

Estratégia de Simulação de Transitórios Eletromecânicos para Sistemas de Transmissão Integrados a FMUs

Ricardo Proba Fava* Marcelo Aroca Tomim**
Igor Borges de Oliveira Chagas***

* Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Juiz de Fora, MG, Brasil (ricardo.proba@estudante.ufjf.br).

** Departamento de Energia Elétrica, Universidade Federal de Juiz de Fora, MG, Brasil (marcelo.tomim@ufjf.br).

*** Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Juiz de Fora, MG, Brasil (igor.borges@engenharia.ufjf.br).

Abstract: This paper presents the implementation of an implicit alternating method for the solution of the transient stability problem for a large electrical transmission system. For that, the generating units are implemented by Functional Mock-up Units (FMU) and the transmission system is computed in *Python*. In this way, the electrical power system is split into smaller subsystems, while preserving the accuracy of a monolithic simulation.

Resumo: Este trabalho apresenta a implementação de um método alternado implícito à solução do problema de estabilidade transitória para um sistema elétrico de grande porte. Para isso, as unidades geradoras são modeladas através de *Functional Mock-up Units* (FMU) e o sistema de transmissão é solucionado em *Python*. Dessa forma, o sistema elétrico de potência é resolvido através de subsistemas menores, preservando a precisão de uma simulação monolítica.

Keywords: Distributed Energy Resources; Model Exchange; FMU; Alternating Iterative Method; Time Simulation; Transient Stability.

Palavras-chaves: Recursos Energéticos Distribuídos; Intercâmbio de Modelos; FMU; Método Iterativo Alternado; Simulação no Tempo; Estabilidade Transitória.

1. INTRODUÇÃO

Nos últimos anos, com o desenvolvimento tecnológico e o crescimento populacional contínuo dos centros urbanos, verifica-se uma demanda por energia cada vez maior nos sistemas elétricos de potência, o que tem acarretado profundas mudanças na sua operação. Além disso, as construções de novas usinas de grande porte (por exemplo, as hidrelétricas) têm sido contestadas, devido aos impactos ambientais associados, ao custo da energia gerada e ao surgimento de novas tecnologias com potencial de reduzir emissões de poluentes. Esses são alguns fatores que têm contribuído para a valorização dos Recursos Energéticos Distribuídos (RED) no âmbito mundial. Como exemplo, pode-se citar o crescimento da geração fotovoltaica, explicado em (de Souza et al., 2020).

Com foco na análise dos sistemas elétricos de potência nesse novo contexto, podem ser necessários modelos matemáticos específicos para o tipo do RED sob avaliação, para que as propriedades e os comportamentos dinâmicos intrínsecos de cada tipo de fonte de energia sejam corretamente capturados, conforme descrito em (de Oliveira Chagas e Tomim, 2022). Além disso, os profissionais da área vêm experimentando a necessidade de ambientes e ferramentas computacionais que permitam o acoplamento de diferentes modelos e domínios em uma mesma simulação

de um sistema elétrico. Integrando-se adequadamente múltiplas ferramentas de simulação, é possível simular grandes sistemas de forma heterogênea, resultando em uma plataforma única de simulação que permite a reutilização de ferramentas comprovadamente confiáveis ao longo de anos de desenvolvimento.

Com base nesse panorama, o presente trabalho tem como objetivo apresentar a estrutura de um programa de estabilidade transitória cujos elementos e subsistemas são atribuídos a ferramentas computacionais distintas. Na presente análise, são considerados fundamentalmente dois subsistemas: a rede elétrica de transmissão e seus equipamentos dinâmicos associados. Para o presente trabalho, de forma a ilustrar a metodologia, geradores elétricos equivalentes de usinas são considerados como integrantes do subsistema de equipamentos dinâmicos. Outros equipamentos, tais como cargas dinâmicas e elementos FACTS, apresentam potencial de integrar esse subsistema.

Estabelece-se que a transmissão, bem como suas cargas lineares e não-lineares, é representada pelo seu circuito monofásico equivalente de sequência positiva através da linguagem *Python*. As máquinas elétricas compõem um subsistema específico porque, ao contrário da transmissão, possuem uma natureza dinâmica. Para essa modelagem dinâmica, adota-se a *Functional Mock-up Interface* (FMI), uma interface padronizada e gratuita que visa o

acoplamento entre modelos matemáticos e simuladores. Os modelos, ou bibliotecas, que adotam a interface FMI são chamados *Functional Mock-up Units* (FMU), e podem ser de dois tipos distintos: *Co-Simulation* (CS) e *Model Exchange* (ME). Enquanto o primeiro tipo tem um *solver* próprio, o segundo deve ser resolvido por um *solver* do ambiente de simulação, conforme mostrado em (Modelica Association, 2021).

