

**UNIVERSIDADE FEDERAL DE MINAS GERAIS**  
**Escola de Engenharia**  
**Programa de Pós-Graduação em Engenharia Elétrica**

João Paulo Ramos Gomes

**ARQUITETURA COMPUTACIONAL DISTRIBUÍDA E  
COLABORATIVA PARA ESTUDOS DE CÁLCULO DE TRANSITÓRIOS  
ELETROMAGNÉTICOS PERMITINDO INTERAÇÃO SOB-DEMANDA**

**Tese de doutorado**

Belo Horizonte  
2022

João Paulo Ramos Gomes

**ARQUITETURA COMPUTACIONAL DISTRIBUÍDA E  
COLABORATIVA PARA ESTUDOS DE CÁLCULO DE TRANSITÓRIOS  
ELETROMAGNÉTICOS PERMITINDO INTERAÇÃO SOB-DEMANDA**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica.

Área de Concentração: Engenharia de Potência

Orientador: Prof. Dr. Wallace do Couto Boaventura

Belo Horizonte

2022

G633a

Gomes, João Paulo Ramos.

Arquitetura computacional distribuída e colaborativa para estudos de cálculo de transitórios eletromagnéticos permitindo interação sob-demanda [recurso eletrônico] / João Paulo Ramos Gomes. - 2022.  
1 recurso online (128 f. : il., color.) : pdf.

Orientador: Wallace do Couto Boaventura.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 127-128.

Bibliografia: f.122-126.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Transitórios (Eletricidade) - Teses.  
3. Arquitetura de computador - Teses. I. Boaventura, Wallace do Couto.  
II. Universidade Federal de Minas Gerais. Escola de Engenharia. III.  
Título.

CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
COLEGIADO DO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

## FOLHA DE APROVAÇÃO

### Arquitetura Computacional Distribuída e Colaborativa para Estudos de Cálculo de Transitórios Eletromagnéticos Permitindo Interação Sob-demanda

**JOÃO PAULO RAMOS GOMES**

Tese de Doutorado defendida e aprovada, no dia 11 abril de 2022, pela Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais constituída pelos seguintes professores:

**Prof. Dr. José Osvaldo Saldanha Paulino (Departamento de Engenharia Elétrica - UFMG)**

**Prof. Dr. Washington Luiz Araujo Neves (Departamento de Engenharia Elétrica - UFCG)**

**Prof. Dr. Hélder de Paula (Faculdade de Engenharia Elétrica - UFU)**

**Prof. Dr. Estácio Tavares Wanderley Neto (Departamento de Eletrotécnica - UNIFEI)**

**Prof. Dr. Wallace do Couto Boaventura (Departamento de Engenharia Elétrica - UFMG) - orientador**

Belo Horizonte, 11 de abril de 2022.

PROF. WALLACE DO COUTO BOAVENTURA

Presidente da Comissão Examinadora



Documento assinado eletronicamente por **Wallace do Couto Boaventura, Presidente de comissão**, em 18/04/2022, às 11:26, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufmg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1377690** e o código CRC **7DBF375B**.

# Agradecimentos

Em primeiro lugar, agradeço à Deus, que entendo ser a inteligência suprema e criadora do universo.

Agradeço aos meus pais, José e Arilda, pelo amor ofertado durante uma vida, pela forma tão doce e delicada de me apresentarem a motivação pelo trabalho e o conhecimento. A vida com eles sempre foi uma oportunidade de alegria, desafio, simplicidade, poesia e grandes sonhos a serem conquistados. Agradeço às minhas irmãs, Poly e Josy, pela parceria e apoio contínuo nos momentos de felicidade e nas horas difíceis.

Agradeço a toda família de minha esposa pelo apoio nessa pequena jornada.

Agradeço ao meu orientador Wallace por ter me proporcionado a chance de conhecer um jeito novo e divertido de trabalhar, conhecido apenas pelos gênios que se dedicam a uma tarefa com amor. Há 11 anos eu procurava um estágio, a porta da sala dele estava aberta, um convite foi feito e aqui estou para agradecer o desenvolvimento de uma tese de doutorado. Muito obrigado Wallace!

Agradeço a todas as pessoas que direta ou indiretamente me ajudaram nessa caminhada. São muitas, não caberia nas páginas desta tese.

Agradeço à Universidade Federal de Minas Gerais por ter me formado como pesquisador e também por ter me ajudado a ser um cidadão mais humano e pacífico. Agradeço ao PPGEE pelo apoio e suporte da estrutura.

Agradeço à turminha do LEAT, LAPIS, LCE, LAPO por dias de trabalho tão especiais.

Agradeço a CAPES, CNPQ e FAPEMIG, os quais representam o povo brasileiro, por financiar parte de meus estudos.

Agradecemos a CEMIG - Companhia Energética de Minas Gerais pelo financiamento parcial deste trabalho através do Projeto de Pesquisa e Desenvolvimento "D722 Análise de Arranjo Técnico e Comercial baseado em uma Planta Piloto de Sistema Distribuído de Armazenamento de Energia em Alimentador Crítico da Rede de Distribuição ao de 13,8 kV". desenvolvido no âmbito do Programa de Pesquisa e Desenvolvimento da ANEEL - Agência Nacional de Energia Elétrica.

Agradeço à minha esposa Marcela Bedeschi Faria Ramos por estar aqui hoje comigo. Penso que o desenvolvimento de um amor durante um curso de doutorado não é coisa fácil, mas é certo que se ambos quiserem ele somente fortalecerá.

*"Se você der o seu melhor, você ficará impressionado com o que pode fazer."*  
(Desconhecido)

*"Eu tenho em mim todos os sonhos do mundo."*  
(Fernando Pessoa)

*"If you can dream it, you can do it. Always remember that this whole thing started with a dream and a mouse."*  
(Walt Disney)

# Resumo

Os seguidores de Maxwell sempre aceitaram um desafio. Desta vez, não seria diferente. Este trabalho apresenta uma Arquitetura Computacional para simulações distribuídas e colaborativas para o cálculo de transitórios eletromagnéticos. Por meio dessa ferramenta, diferentes sistemas, executando em computadores diferentes em locais distribuídos, podem se conectar utilizando protocolos da Internet para formar grandes sistemas. De maneira inovadora, permite que a integração entre os subsistemas possa ocorrer sob-demanda, o que a diferencia das propostas existentes na literatura. Além disso, permite a interação do usuário com o sistema de energia durante a simulação. Para mostrar como a ferramenta funciona, são desenvolvidos estudos de caso contemplando o modo de operação sob-demanda. Além destes, outros são realizados para ilustrar outras oportunidades decorrentes deste tipo de arquitetura. Um é formado por um grande sistema com 3200 barras, 28 parques eólicos e 7 parques solares. Outros usam subsistemas menores com 39 barras. Os resultados mostram o tempo de execução das simulações, uma breve discussão sobre a latência do processo de comunicação entre computadores e as oportunidades de trabalho colaborativo adicionadas pela ferramenta.

**Palavras-chaves:** Trabalho colaborativo, Transitórios eletromagnéticos, Computação distribuída, Sistema de potência.

# Abstract

Maxwell's followers have always accepted a challenge. This time it would be no different. This work presents a Computational Architecture for distributed and collaborative simulations for the calculation of electromagnetic transients. Through this tool, different systems, running on different computers in distributed locations, can connect using Internet protocols to form large systems. In an innovative way, it allows the integration between the subsystems to occur on-demand, which differentiates it from existing proposals in the literature. In addition, it allows user interaction with the power system during the simulation. To show how the tool works, case studies are developed contemplating the on-demand operation mode. In addition to these, others are carried out to illustrate other opportunities arising from this type of architecture. One is formed by a large system with 3200 buses, 28 wind farms and 7 solar parks. Others use smaller subsystems with 39 buses. The results show the execution time of the simulations, a brief discussion about the latency of the communication process between computers and the collaborative work opportunities added by the tool.

**Key-words:** Collaborative work, electromagnetic transients, distributed computing, power systems.



# Lista de ilustrações

Figura 2.1 – Equivalente Norton para indutância. . . . .	24
Figura 2.2 – Circuito equivalente para capacitância. . . . .	24
Figura 2.3 – Modelo de linha sem perdas. . . . .	25
Figura 2.4 – Circuito equivalente da linha de transmissão no domínio da frequência. . . . .	28
Figura 2.5 – Método de compensação. . . . .	34
Figura 2.6 – Circuito exemplo. . . . .	37
Figura 2.7 – Equivalente Norton do circuito exemplo. . . . .	37
Figura 2.8 – Computação sequencial. . . . .	40
Figura 2.9 – Computação paralela. . . . .	41
Figura 2.10–Modelo híbrido com memória compartilhada e distribuída. . . . .	44
Figura 2.11–Modelo híbrido com memória compartilhada e distribuída e adição de GPU. . . . .	45
Figura 2.12–Comunicação entre processos. Memória compartilhada (à esquerda) e Passagem de mensagem (a direita). . . . .	47
Figura 2.13–Comunicação entre processos. Passagem de Mensagem. . . . .	47
Figura 2.14–Comunicação cliente-servidor via socket. . . . .	49
Figura 3.1 – Computação paralela no MATLAB.[1] . . . . .	67
Figura 3.2 – Computação distribuída no Matlab. .[2] . . . . .	67
Figura 3.3 – Plataforma de computação paralela no MATLAB .[2] . . . . .	68
Figura 3.4 – Cálculo de transitórios no PSCAD [3]. . . . .	69
Figura 3.5 – Estrutura multiprojeto para simulação no PSCAD [4]. . . . .	70
Figura 3.6 – Paralelismo de dados.[4] . . . . .	71
Figura 3.7 – Paralelismo de Tarefas.[4] . . . . .	71
Figura 4.1 – Ilustração para uma LAN . . . . .	77
Figura 4.2 – Ilustração para um WAN. . . . .	78
Figura 4.3 – Transformação entre os domínios modal e fase para uma linha trifásica. . . . .	78
Figura 4.4 – Integração dos componentes HCoE cliente e servidor ao cálculo de transitórios eletromagnéticos do PSCAD/EMTDC. . . . .	80
Figura 4.5 – Arquitetura computacional para metodologia <i>sob demanda</i> . . . . .	81
Figura 4.6 – Ilustração do cálculo de EMT síncrono. . . . .	84
Figura 4.7 – Ilustração do cálculo de EMT <i>sob demanda</i> . . . . .	86
Figura 4.8 – Arquitetura <i>sob demanda</i> com os passos do protocolo de funcionamento. . . . .	86
Figura 4.9 – Mapa de complexidade do projeto <i>sob demanda</i> . . . . .	91
Figura 4.10–Mapa de complexidade do projeto síncrono. . . . .	91
Figura 5.1 – Ilustração de circuito DC implementado no ATP e PSCAD. . . . .	93
Figura 5.2 – Circuito DC implementado com o componente HCoE. . . . .	94

Figura 5.3 – Circuito AC implementado no PSCAD. . . . .	94
Figura 5.4 – Circuito AC implementado com o componente HCoE. . . . .	95
Figura 5.5 – Comparativo ATP x HCoE, fase A. . . . .	95
Figura 5.6 – Comparativo ATP x HCoE, fase B. . . . .	96
Figura 5.7 – Comparativo ATP x HCoE, fase C. . . . .	96
Figura 5.8 – Comparativo PSCAD x HCoE, fase A. . . . .	97
Figura 5.9 – Comparativo PSCAD x HCoE, fase B. . . . .	97
Figura 5.10–Comparativo PSCAD x HCoE, fase C. . . . .	98
Figura 5.11–Comparativo DC PSCAD x HCoE, fase A. . . . .	98
Figura 5.12–Comparativo DC PSCAD x HCoE, fase B. . . . .	99
Figura 5.13–Comparativo DC PSCAD x HCoE, fase C. . . . .	99
Figura 5.14–Esquemático do sistema para o modo de operação síncrono. . . . .	100
Figura 5.15–Tensão no lado 2 do disjuntor no subsistema A. . . . .	101
Figura 5.16–Tensão no lado 2 do disjuntor no subsistema B. . . . .	101
Figura 5.17–Tensão no lado 2 do disjuntor no subsistema B (destaque). . . . .	102
Figura 5.18–Esquemático do sistema em modo de operação <i>sob demanda</i> . . . . .	103
Figura 5.19–Tensão no lado 2 do disjuntor no subsistema A, barra 16. . . . .	105
Figura 5.20–Tensão no lado 2 do disjuntor no subsistema A, barra 33. . . . .	105
Figura 5.21–Tensão no lado 1 do disjuntor no subsistema A, barra 16. . . . .	106
Figura 5.22–Tensão no lado 1 do disjuntor no subsistema A, barra 33. . . . .	106
Figura 5.23–Tensão no 2 do disjuntor no subsistema B. . . . .	107
Figura 5.24–Tensão no lado 2 do disjuntor no subsistema C, barra 33. . . . .	107
Figura 5.25–Mapeamento 1 dos locais onde ocorreram as simulações. . . . .	109
Figura 5.26–Mapeamento 2 dos locais onde ocorreram as simulações. . . . .	109
Figura 5.27–Sistema de potência com 3200 barras, 28 parques eólicas(box marrom) e 7 usinas solares (box amarelo).Linha de transmissão (box verde) . . . . .	111

# Lista de tabelas

Tabela 4.1 – Parâmetros para desenvolvimento de uma arquitetura computacional para cálculo de transitórios eletromagnéticos. . . . .	76
Tabela 4.2 – Parâmetros que compõem o desenvolvimento do projeto <i>sob demanda</i> . . . . .	89
Tabela 4.3 – Inter-relação entre parâmetros do projeto <i>sob demanda</i> . . . . .	89
Tabela 4.4 – Parâmetros que compõem o desenvolvimento do projeto síncrono. . . . .	90
Tabela 4.5 – Inter-relação entre parâmetros para o projeto síncrono. . . . .	90
Tabela 5.1 – Recurso computacional utilizado para simulações da tese. . . . .	94
Tabela 5.2 – Cenários de simulação distribuída. . . . .	110
Tabela 5.3 – Resultado das simulações para os cenários de estudo. . . . .	112

# Lista de abreviaturas e siglas

API	<i>Application Program Interface</i>	43
ATP	<i>Alternative Transient Program</i>	56
CHIL	<i>Controller Hardware in Loop</i>	59
COM	<i>Component Object Model</i>	57
COS	<i>Centro de Operação do Sistema</i>	80
CPU	<i>Central Processing Unit</i>	44
DDE	<i>Dynamic Data Exchange</i>	57
EMT	<i>Electromagnetic Transient Program</i>	15
EMTDC	<i>Electromagnetic Transients Including DC</i>	56
ENI	<i>Electrical Network Interface</i>	56
FACT	<i>Flexible AC Transmission System</i>	64
FMI	<i>Functional Mock-up Interface</i>	65
G2V	<i>Grid to Vehicle</i>	18
GPU	<i>Graphic Processing Unit</i>	44
HCoE	<i>Hub Co-simulation Engine</i>	77
HIL	<i>Hardware in Loop</i>	54
HVDC	<i>High Voltage Direct Current</i>	57
IP	<i>Internet Protocol</i>	49
IPC	<i>Inter process communication</i>	46
IRC	<i>Inter-rack communications card</i>	73
LAN	<i>Local Area Network</i>	61
MDCS	<i>Matlab Distributed Computing Server</i>	66
MPI	<i>Message Passing Interface</i>	43
MPMD	<i>Multiple Instruction Multiple Data</i>	43
ONS	<i>Operador Nacional do Sistema</i>	19
OPENMP	<i>Open Multi-Processing</i>	43
OPENMPI	<i>Open Source High Performance Computing</i>	48
PHIL	<i>Power Hardware in Loop</i>	59
PSCAD	<i>Power Systems Computer Aided Design</i>	33
RPC	<i>Remote Process Call</i>	47
SIN	<i>Sistema Interligado Nacional</i>	19
SPMD	<i>Single Instruction Multiple Data</i>	43
TCP	<i>Transmission Control Protocol</i>	49
TRT	<i>Tensão de Restabelecimento Transitória</i>	20
TS	<i>Transient Stability</i>	62

UDP	<i>User Datagram Protocol</i>	49
UMEC	<i>Unified Magnetic Equivalent Circuit</i>	36
V2G	<i>Vehicle to Grid</i>	18
WAN	<i>Wide Area Network</i>	61
WIC	<i>Workstation interface card</i>	73
XGE	<i>Xoreax Grid Engine</i>	56

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	O futuro das redes elétricas de energia e sua relação com a simulação de transitórios eletromagnéticos	18
1.2	Motivação para a pesquisa	19
1.3	Objetivos do trabalho	20
1.4	Contribuições da tese de doutorado	21
1.5	Organização da tese de doutorado	22
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>23</b>
2.1	Cálculo de transitórios eletromagnéticos	23
2.1.1	Resistência, indutância e capacitância	24
2.1.2	Linha de transmissão	25
2.1.2.1	Linhas de transmissão com parâmetros dependentes da frequência	26
2.1.2.2	Modelagem de linha de transmissão no domínio modal	28
2.1.2.3	Modelagem da linha de transmissão no domínio de fases	31
2.1.3	Elementos não lineares	33
2.1.4	Transformadores	34
2.1.5	Outros componentes elétricos	36
2.1.6	Equação nodal para cálculo matricial	36
2.1.7	Considerações parciais	38
2.2	Computação paralela e distribuída	39
2.2.1	Computação paralela	39
2.2.2	Termos utilizados em computação paralela	42
2.2.3	Modelos de programação paralela	43
2.2.4	Considerações parciais	45
2.3	Comunicação entre processos	46
2.3.1	Modelos de comunicação entre processos	46
2.3.2	Interfaces de comunicação - API	48
2.3.3	Comunicação em rede utilizando protocolos TCP e IP com sockets	48
2.3.4	Considerações parciais	50
2.4	Considerações finais	50
<b>3</b>	<b>REVISÃO DA LITERATURA</b>	<b>52</b>
3.1	Computação paralela aplicada a cálculo de transitórios eletromagnéticos	52
3.1.1	Considerações parciais	59

<b>3.2</b>	<b>Cálculo Computacional Colaborativo</b>	<b>60</b>
3.2.1	Considerações parciais	62
<b>3.3</b>	<b>Ferramentas matemáticas e computacionais para integração de diferentes simuladores</b>	<b>62</b>
3.3.1	Considerações parciais	65
<b>3.4</b>	<b>Softwares “off the shelf” para simulação do tipo EMT ( <i>Electromagnetic Transient Program</i>) em ambiente paralelo e distribuído.</b>	<b>66</b>
3.4.1	Considerações parciais	73
<b>3.5</b>	<b>Considerações finais</b>	<b>73</b>
<b>4</b>	<b>ARQUITETURA COMPUTACIONAL DISTRIBUÍDA, COLABORATIVA PARA CÁLCULO DE TRANSITÓRIOS ELETROMAGNÉTICOS COM INTERAÇÃO SOB DEMANDA</b>	<b>75</b>
4.1	Características gerais da arquitetura computacional	75
4.2	Modelo de Componente personalizado — HCoE.	77
4.3	Protocolos de operação da arquitetura computacional nos modos de operação síncrono e <i>sob demanda</i> .	83
4.4	Mapa de complexidade da arquitetura computacional proposta na tese	88
4.5	Considerações finais	92
<b>5</b>	<b>TESTES COMPUTACIONAIS</b>	<b>93</b>
<b>5.1</b>	<b>Validação do componente HCoE da arquitetura computacional</b>	<b>93</b>
5.1.1	Considerações parciais	97
<b>5.2</b>	<b>Testes computacionais para o modo de operação síncrono</b>	<b>99</b>
5.2.1	Considerações parciais	101
<b>5.3</b>	<b>Testes computacionais para o modo de operação <i>sob demanda</i></b>	<b>102</b>
5.3.1	Considerações parciais	108
<b>5.4</b>	<b>Análise da latência no cálculo de EMT em simulações com computação distribuída</b>	<b>108</b>
5.4.1	Considerações parciais	110
<b>5.5</b>	<b>Simulação de grandes sistemas de potência utilizando computação distribuída</b>	<b>110</b>
5.5.1	Considerações parciais	112
<b>5.6</b>	<b>Discussão dos resultados</b>	<b>112</b>
<b>6</b>	<b>CONCLUSÃO E PROPOSTAS DE CONTINUIDADE</b>	<b>118</b>
<b>6.1</b>	<b>Propostas de continuidade</b>	<b>119</b>
6.1.1	Outros softwares off the shelf para a mesma arquitetura computacional	119

6.1.2	Arquitetura computacional voltada para simuladores híbridos com objetivos variados . . . . .	119
6.1.3	Desenvolvimento de interfaces para simulações paralelas e distribuídas . . . . .	119
6.1.4	Avaliação do impacto econômico de uma aplicação utilizando computação paralela e distribuída . . . . .	120
6.1.5	Simulação geograficamente distribuída envolvendo <i>Hardware in the Loop</i> . . . . .	120
6.1.6	Novo projeto de arquitetura computacional com mais funcionalidades e integração de sistemas homogêneos ou heterogêneos . . . . .	120
6.1.7	Desenvolvimento de um centro de pesquisa voltado para sistemas computacionais colaborativos e multidisciplinares. . . . .	121
6.1.8	Aplicação de conceitos desta tese em outras áreas . . . . .	121
	<b>REFERÊNCIAS . . . . .</b>	<b>122</b>
	<b>ANEXOS</b>	<b>127</b>
	<b>ANEXO A – ARTIGOS PUBLICADOS . . . . .</b>	<b>128</b>



# 1 Introdução

Uma proposta de cálculo digital de transitórios eletromagnéticos foi apresentado por Hermann W. Dommel em 1969 à comunidade científica mundial. De acordo com este pesquisador, através desta metodologia, seria possível representar o comportamento de um sistema elétrico real por meio de um modelo digital. A partir daquele momento, pesquisas em torno de como fazer modelos digitais de componentes elétricos ganharam enorme impulso. Concomitantemente, em outras áreas da ciência, a ideia de um modelo virtual, também se desenvolvia. Atualmente, as novas tecnologias computacionais e a pandemia do COVID-19 estão direcionando o mundo para um aperfeiçoamento desses modelos, conhecidos como gêmeos digitais “*Digital Twin*” [5], [6].

Quando Dommel propôs um modelo digital para os sistemas elétricos houve um enorme alvoroço na comunidade científica. Isto porque aquele trabalho traria enormes consequências para o âmbito das pesquisas na área. Ter uma modelagem digital, que permitisse simular o comportamento de um sistema real, significava menos custos para pesquisas em engenharia elétrica. Os efeitos da propagação de surto eletromagnético em uma linha de transmissão, gerados por uma manobra de um disjuntor, ou uma descarga atmosférica poderiam ser estudados sem mobilizar nenhuma equipe de operação, por exemplo.

Embora a ideia fosse extremamente interessante, as análises digitais possuíam muitas limitações, as quais estavam relacionadas à capacidade computacional do hardware disponível na época. Muitos dos componentes dos sistemas elétricos possuem uma natureza não-linear e, somado a isso, quanto maior o tamanho do sistema elétrico, a demanda por um sistema computacional mais robusto aumenta. Tudo isso não é mais um problema hoje. Tecnologias como hardware multinúcleo, algoritmos para aprendizagem de máquina e inteligência artificial, memórias com alta capacidade de armazenamento de dados, rede de comunicação 4G e 5G, consolidam a possibilidade de representação de um modelo digital para o setor elétrico e outras áreas.

Outrossim, é sabido que a introdução de novas tecnologias no cotidiano de trabalho, algumas vezes é motivo de resistência. Primeiro, porque mudar exige um esforço pessoal no sentido de aprender o que está sendo trazido; segundo, pelo fato de que a nova proposta pode transformar relações de trabalho, reduzir funcionários em determinado setor de serviços, redução de custos e impacto nos interesses de muitas pessoas. É fato que a introdução de uma nova tecnologia gera enorme impacto na vida das pessoas, o que pode ser bom para uns, pode ser ruim para outros. Esse problema é resolvido quando se encontra um denominador comum ao qual a solução proposta terá impacto positivo. E é assim, que

a crise econômica, advinda da pandemia do COVID-19, serve como um agente acelerador para a absorção dos conceitos como modelos e metodologias de trabalho. O denominador aqui é redução de custos, agilidade de análises e tempo para pensar em novas soluções para o setor.

O *Digital Twin* realmente veio para ficar. Não somente, porque há um conjunto sólido de novas tecnologias que permitem a representação de sistemas reais com maior confiabilidade. Mas também, pelo fato de que a crise que estamos vivendo estabelece um denominador comum para todos nós, precisamos reduzir custos e sermos mais eficientes. Não há tempo a perder.

## 1.1 O futuro das redes elétricas de energia e sua relação com a simulação de transitórios eletromagnéticos

Um comitê técnico, organizado pela União Europeia [7], apresentou um relatório com foco nas visões e estratégias para redes elétricas europeias do futuro, “Smart Grids”. Neste documento, ela reitera o necessário compromisso dos países membros em investir numa rede elétrica que satisfaça a quatro requisitos principais. O primeiro deles é a flexibilidade, de modo a satisfazer as variabilidades nas necessidades dos clientes, respondendo às mudanças e desafios. O segundo trata do aspecto de acessibilidade da rede, garantindo conexão aos usuários, particularmente para fontes renováveis de energia e geração local com alta eficiência e baixa emissão de carbono. O terceiro aborda a confiabilidade, devendo-se assegurar o constante fornecimento e melhoria na qualidade de energia entregue ao consumidor. E, por último, que estes aspectos sejam considerados de maneira econômica e eficiente. Outros países, tais como os Estados Unidos, Canadá e Japão, têm seguido este mesmo caminho.

Em concordância com esta tendência, uma série de desafios e oportunidades para profissionais do setor elétrico têm surgido, sendo minuciosamente discutido por [8], [9] e [10]. Temas como fluxo bidirecional de energia na rede elétrica; unidades geradoras de energia, solar e eólica; integração de veículos elétricos, ora armazenando, ora consumindo energia *G2V (Grid to Vehicle)*, *V2G (Vehicle to Grid)*; geração distribuída; ilhamento; capacidade de auto-reconfiguração (self-healing) diante de distúrbios indesejáveis; gerenciamento de energia pela demanda; manutenção e operação automatizadas e infraestrutura avançada de monitoramento estão se tornando cada vez mais recorrentes nos principais congressos de engenharia elétrica da atualidade.

Os requisitos de qualidade de energia expostos anteriormente, os quais têm se tornado possíveis devido ao expressivo desenvolvimento da eletrônica de potência, da computação e das telecomunicações, apontam para a crescente penetração de novos dispositivos elétricos às redes de energia. Isto faz com que a modelagem eletromagnética dos

sistemas de energia para eventuais simulações, como a de transitórios eletromagnéticos, se torne mais complexa. Somado a isto, está a demanda por análises cada vez mais exatas e precisas relacionadas com projetos para implementação e operação no setor elétrico. Como discutido por [11] e [12], satisfazer estas necessidades exige muito da capacidade de processamentos dos computadores, além de algoritmos computacionais e modelos matemáticos mais eficazes e adaptáveis à velocidade de transformação destas inovações tecnológicas.

No próximo tópico, destacam-se três exemplos, dentro desse universo de transformações do setor elétrico, que serviram de motivação e direcionamento para este trabalho de pesquisa.

## 1.2 Motivação para a pesquisa

No Brasil, para que uma empresa estabeleça a conexão de uma unidade geradora de energia ao *SIN (Sistema Interligado Nacional)*, um conjunto de requisitos técnicos deve ser satisfeito. Estas normas são definidas pelo *ONS (Operador Nacional do Sistema)*, cujo objetivo é manter a qualidade da energia elétrica na rede básica de energia do país. Estudos de transitórios eletromagnéticos são pré-requisitos para provar que a unidade geradora concorda com o que foi solicitado pelo *ONS*, como, por exemplo, estudos de curto-circuito trifásico. As simulações nesse contexto são realizadas através de um equivalente de rede para o sistema de potência interligado nacional e uma modelagem mais detalhada da unidade geradora e à rede retida (conjunto de barras próximas ao empreendimento de geração).

A necessidade de utilizar um equivalente de rede para um sistema completo está relacionada a aspectos como disponibilidade e tempo para preparar os dados da rede; limitações dos softwares quanto ao número de componentes por simulação; tempo de processamento da simulação completa. Embora seja essa a metodologia de estudo definida pelo *ONS*, ela continua sendo uma tarefa árdua para os projetistas do setor elétrico. A entrega de um relatório técnico para o *ONS* pode levar semanas. E, ainda, pode sofrer muitas revisões, devido a erros no projeto *ONS*, [13]. Conforme o Operador Nacional do Sistema, o número médio de Projetos Básicos que sofrem revisão nos tópicos estudos de transitórios eletromagnéticos é 56%, o que significa retrabalho tanto para o *ONS* quanto para os projetistas. O número médio de revisões dos Projetos Básicos completo tem sido elevado: acima de 3, além da emissão inicial *ONS*.

Um segundo exemplo, trata-se de trabalhos colaborativos entre universidades. Como já citado, são inúmeras as inovações tecnológicas introduzidas às redes elétricas. A consequência dessas mudanças é a demanda por modelagens elétricas mais complexas e multidisciplinares, cujo objetivo é fornecer resultados exatos e precisos das simulações dos eventos em estudo. Isso confirma a necessidade de se trabalhar em simulações colaborati-

vas, o que oportuniza o compartilhamento de recursos computacionais, menos tempo de trabalho envolvido com a modelagem de um problema e disponibilidade de informações aos envolvidos.

Considere o estudo da *TRT (Tensão de Restabelecimento Transitória)* durante a manobra de disjuntores numa linha de transmissão. Dois grupos em laboratórios separados podem realizar uma simulação envolvendo dois subsistemas A e B, separados por uma linha de transmissão. Esses subsistemas podem ser iguais ou distintos, por exemplo, um deles com uma rede elétrica malhada e o outro com uma radial. De forma colaborativa, os grupos de pesquisa poderiam avaliar os efeitos do tipo de rede na *TRT*.

Um terceiro exemplo são simulações envolvendo dispositivos comerciais, nos quais a propriedade intelectual e confidencialidade de dados é tema fundamental. Considere a análise de conexão de um parque eólico, cuja topologia do circuito é confidencial, a um sistema de potência. Nesta situação, deseja-se avaliar o comportamento do sistema durante um curto-circuito no ponto de acoplamento comum. Digamos que o parque eólico seja de uma empresa X e o sistema de potência, seja um modelo fornecido por uma empresa Y.

As situações acima evidenciam três questões gerais: a disponibilidade de recurso, a colaboração e confidencialidade de dados durante simulações computacionais. Neste contexto, evidencia-se uma primeira hipótese de pesquisa: seria possível realizar uma simulação de transitórios eletromagnéticos de forma distribuída, com dois subsistemas A e B localizados em computadores distintos?

As consequências decorrentes de uma resposta positiva a esta pergunta são inúmeras. Para o caso do *ONS*, seria possível ter um sistema completo online análogo ao *SIN*, onde os projetistas do setor elétrico poderiam empreender conexões às barras do *SIN* para fazer estudos relacionados aos projetos de integração de plantas elétricas a este sistema. Para o caso das universidades, seria possível promover trabalhos colaborativos e análises de diferentes cenários de pesquisa; compartilhando recursos computacionais e conhecimentos adquiridos entre as universidades. Para o caso das empresas, onde a propriedade intelectual e confidencialidade significa dinheiro, uma resposta positiva para esta questão seria a proteção dos direitos autorais para a empresa e liberdade para projetos em consórcio; tema muito comum atualmente.

Considerando esse conjunto de necessidades e desafios, os pesquisadores envolvidos nesta tese de doutorado decidiram enveredar por este tema de pesquisa.

### 1.3 Objetivos do trabalho

O principal objetivo deste trabalho é o desenvolvimento de uma Arquitetura Computacional envolvendo hardware e software, capaz de simular transitórios eletromagnéti-

cos em grandes sistemas de energia utilizando computação paralela e distribuída. E com interação sob-demanda entre os subsistemas envolvidos. Para tanto, será feita uma integração entre os conceitos de paralelização computacional e a estrutura de um software para cálculo de transitórios eletromagnéticos. A ideia básica é estender as potencialidades de softwares já estabelecidos para execução no ambiente de computação paralela.

De forma a alcançar o objetivo geral, este trabalho possui os seguintes objetivos específicos, os quais definem também a base metodológica para a sua execução:

- Avaliação dos requisitos para aplicação de computação paralela à solução de determinado problema.
- Considerando o item acima, avaliar se o cálculo de transitórios eletromagnéticos se enquadra nesse conjunto de problemas.
- Supondo que o problema abordado nesta tese é paralelizável, o próximo passo é a identificação dos gargalos técnicos envolvidos na integração de softwares *"off the shelf"* à arquitetura de computação paralela.
- Definição de qual software será utilizado para o desenvolvimento da arquitetura computacional.
- Identificação do processo de comunicação multimáquina entre os softwares para uma arquitetura distribuída.
- Definição de protocolos de comunicação em rede, linguagem de programação e aplicativos para o projeto;
- Estabelecimento de uma metodologia de funcionamento da Arquitetura proposta, considerando as peculiaridades do software escolhido e características da modelagem elétrica para cálculo de transitórios;
- Desenvolvimento da Arquitetura Computacional de simulação (simulador) a partir da adaptação da estrutura do software *"off the shelf"* escolhido.

## 1.4 Contribuições da tese de doutorado

As principais contribuições deste trabalho são:

- Desenvolvimento de uma Arquitetura Computacional, Distribuída e Colaborativa para cálculo de transitórios eletromagnéticos que permite interação sob-demanda.
- Comparativo entre a arquitetura computacional proposta com uma arquitetura síncrona.

- Análise dos impactos da latência em simulações geograficamente distribuídas.
- Análise de uma estratégia de computação paralela local associada à computação distribuída para minimização dos efeitos da latência no tempo total de simulações distribuídas.
- Apresentação de uma ferramenta pedagógica denominada Mapa de Complexidade e um parâmetro para quantificar o grau de complexidade do projeto. Esses dois servem para auxiliar pesquisadores no desenvolvimento de Arquiteturas Computacionais como a apresentada nesta tese.
- Artigos publicados em revista e anais de congressos nacionais e internacionais.

## 1.5 Organização da tese de doutorado

Este trabalho está organizado como apresentado nos tópicos a seguir:

- No Capítulo 1 é feita uma breve explanação sobre o contexto em que a tese de doutorado está envolvida. O Capítulo 2 apresenta, inicialmente, uma fundamentação teórica sobre o cálculo de transitórios eletromagnéticos, evidenciando quais características dele que tornam possível a aplicação de computação paralela a este tipo de cálculo. Em seguida, faz uma breve explanação de conceitos sobre a computação paralela e distribuída, mostrando como aplicá-la a determinado problema e modelos de programação. Ao final, apresenta o assunto sobre comunicação entre processos, destacando modelos e protocolos utilizados para troca de informações em cálculos paralelos.
- O Capítulo 3 faz uma revisão bibliográfica sobre os trabalhos na literatura mundial que unem cálculo de transitórios eletromagnéticos e computação paralela; evidenciando os gargalos e oportunidades que surgem como consequência dessa união.
- O Capítulo 4 apresenta o desenvolvimento da arquitetura computacional distribuída e colaborativa com interação sob-demanda, principal contribuição desta tese de doutorado.
- O Capítulo 5 apresenta testes computacionais com a arquitetura proposta e a compara com uma síncrona; faz uma análise do efeito da latência em simulações geograficamente distribuídas; apresenta simulações de um grande sistema de potência, 3200 barras, para mostrar oportunidades fornecidas pelo cálculo distribuído. E, ao final, faz uma discussão em torno dos resultados apresentados nas simulações.
- No Capítulo 6 são apresentadas as conclusões finais do trabalho e propostas de continuidade.

## 2 Fundamentação teórica

Neste capítulo, serão apresentados os tópicos teóricos necessários à compreensão do trabalho apresentado nesta tese, iniciando-se com uma revisão sobre a forma clássica de resolução do cálculo de transitórios eletromagnéticos. Dentro desse tema, apresentam-se as características desse problema que o fazem passível de paralelização. Em seguida, uma breve revisão sobre computação paralela é feita, destacando os prerequisites para a aplicação juntamente com as oportunidades trazidas por ela. E, por fim, um tópico sobre comunicação entre processos é apresentado.

### 2.1 Cálculo de transitórios eletromagnéticos

O cálculo de transitórios eletromagnéticos em sistemas de energia sofreu um grande avanço após H. W. Dommel ter proposto um programa digital para este tipo de problema. Naquela época, o autor implementou um simulador fazendo a discretização de cada uma das equações que descrevem os componentes da rede elétrica e, posteriormente, estabelecendo um equivalente Norton para estes elementos, baseando-se na técnica de integração trapezoidal[6].

De forma geral, ficaram definidas três tarefas básicas para o programa de cálculo transitório eletromagnéticos do último autor: i) definir o equivalente Norton para cada um dos elementos da rede, ii) construir um sistema matricial de equações nodais, composto pela matriz de condutância, o vetor de tensões e vetor de correntes instantâneos e iii) atualizar, a cada passo de cálculo, o vetor de correntes históricas, que retém a informação sobre o estado do sistema em um tempo passado. Define-se, portanto, um método numérico para a solução das equações do sistema a cada passo de tempo.

Embora a proposta de H. W. Dommel seja um dos trabalhos mais conhecidos sobre cálculo de transitórios eletromagnéticos em sistemas de energia. Há também na literatura trabalhos anteriores ao dele sobre o assunto, por exemplo, os trabalhos de (Barthold e Carter) e (Uram e Miller), [14] e [15].

Para facilitar a leitura e evitar repetições, utilizaremos no texto a sigla cálculo de EMT, no lugar de transitórios eletromagnéticos.

A seguir serão apresentados os circuitos equivalentes Norton de componentes elétricos comumente utilizados em sistemas de energia.

### 2.1.1 Resistência, indutância e capacitância

No trabalho de Dommel [6], foi feita a apresentação dos equivalentes discretos de vários componentes elétricos, sendo o método trapezoidal utilizado para a determinação deles. O modelo de uma resistência entre dois terminais é a própria resistência. No caso de indutâncias e capacitâncias concentradas, ambas são modeladas por uma resistência e uma fonte de corrente histórica em paralelo. Este último elemento guarda a história do circuito em um passo de cálculo anterior.

Na Figura 2.1, apresenta-se o modelo equivalente para indutância, o qual utiliza uma resistência  $R_L$  e uma fonte de corrente  $I_L(t - \Delta t)$ .

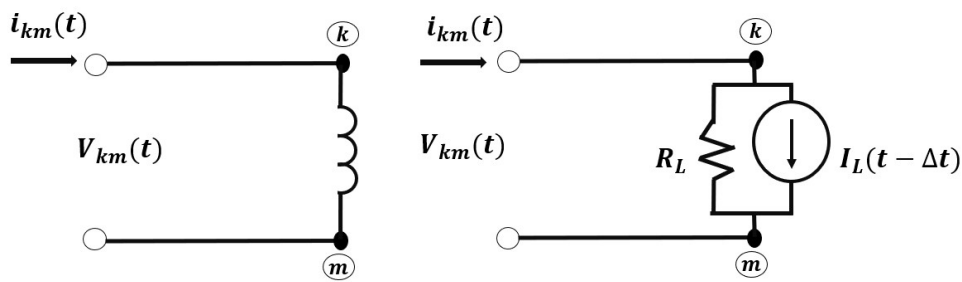


Figura 2.1 – Equivalente Norton para indutância.

Onde:

$$R_L = \frac{2L}{\Delta t} \quad (2.1)$$

$$I_L(t - \Delta t) = \frac{v_{km}(t - \Delta t)}{R_L} + i_{km}(t - \Delta t) \quad (2.2)$$

De maneira análoga, é apresentado o modelo para a capacitância conforme figura 2.2. com uma resistência  $R_C$  e uma fonte de corrente  $I_C(t - \Delta t)$ .

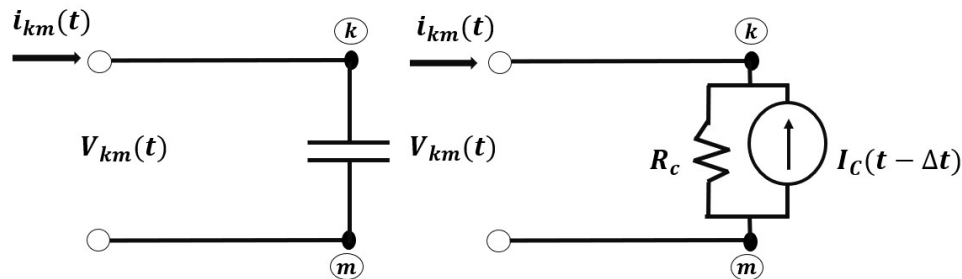


Figura 2.2 – Circuito equivalente para capacitância.

