

Universidade Federal de Juiz de Fora
Faculdade de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Igor Borges de Oliveira Chagas

Metodologias de Co-simulação Aplicadas a Sistemas de Potência

Juiz de Fora

2022

Igor Borges de Oliveira Chagas

Metodologias de Co-simulação Aplicadas a Sistemas de Potência

Dissertação apresentada ao programa de Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas de Energia, como requisito parcial para obtenção do grau de Mestre

Orientador: Prof. Marcelo Aroca Tomim, Ph.D.

Juiz de Fora

2022

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Borges de Oliveira Chagas, Igor.

Metodologias de Co-simulação Aplicadas a Sistemas de Potência / Igor
Borges de Oliveira Chagas. – 2022.

152 f. : il.

Orientador: Prof. Marcelo Aroca Tomim, Ph.D.

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade
de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2022.

1. Co-simulação 2. Simulação de sistemas de potência 3. Functional
Mock-up Interface.

Igor Borges de Oliveira Chagas

Metodologias de Co-simulação Aplicadas a Sistemas de Potência

Dissertação apresentada ao programa de Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas de Energia, como requisito parcial para obtenção do grau de Mestre

Aprovada em:

BANCA EXAMINADORA

Prof. Marcelo Aroca Tomim, Ph.D. - Orientador
Universidade Federal de Juiz de Fora

Prof. João Alberto Passos Filho, D.Sc.
Universidade Federal de Juiz de Fora

Prof. Antônio Carlos Siqueira de Lima, D.Sc.
Universidade Federal do Rio de Janeiro

Paulo César Magalhães Meira, D.Sc.
Universidade Estadual de Campinas

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me concedido o privilégio de chegar até aqui. Ele tornou possível tudo isso, ensinando-me a cada dia e me fortalecendo com sua Presença.

Aos meus pais, Marcelo e Cassiana, que estiveram comigo durante todo o tempo, com todo amor e carinho. Agradeço por seus ensinamentos, por terem me incentivado durante esse período, se colaram a disposição em me ajudar, e principalmente, por estarem sempre orando por mim.

Ao meu orientador, Marcelo Aroca Tomim, por ter me confiado esse importante trabalho. Por toda a paciência, incentivo, tempo, comprometimento ao longo dos últimos anos que trabalhamos juntos, sendo assim, uma inspiração para mim e demais alunos.

À Universidade Federal de Juiz de Fora (UFJF) e ao Programa de Pós-Graduação em Engenharia Elétrica (PPEE) pelo acolhimento, suporte técnico, oportunidade de participar do mestrado. Aos professores, pelo comprometimento, empenho, desempenhando importante papel em minha formação, tanto no caráter profissional como no pessoal.

Aos meus amigos do mestrado, e aqueles que estiveram comigo desde a graduação - José Carlos Faria, Jeanderson Mingorança, Glauber Daniel, Antônio Sobrinho, Fernando Rocha, Amanda Pávila, Pâmela Taveira, Ricardo Proba, e outros que foram também muito importantes no meu desenvolvimento profissional e pessoal. Aos meus amigos de outros cursos da UFJF e aos de Cataguases, que também me apoiaram, se preocuparam comigo e aqueles que investiram em mim.

Agradeço à parceria entre a Petrobras e a Universidade Federal de Juiz de Fora, por meio do projeto de P&D - Setor Elétrico da ANEEL nº 5900.0112828.19.9, a qual contribuiu para a realização do presente estudo.

"Não fui eu que ordenei a você? Seja forte e corajoso! Não se apavore nem desanime, pois o Senhor, o seu Deus, estará com você por onde você andar" Josué 1:9

RESUMO

Essa dissertação apresenta técnicas de co-simulação para simulação de sistemas de transmissão dinâmicos acoplados a sistemas de distribuição. A estratégia da co-simulação permite que sistemas complexos e heterogêneos sejam divididos em subsistemas menores, cuja modelagem e solução podem ser implementadas em plataformas especializadas. Porém, é de vital importância garantir a compatibilidade entre as interfaces do sistema. Nesse âmbito, será utilizada a interface padronizada FMI (*Functional Mock-up Interface*), que permitirá a troca de informações de modelos dinâmicos embutidos em FMUs (*Functional Mock-up Units*). Uma vantagem que a co-simulação oferece é a possibilidade da integração de modelos de domínios diferentes. Além disso, essas unidades podem ser distribuídas como bibliotecas de acesso dinâmico compiladas, que podem ser compartilhadas enquanto a propriedade intelectual é protegida. Embora a co-simulação não tenha sido amplamente explorada para estudos dinâmicos de sistemas de potência, a técnica pode ser eficaz para estudos dinâmicos de sistemas acoplados. Nesse contexto, sistemas de transmissão e distribuição podem ser simulados em diferentes plataformas computacionais, com suas técnicas mais eficientes, que trocam dados entre si de forma cíclica. Dessa forma, cinco técnicas de co-simulação serão apresentadas nesse trabalho. Serão apresentados testes de co-simulação, utilizando a metodologia da linha fictícia, entre sistemas modelados em Modelica e embutidos em FMUs. Os resultados mostraram que a co-simulação replicou de forma bastante fiel os resultados de simulações para o mesmo sistema modelado de forma completa, sem partições, obtidos no OpenModelica. Além disso, a estratégia proposta apresentou tempos computacionais reduzidos e, em algumas situações, se mostrou como a única alternativa para simular os estudos propostos. Nesse trabalho também foram reportadas metodologias para a co-simulação entre FMUs e sistemas de distribuição modelados no OpenDSS. Atrasos observados nas estratégias baseadas em linhas fictícias e no método com atraso foram eliminados completamente através de métodos iterativos alternados. Esses métodos também possibilitaram a redução do tempo computacional ao mesmo tempo que mantiveram a precisão dos resultados.

Palavras-chave: 1. Co-simulação 2. Simulação de sistemas de potência 3. *Functional Mock-up Interface*. 4. OpenDSS 5. Sistemas de transmissão 6. Sistemas de Distribuição

ABSTRACT

This master thesis presents co-simulation techniques for simulating dynamic transmission systems coupled to distribution systems. The co-simulation strategy allows complex and heterogeneous systems to be divided into smaller subsystems, whose modeling and solution can be implemented on specialized platforms. However, it is vitally important to ensure compatibility between system interfaces. In this context, the standardized FMI interface Functional Mock-up Interface will be used, which will allow the exchange of information from dynamic models embedded in FMUs (Functional Mock-up Units). Furthermore, as FMUs can be distributed as compiled dynamic-access libraries, they can be shared while intellectual property is protected. Although co-simulation has not been widely explored for dynamic studies of power systems, the technique can be effective for dynamic studies of coupled systems. In this context, transmission and distribution systems can be simulated on different computational platforms, with their most efficient techniques, that exchange data between them in a cyclical way. Thus, five co-simulation techniques will be presented in this work. Co-simulation tests will be presented, using the fictitious line methodology, between systems modeled in Modelica and embedded in FMUs. The results showed that the co-simulation quite faithfully replicated the simulation results for the same modeled system completely, without partitions, obtained in OpenModelica. In addition, the proposed strategy presented reduced computational times and, in some situations, proved to be the only alternative to simulate the proposed studies. In this work, methodologies for co-simulation between FMUs and distribution systems modeled in OpenDSS were also reported. Delays observed in the fictitious-line-based strategies and in the delayed method were completely eliminated through alternating iterative methods. These methods also made it possible to reduce computational time while maintaining the accuracy of the results.

Key-words: 1. Co-simulation 2. Power system simulation 3. Functional Mock-up Interface. 4. OpenDSS 5. Transmission Systems 6. Distribution Systems

LISTA DE ILUSTRAÇÕES

Figura 1 – Cenários presente e futuro do sistema elétrico	17
Figura 2 – Co-simulação em um único processo computacional	29
Figura 3 – Estrutura de co-simulação distribuída	29
Figura 4 – Estrutura de uma FMU do tipo co-simulação	31
Figura 5 – Resultados da simulação do sistema teste.	38
Figura 6 – Circuito equivalente de uma linha de transmissão ideal no domínio do tempo.	45
Figura 7 – Subsistemas da co-simulação	47
Figura 8 – Fluxograma do processo de co-simulação	51
Figura 9 – Módulo do autovalor para co-simulação entre sistemas de transmissão e 38 barras	55
Figura 10 – Quantidades medidas durante a falta - PyFMI - Sistema de transmissão e 38 barras	57
Figura 11 – Quantidades medidas durante a falta - OMSimulator - Sistema de transmissão e 38 barras	58
Figura 12 – Quantidades medidas durante a falta - PyFMI - Sistema de transmissão e dois alimentadores de 38 barras.	60
Figura 13 – Quantidades medidas durante a falta - OMSimulator - Sistema de transmissão e dois alimentadores de 38 barras.	61
Figura 14 – Tempos computacionais para o sistema de transmissão conectado a vários alimentadores de 38 barras.	62
Figura 15 – Barra do OpenDSS	64
Figura 16 – Terminal do OpenDSS	64
Figura 17 – Inicialização - Processo iterativo entre OpenModelica e OpenDSS.	69
Figura 18 – Iteração entre o OpenDSS e a linha fictícia.	71
Figura 19 – Módulo dos autovalores para o sistema de transmissão e distribuição de 34 barras.	75
Figura 20 – Grandezas medidas - Sistema de transmissão e IEEE 34 barras - Método da linha fictícia	77
Figura 21 – Tensão na barra de interface - Variação de \bar{Z}_c	78
Figura 22 – Módulo dos autovalores para o sistema de transmissão e distribuição Ckt5.	80
Figura 23 – Grandezas medidas - Sistemas de transmissão e Ckt5 - Método da linha fictícia	81
Figura 24 – Co-simulação entre FMU e OpenDSS - Método com atraso	84
Figura 25 – Iteração entre FMU e OpenDSS - Método alternado.	87
Figura 26 – Equivalente de Norton.	92
Figura 27 – Interação entre FMU e OpenDSS através da sensibilidade.	94

Figura 28 – Grandezas medidas - Sistemas de transmissão e IEEE 34 barras - Método com atraso	101
Figura 29 – Grandezas medidas na interface entre sistema de transmissão e IEEE 34 barras - Método com atraso	102
Figura 30 – Grandezas medidas - Sistemas de transmissão e IEEE 34 barras - Método alternado	103
Figura 31 – Grandezas medidas na interface entre sistema de transmissão e IEEE 34 barras - Método alternado	104
Figura 32 – Medidas de desempenho do método alternado	105
Figura 33 – Medidas de desempenho do método alternado "desonesto" com sensibilidade - Sistemas de transmissão de 11 barras e distribuição de 34 barras	105
Figura 34 – Comportamento da admitância - Método alternado "desonesto".	106
Figura 35 – Grandezas medidas - Co-simulação entre sistema de transmissão e distribuição Ckt5 - Método com atraso	108
Figura 36 – Grandezas medidas na interface entre sistema de transmissão e o Ckt5 - Método com atraso	109
Figura 37 – Grandezas medidas - Sistema de transmissão e Ckt5 - Método alternado	110
Figura 38 – Grandezas medidas na interface entre sistema de transmissão e o Ckt5 - Método alternado	111
Figura 39 – Medidas de desempenho do método alternado - Sistemas de transmissão de 11 barras e Ckt5	112
Figura 40 – Medidas de desempenho do método alternado com sensibilidade - Co-simulação entre sistema de transmissão e Ckt5.	113
Figura 41 – Comportamento da admitância - Método alternado "desonesto".	114
Figura 42 – Sistema teste de 11 barras.	124
Figura 43 – Diagrama de blocos do AVR + PSS.	125
Figura 44 – Sistema teste de 38 barras.	127
Figura 45 – Sistema de distribuição IEEE 34-barras.	130
Figura 46 – Sistema de distribuição - Ckt5	137
Figura 47 – Impedância de Thévenin do sistema de distribuição vista do nó de interconexão	146
Figura 48 – Modelo de corrente de compensação de elementos de conversão de energia	147
Figura 49 – Sistema com linha de transmissão	149

LISTA DE TABELAS

Tabela 1 – Valores para $x(t)$	38
Tabela 2 – Estimativa das impedâncias de Thévenin - Sistemas de transmissão e 38 barras	55
Tabela 3 – Estimativas da impedância característica - Sistemas de transmissão e 38 barras	56
Tabela 4 – Valores NIAE - PyFMI (Valores em [%])	57
Tabela 5 – Valores NIAE - OMSimulator (Valores em [%])	57
Tabela 6 – Parâmetros do Elemento <i>Circuit</i>	65
Tabela 7 – Estimativas das impedâncias de Thévenin - Sistemas de transmissão e IEEE 34 barras	74
Tabela 8 – Estimativa da impedância característica - Sistemas de transmissão e IEEE 34 barras	76
Tabela 9 – Valores NIAE - Sistema de transmissão e IEEE 34 barras [%]	76
Tabela 10 – Perfil da co-simulação - Sistemas de transmissão e IEEE 34 barras	77
Tabela 11 – Variação da impedância característica	78
Tabela 12 – Estimativas das impedâncias de Thévenin - Sistemas de transmissão e Ckt5	79
Tabela 13 – Estimativa da impedância característica - Sistemas de transmissão e Ckt5	79
Tabela 14 – Valores NIAE - Sistema de transmissão e Ckt5 [%]	81
Tabela 15 – Perfil da co-simulação - Sistemas de transmissão e Ckt5.	82
Tabela 16 – Estimativa de admitância do Equivalente de Norton - IEEE 34 barras.	102
Tabela 17 – Perfil da co-simulação - Sistemas de transmissão e IEEE-34 barras (tempos em [s]).	106
Tabela 18 – NIAE - Sistemas de transmissão e IEEE-34 barras (Valores em [%]).	107
Tabela 19 – Estimativa de admitância do Equivalente de Norton - Ckt5.	111
Tabela 20 – Perfil da co-simulação - Sistemas de transmissão e o Ckt5 (tempos em [s]).	114
Tabela 21 – NIAE - Sistemas de transmissão e Ckt5 (Valores em [%]).	115
Tabela 22 – Parâmetros elétricos de cada unidade geradora	125
Tabela 23 – Parâmetros dos reguladores de tensão e estabilizadores de potência.	126
Tabela 24 – Parâmetros das linhas de transmissão	126
Tabela 25 – Comprimento da linha de transmissão	126
Tabela 26 – Dados de geração para o sistema de distribuição	127
Tabela 27 – Dados de carga do sistema de 38 barras	128
Tabela 28 – Dados de linha de transmissão do sistema de 38 barras (base de 100 MVA)	129
Tabela 29 – Dados dos transformadores - IEEE 34 barras	130
Tabela 30 – Configuração das linhas de distribuição - IEEE 34 barras	131

Tabela 31 – Comprimento das linhas de distribuição - IEEE 34 barras	133
Tabela 32 – Cargas locais	134
Tabela 33 – Cargas distribuídas	134
Tabela 34 – Códigos dos modelos de carga	135
Tabela 35 – Dados dos bancos de capacitores	135
Tabela 36 – Dados do regulador 1	135
Tabela 37 – Dados do regulador 2	136
Tabela 38 – Características gerais do sistema Ckt5.	138
Tabela 39 – Inicialização da co-simulação para sistemas de transmissão e 38 barras - Sistema de 11 barras	139
Tabela 40 – Inicialização da co-simulação para sistemas de transmissão e 38 barras - Sistema de 38 barras	140
Tabela 41 – Inicialização da co-simulação para sistemas de transmissão e 34 barras - Sistema de 11 barras	141
Tabela 42 – Inicialização da co-simulação para sistemas de transmissão e 34 barras - Sistema de 34 barras	142

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
AVR	<i>Automatic Voltage Regulator</i>
COM	<i>Component Object Model</i>
CS	Co-simulação (do inglês, <i>Co-simulação</i>)
EMT	Transitórios eletromagnéticos (do inglês, <i>Electromagnetic Transient</i>)
EMTP	Programa transitórios eletromagnéticos (do inglês, <i>Electromagnetic Transient Program</i>)
EPRI	<i>Electric Power Research Institute</i>
FMI	<i>Functional Mock-up Interface</i>
FMU	<i>Functional Mock-up Unit</i>
GD	Geração Distribuída
HLA	Estrutura de alto nível (do inglês <i>High Level Architecture</i>)
IAE	Integral do erro absoluto (do inglês, <i>Integral Absolute-Error</i>)
ME	Intercâmbio entre modelos (do inglês, <i>Model Exchange</i>)
MPI	<i>Message Passing Interface</i>
NIAE	Integral do erro absoluto normalizado (do inglês, <i>Normalized Integral Absolute-Error</i>)
OpenDSS	<i>Open System Distribution Simulator</i>
PAC	Ponto de Acoplamento Comum
PSS	<i>Power System Stabilizer</i>
REDS	Recursos Energéticos Distribuídos
RTDS	<i>Real Time Digital Simulator</i>
SEP	Sistemas Elétricos de Potência
SUNDIALS	<i>Suite of Nonlinear and Differential/Algebraic Equation Solvers</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>

TLM	<i>Transmission Line Model</i>
VBA	<i>Visual Basic for Applications</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	IDENTIFICAÇÃO DO PROBLEMA	16
1.2	MOTIVAÇÃO	18
1.3	OBJETIVOS	19
1.4	PUBLICAÇÕES	19
1.5	ESTRUTURA DA DISSERTAÇÃO	20
2	REVISÃO BIBLIOGRÁFICA	22
3	FUNDAMENTOS DA CO-SIMULAÇÃO	27
3.1	PADRÃO FMI (<i>FUNCTIONAL MOCK-UP INTERFACE</i>)	27
3.1.1	Padrão FMI para co-simulação	29
3.1.2	<i>Functional Mock-up Unit</i> (FMU)	30
3.1.3	Solver CVode	31
3.1.4	Algoritmo mestre	32
3.1.5	Ferramentas de suporte à co-simulação	33
3.1.6	Tutorial de exportação e importação de unidades FMU	34
3.2	CONCLUSÕES PARCIAIS	41
4	CO-SIMULAÇÃO ATRAVÉS DE LINHA FICTÍCIA	42
4.1	MODELAGEM DA LINHA FICTÍCIA	43
4.1.1	Procedimento de co-simulação	50
4.2	RESULTADOS DE CO-SIMULAÇÃO COM FMUs	54
4.2.1	Análises de erro	54
4.2.2	Co-simulação entre sistema transmissão e ramais de 38 barras	54
4.2.3	Múltiplos Alimentadores	59
4.3	CO-SIMULAÇÃO ENTRE OPENMODELICA E OPENDSS	63
4.3.1	Software OpenDSS	63
4.3.2	Fluxo de potência do OpenDSS	65
4.3.3	Modo de Inicialização	66
4.3.4	Linha Fictícia entre FMU e OpenDSS	69
4.4	CO-SIMULAÇÃO ENTRE SISTEMA DE TRANSMISSÃO E IEEE - 34 BARRAS	74
4.5	CO-SIMULAÇÃO ENTRE SISTEMA DE TRANSMISSÃO E CKT5	79
4.6	CONCLUSÕES PARCIAIS	82
5	METODOLOGIAS DE CO-SIMULAÇÃO	84

5.1	MÉTODO COM ATRASO	84
5.2	MÉTODO ALTERNADO	86
5.3	MÉTODO ALTERNADO COM SENSIBILIDADE	89
5.3.1	Levantamento do Equivalente de Norton dos Sistemas de Distribuição	92
5.3.2	Modo de Integração	94
5.4	MÉTODO ALTERNADO DESONESTO	95
5.5	RESULTADOS DE CO-SIMULAÇÃO	100
5.5.1	Co-simulação entre sistema de transmissão e IEEE 34 barras .	100
5.5.1.1	<i>Método com atraso</i>	100
5.5.1.2	<i>Método alternado</i>	100
5.5.1.3	<i>Método alternado com sensibilidade</i>	101
5.5.1.4	<i>Método alternado "desonesto" com sensibilidade</i>	103
5.5.2	Análise de Desempenho Computacional	106
5.5.3	Co-simulação entre Sistema de Transmissão e Ckt5	107
5.5.3.1	<i>Método com atraso</i>	108
5.5.3.2	<i>Método alternado</i>	109
5.5.3.3	<i>Método alternado com sensibilidade</i>	110
5.5.3.4	<i>Método alternado "desonesto" com sensibilidade</i>	112
5.5.4	Análise de Desempenho Computacional	113
5.6	CONCLUSÕES PARCIAIS	115
6	CONCLUSÕES E DESENVOLVIMENTOS FUTUROS	117
6.1	TRABALHOS FUTUROS	118
	REFERÊNCIAS	120
	APÊNDICE A – SISTEMAS TESTE	124
A.1	SISTEMA DE TRANSMISSÃO	124
A.2	SISTEMAS DE DISTRIBUIÇÃO	126
A.2.1	Sistema de 38 barras	126
A.2.2	Sistema IEEE 34-barras	130
A.2.3	Sistema Ckt5	137
	APÊNDICE B – INICIALIZAÇÃO DA CO-SIMULAÇÃO .	139
B.1	CO-SIMULAÇÃO ENTRE SISTEMAS DE TRANSMISSÃO DE 11 BARRAS E DE DISTRIBUIÇÃO DE 38 BARRAS	139
B.2	CO-SIMULAÇÃO ENTRE SISTEMAS DE TRANSMISSÃO DE 11 BARRAS E DE DISTRIBUIÇÃO IEEE 34 BARRAS	139

B.3	CO-SIMULAÇÃO ENTRE SISTEMAS DE TRANSMISSÃO DE 11 BARRAS E DE DISTRIBUIÇÃO CKT5	139
	APÊNDICE C – ESTIMATIVA DA IMPEDÂNCIA DE THÉ- VENIN	143
C.1	ESTIMATIVA A PARTIR DE UM SISTEMA EM SEQUÊNCIA POSI- TIVA	143
C.2	ESTIMATIVA A PARTIR DE UM SISTEMA TRIFÁSICO	144
	APÊNDICE D – FLUXO DE POTÊNCIA DO OpenDSS . .	147
	APÊNDICE E – COEFICIENTE DE REFLEXÃO PARA UMA LINHA DE TRANSMISSÃO IDEAL	149
	APÊNDICE F – ESPAÇO DE ESTADOS PARA SISTEMAS NO DOMÍNIO DISCRETO	151

1 INTRODUÇÃO

O presente trabalho está relacionado à linha de investigação da avaliação dos impactos da penetração de Recursos Energéticos Distribuídos (REDs) na dinâmica dos sistemas elétricos de grande porte. Esse é um capítulo introdutório, onde na Seção 1.1 são detalhados os desafios encontrados para a realização de estudos dinâmicos diante das mudanças que vêm acontecendo nos Sistemas Elétricos de Potência (SEP). Na Seção 1.2 são relatadas as motivações encontradas para a proposta desse trabalho enquanto na Seção 1.3 são apresentados os objetivos. Por fim, na Seção 1.5 é apresentada a estrutura do presente documento.

1.1 IDENTIFICAÇÃO DO PROBLEMA

Com o passar dos anos, a demanda por energia elétrica tem aumentado significativamente. E com isso, fatores relacionados a conscientização ambiental, incentivos de descarbonização e a descentralização do sistema impulsionam mudanças no setor elétrico do Brasil e do mundo. Preocupações relativas à busca de novas fontes de energia limpa proporcionam o desenvolvimento e integração de Recursos Energéticos Distribuídos (REDs) e renováveis nos Sistemas Elétricos de Potência [1].

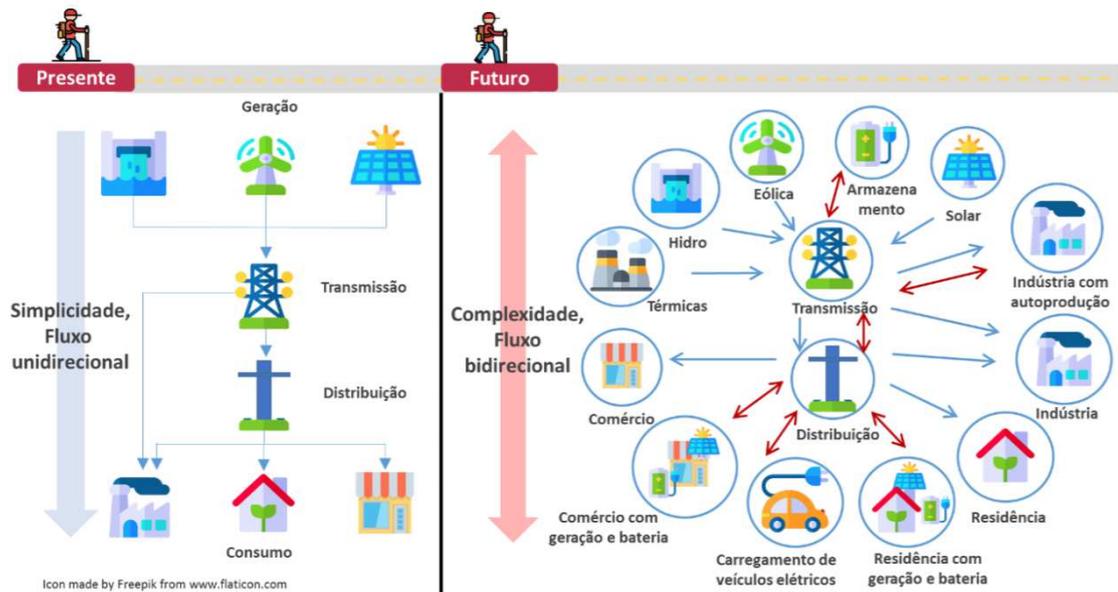
No que diz respeito à matriz elétrica brasileira, cerca de 84,8 % da energia elétrica é proveniente da geração a partir de fontes renováveis sendo 65,1 % proveniente de usinas hidrelétricas [2]. Há uma previsão de crescimento das fontes renováveis não hidráulicas em 9% até 2030 [3].

Com a inserção dos REDs, os sistemas de potência começaram a passar por grandes mudanças em sua topologia, devido às novas necessidades de geração e consumo, que exigem a conexão de novas fontes de geração e cargas. Segundo [4], os REDs podem ser definidos como tecnologias de geração e/ou armazenamento de energia elétrica, localizados dentro dos limites da área de uma determinada concessionária de distribuição. Nessa classificação podem ser incluídas geração distribuída (GD), sistemas de armazenamento de energia, veículos elétricos, estrutura de recarga e micro-redes.

A geração conectada aos sistemas de distribuição e baseada em REDs, começou a ser adotada em um ritmo tão acelerado que surgiram preocupações com relação ao seu impacto na operação dos sistemas elétricos modernos. A Figura 1 mostra o cenário atual e futuro do sistema elétrico em decorrência da inserção dos REDs.

Pode se observar uma transição de um modelo centralizado para um modelo distribuído. Além disso, devido à natureza das cargas e a alta inserção de REDs os fluxos de potência podem deixar de ser unidirecionais, passando a ser bidirecionais. Esse fato implica no aumento da complexidade de planejamento e operação dos sistemas de potência.

Figura 1 – Cenários presente e futuro do sistema elétrico



Fonte: Empresa de Pesquisa Energética (2017)

Além disso, riscos de alterações severas na dinâmica dos sistemas elétricos que tais recursos possam vir a causar têm se tornado motivo de preocupação por parte dos operadores.

Nesse contexto, ferramentas de análise de sistemas de potência mais avançadas tem sido requeridas. Um dos motivos é que a interconexão dos REDs com um sistema de potência pode ser bastante heterogênea, exigindo modelagem e técnicas de computação específicas [5].

O segundo motivo é que as simulações de tais sistemas podem ser bastante complexas e demoradas, podendo exigir até abordagens estocásticas, devido à característica intermitente dos REDs.

Assim, a co-simulação surge como uma alternativa para avaliar sistemas de potência altamente penetrados por REDs. A estratégia de co-simulação permite que sistemas complexos e heterogêneos sejam divididos em subsistemas menores, cuja modelagem e solução podem ser implementadas em plataformas especializadas [5, 6].

Como uma vantagem, a co-simulação permite integrar subsistemas que podem até mesmo ser modelados em diferentes domínios (por exemplo, domínio do tempo e frequência). Outra vantagem seria o tempo computacional reduzido quando comparado à simulação de um determinado sistema utilizando uma única ferramenta. Como uma terceira vantagem, na co-simulação, cada simulador pode ser representado como uma espécie de caixa preta, na qual há uma modelagem detalhada para o sistema desenvolvida por especialistas. Essa abordagem, permite que o desenvolvedor forneça as soluções sem precisar revelar o modelo [7].

Com base nas vantagens mencionadas, as técnicas de co-simulação permitem a integração de modelos através da interface padronizada FMI, *Functional Mock-up Interface*. Essa interface garante compatibilidade de acoplamento entre sistemas heterogêneos. Os subsistemas gerados em conformidade com a interface FMI são encapsulados em modelos padronizados chamados de *Functional Mock-up Unit* (FMU) [5, 6]. Essa abordagem será discutida no próximo capítulo.

1.2 MOTIVAÇÃO

Com a inserção de recursos energéticos distribuídos, os sistemas elétricos de potência tem se tornado cada vez mais complexos, tornando a realização de estudos dinâmicos desses sistemas cada vez mais desafiador. Esses estudos necessitam que os sistemas de transmissão e distribuição bem como os REDs sejam representados adequadamente.

No que se relaciona com a simulação de sistemas de transmissão e distribuição, nas abordagens convencionais, os sistemas de transmissão e distribuição são estudados separadamente. Atualmente, existem ferramentas como PSS/E, PSLF, PSAT, ANAREDE e ANATEM entre outras ferramentas para a análises a nível de sistemas de transmissão e ferramentas como OpenDSS para a análises a nível de sistemas distribuição [8]. Mas há poucas ferramentas disponíveis para análise combinada de sistemas de transmissão e distribuição, com exceção de simuladores EMTP como PSCAD, ATP e *SimPowerSystems* [8], que podem ser demorados quando comparados com simuladores que exigem um menor grau de detalhamento. Em simulações do tipo EMT, onde se deseja alta precisão, são requeridos passos de simulação muito pequenos, e conseqüentemente, eleva-se o esforço computacional.

Normalmente, simulações do sistema de transmissão são feitas considerando os ramais de distribuição como uma carga fixa e equilibrada, enquanto as simulações do sistema de distribuição, em geral, consideram como representação para o sistema de transmissão, uma barra infinita, embora permitam realizar análises trifásicas com desequilíbrios. Simulações do sistema de transmissão com representação de vários tipos de cargas não podem capturar todo o efeito dos componentes da rede de distribuição como transformadores, reguladores, compensadores, GDs, etc. Modelos dinâmicos de cargas como motores de indução e dispositivos de interface eletrônica precisam ser representados com mais precisão para serem incluídos na rede de distribuição. Portanto, há uma grande necessidade de uma ferramenta que permita realizar estudos dinâmicos de sistemas de transmissão e distribuição acoplados [7, 8].

Nesse contexto, o tema a ser abordado no trabalho vem ao encontro dessas necessidades, permitindo o acoplamento entre ferramentas distintas com métodos de solução específicos. As metodologias a serem apresentadas permitirão que cada sistema seja resolvido independentemente do outro, com um ambiente próprio para troca de informações

entre esses sistemas. Além disso, é possível obter ganhos em termos de desempenho computacional e precisão dos resultados em relação a simulações de sistemas sem partições.

1.3 OBJETIVOS

Diante do exposto nas seções anteriores, este trabalho visa esclarecer as investigações sobre a aplicação da co-simulação na avaliação dinâmica de sistemas de potência compostos por sistemas de transmissão e distribuição. Os seguintes tópicos a serem tratados serão:

- Apresentar ferramentas eficazes de co-simulação em linguagem Python para o acoplamento entre sistemas de transmissão e distribuição;
- Desenvolver uma metodologia baseada em linha fictícia para troca de informações entre sistemas de transmissão e distribuição encapsulados em FMUs:
 - Descrever de forma detalhada os passos implementados no algoritmo mestre para gerenciamento da co-simulação com FMUs
- Apresentar o gerenciamento da co-simulação entre as ferramentas OpenModelica e OpenDSS
 - Apresentar um algoritmo mestre para o acoplamento entre as ferramentas pela metodologia da linha fictícia, bem como discutir os desafios encontrados;
 - Desenvolver algoritmos mestre para o acoplamento entre as ferramentas baseadas nos métodos com atraso, alternado, alternado com sensibilidade e o desonesto;
- Apresentar resultados para a co-simulação entre sistemas de transmissão e distribuição modelados em linguagem Modelica encapsulados em FMUs, bem como para a co-simulação entre sistema de transmissão encapsulados em FMUs e de distribuição modelados no OpenDSS.

1.4 PUBLICAÇÕES

A principal publicação desse trabalho é:

- CHAGAS, I. B. O.; TOMIM, M. A. Co-Simulation Applied to Power Systems with High Penetration of Distributed Energy Resources. Power Systems Computation Conference (PSCC), 2022

1.5 ESTRUTURA DA DISSERTAÇÃO

Além da introdução, o trabalho está dividido na seguinte forma:

1. No Capítulo 2 será apresentada uma revisão bibliográfica para esse trabalho. Nesse capítulo, serão relatados alguns trabalhos encontrados na literatura relacionados a abordagem da co-simulação.
2. No Capítulo 3 serão apresentados os conceitos fundamentais de co-simulação, como também a interface padronizada FMI que permite a troca de informações entre subsistemas de co-simulação. Serão ainda apresentadas as principais ferramentas de suporte à co-simulação, tanto para a implementação do algoritmo mestre, como para a geração dos subsistemas de interface padronizada.
3. No Capítulo 4 será apresentada a metodologia da linha fictícia para a interface entre os subsistemas da co-simulação. Será detalhado o gerenciamento da co-simulação com FMUs e apresentados resultados para o acoplamento entre um sistema de transmissão e ramais de distribuição modelados no OpenModelica e embutidos em FMUs. Também será apresentado o gerenciamento da co-simulação entre o OpenModelica e OpenDSS através da linha fictícia. Por fim, serão apresentados resultados para a metodologia e discutidas as limitações encontradas.
4. No Capítulo 5 serão apresentadas outras metodologias de co-simulação que foram bem sucedidas para a co-simulação entre o OpenModelica e o OpenDSS baseadas no método atrasado, alternado, alternado com sensibilidade e alternado com sensibilidade desonesto. Serão apresentados resultados para cada uma das metodologias, e discutidos os resultados em termos de precisão e eficiência computacional.
5. Finalmente, no Capítulo 6 serão apresentadas as conclusões finais e propostas para trabalhos futuros.
6. No Apêndice A são fornecidos os parâmetros dos sistemas teste utilizados nesse trabalho.
7. No Apêndice B são apresentados resultados da inicialização da co-simulação de cada estudo de caso.
8. No Apêndice C são descritos os passos para a estimativa da impedância de Thévenin, a partir de um sistema representado por sequência positiva e também a partir de um sistema trifásico.
9. No Apêndice D explica-se o método de fluxo de potência realizado pelo *software* OpenDSS, descrevendo os passos do algoritmo utilizado.

10. No Apêndice E é apresentado como o coeficiente de reflexão para uma linha de transmissão ideal pode ser obtido, sendo esse entendimento importante para a metodologia da linha fictícia.
11. No Apêndice F é descrito como as amostras podem ser obtidas, a partir de um sistema de espaço de estados no domínio discreto.

2 REVISÃO BIBLIOGRÁFICA

Na literatura, várias abordagens de co-simulação vem sendo exploradas. Primeiramente, é necessário compreender que, para o acoplamento entre ferramentas, um padrão de interface deve ser estabelecido, a fim de evitar incompatibilidades entre a comunicação entre os simuladores. A seguir serão apresentados alguns padrões encontrados, bem como o padrão a ser utilizado na abordagem de co-simulação do presente trabalho.

- **IEEE 1516 HLA**

O primeiro é o IEEE 1516 HLA (*High Level Architecture*), [9, 10]. Essa arquitetura de alto nível, define como criar uma simulação global composta por simuladores distribuídos que interagem sem serem recodificados. Em uma simulação baseada em HLA, cada simulador participante é chamado de federado e interage com outros federados em uma federação HLA. Essa estrutura é feita graças a um componente central chamado *Run Time Infrastructure* (RTI). Esse componente fornece um conjunto de serviços que são necessários para ajudar a coordenação de federados em suas operações e troca de dados durante uma execução, habilita a sincronização das simulações e encaminhamento de comunicações entre os federados. Para permitir a comunicação bidirecional entre os simuladores, o RTI fornece uma API. Essa API deve ser implementada pelos federados, de forma a tornar possível o acesso aos atributos e funções específicas dos simuladores. Ademais, o padrão permite o acoplamento de múltiplos domínios e tem como principais vantagens a interoperabilidade e reutilização. Como desvantagens, pode-se mencionar a dificuldade de compatibilidade de simuladores com o padrão, quando o código fonte não é disponível.

Ainda se tratando do padrão HLA, um exemplo de ferramenta baseada nessa estrutura é o EPOCHS [11]. A ferramenta utiliza a abordagem de co-simulação para combinar simuladores de comunicação NS-2 (*Network Simulator*), simuladores de transitórios eletromagnéticos PSCAD/EMTDC e o simulador de transitórios eletromecânicos PSLF. A combinação tornará possível ao usuário escolher a avaliação de transitórios eletromagnéticos ou eletromecânicos com comunicação. Além disso, o componente central RTI permitirá a troca de dados entre os simuladores.

- **Padrão DEVs**

Esse padrão é uma abordagem baseada em eventos. Ele não apenas permite a descrição de sistemas de eventos discretos, mas também permite integrar sistemas contínuos, expressos com equações diferenciais. Além disso, DEVs pode encapsular solucionadores de equações diferenciais e algébricas com uma abordagem de integrador quantizado como o método QSS (*Quantized State System*). Esse método é baseado na quantização de estado em vez da discretização no tempo usada pelos métodos

tradicionais de integração. Esta estratégia apresenta em alguns casos melhores desempenhos em relação aos métodos tradicionais. Esse método é adequado para modelagem híbrida, pois torna o componente contínuo equivalente a um modelo DEVS, que naturalmente integra eventos de entrada e torna a detecção de eventos de estado trivial e sem custo [12].

A ferramenta MECSYCO (*Multi-agent Environment for Complex-System Co-simulation*) é baseada na especificação DEVS. O objetivo da ferramenta é usar o encapsulamento DEVS para integrar cada ferramenta dentro de um *middleware*. Esse acoplamento ocorre de forma paralela, descentralizada e distribuível graças à sua arquitetura modular multi-agente. Assim, ferramentas heterogêneas podem ser co-simuladas na forma de um sistema DEVS [12].

- **Mosaik**

Mosaik é uma estrutura de co-simulação de código aberto flexível, genérico e modular focado em sistemas de energia, baseado na composição automática de modelos de simulação existentes. É escrito em Python, e permite acoplar simuladores heterogêneos e aplicações de controle e a gerar cenários em larga escala. Nesse caso, os simuladores envolvidos na simulação são executados em seu próprio processo. O Mosaik tenta sincronizar esse processo e gerenciar a troca de dados entre eles. Assim, o Mosaik fornece um API para cada simulador se comunicar com essa estrutura, permite a modelagem de cenários de simulação envolvendo simuladores diferentes e programa a execução passo a passo dos diferentes simuladores e gerencia a troca de dados entre eles. Há uma ampla variedade de simuladores que podem ser integrados como parte do Mosaik, como PyPower, PandaPower, PowerFactory ou Opal-RT e MEGAsim. Ademais, um adaptador FMI também pode estar incluído no ambiente, o que permite que ele seja usado como algoritmo mestre para simuladores compatíveis com o padrão FMI [10, 13].

ASTORIA usa Mosaik como middleware para habilitar a integração entre PyPower, como um simulador de fluxo de potência e NS-3, para simular o comportamento dos componentes do sistema SCADA (*Supervisory Control and Data Acquisition*) [14]. O objetivo é avaliar soluções de segurança em Smart Grids, antes de serem implantadas em sistemas reais. O Mosaik intermedia a comunicação entre os dois simuladores através do uso de conexões apropriadas e a troca de mensagens seguindo o formato JSON. O Mosaik inicializa os dois simuladores e coordena sua execução gerenciando a simulação em passos de tempo. O Mosaik também é usado para combinar os componentes dos dois simuladores, a fim de coletar dados da rede elétrica e encaminhá-los para a rede de comunicação.

- **Padrão FMI**

Outro padrão de comunicação, pode-se citar, o padrão FMI [6]. Essa estrutura define uma interface que tem por objetivo simplificar a criação, armazenamento, troca e a (re-) utilização de modelos em sistemas dinâmicos procedentes de diferentes ferramentas de modelagem. Os modelos de interface padronizada são chamados de FMUs (*Functional Mock-up Interface*). Essas unidades quando especificadas para a co-simulação, tem como vantagem permitir a proteção da propriedade intelectual do modelo, explicitando-se somente as soluções [5, 15].

Em [15], são apresentados estudos de acoplamentos de sistemas físicos baseados no padrão FMI, desde aplicações simples até uma aplicação industrial mais extensa. O referido padrão foi apresentado como a interface mais promissora para a co-simulação de sistemas de tempo contínuo e discreto. Além disso, nesse trabalho são tratados problemas de estabilidade em sistemas fracamente acoplados, que podem vir aparecer. Nesses sistemas, as equações do modelo são ocultadas no modelo e são resolvidas a cada passo de integração, ao contrário dos sistemas fortemente acoplados onde as equações do modelo são resolvidas de forma simultânea. Assim, são propostos novos métodos, no âmbito do padrão FMI, que oferecem propriedades aperfeiçoadas de estabilidade para a co-simulação, em comparação com as abordagens clássicas.

Ainda de [15], é proposto um pacote de co-simulação, o pacote PyFMI. Esse é um pacote de alto nível para trabalhar com modelos de interface FMI. Desse modo, conecta o conjunto completo de métodos na especificação FMI em uma abordagem orientada a objetos. O PyFMI também fornece um algoritmo mestre próprio de co-simulação para simulação de sistemas acoplados.

Uma aplicação do uso do pacote PyFMI para sistemas de potência encontra-se em [5]. Nesse trabalho, o potencial do uso de co-simulação foi demonstrado na simulação de um sistema coletor de parque eólico com 50 geradores de 10 [MW] cada. As simulações tornaram-se possíveis no contexto da co-simulação, dada a partição natural do sistema em partes menores e mais simples, o que favorece simultaneamente a precisão dos cálculos e também a redução do tempo computacional para as simulações. Através da co-simulação do parque eólico baseada em paradigmas de computação paralela, a carga computacional foi dividida entre vários processadores. Essa divisão permitiu uma aceleração computacional 20 vezes maior quando comparado com a co-simulação realizada de forma sequencial. É importante ressaltar que os subsistemas foram processados como FMUs para co-simulação.

No estado-da-arte da co-simulação apresentado em [16], especialistas têm obtido preferência para disseminação de seus trabalhos através do *Modelica Conference*. Nesse caso, o padrão FMI é considerado um dos tópicos principais, pelo fato do padrão ser uma das interfaces mais promissoras para a co-simulação. Além disso, a co-simulação é uma abordagem multidisciplinar. De acordo com [16], o FMI é o padrão que tem sido mais

utilizado para algum tipo de co-simulação. No que diz respeito a simulações de sistemas no tempo contínuo, o padrão FMI foi preferido em cerca de 90% dos trabalhos investigados em comparação com a abordagem de co-simulação pelo *Simulink*. Quanto ao tempo discreto, o padrão FMI também foi preferido em cerca de 68% em comparação com os padrões HLA, DEVS e SystemC. Quanto a co-simulação de sistemas híbridos, o padrão também obteve preferência (cerca de 94%) em comparação com o padrão HLA.

A abordagem da co-simulação permite também acoplamentos de sistemas de domínios diferentes. Em [17] é apresentada uma ferramenta de co-simulação, que combina uma ferramenta de simulação de sistemas de transmissão através de equivalentes monofásicos, com programas de estudo de transitórios eletromagnéticos (EMT), para avaliação do comportamento dinâmico de sistemas de transmissão com alta penetração de dispositivos eletrônicos. Uma técnica semelhante pode ser encontrada em [18] com duas usinas de geração solar fotovoltaica modeladas de acordo com o programa EMT em duas instâncias diferentes do Matlab. Esses modelos se comunicam com o sistema de distribuição modelado no OpenDSS [19]. Os resultados mostraram redução do tempo computacional para co-simulação quando comparado à simulação em um único ambiente de simulação EMT. Estudos semelhantes podem ser encontrados em [20].

No que tange o acoplamento entre os sistemas de transmissão e distribuição, a co-simulação tem-se mostrado uma técnica promissora. Nesse caso, os sistemas podem ser simulados em diferentes plataformas computacionais, com suas técnicas de solução mais eficientes, que agora trocam dados entre si de forma cíclica.

Em [8], é apresentada uma forma de co-simulação para estudos dinâmicos, utilizando computação série e paralela, onde um sistema de transmissão, com geradores modelados dinamicamente, é conectado a sistemas de distribuição penetrados por gerações distribuídas baseadas em inversores, também representados dinamicamente. O estudo mostrou que a estabilidade pode ser melhor avaliada por co-simulação, ao invés de avaliar cada sistema separadamente, mostrando a necessidade de uma plataforma de co-simulação para análise de estabilidade. Outros estudos de co-simulação desta natureza podem ser encontrados em [21, 22].

É importante ressaltar que, de forma a gerenciar uma co-simulação, uma metodologia precisa ser estabelecida. Uma delas adota a inclusão de uma linha de transmissão fictícia baseada no modelo de Bergeron entre sistemas. Em [5], esta metodologia é avaliada para o acoplamento entre subsistemas modelados em linguagem Modelica. Este método é independente da frequência elétrica e baseado na teoria das ondas viajantes. A comunicação entre subsistemas ocorre através da troca de apenas termos históricos computados nos nós de interconexão. Contudo, é enfatizado que para o gerenciamento da co-simulação a impedância característica da linha deve ser escolhida de forma apropriada.