Com a separação proposta, tem-se a vantagem de desmembrar sistemas complexos em subsistemas menores (Pallensky et al., 2017). Esse desmembramento acarreta uma especialização da solução, dado que cada subsistema pode ser modelado em domínios distintos, como por exemplo domínios da frequência e do tempo. Consequentemente, pode-se esperar maior flexibilidade na modelagem, possibilidade de reutilização do modelo e potenciais ganhos de desempenho computacional.

Dentro dessa linha de estudo, alguns trabalhos demonstraram alguns desses aspectos relacionados à cossimulação. Em (Silva, 2020), modelos matemáticos de componentes de um sistema de geração eólica foram desenvolvidos em linguagem Modelica e compilados em FMUs. Depois, foram integrados a um algoritmo mestre, que coordenou e executou a cossimulação em um ambiente de programação de linguagem Python. Nesse ambiente, as simulações foram possíveis graças à divisão do sistema em subsistemas compostos por aerogeradores e rede elétrica. Resultados evidenciaram tanto a precisão como ganhos computacionais via paralelização.

Vislumbrando aplicações futuras, (Theodoro et al., 2018) apresentou uma simulação híbrida conectando dois subsistemas. O primeiro subsistema foi do tipo EMT (transitórios eletromagnéticos), que continha usinas fotovoltaicas modeladas em instâncias distintas do Matlab, enquanto o segundo subsistema foi do tipo quasi-estático no domínio da frequência, que continha uma rede de distribuição modelada pelo OpenDSS (*Open Distribution System Simulator*). Essa modelagem híbrida se mostrou essencial, uma vez que ela economizou esforço de processamento e permitiu a reutilização de ferramentas já amplamente testadas. Já em (de Oliveira Chagas e Tomim, 2022), foram apresentados métodos de cossimulação onde o sistema de transmissão foi compilado em uma FMU e depois acoplado a redes de distribuição do OpenDSS. Esse cenário mostrou a possibilidade de integração de subsistemas modelados em domínios distintos entre si.

Diferentemente dos estudos relatados anteriormente, este trabalho busca empregar modelos com a interface do tipo ME para a solução dos equipamentos dinâmicos. Dessa forma, este artigo é organizado da seguinte forma: na Seção 2, apresenta-se o método alternado utilizado pelo programa escrito em linguagem Python. Na Seção 3, são apresentados os fundamentos da simulação com FMUs para intercâmbio de modelos adotados neste trabalho. Na Seção 4, resultados obtidos para uma rede de transmissão de 11 barras são mostrados e analisados. Por último, na Seção 5, apresentam-se as conclusões do presente trabalho e as propostas de trabalhos futuros.

2. FUNDAMENTOS DA SIMULAÇÃO ELETROMECAÂNICA

Um sistema elétrico pode ser caracterizado através de um sistema de equações algébrico-diferenciais como denotado em (1).

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{v}) \quad (1a)$$

$$\mathbf{Y} \mathbf{v} = \mathbf{i}(\mathbf{x}, \mathbf{v}) \quad (1b)$$

O conjunto (1a) descreve o comportamento dinâmico de equipamentos dinâmicos (geradores, cargas dinâmicas, elementos FACTS, etc.) e seus controladores associados. O conjunto (1b) consiste em um sistema de equações algébricas que descrevem as restrições impostas pela rede de transmissão passiva assim como cargas funcionais dependentes da tensão. Comumente, esta rede é representada através da matriz \mathbf{Y} de admitâncias nodais. Nessas equações, \mathbf{x} é o vetor de variáveis dinâmicas (ou variáveis de estado), \mathbf{v} é o vetor de tensões nodais e \mathbf{i} é o vetor de correntes de injeções nodais (Kundur, 1994; Tomim, 2009).

A estrutura de (1b) se altera em instantes de tempo denominados t_e , nos quais ocorrem eventos. Por exemplo, em uma contingência composta por aplicação e remoção de curto-circuito através de abertura de linha, observam-se três eventos, de acordo com os quais a matriz \mathbf{Y} deve ser alterada. Cada alteração individual exige o cálculo de duas soluções no instante t_e : uma em t_e^- (antes do evento) e outra em t_e^+ (após o evento). Com isso, pode ocorrer uma descontinuidade em \mathbf{v} , ou seja, tem-se $\mathbf{v}(t_e^+) \neq \mathbf{v}(t_e^-)$. Por outro lado, as variáveis de estado em \mathbf{x} em geral são contínuas, implicando em $\mathbf{x}(t_e^+) = \mathbf{x}(t_e^-)$, conforme mostrado em (Dommel e Sato, 1972).