Onde:

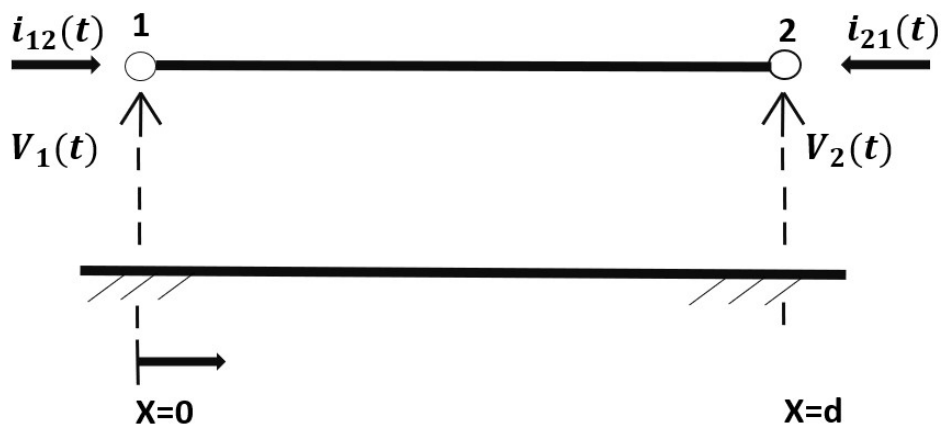
$$R_C = \frac{\Delta t}{2C} \quad (2.3)$$



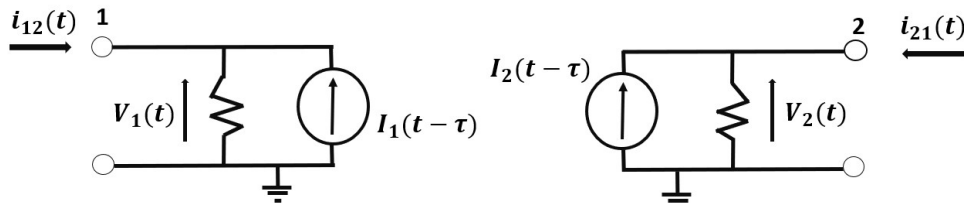
$$I_C(t - \Delta t) = \frac{v_{km}(t - \Delta t)}{R_C} + i_{km}(t - \Delta t) \quad (2.4)$$

### 2.1.2 Linha de transmissão

Neste tópico apresenta-se um dos modelos de linha mais simples que existem, proposto em [6], no qual os parâmetros dela não dependem da frequência e perdas são desconsideradas. Uma ilustração da linha e seu respectivo equivalente Norton são apresentados na Figura 2.3, cujo equacionamento no domínio do tempo se estabelece como nas equações 2.5 e 2.6:



(a) Ilustração de linha de transmissão.



(b) Equivalente Norton para linha de transmissão.

Figura 2.3 – Modelo de linha sem perdas.

$$i_{12}(t) = Y_c v_1(t) - I_1(t - \tau) \quad (2.5)$$

$$i_{21}(t) = Y_c v_2(t) - I_2(t - \tau) \quad (2.6)$$

Em que:

$$I_1(t - \tau) = Y_c v_2(t - \tau) + i_{21}(t - \tau) \quad (2.7)$$

$$I_2(t - \tau) = Y_c v_1(t - \tau) + i_{12}(t - \tau) \quad (2.8)$$

Onde:  $\tau$  é o tempo de trânsito na linha e  $Y_c$  a admitância característica da linha.

Note que este modelo considera a linha como dois blocos desacoplados, os quais podem ser representados por uma fonte de corrente incidente em cada nó das extremidades da linha e uma impedância para a terra. As fontes de corrente  $I_1$  e  $I_2$  nas equações 2.7 e 2.8, são termos históricos que guardam as características das extremidades opostas àquelas nas quais são injetadas.

De 1969 até hoje, grande evolução ocorreu em termos de modelos de linha para transitórios eletromagnéticos, principalmente pelo fato de que os modelos atuais conseguem representar a natureza distribuída dos parâmetros da linha, a variação destes últimos com a frequência, os efeitos pelicular (em inglês: “skin”) e de proximidade, e perdas elétricas. A seguir serão apresentados alguns dos modelos de linhas mais recentes na literatura.

#### 2.1.2.1 Linhas de transmissão com parâmetros dependentes da frequência.

As linhas de transmissão podem ser modeladas de diferentes formas. A complexidade do modelo delas está relacionada ao quanto de precisão e eficiência se deseja alcançar em determinado estudo de caso no qual a linha se faz presente. No tocante aos modelos que avaliam a natureza distribuída da linha, eles podem ser classificados de duas maneiras: a parâmetros concentrados, sendo representados por indutores, capacitores e resistores, muitas vezes, conectados em cascata de seções; muito recomendados para soluções de regime permanente. A outra forma seria a parâmetros distribuídos, neste caso a onda que se propaga na linha sofrerá atenuações ao longo da propagação e existirá um atraso entre sinais de tensão e corrente nos terminais opostos dessa linha [16].

Para a dependência dos parâmetros da linha com a frequência, há duas situações possíveis: a primeira, na qual os parâmetros da linha são calculados em apenas uma frequência, tornando-se assim um modelo limitado; e outra, na qual se considera a dependência dos parâmetros da linha com a frequência, os quais se apresentam mais adequados ao cálculo de transitórios eletromagnéticos.

Para o caso das linhas polifásicas, existem ainda duas classes de modelos especiais. Aqueles em que a modelagem se faz no domínio dos modos, ou seja, usam a técnica de transformação modal, em que um número  $p$  de fases de uma linha polifásica podem ser desacopladas em  $p$  linhas monofásicas independentes; indicados para modelar linhas de transmissão aéreas. Para as situações em que se exige um modelo sem restrição da geometria e natureza da linha, o modelo no domínio das fases é o mais apropriado. Neste, a dinâmica da linha é determinada diretamente no domínio das fases. Uma discussão detalhada e um comparativo dos principais modelos de linha da literatura também podem ser encontrados em Watson et al.[12] e Junior et al. [17].

Os modelos que consideram a dependência dos parâmetros da linha com a frequência são mais complexos e exigem maior esforço computacional. Para entender a complexidade de tais modelos, considere as equações que descrevem o comportamento da tensão e corrente em um determinado ponto  $x$  de uma linha de comprimento  $d$ , com pontos terminais definidos como 1 e 2. São apresentadas a seguir:

$$\frac{d}{dx}[V(x, \omega)] = [Z(\omega)][I(x, \omega)] \quad (2.9)$$

$$\frac{d}{dx}[I(x, \omega)] = [Y(\omega)][V(x, \omega)] \quad (2.10)$$

Onde:  $[V(x, \omega)]$  é o vetor de tensão,  $[I(x, \omega)]$  vetor de corrente,  $[Z(x, \omega)]$  a matriz de impedância série da linha e  $[Y(x, \omega)]$  matriz de admitância.

Partindo das equações 2.9 e 2.10, chega-se a um outro conjunto de equações, 2.11 e 2.12, as quais descrevem o comportamento do perfil de tensão e corrente nos terminais dessa mesma linha de transmissão.

$$[Y_c(\omega)][V_2(x, \omega)] - [I_2(x, \omega)] = e^{-[\Gamma_i(\omega)d]} \{ [Y_c(\omega)][V_1(\omega)] + I_1(\omega) \} \quad (2.11)$$

$$[Y_c(\omega)][V_1(x, \omega)] - [I_1(x, \omega)] = e^{-[\Gamma_i(\omega)d]} \{ [Y_c(\omega)][V_2(\omega)] + I_2(\omega) \} \quad (2.12)$$

Sendo:

$$[Y_c(\omega)] = (\sqrt{Y(\omega)Z(\omega)})^{-1}[Y_c(\omega)] \text{ a matriz de admitância da linha.}$$

$$|\Gamma_i(\omega)| = \sqrt{|P(\omega)|^t}, \text{ com } |P(\omega)|^t = Y(\omega)Z(\omega).$$

Neste contexto, surge um termo muito importante na modelagem de linhas de transmissão, o qual é denominado na literatura como fator de propagação ou função de propagação.

$$[A(\omega)] = e^{-[\Gamma_i(\omega)d]} \quad (2.13)$$

Usando a equação 2.13 e rearranjando os termos das equações 2.11 e 2.12, é possível chegar na seguinte formulação para o comportamento das correntes que entram nos terminais da linha de transmissão.

$$[I_1(\omega)] = [Y_c(\omega)][V_1(\omega)] + [I_{1hist}(\omega)] \quad (2.14)$$

$$[I_2(\omega)] = [Y_c(\omega)][V_2(\omega)] + [I_{2hist}(\omega)] \quad (2.15)$$

Onde:

$$[I_{1hist}(\omega)] = [A(\omega)]\{[Y_c(\omega)][V_2(\omega)] + [I_2(\omega)]\} \quad (2.16)$$

$$[I_{2hist}(\omega)] = [A(\omega)]\{[Y_c(\omega)][V_1(\omega)] + [I_1(\omega)]\} \quad (2.17)$$

Através das equações 2.14 a 2.17 é possível desenvolver um circuito equivalente para linhas polifásicas no domínio da frequência.

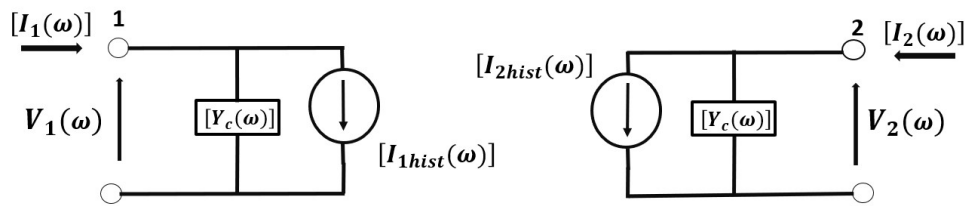


Figura 2.4 – Circuito equivalente da linha de transmissão no domínio da frequência.

A obtenção do fator de propagação  $[A(\omega)]$  é um dos pontos cruciais na modelagem de uma linha. E, como já se sabe, não é uma tarefa fácil. Os modelos de linhas de transmissão disponíveis na literatura têm diferentes propostas para o cálculo daquele fator e, também, para a admitância característica  $[Y_c(\omega)]$ . A determinação desses dois parâmetros é absolutamente necessária para o estabelecimento das equações, 2.14 e 2.15, no domínio do tempo, o que requer o cálculo de convoluções, já que os produtos de matrizes, presentes nestas equações, estão no domínio da frequência.

Marti et al. afirma que o tempo de computação do transitório eletromagnético em um sistema de energia com parâmetros de linha dependentes da frequência é 10 a 30% maior que aquele com parâmetros concentrados. Esse mesmo autor é conhecido por apresentar a comunidade científica um dos modelos de linha mais conhecidos na literatura, o qual será discutido no tópico seguinte [18].

### 2.1.2.2 Modelagem de linha de transmissão no domínio modal

A modelagem de linhas de transmissão no domínio dos modos requer uma transformação de coordenadas sobre as equações que descrevem o comportamento das tensões e correntes na linha. Para tanto são utilizadas matrizes de transformações modais de tensão e corrente,  $[T_v(\omega)]$  e  $[T_i(\omega)]$ . A aplicação dessas matrizes aos vetores de tensão e corrente faz com que um sistema no domínio das fases, com  $n$  destas, seja representado em outro sistema de coordenadas, domínio modal, por  $n$  sistemas monofásicos. As equações

de propagação de onda numa linha de transmissão, no domínio da frequência, podem ser escritas da seguinte forma:

$$\frac{d^2}{dx^2}[V(x, \omega)] = [T_v(\omega)][\lambda(\omega)][T_v(\omega)]^{-1}[V(x, \omega)] \quad (2.18)$$

$$\frac{d^2}{dx^2}[I(x, \omega)] = [T_i(\omega)][\lambda(\omega)][T_i(\omega)]^{-1}[I(x, \omega)] \quad (2.19)$$

Onde  $[\lambda(\omega)]$  é uma matriz diagonal cujos elementos não nulos são autovalores de  $[Y(\omega)][Z(\omega)]$ .

A obtenção das equações de onda desacopladas é feita premultiplicando as equações 2.18 e 2.19, pelas matrizes de transformação modal.

$$\frac{d^2}{dx^2}([T_v(\omega)]^{-1}[V(x, \omega)]) = [\lambda(\omega)][T_v(\omega)]^{-1}[V(x, \omega)] \quad (2.20)$$

$$\frac{d^2}{dx^2}([T_i(\omega)]^{-1}[I(x, \omega)]) = [\lambda(\omega)][T_i(\omega)]^{-1}[I(x, \omega)] \quad (2.21)$$

O que resulta, finalmente, em equações de ondas de propagação no domínio modal.

$$\frac{d^2}{dx^2}([V_{mod}(x, \omega)]) = [\lambda(\omega)][V_{mod}(x, \omega)] \quad (2.22)$$

$$\frac{d^2}{dx^2}([I_{mod}(x, \omega)]) = [\lambda(\omega)][I_{mod}(x, \omega)] \quad (2.23)$$

A solução para estas duas últimas equações estabelece um novo conjunto de equações desacopladas para cada um dos modos  $\mu$  do novo sistema. Sendo representadas como:

$$I_{mod_1}^\mu(\omega) = Y_{c_{mod}}^\mu(\omega)V_{mod_1}^\mu(\omega) + I_{mod_1_{hist}}^\mu(\omega) \quad (2.24)$$

$$I_{mod_2}^\mu(\omega) = Y_{c_{mod}}^\mu(\omega)V_{mod_2}^\mu(\omega) + I_{mod_2_{hist}}^\mu(\omega) \quad (2.25)$$

Nas quais,

$$I_{mod_1_{hist}}^\mu(\omega) = -A_{mod}^\mu(\omega)[Y_{c_{mod}}^\mu(\omega)V_{mod_2}^\mu(\omega) + I_{mod_2}^\mu(\omega)] \quad (2.26)$$

$$I_{mod_2_{hist}}^\mu(\omega) = -A_{mod}^\mu(\omega)[Y_{c_{mod}}^\mu(\omega)V_{mod_1}^\mu(\omega) + I_{mod_1}^\mu(\omega)] \quad (2.27)$$

Até o momento, foi feito o desenvolvimento do modelo de linha no domínio da frequência. A conversão das equações de tensão e corrente para o domínio do tempo exige três passos essenciais:

- Síntese das matrizes de admitância característica modais,  $Y_{c_{mod}}^\mu(\omega)$ , as quais, para os modelos estudados neste trabalho, é feita por meio de funções racionais no plano  $s$ ;
- Síntese de matrizes, para cada modo  $\mu$ , da função de propagação  $[A(\omega)]$ . Isto é feito por meio função racional  $R(s)$ . Onde  $A_{approx_{mod}}^\mu(\omega) = R(s)e^{-s\tau_{min}^\mu}$ . E  $R(s)$  é uma função racional e  $\tau_{min}^\mu$  é o tempo de trânsito da onda mais veloz para o modo  $\mu$ .
- Aproximação das matrizes de transformação  $[T_v(\omega)]$  e  $[T_i(\omega)]$  por meio da síntese de funções racionais no plano  $s$ .

Após o cálculo das aproximações para matrizes de admitância modal, funções de propagação modais e matrizes de transformação, o próximo passo é converter as tensões e correntes modais para o domínio das fases. Mas, antes disso, deve-se solucionar as equações 2.24 a 2.27 para o domínio do tempo.

$$i_{mod_1}^\mu(t) = y_{c_{mod}}^\mu(t) * v_{mod_1}^\mu(t) + i_{mod_1hist}^\mu(t) = i_{1y}^\mu(t) + i_{mod_1hist}^\mu(t) \quad (2.28)$$

$$i_{mod_2}^\mu(t) = y_{c_{mod}}^\mu(t) * v_{mod_2}^\mu(t) + i_{mod_2hist}^\mu(t) = i_{2y}^\mu(t) + i_{mod_2hist}^\mu(t) \quad (2.29)$$

Em que:

$$i_{mod_1hist}^\mu(t) = -r_{mod}^\mu(t) * [i_{2y}^\mu(t - \tau_{min}^\mu) + i_{mod_2}^\mu(t - \tau_{min}^\mu)] \quad (2.30)$$

$$i_{mod_2hist}^\mu(t) = -r_{mod}^\mu(t) * [i_{1y}^\mu(t - \tau_{min}^\mu) + i_{mod_1}^\mu(t - \tau_{min}^\mu)] \quad (2.31)$$

Vale ainda ressaltar que  $r_{mod}^\mu(t)$  é um função exponencial no tempo obtida a partir da função racional  $R_{mod}^\mu(s)$ . Os parâmetros das equações 2.30 e 2.31 são termos históricos.

Para calcular as tensões e correntes de fase, basta utilizar as matrizes de transformações modais como apresentado nas equações 2.32 e 2.33.

$$[v(t)] = [T_v(t)] * [v_{mod}(t)] \quad (2.32)$$

$$[i(t)] = [T_i(t)] * [i_{mod}(t)] \quad (2.33)$$

Após realizado o equacionamento que leva aos vetores de tensão e corrente no domínio do tempo, pode-se entender com maior facilidade a contribuição que [18] forneceu à modelagem de linhas de transmissão no domínio modal. No trabalho proposto por este autor, os elementos das matrizes,  $[Z_{c_{mod}}(\omega)] = [Y_{c_{mod}}(\omega)]^{-1} [A_{mod}(\omega)]$  são aproximados

por funções polinomiais racionais, cuja metodologia é conhecida por método assintótico de bode. Um ponto importante, que vale a pena ser ressaltado, é que a matriz de transformação modal é considerada real e constante com a frequência. Esse quesito, simplifica a transição entre os domínios de fase e de modos, pois evita o cálculo de convoluções. No entanto, implica em limitações na modelagem da linha de transmissão, para casos em que a geometria do sistema requer uma matriz de transformação dependentes da frequência, como por exemplo: em linhas aéreas desbalanceadas, assimétricas e cabos subterrâneos. Para estes casos, o modelo de Morched et. al.[19] é suficiente e adequado. A seguir é feita uma apresentação dele.

### 2.1.2.3 Modelagem da linha de transmissão no domínio de fases

Os modelos de linhas de transmissão no domínio de fases são aqueles em que o comportamento da linha é calculado diretamente no domínio de fases, portanto, não há necessidade de transição entre este domínio e o modal. Os modelos mais recentes conseguem modelar qualquer linha de transmissão considerando a natureza distribuída dela e parâmetros dependentes da frequência, não há restrição quanto a geometria das linhas. Dentre estes, apresenta-se aqui o modelo de Morched, “Universal Line Model”[20].

Assim como apresentado anteriormente, sabe-se que um dos pontos principais no uso das linhas de transmissão, em cálculo de transitórios eletromagnéticos, é a solução das matrizes de admitância  $[Y_c(\omega)]$  e função de propagação  $[A(\omega)]$ . Morched et al.[19] apresenta um modelo de linha de transmissão que usa o método de ajuste vetorial (Vector Fitting) para obter aproximações racionais para os elementos das matrizes da função de propagação e admitância no domínio das fases.

O modelo começa com a aproximação por funções racionais dos elementos da matriz de propagação no domínio modal, conforme a equação 2.34.

$$A_{mod}^{\mu}(\omega) \cong R_{mod}^{\mu}(\omega)e^{-j\omega\tau_{min}^{\mu}} \quad (2.34)$$

Em que:

- $A_{mod}^{\mu}(\omega) \cong e^{-a^{\mu}(\omega)d}e^{-j\beta^{\mu}(\omega)d}$ , modo  $\mu$  da matriz fator de propagação modal, com módulo  $e^{-a^{\mu}(\omega)d}$  e fase  $-\beta^{\mu}(\omega)d$  na frequência;
- $R^{\mu}(\omega)$  = modo  $\mu$  da função racional de fase mínima;
- $\tau_{min}^{\mu}$  = tempo de propagação (ou de trânsito) da mais veloz componente de frequência de modo  $\mu$ .

Neste modelo é necessário determinar o tempo de trânsito, em uma etapa anterior ao processo de ajuste para cada um dos modos do fator de propagação, pois ele não considera que o tempo de trânsito é uma variável do processo de ajuste.

$$\tau_{min}^{\mu} = \frac{\angle R_{mod}^{\mu}(\omega)}{\omega} + \frac{\beta^{\mu}(\omega)d}{\omega} \quad \text{com } \omega = \Omega \quad (2.35)$$

Em que:

- $\angle R_{mod}^{\mu}(\omega)$  = é o ângulo de fase da função de fase mínima na frequência  $\omega$ ;
- $\Omega$  = a frequência mais alta da faixa de interesse.

Após serem calculados os tempos de propagação de cada modo, calculam-se as rotações de fase relacionadas a estes através da equação 2.36.

$$e^{s\tau^{\mu}} A_{mod}^{\mu}(s) \cong \sum_{m=1}^N \frac{c_m^{\mu}}{s - p_m^{\mu}} \quad \text{para } s = j\omega \quad (2.36)$$

Onde:

- $c_m^{\mu}$  = resíduos da função racional aproximada para o modo  $\mu$ ;
- $p_m^{\mu}$  = polos da função racional aproximada para o modo  $\mu$ ;

Um ganho de eficiência computacional é alcançado pelo autor através do agrupamento dos modos com tempos de propagação muito próximos. Após calculados os polos no domínio modal, os elementos da função de propagação no domínio das fases são calculados através da equação 2.37.

$$A_{ij}(s) \cong \sum_{\mu=1}^n [(\sum_{m=1}^{N^{\mu}} \frac{c_{m-ij}^{\mu}}{s - p_m^{\mu}}) e^{-s\tau^{\mu}}] \quad (2.37)$$

Na qual,

- $A_{ij}(s)$  = elemento  $(i, j)$  da matriz  $[A(s)]$  no domínio das fases;
- $c_m^{\mu}$  = resíduos da função racional aproximada para o modo  $\mu$ ;
- $p_m^{\mu}$  = polos da função racional aproximada para o modo  $\mu$ ;
- $\tau^{\mu}$  = tempo de propagação para o modo  $\mu$ ;
- $N^{\mu}$  = ordem da aproximação racional para o modo  $\mu$ ;



- $n$  = número de modos (ou grupos, caso haja agrupamento de modos).

No que diz respeito ao cálculo da matriz de admitância característica  $Y_c(\omega)$  elementos são aproximados diretamente no domínio de fases e com a mesma técnica de compartilhamento dos polos.

$$Y_{c-ij}(s) \cong d_{ij} + \sum_{k=1}^N \frac{c_{k-ij}}{s - p_k}, \quad \text{para } s = j\omega \quad (2.38)$$

Em que:

- $Y_{c-ij}(s)$  = elemento  $(i, j)$  da matriz  $[Y_c(s)]$  no domínio das fases;
- $p_k$  = pólos da função aproximada para o elemento  $(i, j)$ ;
- $N$  = ordem da aproximação racional para o elemento  $(i, j)$ ;
- $c_{k-ij}$  = resíduo da função racional para o elemento  $(i, j)$ ;

O modelo proposto por Morched et. al. é a base para o modelo de linha no domínio das fases apresentado no software *PSCAD (Power Systems Computer Aided Design)*. As deduções das equações foram baseadas em [16], [19].

### 2.1.3 Elementos não lineares

Em sua concepção original, o programa de cálculo de transitórios eletromagnéticos, proposto por Dommel[6], foi implementado para lidar com redes lineares. Entretanto, muitos dos componentes presentes nas rede elétricas possuem um comportamento não linear, por exemplo os disjuntores, para-raios, saturação e histerese nos transformadores. Dommel et. al. [21] apresenta uma discussão sobre como modelar não-linearidades no programa de cálculo de transitórios. Em Gole et. al. [22] e Watson et al.[12] também é possível encontrar uma abordagem sobre esse assunto.

Dommel et. al. [21] apresenta três maneiras de se realizar a modelagem de não linearidades: representação por fonte de corrente, método de compensação e representação linear por partes. No método de compensação, é usado o teorema da substituição e o princípio da superposição. Neste, realiza-se o cálculo de transitórios sem o componente não-linear; depois a contribuição deste último é feita conforme as equações 2.39 e 2.40:

$$v_{km} = v_{km}(0) - r_{thevenin} i_{km} \quad (2.39)$$

Com as características do componente não linear dadas por:

$$v_{km} = f(i_{km}, \frac{di_{km}}{dt}, t, \dots) \quad (2.40)$$

O parâmetro  $v_{km}(0)$  representa a solução da rede para os nós  $k$  e  $m$  considerando-a sem o componente não-linear, o parâmetro  $r_{thevenin}$  é a resistência equivalente de Thévenin. A solução das equações 2.39 e 2.40 é feita através de um método iterativo, por exemplo, Newton-Raphson.

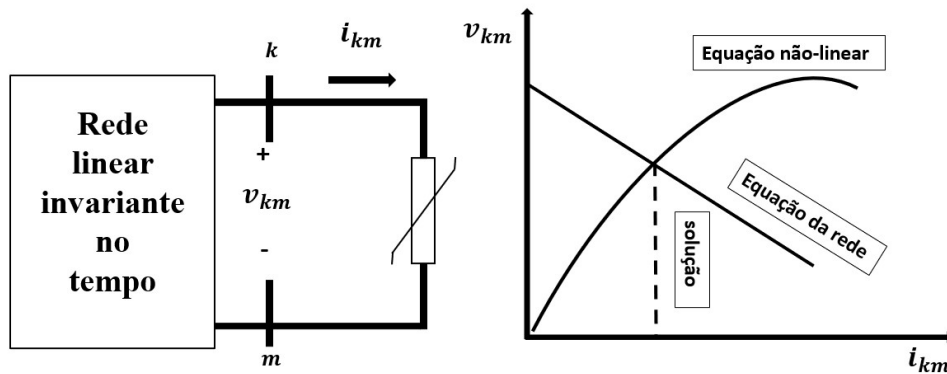


Figura 2.5 – Método de compensação.

O método de compensação pode ser aplicado em redes de energia com diversos elementos não-lineares. No entanto, esta metodologia apenas poderá ser usada com um único elemento por nó; o que representa uma limitação para o modelo [22]. Um ponto importante a ser ressaltado é o aumento do esforço computacional para o cálculo de transitórios ao ter a presença desses componentes. Note que, para cada componente não linear presente no sistema de energia, a aplicação do método de compensação significa o uso do método de Newton-Raphson a cada passo de cálculo.

#### 2.1.4 Transformadores

A modelagem de transformadores para estudos de transitórios possui grandes desafios. Isso se deve às peculiaridades construtivas dos vários dispositivos disponíveis comercialmente. Dentre estas pode-se citar os núcleos, tipo de enrolamentos e materiais. Outra questão de destaque, é a natureza não linear e dependente da frequência apresentada pelo transformador. Esse último quesito define a necessidade de se representar corretamente algumas características físicas deste dispositivo: indutâncias próprias e mútuas; fluxos de dispersão; saturação no núcleo; efeitos capacitivos; perdas por correntes de Foucault. A faixa de frequência em que o transformador estará submetido define quais características são mais ou menos necessárias [23].

O desenvolvimento de um modelo que seja preciso em uma larga faixa de frequências é uma tarefa difícil [24]. Os principais tipos de modelos são Estrela, Matriciais e

Topológicos, adequados para transitórios lentos, os quais são calculados através de testes padronizados e características construtivas. Já os modelos para altas frequências necessitam de medidas não padronizadas para obtenção dos parâmetros dele.

BRANDWAJN et. al. [25] apresenta um modelo relativamente preciso para transitórios em frequências abaixo de  $1kHz$ , o qual é do tipo matricial e denominado BCTRAN. A formulação do modelo parte de uma equação matricial de estado estável através da qual pode se representar o comportamento de um transformador multifásico com  $n$  enrolamentos, assim como em 2.41.

$$[V] = [Z][I] \quad (2.41)$$

No cálculo de transitórios, reescreve-se 2.41 como:

$$[v] = [R][i] + L\left[\frac{di}{dt}\right] \quad (2.42)$$

Os parâmetros  $[R]$  e  $j\omega[L]$  são a parte real e imaginária de  $[Z]$ , obtidos a partir de testes de excitação. Nesta modelagem são considerados todos os acoplamentos fase-fase e características terminais.

Nos casos em que a corrente de excitação é desprezível, a matriz  $[Z]$  não é definida. Para esta situação, uma representação com matriz de admitância pode ser usada.

$$[I] = [Y][V] \quad (2.43)$$

A obtenção dos elementos da matriz de admitância  $[Y]$  é feita através de testes de curto circuito padrão. Uma representação no domínio do tempo, também pode ser feita para a modelagem matricial com admitância.

$$\frac{di}{dt} = [L]^{-1}[v] - [L]^{-1}[R][i] \quad (2.44)$$

A modelagem apresentada anteriormente refere-se à representação da parte linear do transformador. A inclusão do comportamento não linear desse dispositivo, como saturação e histerese, pode ser feita externamente, na forma de um indutor não linear.

O modelo BCTRAN é adequado para fenômenos que envolvem baixas frequências. No entanto, em transitórios de alta frequência, como aqueles causados por descargas atmosféricas em linhas de transmissão, desenergização ou chaveamento de cargas indutivas, uma modelagem de transformador para uma ampla faixa de frequências é necessário.

Gustavsen et. al. [26] apresenta um modelo adequado a estudo de transitórios em altas frequências. Nesta formulação, o autor faz medidas da matriz de admitância do transformador e suas relações de tensão, as quais influenciarão nos parâmetros da modelagem,

na faixa de 50 Hz a 1 MHz. Uma metodologia foi desenvolvida para realizar a aproximação dos parâmetros da matriz de admitância e se baseia na técnica “Vector Fitting” desenvolvida pelo mesmo autor. Um analisador de rede foi utilizado para o desenvolvimento do modelo, os cabos utilizados na medição possuem capacitâncias consideradas ao final da modelagem. A representação deste modelo em um programa de transitórios pode ser feita por uma matriz de condutância com uma fonte de corrente com termos históricos.

Outro modelo robusto de transformador, presente na literatura, é conhecido como *UMEC (Unified Magnetic Equivalent Circuit)*. Esta modelagem se baseia no equivalente Norton e deriva-se da análise de circuito equivalente magnético. Assim como no modelo de Gustavsen [26], a integração deste modelo ao cálculo de EMT é feita por fontes de corrente e admitâncias. A implementação deste modelo encontra-se disponível no software PSCAD [3].

### 2.1.5 Outros componentes elétricos

Em grandes sistemas de energia, muitas vezes a simulação de transitórios eletromagnéticos exige a modelagem de apenas uma parte do sistema. Neste caso, recorre-se ao conceito de Equivalentes de Rede, um procedimento que visa simplificar a modelagem com vistas a reduzir a complexidade e o tempo de computação da parte que não exige um modelo detalhado. Boaventura et. al. [27] apresentam um trabalho sobre este tema cujo foco são equivalentes de rede esparsos e robustos baseados na modelagem paramétrica no domínio do tempo.

Os elementos apresentados até aqui são apenas uma parte dos componentes mais comumente utilizados nos sistemas de potência atual. Dispositivos elétricos como máquinas rotativas, conversores de potência, sistemas de controle e proteção, para-raios, chaves, geradores eólicos, sistemas fotovoltaicos, exigem uma modelagem adequada para estudos de EMT; cuja integração à equação nodal matricial se faz através de uma fonte de tensão, fonte de corrente e matriz de admitância [12], [24].

### 2.1.6 Equação nodal para cálculo matricial

O método proposto por Dommel [6] para cálculo de transitórios possui uma característica peculiar, a qual viabiliza a aplicação de computação paralela e/ou distribuída para solucionar o sistema de equações matriciais que descreve o estado do sistema de energia. Os modelos para as linhas de transmissão, baseados em ondas viajantes e seus respectivos tempos de trânsito, permitem a separação do sistema completo em subsistemas desacoplados.

Os transitórios em cada um dos subsistemas citados podem ser calculados por diferentes núcleos nos processadores, em um mesmo ou vários computadores, independen-

temente. A ligação entre os subsistemas é feita por meio dos termos históricos relativos às ondas viajantes. O sistema da Figura 2.6 será utilizado para ilustrar a possibilidade de desacoplamento oferecida pela presença de linhas de transmissão no sistema de energia.

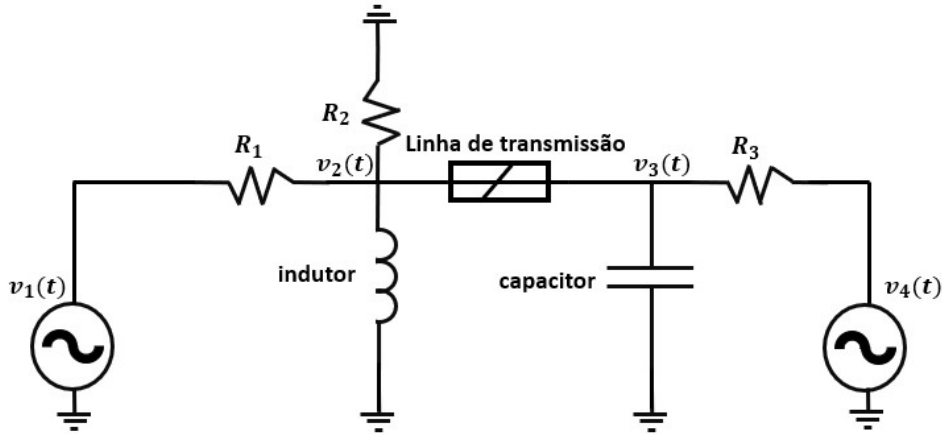


Figura 2.6 – Circuito exemplo.

A aplicação da técnica de integração trapezoidal sobre as equações diferenciais que caracterizam os elementos do circuito acima fornece a base para a determinação do circuito equivalente Norton apresentado na Figura 2.7.

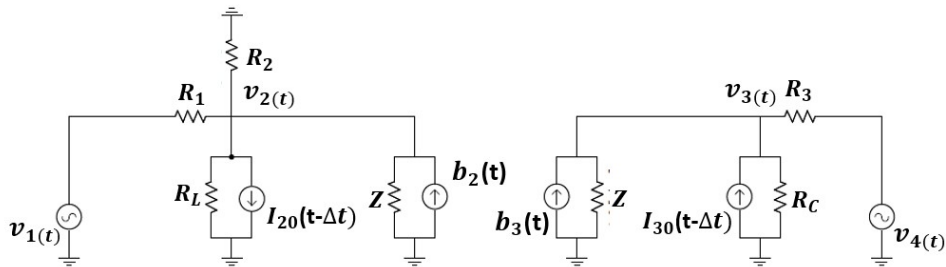


Figura 2.7 – Equivalente Norton do circuito exemplo.

A aplicação da Lei de Kirchhoff para correntes ao equivalente Norton do circuito possibilita o desenvolvimento de uma equação nodal para o sistema completo, cuja forma se apresenta como nas equações 2.45 e 2.46:

$$[i(t)] = [Y][v(t)] \tag{2.45}$$

$$\begin{bmatrix} 0 \\ b_2(t) - I_{20}(t - \Delta t) \\ b_3(t) + I_{30}(t - \Delta t) \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_L} + \frac{1}{Z} & 0 & 0 \\ 0 & 0 & \frac{1}{R_3} + \frac{1}{R_C} + \frac{1}{Z} & -\frac{1}{R_3} \\ 0 & 0 & -\frac{1}{R_3} & 0 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{bmatrix} \tag{2.46}$$

Onde:

- $[i(t)]$  é o vetor com as fontes de corrente conhecidas e as históricas;
- $[Y(t)]$  é a matriz de condutância do sistema completo;
- $v(t)$  é o vetor de tensões de nó a serem determinados no cálculo de transitórios.

Observando o equivalente Norton do circuito exemplo, é fácil perceber o desacoplamento entre as duas partes do sistema. Consequentemente, abrindo possibilidade para que o cálculo de transitórios delas seja feito em dois núcleos diferentes. No primeiro, o equacionamento será formado pela primeira e segunda linhas da equação matricial 2.46, no segundo, pela terceira e quarta linhas da equação. As informações entre as partes, em núcleos diferentes, são trocadas através das fontes de corrente históricas terminais da linha  $b_2(t)$  e  $b_3(t)$ .

A extensão de uma simulação de transitórios em mais de dois núcleos depende de quantos subsistemas forem definidos pelo pesquisador em sua modelagem; o que está intimamente relacionado à quantidade de linhas de transmissão presentes no circuito a ser modelado. No caso do circuito exemplo, apenas dois núcleos podem ser utilizados. Mas para grandes sistemas, como é o caso dos alimentadores das concessionárias de energia, a simulação deste tipo de problema pode ser feita em mais de dois núcleos.

O exemplo apresentado anteriormente é uma demonstração de que o problema envolvendo cálculo de transitórios em sistemas elétricos, com a presença de linhas de transmissão, é paralelizável. Esta afirmação é o primeiro requisito para se iniciar uma discussão sobre a aplicação de computação paralela na solução de um determinado problema. Outro ponto que merece destaque, neste pequeno exemplo, é sobre entender quando se torna interessante trocar uma computação usando um núcleo por outra paralela em dois ou mais núcleos. Note que inicia-se aqui uma busca que transpõe a teoria de cálculo de transitórios eletromagnéticos. O convite à pesquisa em computação paralela é natural, e foi feito.

### 2.1.7 Considerações parciais

Neste tópico, foi apresentado um panorama sobre o cálculo de transitórios eletromagnéticos no domínio do tempo. Apresentaram-se modelos de resistência, indutância, capacitância e linhas de transmissão, transformadores, dentre aqueles existentes na literatura, adequados ao estudo de transitórios. Além disso, foi explicitada como esses elementos são integrados a equação nodal matricial (fontes de corrente e condutâncias, principalmente); necessária à determinação das tensões de nó. Ao final, um pequeno exemplo foi introduzido para mostrar que o problema exposto é paralelizável.

Ao longo do texto, foram destacadas as características da modelagem dos componentes elétricos que justificam a necessidade de sistemas computacionais mais robustos para cálculo de EMT. Dentre aquelas, a presença de não linearidades, tais como em transformadores, máquinas rotativas, para-raios, conversores de potência; parâmetros de componentes elétricos dependentes da frequência; cálculo de integrais de convoluções; o tamanho do sistema de energia a ser modelado e a natureza dos componentes presentes no mesmo, isto vai definir o tamanho da matriz da equação nodal e a complexidade do cálculo computacional. E, por fim, o evento eletromagnético em estudo, pois ele define a necessidade de modelos elétricos mais ou menos complexos e o passo de cálculo utilizado nas modelagens.

Após o fornecimento dessa base conceitual em cálculo de transitórios eletromagnéticos, convida-se o leitor a fazer uma imersão na área ciência da computação paralela e distribuída.

## 2.2 Computação paralela e distribuída

Este tópico aborda alguns conceitos acerca do tema computação paralela. Objetiva-se com isso, responder a três questões primordiais para a aplicação desta técnica ao cálculo de um determinado problema: O que é? O porquê da utilização dela. E os gargalos técnicos envolvidos na aplicação dela à solução de algum problema.

### 2.2.1 Computação paralela

O primeiro passo para uma compreensão do tema computação paralela é entender a necessidade de utilização dela. Para isso, é preciso adquirir conhecimento sobre a computação sequencial, também conhecida como serial. Nesta última, as instruções são executadas, sequencialmente, em uma mesma unidade de processamento. A velocidade com que esta tarefa será finalizada está relacionada com a capacidade de execução deste último dispositivo. Na Figura 2.8, têm-se uma ilustração sobre a solução de um problema usando computação sequencial.

A Figura 2.8 mostra um problema denominado *EMT\_HPC*, o qual é representado por um conjunto de instruções; sendo que estas são executadas em um processador de forma sequencial. A quantidade de instruções e a complexidade de execução delas depende da natureza do problema. Este último parâmetro somado à capacidade de processamento do hardware dedicado à solução do problema vão definir o tempo total para a solução de *EMT\_HPC*; um dos maiores fatores de preocupação em se tratando de simulação computacional. Chega-se aqui, no elemento chave que conduz à aplicação de computação paralela para solucionar determinados problemas, que outrora seriam resolvidos de forma sequencial, o tempo.

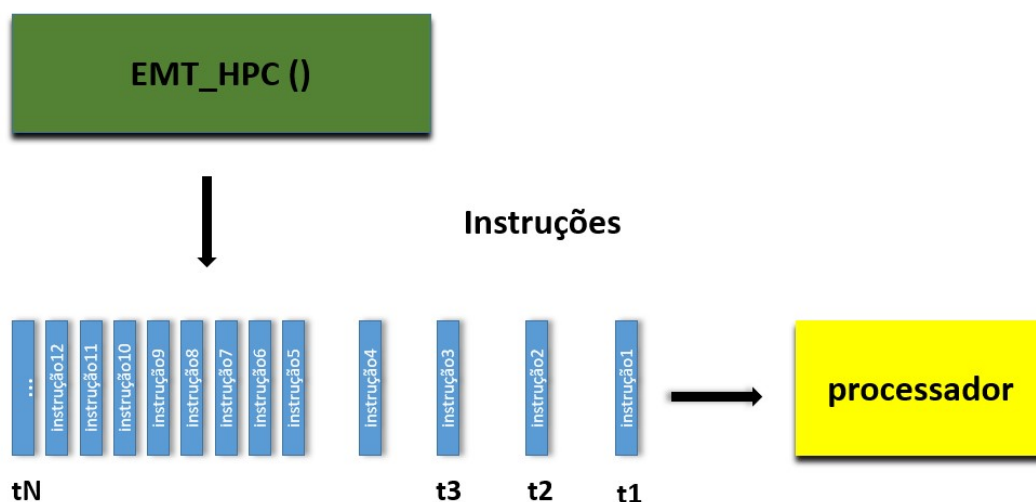


Figura 2.8 – Computação sequencial.

De acordo com Kirk et al. [28] existem três motivos principais para se utilizar a computação paralela. Para ilustrá-los, os autores apresentam o estudo de caso de uma Firma de investimentos. Na primeira situação desse estudo, a Firma precisava fazer uma análise de risco em seu portfólio de negócio em um tempo não superior a 4 horas após o encerramento dos negócios, entretanto num computador sequencial essa análise poderia gastar 200 horas. Na segunda, a Firma conseguia fazer a análise de risco em seu portfólio em um tempo aceitável, no computador sequencial, porém uma expansão na quantidade de ações poderia fazer com que o tempo da simulação, no computador sequencial, excedesse o limite de horas exigido. No terceiro, caso a Firma, que usava um modelo aproximado em sua análise de risco, quisesse utilizar um modelo mais preciso, um tempo de computação maior seria exigido, o que poderia inviabilizar a análise.

As situações apresentadas por Kirk et al. [28] no estudo de caso da Firma de investimento, ilustram três motivos para a computação paralela: o primeiro, é a solução de um determinado problema em menos tempo; o segundo, é solucionar problemas maiores em uma janela de tempo exigida para o caso proposto; o terceiro, é ter melhores soluções para determinado problema em uma quantidade tempo aceitável. Note que em todos, o ponto central é a redução do tempo de execução de um problema. Além disso, em termos de modelagem computacional, a utilização de computação paralela estava em meio a situações envolvendo grande quantidade de dados e modelos com alto nível de complexidade. Partindo deste último trabalho, definem-se, assim, três parâmetros para indicar a utilização da computação paralela: redução de tempo de simulação, tamanho e complexidade do problema.

