

Power System Real-Time Simulation using Modelica and the FMI

Marcelo de Castro¹ Giuseppe Laera¹ Fernando Fachini¹ Sergio A. Dorado-Rojas¹ L. Vanfretti¹
Shehab Ahmed² Chetan Mishra³ Kevin D. Jones³ R. Matthew Gardner³

¹ECSE Department, Rensselaer Polytechnic Institute, USA, {decasm3, laerag, fachif, dorads, vanfr1}@rpi.edu

²King Abdullah University of Science and Technology, Saudi Arabia, shehab.ahmed@kaust.edu.sa

³Dominion Energy, USA, {chetan.mishra, kevin.d.jones, matthew.gardner}@dominionenergy.com

Abstract

Real-time digital simulation of power systems is incredibly important for the testing of appropriate control and protection strategies in the power system industry. However, the case in which one single model can be used in offline simulations and then for testing in real-time studies is rare, if existing at all, due to the lack of adequate standard development in the power industry or the adoption of successful standards elsewhere. A direct consequence of this lack of portability is the large amount of time and resources spent in re-implementation and validation of models for real-time simulation of power grids. The present study proposes the usage of Modelica and the FMI standard in order to address this issue. To test the proposed approach, power system models are built offline using the OpenIPSL library and are exported as FMUs. Real-time simulations of two typical power system models are performed using dSPACE SCALEXIO™, proving that the proposed framework using Modelica and the FMI can greatly contribute to the enhancement of today's current practice in the power industry by providing portability and tractability between offline and real-time power system models.

Keywords: Power Systems, Real-Time Digital Simulation, Modelica, Functional Mockup Interface, FMI

1 Introduction

Modern power systems depend heavily on the ability of engineers to anticipate outcomes that can harm the grid's safe operation (Kundur 2007; Chow and Sanchez-Gasca 2020). Hence, it is obvious that Modeling and Simulation (M&S) plays a crucial role in power system studies. Consequently, engineers have developed several strategies through the decades in order to be able to better represent the electric grid in all its intrinsic complexity. Real-time simulation is, of course, one of those strategies and it is used as a last step before deploying one piece of equipment in real-world conditions. In order to understand the context that motivates this present study, it is first necessary to understand the background on simulation of power systems.

1.1 Background

Before digital simulation became possible, hardware-based test beds in laboratories emulating equivalent models of power apparatus were used since the late 1920s for representing the dynamic behavior of the bulk power system in a smaller scale (Evans and Bergvall 1924). It was not until the 1950s, though, that larger portions of the grid started being represented by analog circuits, composed by amp-ops, capacitors, resistors and inductors (Baldini and Fugill 1952). These transient network analyzers were so important for the simulation of power systems that even today their causality-orientation principle used for modeling is the basis for most conventional model development in the power industry. Although the analog simulators required a myriad of different solutions to properly mimic a real system, they allowed to actively test the prototypes of actual controllers using real measurements (Isaacs 2017).

With the development of modern computers, digital simulation was born and it rapidly became a very attractive approach for the analysis of power systems (Brown and Tinney 1957; Stott 1979). After many years of important advances in simulation technology, different parties could program Dommel's (1969) method for electromagnetic transient (EMT) solutions into a DSP (Throckmorton and Wozniak 1994), starting what is known today as real-time digital simulation (Isaacs 2017). If compared to their analog counterparts, these digital network analyzers were cheaper to assemble, had greater flexibility in terms of operation, allowed for tests to be prepared and conducted more rapidly, and enabled better reproducibility of experiments (Watson and Arrillaga 2003). Due to these notable features, real-time simulators became very common in the assessment of power systems' dynamic operation, control and protection testing.

Simulation under real-time, however, has several limitations with respect to the solution method applied to the system of equations. Simulation of a model in real-time is done with fixed time step solvers due to easier implementation when compared to variable time step, especially considering the strict requirements related to the tasks executed in each time-step. In the interval of one time step, the simulator needs to read inputs, if available, perform the numeric calculations and make outputs available for measurement, as it is depicted in

Figure 1. When this is not achieved under one time step, there is an overrun. As one may note, the time step shown in Figure 1 might not be of importance for human interaction. However, the purpose of real-time simulation is to be able to prototype real-world hardware-based controllers and protection devices and to test their dynamic performance within a power system model (Watson and Arrillaga 2003).

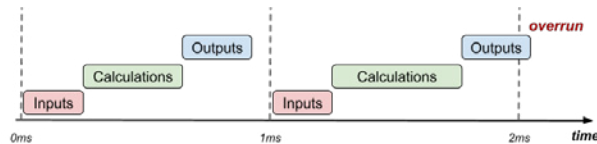


Figure 1. Tasks performed by real-time simulator within the range of a time step.