Para a solução numérica do sistema sintetizado em (1), pode-se empregar o método de integração trapezoidal, que aproxima a integral para um passo de tempo suficientemente próximo à área de um trapézio. Dessa forma, a equação em (1a) é discretizada conforme mostrado em (2).

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{x}(t - \Delta t) + \int_{t-\Delta t}^t \mathbf{f}(\mathbf{x}(\tau), \mathbf{v}(\tau)) d\tau \approx \\ &\approx \mathbf{x}(t - \Delta t) + \frac{1}{2} [\mathbf{f}(t) + \mathbf{f}(t - \Delta t)] \Delta t \end{aligned} \quad (2)$$

Separando os termos presentes (instante t) dos termos passados (instante $t - \Delta t$), tem-se a nova expressão de $\mathbf{x}(t)$ apresentada em (3a), onde o vetor $\mathbf{x}_h(t)$ corresponde ao termo histórico das variáveis de estado (equação (3b)). O termo histórico é totalmente conhecido para a solução referente ao instante t e precisa ser atualizado a cada passo de integração.

$$\mathbf{x}(t) = \frac{\Delta t}{2} \mathbf{f}(t) + \mathbf{x}_h(t) \quad (3a)$$

$$\mathbf{x}_h(t) = \mathbf{x}(t - \Delta t) + \frac{\Delta t}{2} \mathbf{f}(t - \Delta t) \quad (3b)$$

Combinando as expressões (3a), (3b) e (1b), obtém-se o modelo discretizado do sistema de potência que deve ser solucionado a cada instante t . Para isso, pode-se empregar métodos de solução de sistemas não-lineares tais como o de Newton. Outra alternativa, explorada em maiores detalhes no momento, é o método alternado que, para um instante t específico, alterna entre soluções do conjunto dinâmico e algébrico até que a convergência na interface dos modelos seja verificada.

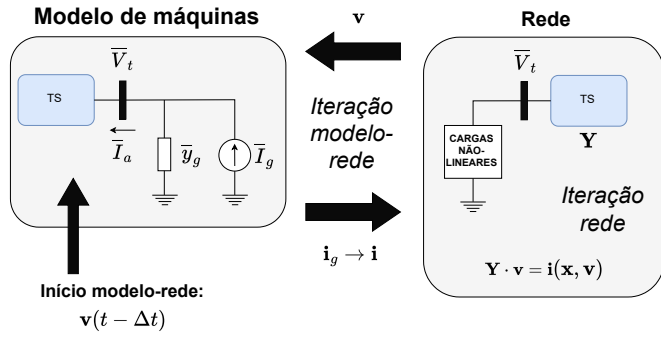


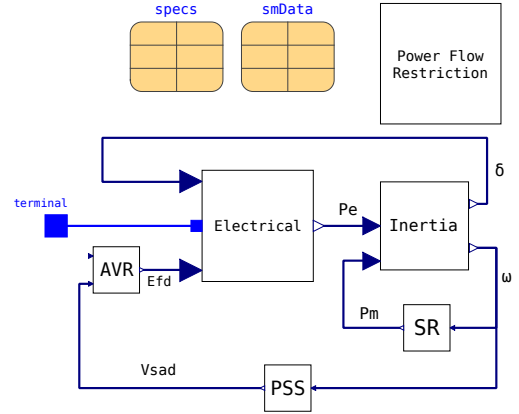
Figura 1. Método alternado do conjunto modelo-rede.

Em linhas gerais, o método alternado do conjunto modelo-rede pode ser ilustrado na Figura 1, que evidencia o equivalente de Norton para os geradores do sistema. Também é mostrado que o método alternado do instante t inicia a primeira iteração com base nas tensões nodais no instante $t - \Delta t$, que serve como entrada para o bloco “Modelo de máquinas” (dinâmico). Na sequência, calculam-se as variáveis de estado através da solução de (1a), obtida pelo método trapezoidal. Com as variáveis de estado calculadas, pode-se obter as injeções de corrente \mathbf{i} , que são necessárias para a solução da equação (1b). Nesse momento, cada equipamento, dinâmico ou estático, é caracterizado através de um equivalente de Norton. Dentro do bloco “Rede”, são calculadas novas tensões nodais \mathbf{v} , que são retornadas ao bloco “Modelo de máquinas”, para que uma nova iteração seja iniciada. As iterações para o instante t se encerram quando a diferença entre duas soluções sucessivas de \mathbf{x} e \mathbf{v} respeitar determinadas tolerâncias de convergência.