Até agora, foram apresentados o conceito de computação sequencial e os gargalos técnicos associados a esta que motivam a demanda por uma técnica de processamento mais eficiente, que, neste trabalho, escolheu-se a computação paralela. Para esta ser in-



troduzida na execução de um problema, falta apenas a consideração de uma característica, associada a este último, que é essencial, a qual se define como a natureza de ser paralelizável; aqui conceituada como a capacidade de determinado problema ser decomposto em subproblemas.

A computação paralela pode ser entendida como a utilização de duas ou mais unidades de processamento para a resolução de um problema. Neste contexto, a solução global deste último, utilizando paralelismo, é representada pela união dos resultados individuais de cada subproblema, resolvidos, simultaneamente, nas unidades supracitadas [29], [30]. A Figura 2.9 ilustra um exemplo no qual a computação paralela é aplicada a resolução do problema  $EMT\_HPC(x, y)$  com quatro conjuntos de dados como entrada.

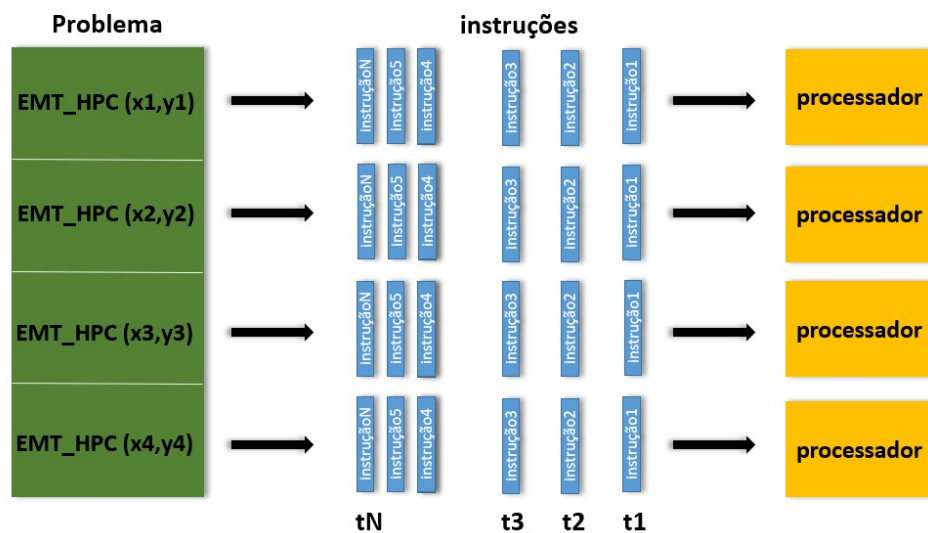


Figura 2.9 – Computação paralela.

A computação paralela pressupõe a divisão de um problema em vários processadores, o que pode ser feito em um único ou em vários computadores. Em ambas as situações, a definição de computação paralela é adequada. No entanto, para o caso em que se usa mais de um computador para o processamento paralelo, é comum utilizar o termo computação distribuída. E ao conjunto de computadores dá-se o nome “*cluster*”.

Como citado anteriormente, a aplicação de computação paralela para resolução de determinado problema, na maioria das vezes, está relacionada a redução de tempo. No entanto, é preciso maturidade de trabalho para perceber que, em algumas situações, a computação paralela pode não ser mais interessante que a computação sequencial, ou até mesmo, ser menos eficiente. Neste momento, cabe ao pesquisador da área avaliar se vale a pena ou não a utilização daquela no lugar desta última. Para tomar essa decisão com mais propriedade, é necessário o entendimento de alguns conceitos, dentre os quais destacam-se aqui: arquiteturas, linguagens de programação e algoritmos [29], [31].

Nos tópicos seguintes, serão apresentados alguns conceitos importantes quando

se trata de computação paralela, fornecendo ao leitor uma base para a compreensão e maturidade para a utilização desta técnica computacional.

### 2.2.2 Termos utilizados em computação paralela

O aprofundamento em torno da computação paralela mostra a necessidade de se compreender uma série de termos que serão recorrentes em todas as leituras relacionadas ao tema [30], [32], [33], [34]. A seguir são listados alguns destes termos:

- Computação de alto desempenho: refere-se ao uso de supercomputadores ou um conjunto de vários computadores, clusters, em tarefas que exigem grandes recursos computacionais.
- Nodo ou nó: trata-se da definição de um computador, que em computação paralela está, muitas vezes conectado a outros computadores para produzir um cluster.
- CPU (em inglês, “*central processing unit*”): é a unidade de processamento central, nela são controladas e realizadas as tarefas definidas pelo usuário do computador.
- Tarefa (em inglês, “*task*”): grupo de instruções que é executado pelo computador.
- Pipeline: é uma técnica na qual o processador é dividido em estágios distintos.
- Desempenho: relaciona-se a diminuição do tempo de resolução do problema à medida que os recursos computacionais aumentam.
- Escalabilidade: relaciona-se ao aumento do desempenho à medida que a complexidade do problema aumenta.
- Granularidade: Razão entre o tempo de computação e o tempo de comunicação.
- Aceleração (em inglês, “*speed up*”): é a relação entre os tempos de processamento serial e com paralelismo.
- Threads: são instruções de um processo ou programa a ser executada pelo processador.
- Sincronização: é um processo de coordenação entre as tarefas de um programa com base em restrições preestabelecidas. Por exemplo, em um programa que faz troca de mensagens, se aquele que envia uma mensagem puder continuar o envio antes de saber se o receptor recebeu a mensagem, o programa não tem sincronização.
- Latência: tempo mínimo para enviar um pacote de dados de um ponto a outro designado em um sistema.

- Multiprocessadores simétricos (em inglês: “*SMP*”): arquitetura de memória compartilhada em que múltiplos processadores compartilham um único espaço de endereço e todos têm igual acesso a todos os recursos.
- Kernel: núcleo do sistema operacional.
- API: Interface de programa de aplicativos (em inglês: “*Application Program Interface*”).

### 2.2.3 Modelos de programação paralela

O cálculo computacional paralelo pode ser feito através de vários modelos, dentre os quais destacam-se: Modelos de memória compartilhada, Modelos de Threads, Modelo de Passagem de Mensagem, Modelo de Dados Paralelos, Modelo *SPMD* (*Single Instruction Multiple Data*), Modelo *MPMD* (*Multiple Instruction Multiple Data*).

O modelo de memória compartilhada é aquele em que os processos compartilham um espaço de endereço comum, lendo e escrevendo nesta de maneira assíncrona. Mattson et. al, [34] cita que este modelo pode ser considerado um dos mais simples pela sua similaridade com programação de sistemas operacionais. O modelo de Threads é um subgrupo do modelo de memória compartilhada, neste, um programa pesado computacionalmente pode ser dividido em subtarefas, threads, mais leves; as quais podem ser processadas concorrentemente. É recomendado uma atenção redobrada quanto a sincronização dos dados para que não haja atualização simultânea do mesmo espaço de memória pelas subtarefas. A comunicação entre processos em uma estrutura com memória compartilhada exige um *API* (*Application Program Interface*), como exemplo têm-se o *OPENMP* (*Open Multi-Processing*) [31].

O modelo de Passagem de Mensagens é aquele no qual há comunicação entre processos para envio e recebimento de dados. Em sua concepção original, este modelo foi elaborado para sistemas computacionais com memória distribuída<sup>1</sup>. No entanto, com os avanços das tecnologias de hardware, introduzindo os multiprocessadores simétricos, hoje é possível encontrar esse modelo em arquiteturas com memórias compartilhadas e distribuídas. Para a implementação deste modelo é necessário a utilização de uma interface para comunicação entre processos, a *MPI* (*Message Passing Interface*) é um exemplo [35]

No que tange ao modelo de Passagem de Mensagens, Pacheco et al. [35] chama atenção para pontos importantes em torno da comunicação, mais especificamente em torno das linguagens de programação e bibliotecas utilizadas no algoritmo de resolução de determinado problema. Por exemplo, considere o caso onde um processo 0 está sendo

<sup>1</sup> Na arquitetura de memória distribuída, os processadores têm sua própria memória local, Isso significa que mudanças ocorridas nesta última não têm efeito sobre a memória de outros processadores na arquitetura.

executado em um nó A e o processo 1, em um nó B. E, além disso, que haverá comunicação entre ambos. Para algumas bibliotecas, o comando de envio de mensagem partindo de A para B somente enviará a mensagem de fato, após receber do segundo nó uma mensagem dizendo que está pronto para o recebimento, chama-se isso de sincronismo de comunicação. De forma diferente, existem bibliotecas que permitiriam o envio da mensagem do 0 para 1, com o conteúdo sendo armazenado em B até que 1 estivesse pronto a recebê-lo, mas sem que o processo 0 tivesse que esperar por isso.

O modelo paralelo de dados é aquele em que as tarefas atuam sobre um grupo de dados. Sendo que estes últimos podem estar organizados na forma de “array” ou “cubo”. Uma partição sobre este grupo de dados deverá ser feita, com cada uma das tarefas assumindo uma parte. Esse modelo de programação e os outros citados podem ser combinados para formar o que é classificado como Modelo híbrido. Um exemplo deste último é a combinação do Modelo de Passagem de Mensagens com o modelo de Threads. Outro exemplo que merece destaque, é a combinação de um paralelismo usando **MPI** e a estrutura **CPU ( Central Processing Unit)-GPU (Graphic Processing Unit)**. Neste caso, é importante ressaltar que haverá a necessidade de um software para fazer a comunicação entre a **CPU** e a **GPU**, como o CUDA. A **GPU** pode ser usada para problemas com demanda de computação intensa, como multiplicação de grandes matrizes. A Figura 2.10 ilustra o modelo híbrido com memória compartilhada e distribuída. Já na Figura 2.11, há um adendo a esta estrutura híbrida com a introdução da **GPU**.

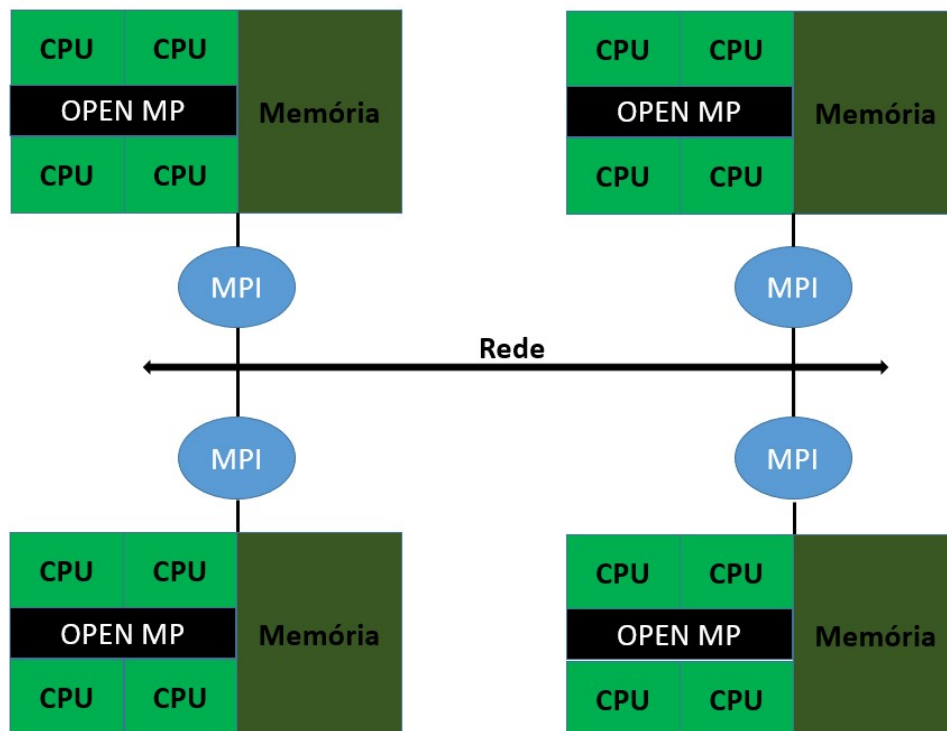


Figura 2.10 – Modelo híbrido com memória compartilhada e distribuída.

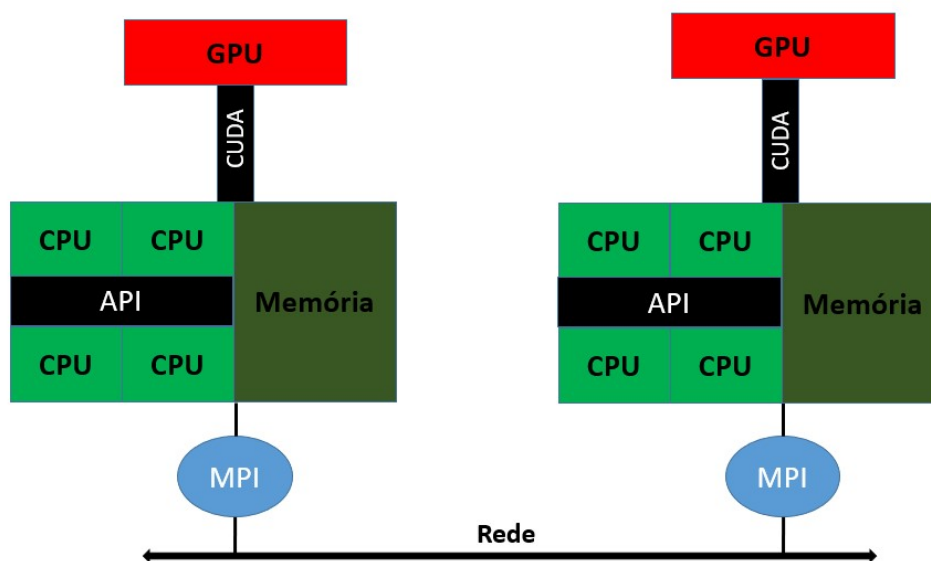


Figura 2.11 – Modelo híbrido com memória compartilhada e distribuída e adição de GPU.

O modelo **SPMD** é uma programação em que várias cópias de um mesmo programa, executadas em vários processadores, atuam em diferentes grupos de dados. Já no modelo **MPMD**, ocorre que diferentes programas, cada um destes direcionados a um processador próprio, atuam em diferentes grupos de dados. Como se sabe, a computação dos dados em cada processador é realizada simultaneamente, tanto para **SPMD** quanto **MPMD**. Note ainda, que os programas das duas estruturas supracitadas podem ter as características dos modelos anteriores a estas, conforme explica [31].

#### 2.2.4 Considerações parciais

Este tópico foi introduzido para que o leitor tivesse a compreensão de três parâmetros essenciais àqueles que objetivam trabalhar com o tema computação paralela: o que é este conceito? O porquê da aplicação do mesmo a um determinado problema. E qual o tipo de modelo de programação paralela será escolhido para o desenvolvimento de um determinado projeto. A escolha correta desses parâmetros determina a viabilidade de um projeto e o grau de desempenho que o mesmo pode alcançar. Ao longo do texto, muitos dos conceitos apresentados neste tópico serão revisitados e compreendidos seja nos exemplos da revisão de literatura ou mesmo no projeto da tese.

No tópico a seguir, o tema é sobre comunicação entre processos; uma função tão importante para a computação paralela e distribuída que sem a qual esta técnica não seria possível.

## 2.3 Comunicação entre processos

A computação paralela tem como uma de suas características o fato de tarefas de um determinado problema serem executadas concorrentemente. Para isso, diferentes unidades de processamento são solicitadas pelo sistema operacional do computador. Dependendo da natureza do problema, pode ser que haja a necessidade de que essas tarefas comuniquem-se entre si. Neste caso, uma investigação acerca da comunicação entre processos<sup>2</sup> se faz extremamente necessária ( *IPC (Inter process communication)*); principalmente sobre o que é processo e sua estruturação computacional.

### 2.3.1 Modelos de comunicação entre processos

A implementação de uma estrutura de comunicação, entre um ou mais processos, exige que se considere uma série de questões inerentes à natureza do problema ao qual ela será acoplada para transferência de informações; a saber:

- Localização: se os processos estão em uma (local) ou mais máquinas (distribuído);
- Característica de cooperação: se a atividade de comunicação é uni ou bidirecional, com processos recebendo e ou enviando mensagens.
- Cooperação e quantidade de participantes: difusão, um processo enviando mensagem a muitos outros; produtor-consumidor, comunicação unidirecional; cliente-servidor, cujo cliente é o processo que requisita uma informação e o servidor é aquele que disponibiliza determinado serviço ao cliente; peer-to-peer, vários nós numa rede realizando funções de cliente e servidor, simultaneamente.

As características supracitadas, com relação a troca de informações entre processos, estão atreladas a um modelo de comunicação, o qual será utilizado para o desenvolvimento de um canal de comunicação, os quais são definidos como: Memória compartilhada ou Transferência de dados através do núcleo do sistema operacional; como exemplo deste modelo conhecido como passagem de mensagem (inglês: “*Message Passing*”). No primeiro, há um compartilhamento de um espaço de endereçamento de memória; no segundo, ocorre um acesso ao núcleo do sistema operacional, o qual está intimamente ligado com o transporte das mensagens [36], [31]. As Figuras 2.12 e 2.13 ilustram os dois modelos de comunicação.

Uma análise comparativa entre os dois modelos, mostra que a comunicação usando compartilhamento de memória é mais rápida que aquela onde há transferência de dados por meio do núcleo do sistema operacional, mas com a desvantagem de ser adequada somente a aplicações cujos processos estão na mesma máquina. No caso da Passagem de

<sup>2</sup> Em ciência da computação, um processo é definido como um programa em execução.

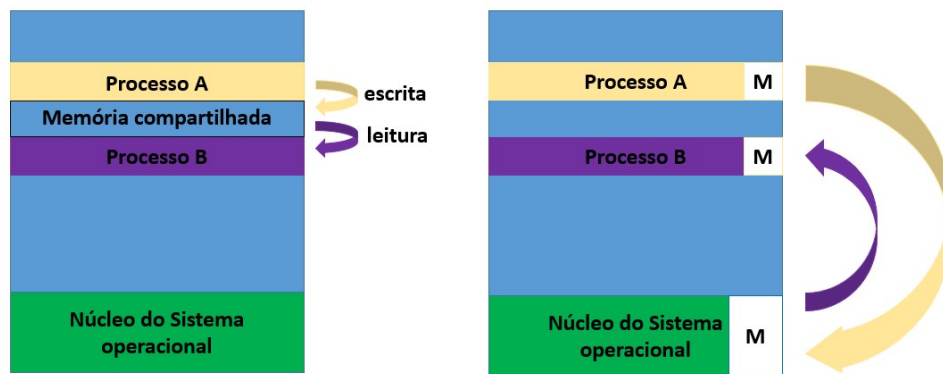


Figura 2.12 – Comunicação entre processos. Memória compartilhada (à esquerda) e Passagem de mensagem (a direita).

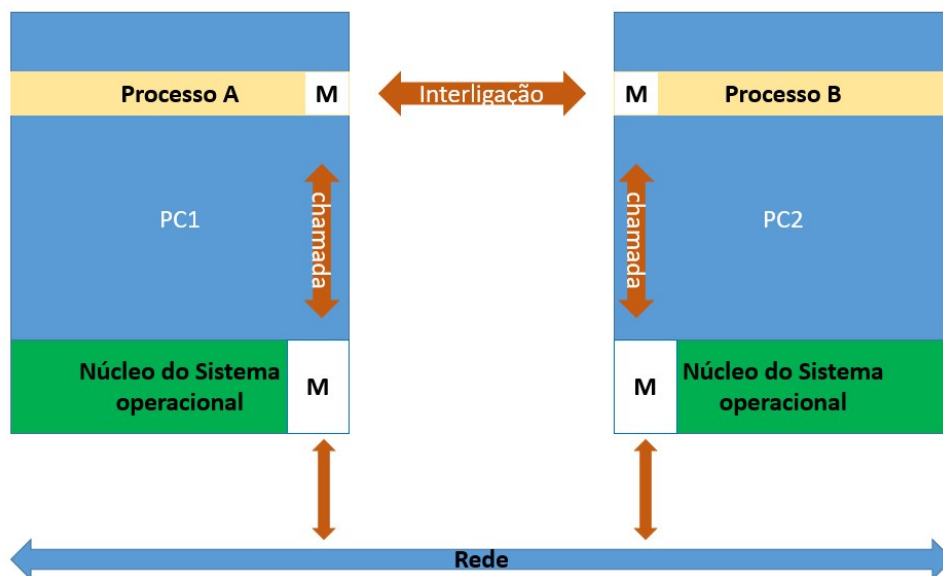


Figura 2.13 – Comunicação entre processos. Passagem de Mensagem.

mensagens, é possível utilizá-la para comunicação local ou em diferentes computadores, mostrando, assim, um potencial de aumento de desempenho em sistemas computacionais muito grandes, pois fornece a oportunidade de usar maior número de processadores.

Os modelos de comunicação são implementados com primitivas baseadas neles. Para o primeiro modelo apresentado, têm-se: as primitivas de memória partilhada com ou sem Threads. Para o segundo têm-se: os Pipes, canais de comunicação local; sockets, canais de comunicação local ou em ambiente distribuído; *RPC (Remote Process Call)* comunicação em ambiente distribuído, dentre outros. O desenvolvimento das primitivas é baseado em uma série de aplicativos disponíveis comercialmente, os APIs. No tópico seguinte há uma apresentação para eles.

### 2.3.2 Interfaces de comunicação - API

A seguir serão apresentados alguns dos aplicativos destinados a interface de comunicação entre processos.

**MPI** é uma biblioteca onde há especificação para nomes, sequências de chamadas, e os resultados das sub-rotinas que serão chamadas por códigos computacionais. Sendo estes podendo ser escritos em linguagens como Fortran, C e C++. O **MPI** possui uma estrutura que permite o gerenciamento de processos (estabelecer, gerenciar, finalizar e identificar processos); rotinas de comunicação, com primitivas para envio e recebimento de mensagens; rotinas de comunicação em grupo, sendo determinante para sincronização de processos, como apresentado em [29].

**OPENMPI** (*Open Source High Performance Computing*) é uma biblioteca em código aberto que segue o modelo de passagem de mensagens e foi baseada em implementações de código aberto do padrão **MPI**: MPI LAM / MPI, LA / MPI (Los Alamos MPI) e FT-MPI (Fault Tolerant MPI). Necessita de compiladores C ou Fortran (OPENMPI, 2017). **OPENMP** — é um **API** que suporta uma programação multithread como modelo de programação em ambiente de memória compartilhada. Em um ambiente de computação paralela com o modelo híbrido, pode-se utilizar **OPENMP** e **MPI** em conjunto. Pode ser utilizado em códigos escritos em C, Fortran e C++ [31].

**WINSOCK** - Windows Sockets 2 (Winsock) é uma biblioteca para comunicação entre aplicações em ambiente do sistema operacional do Windows. Esta **API** possibilita troca de informações entre processos em um mesmo ou vários computadores, neste último caso via inter-rede, independe do protocolo de comunicação sendo utilizado. Além de fornecer acesso a avançadas aplicações em rede disponibilizadas pela Microsoft como Multicast and Quality of Service.

No próximo tópico, será apresentado o mecanismo de comunicação de processos em rede via socket por meio do protocolo TCPIP.

### 2.3.3 Comunicação em rede utilizando protocolos TCP e IP com sockets

A passagem de mensagem entre processos é suportada por duas operações essenciais, enviar e receber. Com base nestas, fica definido que um processo realizará o envio de uma mensagem, para um determinado destino, e um segundo processo, na localidade definida por quem fez o envio, irá realizar a recepção. Sendo assim, para que se realize essa conexão entre processos, é preciso que a **API** escolhida permita, essencialmente, as definições de receptor, emissor, e local de destino. Este último sendo uma identificação para as máquinas onde as aplicações estão situadas. Além disso, é preciso que se defina, também, uma identificação para a aplicação cujo processo pertence, uma vez que existem várias aplicações em uma mesma máquina, como explica [36].



A primitiva “*Socket*”, em português soquete, foi desenvolvida com base nas necessidades de comunicação supracitadas. A transmissão de mensagens via soquete define que se um processo é receptor, seu soquete está associado a uma porta local, identificação da aplicação, e a um endereço **IP** (*Internet Protocol*) do computador no qual é executado. Este endereço é a localização do computador na rede em que este último dispositivo se encontra. O processo emissor, que também possui um soquete com porta e **IP** associados, precisa saber das informações do soquete do receptor para que a mensagem seja transmitida corretamente. A figura 2.14 ilustra o conceito de sockets.

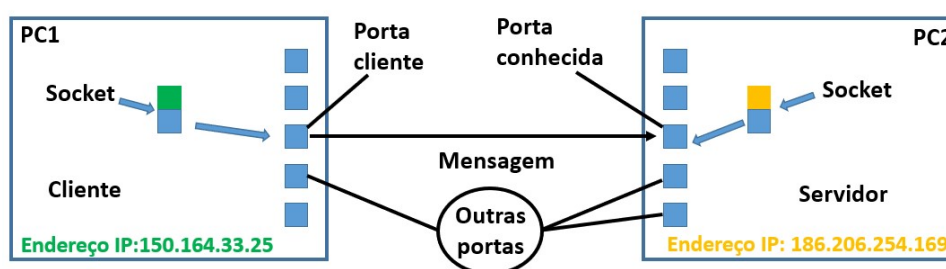


Figura 2.14 – Comunicação cliente-servidor via socket.

Um ponto importante acerca da utilização de soquetes é a escolha de um protocolo de comunicação, ou seja, um conjunto de regras e padronizações que tornam possíveis a comunicação entre processos localizados em dispositivos de diferentes fornecedores. Na comunicação em rede, existem vários tipos de protocolos, os quais podem ser classificados como abertos ou proprietários. Os primeiros são padrões da internet, já os últimos são definidos para ambientes específicos, necessitam para sua implementação a estrutura de hardware e software, para o qual foram projetados. Para os primeiros, têm-se como exemplo os protocolos **TCP** (*Transmission Control Protocol*) e **UDP** (*User Datagram Protocol*) [37].

O protocolo **TCP** faz parte da camada de transporte do Modelo OSI. Esse padrão de comunicação têm características como: orientação a conexão, na qual dois processos só podem interagir com transferência de dados entre eles após o estabelecimento de uma conexão entre ambos; confiabilidade, garantia de entrega de pacote dados ao destino sem erro e na ordem correta de emissão; *Full duplex*, quando dois processos estão conectados é possível transferência de dados bidirecionalmente; controle de fluxo, no qual o receptor confirma o recebimento dos pacotes à medida que vai recebendo [37].

O protocolo **UDP** também pertence a camada de transporte do Modelo OSI. No entanto, não possui as garantias de confiabilidade ofertadas pelo **TCP** para a entrega de pacotes de dados. Estes são enviados sem a garantia de que serão recebidos ou que estarão intactos. Por não garantir esse controle e confiabilidade na emissão de dados, a transmissão **UDP** se torna mais rápida que aquela feita pelo **TCP**. Para situações nas quais a perda ou defeito de parte do pacote de dados, como por exemplo em vídeo ou

chamada de voz, ele se torna interessante. Percebe-se que a escolha entre os protocolos abertos supracitados depende de quais são as principais necessidades das aplicações, tais como confiabilidade nos dados transmitidos e velocidade [37].

Os protocolos **UDP** e **TCP** da camada de transporte do Modelo OSI se associam ao protocolo de internet **IP** para que haja a comunicação em rede (INTERNET); vem daí a nomenclatura TCPIP. Após a definição do mecanismo e protocolo de comunicação a ser utilizado, existe uma questão importante que merece destaque, a sincronização. De acordo com a **API** utilizada, haverá a possibilidade de se definir a troca de mensagens entre processos como sendo síncrona ou assíncrona. Para o exemplo cliente-servidor, dependendo da programação realizada, existe a possibilidade de escolher um padrão de sincronia no envio e recebimento de mensagens. Quando se escolhe um padrão de bloqueio, para o emissor significa que este processo fica bloqueado até a confirmação de recebimento da mensagem; já para o receptor, o mesmo fica bloqueado até que a mensagem esteja disponível. Em contraposição, os processos podem ser não bloqueantes, situação em que não há bloqueio de emissor nem receptor.

A implementação de comunicação entre processos pode ser feita usando diferentes linguagens computacionais, dentre elas têm-se C, Fortran, Java, Python, Cuda, Matlab.

### 2.3.4 Considerações parciais

Este tópico apresentou alguns conceitos sobre a comunicação entre processos; destacadamente modelos característicos, interfaces e protocolos associados a eles. A escolha correta do modelo de comunicação relaciona-se com a arquitetura adotada em um projeto computacional paralelo, já o protocolo e interface, com a velocidade e confiabilidade das funções de comunicação. Isso significa que são premissas essenciais ao projeto de uma solução computacional em paralelo. Logo, a solução paralela do cálculo de **EMT** precisa, necessariamente, se envolver com essa área.

## 2.4 Considerações finais

Neste capítulo foram apresentados os conceitos de computação paralela e distribuída. Mostrou-se que, associado à utilização dele, está a demanda por sistemas computacionais que se caracterizem pela escalabilidade, precisão e rápida capacidade de processamento. E, para que o pesquisador seja capaz de dominar essa infraestrutura tecnológica, seja utilizando de maneira eficiente ou projetando tais sistemas, é preciso que ele tenha conhecimento dos gargalos técnicos que tangenciam esse conceito. Os quais estão direta ou indiretamente relacionados aos tópicos abordados: modelos de programação paralela, arquitetura de memória, comunicação entre processos e protocolos de comuni-

cação, interfaces de programa, hardware adequados a programas paralelos, linguagens de programação.

Tomando como base os tópicos supracitados, o leitor terá melhores condições de entender as escolhas que foram feitas para o desenvolvimento do projeto desta tese. Mas antes disso, a seguir será apresentada uma revisão na literatura sobre trabalhos que fazem uma conexão entre cálculo de transitórios eletromagnéticos e computação paralela.

## 3 Revisão da literatura

Este capítulo tem como objetivo apresentar uma revisão bibliográfica sobre cálculo de transitórios eletromagnéticos em sistemas de energia; com atenção especial para os trabalhos que aplicam computação paralela à solução deste problema. E nesse contexto, os parâmetros que delimitaram a pesquisa foram o cálculo de EMT; hardware; software e algoritmos para modelagem elétrica em ambientes computacionais paralelos; ferramentas matemáticas para integração de simuladores com dinâmicas de processamento lentas e rápidas.

### 3.1 Computação paralela aplicada a cálculo de transitórios eletromagnéticos

A demanda por plataformas computacionais robustas para realizar cálculo de transitórios é crescente. O desenvolvimento das redes elétricas inteligentes; a penetração de unidades geradoras de energia, com destaque para solar e eólica; o aumento da presença de conversores de potência na rede, cuja dinâmica é rápida e não linear; os dispositivos dissipadores de surto e transformadores, que também introduzem não linearidades são alguns dos fatores responsáveis por aquela demanda.

Considerando as transformações supracitadas para o setor elétrico, é certo que as simulações de transitórios eletromagnéticos estarão sujeitas a sistemas de energia grandes e complexos. A computação paralela é uma das ferramentas disponíveis para contribuir com soluções rápidas e precisas nesse contexto. Além de fornecer a oportunidade de trabalho colaborativo.

A associação de computação paralela ao cálculo de transitórios eletromagnéticos é um tema que já vem sendo desenvolvido por muitos pesquisadores [38], [39],[40]. Quando H. W. Dommel [6] desenvolveu a técnica para implementação de soluções relacionadas a transitórios eletromagnéticos, nem se imaginava que na modelagem proposta para as redes elétricas havia uma oportunidade para que esse problema se tornasse paralelizável. O entendimento de como transformar esse cálculo sequencial em paralelo começa com a compreensão do modelo de linhas de transmissão proposto naquela época.

O modelo de linha de transmissão, baseado na teoria de onda viajante, permite que um sistema elétrico seja particionado em uma determinada quantidade de subsistemas. Cada linha permite uma subdivisão. Do ponto de vista da equação matricial de estado no tempo, isso significa que há um desacoplamento entre as equações que representam as partes do sistema que estão conectadas em cada terminação da linha. Embora o cálculo

de transitórios para os subsistemas possa ser feito separadamente, a análise global, que é a determinação das tensões de nó do sistema completo, permanece com os mesmos resultados[38].

Um trabalho que exemplifica a aplicação de computação paralela ao cálculo de transitórios é apresentado por Falcão et al. [38]. Neste estudo, um hardware multiprocessador com oito núcleos foi utilizado para o processamento do problema. A divisão de tarefas que serão executadas em cada núcleo é um ponto importante para o desempenho da solução. Sendo este, um parâmetro escolhido para desenvolver o algoritmo de balanceamento das cargas nos núcleos de processamento. Para tanto, o quesito escolhido foi separar subsistemas cujas variáveis trocam informações elétricas. Outro parâmetro que influenciou na divisão de tarefas foi a quantidade de operações de ponto flutuante.

Após a definição do algoritmo de balanceamento de cargas, um conjunto de simulações foi realizado [38]. Depois de um comparativo feito neste estudo, concluiu-se que um cálculo mais eficiente, ou seja, que objetiva menor tempo de simulação deve considerar a quantidade de núcleos disponíveis, o balanceamento de cargas em cada núcleo, os locais do sistema onde será feita a partição para a formação de subsistemas menores. Esses são parâmetros que influenciarão o tempo de processamento em cada núcleo e as trocas de informações entre subsistemas em núcleos diferentes; definindo a eficiência da solução.

Seguindo a mesma linha de pesquisa, relacionadas à integração de ambos os temas, Falcão [41] faz uma revisão geral sobre computação de alto desempenho aplicada a estudos envolvendo sistemas de potência. O autor enfatiza a complexidade matemática da modelagem de componentes do setor elétrico, por exemplo, dispositivos de controle baseados em eletrônica de potência; o que culmina na necessidade de alta capacidade de processamento computacional. O autor mostra que apesar da evolução da capacidade de processamento dos hardwares, a requisição por estruturas de hardware multitarefas, trabalhando em paralelo é uma realidade que veio para ficar. Uma simples pesquisa de mercado em arquiteturas computacionais, na atualidade, mostra que as previsões de Falcão se confirmaram.

Ainda neste estudo [41], Falcão faz uma pesquisa sobre arquiteturas computacionais com a intenção de mostrar possibilidades de aplicação de computação de alto desempenho a análises do sistema de potência. Tendo algumas como destaque: Processadores Vetoriais, Processadores Superescalares, Multiprocessadores com memória compartilhada, Multiprocessadores com memória distribuída, Máquinas Massivamente Paralelas. A pesquisa mostrou que a computação de alto desempenho pode ser estendida para análises como: simulação em tempo real; controle em tempo real; otimização; ferramentas inteligentes para análise e síntese. É importante ressaltar que, atualmente, as arquiteturas citadas já estão em uso em sistemas comerciais como o Simulador de tempo real da TEPCO; RTDS da Manitoba HVDC Research Centre e o Supercomputador da Hydro

Quebec, no Canadá.

Além de Falcão [41], outros autores, Watson et al. [12] e Velasco et al. [11], mostram que a computação paralela aplicada a simulações envolvendo sistemas elétricos é um tema que há muito tempo vem sendo trabalhado. Uma das linhas de aplicação é a simulação digital em tempo real. Duas empresas dominam o mercado dessa tecnologia, RTDS Technologies e OPAL-RT. No portfólio de aplicações de seus produtos ao redor do mundo estão: Manitoba-Hydro e Hydro-Québec, no Canadá; EDF, na França, projeto Concept Grid; PowerLabDK, laboratório na Dinamarca; SESA-LAB na Alemanha, com o projeto Mosaik; centros de pesquisa como o CAPS, na Universidade da Flórida, o EPRI nos USA e o Instituto Tecnológico de Nagoya, no Japão.

Venugopal et al. [42] são outros autores que investigam as transformações pelas quais o setor elétrico tem passado e suas implicações para as simulações de sistemas elétricos. De acordo com ele, uma área que está em desenvolvimento são as pesquisas com **HIL** (*Hardware in Loop*), a qual está relacionada a simulações em tempo real. O autor mostra que os conceitos e dispositivos relacionados a redes inteligentes de energia e microrrede, plantas fotovoltaicas, alta penetração de inversores, geração eólica, infraestrutura de comunicação para rede elétrica impõem maior complexidade às simulações. Essas inovações trazem grandes desafios à estudos para testes de confiabilidade, falta e funcionalidade relacionados com o sistema elétrico, dos quais destacam-se aqueles relacionados ao passo de cálculo das simulações do tipo **HIL**, linhas de transmissão curta, grande número de dispositivos da eletrônica de potência, flexibilidade e escalabilidade.

Para exemplificar, Venugopal et al. [42] citam que simulações em tempo real de sistemas complexos podem exigir passos de cálculo que estão numa escala de 10 a 50 microssegundos para garantir resultados precisos. Neste contexto, um primeiro desafio está relacionado à modelagem de sistemas grandes e complexos como linhas de transmissão curtas, cujo atraso de tempo da propagação de onda nesta linha costuma ser menor que os passos de cálculo anteriormente citados. Isto inviabiliza a partição desses sistemas em subsistemas menores, como é feito no caso de linhas longas. Buscando contornar essa limitação, o autor apresenta uma técnica denominada solucionador nodal no espaço de estado (em inglês: “*state-space nodal solver*” (SSN)), através da qual é possível simular linhas curtas em grandes sistemas com elevado grau de precisão.

A complexidade da simulação em tempo real de redes inteligentes e microrredes no contexto de **HIL** também está relacionada com a alta penetração de conversores de potência para conectar múltiplas fontes e cargas. Devido a natureza não linear, a alta velocidade para capturar e interpolar rápidas frequências de chaveamento desses dispositivos, a demanda por computação de alto desempenho, paralela e distribuída, envolvendo hardwares com múltiplos processadores e FPGA são necessários [42].

Outras duas características importantes para simulações em tempo real é a escala-

bilidade e flexibilidade. Estas surgem como consequência da demanda por análises envolvendo a presença de múltiplos sistemas, com diferentes características de simulações. Por exemplo a integração de simulações de grandes e complexos sistemas com passos de cálculo diferentes para a solução de cada subsistema. O simulador eMEGAsim, da OPAL-RT, é uma das tecnologias comerciais de destaque para esse tipo de simulação [42].

Seguindo na direção de simulações off-line, mas, também, usando computação paralela, Debnath et al. [39] apresenta estudos envolvendo cálculo de EMT. Estes autores mostram que a integração de elementos não lineares aos sistemas de energia, tais como dispositivos de chaveamento da eletrônica de potência, fazem com que o tempo de simulação usando computador convencional seja crescente. Uma nova arquitetura, composta por GPU da NVIDIA e CPU é proposta neste trabalho como estratégia para melhorar o desempenho da simulação de transitórios. No estudo de caso, os autores avaliam quatro estratégias de simulação do cálculo de transitórios para testarem a eficiência do processamento usando GPU: na primeira, apenas a CPU faz o processamento computacional; na segunda, a GPU executa a multiplicação matriz-vetor; na terceira, a implementação dos termos históricos junto às multiplicações de matriz-vetor da computação são implementados pela GPU; na quarta, a GPU executa todo o processo de computação requerido e a CPU, o controle do fluxo da simulação e armazenamento dos dados de saída. Esta última foi a que se mostrou mais eficiente para simulações de transitórios em sistemas de potência acima de 700 barras. Este trabalho mostrou que a aplicação de GPU para cálculo de transitórios eletromagnéticos torna o processamento mais eficiente se comparado àqueles usando apenas CPU. No entanto, uma atenção deve ser dada a diferentes estratégias de distribuição das tarefas do programa entre CPU e GPU.

A aplicação de computação com GPU pode melhorar o desempenho de simulações reduzindo o tempo de simulação. Uma atenção especial deve ser dada a estrutura do sistema que será implementado na plataforma de paralelismo. A primeira pergunta que dever ser feita è: O problema matemático é paralelizável? A segunda questão é sobre o balanceamento de carga computacional em cada núcleo da GPU. Uma grande quantidade de núcleos envolve mais tempo de comunicação entre as partes do sistema. Um terceiro ponto, é sobre a partição de tarefas a serem realizadas na CPU e GPU. Partes sequenciais, geralmente são colocadas na CPU, mas nem sempre é uma boa solução. Quando tarefas implementadas na GPU precisam se comunicar com essas partes do programa na CPU, o tempo de simulação aumenta. Por fim, destacam-se os seguintes pontos em se tratando de simulações envolvendo GPU: A distribuição do problema nos núcleos da GPU; tempo de comunicação entre os núcleos e tempo de processamento; técnicas de esparsidade aplicadas à multiplicação de matrizes [43].

Diferentemente da proposta de arquitetura computacional com GPU, Singh et al. [44] apresentam um trabalho no qual utilizam computação paralela, multinúcleo, para

realizar simulação de transitórios eletromagnéticos. Os autores desenvolvem uma arquitetura utilizando os softwares PSCAD<sup>TM</sup>/EMTDC<sup>TM</sup> como solucionador do cálculo de EMT, XGE (*Xorea: Grid Engine*) para gerenciar as tarefas entre os diversos agentes, no caso os núcleos dos processadores. Uma interface de comunicação desenvolvida para o EMTDC (*Electromagnetic Transients Including DC*) foi utilizada, denominada ENI (*Electrical Network Interface*). Esta permite a comunicação elétrica entre os diversos subsistemas sendo simulados em cada processador. A atividade de dividir as tarefas para cada um dos agentes é feita pelo XGE.

O trabalho de Singh et al. compara duas simulações de transitórios usando a computação sequencial e paralela. Na primeira, apenas um processador é utilizado; na segunda, o sistema completo é distribuído em até sete processadores. Estes autores mostram que a computação paralela se torna mais interessante que a sequencial quando o tempo de comunicação entre os processadores é muito menor que o tempo de execução das subdivisões dos sistemas em cada processador. Acrescentam, ainda, a possibilidade de se realizar as mesmas simulações utilizando a computação distribuída, com o mesmo solucionador de EMT, o software PSCAD<sup>TM</sup>/EMTDC<sup>TM</sup>, e a mesma interface ENI. No final, Singh et al. sugerem, como pesquisas futuras, estratégias otimizadas de divisão de um grande sistema elétrico entre os processadores, com o objetivo de reduzir tempo de simulação computacional.

