Initial Steps in Deploying and Calibrating Power System Models on a Synchrophasor Data Cloud Platform using FMI

Luigi Vanfretti[1] Giuseppe Laera[1] Marcelo de Castro Fernandes[1] Chen Wang[2] Chetan Mishra[2] Kevin D. Jones[2]

[1]ECSE, Rensselaer Polytechnic Institute, USA, {vanfrl,laerag,decasm3}@rpi.edu
[2]Dominion Energy, USA, {chen.wang,chetan.mishra,kevin.d.jones}@dominionenergy.com

Abstract

This presentation gives a proof-of-concept of how to deploy power systems models to the cloud and how to couple them with measurements for model validation.

The presentation describes the re-implementation of models of a specific power plant in the Modelica language together with the OpenIPSL library, and how to bring confidence to Dominion engineers of the capability of the re-implementation to reproduce the results of the "trusted" proprietary tool. The models need to be ready to be coupled with PMU data and to be used for parameter estimation, which is part of a typical model validation process. Thus, we show how to prepare the models for export via the Functional Mock-Up Interface (FMI) standard. Finally, we illustrate how the models are deployed in the PredictiveGrid, a Phasor Measurement Unit (PMU) data analytics cloud platform, how they are coupled with PMU measurements and some calibration experiments results.
Keywords: power grid, power systems, OpenIPSL, cloud, model validation

1 Introduction

Power system modeling and simulation (M&S) are used to support decision making in planning and operation of electrical grids. However, for such studies, the models need to be updated to reflect the "real-world" exposed displayed in measurements (Z. Huang, P. Du, D. Kosterev, and S. Yang 2013). Keeping models updated is a complex task requiring measurement data collection and processing, validation of the model against the measured data, etc., involving laborious human intervention (NASPI 2015).

Dominion Energy is an American energy company owning more than 27,000 MW of power generation assets and 6,660 miles of power transmission lines across three states. Recently, Dominion adopted a cloud-based platform for synchrophasor measurement (a.k.a. Phasor Measurement Units (PMU)) analysis called PredictiveGrid. A great number of PMU data-based applications for grid analysis have been implemented and have provided tremendous value identifying existing problems in Dominion's grid. However, the lack of model-based analysis features in the platform limits the potential to explain the data from a "physics-based" perspective and to provide model-based solutions to observed problems.

As all utilities in North America, Dominion uses conventional power system M&S tools in their engineering tasks. However, the used models are "locked-in" to a specific tool (Z. Huang, P. Du, D. Kosterev, and S. Yang 2013; NASPI 2015), without means to deploy their equations, which define their physics, into modern computing environments, such as cloud platforms. This makes the model calibration to be in the sole purview of few tool-specific experts and limit the processes to update the models. Hence, it is obvious that an alternative to deploy models into PredictiveGrid was needed.

Dominion established a project with Rensselaer Polytechnic Institute (RPI) to address some of the challenges listed above. In the sequel, we describe the process that resulted in a proof-of-concept "workflow" that allows to deploy power systems models to the cloud and to couple them with measurements for model validation and calibration.

2 User Requirements

2.1 User Requirements

The goal of this project was to provide a proof-of-concept "workflow" to assist engineers in the model validation process and to exploit the models in combination with measurement data for other applications.

Dominion provided the following requirements to RPI to build a proof-of-concept prototype:

1. Use Jupyter Notebooks (JN) as the main user interaction environment and Python 3.6 (or newer) to develop the functions required.
2. Implement models of specific power plant(s) that can be deployed in the platform and accessed via JN.
3. The implemented models (a) must give the same response as the existing tool and (b) can be altered by the user.
4. Develop a prototype workflow, in JN, to compare the response of the model(s) to measurements corresponding to "ambient" conditions or disturbances.
5. Make the models available in the platform to develop other applications, e.g. parameter estimation.

A list of power plants, their models and existing parameter information was provided by Dominion's team, who also identified disturbances for different power plants. Additionally, access to PredictiveGrid was provided to RPI via OpenVPN connections.

2.2 Platform Constraints

The constraints of the cloud based platform were represented by its operating system, which is Ubuntu 18.04.3 LTS (Bionic Beaver), and the constraint of not being able to deploy third party proprietary software into the platform. Hence, these constraints made it impossible to leverage alternative tools for Modelica/FMI-based model calibration that have been used in the past by the authors' for similar purposes (Vanfretti et al. 2016; Luigi Vanfretti 2014; M. Podlaski, L. Vanfretti, J. Pesente and P. H. Galassi 2019; Podlaski et al. 2020).

The next section describes the building blocks and process adopted to develop a proof-of-concept prototype that meets user requirements under these constraints.