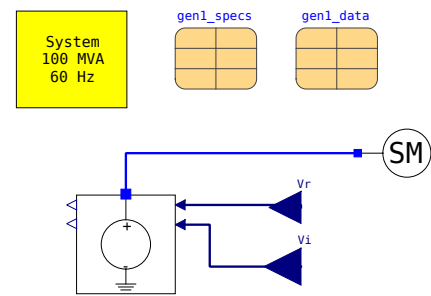
3. SIMULAÇÃO ELETROMECAÂNICA COM FMUS

Conforme citado anteriormente, o padrão FMI define uma interface a ser implementada por uma biblioteca de acesso dinâmico chamada FMU (*Functional Mockup Unit*). Essa interface padronizada é chamada através de um ambiente de simulação de forma que uma ou mais instâncias de um modelo específico (associado a uma FMU) possam ser criadas e resolvidas individualmente ou em conjunto com outros modelos. Neste trabalho, dentro de um ambiente de simulação desenvolvido em *Python*, são desenvolvidas FMUs que contêm os modelos das máquinas elétricas que devem ser acopladas à rede, assim como seus controladores, como reguladores automáticos de tensão, reguladores de velocidade e estabilizadores de potência.

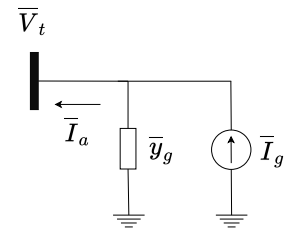
Antes de gerar as FMUs, os modelos das máquinas síncronas são implementados através do programa OMEdit, que proporciona uma interface gráfica para desenvolvimento de modelos matemáticos complexos como o mostrado na Figura 2. A Figura 2a resume os blocos principais que compõem o modelo da máquina, a saber, parte elétrica (*Electrical*), parte mecânica (*Inertial*), regulador de tensão (*AVR*), regulador de velocidade (*SR*) e estabilizador de potência (*PSS*). O bloco *Power Flow Restriction* auxilia na definição da condição de operação da máquina obtida a partir da solução do fluxo de potência. Os demais blocos contêm parâmetros elétricos e definições do tipo de barra a ser utilizada para uma determinada máquina



(a) Modelo genérico de máquina síncrona.



(b) Máquina síncrona com interface.



(c) Equivalente de Norton da máquina.

Figura 2. Máquina síncrona modelada nas FMUs.

no fluxo de potência. A Figura 2c ilustra a estrutura do modelo da máquina compilada na FMU. Nota-se a máquina conectada a uma fonte de tensão controlada, que permite a interface desse modelo com a rede distribuição. As entradas da FMU consistem na tensão terminal nas suas componentes real (V_r) e imaginária (V_i). No cálculo realizado pela FMU, a tensão sub-transitória da máquina é obtida a partir das condições internas do modelo e da tensão terminal definida pela rede elétrica. Esta tensão, junto com a reatância sub-transitória da máquina, compõe o equivalente de Norton conectado à rede. Detalhes desse modelo são apresentados por Kundur (1994).

Após a implementação do modelo, pode-se gerar as FMUs correspondentes empregando o compilador de modelos escritos em linguagem Modelica, disponibilizado pela plataforma OpenModelica ([Open Source Modelica Consortium](https://openmodelica.org/), 2023). Para a utilização da FMU gerada, utiliza-se a biblioteca PyFMI (Andersson et al., 2016), projetada para fornecer uma interface de alto nível e de fácil carregamento de FMUs em *Python*. O algoritmo mestre do simulador é

responsável por definir as condições iniciais do sistema, estabelecer e sincronizar a comunicação entre as FMUs e o subsistema referente à rede de transmissão. Além disso, o algoritmo mestre realiza a integração numérica das FMUs, já que este trabalho adota FMUs para intercâmbio de modelos.

Em suma, o protótipo de programa de estabilidade transitória para sistemas de potência deste trabalho pode ser estruturado da seguinte forma: no ambiente *Python*, soluciona-se o sistema de transmissão passivo através do método nodal, enquanto um algoritmo de integração numérica baseado no método trapezoidal serve como *solver* das FMUs. Para a integração da solução das FMUs com a solução de rede, adota-se o método alternado, conforme detalhado na seção anterior.