Assim como Singh et al. [44], Barros et al. [45] realizam um trabalho no qual utilizam o PSCAD<sup>TM</sup>/EMTDC<sup>TM</sup> para realizar computação paralela. Os autores citam algumas alternativas para aplicação de computação paralela ao cálculo de transitórios. Em [38], a linha de transmissão é o elemento chave para isso. Além deste, Barros et al. citam, também, stub lines ou elementos modificados que incluem atraso de tempo. De acordo com os autores, para o cálculo de transitórios em grandes sistemas, é necessário que cada subparte deste último seja processada em núcleos diferentes e que, naquelas que estavam interligadas, haja comunicação dos dados elétricos nos pontos em que houve partição do sistema.

Barros et al. [45] apresentam três softwares do tipo EMT, utilizados pela indústria, através dos quais é possível aplicar computação paralela para cálculo de transitórios: ATP (*Alternative Transient Program*), PSCAD<sup>TM</sup>/EMTDC<sup>TM</sup>C, EMTP-RV. Esses softwares permitem que o usuário possa desenvolver componentes próprios dentro dessas plataformas e integrá-los à estrutura de cálculo de transitórios do software. Sendo assim, é possível desenvolver um componente que faça a interligação entre subsistemas, permitindo a comunicação entre eles. Isso habilita que subsistemas menores sejam interligados para formar um grande sistema.

Antes de apresentar o estudo de caso, Barros et al. [45] citam oito mecanismos para comunicação entre processos disponíveis para sistemas operacionais Windows: Clip-



board, COM (*Component Object Model*), Data Copy, DDE (*Dynamic Data Exchange*), File Mapping, Mailslot, Pipes, RPC e Windows Sockets. Dentre estes, escolhem o Named Pipes para ser responsável pela comunicação entre os subsistemas por ser simples, flexível e fazer com que os tempos de comunicação não sejam expressivos em relação ao tempo total de computação do sistema completo.

O estudo de caso apresenta dois grandes sistemas de energia para a simulação com falta trifásica para terra. O primeiro é composto por uma rede que inclui 30 barras, 41 elementos em série, 3 geradores síncronos, 2 modelos completos SVC e 2 conexões back-to-back com controles correspondentes. O segundo possui 630 nós trifásicos, composto pela rede anterior adicionado de um link HVDC (*High Voltage Direct Current*) e duas unidades geradoras. O PSCAD foi o software escolhido para o estudo. Um componente para a comunicação entre os subsistemas foi desenvolvido. Um comparativo entre computação sequencial e paralela foi realizado. Para tanto, foi feita uma partição do sistema completo em até quatro núcleos, na mesma máquina. Os resultados mostraram que o tempo de processamento do transitório em paralelo foi mais rápido que o sequencial. Níveis de aceleração computacional de até 20 vezes foram alcançados no trabalho. Ao final, Barros et al. [45] chamam a atenção para os benefícios da computação paralela e propõem a simulação em cluster, integrando subsistemas com diferentes passos de cálculo.

A redução do tempo de simulação é um dos principais motivos para aplicação de computação paralela a um determinado problema. Os resultados alcançados por Barros et al. mostram que uma estrutura de hardware e software adequados permitem atingir esse objetivo. Outro trabalho, cuja computação paralela é aplicada ao cálculo de transitórios e houve diminuição de tempo de simulação em relação ao processamento sequencial, é apresentado por Yonezawa et al. [46]. Estes autores utilizam a teoria desenvolvida em [6] para o problema do tipo EMT. Conforme afirmam Yonezawa et al, na equação matricial de estados que modela o problema de transitórios estão as equações relacionadas aos subsistemas elétricos, provenientes da partição do sistema completo. Considerando isso, ele desenvolve uma metodologia baseada em threads para a solução do cálculo de transitórios.

A metodologia desenvolvida por Yonezawa et al. [46] estabelece um fluxograma com três regiões distintas, relacionadas ao cálculo de transitórios, para explicar como foi aplicado o paralelismo neste tipo de problema. Na primeira região, ocorre a formulação e inicialização do sistema com a entrada em um loop para atualização de variáveis e estado dos componentes do sistema; na segunda região, ocorre o estabelecimento dos subsistemas e solução do cálculo de transitórios neles; na terceira região, avalia-se a tomada de decisão pela continuidade ou finalização do programa. Na segunda região do cálculo é onde os autores apresentam sua contribuição, dois algoritmos para distribuição de tarefas entre threads do programa são apresentados: “*Static Scheduling*” e “*Dynamic Scheduling*.”

A aplicação do algoritmo proposto por Yonezawa et al. é feita utilizando um sis-

tema trifásico composto por linhas de transmissão, geradores, cargas trifásicas e transformadores. A simulação é implementada em um programa denominado XTAP. A estratégia de paralelização considera as características de atraso de tempo em linhas de transmissão. Com base nela, a solução do cálculo para o sistema elétrico é subdividida em 17 partes. A interface [OPENMP](#), adequada para arquiteturas de memória compartilhada, foi utilizada. Essa [API](#) é projetada para algoritmos multi-thread cuja programação pode ser feita em C, C++ e Fortran.

Os resultados apresentados por [46] mostraram que a aplicação de paralelismo diminuiu significativamente o tempo de simulação quando comparada ao cálculo sequencial. Além disso, mostraram, também, que o algoritmo de distribuição de tarefas influenciou na velocidade de processamento, sendo que para o estudo de caso proposto, o *Static Scheduling* foi mais eficiente que o *Dynamic Scheduling*.

O tempo de processamento está ligado ao tamanho e complexidade do problema a ser solucionado. Como visto anteriormente, a distribuição de tarefas em vários núcleos é uma alternativa para redução de tempo. De acordo com Huy et al. [47], a redução de sistemas ou equivalentes de rede é usada, com frequência, para facilitar o processamento computacional de grandes sistemas de potência em simulação de tempo real. No entanto, existem situações em que a complexa natureza do problema exige que a modelagem completa do sistema de potência seja feita. Casos em que se analisa correntes induzidas geomagneticamente e estudos de [HIL](#) envolvendo múltiplos dispositivos são exemplos. Uma estrutura computacional para suportar esses tipos de simulações precisa de robustez e flexibilidade. Tais requisitos têm o paralelismo de hardware e software como uma das soluções [47].

Devido a esses requisitos, Huy et al. [47] apresentam uma arquitetura computacional composta por um simulador de tempo real, o Hypersim. Este dispositivo foi modificado, ao nível de hardware e software, agregando à sua estrutura um supercomputador SGI UV300 com vistas a melhorar o desempenho de comunicação do mesmo; o que possibilitou mais unidades de processamento computacional trabalhando em conjunto. Com as novas modificações, a arquitetura se tornou um simulador de [EMT](#) em tempo real massivamente paralelo.

Um sistema de potência com 16432 nós, implementado com base em uma referência de sistema da empresa canadense Hydro-Québec, foi utilizado para validar a plataforma com alto grau de paralelismo. Os resultados mostram sucesso na proposição, isso porque os fatores de aceleração, com base na lei de Amdahl, foram da ordem de 50 vezes para 65 unidades de processamento utilizadas, e 83 vezes, para 200 unidades de processamento em paralelo.

Outro simulador de tempo real, disponível comercialmente, é citado por Dyck et al. [40]. Segundo os autores, o simulador RTDS pode ser utilizado para simular operações

do sistema distribuição com fontes de energia distribuída. O software de EMT disponível neste simulador resolve cálculo de EMT da mesma forma que o algoritmo proposto por [6], em um hardware com processadores paralelos de alta velocidade. Considerando estas características, afirmam que a estrutura desse simulador é adequada a simulações de um grande sistema de distribuição com recursos energéticos distribuídos (em inglês: “*DERs*”).

Para testar o simulador, Dyck et al. [40] apresentam um trabalho no qual realizam a simulação de um grande sistema de transmissão, “*IEEE123 Node Test Feeder*”, com geração distribuída. Este sistema foi modelado de duas formas, uma simplificada e outra complexa. Os resultados das simulações de ambas as modelagens foram comparados a um software de simulação off-line, CYMDIST da empresa CYME, já conhecida na literatura técnica. O comparativo entre este último e o simulador de tempo real, considerando os níveis de tensão nas barras, mostrou resultados bastante satisfatórios; apontando o RTDS como uma alternativa para simulação de grandes sistemas. É importante ressaltar que a modelagem mais complexa necessitou de 6 processadores enquanto que a outra, 2.

Outra funcionalidade apresentada por Dyck et al. [40] neste trabalho é a possibilidade de aplicações de Hardware in the loop. Conforme os autores, esse tipo de aplicação pode ser dividido em duas classes: CHIL (*Controller Hardware in Loop*) e PHIL (*Power Hardware in Loop*). Nesta, há o envolvimento da conexão física de um componente físico a um sistema de potência, sendo este sistema modelado na plataforma RTDS; naquela, todo o sistema de potência é modelado no simulador e um dispositivo de controle a ser testado é conectado ao simulador. Neste último caso não há troca energética na interface de conexão entre sistema físico e sistema simulado. Um teste PHIL, cujo componente físico era um inversor PV, foi realizado para mostrar a funcionalidade do simulador.

### 3.1.1 Considerações parciais

Os trabalhos apresentados anteriormente confirmaram uma gama de possibilidades de aplicação da computação paralela a simulações de transitórios eletromagnéticos. A linha de transmissão foi considerada um elemento chave para que houvesse a junção entre essas duas grandes áreas do conhecimento, embora outras técnicas para paralelizar o cálculo sejam possíveis. A maioria dos trabalhos de pesquisa focaram nesta técnica computacional como forma de reduzir o tempo de simulação dos eventos em sistemas de energia. A estratégia de participação do sistema elétrico em diferentes núcleos, o modelo de programação adotado, a redução do tempo de comunicação entre processos e o hardware disponível foram alguns dos parâmetros que contribuíram para aceleração do processamento.

O cálculo de EMT associado à computação paralela pode ser acelerado se o projeto for implementado de forma adequada. A redução do tempo de simulação é um dos benefícios alcançados neste tipo de trabalho, no entanto, o cálculo paralelo, que pode ser

distribuído, fornece novas oportunidades de trabalho e desafios, como será visto a seguir.

## 3.2 Cálculo Computacional Colaborativo

A computação paralela é geralmente utilizada para os casos em que se pretende reduzir tempo de simulação em um determinado projeto. Todavia, ela possibilita outras oportunidades, quais sejam: na disponibilidade de *hardware* para satisfazer requerimentos de simulações com grandes e complexos sistemas; trabalhos colaborativos com a garantia da manutenção da propriedade intelectual, compartilhamento de dados do cálculo de transitórios sem mostrar dados dos subsistemas envolvidos, os quais, talvez, podem ser confidenciais [48], [49].

M. Stevic et al. [49] apresentam uma arquitetura computacional para o desenvolvimento de uma simulação de tempo real (RT) colaborativa. Neste estudo, dois simuladores localizados nos laboratórios da SINTEF (Trondheim, Norway) e ACS (RWTH Aachen University, Germany) são utilizados para a simulação de um link HVDC ponto a ponto que conecta dois sistemas AC. A interface que faz a conexão entre estes últimos utiliza uma técnica denominada *ITM-Ideal Transformer Model*, a qual utiliza fontes de controle de corrente e tensão ideais. O protocolo de comunicação utilizado foi o UDP e a rede de conexão entre os laboratórios foi estabelecida via *GÉANT* e *NORDUnet*, redes que interconectam centros de pesquisa e educação através da Europa e países nórdicos.

Os resultados apresentados por M. Stevic et al. [49] para o estado estável e transitório lento foram satisfatórios comparativamente a simulações realizadas de forma não distribuída. Embora, ainda os autores citem a necessidade de aperfeiçoar a interface com relação a problemas de atraso e perdas de pacote relacionados à comunicação entre processos.

Em [50], M. Stevic et al. apresentam uma arquitetura computacional para integração de laboratórios, na Europa, para o desenvolvimento de co-simulações de tempo real. Neste projeto, quatro laboratórios estão envolvidos com uma determinada função específica: ACS, em Aachen, Germany, hospeda um simulador de tempo real (RTDS); Polito, em Turim, Itália, hospeda outro simulador de tempo real (OPAL-RT); JCR IET, Ispra, Itália, hospeda um sistema de monitoramento de sinais da arquitetura; JCR IET, em Petten, Holanda, hospeda um sistema para fornecer entrada de dados para cenários de simulação.

Na arquitetura proposta, uma interface denominada *VILLAS* faz a interconexão entre os simuladores com o protocolo UDP/IP. Para o sistema de monitoramento e os simuladores, o protocolo é o OMA NGI-10. No caso do simulador em Turim e o sistema em Petten, o protocolo de comunicação também é o UDP. Esta interface também utiliza a técnica ITM, como proposta em [49]. Embora, agora esteja mais evoluída, já que trata

melhor das perdas e atrasos de pacotes de dados. A rede europeia que conecta instituições e centros de pesquisa foi utilizada para a comunicação de dados

Para ilustrar o funcionamento da arquitetura computacional, M. Stevic et al. propõem a simulação envolvendo a interconexão de dois sistemas baseados em dados reais da Itália. O primeiro é um sistema de transmissão hospedado em Aachen, formado por geradores, linhas e cargas; o segundo é um sistema de distribuição hospedado em Turim, formado por uma subestação, cargas, linhas, consumidores e geradores fotovoltaicos. O estudo de caso analisa o impacto da redução da geração fotovoltaica na frequência do sistema de transmissão e na tensão no sistema de distribuição. Os dados de perfil da geração solar são fornecidos pelo laboratório de Petten e os resultados de monitoramento são apresentados com o sistema de Ispra.

Através deste trabalho, é possível identificar novos cenários de estudos para o setor elétrico. A computação distribuída fornece um variado conjunto de oportunidades, sendo que o principal deles é o trabalho colaborativo. Aumentando, assim, o potencial de inovação dos projetos de pesquisa a serem realizados.

Outro exemplo de computação distribuída aplicada na simulação de RT foi apresentado por Ren Liu et al. [51]. Uma estrutura de co-simulação é projetada com dois simuladores de RTDS dentro de uma rede local LAN (*Local Area Network*) para estudar o efeito da latência de comunicação nas simulações de RT. Os testes de casos mostram que pode causar imprecisão numérica ou cálculo incorreto. Para resolvê-lo, Ren Liu et al. aplicaram um método de previsão linear em simulações de RT. O protocolo de comunicação da simulação é o UDP. A latência é induzida na simulação com as ferramentas do RTDS. O valor fixo de 15 ms é escolhido para esse parâmetro, mas no mundo real esse parâmetro pode variar.

Segundo Ren Liu et al. [51], analisar a influência da latência em simulações distribuídas é um estudo complexo, pois depende de vários fatores. Alguns deles são listados como responsáveis pelo atraso total associado à latência: meio de propagação física; tecnologia de equipamentos de rede e função de processamento específico; camada de link físico e tamanho de pacotes e tecnologia de roteamento de pacotes. Este artigo apresenta influência da latência, estipulada artificialmente, no cálculo de uma simulação. Um avanço interessante para este trabalho seria avaliar como a latência da comunicação influencia no tempo total das simulações, cujos computadores estão em locais dispersos usando a WAN (*Wide Area Network*) para comunicação.

Monti et al. [52] apresentam uma plataforma computacional para simulações de sistemas elétricos de potência. Neste trabalho, simuladores de tempo real e HIL, em oito laboratórios geograficamente distribuídos entre Europa e Estados Unidos, são interconectados, via rede internet, para a formação de um super laboratório global de pesquisa e desenvolvimento.

Monti et al. mostram que os sistemas de energia estão deixando de ser eletromecânicos para se tornarem eletrônicos. Tal mudança se deve à alta penetração de recursos elétricos distribuídos, cuja interface com a rede é um dispositivo baseado em eletrônica de potência. E, com base nessas transformações, discutem a necessidade de desenvolver novas infraestruturas de laboratórios computacionais para lidar com os desafios que elas impõem. Um estudo de caso envolvendo sistemas de transmissão, distribuição, PHIL de uma turbina eólica e inversores fotovoltaicos é conduzido para ilustrar a proposta do trabalho.

As inovações propostas para os sistemas elétricos de potência têm como consequência um aumento significativo na complexidade das pesquisas para este setor. O trabalho de Monti et al. é mais um exemplo que consolida o direcionamento para trabalhos colaborativos em simulações computacionais, o que é viabilizado por meio de arquiteturas computacionais distribuídas.

ZAMRONI [53] desenvolve uma arquitetura computacional para co-simulação híbrida de Transitórios eletromagnéticos EMT – Estabilidade transitória (TS (*Transient Stability*)). A estrutura do projeto envolve os softwares Power Factory (EMT) e PSS/E (TS). Neste trabalho, os fluxos de potência ativa e reativa são analisados nas simulações com um sistema de potência. Os resultados apresentaram diferenças consideráveis em relação ao caso base de referência; o que Zamroni atribui a atrasos na resposta de cada simulador e a possíveis erros na solução do sistema. Adicionalmente, propõe o cálculo colaborativo e distribuído como promissora pesquisa futura.

### 3.2.1 Considerações parciais

As mudanças no setor elétrico estão ocorrendo em uma velocidade muito alta. Para enfrentar os desafios que essa dinâmica impõe às simulações de sistemas elétricos, é imperativo o trabalho colaborativo entre os pesquisadores. Neste caso, a computação distribuída vem como uma aliada a essa nova demanda. Um caminho de muitas oportunidades é aberto quando se associa esta técnica ao cálculo de transitórios eletromagnéticos. Os exemplos deste tópico mostraram um pouco desse caminho de possibilidades.

## 3.3 Ferramentas matemáticas e computacionais para integração de diferentes simuladores

O ambiente paralelo para a simulação de transitórios eletromagnéticos possibilitou novas oportunidades além de redução de tempo de processamento, tais como interação entre softwares com diferentes funcionalidades, simulações com múltiplos passos de cálculo, uso de funções pós processamento para cálculos ótimos no projeto e design de elementos

dos dispositivos do setor elétrico. No entanto, para isso ser possível de fato, ferramentas matemáticas e/ou computacionais precisam ser desenvolvidas. Os trabalhos a seguir pretendem trazer algumas soluções já consolidadas nesse sentido.

Semlyen et al. [54] descrevem uma metodologia para cálculo de transitórios eletromagnéticos através da qual é possível usar múltiplos passos de cálculo para a integração trapezoidal; isso no contexto de programas envolvendo sistemas elétricos, cujo modelo segue a teoria de ondas viajantes; como [ATP](#), [PSCAD/EMTDC](#) e [EMTP](#). A formulação apresentada explora o desacoplamento promovido pelas linhas de transmissão em sistemas de potência. Um sistema maior  $X$  pode ser particionado em sistemas menores, sendo que o efeito entre eles é computado através dos termos históricos, presentes no modelo da linha, os quais são integrados em cada uma das terminações em que ela está localizada. Outra questão abordada, é como fazer uma interação entre sistemas com passos de cálculo diferentes. O efeito de um sistema mais lento em um mais rápido, baseado na onda eletromagnética que se propaga na linha, é realizado por meio de interpolação. E o mais rápido sobre o mais lento por meio de uma dizimação.

Do ponto de vista da computação paralela, a metodologia apresentada por [54] permite que um grande sistema de potência seja particionado em subsistemas menores, tendo cada um destes uma solução de cálculo de transitórios com um passo de cálculo diferente. Isso significa que além de aproveitar o paralelismo para diminuir o tempo de simulação, uma partição ótima do sistema poderia ser elaborada levando em conta as características intrínsecas a cada uma das partes do sistema, de forma a considerar um adequado passo de cálculo para cada uma delas. Por exemplo, em um sistema envolvendo linhas de transmissão, elementos a parâmetros concentrados, plantas de geração fotovoltaica com conversores estáticos e fontes de tensão. Os elementos a parâmetros concentrados, com passo de cálculo grande, e plantas fotovoltaicas com conversores, passo de cálculo pequeno, podem ser separados em subsistemas diferentes. Isso geraria um benefício duplo na computação paralela, mais núcleos para processamento e menos operações de cálculo matemático.

Gole et al. [55] apresentam um trabalho no qual desenvolvem um componente dentro da plataforma do [PSCAD/EMTDC](#) para conexão com o MATLAB. A possibilidade de comunicação entre ambos os softwares acontece devido a estrutura organizacional em que o [PSCAD/EMTDC](#) foi desenvolvido. O arquivo executável do [PSCAD/EMTDC](#) possui uma parte contendo os dados da rede fixa e um modelo dinâmico denominado DSDYN, o qual permite ao usuário desenvolver seu próprio componente e integrá-lo ao software. Vale a pena ressaltar as possibilidades que esse componente viabiliza, pois, como consequência dessa interação todas as poderosas ferramentas matemáticas e estrutura de computação paralela disponíveis no MATLAB poderão ser utilizadas pelo [PSCAD/EMTDC](#) para aperfeiçoar estudos envolvendo cálculo de transitórios. Duas estruturas de interface são

construídas. Na primeira, a comunicação entre ambos os softwares é feita antes e depois da simulação; na segunda, a comunicação é feita durante a simulação de um estudo de caso.

Seguindo a mesma linha de trabalho de [55], Filizadeh et al. [56] avançam um pouco mais e apresentam, de forma mais geral, diversos modelos de interface de programa do tipo EMT com ferramentas matemáticas em geral. Dentre elas, destacam-se: Interfaces internas, dentro do próprio programa de cálculo de transitórios, citando como exemplo o PSCAD/EMTDC, no qual é possível projetar um componente como uma ferramenta a ser utilizada em estudos de caso; Interfaces externas, nas quais o software de EMT utiliza as potencialidades de outro software no auxílio de simulações de transitórios. Um exemplo desta última, foi a utilização do PSCAD/EMTDC e MATLAB para fazer o design ótimo de parâmetros de dispositivos da eletrônica de potência, como conversores, HVDC e FACT (*Flexible AC Transmission System*).

Filizadeh et al. [56] mostram um exemplo ao se utilizar um STATCOM para controlar a tensão em uma barra de um sistema elétrico com carga variável. Através de um algoritmo de otimização, dentro do MATLAB, parâmetros daquele dispositivo eletrônico foram modificados e definidos com vistas a se manter o nível de tensão adequado na barra, apesar das variações de carga nela.

Heidari et al. [57] apresentam um trabalho que consolida as metodologias de interfaces apresentadas por [56]. Aqueles autores desenvolvem uma tese na qual propõem duas ferramentas importantes para o design de sistemas elétricos. A primeira é um algoritmo de otimização baseado em simulação de transitórios eletromagnéticos. Ela faz escolha de parâmetros tomando como base os resultados de simulações feitas no PSCAD. A segunda ferramenta ajuda o designer a analisar os efeitos de incertezas no desempenho do sistema. No trabalho de Heidari et al., ressalta-se um ponto importante, o fato dele apresentar a possibilidade de se paralelizar o algoritmo de otimização, integrando, nesse caso, computação de alto desempenho às ferramentas matemáticas utilizadas no auxílio ao cálculo de transitórios.

Boaventura et al. [58] desenvolvem um trabalho no qual é avaliada uma proposta metodológica para simulação de transitórios em tempo real aplicado a grandes sistemas. Nesta abordagem, destacam-se as seguintes peculiaridades: modelagem de sistemas elétricos por meio de equivalentes de rede e uso de múltiplos passos de cálculo. Um estudo de caso é feito para validar a proposta. Neste, um sistema completo, contendo uma porção analógica, circuito equivalente de Thévenin AC, e uma porção digital, composta de três subsistemas modelados com equivalentes de rede, foi simulado. Numa das simulações realizadas no trabalho, a porção analógica foi simulada com o menor passo de cálculo, 4 microssegundos, e a outra, com passos iguais a 12, 16 e 20 microssegundos, um para cada subsistema. A modelagem com múltiplos passos foi feita no software Matlab.



De acordo com [58], um ponto central nesse estudo é a interface de comunicação entre subsistemas cooperantes com diferentes passos de cálculo. Toda vez que informações elétricas são enviadas de um subsistema para outro, o sinal elétrico passa por uma interpolação, caso saia de um sistema mais lento em direção a um mais rápido; e, por uma dizimação, caso ocorra o inverso. Um comparativo entre essa proposta com múltiplos passos de tempo é feito com uma simulação do sistema completo da rede, implementado no **PSCAD**; o que mostrou resultados satisfatórios.

O estudo de [58] é importante pois abre caminho para que simulações de grandes sistemas, com subsistemas modelados por equivalentes digitais, sejam feitas com computação paralela. A validação do uso de equivalentes de rede para o sistema de potência, modelado de forma digital, e a integração destes em ambientes de cálculo com múltiplos passos de tempo, traz robustez e flexibilidade nas simulações de transitórios eletromagnéticos. Outro ponto de destaque, fundamentado pela conexão de modelos digitais de circuitos elétricos com analógicos, é a possibilidade de integrar essa estrutura de grandes sistemas modeladas computacionalmente a hardware físicos, num conjunto do tipo **HIL**.

Gonçalves et al. [59] desenvolvem uma plataforma para cálculo de transitórios com processamento paralelo. Neste trabalho, a estrutura do software EMTP foi utilizada para a modelagem e solução dos sistemas de potência implementados. Associado a este solucionador está uma interface para comunicação denominada **FMI (Functional Mock-up Interface)**. Por meio desta **API** é possível estabelecer uma interface entre ferramentas de simulações, neste caso entre as instâncias do EMTP, em um ambiente paralelo de co-simulação usando o método mestre escravo [60]. E, por último, [59] escolhem uma primitiva de sincronização denominada semáforo.

Gonçalves et al. [59] realizam diferentes simulações em seu estudo de caso. Para tanto, utilizam como base o sistema de referência *IEEE 39 barras*, como linhas a parâmetros distribuídos e independentes da frequência, geradores síncronos, parques eólicos, dentre outros. O passo de cálculo, para todas as simulações, é 50 microssegundos. A implementação completa desse estudo é feita em um computador de quatro núcleos com sistema operacional Windows 7. Como resultados, Gonçalves et al. [59] apresentam um comparativo, em termos de tempo de simulação, entre a execução do cálculo sequencial versus as propostas de execução em paralelo. Os autores apresentam como resultado mais significativo, uma redução no tempo de simulação de 47,5%.

### 3.3.1 Considerações parciais

As técnicas apresentadas proporcionam, não somente a integração de ferramentas matemáticas disponíveis em softwares como os de **EMT**, mas também, a possibilidade de utilizar paralelismo dentro de paralelismo. Ou seja, além de paralelizar um problema dentro dos softwares do tipo **EMT**, as ferramentas disponíveis em outras plataformas, tal

como MATLAB, serem paralelizadas; o que, conseqüentemente, aumentará o desempenho global da solução do problema. As ferramentas matemáticas e interfaces computacionais só podem ser utilizadas se a estrutura do software do tipo EMT for adequada para tal intento. A seguir alguns destes softwares são apresentados.

### 3.4 Softwares “*off the shelf*” para simulação do tipo EMT em ambiente paralelo e distribuído.

O MATLAB é um software muito conhecido na literatura. Ele possui um gama de possibilidades, como, por exemplo, o cálculo numérico, processamento de sinais, sistemas de controle. Além disso, o MATLAB possibilita ao usuário a utilização de computação de alto desempenho para desenvolvimento de simulações, desde que as mesmas sejam paralelizáveis. A estrutura dele permite a solução de problemas com grandes quantidades de dados e com intensa demanda computacional. Isso é feito utilizando diferentes estratégias computacionais, as quais poderão utilizar processadores multinúcleo, unidades gráficas de processamento (GPU) e *clusters* [2].

A realização de computação paralela no MATLAB requer a aquisição de pacotes separados da licença padrão: pacote de processamento paralelo (em inglês: “*Parallel computing Toolbox*”) e, para o caso em que o estudo necessita de computação distribuída, há a necessidade de aquisição do software MDCS (*Matlab Distributed Computing Server*). Esses pacotes fornecem funções prontas para a realização da paralelização do software no MATLAB. Por exemplo funções de envio e recebimento de dados (“*labSend*”) e (“*labReceive*”); partição de tarefas em um loop (“*parfor*”). O suporte interno disponível para paralelização de um algoritmo é muito vasto; embora traga limitações, por exemplo, quanto ao número de instâncias (partes do algoritmo a serem processadas em núcleos individuais).

A Figura 3.1 ilustra a computação paralela em uma única máquina. Os elementos denominados “*worker*” são instâncias que resultaram da partição do problema original modelado no software.

No caso da computação distribuída, como visto na figura 3.2, para a montagem de um *cluster*, o software adquirido adicionalmente, MDCS, realizará o planejamento da divisão de tarefas nos núcleos de computadores distribuídos e a sincronização de resultados na arquitetura computacional.

O desenvolvimento da arquitetura de computação paralela com o MATLAB possui uma série de requisitos quanto a hardware e software [61]. Dentre estas, citam-se: recomenda-se 1 MATLAB worker por núcleo de CPU; mínimo de 2 GB de RAM por MATLAB worker; 10 GB de espaço em disco para a instalação do MDCS; 5GB em disco para

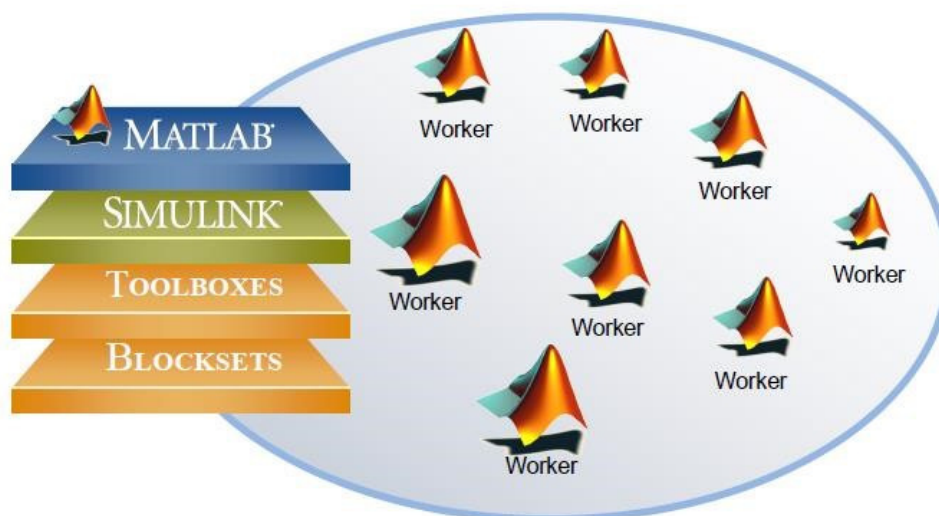


Figura 3.1 – Computação paralela no MATLAB.[1]

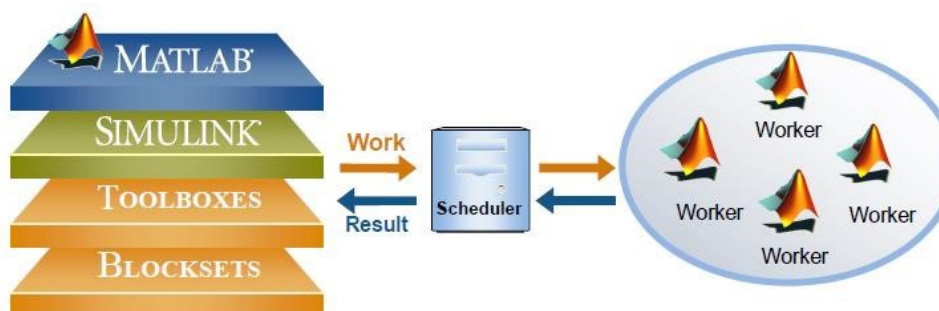


Figura 3.2 – Computação distribuída no Matlab. .[2]

acomodar dados temporários no computador onde o **MDCS** for instalado; diversas portas TCP são requeridas para intercomunicação entre os MATLAB *worker*. É importante destacar essas requisições do MATLAB, pois em um desenvolvimento de computação paralela e distribuída, independentemente de softwares comerciais, o usuário poderá se deparar com os mesmos requisitos.

Outra ferramenta de aceleração computacional no MATLAB são as funções que permitem o uso da **GPU** para operações matriciais [2]. Em muitos casos, o processamento na **GPU** é mais rápido que aquele realizado na **CPU**. Dependendo do tipo, uma **GPU** pode ter mais de 1000 núcleos disponíveis para a realização da computação de alto desempenho. Uma interface de comunicação entre **GPU** e **CPU** é necessária, o que é feito por funções já predefinidas no pacote de paralelismo. A unidade gráfica de processamento compatível com a plataforma do MATLAB é produzida pela NVIDIA. A estrutura completa de paralelismo disponível com o Matlab é ilustrada na figura 3.3. Dentro dessa estrutura é possível implementar a modelagem de cálculo de transitórios e a integração de computação de alto desempenho.

Um outro software disponível para cálculo de transitórios eletromagnéticos é o

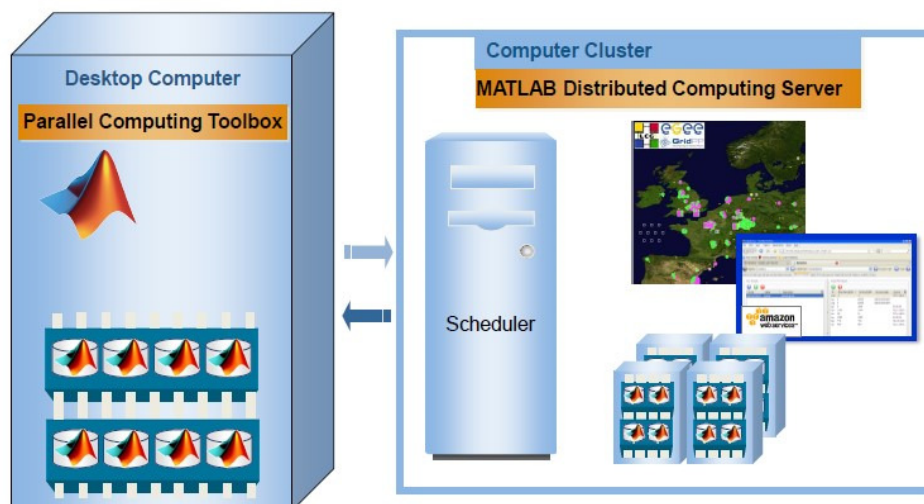


Figura 3.3 – Plataforma de computação paralela no MATLAB .[2]

**PSCAD/EMTDC**. O **PSCAD** é uma interface gráfica para o programa de cálculo de transitórios eletromagnéticos **EMTDC**. Juntos, os dois formam uma poderosa ferramenta para estudos acerca de eventos desse tipo nos sistemas de energia, como por exemplo: coordenação de isolamento de transformadores, chaves e para-raios; saturação de transformadores, faltas em linhas de transmissão e distribuição devido a descargas atmosféricas, design dos sistemas de controle de **FACT** e **HVDC**, acionamentos de máquinas elétricas, integração de geração distribuída à rede [20], [3].

O **EMTDC**, desenvolvido, inicialmente, por Dennis Woodford com a ajuda de outros pesquisadores, possui algumas peculiaridades que valem a pena serem destacadas neste trabalho:

- A preparação e o tempo de teste de estudo de caso são reduzidos devido a interface gráfica.
- Muitos elementos série e paralelo (por exemplo RLC) são colapsados matematicamente para reduzir a quantidade de nós e ramos.
- Algoritmo de otimização para aumentar a velocidade da decomposição LDU de matriz.
- Usuários do **EMTDC** podem escrever e desenvolver seu próprio modelo de componente. Interfaces para variáveis do programa e elementos de armazenamento são fornecidos.
- Os usuários do programa podem escrever em linguagem C, Fortran e Matlab.
- Permissão para alocação dinâmica no início do software.
- Modelos de linha e cabo do **EMTDC** são um dos mais completos do mercado.

- Inicialização de sistemas com Snapshot File.
- Algoritmo de otimização é utilizado em operações de chaveamento. Isso permite que o chaveamento ocorra no exato instante de chaveamento, mesmo que seja entre o passo de cálculo.

A figura 3.4 ilustra como é feito o cálculo de transitório no PSCAD.

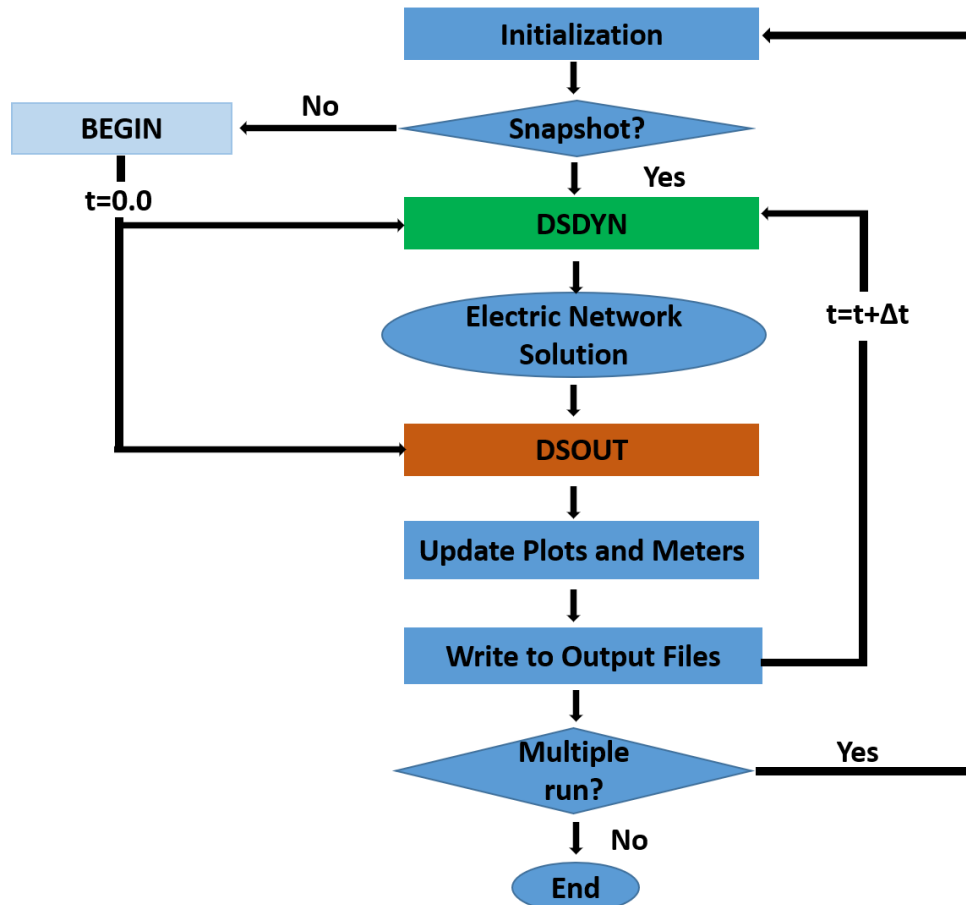


Figura 3.4 – Cálculo de transientes no PSCAD [3].

O cálculo de transientes é iniciado com a montagem do sistema elétrico no PSCAD. Essa interface traduz esse sistema e suas funcionalidades na forma de um arquivo Fortran, o qual será executado posteriormente. A estrutura deste é constituída por duas partes principais: uma dinâmica, formada pelas sub-rotinas DSDYN, DSDOUT e BEGIN; e uma segunda que representa a solução da rede elétrica. Em BEGIN, as variáveis são inicializadas e armazenadas. Na subrotina DSDYN, funções dinâmicas são executadas, as quais estão relacionadas com a preparação da rede elétrica, tal como fontes, e outros, e a manipulação de sinais de controle. Vale a pena ressaltar que nessa subrotina existe a possibilidade de serem anexados ao sistema elétrico eventuais componentes desenvolvidos pelo usuário, nos quais podem ser implementados dispositivos de comunicação com outros softwares em diferentes computadores. Ou ainda, a comunicação de diferentes partes

do sistema elétrico em diferentes processadores. Depois que o cálculo de transitórios é realizado, um pós-processamento dos dados é feito em DSOUT [3].

A versão comercial mais atualizada do PSCAD oferece aos usuários a possibilidade de utilizar computação de alto desempenho no cálculo de transitórios. Para isso ocorrer, é preciso que se tenha uma combinação balanceada entre os núcleos do computador e as instâncias (cada executável do EMTDC é uma instância gerada pelo PSCAD). No caso de o paralelismo usar mais de um computador, um terceiro software é necessário para fazer a conexão entre os mesmos, denominado IncrediBuild Xoreax Grid Engine (XGE) [4]. A figura 3.5 ilustra uma estrutura multiprojeto do PSCAD.

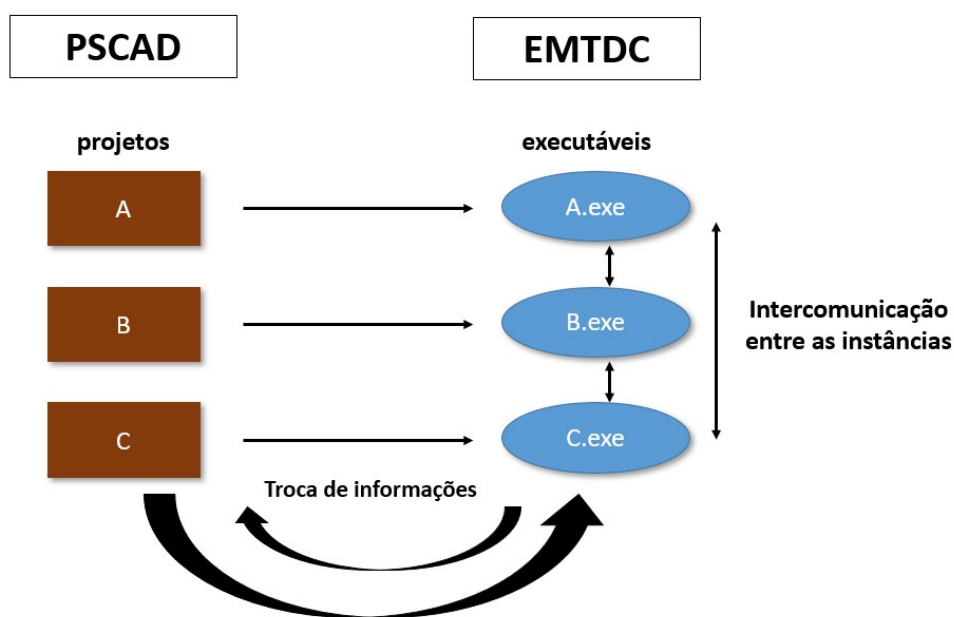


Figura 3.5 – Estrutura multiprojeto para simulação no PSCAD [4].