Outra possibilidade seria a interação por meio de fontes controladas por corrente e

tensão. Esta abordagem pode ser encontrada em [17, 18, 23]. No contexto da simulação híbrida apresentada nesses trabalhos, tem-se que uma parte do sistema é vista pela outra através de um circuito equivalente. Assim, foi utilizado uma fonte controlada por tensão para representar o sistema no domínio da frequência no lado do sistema EMPT, enquanto uma fonte controlada por corrente representa o sistema no domínio do tempo, no lado do sistema modelado no domínio da frequência. Os valores das duas fontes são atualizadas a cada passo de comunicação.

Dessa forma, diante dos padrões de co-simulação investigados e metodologias de interface explorados, será utilizado o padrão FMI, na tentativa de uniformização entre sistemas modelados no presente trabalho. Os modelos são baseados em linguagem Modelica, sendo esta uma linguagem para modelagem de sistemas complexos, orientada a objetos. Em relação a metodologia de interface, primeiramente o acoplamento entre os sistemas envolvidos na co-simulação será pela implementação de uma linha de transmissão fictícia baseada no modelo de Bergeron, porém para sistemas modelados no domínio fasorial. Logo em seguida, serão apresentadas outras metodologias de co-simulação, onde a interface entre os sistemas dar-se-á através das fontes de corrente e tensão.

3 FUNDAMENTOS DA CO-SIMULAÇÃO

Com a inserção dos recursos energéticos distribuídos, aumenta-se a complexidade dos sistemas elétricos de potência, bem como desafios nos estudos dos impactos dinâmicos dessa integração. Os REDs acoplados a um sistema de potência podem ser bastante heterogêneos, exigindo ferramentas e técnicas de modelagem específicas para a realização de certos estudos. Mesmo assim, torna-se desejável que a interação dessas fontes com o sistema fosse caracterizada em um mesmo ambiente de simulação, embora o procedimento seja realmente desafiador [5].

Uma alternativa que vem sendo explorada na área de simulação de sistemas de potência é o emprego de técnicas de co-simulação. A estratégia básica da co-simulação é dividir sistemas complexos e heterogêneos em subsistemas menores, cujas modelagem e solução podem ser implementadas nas plataformas mais apropriadas. Após essa divisão, deve-se estabelecer canais de comunicação apropriados que permitam a troca de informações entre os subsistemas. Essas informações podem ser trocadas a uma determinada taxa, de forma que, durante um intervalo entre dois pontos de comunicação, cada subsistema seja resolvido independentemente [5, 6].

A co-simulação pode ser utilizada para acoplamento de sistemas de múltiplos domínios, como por exemplo, o acoplamento de um sistema no domínio do tempo, que leva em consideração transitórios eletromagnéticos e um sistema no domínio fasorial [24]. A técnica permite ainda que os passos de integração sejam distintos entre subsistemas, o que é essencial para o acoplamento entre sistemas de múltiplos domínios. A co-simulação também permite que a simulação seja realizada com o emprego de vários processadores. Dessa forma, com o compartilhamento de esforço computacional, ganha-se em tempo de execução ou precisão nas soluções [5, 6].

Nas próximas seções, serão apresentados alguns conceitos sobre co-simulação e sobre o padrão de acoplamento de subsistemas utilizado nesse trabalho.

3.1 PADRÃO FMI (*FUNCTIONAL MOCK-UP INTERFACE*)

Com o suporte de diversas empresas, tais como Daimler, ABB, Siemens, EDF, BMW, Ford, Toyota, Bosch, SAAB, entre outros, e com o intuito de se aprimorar os meios de troca de modelos de simulação entre fornecedores e fabricantes foi desenvolvido o padrão FMI, (do inglês, *Functional Mock-up Interface*) [6].

A FMI define uma interface de programação, também denominada API (do inglês *Application Programming Interface*), que visa a troca de modelos dinâmicos utilizando uma combinação de arquivos XML, binários e código C, distribuídos em um arquivo compactado, denominado FMU (do inglês, *Functional Mock-up Unit*). Essa API é utilizada por um

ambiente de simulação, importador, para criar uma ou mais instâncias de uma FMU e simulá-las tanto de forma isolada, como integrada a outros modelos. Uma FMU pode ser dotada de seu próprio método de solução (normalmente chamado de *solver*) ou permitir que o ambiente de simulação execute a sua integração numérica. Essa interface permite que a chamada de uma unidade FMU seja razoavelmente simples [6].

Outro ponto a salientar é que através da interface FMI, pode-se garantir que uma FMU será compatível com as ferramentas de co-simulação compatíveis com esse modo de solução [5, 6].

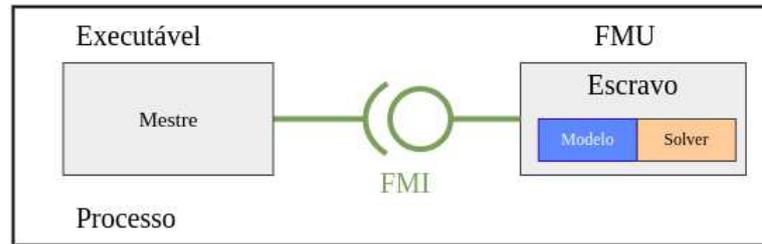
O padrão FMI 2.0 define critérios para estabelecimento de dois tipos de interfaces, a saber, o de co-simulação e o de intercâmbio de modelos (*Model Exchange*) [6], como detalhados a seguir.

- *Model Exchange* (ME): um ambiente de modelagem gera uma representação em código C para o modelo dinâmico de forma que possa ser utilizado por outros ambientes de simulação. Neste modo, um *solver* implementado no ambiente de simulação do sistema hospedeiro deve ser capaz de resolver o sistema importado. Nesse tipo de interface, a modelagem matemática é transparente aos usuários.
- *Co-Simulation* (CS): uma interface padrão para acoplar um modelo de sistema a ferramentas de simulação em um ambiente de co-simulação. A troca de dados entre subsistemas é restrita a pontos de comunicação discretos. No tempo entre os pontos de comunicação, os subsistemas são resolvidos independentemente uns dos outros por seus *solvers* próprios. Os subsistemas de co-simulação são internamente representados por variáveis de estado $\mathbf{x}(t)$. Além disso, são empregadas na interface com outros subsistemas, entradas $\mathbf{u}(t)$ e saídas $\mathbf{y}(t)$.

Esses tipos de interface têm partes em comum, como mostradas nos itens a seguir:

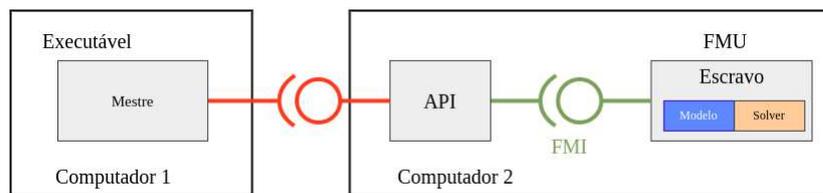
- *FMI Application Programming Interface* (C) - Todas as equações necessárias ou cálculos de acoplamento de ferramenta são avaliados chamando-se funções C padronizadas. A linguagem C é utilizada porque é a linguagem de programação mais portátil hoje, sendo a única linguagem de programação que pode ser utilizada em todos os sistemas de controle embarcados.
- Esquema de descrição FMI (XML) - O esquema define a estrutura e o conteúdo de um arquivo XML gerado por um ambiente de modelagem. Este arquivo XML contém a definição de todas as variáveis da FMU de forma padronizada.
- Uma FMU é distribuída em um arquivo compactado, tipo ZIP. Esse arquivo contém o arquivo de descrição FMI (XML), os binários e bibliotecas necessárias para executar as funções FMI (arquivos *.dll* ou *.so*), as fontes das funções FMI (opcionais) e

Figura 2 – Co-simulação em um único processo computacional



Fonte: Elaborado pelo autor (2022)

Figura 3 – Estrutura de co-simulação distribuída



Fonte: Elaborado pelo autor (2022)

outros dados usados pela FMU (por exemplo, tabelas ou mapas). É possível que uma FMU oculte o código-fonte para proteger o código contido ou para permitir uma importação totalmente automática da FMU em outro ambiente de simulação.

3.1.1 Padrão FMI para co-simulação

No que diz respeito à co-simulação, a FMI é projetada tanto para acoplamento com modelos de subsistema, que foram exportados por seus simuladores juntamente com seus *solvers* e códigos executáveis, como também para acoplamento de ferramentas de simulação. O processo de solução baseia-se no conceito de mestre-escravo, no qual os processos escravos, realizam a solução do subsistema, enquanto o mestre é responsável pela coordenação de troca de informações entre os processos escravos [6]. A Figura 2 ilustra a co-simulação em um único processo.

O padrão também permite o acoplamento de ferramentas, em sistemas distribuídos. Nesse caso, os subsistemas podem ser tratados por diferentes processadores com sistemas operacionais diferentes [6, 25]. A troca de dados e a comunicação entre os subsistemas podem ser feitas utilizando alguma tecnologia de comunicação de rede, como por exemplo, MPI (do inglês, *Message Passing Interface*), TCP/IP (do inglês, *Transmission Control Protocol/Internet Protocol*). A definição desta camada de comunicação não faz parte do padrão FMI. Porém, o padrão permite cenários de co-simulação distribuída. Nesse caso, uma API fornecida pela ferramenta de simulação precisa ser utilizada, para a comunicação entre o mestre e os processos escravos. A Figura 3 ilustra o processo.

3.1.2 *Functional Mock-up Unit* (FMU)

Por meio da interface FMI, um modelo de subsistema pode ser gerado para troca de modelos e/ou co-simulação. Este modelo é então denominado FMU (do inglês *Functional Mock-Up Unit*).

Uma FMU é um arquivo compactado com vários outros arquivos [6]:

- Um arquivo XML contendo a definição de todas as variáveis expostas e outras informações estruturais.
- Um conjunto de funções C com todas as equações necessárias para resolver um sistema ou acessar ferramentas de co-simulação. Este conjunto pode ser fornecido na forma binária, como uma biblioteca compartilhada ou em código-fonte.
- Possíveis recursos adicionais para acoplar um sistema específico a outras ferramentas de simulação.

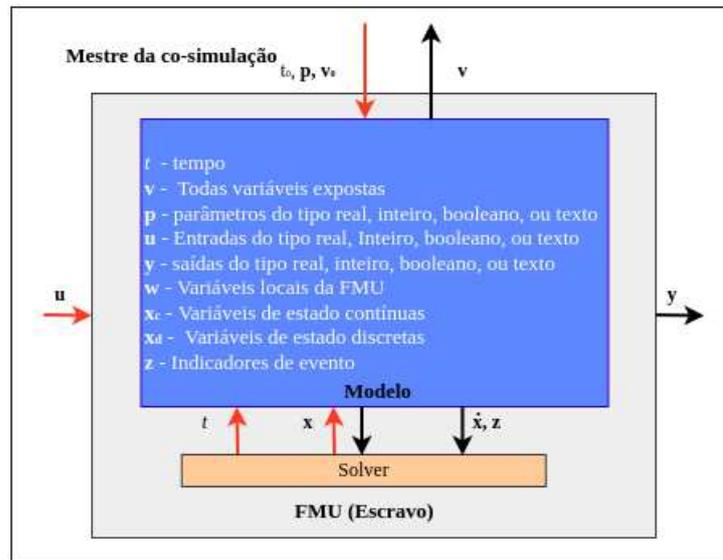
A principal diferença entre os modos de co-simulação e de intercâmbio de modelos é que, no modo CS, o *solver* é integrado à FMU. O ambiente hospedeiro é responsável por inicializar a FMU, definir suas entradas, passo de simulação, bem como a leitura das suas saídas. No caso do modo ME, o *solver* é nativo do ambiente hospedeiro de simulação.

No caso de FMUs para co-simulação, os simuladores escravos são vistos pelo mestre como sistemas puramente amostrados. Tais simuladores podem ser um sistema de dados amostrados real, ou uma equação diferencial híbrida que é integrada entre pontos de comunicação. Nessa interface, eventos internos podem ocorrer, mas no caso de FMUs do tipo co-simulação, estes são tratados internamente, e não são visíveis pelo mestre. Esse conjunto de pontos de comunicação são os instantes de comunicação entre o mestre e os processos escravos [6].

Através de uma representação em diagrama de blocos, os subsistemas podem ser representados por blocos, com variáveis de estado internas $\mathbf{x}(t)$, que podem se conectar à outros subsistemas de um problema acoplado por entradas $\mathbf{u}(t)$ e saídas $\mathbf{y}(t)$. As variáveis de entrada são definidas fora do modelo embutido na FMU, enquanto as variáveis de saída são computadas na própria FMU e projetadas para se conectarem com outros modelos. Assim, as variáveis de saída podem ser usadas no ambiente como valores de entrada para outras FMUs ou simuladores. Nesta estrutura, as conexões físicas entre os subsistemas são representadas por condições matemáticas de acoplamento entre as entradas $\mathbf{u}(t)$ e as saídas $\mathbf{y}(t)$ de todos os subsistemas [6]. A Figura 4 ilustra a estrutura da FMU para co-simulação.

Na estrutura apresentada, além das entradas e saídas já mencionadas, um vetor \mathbf{v} representa todas as variáveis do modelo que são expostas, ou seja, são acessíveis pelo mestre.

Figura 4 – Estrutura de uma FMU do tipo co-simulação



Fonte: Elaborado pelo autor (2022)

Estas variáveis podem ser parâmetros independentes e valores iniciais de outras variáveis, como valores iniciais para variáveis de estados, derivadas de estado ou saídas. O mestre também fornece o instante inicial da co-simulação t_0 , como também \mathbf{p} , que representa o vetor de parâmetros que são constantes durante a co-simulação, sendo fornecido pelo mestre. Os valores iniciais também fornecidos pelo mestre, podem ser representados por \mathbf{v}_0 . O vetor \mathbf{w} representa as variáveis locais, e que não podem ser usadas para conexão entre FMUs. Quanto as variáveis de estado, o vetor $\mathbf{x}_c(t)$ se relaciona às variáveis reais de tempo contínuo representando os estados de tempo contínuo, enquanto \mathbf{x}_d se refere às variáveis de tempo discreto representando os estados discretos, que não são visíveis na interface de uma FMU. Indicadores de eventos \mathbf{z} , podem ser fornecidos pelo modelo ao *solver* porém, são variáveis internas às FMUs [6].

A escolha do método de solução de uma FMU também merece atenção. É necessário observar quais métodos estão disponíveis em uma ferramenta de exportação de FMU e se esses são adequados para a solução do sistema em estudo. No presente desenvolvimento, as FMUs foram geradas pela plataforma de simulação OpenModelica com o *solver* CVode [15, 26, 27].

3.1.3 Solver CVode

O CVode é parte do SUNDIALS (do inglês, *Suite of Nonlinear and Differential/Algebraic Equation Solvers*), um solucionador de problemas de valor inicial em um espaço

real de n dimensões. O problema pode ser escrito na forma em (3.1).

$$\dot{y} = f(t, y) \quad (3.1a)$$

$$y(t_0) = y_0 \quad (3.1b)$$

O método pode ser utilizado para solução de sistemas rígidos e não rígidos. A rigidez é caracterizada pela presença de pelo menos um modo rapidamente amortecido, cuja constante de tempo é pequena em relação a escala de tempo da própria solução. Os métodos usados no CVode são métodos multi-passos de ordem variável e passos variáveis, baseados em fórmulas do tipo em (3.2).

$$\sum_{i=0}^{K_1} \alpha_{n,i} y_{n-i} + h_n \sum_{i=0}^{K_2} \beta_{n,i} \dot{y}_{n-i} = 0 \quad (3.2)$$

Onde y_n são aproximações de $y(t_n)$ e $h_n = t_n - t_{n-1}$ é o tamanho do passo. Duas fórmulas multi-passos podem ser escolhidas. Para problemas não rígidos, CVode inclui as fórmulas de Adams-Moulton, onde $K_1 = 1$ e $K_2 = q$, com $q - 1$ variando de 1 a 12. Para problemas rígidos, CVode inclui as fórmulas de Diferenciação Regressiva (BDF), onde $K_1 = q$ e $K_2 = 0$, com q variando de 1 a 5 [27].

Para qualquer escolha do método de discretização, o sistema não linear resultante é resolvido a cada passo de integração. Para isso, o CVode oferece a opção de iteração funcional, adequada apenas para sistemas não rígidos, e várias versões de iteração de Newton.

3.1.4 Algoritmo mestre

Para estabelecer uma comunicação adequada entre as FMUs, um algoritmo mestre deve ser implementado, pois não há comunicação direta entre os subsistemas. Este algoritmo deve ser responsável por definir as interconexões entre os subsistemas, inicializar variáveis, atribuir parâmetros e sincronizar o fluxo de dados entre os diversos subsistemas que compõem uma simulação. O padrão FMI define as rotinas de interface que permitem a interação entre o algoritmo mestre e as unidades FMU, embora o algoritmo mestre não faça parte do padrão FMI [5, 6].

O algoritmo mestre mais comum é aquele que, para cada instante tc_i de comunicação entre as FMUs, coleta-se as saídas $y(tc_i)$ e atribui-se os valores para as entradas $u(tc_i)$ de todas as unidades FMU associadas a uma simulação. Nesse ponto, cada FMU tem a habilidade de calcular, de forma independente, os valores de suas variáveis de estado e saídas, empregando-se um *solver*, interno ou externo à unidade FMU. Ao fim do cálculo do passo de integração presente, avança-se a linha do tempo com o passo calculado

($tc_{i+1} = tc_i + \Delta t$) e repete-se o procedimento de atribuição de entradas e aquisição de saídas até que a simulação seja terminada [6].

É importante ressaltar que a FMI para a co-simulação suporta desde algoritmos mestre mais comuns, como também mais sofisticados. No entanto, ainda assim, o padrão não define um algoritmo mestre próprio.

Há alguns recursos que permitem a implementação de algoritmos mestre mais sofisticados, porém há uma necessidade de que as FMUs tenham capacidade de suportar tais recursos, como a capacidade de lidar com tamanhos de passo de integração variável; capacidade de repetir um passo de integração rejeitado com tamanho de passo reduzido; e fornecimento de jacobianas [6].

O cálculo de derivadas direcionais pode ser um recurso útil, quando derivadas parciais de FMUs conectadas precisam ser calculadas. Com as derivadas direcionais disponíveis, a Jacobiana pode ser computada analiticamente, e os algoritmos numéricos baseados nessas derivadas podem ser mais eficientes [6]. A plataforma JModelica possui um algoritmo mestre próprio, que permite trabalhar com as FMUs que possuem essa capacidade [28]. Porém, como é um recurso adicional, nem todas as ferramentas possibilitam a exportação de FMUs de co-simulação com essa capacidade.

3.1.5 Ferramentas de suporte à co-simulação

Técnicas de co-simulação vêm se difundindo cada vez mais, tanto no setor industrial como acadêmico, devido às suas vantagens e por se apresentar como única alternativa para o acoplamento de certos modelos. Muitas são as ferramentas que possibilitam a integração de modelos empregando o padrão FMI. Por essa razão, nessa subseção serão apresentadas ferramentas que possibilitaram a implementação da co-simulação na proposta desse trabalho.

Como ferramentas de implementação da co-simulação foram utilizados o PyFMI e OMSimulator. O módulo PyFMI é desenvolvido em Python, com o intuito de possibilitar a interação com as unidades FMU tanto do tipo *model exchange* como do tipo co-simulação. Permite o acesso à interface FMI através de uma linguagem de programação de alto nível. Além disso, o módulo PyFMI implementa um mapeamento dos métodos estabelecidos pelo padrão FMI através de uma abordagem orientada a objetos típica da linguagem de programação Python. Assim, o módulo PyFMI permite o carregamento das FMUs, além da atribuição de parâmetros, condições iniciais de simulação do modelo. O módulo PyFMI faz parte da plataforma de simulação JModelica [5].

O módulo OMSimulator [29] também é disponibilizado na linguagem Python que integra o conjunto de ferramentas disponibilizado pela plataforma OpenModelica. Este módulo permite o acoplamento de FMUs através de conexões comuns, sem atraso, e através

de conexões TLM (do inglês, *Transmission Line Model*), com atrasos introduzidos. Através de conexões comuns, portas de entrada e saída das FMUs podem ser interconectadas. A ferramenta permite simulações de FMUs tanto do tipo *model-exchange* como de co-simulação.

Como ferramenta de geração de FMUs, foi utilizado o módulo OMPython, que faz parte da plataforma de simulação OpenModelica. Este módulo fornece componentes para a criação de um ambiente completo de modelagem, compilação e simulação de modelos baseados na linguagem Modelica. Além disso, OMPython permite exportar FMUs nos modos *model exchange* e/ou co-simulação. Especificamente para o modo de co-simulação, o módulo OMPython permite que FMUs sejam exportadas com um de dois *solvers* distintos: Euler explícito e o CVode [26]. Porém, como já mencionado, o *solver* escolhido foi o CVode. Após a geração da FMU, esta unidade pode ser importada e simulada dentro de um ambiente Python com o auxílio dos módulos PyFMI ou OMSimulator.

3.1.6 Tutorial de exportação e importação de unidades FMU

De forma a ilustrar o processo de exportação e importação de FMUs assim como sua simulação, considera-se o modelo matemático descrito através do conjunto de equações dado em (3.3). Nesse modelo, x corresponde à variável de estado do modelo, enquanto u e y correspondem às suas variáveis de entrada e saída, respectivamente. Adicionalmente, supõe-se também que a condição inicial do problema consiste em um ponto de equilíbrio do sistema onde a taxa de variação da variável de estado x é nula em $t = 0$.

$$\left\{ \begin{array}{l} 0,5 \frac{dx}{dt} = -x + u \\ y = 2x \\ \frac{dx}{dt} \Big|_{t=0} = 0 \end{array} \right. \quad (3.3)$$

Esse modelo matemático pode ser descrito empregando-se a linguagem Modelica como mostrado no Algoritmo 1. Note aqui que as variáveis de estado, entrada e saída são inicialmente declaradas como números reais, ou seja, do tipo `Real`. A identificação das entradas e saídas é feita através das palavras-chave `input` e `output`, respectivamente. Na sequência, são descritas as restrições associadas às condições iniciais do modelo dentro do campo `initial equation`. Nesse caso, a derivada de x , identificada por `der(x)`, deve ser nula. Por fim, as equações que devem ser resolvidas ao longo do tempo são descritas no campo `equation`. Pode-se observar que o modelo matemático pode ser escrito diretamente como concebido em (3.3).

Uma vez desenvolvido o modelo matemático em linguagem Modelica pode-se prosseguir para a exportação da unidade FMU associada. O Algoritmo 2 descreve um

script Python que permite exportar uma unidade FMU referente ao modelo descrito em linguagem Modelica no arquivo `SistemaTeste.mo`. Além disso, a unidade FMU exportada através desse *script* é compatível com o padrão FMI 2.0, do tipo co-simulação e que contém o *solver* CVode. A definição de *solver* é realizada através das opções em `fmiFlags`.

Com a FMU exportada, pode-se realizar sua importação e simulação através dos módulos anteriormente citados, PyFMI ou OMSimulator. O emprego do módulo PyFMI para a simulação do modelo teste será desenvolvido, cujo resultado é resumido no Algoritmo 3.

Primeiramente, importa-se a unidade FMU através do comando `pyfmi.load_fmu` que retorna o objeto `system_cs` que permite interagir com o modelo carregado de acordo com o padrão FMI. Na sequência, declara-se a função `entrada` que gera os valores que devem ser assumidos pela variável u do modelo. Para a presente simulação, a variável de entrada u foi definida como mostrado em (3.4).

$$u(t) = \begin{cases} 1 & \text{se } 0 \leq t < 2 \\ -1 & \text{se } t \geq 2 \end{cases} \quad (3.4)$$

Também são definidos parâmetros associados à simulação tais como passo de integração e duração da simulação através das variáveis `dt` e `tf`. Logo após, são alocados os vetores que armazenarão os resultados para cada uma das variáveis do modelo.

O cálculo da condição inicial depende do valor da entrada u e por isso deve ser definida antes de se executar a inicialização do modelo propriamente dita. A atribuição

Algoritmo 1: Descrição do sistema teste em linguagem Modelica (arquivo `SystemaTeste.mo`)

```

model SistemaTeste
// declaracao da entrada do modelo
input Real u;

// declaracao da saida do modelo
output Real y;

// declaracao da variavel de estado do modelo
Real x;

// restricoes para as condicoes iniciais
initial equation
  der(x) = 0.0;

// equacoes que descrevem o modelo
equation
  0.5*der(x) = - x + u;
  y = 2*x;
end SistemaTeste;

```

Algoritmo 2: Descrição do sistema teste em linguagem Modelica (arquivo `SystemaTeste.mo`)

```

# Importacao de modulos
from OMPython import OMCSessionZMQ, ModelicaSystem
import os

# Abertura do canal de comunicacao com o OpenModelica
omc = OMCSessionZMQ()

# Definicao do arquivo com o modelo a ser exportado
model_basename = "SystemaTeste"
mo_file = "../../{:s}.mo".format(model_basename)
print(omc.sendExpression('cd("./temp/")'))
os.chdir('./temp/')

# Definicao de opcoes para a exportacao da FMU
options = "-d=initialization"
options += "--fmiFlags=s:cvode"

# Carrega o modelo em memoria
mod = ModelicaSystem(mo_file,
model_basename,
["Modelica"],
commandLineOptions=options)

# Conversao do modelo em FMU, tipo co-simulacao
mod.convertMo2Fmu(version="2.0", fmuType="cs")

```

de valor à entrada é realizada através da função `set` do objeto `system_cs`, enquanto a inicialização é realizada através da função `initialize`. Essa última função calcula os valores das variáveis x e y de forma que a derivada de x no tempo seja nula no início da simulação.

Depois do pré-processamento e inicialização descritos nos parágrafos anteriores, pode-se prosseguir para a simulação propriamente dita que consiste num laço de computação realizada para cada instante de tempo armazenado no vetor \mathbf{t} . Em cada iteração desse laço um valor é atribuído para a entrada u e o cálculo do modelo é realizado através da função `do_step`, que avança a simulação em um passo de tempo dt . Ao fim de cada iteração do laço, os resultados são extraídos da FMU, através da função `get`, e armazenados para processamento posterior. Assim, pode-se ver traçado do gráfico mostrado na Figura 5 com os resultados da simulação do modelo proposto.

Nos resultados ilustrados na Figura 5 pode-se observar que a inicialização do modelo foi realizada corretamente dado que a variável de estado x assumiu o valor 1 que, de fato, torna a sua derivada nula no início da simulação. O estado de equilíbrio do sistema só é alterado a partir do instante 2 [s], momento este em que a entrada muda de 1 para -1 . Por fim, observa-se que a variável de estado x tende ao novo valor de -1 , para o qual sua

derivada se anula novamente.

Algoritmo 3: Importação e simulação do modelo teste através do módulo PyFMI

```

import pyfmi
import numpy as np

# Importacao da FMU
system_cs = pyfmi.load_fm("SistemaTeste.fmu")

# Definicao do sinal de entrada aplicado ao modelo
def entrada(time):
    if time >= 2:
        u = -1
    else:
        u = 1
    return u

# Definicao de parametros da co-simulacao
dt = 1e-4 # passo de integracao para a simulacao
tf = 5.0 # duracao da simulacao

# Alocacao de vetores para armazenamento de resultado
t = np.arange(0, tf+dt, dt)
x = np.zeros_like(t)
y = np.zeros_like(t)
u = np.zeros_like(t)

# Calculando a condicao inicial
system_cs.time = 0
system_cs.set("u", entrada(0.0))
system_cs.initialize()

# Laco de tempo para a co-simulacao
for k, ts in enumerate(t):

    # Estabelece o valor de entrada u para o proximo passo
    u[k] = entrada(ts)
    system_cs.set("u", u[k])

    # Calculo de um passo de integracao
    status = system_cs.do_step(ts, dt)

    # Verificando se o passo de integracao foi completado
    if status is not pyfmi.fmi.FMI_OK:
        break

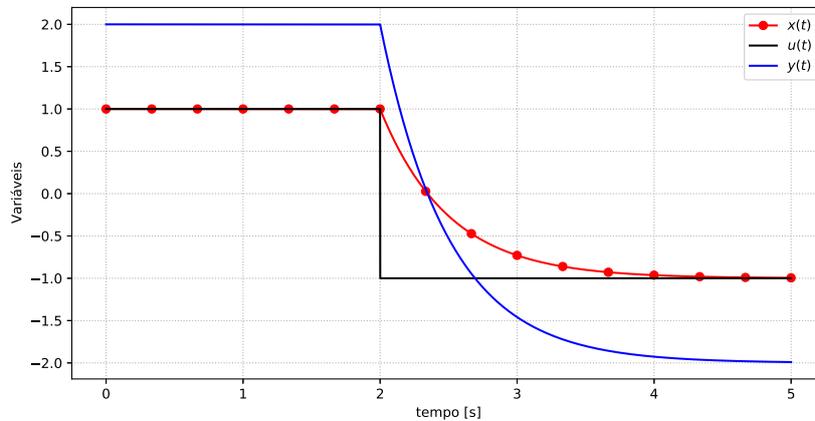
    # Armazenamento dos resultados
    x[k] = system_cs.get("x")[0]
    y[k] = system_cs.get("y")[0]

```

Resolvendo-se (3.3), encontra-se (3.5) como solução para $x(t)$:

$$x(t) = 1 - 2u_{-1}(t - 2) + 2e^{-2(t-2)}u_{-1}(t - 2) \quad (3.5)$$

Figura 5 – Resultados da simulação do sistema teste.



Fonte: Elaborado pelo autor (2022)

onde $u_{-1}(t)$ representa a função degrau. Alguns valores que satisfazem $x(t)$ estão mostrados na Tabela 1. Percebe-se que comportamento mostrado na Figura 5 está coerente com os valores exatos.

Tabela 1 – Valores para $x(t)$

t [s]	$x(t)$
0	1
1	1
2_-	1
2_+	1
∞	-1

No que diz respeito ao OMSimulator, um algoritmo semelhante pode ser implementado. O Algoritmo 4 descreve o processo de importação e simulação da FMU.

Ao se criar o objeto `oms`, é possível interagir com o modelo embutido da FMU. Cria-se então, um modelo através da função `newModel`. Em seguida, um modelo raiz deve ser criado através da função `addSystem`. A partir da criação de um sistema raiz, é possível adicionar as FMUs a serem simuladas no ambiente de simulação. Além disso, pode-se declarar um arquivo de resultados de extensão `.mat` a ser gerado ao final da simulação.

Após a declaração da função `entrada`, os parâmetros de simulação podem ser definidos através de funções específicas, como as funções `setStopTime` para atribuição do tempo final de simulação, e `setFixedStepSize` para atribuição do passo fixo de integração. Em seguida, pode-se atribuir valor à entrada através da função `setReal`, ao passo que, a inicialização é feita através da função `initialize`.

Finalmente, entra-se no laço de integração, de estrutura semelhante ao Algoritmo 3,

devendo-se utilizar apropriadamente as funções para cálculo do passo de integração e extração dos resultados. Os mesmos resultados da Figura 5 foram encontrados, utilizando o módulo OMSimulator.

Algoritmo 4: Importação e simulação do modelo através do módulo OMSimulator

```

import numpy as np
from OMSimulator import OMSimulator
import DyMat

oms = OMSimulator()
oms.newModel("model")
oms.addSystem("model.root", oms.system_wc)
oms.addSubModel("model.root.system", 'SistemaTeste.fmu')

# Arquivo de resultados a ser gerado ao final da simulacao
oms.setResultFile("model", "SistemaTeste.mat")

# Definicao do sinal de entrada aplicado ao modelo
def entrada(time):
    if time >= 2:
        u = -1
    else:
        u = 1
    return u

# Definicao de parametros da co-simulacao
dt = 1e-4 # passo de integracao para a simulacao
tf = 5.0 # duracao da simulacao

# Alocacao de vetores para armazenamento de resultado
t = np.arange(0, tf + dt, dt)
x = np.zeros_like(t)
y = np.zeros_like(t)
u = np.zeros_like(t)

oms.setStopTime("model", tf)
oms.setFixedStepSize("model", dt)
oms.instantiate("model")
oms.setReal('model.root.system.u', entrada(0.0))
oms.initialize("model")

for k, ts in enumerate(t):

    u[k] = entrada(ts)
    oms.setReal('model.root.system.u', u[k])

    # Calcula um passo de integracao:
    status = oms.doStep("model")

    if status is oms.status_error:
        break

    # Salva as variaveis de interesse:
    x[k] = oms.getReal("model.root.system.x")[0]
    y[k] = oms.getReal("model.root.system.y")[0]

results_file = oms.getResultFile("model")[0]

```

3.2 CONCLUSÕES PARCIAIS

Nesse capítulo, foram apresentados conceitos fundamentais de co-simulação. Nesse âmbito, foi citada a interface FMI, a qual garante compatibilidade de acoplamento entre sistemas heterogêneos. Foram definidas as FMUs (*Functional Mock-up Unit*) como subsistemas gerados em conformidade com a interface FMI. Foi especificado que essas unidades são arquivos compactados, contendo o arquivo de descrição FMI, binários e bibliotecas compartilhadas. Foi discutido também que quando a FMU tem a finalidade de co-simulação, esta pode funcionar como uma espécie de caixa preta, explicitando-se entradas e saídas. Além disso, o *solver* é embutido na própria FMU. Dessa forma, foi definido para esse trabalho, que as FMUs serão geradas pela plataforma de simulação OpenModelica, embutidas com o *solver* CVode.

Para o gerenciamento da co-simulação, foi apresentado o procedimento para a implementação do algoritmo mestre, que permite a interconexão, inicialização, atribuição de parâmetros e sincronização de dados entre os subsistemas. Nesse contexto, foi definido o ambiente Python, para a implementação do algoritmo mestre. Ademais, foram definidas como ferramentas de suporte, os módulos em Python, PyFMI e OMSimulator. Para a geração de FMUs, foi definido o módulo OMPython, que faz parte da plataforma OpenModelica.

Por fim, foi apresentado um caso teste ilustrando o ciclo de trabalho para a obtenção de simulações no tempo a partir de unidades FMU em Python. Primeiramente, deve-se modelar matematicamente o sistema de interesse através da linguagem Modelica. Com esse modelo, gera-se a unidade FMU associada por intermédio do módulo OMPython. A FMU gerada pode então ser importada com o emprego do módulo PyFMI ou OMSimulator que implementa a interface de acesso ao modelo e *solver* armazenados dentro da FMU. Através da interface FMI pode-se então realizar as simulações de interesse. Assim, toda fundamentação teórica sobre co-simulação, interface FMI, assim como o tutorial de utilização da FMU apresentados nesse capítulo são de extrema importância para o entendimento das metodologias de co-simulação que serão detalhadas nos próximos capítulos.

4 CO-SIMULAÇÃO ATRAVÉS DE LINHA FICTÍCIA

Como discutido no capítulo anterior, a ideia fundamental da co-simulação reside no fato de um sistema complexo poder ser dividido em subsistemas, os quais são modelados em uma ferramenta especializada. Porém, para o acoplamento entre os subsistemas, uma interface de comunicação precisa ser estabelecida. A proposta apresentada nesse capítulo consiste numa metodologia baseada na implementação de uma linha de transmissão fictícia. Essa técnica permitirá que subsistemas sejam resolvidos de forma independente, sendo necessário explicitar as entradas e saídas que serão trocadas via algoritmo mestre.

A técnica de acoplamento de sistemas por um modelo de linha de transmissão, vem sendo explorada em alguns trabalhos e por algumas ferramentas. A plataforma OpenModelica, através de seu ambiente gráfico OMEdit, permite o acoplamento de modelos conforme o padrão FMI e via modelos baseados em linhas de transmissão ideais, chamado modelos TLM (*Transmission Line Model*) [30].

Ainda da plataforma OpenModelica, em [31], é apresentado o editor gráfico de modelos compostos, que é integrado às estruturas de co-simulação do OpenModelica para dar suporte à edição e simulação de modelos baseado em FMI e TLM. Esses modelos compostos podem ser oriundos de diferentes ferramentas de modelagem de sistemas, como OpenModelica, SimuLink e modelos Adams. Para aplicações de co-simulação de sistemas mecânicos, a estrutura TLM mostrou-se promissora. Porém, tal abordagem necessita de um conhecimento mais aprofundado dos *scripts* inerentes à estrutura TLM do OpenModelica, tornando inadequada a sua utilização para a proposta desse trabalho.

Entretanto, uma segunda investigação foi explorada. Uma proposta de co-simulação bem sucedida foi verificada em [5]. Nessa estratégia, sistemas acoplados foram particionados em subsistemas menores. Dessa forma, cada subsistema foi implementado em linguagem Modelica no OpenModelica e encapsulados em FMUs. Ademais, a troca de dados entre os subsistemas sucedeu-se através da implementação de um modelo de linha fictícia baseada no modelo de Bergeron. Através de uma análise numérica, o modelo proposto mostrou-se bastante adequado para quando a impedância característica da linha é bem ajustada. Como ferramenta de co-simulação para a referida proposta, foi utilizado o módulo PyFMI.

Diante dos resultados bem sucedidos da proposta supracitada, testes de co-simulação baseados no modelo de Bergeron, foram realizados nesse trabalho. Além disso, como já mencionado no capítulo anterior, os módulos PyFMI e OMSimulator foram utilizados para testes de co-simulação.

Portanto, o método da linha fictícia será explorado na seguinte seção, bem como o gerenciamento da co-simulação através da implementação do algoritmo mestre. De antemão, tal metodologia foi bem sucedida para a co-simulação com FMUs, além de proporcionar um ganho em desempenho computacional com a inserção de vários ramais

de distribuição. Do mesmo modo, para o acoplamento entre o OpenModelica e OpenDSS, resultados bem-sucedidos foram encontrados, porém algumas limitações da metodologia foram encontradas e serão discutidas na apresentação dos resultados.

4.1 MODELAGEM DA LINHA FICTÍCIA

Na metodologia proposta nesse capítulo, a interconexão entre subsistemas elétricos é realizada por meio de uma linha de transmissão sem perdas representada pelo modelo de Bergeron. Este modelo é amplamente utilizado em muitos simuladores de transitórios eletromagnéticos no domínio do tempo [32]. No entanto, as simulações de interesse desse trabalho se relacionam com grandezas no domínio fasorial. Nesse contexto, primeiramente será apresentado o modelo de Bergeron e posteriormente, será apresentada como a metodologia será utilizada para a proposta desse capítulo.

Seja então, uma linha de transmissão sem perdas. A propagação da tensão e corrente ao longo dessa linha, estão relacionados com os seus parâmetros distribuídos L e C , dados por (4.1).

$$-\frac{\partial v}{\partial x} = L \frac{\partial i}{\partial t} \quad (4.1a)$$

$$-\frac{\partial i}{\partial x} = C \frac{\partial v}{\partial t} \quad (4.1b)$$

Nesse caso, L e C se referem a indutância e capacitância por unidade de distância dadas em H/m e F/m respectivamente. As soluções gerais de (4.1) são vistas em (4.2):

$$i(x,t) = f_1(x - ct) + f_2(x + ct) \quad (4.2a)$$

$$v(x,t) = Z_c f_1(x - ct) - Z_c f_2(x + ct) \quad (4.2b)$$

onde f_1 e f_2 são funções arbitrárias de $(x - ct)$ e $(x + ct)$ que podem ser completamente definidas através das condições iniciais e de contorno. Fisicamente, $f_1(x - ct)$ consiste em uma onda viajando na direção progressiva enquanto $f_2(x + ct)$ consiste em uma onda viajando na direção regressiva. Os termos Z_c e c referem-se a impedância característica da linha e a velocidade de propagação da onda de tensão e de corrente, respectivamente. Para uma linha ideal, os seus valores são mostrados em (4.3).

$$c = 1/\sqrt{LC} \quad (4.3a)$$

$$Z_c = \sqrt{\frac{L}{C}} \quad (4.3b)$$

A partir de agora, será mostrado como pode ser obtido um circuito equivalente para o modelo de Bergeron, a partir de uma onda se deslocando na direção progressiva. Multiplicando-se (4.2a) por Z_c , e somando-se a (4.2b), obtém-se em (4.4).

$$v(x,t) + Z_c i(x,t) = 2Z_c f_1(x - ct) \quad (4.4)$$

Para o entendimento de (4.4), sejam primeiramente definidos os terminais k e m como receptor e emissor respectivamente. Agora, considere um observador no início do terminal k onde $x = 0$ e $t = 0$. Assim, conclui-se em (4.5).

$$v(0,0) + Z_c i(0,0) = 2Z_c f_1(0) \quad (4.5)$$

Ao chegar ao final dessa linha, no terminal m , esse observador estará a uma posição $x = d$, tendo transcorrido um tempo $t = \tau$. Desse modo, tem-se em (4.6).

$$v(d,\tau) + Z_c i(d,\tau) = 2Z_c f_1(d - c\tau) \quad (4.6)$$

Para uma onda de tensão e corrente se propagando em uma linha de transmissão sem perdas, tem-se que $v(d,\tau)$ é igual a $v(0,0)$ e $i(d,\tau)$ é igual a $i(0,0)$. Essa relação será válida, se $f_1(d - c\tau)$ for igual a $f_1(0)$, ou seja, se forem válidas as relações em (4.7).

$$c = d/\tau = \frac{1}{\sqrt{LC}} \quad (4.7a)$$

$$\tau = d\sqrt{LC} \quad (4.7b)$$

Assim, se as ondas estiverem se propagando na velocidade da luz, o termo $v + Z_c i$ se mantém constante. Nesse âmbito, quando uma onda sai do terminal k em um instante $t - \tau$, chegará de forma integral no terminal m em um instante t . A mesma analogia pode ser feita quando a onda sai do terminal m para o terminal k . Assim, relacionado-se a corrente e tensão entre os terminais k e m da linha de transmissão, são válidas as relações em (4.8).

$$v_k(t - \tau) + Z_c i_k(t - \tau) = v_m(t) - Z_c i_m(t) \quad (4.8a)$$

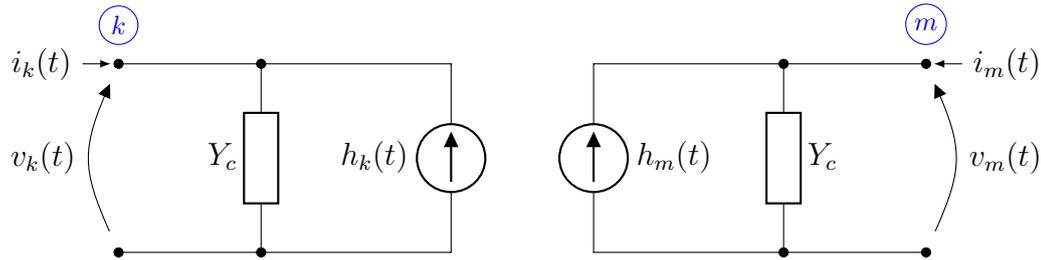
$$v_m(t - \tau) + Z_c i_m(t - \tau) = v_k(t) - Z_c i_k(t) \quad (4.8b)$$

De (4.8), as correntes injetadas $i_k(t)$ e $i_m(t)$ podem ser reescritas em (4.9).

$$i_k(t) = Y_c v_k(t) - h_k(t) \quad (4.9a)$$

$$i_m(t) = Y_c v_m(t) - h_m(t) \quad (4.9b)$$

Figura 6 – Circuito equivalente de uma linha de transmissão ideal no domínio do tempo.



Fonte: Elaborado pelo autor (2022)

O termo Y_c representa a admitância característica da linha, enquanto os termos $h_k(t)$ e $h_m(t)$ representam respectivamente a influência do terminal m no terminal k e vice-versa, e são descritos por (4.10).

$$h_k(t) = Y_c v_m(t - \tau) + i_m(t - \tau) \quad (4.10a)$$

$$h_m(t) = Y_c v_k(t - \tau) + i_k(t - \tau) \quad (4.10b)$$

Como $h_k(t)$ e $h_m(t)$ dependem exclusivamente de valores passados, portanto, são conhecidos como correntes históricas. O conjunto (4.9) e (4.10) pode ser disposto no circuito equivalente na Figura 6. Pode-se perceber que embora os subsistemas não estejam interligados diretamente, há um acoplamento mútuo entre esses, através das fontes de corrente.

O modelo da Figura 6 é conhecido como modelo de Bergeron. Como consequência, no modelo descrito, o tempo de propagação τ da linha de transmissão naturalmente introduz um atraso de tempo entre os terminais k e m , permitindo resolver sistemas conectados a cada terminal independentemente um do outro, desde que nenhuma outra interconexão entre os dois sistemas seja observada. Do ponto de vista da co-simulação, esta característica já seria suficiente para desacoplar completamente grandes porções de um sistema heterogêneo que poderia, por sua vez, ser resolvido de forma independente.

Em simulações de transitórios eletromagnéticos, ou simplesmente simulações EMT, essa característica já é bem aproveitada, possibilitando até simuladores em tempo real, como os comercializados pela RTDS e OpalRT [33, 34]. Nesses simuladores em tempo real, linhas de transmissão já existentes podem ser modeladas de acordo com o modelo de Bergeron, desacoplando, assim, partes dos sistemas e melhorando consideravelmente a eficiência computacional das simulações, que agora podem melhor empregar arquiteturas computacionais de processamento paralelo.

Entretanto, as simulações de interesse no presente desenvolvimento são de domínio

fasorial. Nesse modo, as linhas de transmissão com retardo de tempo não existem mais, uma vez que os sistemas de transmissão são agora considerados operando muito próximos de sua condição de regime permanente. Sob tais circunstâncias, as linhas de transmissão são representadas por circuitos a parâmetros concentrados na frequência industrial e utilizando o modelo π , tornando todos os subsistemas totalmente interconectados novamente.

