

FERRAMENTA DE CO-SIMULAÇÃO DE SISTEMAS DE POTÊNCIA E REDES DE COMUNICAÇÃO PARA O ESTUDO DE SISTEMAS DE DISTRIBUIÇÃO DO FUTURO

FERNANDA C. T. ARIOLI, RODRIGO S. BONADIA, HÉLDER S. FERREIRA, GUSTAVO O. TROIANO

Depto. De Sistemas de Energia Elétrica, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas

Av. Albert Einstein, nº 400, 13083-852, Campinas, SP - BRASIL

E-mails: fernanda@dsee.fee.unicamp.br, rodrigobonadia@gmail.com, hsf.ferreira@gmail.com, gtroiано@dsee.fee.unicamp.br

Abstract— Electric power distribution systems have undergone several changes, among them the connection of smart meters, electric vehicles as loads, and the integration of different distributed generation technologies, such as those based on renewable energy sources. Due to the technical impacts associated with these changes, this scenario will potentially require a better capacity to control and manage these systems. Better management of network resources can be achieved by the integration of data communication technologies with the future distribution systems. To determine the most appropriate communication technology and the use of the integrated communication network to the distribution systems, specialized tools for the analysis of the system as a whole become crucial. In this context, this work describes a co-simulator that employs OpenDSS and OMNeT++ working together to compile and simulate scenarios involving power distribution systems and communication networks. The co-simulator is characterized, as well as its architecture and internal functioning. In addition, a case study is presented.

Keywords— Co-simulation, Electric Power Distribution Systems, Smart Grids

Resumo— Os sistemas de distribuição de energia elétrica vêm sendo submetidos a diversas mudanças, entre elas destacam-se a conexão de medidores inteligentes, veículos elétricos como cargas, e a integração de diferentes tecnologias de geração distribuída, como as baseadas em fontes renováveis de energia. Devido aos impactos técnicos associados a essas mudanças, este cenário potencialmente exigirá uma maior capacidade de controle e gerenciamento destes sistemas. Um melhor gerenciamento dos recursos da rede poderá ser viabilizado pela integração de tecnologias de comunicação de dados aos sistemas de distribuição do futuro. Visando balizar a escolha da tecnologia de comunicação mais adequada e do uso da rede de comunicação integrada aos sistemas de distribuição, ferramentas especializadas para a análise conjunta dos sistemas de distribuição e das redes de comunicação tornam-se fundamentais. Neste contexto, este trabalho descreve um co-simulador de redes inteligentes que emprega o OpenDSS e o OMNeT++ trabalhando em conjunto para compilar e simular cenários envolvendo sistemas de distribuição de energia e redes de comunicação. Os recursos do software são caracterizados, bem como sua arquitetura e funcionamento interno. Ademais, um estudo de caso utilizando a ferramenta de software desenvolvida é apresentado.

Palavras-chave— Co-simulação, Redes Inteligentes, Sistemas de Distribuição de Energia Elétrica

1 Introdução

Devido a uma vasta gama de mudanças que vem sendo introduzidas nos sistemas de distribuição de energia elétrica, como a inserção de veículos elétricos (Lévesque et al., 2012), e geradores fotovoltaicos (Amarasekara et al., 2015), (Ho et al., 2003), torna-se fundamental o uso de técnicas mais avançadas de controle e de gerenciamento destes sistemas.

Os geradores fotovoltaicos, por exemplo, podem gerar problemas relacionados a intermitência, desequilíbrio de tensão e sobretensão em determinados níveis de penetração. É imprescindível que o sistema elétrico seja capaz de suprir a demanda energética e funcionar dentro dos devidos limites de operação. Quanto à presença de veículos elétricos, pode-se afirmar que, no sistema atual, determinados níveis de penetração e recargas descoordenadas podem ocasionar problemas como subtensão, desequilíbrio de tensão e sobrecarga do sistema elétrico (Zheng, Lin e Gao, 2013).