Os modelos de programação presentes no PSCAD são SPMD, e MPMD, também chamados de Paralelismo de dados (em inglês: “*Data Parallelism*”) e Paralelismo de tarefas (em inglês: “*Task Parallelism*”). Para o SPMD, múltiplas instâncias de uma simulação podem ser liberadas simultaneamente, em paralelo, com variações de dados de entrada. Isto irá acelerar a velocidade da simulação. A quantidade de aceleração depende da quantidade de núcleos do computador e a quantidade de instâncias a serem simuladas. Na plataforma, o nome dado para esse modelo de paralelismo é volley launch. Já no modelo MPMD, um grande projeto pode ser dividido em projetos menores, sendo que cada um dos subprojetos pode ser simulado em núcleos de um mesmo computador ou de vários. Na plataforma, o *Task Parallelism* é chamado de ENI [4]. As figuras 3.6 e 3.7 ilustram essas arquiteturas.

Manitoba Hydro International apresenta um estudo em que mede a aceleração de uma simulação paralela, SPMD, em relação a uma serial. Neste estudo, o número de processos foi sendo aumentado na simulação e, também, o número de núcleos disponíveis.

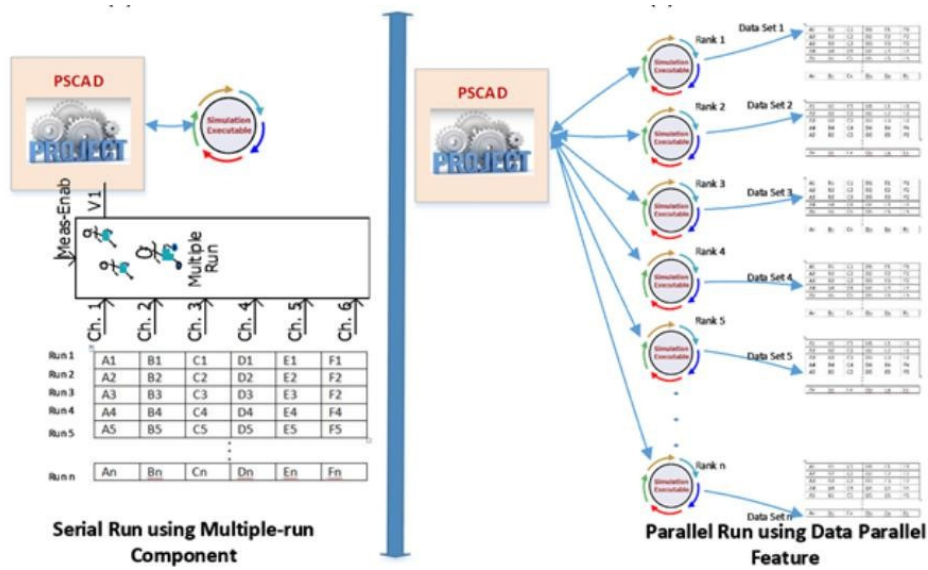


Figura 3.6 – Paralelismo de dados.[4]

Para a situação em que 40 casos foram simulados em 40 núcleos, o tempo de simulação com paralelismo foi 33 vezes maior que o serial, no caso em que se tinha 64 processos e 40 núcleos, o tempo de simulação foi 29 vezes maior. A aceleração não acompanhou o aumento do número de processos, isso se deve ao fato de que quando se tem mais processos que núcleos, aqueles devem esperar a disponibilidade do hardware para serem executados [4].

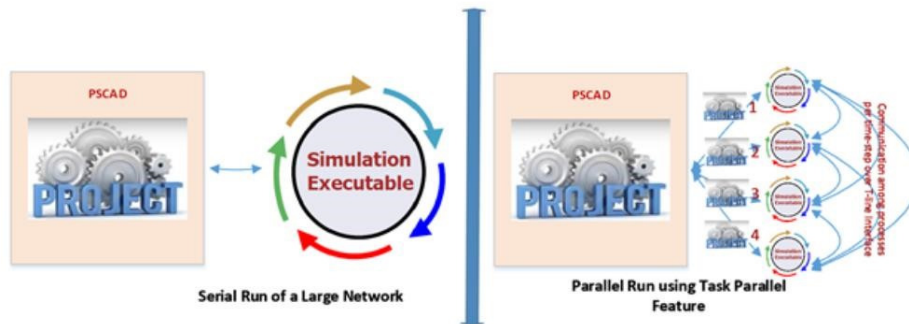


Figura 3.7 – Paralelismo de Tarefas.[4]

Neste mesmo documento, M. H. International aplica computação de alto desempenho, modelo MPMD, para realizar a simulação de uma Rede DC de nove terminais “Cigre benchmark”. A simulação paralela, usando nove núcleos foi 4,4 vezes maior que a serial. Vários estudos de caso são executados envolvendo paralelismo de dados e tarefas com vistas a confirmar essa característica do software. Além disso, uma discussão foi feita acerca da estrutura de comunicação utilizada na arquitetura do software, a qual, até aquele momento, era a TCP/IP. Uma nova estrutura de comunicação foi apresentada e testada, chamada INFINIBAND, através da qual se conseguiu reduzir, significativamente,

o tempo de comunicação entre instancias do software durante cálculo de transitórios multimáquinas.

M. H. International mostra que a chave para uma maior desempenho é alcançar uma baixa granularidade. Isto significa ter um tempo de comunicação, entre núcleos, menor que o tempo de computação nos mesmos. Com base nisso, discutiu-se a variação da granularidade conforme o tamanho do sistema a ser simulado. E, depois, realizou-se um comparativo entre a simulação de grandes sistemas utilizando o protocolo TCP/IP com a tecnologia INFINIBAND. Notou-se que esta última é muitas vezes superior à outra.

A possibilidade de se aplicar computação de alto desempenho ao cálculo de transitórios usando [PSCAD/EMTDC](#) é confirmada conforme as referências supracitadas. No entanto, muitas são as limitações para o pesquisador que deseja utilizar os pacotes desse programa. Dentre elas, têm-se: o fato de que a quantidade de instâncias possíveis de serem geradas pelo [PSCAD](#) está vinculada à licença comprada pelo usuário; a interface de comunicação multimáquinas, para aquele que não tem um conhecimento sobre programação, é a TCP/IP; para se desenvolver um cluster é necessário a aquisição de mais um software, [XGE](#). E, o principal, versões mais potentes do programa custam um preço elevado; questão crucial para muitos dos pesquisadores da área.

Porém, para aqueles pesquisadores que quiserem se envolver com pesquisas na área de computação de alto desempenho, existe a possibilidade de interagir com o software [PSCAD/EMTDC](#), desenvolvendo componentes, e montar sua própria plataforma computacional. Isto fornece a possibilidade de usar todos os recursos do [PSCAD/EMTDC](#), limitados a licença disponível do pesquisador, e expandi-los para um sistema com maior capacidade. Por exemplo, usar uma interface de comunicação diferente daquela disponível, no caso TCP/IP, e montar um sistema multimáquina. E ainda, fazer interface com outros softwares; característica essa já disponível para o caso de esse software ser o MATLAB.

Outro software para cálculo de transitórios é o [ATP](#). De acordo com [62] é o programa de simulação digital de fenômenos transitórios eletromagnéticos e eletromecânicos em sistemas de potência mais utilizado no mundo. Este software é um aperfeiçoamento do conhecido EMTP, cujas bases de cálculo são o trabalho proposto por [6]. O [ATP](#) possui um diferencial com relação às outras plataformas pois é um software livre.

O [ATP](#), assim como o [PSCAD](#), possibilita fazer a modelagem de complexas redes elétricas e sistemas de controle, com modelos já preestabelecidos de linhas de transmissão a parâmetros distribuídos e dependentes da frequência, resistência, capacitores, indutores, fontes de tensão, máquinas elétricas, transformadores e outros, componentes não lineares, dispositivos da eletrônica de potência. Além disso, possibilita ao usuário desenvolver componentes próprios e integrá-los ao cálculo de transitórios. Isto é definido na base do programa como uma estrutura denominada *Models*. Com este componente, além de componentes próprios é possível integrar a simulação de outros softwares ao [ATP](#), assim como



o que é realizado com o MATLAB e PSCAD. O ATP não possui uma estrutura pronta de computação paralela e distribuída, embora ela possa ser implementada com o auxílio de interfaces de softwares já consolidadas na literatura técnica, como Windows Socket, OpenMP, dentre outros.

A estrutura do simulador da empresa RTDS Technologies é apresentada por Watson et al. [12]. De acordo com este último, o simulador utiliza uma arquitetura composta por um ou mais racks, sendo cada um destes compostos por até 18 processadores. Dois tipos de cartões de comunicação são utilizados para executar as simulações, um cartão de comunicação com uma Workstation (WIC (*Workstation interface card*)) e um cartão de comunicação entre racks (IRC (*Inter-rack communications card*)). O software utilizado na simulação vai depender da aplicação para o qual foi designado o simulador.

### 3.4.1 Considerações parciais

A utilização de softwares "*off the shelf*" para simulações em arquiteturas computacionais envolvendo co-simulação é uma tendência forte; uma vez que permite a utilização de um recurso, já consolidado em uma área específica, contribuir em estudos mais amplos e multidisciplinares. Adicionalmente, uma grande parte desses softwares possuem funcionalidades que viabilizam a utilização de computação paralela e distribuída aos problemas que se propõem a resolver. Para o cálculo de transitórios eletromagnéticos, a utilização de softwares com essas características significa maior eficiência, flexibilidade e redução de tempo nas simulações.

## 3.5 Considerações finais

Neste capítulo foi apresentado o estado da arte acerca da simulação de transitórios eletromagnéticos em grandes sistemas utilizando computação paralela e distribuída. Por meio de estudos realizados por pesquisadores ao redor mundo, ficou evidente que o primeiro passo para integrar essas duas áreas foi compreender em que ponto a natureza do cálculo proposto é paralelizável; o que se confirmou no modelo de linha de transmissão no contexto de sistemas de potência. Por causa disso, uma área que já era ampla e consolidada para os cientistas da computação, pode ser integrada às simulações que envolvem sistemas de energia.

Diante das referências escolhidas, foi possível perceber quais os aspectos e situações que levaram os pesquisadores, na área de transitórios eletromagnéticos, avançarem nessa direção. Muitos são os motivos, mas como síntese destes destacam-se os componentes não-lineares presentes na rede; os dispositivos da eletrônica de potência com alta frequência de chaveamento; as gerações renováveis e distribuídas, solar e eólica, que trazem alta penetração de conversores; os conceitos de Smart Grids; as simulações em tempo

real e a própria natureza do transitório eletromagnético, o qual se situa num curtíssimo intervalo de tempo, da ordem de microssegundos. Somado a esses, está **a oportunidade do trabalho colaborativo, o qual proporciona inúmeras vantagens no âmbito de projetos multidisciplinares.**

O contexto das análises a serem feitas em torno do estudo de transitórios eletromagnéticos exige que o simulador de tais eventos tenha como características principais exatidão e precisão, a flexibilidade e que seja escalável; definindo assim, a robustez do simulador. Seguindo nessa direção, estão os simuladores aqui apresentados: [ATP](#), [PSCAD](#), [MATLAB](#), [RTDS](#) e [eMEGAsim](#), da OPAL-RT.

Sendo assim, pode-se concluir que o investimento em computação paralela é um direcionamento positivo quando se trata das requisições para simuladores do setor elétrico; o que traz oportunidades, não somente de eficiência computacional das simulações, mas, também, outras possibilidades como a integração de subsistemas com dinâmicas de processamento diferentes; trabalhos colaborativos de pesquisa com manutenção de propriedade intelectual por parte dos atores do projeto; desenvolvimento de arquiteturas computacionais para estudos multidisciplinares.

Diante do que foi apresentado nesta revisão bibliográfica, ***uma lacuna sobre cálculo de transitórios eletromagnéticos com interação de subsistemas elétricos sob demanda foi identificada***; sendo este o assunto que essa tese se propôs a investigar.

## 4 Arquitetura computacional distribuída, colaborativa para cálculo de transitórios eletromagnéticos com interação sob demanda

Este capítulo apresenta a arquitetura computacional projetada nesta tese; duas propostas metodológicas para a operação dela, definidas como protocolo de operação síncrono e sob demanda; um comparativo entre simulações realizadas com esta arquitetura e os softwares [ATP](#) e [PSCAD/EMTDC](#). Adicionalmente, apresenta um Mapa de complexidade – ferramenta pedagógica para auxiliar pesquisadores no desenvolvimento de projetos deste tipo de arquitetura computacional.

### 4.1 Características gerais da arquitetura computacional

O projeto da arquitetura computacional desta tese está fundamentado na premissa de que é possível desenvolver uma solução para o cálculo de transitórios eletromagnéticos em que um Subsistema A, executando continuamente em um computador, pode agregar um Subsistema B, executando em outro computador; com ambos formando um sistema maior. A esta metodologia, dá-se o nome de simulação colaborativa sob demanda. Diferentemente desta, está a simulação síncrona, em que o primeiro subsistema inicia o cálculo, mas mantém a execução parada até que o subsistema B se conecte a ele; continuando somente após a conexão entre ambos.

Outra característica da arquitetura é ser distribuída, ou seja, realiza a solução de problemas utilizando computação paralela agregando mais de um computador para tal intento. E, somado a isso, está a proposta de que seja colaborativa. Assim sendo, o cálculo de transitórios eletromagnéticos ocorre pela integração de dois subsistemas que, a princípio, são inteiramente independentes, mas que se unem para formar um terceiro em um dado cenário ou conjunto de eventos.

Além das características supracitadas, é importante definir o conceito de arquitetura computacional, qual seja o conjunto de funções, softwares, hardware e procedimentos metodológicos que permite a ela atingir o objetivo para o qual foi projetada. Somente após a compreensão dessas definições é possível iniciar a implementação dela, como prevista nesta tese. Para tanto, é necessário decidir sobre os parâmetros que compõem a estrutura dessa arquitetura, os quais enumeram-se a seguir juntamente com as escolhas para a tese:

Os parâmetros apresentados na Tabela 4.1 são responsáveis pelo nível de desempe-

Tabela 4.1 – Parâmetros para desenvolvimento de uma arquitetura computacional para cálculo de transitórios eletromagnéticos.

Parâmetros		Escolhas para a tese
1	O tipo de problema que será analisado na simulação.	Cálculo de transitórios eletromagnéticos em sistemas de energia.
2	A linguagem de programação.	Linguagens C e Fortran.
3	O modelo de programação paralela.	Múltiplos programas com múltiplas entradas(“MPMD”).
4	O hardware computacional disponível, ou seja, a capacidade de processamento dos computadores envolvidos no projeto. Deve ser considerado o número de núcleos do processador.	Computador com processador multinúcleo, quatro ou mais.
5	Interface de comunicação	Windows Sockets (Winsock)
6	Protocolos de comunicação para troca de dados entre simulações distribuídas.	Protocolos TCP/IP
7	Se a rede de computadores será local ou geograficamente distribuída.	A rede foi desenvolvida para uma integração local ou distribuída.
8	Software “off the shelf” utilizado na solução do problema em análise. Este parâmetro pode ser excluído caso o profissional decida fazer programação própria.	PSCAD/EMTDC

nho e robustez que a arquitetura computacional pode atingir. Ou seja, redução no tempo de simulação quando é feita a comparação entre computação paralela com a plataforma e uma computação sequencial; alcance de resultados precisos e exatos, elevação do grau de escalabilidade, integração de simulações entre diferentes usuários. No conjunto de parâmetros apresentados, três assumem papel essencial para arquitetura computacional proposta: O software [PSCAD/EMTDC](#), os protocolos TCP/IP associados a interface Socket.

O primeiro deles é o software [PSCAD/EMTDC](#), cuja escolha foi influenciada pela enorme variedade de modelos de componentes do sistema de potência; por permitir o desenvolvimento de componentes internos, os quais poderiam ser integrados à formação da equação matricial do cálculo de transitórios eletromagnéticos e promover a conexão entre instâncias do software localizadas em computadores distintos. A escolha pelo [PSCAD/EMTDC](#) como software, definiu, conseqüentemente, as linguagens de programação a serem utilizadas no projeto – linguagens de programação C e Fortran.

A presença de um componente personalizado, que se integra ao cálculo de transitórios do software e permite conexão com simulações externas, é uma das peças-chave desse projeto, pois através dele será possível integrar subsistemas formando um grande sistema elétrico. Para tanto, este dispositivo utilizará a teoria de linhas de transmissão, estabelecendo pontos de integração entre os subsistemas e trocas de dados. Define-se aqui

a possibilidade de computação paralela e distribuída aplicada ao cálculo de transitórios na arquitetura.

O mecanismo de comunicação escolhido para o estabelecimento de comunicação foi o Socket associado aos protocolos TCP/IP. Essa escolha foi feita porque, além de consolidados tecnicamente, podem ser implementados em diferentes hardwares de computadores, e vários tipos de sistemas operacionais; dando um caráter de interoperabilidade para a arquitetura computacional. Outra contribuição dos protocolos TCP/IP é viabilizar a integração de computadores tanto em uma LAN quanto em uma WAN; com ilustrações nas Figuras 4.1 e 4.2. Sendo assim, áreas para além do local de um laboratório poderão ser abrangidas, por exemplo, computadores em diferentes cidades.

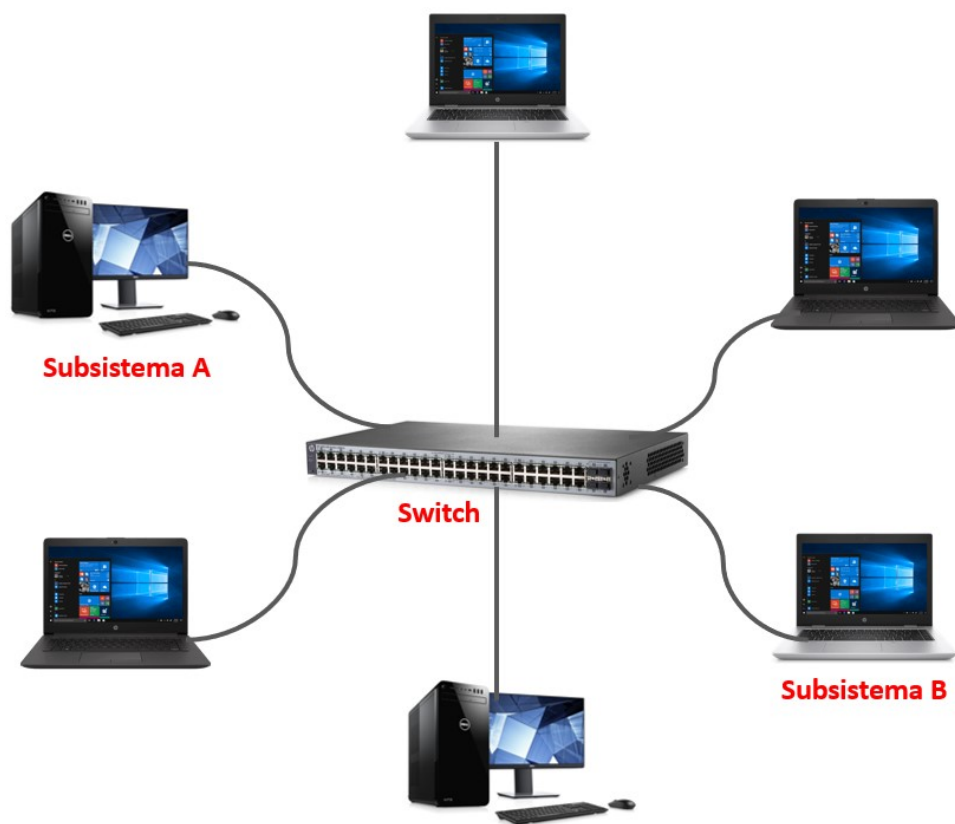


Figura 4.1 – Ilustração para uma LAN

Neste tópico foram apresentadas as características gerais da arquitetura computacional desenvolvida nesta tese. Nos tópicos a seguir, componentes específicos dela serão descritos detalhadamente.

## 4.2 Modelo de Componente personalizado — HCoE.

O componente personalizado HCoE (*Hub Co-simulation Engine*) foi desenvolvido para permitir que o cálculo de transitórios eletromagnéticos fosse feito de forma distribuída e colaborativa. Para tal intento, ele assume, no software, uma representação na forma de

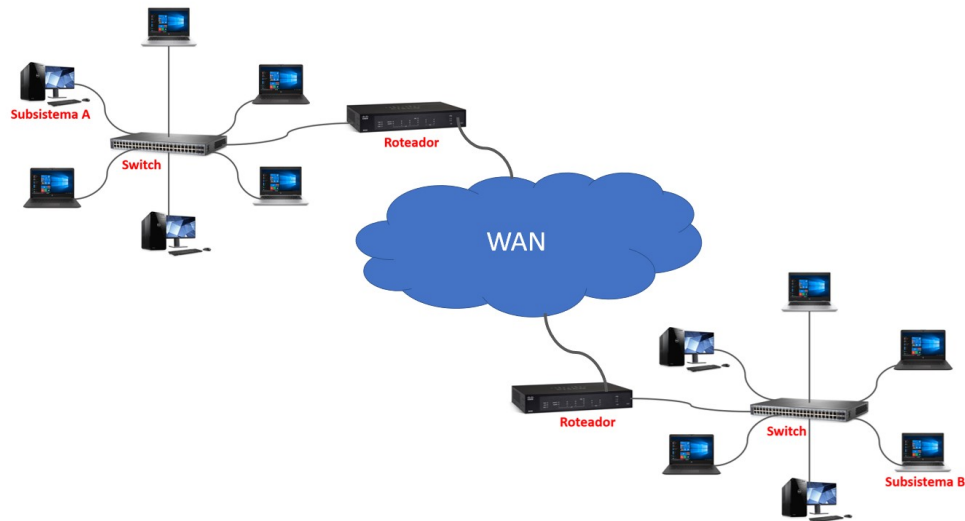


Figura 4.2 – Ilustração para um WAN.

uma linha de transmissão trifásica, cujo modelo elétrico é o Bergeron. Como se sabe, da teoria de ondas viajantes no cálculo de EMT, a presença de uma linha de transmissão entre dois pontos distintos do circuito causa um desacoplamento natural entre os subsistemas conectados à extremidade dela, possibilitando, assim, a computação paralela. Por isso, para a integração entre dois subsistemas, são necessários dois componentes HCoE.

O software PSCAD/EMTDC realiza o cálculo de EMT através da formação de uma equação matricial cujos elementos de circuitos são representados pelos seus respectivos equivalentes Norton. Para o componente HCoE, essa integração é feita por meio de um conjunto de admitâncias que caracterizam a linha e fontes de corrente monofásicas. Estas correntes são calculadas com base na teoria de transformação modal. A Figura 4.3 ilustra o procedimento para integração do modelo de linha trifásica ao cálculo de EMT. As tensões de nó, no domínio das fases, são manipuladas no domínio dos modos dando origem a correntes modais. Estas por sua vez são manipuladas, por meio das equações modais, para a obtenção das correntes de fase que serão injetadas na rede elétrica à qual a linha se encontra conectada.

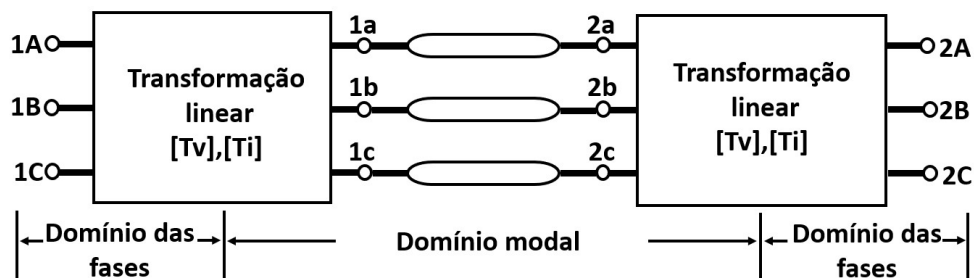


Figura 4.3 – Transformação entre os domínios modal e fase para uma linha trifásica.

Durante o cálculo de transitórios eletromagnéticos desacoplado, executando em dois computadores, o modelo HCoE realiza uma troca de vetores de correntes históricas

modais. Estes dados são necessários para o cálculo das correntes de fase que serão injetadas em cada extremidade da linha de transmissão. Até esse ponto, funções se tornam necessárias no algoritmo que representa este componente: a primeira, está relacionada ao cálculo dos parâmetros modais da linha e as correntes que serão injetadas na rede elétrica; a segunda, tem o papel de fazer a transferência de correntes históricas modais entre os subsistemas. Estas últimas, são utilizadas para fazer o cálculo das correntes injetadas nos terminais da linha de transmissão.

Ainda sobre a conexão do componente HCoE à rede elétrica, um ponto importante a ser destacado é que ela ocorre por meio de disjuntores em ambas as extremidades da linha. Isto garante que nas situações onde os subsistemas não estão conectados, o sistema elétrico original pode executar o cálculo de transitórios sem a influência das admitâncias do circuito do HCoE. Esta proposta foi estabelecida porque o circuito que representa a linha de transmissão, com interfaces conectadas aos nós elétricos nas extremidades da linha somente têm sentido quando já foi estabelecido um canal de comunicação entre os subsistemas, fazendo com que as simulações estejam sincronizadas. Nesse ponto ocorre a necessidade de uma terceira definição para os componentes, um deles deve ter um algoritmo de comunicação com o papel de servidor e, o outro, com o papel de cliente. E, assim, ficou definido.

A transferência de dados entre as interfaces do circuito é feita através do mecanismo de comunicação WinSocket. Esta API permite a conexão entre processos hospedados em computadores distintos, desde que estes últimos estejam acessando a Internet. Para tanto, o algoritmo deve ser identificado por um endereço, "IP", e uma porta de conexão, por meio da qual os dados serão enviados e recebidos.

A escolha do protocolo TCP/IP para o componente garante quatro características fundamentais para a comunicação de dados na arquitetura computacional: orientação a conexão, em que dois processos podem fazer transferência de dados somente depois que a conexão entre ambos foi estabelecida; confiabilidade, garante que a entrega do pacote de dados ao destino ocorreu sem erro e na ordem que foi transmitida; full-duplex, após a conexão entre os processos, os dados podem ser transferidos de forma bidirecional; controle de fluxo, no qual o receptor confirma o recebimento dos pacotes à medida que vai recebendo.

A comunicação entre os subsistemas na plataforma é estruturada de acordo como o modelo cliente-servidor. Um dos subsistemas terá uma interface programada para ser o servidor e a outra, como o cliente. O projeto da função que executa a comunicação, a qual faz a transferência de dados elétricos, foi feito por um algoritmo que bloqueia a simulação enquanto a conexão entre cliente e servidor não é estabelecida.

A Figura 4.4 ilustra a integração do componente HCoE ao cálculo de EMT na plataforma do PSCAD/EMTDC. No computador 1, por exemplo, o HCoE terá caracte-

rísticas de servidor e, no computador 2, cliente. Note que o componente está ligado a uma estrutura denominada “DSDYN”. É nesse ambiente computacional que é feita a representação de componente personalizado pelo usuário como dispositivo elétrico. Com isso, o software integra esse componente a solução da rede elétrica para o cálculo de EMT.

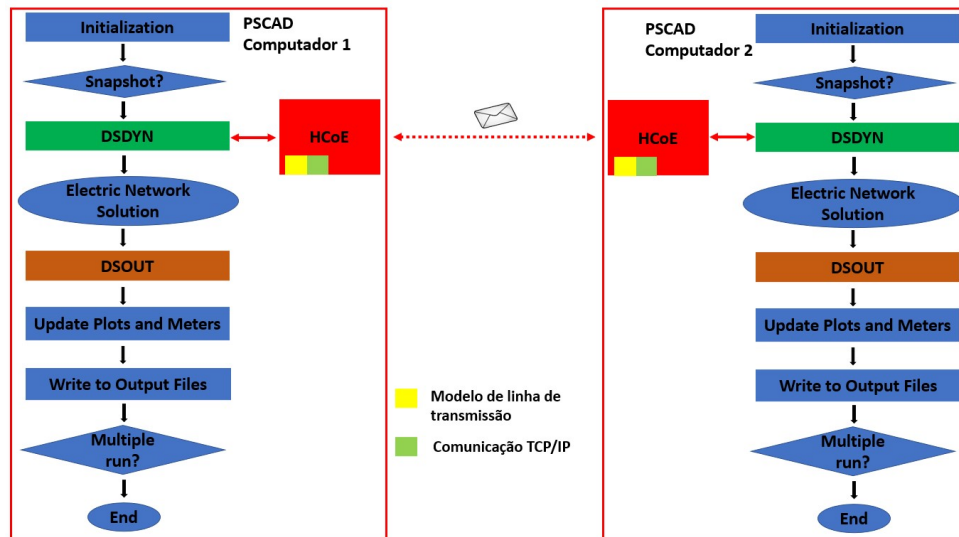


Figura 4.4 – Integração dos componentes HCoE cliente e servidor ao cálculo de transitórios eletromagnéticos do PSCAD/EMTDC.

A estrutura apresentada na Figura 4.4 é suficiente para o cálculo de transitórios eletromagnéticos de forma síncrona e distribuída, discutiremos esse algoritmo mais a frente (*Algoritmo 3*). Neste caso, o algoritmo que descreve o componente HCoE servidor e HCoE cliente se diferem apenas nas funções que fazem a transferência de dados elétricos. O computador 1 precisa ser inicializado primeiro, o que é consequência da comunicação modelo cliente-servidor, em que o servidor aguarda o contato do cliente para iniciar a troca de informações.

A operação no modo *sob demanda* demandou o desenvolvimento de novos elementos para arquitetura computacional proposta na tese, quando comparado à estrutura da Figura 4.4. Para tanto, ela é inspirada no sistema elétrico e seus centros de operação do sistema, também chamados de COS (*Centro de Operação do Sistema*). Estes órgãos sendo responsáveis pela tomada de decisão de manobras nos sistemas de energia, como as que serão experimentadas neste trabalho. Na Figura 4.5 têm-se uma ilustração da arquitetura computacional *sob demanda*.

A primeira modificação foi o “Centro de Operação do Sistema (COS) – servidor”. Este módulo é um algoritmo cuja função é receber um conjunto de informações que definem o pedido de conexão do cliente: a barra do subsistema A que o cliente quer se conectar; um parâmetro denominado janela temporal, o qual define o tempo em que os subsistemas cliente e servidor ficarão conectados. Esses dados são transferidos para outro módulo, denominado “Gerente de conexão”. Depois que o sistema controlado pelo COS-



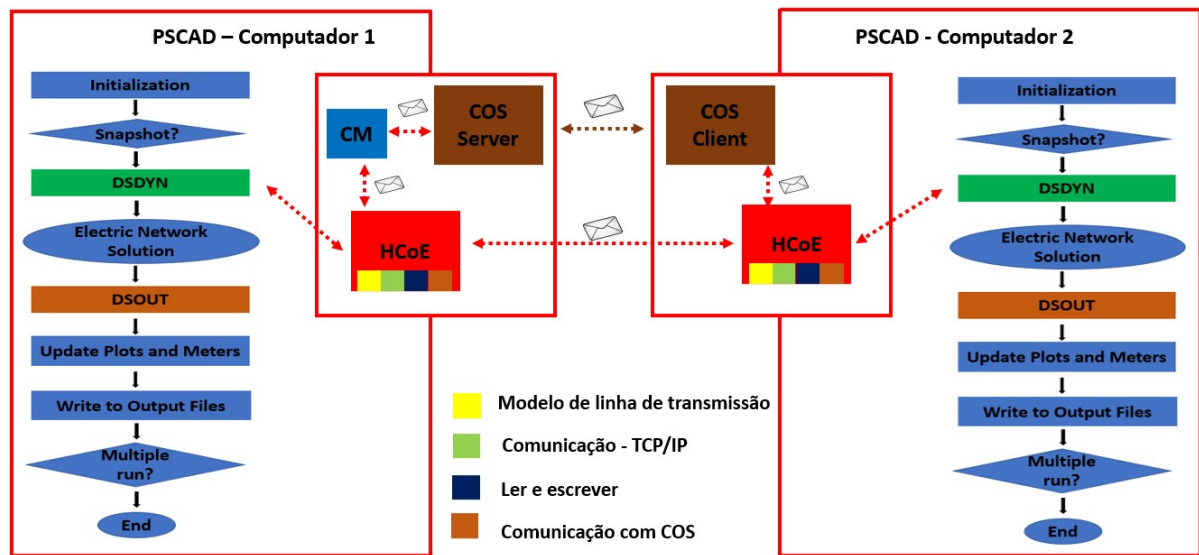


Figura 4.5 – Arquitetura computacional para metodologia *sob demanda*.

servidor está pronto, este módulo envia uma resposta para o cliente habilitando a conexão entre os subsistemas cliente e servidor.

A segunda foi o “Centro de Operação do Sistema (COS) – cliente”. Este módulo é um algoritmo cuja função é enviar dados do subsistema cliente para o servidor: barra a ser conectada e janela temporal. Este software mantém o cálculo de EMT bloqueado até que a resposta do COS-servidor chegue habilitando a conexão entre os subsistemas envolvidos. Vale ressaltar que a comunicação entre os módulos COS, também, é feita por meio da API WinSocket.

A terceira foi o desenvolvimento de um módulo denominado “Gerente de conexão”. Este componente têm a função de habilitar o componente HCoE, associado à barra que o cliente quer se conectar, e definir os instantes de abertura e fechamento do disjuntor conectado a este último componente. Neste módulo, um ponto importante a esclarecer é que a atividade habilitar significa introduzir a função que modela o componente HCoE como uma linha de transmissão ao cálculo de EMT e permitir a formação do canal de comunicação; do contrário ele é apenas um conjunto de admitâncias sem definição para o software PSCAD/EMTDC. Esse componente melhora o desempenho do software, porque somente o componente conectado à barra de interesse integrará a equação matricial do cálculo de transitórios eletromagnéticos; o que diminui o tempo de simulação.

Além de habilitar a conexão em determinada barra e a manobra dos disjuntores, o módulo Gerente de conexão, comunica com o COS-servidor para mostrar que o subsistema do servidor está pronto para a conexão do cliente. Isso significa que o cálculo de EMT está parado esperando a conexão do subsistema cliente.

A quarta modificação ocorreu no componente HCoE do subsistema cliente. Além das funções que definem a linha de transmissão trifásica e a comunicação cliente-servidor,

foram integradas ao módulo do componente HCoE uma função para calcular a janela temporal de conexão entre cliente e servidor; uma função que faz a comunicação entre HCoE e COS-cliente, permitindo a este último controlar o instante que aquele irá estabelecer um canal de comunicação com o servidor.

A janela temporal assume um papel muito importante na metodologia *sob demanda* proposta para a arquitetura. Pois é através dela que o Gerente de conexão define o intervalo de tempo que o subsistema servidor manterá a conexão com o cliente. Para calculá-la são necessários três parâmetros: O passo de cálculo, o tempo final da simulação, e, no caso deste trabalho, o comprimento do vetor de dados elétricos de sequência positiva. Os cálculos dela basearam-se na seguinte pergunta: Após a conexão do cliente ao servidor, quanto tempo decorrerá até que a última troca de dados entre eles ocorra? Por exemplo:

**Algoritmo 1: Exemplo para cálculo de janela temporal**

- Seja o passo de cálculo  $\Delta t = 34 \times 10^{-6} s$ .
- Comprimento do vetor de sequência positiva  $N_{POS} = 40\Delta t = 0.00136s$ .
- Tempo final definido pelo usuário  $3,0s$ .
- Tempo final definido pelo software PSCAD  $t_{vfinal} = 2,9999s$ ;
- $aux1 = \frac{t_{vfinal}}{\Delta t} = 88232,352$ .
- $aux2 = int(\frac{aux1}{N_{POS}}) \times N_{POS} \times \Delta t$ .
- Janela temporal =  $2,9998s$

A janela temporal estabelece o tempo em que os subsistemas cliente e servidor estarão conectados através do canal de comunicação, o que não significa, necessariamente, a conexão elétrica entre eles. Pois, os disjuntores podem estar abertos. O Gerente de conexão definirá os instantes de manobra, mas o funcionamento global da arquitetura computacional impõe algumas regras, por exemplo: os disjuntores fecham somente após a conexão cliente-servidor; a abertura dos disjuntores deve acontecer antes que seja finalizada a comunicação entre os subsistemas.

O tempo final da simulação no PSCAD/EMTDC não é exatamente o tempo definido pelo usuário. Há um algoritmo que relaciona o passo de cálculo, o tempo definido pelo usuário e o instante em que a simulação finalizará de fato. O tempo final é um múltiplo inteiro do passo de cálculo, cujo valor pode ser maior, igual ou menor que o tempo

final fornecido pelo usuário. Isso vai depender da parte fracionária da divisão do tempo final pelo passo de cálculo.

A quinta modificação foi feita no módulo **HCoE** servidor para habilitar a comunicação com o algoritmo do Gerente de conexão; que consolida todos os componentes necessários para a definição da operação *sob demanda* da arquitetura proposta. No próximo tópico, os protocolos de funcionamento desta última serão apresentados.

### 4.3 Protocolos de operação da arquitetura computacional nos modos de operação síncrono e *sob demanda*.

O cálculo de transitórios eletromagnéticos síncrono será apresentado para que o leitor possa fazer um comparativo entre os cálculos síncrono e *sob demanda*. Antes disso, o cálculo de **EMT** do **PSCAD/EMTDC** será revisado no Algoritmo 2.

#### **Algoritmo 2: Cálculo de transitórios eletromagnéticos no PSCAD/EMTDC**

- *Inicialização e armazenamento de variáveis.*
- *Snapshot: Possibilidade de começar a simulação atual, a partir de um tempo  $T_1$ , considerando que o mesmo circuito elétrico atual tenha sido simulado e dados da simulação tenham sido gravados.*
- *DSDYN: Preparação de circuitos elétricos dinâmicos, como exemplo uma fonte, e manipulação de sinais de controle.*

*“É neste momento que os algoritmos dos módulos desenvolvidos para arquitetura computacional são integrados ao algoritmo do software.”*

- *Solução da rede elétrica.*
- *Preparação das variáveis de saída.*
- *Atualização de dados medidos.*
- *Escrita de dados de saída em arquivos.*
- *Verifica se haverá a repetição de algum cenário. E finaliza.*

A definição de um protocolo para o cálculo de transitórios eletromagnéticos síncrono considera as regras do software **PSCAD/EMTDC**, dos protocolos de comunicação TCP/IP e a **API WinSocket**. A premissa base da simulação síncrona distribuída é a de

que dois subsistemas, sendo executados em computadores distintos, começam e terminam um cenário de estudo no mesmo intervalo de tempo. Sendo assim, considere para este entendimento a Figura 4.6 e o Algoritmo 3 a seguir:

Para uma melhor compreensão dos protocolos de operação considere algumas definições: Três Computadores são utilizados, 1, 2 e 3; o subsistema servidor será denominado Subsistema A e os subsistemas cliente, B e C.

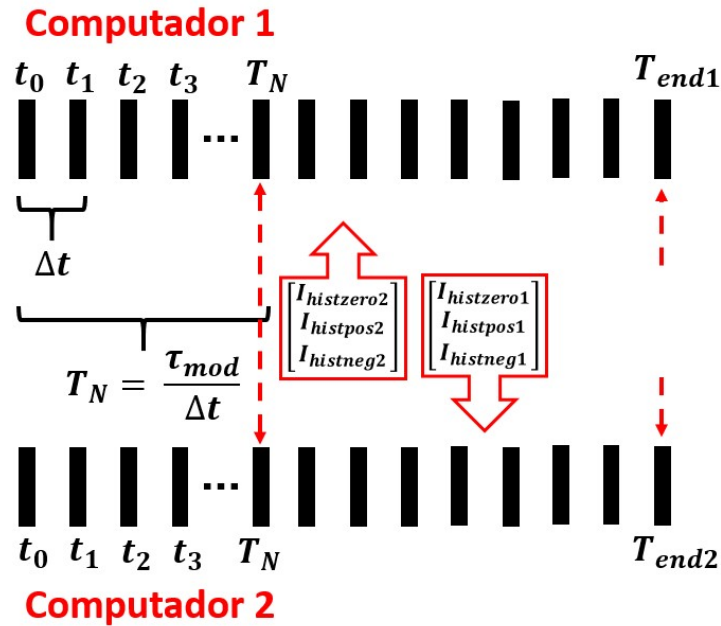


Figura 4.6 – Ilustração do cálculo de EMT síncrono.