1.2 Motivation and Objectives

After years of development, real-time digital simulators have become a broadly adopted approach in the testing and prototyping of solutions for power systems over the last decade (Faruque et al. 2015). This is mainly due to the increase in their capabilities for power network representation and the possibility to conduct real-time hardware-in-the-loop simulations. With exception of solutions that adopt MATLAB/SIMULINK™ tools, such as eMEGAsim™ from Opal-RT (OPAL-RT Technology Inc. 2022), the traditional approach requires different proprietary M&S software tools to be used for offline M&S, control design, and real-time simulation. Due to that, engineers have to spend a great amount of time, effort, and resources being allocated in the re-implementation and verification of models, having great impacts on project's costs and the reliability of simulation results. Therefore, this clear lack of portability and tractability can be a tremendous bottleneck that should be urgently solved.

This paper, then, addresses this problem by proposing the usage of standardized dynamical models using the Modelica language and exploiting the Functional Mockup Interface (FMI) standard (Modelica Association Project 2021) to deploy these models into different platforms. Offline power system models can be developed using the Modelica language (Modelica Association 2022) and the Open-Instance Power System Library (OpenIPSL) (Baudette et al. 2018), an open-source library of power system models for stability studies using the “phasor” representation. The OpenIPSL has a set of models that are present in traditional proprietary software tools used in the study of power systems, such as PSS/E™. Moreover, as most Modelica tools implement the FMI standard, the language becomes a strong candidate along with Modelica-compliant software being the M&S tools in which real-time models are configured. Furthermore, the dSPACE SCALEXIO™ (dSPACE GmbH 2022) real-time simulator usually employed in aviation and automobile

industries, has adopted the FMI standard, enabling it to natively execute models exported as Functional Mockup Units (FMUs).

1.3 Contributions

The contributions of this paper are twofold.

- The introduction and the assessment of a proposed framework using Modelica language and the FMI standard as means to address the portability and tractability challenges that are common in power system studies involving real-time simulation;
- The assessment of power system models assembled with the OpenIPSL library, which were made originally developed for offline simulation, in a real-time execution environment, with minimal modifications.

1.4 Paper Organization

The remainder of this paper is organized as follows: Section 2 presents the methodology for offline to real-time simulation proposed in this paper; Section 3 presents the example power systems used in this study; Section 4 describes the laboratory setup for the real-time simulations; Section 5 shows the measured results for the different simulations; Section 6 provides the concluding remarks of this paper.

2 Model Configuration Framework

The framework adopted in this paper to perform power system modeling, offline and real-time simulations, can be summarized into three steps. They are depicted in Figure 2 as a workflow, starting with the offline Modelica model and resulting in the real-time simulation on a dSPACE SCALEXIO™ real-time simulator.

2.1 Modelica Model Configuration

The first step, as shown in Figure 2, concerns the offline power system network model. It is assembled in Modelica and, in this study, the Dymola™ 2019 software is used to make modifications on the model. The network model is built using the OpenIPSL (Baudette et al. 2018), a library that has been under development during the last decade (Bogodorova et al. 2013) and that has shown potential for the study of power systems by the means of offline simulations for dynamic performance assessment (Baudette et al. 2018; Winkler 2017). For real-time simulation, the original Modelica model needs to be slightly modified to have outputs added, to specify the quantities to be measured during simulation. In addition, simulation parameters, such as solver tolerance and total simulated time, also need to be specified. These tasks can be easily done by using Modelica Standard Library's `Blocks.Interfaces.RealOutput` and by the addition of experiment annotations, respectively.

After performing these minor modifications in the original models, the FMU export procedure can be



Figure 2. Three-step workflow adopted for simulation of FMUs in a real-time processing unit.

performed. The export should be configured in such way that a FMI v.2.0 Co-Simulation FMU with CVODE solver is created, with 64-bit binaries, including the model's source code. Although CVODE solvers are variable-order variable-step solvers, a fixed time-step variable-order CVODE solver is exported within the FMU. It is also necessary to highlight that one should check the compatibility between the versions of Dymola™ and dSPACE™'s software ConfigurationDesk® (the latter is the target in which the FMU will be loaded).

2.2 Application Configuration

After the export procedure, the FMU is loaded into dSPACE™'s software for real-time application configuration, ConfigurationDesk®. There, hardware resources such as the SCALEXIO™ and LabBox™ can be assigned to the project and configured. The former is the real-time processing unit and is responsible for the calculations that are executed during the real-time simulation. The latter is the unit responsible making outputs and inputs available. More information on these two pieces of hardware is presented in Section 4.

