ \vdots \\ \delta_i^{(\nu)}(t_0) = \omega_i^{(\nu-1)}(t_0) = \frac{-P_{ei}^{(\nu-2)}(t_0)}{M_i} \end{cases} \quad (\text{E.6})$$

Observa-se nas equações (E.3)-(E.5) e (E.6), que o cálculo dos coeficientes da série exige o cálculo de derivadas de alta ordem de  $P_{ei}(t)$ ,  $i = 1, 2, \dots, m$ , até duas ordens abaixo da derivada de mais alta ordem de  $\delta_i(t)$ .

## E.4 Potência Elétrica e suas Derivadas

As equações para o cálculo de  $P_{ei}(t)$ ,  $i = 1, 2, \dots, m$ , e suas derivadas, até a segunda ordem, no instante de tempo  $t = t_0$ , são apresentadas a seguir. Definindo-se  $K_i = \frac{E'_{gi}}{x'_{di}}$ , em (E.2), resulta:

$$\left\{ \begin{array}{l} P_{ei} = K_i [V_{ri} \sin \delta_i - V_{mi} \cos \delta_i] \\ P_{ei}^{(1)} = K_i \sin \delta_i [V_{ri}^{(1)} + V_{mi} \delta_i^{(1)}] + K_i \cos \delta_i [V_{ri} \delta_i^{(1)} - V_{mi}^{(1)}] \\ P_{ei}^{(2)} = K_i \sin \delta_i [V_{ri}^{(2)} - V_{ri} [\delta_i^{(1)}]^2 + 2V_{mi}^{(1)} \delta_i^{(1)} + V_{mi} \delta_i^{(2)}] + \\ \quad K_i \cos \delta_i [V_{ri} \delta_i^{(2)} + V_{mi} [\delta_i^{(1)}]^2 + 2V_{ri}^{(1)} \delta_i^{(1)} - V_{mi}^{(2)}] \end{array} \right. \quad (E.7)$$

## E.5 Determinação das Tensões e suas Derivadas

Analisando-se as equações (E.3)-(E.5), (E.6) e (E.7), verifica-se a necessidade de se calcular as tensões  $V_{ri}$  e  $V_{mi}$ ,  $i = 1, 2, \dots, m$  e suas derivadas, até a segunda ordem, de modo a obter-se uma aproximação de quarta ordem para  $\delta(t)_i$  e  $\omega_i(t)$ ,  $i = 1, 2, \dots, m$ . Além disso, os esquemas de solução do comportamento dinâmico de SEE, exigem, também, condições iniciais para as tensões em todas as barras do sistema. Estes valores também podem ser obtidos por série de Taylor, conforme descreve-se a seguir.

Supondo as cargas representadas como impedâncias constantes, a equação matricial da rede elétrica pode ser escrita na seguinte forma:

$$\begin{bmatrix} I_g \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{gg} & Y_{gl} \\ Y_{lg} & Y_{ll} \end{bmatrix} \cdot \begin{bmatrix} V_g \\ V_l \end{bmatrix} \quad (E.8)$$

ou, conforme a equação (E.9), usando-se a forma expandida nas partes real e imaginária, e a matriz impedância nodal.

$$\begin{bmatrix} V_r \\ V_m \end{bmatrix} = [Z^e] \cdot \begin{bmatrix} I_r \\ I_m \end{bmatrix} \quad (E.9)$$

onde:

$$\begin{cases} I_{ri} = \frac{E'_{qi}}{x'_{di}} \operatorname{sen} \delta_i - \frac{V_{mi}}{x'_{di}} \\ I_{mi} = \frac{-E'_{qi}}{x'_{di}} \cos \delta_i + \frac{V_{ri}}{x'_{di}} \\ i = 1, 2, \dots, m \end{cases} \quad (\text{E.10})$$

$$\begin{cases} I_{ri} = 0 \\ I_{mi} = 0 \\ i = m + 1, \dots, n \end{cases} \quad (\text{E.11})$$

Note que nas equações (E.8)-(E.11), as cargas são consideradas como impedâncias constantes e incluídas nas matrizes nodais de admitância e impedância.