**Algoritmo 3: Protocolo de operação síncrono**

- *Inicie a simulação no Computador 1. Este dispositivo hospeda um Subsistema A, com HCoE-servidor. A simulação espera até que algum subsistema cliente entre em conexão com o servidor. “Neste caso há a possibilidade de conexão apenas em uma barra. A simulação fica bloqueada, conforme o modelo de programação feito com WinSocket, até que se estabeleça o canal de comunicação cliente-servidor.”*
- *Inicie a simulação no Computador 2. Este dispositivo hospeda um Subsistema B, com o HCoE-cliente. Após a conexão entre cliente e servidor ser estabelecida, inicia-se a troca de dados elétricos entre os componentes. Estes são formados por vetores de correntes históricas modais. As trocas ocorrem a cada intervalo de tempo  $T_N$ . Este parâmetro é calculado pela relação entre o menor tempo de propagação modal  $\tau_{mod}$ , referente a linha de transmissão, e o passo de cálculo  $\Delta t$ ;  $T_N = \frac{\tau_{mod}}{\Delta t}$  ;*
- *Depois de um tempo,  $T_{COMPUTER2}$ , o disjuntor no computador 2 pode ser fechado; “Este tempo deve ser após o estabelecimento do canal de comunicação entre os subsistemas e antes do término da simulação. Obedecendo esta premissa, pode ser definido arbitrariamente pelo usuário.”*
- *Depois de um tempo,  $T_{COMPUTER1}$ , os subsistemas A e B podem ser conectados com o fechamento do disjuntor do subsistema A; “Este tempo também deve ser após o estabelecimento do canal de comunicação entre ambos os subsistemas.”*

O protocolo síncrono serve como um exemplo comparativo a fim de que o leitor tenha uma melhor compreensão do que é o funcionamento *sob demanda* da arquitetura computacional. Na figura 4.7 têm uma ilustração da execução de três computadores durante o protocolo *sob demanda*; com alguns instantes em destaque:  $t_0$ , início do cálculo de EMT;  $T_N$ , instante em que ocorre a troca de dados;  $T_{d1}$ , abertura do disjuntor no Subsistema A;  $T_{end2}$ , final da simulação do Subsistema B;  $T_{d2}$ , abertura do disjuntor no Subsistema A após a conexão do Sub. C;  $T_{end1}$ , final da simulação do Sub. A;  $T_{end3}$ , final da simulação do Sub. C. Todos esses instantes de tempo serão mais bem compreendidos no protocolo de funcionamento *sob demanda* e durante resultados de simulações (Seção 5.3).

Na Figura 4.7 é apresentado um esquemático resumido da arquitetura computacional para servir de auxílio no entendimento do protocolo de funcionamento *sob demanda*.

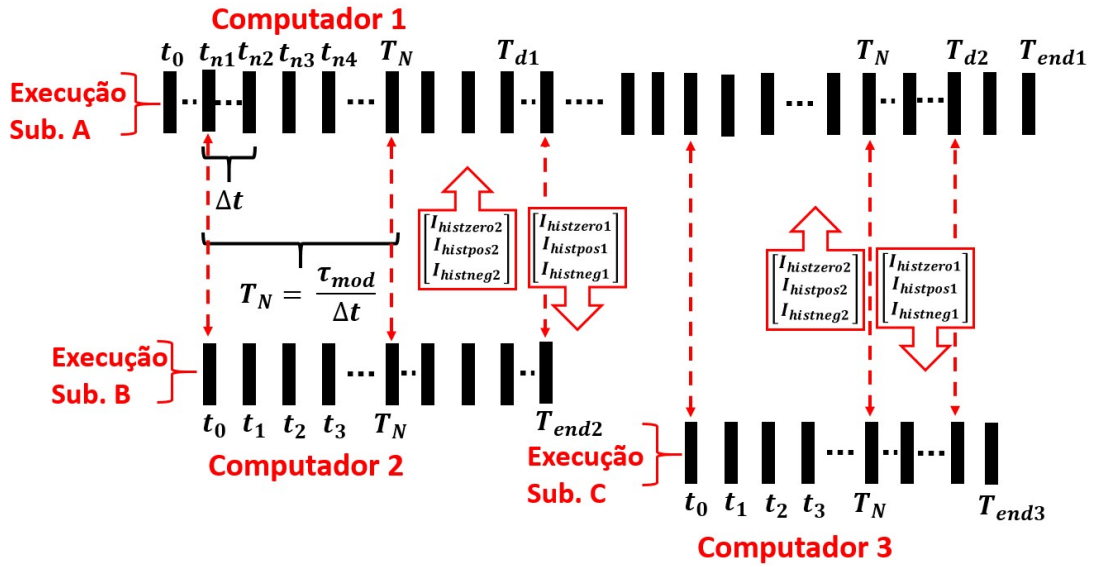


Figura 4.7 – Ilustração do cálculo de EMT *sob demanda*.

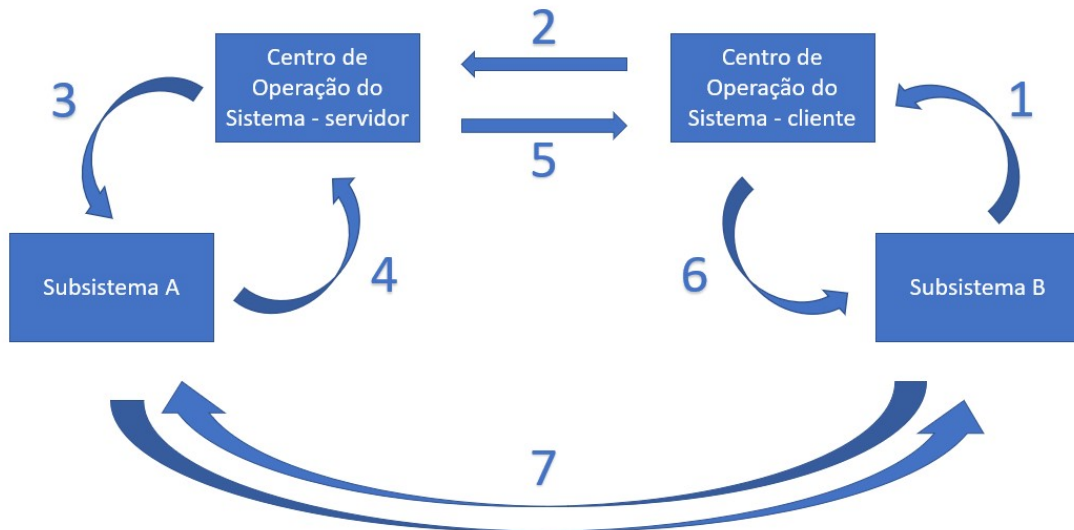


Figura 4.8 – Arquitetura *sob demanda* com os passos do protocolo de funcionamento.

**Algoritmo 4: Protocolo de operação sob demanda**

- 0 - Inicie a simulação no Computador 1. Este dispositivo hospeda um Sub-sistema A, com HCoE-servidor. O COS-servidor é inicializado.  
 “Neste caso a simulação funciona continuamente até a conexão do sub-sistema cliente.”
- 1- Inicie a simulação no Computador 2 e o COS-cliente. Este dispositivo hospeda um Sub-sistema B, com o HCoE-cliente. O sub-sistema B envia uma mensagem para o COS-cliente com os dados da barra que pretende se conectar e o valor da janela temporal.
- 2- O COS-cliente envia uma mensagem para o COS-servidor para entregar a requisição do sub-sistema B.

**Algoritmo 4: Protocolo de operação sob demanda**

- 3- O COS-servidor envia os dados recebidos para o gerente de conexão, o qual irá habilitar o componente HCoE associado à barra que o cliente pretende se conectar.
- 4- O componente HCoE da barra adequada abre o canal de comunicação via socket. E, depois, o gerente de conexão envia uma mensagem para o COS-servidor dizendo estar pronto.

“A simulação do subsistema A fica bloqueada esperando a conexão do subsistema B”.

- 5- O COS-servidor envia uma mensagem ao COS-cliente dizendo que o subsistema B pode se conectar ao subsistema A.
- 6- O COS-cliente envia um comando para o subsistema B habilitando a abertura do canal de comunicação com o subsistema A.
- 7- O canal de comunicação entre componentes HCoE é iniciado. Neste momento o modelo de linha está definido. Agora os disjuntores podem ser fechados. No subsistema B, isso deve ocorrer após 0,3 s, que é o tempo para o sistema atingir o regime permanente, e no subsistema A, definiu-se que o fechamento seria um pouco depois da chegada do sinal de tensão à extremidade da linha. **A abertura do disjuntor no subsistema A deve ser antes da última troca de dados entre os componentes HCoE, pois é neste momento que ocorrerá o fechamento do canal de comunicação e o servidor continuará a simulação dinamicamente.** Já no subsistema B, o disjuntor pode ser aberto no mesmo instante que ocorreu a manobra no subsistema A.
- No subsistema B, a simulação terminou. Mas, no subsistema A, a simulação continua. No caso deste último, o COS-servidor espera o pedido de conexão por parte de um novo subsistema C, hospedado em um Computador 3, por exemplo.

“Neste caso, a disponibilidade do subsistema A depende do tempo total de simulação.”

- Após a conexão do subsistema C, os passos que se seguem no protocolo são iguais aqueles adotados após a conexão do subsistema B.

Os protocolos apresentados fornecem um direcionamento para quem pretende utilizar a arquitetura computacional desenvolvida.

O projeto de uma arquitetura computacional *sob demanda*, colaborativa e distribuída para cálculo de EMT é um problema complexo e multidisciplinar. Consequentemente, traz inúmeros desafios para que um pesquisador, formado em uma área específica, possa desenvolvê-lo e avançar com pesquisas em torno do tema. Devido a isso, foi elaborado, também nesta tese, uma ferramenta pedagógica denominada Mapa de Complexidade, cujo objetivo é auxiliar estes profissionais quando forem trabalhar com este tipo de problema.

## 4.4 Mapa de complexidade da arquitetura computacional proposta na tese

A proposta da arquitetura computacional possui uma série de funções que são interdependentes. Devido a isso, uma pequena modificação em uma delas pode afetar todas as outras. Esta característica se aproxima muito da definição de problemas complexos. Buscando facilitar o trabalho de profissionais em torno desse tema, um Mapa de complexidade é proposto para esta plataforma. Nesta ferramenta, serão apresentados parâmetros e funções associados ao projeto que caracterizam o problema. Nas tabelas 4.2 e 4.4 são apresentados os parâmetros selecionados para apresentação dos mapas de complexidade dos projetos síncrono e *sob demanda*.

Nas Tabelas 4.3 e 4.5, é apresentado como os parâmetros se inter-relacionam. Na Tabela 4.3, tome como exemplo o Centro de operação do servidor. Uma mudança neste parâmetro afeta a metodologia *sob demanda*, o módulo é essencial para esse modo de operação da arquitetura; o Centro de operação cliente, a comunicação entre ambos precisa que eles possuam um protocolo de comunicação e conjunto de ações combinadas; o Gerente de Conexão, pois este recebe os dados que vem do COS-servidor para definir qual barra será habilitada e instantes de manobra do disjuntor; os componentes HCoE, comunicam-se entre si e o HCoE-servidor depende da interação do subsistema A com COS-servidor. O protocolo de operação e a estratégia *sob demanda* também dependem das ações do módulo COS-servidor.

O mapa de complexidade é uma contribuição não somente para o projeto e desenvolvimento de uma arquitetura *sob demanda* ou síncrona, mas também durante a verificação de erros durante a simulação. Considere a situação em que a leitura de um arquivo não foi feita corretamente, parâmetro 4 na Tabela 4.2. Note que o software pode apresentar erros nos outros módulos relacionados a ele, tornando a correção do erro mais difícil. Mas com o mapa de complexidade, esta tarefa se torna mais simples. Tentando ilustrar o quão complexa é uma determinada proposta de projeto, foi desenvolvido um







### Mapa de complexidade do projeto *Sob demanda*

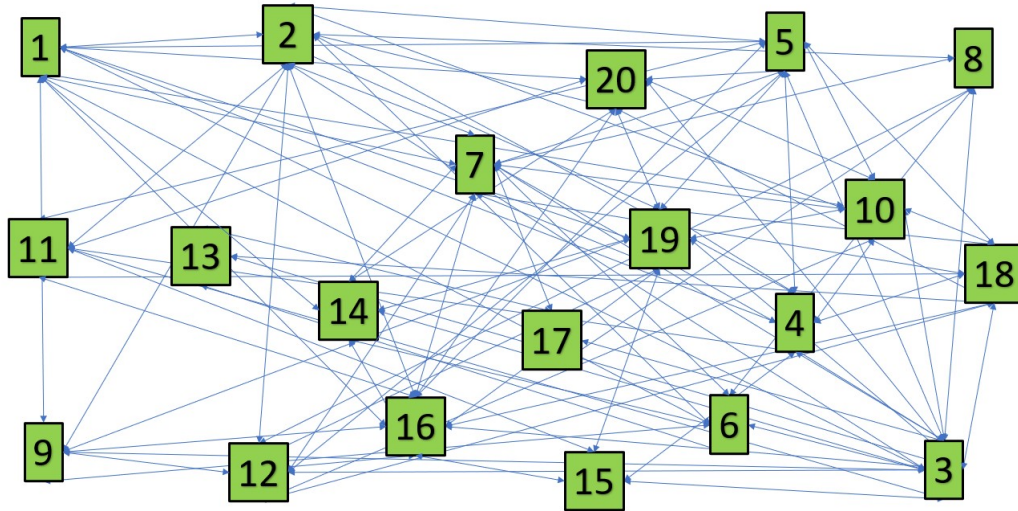


Figura 4.9 – Mapa de complexidade do projeto *sob demanda*.

### Mapa de complexidade do projeto síncrono

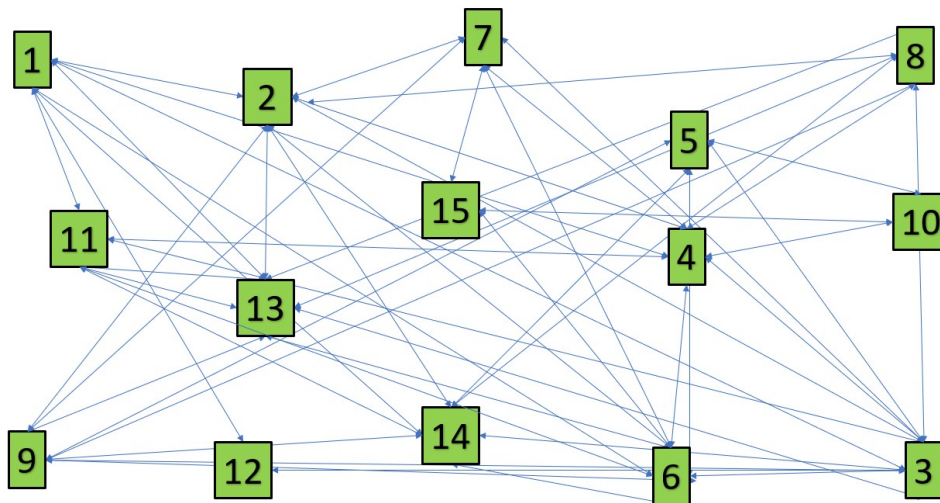


Figura 4.10 – Mapa de complexidade do projeto síncrono.

parâmetro denominado grau de complexidade  $grauComp$ , exposto na equação 4.1 a seguir:

$$grauComp = \sum_{i=1}^{n_{max}} \frac{(n_{Variavel_i})^2}{n_{max} - 1} \quad (4.1)$$

Onde:  $n_{max}$  é o número de parâmetros envolvidos no mapa de complexidade do projeto;  $n_{Variavel_i}$  é a quantidade de outros parâmetros dentro do mapa de complexidade que o parâmetro selecionado " $Variavel_i$ " influencia. Isto significa que quanto maior a interdependência entre as variáveis do projeto, maior será o grau de complexidade dele.

No caso do projeto para a arquitetura *sob demanda*, o parâmetro  $grauComp = 69,53$ ; para a arquitetura síncrona,  $grauComp = 39,21$ .

## 4.5 Considerações finais

Este capítulo apresentou o projeto de uma arquitetura computacional para cálculo de transitórios eletromagnéticos. Esta por sua vez, teve como características principais: permitir interação *sob demanda*, um subsistema A é simulado continuamente enquanto recebe a conexão de outros subsistemas, a qualquer momento da simulação do primeiro; ser distribuída, as simulações propostas para a mesma são formadas por dispositivos dentro de uma rede de computadores, que pode ser local ou geograficamente distribuída; ser colaborativa, o cálculo depende da interação entre os computadores.

Adicionalmente, um projeto de uma arquitetura síncrona foi apresentado para que o leitor pudesse comparar um sistema mais simplificado com aquele que foi proposto neste trabalho. Nesta última, dois computadores realizam o cálculo de transitórios em conjunto, porém a simulação no subsistema principal, servidor, somente ocorre após a conexão do subsistema, cliente. Ou seja, o início e final da execução das simulações ocorre no mesmo intervalo de tempo.

O desenvolvimento do projeto possui um caráter complexo e multidisciplinar, já que envolve diferentes áreas do conhecimento, computação e engenharia elétrica. Buscando facilitar o trabalho daqueles que, num futuro próximo, decidirem se aventurar nesse caminho de pesquisa, foi proposta uma ferramenta educacional, denominada Mapa de Complexidade. Neste Mapa estão expostos um conjunto de parâmetros que são necessários ao desenvolvimento do projeto em questão e como eles se inter-relacionam. Com isto, definiu-se um último parâmetro, o grau de complexidade, cuja proposta é medir a complexidade de desenvolvimento do projeto e mostrá-la matematicamente.

No próximo capítulo, serão apresentados os resultados de simulações realizadas com a arquitetura computacional.

## 5 Testes Computacionais

Este capítulo apresenta um conjunto de testes computacionais cujo principal objetivo é apresentar a arquitetura computacional proposta e suas funcionalidades. Na primeira seção mostra-se um estudo comparativo entre a interface de conexão **HCoE** e modelos de linha Bergeron nos softwares **ATP** e **PSCAD**. Na segunda, um teste computacional é feito para mostrar o modo de operação síncrono. Na terceira, uma simulação é realizada para mostrar o modo de operação *sob demanda* da arquitetura computacional, a principal contribuição deste trabalho. Na quarta, apresenta-se um estudo de caso para analisar os efeitos da latência durante simulações geograficamente distribuídas. E, por último, na quinta seção, um estudo de caso com um grande sistema de potência, envolvendo parques eólicos e usinas solares, é simulado em diferentes cenários para mostrar as oportunidades de trabalho colaborativo provenientes de simulações distribuídas.

### 5.1 Validação do componente HCoE da arquitetura computacional

Este tópico apresenta um conjunto de simulações cujo objetivo é validar o funcionamento do componente **HCoE**. Para tanto, dois circuitos elétricos modelos foram definidos. Os resultados de sinais de tensão foram utilizados para um comparativo com implementações feitas nos softwares **ATP** e **PSCAD/EMTDC**.

Os circuitos escolhidos foram projetados em duas simulações. A primeira utiliza uma fonte *DC* ( $100kV, 1ohm$ ), resistores e linhas de transmissão trifásica, como mostrado nas Figuras 5.1 e 5.2. Na segunda, uma fonte trifásica senoidal ( $230kV, 60Hz$ ), cargas  $0,2MW$  e  $0,6MW$  e linhas de transmissão trifásica formam o sistema elétrico, como mostrado nas Figuras 5.3 e 5.4. Nas figuras são destacados os pontos dos terminais das linhas e os computadores nos quais ocorreram as simulações. Os computadores utilizados estão definidos na Tabela 5.1. O componente foi comparado com modelos de linha Bergeron dos softwares **ATP** e **PSCAD/EMTDC**.

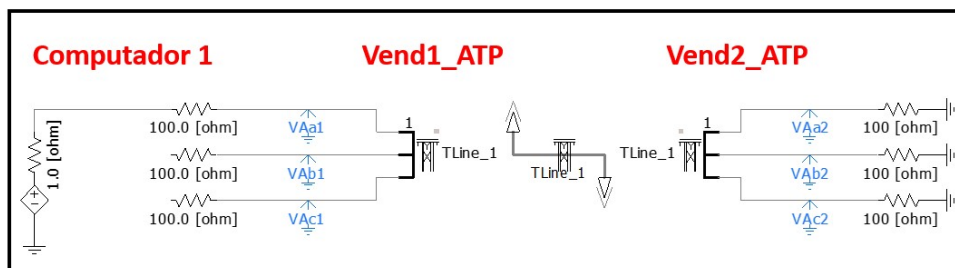


Figura 5.1 – Ilustração de circuito DC implementado no ATP e PSCAD.

Tabela 5.1 – Recurso computacional utilizado para simulações da tese.

Computador 1 (PC1)	
Sistema Operacional	Windows 10, 64 bits
Processador	AMD RIZEN 7 2700 Eight-Core Processor 3,20 GHz
Computador 2 (PC2)	
Sistema Operacional	Windows 10; 64 Bits
Processador	Intel Core i5-3470
Computador 3 (PC3)	
Sistema Operacional	Windows 10, 64 Bits
Processador	Intel core i7 4500U

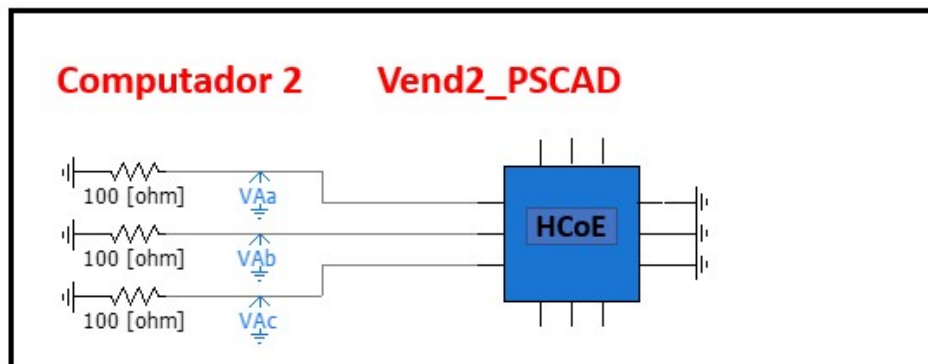
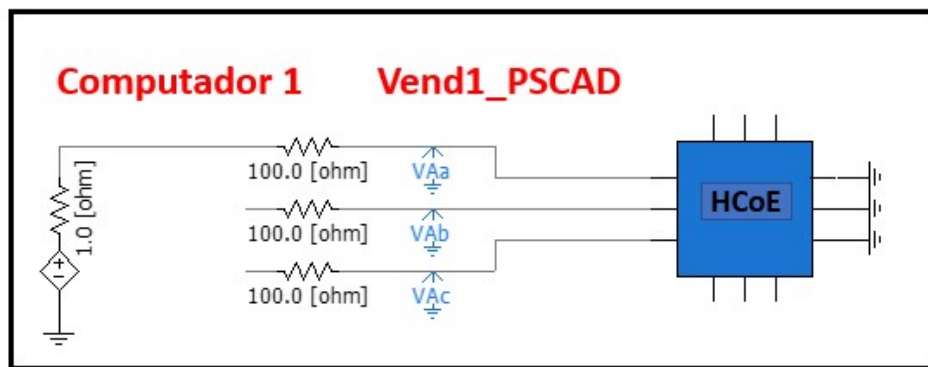


Figura 5.2 – Circuito DC implementado com o componente HCoE.

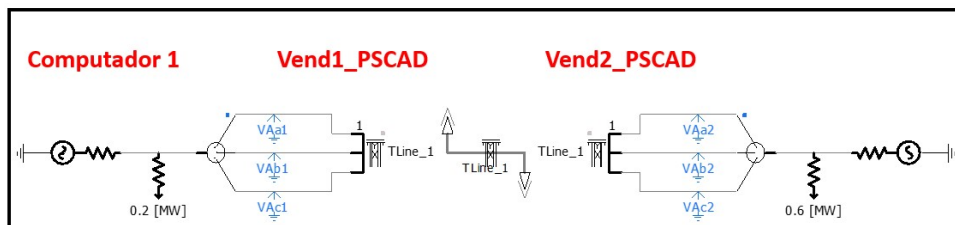


Figura 5.3 – Circuito AC implementado no PSCAD.

Os gráficos 5.5 a 5.7 apresentam um comparativo entre o circuito DC implementado com o modelo de linha Bergeron no ATP e com o componente HCoE.

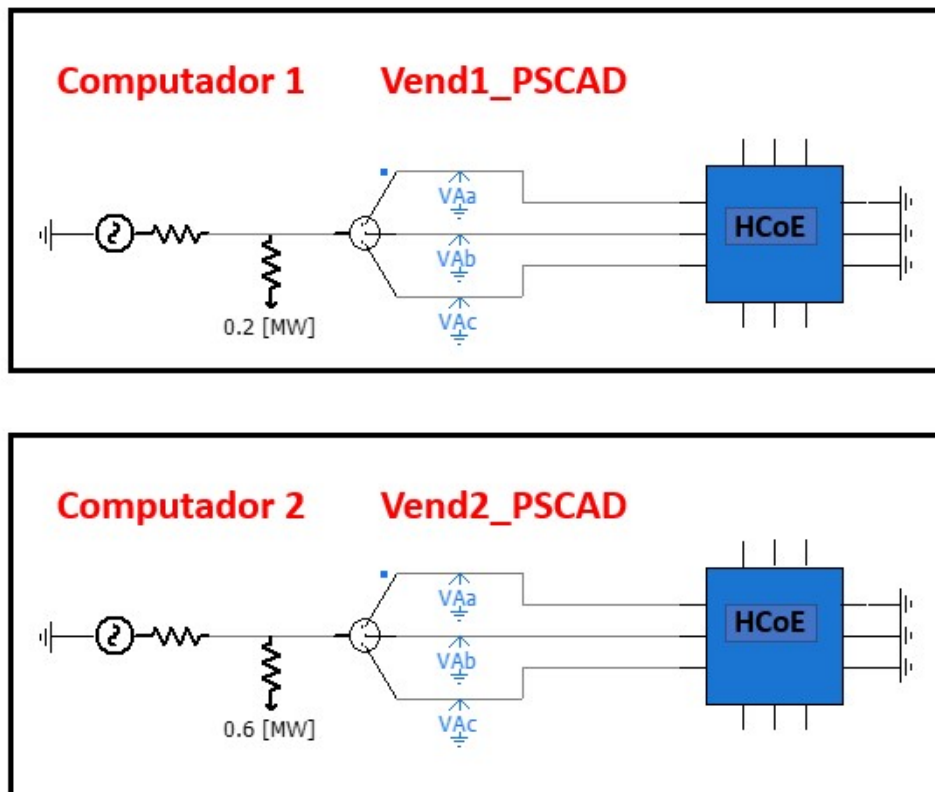


Figura 5.4 – Circuito AC implementado com o componente HCoE.

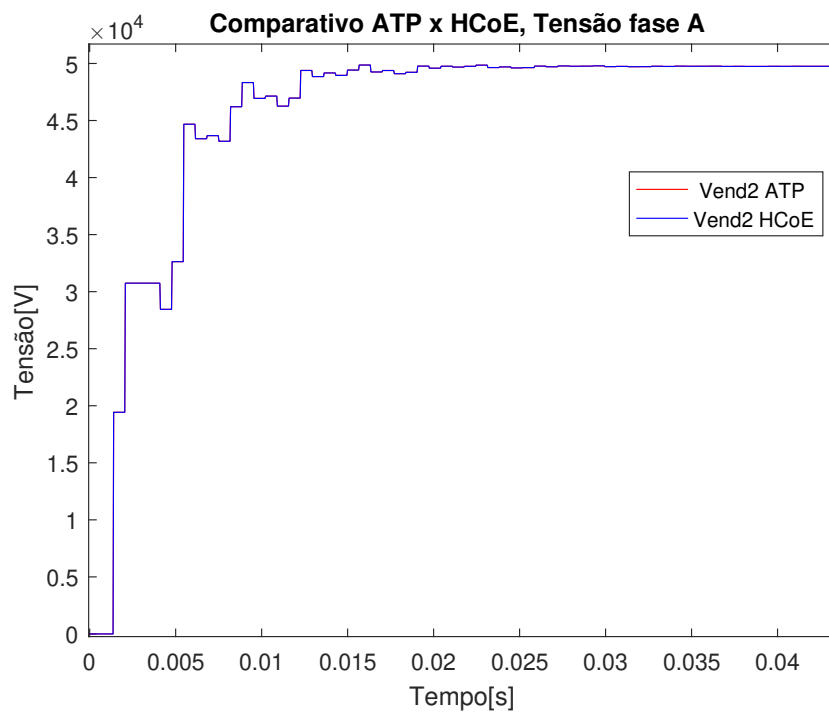


Figura 5.5 – Comparativo ATP x HCoE, fase A.

Os gráficos 5.8 a 5.10 apresentam um comparativo entre o circuito AC implementado com o modelo de linha Bergeron no PSCAD e com o componente HCoE.

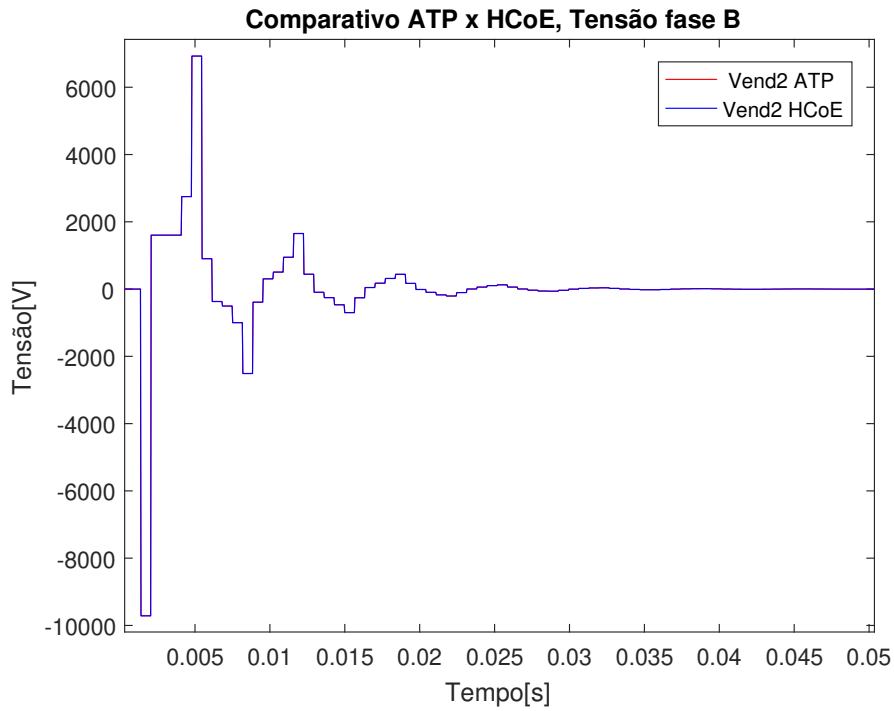


Figura 5.6 – Comparativo ATP x HCoE, fase B.

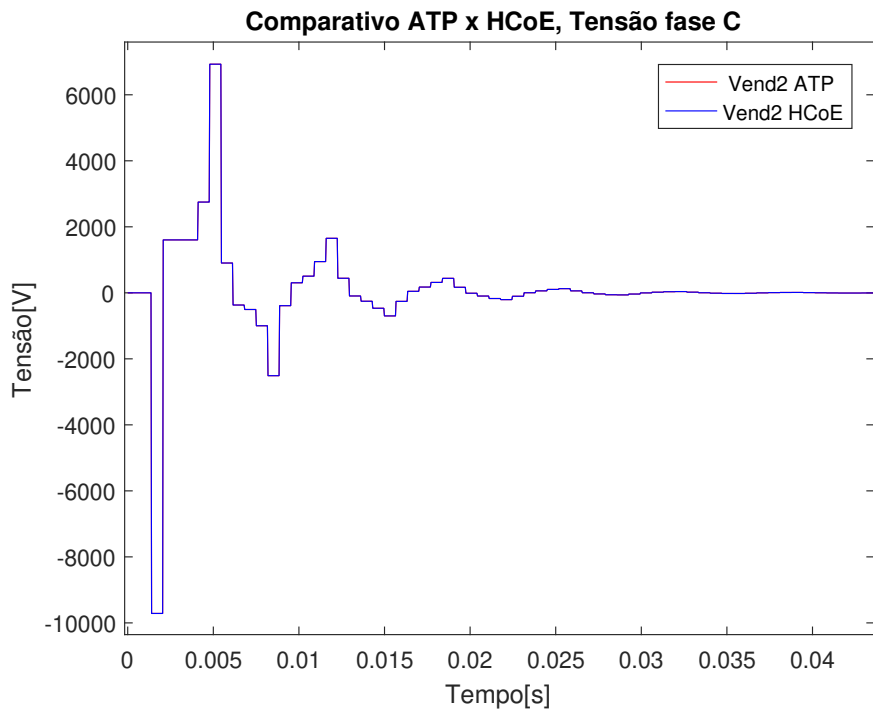


Figura 5.7 – Comparativo ATP x HCoE, fase C.

Os gráficos 5.11 a 5.13 apresentam um comparativo entre o circuito DC implementado com o modelo de linha Bergeron no PSCAD e com o componente HCoE.



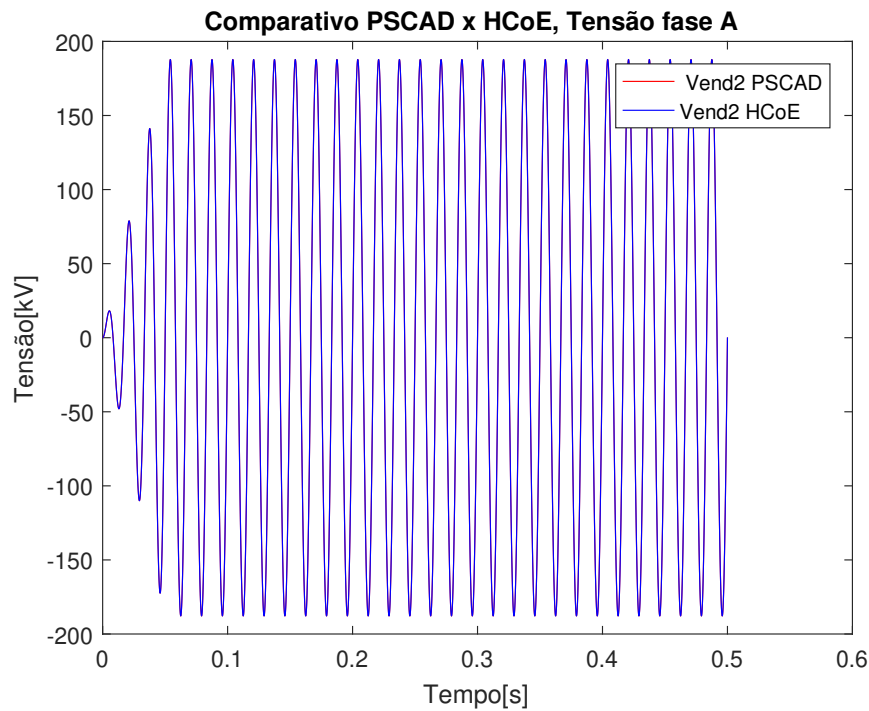


Figura 5.8 – Comparativo PSCAD x HCoE, fase A.

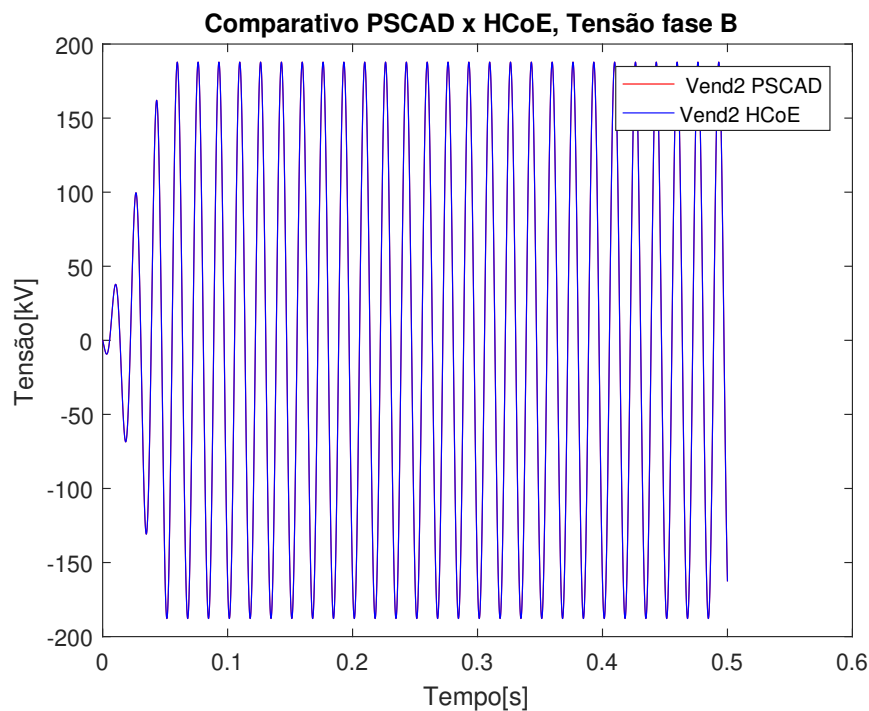


Figura 5.9 – Comparativo PSCAD x HCoE, fase B.

### 5.1.1 Considerações parciais

Este tópico apresentou um comparativo entre o modelo de linha desenvolvido no componente **HCoE** e o modelo de Bergeron, implementado nos softwares **PSCAD** e **ATP**. Os resultados para o componente **HCoE**, com relação aos softwares supracitados, foram

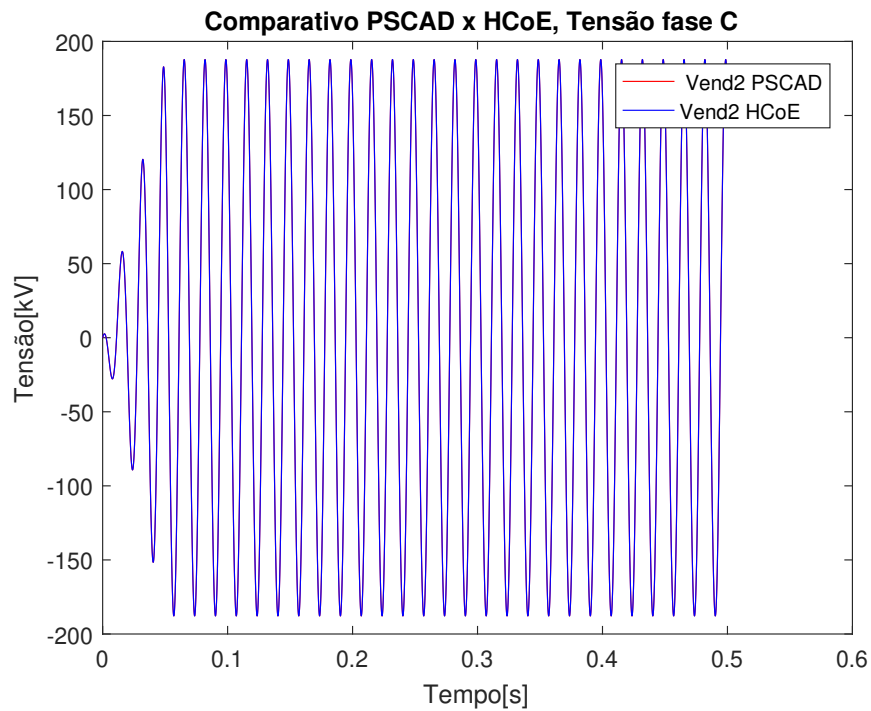


Figura 5.10 – Comparativo PSCAD x HCoE, fase C.

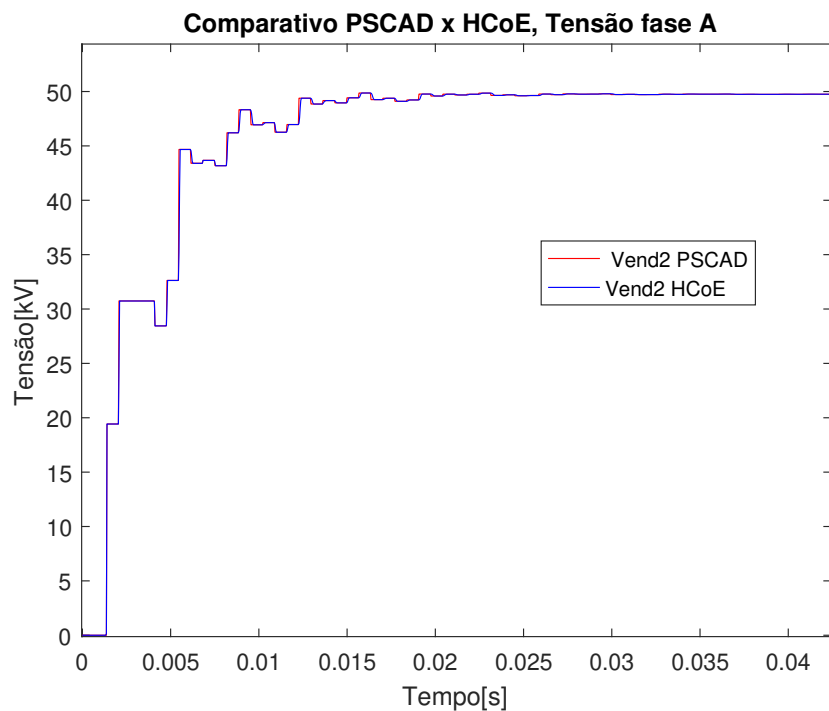


Figura 5.11 – Comparativo DC PSCAD x HCoE, fase A.

satisfatórios; o que valida a utilização deste componente para um cálculo de EMT em paralelo. As simulações deste estudo de caso foram feitas de forma distribuída, mas localmente, no laboratório de Aplicações e Processamento de Informação e Sinais (LAPIS).