Na sequência, a estratégia proposta é apresentada em maior detalhamento:

- (i) Estabelecimento das condições iniciais: solução de fluxo de potência para a obtenção de $\mathbf{v}(0)$. Neste trabalho, essa solução é obtida por um programa próprio baseado no método iterativo de *Newton-Raphson*.
- (ii) Inicialização dos equipamentos dinâmicos (geradores) e estáticos (cargas) tendo como premissa básica derivadas nulas, ou seja, $\dot{\mathbf{x}} = 0$.
 - Geradores acoplados à rede de transmissão e modelados através de equivalentes de Norton. Essa modelagem possui a vantagem de não exigir um aumento das dimensões da matriz de admitância nodal \mathbf{Y} , exigindo menor esforço computacional da simulação.
 - Cargas estáticas ou funcionais representadas por modelos ZIP. Essa modelagem intrinsecamente algébrica depende apenas da magnitude das tensões terminais da rede. Essas cargas possuem parcelas de impedância constante (\mathbf{Z}), corrente constante (\mathbf{I}) e potência constante (\mathbf{P}). Da mesma forma que os geradores, essas cargas também permitem uma representação através de equivalentes de Norton, que é atualizado junto com a rede.

Os passos de (i) a (ix) caracterizam o método alternado do conjunto modelo-rede, conforme apresentado no algoritmo 1.

- (i) Extrapolação das tensões nodais: conforme a equação (4), é realizada com base em valores de um instante anterior ($t - \Delta t$) e de dois instantes anteriores ($t - 2\Delta t$). Possui a vantagem de aumento da velocidade de convergência, conforme mostrado em (Stott, 1979).

$$\mathbf{v}^{(\text{extrapol})}(t) = \frac{\mathbf{v}^2(t - \Delta t)}{\mathbf{v}(t - 2\Delta t)} \quad (4)$$

- (ii) Solução das equações diferenciais em (1a) para obtenção de $\mathbf{x}(t)$: através de qualquer técnica de integração numérica. Neste trabalho, as equações diferenciais em (1a) estão compiladas em FMUs e são resolvidas por *Python* via integração trapezoidal (vide Seção 2).
- (iii) Atualização dos equivalentes de Norton associados aos modelos dinâmicos para montagem de $\mathbf{i}(t)$.

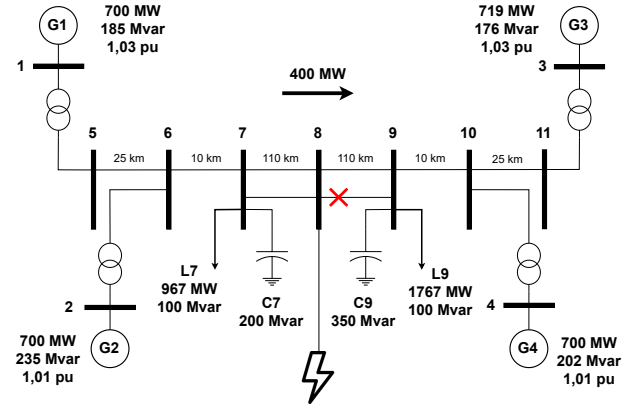


Figura 3. Sistema de transmissão de 11 barras.

- (iv) Atualização dos equivalentes de Norton associados aos modelos funcionais dependentes da tensão para a complementação de $\mathbf{i}(t)$.
- (v) Solução das equações algébricas em (1b) para obtenção de $\mathbf{v}(t)$ através do método nodal.
- (vi) Verificação do erro entre duas soluções sucessivas de \mathbf{v} . Se a diferença respeitar tolerâncias absolutas e relativas de convergência, pode-se prosseguir para (vii). Caso contrário, deve-se regredir para (iv).
- (vii) Verificação do erro entre duas soluções sucessivas de \mathbf{x} no instante t . Se a diferença respeitar tolerâncias relativas e absolutas de convergência, pode-se prosseguir para o próximo passo. Caso contrário, deve-se regredir para o passo (ii).
- (viii) Tratamento de eventos, caso necessário. Após a convergência simultânea das equações (1a) e (1b) no instante t^- , calcula-se o vetor $\mathbf{v}(t^+)$ de acordo com o evento de rede associado ao instante $t = t_e$. Em t^+ , também deve-se reiniciar o processo de integração, atualizando-se os termos históricos \mathbf{x}_h em (3b), para utilização no passo de tempo seguinte.
- (ix) Armazenamento de variáveis de interesse, avanço de tempo em um passo de integração e retorno a (i).

4. RESULTADOS OBTIDOS

A estratégia proposta é aplicada ao sistema de transmissão de 11 barras mostrado na Figura 3 e descrito em detalhes em (Kundur, 1994).

