

Modeling and Validation of Renewable Energy Sources in the OpenIPSL Modelica Library

Fernando Fachini, Luigi Vanfretti, Marcelo de Castro, Tetiana Bogodorova, Giuseppe Laera

Electrical, Computer and Systems Engineering Department

Rensselaer Polytechnic Institute

Troy, New York, U.S.A.

{fachif,vanfrl,decasm3,bogodt2,laerag}@rpi.edu

Abstract—This paper presents the implementation of industry grade renewable energy source phasor time-domain models from the Western Electricity Coordinating Council (WECC) Renewable Energy Modeling Task Force (REMTF) utilizing the OpenIPSL Modelica library. The modules described in this paper are generic, thus being able to be used to represent photovoltaic (PV), battery (BESS), and wind energy sources in transient stability scenarios, with added modularity given the object-oriented nature of Modelica models. The modeled components are briefly described, giving emphasis on how they were modeled in Modelica and coupled to the OpenIPSL library. Finally, the models described in this work were validated against the commercial software tool Siemens PTI PSS®E and showed to accurately replicate its results bringing validity to the Modelica-based implementation.

Index Terms—Modeling, simulation, power system dynamics, power system simulation, Modelica, OpenIPSL, solar PV, battery energy storage.

I. INTRODUCTION

The traditional approach to power system generation and transmission has been historically driven by large centralized power plants, mainly fossil fuel and hydro-power based, generating the bulk of the electrical power and making it readily available for distribution. In the US, for example, the power generation spectrum is broad, ranging from renewable energy sources to nuclear and fossil fuel sources, such as coal and gas. In 2020, roughly 60% of all power generation came from coal and gas [1]. However, fossil fuel based generation has sped up the amount of CO₂ injected into the atmosphere and is also linked to environmental issues regarding pollutants, mainly ash from coal combustion.

Based on the aforementioned issues, future energy sources have to meet the requirements of being flexible, accessible, reliable, economically viable, and environment friendly. The

This material is based upon work supported 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.

The work of M. de Castro and L. Vanfretti was supported in part by the New York State Energy Research and Development Authority (NYSERDA) under agreement numbers 137948 in part by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under Award EEC-1041877, in part by the CURENT Industry Partnership Program, in part by the Center of Excellence for NEOM Research at King Abdullah University of Science and Technology, and in part by Dominion Energy.

penetration of renewable energy sources in today's power grid, such as photovoltaic (PV), wind turbines (WT), and battery energy storage systems (BESS), has increased significantly and thus there is a need for better models that account for inverter-based generation sources. The Western Electricity Coordinating Council (WECC) Renewable Energy Modeling Task Force (REMTF) and the International Electrotechnical Commission (IEC) established renewable energy source (RES) generic models intended for power system dynamic studies, and that can somewhat emulate different vendor specific real life RES's. As described in [2], WECC's renewable models are implemented in several power system proprietary/commercial tools, each tool requiring its own implementation, which limits model portability and the possibility of using such models in heterogeneous simulation environments, and more importantly, substantially increases development cost and effort [3].

With the goal of providing model portability beyond power system specific tools [3], in this work, an implementation of these "industry grade" models is carried out using the Modelica language. The Modelica language offers an open access standardized specification [4], which enables model portability to multiple tools supporting the language specification. The portability between Modelica-based tools is ensured using the Functional Mock-up Interface (FMI) [5] standard. Considering the aforementioned advantages of Modelica, the engineers in different fields of study use the language for multi-domain modeling which expands the usability and applicability of the language.

The contributions of this work are:

- 1) Description and modeling of "industry grade" generic phasor time-domain renewable source models in the Modelica language and integrated with the OpenIPSL library [6].
- 2) Implementation of modular renewable models using object-oriented computer-based modeling techniques and tools (i.e. Dymola).
- 3) Validation of the output results against Siemens PTI PSS®E.

II. RENEWABLE ENERGY MODELS IN MODELICA

The models described in this paper are the so-called "second generation" generic renewable energy models from the

WECC. A detailed description of the models can be found in the Solar Photovoltaic Power Plant Modeling Guideline and the Battery Storage Dynamic Modeling Guideline from the Western Electricity Coordinating Council (WECC) [7], [8], and other sources, including software manuals [9], [10], and papers [2], etc.