It is important to note that, once the FMU is loaded into ConfigurationDesk®, the outputs added on the first step are recognized and can be configured to be interfaced (via functions) to the real hardware outputs in the LabBox™ hardware, via the DS6101® board. The signals connected to these ports are converted into non-negative voltages of same magnitude and, therefore, special caution needs to be considered in output preparation. In fact, the DS6101® board supports outputs within the range of 0 to 10 V, justifying why some outputs of interest might need to be pre-configured to allow for measurements to be performed during the experiment.

2.3 Real-Time Simulation

After all the simulation configurations are set up and the outputs are properly configured, a “real-time application” is built. This procedure is executed automatically by ConfigurationDesk®. The real-time application can then be loaded into SCALEXIO™ and the simulation can be controlled by using another software tool made

available by dSPACE: ControlDesk®. In this software tool, the FMU can be set up for simulation and the real-time processing unit will execute the necessary calculations. Although the quantities are physically measured in this paper using both USB-based and conventional oscilloscopes, the results from the real-time simulations can also be verified in ControlDesk®. More information about it can be found in Section 4.

3 Test Cases

In order to assess the framework, two typical test power system models are developed in Modelica using the OpenIPSL library and are configured for export using FMI, as described in Section 2. A brief description of these two system models is presented next.

3.1 Single-Machine Infinite-Bus

The Single-Machine Infinite-Bus (SMIB) is a very basic representation of a power network, with five buses, one machines and one infinite bus. The diagram representation of this power grid is presented on Figure 3. The infinite bus is connected at bus GEN2 while the generator connected at bus GEN1 is composed by a round rotor synchronous machine (GENROE), a fast static excitation system model (ESST1A) and a stabilizer (PSS2B). During the simulation, the system undergoes a three-phase-to-ground fault on bus FAULT. In addition, note that these models are also present in PSS/E™, a traditional software tool used in offline power system transient stability studies.

As mentioned in Section 2, the model needs minor additions, such as the placement of outputs to gather measurements during the real-time experiment. Four outputs are placed in the model and their relationship with model variables are determined via equations, as shown in Listing 1. Note that, because the output board does not allow for non-negative voltages to be measured, the frequency deviation, in Hz, is offset by 2.

In addition, although the SMIB system is quite simple, it can be helpful considerably helpful in basic experiments (Kundur 2007) and can be used in controller design (De Marco, Martins, and Rullo 2021; De Marco,

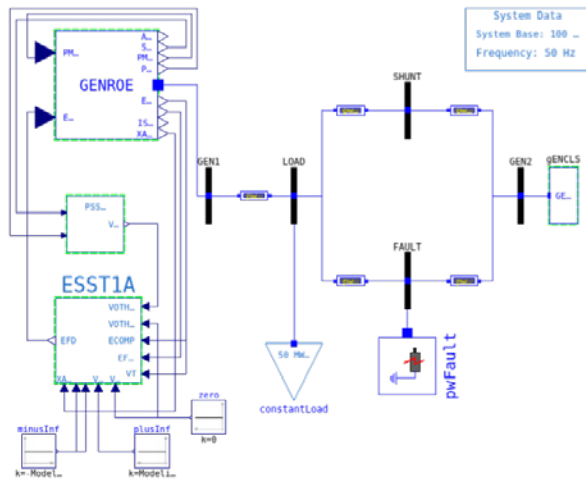


Figure 3. Implementation of the SMIB system in Modelica using the OpenIPSL.

Rullo, and Martins 2021).

Listing 1. Equations added to SMIB model for representing outputs of interest.

```
equation
dfreq = 2 + gENROE.SPEED*SysData.fn
"Frequency deviation with offset";
rotor_angle = gENROE.ANGLE
"Rotor angle measurement";
voltage = gENROE.ETERM
"Generator's terminal voltage magnitude";
power = gENROE.PELEC
"Generator's active power output";
```

Listing 2. Equations added to IEEE 9-bus model for representing outputs of interest.

```
equation
statcom_out = sSTATCOM.Q + 0.5
"STATCOM's reactive output with offset";
gen_Q = gen1.gen.Q + 0.5
"Gen 1 reactive output with offset";
freq_gen1 = gen1.gen.w*SysData.fn-59
"Gen 1 frequency deviation with offset";
freq_gen3 = gen3.gen.w*SysData.fn-59
"Gen 3 frequency deviation with offset";
```

3.2 IEEE 9-Bus 3-Machine

The IEEE 9-bus 3-machine system is the second network used in this study. The one-line diagram of this test system, built using OpenIPSL, is presented in Figure 4. The machines are connected at buses B1, B2 and B3 and all of them are composed of a machine and an excitation system models verified against those of PSAT (Milano 2005). Moreover, a Static Compensator (STATCOM) is connected at bus B8 and a Fault element, representing a three-phase-to-ground event is connected at bus B9.