Ao aplicar a estratégia proposta neste trabalho, realiza-se uma simulação de tempo total de 10s, na qual um curto-circuito com impedância Z_{flt} de $j0,001$ pu é aplicado à barra 8 no instante t_{flt} de 0,2s e removido 100 ms depois, junto com a linha defeituosa entre as barras 8 e 9. Os outros parâmetros adotados para a simulação no tempo são: modelagem do tipo impedância constante para as cargas L_7 e L_9 ; passo de integração igual a 1ms; tolerância absoluta igual a 10^{-4} pu e tolerância relativa igual a 10^{-2} %. Para validar os resultados obtidos pelo programa desenvolvido no ambiente *Python*, utiliza-se como referência os resultados do programa ANATEM, desenvolvido pelo CEPTEL. Algumas variáveis referentes à simulação são mostradas nos seguintes gráficos: número de iterações ao longo do tempo (Figura 4), ângulos relativos das máquinas (Figura 5) e módulo de tensão da barra 8 (Figura 6), onde ocorre o curto-circuito de 100 ms.

Algoritmo 1: Método alternado do conjunto modelo-rede.

```

from pyfmi.fmi import FMUModelME2 ;

# Condições iniciais da rede por Newton-Raphson
 $\mathbf{Y}, \mathbf{v} \leftarrow \text{newton\_raphson}(\text{Dados\_rede})$ 

# Inicialização da FMU
fmu[k]  $\leftarrow$  FMUModelME2("Generic_Machine.fmu")
fmu[k].instantiate()
fmu[k].setup_experiment()
fmu[k].enter_initialization_mode()
fmu[k].set("Pesp",  $\mathbf{P}_{esp}$ )
fmu[k].set("Vreal", Real( $v_k$ ))
fmu[k].set("Vimag", Imag( $v_k$ ))
fmu[k].exit_initialization_mode()
integrator[k]  $\leftarrow$  TrapezoidalIntegrator(fmu[k])
integrator[k].init_events()
fmu[k].enter_continuous_time_mode()

# Inicialização da cargas
inicia_cargas( $\mathbf{v}$ )

# Simulação no tempo
t = 0
flag_stop_time  $\leftarrow$  False
enquanto t  $\leq$  tfin faça

    # Inicializações para cada instante de tempo
     $\mathbf{v} \leftarrow \mathbf{v}^2(t - \Delta t) / \mathbf{v}(t - 2\Delta t)$ 
    itermodelo-rede  $\leftarrow$  0
    doNextItermodelo-rede  $\leftarrow$  True

    # Método alternado do conjunto modelo-rede
    enquanto doNextItermodelo-rede faça
         $\mathbf{v}_{ant} \leftarrow \mathbf{v}$ 
         $\mathbf{x}_{ant} \leftarrow \mathbf{x}$ 
         $\mathbf{x}, \mathbf{i}_g \leftarrow$  calcula_modelos(integrator,  $\mathbf{v}$ ,
            flag_stop_time)
         $\mathbf{v} \leftarrow$  calcula_rede( $\mathbf{v}, \mathbf{i}_g$ )
        doNextItermodelo-rede  $\leftarrow$  max( $\|\mathbf{x} - \mathbf{x}_{ant}\| > \varepsilon$  or
            max( $\|\mathbf{v} - \mathbf{v}_{ant}\| > \varepsilon$ ))
        itermodelo-rede  $\leftarrow$  itermodelo-rede + 1

    # Atualização dos termos históricos
    update_history(integrator)

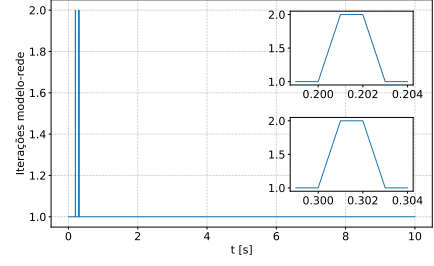
    # Tratamento de eventos
    se t == te faça
         $\mathbf{Y} \leftarrow$  aplica_evento( $\mathbf{Y}$ )
         $\mathbf{v} \leftarrow$  calcula_rede( $\mathbf{v}, \mathbf{i}_g$ )
        flag_stop_time  $\leftarrow$  True
         $\mathbf{x}, \mathbf{i}_g \leftarrow$  calcula_modelos(integrator,  $\mathbf{v}$ ,
            flag_stop_time)
        flag_stop_time  $\leftarrow$  False

        # Atualização dos termos históricos
        update_history(integrator)

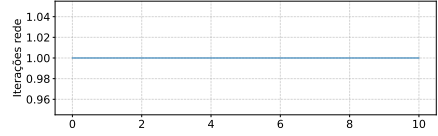
    # Avanço no tempo em um passo de integração
    complete_integration_step(integrator)
    t  $\leftarrow$  t +  $\Delta t$ 