These models are suitable for electrical transient stability dynamic studies, allowing for a phasor representation of generic renewable energy models in the electrical power system. The models not only are simpler to represent but are also much faster to simulate when compared to Electro-Magnetic models (EMT) [11]. The practicality of using phasor domain instead of waveform representation is that, by having a time-invariant frequency assumption, the set of differential equations that describes the states can now be characterized by algebraic equations [12]. This reduces the number of ultra-fast dynamic states that would otherwise need to be modeled. The new renewable models described in further subsections were implemented to expand the portfolio of existing Modelica models for power system dynamic simulation library in the OpenIPSL, an open-source Modelica based power system library [6]. There are three main functional blocks (i.e. modules) that constitute the second generation renewable energy models, as shown in Fig. 1:

- REGCA module: the name stands for renewable generator/converter model and it represents the current injection model for renewable sources, such as photovoltaic, batteries, and wind turbines.
- REEC module family: the name stands for renewable electrical controller and it represents the electrical controls of the electrical inverters. The electrical controller is connected to the REGCA module in a close loop connection, meaning that it acts on active and reactive power based on the output voltage and power from the REGCA module. For the PV model, the model to use is the REECB module while the BESS model requires using the REECC model.
- REPCA module: the name stands for renewable plant controller. It establishes the controlling scheme for a conglomerate of renewable sources. Similar to the electrical controller, it has all the aforementioned control functions, which are active and reactive power and voltage control, with the addition of frequency control for active power control.

In the sequel, names and variables in the figures, e.g. Fig. 1, are referred to in text using *italics*, while their counterparts in the Modelica language appear in `courier` font.

III. REGCA MODEL

The REGCA is a generic current source model that outputs active and reactive current based on the active and reactive power values of the renewable generator model and its voltage magnitude and angle. The renewable generator/converter model has two input signals, which are the real and imaginary command currents (*I_pcmd* and *I_qcmd*), determined via the electrical controller model, and two output signals which are

the real and imaginary currents that are injected into the system (*I_p* and *I_q*). The `p` pin, represented by the blue square in Fig. 2, is a Modelica interface [4] used in the OpenIPSL library to couple electrical variables and it contains four different variables that can be divided into two connector type groups:

- Potential Variables: *v_r* (Real component of the voltage) and *v_i* (Imaginary component of the voltage).
- Flow Variables: *i_r* (Real component of the current) and *i_i* (Imaginary component of the current).

In Modelica, potential variables represent quantities that are equal across a connection while flow variables sum up to zero across a connection [4]. For the REGCA model, the real and imaginary voltage values in the `p` pin are defined by the guess voltage value of the bus in which the generator model is connected to. The guess voltage value is assumed to be known, in practice this is provided by a static power flow solution, see [6], while a Modelica tool will solve for the initial value [13].

The currents that will be injected into the system are *I_p* and *I_q*, both of which are available in the form of real outputs of the REGCA model. The assignment from real valued variables to `p` pin flow variables is done through the equation:

$$p.ir = I_p \text{ and } p.ii = I_q. \quad (1)$$

The Modelica model also has five specific real-valued outputs, three of which are within the red hatched area and are used to provide the generator output power and voltage magnitude to the electrical controller block, and two within the green hatched area that are solely used to pass the values of active and reactive current values to the electrical controller block solved at initialization (i.e. at $t = 0$ [sec.]).

A. REGCA Initial Condition Parameter Calculation

Providing a good initial guess for the current source model and its controllers is key in guaranteeing that the system, as a whole, will successfully initialize to a valid steady state [13]. During initialization, Modelica tools attempt to find a solution to the entire set of Differential and Algebraic Equations (DAEs), at $t = 0$, which will be used to start the simulation. In order to do so, the user must provide the following guess values: generator's base apparent power, active power, base power, reactive power, and voltage magnitude. With the active & reactive power and the terminal voltage angle, it is possible to determine the real and imaginary components of the current with expression (2),

$$\begin{aligned} I_{p0} &= (p_0 \cdot v_{r0} + q_0 \cdot v_{i0}) / (v_{r0}^2 + v_{i0}^2) \\ I_{q0} &= (p_0 \cdot v_{i0} - q_0 \cdot v_{r0}) / (v_{r0}^2 + v_{i0}^2) \end{aligned} \quad (2)$$

where I_{p0} and I_{q0} are the initial active and reactive current in per unit, p_0 and q_0 are the initial active and reactive power of the renewable model in per unit, and v_{r0} and v_{i0} are the active and reactive voltage value at the terminal bus in per unit. To aid the initialization process, the I_{p0} and I_{q0} , which are the initialization values of the dynamic model, are