Once again it is necessary to slightly modify the original network in order to introduce the outputs to be

measured during the real-time experiment. The outputs are associated with the model's variables via the equations in Listing 2. Because the reactive power output of the STATCOM model and of machine 1 have negative values, at some time instants in the simulation, their values are offset by 0.5 per unit to conduct the experiment. The frequency deviation, in Hz, for generators 1 and 3 is also offset and centered at 1.

4 Simulator Setup

After the power system models are exported as FMUs, they are loaded into ConfigurationDesk® and are configured as described in Section 2. The real-time applications are then built and loaded into the assigned SCALEXIO™ real-time processing unit. Hardware specifications of the processing unit together with simulation parameters are summarized on Table 1. The processing unit is based on CPU architecture and from the 4 cores available, only one is used in all tests performed in this paper, indicating that larger power system models can be simulated if they are broken into different FMUs and assigned to different cores. Furthermore, the simulation time step used in all simulations performed in this paper is 1 ms, a common value for power system studies related to electromechanical transient stability. In addition, the maximum number of overruns is set to 150. After that, the real-time simulation stops.

Table 1. Hardware and simulation specifications.

Parameter	Description
Processor	Intel® Xeon® E3-1275 v3
Number of cores (used)	4 (1)
Clock Frequency	2.8 GHz
RAM size	8 Gb
Flash size	512 Mb
Allocated task stack size	1024 Kb
Task time step	1 ms
Overrun Count Max.	150

The outputs are configured in such way that measurements are made directly in the pins of the input/output (I/O) boards. The FMU's outputs are assigned to analog outputs and are measured using USB and conventional oscilloscopes, as displayed in Figure 5. Note that the behavior of determined variables is also available for display on ControlDesk®. The results displayed in Section 5 are obtained using the USB oscilloscope due to the convenience it offers to import results.

5 Real-Time Simulation Results

In this section, the results for the simulation of both systems is presented together with some discussion on the results and on the performance of the real-time processing

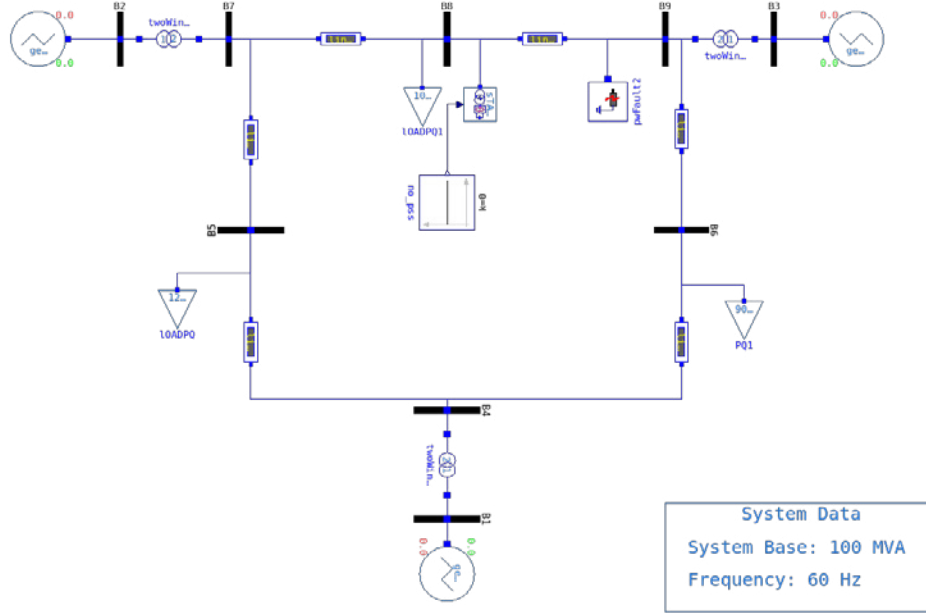


Figure 4. Implementation of the IEEE 9-bus 3-machine system in Modelica using the OpenIPSL.