Para solucionar tais complicações, um aproveitamento mais inteligente dos recursos do sistema de distribuição pode ser proporcionado juntamente com

a incorporação de tecnologias de comunicação aos sistemas de distribuição de energia elétrica. Certamente, há numerosas vantagens em implementar uma rede mais inteligente. Segundo Fang et al. (2012), o sistema pode ser aperfeiçoado de inúmeras maneiras: melhoria na qualidade da energia e confiabilidade; aumento da capacidade de transmissão/distribuição de energia, eficiência, segurança e resiliência a faltas da rede; possibilidade da introdução de mais fontes de energia renováveis, veículos elétricos e tecnologias de armazenamento de energia; redução no uso de fontes de energia a combustão em horários de pico de consumo; possibilidade de uso de técnicas de regeneração da rede após incidentes; automação da manutenção e operação do sistema elétrico, além da redução dos custos operacionais; aumento na quantidade de produtos e serviços fornecidos e associados, e mais possibilidades para os clientes em geral.

Contudo, há problemas a serem superados. As redes de comunicação estão sujeitas a diversos problemas, como latência, *jitter*, perdas de pacotes, e ruído. Ademais, redes sem fio também podem ser afetadas por problemas topográficos e de propagação (p. ex., transmissão sem visada direta, atenuação por chuva, oxigênio) (Garau et al., 2015). Todos estes problemas comuns às redes de comunicação podem

afetar funções essenciais dos sistemas de distribuição como o gerenciamento e o controle. Assim, estudos baseados na simulação integrada dos sistemas de energia e da rede de comunicação são necessários para a avaliação de seu desempenho em sua totalidade. Para isso, torna-se necessário desenvolver ferramentas computacionais para suprir tal necessidade.

Neste contexto, este trabalho apresenta uma ferramenta computacional de co-simulação dos sistemas de distribuição de energia elétrica e redes de comunicação que vem sendo desenvolvida pelos autores deste trabalho.

2 Co-Simulação

Nesta seção, uma breve descrição do conceito de co-simulação e sua utilidade prática como uma ferramenta são apresentadas. Além disso, retrata-se o co-simulador desenvolvido pelo grupo de pesquisa e o funcionamento do mesmo.

O desenvolvimento de uma ferramenta para estudar conjuntamente redes de comunicação e sistemas elétricos pode ser realizado de três formas distintas:

- Desenvolvimento de um software novo: conquanto seja uma alternativa válida, sua implementação é muito trabalhosa, além de frequentemente não permitir análises tão completas (Amarasekara et al., 2015).
- Extensões de software: podem ser extensões focadas nas redes de comunicação aplicadas a softwares de sistemas de distribuição ou vice-versa. Todavia, geralmente este tipo de abordagem não permite abranger todas as características de um sistema real, o que torna sua utilização inapropriada para alguns cenários (Li et al., 2014), (Al-Hammouri, 2012).
- Co-simulação: consiste na integração conjunta de dois ou mais softwares através de uma plataforma independente (Amarasekara et al., 2015). É a alternativa mais factível, tanto do ponto de vista da reprodução dos fenômenos de uma rede inteligente, quanto do custo e dificuldade de implementação. Contudo, trata-se de uma ferramenta interdisciplinar e há pouca informação disponível na literatura.

Os co-simuladores dedicados à análise de sistemas de distribuição do futuro podem ser implementados usando variadas combinações de softwares, tanto de sistemas de potência, quanto de redes de comunicação. Cada uma dessas combinações apresenta uma determinada característica de simulação, de acordo com o tipo de estudo a ser realizado.