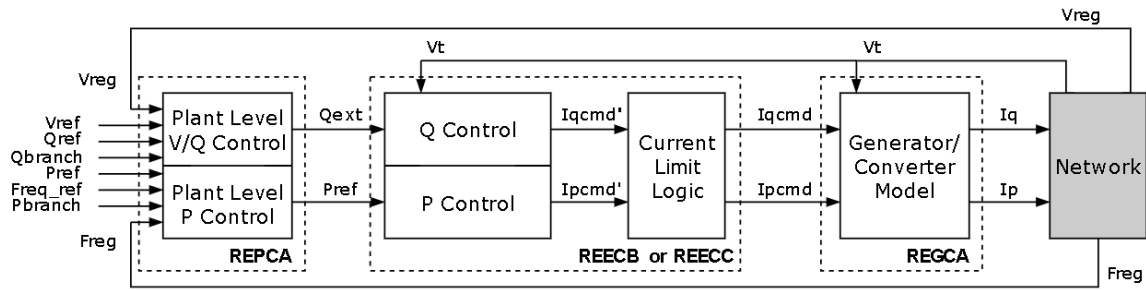


Fig. 1: WECC Block Diagram of the RES modules.

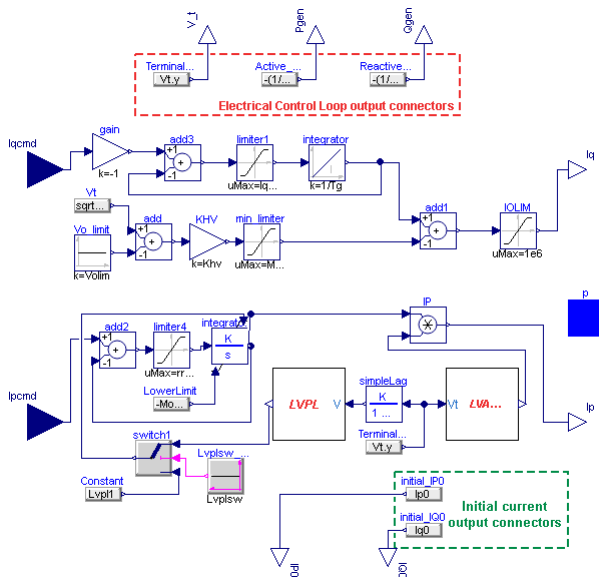


Fig. 2: REGCA Module in Modelica and OpenIPSL.

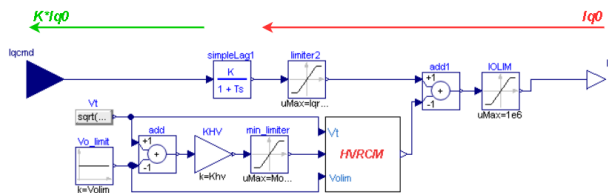


Fig. 3: Initialization procedure of the reactive current segment in the REGCA module.

computed as parameters. Then, they are specified as the guess start value through the modifier `start` in the `protected` section of model [4]. Then they are used to initialize the I_q and I_p output connectors as well as the starting output value of the transfer function blocks within the REGCA model, which is exemplified in Fig. 3 in the `simpleLag1` block. Figure 3 illustrates how these variables are computed to ensure that the model starts in steady state. The current I_{q0} , defined in equation (2), is used to initialize the state of the `simpleLag1` transfer function block, which is its output value. By the final value theorem, the input of the `simpleLag1` transfer function is $K \cdot I_{q0}$, shown by the green arrow in Fig. 3. Eventually, $K \cdot I_{q0}$ becomes the initialization value to the I_{qcmd} connector on the electrical controller side. This is similar for all other transfer function blocks from all modules from Fig. 1.