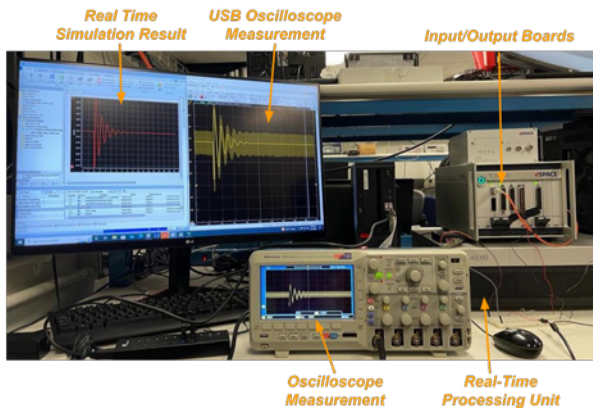


Figure 5. Experiment set-up and comparison among real-time simulation result and observed measurements coming from two different devices.

unit.

5.1 Single-Machine Infinite-Bus

The results for the real-time simulation of the SMIB system are shown in Figure 6. The generator's terminal voltage is depicted in channel 1, in red, of Figure 6a while the generator's deviation from nominal frequency of 50 Hz is displayed in channel 2, in blue. Note that the latter is centered ≈ 2 V, as designated with Listing 1. In Figure 6b, channel 1, in red, represents the generator's electrical power output, in per unit, while channel 2, in blue represents the generator's rotor angle, in radians. All result windows present 20 seconds of measured real-time results captured at a sampling rate of 400 Hz. Also note that the solid lines on all channels represent the average value, while the shaded area around it is the measured noise. Furthermore, it is worth highlighting that the curves

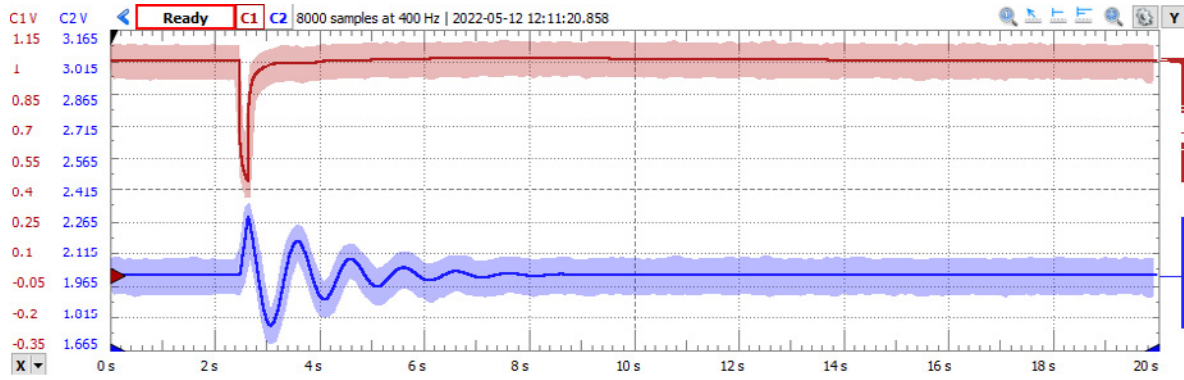
have different axes, as displayed in the left part of Figures 6a and 6b, and that they are measured in volts.

The averaged values behave as expected, and it is possible to clearly observe that the event occurs at approximately 2.5 s in Figure 6a and at 2.8 s in Figure 6b. This discrepancy between the event instant is due to a display time shift, since the measurement was done using two channels simultaneously. In addition, the simulation and measurement acquisition are not synchronized, which is why none of the curves present the result of the fault event at exactly at 2.0 s.

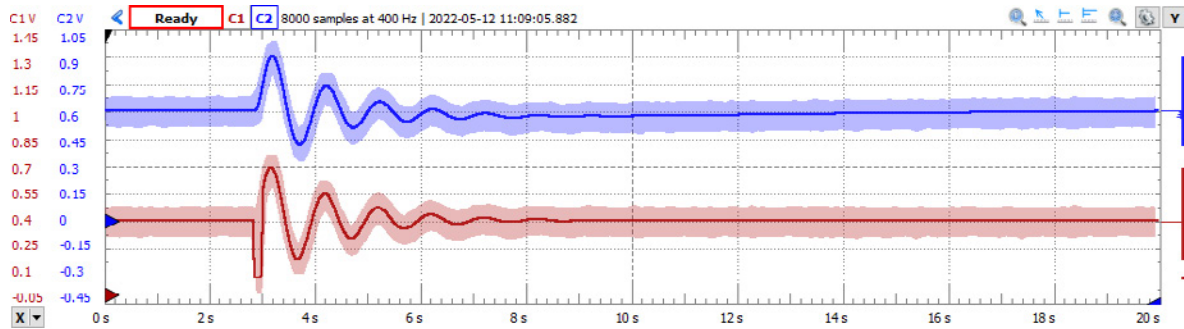
5.2 IEEE 9-Bus 3-Machine

The IEEE 9-bus 3-machine model is simulated in real-time in order to perform the tests using a slightly larger system. The measured results are depicted on Figure 7. The reactive power outputs of the STATCOM and generator 1 are represented, in per unit, by channels 1 and 2, respectively, in Figure 7a. Note that they are displayed with the same offset of 0.5 V and that the curves have different displacement in the y-axis, as shown in the left part of the plot. Note that, as the event occurs, both power system components peak in their reactive power generation.