Considerando-se que  $Z^e$  é uma matriz de constantes e invariante entre dois instantes quaisquer de chaveamentos na rede, a  $\nu$ -ésima derivada das tensões é dada por:

$$\begin{bmatrix} V_r \\ V_m \end{bmatrix}^{(\nu)} = [Z^e] \cdot \begin{bmatrix} I_r \\ I_m \end{bmatrix}^{(\nu)} \quad (\text{E.12})$$

ou

$$\left[ I_r^{(\nu)} + jI_m^{(\nu)} \right] = [Y] \cdot \left[ V_r^{(\nu)} + jV_m^{(\nu)} \right] \quad (\text{E.13})$$

onde  $Y$  é a matriz admitância nodal complexa.

Portanto, a  $\nu$ -ésima derivada das tensões pode ser obtida resolvendo-se a equação (E.13), a partir do conhecimento das derivadas das injeções de correntes. No caso de cargas representadas por impedâncias constantes, a equação (E.13) é linear.

## E.6 Determinação das Injeções de Correntes e suas Derivadas

Incorporando-se a reatância transitória de eixo direto ( $x'_d$ ) na matriz  $Y$ , o sistema de equações (E.10) pode ser reescrito na seguinte forma:

$$\begin{cases} I_{ri} = K_i \operatorname{sen} \delta_i \\ I_{mi} = -K_i \operatorname{cos} \delta_i \\ i = 1, 2, \dots, m \end{cases} \quad (\text{E.14})$$

As derivadas dessas injeções de correntes, até a quarta ordem e para  $i = 1, 2, \dots, m$ , são apresentadas a seguir: são:

$$\begin{cases} I_{ri}^{(1)} = K_i \operatorname{cos} \delta_i [\delta_i^{(1)}] \\ I_{mi}^{(1)} = K_i \operatorname{sen} \delta_i [\delta_i^{(1)}] \end{cases} \quad (\text{E.15})$$

$$\begin{cases} I_{ri}^{(2)} = -K_i \operatorname{sen} \delta_i [\delta_i^{(1)}]^2 + K_i \operatorname{cos} \delta_i [\delta_i^{(2)}] \\ I_{mi}^{(2)} = K_i \operatorname{sen} \delta_i [\delta_i^{(2)}] + K_i \operatorname{cos} \delta_i [\delta_i^{(1)}]^2 \end{cases} \quad (\text{E.16})$$

$$\begin{cases} I_{ri}^{(3)} = -K_i \operatorname{sen} \delta_i [3\delta_i^{(1)}\delta_i^{(2)}] + K_i \operatorname{cos} \delta_i [\delta_i^{(3)} - (\delta_i^{(1)})^3] \\ I_{mi}^{(3)} = K_i \operatorname{sen} \delta_i [\delta_i^{(3)} - (\delta_i^{(1)})^3] + K_i \operatorname{cos} \delta_i [3\delta_i^{(1)}\delta_i^{(2)}] \end{cases} \quad (\text{E.17})$$

$$\begin{cases} I_{ri}^{(4)} = -K_i \operatorname{sen} \delta_i [3(\delta_i^{(2)})^2 + 4\delta_i^{(1)}\delta_i^{(3)} - (\delta_i^{(1)})^4] + K_i \operatorname{cos} \delta_i [\delta_i^{(4)} - 6(\delta_i^{(1)})^2\delta_i^{(2)}] \\ I_{mi}^{(4)} = K_i \operatorname{sen} \delta_i [\delta_i^{(4)} - 6(\delta_i^{(1)})^2\delta_i^{(2)}] + K_i \operatorname{cos} \delta_i [3(\delta_i^{(2)})^2 + 4\delta_i^{(1)}\delta_i^{(3)} - (\delta_i^{(1)})^4] \end{cases} \quad (\text{E.18})$$

Observa-se nas equações (E.7) e (E.14)-(E.18), um elevado número de termos comuns e repetidos. Assim, explorando-se tais características, tem-se meios muito eficientes computacionalmente para se obter uma solução aproximada. O fluxograma da figura E.1 fornece uma visão global de uma implementação paralela desta metodologia, tomando-se uma janela de integração numérica com  $p$  passos e truncando-se a série após o termo de quarta ordem para os ângulos ( $\delta$ ).