For this reason, I_{p0} and I_{q0} are also used to initialize the electrical controller block, therefore the value of both real and imaginary currents is outputted by the REGCA model to be used in the REECB/REECC electrical models, depicted by the green hatched area in Fig. 2. This is a common practice used in several other OpenIPSL source/controller components, see [13].

B. REGCA Model Description

The REGCA generator/inverter model, has the capability of adjusting the active and reactive current of the model when its terminal voltage reaches its limits. For instance, when the terminal voltage of the generator exceeds a maximum value, V_{o_lim} , the model decreases the amount of reactive current that will be injected into the system and consequently reduces the voltage magnitude.

The `min_limiter` block in the blue hatched area from Fig. 2 is a limiter block with a specified min. and infinite max. limits. While the generator's terminal voltage (V_t) is lower than a maximum value set by the user (V_{o_lim}), the difference between these two values is negative and the output of the `min_limiter` block is zero. If the generator's terminal voltage increases and surpasses the limiting voltage value, the difference in the addition block becomes positive and the output of the `min_limiter` block becomes positive. This signal is then subtracted in the reactive current segment of the REGCA model and I_q reduces its values as a consequence. In situations where the terminal voltage drops below a minimum threshold, set by the user, the model is capable of limiting the value of the active current (I_p) in order to protect the inverter circuit of the PV. This mode of operation is achieved through the Low Voltage Active Current Management component of the REGCA model, shown by the cyan hatched area in Fig. 2.

IV. REECB AND REECC MODEL

This component has five different input signals, as shown in Fig. 4. Of these, the ones shown in the red hatched area come from the REGCA model outputs. Next, as shown in Fig. 4, the inputs V_t , P_e , and Q_{gen} are the terminal voltage, and active and reactive power generated by the REGCA module. The set of real-valued inputs marked by the purple hatched area are the active and reactive power references, where the user must connect to either a constant value reference or to the REPCA component. The guess values used during initialization of

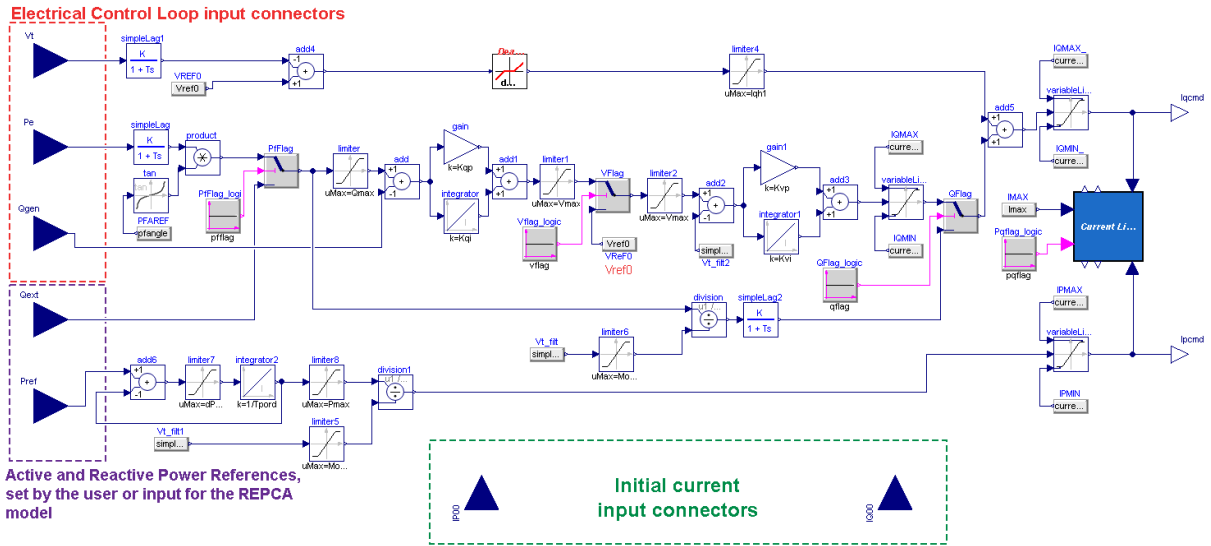


Fig. 4: REECB Module in Modelica and OpenIPSL.

the electrical controller model are computed similarly to the initialization example in Fig. 3. A key distinction from the REGCA and REECB/REECC models is that the electrical controller models have four logical switches enabling for four distinct control schemes.