The frequency deviation from the nominal value of 60 Hz, in generators 1 and 3, is displayed in Figure 7b. Channel 1, in red, depicts the quantity related to generator 1, while channel 2, in blue, pictures the quantity associated with generator 3. Note that both channels are, now, centered at the same value and they both have an offset of 1 V. The frequency deviations have opposing phase, suggesting that these two generators are not coherent. Moreover, observe that the event appears to occur at 3.05 s in Figure 7a and at 3.2 s in Figure 7b. The reason for this discrepancy is the same

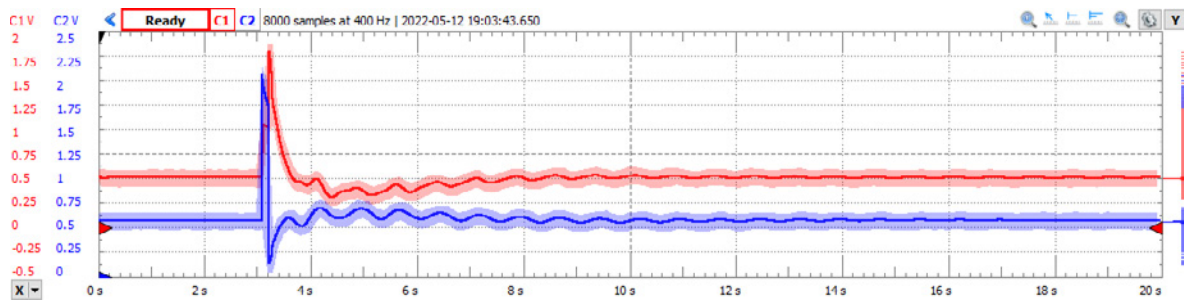


(a) Channel 1 represents the generator's output voltage, in per unit. Channel 2 represents the generator's deviation from nominal frequency, in hertz, and centered at 2 V.

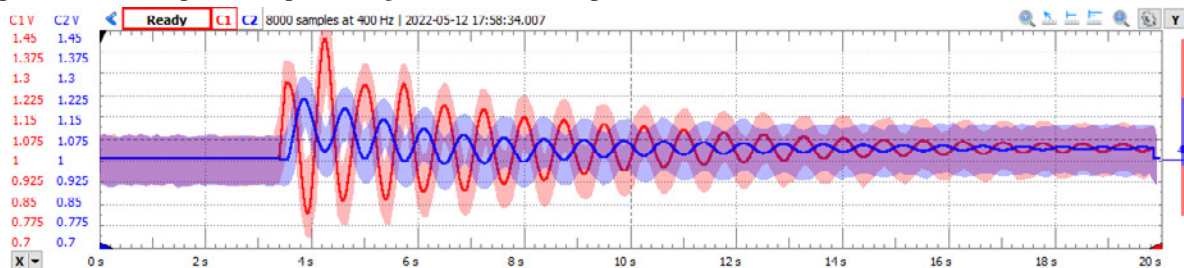


(b) Channel 1 represents the generator's electrical power output, in per unit. Channel 2 represents the generator's rotor angle, in radians.

Figure 6. SMIB system real-time simulation results measured with an USB oscilloscope.



(a) Channel 1 represents the reactive power output from the STATCOM, in per unit and centered at 0.5 V. Channel 2 represents reactive power output from generator 1, also in per unit and centered at 0.5 V.



(b) Channels 1 and 2 represent the deviation from nominal frequency, in hertz, on generator 1 and 3, respectively. Both values are centered at 1 V.

Figure 7. IEEE 9-bus 3-machine system real-time simulation results measured with an USB oscilloscope.

lack of synchronization between the measurement data acquisition and the real-time simulator, and the fact that both quantities are measured concurrently per experiment.

5.3 Discussion

Notable features observed in each simulation experiment are displayed in Table 2. Note that, although the IEEE 9-bus 3-machine model is $\approx 29\%$ larger than the SMIB in terms of the number of Differential and Algebraic Equations (DAEs), it has approximately $\times 6$ times more overruns during real-time execution. The IEEE 9-bus also has a task turnaround time 3.22 times higher, indicating that as the model increases in complexity, the time required for the simulator to execute each time step increases substantially. It is important to note that only one CPU from the processing unit was used in this study, meaning that for larger system models than the IEEE 9-bus, the model needs be separated into different FMUs for process parallelization, reducing the overruns. However, this is not assessed in this work.