Uma das principais dificuldades encontradas durante a implementação de um co-simulador refere-se ao problema de sincronização dos softwares envolvidos. O desenvolvimento das simulações de softwares de sistemas de potência é baseado no tempo, enquan-

to os simuladores das redes de comunicação são baseados em eventos. Ou seja, somente quando ocorre um evento, o tempo de simulação de um software de comunicação pode avançar (Dede et al., 2015). Há três principais formas com que este problema de sincronização pode ser resolvido:

- Sincronização por intervalos de tempo (*time-step synchronization*): este método consiste em fazer com que ambos os simuladores sejam executados independentemente, mas, em tempos predeterminados, pausam e trocam informações entre si. Infelizmente, esta estratégia tende a gerar imprecisões na simulação e a acumulação progressiva de erros temporais causados por atrasos entre a ocorrência de algum problema e a sua respectiva ação de controle.
- Sincronização por eventos: preliminarmente ao início da simulação, uma lista de eventos compartilhada entre ambos os programas é gerada. Imediatamente após, os simuladores executam de forma independente. A cada evento, no entanto, os programas trocam dados e são sincronizados. Contudo, esta abordagem pode comprometer a velocidade da simulação.
- Sincronização Mestre/Escravo: nesta arquitetura, um dos simuladores (o mestre) controla a simulação e quando haverá comunicação entre os programas. Pode ser mais facilmente implementado, porém gera imprecisões nos intervalos de tempo. A solução adotada neste trabalho não é exatamente essa, mas é uma versão modificada do Mestre-Escravo, cujas modificações servem justamente para atenuar o problema supramencionado das imprecisões de tempo.

2.1 Co-Simulador Implementado – Visão Geral

O projeto do co-simulador, que vem sendo desenvolvido emprega ambos os simuladores OpenDSS e OMNeT++ trabalhando em conjunto. O OpenDSS atua como simulador de sistemas de potência, enquanto o OMNeT++ atua como simulador de redes de comunicação. O co-simulador, desenvolvido em C#, integra ambos de forma sincronizada para realizar simulações de redes inteligentes. Pretende-se disponibilizar a ferramenta construída de forma gratuita para a comunidade acadêmica em breve.

Uma versão mais simples desta ferramenta é descrita em (Troiano, Ferreira, Trindade e Ochoa, 2016), sendo que o presente artigo descreve uma versão mais recente e mais aperfeiçoada da ferramenta.

O OpenDSS (do inglês, *Open Distribution System Simulator*) é um simulador nativo em Windows de código livre, capaz de realizar simulações em redes elétricas. É baseado em tempo discreto, podendo realizar simulações instantâneas (apenas um cálculo de fluxo de carga), ao longo de um dia todo (modo *daily*) e ao longo de um ano (modo *yearly*). Há uma infinidade de dispositivos disponibilizados, de forma oficial ou não, para o OpenDSS. Dentre eles, inclu-

em-se: linhas de transmissão, transformadores, reguladores, chaves, capacitores *shunt*, cargas, usinas de geração de energia, sistemas fotovoltaicos, geradores eólicos e baterias.

Pode-se utilizar a linguagem própria do OpenDSS (DSS) para descrever o sistema elétrico. Além disso, pode-se usar a interface COM para invocar o OpenDSS dentro de um programa em outra linguagem de programação, como por exemplo Visual Basic, MATLAB, Python, C#, R e C++. Destaca-se o C#, pois é a linguagem que é utilizada neste co-simulador para acionar o cálculo da simulação e o controle da rede no OpenDSS.

Por sua vez, o OMNeT++ é um simulador de redes de comunicação de código aberto para uso não-comercial. É amplamente documentado, modular e é capaz de simular redes de comunicação com diversas topologias e protocolos de comunicação. É nativo em sistemas operacionais tipo Unix, porém pode ser executado em Windows utilizando MinGW. Modela as quatro primeiras camadas de uma rede no modelo OSI (camadas de transporte, rede, enlace e física). Na camada física, o OMNeT++ pode simular tanto redes com ou sem fio, além de fenômenos físicos pertinentes, como por exemplo dispersão e reflexão.

Para estipular o comportamento dos componentes que fazem parte do sistema (clientes, repetidores, roteadores e servidores), programa-se a lógica em C++ e em NED (uma linguagem puramente descritiva que indica os elementos existentes na rede e determinados parâmetros físicos do sistema) (OpenSim Ltd., *What is OMNeT++?*, s.d.).