Due to page limit restrictions, the combination of control flags will not be discussed. The reader is referred to [7], [8] for more information.

A. Differences between REECB and REECC Models

The Renewable Electrical Control Component (REECC) model of the WECC is used to represent BESS models with control subsystems that are meant to control I_{pcmd} and I_{qcmd} to the REGCA model, see Fig. 1. This is a similar model to the REECB, however, the State Of Charge (SOC) of the battery is added. Since the REECC component models the electrical controller of a battery, charging and discharging the battery will certainly have a different behavior when compared to its PV counterpart. The battery has two limit thresholds that are the max. and min. SOC allowable in the BESS. Once the charge of the battery surpasses the maximum limit or is lower than the minimum limit, the battery will stop injecting or consuming power, resulting in a null-valued active power injection. Figure 5 is the SOC component that differentiates REECC from REECB. The SOC_logic block in Fig. 5 contains

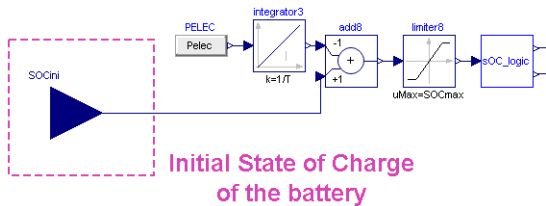


Fig. 5: Fragment of the REECC electrical controller responsible for the SOC of the battery.

the expression responsible for tracking the SOC of the battery, which is:

$$\begin{aligned} ipmax_SOC &= \text{if } SOC \leq SOCmin \text{ then } 0 \text{ else } 1 \\ ipmin_SOC &= \text{if } SOC \geq SOCmax \text{ then } 0 \text{ else } 1 \end{aligned} \quad (3)$$

where $ipmax_SOC$ and $ipmin_SOC$ are the output signals from the SOC_logic block, SOC is the input signal, and $SOCmin$ and $SOCmax$ are the user-defined parameters for max. and min. charge limits in the battery.

V. REPCA MODEL

The REPCA model, shown in Fig. 6, is the renewable plant controller component, and is used to emulate plant level control of injected/consumed active and reactive power of the RES models. As shown in Fig. 6, the REPCA component has four inputs (Q_{ref} , $Plant_pref$, $Freq$, and $Freq_ref$), two outputs (Q_{ext} , and $Pref$), and two OpenIPSL interfaces, i.e. ($BRANCH_p$ and $REGULATE$). $Plant_pref$, and Q_{ref} are the active and reactive power reference inputs of the model. In the case of PSS@E, these reference inputs are constant and are derived via power flow solution, meaning that a simulation is valid for a limited time window where solar irradiance and power are roughly the same throughout. For Modelica models, these inputs can be varied during simulation, for example using tabulated data with irradiance to power information. This allows to conduct simulation studies ranging from 10s of minutes to hours. In Fig. 6, $Freq$, and $Freq_ref$ are the frequency and reference frequency of the system; Q_{ext} , and $Pref$ are REPCA's output values of active and reactive power. The blue square pins $BRANCH_p$, and $REGULATE$ are used to obtain the active and reactive power injection of the park and the voltage of the chosen regulation bus.

Once again, due to page limit constraints, the combination of control flags and in depth characteristics of the REPCA module will not be discussed here. The reader is referred to [7], [8] for more information.