Table 2. Comparison between studied systems and their performance in real-time simulation.

Studied System	SMIB	IEEE 9-bus
DAEs	369	476
Initialization Overruns	2	3
Simulation Overruns	16	106
Task Turnaround Time (μs)	≈ 83	≈ 268

In order to understand the reasons why these overruns are observed, it is possible to perform profiling of execution time using DymolaTM, before generating the FMU. By setting some simulation flags, it is possible to enable the plotting of the execution time needed for each time step throughout simulation. Figure 8 depicts the execution time needed for DymolaTM to compute each time step during simulation marked as green \times , while the simulator time step is highlighted in a dashed red line. Note that the number of execution times that are larger than the time step adopted for the simulator is similar to the value found in rows representing Initialization and Simulation Overruns, on Table 2. Furthermore, it is possible to observe that a large number of overruns result from the event that occurs at 2 s and 2.15 s, which correspond to a three-phase fault being applied and its clearing, respectively. It can also be found that a state associated with one lead-lag block from the excitation system dominates the error and, therefore, should be the responsible for the majority of overruns. This is because the excitation system will act to provide synchronizing torque so to help the generator keep in synchronism when the fault is applied, and thus, the equations of the excitation system will be governing the dynamic response of the system.

It is also possible to conduct a similar analysis for the

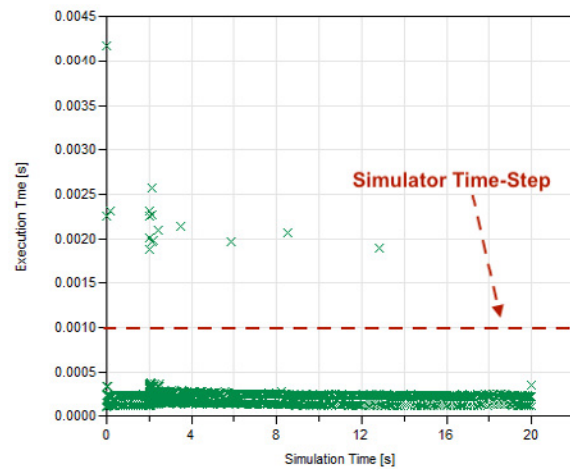


Figure 8. Comparison between execution time in DymolaTM and the simulator time step for the SMIB test system.

IEEE 9-bus 3-machine system and the execution time for DymolaTM, as shown in Figure 9 in green \times . Once again the simulator time step is highlighted in a dashed red line and it reveals when the overruns related on Table 2 that occur during simulation. Once again, the majority of the overruns occur during the fault being applied and its clearing. In this case, three states dominate the error; two related to first order blocks in the excitation systems of generators 1 and 2 and one related to a lead-lag control block used in the STATCOM. Similarly to the previous case, this is expected as both generators aid in stabilizing the system while the STATCOM aims to control its terminal voltage, thus engaging the equations that govern the dynamics of these three components.

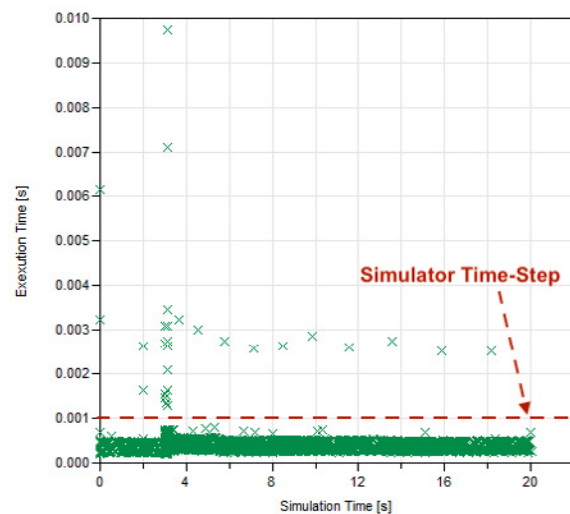


Figure 9. Comparison between execution time in DymolaTM and the simulator time-step for IEEE 9-bus 3-machine test system.

6 Conclusion

This paper presented a framework that allows power system simulation studies to be performed using a power network model, built using Modelica and OpenIPSL, that is exported using the FMI standard for execution in a real-time simulator. The framework uses dSPACE hardware and software tools, since it natively runs FMUs, allowing for the exported Modelica-based power system models to be readily loaded, with minimal modifications. The results from the real-time simulation can be measured and verified by the usage of I/O boards that are connected to the processing unit.