3 Developing a Proof-of-Concept Prototype

3.1 Building Blocks

After analyzing the requirements and constraints, several building blocks and needs were identified. They are listed first and discussed afterwards:

1. Additional Modelica models needed to be implemented for the specific power plants.
2. The models needed to conform to the response of the proprietary tool, PSS/E.
3. Signal processing methods needed to be implemented to access, process and use the measurement data.
4. Means to simulate the Modelica models within the platform were needed.
5. Optimization tools to solve the parameter estimation problem, i.e. calibrate the model.

Instead of building models within the Python language directly, it was preferred to leverage previous work that has shown the ability of Modelica-based OpenIPSL library power system models to reproduce the proprietary tools used by the power industry. However, the OpenIPSL did not have all of the component models needed to assemble the power plant models according to Dominion's description. Those components needed not only to be implemented, but verification against PSS/E was required. Results from this efforts will be illustrated in the presentation.

Furthermore, it was not possible to deploy a Modelica tool natively in the cloud platform due to different concerns of the platform provider. Hence, we decided to exploit the capability to export models using the Functional Mock-up Interface (FMI) standard. This also facilitated to address item 4 via the FMpy ((https://github.com/CATIA-Systems/FMPy)) and item 5 via ModestPy (Arendt et al. 2018).

Finally, the remaining component to develop was item 3, which was implemented from scratch in Python.

3.2 Preparing the Model for Export

After implementing the power plant models in Modelica and verifying them against PSS/E, the models were prepared for export. In Figure 1 the Modelica model of the plant under study exported as FMU and used in PredictiveGrid for calibration is illustrated. The components of the generating unit from the OpenIPSL library (Baudette et al. 2018) have been validated against the reference software PSS/E (Laera 2016), and additional ones that had to be implemented and verified for this work. Then they have been assembled and parametrized according to the original model of the plant.

The individual blocks of Figure 1 are numbered to describe easily their functions. Block 1 represents the record containing the system data. The constant blocks in 2 are used for power flow data for model initialization. The inputs (9) drive the controlled voltage source (3) through the real measurements of the voltage with its real and imaginary parts. The plant to be calibrated is represented with its components: generator GENROE (4), turbine governor IEEEG1 (5), exciter ESST1A (7) and power system stabilizer PSS2A (6). In block 8 the outputs linked to the active power P and reactive power Q of the generator are given and used to compare the real trajectories with the response of the system during the calibration process. A similar methodology has been implemented in (Podlaski et al. 2020).

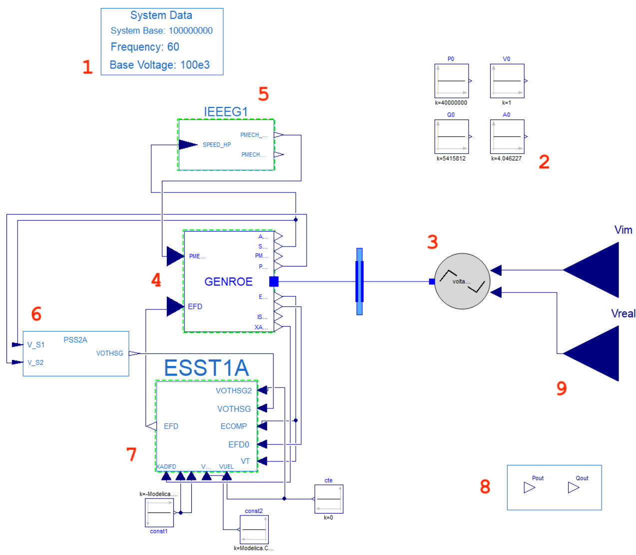

Figure 1. Modelica model of the plant exported as FMU.

3.3 Signal Processing

The main workflow structure is shown in Figure 2. The task is basically divided into two parts. First, it is necessary to retrieve data from the platform by selecting the data, the time window and the sampling rate. Then, after the data is retrieved, it needs to pass high-pass and low-pass filters to remove unwanted dynamics. Finally, the data is re-sampled in order to match with the FMU solver time step.

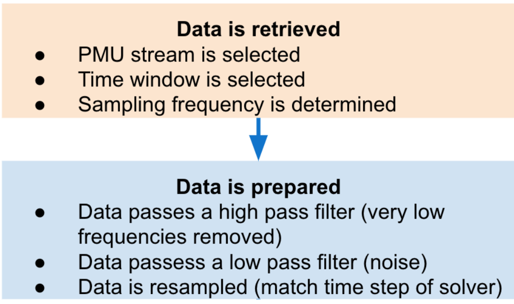

Figure 2. Signal processing basic flowchart.