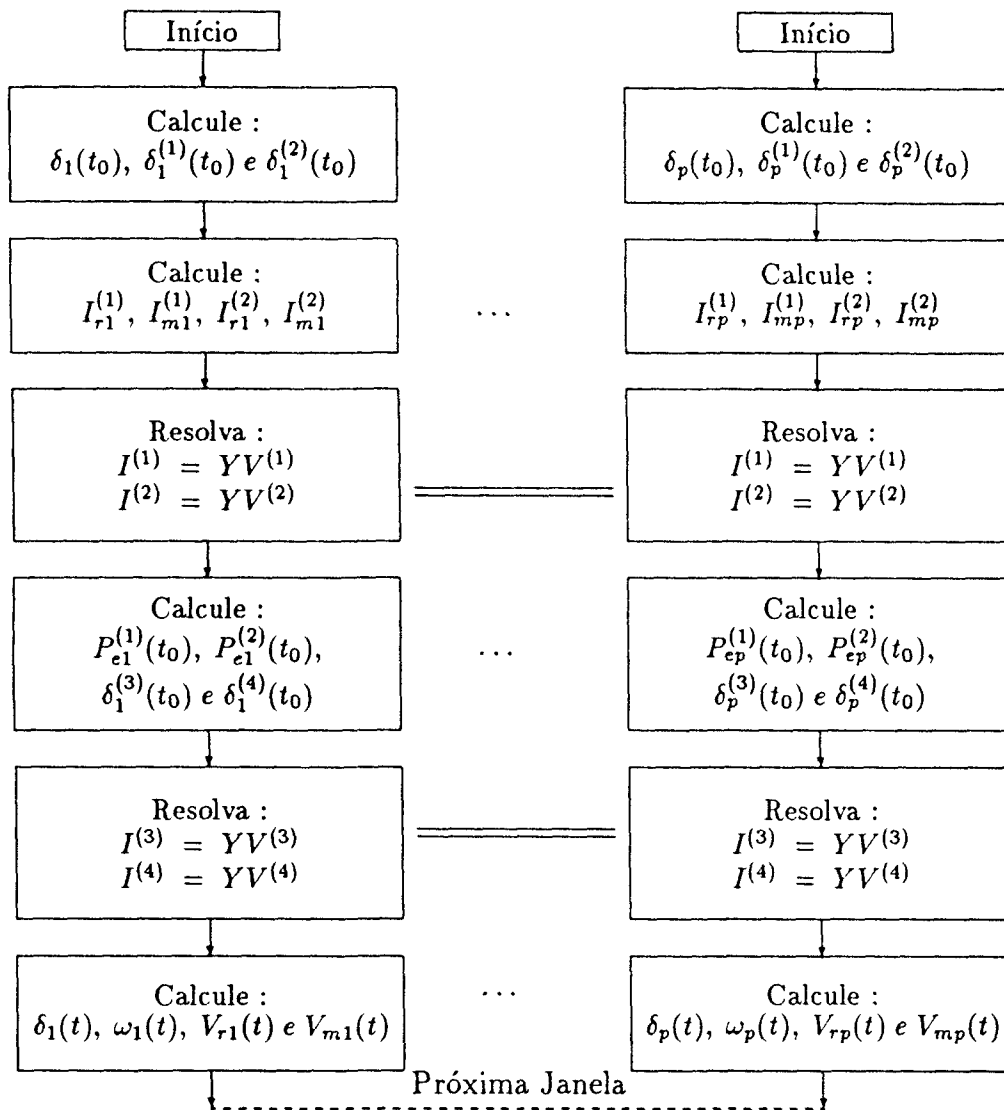


Figura E.1: Solução paralela aproximada por série de Taylor para uma janela, de integração com  $p$  passos