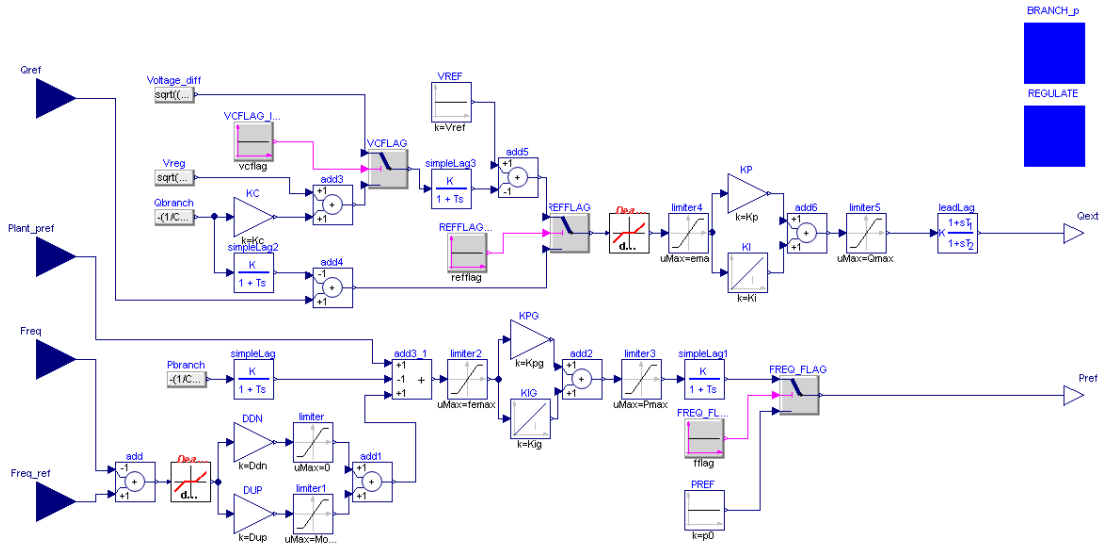


Fig. 6: REPCA Module in Modelica and OpenIPSL.

VI. MODEL VALIDATION AGAINST PSS@E

The validation of the aforementioned models is carried out by comparing simulation results obtained from running simulations in Dymola, a simulation tool for Modelica models, and comparing the output results against the solution from PSS@E. For that, the same test scenario is used in both models and a depiction of test case model in Dymola is displayed in Fig. 7. Within Dymola, all simulations were obtained in a time window of 10 [sec.], with 10000 simulation intervals and a tolerance of 0.001.

A. Test System Information

The *pwFault* component in Fig. 7 represents the three phase to ground fault in a test system model and is used to generate a transient in the system to verify the models' response similar to [2]. The voltage magnitude at the *FAULT* bus will decrease as a result of the current splitting its flow between the fault to ground and the transmission lines *pwLine* and *pwLine1*. The component connected in the right hand side of the bus *GEN2* is an infinite bus model from OpenIPSL library. An equivalent model was implemented in PSS@E to perform the comparisons. Finally, the test model parameters are:

- *pwLine2*: line impedance $\tilde{Z} = 0.0025 + j0.0025$ [p.u.].
- *pwLine*: line impedance $\tilde{Z} = 0.025 + j0.025$ [p.u.] and $\frac{1}{2}$ shunt susceptance $\hat{B}/2 = j0.025$ [p.u.]. The same values are set in *pwLine1*.

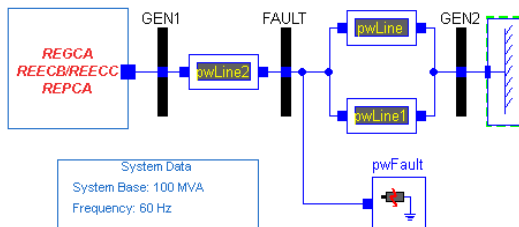


Fig. 7: Validation test system model template.

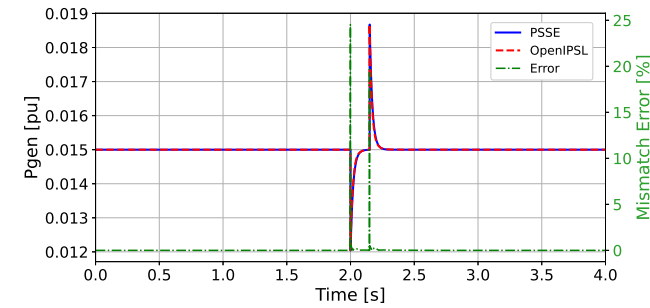
- *pwFault*: short circuit impedance $\tilde{Z} = 0.5 + j0.5$ [p.u.]. The fault is applied at $t=2$ seconds and removed at $t=2.15$ [sec.].
- Power flow: the initial active and reactive power injection of the renewable generator model is 1.5 [MW] and -5.6658 [Mvar], respectively. The terminal voltage magnitude is 1 [p.u.] with initial bus angle of 1.47 [°].