```

Com base no gráfico da Figura 4a, é possível analisar que, depois da aplicação do curto-circuito, a simulação do método alternado do conjunto modelo-rede exige uma média de 1 iteração por instante de tempo, exceto nos instantes 0.201, 0.202, 0.301 e 0.302 s, que exigem 2 iterações cada. Devido à descontinuidade do vetor \mathbf{v} (causada pelas ocorrências de eventos na rede), não é possível aplicar a extrapolação das tensões nodais nos referidos instantes de tempo, o que exige um maior número de iterações.



(a) Número de iterações do conjunto modelo-rede.



(b) Número de iterações da rede.

Figura 4. Número de iterações ao longo do tempo.

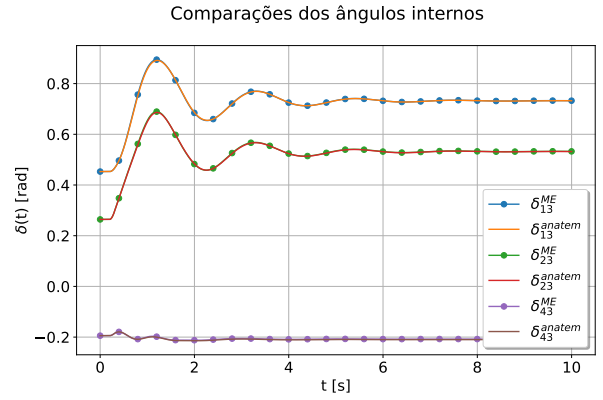


Figura 5. Ângulos relativos das máquinas do sistema de transmissão de 11 barras.

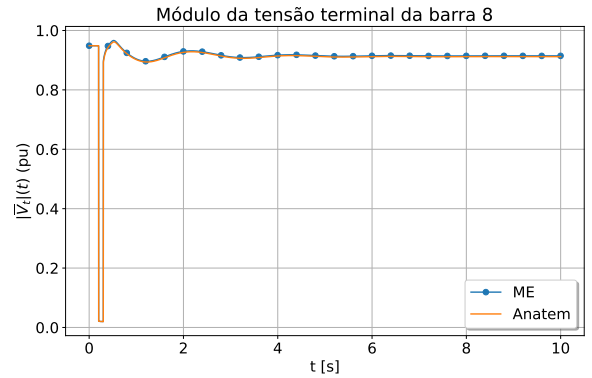


Figura 6. Tensão da barra 8, onde ocorre o curto-circuito de 100 ms.

Com base no gráfico da Figura 4b, o número de iterações para a solução da rede se mantém unitário em todos os instantes, devido à modelagem de carga do tipo impedância constante.

Com base nos gráficos de ângulos relativos e de tensão da barra 8, verifica-se quantitativamente a proximidade das curvas da simulação em relação às curvas de referência. Para isso, adota-se um índice de desempenho chamado

Tabela 1. Valores de NIAE dos ângulos relativos e da tensão da barra 8.

Grandeza	NIAE
δ_{13}	0,9989
δ_{23}	0,9979
δ_{43}	0,9907
$ \bar{V}_{t,8} $	0,9970

Integral do Erro Absoluto Normalizado (NIAE), descrito em (Theodoro, 2016) e encontrado pela equação (5). Se a relação $NIAE \geq 0,95$ for satisfeita, a simulação pode ser considerada adequada.

$$NIAE = 1 - \frac{\int_0^t |\mathbf{x}(\tau) - \mathbf{x}_{ref}(\tau)| d\tau}{\int_0^t |\mathbf{x}_{ref}(\tau)| d\tau} \quad (5)$$

Calculando as integrais da equação (5) através da regra dos trapézios, tem-se na Tabela 1 os valores de NIAE dos ângulos relativos e da tensão da barra 8. Uma vez que todos os valores encontrados são maiores do que 0.95, tem-se que a estratégia de simulação eletromecânica apresentada neste trabalho pode ser considerada adequada.

5. CONCLUSÕES

Este trabalho propõe uma estratégia para a simulação de sistemas de potência visando a estabilidade transitória através da interface FMI para o intercâmbio de modelos.

A divisão da solução de um sistema maior em subsistemas menores e de fácil solução se mostra uma ferramenta com um grande potencial para representar sistemas de grande porte, principalmente aqueles que possuem as mais diversas fontes de energia. Com a implementação de FMUs, é possível realizar simulações e estudos cada vez mais complexos, como por exemplo Recursos Energéticos Distribuídos (RED).