# Bibliografia

- [1] B. Stott, "Power system dynamic response calculations," *Proceedings of the IEEE*, vol. 67, pp. 219–241, Feb. 1979.
- [2] IEEE Committee Report, "Parallel processing in power systems computation," *IEEE Trans. on Power Systems*, vol. 7, pp. 629–638, May 1992.
- [3] M. Ilić-Spong, M. L. Crow, and M. A. Pai, "Transient stability simulation by waveform relaxation method," *IEEE Trans. on Power Systems*, vol. PWRS-2, pp. 943–952, Nov. 1987.
- [4] G. Meurant, "The evolution of scientific computing on parallel computers," Aug. 1992. I Escola de Computação Científica de Alto Desempenho, LNCC/CNPq, Rio de Janeiro, Brasil.
- [5] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. Van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM - Society for Industrial and Applied Mathematics, 1991.
- [6] K. Hwang and F. Briggs, *Computer Architecture and Parallel Processing*. New York: McGraw-Hill Book Company, 1984.
- [7] K. Hwang, "Advanced parallel processing with supercomputer architectures," *Proceedings of the IEEE*, vol. 75, pp. 1348–1379, Oct. 1987.
- [8] F. M. Brasch, J. E. Van Ness, and S. C. Kang, "Simulation of a multiprocessor network for power system problems," *IEEE Trans Power Apparatus and Systems*, vol. PAS-101, no. 2, pp. 295–301, 1982.
- [9] M. Rodrigues, O. R. Saavedra, and A. Monticelli, "Modelo de programação assíncrona para a solução concorrente do fluxo de potência Ótimo com restrições

- de segurança,” in *9o CBA - Congresso Brasileiro de Automática*, (UFES - Vitória/ES), pp. paper WSP2-3, Setembro 14-18 1992.
- [10] A. Gomes and R. Betancourt, “Implementation of the fast decoupled load flow on a vector computer,” *IEEE Trans. on Power Systems*, vol. 5, no. 3, pp. 977-983, 1990.
- [11] C. L. T. Borges, *Investigação do desempenho de métodos de solução do fluxo de potência para processamento paralelo e vetorial*. Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ., 1991.
- [12] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, “An efficient parallel method for transient stability analysis,” in *Proc. of 10<sup>th</sup> PSCC, Power Systems Computation Conference*, (Graz, Austria), pp. 509-516, Aug. 19-24 1990.
- [13] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, “Parallel implementation of power system dynamic simulation methodology using the conjugate gradient method,” *IEEE Trans. on Power Systems*, vol. 7, pp. 458-465, Feb. 1992.
- [14] M. Crow and M. Ilić, “The parallel implementation of the waveform relaxation method for transient stability simulations,” *IEEE Trans. on Power Systems*, vol. 5, pp. 922-931, Aug. 1990.
- [15] M. LaScala, M. Brucoli, F. Torelli, and M. Trovato, “A Gauss-Jacobi-block-Newton method for parallel transient stability analysis,” *IEEE Trans. on Power Systems*, vol. 5, pp. 1168-1177, Nov. 1990.
- [16] M. A. Pai, P. W. Sauer, and A. Y. Kulkarni, “Conjugate gradient approach to parallel processing in dynamic simulation of power systems,” in *1992 ACC/TA10*, pp. 1644-1647, 1992.
- [17] J. S. Chai and A. Bose, “Bottlenecks in parallel algorithms for power systems stability analysis,” *IEEE Trans. on Power Systems*, vol. 8, pp. 9-15, Feb. 1993.
- [18] H. Taoka, I. Iyoda, H. Noguchi, N. Sato, and T. Nakazawa, “Real-time digital simulator for power system analysis on a hypercube computer,” *IEEE Trans. on Power Systems*, vol. 7, pp. 1-10, Feb. 1992.

- [19] D. M. Falcão, E. Kaszkurewicz, and H. L. S. Almeida, "Application of parallel processing techniques to the simulation of power system electromagnetic transients," *IEEE Trans. on Power Systems*, vol. 8, pp. 84–89, Feb. 1993.
- [20] 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," in *IEEE/PES Summer Meeting*, Jul. 1993.
- [21] J. S. Chai, N. Zhu, A. Bose, and D. Tylavsky, "Parallel Newton type methods for power systems stability analysis using local and shared memory multiprocessors," *IEEE Trans. on Power Systems*, vol. 6, pp. 1539–1545, Nov. 1991.
- [22] S. Y. Lee, H. D. Chiang, K. G. Lee, and B. Y. Ku, "Parallel power system transient stability analysis on hipercube multiprocessors," *IEEE Trans. on Power Systems*, vol. 6, pp. 1337–1343, Aug. 1991.
- [23] M. L. Crow, *Waveform relaxation methods for the simulation of systems of differential/algebraic equations with applications to electric power systems*. PhD thesis, University of Illinois at Urbam-Cahampaign, 1990.
- [24] F. L. Alvarado, "Parallel solution of transient problems by trapezoidal integration," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-98, pp. 1080–1090, May/June 1979.
- [25] M. LaScala, A. Bose, D. J. Tylavsky, and J. S. Chai, "A highly parallel method for transient stability analysis," *IEEE Trans. on Power Systems*, vol. 5, pp. 1439–1446, Nov. 1990.
- [26] W. L. Hatcher, F. M. Brasch, and J. E. Van Ness, "A feasibility study for the solution of transient stability problems by multiprocessor structures," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-96, pp. 1789–1797, Nov./Dec. 1977.
- [27] J. Fong and C. Potle, "Parallel processing on power system analysis problems via simple parallel microcomputer structures," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-97, pp. 1834–1841, Sep./Oct. 1978.