B. Comparison Tests

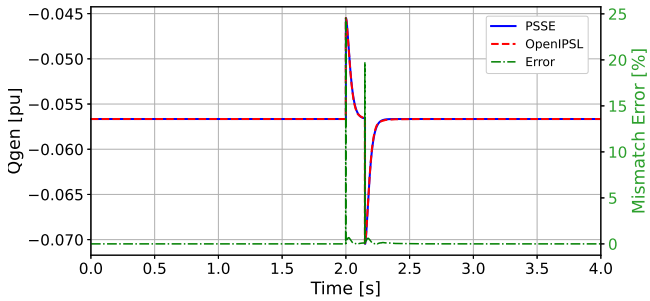
WECC's PV and BESS Modeling Guideline Reports discusses all the possible control flag combinations available, eight different control schemes per model. Due to space limitations, only two tests will be presented to prove the validation process.

1) *Validation Test 1 — Constant Local Reactive Power Control for PV model*: The control scheme in the first test utilizes the REGCA and REECB modules. The flag configuration in the electrical controller for this particular test is: *pfFlag* = false, *vFlag* = true or false, *qFlag* = false, and *pqFlag* = true or false. In the case of this test configuration, the external reactive power reference Q_{ext} is constant, which explains the name of the control scheme. The output curves for active and reactive power are shown in Fig. 8a, and 8b. Both curves match the reference software PSS@E, with a root mean square error (RMSE) of 6.75×10^{-5} for P_{gen} and 2.56×10^{-4} for Q_{gen} . The mismatch error in percent using PSS@E as reference is shown in the green curve. The error plot displays two spikes that are due a 1×10^{-3} [sec.] difference in fault occurrence and clearance between PSS@E and Dymola. As time progresses the error decays and the error curve goes to zero.

2) *Validation Test 2 — Plant Level Reactive Power Control + Local Coordinated Voltage/Reactive Power Control for BESS model*: The control scheme in the second test utilizes REGCA, REECC, and the REPCA modules. The flag configuration in the electrical controller for this particular test is: *pfFlag* = false, *vFlag* = true, *qFlag* = true,



(a) P_{gen} RMSE = 6.75×10^{-5} .



(b) Q_{gen} RMSE = 2.56×10^{-4} .

Fig. 8: (a) Active Power response for PV, Validation Test 1, and (b) Reactive Power response for PV, Validation Test 1.

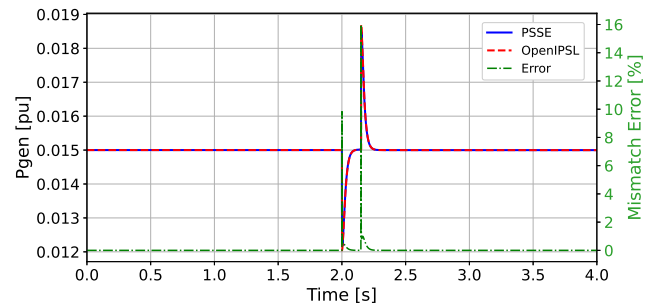
and `pqflag = true` or `false`. The flag configuration in the plant controller is: `vcflag = true`, `refflag = false`, and `fflag = false`. In this test configuration, the reactive power from the measured `BRANCH_p` is the signal that is compared against the active power reference Q_{ref} in the REPCA module. In addition, there is a coordinated voltage/reactive power controller within the electrical controller. The output curves for active and reactive power are shown in Fig. 9a, and 9b. Both curves match the reference software PSS@E, with a RMSE of 3.89×10^{-5} for P_{gen} and 1.49×10^{-4} for Q_{gen} . Once again, the green error plot displays error spikes at both fault occurrence and fault clearance times, similar to Test 1. The error decays over time, reaching zero.