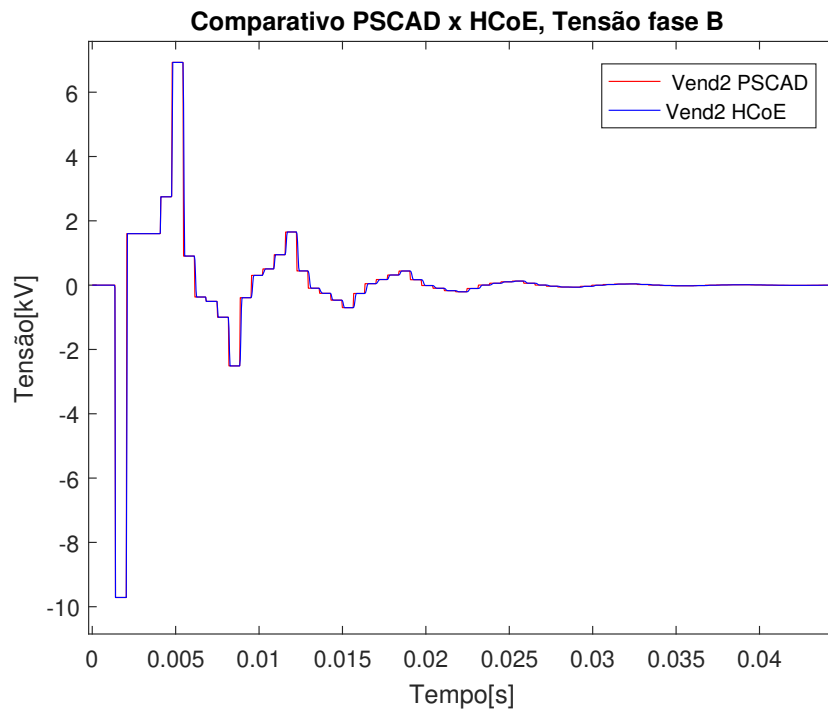


Figura 5.12 – Comparativo DC PSCAD x HCoE, fase B.

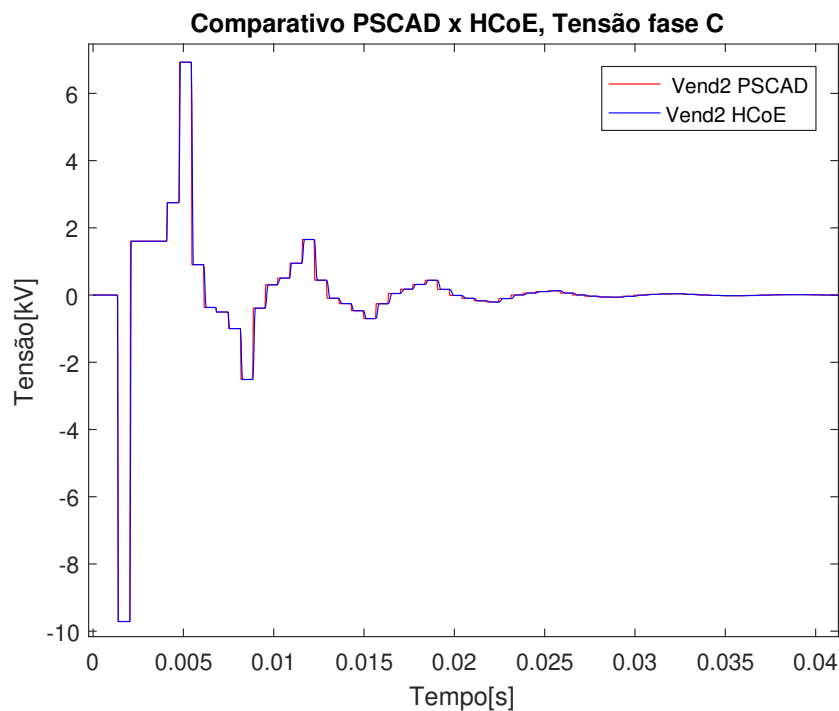


Figura 5.13 – Comparativo DC PSCAD x HCoE, fase C.

## 5.2 Testes computacionais para o modo de operação síncrono

Este caso apresenta uma simulação ilustrando a característica de operação síncrona que poderia ser implementada com uma arquitetura computacional, conforme Figura 4.4 na seção 4.2. O circuito de benchmark *IEEE 39-BUS* é escolhido para construir a rede

elétrica do sistema de teste. Este circuito é duplicado para formar um grande sistema com 78 barras, compreendendo dois subsistemas. O subsistema A, servidor, e o subsistema B, cliente foram conectados conforme a Figura 5.14. Dois disjuntores são conectados ao barramento 16 em ambos os circuitos elétricos. Os terminais do disjuntor são nomeados 1 e 2, visto que o lado 2 é conectado ao componente HCoE para permitir a interação entre os subsistemas A e B, através da arquitetura desenvolvida. Especificamente, os dados para o sistema de energia elétrica são extraídos do benchmark *IEEE 39-BUS* modificado, de acordo com o relatório do centro de pesquisa HVDC de Manitoba [63]. Os geradores são representados como fontes de tensão com 10 ohm impedância, 100MV A potência nominal. O modelo da linha de transmissão é de Bergeron. As cargas são do tipo constante PQ-Load. A linha de transmissão usada para desacoplar o grande sistema no subsistema A e B tem um tempo de viagem de 1,36ms.

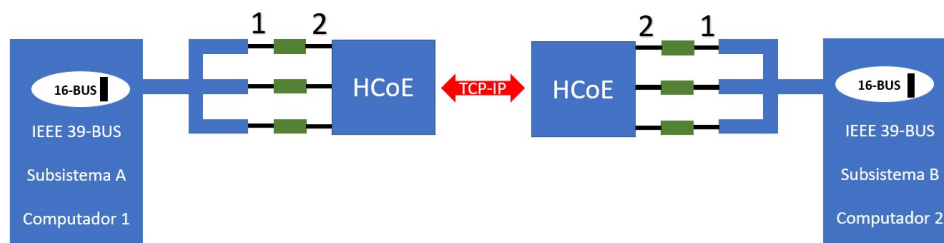


Figura 5.14 – Esquemático do sistema para o modo de operação síncrono.

O teste analisa as tensões nos terminais do disjuntor durante um evento de comutação. A duração da simulação é de 0,7s e o passo de cálculo ( $\Delta t$ ) é, 34,0 $\mu$ s, que dá  $40\Delta t$  para o tempo de viagem, 1,36ms, da linha de transmissão de interconexão. Os dois subsistemas começam do estado estacionário (Utilizou-se aqui a função snapshot do PSCAD). Os instantes de tempo para as operações do disjuntor foram predefinidos para ambos os subsistemas de forma independente. O cenário simulado é o seguinte:

- Inicie a simulação no computador com subsistema A e, depois, no subsistema B;
- o disjuntor no subsistema B é fechado em 0,02s;
- O transitório de tensão chega no terminal 2 do disjuntor no subsistema A em 0,02136s. O valor 0,02132s; marcado na Figura 5.15 é a última etapa para que o valor da tensão seja zero;
- o disjuntor no subsistema A fecha-se em 0,22s;
- o transitório resultante aparece nos terminais do disjuntor no subsistema B em 0,22136s.

Todos os detalhes sobre esses transitórios estão representados nas Figuras 5.15, 5.16 e 5.17.

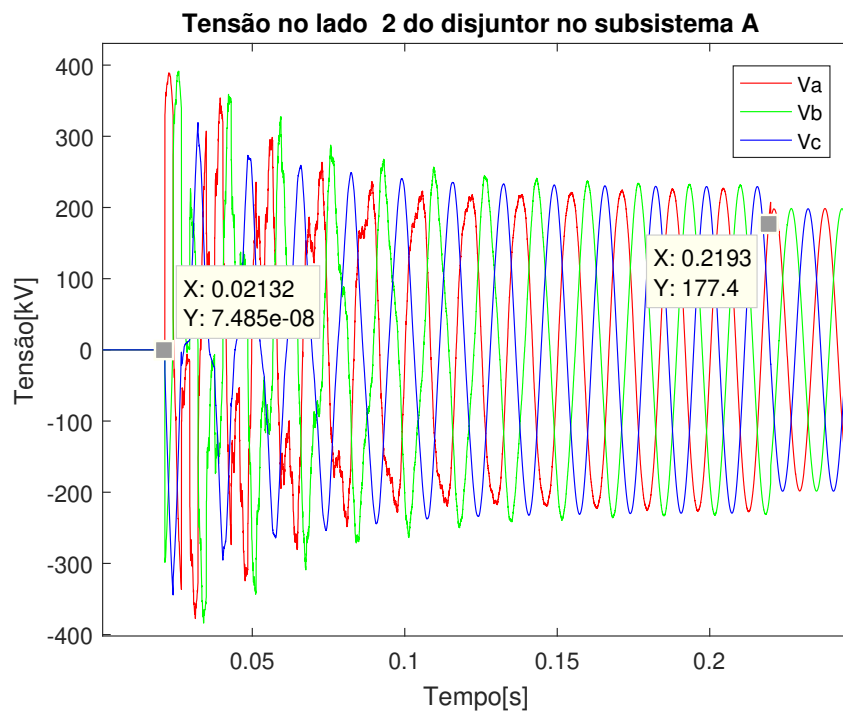


Figura 5.15 – Tensão no lado 2 do disjuntor no subsistema A.

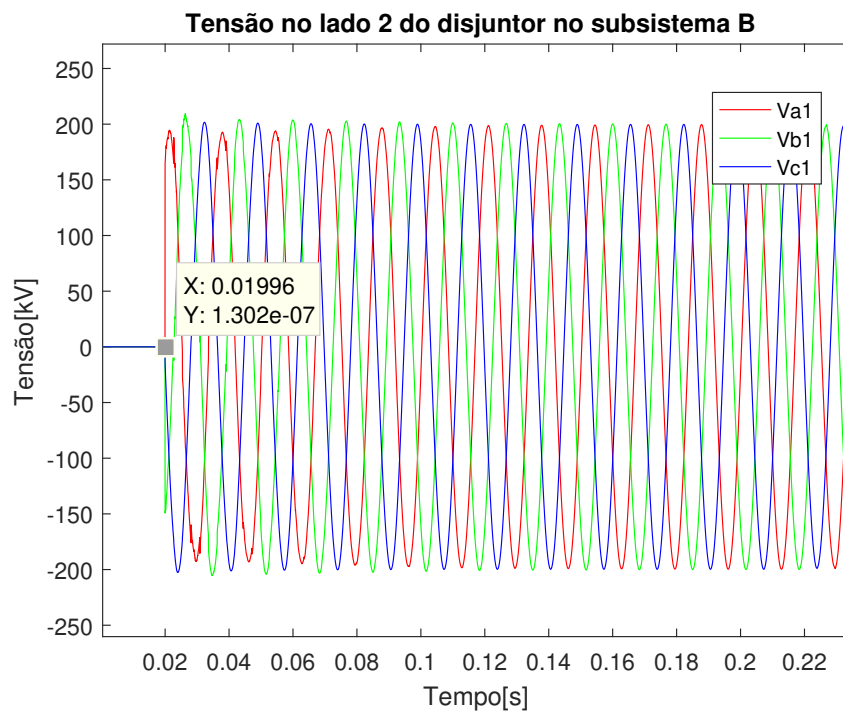


Figura 5.16 – Tensão no lado 2 do disjuntor no subsistema B.

### 5.2.1 Considerações parciais

Este estudo de caso apresentou uma simulação no modo de operação síncrono. Um enfoque foi dado aos eventos de manobra nos disjuntores associados aos componentes HCoE. Neste estudo de caso, o HCoE precisa apenas representar uma linha de transmissão

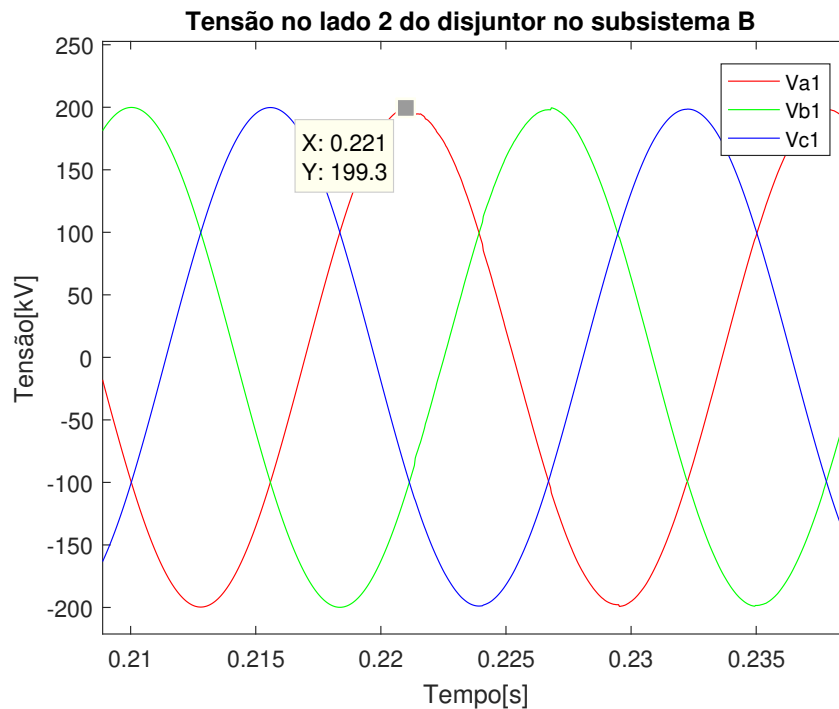


Figura 5.17 – Tensão no lado 2 do disjuntor no subsistema B (destaque).

e ter as funções de comunicação entre cliente e servidor. Os transitórios eletromagnéticos nos terminais dos disjuntores do cliente e servidor confirmam o funcionamento correto do componente [HCoE](#).

### 5.3 Testes computacionais para o modo de operação *sob demanda*

Este caso apresenta o modo de operação *sob demanda* da arquitetura computacional. O circuito elétrico utilizado para as simulações é um “*IEEE 39-Bus benchmark*”. Três computadores são utilizados para a simulação, cujo hardware é apresentado na Tabela 5.1. O computador 1 hospeda o Subsistema A e possui a função de servidor. Os computadores 2 e 3 hospedam os Subsistemas B e C e possuem a função de cliente. Estes últimos estabeleceram conexão com o primeiro computador. A Figura 5.18 mostra o esquemático para o teste:

O teste analisa a tensão nos terminais dos disjuntores após eventos de chaveamento. O passo de cálculo da simulação é  $34\mu s$ . No computador 1, a simulação tem duração de  $5s$ , e nos computadores 2 e 3, a duração é de  $1,3s$ . O cenário de simulação é apresentado nos passos a seguir:

- 1- Inicie a simulação no Subsistema A;
- 2- Inicie a simulação no Subsistema B;

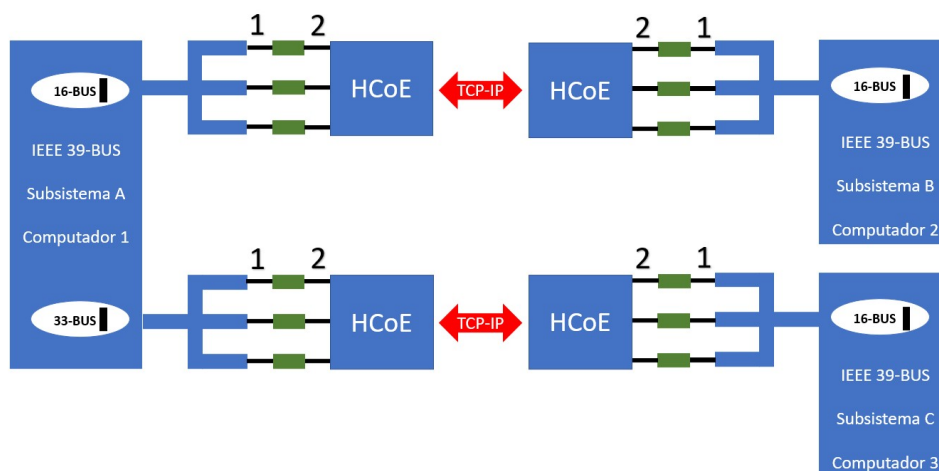


Figura 5.18 – Esquemático do sistema em modo de operação *sob demanda*.

- 3- Uma solicitação de conexão do subsistema B é feita para o Centro de operação do cliente. Este encaminha uma solicitação ao Centro de operação do servidor, o qual determina que o subsistema A pare a simulação e espere a conexão do subsistema B. Neste momento, uma mensagem é enviada ao cliente para que se estabeleça a conexão entre ambos os subsistemas. Em seguida, a simulação continua.
- 4- O disjuntor do Subsistema B, conectado à barra 16, é fechado em 0,32 segundos;
- 5- Depois de um intervalo igual ao um tempo de viagem na linha, o transitório eletromagnético chega ao terminal 2 do disjuntor no Subsistema A;
- 6- O disjuntor do Subsistema A, conectado à barra 16, é fechado em 0,52 segundos após o início da conexão entre os subsistemas.
- 7- O resultado do transitório eletromagnético aparece nos terminais do disjuntor no subsistema B após um tempo de viagem na linha;
- 8- Após um tempo de 1 segundo de conexão, o disjuntor no subsistema A é aberto. Mas os computadores ainda continuam trocando mensagem.
- 9- Após a última troca de mensagens entre os subsistemas, o servidor fecha o canal de comunicação. E continua a simulação com o disjuntor aberto.
- 10- Após 1,3 segundos, a simulação do subsistema B é finalizada.
- 11 – Habilitado pela operação *sob demanda*, o subsistema C faz uma solicitação de conexão ao COS do cliente. Este encaminha uma solicitação ao Centro de operação do servidor, o qual determina que o subsistema A pare a simulação e espere a conexão do subsistema C. Neste momento, uma mensagem é enviada ao cliente para que se estabeleça a conexão entre ambos os subsistemas. Em seguida, a simulação continua.

- 12- O disjuntor do Subsistema C, conectado à barra 16, é fechado em 0,32 segundos;
- 13- Depois de um intervalo igual ao um tempo de viagem na linha, o transitório eletromagnético chega ao terminal 2 do disjuntor no Subsistema A;
- 14- O disjuntor do Subsistema A, conectado à barra 33, é fechado em 0,52 segundos após o início da conexão entre os subsistemas.
- 15- O resultado de transitório aparece nos terminais do disjuntor no subsistema C após um tempo de viagem na linha;
- 16- Após um tempo de 1 segundo de conexão, o disjuntor no subsistema A é aberto. Mas os computadores ainda continuam trocando mensagem.
- 17- Após a última troca de mensagens entre os subsistemas, o servidor fecha o canal de comunicação. E continua a simulação com o disjuntor aberto.
- 18- Após 1,3 segundos, a simulação do subsistema C é finalizada.
- 19- Após 5 segundos de cálculo de transitórios eletromagnéticos, a simulação do subsistema A é finalizada.

A Figura 5.19 apresenta os transitórios eletromagnéticos no lado 2 do disjuntor do subsistema A, na barra 16, após a conexão do subsistema B. Os valores destacados no gráfico mostram os instantes em que o sinal de tensão, proveniente do subsistema B, chega ao disjuntor no subsistema A. Em 0,7005s ocorre a chegada deste sinal ao disjuntor no subsistema A; em 0,9062s ocorre o fechamento do disjuntor no subsistema A; em 1,388s ocorre a abertura do disjuntor no subsistema A. Note que a diferença de 0,2s, entre a chegada do sinal de tensão de B em A e fechamento do disjuntor em A, combina com as ações dos passos 4, 5 e 6 projetados para o cenário simulado. Além disso, a diferença de aproximadamente 0,48 s entre o fechamento e abertura do disjuntor em A estão de acordo com o que foi projetado nos passos 6 e 8 para a simulação.

A Figura 5.20 apresenta os transitórios eletromagnéticos no lado 2 do disjuntor do subsistema A, na barra 33, após a conexão do subsistema C. Os valores destacados no gráfico mostram os instantes em que o sinal de tensão, proveniente do subsistema C, chega ao disjuntor no subsistema A, 3,141s. Em 3,341s ocorre o fechamento do disjuntor no subsistema A. Em 3,826s ocorre a abertura do disjuntor no subsistema A. Note que o gráfico de tensão para as barras 16 e 33 são semelhantes, mas não são iguais, veja a escala de tempo.

As Figuras 5.21 e 5.22 apresentam os sinais de tensão no lado 1 do disjuntor para as barras 16 e 33. Os instantes de fechamento e abertura dos disjuntores, em cada uma das barras, estão destacados nos gráficos. No primeiro intervalo, o subsistema B



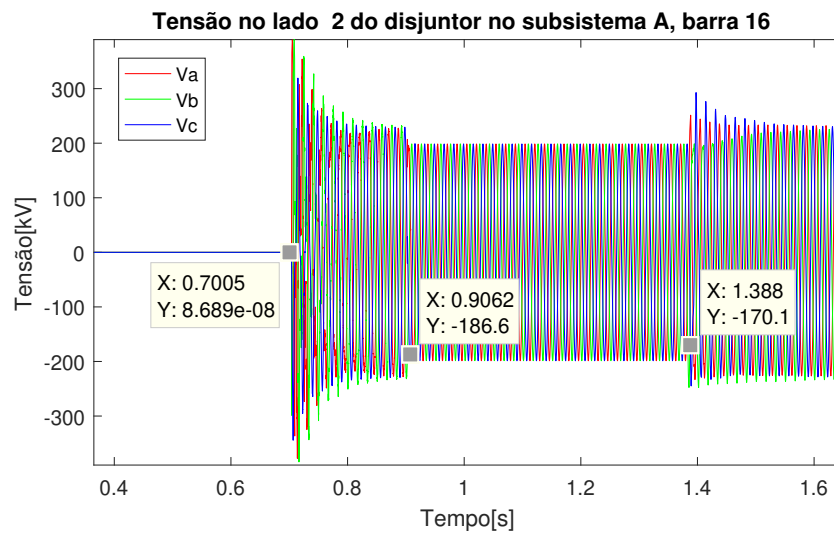


Figura 5.19 – Tensão no lado 2 do disjuntor no subsistema A, barra 16.

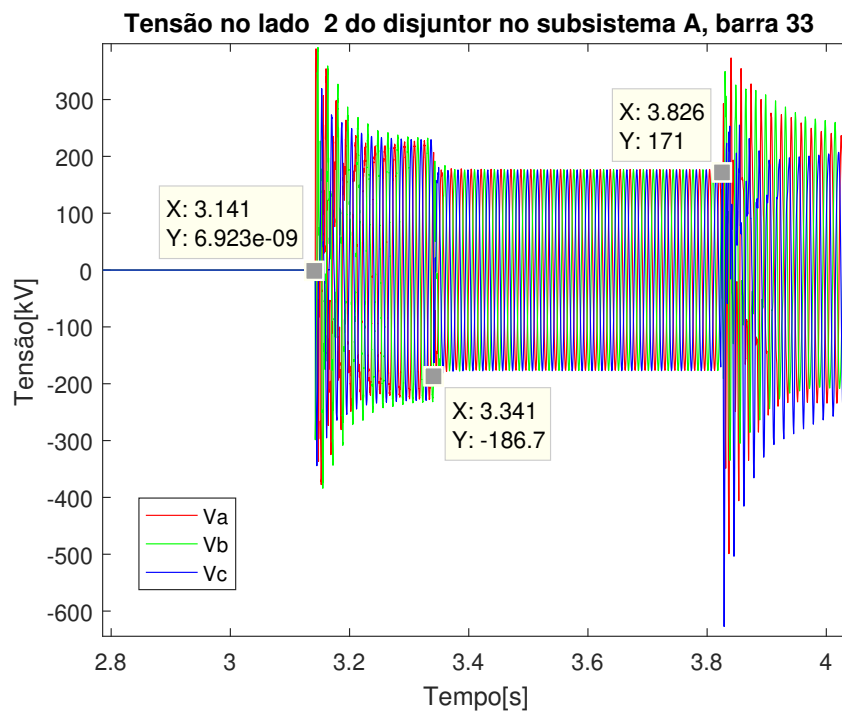


Figura 5.20 – Tensão no lado 2 do disjuntor no subsistema A, barra 33.

estava conectado ao A; no segundo o subsistema C. De acordo com os resultados, pode-se afirmar que a conexão do subsistema C ao subsistema A teve maior impacto nos níveis de tensão que a conexão do subsistema B. Os instantes de tempo destacados evidenciam a característica de interação *sob demanda* proposta neste trabalho; observe a escala de tempo.

A Figura 5.23 apresenta a tensão no lado 2 do disjuntor no subsistema B durante o cenário de simulação. O gráfico destaca os instantes aproximados de fechamento do disjuntor no subsistema B, 0,3177 s; de fechamento e abertura do disjuntor no subsistema

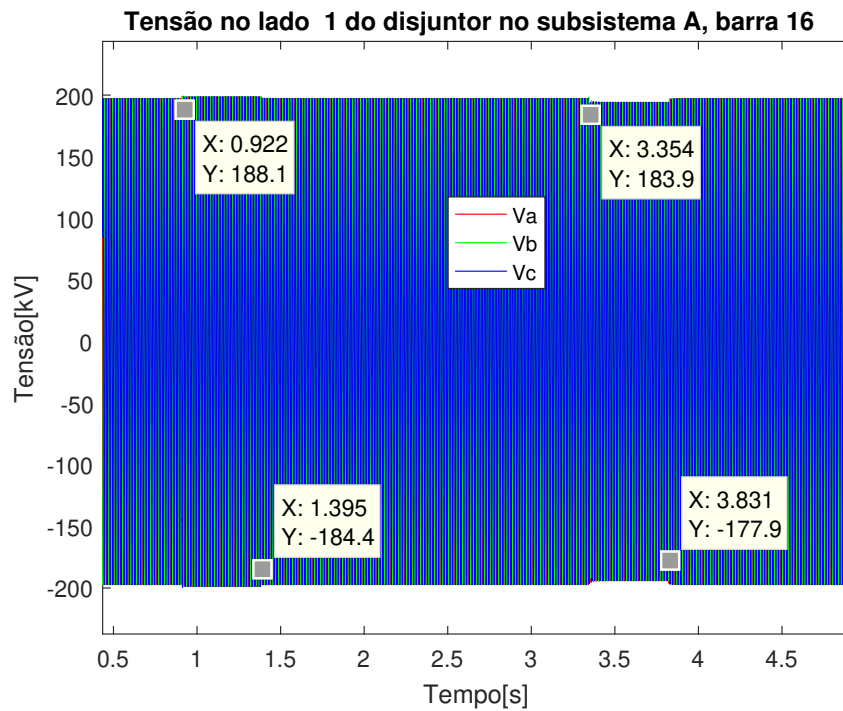


Figura 5.21 – Tensão no lado 1 do disjuntor no subsistema A, barra 16.

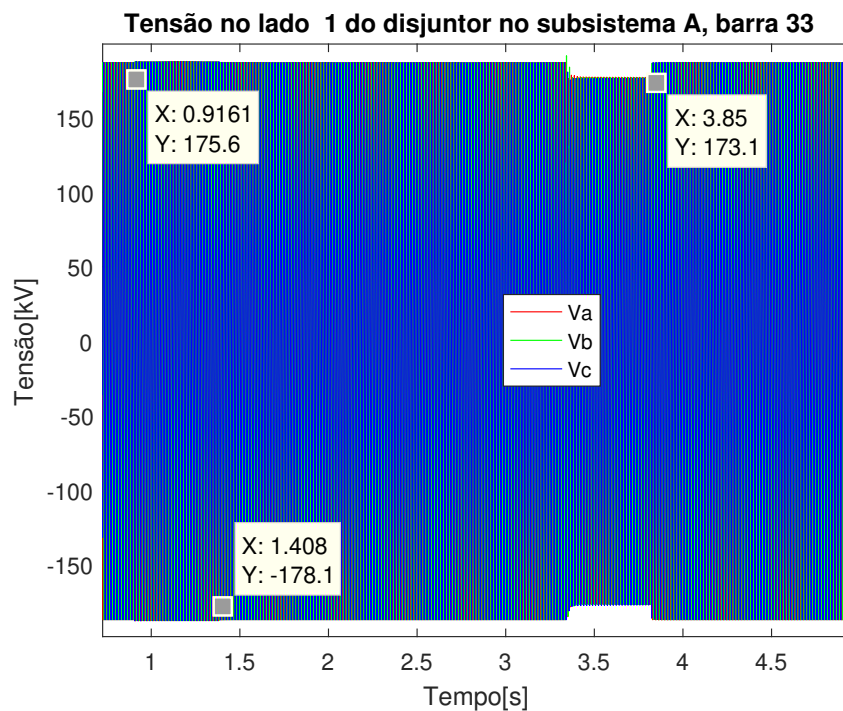


Figura 5.22 – Tensão no lado 1 do disjuntor no subsistema A, barra 33.

A, na barra 16, em 0,5246 s e 1,006 s.

A Figura 5.24 apresenta a tensão no lado 2 do disjuntor no subsistema C durante o cenário de simulação. O gráfico destaca os instantes aproximados de fechamento do disjuntor no subsistema C, 0,3145 s; de fechamento e abertura do disjuntor no subsistema

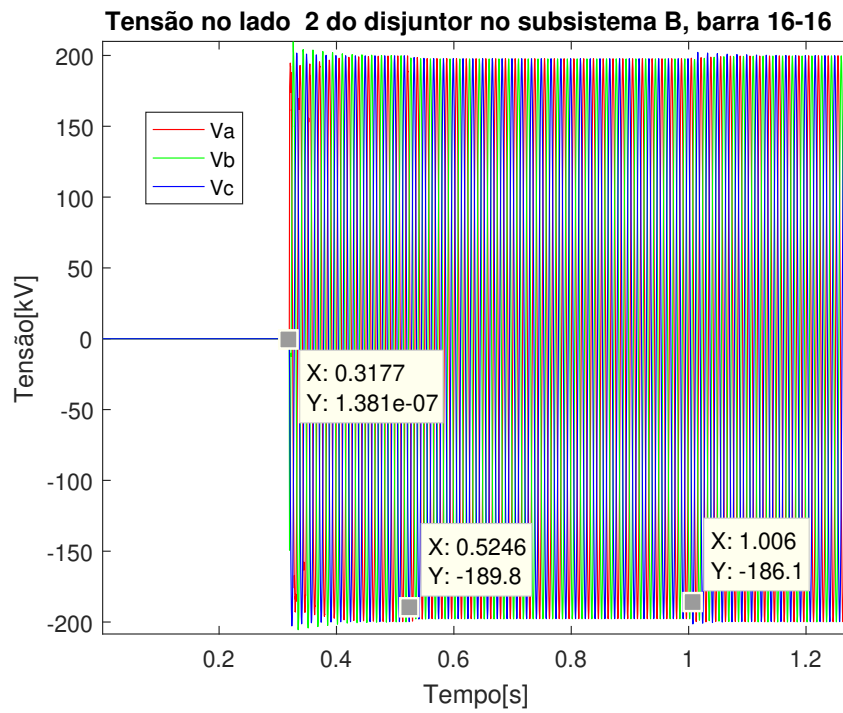


Figura 5.23 – Tensão no 2 do disjuntor no subsistema B.

A, na barra 16, em 0,5237 s e 1,006 s.

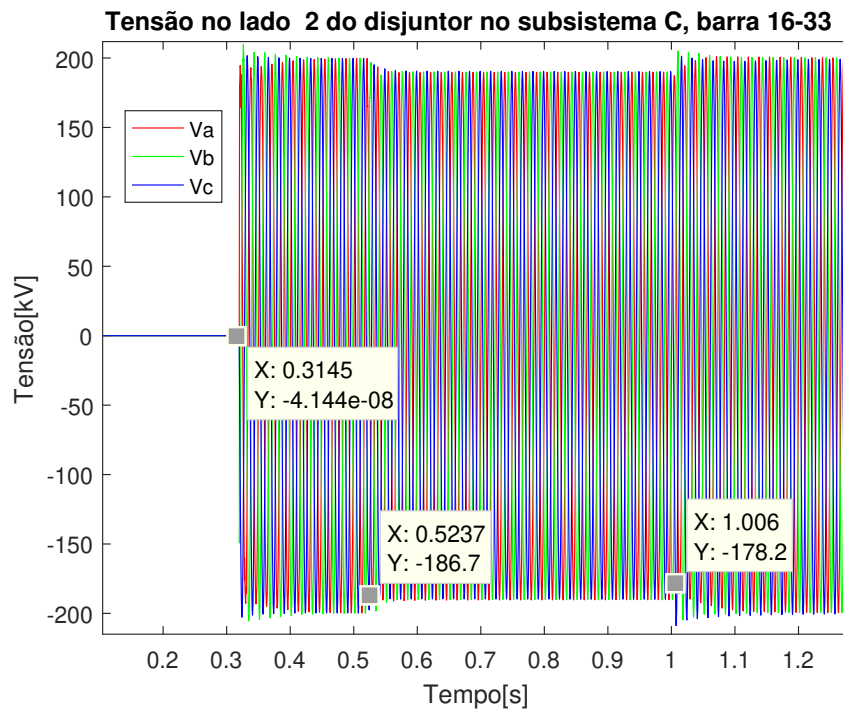


Figura 5.24 – Tensão no lado 2 do disjuntor no subsistema C, barra 33.

### 5.3.1 Considerações parciais

Este tópico apresentou o **modo de operação *sob demanda*** da arquitetura computacional desenvolvida nesta tese. Um conjunto de cenários envolvendo a conexão de dois subsistemas cliente, B e C, a um terceiro subsistema servidor, A, foi desenvolvido para tal intento. Os sinais de tensão nos terminais dos disjuntores foram as variáveis observadas para validar o correto funcionamento da arquitetura computacional.

## 5.4 Análise da latência no cálculo de EMT em simulações com computação distribuída

Este estudo de caso apresenta um conjunto de experimentos para analisar a latência de comunicação durante simulações distribuídas. Os cenários de estudo foram desenvolvidos com dois computadores, distribuídos em diferentes localidades. Cada um hospedou um circuito elétrico projetado como o “*IEEE 39-BUS BENCHMARK*”. Sendo assim, formando um sistema completo de 78 barras durante as simulações colaborativas; da mesma forma como foi feito na parte de resultados 5.2 desta tese. O tempo total escolhido para o estudo foi de 0,7s e o passo de cálculo de 34.0us. Os cenários de simulação são descritos a seguir:

- Simulação 1: O circuito que serve como referência é composto por 78 barras e simulado em um único computador. Para tanto, uma conexão entre dois subsistemas *IEEE 39-bus* é feita através de uma linha de transmissão, modelo Bergeron do PS-CAD. Este caso é simulado no Computador 1, vide Tabela 5.1.
- Simulação 2: O circuito é projetado de forma distribuída, similar ao que foi feito na parte de resultado da seção 5.2 da tese. O subsistema A está localizado no laboratório LAPIS (Laboratório de Aplicações de Processamento e Informação de Sinais) e o subsistema B, no LCE (Laboratório de Compatibilidade Eletromagnética); ambos na Escola de Engenharia da UFMG.
- Simulação 3: O circuito é projetado de forma distribuída, como no resultado da seção 5.2 da tese. O subsistema A está localizado no LAPIS e o subsistema B, no CEFET CAMPUS II (Centro Federal para Educação Tecnológica de Minas Gerais).
- Simulação 4: O circuito é projetado de forma distribuída, como no resultado da seção 5.2 da tese. O subsistema A está localizado no LAPIS e o subsistema B, no LEAT (Laboratório de Extra Alta Tensão).
- Simulação 5: O circuito é projetado de forma distribuída, como no resultado seção 5.2 da tese. O subsistema A está localizado no LAPIS e o subsistema B, na residência de João Paulo Ramos Gomes, no bairro Caiçaras, em Belo Horizonte.

É importante ressaltar que o subsistema A foi projetado no Computador 1 e o subsistema B no Computador 2. A rede de comunicação para os casos simulados 1 a 4 é a rede Ipê - Rede de comunicação das universidades brasileiras [64]. No caso da simulação 5, a simulação envolve uma rede comercial privada e a rede Ypê. As localizações dos laboratórios onde foram feitas as simulações estão destacadas nos mapas das Figuras 5.25 e 5.26.

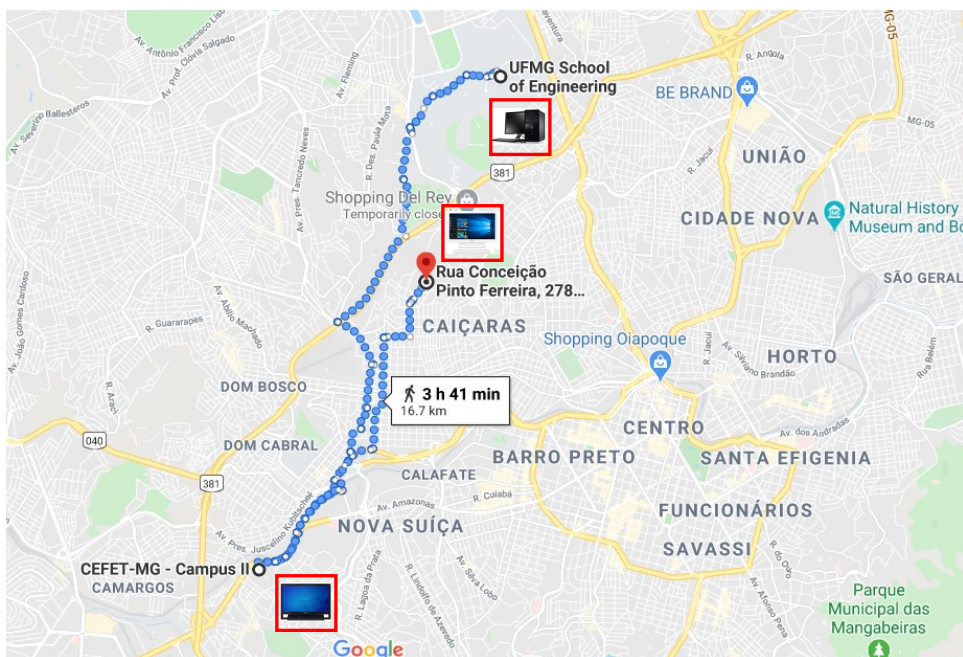


Figura 5.25 – Mapeamento 1 dos locais onde ocorreram as simulações.

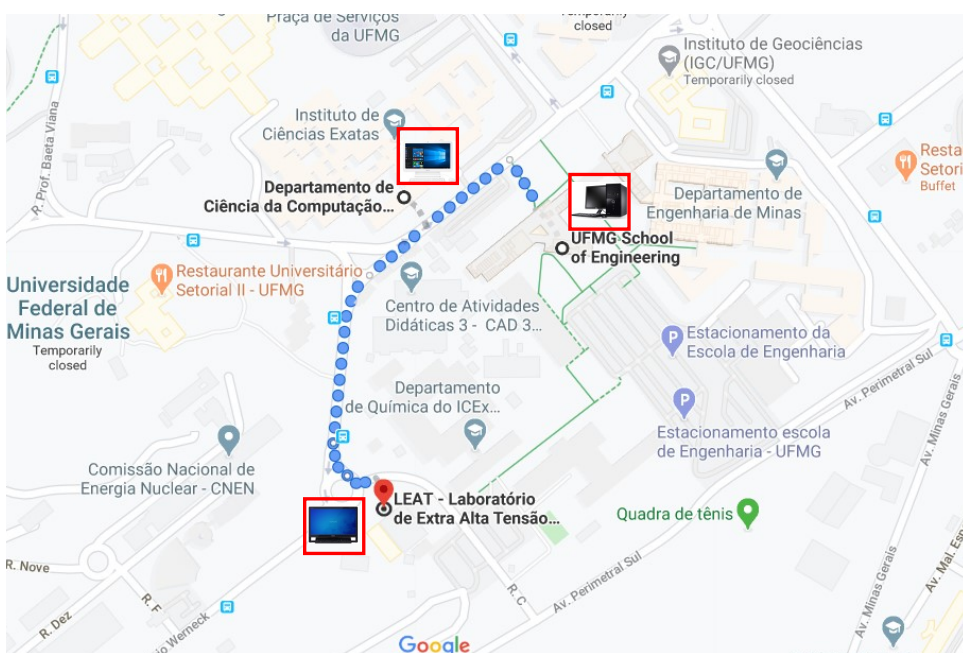


Figura 5.26 – Mapeamento 2 dos locais onde ocorreram as simulações.

A Tabela 5.2 apresenta os resultados com a duração da simulação em cada um dos

Tabela 5.2 – Cenários de simulação distribuída.

Cenários simulados		
Simulação	Tempo total [s]	Tempo de execução [s]
<b>1</b>	0,7	7
<b>2</b>	0,7	116
<b>3</b>	0,7	164
<b>4</b>	0,7	297
<b>5</b>	0,7	2321 = 38 min e 41s

cenários em estudo. Através do comparativo entre o tempo de simulação para o caso de referência e outros casos simulados, é possível ter uma ideia do impacto da latência neste tipo de simulação.

#### 5.4.1 Considerações parciais

Este tópico apresentou um conjunto de simulações cujo cálculo computacional de EMT é feito de forma distribuída e colaborativa. Este estudo de caso foi projetado para subsidiar a análise da latência da comunicação em simulações geograficamente distribuídas; fornecendo ao leitor uma noção do tempo adicionado a simulações executadas de forma distribuída quando comparada a simulações executadas localmente.

## 5.5 Simulação de grandes sistemas de potência utilizando computação distribuída

Este estudo caso objetiva mostrar oportunidades que uma plataforma colaborativa e distribuída proporciona. Para tanto, um Sistema de potência de 3200 barras, 7 usinas solares e 28 parques eólicos ( com geradores e malhas de controle ) foi desenvolvido. Os dados para o Sistema de potência são extraídos dos relatórios [65], [66], [67] , e “*IEEE 118-BUS BENCHMARK*”. O tempo total de simulação é 0,7s e passo de cálculo é 34.0us. A Figura 5.27 apresenta o esquemático do circuito desenvolvido. Um conjunto de três cenários foi desenvolvido para tal intento:

A Tabela 5.3 apresenta os resultados com os tempos totais de execução das simulações:

- Simulação 1: O circuito completo da Figura 5.27 é projetado para ser simulado em apenas um núcleo do Computador 1.