Nesse contexto, é proposto um ambiente de co-simulação que se baseia em linhas de transmissão fictícias interligando subsistemas baseados em domínio fasorial. Essas linhas reintroduzem atrasos de transporte entre os subsistemas, permitindo computá-los independentemente. Apesar da possibilidade de ocorrência de oscilações numéricas, principalmente na vizinhança de interconexão, também é esperado que logo após distúrbios de alta frequência, a linha de transmissão fictícia acabe operando em estado estacionário e excitada por quantidades complexas constantes. Nessa situação, o efeito da linha de transmissão fictícia praticamente desaparece, restando apenas as partes pré-existentes do sistema.

Para a modelagem da linha fictícia, as relações (4.9) e (4.10) podem ser colocadas na forma complexa, como mostrada em (4.11).

$$\bar{I}_k(t) = \bar{Y}_c \bar{V}_k(t) - \bar{H}_k(t) \quad (4.11a)$$

$$\bar{I}_m(t) = \bar{Y}_c \bar{V}_m(t) - \bar{H}_m(t) \quad (4.11b)$$

$$\bar{H}_k(t) = \bar{Y}_c \bar{V}_m(t - \tau) + \bar{I}_m(t - \tau) \quad (4.11c)$$

$$\bar{H}_m(t) = \bar{Y}_c \bar{V}_k(t - \tau) + \bar{I}_k(t - \tau) \quad (4.11d)$$

Para a metodologia proposta, todos os atrasos de propagação fictícios são iguados ao intervalo de tempo fixo (Δt) utilizado nas simulações, deixando sua admitância característica \bar{Y}_c indefinida.

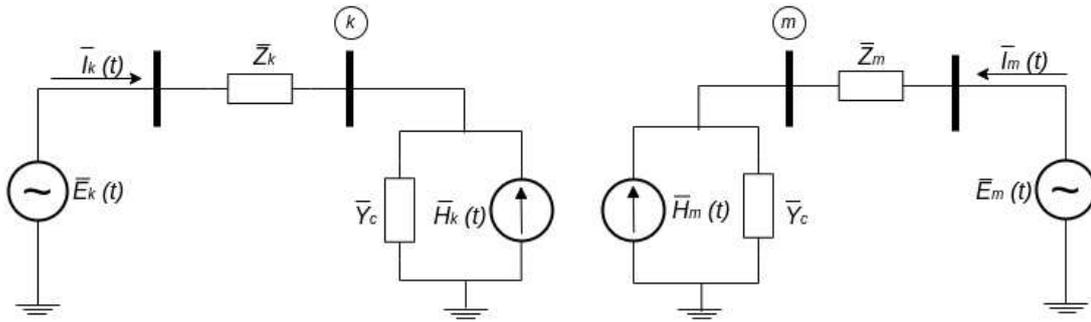
Algumas análises foram realizadas, com objetivo de estimar a admitância característica \bar{Y}_c adequada para a co-simulação. Portanto, seja o sistema radial particionado em dois subsistemas, apresentado na Figura 7.

Nesse sistema, $\bar{E}_k(t)$ e $\bar{E}_m(t)$ representam fontes equivalentes de Thévenin, enquanto \bar{Z}_k e \bar{Z}_m representam as impedâncias de Thévenin vistas a partir do nó de interconexão. Esses equivalentes podem representar os sistemas de transmissão e distribuição da co-simulação. Para os desenvolvimentos subsequentes, análises no domínio discreto serão realizadas.

Em processamento digital de sinais, os sinais são representados como sequências. Essas sequências podem ser amostradas no tempo como mostrado em (4.12):

$$x[n] = x_c(nT) \quad (4.12)$$

Figura 7 – Subsistemas da co-simulação



Fonte: Elaborado pelo autor (2022)

No qual $x[n]$ representa o sinal amostrado no tempo, sendo x_c um sinal no tempo contínuo, n um número inteiro e T representa o período de amostragem [35]. Dessa forma, as relações obtidas no domínio do tempo para o modelo de Bergeron, serão amostradas, para análises no domínio discreto.

Assim, a partir das relações de (4.11), os termos \bar{E}_k e \bar{E}_m podem ser descritos em (4.13).

$$\bar{E}_k = \bar{Z}_k \bar{I}_k[n] + \bar{V}_k[n] \quad (4.13a)$$

$$\bar{E}_m = \bar{Z}_m \bar{I}_m[n] + \bar{V}_m[n] \quad (4.13b)$$

Os termos $\bar{I}_k[n]$ e $\bar{I}_m[n]$, $\bar{V}_k[n]$ e $\bar{V}_m[n]$ representam as sequências de corrente e tensões atuais, enquanto os termos $\bar{I}_k[n-1]$ e $\bar{I}_m[n-1]$ referem-se às sequências de correntes atrasadas de uma amostra. As injeções nos terminais k e m são obtidas em (4.14).

$$\bar{I}_k[n] = \bar{Y}_c \bar{V}_k[n] - \bar{H}_k[n] \quad (4.14a)$$

$$\bar{I}_m[n] = \bar{Y}_c \bar{V}_m[n] - \bar{H}_m[n] \quad (4.14b)$$

Os termos históricos $\bar{H}_k[n]$ e $\bar{H}_m[n]$ podem ser descritos em (4.15):

$$\bar{H}_k[n] = \bar{Y}_c \bar{V}_m[n-1] + \bar{I}_m[n-1] \quad (4.15a)$$

$$\bar{H}_m[n] = \bar{Y}_c \bar{V}_k[n-1] + \bar{I}_k[n-1] \quad (4.15b)$$

Os conjuntos (4.13) a (4.15) formam um sistema de equações, no qual serão assumidas como variáveis, os termos $\bar{I}_k[n]$ e $\bar{I}_m[n]$. Para isso, as tensões terminais $\bar{V}_k[n]$ e

$\bar{V}_m[n]$ podem ser obtidas de (4.13), como concebido em (4.16).

$$\bar{V}_k[n] = \bar{E}_k - \bar{Z}_k \bar{I}_k[n] \quad (4.16a)$$

$$\bar{V}_m[n] = \bar{E}_m - \bar{Z}_m \bar{I}_m[n] \quad (4.16b)$$

É importante ressaltar que as tensões \bar{E}_k e \bar{E}_m são consideradas constantes. Assim, são válidas também as relações de (4.17):

$$\bar{V}_k[n-1] = \bar{E}_k - \bar{Z}_k \bar{I}_k[n-1] \quad (4.17a)$$

$$\bar{V}_m[n-1] = \bar{E}_m - \bar{Z}_m \bar{I}_m[n-1] \quad (4.17b)$$

Substituindo-se (4.17) em (4.15), obtém-se em (4.18):

$$\bar{H}_k[n] = \frac{\bar{E}_m + (\bar{Z}_c - \bar{Z}_m) \bar{I}_m[n-1]}{\bar{Z}_c} \quad (4.18a)$$

$$\bar{H}_m[n] = \frac{\bar{E}_k + (\bar{Z}_c - \bar{Z}_k) \bar{I}_k[n-1]}{\bar{Z}_c} \quad (4.18b)$$

Por fim, substituindo-se (4.17) e (4.18) em (4.14) pode-se obter como solução, os termos mostrados em (4.19)

$$\bar{I}_k[n] = \frac{\bar{E}_k - \bar{E}_m - \bar{Z}_c \bar{I}_m[n-1] + \bar{Z}_m \bar{I}_m[n-1]}{\bar{Z}_c + \bar{Z}_k} \quad (4.19a)$$

$$\bar{I}_m[n] = \frac{-\bar{E}_k + \bar{E}_m - \bar{Z}_c \bar{I}_k[n-1] + \bar{Z}_k \bar{I}_k[n-1]}{\bar{Z}_c + \bar{Z}_m} \quad (4.19b)$$

Como se pode observar, cada variável é função das correntes históricas. As correntes $\bar{I}_k[n]$, $\bar{I}_m[n]$ podem ser escritas na forma de espaço de estados discretos, a partir de uma fórmula recursiva, descrita no Apêndice F. Assumindo que $\mathbf{x}[n]$ é função somente das amostras passadas, obtém-se em (4.20).

$$\mathbf{x}[n] = \mathbf{A}\mathbf{x}[n-1] + \mathbf{B}\mathbf{u}[n] \quad (4.20)$$

onde,

$$\mathbf{x} = \begin{bmatrix} \bar{I}_k \\ \bar{I}_m \end{bmatrix} \quad (4.21a)$$

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{\bar{Z}_m - \bar{Z}_c}{\bar{Z}_k + \bar{Z}_c} \\ \frac{\bar{Z}_k - \bar{Z}_c}{\bar{Z}_m + \bar{Z}_c} & 0 \end{bmatrix} \quad (4.21b)$$

$$\mathbf{u} = \begin{bmatrix} \bar{E}_k \\ \bar{E}_m \end{bmatrix} \quad (4.21c)$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{\bar{Z}_k + \bar{Z}_c} & -\frac{1}{\bar{Z}_k + \bar{Z}_c} \\ -\frac{1}{\bar{Z}_m + \bar{Z}_c} & \frac{1}{\bar{Z}_m + \bar{Z}_c} \end{bmatrix} \quad (4.21d)$$

A estabilidade de (4.20) é determinada pela magnitude dos autovalores $\bar{\lambda}_i$ da matriz de transição \mathbf{A} . Se os autovalores $\bar{\lambda}_i$ satisfazem a condição $|\bar{\lambda}_i| < 1$, o sistema é assintoticamente estável. Por outro lado, é instável se $|\bar{\lambda}_i| > 1$. Se $|\bar{\lambda}_i| = 1$, o sistema pode ser estável ou instável, dependendo da estrutura de \mathbf{A} .

Dessa forma, retornando-se ao sistema definido em (F.4), os autovalores de \mathbf{A} são descritos em (4.22).

$$\bar{\lambda}_{1,2} = \pm \sqrt{\frac{(\bar{Z}_c - \bar{Z}_k)(\bar{Z}_c - \bar{Z}_m)}{(\bar{Z}_c + \bar{Z}_k)(\bar{Z}_c + \bar{Z}_m)}} \quad (4.22)$$

Analisando-se (4.22), pode-se inferir que os autovalores correspondem a média geométrica entre os coeficientes de reflexão $\bar{\Gamma}_k$ e $\bar{\Gamma}_m$, obtidos separadamente para cada terminal k e m . O Apêndice E mostra como o coeficiente de reflexão pode ser obtido. Assim, sejam os termos descritos em (4.23).

$$\bar{\Gamma}_k = \frac{\bar{Z}_k - \bar{Z}_c}{\bar{Z}_k + \bar{Z}_c} \quad (4.23a)$$

$$\bar{\Gamma}_m = \frac{\bar{Z}_m - \bar{Z}_c}{\bar{Z}_m + \bar{Z}_c} \quad (4.23b)$$

Os autovalores mostrados em (4.22) podem ser reescritos em (4.24).

$$\bar{\lambda}_{1,2} = \pm \sqrt{\bar{\Gamma}_k \bar{\Gamma}_m} \quad (4.24)$$

Desse modo, para que o sistema descrito em (4.20) seja estável, tem-se que

$\left| \sqrt{\bar{\Gamma}_k \bar{\Gamma}_m} \right| < 1$. Sejam, então, as impedâncias \bar{Z}_c , \bar{Z}_k , \bar{Z}_m descritas por (4.25).

$$\bar{Z}_c = Z_c(\cos \theta_c + j \operatorname{sen} \theta_c) \quad (4.25a)$$

$$\bar{Z}_k = Z_k(\cos \theta_k + j \operatorname{sen} \theta_k) \quad (4.25b)$$

$$\bar{Z}_m = Z_m(\cos \theta_m + j \operatorname{sen} \theta_m) \quad (4.25c)$$

em que,

$$\theta_c = \tan^{-1} \left(\frac{\operatorname{Im} \{ \bar{Z}_c \}}{\operatorname{Re} \{ \bar{Z}_c \}} \right) \quad (4.26a)$$

$$\theta_k = \tan^{-1} \left(\frac{\operatorname{Im} \{ \bar{Z}_k \}}{\operatorname{Re} \{ \bar{Z}_k \}} \right) \quad (4.26b)$$

$$\theta_m = \tan^{-1} \left(\frac{\operatorname{Im} \{ \bar{Z}_m \}}{\operatorname{Re} \{ \bar{Z}_m \}} \right) \quad (4.26c)$$

Substituindo-se (4.25) em (4.22), uma função $f(Z_c, \theta_c)$ pode ser obtida, descrita em (4.27).

$$f(Z_c, \theta_c) = \sqrt{\frac{\sqrt{Z_c^2 - 2Z_c Z_k \cos(\theta_c - \theta_k) + Z_k^2} \sqrt{Z_c^2 - 2Z_c Z_m \cos(\theta_c - \theta_m) + Z_m^2}}{\sqrt{Z_c^2 + 2Z_c Z_k \cos(\theta_c - \theta_k) + Z_k^2} \sqrt{Z_c^2 + 2Z_c Z_m \cos(\theta_c - \theta_m) + Z_m^2}}} \quad (4.27)$$

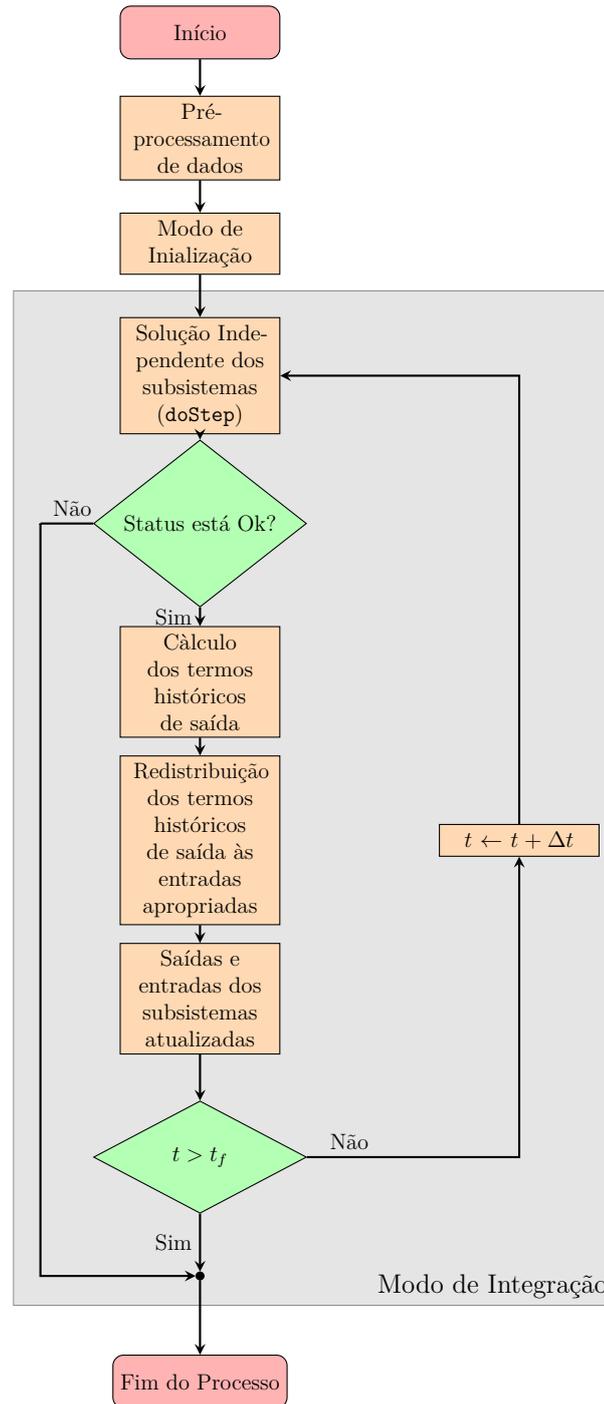
Para se garantir a estabilidade da co-simulação, deve-se escolher apropriadamente os valores de Z_c e θ_c , de forma que $|f|$ seja menor que 1. A seleção de Z_c e θ_c será ilustrada através de exemplos.

4.1.1 Procedimento de co-simulação

Dada a apresentação da metodologia da linha fictícia, um fluxograma descrevendo todo o procedimento é mostrado na Figura 8. Cada etapa será descrita a seguir.

Pré-processamento de dados No início, os sistemas são preparados para serem importados para o ambiente de co-simulação. Para as atuais FMUs baseadas em co-simulação, cada sistema foi modelado por meio da linguagem Modelica no ambiente gráfico do OMEEdit. As linhas fictícias são acopladas ao modelo dinâmico no nó de interconexão, definindo suas entradas e saídas associadas. As FMUs foram geradas utilizando-se o módulo OMPython.

Figura 8 – Fluxograma do processo de co-simulação



Fonte: Elaborado pelo autor (2022)

Modo de inicialização Neste modo, um tempo de simulação inicial t_0 é definido e as condições iniciais para as variáveis de estado internas da FMU são calculadas. Nesta etapa, um conjunto de equações algébricas, com todas as derivadas definidas como zero, é resolvido para se obter um estado estacionário para o sistema. Todas as FMUs envolvidas precisam ser inicializadas corretamente. Durante esta fase, o padrão FMI também permite a atribuição de valores iniciais às variáveis de estado e entradas

internas da FMU.

Mais especificamente para o presente esquema de co-simulação, é necessário fornecer às FMUs suas entradas, para que sejam definidos os valores iniciais das correntes históricas.

As etapas executadas para a inicialização do sistema serão as seguintes:

- Estimativa das impedâncias de Thévenin vistas a partir dos nós de interconexão de cada subsistema: Estimativas razoáveis podem ser obtidas por um cálculo de curto-circuito realizado nos nós de interconexão. A Seção C.1 descreve como o procedimento pode ser feito. Assim, as impedâncias \bar{Z}_k e \bar{Z}_m podem ser obtidas.
- Cada linha de transmissão fictícia é definida por meio da impedância característica \bar{Z}_c , analisando-se (4.27).
- A partir de um cálculo de fluxo de potência, e levando em consideração que $\bar{V}_m = \bar{V}_k$ e $\bar{I}_m = -\bar{I}_k$, as correntes históricas iniciais podem ser obtidas através de (4.28).

$$\bar{H}_{k0} = \bar{Y}_c \bar{V}_{k0} + \bar{I}_{k0} \quad (4.28a)$$

$$\bar{H}_{m0} = \bar{Y}_c \bar{V}_{k0} - \bar{I}_{k0} \quad (4.28b)$$

Solução independente dos sistemas Nesta etapa, entra-se no modo de integração, onde para cada FMU, os valores para todas as variáveis de tempo contínuas e discretas reais são calculados numericamente por meio de métodos apropriados para sistemas de equações algébrico-diferenciais. As etapas a seguir fornecem uma visão geral do modo de integração, de acordo com [6]:

1. Em um dado instante t_i , um passo de integração Δt é calculado através da função `doStep`, que invoca o *solver* embutido nas FMUs;
2. O status do cálculo realizado em (1) é verificado para cada FMU;
3. Se o cálculo do passo foi bem sucedido, as saídas são coletadas (correntes de saída históricas, por exemplo);
4. As correntes históricas são enviadas ao algoritmo mestre, que as redistribui aos subsistemas associados;
5. Os valores de entrada das FMUs agora são atualizados;
6. O tempo avança $t \leftarrow t + \Delta t$ com um tamanho de passo de comunicação fixo;
7. Itens (1) a (6) são repetidos enquanto $t < t_{final}$.

De forma a contribuir para o entendimento do processo de co-simulação com FMUs, o Algoritmo 5 apresenta na forma de uma mistura de pseudocódigo com linguagem Python, o processo de co-simulação com FMUs. É importante ressaltar que nas FMUs geradas, as únicas entradas são as correntes históricas do modelo de Bergeron, que são calculadas no subsistema conectado à outra ponta da linha de transmissão fictícia. Assim, os termos h_{in-re} e h_{in-im} referem-se as partes real e imaginária das entradas das FMUs, enquanto h_{out-re} e h_{out-im} referem-se às saídas.

Algoritmo 5: Co-simulação com duas FMUs

```

entrada :  $\bar{H}_{k0}, \bar{H}_{m0}, \bar{Z}_c$ 
saída : Grandezas de interesse

# Carregamento das FMUs
fmu_TS ← load_fmu("sistema1.fmu")
fmu_DS ← load_fmu("sistema2.fmu")

# Configurações de simulação
Atribuir valores para  $t_f, \Delta t$ 

# Inicialização da FMU do sistema de transmissão
fmu_TS.instantiate()
fmu_TS.setup_experiment()
fmu_TS.enter_initialization_mode()
fmu_TS.set("hin_re", Real( $\bar{H}_{k0}$ ))
fmu_TS.set("hin_im", Imag( $\bar{H}_{k0}$ ))
fmu_TS.set("Zc.re", Real( $\bar{Z}_c$ ))
fmu_TS.set("Zc.im", Imag( $\bar{Z}_c$ ))
fmu_TS.time = 0
fmu_TS.exit_initialization_mode()

# Inicialização da FMU do sistema de distribuição
(inicialização anterior repetida para fmu_DS)

# Modo de integração
enquanto  $t < t_{final}$  faça
  # Cálculos independentes para cada subsistema
  status1 ← fmu_TS.do_step(ts,  $\Delta t$ )
  status2 ← fmu_DS.do_step(ts,  $\Delta t$ )

  # Verifica se os cálculos do passo de integração foram bem-sucedidos
  se  $status1$  ou  $status2 = pyfmi.fmi.FMI\_ERROR$  então
    └ break

  # Armazenamento das variáveis de interesse
   $H_{m-re}$  ← fmu_TS.get("hout_re")
   $H_{m-im}$  ← fmu_TS.get("hout_im")
   $H_{k-re}$  ← fmu_DS.get("hout_re")
   $H_{k-im}$  ← fmu_DS.get("hout_im")

  # Atualização das entradas das FMUs
  fmu_TS.set("hin_re",  $H_{k-re}$ )
  fmu_TS.set("hin_im",  $H_{k-im}$ )
  fmu_DS.set("hin_re",  $H_{m-re}$ )
  fmu_DS.set("hin_im",  $H_{m-im}$ )

  # Incremento do tempo de simulação
   $t \leftarrow t + \Delta t$ 

```

4.2 RESULTADOS DE CO-SIMULAÇÃO COM FMUs

Na sequência serão apresentados resultados obtidos para a co-simulação com FMUs, assim como uma medida de proximidade entre curvas obtidas através de co-simulação e através de simulações com sistemas monolíticos completos, ou seja, sem partição alguma.

4.2.1 Análises de erro

Um critério que foi utilizado para a análise do erro da co-simulação é o Erro Integral Absoluto Normalizado (NIAE). Este erro é uma variação do erro absoluto integral (IAE), normalizado pela integral do sinal de referência. Os critérios NIAE podem ser descritos matematicamente por (4.29). De acordo com [25], quando $NIAE \geq 0,95$ o modelo, representado por x pode ser considerado uma aproximação adequada para x^* .

$$NIAE = 1 - \frac{\int_0^t |x^* - x| dt}{\int_0^t x^* dt} \quad (4.29)$$

4.2.2 Co-simulação entre sistema transmissão e ramais de 38 barras

O primeiro teste realizado foi a co-simulação com os sistemas de transmissão de 11 barras [36] e o sistema de distribuição de 38 barras [37]. As características de ambos os sistemas podem ser consultados no Seção A.1 e Subseção A.2.1.

Quanto às simulações, um curto-circuito trifásico é aplicado no barramento 8 do sistema de 11 barras e eliminado 100 [ms] depois, quando a linha em falha é aberta.

Quanto ao sistema de distribuição, este foi considerado no modo estático, sem nenhuma dinâmica. Além disso, este sistema é alimentado por um grande sistema, modelado como uma barra infinita, conectada à barra 1. Nessas condições, o sistema demanda 187 MW e 123 Mvar.

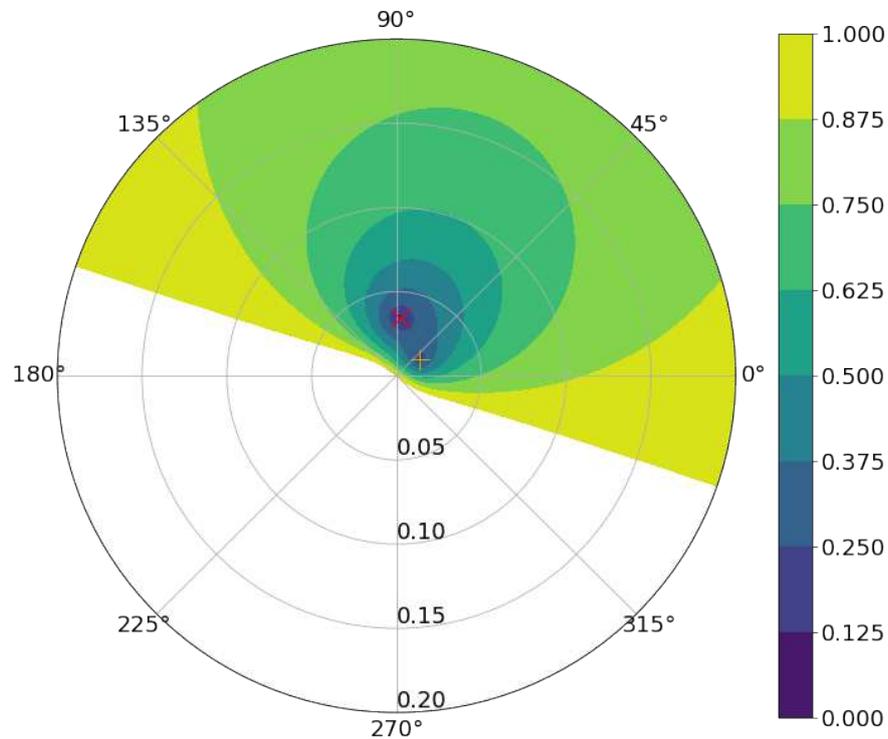
Para o acoplamento do sistema de distribuição de 38 barras ao sistema de transmissão de 11 barras, o sistema de distribuição, foi considerado parte da carga conectada à barra 7 do sistema de transmissão que totaliza 936 MW.

Em relação a inicialização, a Seção B.1 apresenta os valores iniciais obtidos da solução do sistema completo.

Quanto às linhas de transmissão fictícias, necessárias para estabelecer o esquema de co-simulação proposto, as estimativas das impedâncias de Thévenin vistas dos sistemas de transmissão e distribuição são necessárias e apresentadas na Tabela 2. Nesta tabela, a linha está relacionada às estimativas de impedância para o sistema de transmissão de 11 barras e sistemas de distribuição de 38 barras.

Assim, a expressão (4.27) pode ser analisada, substituindo-se os parâmetros \bar{Z}_k e \bar{Z}_m dados na Tabela 2.

Figura 9 – Módulo do autovalor para co-simulação entre sistemas de transmissão e 38 barras



Fonte: Elaborado pelo autor (2022)

Para a convergência, tem-se que (4.27) deve ser menor que 1. Assim, f pode ser analisada através de um gráfico polar, como mostrada na Figura 9. Os contornos referem-se ao módulo da função f .

Pelo gráfico polar, pode se observar que, a partir do limite da estabilidade, quando $|f| = 1$, a medida que $|f|$ reduz, os valores \bar{Z}_c que satisfazem f são mais restritos. Além disso, quando $|f|$ se aproxima de zero, os valores de \bar{Z}_c tendem a se aproximar das impedâncias \bar{Z}_k e \bar{Z}_m , representados pelos marcadores \times e $+$ respectivamente.

Para as análises subsequentes, duas análises serão realizadas: A primeira será para quando \bar{Z}_c é um valor próximo ao de \bar{Z}_k , enquanto a segunda, para quando \bar{Z}_c é próximo a de \bar{Z}_m . Dessa forma, a Tabela 3 apresenta os dois casos que serão analisados.

Tabela 2 – Estimativa das impedâncias de Thévenin - Sistemas de transmissão e 38 barras

Transmissão [pu]	Distribuição [pu]
$0,00176 + j0,034491$	$0,0133 + j0,0097$

Tabela 3 – Estimativas da impedância característica - Sistemas de transmissão e 38 barras

Casos	\bar{Z}_c [pu]	f
I	$0,035057/86,73^\circ$	0,068
II	$0,019/31,37^\circ$	0,085

Fonte: Elaborado pelo autor (2022)

Pela escolha dos valores de \bar{Z}_c percebe-se que o módulo de (4.27) é menor que 10% do limite da estabilidade, já sendo suficiente para a co-simulação.

Os testes para as estratégias de co-simulação propostas foram realizados utilizando os módulos PyFMI e OMSimulator. O passo de integração adotado para as simulações foi de 1 [ms].

As grandezas dinâmicas apresentadas na sequência comparam os resultados obtidos pela co-simulação utilizando o módulo PyFMI do sistema de 11 barras e um único alimentador de 38 barras com os resultados obtidos com um único sistema modelado no OpenModelica. A Figura 10(a) mostra o ângulo relativo de G_1 em relação a G_4 , enquanto a Figura 10(b) mostram a potência fluindo através da barra de referência do sistema de transmissão. E por último, as Figuras 10(c) e 10(d) mostram a tensão fluindo através da barra de interface para os casos I e II. As mesmas grandezas mostradas anteriormente foram também medidas utilizando o módulo OMSimulator, sendo mostradas na Figura 11.

Observa-se que, os dois casos de co-simulação realizadas com ambos os módulos, PyFMI e OMSimulator, apresentaram resultados muito semelhantes aos obtidos com o sistema monolítico ou seja, sem partições.

Apenas um pequeno atraso entre as formas de onda pode ser observado, devido ao passo de comunicação entre os terminais da linha de transmissão fictícia.

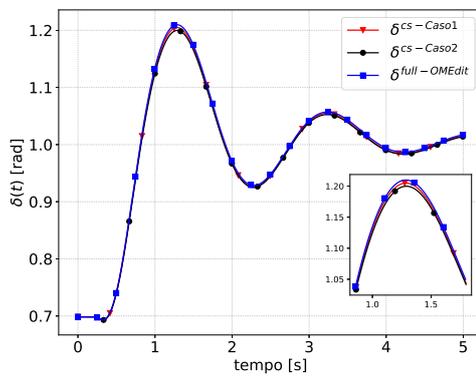
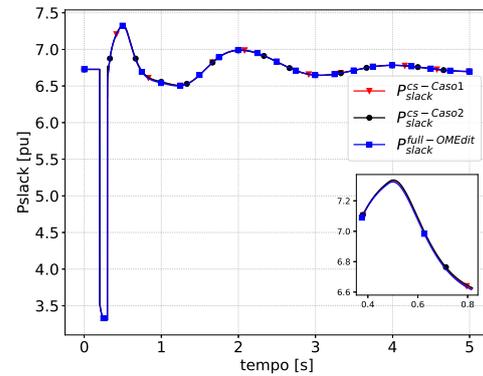
A análise de erro através do critério NIAE para as curvas relacionadas ao ângulo relativo δ_{14} , tensão da interface (V_{bus7}) e potência ativa fluindo pela interface (P_{slack}) também foram realizadas. As Tabelas 4 e 5 apresentam os valores obtidos para cada grandeza extraída. Os resultados mostraram que a abordagem de co-simulação proposta pode ser considerada adequada com base no fato de que todos os valores NIAE excederam 99,5%.

Tabela 4 – Valores NIAE - PyFMI (Valores em [%])

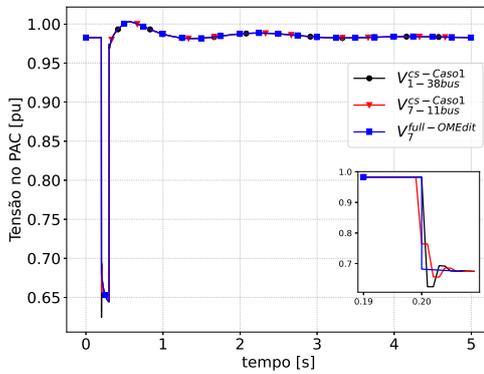
Casos	δ_{14}	V_{bus7}	P_{slack}
I	99,745	99,947	99,898
II	99,570	99,961	99,878

Fonte: Elaborado pelo autor (2022)

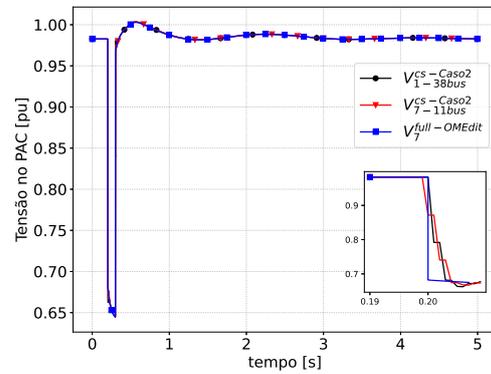
Figura 10 – Quantidades medidas durante a falta - PyFMI - Sistema de transmissão e 38 barras

(a) Ângulo relativo entre G_1 e G_4 

(b) Potência na barra de referência



(c) Tensão na barra de interface - Caso 1



(d) Tensão na barra de interface - Caso 2

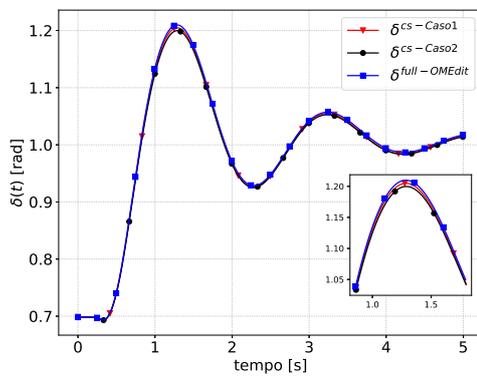
Fonte: Elaborado pelo autor (2022)

Tabela 5 – Valores NIAE - OMSimulator (Valores em [%])

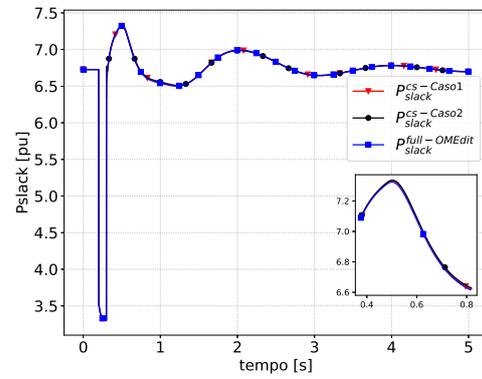
Casos	δ_{14}	V_{bus7}	P_{slack}
I	99,744	99,947	99,898
II	99,570	99,961	99,878

Fonte: Elaborado pelo autor (2022)

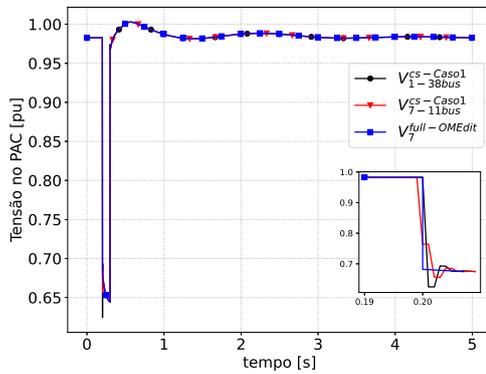
Figura 11 – Quantidades medidas durante a falta - OMSimulator - Sistema de transmissão e 38 barras



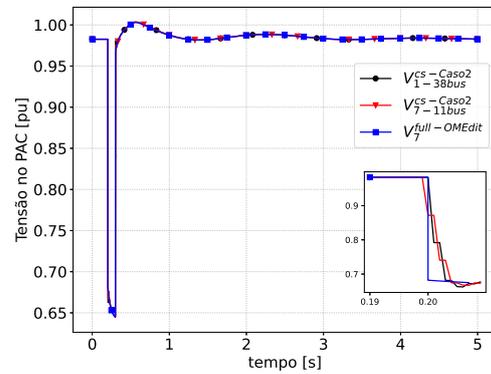
(a) Ângulo relativo entre G_1 e G_4



(b) Potência na barra de referência



(c) Tensão na barra de interface - Caso 1



(d) Tensão na barra de interface - Caso 2

Fonte: Elaborado pelo autor (2022)

4.2.3 Múltiplos Alimentadores

Testes de co-simulação com múltiplos alimentadores também foram realizados. De dois a cinco alimentadores de 38 barras foram conectados ao sistema de transmissão de 11 barras. Cada alimentador é conectado ao sistema de transmissão com sua própria linha de transmissão fictícia. E a fim de manter a carga total, a cada inserção de ramal de distribuição, foi descontada 187 MW da carga total, o que é equivalente a potência demandada pelo ramal. Resultados para a inserção de dois alimentadores de distribuição utilizando o módulo PyFMI estão mostrados na Figura 12, enquanto os resultados com o módulo OMSimulator são mostrados na Figura 13.

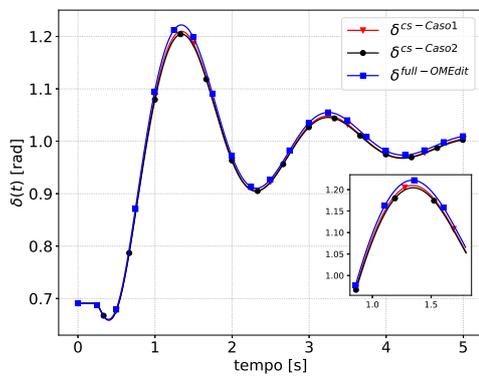
Os resultados para as duas estimativas de \bar{Z}_c , mostraram-se coerentes com a simulação do sistema sem partições, para ambos módulos de co-simulação.

É importante ressaltar, entretanto, que para três alimentadores ou mais, nenhum resultado pôde ser obtido para a simulação do sistema completo, devido ao alto tempo computacional e problemas de convergência na simulação. Assim, para tais casos, a co-simulação tornou-se a única alternativa para a realização dos estudos dinâmicos com a inserção de múltiplos alimentadores.

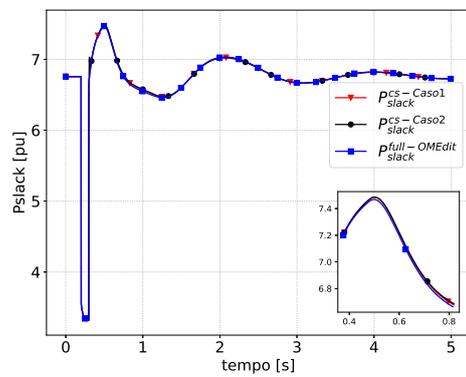
Além disso, a co-simulação também reduziu o tempo computacional em comparação com a simulação do sistema completo. Para efeito de comparação, a Figura 14 apresenta os tempos computacionais para os módulos PyFMI e OMSimulator, assim como a simulação do sistema completo realizada com o OpenModelica. Como se pode observar, os tempos do PyFMI e OMSimulator são bastante semelhantes.

Apesar da escalabilidade da co-simulação sugerida pela Figura 14, novas investigações ainda são necessárias para reduzir os tempos absolutos de simulação de sistemas de potência por meio das FMUs.

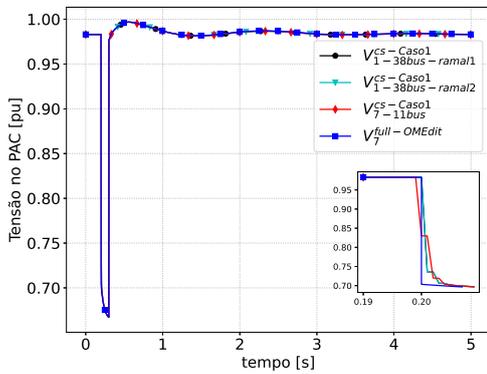
Figura 12 – Quantidades medidas durante a falta - PyFMI - Sistema de transmissão e dois alimentadores de 38 barras.



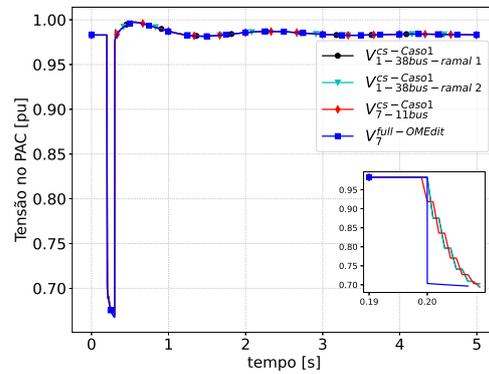
(a) Ângulo relativo entre G_1 e G_4



(b) Potência na barra de referência



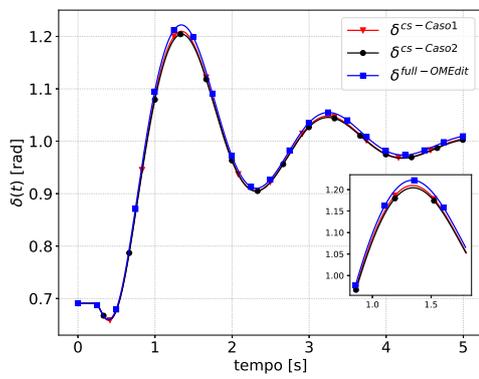
(c) Tensão na barra de interface - Caso 1



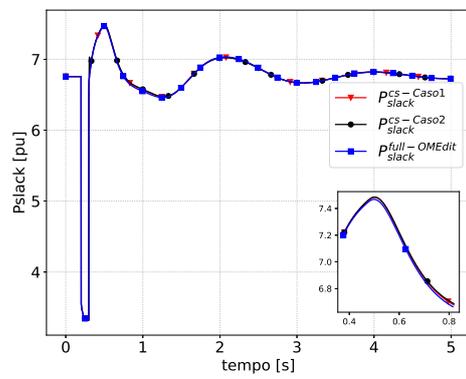
(d) Tensão na barra de interface - Caso 2

Fonte: Elaborado pelo autor (2022)

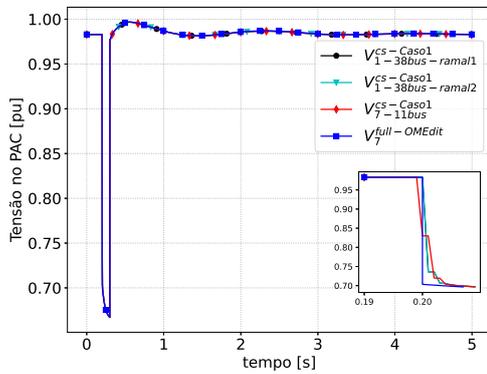
Figura 13 – Quantidades medidas durante a falta - OMSimulator - Sistema de transmissão e dois alimentadores de 38 barras.



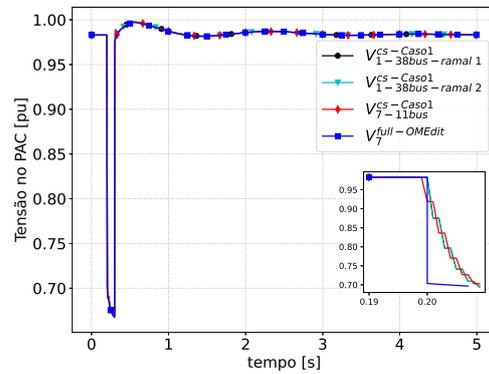
(a) Ângulo relativo entre G_1 e G_4



(b) Potência na barra de referência



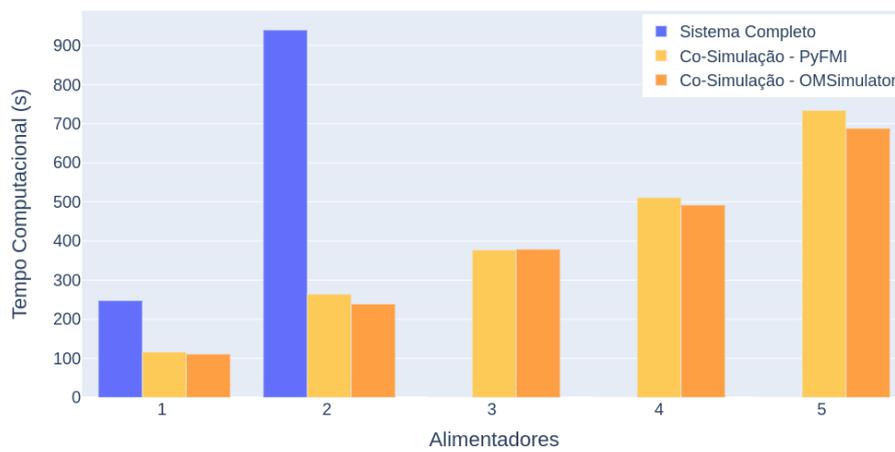
(c) Tensão na barra de interface - Caso 1



(d) Tensão na barra de interface - Caso 2

Fonte: Elaborado pelo autor (2022)

Figura 14 – Tempos computacionais para o sistema de transmissão conectado a vários alimentadores de 38 barras.



Fonte: Elaborado pelo autor (2022)

4.3 CO-SIMULAÇÃO ENTRE OPENMODELICA E OPENDSS

Diante dos resultados bem sucedidos da co-simulação com FMUs, a proposta pode-se estender para o acoplamento com ferramentas distintas. Dessa forma, será utilizada a plataforma OpenModelica para a implementação do sistema de transmissão enquanto a plataforma OpenDSS será empregada para o sistema de distribuição.

Na abordagem proposta, um ambiente Python foi empregado como ponte entre um sistema de transmissão embutido em uma FMU e o OpenDSS que resolve o fluxo de potência do sistema de distribuição. O sistema de transmissão, por sua vez, é modelado através de um circuito fasorial de sequência positiva, enquanto o OpenDSS modela o sistema de distribuição no domínio fasorial e trifásico.