Power system models built using Modelica, via the OpenIPSL, are usually used for offline simulation. However, in this study, their potential for real-time simulation and their real-time execution performance were assessed and determined to be satisfactory, i.e. with limited overruns. More importantly, the process of “going from off-line to real-time” involved very little additional effort. Therefore, by adopting this framework, the authors were able to demonstrate that the same model might be used in *both* offline and real-time simulation, eliminating the time and effort allocated for model re-implementation and verification, resulting from enhanced power system model portability and tractability. This work hopes to serve for both the Modelica and power engineering communities, as a proof of concept of the potential of the Modelica and FMI open access standards for model portability and interoperability across different simulation use cases, which are currently performed using disparate and siloed tools and hardware.

As future works, the authors aim at assessing the real-time simulation of larger power systems by exploring process parallelization and the connection of multiple FMUs. Furthermore, the authors aim to perform control-hardware-in-the-loop experiments using this framework, allowing for fast prototyping of power system controllers while providing maximum model portability and traceability.

Acknowledgements

This material is based upon work supported in whole or in part by Dominion Energy, by the Center of Excellence for NEOM Research at the King Abdullah University of Science and Technology under grant OSR-2019-CoE-NEOM-4178.12, and by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office, Award Number DE-EE0009139. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

Baldini, EA and AP Fugill (1952). “A power system analogue and network computer [includes discussion]”.

- In: *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems* 71.1, pp. 291–297.
- Baudette, Maxime et al. (2018). “OpenIPSL: Open-instance power system library—update 1.5 to “iTesla power systems library (iPSL): A modelica library for phasor time-domain simulations””. In: *SoftwareX* 7, pp. 34–36.
- Bogodorova, T. et al. (2013). “A Modelica power system library for phasor time-domain simulation”. In: *IEEE PES ISGT Europe 2013*.
- Brown, Rodney J and William F Tinney (1957). “Digital solutions for large power networks”. In: *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems* 76.3, pp. 347–351.
- Chow, Joe H and Juan J Sanchez-Gasca (2020). *Power system modeling, computation, and control*. John Wiley & Sons.
- De Marco, Fernando, Nelson Martins, and Pablo Rullo (2021). “Using a SMIB Model to Analyse and Damp the Electromechanical Modes in Multigenerator Plants”. In: *2021 IEEE URUCON*. IEEE, pp. 252–257.
- De Marco, Fernando, Pablo Rullo, and Nelson Martins (2021). “Synthetic Power System Models for PSS Tuning and Performance Assessment”. In: *2021 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, pp. 107–112.
- Dommel, Hermann W (1969). “Digital computer solution of electromagnetic transients in single-and multiphase networks”. In: *IEEE transactions on power apparatus and systems* 4, pp. 388–399.
- dSPACE GmbH (2022-05). *Home-dSPACE*. <https://www.dspace.com/>.
- Evans, RD and RC Bergvall (1924). “Experimental analysis of stability and power limitations”. In: *Transactions of the American Institute of Electrical Engineers* 43, pp. 39–58.
- Faruque, MD Omar et al. (2015). “Real-time simulation technologies for power systems design, testing, and analysis”. In: *IEEE Power and Energy Technology Systems Journal* 2.2, pp. 63–73.
- Isaacs, Andrew (2017). “Simulation technology: The evolution of the power system network [history]”. In: *IEEE Power and Energy Magazine* 15.4, pp. 88–102.
- Kundur, Prabha (2007). *Power System Stability and Control*. McGraw Hill, Inc.
- Milano, Federico (2005). “An open source power system analysis toolbox”. In: *IEEE Transactions on Power systems* 20.3, pp. 1199–1206.
- Modelica Association (2022-05). *Modelica Language*. <https://modelica.org/modelicalanguage.html>.
- Modelica Association Project (2021-07). *FMI Standard*. <https://fmi-standard.org/>.
- OPAL-RT Technology Inc. (2022-05). *OPAL-RT website*. <http://opal-rt.com>.
- Stott, Brian (1979). “Power system dynamic response calculations”. In: *Proceedings of the IEEE* 67.2, pp. 219–241.
- Throckmorton, Paul Jeffrey and Louis Wozniak (1994). “A generic DSP-based real-time simulator with application to hydrogenerator speed controller development”. In: *IEEE Transactions on Energy Conversion* 9.2, pp. 238–242.
- Watson, Neville and Jos Arrillaga (2003). *Power systems electromagnetic transients simulation*. Vol. 39. IET.
- Winkler, Dietmar (2017). “Electrical Power System Modelling in Modelica - Comparing Open-source Library Options”. In: *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58)*, pp. 263–270.