Empregando-se as devidas extensões de software e/ou *frameworks*, pode-se utilizar protocolos de comunicações populares muito mais facilmente em projetos do OMNeT++. Neste projeto em específico, emprega-se o *framework* INET, o qual possui implementações de diversos numerosos protocolos de comunicação de dados (principalmente nas camadas de transporte e rede), em diferentes camadas. *Internet Protocol v4* (IPv4), *Internet Protocol v6* (IPv6), *Transmission Control Protocol* (TCP), *Scalable Transmission Control Protocol* (STCP), *User Datagram Protocol* (UDP), 802.11 (Wi-Fi), Ethernet e *Point-to-Point Protocol* (PPP) são alguns exemplos de protocolos disponíveis neste *framework* (OpenSim Ltd., *INET Framework for OMNeT++/OMNEST*, s.d.). Este *framework* é empregado neste co-simulador, portanto é possível construir cenários de co-simulação utilizando qualquer protocolo presente no INET.

No co-simulador, comunicando-se via *socket*, o OpenDSS é executado em sistema operacional Windows, e o OMNeT++ é executado em Linux. Esta necessidade de rodá-los em sistemas operacionais diferentes se deve ao fato de que o OpenDSS é nativo em Windows, enquanto o OMNeT++ é nativo em sistemas tipo Unix (Linux, OSX, etc.). Para isso, pode-se usar dois computadores distintos ou um único computador com máquina virtual. O software C# que

realiza a co-simulação é executado no mesmo sistema que executa o OpenDSS.

Na rede do OpenDSS, constrói-se o sistema elétrico de distribuição a ser estudada, com os devidos parâmetros elétricos. Na rede do OMNeT++, introduzem-se os medidores inteligentes e configura-se a topologia da rede de comunicações desejada, integrando as tecnologias de comunicação a serem estudadas.

Para a redação deste artigo, o co-simulador mostra a simulação de sistemas com geradores fotovoltaicos e executar ações de controle nos mesmos para mitigar efeitos de sobretensão (o usuário do programa é quem estipula os limites de sobretensão e subtensão) causados pela introdução de altas penetrações de geradores fotovoltaicos nos sistemas de distribuição de energia elétrica. A partir de medições nas barras da rede de distribuição, um algoritmo de controle reduz o fator de potência dos painéis (até um limite mínimo estipulado pelo usuário) para tentar reduzir a sobretensão.

É possível ajustar o tempo inicial e a duração da simulação (sendo este tempo o tempo de simulação, e não o tempo real). Isto é relevante para o estudo como um todo, pois consumidores conectados ao sistema elétrico possuem curvas de variação de consumo de potência ao longo do dia. O mesmo também se aplica à incidência de luz solar, o que afeta a geração fotovoltaica.

Além disso, o programa possui alarmes. Alarmes são gerados quando a tensão de um consumidor excede algum limite de sobretensão ou subtensão, desencadeando a ação de controle. Este recurso será melhor esclarecido na seção 2.2.

O programa também salva os resultados da simulação (todas as medições obtidas, alarmes gerados, alarmes recebidos, histórico de ações de controle, etc.) em um banco de dados SQLite. Este banco de dados pode ser posteriormente examinado e estudado para verificar a atuação integrada entre sistema elétrico, rede de comunicações e lógica de controle e, por conseguinte, obter conclusões sobre o sistema como um todo. É relevante salientar que este projeto está em constante aprimoramento.

2.2 Funcionamento do Co-Simulador

O co-simulador é dividido em três componentes principais, conforme mostrado na Figura 1 e descritos a seguir.

- Simulação da rede de telecomunicações: é realizada pelo OMNeT++. Há algumas classes de suporte implementadas dentro dos modelos da rede de comunicações, cuja responsabilidade é a comunicação entre os simuladores, bem como a recepção dos alarmes enviados pelo OpenDSS.
- Simulação do sistema elétrico: é realizada pelo OpenDSS e controlada externamente através da interface COM por meio de um controlador. O Controlador é responsável por avançar a simula-

ção no tempo, manter e atualizar o cache (área de memória que, de modo grosseiro, mantém o estado do último passo de simulação), e gerar os alarmes.