- [28] N. Martins, "Decomposition technique for parallel computation of small-signal stability problems in large power systems," in *IASTED Conference on High Technology in the Power Industry*, (Tempe, Arizona), 1988.
- [29] A. P. Feltrin, *Cálculo da estabilidade transitória em sistemas de energia elétrica utilizando o esquema simultâneo implícito*. Tese de Mestrado, Fec/Unicamp, Campinas, SP, 1986.
- [30] C. P. Arnold, *Solutions of the multi-machine power system stability problem*. PhD thesis, Univ. Manchester, 1976.
- [31] H. W. Dommel and N. Sato, "Fast transient stability solutions," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-91, pp. 1643-1650, July/Aug. 1972.
- [32] Boeing Computer Service, "Power system dynamic analysis phase I," Final report EL-0484 project 670-1, EPRI, Palo Alto, Califórnia, 1977.
- [33] V. F. Da Costa, *Análise de estabilidade transitória em sistemas de energia elétrica*. Tese de Mestrado, Fec/Unicamp, Campinas, SP., 1981.
- [34] A. L. M. França, "Simulação da dinâmica de curto prazo em sistemas de energia elétrica: comportamento do esquema alternado implícito," in *V CBA/I CLA*, (Campina Grande, Brasil), 1984.
- [35] Boeing Computer Service, "Dynamic stability calculations using vector and array processors," Final report EL-3335 project 670-2, EPRI, Palo Alto, Califórnia, 1984.
- [36] J. M. Ortega, *Introduction to parallel and vector solution of linear systems*. New York: Plenum Press, 1988.
- [37] M. K. Enns, W. F. Tinney, and F. L. Alvarado, "Sparse matrix inverse factors," *IEEE Trans. on power systems*, vol. 5, no. 2, pp. 466-473, 1990.
- [38] A. Padilha and A. Morelato, "A  $W$  matrix methodology for solving sparse network equations on multiprocessor computers," *IEEE Trans on Power Systems*, vol. 7, pp. 1023-1030, Aug. 1992.

- [39] F. F. Wu, "Real-time network security monitoring, assessment and optimization," *Electrical Power Energy Systems*, vol. 10, no. 2, pp. 83–99, 1988.
- [40] K. J. White and A. Sangiovanni Vincentelli, *Relaxation techniques for the simulation of VLSI circuits*. Boston, Massachusetts: Kluwer Academic Publishers, 1987.
- [41] M. L. Crow and M. Ilić, "Parallel algorithms for large scale power system dynamic simulation," in *Proceedings of 1988 American Control Conference*, pp. 1795–1798, 1988.
- [42] M. L. Crow and M. Ilić, "The parallel implementation of the waveform relaxation method for the simulation of structure-preserved power systems," in *Paper IEEE CH2868-8/90*, pp. 1285–1288, 1990.
- [43] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, "Simulação da dinâmica de sistemas de energia elétrica usando técnicas de processamento paralelo," Relatório Interno RT-01/93, Programa de Engenharia Elétrica, COPPE/UFRJ, Ilha do Fundão, Bloco H, Rio de Janeiro -RJ., 1991.
- [44] J. S. Chai, A. Bose, D. Tylavsky, and N. Zhu, "Discussão da referência Crow e Ilić (1990)," *IEEE Trans. on Power Systems*, vol. 5, pp. 930–931, Aug. 1990.
- [45] J. Zaborszky, G. Huang, and K. W. Lu, "A textured model for computationally efficient reactive power control and management," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-104, pp. 1718–1727, July 1985.
- [46] D. J. Tylavsky, "Parallel-vector processing synergy and systems dynamic simulations," in *Proceedings of the EPRI/NSF Workshop on Application of Advanced Mathematics to Power Systems*, Sep. 1991. Redwood City, CA.
- [47] A. Bhaya, D. M. Falcão, E. Kaszkurewicz, and N. Roqueiro, "Parallel block iterative methods: application to the load flow problem," in *Proceedings of the IFAC/IFORS/IMACS Symposium on Large Scale Systems: Theory and Applications*, (Berlim GDR), pp. 364–368, 1989.
- [48] C. Aykanat, F. Özgüner, F. Ercal, and P. Sadayappan, "Iterative algorithms for solution for large sparse systems of linear equations on hypercubes," *IEEE Trans on Computers*, vol. 37, pp. 1554–1568, Dec. 1988.