No que diz respeito ao OpenDSS, torna-se importante o conhecimento de alguns aspectos da ferramenta, de forma a possibilitar sua utilização para o acoplamento com a FMU. Na Subseção 4.3.1 serão apresentados aspectos gerais do OpenDSS, e o algoritmo de fluxo de potência inerente à ferramenta. Será apresentado também o módulo em Python utilizado, para a utilização do OpenDSS.

4.3.1 Software OpenDSS

O OpenDSS (*Open System Distribution Simulator*) é uma ferramenta de simulação do sistema elétrico para simulação de sistemas de distribuição desenvolvido pela EPRI. Essa ferramenta se refere a uma implementação de código aberto do DSS. É baseado em linhas de comandos, sendo essas linhas provenientes do usuário, de um arquivo texto ou programas externos [19, 38].

A ferramenta pode ser implementada como um programa executável (OpenDSS.exe), pelo qual os usuários podem definir os circuitos. Outra forma, seria através da interface COM (*Component Object Model*) da *Microsoft*, de forma que os usuários possam projetar e executar modos e recursos de solução personalizados a partir de um programa externo e executar as funções do simulador, incluindo a definição dos dados do modelo. Nesse contexto, o programa pode ser implementado de forma totalmente independente de qualquer banco de dados ou arquivo de texto fixo. Outra característica da ferramenta é a possibilidade de desenvolvedores personalizarem o programa para solucionar necessidades específicas, através de bibliotecas de acesso dinâmico [19].

No que diz respeito aos componentes da rede elétrica, em [19, 38] são caracterizados cada elemento, de forma a serem utilizados na ferramenta OpenDSS. Alguns elementos serão apresentados a seguir sendo de interesse da co-simulação, para o acoplamento com a FMU:

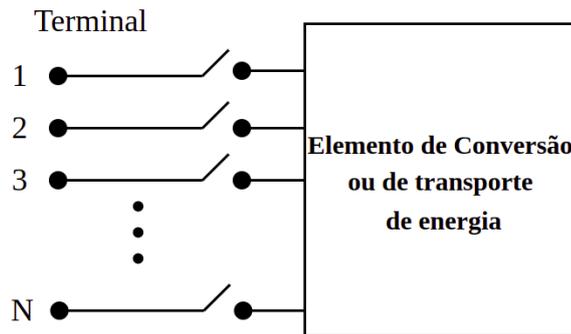
1. Barra: A barra é um elemento de circuito que tem N nós, como vista na Figura 15.

Figura 15 – Barra do OpenDSS



Fonte: Elaborado pelo autor (2022)

Figura 16 – Terminal do OpenDSS



Fonte: Elaborado pelo autor (2022)

Barras são os pontos de conexão para todos os outros elementos do circuito. Na ferramenta, as barras são definidas em decorrência da criação de novos elementos.

2. Terminal: Os elementos do OpenDSS possuem um ou mais terminais que podem conter diversos conectores, como mostrado na Figura 16. Cada terminal deve ser conectado a apenas uma barra.
3. Equivalente de Thévenin: O equivalente de Thévenin pode representar o sistema de transmissão vista pela subestação ou o equivalente da subestação vista por um alimentador. No OpenDSS, todo circuito estabelecido deve ser inicializado pelo elemento de circuito *Circuit*. Esse elemento é um equivalente de Thévenin, ou seja, uma fonte de tensão (\bar{V}_s) atrás de uma impedância (\bar{Z}_s). Internamente, quando o elemento é criado, uma fonte de tensão trifásica nomeada *Source* é conectada à uma barra, chamada *SourceBus*. Esse equivalente é considerado a barra de referência para cálculos de fluxo de potência. Os principais parâmetros a serem definidos no elemento *Circuit*, são mostrados na Tabela 6.
4. Elementos: Linhas de distribuição, transformadores, cargas e outros elementos de entrega de energia, são modelados por suas matrizes de admitância nodal, chamadas também de matrizes primitivas. Em [19] e [38] é descrita a modelagem de cada elemento, para a formação da matrizes primitivas. Cargas, com características não

lineares, podem ser representadas por fontes de corrente de compensação, em um equivalente de Norton. Isso permite acomodar, uma ampla variedade de modelos de carga em que as correntes de injeção sejam funções da tensão. O OpenDSS constrói as matrizes primitivas de cada elemento e as entrega para um solucionador de matrizes esparsas, chamado *KLU solve*, que permite a formação da matriz de admitância nodal do sistema completo [39].

4.3.2 Fluxo de potência do OpenDSS

Para o cálculo de fluxo de potência no OpenDSS, o algoritmo utilizado é o método Normal, que é padrão do OpenDSS. O Apêndice D descreve o referido método de solução. O método normal para o estudo vigente, realizará o cálculo do fluxo de potência, utilizando o modo *Snapshot*.

Para a comunicação entre o OpenDSS e o ambiente Python, será utilizado o pacote *OpenDSSDirect.py* implementa uma interface direta com o OpenDSS no ambiente python utilizando a extensão *DSS_Python*. Esta extensão foi construída baseada na biblioteca C-API que expõe uma API compatível com C plano para facilitar o uso em diversas linguagens, entre elas a linguagem Python. Nesse âmbito, no ambiente de simulação o pacote pode ser importado como um objeto *dss*.

No que diz se refere a modelagem da rede, esta foi descrita em um arquivo de texto de extensão *.dss*. O objeto importado possui uma função que utiliza a interface de texto do OpenDSS, a função *run_command*. Através da interface de texto, o arquivo de extensão *.dss* pode ser redirecionado, como mostrado na Algoritmo 6.

Uma vantagem da utilização do objeto *dss* para a co-simulação é que, uma vez que o sistema modelado no OpenDSS foi importado, o objeto pode ser reutilizado sempre

Tabela 6 – Parâmetros do Elemento *Circuit*

Parâmetro	Descrição
<i>pu</i>	Módulo da tensão [pu]
<i>angle</i>	Ângulo da tensão [°]
<i>kV</i>	Tensão nominal da linha [kV]
<i>frequency</i>	Frequência da fonte [Hz]
<i>bus1</i>	Nome da barra a qual o terminal do elemento é conectado
<i>R1</i>	Resistência de sequência positiva da fonte [pu]
<i>X1</i>	Reatância de sequência positiva da fonte [pu]
<i>mvasc3</i>	Potência de curto circuito trifásico [MVA]
<i>mvasc1</i>	Potência de curto circuito monofásico [MVA]
<i>baseMVA</i>	Potência base do sistema [MVA]

Fonte: Elaborado pelo autor (2022)

Algoritmo 6: Interface OpenDSS

```

import opendssdirect as dss
dss.run_command('Redirect "sistema.dss"')

```

que necessário, sem a necessidade de ser importado várias vezes. Após o carregamento do sistema, pode-se utilizar outras funções referentes ao objeto criado de forma a atender as necessidades da co-simulação. A seguir serão apresentados os passos que foram utilizados para o acoplamento entre os sistemas modelados no OpenModelica e OpenDSS através da linha fictícia.

4.3.3 Modo de Inicialização

Como primeiro passo do processo, assim como na co-simulação com FMUs, é a inicialização.

Nesse estágio, uma condição inicial adequada só é possível combinando os dois sistemas em uma única estratégia de solução de fluxo de potência. O problema principal nesse momento é que não se pode determinar com precisão a potência complexa drenada pelo sistema de distribuição ou a potência complexa fornecida pelo sistema de transmissão sem que a tensão do ponto de acoplamento comum seja completamente conhecida, tanto em módulo como em ângulo. As únicas restrições que necessitam ser aplicadas são que a tensão no ponto de acoplamento comum (PAC) deve ser a mesma para ambos os sistemas, transmissão e distribuição, e a potência complexa que é drenada pelo sistema de distribuição deve ser integralmente fornecida pelo sistema de transmissão.

Para a aplicação desta restrição, um processo iterativo foi implementado entre OpenModelica e o OpenDSS de acordo com a Figura 17. As etapas desse procedimento são descritas a seguir [40]:

1. Atribuir um valor inicial para a tensão $\bar{V}_s = \bar{V}_{PAC-DSS}$ da barra de referência do sistema modelado no OpenDSS (por exemplo, $1\angle 0^\circ$ [pu]);
2. Para a tensão \bar{V}_s definida, emprega-se o OpenDSS para se resolver o fluxo de potência para o sistema de distribuição (DS). Assim, a potência complexa de sequência positiva drenada da barra de referência, \bar{S}_{DS} , é calculada;
3. Por sua vez, considera-se uma carga de potência \bar{S}_{DS} conectada ao ponto de acoplamento comum do sistema de transmissão (TS);
4. Agora, outro fluxo de potência para o sistema de transmissão (TS) é calculado. A tensão \bar{V}_{PAC-OM} na barra de interface, assim como a corrente que flui na interface \bar{I}_{m-OM} são retornados;

Algoritmo 7: Algoritmo para solução conjunta dos sistemas de transmissão e distribuição

entrada :

- \bar{V}_{s0} : Estimativa inicial da tensão interna da subestação

saída :

- \bar{V}_s : Tensão interna da subestação calculada
- \bar{I}_{m-DSS} : Corrente drenada da subestação
- $\bar{S}_{PAC-DSS}$: Potência complexa de sequência positiva drenada da barra de referência

Inicialização das variáveis internas

$\bar{V}_s \leftarrow \bar{V}_{s0}$

Inicialização do contador de iterações

$h \leftarrow 0$

$nao_convergado \leftarrow True$

Solução iterativa para o PAC

enquanto $nao_convergado$ **faça**

$\bar{V}_{PAC-DSS}, \bar{I}_{m-DSS}, \bar{S}_{PAC-DSS} \leftarrow DSS(\bar{V}_s)$
$\bar{V}_{PAC-OM}, \bar{I}_{m-OM} \leftarrow OM(\bar{S}_{PAC-DSS})$
$\bar{V}_s \leftarrow \bar{V}_{PAC-OM} - \bar{Z}_s \bar{I}_{m-OM}$
$nao_convergado \leftarrow \ \bar{V}_{PAC-DSS} - \bar{V}_{PAC-OM}\ > \varepsilon$
$h \leftarrow h + 1$

5. Defina a tensão $\bar{V}_s = \bar{V}_{PAC-OM} - \bar{Z}_s \bar{I}_{m-OM}$;

6. Os passos (2) a (5) são repetidos até que a convergência seja atingida, ou seja, $\|\bar{V}_{PAC-OM} - \bar{V}_{PAC-OpenDSS}\| < \varepsilon$, onde ε é uma tolerância aceitável.

Essa estratégia é equivalente a um método iterativo de ponto-fixo, onde se alternam soluções entre sistemas até que as grandezas de interface se aproximem o suficiente. De forma a auxiliar no entendimento dos passos citados anteriormente, o Algoritmo 7 descreve o processo iterativo de solução para a inicialização.

Os termos referentes a $DSS(\bar{V}_s)$ e $OM(\bar{S}_{PAC-DSS})$ são funções de \bar{V}_s e $\bar{S}_{PAC-DSS}$ para cálculo do fluxo de potência nos sistemas modelados no OpenDSS, e no OpenModelica, respectivamente.

Na sequência, serão explicados os passos referentes a função $DSS(\bar{V}_s)$.

O Algoritmo 8 descreve os passos para se obter a tensão terminal $\bar{V}_{PAC-DSS}$ e corrente \bar{I}_m , ambos de sequência positiva.

Primeiramente, a função recebe como parâmetro a tensão \bar{V}_s . A fonte equivalente é um objeto `Vsources` do pacote `opendssdirect.py`. Esse é um objeto que mapeia a classe `Vsource` do OpenDSS. Nesse objeto, as funções `PU` e `AngleDeg` são disponíveis e podem receber parâmetros de módulo em pu e fase em graus da tensão. Recebidos os parâmetros anteriormente citados, calcula-se o fluxo de potência através da função `solve` pertencente

Algoritmo 8: Solução de rede - OpenDSS

```

Função DSS( $\bar{V}_s$ )
  entrada :
    •  $\bar{V}_s$ : Tensão interna da fonte que representa a subestação do sistema
      de distribuição

  saída :
    •  $\bar{V}_{PAC-DSS}$ : Tensão no PAC do sistema de distribuição
    •  $\bar{I}_m$ : Corrente drenada da subestação
    •  $\bar{S}_{DSS}$ : Potência complexa de sequência positiva drenada da barra de referência

   $V_s \leftarrow |\bar{V}_s|$ 
   $\theta_s \leftarrow \tan^{-1} \left( \frac{\text{Im} \{ \bar{V}_s \}}{\text{Re} \{ \bar{V}_s \}} \right)$ 
  dss.Vsources.PU( $V_s$ )
  dss.Vsources.AngleDeg( $\theta_s$ )
  dss.Solution.solve()
  status = dss.Solution.Converged()
  dss.Circuit.SetActiveBus('sourcebus')
  dss.Circuit.SetActiveElement('SOURCE')
   $V_{PAC-DSS-re} \leftarrow \text{dss.Bus.CplxSeqVoltages}()[2]/V_{base}$ 
   $V_{PAC-DSS-im} \leftarrow (\text{dss.Bus.CplxSeqVoltages}()[3])/V_{base}$ 
   $\bar{V}_{PAC-DSS} \leftarrow V_{PAC-DSS-re} + jV_{PAC-DSS-im}$ 
   $I_{m-re} \leftarrow (\text{dss.CktElement.CplxSeqCurrents}()[2])/I_{base}$ 
   $I_{m-im} \leftarrow (\text{dss.CktElement.CplxSeqCurrents}()[3])/I_{base}$ 
   $\bar{I}_m \leftarrow I_{m-re} + jI_{m-im}$ 
   $P_{PAC-DSS} \leftarrow \text{dss.Circuit.TotalPower}()[0]$ 
   $Q_{PAC-DSS} \leftarrow \text{dss.Circuit.TotalPower}()[1]$ 
   $\bar{S}_{PAC-DSS} \leftarrow P_{PAC-DSS} + jQ_{PAC-DSS}$ 

  retorna  $\bar{V}_{PAC-DSS}, \bar{I}_m, \bar{S}_{PAC-DSS}$ 

```

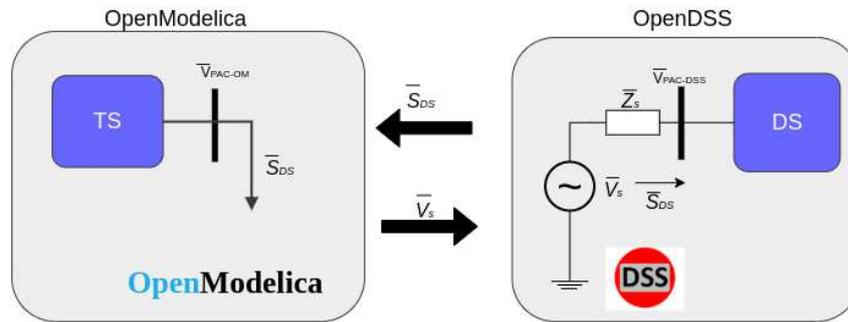
ao objeto **Solution**. Além disso, a convergência do fluxo de potência pode ser verificada através da função **Converged**, retornando *True* se convergido, e *False* caso contrário.

Posteriormente, são extraídos os componentes de sequência positiva da tensão e corrente da interface, assim também como a potência total. Para isso, podem ser utilizados as funções do objeto **Circuit**. Esse objeto permite o acesso de valores relacionados ao circuito, à solução atual e aos elementos do circuito.

Nesse contexto, ativa-se a barra de referência *sourcebus*, através da função **SetActiveBus**, assim como o elemento *SOURCE* referente a fonte atrás da impedância, através da função **SetActiveElement**. Desse modo, através das funções **CplxSeqVoltages** e **CplxSeqCurrents** do objeto **CktElement**, são extraídas os componentes de sequência positiva (partes real e imaginária) de $\bar{V}_{PAC-DSS}$ e \bar{I}_m . É importante salientar que no OpenDSS, as correntes são injetadas nos terminais do elemento **SOURCE**, necessitando-se dessa forma, atentar quanto ao sinal da corrente. Ademais, as potências ativa e reativa totais podem ser extraídas através da função **TotalPower**.

Resultados apresentados para essa estratégia de inicialização também foram com-

Figura 17 – Inicialização - Processo iterativo entre OpenModelica e OpenDSS.



Fonte: Elaborado pelo autor (2022)

parados com métodos de solução iterativos disponibilizados pelo módulo SciPy, mais especificamente a função `root`, que em geral apresentou desempenho computacional inferior à estratégia descrita acima. Essa diferença deve-se à necessidade da função `root` de estimar matrizes jacobianas numericamente como parte dos cálculos em cada iteração.

4.3.4 Linha Fictícia entre FMU e OpenDSS

A adoção de linhas fictícias, detalhada na seção anterior, também foi testada para o acoplamento entre as ferramentas OpenModelica e OpenDSS. Durante a solução no tempo dos modelos implementados no OpenModelica e no OpenDSS, a estratégia adotada manteve a linha de transmissão fictícia externa ao OpenDSS, como parte do próprio ambiente de co-simulação. Nesse caso, um processo iterativo entre a linha de transmissão fictícia e o OpenDSS foi implementado para cada cálculo referente a cada passo de integração. Essa estratégia foi adotada por questões de convergência das soluções do OpenDSS, quando parte da linha fictícia foi integrada ao sistema modelado no OpenDSS. Também por questões de convergência, ao invés de se utilizar equivalentes de Norton, previamente descritos, a linha de transmissão fictícia foi modelada por meio de equivalentes de Thévenin. Portanto, os termos históricos foram representados pelas tensões históricas $\bar{E}_{hk}(t)$ e $\bar{E}_{hm}(t)$.

O esquema apresentado na Figura 18 ilustra o procedimento.

Os passos para a co-simulação serão detalhados a seguir. De forma a corroborar com o entendimento dos passos, o Algoritmo 9 descreve o processo de co-simulação entre o OpenDSS e o OpenModelica pelo método da linha de transmissão fictícia.

Na estratégia apresentada, o algoritmo recebe como entrada a tensão interna \bar{V}_{s0} que foi calculada previamente durante a inicialização, bem como os valores iniciais das tensões históricas da linha de transmissão fictícia, descritos em (4.30). Essas tensões históricas podem ser obtidas a partir da tensão no ponto de acoplamento comum \bar{V}_{PAC} e

Algoritmo 9: Co-simulação entre o OpenModelica e OpenDSS

```

entrada :  $\overline{E}_{hk0}, \overline{E}_{hm0}, \overline{V}_{s0}, \overline{Z}_c$ 
saída : Grandezas de interesse

# Carregamento dos sistemas no ambiente de simulação
fmu_TS ← load_fm(" sistema1.fmu")
opendssdirect.run_command('Redirect sistema2.dss')

# Configurações da simulação
Atribuir valores para  $t_f, \Delta t$ 

# Inicialização da FMU do sistema de transmissão
fmu_TS.instantiate()
fmu_TS.setup_experiment()
fmu_TS.enter_initialization_mode()
fmu_TS.set("hin_re", Real( $\overline{E}_{hk0}$ ))
fmu_TS.set("hin_im", Imag( $\overline{E}_{hk0}$ ))
fmu_TS.set("Zc.re", Real( $\overline{Z}_c$ ))
fmu_TS.set("Zc.im", Imag( $\overline{Z}_c$ ))
fmu_TS.time = 0
fmu_TS.exit_initialization_mode()

# Inicialização do sistema modelado no OpenDSS
(Fluxo de potência calculado de acordo com o Algoritmo 7)

# Modo de integração
enquanto  $t < t_{final}$  faça
  # Solução independente de cada sistema
  status1 ← fmu_TS.do_step(t,  $\Delta t$ )
  status2,  $\overline{V}_s, \overline{E}_{hk} \leftarrow$  calcula_Ehk( $\mathbf{v}_s, \overline{E}_{hm}, t$ )

  # Verifica se os cálculos do passo e do fluxo de potência foram bem sucedidos
  se  $status1 = \text{pyfmi.fmi.FMI\_ERROR}$  ou  $status2 = \text{False}$  então
    └ break

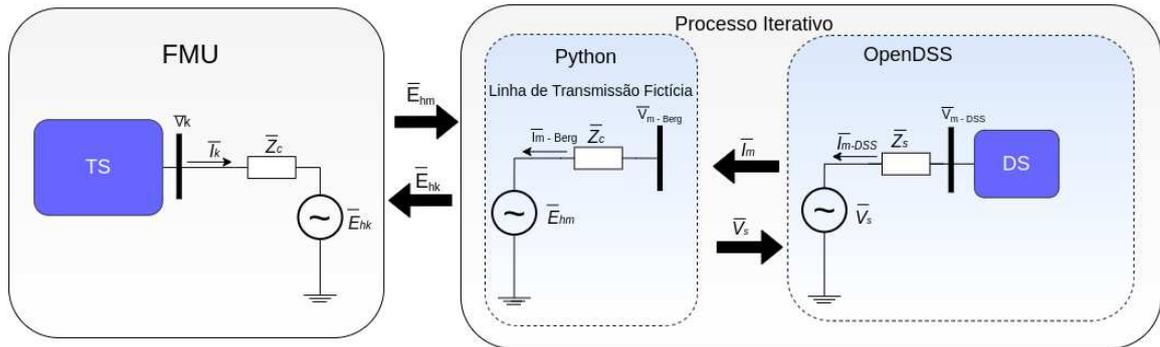
  # Armazenamento das variáveis de interesse
   $E_{hm-re} \leftarrow$  fmu_TS.get("hout_re")
   $E_{hm-im} \leftarrow$  fmu_TS.get("hout_im")

  # Atualização das entradas
   $\overline{E}_{hm} \leftarrow E_{hm-re} + jE_{hm-im}$ 
  fmu_TS.set("hin_re", Real( $\overline{E}_{hk}$ ))
  fmu_TS.set("hin_im", Imag( $\overline{E}_{hk}$ ))

  # Incremento do tempo de simulação
   $t \leftarrow t + \Delta t$ 

```

Figura 18 – Iteração entre o OpenDSS e a linha fictícia.



Fonte: Elaborado pelo autor (2022)

corrente fluindo na interface \bar{I}_{m0} obtidas ao final da inicialização.

$$\bar{E}_{hk0} = \bar{V}_{PAC} + \bar{Z}_c \bar{I}_{m0} \quad (4.30a)$$

$$\bar{E}_{hm0} = \bar{V}_{PAC} - \bar{Z}_c \bar{I}_{m0} \quad (4.30b)$$

Para o valor de \bar{Z}_c , as impedâncias \bar{Z}_k e \bar{Z}_m precisam ser estimadas. A estimativa da impedância \bar{Z}_k encontra-se no Seção C.1. No entanto, para estimativa de \bar{Z}_m , deve-se levar em consideração, que o sistema modelado no OpenDSS é trifásico. Assim, a Seção C.2 apresenta a técnica utilizada para a estimativa de \bar{Z}_m . Desse modo, \bar{Z}_c é obtida, através do módulo do autovalor dado em (4.27).

1. Como primeira parte do processo da co-simulação, a FMU deve ser carregada utilizando a função apropriada do módulo PyFMI ou OMSimulator que permita realizar esse procedimento. Já o sistema modelado no OpenDSS, deve ser lido utilizando função apropriada, como já mencionado.
2. Em seguida, define-se os parâmetros de simulação como o tempo final t_{final} e passo de integração Δt .
3. Realiza-se a inicialização da modelos, como descrita na seção anterior, e faz-se $t = 0$.
4. No laço de integração, para cada instante de tempo t , são realizadas as seguintes etapas:
 - a) Calcula-se um passo de integração Δt , através da função `doStep`, que invoca o *solver* embutido na FMU .
 - b) Na próxima etapa, entra-se no modo de solução do OpenDSS, para o cálculo do fluxo de potência, e por fim, da tensão histórica de saída. No algoritmo, uma função nomeada de `calcula_Ehk` executa esse procedimento. Para cada passo

de integração, esta função recebe como entrada a tensão histórica \bar{E}_{hm} , um vetor de tensões internas \mathbf{v}_s que armazena \bar{V}_s a cada passo de integração, e o instante atual t , retornando para o mestre o status de convergência da solução de rede, tensão histórica \bar{E}_{hk} e a tensão \bar{V}_s no instante atual. O algoritmo que descreve a função pode ser visto no Algoritmo 10.

Algoritmo 10: Algoritmo para cálculo da tensão histórica \bar{E}_{hk}

Função calcula_Ehk($\mathbf{v}_s, \bar{E}_{hm}, t$)

entrada :

- \mathbf{v}_s : Vetor de tensões internas da subestação
- \bar{E}_{hm} : Tensão histórica no lado do sistema de distribuição
- t : Instante de tempo

saída :

- status: Status de convergência do fluxo de potência
- \bar{E}_{hk} : Tensão histórica no lado do sistema de transmissão
- \bar{V}_s : Tensão interna da subestação calculada

se $t > 2\Delta t$ **então**

$$\left[\bar{V}_s^{est} \leftarrow \frac{\bar{V}_s(t - \Delta t)^2}{\bar{V}_s(t - 2\Delta t)} \right]$$

senão

$$\left[\bar{V}_s^{est} \leftarrow \bar{V}_s(t - \Delta t) \right]$$

status, $\bar{V}_s \leftarrow \text{itera_linha_opendss}(\bar{E}_{hm}, \bar{V}_s^{est})$

$\bar{V}_m, \bar{I}_m \leftarrow \text{DSS}(\bar{V}_s)$

$\bar{E}_{hk} \leftarrow \bar{V}_m + \bar{I}_m \bar{Z}_c$

retorna status, \bar{E}_{hk}, \bar{V}_s

- c) A função `calcula_Ehk` recebe como parâmetros de entrada o vetor de tensões internas \mathbf{v}_s . Nessa etapa, um processo iterativo entre o OpenDSS e o elo da linha fictícia acontece, como já mencionado anteriormente. Porém, como esse processo é feito a cada passo de integração t , a estimativa inicial \bar{V}_s^{est} , pode ser fornecida por extrapolação como dado por (4.31) [41]. Nesse caso, dois valores passados precisam ser armazenados, ou seja, a extrapolação acontece quando $t > 2\Delta t$. Caso contrário, $\bar{V}_s^{est} = \bar{V}_s(t - \Delta t)$. A principal vantagem de se empregar a referida extrapolação é aumentar a velocidade de convergência do processo iterativo necessário em cada passo de tempo da co-simulação.

$$\bar{V}_s^{est}(t) = \frac{\bar{V}_s(t - \Delta t)^2}{\bar{V}_s(t - 2\Delta t)} \quad (4.31)$$

- d) Na próxima etapa, uma função `itera_linha_opendss` pode ser implementada, com a finalidade iterar o elo da linha conectado no lado do sistema de distribuição com o sistema modelado no OpenDSS em cada passo de iteração. Essa função recebe como parâmetros de entrada, a tensão histórica \bar{E}_{hm} que se mantém fixa naquele instante e a tensão estimada \bar{V}_s^{est} . A função que implementa o

processo iterativo pode ser vista no Algoritmo 11. Os seguintes passos podem ser realizados:

Algoritmo 11: Algoritmo para iteração entre o elo da linha de transmissão e o OpenDSS

Função *itera_linha_opendss*($\bar{E}_{hm}, \bar{V}_{s0}$)

entrada :

- \bar{E}_{hm} : Tensão histórica no lado do sistema de distribuição
- \bar{V}_{s0} : Estimativa inicial da tensão interna da subestação

saída :

- \bar{V}_s : Tensão interna da subestação calculada

Inicialização das variáveis internas

$\bar{V}_s \leftarrow \bar{V}_{s0}$

Inicialização do contador de iterações

$h \leftarrow 0$

$nao_convergado \leftarrow True$

Solução iterativa para o PAC

enquanto *nao_convergado* **faça**

$\bar{I}_m, \bar{V}_{m-DSS} \leftarrow DSS(\bar{V}_s)$

$\bar{V}_{m-Berg} \leftarrow \bar{E}_{hm} + \bar{Z}_c \bar{I}_m$

$nao_convergado \leftarrow \|\bar{V}_{m-Berg} - \bar{V}_{m-DSS}\| \geq \varepsilon$

$\bar{V}_s \leftarrow \bar{V}_{m-Berg} - \bar{Z}_s \bar{I}_m$

$h \leftarrow h + 1$

retorna \bar{V}_s

- i. O OpenDSS resolve o fluxo de potência do sistema de distribuição com $\bar{V}_s = \bar{V}_{m-OpenDSS}$ fixo. O resultado é corrente drenada da barra de referência \bar{I}_m .
 - ii. A corrente \bar{I}_m atual agora é drenada do terminal m da linha de transmissão fictícia, conforme ilustrado na Figura 18, para posterior solução da tensão \bar{V}_{m-Berg} .
 - iii. Os passos (4(d)i) a (4(d)ii) são repetidos até que *nao_convergado* seja falso. A constante ε corresponde a uma tolerância aceitável para o processo iterativo.
- e) Quando a convergência for atingida, pode-se calcular a tensão histórica \bar{E}_{hk} , como descrito no Algoritmo 10.
 - f) Proceder com a aquisição das tensões históricas e variáveis de interesse de ambos os sistemas, de transmissão e distribuição, para o passo de integração t .
 - g) Incrementar o passo de integração $t \leftarrow t + \Delta t$. Se $t < t_{final}$, retornar ao passo (4a), senão terminar a simulação.

É importante ressaltar que durante o processo iterativo descrito para cálculo da tensão no ponto de interconexão entre a linha de transmissão fictícia e o OpenDSS, os

Tabela 7 – Estimativas das impedâncias de Thévenin - Sistemas de transmissão e IEEE 34 barras

Transmissão [pu]	Distribuição [pu]
$0,00176 + j0,034491$	$48,374 + j6,94$

Fonte: Elaborado pelo autor (2022)

termos históricos são mantidos constantes. Esses termos só são atualizados quando o tempo realmente avança.

4.4 CO-SIMULAÇÃO ENTRE SISTEMA DE TRANSMISSÃO E IEEE - 34 BARRAS

Para a validação da metodologia discutida na última seção, foram realizados testes de co-simulação, fazendo-se a interface entre FMUs geradas pelo OpenModelica e o sistema modelado no OpenDSS.

O primeiro teste realizado nesse âmbito, foi a co-simulação entre o sistema de transmissão 11 barras modelado no OpenModelica, embutido em uma FMU e uma única instância do sistema de distribuição IEEE - 34 barras modelado no OpenDSS. A descrição do ramal de distribuição encontra-se no Apêndice A.

Em relação à inicialização, a Seção B.2 apresenta os valores iniciais obtidos da solução conjunta dos sistemas envolvidos na co-simulação. Quanto as estimativas para \bar{Z}_k e \bar{Z}_m , estas podem ser verificadas na Tabela 7.

Analogamente ao primeiro caso, uma única linha de transmissão fictícia foi incluída no modelo, agora representada por meio de equivalentes Thévenin conforme descrito anteriormente.

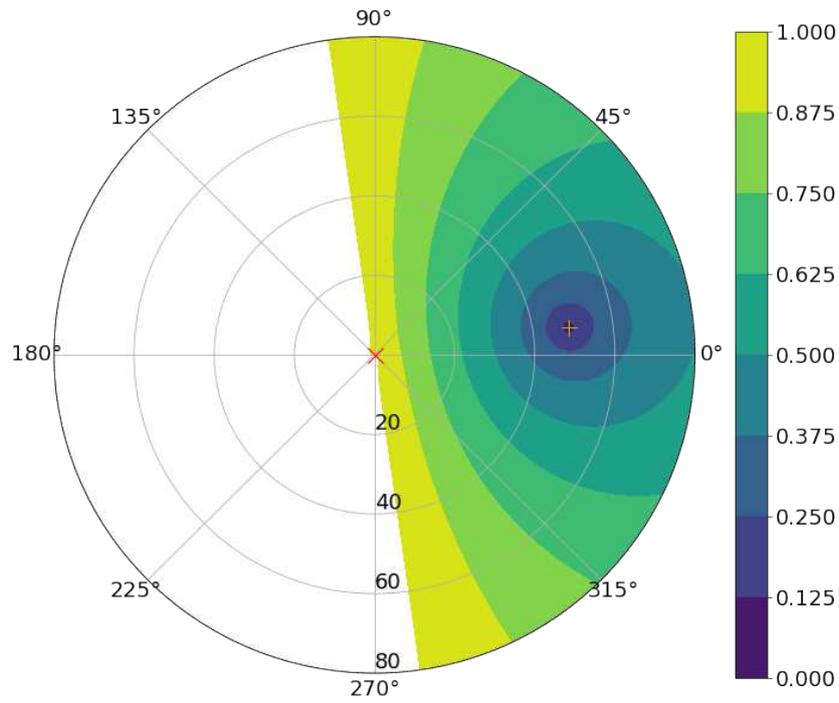
De forma a escolher a impedância \bar{Z}_c para a linha fictícia, a mesma análise realizada para o teste anterior, será para esse estudo de caso. Assim a expressão em (4.27) pode ser analisada substituindo-se os parâmetros de \bar{Z}_k e \bar{Z}_m , dados na Tabela 7.

Seja $f(Z_c, \theta_c)$ ilustrada no gráfico polar na Figura 19(a). De forma análoga a análise feita no estudo anterior, na medida que $|f|$ se reduz, observa-se que, \bar{Z}_c tende a se aproximar das impedâncias \bar{Z}_k e \bar{Z}_m . Devido à diferença de ordem de grandeza entre as impedâncias de Thévenin, a Figura 19(b) mostra de forma ampliada que \bar{Z}_c também tende a se aproximar de \bar{Z}_k .

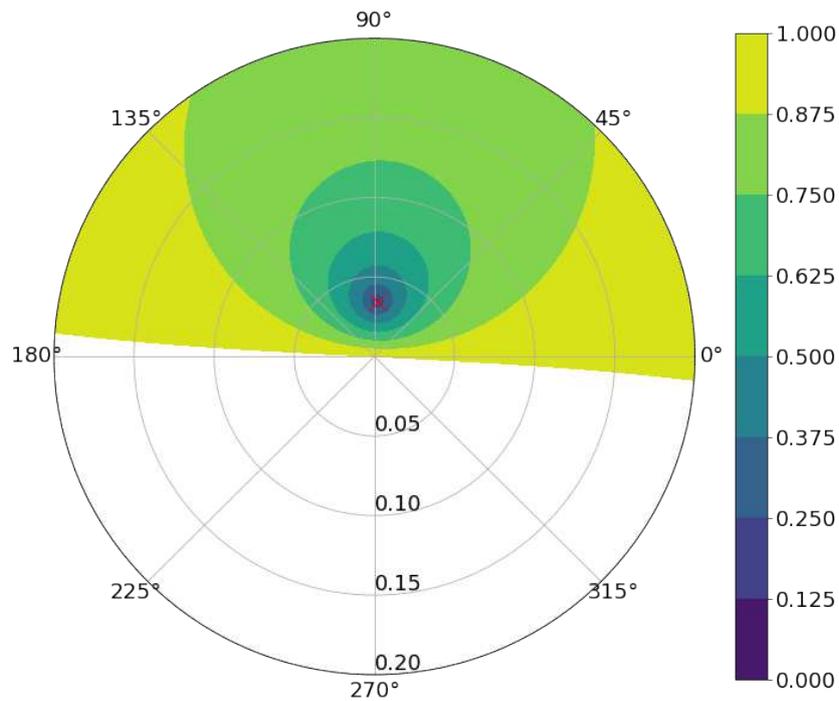
Uma limitação encontrada para essa metodologia, é que quando \bar{Z}_c começa a ter valores elevados, o fluxo de potência no OpenDSS apresenta problemas de convergência, não sendo possível de ser utilizado para a co-simulação. Desse modo, não foi possível obter resultados para quando \bar{Z}_c é próximo de \bar{Z}_m .

Uma possível explicação para esse fato, é que o fluxo de potência do OpenDSS,

Figura 19 – Módulo dos autovalores para o sistema de transmissão e distribuição de 34 barras.



(a) Visualização ampliada considerando \bar{Z}_c próximo a \bar{Z}_m



(b) Visualização ampliada considerando \bar{Z}_c próximo a \bar{Z}_k

Tabela 8 – Estimativa da impedância característica - Sistemas de transmissão e IEEE 34 barras

\bar{Z}_c [pu]	f
$0,03503/87,3825^\circ$	0,05168

Fonte: Elaborado pelo autor (2022)

Tabela 9 – Valores NIAE - Sistema de transmissão e IEEE 34 barras [%]

δ_{14}	V_{bus7}	P_{slack}
99,785	99,939	99,895

Fonte: Elaborado pelo autor (2022)

utiliza um algoritmo iterativo no modo Normal, método de ponto fixo, que funciona bem para sistemas de distribuição com fontes de tensão rígidas [42], o que não é o caso do sistema IEEE 34 barras, para o método da linha fictícia, devido a inclusão de \bar{Z}_m à fonte.

Assim, para as análises subsequentes, \bar{Z}_c será considerado próximo ao valor de \bar{Z}_k . O valor adotado pode ser visto na Tabela 8. Pela escolha percebe-se que $|f|$ é menor que 10% do limite da estabilidade, já sendo suficiente para a co-simulação.

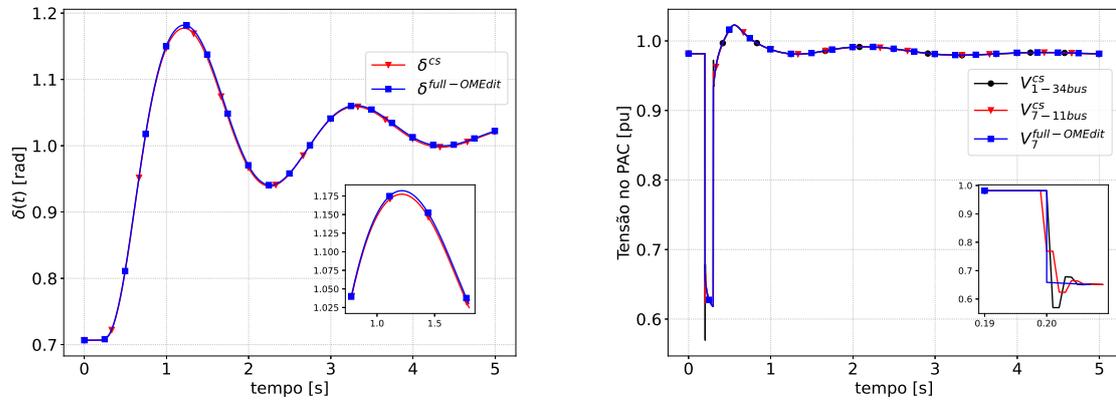
Os resultados da co-simulação são comparados com os obtidos para o sistema de transmissão de 11 barras com apenas cargas de potência constantes. Para esse teste, o passo de integração também foi de $\Delta t = 1$ [ms].

Como pode ser observado, os resultados para ângulo relativo entre G_1 e G_4 , bem como a potência na barra de referência do sistema de transmissão, ficaram bastante próximas dos resultados obtidos com o sistema completo, da mesma forma que a tensão no ponto de acoplamento comum e a potência na barra de referência. Em relação aos valores do NIAE, estes podem ser descritos na Tabela 9

Os valores do NIAE ratificam que o comportamento das grandezas foi muito próximo daqueles obtidos na simulação realizada com o sistema completo. Além disso, os valores obtidos foram maiores que 99,5%. Quanto ao desempenho computacional, o tempo de inicialização referente a solução conjunta OpenModelica-OpenDSS foi de 8,108 [s]. A Tabela 10 apresenta o perfil de tempo computacional para a co-simulação entre o sistema de transmissão e o IEEE 34 barras.

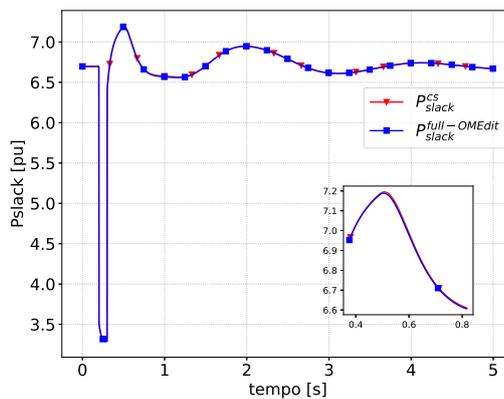
Percebe-se que o tempo de execução total para as funções inerentes ao OpenDSS foi menor que o tempo de execução da FMU, tendo em vista que o OpenDSS é um programa dedicado a simulações de sistemas de potência, enquanto o OpenModelica é dedicado a simulações mais genéricas. Outra situação a ser analisada, é quando \bar{Z}_c assume valores de ordem de grandeza diferentes de \bar{Z}_k . Para isso, variando-se a parte real de \bar{Z}_c , foram realizados testes de acordo com a Tabela 11. O caso I é o caso base para quando \bar{Z}_c é

Figura 20 – Grandezas medidas - Sistema de transmissão e IEEE 34 barras - Método da linha fictícia



(a) Ângulo relativo entre G_1 e G_4

(b) Tensão na barra de interface



(c) Potência na barra de referência

Fonte: Elaborado pelo autor (2022)

Tabela 10 – Perfil da co-simulação - Sistemas de transmissão e IEEE 34 barras

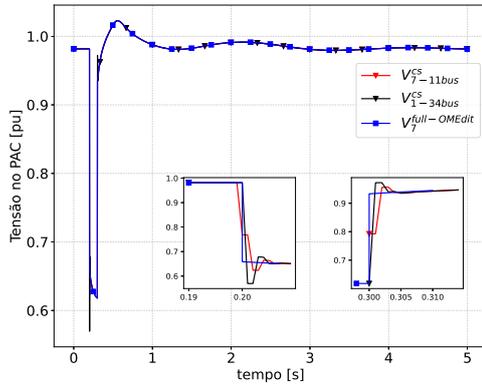
Tempo de execução FMU [s]	Tempo de execução OpenDSS [s]	Proc. Variáveis [s]	Tempo de execução total [s]
3,9	1,5	3,6	9

Fonte: Elaborado pelo autor (2022)

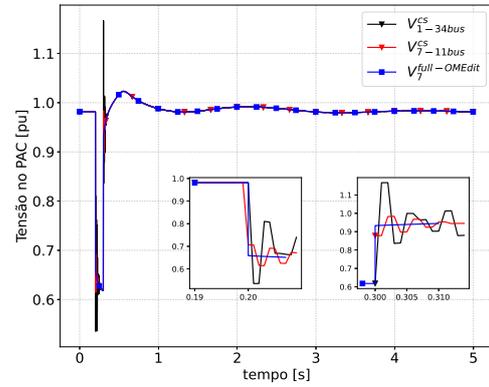
próximo a \bar{Z}_k .

A Tabela 4.4 apresenta uma comparação entre a tensão na barra de interface, para os casos mostrados na Tabela 11. São descritos também valores de NIAE da tensão obtida no PAC do lado do sistema de distribuição.

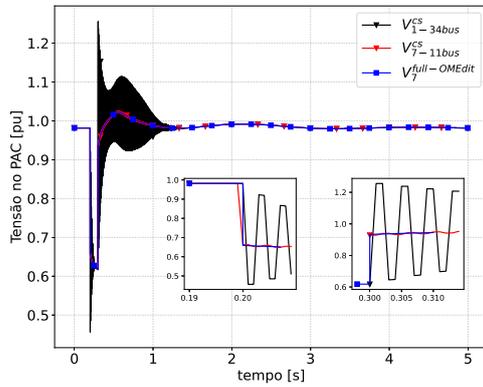
Como observado, quando a ordem de grandeza começa a aumentar em relação ao

Figura 21 – Tensão na barra de interface - Variação de \bar{Z}_c 

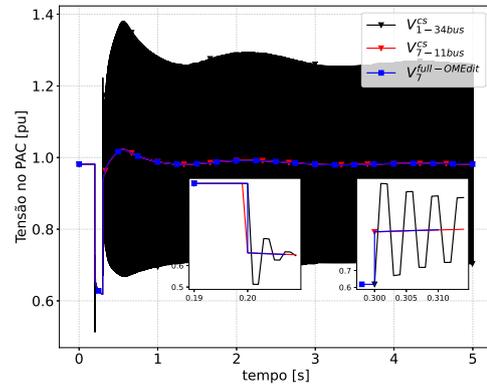
(a) Caso I - NIAE - 99,939%



(b) Caso II - NIAE - 99,874 %



(c) Caso III - NIAE - 98,728 %



(d) Caso IV - NIAE - 72,540 %

Fonte: Elaborado pelo autor (2022)

caso base, oscilações após as discontinuidades começam a aparecer. Ademais, os valores do NIAE decrescem com o aumento das referidas oscilações. Como mencionado anteriormente, o OpenDSS trabalha melhor para fontes equivalentes rígidas. Dessa forma, a topologia do sistema é um fator limitante, pois quando a impedância de Thévenin apresenta valores elevados, oscilações numéricas podem aparecer. Portanto, é importante que a escolha da impedância \bar{Z}_c seja feita de forma apropriada.

Tabela 11 – Variação da impedância característica

Casos	\bar{Z}_c [pu]	f [pu]
I	0,0016 + j0,0344	0,051
II	0,16 + j0,0344	0,9455
III	1,6 + j0,0344	0,9664
IV	16 + j0,0344	0,7149

4.5 CO-SIMULAÇÃO ENTRE SISTEMA DE TRANSMISSÃO E CKT5

Outro teste realizado, foi a co-simulação entre o sistema de transmissão de 11 barras e o sistema de distribuição Ckt5. A rede de distribuição Ckt5 foi criada e publicada pelo EPRI com propósitos de estudos de sistemas de distribuição [43]. Os dados referentes ao sistema de distribuição Ckt5 podem ser encontrados no Subseção A.2.3.

Os resultados da co-simulação aqui apresentados são comparados com os obtidos para o sistema de transmissão de 11 barras suprindo apenas cargas de potência constante. Em relação à inicialização, a Seção B.3 apresenta os valores iniciais obtidos da solução conjunta dos sistemas envolvidos na co-simulação.