- **Engine de Co-Simulação:** é implementada em C# e recebe todas as medições regulares e alarmes. Essas medições e esses alarmes são armazenados em um banco de dados (SQLite) com finalidade de efetuar posteriores estudos. Quando os alarmes são recebidos, respectivas ações de controle podem ou não serem geradas. Também integra o gerenciador, que controla toda a comunicação entre as outras partes e é responsável pelo encapsulamento e desencapsulamento das mensagens transmitidas entre os diferentes componentes do co-simulador.

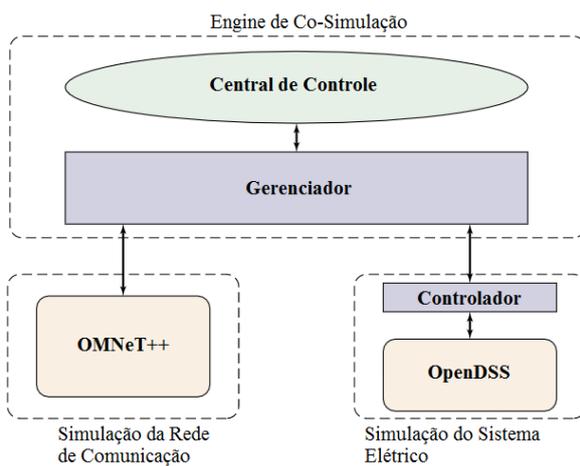


Figura 1. Arquitetura do co-simulador desenvolvido.

Quanto à sincronização entre os softwares, a solução utilizada no co-simulador implementado pelo grupo foi uma estrutura Mestre com Escravo Ativo (uma forma modificada do modelo de sincronização Mestre-Escravo descrito previamente na Seção 2.1), em que o mestre é o OMNeT++ e o escravo ativo é o OpenDSS. Esta implementação mitiga os problemas de temporização pertinentes ao Mestre-Escravo clássico, sem dificultar muito a implementação em software da estrutura de sincronização em questão.

Em uma estrutura clássica de mestre-escravo, apenas o mestre pode se comunicar livremente com o escravo, mas o escravo só pode se comunicar com o mestre caso o mesmo tenha previamente requisitado uma resposta. Um dos diferenciais deste co-simulador é o que dá ao escravo o nome de “escravo ativo”: a capacidade de, em algumas situações em específico, iniciar uma comunicação com o mestre sem a sua requisição, o que não é lugar-comum na estrutura usual de sincronia por mestre-escravo.

Neste trabalho, tais situações específicas constituem-se em alarmes. Podem existir diversos tipos de alarmes, como por exemplo violações de tensão (que constitui o tipo de alarme implementado no co-simulador apresentado neste artigo).

Em tais ocorrências de alarmes, o OpenDSS pode iniciar uma comunicação com o OMNeT++ sem ser requisitado, para que o sistema tome suas ações de controle adequadas (Troiano, Ferreira, Trindade e Ochoa, 2016). Na Figura 2, representa-se uma ilustração da comunicação entre os softwares.