- [49] A. L. G. A. Coutinho, J. L. D. Alves, L. Landau, N. F. F. Ebecken, and L. M. Troina, "Comparasion of Lanczos and Conjugate Gradients for the element-by-element solution of finite element equations on the IBM 3090 vector computer," *Computers & Structures*, vol. 39, no. 1/2, pp. 47-55, 1991.
- [50] F. Angereli, V. Sonnad, and K. Bathe, "Studies of finite element procedures - an evaluation of preconditioned iterative solvers," *Computers & Sturctures*, vol. 32, no. 3/4, pp. 671-677, 1989.
- [51] W. J. Mansur, F. C. Araújo, and J. E. B. Malaghini, "Solution of BEM systems of equations via iterative tecniques," *International Journal for Numerical Methods in Engeneering*, vol. a ser publicado, p. , 1992.
- [52] F. Shakib, T. J. R. Hughes, and Z. Johan, "A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis," *Computer Methods in applied Mechanics and Engineering*, vol. 75, pp. 415-456, 1989. North-Holland.
- [53] H. A. Van der Vorst, "BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 2, pp. 631-644, 1992.
- [54] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, "Esquema Alternado Entrelaçado Implícito Paralelo - Aplicação do método Gradiente Conjugado pré-condicionado as equações da rede elétrica," Relatório Interno RT-02/93, Programa de Engenharia Elétrica, COPPE/UFRJ, Ilha do Fundão, Bloco H, Rio de Janeiro -RJ., 1993.
- [55] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, "Esquema Simultâneo Implícito Paralelo no Tempo - Aplicação dos métodos Gradiente Biconjugado e Bi-CGSTAB," Relatório Interno RT-03/93, Programa de Engenharia Elétrica, COPPE/UFRJ, Ilha do Fundão, Bloco H, Rio de Janeiro -RJ., 1993.
- [56] R. G. Cabral, *Avaliação de desempenho do método dos gradientes conjugados em multiprocessadores com arquitetura hipercúbica*. Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, 1991.

- [57] A. L. G. A. Coutinho, J. L. D. Alves, L. Landau, and N. F. F. Ebecken, "Solução iterativa de sistemas de equações do método dos elementos finitos no computador IBM 3090," in *II Simp. Bras. de Arq. de Comp.*, (Lindóia, SP.), set. 26-28 1989.
- [58] M. Ribbens-Pavella, B. Lemal, and W. Pirard, "On-line operation of Lyapunov criterion for transient stability studies," in *Proc. of IFAC Symposium*, (Melbourne, Australia), pp. 292-296, 1977.
- [59] I. C. Decker, *Análise de estabilidade transitória em sistemas de potência usando o conceito de superfície limite de energia potencial*. Tese de Mestrado, UFSC, Florianópolis, SC., 1984.
- [60] *Parallel Fortran User Guide*. 3L Ltd and EPCL, Scotland, versão 2.0 ed., 1988.
- [61] M. H. M. Vale, D. M. Falcão, and E. Kaszkurewicz, "Electrical power network decomposition for parallel computations," in *1992 IEEE ISCAS, International Symposium on Circuits and Systems*, (San Diego, CA.), May 10-13 1992.
- [62] IEEE Committee Report, "Computer representation of excitation systems," *IEEE Trans. on Power App. Syst.*, vol. PAS-87, no. 6, pp. 50-54, 1968.
- [63] IEEE Committee Report, "Excitation system models for power system stability studies," *IEEE Trans on Power App. and Syst.*, vol. PAS-100, no. 2, pp. 494-509, 1981.