De forma a ser escolhida a impedância \bar{Z}_c , a mesma análise realizada para os estudos de caso anteriores, será realizada para o estudo atual. Os valores referentes a \bar{Z}_k e \bar{Z}_m podem ser encontrados na Tabela 12.

Substituindo-se os parâmetros \bar{Z}_k e \bar{Z}_m da Tabela 12 em (4.27), pode ser obtida a expressão a ser analisada. Pode-se assim, obter as regiões descritas na Figura 22(a).

As mesmas conclusões tiradas dos estudos anteriores são válidas nesse caso. Os valores de impedância \bar{Z}_c tendem a se aproximar de \bar{Z}_k e \bar{Z}_m , ao reduzir $|f|$. E da mesma forma, \bar{Z}_c será considerado próximo ao valor de \bar{Z}_k para a co-simulação.

Assim, para as análises subsequentes, \bar{Z}_c será considerado próximo ao valor de \bar{Z}_k . O valor adotado pode ser visto na Tabela 13. Pela escolha, percebe-se que $|f|$ é menor que 10% do limite da estabilidade, já sendo suficiente para a co-simulação.

Considerando a impedância \bar{Z}_c adotada, a Figura 23(a) mostra o ângulo relativo de G_1 em relação a G_4 , enquanto as Figura 23(b) e 23(c) mostram a tensão e potência fluindo através da barra de interface.

Como observado, os resultados para ângulo relativo entre G_1 e G_4 , bem como a potência na barra de referência do sistema de transmissão, ficaram bastante próximas dos resultados obtidos com o sistema completo, da mesma forma que a tensão no ponto

Tabela 12 – Estimativas das impedâncias de Thévenin - Sistemas de transmissão e Ckt5

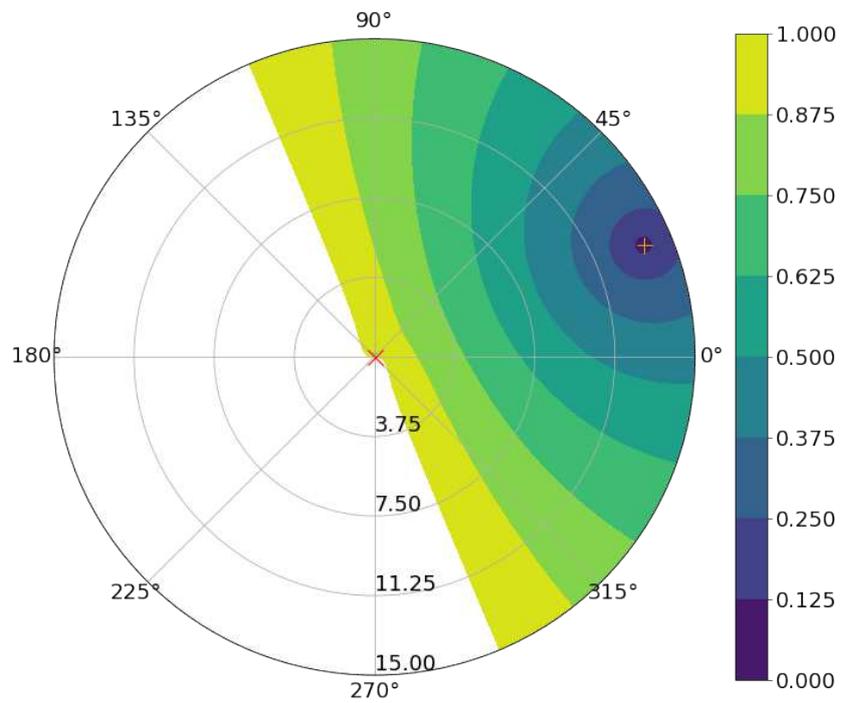
Transmissão [pu]	Distribuição [pu]
0,00176 + j0,0344915	12,5854 + j5,2785

Fonte: Elaborado pelo autor (2022)

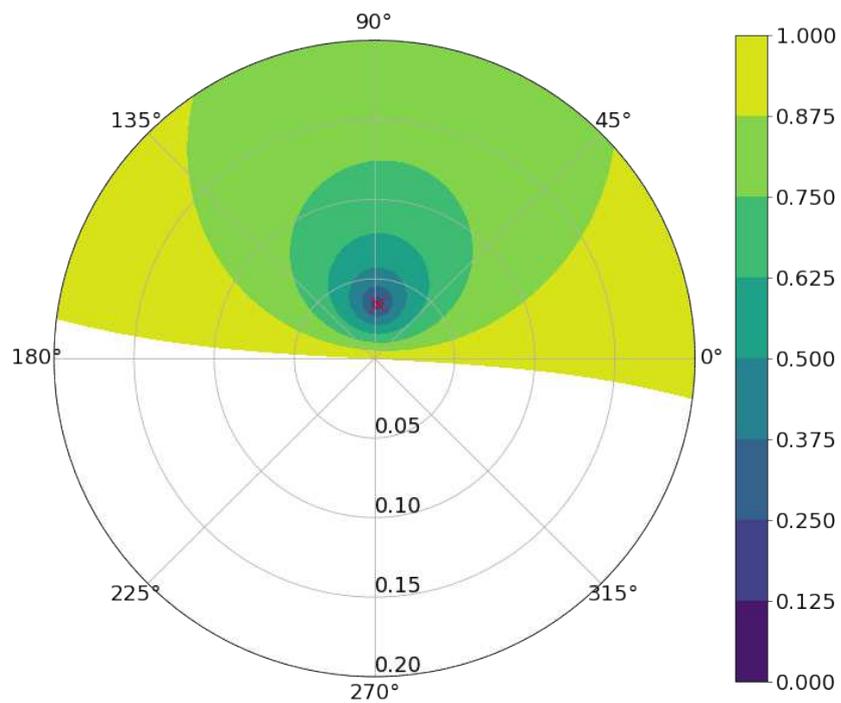
Tabela 13 – Estimativa da impedância característica - Sistemas de transmissão e Ckt5

\bar{Z}_c [pu]	f
0,03503/ <u>87,3825°</u>	0,05163

Figura 22 – Módulo dos autovalores para o sistema de transmissão e distribuição Ckt5.

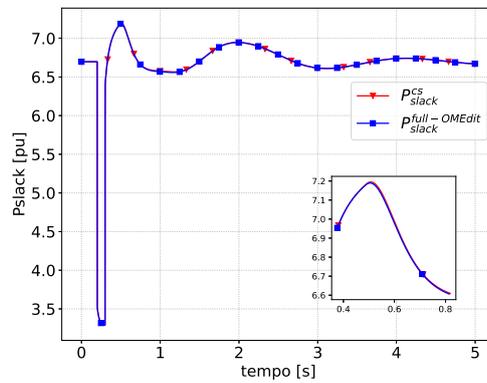
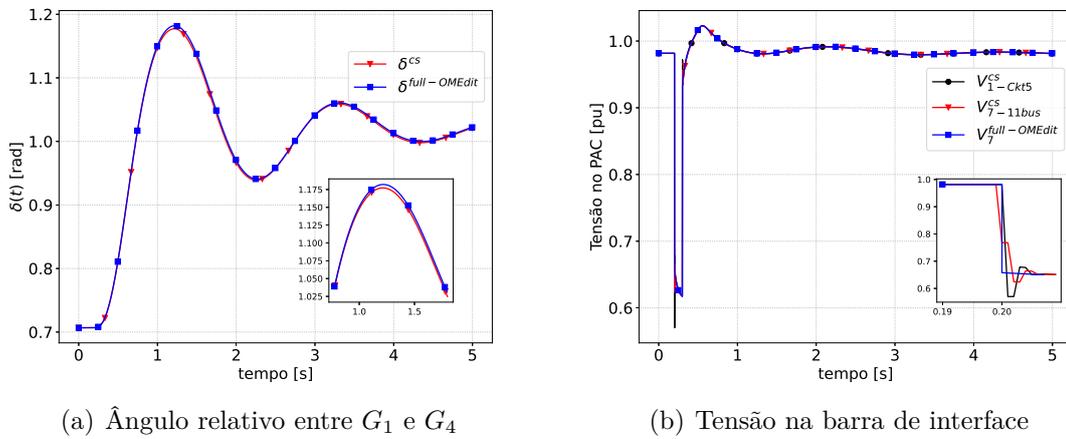


(a) Visualização ampliada considerando \bar{Z}_c é próximo a \bar{Z}_m



(b) Visualização ampliada considerando \bar{Z}_c é próximo a \bar{Z}_k

Figura 23 – Grandezas medidas - Sistemas de transmissão e Ckt5 - Método da linha fictícia



Fonte: Elaborado pelo autor (2022)

Tabela 14 – Valores NIAE - Sistema de transmissão e Ckt5 [%]

δ_{14}	V_{bus7}	P_{slack}
99,774	99,936	99,893

Fonte: Elaborado pelo autor (2022)

de acoplamento comum e a potência na barra de referência. Os valores do NIAE estão mostrados na Tabela 14

Pelo critério do NIAE, as grandezas extraídas foram muito próximas daquelas obtidos na simulação realizada com o sistema completo. Ademais, os valores obtidos foram maiores que 99,7%. Quanto ao desempenho computacional, o tempo de inicialização referente a solução conjunta OpenModelica-OpenDSS foi de 11,0 [s]. A Tabela 15 apresenta o perfil de tempo computacional para a co-simulação entre o sistema de transmissão e o Ckt5.

Tabela 15 – Perfil da co-simulação - Sistemas de transmissão e Ckt5.

Tempo de execução FMU [s]	Tempo de execução OpenDSS [s]	Proc. Variáveis [s]	Tempo de execução total [s]
4,878	16,383	4,108	36

Fonte: Elaborado pelo autor (2022)

Nesse caso, o tempo de execução para as funções inerentes ao OpenDSS foi maior que o tempo de execução da FMU, levando em consideração que o porte do sistema Ckt5 é muito maior em relação ao IEEE 34 barras, formado por 2998 nós elétricos.

4.6 CONCLUSÕES PARCIAIS

Nesse capítulo foi apresentada a metodologia da linha fictícia para o acoplamento entre sistemas elétricos. A metodologia foi explorada tanto para o acoplamento de sistemas modelados em uma mesma plataforma como também, para aqueles modelados em plataformas distintas.

No que diz respeito a modelagem da linha fictícia, a metodologia foi baseada no modelo de Bergeron para uma linha de transmissão sem perdas e foi desenvolvida ao longo do capítulo. Além disso, foi mostrada a necessidade de se escolher uma impedância característica de forma apropriada para a co-simulação, conservando-se o tempo de propagação igual ao passo de integração.

Nesse âmbito, análises do modelo da linha fictícia no domínio discreto foram realizadas para a investigação da impedância apropriada. Assim, a partir da matriz de transição de estados discretos, concluiu-se que os autovalores da referida matriz são iguais as médias geométricas dos coeficientes de reflexão obtidos separadamente para cada subsistema. Ademais, a impedância característica pode ser melhor avaliada através de uma análise de estabilidade associada aos autovalores obtidos.

Posteriormente, foram apresentadas as etapas do gerenciamento da co-simulação, com o auxílio de fluxograma e pseudocódigos. Com as etapas bem definidas, foram realizados três estudos de caso. No primeiro estudo, foi realizada a co-simulação dos sistemas de transmissão de 11 barras e de distribuição de 38 barras, ambos modelados no OpenModelica. Através de gráficos polares do módulo dos autovalores do sistema, foram analisadas as regiões que garantem a estabilidade do modelo. Além disso, foi verificado que valores ideais para a impedância característica tendem a se aproximar das impedâncias de Thévenin de cada sistema envolvido, visto que o módulo do autovalor tende a se aproximar de zero, estando assim, longe do limite da estabilidade.

Acerca dos resultados do estudo supracitado, a co-simulação replicou de forma

fiel os resultados de simulações para o mesmo sistema modelado de forma completa, sem partições, obtidos no OpenModelica. A análise de erros ratificou a precisão dos resultados, visto que os valores de NIAE foram superiores ao valor de referência. Além disso, a estratégia proposta apresentou tempos computacionais reduzidos, se comparada à simulação do sistema completo quando testada com até dois alimentadores. A partir de três alimentadores, a co-simulação se apresentou como a única estratégia capaz de realizar os estudos propostos, devido ao elevado tempo computacional e problemas de convergência.

Na sequência, a co-simulação também foi testada para o acoplamento entre FMUs e o simulador de redes de distribuição trifásicas, OpenDSS. Foram apresentados os passos para a implementação do processo, sendo corroborados com pseudocódigos. Para a validação da metodologia, no segundo e terceiro estudo de co-simulação foram realizados a co-simulação do sistema de transmissão de 11 barras com os sistemas de distribuição IEEE 34 barras e Ckt5, respectivamente. Em relação a avaliação da impedância característica, conclusões semelhantes ao primeiro estudo foram obtidas.

Todavia, devido a discrepância entre os valores de impedâncias de Thévenin para os sistemas da co-simulação, algumas limitações foram encontradas. Não foram obtidos resultados para quando a impedância característica é próxima à impedância de Thévenin do lado do sistema de distribuição. A baixa regidez da fonte equivalente do alimentador modelado no OpenDSS para a co-simulação, foi considerado como uma das possibilidades para essa limitação. Além disso, uma análise variando-se a impedância característica foi realizada, mostrando que aumentando-se a parte real da impedância característica, oscilações na interface entre os sistemas, podem vir a aparecer. Consequentemente, os valores de NIAE são reduzidos e a precisão dos resultados é afetada.

Contudo a metodologia mostrou-se eficiente do ponto de vista computacional, mas precisa ser avaliada a cada caso, de forma a se obter a impedância característica adequada para aquela situação. No capítulo seguinte serão abordadas outras metodologias que foram consideradas bem sucedidas em termos de eficiência computacional e precisão.

5 METODOLOGIAS DE CO-SIMULAÇÃO

No último capítulo, foi apresentado o método da linha de transmissão fictícia. A metodologia, apesar de ter sido bem sucedida em algumas situações, apresentou limitações em decorrência de sua dependência com a impedância característica. Nesse capítulo serão apresentadas outras metodologias, que mostraram-se precisas para a co-simulação entre ferramentas distintas. As principais motivações da implementação de outras metodologias é a não dependência da impedância característica, e o método alternado para estabilidade. O referido método elimina os atrasos existentes entre as grandezas de interface, observados no método da linha fictícia.

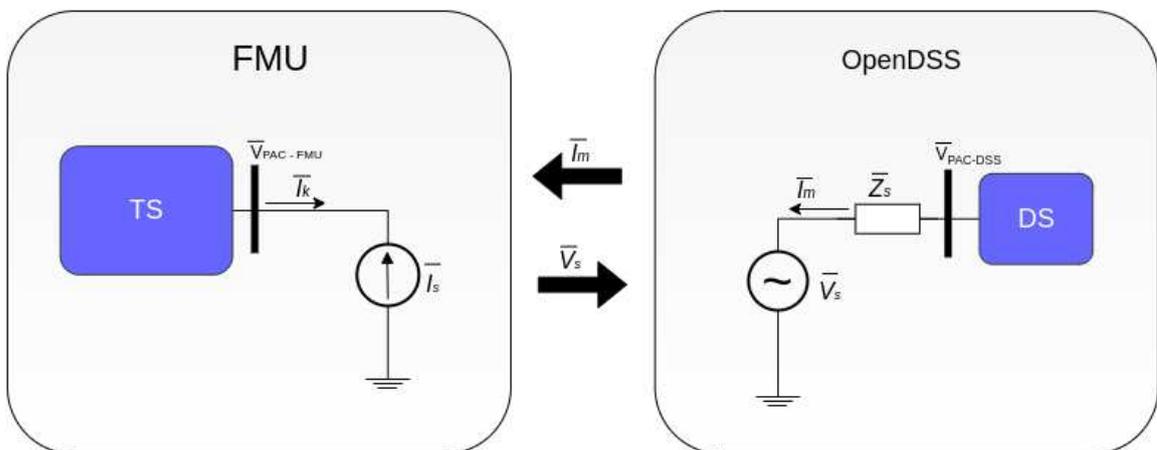
Contudo, antes da apresentação do método alternado, primeiramente será introduzido o método com atraso. Esse é um método eficiente do ponto de vista computacional, porém não é tão preciso como o método alternado, devido ao atraso existente entre as grandezas de interface. Na sequência serão apresentados o método alternado, alternado com sensibilidade e o método alternado "desonesto".

5.1 MÉTODO COM ATRASO

O primeiro método, é o método com atraso. Nessa metodologia, a cada passo de integração, sistemas de transmissão e distribuição são resolvidos separadamente, com a troca de informações entre eles. Assim, há um atraso de comunicação de Δt entre os sistemas envolvidos.

No que diz respeito a proposta de co-simulação desse trabalho, sistemas de transmissão e distribuição são interfaceados diretamente por fontes controladas por corrente e tensão, respectivamente, como mostrado na Figura 24.

Figura 24 – Co-simulação entre FMU e OpenDSS - Método com atraso



Fonte: Elaborado pelo autor (2022)

No processo de co-simulação, em cada passo de integração, ocorre a troca de termos de corrente e tensão entre o sistema de transmissão embutido na FMU e o OpenDSS. Desse modo, o seguinte procedimento pode ser realizado:

1. Após o carregamento da FMU e a leitura do sistema modelado no OpenDSS, realiza-se a inicialização dos modelos, e faz-se $t = 0$.
2. Nessa etapa, as componentes da corrente na interface entre os sistemas, obtidas na inicialização são injetadas nas entradas associadas à FMU. Na sequência, calcula-se o sistema contido na FMU, possibilitando extrair a tensão no ponto de acoplamento comum $\bar{V}_{PAC-FMU}$.
3. Para cada passo de integração, o seguinte procedimento é realizado:
 - a) Com a corrente injetada no sistema de transmissão, calculada no passo anterior, $\bar{I}_m(t - \Delta t)$, calcula-se um passo de integração Δt , através da função `doStep`, que invoca o *solver* embutido na FMU. Nesse passo, é extraída a tensão $\bar{V}_{PAC-FMU}$. Também deve-se verificar se o cálculo do passo foi bem sucedido;
 - b) Resolve-se o fluxo de potência do OpenDSS para o sistema de distribuição, com a tensão calculada no passo anterior $\bar{V}_s(t - \Delta t)$. Esta etapa resulta a corrente drenada da barra de referência \bar{I}_m . A convergência do cálculo do fluxo de potência também deve ser verificada nesse passo;
 - c) Proceder com a aquisição de variáveis de interesse de ambos os sistemas, de transmissão e distribuição, para o passo de integração t .
 - d) Atualiza-se as entradas. Nessa etapa, as componentes da corrente \bar{I}_m (partes real e imaginária) são injetadas nas entradas da FMU, enquanto, no OpenDSS, é feito $\bar{V}_s = \bar{V}_{PAC-FMU} - \bar{Z}_s \bar{I}_s$.
 - e) Incrementar o passo de integração $t \leftarrow t + \Delta t$. Se $t < t_{final}$, retornar ao passo (3a), senão terminar a simulação.

De forma a contribuir com a descrição do processo de co-simulação, o Algoritmo 12 apresenta na forma de uma mescla de pseudocódigo com a linguagem Python, o procedimento de co-simulação entre o OpenModelica e o OpenDSS pelo método com atraso.

Duas vantagens que o método com atraso oferece é quanto a facilidade de implementação e o desempenho computacional. Porém, há outros métodos que são mais precisos, que eliminam o atraso inerente da co-simulação, e serão descritos a seguir.

Algoritmo 12: Co-simulação entre o OpenModelica e OpenDSS - Método com atraso

```

entrada :  $\bar{V}_{s0}, \bar{I}_{m0}$ 
saída : Grandezas de interesse

# Carregamento dos sistemas no ambiente de simulação
fmu_TS ← load_fm(" sistema1.fmu")
opendssdirect.run_command('Redirect sistema2.dss')

# Configurações da simulação
Atribuir valores para  $t_f, \Delta t$ 
 $\bar{I}_s \leftarrow \bar{I}_{m0}$ 

# Inicialização da FMU do sistema de transmissão
fmu_TS.instantiate()
fmu_TS.setup_experiment()
fmu_TS.enter_initialization_mode()
fmu_TS.set("Ir", Real( $\bar{I}_{m0}$ ))
fmu_TS.set("Ii", Imag( $\bar{I}_{m0}$ ))
fmu_TS.time = 0
fmu_TS.exit_initialization_mode()

# Inicialização do sistema modelado no OpenDSS
(Fluxo de potência calculado de acordo com o Algoritmo 7)

# Modo de Integração
enquanto  $t < t_{final}$  faça
  # Solução independente de cada sistema
  status1 ← fmu_TS.do_step(t,  $\Delta t$ )
  status2,  $\bar{I}_m \leftarrow \text{DSS}(\bar{V}_s(t - \Delta t))$ 
  # Verifica se os cálculos do passo e do fluxo de potência foram bem sucedidos
  se  $status1 = \text{pyfmi.fmi.FMI\_ERROR}$  ou  $status2 = \text{False}$  então
    | break

  # Armazenamento das variáveis de interesse
   $V_{PAC-FMU-re} \leftarrow \text{fmu\_TS.get("Vr")}$ 
   $V_{PAC-FMU-im} \leftarrow \text{fmu\_TS.get("Vi")}$ 
   $\bar{V}_{PAC-FMU} \leftarrow V_{PAC-FMU-re} + jV_{PAC-FMU-im}$ 

  # Atualização das entradas
  fmu_TS.set("Ir", Real( $\bar{I}_m$ ))
  fmu_TS.set("Ii", Imag( $\bar{I}_m$ ))
   $\bar{V}_s \leftarrow \bar{V}_{PAC-FMU} - \bar{Z}_s \bar{I}_s$ 
   $\bar{I}_s \leftarrow \bar{I}_m$ 

  # Incremento do tempo de simulação
   $t \leftarrow t + \Delta t$ 

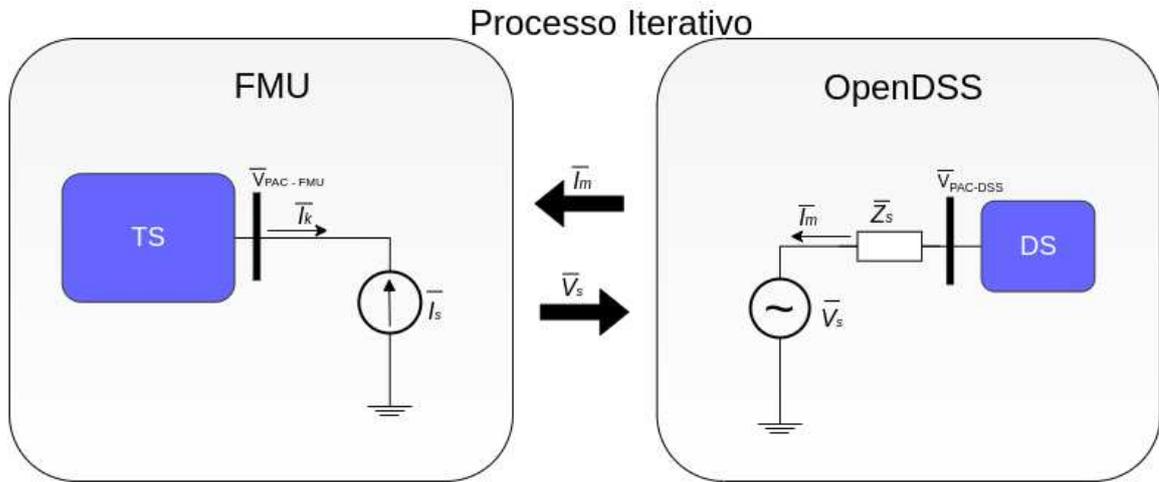
```

5.2 MÉTODO ALTERNADO

O método com atraso, embora seja eficiente do ponto de vista computacional, não é tão preciso devido a existência de um atraso durante a troca de informações entre as variáveis de interface. Outros dois esquemas comumente utilizados em análise de sistemas algébrico-diferencial são: método simultâneo e alternado [44, 45]. Esses são métodos mais precisos em relação ao método com atraso.

Na abordagem simultânea, é aplicado um método de integração numérica sobre as equações diferenciais convertendo-as em equações de diferenças. Posteriormente, essas

Figura 25 – Iteração entre FMU e OpenDSS - Método alternado.



Fonte: Elaborado pelo autor (2022)

equações são combinadas às outras equações algébricas formando um único sistema que deve ser solucionado por meio da aplicação de um método numérico, por exemplo, o método de Newton. Uma desvantagem desse método, é que este é um método mais custoso por iteração. Em contrapartida, no método alternado, as equações diferenciais são resolvidas separadamente das equações algébricas, em cada passo de integração, alternando entre o conjunto de equações, e de forma iterativa, até que algum critério de convergência seja satisfeito [44, 45].

Nesse contexto, nessa seção, será utilizado o método alternado para o acoplamento entre sistemas de transmissão e distribuição. Da mesma forma como para o método com atraso, os sistemas de transmissão e distribuição são interfaceados diretamente por fontes controladas por corrente e tensão, respectivamente, como mostrado na Figura 25.

Durante a co-simulação, diferente do método com atraso, a cada passo de integração um processo iterativo entre a FMU e o OpenDSS é realizado, com finalidade de satisfazer as leis de Kirchhoff tanto de tensão como corrente no ponto de acoplamento comum entre os sistemas de transmissão e distribuição, observando apenas as componentes de sequência positiva. Com isso, é possível eliminar o erro de interface existente no método da linha fictícia. Para essa estratégia, serão considerados os seguintes passos:

1. Após o carregamento da FMU e a leitura do sistema modelado no OpenDSS, realiza-se a inicialização dos modelos, e faz-se $t = 0$.
2. Nessa etapa, as componentes da corrente que flui na interface entre os sistemas, obtidas na inicialização são injetadas nas entradas associadas à FMU. Na sequência, calcula-se o sistema contido na FMU, possibilitando extrair a tensão no ponto de acoplamento comum $\bar{V}_{PAC-FMU}$.

3. Para cada passo de integração, o seguinte procedimento iterativo pode ser realizado:
 - a) Define-se $\bar{V}_s = \bar{V}_{PAC-FMU} - \bar{Z}_s \bar{I}_m$ como a tensão na barra de referência do sistema de distribuição modelado no OpenDSS;
 - b) Resolve-se o fluxo de potência do OpenDSS. Esta etapa resulta a corrente drenada da barra de referência \bar{I}_m .
 - c) A corrente \bar{I}_m é agora injetada na entradas da FMU.
 - d) Utiliza-se, agora, o comando `doStep`, para a solução do sistema contido na FMU para o instante de tempo t atual.
 - e) Extraí-se as tensões da barra de interface do sistema de transmissão, obtendo-se dessa forma $\bar{V}_{PAC-FMU}$.
 - f) Repetir os passos (3a) a (3e) enquanto $\|\bar{V}_{PAC-FMU} - \bar{V}_{PAC-OpenDSS}\| > \varepsilon$, onde ε corresponde à tolerância do processo iterativo.
4. Quando a convergência for atingida, prosseguir para o próximo passo.
5. Calcula-se uma última vez a FMU bem como o fluxo de potência no OpenDSS de forma a compatibilizar a interface. Nesse passo deve-se verificar também se o cálculo do passo de integração e a convergência do cálculo do fluxo de potência foram bem sucedidos.
6. Proceder com a aquisição de variáveis de interesse de ambos os sistemas, de transmissão e distribuição, para o passo de integração t .
7. Incrementar o passo de integração $t \leftarrow t + \Delta t$. Se $t < t_{final}$, retornar ao passo (2), senão terminar a simulação.

De forma a contribuir para a descrição da co-simulação, o Algoritmo 13 apresenta na forma de uma mistura de pseudocódigo com a linguagem Python, o procedimento de co-simulação entre o OpenModelica e o OpenDSS pelo método alternado. A função `calcula_Vs` descrita no Algoritmo 14 realiza o processo iterativo a cada passo de integração, recebendo como entrada, a tensão $\bar{V}_s(t - \Delta t)$ e retornando a tensão \bar{V}_s após a convergência.

Uma das principais vantagens do método alternado é que as soluções de ambos os sistemas, de transmissão e distribuição, são obtidos para um mesmo instante de tempo, tornando, assim, a solução simultânea. Dessa forma, a solução obtida com o método alternado deve ser idêntica à solução do sistema completo, a menos de erros associados à tolerância adotada. Uma vantagem com relação ao método da linha fictícia, e ao método com atraso é a eliminação total do atraso nas variáveis de interface. No entanto, exige um esforço computacional adicional.

Algoritmo 13: Co-simulação entre o OpenModelica e OpenDSS - Método alternado

```

entrada :  $\bar{V}_{s0}, \bar{I}_{m0}$ 
saída : Grandezas de interesse

# Carregamento dos sistemas no ambiente de simulação
fmu_TS ← load_fm(" sistema1.fmu")
opendssdirect.run_command('Redirect sistema2.dss')

# Configurações da simulação
Atribuir valores para  $t_f, \Delta t$ 

# Inicialização da FMU do sistema de transmissão
fmu_TS.instantiate()
fmu_TS.setup_experiment()
fmu_TS.enter_initialization_mode()
fmu_TS.set("Ir", Real( $\bar{I}_{m0}$ ))
fmu_TS.set("Ii", Imag( $\bar{I}_{m0}$ ))
fmu_TS.time = 0
fmu_TS.exit_initialization_mode()

# Inicialização do sistema modelado no OpenDSS
(Fluxo de potência calculado de acordo com o Algoritmo 7)

# Modo de Integração
enquanto  $t < t_{final}$  faça
   $\bar{V}_s, \bar{I}_m = \text{calcula\_Vs}(\bar{V}_s(t - \Delta t))$ 
  fmu.set("Ir", Real( $\bar{I}_m$ ))
  fmu.set("Ii", Imag( $\bar{I}_m$ ))

  # Solução independente de cada sistema
  status1 ← fmu_TS.do_step(t,  $\Delta t$ )
  status2,  $\bar{V}_{PAC-OpenDSS}, \bar{I}_m \leftarrow DSS(\bar{V}_s)$ 

  # Verifica se os cálculos do passo e fluxo de potência foram bem sucedidos
  se status1 = pyfmi.fmi.FMI_ERROR or status2 = False então
    ⊥ break

  # Armazenamento das variáveis de interesse
   $V_r \leftarrow \text{fmu.get}("V_r")$ 
   $V_i \leftarrow \text{fmu.get}("V_i")$ 
   $\bar{V}_{PAC-FMU} \leftarrow V_r + jV_i$ 
   $\bar{V}_s \leftarrow \bar{V}_{PAC-FMU} - \bar{Z}_s \bar{I}_m$ 

  # Incremento o tempo de simulação
   $t \leftarrow t + \Delta t$ 

```

5.3 MÉTODO ALTERNADO COM SENSIBILIDADE

O método alternado, descrito anteriormente, pode incorrer em problemas de convergência. Visando tornar o método ainda mais robusto, o presente trabalho, apresenta a proposta de uma metodologia baseada em sensibilidades extraídas no ponto de acoplamento comum.

De forma a motivar esse novo desenvolvimento, seja, então, uma função vetorial $\mathbf{y} = \mathbf{f}(\mathbf{x})$. Considerando um ponto qualquer \mathbf{x}_0 , pode-se aproximar a variação em \mathbf{y} através

Algoritmo 14: Processo iterativo - Método alternado

Função calcula_Vs(\bar{V}_{s0})

```

entrada :
  •  $\bar{V}_{s0}$ : Estimativa inicial da tensão interna da subestação

saída :
  •  $\bar{V}_s$ : Tensão interna da subestação calculada

# Inicialização das variáveis internas
 $\bar{V}_s \leftarrow \bar{V}_{s0}$ 

# Inicialização do contador de iterações
 $h \leftarrow 0$ 
 $nao\_convergado \leftarrow True$ 

# Solução iterativa para o PAC
enquanto  $nao\_convergado$  faça
   $\bar{V}_{PAC-DSS}, \bar{I}_m \leftarrow DSS(\bar{V}_s)$ 
   $fmu\_TS.set("Ir", Real(\bar{I}_m))$ 
   $fmu\_TS.set("Ii", Imag(\bar{I}_m))$ 
   $fmu\_TS.doStep(t, \Delta t)$ 
   $V_r \leftarrow fmu\_TS.get("Vr")$ 
   $V_i \leftarrow fmu\_TS.get("Vi")$ 
   $\bar{V}_{PAC-FMU} \leftarrow V_r + jV_i$ 
   $nao\_convergado \leftarrow \|\bar{V}_{PAC-DSS} - \bar{V}_{PAC-FMU}\| > \varepsilon$ 
   $\bar{V}_s \leftarrow \bar{V}_{PAC-FMU} - \bar{I}_m \bar{Z}_s$ 
   $h \leftarrow h + 1$ 

retorna  $\bar{V}_s$ 

```

da série de Taylor truncada como denotada em (5.1), onde $\Delta \mathbf{y} = \mathbf{y} - \mathbf{y}_0$ e $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_0$.

$$\Delta \mathbf{y} \approx \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0} \Delta \mathbf{x} \quad (5.1)$$

O termo $\left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0}$, que corresponde à matriz jacobiana da função \mathbf{f} calculada em \mathbf{x}_0 , traduz uma medida de sensibilidade das variáveis \mathbf{y} para uma variação em torno de \mathbf{x}_0 .

Considerando agora que a variável \mathbf{x} esteja associada às componentes da tensão \bar{V}_{PAC} , medida no ponto de acoplamento comum dos sistemas de transmissão e distribuição, pode-se medir a variação da corrente drenada pelo sistema de distribuição através da matriz de admitância nodal vista do ponto de acoplamento comum. Matematicamente, pode-se modelar essa relação através de (5.2) e (5.3), cujos parâmetros são detalhados em (5.4). Nessas expressões, as correntes em \mathbf{i} e tensões em \mathbf{v} são descritas através de vetores cujas componentes correspondem às suas partes real e imaginária, respectivamente,

enquanto a admitância é representada através da condutância G e susceptância B ¹.

$$\mathbf{i} = \mathbf{f}(\mathbf{v}) \quad (5.2)$$

$$\Delta \mathbf{i} = \mathbf{Y} \Delta \mathbf{v} \quad (5.3)$$

$$\mathbf{i} = \begin{bmatrix} I_r \\ I_i \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} V_r \\ V_i \end{bmatrix} \quad \Delta \mathbf{i} = \begin{bmatrix} \Delta I_r \\ \Delta I_i \end{bmatrix} \quad \Delta \mathbf{v} = \begin{bmatrix} \Delta V_r \\ \Delta V_i \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} G & B \\ -B & G \end{bmatrix} \quad (5.4)$$

Comparando-se (5.3) com (5.1), pode-se verificar que a matriz de admitâncias \mathbf{Y} corresponde à matriz jacobiana das componentes da corrente injetada no ponto de acoplamento comum, dadas no vetor \mathbf{i} , que é uma função da tensão do ponto de acoplamento \mathbf{v} , como denotado em (5.2). Assim as relações que definem a condutância G e a susceptância B podem ser verificadas em (5.5) e (5.6), respectivamente.

$$G = \left. \frac{\partial I_r}{\partial V_r} \right|_{V_r, V_i} = \left. \frac{\partial I_i}{\partial V_i} \right|_{V_r, V_i} \quad (5.5)$$

$$B = \left. \frac{\partial I_r}{\partial V_i} \right|_{V_r, V_i} = - \left. \frac{\partial I_i}{\partial V_r} \right|_{V_r, V_i} \quad (5.6)$$

Considerando agora que a variação seja aplicada em torno de uma tensão e corrente iniciais \bar{V}_0 e \bar{I}_0 , a relação (5.4) pode ser reescrita como dado em (5.7).

$$\mathbf{i} = \mathbf{Y} \mathbf{v} - \mathbf{i}_N \quad (5.7a)$$

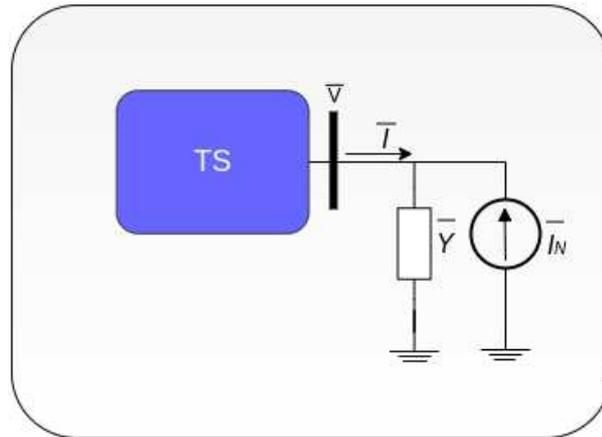
$$\mathbf{i}_N = \mathbf{Y} \mathbf{v}_0 - \mathbf{i}_0 \quad (5.7b)$$

Pode-se observar que as expressões dadas em (5.7) definem um equivalente de Norton, como mostrado na Figura 26. Esse equivalente representa, de forma aproximada, a relação entre a corrente drenada pelo sistema de distribuição e a tensão aplicada ao ponto de acoplamento comum.

Nos casos em que o sistema de distribuição sob estudo for linear, esse equivalente de Norton representa fielmente a dependência entre variações de tensão e corrente no sistema de distribuição vista do ponto de acoplamento comum. Assim, o equivalente de Norton do sistema de distribuição pode ser integrado à solução do cálculo do sistema de transmissão, não havendo necessidade do cálculo iterativo com o sistema de distribuição, como realizado no método alternado, descrito na seção anterior. O processo iterativo ficaria restrito apenas à solução do sistema de transmissão.

¹ Nas análises subsequentes, a susceptância B é considerada positiva para os casos em o sistema analisado seja indutivo.

Figura 26 – Equivalente de Norton.



Fonte: Elaborado pelo autor (2022)

Para situações mais gerais, no entanto, a relação entre corrente e tensão no ponto de acoplamento dos sistemas de distribuição, como mostrado em (5.2), não são lineares. Nessas situações, espera-se que os elementos da matriz de admitância \mathbf{Y} variem, na medida em que a tensão no ponto de acoplamento comum varie. Dessa forma, para o caso não-linear, tanto os parâmetros de condutância G e susceptância B como a injeção de corrente \mathbf{i}_N devem ser recalculados a cada variação da tensão no ponto de acoplamento comum. Essa atualização, em princípio, deve ocorrer a cada variação da tensão no ponto de acoplamento comum dos sistemas de transmissão e distribuição. Ou seja, considerando que os sistemas são calculados dentro de processos iterativos, essa atualização deve, em princípio, ocorrer a cada iteração do processo.

Antes do detalhamento do processo alternado com sensibilidades em si, faz-se necessário detalhar o procedimento adotado para a atualização dos parâmetros do circuito equivalente de Norton que caracteriza o sistema de distribuição a partir de um nó elétrico de interface.

5.3.1 Levantamento do Equivalente de Norton dos Sistemas de Distribuição

Considerando que o sistema de distribuição opera com a sua barra de interface sob tensão $\bar{V}_0 = V_{r0} + jV_{i0}$, drenando a corrente $\bar{I}_0 = I_{r0} + jI_{i0}$, pode-se levantar o seu equivalente de Norton a partir de variações controladas da tensão de interface.

Considerando uma variação somente na parte real da tensão \bar{V}_0 , ou seja, ΔV_r , a partir de (5.3), pode-se obter (5.8). Nessa expressão, a corrente sofre uma variação $\bar{I}_1 = I_{r1} + jI_{i1}$ correspondente à corrente drenada na interface do sistema de distribuição

caso a tensão de interface seja $\bar{V}_0 + \Delta V_r$.

$$I_{r1} - I_{r0} = G \Delta V_r \quad (5.8a)$$

$$I_{i1} - I_{i0} = -B \Delta V_r \quad (5.8b)$$

Dessa forma, a condutância G e a susceptância B podem ser obtidas como mostrado em (5.9).

$$G = \frac{I_{r1} - I_{r0}}{\Delta V_r} \quad (5.9a)$$

$$B = \frac{I_{i0} - I_{i1}}{\Delta V_r} \quad (5.9b)$$

Nas estimativas de G e B dadas em (5.9), a variação ΔV_r pode, em princípio, ser tanto positiva como negativa. Estimativas distintas, no entanto, podem resultar para diferentes variações, principalmente, se equipamentos conectados ao sistema de distribuição estiverem operando próximos de limites que exijam mudança drástica de seu comportamento para pequenas variações de tensão e corrente.

De forma a minimizar os efeitos de não linearidade muito acentuada, uma segunda estratégia de estimativa para as sensibilidades considera variações reais positiva e negativa em torno da \bar{V}_0 . Assim são estabelecidos dois experimentos: um para $\bar{V}_0 + \Delta V_r$ que resulta na corrente $\bar{I}_p = I_{rp} + jI_{ip}$ e, outro para $\bar{V}_0 - \Delta V_r$ que resulta na corrente $\bar{I}_n = I_{rn} + jI_{in}$. Com base em (5.9), pode-se inferir que para ambos os experimentos pode-se escrever as relações dadas em (5.10).

$$I_{rp} - I_{r0} = G \Delta V_r \quad I_{rn} - I_{r0} = -G \Delta V_r \quad (5.10a)$$

$$I_{ip} - I_{i0} = -B \Delta V_r \quad I_{in} - I_{i0} = B \Delta V_r \quad (5.10b)$$

Subtraindo-se as expressões em (5.10a), permite o cálculo da condutância G como mostrado em (5.11a). De forma análoga, a susceptância B é calculada subtraindo-se as expressões em (5.10b).

$$G = \frac{I_{rp} - I_{rn}}{2 \Delta V_r} \quad (5.11a)$$

$$B = \frac{I_{in} - I_{ip}}{2 \Delta V_r} \quad (5.11b)$$

As expressões de G e B , dadas em (5.11), levam em conta variações em dois sentidos distintos ao longo do eixo real para a tensão de interface. Ainda podem minimizar

problemas que possam aparecer devido à presença de forte não-linearidades no sistema de distribuição. Espera-se, no entanto, um comportamento mais bem comportado que das estimativas de sentido único apresentadas em (5.9).

De posse dos parâmetros de condutância G e susceptância B , a admitância \bar{Y} do equivalente de Norton fica completamente definida, restando o cálculo da corrente de Norton \bar{I}_N , dado em (5.12).

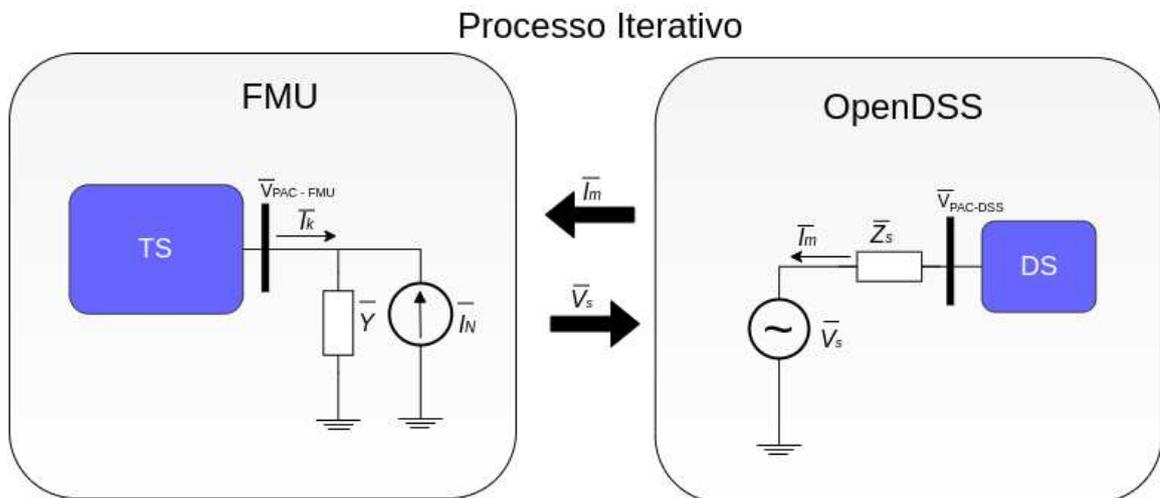
$$\bar{I}_N = \bar{I}_0 - \bar{Y} \bar{V}_0 \quad (5.12)$$

5.3.2 Modo de Integração

Para o método alternado com sensibilidade, uma solução iterativa entre a FMU e o OpenDSS ainda é realizada a cada passo de integração, só que levando em conta o equivalente de Norton inserido no sistema de transmissão modelado no OpenModelica. A Figura 27 apresenta um esquema de interação entre a FMU e o OpenDSS nesse novo contexto. Assim, os passos para o processo iterativo podem ser listados a seguir:

1. Após o carregamento da FMU e a leitura do sistema modelado no OpenDSS, realiza-se a inicialização dos modelos, e faz-se $t = 0$.
2. Dentro do laço de integração, para cada passo de integração, o seguinte processo iterativo pode ser realizado:
 - a) Define-se $\bar{V}_s = \bar{V}_{PAC-FMU} - \bar{I}_m \bar{Z}_s$ como a tensão a ser inserida na barra de referência do sistema de distribuição modelado no OpenDSS;
 - b) Resolve-se o fluxo de potência no OpenDSS. Esta etapa resulta na corrente drenada da barra de referência \bar{I}_m ;

Figura 27 – Interação entre FMU e OpenDSS através da sensibilidade.