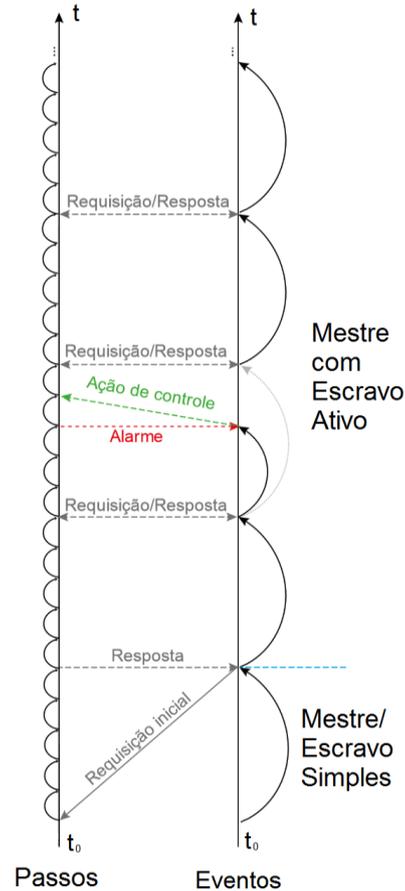


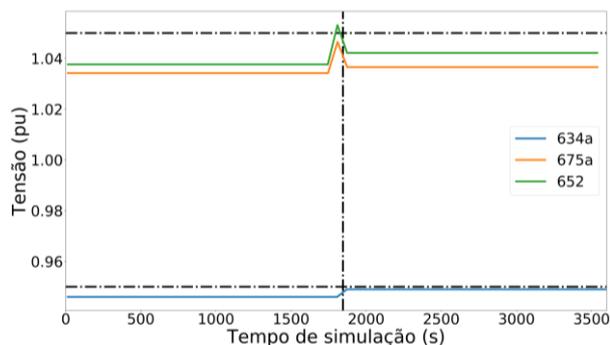
Figura 2. Ilustração da comunicação entre softwares utilizando Mestre com Escravo Ativo.

3 Estudo de Caso

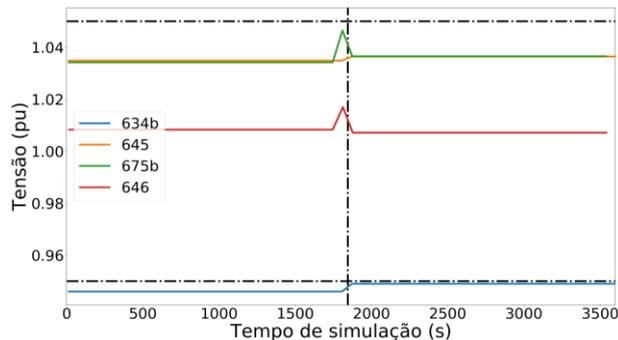
Nesta seção, apresenta-se a descrição do estudo de caso realizado com o co-simulador, bem como os resultados advindos desta simulação. Ressalta-se que este estudo de caso não necessariamente representa um sistema elétrico real ou verossímil. Alguns dos parâmetros da simulação, como por exemplo a penetração de geração fotovoltaica e a redução no fator de potência dos inversores, foram exagerados para melhor visualização dos efeitos de controle da co-simulação.

3.1 Descrição Básica – Sistema Elétrico

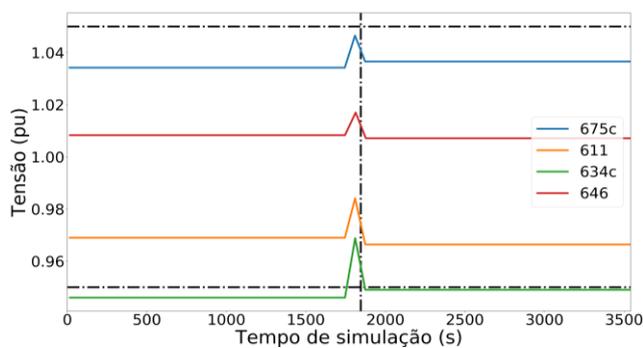
O circuito deste estudo de caso é uma versão modificada do IEEE 13 barras (IEEE PES, *Test Feeders*, s.d.), um sistema de teste simples com 13 barras, constituído por uma pequena rede de distribuição



(a) Tensões na fase 1.



(b) Tensões na fase 2.



(c) Tensões na fase 3.

Figura 5. Tensões registradas por cada medidor instalado em consumidores com gerador fotovoltaico. Linhas horizontais tracejadas representam os limites de 1,05 e 0,95 pu; linhas verticais representam ações de controle (Ação de controle **ativada**).