VII. CONCLUSION

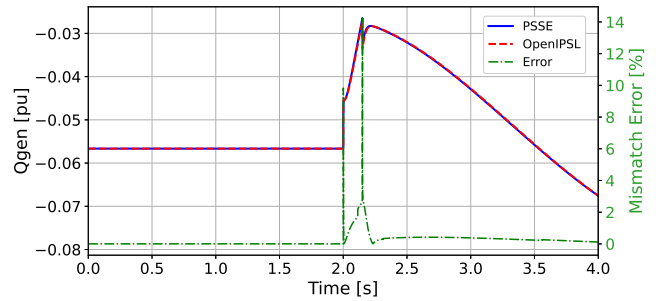
This work presented the implementation and validation of generic renewable energy source models using the Modelica language and the OpenIPSL library, based on the modeling guidelines from WECC. The models are based on open-source software, they are reusable and generic, and capable of emulating a RMS equivalent of inverter based renewable energy sources, such as PV, Wind, and BESS. The validation results point to the success of the modeling task, and the models will be released as open source software by integrating them in a future version of the OpenIPSL library once included in the continuous integration process [14]. The library can be found at: <https://github.com/OpenIPSL/OpenIPSL>.

REFERENCES

- [1] US Energy Information Administration, “Electricity Explained,” <https://tinyurl.com/ieaWEECModels>, 2021.



(a) P_{gen} RMSE = 3.89×10^{-5} .



(b) Q_{gen} RMSE = 1.49×10^{-4} .

Fig. 9: (a) Active Power response for BESS, Validation Test 2, and (b) Reactive Power response for BESS, Validation Test 2.

- [2] G. Lammert *et al*, “Implementation and validation of WECC generic photovoltaic system models in DiGSILENT PowerFactory,” in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, 2016, pp. 1–5.
- [3] L. Vanfretti *et al*, “Unambiguous power system dynamic modeling and simulation using Modelica tools,” in *2013 IEEE Power and Energy Society General Meeting*, 2013, pp. 1–5.
- [4] P. Fritzson, *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*. John Wiley & Sons, 2014.
- [5] T. Blochwitz *et al*, “Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models,” in *Proceedings of the 9th International Modelica Conference*, 2012, pp. 173–184.
- [6] M. Baudette, M. Castro, T. Rabuzin, J. Lavenius, T. Bogodorova, and L. Vanfretti, “OpenIPSL: Open-instance power system library—update 1.5 to “iTesla power systems library (iPSL): A Modelica library for phasor time-domain simulations,”” *SoftwareX*, vol. 7, pp. 34–36, 2018.
- [7] WREM Task Force, “WECC battery storage dynamic modeling guideline,” *Western Electricity Coordinating Council, Tech. Rep.*, 2016.
- [8] WECC Renewable Energy Modeling Task Force, “Solar photovoltaic power plant modeling and validation guideline,” 2019.
- [9] Siemens PTI, “PSS@e 34.2.0 model library,” *Siemens Power Technologies International, Schenectady, NY*, 2017.
- [10] E. Muljadi, M. Singh, and V. Gevorgian, “User guide for PV dynamic model simulation written on PSCAD platform,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2014.
- [11] X. Mao and R. Ayyanar, “Average and phasor models of single phase pv generators for analysis and simulation of large power distribution systems,” in *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*. IEEE, 2009, pp. 1964–1970.
- [12] J. A. Svoboda and R. C. Dorf, *Introduction to electric circuits*. John Wiley & Sons, 2013.
- [13] M. Zhang, M. Baudette, J. Lavenius, S. Løvlund, and L. Vanfretti, “Modelica implementation and software-to-software validation of power system component models commonly used by nordic tsos for dynamic simulations,” in *Proceedings of the 56th Conference on Simulation and Modelling (SIMS 56), October, 7-9, 2015, Linköping University, Sweden*, no. 119. Linköping University Electronic Press, 2015, pp. 105–112.
- [14] T. Rabuzin, M. Baudette, and L. Vanfretti, “Implementation of a continuous integration workflow for a power system Modelica library,” in *2017 IEEE Power and Energy Society General Meeting*, 2017, pp. 1–5.