- c) Calcula-se o equivalente de Norton associado ao sistema de distribuição (vide subseção anterior). A admitância \bar{Y} pode ser estimada por algum dos métodos detalhados anteriormente (variação positiva, negativa ou central).
 - d) As entradas da FMU, definidas para I_r e I_i , e a admitância equivalente conectada ao nó de interface são atualizadas com o equivalente de Norton estimado;
 - e) Utiliza-se, agora, o comando `doStep`, para a solução do sistema contido na FMU para o instante de tempo t atual.
 - f) Extraí-se as tensões (parte real e imaginária) da barra de interface do sistema de transmissão, obtendo-se dessa forma $\bar{V}_{PAC-FMU}$;
 - g) Repetir os passos (2a) a (2f) até que $\|\bar{V}_{PAC-FMU} - \bar{V}_{PAC-OpenDSS}\| < \varepsilon$, onde ε corresponde à tolerância do processo iterativo.
3. Calcula-se uma última vez a FMU bem como o fluxo de potência no OpenDSS de forma a compatibilizar a interface. Verifica-se também se o cálculo do passo de integração e a convergência do cálculo do fluxo de potência foram bem sucedidos;
 4. Proceder com a aquisição de variáveis de interesse de ambos os sistemas, de transmissão e distribuição, para o passo de integração t .
 5. Incrementar o passo de integração $t \leftarrow t + \Delta t$. Se $t < t_{final}$, retornar ao passo (2a), senão terminar a simulação.

É importante notar que os métodos alternados com e sem sensibilidades apresentam algoritmos praticamente idênticos. A principal diferença é encontrada no cálculo adicional do equivalente de Norton associado ao sistema de distribuição no caso do método com sensibilidades. A inclusão da admitância equivalente do sistema de distribuição no sistema de transmissão tende a melhorar as estimativas obtidas durante o processo iterativo, como observado no método de Newton-Raphson, que apresenta, normalmente, característica de convergência quadrática.

O Algoritmo 15 apresenta forma de pseudocódigos misturado com a linguagem Python, o algoritmo principal do processo de co-simulação com o método baseado em sensibilidades. A função `calcula_Vs` descrita no Algoritmo 16 irá executar o processo iterativo. Os Algoritmo 17 a Algoritmo 19 apresentam as funções utilizadas para os cálculos de sensibilidade, baseadas nos métodos de variação positiva, negativa e central em \bar{V}_0 respectivamente.

5.4 MÉTODO ALTERNADO DESONESTO

Dois importantes aspectos a serem considerados para o processo de co-simulação devem ser a sua eficiência computacional e a precisão.

Algoritmo 15: Co-simulação entre o OpenModelica e OpenDSS - Método alternado com sensibilidade

```

entrada :  $\bar{V}_{s0}, \bar{I}_{m0}$ 
saída : Grandezas de interesse

# Carregamento dos sistemas no ambiente de simulação
fmu_TS ← load_fmú(" sistema1.fmu")
opendssdirect.run_command('Redirect sistema2.dss')

# Configurações da simulação
Atribuir valores para  $t_f, \Delta t$ 

# Inicialização do equivalente de Norton
 $\bar{V}_{PAC} \leftarrow \bar{V}_s + \bar{Z}_s \bar{I}_{m0}$ 
 $\bar{Y} \leftarrow \text{calcula\_Y\_positiva}(\bar{V}_{PAC})$ 
 $\bar{I}_N \leftarrow \bar{I}_{m0} - \bar{Y} \bar{V}_{PAC}$ 

# Inicialização da FMU do sistema de transmissão
fmu_TS.instantiate()
fmu_TS.setup_experiment()
fmu_TS.enter_initialization_mode()
fmu_TS.set("Ir", Real( $\bar{I}_N$ ))
fmu_TS.set("Ii", Imag( $\bar{I}_N$ ))
fmu_TS.set("Y.re", Real( $\bar{Y}$ ))
fmu_TS.set("Y.im", Imag( $\bar{Y}$ ))
fmu_TS.time = 0
fmu_TS.exit_initialization_mode()

# Inicialização do sistema modelado no OpenDSS
(Fluxo de potência calculado de acordo com o Algoritmo 7)

# Modo de Integração
enquanto  $t < t_{final}$  faça
   $\bar{V}_s, \bar{I}_N, \bar{Y} \leftarrow \text{calcula\_Vs}(\bar{V}_s(t - \Delta t))$ 
  fmu_TS.set("Ir", Real( $\bar{I}_N$ ))
  fmu_TS.set("Ii", Imag( $\bar{I}_N$ ))
  fmu_TS.set("Y.re", Real( $\bar{Y}$ ))
  fmu_TS.set("Y.im", Imag( $\bar{Y}$ ))

  # Solução independente de cada sistema
  status1 ← fmu_TS.doStep(t,  $\Delta t$ )
  status2,  $\bar{V}_{PAC-OpenDSS}, \bar{I}_m \leftarrow \text{DSS}(\bar{V}_s)$ 

  # Verifica se os cálculos do passo e fluxo de potência foram bem sucedidos
  se status1 = pyfmi.fmi.FMI_ERROR or status2 = False então
     $\perp$  break

  # Armazenamento das variáveis de interesse
   $V_r \leftarrow \text{fmu.get}("V_r")$ 
   $V_i \leftarrow \text{fmu.get}("V_i")$ 
   $\bar{V}_{PAC-FMU} \leftarrow V_r + jV_i$ 
   $\bar{V}_s \leftarrow \bar{V}_{PAC-FMU} - \bar{Z}_s \bar{I}_m$ 

  # Incremento do tempo de simulação
   $t \leftarrow t + \Delta t$ 

```

Algoritmo 16: Processo iterativo - Método alternado com sensibilidade

```

Função calcula_Vs( $\bar{V}_{s0}$ )
  entrada :
    •  $\bar{V}_{s0}$ : Estimativa inicial da tensão interna da fonte equivalente

  saída :
    •  $\bar{V}_s$ : Tensão interna da subestação calculada
    •  $\bar{I}_N$ : Corrente de Norton calculada
    •  $\bar{Y}$ : Admitância de Norton calculada

  # Inicialização das variáveis internas
   $\bar{V}_s \leftarrow \bar{V}_{s0}$ 

  # Inicialização do contador de iterações
   $h \leftarrow 0$ 
   $nao\_convergado \leftarrow True$ 

  # Solução iterativa para o PAC
  enquanto  $nao\_convergado$  faça
     $\bar{V}_{PAC-DSS}, \bar{I}_m \leftarrow DSS(\bar{V}_s)$ 
     $\bar{Y} \leftarrow calcula\_Y\_(\bar{V}_{PAC-DSS})$ 
     $\bar{I}_N \leftarrow \bar{I}_m - \bar{Y}\bar{V}_{PAC-DSS}$ 
     $fmu\_TS.set("Ir", Real(\bar{I}_N))$ 
     $fmu\_TS.set("Ii", Imag(\bar{I}_N))$ 
     $fmu\_TS.set("Y.re", Real(\bar{Y}))$ 
     $fmu\_TS.set("Y.im", Imag(\bar{Y}))$ 
     $fmu\_TS.doStep(t_i, \Delta t)$ 
     $V_r \leftarrow fmu\_TS.get("Vr")$ 
     $V_i \leftarrow fmu\_TS.get("Vi")$ 
     $\bar{V}_{PAC-FMU} \leftarrow V_r + jV_i$ 
     $nao\_convergado \leftarrow \|\bar{V}_{PAC-DSS} - \bar{V}_{PAC-FMU}\| > \varepsilon$ 
     $\bar{V}_s \leftarrow \bar{V}_{PAC-FMU} - \bar{I}_m\bar{Z}_s$ 
     $h \leftarrow h + 1$ 

  retorna  $\bar{V}_s, \bar{I}_N, \bar{Y}$ 
  
```

Na seção anterior, foi detalhado o método alternado com inclusão de sensibilidades para a solução do sistema de transmissão. Pode-se traçar um paralelo entre esse método e o de Newton-Raphson, onde a admitância equivalente é atualizada a cada iteração do processo iterativo. Porém, pode-se identificar que o método alternado com sensibilidades não corresponde exatamente ao método de Newton-Raphson, já que a inclusão da sensibilidade ocorre somente na interface do sistema de transmissão.

O cálculo da sensibilidade é importante para a convergência. A raiz de uma função é calculada de forma exata, dentro de uma dada tolerância, e normalmente, o processo deve partir de uma boa estimativa inicial. A precisão do método de sensibilidade necessita de aproximações para \bar{Y} a cada iteração entre as ferramentas que compõe a co-simulação, de forma que uma tensão comum entre os sistemas seja atingida.

No que diz respeito a eficiência, o método de sensibilidade pode aumentar o tempo computacional, apesar da precisão ser aumentada. Um procedimento comumente utilizado, seria atualizar a admitância, somente em algumas situações, como diante da mudança da

Algoritmo 17: Função para cálculo da admitância Y - Variação positiva

Função calcula_Y_positiva(\bar{V}_0)

entrada :

- \bar{V}_0 : Tensão no PAC do sistema de distribuição

saída :

- \bar{Y} : Admitância do equivalente de Norton

$$\bar{I}_0 \leftarrow DSS(\bar{V}_0)$$

$$\bar{I}_1 \leftarrow DSS(\bar{V}_0 + \Delta V_r)$$

$$G \leftarrow \frac{I_{r1} - I_{r0}}{\Delta V_r}$$

$$B \leftarrow \frac{I_{i0} - I_{i1}}{\Delta V_r}$$

$$\bar{Y} \leftarrow G - jB$$

retorna \bar{Y}

Algoritmo 18: Função para cálculo da admitância Y - Variação negativa

Função calcula_Y_negativa(\bar{V}_0)

entrada :

- \bar{V}_0 : Tensão no PAC do sistema de distribuição

saída :

- \bar{Y} : Admitância do equivalente de Norton

$$\bar{I}_0 \leftarrow DSS(\bar{V}_0)$$

$$\bar{I}_1 \leftarrow DSS(\bar{V}_0 - \Delta V_r)$$

$$G \leftarrow \frac{I_{r1} - I_{r0}}{\Delta V_r}$$

$$B \leftarrow \frac{I_{i0} - I_{i1}}{\Delta V_r}$$

$$\bar{Y} \leftarrow G - jB$$

retorna \bar{Y}

Algoritmo 19: Função para cálculo da admitância Y - Variação central

Função calcula_Y_central(\bar{V}_0)

entrada :

- \bar{V}_0 : Tensão no PAC do sistema de distribuição

saída :

- \bar{Y} : Admitância do equivalente de Norton

$$\bar{I}_p \leftarrow DSS(\bar{V}_0 + \Delta V_r)$$

$$\bar{I}_n \leftarrow DSS(\bar{V}_0 - \Delta V_r)$$

$$G \leftarrow \frac{I_{rp} - I_{rn}}{2\Delta V_r}$$

$$B \leftarrow \frac{I_{ip} - I_{in}}{2\Delta V_r}$$

$$\bar{Y} \leftarrow G - jB$$

retorna \bar{Y}

topologia do sistema, ou quando o número de iterações aumentar demasiadamente. Essa estratégia é conhecida como Newton. Esse é um método que pode possibilitar ganho em eficiência computacional, conservando-se a precisão [46].

Na metodologia aplicada à co-simulação, pode-se alcançar a mesma ideia do Newton Desonesto mantendo-se a admitância equivalente do sistema de distribuição constante no

método alternado com sensibilidade, a menos que ocorram mudanças na topologia na rede ou o número de iterações cresça de forma excessiva. O critério estabelecido no presente nível de desenvolvimento foi a atualização da admitância quando o número de iterações for maior que dois.

O Algoritmo 20 apresenta a função para o processo iterativo baseado no método de Alternado "desonesto". Essa função, recebe como argumento, além da tensão $\bar{V}_s(t - \Delta t)$, a admitância $\bar{Y}(t - \Delta t)$ calculada no passo anterior.

Algoritmo 20: Processo iterativo - Método alternado desonesto

Função calcula_vs(\bar{V}_s, \bar{Y}_0)

entrada :

- \bar{V}_{s0} : Estimativa inicial da tensão interna da subestação
- \bar{Y}_{s0} : Estimativa inicial para a admitância de Norton

saída :

- \bar{V}_s : Tensão interna da subestação calculada
- \bar{I}_N : Corrente de Norton calculada
- \bar{Y} : Admitância de Norton calculada

Inicialização das variáveis internas

$\bar{V}_s \leftarrow \bar{V}_{s0}$

Inicialização do contador de iterações

$h \leftarrow 0$

$nao_convergado \leftarrow True$

Solução iterativa para o PAC

enquanto $nao_convergado$ **faça**

$\bar{V}_{PAC-DSS}, \bar{I}_m \leftarrow DSS(\bar{V}_s)$

se $h \geq 2$ **então**

$\bar{Y} \leftarrow calcula_Y(\bar{V}_{PAC-DSS})$

senão

$\bar{Y} \leftarrow \bar{Y}_0$

$\bar{I}_N \leftarrow \bar{I}_m - \bar{Y} \bar{V}_{PAC-DSS}$

$fmu_TS.set("Ir", Real(\bar{I}_N))$

$fmu_TS.set("Ii", Imag(\bar{I}_N))$

$fmu_TS.set("Y.re", Real(\bar{Y}))$

$fmu_TS.set("Y.im", Imag(\bar{Y}))$

$fmu_TS.doStep(t, \Delta t)$

$V_r \leftarrow fmu_TS.get("Vr")$

$V_i \leftarrow fmu_TS.get("Vi")$

$\bar{V}_{PAC-FMU} \leftarrow V_r + jV_i$

$nao_convergado \leftarrow \|\bar{V}_{PAC-DSS} - \bar{V}_{PAC-FMU}\| > \varepsilon$

$\bar{V}_s \leftarrow \bar{V}_{PAC-FMU} - \bar{I}_m \bar{Z}_s$

$h \leftarrow h + 1$

retorna $\bar{V}_s, \bar{I}_N, \bar{Y}$

5.5 RESULTADOS DE CO-SIMULAÇÃO

Essa seção tem por objetivo apresentar resultados obtidos para todas metodologias discutidas nesse capítulo. Serão apresentados resultados para a co-simulação entre os sistemas de transmissão de 11 barras e o de distribuição IEEE 34 barras, como também, entre o sistema de transmissão citado anteriormente e o sistema de distribuição Ckt5. Para todas as metodologias serão apresentados resultados da co-simulação comparados aos do sistema de transmissão de 11 barras com apenas cargas de potência constantes. O passo de integração adotado foi de $\Delta t = 1$ [ms].

5.5.1 Co-simulação entre sistema de transmissão e IEEE 34 barras

Quanto ao primeiro estudo de caso desse capítulo, os resultados de co-simulação serão apresentados na sequência. Em relação à inicialização, a Seção B.2 apresenta os valores iniciais obtidos da solução conjunta dos sistemas envolvidos na co-simulação.

5.5.1.1 Método com atraso

A Figura 28(a) mostra o ângulo relativo de G_1 em relação a G_4 , enquanto as Figura 28(b) e Figura 28(c) mostram a tensão e potência fluindo através da barra de interface.

Pode ser observado que os comportamentos das grandezas ficaram bem coerentes com a simulação completa realizada no OpenModelica. Um pequeno atraso entre as grandezas pode ser observado, devido à característica da metodologia. A Figura 29 apresenta a potência ativa, reativa e corrente fluindo do sistema de transmissão para o alimentador do IEEE 34 barras. A comparação foi feita entre grandezas medidas do lado do sistema de transmissão e do sistema modelado no OpenDSS.

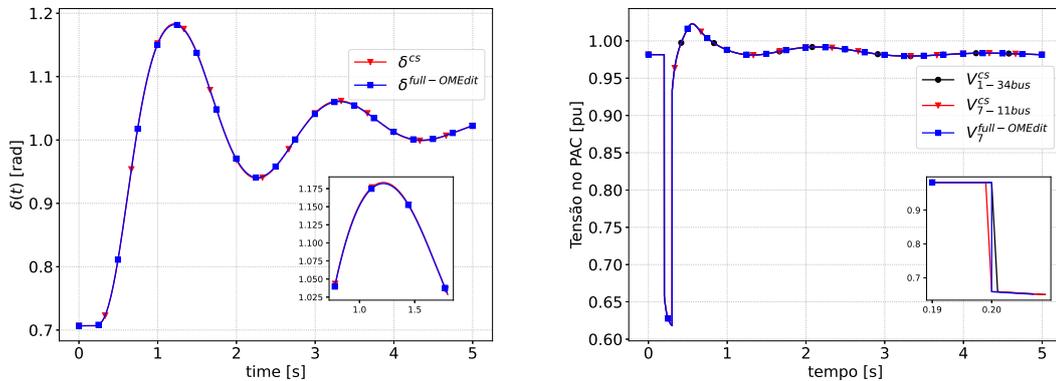
Da mesma forma, as grandezas extraídas na interface ficaram bem comportadas. Nos instantes de descontinuidade, foram observados picos de potência reativa demandada do alimentador. Esse comportamento ocorre devido ao atraso existente na metodologia adotada.

5.5.1.2 Método alternado

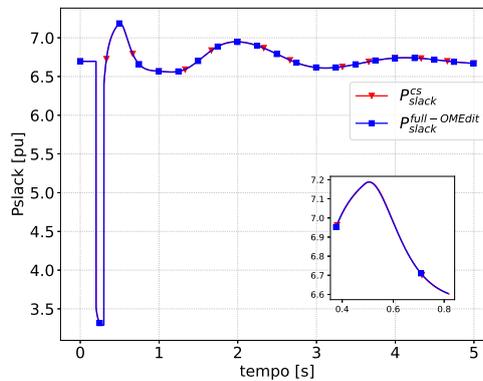
Para esse método, a tolerância foi definida em 10^{-5} [pu]. A Figura 30(a) mostra o ângulo relativo de G_1 em relação a G_4 , enquanto as Figuras 30(b) e 30(c) mostram a tensão e potência fluindo através da barra de interface.

A Figura 31 apresenta a potência ativa, reativa e corrente fluindo do sistema de transmissão para o alimentador de 34 barras. A comparação foi feita entre grandezas medidas do lado do sistema de transmissão e do sistema modelado no OpenDSS.

Figura 28 – Grandezas medidas - Sistemas de transmissão e IEEE 34 barras - Método com atraso

(a) Ângulo relativo entre G_1 e G_4

(b) Tensão na barra de interface



(c) Potência na barra de referência

Fonte: Elaborado pelo autor (2022)

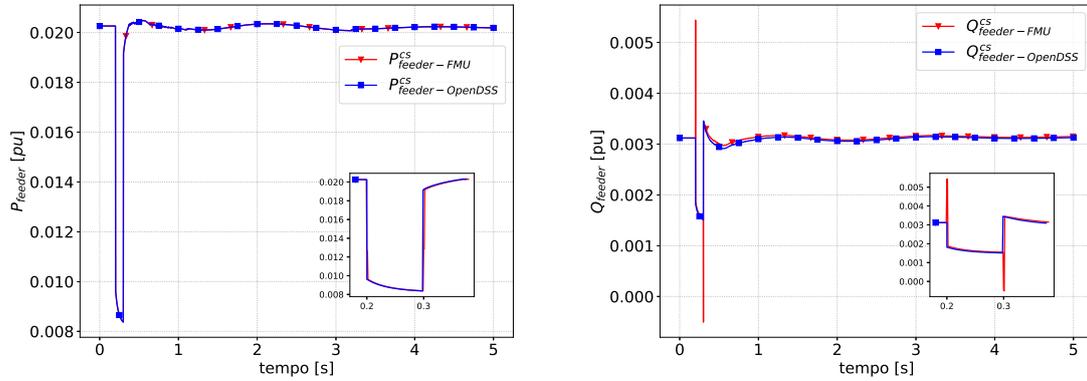
Pode ser observado que os comportamentos das grandezas, novamente ficaram bem coerentes com a simulação completa realizada no OpenModelica. Além disso, em comparação com o método com atraso, picos associados as discontinuidades foram eliminados.

De forma a ilustrar o comportamento numérico do método, a Figura 32(a) apresenta o número de iterações por passo de integração do processo iterativo entre o OpenDSS e o sistema de transmissão interfaceado pela fonte de corrente. Como visto, o processo iterativo não levou mais que três iterações para garantir um erro de tensão de interface menor que 10^{-5} [pu].

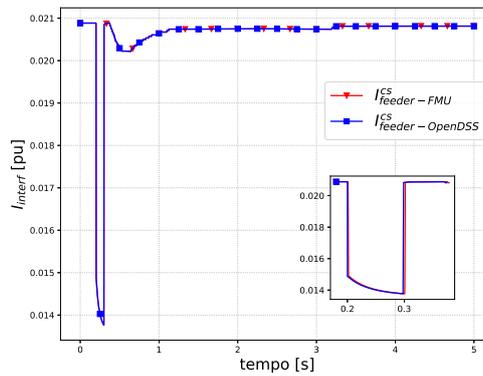
5.5.1.3 Método alternado com sensibilidade

Nessa metodologia, o sistema de distribuição é representado através de um equivalente de Norton para o sistema de transmissão a cada iteração do processo iterativo entre OpenDSS e OpenModelica. Para o cálculo da admitância, foi considerada uma variação na parte real da tensão aplicada ao ponto de acoplamento comum do sistema de

Figura 29 – Grandezas medidas na interface entre sistema de transmissão e IEEE 34 barras - Método com atraso



(a) Potência ativa drenada do sistema de transmissão para o alimentador (b) Potência reativa drenada do sistema de transmissão para o alimentador



(c) Corrente fluindo do sistema de transmissão para o alimentador

Fonte: Elaborado pelo autor (2022)

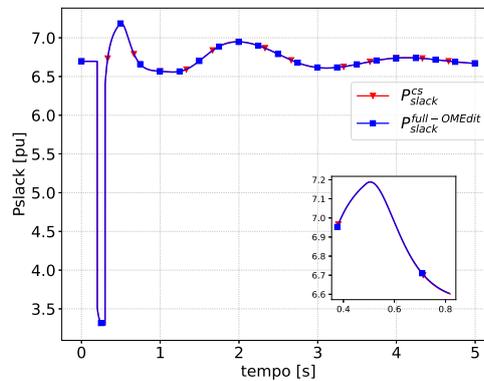
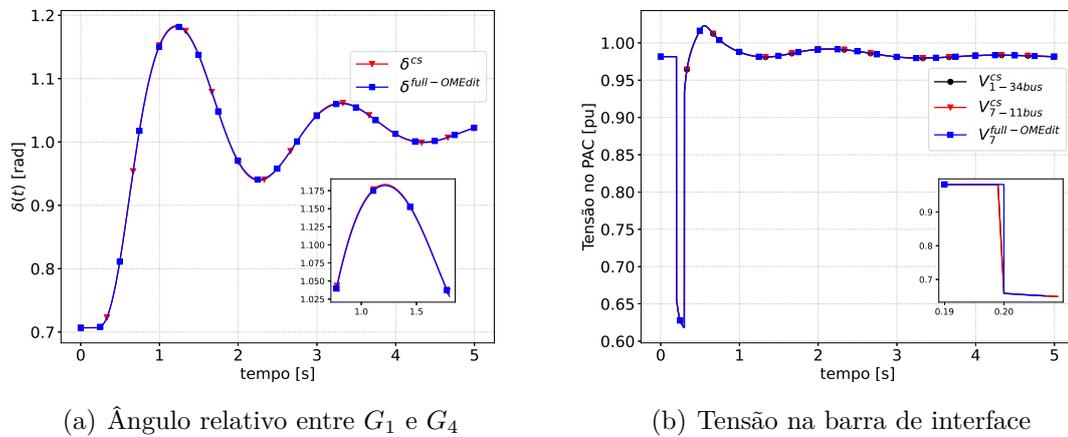
distribuição $\Delta V_r = 10^{-3}$ [pu]. A Tabela 16 apresenta os valores de admitância para uma variação positiva, negativa e central a partir de \bar{V}_0 . O valor de \bar{V}_0 obtido na inicialização é $\bar{V}_0 = 0,951257 + j0,242173$ [pu]. Comparando-se as três estimativas de sensibilidade, pode ser observado que os valores de sensibilidade ficaram muito próximos.

Tabela 16 – Estimativa de admitância do Equivalente de Norton - IEEE 34 barras.

Métodos	\bar{Y} [pu]
Positivo	0,000324 + j0,005696
Negativo	0,000307 + j0,005691
Central	0,000317 + j0,005692

Fonte: Elaborado pelo autor (2022)

Figura 30 – Grandezas medidas - Sistemas de transmissão e IEEE 34 barras - Método alternado



(c) Potência na barra de referência

Fonte: Elaborado pelo autor (2022)

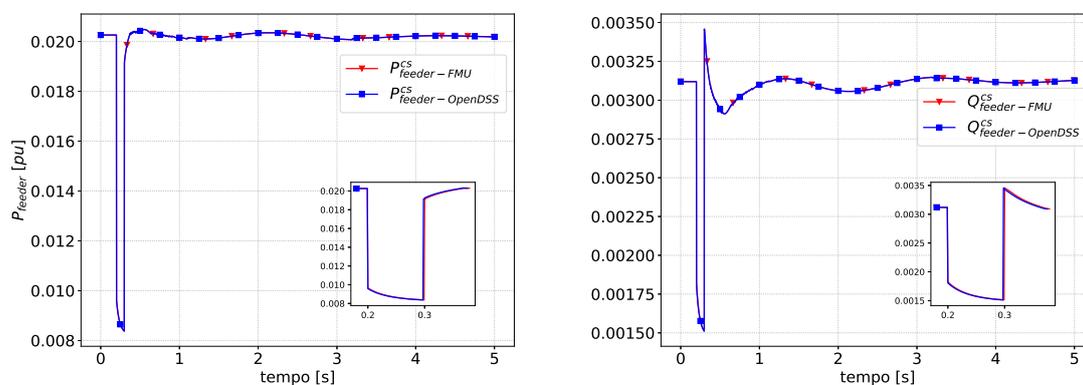
Os comportamentos das grandezas obtidas através da presente metodologia são idênticos aos mostrados na Figura 30, ratificando a coerência entre resultados da co-simulação e do sistema completo.

Já a Figura 32(b) apresenta o número de iterações requeridos por passo de integração do processo iterativo entre o OpenDSS e o sistema de transmissão interfaceado pelo equivalente de Norton. Nota-se que o processo iterativo não levou mais que três iterações, sendo que o maior número de iterações ocorreu nos instantes de aplicação e eliminação da falta, seguida da abertura da linha. Em comparação com o método alternado, não apresentou diferença quanto ao número de iterações.

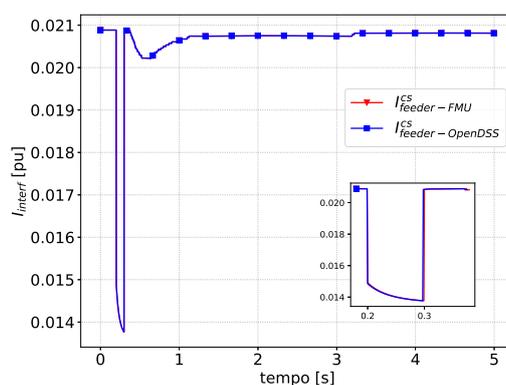
5.5.1.4 Método alternado "desonesto" com sensibilidade

Nessa metodologia, mantém-se a admitância equivalente de Norton, referente ao sistema de distribuição, constante ao longo da co-simulação, sendo atualizada apenas quando o número de iterações do processo iterativo for maior que dois. A corrente

Figura 31 – Grandezas medidas na interface entre sistema de transmissão e IEEE 34 barras - Método alternado



(a) Potência ativa drenada do sistema de transmissão para o alimentador (b) Potência reativa drenada do sistema de transmissão para o alimentador



(c) Corrente fluindo do sistema de transmissão para o alimentador

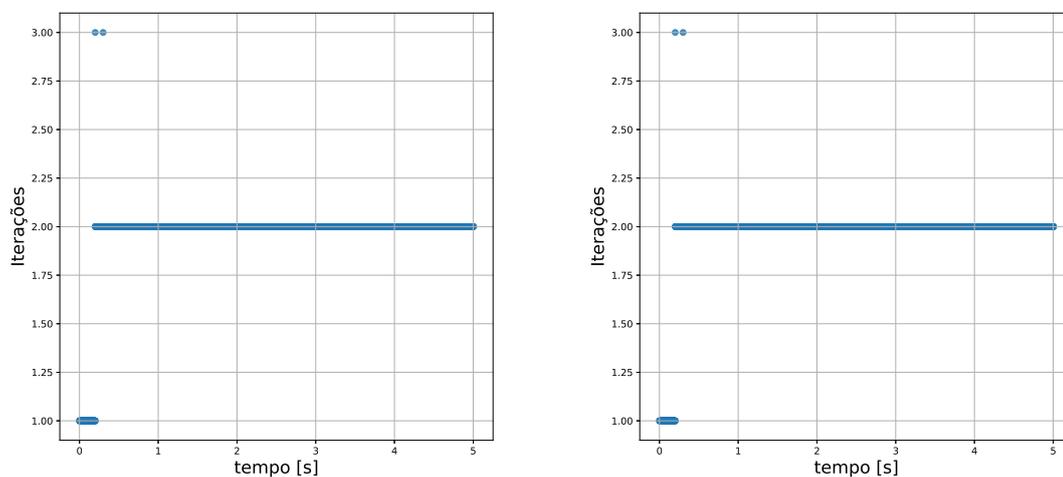
Fonte: Elaborado pelo autor (2022)

equivalente de Norton continua a ser calculada a cada iteração do processo iterativo.

Adotando essa estratégia, pode-se verificar a Figura 33(a) que apresenta o número de iterações de cada processo iterativo por passo, enquanto a Figura 33(b) apresenta o número de atualizações da admitância equivalente por passo de integração. Dos resultados extraídos, o número máximo de iterações observado é três, ocorrendo exatamente nos instantes de descontinuidade. Assim, foram obtidas somente duas atualizações da impedância durante todo o período de simulação. Em comparação com métodos alternado, e alternado com sensibilidade, o número de iterações por passo de integração se manteve. Pode-se então concluir que, o método foi eficaz para o processo de co-simulação, conservando, simultaneamente, eficiência e precisão da solução.

A atualização da admitância pode ser evidenciada também pelas Figura 34(a), Figura 34(b) que representam respectivamente o comportamento da condutância e suscep-

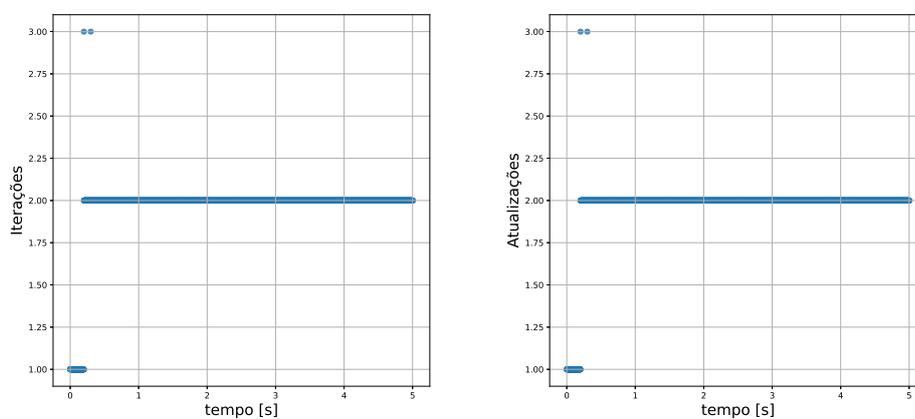
Figura 32 – Medidas de desempenho do método alternado



(a) Número de iterações por passo do método alternado (b) Número de iterações por passo do método alternado com sensibilidade

Fonte: Elaborado pelo autor (2022)

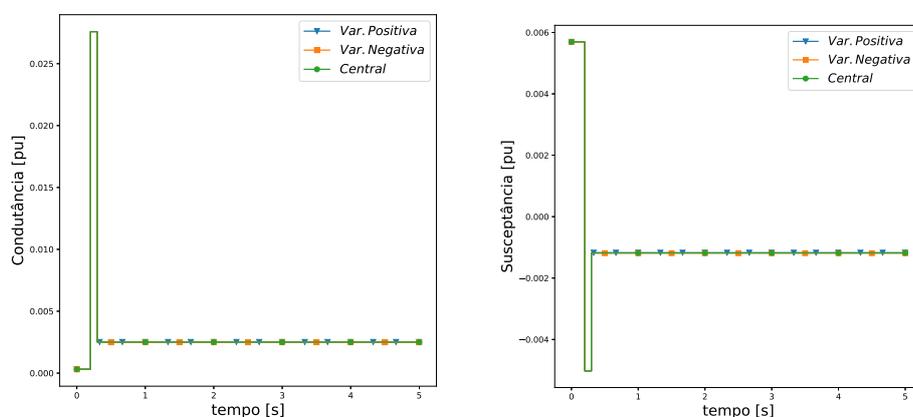
Figura 33 – Medidas de desempenho do método alternado "desonesto" com sensibilidade - Sistemas de transmissão de 11 barras e distribuição de 34 barras



(a) Número de iterações por passo. (b) Número de atualizações por passo.

Fonte: Elaborado pelo autor (2022)

Figura 34 – Comportamento da admitância - Método alternado "desonesto".



(a) Comportamento da condutância. (b) Comportamento da susceptância.

Fonte: Elaborado pelo autor (2022)

Tabela 17 – Perfil da co-simulação - Sistemas de transmissão e IEEE-34 barras (tempos em [s]).

Discriminação	Tempo de execução FMU	Tempo de execução OpenDSS	Proc. Variáveis	Tempo de execução Total
Linha fictícia	3,9	1,5	3,6	9
Método com atraso	3,7	0,7	2,6	7
Método Alternado	8,8	1,8	2,8	14
Sensibilidade - Positiva	10,9	3,9	2,8	19
Sensibilidade - Negativa	11,2	4,1	2,9	20
Sensibilidade - Central	11,9	4,9	3,0	21
Alternado "desonesto" - Positiva	11,6	2,0	2,9	18
Alternado "desonesto" - Negativa	11,0	1,8	2,8	17
Alternado "desonesto" - Central	11,5	2,0	3,0	18

Fonte: Elaborado pelo autor (2022)

tância ao longo da co-simulação.

Do comportamento extraído, pode ser observado que, novamente a admitância foi atualizada duas vezes. Além disso, os valores de sensibilidade obtidos para cada método, foram muito próximos entre si.

5.5.2 Análise de Desempenho Computacional

Um resumo dos tempos computacionais de cada metodologia empregada na realização da co-simulação entre a FMU dedicada ao sistema de transmissão de 11 barras e o sistema de 34 barras modelado no OpenDSS é apresentado na Tabela 17.

Tabela 18 – NIAE - Sistemas de transmissão e IEEE-34 barras (Valores em [%]).

Discriminação	δ_{14}	V_{bus7}	P_{slack}
Linha fictícia	99,785	99,939	99,895
Método com atraso	99,876	99,946	99,905
Método Alternado	99,876	99,943	99,899
Sensibilidade - Positiva	99,877	99,943	99,899
Sensibilidade - Negativa	99,877	99,943	99,899
Sensibilidade - Central	99,877	99,943	99,899
Alternado "desonesto" - Positiva	99,877	99,943	99,899
Alternado "desonesto" - Negativa	99,877	99,943	99,899
Alternado "desonesto" - Central	99,877	99,943	99,899

Fonte: Elaborado pelo autor (2022)

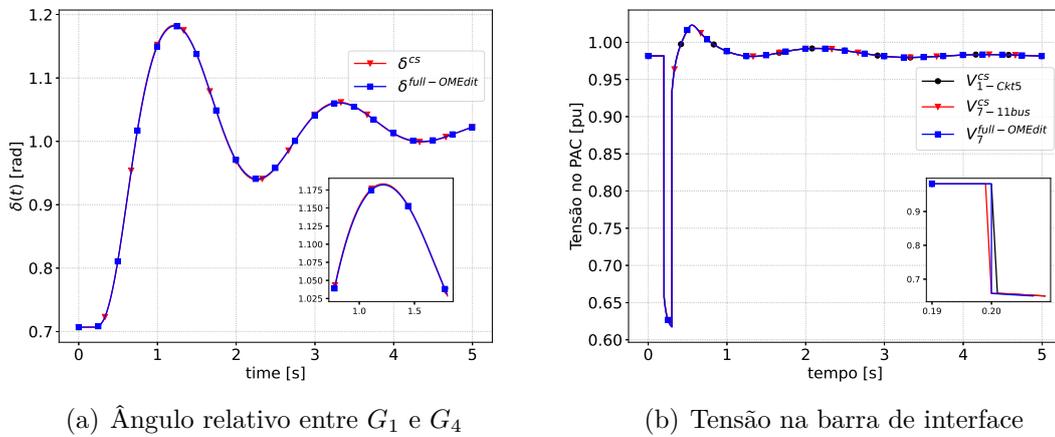
Pode-se perceber que o tempo de execução para as funções inerentes ao OpenDSS é, em geral, muito menor que o tempo gasto pela FMU. O menor tempo de execução total foi observado para o método com atraso que conta com o menor número de chamadas do OpenDSS. Na sequência, o método da linha fictícia apresentou menor tempo computacional em relação aos métodos alternado, que são mais precisos. Os métodos que demandam cálculos de sensibilidade, por outro lado, acabam apresentando os maiores tempos computacionais, devido à necessidade de atualização do equivalente do sistema de distribuição a cada iteração, realizada via chamadas extras do OpenDSS. Comparando-se os métodos baseados em sensibilidades, observa-se que não foram registradas variações significativas entre os tempos computacionais. Como esperado, os tempos de execução total para os métodos alternado "desonesto", foram menores em relação aos demais métodos com sensibilidade. Pode-se verificar uma redução de aproximadamente 15% em relação aos métodos de sensibilidade. Em relação a precisão dos resultados, a Tabela 18 compara os valores obtidos pelo critério NIAE para todas metodologias implementadas para o presente estudo de caso.

Essa redução deve-se à atualização da admitância equivalente ser realizada somente em instantes de descontinuidade. Pode se observar que os valores do critério NIAE foram superiores ao valor de referência, mostrando assim que todas as metodologias foram adequadas para esse estudo. Além disso, o método alternado conservou a precisão dos resultados.

5.5.3 Co-simulação entre Sistema de Transmissão e Ckt5

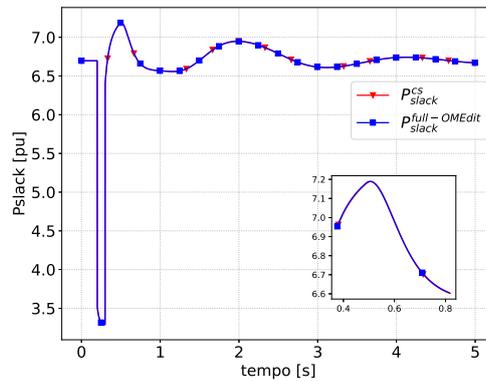
A seguir serão apresentados resultados da co-simulação entre o sistema de transmissão de 11 barras e o sistema de distribuição Ckt5. Como feito anteriormente, os resultados apresentados na sequência são comparados com os obtidos para a simulação do sistema de transmissão de 11 barras com apenas cargas de potência constante. Acerca

Figura 35 – Grandezas medidas - Co-simulação entre sistema de transmissão e distribuição Ckt5 - Método com atraso



(a) Ângulo relativo entre G_1 e G_4

(b) Tensão na barra de interface



(c) Potência na barra de referência

Fonte: Elaborado pelo autor (2022)

da inicialização, a Seção B.2 apresenta os valores iniciais obtidos da solução conjunta dos sistemas envolvidos na co-simulação.

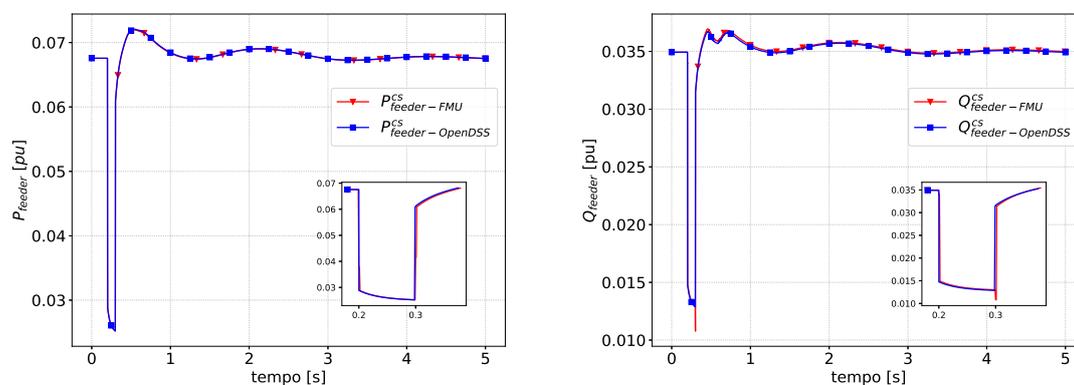
5.5.3.1 Método com atraso

A Figura 35(a) mostra o ângulo relativo de G_1 em relação a G_4 , enquanto as Figura 35(b) e Figura 35(c) mostram a tensão e potência fluindo através da barra de interface.

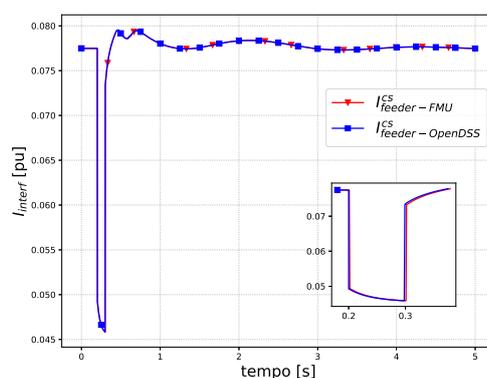
A Figura 36 apresenta a potência ativa, reativa e corrente fluindo do sistema de transmissão para o alimentador Ckt5. A comparação foi feita entre grandezas medidas do lado do sistema de transmissão e do sistema modelado no OpenDSS.

Da mesma forma que os resultados para a co-simulação com o sistema IEEE 34 barras, os comportamentos das grandezas analisadas ficaram bem próximos daqueles verificados para a simulação do sistema completo. Um pico no instante de aplicação da

Figura 36 – Grandezas medidas na interface entre sistema de transmissão e o Ckt5 - Método com atraso



(a) Potência ativa drenada do sistema de transmissão para o alimentador (b) Potência reativa drenada do sistema de transmissão para o alimentador



(c) Corrente fluindo do sistema de transmissão para o alimentador

Fonte: Elaborado pelo autor (2022)

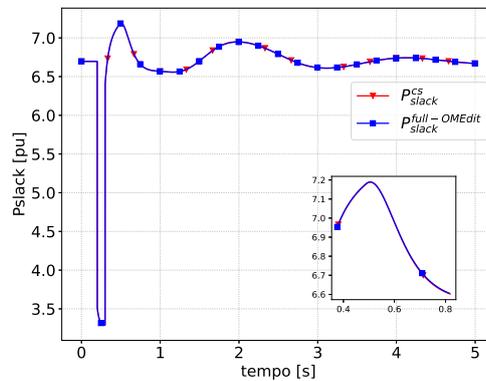
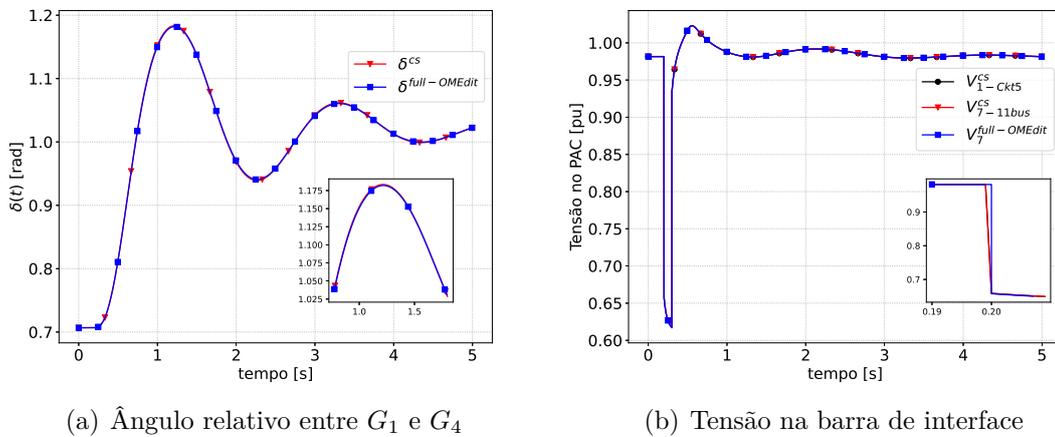
falta pode ser observado, em decorrência do atraso existente na metodologia.

5.5.3.2 Método alternado

A Figura 37(a) mostra o ângulo relativo de G_1 em relação a G_4 , enquanto as Figuras 37(b) e 37(c) mostram a tensão e potência fluindo através da barra de interface. Da mesma forma que os resultados para a co-simulação com o sistema IEEE 34 barras, os comportamentos das grandezas analisadas ficaram bem próximos daqueles verificados para a simulação do sistema completo.

A Figura 38(a), Figura 38(b) e Figura 38(c) apresentam a potência ativa, reativa e corrente fluindo do sistema de transmissão para o alimentador Ckt5. Na interface entre os sistemas, o erro de interface foi eliminado, e nenhum pico foi observado nos instantes de descontinuidades.