Observe que, em todos os gráficos, ocorre um pico de tensão um pouco antes dos 2.000 s. Isto ocorre pois há uma violação de tensão (sobretensão) na fase 1 do medidor inteligente da barra 652 (Figura 5(a), no tempo 1.845 s).

Logo após a violação, é gerado um alarme no tempo de 1.845,23 s. Pouco tempo após, quando a mensagem de alarme se propaga pela rede do OMNeT++ e é processada, ocorre uma ação de controle (linha vertical). Um pouco depois da ação de controle, a tensão em todos os medidores e em todas as fases é significativamente reduzida, e a violação de sobretensão na barra é removida com sucesso.

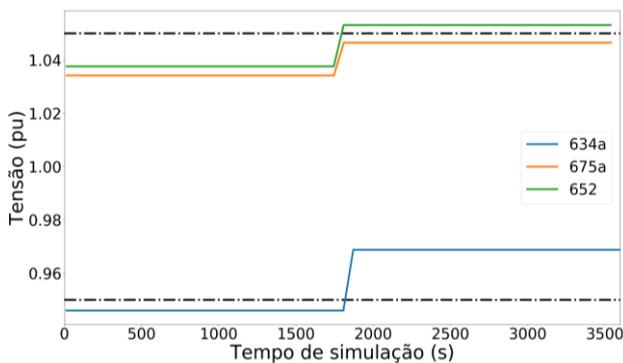
Para realizar uma comparação, foi executada também uma simulação com o sistema elétrico funcionando normalmente, sem realizar nenhuma ação de

controle sobre os inversores dos geradores fotovoltaicos. Os resultados estão apresentados na Figura 6.

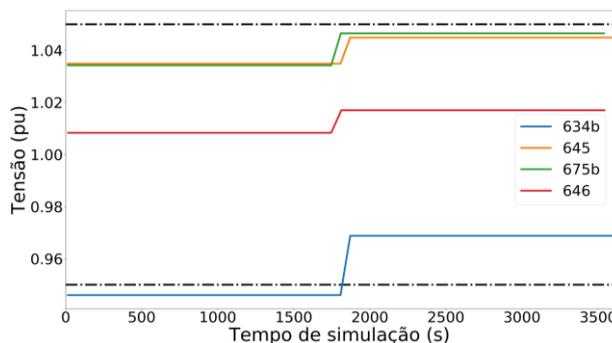
Observe que a tensão aumenta no tempo 1.845 s, e há uma violação de sobretensão na barra 652, porém nenhuma ação é tomada e a violação de tensão persiste até o fim da simulação.

Todos os dados obtidos para a construção dos gráficos foram retirados do banco de dados SQLite, o qual é gerado pelo co-simulador.

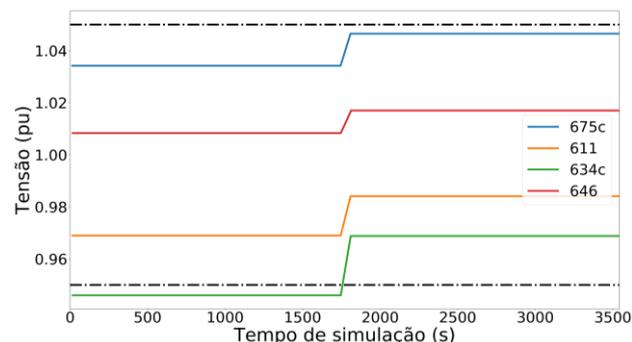
Cabe salientar também que vários dos parâmetros utilizados nesta simulação (como por exemplo o tempo de simulação, limites de sobretensão e subtenção, intervalo de tempo entre as medições, etc.) são ajustáveis e podem ser modificados para outros estudos.



(a) Tensões na fase 1.



(b) Tensões na fase 2.



(c) Tensões na fase 3.