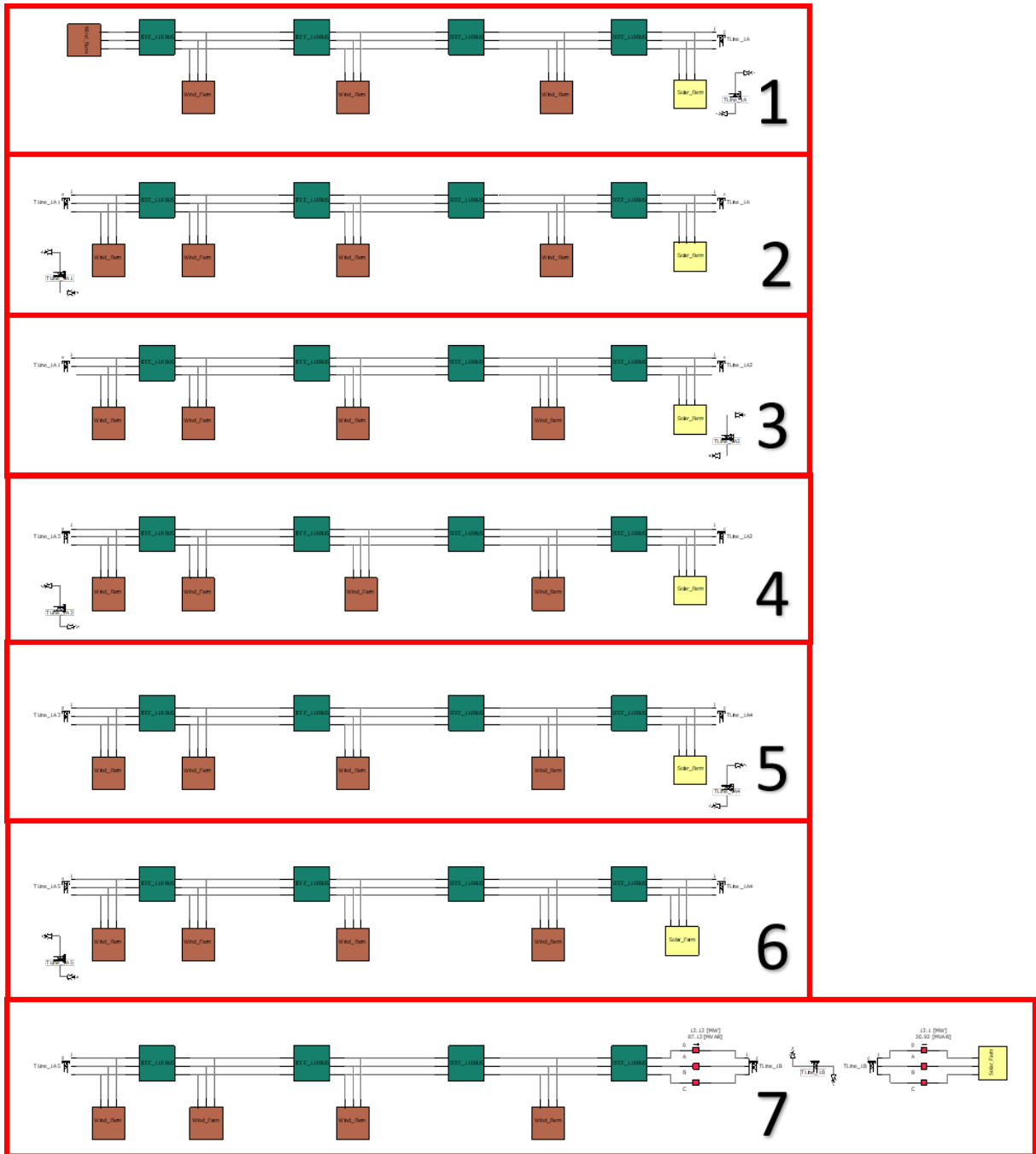


Figura 5.27 – Sistema de potência com 3200 barras, 28 parques eólicas(box marrom) e 7 usinas solares (box amarelo).Linha de transmissão (box verde)

- Simulação 2: O circuito completo é projetado para ser simulado em paralelo, na mesma máquina, utilizando sete núcleos do Computador 1, como destacado na Figura 5.27.
- Simulação 3: O circuito completo é projeto para uma simulação colaborativa e distribuída, de forma síncrona. O Computador 1, localizado no LAPIS, hospeda o circuito com 28 parques eólicos e 6 usinas solares. A simulação, neste computador, é feita em paralelo dividindo o sistema em sete núcleos. A única diferença em relação à simulação 2, é que uma usina é retirada do projeto. Em um Computador 2, localizado no LAPO (Laboratório de Pesquisa Operacional), no Departamento de Ciência da Computação da UFMG, foi simulado apenas uma das usinas solares do projeto completo original. Sobre a localização do LAPO, veja Figura 5.26 na parte de resultados da seção 5.4.

Tabela 5.3 – Resultado das simulações para os cenários de estudo.

Cenários simulados		
Simulação	Tempo total [s]	Tempo de execução [min]
1	0,7	128
2	0,7	37
3	0,7	44

### 5.5.1 Considerações parciais

Este tópico apresentou um estudo de caso envolvendo a simulação de um grande sistema de potência; o qual envolve parque eólicos, usinas solares, cargas dentre outros componentes. Este estudo foi projetado para mostrar as oportunidades que a computação paralela fornece em termos de trabalho colaborativo e redução de tempo de simulação. E, somado a isso, o fato de que em arquiteturas computacionais envolvendo co-simulação, a estrutura dos softwares *off the shelf* pode ser utilizada para melhorar a eficiência da simulação.

## 5.6 Discussão dos resultados

O estudo de caso apresentado na seção 5.1 tem como objetivo validar o funcionamento do componente HCoE que integra a arquitetura computacional proposta nesta tese. Para tanto, circuitos elétricos foram elaborados a fim de comparar simulações realizadas no PSCAD/EMTDC, utilizando o componente HCoE, com outras realizadas no ATP e no próprio PSCAD/EMTDC. Os resultados das curvas de tensão desses testes de validação



apresentaram diferenças com valores máximos na faixa de mV em relação aos softwares ATP e PSCAD. Um valor muito baixo, considerando que a operação de comutação causou sobretensões na faixa de centenas de kV.

Na seção 5.2, uma simulação de uma operação de comutação em um sistema elétrico, que consiste em dois subsistemas IEEE 39 BUS, é realizada. A simulação mostra uma arquitetura computacional síncrona, na qual é possível integrar sistemas elétricos para realizar o cálculo de transitórios eletromagnéticos colaborativamente (cada subsistema tem suas próprias decisões), ao usar um ambiente computacional distribuído, empregando dois computadores diferentes conectados através da Internet. Este cenário de estudo de caso fornece ao leitor a oportunidade de analisar uma simulação síncrona e paralela; o que é muito similar aos trabalhos apresentados em [48],[49], [40] e [45].

A introdução de disjuntores nas terminações da linha foi uma estratégia de simulação muito importante. Através dela, e com a possibilidade intrínseca de funcionamento independente dos disjuntores, tornou-se possível que a conexão entre subsistemas pudesse ocorrer em instantes arbitrariamente selecionados, dependendo dos requisitos próprios de cada subsistema. Isto teve um impacto tanto no modo de operação síncrono quanto no modo de operação *sob demanda* das simulações. Além de garantir mudanças topológicas dos sistemas em tempo de execução. Esta estratégia de simulação não foi vista na bibliografia apresentada.

A possibilidade de realizar cálculos de transitórios eletromagnéticos paralela e colaborativamente deve-se à arquitetura computacional desenvolvida. Isso é semelhante ao trabalho apresentado em [49], embora seja diferente em termos do software "off the shelf" utilizados e também na arquitetura. Comparando esta arquitetura síncrona com as implementadas em [43], [44] e [40] também existem diferenças entre as arquiteturas de hardware e software implementadas.

O estudo de caso sobre uma arquitetura síncrona foi apresentado para que o leitor tivesse um parâmetro para entender as contribuições da arquitetura computacional desenvolvida nesta tese. Todos os trabalhos vistos na revisão bibliográfica ou apresentam uma simulação síncrona, ou não apresentam nenhum comentário a respeito de como é o modo de operação das simulações. E, nenhum deles, apresenta uma proposta permitindo interação *sob demanda*, como a que foi apresentada aqui.

Na seção 5.3 está a principal contribuição da tese, uma *Arquitetura Computacional Distribuída e Colaborativa permitindo Interação Sob demanda* (explicação em 4.1). Um estudo de caso envolvendo três subsistemas, sendo um deles servidor, é apresentado. O modo de operação *sob demanda* desta arquitetura computacional garante que o estado da rede seja mantido após as conexões que o sistema servidor faz com outros subsistemas. Uma das contribuições desse modo de operação, por exemplo, está em situações nas quais se deseja avaliar os efeitos que um transitório eletromagnético, que aconteceu no

subsistema B, provoca em um subsistema C após a conexão de ambos ao subsistema A. Por exemplo, quando centros de operação de concessionárias de energia realizam manobras no sistema de potência para manter a qualidade da energia fornecida.

Na seção 5.3, as Figuras 5.21 e 5.22 ilustram o comportamento *sob demanda* da Arquitetura computacional. O sinal de tensão no lado 1 do disjuntor, nas barras 16 e 33, é apresentado durante as conexões dos subsistemas B e C ao subsistema A. Note que a simulação total no subsistema A ocorre durante cinco segundos e os intervalos em destaque ressaltam os instantes em que os subsistemas estão conectados com os disjuntores fechados.

No caso da Figura 5.21 ocorre em, (0, 922-1, 395) e (3, 354-3, 831); e para a Figura 5.22 em, (0, 9161-1, 408) e (3, 354-3, 850). É neste modo de operação que reside uma das principais diferenças entre esta arquitetura e outras propostas na literatura [50], [51].

A comunicação de dados entre computadores é uma tarefa fundamental no cálculo paralelo de transitórios eletromagnéticos. Se as informações enviadas forem perdidas ou modificadas durante a transmissão de dados, todo o cálculo de transitório será comprometido. No presente trabalho, a escolha do protocolo TCP proporcionou maior confiabilidade na transmissão de dados, uma vez que garantiu que os pacotes enviados não fossem perdidos ou corrompidos, caso contrário, a simulação completa simplesmente não ocorreria. Isto vem com um custo, o tempo de comunicação se torna mais lento quando comparado aos protocolos que não têm esta funcionalidade. O protocolo UDP, por exemplo, não tem confiabilidade como um recurso, embora seja mais rápido que o TCP, os dados enviados usando o UDP podem ser perdidos. No artigo apresentado em [49], o protocolo UDP foi utilizado, de modo que a confiabilidade foi um dos gargalos desta proposta.

No estudo de caso da seção 5.4, dois computadores executam uma simulação colaborativa. No experimento, um laboratório hospeda o computador 1 e outros quatro laboratórios hospedam o computador 2, em diferentes momentos. Uma simulação em um único computador é feita com o sistema completo para servir de referência. Os cenários de simulação 2 e 3 mostram que o impacto da latência nas simulações foram aproximadamente 2 min, na primeira, e 2,5 min, na segunda. No caso do cenário 4, era de se esperar que os efeitos da latência na simulação fossem menores que o caso 3, já que o LEAT se encontra no CAMPUS da UFMG e o laboratório do CEFET, não. O que pode ser explicado pelo meio de propagação física, tecnologia de equipamentos de rede e função de processamento específico, camada de link físico e tecnologia de roteamento de pacotes, como explicitado por Ren Liu et al. [51]. É importante destacar que a rede de comunicação nestes cenários é a rede Ypê; o que garante maior velocidade na comunicação de dados em relação à rede comercial.

Por último, no cenário de simulação 5, uma simulação distribuída é feita com o computador 1, no LAPIS, e um computador 2, em um endereço residencial. Os pacotes de dados trocados entre estes computadores passam por uma rede comercial. Nota-se, por

isso, o impacto no tempo de execução, o qual resultou em 38 minutos e 41 segundos.

A latência na comunicação é um dos principais desafios a serem enfrentados em simulações realizadas de forma distribuída. Neste contexto, a rede de comunicação nacional Ypê [64], que serve a universidades, institutos e centros federais é uma grande oportunidade para minimizar os efeitos desse parâmetro em pesquisas desse tipo. Em [50], uma estrutura de rede de comunicação entre instituições de ensino semelhante também é utilizada.

No artigo apresentado em [44], os autores utilizam as ferramentas de computação de alto desempenho já disponíveis no PSCAD/EMTDC para realizar simulações. É importante notar que as ferramentas disponíveis no PSCAD/EMTDC também continuarão a estar disponíveis na arquitetura computacional desenvolvida. Assim, os sistemas grandes podem agora ser simulados colaborativamente com computação distribuída. Os subsistemas que compõem o sistema maior podem ser simulados localmente com as ferramentas de computação paralela do software, o que pode aumentar o desempenho geral da simulação. Nesta situação, é importante lembrar que o hardware tem de ser compatível com o design de simulação. Para simular um sistema que é dividido localmente em três núcleos, o computador deve ter pelo menos 4 núcleos porque um deles é usado pela interface gráfica do software.

Baseando nessas considerações, o estudo de caso da seção 5.5 foi elaborado para ilustrar as oportunidades que o trabalho colaborativo e distribuído fornece. Três estratégias para a simulação de um grande sistema de potência, 3200 barras, foram utilizadas. A comparação entre as simulações 1 e 2, conforme Tabela 5.3, mostrou que a computação paralela é 3,5 vezes mais eficiente que a sequencial. A simulação 3 é cerca de três vezes mais rápida que a simulação 1, sequencial, e demora apenas 7 minutos a mais que a simulação 2, executada em paralelo no mesmo computador. Vale lembrar que a simulação 3 foi feita de forma distribuída, com um dos computadores localizados no LAPO. Nesta última comparação, entre os cenários 2 e 3, o efeito da latência na comunicação é evidenciado.

As simulações 1, 2 e 3 mostram as oportunidades que surgem ao se utilizar um recurso computacional para melhorar a eficiência das simulações. A partição do sistema completo em sete núcleos fez com que a simulação em paralelo se tornasse mais rápida que no caso sequencial. E, no caso da simulação 3, mostra que a utilização de uma correta estratégia de computação paralela pode minimizar os efeitos da latência no tempo total de simulação. Este resultado fornece um interessante caminho de pesquisa e desenvolvimento para os profissionais envolvidos em estudos de conexão de circuitos elétricos ao sistema interligado nacional (SIN), operado pelo ONS, como exemplificado na seção 1.2.

No Brasil, para que uma empresa estabeleça a conexão de uma unidade geradora de energia elétrica ao SIN, para a qual a tensão de operação se inicia em 230 kV, é necessário que um conjunto de requisitos técnicos seja atendido, conforme definido pelo

**ONS.** Estudos de transitórios eletromagnéticos são pré-requisitos para comprovar que a nova planta está de acordo com o que é solicitado pelo **ONS**, como o impacto nas correntes de curto-circuito trifásico, por exemplo. Geralmente, sem a utilização de novas ferramentas, como a proposta neste trabalho, as simulações neste contexto são realizadas utilizando uma rede equivalente para o **SIN** e uma modelagem detalhada para unidade geradora de energia, subsistema de estudo visado.

A utilização de um equivalente de rede para todo o sistema está relacionada a aspectos como disponibilidade e tempo para preparar os dados da rede, limitações do software quanto ao número de componentes por simulação e tempo de processamento para a simulação completa. Embora esta seja a metodologia de estudo definida pela NSO, continua a ser uma tarefa árdua para os que a ela se dirigem. A entrega de um relatório técnico ao NSO pode levar semanas e sofrer muitas revisões. De acordo com a NSO, o número médio de revisões para o Projeto Básico completo foi alto: três além da edição inicial [68].

Além disso, a Arquitetura Computacional *Sob demanda* permite que um grande sistema de potência esteja disponível em um laboratório funcionando online enquanto outros subsistemas, operados remotamente, possam conectar-se a uma de suas barras, a qualquer momento da simulação, com a garantia de que a propriedade intelectual dos dados do circuitos seja mantida. Isso destaca a importância do modo de operação *sob demanda*, do caráter distribuído do cálculo e a presença do componente gerenciador de conexões. Com este último, contribuindo para uma maior eficiência nas simulações (explicação em 4.1).

Os estudos de caso das seções 5.4 e 5.5 mostraram a influência da latência no tempo de execução da simulação. Quando observamos as simulações 2 e 3, na seção 5.5, cujo tempo total definido pelo usuário é 0,7 segundos, uma latência de 7 minutos pode ser um valor muito grande. Entretanto, quando consideramos o tempo que um pesquisador independente economizaria quando executasse uma simulação colaborativa tal como a simulação 3, assumindo apenas o circuito projetado para o computador no LAPO, a latência na simulação distribuída é muito menor que o tempo para projetar um sistema de 3200 barras como aquele disponível no LAPIS. Sendo assim, as simulações colaborativas se tornam excelentes oportunidades de compartilhar recursos computacionais, dados de sistemas e conhecimentos adquiridos em modelagens de circuitos elétricos.

Neste trabalho, foi utilizado o mesmo passo de cálculo para simulação de subsistemas. Se diferentes passos de cálculo forem usados, técnicas de interpolação devem ser empregadas, como as propostas em [69] e [70]. Não prevemos nenhum inconveniente ou dificuldade para a implementação de técnicas de interpolação em nossa metodologia. Outra questão é sobre a partição do sistema ou partição ideal do sistema. Este trabalho utiliza nós elétricos com modelo de linha de transmissão para dividir em subsistemas

conforme previamente acordado entre servidor e cliente. No entanto, outras interações podem ser usadas, conforme descrito em [48]. Além disso, como o foco do trabalho é demonstrar que as interações sob demanda são viáveis, o que foi alcançado por meio de um modelo servidor-cliente, a interação direta entre clientes não foi implementada. De qualquer forma, mediante hierarquização, a proposta pode ser estendida para permitir interação generalizada entre clientes diretamente.

No próximo tópico serão apresentadas as conclusões finais desta tese de doutorado.

## 6 Conclusão e Propostas de Continuidade

Esta tese de doutorado apresentou o projeto de uma arquitetura computacional para cálculo de transitórios eletromagnéticos. Esta, no que lhe concerne, teve como características principais: ser *sob demanda*, um subsistema A é simulado continuamente enquanto recebe a conexão de outros subsistemas, a qualquer momento da simulação do primeiro; ser distribuída, as simulações propostas para ela são formadas por dispositivos dentro de uma rede de computadores, que pode ser local ou geograficamente distribuída; ser colaborativa, o cálculo depende da interação entre os computadores. Os testes computacionais realizados mostraram êxito no desenvolvimento desta proposta.

Devido a seu caráter multidisciplinar, o desenvolvimento desse projeto veio acompanhado de uma série de desafios, os quais estiveram relacionados à junção de duas áreas do conhecimento, computação paralela e cálculo de transitórios eletromagnéticos. Como forma de contribuir com os pesquisadores que seguirem este caminho, uma ferramenta pedagógica denominada Mapa de Complexidade, juntamente com um parâmetro para quantificar a complexidade do projeto, foram propostos, neste contexto.

Um conjunto de simulações foi realizado para analisar o impacto da latência no cálculo computacional distribuído, o que ficou evidente com os resultados apresentados. Computadores foram distribuídos em laboratórios do CAMPUS da UFMG, CEFET e um endereço residencial, todos na cidade de Belo Horizonte. Os experimentos mostraram que a latência na comunicação foi menor para os casos que utilizaram a rede Ypê em relação à rede comercial.

O cálculo [EMT](#) para um grande sistema, composto por 3200 barras, 7 usinas solares e 28 parques eólicos, é utilizado para demonstrar a viabilidade da arquitetura computacional proposta. Uma comparação é estabelecida entre três casos: um caso sequencial, um caso paralelo, ambos rodando no mesmo computador; e um caso distribuído utilizando a arquitetura computacional proposta. O tempo gasto para as simulações foi de 128, 37 e 44 minutos, respectivamente. Isso mostra que a latência pode não ser um problema para a arquitetura proposta. Adicionalmente, esses resultados ilustram as oportunidades que podem surgir quando pesquisadores trabalham, livremente, com trabalho distribuído e colaborativo. Além disso, ilustrou-se a oportunidade que o trabalho colaborativo traz em relação ao compartilhamento de recursos computacionais e garantia de propriedade intelectual e confidencialidade de dados.

## 6.1 Propostas de continuidade

### 6.1.1 Outros softwares off the shelf para a mesma arquitetura computacional

A Arquitetura computacional apresentada faz cálculo de transitórios eletromagnéticos de forma paralela e distribuída tendo como base o software “*off-the-shelf*” [PS-CAD/EMTDC](#). Existem outros softwares comerciais que poderiam ser utilizados para tal intento, tais como o [OPENDSS](#), [ATP](#), [OPAL-RT](#), [MATLAB](#), [Power Factory](#), [RTDS](#), [EMTDC-RV](#). Para tanto, uma investigação sobre as linguagens de programação adotadas pelo software, a metodologia para integração de componentes de usuário à simulação e definição do modelo de interface seriam necessárias.

### 6.1.2 Arquitetura computacional voltada para simuladores híbridos com objetivos variados

Neste trabalho, o foco do projeto foi a simulação de transitórios eletromagnéticos com simuladores homogêneos em computadores diferentes. No entanto, o projeto da interface de comunicação pode ser ampliado, de tal forma que a estrutura dessa plataforma seja estendida para ambientes de co-simulação híbridos. Nestes, softwares com objetivos diferentes podem ser integrados. Por exemplo, análises de efeitos térmicos e elétricos em sistemas de potência poderiam ser utilizados. Ou, ainda, a modelagem para transitórios eletromagnéticos associada à telecomunicações, como é feito para estudar o conceito de redes inteligentes de energia.

### 6.1.3 Desenvolvimento de interfaces para simulações paralelas e distribuídas

O componente [HCoE](#) foi implementado com a premissa de que ambos os subsistemas funcionam com o mesmo passo de cálculo. Outra proposta seria desenvolver este componente considerando o acoplamento entre subsistemas com dinâmicas rápidas e lentas, por meio de técnicas de dizimação e interpolação. Por exemplo, a integração entre um sistema modelado para cálculo de transitórios e outro para estabilidade transitória. Ou, ainda, subsistemas com diferentes passos de cálculo na modelagem [EMT](#).

O projeto do componente [HCoE](#) teve como base a modelagem de linhas de transmissão baseadas em ondas viajantes. Analisar outras propostas para desacoplar os sistemas elétricos de forma a promover a computação distribuída seria interessante, principalmente para situações onde se deseja separar um sistema onde não haja linha de transmissão.

#### 6.1.4 Avaliação do impacto econômico de uma aplicação utilizando computação paralela e distribuída

A mudança na metodologia de análise da viabilidade de projetos e empreendimentos para o setor elétrico é algo complexo, envolve questões técnicas, econômicas e culturais. Uma avaliação do impacto econômico da aplicação de arquiteturas computacionais distribuídas em projetos de integração de novos empreendimentos ao SIN seria de grande importância para o setor elétrico. Este tipo de estudo forneceria à ANEEL argumentos para implementar uma nova metodologia para os interessados em fazer integração ao sistema elétrico brasileiro. Um estudo de caso poderia ser desenvolvido com a parceria entre Universidade Federal de Minas Gerais, o Operador Nacional do Sistema Elétrico e uma empresa, a multinacional francesa ENGIE, por exemplo. Esta última faz estudos de integração de empreendimentos elétricos ao SIN.

#### 6.1.5 Simulação geograficamente distribuída envolvendo *Hardware in the Loop*

As simulações geograficamente distribuídas de cálculo de transitórios eletromagnéticos são uma realidade. Nesse contexto, o avanço para simulações de tempo real envolvendo cálculo de EMT com Hardware in the Loop é uma linha de pesquisa que vem ganhando muitos adeptos.

#### 6.1.6 Novo projeto de arquitetura computacional com mais funcionalidades e integração de sistemas homogêneos ou heterogêneos

A arquitetura computacional desenvolvida possui um protocolo específico de funcionamento. Nesta proposta o servidor deve ser ligado primeiro e os sistemas elétricos clientes, depois. Isso está relacionado ao modelo de comunicação cliente-servidor. Uma nova arquitetura computacional com um sistema de gerenciamento central para as conexões pode ser uma eficiente opção. Neste dispositivo, haveria a possibilidade de virtualização dos sistemas a serem interconectados. Nesta arquitetura, os sistemas elétricos a serem interconectados poderiam ser de plataformas homogêneas ou heterogêneas. Como consequência disso, modelos de comunicação poderiam ser experimentados. Adicionalmente, uma biblioteca com diferentes dispositivos de interface entre subsistemas, com novos protocolos de conexão, poderiam ser testados. Aqui o padrão FMU-FMI poderia ser adotado. A nova arquitetura se configuraria em uma plataforma online para simulação distribuída.

O *know how* apresentado nesta tese, associado a este tipo de arquitetura, fornece as bases para que se possa estudar um novo conceito denominado "*Virtual Power Plant*". Esta definição pode ser entendida com um sistema de controle central que faz o gerenciamento de independentes geradores de energia renovável, os quais, quando somados, formam um gerador de energia maior.



### 6.1.7 Desenvolvimento de um centro de pesquisa voltado para sistemas computacionais colaborativos e multidisciplinares.

A computação paralela e distribuída vai ganhar um novo impulso com a pandemia do COVID-19. Associado a isso, estão tecnologias que fazem essa técnica mais eficiente, como exemplo: hardware com multiprocessadores com preços mais acessíveis, técnicas de inteligência artificial e aprendizagem de máquina, aumento da velocidade da internet por tecnologia de fibra ótica, tecnologia 5G. Investigar estruturas laboratoriais, não necessariamente físicas, para arquiteturas computacionais distribuídas é um caminho promissor.

### 6.1.8 Aplicação de conceitos desta tese em outras áreas

E se essa arquitetura computacional proposta nesta tese fosse utilizada em problemas paralelizáveis de outras áreas do conhecimento?

# Referências

- 1 MATHWORKS. *MATLAB Distributed Computing Server User's Guide*. 2019. <https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/products/cloud.pdf>. Online; accessed 28-August-2019. 9, 67
- 2 MATHWORKS. *MATLAB Parallel Computing Toolbox User's Guide*. 2019. <https://www.pdfdrive.com/parallel-computing-toolbox-users-guide-mathworks-e5805204.html>. Online; accessed 28-August-2019. 9, 66, 67, 68
- 3 INTERNATIONAL, M. H. *PSCAD/EMTDC - version 4.6. EMTDC User's Guide*. 2019. <https://hvdc.ca/knowledge-base/read,article/163/emtdc-user-s-guide-v4-6/v>. Online; accessed 03-August-2019. 9, 36, 68, 69, 70
- 4 INTERNATIONAL, M. H. *Parallel and High Performance Computing*. 2019. <https://hvdc.ca/knowledge-base/topic:299/v>. Online; accessed 03-August-2019. 9, 70, 71
- 5 Rasheed, A.; San, O.; Kvamsdal, T. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, v. 8, p. 21980–22012, 2020. 17
- 6 DOMMEL, H. Digital computer solution of electromagnetic transients in single-and multiphase networks. *IEEE Transactions on Power Apparatus and Systems*, Institute of Electrical and Electronics Engineers (IEEE), PAS-88, n. 4, p. 388–399, apr 1969. 17, 23, 24, 25, 33, 36, 52, 57, 59, 72
- 7 COMMISSION, E. *European Smart Grids Technology Platform: Vision and Strategy for Europ's Eletricity Networks of the Future*. Brussels, Belgium:: EUROPEAN COMMISSION, 2006. ISBN 92-79-01414-5. 18
- 8 FANG, X. et al. Smart grid — the new and improved power grid: A survey. *IEEE Communications Surveys & Tutorials*, Institute of Electrical and Electronics Engineers (IEEE), v. 14, n. 4, p. 944–980, 2012. 18
- 9 METS, K.; OJEA, J. A.; DEVELDER, C. Combining power and communication network simulation for cost-effective smart grid analysis. *IEEE Communications Surveys & Tutorials*, Institute of Electrical and Electronics Engineers (IEEE), v. 16, n. 3, p. 1771–1796, 2014. 18
- 10 M.GUIZANI; ANAN, M. Smart grid opportunities and challenges of inte grating renewable sources: A survey. In: *Wireless Communications and Mobile Computing Conference*. [S.l.: s.n.], 2014. 18
- 11 MARTINEZ-VELASCO, J. A. Introduction to electromagnetic transient analysis of power systems. In: *Transient Analysis of Power Systems*. [S.l.]: John Wiley & Sons, Ltd, 2014. p. 1–8. 19, 54
- 12 WATSON, N.; J.ARRILAGA. *Power Systems Electromagnetic Transients Simulation*. [S.l.]: London: The Institution of Engineering and Technology, 2007. ISBN 978-0-85296-106-3. 19, 26, 33, 36, 54, 73

- 13 JR. C. E. M. PEREIRA, A. A. C. A. L. C. Z. *Equivalentes de Redes Eléctricas para Cálculo de Transitórios Eletromagnéticos no Programa ATP*. 2020. <https://docplayer.com.br/10308919-Equivalentes-de-redes-eletricas-para-calculo-de-transitorios-eletromagneticos-no-programa-atp.html>. Online; accessed 15-July-2020. 19
- 14 BARTHOLD, L. O.; CARTER, G. K. Digital traveling-wave solutions i-single-phase equivalents. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, v. 80, n. 3, p. 812–818, 1961. 23
- 15 URAM, R.; MILLER, R. W. Mathematical analysis and solution of transmission-line transients i-theory. *IEEE Transactions on Power Apparatus and Systems*, v. 83, n. 11, p. 1116–1123, 1964. 23
- 16 ARAÚJO, A. E.; NEVES, W. *Cálculo de transitórios eletromagnéticos em sistemas de energia*. [S.l.]: UFMG, 2005. ISBN 85-7041-448-x. 26, 33
- 17 JÚNIOR, L. C. Z. *Transitórios eletromagnéticos em sistemas de potência*. [S.l.]: Editora da Universidade de São Paulo, 2003. 26
- 18 MARTI, L. Simulation of transients in underground cables with frequency-dependent modal transformation matrices. *IEEE Transactions on Power Delivery*, Institute of Electrical and Electronics Engineers (IEEE), v. 3, n. 3, p. 1099–1110, jul 1988. 28, 30
- 19 MORCHED, A.; GUSTAVSEN, B.; TARTIBI, M. A universal model for accurate calculation of electromagnetic transients on overhead lines and underground cables. *IEEE Transactions on Power Delivery*, Institute of Electrical and Electronics Engineers (IEEE), v. 14, n. 3, p. 1032–1038, jul 1999. 31, 33
- 20 INTERNATIONAL, M. H. *PSCAD/EMTDC version 4.6, User's Guide*. 2019. <https://hvdc.ca/knowledge-base/read,article/160/pscad-user-s-guide-v4-6/v..> Online; accessed 03-August-2019. 31, 68
- 21 DOMMEL, H. *Electromagnetic Transients Program. Reference Manual (EMTP Theory Book)*. Portland, 1986. 33
- 22 GOLE, A. M.; VELASCO, J. M.; KERI, A. J. F. *Modeling and Analysis of System Transients Using Digital Programs*. [S.l.], 2013. 33, 34
- 23 MARTINEZ, J.; MORK, B. Closure on "transformer modeling for low- and mid-frequency transients - a review". *IEEE Transactions on Power Delivery*, Institute of Electrical and Electronics Engineers (IEEE), v. 23, n. 3, p. 1697–1697, jul 2008. 34
- 24 GROUP, I. W. *Modeling and Analysis of System Transients Using Digital Programs*. [S.l.], 2013. 34, 36
- 25 BRANDWAJN, V.; DOMMEL, H. W.; DOMMEL, I. Matrix representation of three-phase n-winding transformers for steady-state and transient studies. *IEEE Power Engineering Review*, Institute of Electrical and Electronics Engineers (IEEE), PER-2, n. 6, p. 27–28, jun 1982. 35
- 26 GUSTAVSEN, B.; SEMLYEN, A. Simulation of transmission line transients using vector fitting and modal decomposition. *IEEE Transactions on Power Delivery*, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 2, p. 605–614, apr 1998. 35, 36

- 27 BOAVENTURA, W. C. *Equivalentes de rede esparsos e robustos baseados na modelagem paramétrica no domínio do tempo*. Tese (Doutorado) — UNICAMP, 2002. 36
- 28 KIRK, D. B.; HWU, W. W. *Programando para processadores paralelos: Uma abordagem prática a programação de GPU*. Rio de Janeiro: Editora Campus/ELSEVIER, 2011. 40
- 29 JUNIOR, E. E. *Estudos de Tecnologias para Computação Paralela e Distribuída: Implementação de um cluster Beowulf*. Dissertação (Mestrado) — Universidade Federal Fluminense, 2013. 41, 48
- 30 KARYPIS, A. A. G. G. *Introduction to Parallel Computing*. Rio de Janeiro: Pearson Education Limited, 2003. 41, 42
- 31 BARNEY, B. *Introduction to parallel computing*. Califórnia, 2017. 41, 43, 45, 46, 48
- 32 TANENBAUM, A. S. *Organização Estruturada de Computadores*. São Paulo: Pearson Prentice-Hall, 2007. 42
- 33 KHAITAN, S. K. A survey of high-performance computing approaches in power systems. In: *2016 IEEE Power and Energy Society General Meeting (PESGM)*. [S.l.]: IEEE, 2016. 42
- 34 MATTSON, T. G.; SANDERS, B.; MASSINGILL, B. *Patterns for Parallel Programming*. Westford, 2010: Pearson Education, 2010. 42, 43
- 35 SPACHECO, P. *Introduction to Parallel Computing*. São Francisco: Parallel Programming with MPI, 1997. 43
- 36 COULOURIS, G. *Sistemas distribuídos: conceitos e projeto*. 5. ed. Porto Alegre: Bookman, 2013. 46, 48
- 37 COMER, D. E. *Interligação em redes com TCP/IP: Princípios, protocolo e arquitetura*. 3 ed. Rio de Janeiro: Campus, 1998. 49, 50
- 38 FALCAO, D.; KASZKUREWICZ, E.; ALMEIDA, H. Application of parallel processing techniques to the simulation of power system electromagnetic transients. *IEEE Transactions on Power Systems*, Institute of Electrical and Electronics Engineers (IEEE), v. 8, n. 1, p. 90–96, 1993. 52, 53, 56
- 39 DEBNATH, J. K. et al. Electromagnetic transient simulation of large-scale electrical power networks using graphics processing units. In: *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. [S.l.]: IEEE, 2012. 52, 55
- 40 DYCK, M.; NZIMAKO, O. Real-time simulation of large distribution networks with distributed energy resources. *CIREN - Open Access Proceedings Journal*, Institution of Engineering and Technology (IET), v. 2017, n. 1, p. 1402–1405, oct 2017. 52, 58, 59, 113
- 41 FALCÃO, D. M. High performance computing. *2nd International Meeting on Vector and Parallel Processing*, 1996. 53, 54
- 42 VENUGOPAL, R.; WANG, W.; BÉLANGER, J. Advances in real-time simulation for power distribution systems. In: *International Conference on Energy, Automation, and Signal (ICEAS)*. Odisha, India: [s.n.], 2011. 54, 55

- 43 DEBNATH, J.; GOLE, A.; FUNG, W.-K. Graphics processing unit based acceleration of electromagnetic transients simulation. In: *2016 IEEE Power and Energy Society General Meeting (PESGM)*. [S.l.]: IEEE, 2016. 55, 113
- 44 SINGH, R. et al. Using local grid and multi-core computing in electromagnetic transients simulation. In: *International Conference Power System Transients*. Vancouver, Canada: [s.n.], 2013. 55, 56, 113, 115
- 45 BARROS, H. M. et al. Efficient application of parallel processing with standard tools for electromagnetic transients simulation. *IPST*, jul. 2013. 56, 57, 113
- 46 R.YONEZAWA; NODA, T. Study of solution process parallelization for an emt analysis program using openmp ipst 2013. In: *International Power System Conference*. Seoul, Korea: [s.n.], 2017. 57, 58
- 47 HUY, P. L. et al. Massively parallel real-time simulation of very- large-scale power system. In: *International Power System Transient*. Seoul, Korea: [s.n.], 2017. 58
- 48 STEVIC, M.; MONTI, A.; BENIGNI, A. Development of a simulator-to-simulator interface for geographically distributed simulation of power systems in real time. In: *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*. [S.l.]: IEEE, 2015. 60, 113, 117
- 49 STEVIC, M. et al. Feasibility of geographically distributed real-time simulation of (hvdc) system interconnected with (ac) networks. In: *2015 IEEE Eindhoven PowerTech*. [S.l.]: IEEE, 2015. 60, 113, 114
- 50 Stevic, M. et al. Multi-site european framework for real-time co-simulation of power systems. *IET Generation, Transmission Distribution*, v. 11, n. 17, p. 4126–4135, 2017. 60, 114, 115
- 51 LIU, R. et al. Geographically distributed real-time digital simulations using linear prediction. *International Journal of Electrical Power & Energy Systems*, International Journal of Electrical Power & Energy Systems, v. 84, p. 308–317, Jan 2017. 61, 114
- 52 Monti, A. et al. A global real-time superlab: Enabling high penetration of power electronics in the electric grid. *IEEE Power Electronics Magazine*, v. 5, n. 3, p. 35–44, 2018. 61
- 53 ZAMRONI, M. *Development of EMT/TS Co-simulation Using PowerFactory and PSS/E*. Dissertação (Mestrado) — Technische Universiteit Delft, 2017. 62
- 54 SEMLYEN, A.; LEON, F. de. Computation of electromagnetic transients using dual or multiple time steps. *IEEE Transactions on Power Systems*, Institute of Electrical and Electronics Engineers (IEEE), v. 8, n. 3, p. 1274–1281, 1993. 63
- 55 GOLE, A. et al. Integrating electromagnetic transient simulation with other design tools. *International Conference on Power Systems Transients*, jun. 2001. 63, 64
- 56 FILIZADEH, S. et al. Techniques for interfacing electromagnetic transient simulation programs with general mathematical tools IEEE taskforce on interfacing techniques for simulation tools. *IEEE Transactions on Power Delivery*, Institute of Electrical and Electronics Engineers (IEEE), v. 23, n. 4, p. 2610–2622, oct 2008. 64

- 57 HEIDARI, M. *Decision Support Algorithms for Power System and Power Electronic Design*. Tese (Doutorado) — University of Manitoba, Winnipeg, Manitoba, Canadá, 2010. 64
- 58 BOAVENTURA, W. C. Large systems real-time transient calculation by means of simultaneous simulation of multiple equivalents. *IPST*, jun. 2015. 64, 65
- 59 GONÇALVES, S. M. et al. A semaphore-based parallelization of networks for electromagnetic transients. *International Conference on Power Systems Transients*, 2015. 65
- 60 MATHWORKS. *Functional Mock-up Interface for Co-Simulation*. 2019. <https://fmi-standard.org/docs/2.0.1-develop/>. Online; accessed 28-August-2019. 65
- 61 MATHWORKS. *MATLAB Distributed Computing Server User's Guide*. 2019. <https://www.mathworks.com/support/requirements/parallel-computing-toolbox.html>. Online; accessed 05-September-2019. 66
- 62 PRIKLER L.; HØIDALEN, H. K. *ATPDRAW version 5.6 for Windows 9x/NT/2000/XP /Vista*. [S.l.], 2009. 72
- 63 CENTRE, M. H. R. *PSCAD Model of IEEE 39-bus system*. 2019. <https://hvdc.ca/knowledge-base/read,article/28/ieee-39-bus-system/v.> Online; accessed 14-September-2019. 100
- 64 TECNOLOGIA, M. da Ciência e. *Rede Nacional de Pesquisa*. 2020. <https://www.rnp.br/en>. Online; accessed 09-July-2020. 109, 115
- 65 INTERNATIONAL, M. H. *PSCAD Engineering Applications*. 2020. <https://www.pscad.com/knowledge-base/topic-35/v->. Online; accessed 09-July-2020. 110
- 66 INTERNATIONAL, M. H. *PSCAD Engineering Applications(Solar)*. 2020. <https://www.pscad.com/knowledge-base/article/521>. Online; accessed 09-July-2020. 110
- 67 INTERNATIONAL, M. H. *PSCAD Engineering Applications(Wind)*. 2020. <https://www.pscad.com/knowledge-base/article/227>. Online; accessed 09-July-2020. 110
- 68 ELÉTRICO, O. N. do S. *Diretrizes para Elaboração de Projetos Básicos para Empreendimentos de Transmissão*. Rio de Janeiro: ONS, 2013. 116
- 69 AL, J. H. et. Multi-rate and parallel electromagnetic transient simulation considering nonlinear characteristics of a power system. *Energies*, 2018. 116
- 70 MU, Q. e. a. Parallel multi-rate electromagnetic transient simulation algorithm based on network division through transmission line. *Automation of Electric Power System*, v. 38, 2014. 116

# Anexos

## ANEXO A – Artigos publicados

1. Gomes, J. P. R. ; Boaventura, W.C. A TOOL FOR DISTRIBUTED COLLABORATIVE ELECTROMAGNETIC TRANSIENTS STUDIES ALLOWING ON-DEMAND INTERACTION. International Journal of Electrical Power and Energy Systems, 7 Feb 2022. ISSN 0142-0615.

2. Gomes, J. P. R.; Boaventura, W.C. ; Jose Osvaldo Saldanha Paulino . Computação Paralela Aplicada ao Cálculo de Transitórios Eletromagnéticos: Incidência de Descargas Atmosféricas em Linhas de Distribuição. In: XXII CONGRESSO BRASILEIRO DE AUTOMÁTICA, 2018, João Pessoa. XXII Congresso Brasileiro de Automática, 2018.

3. Oliveira, J. H. ; Gomes, J. P. R. ; Boaventura, W.C. ; Mendes, V. F. Impact of Integrating a Photovoltaic Power Plant in a Distribution Feeder. In: 17th International Conference on Harmonics and Quality of Power (ICHQP), 2016.

4. Gomes, J. P. R. ; Boaventura, W.C. ; Cardoso, E. N. ; Lopes, B. M. . Impact of integrating a solar power plant in a distribution feeder considering short circuit events.. In: 13th Brazilian Power Electronics Conference - COBEP, 2015.