Figura 37 – Grandezas medidas - Sistema de transmissão e Ckt5 - Método alternado



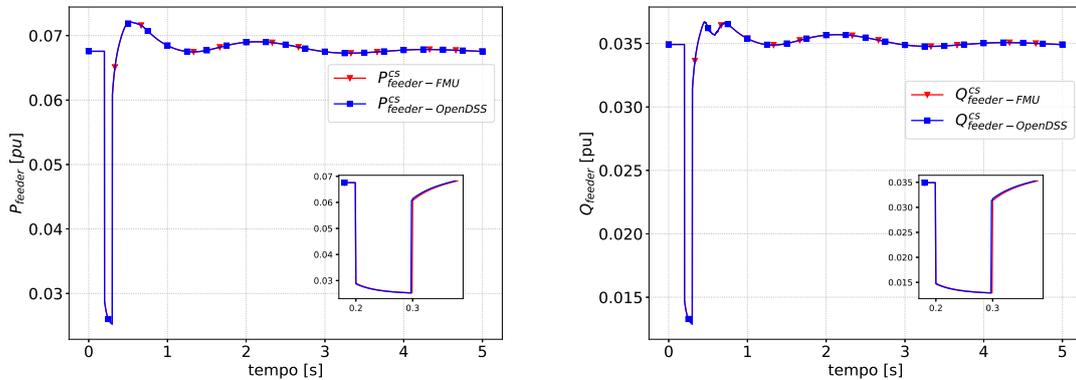
Fonte: Elaborado pelo autor (2022)

A Figura 39(a) apresenta o número de iterações por passo de integração do processo iterativo entre o OpenDSS e o sistema de transmissão interfaceado pela fonte de corrente. Pode-se observar que o processo iterativo não tomou mais que três iterações para convergir, sendo o maior número de iterações verificado nos instantes de aplicação e eliminação da falta ocorrida no sistema de transmissão.

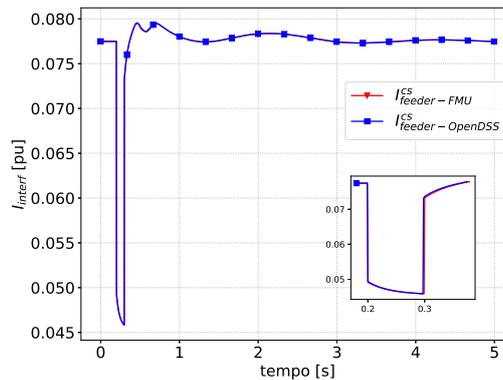
5.5.3.3 Método alternado com sensibilidade

Nessa metodologia, o sistema de distribuição é representado através de um equivalente de Norton para o sistema de transmissão a cada iteração do processo iterativo entre OpenDSS e OpenModelica. Para o cálculo da admitância, foi considerada uma variação na parte real da tensão aplicada ao ponto de acoplamento comum do sistema de distribuição $\Delta V_r = 10^{-3}$ [pu]. A Tabela 19 apresenta os valores de admitância para uma variação positiva, negativa e central a partir de $\bar{V}_0 = 0,951257 + 0,242173j$ [pu]. Pode-se observar, mais uma vez, que o processo iterativo não tomou mais que três iterações para convergir,

Figura 38 – Grandezas medidas na interface entre sistema de transmissão e o Ckt5 - Método alternado



(a) Potência ativa drenada do sistema de transmissão para o alimentador (b) Potência reativa drenada do sistema de transmissão para o alimentador



(c) Corrente fluindo do sistema de transmissão para o alimentador

Fonte: Elaborado pelo autor (2022)

sendo o maior número de iterações verificado, novamente, nos instantes de aplicação e eliminação da falta ocorrida no sistema de transmissão.

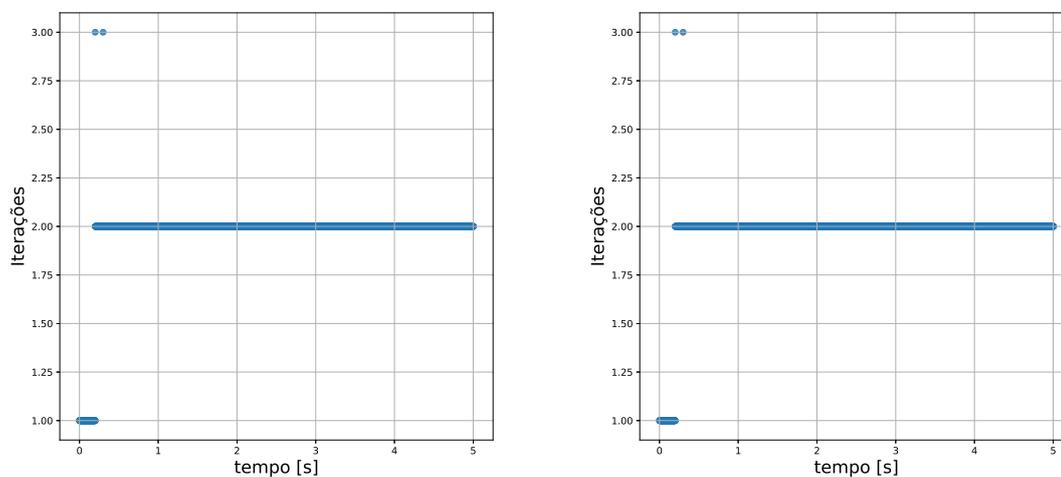
Os comportamentos das grandezas obtidas através da presente metodologia são idênticos aos mostrados na Figura 38, ratificando a coerência entre resultados da co-

Tabela 19 – Estimativa de admitância do Equivalente de Norton - Ckt5.

Métodos	\bar{Y} [pu]
Positivo	0,0779 - j0,0386
Negativo	0,0779 - j0,0387
Central	0,0779 - j0,0386

Fonte: Elaborado pelo autor (2022)

Figura 39 – Medidas de desempenho do método alternado - Sistemas de transmissão de 11 barras e Ckt5



(a) Número de iterações por passo do método alternado

(b) Número de iterações por passo do método alternado com sensibilidade

Fonte: Elaborado pelo autor (2022)

simulação e do sistema completo.

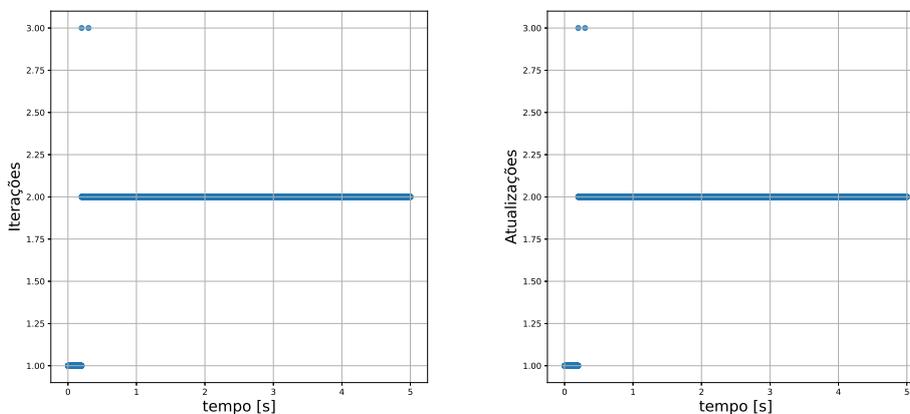
A Figura 39(b) apresenta o número de iterações requeridos por passo de integração do processo iterativo entre o OpenDSS e o sistema de transmissão interfaceado pelo equivalente de Norton. Nota-se que o processo iterativo não levou mais que três iterações, sendo que o maior número de iterações ocorreu nos instantes de aplicação e eliminação da falta, seguida da abertura da linha. Pode ser observado que o processo iterativo não levou mais que três iterações, sendo o maior número de iterações nos instantes da falta, bem como na abertura da linha de transmissão. Em comparação com o método alternado sem a inclusão de sensibilidade, não foi percebido alteração no número de iterações, conservando no entanto, a precisão dos resultados.

5.5.3.4 Método alternado "desonesto" com sensibilidade

Nessa metodologia, mantém-se a admitância equivalente de Norton, referente ao sistema de distribuição, constante ao longo da co-simulação, sendo atualizada apenas quando o número de iterações do processo iterativo for maior que dois. A corrente equivalente de Norton continua a ser calculada a cada iteração do processo iterativo.

Adotando essa estratégia, pode-se verificar a Figura 40(a) que apresenta o número de iterações de cada processo iterativo por passo, enquanto a Figura 40(b) apresenta o número de atualizações da admitância equivalente por passo de integração. Dos resultados extraídos, o número máximo de iterações observado é três, ocorrendo exatamente nos instantes de

Figura 40 – Medidas de desempenho do método alternado com sensibilidade - Co-simulação entre sistema de transmissão e Ckt5.



(a) Número de iterações por passo.

(b) Número de atualizações por passo.

Fonte: Elaborado pelo autor (2022)

descontinuidade. Assim, foram obtidas somente duas atualizações da impedância durante todo o período de simulação. Em comparação com os métodos alternado, e alternado com sensibilidade, o número de iterações por passo de integração se manteve. Dessa forma, o método foi eficaz para o processo de co-simulação, conservando, simultaneamente, a precisão da solução.

A atualização da admitância pode ser evidenciada também pelas Figura 41(a), Figura 41(b) que representam o comportamento da condutância e susceptância ao longo da co-simulação.

Do comportamento extraído, pode ser observado novamente que a admitância foi atualizada duas vezes. Além disso, os valores para os métodos de sensibilidade foram muito próximos entre si.

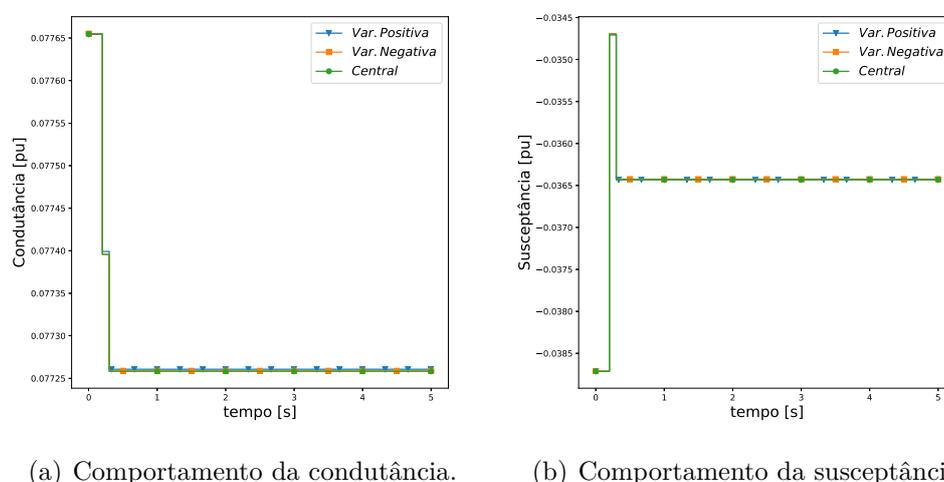
5.5.4 Análise de Desempenho Computacional

Um resumo dos tempos computacionais de cada metodologia empregada na realização da co-simulação entre a FMU dedicada ao sistema de transmissão de 11 barras e o sistema Ckt5 modelado no OpenDSS é apresentado na Tabela 20.

Pode-se perceber que o tempo de execução para as funções inerentes ao OpenDSS é, em geral, maior que o tempo gasto pela FMU, tendo em vista que o sistema Ckt5 é formado por 2998 nós elétricos.

O menor tempo de execução total foi observado para o método alternado, seguido de perto do método da linha fictícia. Porém, como já observado, os métodos da linha

Figura 41 – Comportamento da admitância - Método alternado "desonesto".



Fonte: Elaborado pelo autor (2022)

Tabela 20 – Perfil da co-simulação - Sistemas de transmissão e o Ckt5 (tempos em [s]).

Discriminação	Tempo de execução FMU	Tempo de execução OpenDSS	Proc. Variáveis	Tempo de execução Total
Linha Fictícia	4,878	16,383	4,108	36
Método com atraso	4,554	11,152	3,485	20
Método alternado	9,519	26,186	3,272	40
Sensibilidade - Positiva	12,576	58,740	3,608	77
Sensibilidade - Negativa	12,654	58,654	3,326	76
Sensibilidade - Central	12,616	61,491	3,430	79
Alternado "desonesto" - Positiva	12,238	27,735	3,406	45
Alternado "desonesto" - Negativa	11,983	26,631	3,286	43
Alternado "desonesto" - Central	11,721	26,028	3,274	42

Fonte: Elaborado pelo autor (2022)

fictícia e com atraso não foi tão precisos como os demais métodos, devido ao atraso inerente à co-simulação. Já o método alternado apresenta alta precisão enquanto requer o menor número de chamadas do OpenDSS. Os métodos que demandam cálculos de sensibilidade, por outro lado, acabam apresentando os maiores tempos computacionais, devido à necessidade de atualização do equivalente do sistema de distribuição a cada iteração, realizada via chamadas extras do OpenDSS.

Mais uma vez, não foram observadas variações significativas entre os tempos computacionais dos métodos baseados em sensibilidades. E como esperado, os tempos de execução total para os métodos alternado "desonesto", foram menores em relação aos

Tabela 21 – NIAE - Sistemas de transmissão e Ckt5 (Valores em [%]).

Discriminação	δ_{14}	V_{bus7}	P_{slack}
Linha fictícia	99,774	99,936	99,893
Método com atraso	99,864	99,946	99,904
Método Alternado	99,865	99,943	99,897
Sensibilidade - Positiva	99,865	99,943	99,897
Sensibilidade - Negativa	99,865	99,943	99,897
Sensibilidade - Central	99,865	99,943	99,897
Alternado "desonesto" - Positiva	99,865	99,943	99,897
Alternado "desonesto" - Negativa	99,865	99,943	99,897
Alternado "desonesto" - Central	99,865	99,943	99,897

Fonte: Elaborado pelo autor (2022)

demais métodos de sensibilidade. Pode-se verificar uma redução de aproximadamente 45% em relação aos métodos de sensibilidade. Essa redução deve-se à atualização da admitância equivalente ser realizada somente em instantes de descontinuidade. Os valores referentes ao critério NIAE podem ser vistos na Tabela 21.

Pode se notar que os valores do critério NIAE foram superiores ao valor de referência, mostrando assim que todas as metodologias foram adequadas para esse estudo. Ademais, o método alternado "desonesto" conservou a precisão dos resultados assim como observado para o estudo anterior.

5.6 CONCLUSÕES PARCIAIS

Esse capítulo teve por objetivo explorar outras metodologias de co-simulação de forma a atender os objetivos desse trabalho. Todas as metodologias apresentadas nesse capítulo foram bem sucedidas.

O primeiro método exposto foi o com atraso, o qual baseou-se na troca de termos entre corrente e tensão no laço de integração. Esse método mostrou-se eficiente do ponto de vista computacional porém outros métodos mais precisos foram explorados como os métodos alternado, alternado com sensibilidade e o "desonesto".

Foi também apresentado o método alternado, no qual, a cada passo de integração, realiza-se um processo iterativo entre a FMU e o OpenDSS, com o objetivo de compatibilizar a interface. Como vantagem desse método, tem-se que o erro de interface é praticamente eliminado. Portanto, mostrou-se um método mais preciso que o método com atraso, embora ainda possa incorrer em problemas de convergência.

Na sequência, foi também apresentado o método alternado com sensibilidade. Nessa metodologia, o sistema de distribuição é representado através de um equivalente de Norton para o sistema de transmissão a cada iteração do processo iterativo entre OpenDSS e

OpenModelica. Além disso, a admitância do equivalente de Norton é atualizada a cada iteração, o que pode aumentar o tempo computacional. Assim, foi utilizada a estratégia de atualizar a admitância equivalente somente, quando fosse verificado um aumento no número de iterações. Técnica conhecida como desonesta, o que possibilita além de garantir a precisão dos resultados, reduzir o tempo computacional do método alternado com sensibilidade.

Por fim, testes de co-simulação foram realizados com o sistema de transmissão de 11 barras e o sistema de distribuição IEEE 34 barras, como também o sistema Ckt5. Para ambos estudos, resultados para todas as metodologias foram bem sucedidos. Embora no último capítulo o método da linha fictícia tenha sido verificada como um método eficiente do ponto de vista computacional, assim como o método com atraso, foram verificadas limitações da metodologia, diante da dependência da impedância característica. Contudo, todos os métodos apresentados nesse capítulo eliminaram as limitações impostas pela linha fictícia, devido à não dependência de uma impedância característica. Além disso, o método alternado "desonesto" possibilitou a redução do tempo computacional, e manteve a precisão dos resultados.

6 CONCLUSÕES E DESENVOLVIMENTOS FUTUROS

Os sistemas elétricos de potências tem se tornado cada vez mais complexos nos últimos anos, tornando a realização de estudos dinâmicos cada vez mais desafiador, principalmente com a inserção dos REDs. Um desafio é que esses recursos podem exigir técnicas de modelagem específicas de solução. Ademais, é desejável que a interação dessas fontes com o sistema seja realizada em uma mesma plataforma porém, o procedimento é desafiador. Assim a co-simulação vem ao encontro dessas necessidades, permitindo que sistemas complexos e heterogêneos sejam divididos em subsistemas menores, cuja modelagem e solução podem ser implementadas em plataformas apropriadas.

Neste trabalho, foram apresentadas metodologias de co-simulação para estudos dinâmicos de sistemas de transmissão e distribuição. Inicialmente foram apresentados fundamentos de co-simulação, e foi explorada a interface padronizada FMI, a qual garante a compatibilidade de acoplamento entre sistemas.

Posteriormente, foi desenvolvida a metodologia da linha fictícia, e discutida a co-simulação entre sistemas de transmissão e ramos de distribuição, modelados em linguagem Modelica e embutidos em FMUs. O algoritmo mestre da co-simulação foi realizada em ambiente Python, com o auxílio dos módulos PyFMI e OMSimulator. Os resultados mostraram-se satisfatórios, replicando de forma fiel os resultados de simulações para o mesmo sistema modelado de forma completa, sem partições, obtidos no OpenModelica. A análise de erros validou a precisão dos resultados. Ademais, tempos computacionais foram reduzidos pela técnica adotada, comparando-se à simulação do sistema completo quando testada com até dois alimentadores. A partir de três alimentadores, a co-simulação se apresentou como a única estratégia capaz de realizar os estudos propostos.

A metodologia supracitada também foi testada para a co-simulação entre FMUs geradas pelo OpenModelica e o simulador de redes de distribuição trifásicas, OpenDSS. Nesse caso, o elo da linha referente ao sistema de distribuição, manteve-se externo ao OpenDSS, sendo iterado com a ferramenta a cada passo de integração.

No que tange aos resultados, foram realizados testes de co-simulação de um sistema de transmissão de 11 barras, e os sistemas de distribuição IEEE 34 barras e Ckt5. Para ambos os estudos, resultados bem sucedidos foram obtidos somente para quando a impedância característica foi considerada próxima a impedância de Thévenin do sistema de transmissão, vista do ponto de acoplamento comum. Além disso, foi discutida a influência da topologia da rede de distribuição, para a co-simulação. Foi mostrado que quando a estimativa da impedância de Thévenin apresenta valores de ordem de grandeza elevada, oscilações na interface podem vir a aparecer.

Embora algumas limitações tenham sido encontradas com o método da linha fictícia, outras quatro metodologias foram testadas para a co-simulação. Métodos com atraso,

alternado, alternado com sensibilidade e o desonesto, mostraram-se também eficazes para o acoplamento entre as ferramentas. Foi mostrado que o método com atraso, apesar de eficiente do ponto de vista computacional, não é tão preciso como o método alternado. Isso porque, o método alternado, possibilita compatibilizar a interface entre os sistemas a cada passo de integração. Ademais, foi mostrado que a inclusão de uma sensibilidade tende a melhorar as estimativas durante o processo iterativo, embora aumente o tempo computacional. Finalmente o método alternado desonesto com sensibilidade possibilitou a redução do tempo computacional ao mesmo tempo que manteve a precisão dos resultados.

Por fim, resultados de co-simulação com o sistema de transmissão de 11 barras e os sistemas de distribuição IEEE 34 barras e Ckt5, foram apresentados. Os comportamentos das grandezas extraídas foram coerentes com a simulação realizada no OpenModelica. Em relação aos tempos computacionais, o menor observado foi para o método com atraso seguido da linha fictícia. Os métodos com sensibilidade obtiveram os maiores tempos computacionais. Contudo, uma redução significativa do tempo computacional foi obtida pelo método alternado desonesto, quando comparado com os métodos com sensibilidade.

Portanto, as metodologias apresentadas atenderam os objetivos da proposta, uma vez que permitiram tanto realizar o acoplamento entre ferramentas distintas para estudos dinâmicos como também obter ganhos em termos de desempenho computacional e precisão dos resultados em relação a simulações de sistemas monolíticos. Deseja-se que as metodologias apresentadas contribuam para outras aplicações e investigações, como a avaliação de impactos dinâmicos em sistemas de potência devido a inserção de REDs aos sistemas de distribuição.

6.1 TRABALHOS FUTUROS

Alguns pontos de melhoria, bem como novas investigações podem ser propostas a partir dos resultados obtidos:

1. Investigação da impedância característica do modelo da linha de transmissão fictícia de forma mais rigorosa, e do seu impacto na simulação de sistemas modelados no OpenDSS
2. Inclusão de modelos dinâmicos de REDs aos sistemas de distribuição modelados no simulador OpenDSS
3. Avaliação dos impactos dos REDs na dinâmica dos sistema de potência como um todo pelas técnicas de co-simulação abordadas nesse trabalho.
4. Co-simulação do sistema de transmissão com múltiplos ramais de distribuição modelados no OpenDSS

5. Investigação do comportamento das sensibilidades diante das não-linearidades de sistemas elétricos de potência
6. Realização de análises de métricas de desempenho das metodologias de co-simulação apresentadas nesse trabalho, quando comparadas às simulações de sistemas monolíticos.

REFERÊNCIAS

- [1] ISSICABA, D., NASCIMENTO, L. C., DE SILVA, A. M. L., ROSA, M., LOPES, J. P., “Impacto na confiabilidade de sistemas de distribuição de energia elétrica sujeitos à restrição de carga e integrados a recursos energéticos distribuídos”, 2012.
- [2] DE PESQUISA ENERGÉTICA, E., “Balço Energético Nacional 2021”, Disponível em: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/balanco-energetico-nacional-2021>.
- [3] DE PESQUISA ENERGÉTICA, E., “Plano Decenal de Expansão de Energia 2030”, Disponível em: <https://www.epe.gov.br/>.
- [4] DE PESQUISA ENERGÉTICA, E., “Recursos Energéticos Distribuídos: Impactos no Planejamento Energético”, Disponível em: <https://www.epe.gov.br/sites-pt/sala-de-imprensa/noticias/Documents/ND>
- [5] SILVA, L. T., *Modelagem e Simulação de Sistemas de Geração de Energia Eólica através de Co-simulação*, Master’s Thesis, Universidade Federal de Juiz de Fora (UFJF), 2020.
- [6] MODELICA ASSOCIATION PROJECT FMI, “Functional Mock-up Interface for model exchange and co-simulation”, 2021, Disponível em: <https://fmi-standard.org/>.
- [7] MOHSENI-BONAB, S. M., HAJEBRAHIMI, A., KAMWA, I., MOEINI, A., “Transmission and distribution co-simulation: a review and propositions”, *IET Generation, Transmission & Distribution*, v. 14, n. 21, pp. 4631–4642, 2020.
- [8] VENKATRAMAN, R., KHAITAN, S. K., AJJARAPU, V., “A combined transmission-distribution system dynamic model with grid-connected DG inverter”. In: *2017 IEEE Power & Energy Society General Meeting*, pp. 1–5, 2017.
- [9] IEEE, “IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules”, , n. IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000), pp. 1–38, 2010.
- [10] GODFRAIND, A., OTHERS, “Master thesis: Co-simulation platform for cyber-physical transmission systems”, 2021.
- [11] HOPKINSON, K., WANG, X., GIOVANINI, R., THORP, J., BIRMAN, K., COURRY, D., “EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components”, *IEEE Transactions on Power Systems*, v. 21, n. 2, pp. 548–558, 2006.
- [12] CAMUS, B., PARIS, T., VAUBOURG, J., PRESSE, Y., BOURJOT, C., CIARLETTA, L., CHEVRIER, V., “Co-simulation of cyber-physical systems using a DEVS wrapping strategy in the MECSYCO middleware”, *Simulation*, v. 94, n. 12, pp. 1099–1127, 2018.
- [13] STEINBRINK, C., BLANK-BABAZADEH, M., EL-AMA, A., HOLLY, S., LÜERS, B., NEBEL-WENNER, M., RAMÍREZ ACOSTA, R. P., RAUB, T., SCHWARZ, J. S., STARK, S., OTHERS, “Cpes testing with mosaik: Co-simulation planning, execution and analysis”, *Applied Sciences*, v. 9, n. 5, pp. 923, 2019.

- [14] WERMANN, A. G., BORTOLOZZO, M. C., DA SILVA, E. G., SCHAEFFER-FILHO, A., GASPARY, L. P., BARCELLOS, M., “ASTORIA: A framework for attack simulation and evaluation in smart grids”. In: *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 273–280, 2016.
- [15] ANDERSSON, C., *Methods and tools for co-simulation of dynamic systems with the functional mock-up interface*, Ph.D. Thesis, Lund University, 2016.
- [16] SCHWEIGER, G., GOMES, C., ENGEL, G., HAFNER, I., SCHOEGGL, J., POSCH, A., NOUIDUI, T., “An empirical survey on co-simulation: Promising standards, challenges and research needs”, *Simulation modelling practice and theory*, v. 95, pp. 148–163, 2019.
- [17] SANTOS, T. T., TOMIM, M. A., BARBOSA, P. G., DE LIMA, A. C. S., DE BARROS, M. T. C., “Co-simulation of a doubly fed induction generator connected to a power network: The use of DSOGI for phasor extraction”, *IEEE Latin America Transactions*, v. 17, n. 07, pp. 1070–1079, 2019.
- [18] THEODORO, T., BARBOSA, P., TOMIM, M., DE LIMA, A., DE BARROS, M. C., “MatLab-OpenDSS co-simulation environment: An alternative tool to investigate DSG connection”. In: *2018 Simposio Brasileiro de Sistemas Eletricos (SBSE)*, pp. 1–7, 2018.
- [19] DUGAN, R., *Reference Guide: The Open Distribution System Simulator (OpenDSS)*.
- [20] HARIRI, A., FARUQUE, M. O., “A hybrid simulation tool for the study of PV integration impacts on distribution networks”, *IEEE Transactions on Sustainable Energy*, v. 8, n. 2, pp. 648–657, 2016.
- [21] VENKATRAMAN, R., KHAITAN, S. K., AJJARAPU, V., “Dynamic co-simulation methods for combined transmission-distribution system with integration time step impact on convergence”. v. 34, n. 2, pp. 1171–1181, IEEE, 2018.
- [22] WANG, W., FANG, X., CUI, H., LI, F., “Transmission-and-Distribution Frequency Dynamic Co-Simulation Framework for Distributed Energy Resources Frequency Response”, *arXiv preprint arXiv:2101.05894*, 2021.
- [23] HUANG, R., FAN, R., DAILY, J., FISHER, A., FULLER, J., “Open-source framework for power system transmission and distribution dynamics co-simulation”, *IET Generation, Transmission & Distribution*, v. 11, n. 12, pp. 3152–3162, 2017.
- [24] SANTOS, T. T., TOMIM, M. A., BARBOSA, P. G., DE LIMA, A. C. S., DE BARROS, M. T. C., “Co-simulation of a doubly fed induction generator connected to a power network: The use of DSOGI for phasor extraction”, *IEEE Latin America Transactions*, v. 17, n. 07, pp. 1070–1079, 2019.
- [25] SANTOS, T. T., *Simulação híbrida no domínio do tempo de transitórios eletromecânicos e eletromagnéticos: integração de um aerogerador de indução duplamente excitado.*, Master’s Thesis, 2016.
- [26] LAWRENCE LIVERMORE NATIONAL LABORATORY, “SUNDIALS: SUite of Nonlinear and Differential/ALgebraic Equation Solvers”, 2021, Disponível em: <https://computing.llnl.gov/projects/sundials>.

- [27] HINDMARSH, A. C., BROWN, P. N., GRANT, K. E., LEE, S. L., SERBAN, R., SHUMAKER, D. E., WOODWARD, C. S., “SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers”, *ACM Transactions on Mathematical Software (TOMS)*, v. 31, n. 3, pp. 363–396, 2005.
- [28] MODELON AB, *JModelica.org User Guide*, 2018, Disponível em: <https://jmodelica.org/downloads/UsersGuide.pdf>.
- [29] OPEN SOURCE MODELICA CONSORTIUM, “OpenModelica Simulator (OM-Simulator)”, 2021, Disponível em: <https://openmodelica.org/uncategorised/191-omsimulator>.
- [30] OPEN SOURCE MODELICA CONSORTIUM, *OpenModelica Users Guide*, 2021, Disponível em: <https://openmodelica.org/doc/OpenModelicaUsersGuide/OpenModelicaUsersGuide-latest.pdf>.
- [31] MENGIST, A., ASGHAR, A., POP, A., FRITZSON, P., BRAUN, W., SIEMERS, A., FRITZSON, D., “An open-source graphical composite modeling editor and simulation tool based on FMi and TLM co-simulation”. In: *11th International Modelica Conference, Versailles, France, September 21-23, 2015*, pp. 181–188, 2015.
- [32] ARRILLAGA, J., ARNOLD, C., *Computer analysis of power systems*. Wiley Online Library, 1990.
- [33] KWON PARK, I., FORSYTH, P., KIM, H., HUR, K., “A Study on synchronizing two separate RTDS simulation instances”. In: *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5, 2016.
- [34] DUFOUR, C., PAQUIN, J.-N., LAPOINTE, V., BÉLANGER, J., SCHOEN, L., “PC-Cluster-Based Real-Time Simulation of an 8-synchronous machine network with HVDC link using RT-LAB and TestDrive”. In: *Proc. Int. Conf. Power Systems Transients*, 2007.
- [35] WILLSKY, A. S., YOUNG, I. T., *Signals and systems*. Prentice-Hall International, 1997.
- [36] KUNDUR, P., *Power System Stability and Control*. McGraw-Hill: New York, 1994.
- [37] SINGH, D., MISRA, R. K., SINGH, D., “Effect of load models in distributed generation planning”, *IEEE Transactions on Power Systems*, v. 22, n. 4, pp. 2204–2212, 2007.
- [38] DE FREITAS, P. R. R., “Modelos avançados de análise de redes elétricas inteligentes utilizando o software OpenDSS”, .
- [39] DAVIS, T. A., PALAMADAI NATARAJAN, E., “Algorithm 907: KLU, a direct sparse solver for circuit simulation problems”, *ACM Transactions on Mathematical Software (TOMS)*, v. 37, n. 3, pp. 1–17, 2010.
- [40] SUN, H., GUO, Q., ZHANG, B., GUO, Y., LI, Z., WANG, J., “Master–slave-splitting based distributed global power flow method for integrated transmission and distribution analysis”, *IEEE Transactions on Smart Grid*, v. 6, n. 3, pp. 1484–1492, 2014.

- [41] STOTT, B., “Power system dynamic response calculations”, *Proceedings of the IEEE*, v. 67, n. 2, pp. 219–241, 1979.
- [42] ELECTRIC POWER RESEARCH INSTITUTE, *The Open Distribution System Simulator*, 2020, Disponível em: <https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/Doc/OpenDSSManual.pdf>.
- [43] INSTITUTE, E. P. R., “Smart Grid Resource Center”, Disponível em: <https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/EPRITestCircuits/>.
- [44] MANZONI, A., “Desenvolvimento de um módulo dinâmico para simuladores de ensino e treinamento em sistemas de energia elétrica usando programação orientada a objetos”, 1996.
- [45] SILVA, J. P. A., *Metodologia para Modelagem e Simulação de Curto e Longo Termo em Sistemas Elétricos de Potência*, Master’s Thesis.
- [46] NETO, A. B., MAGALHAES, E. M., ALVES, D. A., “Dishonest newton method applied in continuation power flow through a geometric parameterization technique”, *IEEE Latin America Transactions*, v. 14, n. 1, pp. 161–170, 2016.
- [47] IEEE-PES, “IEEE PES Test Feeder”, Disponível em: <http://sites.ieee.org/pes-testfeeders/resources>.
- [48] ELECTRIC POWER RESEARCH INSTITUTE (EPRI), “EPRI Teste Circuits”, 2015, Disponível em: <http://svn.code.sf.net/p/electricdss/code/trunk/Distrib/EPRITestCircuits/Readme.pdf>.
- [49] JR., W. D. S., *Elementos de análise de sistemas de Potência*. Editora McGraw-Hill do Brasil, 1986.
- [50] ZAHN, M., *Electromagnetic Field Theory: a problem solving approach*. Wiley, 1979.

APÊNDICE A – SISTEMAS TESTE

Nesse apêndice serão descritos os sistemas teste utilizados para os estudos de co-simulação apresentados nos Capítulo 4 e Capítulo 5.

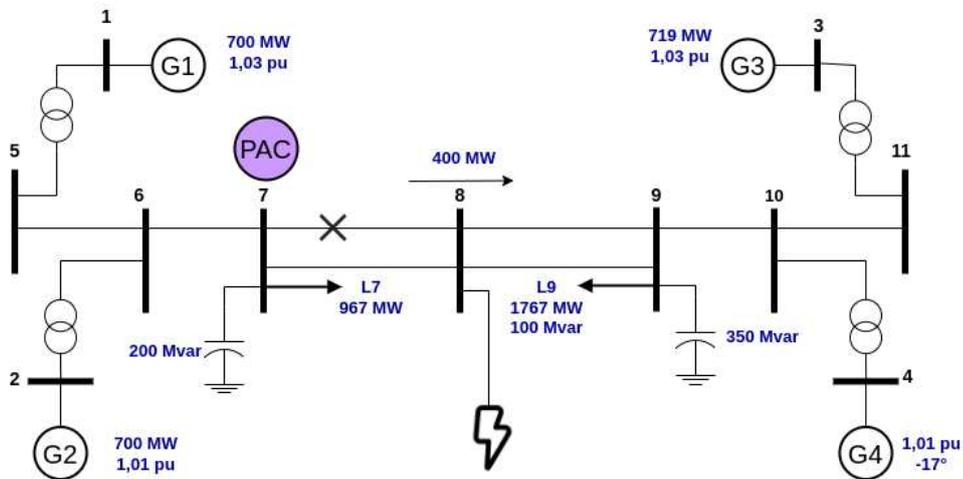
A.1 SISTEMA DE TRANSMISSÃO

O sistema de transmissão de 11 barras ilustrado na Figura 42 e detalhado em [36], foi usado para todos os testes de co-simulação de trabalho. O sistema transmissão é formado por duas áreas e 11 barras. As duas áreas desse sistema são conectadas através de duas linha de transmissão de 220 [km] com quatro usinas representadas por G_1 , G_2 , G_3 e G_4 . Cada usina do sistema, tem uma capacidade de 900 [MVA]. As usinas G_1 , G_2 e G_3 são despachadas em 700 [MW], 700 [MW] e 719 [MW], respectivamente, enquanto G_4 serve de referência do sistema. Além disso, foram inseridos também bancos de capacitores em derivação de 200 [Mvar] na barra 7 e 350 [Mvar] na barra 9. As usinas 1 e 2 possuem constante de inércia de $H = 6,5$ [s] enquanto as usinas 3 e 4 possuem como constante, o valor de $H = 6,175$ [s];

Os parâmetros das máquinas síncronas podem ser descritas na Tabela 22.

Além da definição do número de máquinas, é importante considerar que cada máquina é equipada com regulador automático de tensão (AVR) e estabilizador de potência (PSS). O diagrama de bloco que representa o regulador automático de tensão (AVR) juntamente com o estabilizador de potência (PSS), é ilustrado na Figura 43. Os ganhos e constantes de tempo adotados para esses reguladores de tensão são descritos na Tabela 23.

Figura 42 – Sistema teste de 11 barras.



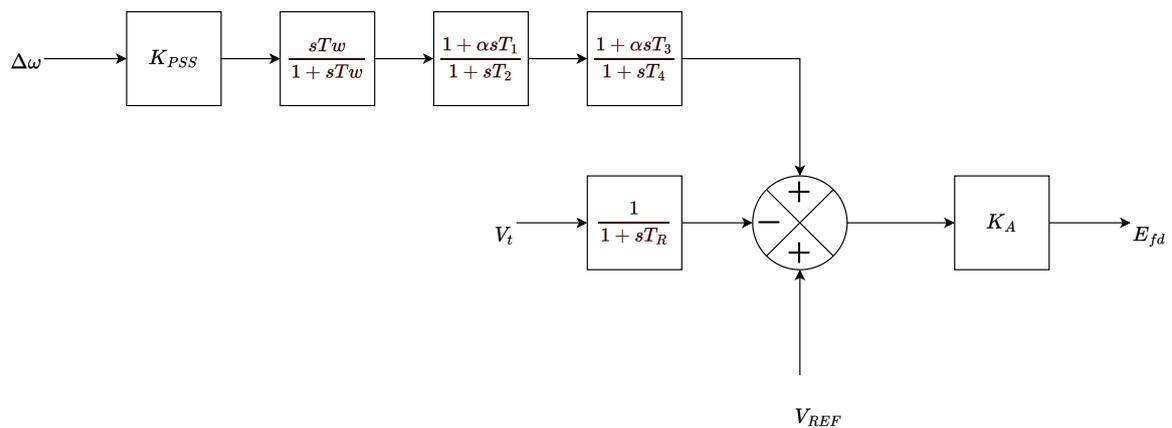
Fonte: Elaborado pelo autor (2022)

Tabela 22 – Parâmetros elétricos de cada unidade geradora

Parâmetro	Valor
R_a	0,0025 [pu]
T'_{d0}	8 [s]
T'_{q0}	0,4 [s]
T''_{d0}	0,03 [s]
T''_{q0}	0,05 [s]
X'_d	0,3 [pu]
X'_q	0,55 [pu]
X''_d	0,25 [pu]
X''_q	0,25 [pu]
X_d	1,8 [pu]
X_l	0,2 [pu]
X_q	1,7 [pu]

Fonte: Elaborado pelo autor (2022)

Figura 43 – Diagrama de blocos do AVR + PSS.



Fonte: Elaborado pelo autor (2022)

Os parâmetros da linha em [pu] na base de 100 [MVA] e 230 [kV] são vistos na Tabela 24, enquanto os seus comprimentos são dados na Tabela 25.

Tabela 23 – Parâmetros dos reguladores de tensão e estabilizadores de potência.

Parâmetro	Valor	Parâmetro	Valor
K_A	200	T_R	0,01
K_{PSS}	20,0	T_W	10,0
T_1	0,05	T_2	0,02
T_3	3,0	T_4	5,4

Fonte: Elaborado pelo autor (2022)

Tabela 24 – Parâmetros das linhas de transmissão

Parâmetro	Valor [pu/km]
r	0,0001
x_L	0,001
b_C	0,00175

Fonte: Elaborado pelo autor (2022)

Tabela 25 – Comprimento da linha de transmissão

Linha	Comprimento [km]
5-6	25
6-7	10
7-8	110
8-9	110
9-10	10
10-11	25

Fonte: Elaborado pelo autor (2022)

A.2 SISTEMAS DE DISTRIBUIÇÃO

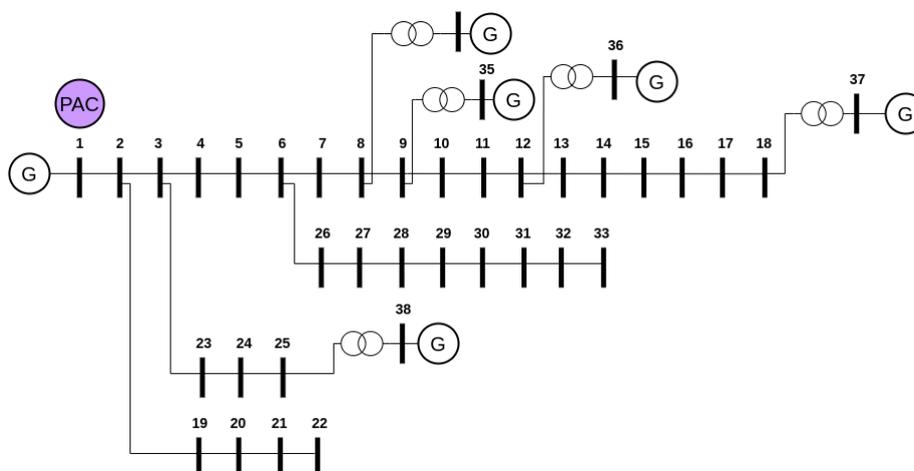
Foram considerados três sistemas de distribuição para os testes desenvolvidos, que serão descritos na sequência.

A.2.1 Sistema de 38 barras

O primeiro sistema de distribuição é composto por 38 barras [37]. Para este estudo, o sistema é representado por meio de seu equivalente em sequência positiva, ilustrado na Figura 44.

Vale ressaltar que neste sistema também estão incluídas cinco unidades geradoras conectadas das barras 34 a 38. Em relação as unidades de geração deste sistema de distribuição, os dados de geração podem ser vistos na Tabela 26. Quanto aos dados de

Figura 44 – Sistema teste de 38 barras.



Fonte: Elaborado pelo autor (2022)

Tabela 26 – Dados de geração para o sistema de distribuição

Gerador	P [MW]	V [p.u]
G_{34}	88,75	1,01
G_{35}	44,4	1,01
G_{36}	14,8	1,01
G_{37}	29,59	1,01
G_{38}	14,4	1,01

Fonte: Elaborado pelo autor (2022)

carga, estes podem ser verificado na Tabela 27. Também, os dados de linha de transmissão do sistema de 38 barras, podem ser vistos na Tabela 28. É importante ressaltar que, os parâmetros R e X estão dados em [%], considerando uma potência base em 100 [MVA].

Tabela 27 – Dados de carga do sistema de 38 barras

Barra	P [MW]	Q [Mvar]
1	-	-
2	10,0	6,0
3	9,0	4,0
4	12,0	8,0
5	6,0	3,0
6	6,0	2,0
7	20,0	10,0
8	20,0	10,0
9	6,0	2,0
10	6,0	2,0
11	4,5	3,0
12	6,0	3,505
13	6,0	3,505
14	12,0	8,0
15	6,0	1,0
16	6,0	2,0
17	6,0	2,0
18	9,0	4,0
19	9,0	4,0
20	9,0	4,0
21	9,0	4,0
22	9,0	4,0
23	9,0	5,0
24	42,0	20,0
25	42,0	20,0
26	6,0	2,505
27	6,0	2,505
28	6,0	2,0
29	12,0	7,0
30	20,0	60,0
31	15,0	7,0
32	21,0	10,0
33	6,0	4,0

Fonte: Elaborado pelo autor (2022)

Tabela 28 – Dados de linha de transmissão do sistema de 38 barras (base de 100 MVA)

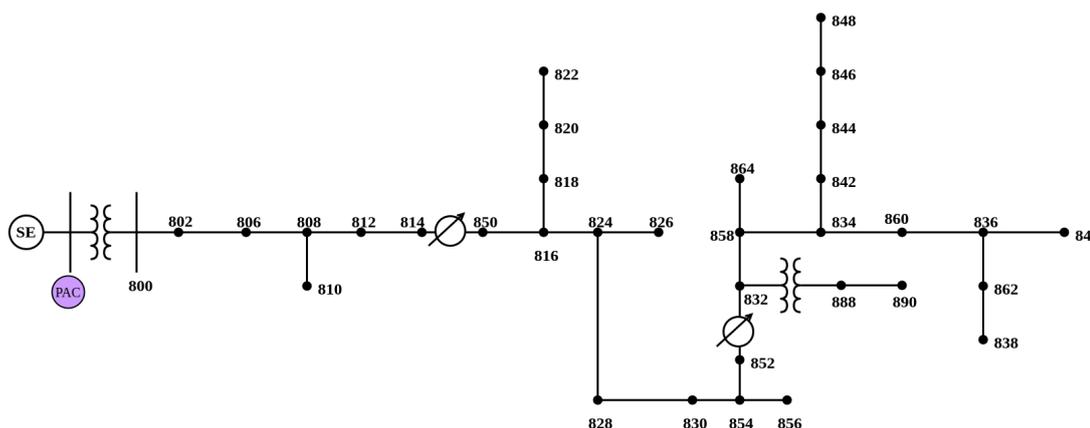
Barra de	Barra para	R [%]	X [%]	Tape
1	2	0,0574	0,0293	
2	3	0,307	0,1564	
3	4	0,2279	0,1161	
4	5	0,2373	0,1209	
5	6	0,5100	0,4402	
6	7	0,1166	0,3853	
7	8	0,443	0,1464	
8	9	0,6413	0,4608	
9	10	0,6501	0,4608	
10	11	0,1224	0,0405	
11	12	0,2331	0,0771	
12	13	0,9141	0,7192	
13	14	0,3372	0,4439	
14	15	0,3680	0,3275	
15	16	0,4647	0,3394	
16	17	0,8026	1,0716	
17	18	0,4558	0,3574	
2	19	0,1021	0,0974	
19	20	0,9366	0,8440	
20	21	0,2550	0,2979	
21	22	0,4414	0,5836	
3	23	0,2809	0,1920	
23	24	0,5592	0,4415	
24	25	0,5579	0,4366	
6	26	0,1264	0,0644	
26	27	0,1770	0,0901	
27	28	0,6594	0,5814	
28	29	0,5007	0,4362	
29	30	0,3160	0,1610	
30	31	0,6067	0,5996	
31	32	0,1933	0,2253	
32	33	0,2123	0,3301	
8	34	1,2453	1,2453	1,000
9	35	1,2453	1,2453	1,000
12	36	1,2453	1,2453	1,000
18	37	0,3113	0,3113	1,000
25	38	0,3113	0,3113	1,000

Fonte: Elaborado pelo autor (2022)

A.2.2 Sistema IEEE 34-barras

O segundo sistema de distribuição considerado aqui é o sistema IEEE 34-barras mostrado na Figura 45 [47]. Corresponde a um sistema trifásico desequilibrado em 24,9 [kV] conectado à subestação em 69 [kV], consumindo aproximadamente 2026,29 [kW] e 311,94 [kvar].