Vale ressaltar que o programa de estabilidade transitória descrito e testado neste trabalho se trata de um protótipo, cujo objetivo é possibilitar o desenvolvimento de equipamentos de maior porte e complexidade. Para dar prosseguimento a esse trabalho, uma possível alternativa é uma reformulação do programa de linguagem *Python*, de modo que, além de acoplar FMUs, também seja capaz de acoplar sistemas de distribuição através do programa OpenDSS, que se mostrou uma ferramenta confiável e eficiente em trabalhos citados na introdução deste artigo.

Por fim, uma outra possibilidade é considerar a FMU para *Scheduled Execution* (SE), ou execução programada, onde o importador inicia a execução dos subsistemas do modelo de acordo com instantes de ativação e definições de prioridade. Para isso, utiliza-se o conceito de *clock* para uma melhor sincronização de eventos e avaliação dos subsistemas através de FMUs. Esse é um dos recursos introduzidos na FMI 3.0, que se trata de uma melhoria do padrão FMI, conforme descrito em (MODELISAR Consortium, 2022).

AGRADECIMENTOS

Os autores gostariam de agradecer à PETROBRAS, à Coordenação de Aperfeiçoamento de Pessoal de Nível Super-

rior (CAPES), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e ao Instituto Nacional de Energia Elétrica (INERGE). Esse trabalho foi desenvolvido em parceria com a PETROBRAS dentro do projeto intitulado “Gerenciamento de Recursos Energéticos Distribuídos para Prestação de Novos Serviços à Rede Elétrica”, PD-00553-0064/2019, no âmbito do programa de P&D do sistema elétrico regulado pela ANEEL.

REFERÊNCIAS

- Andersson, C., Åkesson, J., e Führer, C. (2016). PyFMI: A Python Package for Simulation of Coupled Dynamic Models with the Functional Mock-up Interface.
- de Oliveira Chagas, I.B. e Tomim, M.A. (2022). Co-simulation applied to power systems with high penetration of distributed energy resources. *Electric Power Systems Research*, 212, 108413. doi: <https://doi.org/10.1016/j.epr.2022.108413>. URL <https://www.sciencedirect.com/science/article/pii/S0378779622005508>.
- de Souza, J.P.A., Ferreira, J.K., e Campos, G.E.M. (2020). Uso do openss em aplicações reais para concessionárias de energia. Em *Congresso Brasileiro de Automática-CBA*, volume 2.
- Dommel, H. e Sato, N. (1972). Fast transient stability solutions. *IEEE Transactions on Power Apparatus and Systems*, PAS-91, 1643–1650. doi:10.1109/TPAS.1972.293341.
- Kundur, P. (1994). *Power system stability and control*. McGraw-Hill, Inc.
- Modelica Association (2021). *Functional Mock-up Interface for Model Exchange and Co-Simulation*. URL <https://fmi-standard.org/>.
- MODELISAR Consortium, T.M.A.P.F. (2022). *Functional Mock-up Interface Specification - Version 3.0*.
- Open Source Modelica Consortium (2023). *Open Modelica User’s Guide*.
- Palensky, P., Meer, A.A.V.D., López, C.D., Joseph, A., e Pan, K. (2017). Cosimulation of intelligent power systems. *IEEE Industrial Electronics Magazine*.
- Silva, L.T. (2020). *Modelagem e Simulação de Sistemas de Geração de Energia Eólica através de Co-simulação*. Master’s thesis, Universidade Federal de Juiz de Fora (UFJF). URL <https://repositorio.ufjf.br/jspui/handle/ufjf/11593>.
- Stott, B. (1979). Power system dynamic response calculations. *Proceedings of the IEEE*, 67(2), 219–241. URL <https://ieeexplore.ieee.org/document/1455502>.
- Theodoro, T.S., Barbosa, P.G., Tomim, M.A., de Lima, A.C.S., e de Barros, M.T.C. (2018). Matlab-openss co-simulation environment: An alternative tool to investigate dsg connection. Em *2018 Simposio Brasileiro de Sistemas Eletricos (SBSE)*, 1–7. doi:10.1109/SBSE.2018.8395643.
- Theodoro, T.S. (2016). Simulação híbrida no domínio do tempo de transitórios eletromecânicos e eletromagnéticos: integração de um aerogerador de indução duplamente excitado.
- Tomim, M.A. (2009). *Parallel computation of large power system networks using the multi-area Thévenin equivalents*. Ph.D. thesis, University of British Columbia (UBC), Vancouver, Canada.