Figura 6. Tensões registradas por cada medidor instalado em consumidores com gerador fotovoltaico. Linhas horizontais tracejadas representam os limites de 1,05 e 0,95 pu; linhas verticais representam ações de controle (Ação de controle **desativada**).

4 Conclusão

Neste trabalho, foi retratado o co-simulador, ferramenta construída pelo grupo de pesquisa para auxiliar o estudo de redes de distribuição do futuro. Com essa ferramenta, é possível que técnicas de controle e gerenciamento do sistema elétrico sejam investigadas com mais facilidade, levando em consideração fatores imprescindíveis a uma análise mais profunda, como por exemplo a qualidade e velocidade da transmissão de dados via uma determinada rede de comunicações e como isso afeta do desempenho o gerenciamento do sistema elétrico e a qualidade do serviço de fornecimento de energia.

O estudo de caso realizado demonstra o funcionamento da ferramenta em um cenário simples, mas que evidencia as suas capacidades técnicas. Esta é apenas uma demonstração do que pode ser feito com essa ferramenta.

Agradecimentos

Este trabalho é fomentado pela FAPESP, processo 2017/03271-6.

Referências Bibliográficas

- Al-Hammouri, A. T. (2012). *A comprehensive co-simulation platform for cyber-physical systems*. *Computer Communications*, vol. 36, no. 1, pp. 8-19.
- Amarasekara, B., Ranaweera, C., Nirmalathas, A. and Evans, R. (Nov. 2015). *Co-simulation Platform for Smart Grid Applications*. In: *IEEE Innovative Smart Grid Technologies - Asia (ISGT)*, Bangkok, Thailand.
- Dede, J., Kuladinithi, K., Foster, K., Nannen, K. and Lehnhoff, S. (2015). *OMNeT++ and mosaik: Enabling Simulation of Smart Grid Communications Position Paper*. In: *2nd OMNeT++ Community Summit, IBM Research, Zurich, Switzerland*.
- Fang, X., Misra, S., Xue, G. and Yang, D. (2012). *Smart Grid — The New and Improved Power Grid: A Survey*. In: *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944-980.
- Garau, M. et al. (2015). *ICT reliability modelling in co-simulation of smart distribution networks*. In: *IEEE 1st International Forum on Research and Technologies for Society and Industry (RTSI) Leveraging a better tomorrow*, Turin, Italy.
- Ho, Q. D., Gao, Y., Rajalingham, G., Ngoc, T. L. (2003). *Wireless Communications Networks for the Smart Grid*. In: CRC Press.
- IEEE PES, *IEEE Distribution Test Feeders*. [Online]. Available at: <http://sites.ieee.org/pes-testfeeders/> [Accessed at Mar. 2018].
- Lévesque, M., Maier, M., Xu., D. Q. e Joós, G. (2012). *Communications and Power Distribution Network Co-Simulation for Multidisciplinary Smart Grid Experimentations*. In: *Proceedings of the 45th Annual Simulation Symposium (ANSS'12)*, Orlando, FL, USA.
- Li, W., Ferdowsi, M., Stevic, M., Monti, A. e Ponci, F. (2014). *Cosimulation for Smart Grid Communications*. In: *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2374-2384.
- OpenSim Ltd., *INET Framework for OMNeT++/OMNEST* [Online]. Available at <https://omnetpp.org/doc/inet/api-current/neddoc/index.html> [Accessed at Aug. 2017].
- OpenSim Ltd., *What is OMNeT++?* [Online]. Available at <https://omnetpp.org/intro> [Accessed at Aug. 2017].
- Troiano, G. O., Ferreira, H. S., Trindade, F. C. L., Ochoa, L. F. (2016). *Co-simulator of power and communication networks using OpenDSS and OMNeT++*. In: *2016 IEEE Innovative Smart Grid Technologies Asia (ISGTAsia)*, Melbourne, Australia.
- Zheng, J., Lin, L., Gao, D. W. (2013). *Smart Meters in Smart Grid: An Overview*. In: *IEEE Green Technologies Conference*, Denver, CO, EUA.