Figura 45 – Sistema de distribuição IEEE 34-barras.



Fonte: Elaborado pelo autor (2022)

O alimentador está conectado ao ponto de acoplamento comum (PAC) por um transformador na subestação, entre as barras PAC e 800. Outro transformador nomeado de XFM-1, pode ser visto conectado entre as barras 832 e 888. Os dados dos transformadores anteriormente citados encontram-se na Tabela 29. Os termos D e Yt referem-se as ligações em Delta e Estrela aterrada respectivamente.

Em relação aos dados de linha de distribuição, a Tabela 30 apresenta os dados de linha bem como os tipos de conexões para cada Configuração, enquanto a Tabela 31 mostra o comprimento de cada linha de distribuição.

Os dados de cada configuração de linha, podem ser vistos na sequência. De (A.1) à (A.5), são apresentadas matrizes de resistências, reatâncias e capacitâncias para cada configuração em componentes de fase. É importante salientar que essas matrizes são simétricas. Dados de cargas locais e distribuídas, são verificadas na Tabela 32 e Tabela 33. A Tabela 34 explica os códigos para cada modelo de carga.

Tabela 29 – Dados dos transformadores - IEEE 34 barras

Discriminação	kVA	AT [kV]	BT [kV]	R [%]	X [%]
Subestação	2500	69 - D	24,9 - Yt	1	8
XFM-1	500	24,9 - Yt	4,16 - Yt	1,9	4,08

Fonte: *IEEE PES Test Feeder* [47] (1992).

Tabela 30 – Configuração das linhas de distribuição - IEEE 34 barras

Configuração	Conexão	Fase/ACSR	Neutro/ACSR	Espaçamento ID
300	B A C N	1/0	1/0	500
301	B A C N	2#6/1	2#6/1	500
302	A N	4#6/1	4#6/1	510
303	B N	4#6/1	4#6/1	510
304	B N	2#6/1	2#6/1	510

Fonte: *IEEE PES Test Feeder* [47] (1992).

Dados de bancos de capacitores são verificados na Tabela 35. E por fim, dados dos reguladores para esse sistema, são vistos na Tabela 36 e Tabela 37 .

1. Configuração 300

$$\mathbf{R}_{abc} = \begin{bmatrix} 0,253181818 & & \\ 0,039791667 & 0,250719697 & \\ 0,040340909 & 0,039128788 & 0,251780303 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.1a})$$

$$\mathbf{X}_{abc} = \begin{bmatrix} 0,252708333 & & \\ 0,109450758 & 0,256988636 & \\ 0,094981061 & 0,086950758 & 0,255132576 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.1b})$$

$$\mathbf{B}_{abc} = \begin{bmatrix} 2,680150309 & & \\ -0,769281006 & 2,5610381 & \\ -0,499507676 & -0,312072984 & 2,455590387 \end{bmatrix} \quad [\text{S}/\text{mi}] \quad (\text{A.1c})$$

2. Configuração 301

$$\mathbf{R}_{abc} = \begin{bmatrix} 0,365530303 & & \\ 0,04407197 & 0,36282197 & \\ 0,04467803 & 0,043333333 & 0,363996212 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.2a})$$

$$\mathbf{X}_{abc} = \begin{bmatrix} 0,267329545 & & \\ 0,122007576 & 0,270473485 & \\ 0,107784091 & 0,099204545 & 0,269109848 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.2b})$$

$$\mathbf{B}_{abc} = \begin{bmatrix} 2,572492163 & & \\ -0,72160598 & 2,464381882 & \\ -0,472329395 & -0,298961096 & 2,368881119 \end{bmatrix} \quad [\text{S}/\text{mi}] \quad (\text{A.2c})$$

3. Configuração 302

$$\mathbf{R}_{abc} = \begin{bmatrix} 0,530208 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.3a})$$

$$\mathbf{X}_{abc} = \begin{bmatrix} 0,281345 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.3b})$$

$$\mathbf{B}_{abc} = \begin{bmatrix} 2,12257 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\text{S}/\text{mi}] \quad (\text{A.3c})$$

4. Configuração 303

$$\mathbf{R}_{abc} = \begin{bmatrix} 0,530208 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.4a})$$

$$\mathbf{X}_{abc} = \begin{bmatrix} 0,281345 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.4b})$$

$$\mathbf{B}_{abc} = \begin{bmatrix} 2,12257 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\text{S}/\text{mi}] \quad (\text{A.4c})$$

5. Configuração 304

$$\mathbf{R}_{abc} = \begin{bmatrix} 0,363958 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.5a})$$

$$\mathbf{X}_{abc} = \begin{bmatrix} 0,269167 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\Omega/\text{mi}] \quad (\text{A.5b})$$

$$\mathbf{B}_{abc} = \begin{bmatrix} 2,1922 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [\text{S}/\text{mi}] \quad (\text{A.5c})$$

Tabela 31 – Comprimento das linhas de distribuição - IEEE 34 barras

Barra de	Barra Para	Comprimento [km]	Config.
800	802	0,786384	300
802	806	0,527304	300
806	808	9,823704	300
808	810	1,7690592	303
808	812	11,43	300
812	814	9,061704	300
814	850	0,003048	301
816	818	0,521208	302
816	824	3,112008	301
818	820	14,67612	302
820	822	4,187952	302
824	826	0,923544	303
824	828	0,256032	301
828	830	6,230112	301
830	854	0,158496	301
832	858	1,49352	301
832	888	0	XFM-1
834	860	0,615696	301
834	842	0,085344	301
836	840	0,262128	301
836	862	0,085344	301
842	844	0,41148	301
844	846	1,109472	301
846	848	0,161544	301
850	816	0,094488	301
852	832	0,003048	301
854	856	7,110984	303
854	852	11,225784	301
858	864	0,493776	302
858	834	1,776984	301
860	836	0,816864	301
862	838	1,481328	304
888	890	3,218688	300

Fonte: *IEEE PES Test Feeder* [47] (1992).

Tabela 32 – Cargas locais

Barra	Carga Modelo	Fase 1 kW	Fase 1 kvar	Fase 2 kW	Fase 2 kvar	Fase 3 kW	Fase 3 kvar
860	Y-PQ	20	16	20	16	20	16
840	Y-I	9	7	9	7	9	7
844	Y-Z	135	105	135	105	135	105
848	D-PQ	20	16	20	16 2	0	16
890	D-I	150	75	150	75	150	75
830	D-Z	10	5	10	5	25	10
Total		344	224	344	224	359	229

Fonte: *IEEE PES Test Feeder* [47] (1992).

Tabela 33 – Cargas distribuídas

Barras	Carga Modelo	Fase 1 kW	Fase 1 kvar	Fase 2 kW	Fase 2 kvar	Fase 3 kW	Fase 3 kvar	
802	806	Y-PQ	0	0	30	15	25	14
808	810	Y-I	0	0	16	8	0	0
818	820	Y-Z	34	17	0	0	0	0
820	822	Y-PQ	135	70	0	0	0	0
816	824	D-I	0	0	5	2	0	0
824	826	Y-I	0	0	40	20	0	0
824	828	Y-PQ	0	0	0	0	4	2
828	830	Y-PQ	7	3	0	0	0	0
854	856	Y-PQ	0	0	4	2	0	0
832	858	D-Z	7	3	2	1	6	3
858	864	Y-PQ	2	1	0	0	0	0
858	834	D-PQ	4	2	15	8	13	7
834	860	D-Z	16	8	20	10	110	55
860	836	D-PQ	30	15	10	6	42	22
836	840	D-I	18	9	22	11	0	0
862	838	Y-PQ	0	0	28	14	0	0
842	844	Y-PQ	9	5	0	0	0	0
844	846	Y-PQ	0	0	25	12	20	11
846	848	Y-PQ	0	0	23	11	0	0
Total		262	133	240	120	220	114	

Fonte: *IEEE PES Test Feeder* [47] (1992).

Tabela 34 – Códigos dos modelos de carga

Modelo	Descrição
PQ	Potências ativa e reativa constantes
I	Corrente constante
Z	Impedância constante

Fonte: Elaborado pelo autor (2022)

Tabela 35 – Dados dos bancos de capacitores

Barra	Fase-A kvar	Fase-B kvar	Fase-C kvar
844	100	100	100
848	150	150	150
Total	250	250	250

Fonte: *IEEE PES Test Feeder* [47] (1992).

Tabela 36 – Dados do regulador 1

Nº do regulador	1		
Segmento da linha	814 - 850		
Localização	814		
Fases	A - B -C		
Conexão	3Ø, LG		
Fases de monitoramento	A-B-C		
Largura de banda [V]	2,0		
Relação TP	120		
Classificação primária do TC	100		
Configuração do compensador	Fase-A	Fase-B	Fase-C
Configuração R	2,7	2,7	2,7
Configuração X	1,6	1,6	1,6
Nível de tensão	122	122	122

Fonte: *IEEE PES Test Feeder* [47] (1992).

Tabela 37 – Dados do regulador 2

Nº do regulador	2		
Segmento da linha	852 - 832		
Localização	852		
Fases	A - B - C		
Conexão	3Ø, LG		
Fases de monitoramento	A-B-C		
Largura de banda [V]	2.0		
Relação TP	120		
Classificação primária do TC	100		
Configuração do compensador	Fase-A	Fase-B	Fase-C
Configuração R	2,5	2,5	2,5
Configuração X	1,5	1,5	1,5
Nível de tensão	124	124	124

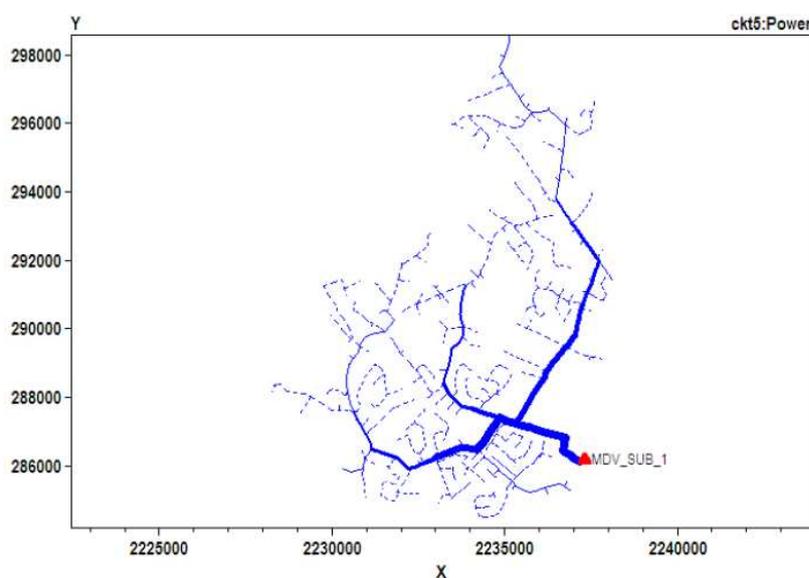
Fonte: *IEEE PES Test Feeder* [47] (1992).

A.2.3 Sistema Ckt5

A rede de distribuição Ckt5, mostrada na Figura 46, foi criada e publicada pelo EPRI com propósitos de estudos de sistemas de distribuição [43]. O triângulo em vermelho representa a subestação do sistema. O transformador da subestação, denominado MDV SUB1, tem uma potência aparente de 10 [MVA] e possui relação de transformação de 115 [kV] para 12,47 [kV]. O sistema contém 2998 barras. Além disso, o alimentador é trifásico e modelado apenas com cargas monofásicas, com diferentes potências. Características desse sistema podem ser vistas na Tabela 38.

Esse sistema consome aproximadamente 6756,32 [kW] e 3492,66 [kvar]. Os demais dados desse alimentador podem ser encontrados em [43].

Figura 46 – Sistema de distribuição - Ckt5



Fonte: *EPRI Test Circuits* [48] (2016).

Tabela 38 – Características gerais do sistema Ckt5.

Características	Valor
Tensão do Sistema em [kV]	12,47
Número de consumidores	1379
Potência total de Transformadores conectados [kVA]	16310
Potência reativa total dos alimentadores em [kvar]	1950
Tensão de subtransmissão em [kV]	115
Potência de CC Sec. Sub [MVA]	114
Comprimento do circuito primário [km]	77,3
Porcentagem de carga residencial	96
Ramais alimentadores	1

Fonte: *EPRI Test Circuits* [48] (2016).

APÊNDICE B – INICIALIZAÇÃO DA CO-SIMULAÇÃO

Esse apêndice tem por objetivo apresentar os dados da inicialização da co-simulação para cada estudo de caso

B.1 CO-SIMULAÇÃO ENTRE SISTEMAS DE TRANSMISSÃO DE 11 BARRAS E DE DISTRIBUIÇÃO DE 38 BARRAS

Para o acoplamento entre os sistemas de transmissão de 11 barras e de distribuição de 38 barras, as Figuras 39 e 40 apresentam o ponto operativo para a co-simulação entre os sistemas.

Tabela 39 – Inicialização da co-simulação para sistemas de transmissão e 38 barras - Sistema de 11 barras

Barra	V[pu]	θ [°]	P_G [MW]	Q_G [Mvar]
1	1,03	38,5135	700,0	154,5198
2	1,01	28,8286	700,0	161,1557
3	1,03	10,4399	719,1	170,3965
4	1,01	0,0	672,7121	185,4196
5	1,0113	32,0831	0,0	0,0
6	0,9902	22,1292	0,0	0,0
7	0,9827	13,9491	0,0	0,0
8	0,9585	-0,3672	0,0	0,0
9	0,9749	-14,6767	0,0	0,0
10	0,9856	-6,4665	0,0	0,0
11	1,0091	3,8188	0,0	0,0

Fonte: Elaborado pelo autor (2022)

B.2 CO-SIMULAÇÃO ENTRE SISTEMAS DE TRANSMISSÃO DE 11 BARRAS E DE DISTRIBUIÇÃO IEEE 34 BARRAS

Para o acoplamento entre os sistemas de transmissão de 11 barras e de distribuição IEEE 34 barras, as Figuras 41 e 42 apresentam o ponto operativo para a co-simulação entre os sistemas.

B.3 CO-SIMULAÇÃO ENTRE SISTEMAS DE TRANSMISSÃO DE 11 BARRAS E DE DISTRIBUIÇÃO CKT5

Para a co-simulação entre os sistemas de transmissão de 11 barras e o sistema de distribuição Ckt5, o ponto operativo é o mesmo da Tabela 41. Devido ao elevado número de nós, não serão apresentados resultados do fluxo de potência dos nós referentes ao sistema de distribuição Ckt5,

Tabela 40 – Inicialização da co-simulação para sistemas de transmissão e 38 barras - Sistema de 38 barras

Barra	V[p.u]	θ [°]	P_G [MW]	Q_G [Mvar]
1	0,9826	13,9491	0,0	0,0
2	0,9815	13,9150	0,0	0,0
3	0,9774	13,7348	0,0	0,0
4	0,9766	13,6351	0,0	0,0
5	0,9762	13,5286	0,0	0,0
6	0,9768	13,2597	0,0	0,0
7	0,9837	13,2990	0,0	0,0
8	0,9900	12,9381	0,0	0,0
9	0,9962	12,5952	0,0	0,0
10	0,9970	12,3911	0,0	0,0
11	0,9970	12,3574	0,0	0,0
12	0,9972	12,2913	0,0	0,0
13	0,9972	12,1176	0,0	0,0
14	0,9979	12,0459	0,0	0,0
15	0,9991	11,9822	0,0	0,0
16	1,0006	11,9101	0,0	0,0
17	1,0055	11,8367	0,0	0,0
18	1,0078	11,7819	0,0	0,0
19	0,9810	13,9039	0,0	0,0
20	0,9773	13,8349	0,0	0,0
21	0,9766	13,8150	0,0	0,0
22	0,9760	13,7940	0,0	0,0
23	0,9748	13,6694	0,0	0,0
24	0,9703	13,5172	0,0	0,0
25	0,9692	13,4099	0,0	0,0
26	0,9750	13,2968	0,0	0,0
27	0,9725	13,3497	0,0	0,0
28	0,9614	13,4279	0,0	0,0
29	0,9534	13,5012	0,0	0,0
30	0,9500	13,6001	0,0	0,0
31	0,9460	13,5208	0,0	0,0
32	0,9451	13,4992	0,0	0,0
33	0,9448	13,4919	0,0	0,0
34	1,0100	13,0514	88,75	72,8818
35	1,0100	12,4330	44,4	67,2680
36	1,0027	12,1828	14,8	30,0
37	1,0100	11,7611	29,59	41,4528
38	0,9706	13,3803	14,4	30,0

Fonte: Elaborado pelo autor (2022)

Tabela 41 – Inicialização da co-simulação para sistemas de transmissão e 34 barras - Sistema de 11 barras

Barra	V[p.u]	θ [°]	P_G [MW]	Q_G [Mvar]
1	1,03	38,866	700,0	155,9888
2	1,01	29,177	700,0	164,6951
3	1,03	10,471	719,10	171,0620
4	1,01	0,000	669,6537	186,6729
5	1,0111	32,434	0,0	0,0
6	0,9896	22,474	0,0	0,0
7	0,9816	14,283	0,0	0,0
8	0,9571	-0,183	0,0	0,0
9	0,9744	-14,636	0,0	0,0
10	0,9854	-6,439	0,0	0,0
11	1,0091	3,850	0,0	0,0

Fonte: Elaborado pelo autor (2022)

Tabela 42 – Inicialização da co-simulação para sistemas de transmissão e 34 barras - Sistema de 34 barras

Barra	V_A [p.u]	θ_A [°]	V_B [p.u]	θ_B [°]	V_C [p.u]	θ_C [°]
sourcebus	0,9816	14,292	0,9816	-105,708	0,9816	134,292
800	0,9816	-15,709	0,9816	-135,709	0,9816	104,291
802	0,979	-15,758	0,9798	-135,787	0,9799	104,231
806	0,9772	-15,792	0,9787	-135,84	0,9788	104,191
808	0,9439	-16,449	0,9585	-136,826	0,959	103,464
810	-	-	0,9584	-136,827	-	-
812	0,9052	-17,277	0,9362	-137,98	0,9357	102,602
814	0,8745	-17,97	0,9187	-138,919	0,9171	101,9
814r	0,9617	-17,992	1,0104	-138,938	1,0087	101,881
850	0,9617	-17,992	1,0104	-138,938	1,0087	101,881
816	0,9613	-17,997	1,0102	-138,946	1,0084	101,875
818	0,9603	-18,0	-	-	-	-
824	0,9522	-18,101	1,0005	-139,19	0,9999	101,619
820	0,9355	-18,06	-	-	-	-
822	0,9322	-18,064	-	-	-	-
826	-	-	1,0003	-139,191	-	-
828	0,9515	-18,111	0,9998	-139,209	0,9992	101,598
830	0,9335	-18,349	0,9823	-139,685	0,982	101,087
854	0,9331	-18,355	0,9819	-139,697	0,9816	101,074
832	0,9922	-18,812	1,0462	-140,567	1,0466	100,124
858	0,9899	-18,867	1,0439	-140,662	1,0445	100,016
834	0,9873	-18,934	1,0411	-140,774	1,042	99,886
860	0,9868	-18,926	1,0407	-140,774	1,0416	99,884
842	0,9872	-18,938	1,0411	-140,779	1,042	99,881
836	0,9866	-18,924	1,0403	-140,776	1,0415	99,884
840	0,9866	-18,924	1,0403	-140,776	1,0415	99,885
862	0,9866	-18,923	1,0403	-140,776	1,0415	99,884
844	0,987	-18,961	1,0407	-140,805	1,0417	99,854
846	0,9872	-19,004	1,0407	-140,852	1,042	99,804
848	0,9873	-19,01	1,0408	-140,859	1,0421	99,797
852r	0,9922	-18,812	1,0462	-140,567	1,0466	100,124
888	0,956	-20,376	1,0102	-142,132	1,0102	98,629
856	-	-	0,9818	-139,704	-	-
852	0,9022	-18,797	0,9512	-140,551	0,9516	100,14
864	0,9899	-18,867	-	-	-	-
838	-	-	1,0402	-140,778	-	-
890	0,8738	-20,922	0,9351	-143,227	0,9276	97,836

Fonte: Elaborado pelo autor (2022)

APÊNDICE C – ESTIMATIVA DA IMPEDÂNCIA DE THÉVENIN

Esse apêndice visa descrever os passos utilizados para as estimativas da impedância de Thévenin a partir de um sistema representado por sequência positiva e também a partir de um sistema trifásico.

C.1 ESTIMATIVA A PARTIR DE UM SISTEMA EM SEQUÊNCIA POSITIVA

Seja o sistema de potência descrito pela sua relação entre tensões nodais e correntes de injeção na (C.1).

$$\mathbf{Y}_{bus} \mathbf{V} = \mathbf{I} \quad (\text{C.1})$$

Para um sistema elétrico representado por sequência positiva, com n barras, tem-se que \mathbf{V} representa o vetor de tensões em cada nó do sistema com dimensão $n \times 1$, enquanto \mathbf{I} refere-se a corrente injetada em cada elemento, também com dimensão $n \times 1$. E por fim, a matriz de admitância nodal \mathbf{Y}_{bus} , que possui dimensão $n \times n$.

Para se encontrar uma impedância vista de uma barra k , um procedimento útil é através de uma medição realizada na barra de interesse.

Primeiramente, deve-se considerar que, como feito em estudos de curto-circuito, algumas simplificações devem ser feitas, de acordo com [49]. Nesse contexto, os geradores são representados por fontes de tensão \bar{E}_g em série com suas reatâncias sub-transitórias X''_d . Para a medição da impedância de Thévenin, o método consiste em obter uma medida de sensibilidade na barra k , a partir da variação de uma corrente \bar{I}_k injetada na barra k . As demais fontes de corrente do sistema precisam ser abertas. Injetando-se \bar{I}_{k0} , mede-se \bar{V}_{k0} na barra k , enquanto injetando-se \bar{I}_{k1} , obtém-se \bar{V}_{k1} . A impedância de Thévenin \bar{Z}_{th} vista na barra k , é concebida em (C.2).

$$\bar{Z}_{th} = \frac{\Delta \bar{V}_k}{\Delta \bar{I}_k} = \frac{\bar{V}_{k1} - \bar{V}_{k0}}{\bar{I}_{k1} - \bar{I}_{k0}} \quad (\text{C.2})$$

No caso em que o sistema é linear, a estimativa da impedância de Thévenin é fiel para cada duas medições realizadas. Todavia, para medições em sistemas não lineares, a estimativa será um valor aproximado. Da equação nodal de (C.1), tem-se a relação expressa em (C.3).

$$\begin{bmatrix} \bar{Y}_{11} & \cdots & \bar{Y}_{1k} & \cdots & \bar{Y}_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{Y}_{k1} & \cdots & \bar{Y}_{kk} & \cdots & \bar{Y}_{kn} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{Y}_{n1} & \cdots & \bar{Y}_{nk} & \cdots & \bar{Y}_{nn} \end{bmatrix} \begin{bmatrix} \Delta \bar{V}_1 \\ \vdots \\ \Delta \bar{V}_k \\ \vdots \\ \Delta \bar{V}_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \Delta \bar{I}_k \\ \vdots \\ 0 \end{bmatrix} \quad (\text{C.3})$$

Algoritmo 21: Exportação da matriz Y_{bus}

```
dss.run_command(Redirect "file.dss")
dss.Solution.Solve()
dss.run_command("Export Y")
```

Sabendo-se que $\Delta \mathbf{V} = \mathbf{Z}_{bus} \Delta \mathbf{I}$, onde, $\mathbf{Z}_{bus} = \mathbf{Y}_{bus}^{-1}$, tem-se assim, que a impedância de Thévenin \bar{Z}_{th} é a impedância própria da barra k , referente ao elemento \bar{Z}_{kk} . Desse modo, a impedância de Thévenin \bar{Z}_{th} vista na barra k , é vista também em (C.4).

$$\bar{Z}_{th} = \frac{\Delta \bar{V}_k}{\Delta \bar{I}_k} = \bar{Z}_{kk} \quad (\text{C.4})$$

C.2 ESTIMATIVA A PARTIR DE UM SISTEMA TRIFÁSICO

No que diz respeito as estimativas das impedâncias de Thévenin dos sistemas modelados no OpenDSS, serão apresentados os passos utilizados na sequência.

Primeiramente, parte-se da mesma relação apresentada em (C.1). Porém, para o presente trabalho, os sistemas modelados no OpenDSS, são trifásicos. Nesse âmbito, a matriz de admitâncias nodais é de modelagem trifásica, sendo construída a partir das matrizes de admitâncias nodais (matrizes primitivas) de cada elemento que compõe o sistema.

Desse modo, para um sistema elétrico de modelagem trifásica com n barras, tem-se que o vetor de tensões \mathbf{V} apresenta dimensão $3n \times 1$, enquanto \mathbf{I} apresenta dimensão $3n \times 1$. E por fim, a matriz de admitância nodal \mathbf{Y}_{bus} , que possui dimensão $3n \times 3n$.

Uma vantagem que o OpenDSS oferece, é a possibilidade de exportar matriz \mathbf{Y}_{bus} . O seguinte comando pode ser verificado no Algoritmo 21:

Para encontrar a impedância de Thévenin vista a partir do nó k , um procedimento útil pode ser utilizado. O método consiste em realizar uma medição a partir da barra k , para cada fase a , b e c . Nesse caso, uma corrente pode ser injetada em uma das fases, para cada medição. As demais fontes de corrente do sistema precisam ser abertas. Esse é um procedimento comumente utilizado em análises de curto-circuito e muito útil do ponto de vista computacional, principalmente quando aumenta-se o número de nós nos sistemas de distribuição. Sendo assim, para obter a matriz de impedâncias próprias a partir do nó

k , a relação (C.5) pode ser obtida.

$$\begin{bmatrix} (\mathbf{Y}_{11}^*)^{abc} & \dots & (\mathbf{Y}_{1k}^*)^{abc} & \dots & (\mathbf{Y}_{1n}^*)^{abc} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\mathbf{Y}_{k1}^*)^{abc} & \dots & (\mathbf{Y}_{kk}^*)^{abc} & \dots & (\mathbf{Y}_{kn}^*)^{abc} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\mathbf{Y}_{n1}^*)^{abc} & \dots & (\mathbf{Y}_{nk}^*)^{abc} & \dots & (\mathbf{Y}_{nn}^*)^{abc} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^{abc} \\ \vdots \\ \mathbf{V}_k^{abc} \\ \vdots \\ (\mathbf{V}_n)^{abc} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{I}_k^{abc} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (\text{C.5})$$

Aonde \mathbf{I}_k^{abc} é formado pelos vetores de injeção de corrente em cada fase para cada medição nó k , como mostrado em (C.6).

$$\mathbf{I}_k^{abc} = \begin{bmatrix} \bar{I}_{ka} & 0 & 0 \\ 0 & \bar{I}_{kb} & 0 \\ 0 & 0 & \bar{I}_{kc} \end{bmatrix} \quad (\text{C.6})$$

Resolvendo-se (C.5), a matriz de tensões pode ser obtida. Assim, pode ser obtida a submatriz \mathbf{V}_k^{abc} .

Injetando-se correntes unitárias, é válido que $\mathbf{V}_k^{abc} = \mathbf{Z}_{kk}^{abc}$. Nesse sentido, a matriz \mathbf{Z}_{kk}^{abc} é a matriz de impedâncias de Thévenin \mathbf{Z}_{th}^{abc} , vistas a partir do nó k , representada em (C.7).

$$\mathbf{Z}_{th}^{abc} = \begin{bmatrix} \bar{Z}_{aa} & \bar{Z}_{ab} & \bar{Z}_{ac} \\ \bar{Z}_{ba} & \bar{Z}_{bb} & \bar{Z}_{bc} \\ \bar{Z}_{ca} & \bar{Z}_{cb} & \bar{Z}_{cc} \end{bmatrix} \quad (\text{C.7})$$

Levando em consideração o interesse em se obter a componente de sequência positiva, da impedância de Thévenin a partir do nó de interconexão, a relação de Fortescue pode ser aplicada em (C.7), como visto em (C.8).

$$\mathbf{Z}_{th-012} = \mathbf{A}^{-1} \cdot \mathbf{Z}_{th-abc} \cdot \mathbf{A} \quad (\text{C.8})$$

Onde \mathbf{A} representa a matriz de transformação de Fortescue, descrita por (C.9).

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \quad (\text{C.9})$$

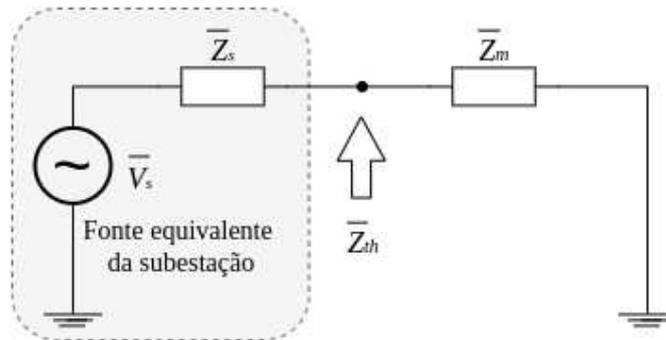
Sendo $a = 1e^{j\frac{2\pi}{3}}$. Assim, é possível encontrar a componente de sequência positiva \bar{Z}_{th-1} .

É importante salientar que o OpenDSS exporta a matriz \mathbf{Y}_{bus} em siemens. Desse modo, deve-se transformar \bar{Z}_{th-1} dada em ohms, para pu, dividindo-se pela impedância base Z_{base} , dada por (C.10):

$$Z_{base} = \frac{V_{base}^2}{S_{base}} \quad (\text{C.10})$$

Outra consideração importante, é que a barra de referência do OpenDSS pode ser representada por uma fonte de tensão em série com uma impedância \bar{Z}_s , como mostrado na Figura 47.

Figura 47 – Impedância de Thévenin do sistema de distribuição vista do nó de interconexão



Fonte: Elaborado pelo autor (2016)

Dessa forma, a impedância de Thévenin \bar{Z}_m vista do nó de interconexão, pode ser escrita por (C.11):

$$\bar{Z}_m = \frac{1}{\frac{1}{\bar{Z}_{th1-pu-pu}} - \frac{1}{\bar{Z}_s}} = \frac{\bar{Z}_s \bar{Z}_{th1-pu}}{\bar{Z}_s - \bar{Z}_{th1-pu}} \quad (C.11)$$

Deste modo, \bar{Z}_m representa o valor da impedância de Thévenin do sistema de distribuição vista a partir do nó de interconexão.

APÊNDICE D – FLUXO DE POTÊNCIA DO OpenDSS

Para o cálculo de fluxo de potência de sistemas de transmissão, métodos tradicionais como Gauss-Seidel, Newton Raphson e Desacoplado Rápido, podem ser utilizados. Nesse âmbito, os referidos métodos representam o sistema trifásico utilizando somente o equivalente monofásico, devido a suposição da rede ser equilibrada. Todavia, no que diz respeito aos sistemas de distribuição, esses métodos não são recomendados, devido a algumas características dos referidos sistemas como: desequilíbrios, cargas desbalanceadas, inserção de GDs. Assim, há métodos de solução para cálculo de fluxo de potência de sistemas de distribuição, como por exemplo, o método de varredura (*Backward-Forward Sweep*) e o método de desacoplamento da matriz de admitância nodal [38].

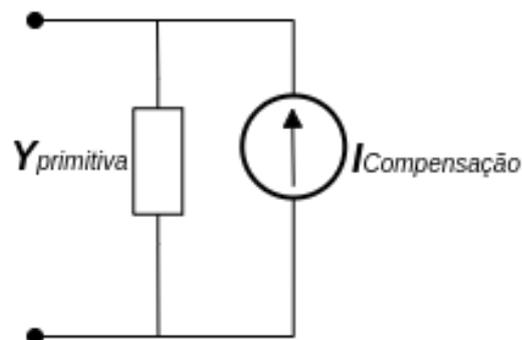
O OpenDSS utiliza métodos polifásicos para a solução do fluxo de potência, sendo o método Normal, o algoritmo padrão utilizado pelo OpenDSS. É um método mais rápido computacionalmente, e trabalha com a técnica de desacoplamento da matriz de admitância nodal.

Nesse método, as tensões nas barras (\bar{V}_n) são tratadas como incógnitas e as correntes (\bar{I}_n) em função dessas tensões, ou seja, as fontes lineares de tensão são representadas pelos seus equivalentes de Norton, enquanto os geradores e cargas não lineares são representados por uma admitância em paralelo com uma fonte de corrente, também chamada de fonte de corrente de compensação, como mostrado na Figura 48.

Essas correntes de compensação são correntes de ajuste, representadas pela diferença entre as correntes terminais calculadas e as correntes do ramo da matriz primitiva do elemento.

No que tange ao método de solução, o OpenDSS constrói a matriz de admitâncias nodais de cada elemento, e entrega para o *KLU solve*, que é um solucionador de matriz esparsa, que decompõe a matriz de admitância nodal e resolve o sistema linear associado.

Figura 48 – Modelo de corrente de compensação de elementos de conversão de energia



Fonte: Elaborado pelo autor (2022)

Ademais, o OpenDSS deve fornecer o mapeamento da conexão dos elementos, as correntes injetadas e as correntes de compensação para o *KLUSolve*, que retorna as tensões nodais da próxima iteração, as quais são utilizadas para calcular as novas correntes injetadas e as correntes de compensação. Assim, os principais passos serão descritos a seguir:

1. Fornecimento dos valores iniciais, próximos a solução: As tensões nodais são calculadas, desconsiderando-se as cargas e geradores, e considerando-se somente o elemento da fonte equivalente do sistema (elemento *Circuit*). Assim, o OpenDSS fornece ao solucionador, as matrizes primitivas de cada elemento, e do elemento *Circuit*, com a corrente injetada. Posteriormente, o solucionador entrega as tensões nodais a serem utilizadas na próxima iteração, através da solução do sistema de (D.1):

$$\begin{bmatrix} \bar{Y}_{11} & \cdots & \bar{Y}_{1k} & \cdots & \bar{Y}_{1n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \bar{Y}_{k1} & \cdots & \bar{Y}_{kk} & \cdots & \bar{Y}_{kn} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \bar{Y}_{n1} & \cdots & \bar{Y}_{nk} & \cdots & \bar{Y}_{nn} \end{bmatrix} \begin{bmatrix} \bar{V}_1^{(1)} \\ \vdots \\ \bar{V}_k^{(1)} \\ \vdots \\ \bar{V}_n^{(1)} \end{bmatrix} = \begin{bmatrix} \bar{I}_1^{(0)} \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (\text{D.1})$$

2. Para cada iteração h , são calculadas correntes injetadas e de compensação nas cargas e nos geradores;
3. Entrega das correntes injetadas e de compensação para o solucionador, que calcula as tensões nodais \bar{V}_{n+1} , da próxima iteração h , como mostrado na (D.2)

$$\begin{bmatrix} \bar{Y}_{11} & \cdots & \bar{Y}_{1k} & \cdots & \bar{Y}_{1n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \bar{Y}_{k1} & \cdots & \bar{Y}_{kk} & \cdots & \bar{Y}_{kn} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \bar{Y}_{n1} & \cdots & \bar{Y}_{nk} & \cdots & \bar{Y}_{nn} \end{bmatrix} \begin{bmatrix} \bar{V}_1^{(h+1)} \\ \vdots \\ \bar{V}_k^{(h+1)} \\ \vdots \\ \bar{V}_n^{(h+1)} \end{bmatrix} = \begin{bmatrix} \bar{I}_1^{(h)} \\ \vdots \\ \bar{I}_k^{(h)} \\ \vdots \\ \bar{I}_n^{(h)} \end{bmatrix} \quad (\text{D.2})$$

4. Os passos de (2) a (3) são repetidos até $\|\bar{V}_{n+1} - \bar{V}_n\| < \varepsilon$, , onde ε é uma tolerância aceitável.

É importante ressaltar, que a matriz de admitância nodal \mathbf{Y}_{bus} do sistema normalmente não é reconstruída durante esse processo, portanto, as iterações ocorrem rapidamente. Ganha-se assim, em eficiência computacional. A matriz de admitâncias nodais é modificada apenas quando capacitores e/ou indutores *shunt* são chaveados e, além disso, quando os *taps* dos transformadores são alterados de uma simulação para outra, ou seja, a matriz \mathbf{Y}_{bus} deve ser modificada quando alguma das matrizes primitivas se altera, pois para os elementos de conversão de energia, as correntes de compensação resolvem o problema [19, 38].

APÊNDICE E – COEFICIENTE DE REFLEXÃO PARA UMA LINHA DE TRANSMISSÃO IDEAL

A linha de transmissão sem perdas pode ser descrita pelo sistema de equações diferenciais de tensão e corrente, dado por (E.1).

$$\frac{\partial v^2}{\partial t^2} = c^2 \frac{\partial v^2}{\partial x^2} \quad (\text{E.1a})$$

$$\frac{\partial i^2}{\partial t^2} = c^2 \frac{\partial i^2}{\partial x^2} \quad (\text{E.1b})$$

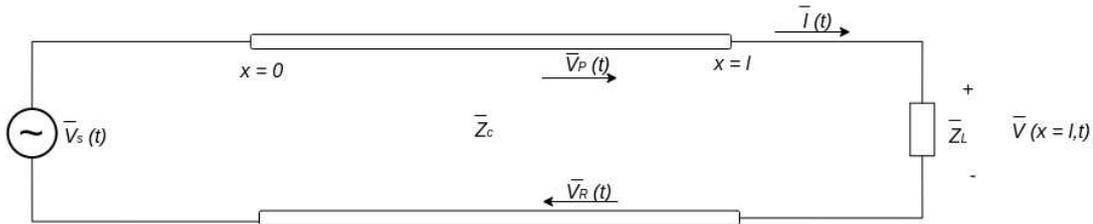
A solução de (E.1), em sua representação fasorial, pode ser dada por (E.2) [50], onde c é a velocidade de propagação.

$$\bar{V}(x, t) = \bar{V}_P(t - l/c) + \bar{V}_R(t + l/c) \quad (\text{E.2a})$$

$$\bar{I}(x, t) = \bar{I}_P(t - l/c) + \bar{I}_R(t + l/c) \quad (\text{E.2b})$$

Todas as linhas de transmissão tem um fim. Na Figura 49, vê-se uma onda viajando positivamente sobre um resistor R_L em $x = l$. A onda refletida, retornará em direção a fonte em $x = 0$ como uma onda \bar{V}_R .

Figura 49 – Sistema com linha de transmissão



Fonte: Elaborado pelo autor (2022)

Na posição $x = l$, a tensão no final da linha pode ser obtida em (E.3).

$$\bar{V}(l, t) = \bar{V}_P(t - l/c) + \bar{V}_R(t + l/c) \quad (\text{E.3})$$

Sabendo-se que a tensão na carga também pode ser descrita por $v(l, t) = i_L(l, t)\bar{Z}_L$, tem-se em (E.4).

$$\bar{V}(l, t) = \frac{\bar{Z}_L}{\bar{Z}_c} (V_P(t - l/c) - V_R(t + l/c)) \quad (\text{E.4})$$

Igualando-se (E.3) a (E.4), e dividindo a expressão por $\bar{V}_P(t - l/v)$, a relação (E.5) pode ser obtida:

$$\frac{\bar{V}_P(t - l/c)}{\bar{V}_P(t - l/c)} + \frac{\bar{V}_R(t + l/c)}{\bar{V}_P(t - l/c)} = \frac{\bar{Z}_L}{\bar{Z}_c} \left(\frac{\bar{V}_P(t - l/c)}{\bar{V}_P(t - l/c)} - \frac{\bar{V}_R(t + l/c)}{\bar{V}_P(t - l/c)} \right) \quad (\text{E.5})$$

A amplitude da onda de tensão regressiva (\bar{V}_R) em relação a onda de tensão progressiva (\bar{V}_P) pode ser chamada de coeficiente de reflexão $\bar{\Gamma}_L$. Dessa forma, (E.5) pode ser reescrita em (E.6).

$$1 + \bar{\Gamma}_L = \frac{\bar{Z}_L}{\bar{Z}_c} (1 - \bar{\Gamma}_L) \quad (\text{E.6})$$

Finalmente, o coeficiente de reflexão $\bar{\Gamma}_L$ pode ser obtida de (E.7).

$$\bar{\Gamma}_L = \frac{\bar{Z}_L - \bar{Z}_c}{\bar{Z}_L + \bar{Z}_c} \quad (\text{E.7})$$

APÊNDICE F – ESPAÇO DE ESTADOS PARA SISTEMAS NO DOMÍNIO DISCRETO

O modelo matemático de um sistema no domínio discreto pode ser escrito em termos de uma fórmula recursiva, utilizando equações de diferença como em (F.1):

$$\mathbf{x}[n] = \mathbf{A}\mathbf{x}[n-1] + \mathbf{B}\mathbf{u}[n-1] \quad (\text{F.1a})$$

$$\mathbf{y}[n] = \mathbf{C}\mathbf{x}[n-1] + \mathbf{D}\mathbf{u}[n-1] \quad (\text{F.1b})$$

Onde \mathbf{x} , \mathbf{u} , \mathbf{y} são vetores com variáveis discretas e \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} são matrizes constantes. O índice n refere-se ao índice de amostragem. Levando em consideração as amostras $n = 0, 1, 2, \dots$ tem-se que:

$$\mathbf{x}[1] = \mathbf{A}\mathbf{x}[0] + \mathbf{B}\mathbf{u}[0] \quad (\text{F.2a})$$

$$\mathbf{x}[2] = \mathbf{A}\mathbf{x}[1] + \mathbf{B}\mathbf{u}[1] = \mathbf{A}^2\mathbf{x}[0] + \mathbf{A}\mathbf{B}\mathbf{u}[0] + \mathbf{B}\mathbf{u}[1] \quad (\text{F.2b})$$

$$\mathbf{x}[3] = \mathbf{A}\mathbf{x}[2] + \mathbf{B}\mathbf{u}[2] = \mathbf{A}^3\mathbf{x}[0] + \mathbf{A}^2\mathbf{B}\mathbf{u}[0] + \mathbf{A}\mathbf{B}\mathbf{u}[1] + \mathbf{B}\mathbf{u}[2] \quad (\text{F.2c})$$

$$\vdots \quad (\text{F.2d})$$

$$\mathbf{x}[n] = \mathbf{A}^n\mathbf{x}[0] + \sum_{k=0}^{n-1} \mathbf{A}^{n-k}\mathbf{B}\mathbf{u}[k] \quad (\text{F.2e})$$

De (F.2e), a sequência $\mathbf{x}[n]$ pode ser escrita como a soma de duas parcelas: $\mathbf{x}[n] = \mathbf{x}_h[n] + \mathbf{x}_p[n]$, onde $\mathbf{x}_h[n]$ representa a solução homogênea e $\mathbf{x}_p[n]$ refere-se a uma solução particular. As referidas parcelas estão denotadas em (F.3):

$$\mathbf{x}_h[n] = \mathbf{A}^n\mathbf{x}[0] \quad (\text{F.3a})$$

$$\mathbf{x}_p[n] = \sum_{k=0}^{n-1} \mathbf{A}^{n-k}\mathbf{B}\mathbf{u}[k] \quad (\text{F.3b})$$

As soluções em (F.3) podem ser melhor avaliadas a partir da decomposição da matriz de transição \mathbf{A} em termos de seus autovalores e autovetores. Contanto que \mathbf{A} possua autovalores distintos, cada autovalor $\bar{\lambda}_i$ está associado ao autovetor \mathbf{v}_i como disposto em (F.4).

$$\mathbf{A}\mathbf{v}_i = \bar{\lambda}_i\mathbf{v}_i \quad (\text{F.4})$$

Combinando todos os autovalores e autovetores de \mathbf{A} na forma matricial, tem-se a relação mostrada em (F.5)

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \quad (\text{F.5})$$

em que,

$$\mathbf{V} = [\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_n] \quad (\text{F.6a})$$

$$\mathbf{\Lambda} = \text{diag}(\bar{\lambda}_1, \cdots, \bar{\lambda}_n) \quad (\text{F.6b})$$

De (F.5), a matriz de transição \mathbf{A} pode ser obtida como visto em (F.7).

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} \quad (\text{F.7})$$

De (F.3a), sabendo-se que $\mathbf{A}^n = (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})^n$, é válida a relação mostrada em (F.8).

$$\left(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}\right)^n = \underbrace{\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}}_1 \cdot \underbrace{\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}}_2 \cdots \underbrace{\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}}_n = \mathbf{V}\mathbf{\Lambda}^n\mathbf{V}^{-1} \quad (\text{F.8})$$

Como $\mathbf{\Lambda}^n$ é uma matriz diagonal, tem-se que:

$$\mathbf{\Lambda}^n = \text{diag}(\bar{\lambda}_1^n, \cdots, \bar{\lambda}_n^n) \quad (\text{F.9})$$

Por fim, a relação de (F.3), pode ser reescrita em (F.10).

$$\mathbf{x}_h[n] = \mathbf{V}\mathbf{\Lambda}^n\mathbf{V}^{-1}\mathbf{x}[0] \quad (\text{F.10a})$$

$$\mathbf{x}_p[n] = \sum_{k=0}^n \mathbf{V}\mathbf{\Lambda}^{n-k}\mathbf{V}^{-1}\mathbf{u}[k] \quad (\text{F.10b})$$

A estabilidade de (F.2e) é determinada avaliando-se $\mathbf{\Lambda}^n$. Se os autovalores $\bar{\lambda}_i^n$ satisfazem a condição $|\bar{\lambda}_i|^n < 1$, o sistema é assintoticamente estável. Por outro lado, é instável se $|\bar{\lambda}_i|^n > 1$. Se $|\bar{\lambda}_i|^n = 1$, o sistema pode ser estável ou instável, dependendo da estrutura de \mathbf{A}^n .