3.4 Parameter Estimation

In Figure 3 the results of a calibration using measurements for a transient event are plotted. The red curves represent the real measurements of P and Q and the blue curves are the corresponding powers of the model after the calibration. In this case only four parameters have been calibrated and they are the gain and time constant of the exciter, the active and reactive powers (P0, Q0) of the blocks of power flow data. The parameters calibration has been performed through the open-source Python tool ModestPy (Arendt et al. 2018). The tool has been imported in the cloud through JN and the estimation of the chosen parameters has been carried out with a combination of algorithms: GA (Katoch, Chauhan, and Kumar 2021) + Nelder-Mead (Fuchang and Lixing 2018).

The results show that more parameters with different combinations need to be considered for a better calibration. It also important to run the parameters calibration with several scenarios for sensitivity analysis. As this is an on-going work, newer results will be shown during the presentation.

4 Conclusion

This abstract gave an overview of the current status of work towards deploying power system models into a cloud-hosted measurement data platform. The proof-of-concept prototype develop shows the value of Modelica and the FMI standard for model portability in new computing platforms, such as the cloud.

The proposed presentation will present in more detail examples of the models developed, Jupyter Notebooks developed and additional work currently being conducted to calibrate the developed models.

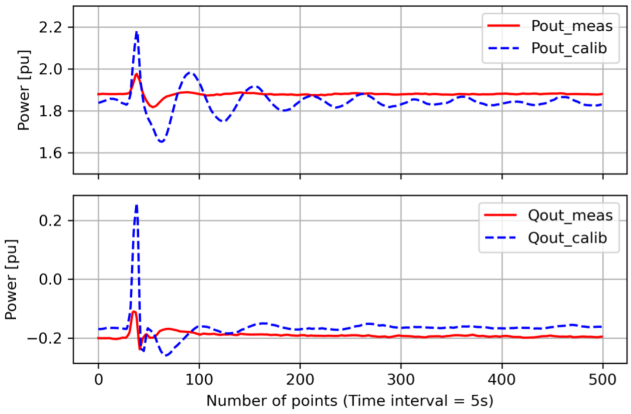

Figure 3. Calibration results for a transient event.

References

Arendt, Krzysztof et al. (2018-10). "ModestPy: An Open-Source Python Tool for Parameter Estimation in Functional Mock-up Units". In: pp. 121–130. DOI: 10.3384/ecp18154121.

Baudette, Maxime et al. (2018-01). "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. DOI: 10.1016/j.softx.2018.01.002.

Fuchang, Gao and Han Lixing (2018-05). "Implementing the Nelder-Mead simplex algorithm with adaptive parameters". In: *Computational Optimization and Applications* 51, pp. 259–277. DOI: 10.1007/s10589-010-9329-3.

Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar (2021-02). "A review on genetic algorithm: past, present, and future". In: *Multimedia Tools and Applications* 80, pp. 8091–8126. DOI: 10.1007/s11042-020-10139-6.

Laera, Giuseppe (2016). *Modelica Norwegian Grid Models for iTesla and Model Validation Tasks*. Tech. rep.

Luigi Vanfretti Tetiana Bogodorova, Maxime Baudette (2014). "A Modelica Power System Component Library for Model Validation and Parameter Identification". In: *Proceedings of the 10th International Modelica Conference* 96, pp. 1195–1203. DOI: 1650-3740.

M. Podlaski, L. Vanfretti, J. Pesente and P. H. Galassi (2019). "Automated Parameter Identification and Calibration for the Itaipú Power Generation System using Modelica, FMI, and RaPId". In: *7th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pp. 1–6. DOI: 10.1109/MSCPES.2019.8738793.

NASPI (2015-03). *Model Validation Using Measurement Data*. Tech. rep. https://tinyurl.com/NASPImodelVal. North American Synchrophasor Initiative.

Podlaski, Meaghan et al. (2020-11). "Parameter Estimation of User-Defined Control System Models for Itaipú Power Plant using Modelica and OpenIPSL". In: pp. 139–148. DOI: 10.3384/ecp20169139.

Vanfretti, Luigi et al. (2016). "RaPId: A modular and extensible toolbox for parameter estimation of Modelica and FMI compliant models". In: *SoftwareX* 5, pp. 144–149. ISSN: 2352-7110. DOI: 10.1016/j.softx.2016.07.004.

Z. Huang, P. Du, D. Kosterev, and S. Yang (2013). "Generator dynamic model validation and parameter calibration using phasor measurements at the point of connection". In: *IEEE transactions on power systems* 28.2, pp. 1939–1949.