OpenIWPI: Open-Instance Wave-Phasor Interface Library for Power System Simulation Studies in Modelica

Marcelo de Castro[1] Luigi Vanfretti[2]

[1]Mitsubishi Electric Power Products, Inc., USA marcelo.decastro@meppi.com
[2]Rensselaer Polytechnic Institute, USA vanfrl@rpi.edu

Abstract

In the past decades, a large number of power electronic converter-based energy resources have been connected to grids around the world. The increasing number of such devices is pushing for both electromechanical and electromagnetic dynamic simulations to be part of routine stability studies for transmission planning studies. Hybrid wave-phasor simulations have been proposed in the literature to address this challenge. This paper describes Open-Instance Wave-Phasor Interface (OpenIWPI), a Modelica library used to couple phasor-based and electromagnetic power system models, allowing simulation and linearization of such hybrid models to be performed altogether. Examples are presented to describe how the library can be useful in power system studies.
Keywords: power grid, power systems, electromagnetic transients, phasor-based simulation, RMS simulation

1 Introduction

1.1 Background and Motivation

The planning of the grid and its safe operation requires many different simulation-based studies to be performed. The dynamic stability analysis of electric grids is one of these studies, and it is usually assessed using two different approaches, depending on the time scale of the phenomena that need to be assessed. In electromechanical transient stability (usually named just Transient Stability, or TS) simulations, the power system is represented in its positive sequence component equivalent to the fundamental frequency (Kundur, Balu, and Lauby 1994). In essence, a phasor-based representation of the power system is used to create a system of differential and algebraic equations (DAEs). The time scale of electromechanical system dynamics is in the order of milliseconds to tens of seconds and, with appropriate time step selection, a large number of DAEs can be solved via low-order integration methods with high computational efficiency (Kundur, Balu, and Lauby 1994; Chow and Sanchez-Gasca 2020). Electromagnetic transient stability (EMT) simulations have more detailed models of the components of the power system, capturing ultra-fast switching dynamics due to power line tripping or those of the power electronic converter (Dommel 1986; Watson and Arrillaga 2003; Mahseredjian, Venkata Dinavahi, and Martinez 2009). Once again, with an appropriate time step selection, the differential equations describing the EMT simulation are solved for a large frequency spectrum using trapezoidal integration methods. However, the time scale of electromagnetic phenomena is very small, and solving these large systems of differential equations with very small time steps is cumbersome for computers, limiting the study of EMT simulations in extremely large systems (Watson and Arrillaga 2003). Moreover, the information required for the parameterization of EMT models is substantially larger than that used for phasor-based models, and some of this information may not be available, making it challenging to develop large-scale grid models, with some exceptions (Mahseredjian, Venkata Dinavahi, and Martinez 2009; Dennetiere et al. 2019; Thapa et al. 2023).

Traditionally, TS and EMT studies were carried out separately, in different tools. However, there have already been efforts, since the 1980s, for hybrid EMT-TS simulations motivated by the need to understand the interaction between high voltage direct current (HVdc) transmission lines and the power grid (Heffernan et al. 1981; Adapa and Reeve 1988). In recent decades, the electric power system has grown not only in size but also in complexity. Currently, many converter-based resources are operating in modern power systems, and many more will be in service in the next years. These ubiquitous devices deal with large amounts of power and are crucial for the operation of the current power grid. Therefore, it is important for studies to consider how the electromagnetic response of such devices can affect the electromechanical transient stability of the entire system and vice versa.

Many power system libraries have already been developed in Modelica for both TS and EMT studies (Winkler 2017; de Castro, Winkler, et al. 2023; Masoom, Ould-Bachir, et al. 2020; Masoom, Mahseredjian, et al. 2021). Engineers have created these libraries to explore the rich features of Modelica (Modelica Association 2025) while performing EMT or TS studies. Therefore, it is attractive to explore how these can be combined and exploited for hybrid EMT-TS simulations without the typical drawbacks of co-simulation (Schweiger et al. 2019). To this end, hybrid wave-phasor interfaces using Modelica would enable one to adequately model and simulate modern power systems that are rich in converter-based resources (Chang and Vanfretti 2024; Fachini et al. 2025).

Furthermore, in recent years hybrid EMT-TS studies have become more common (Watson and Arrillaga 2003). However, they are usually conducted mainly as co-simulation, with the different parts being simulated as if they were standing alone (Heffernan et al. 1981; Adapa and Reeve 1988; Theodoro, Tomim, Lima, et al. 2017; Theodoro, Tomim, Barbosa, Lima, and Santiago 2018; Theodoro, Tomim, Barbosa, Lima, and Barros 2019). While the EMT part is discretized to be solved using trapezoidal integration method, the TS part uses modified Euler algorithms. This is not done explicitly; instead, the dominant approach used by the power systems community over the last 15 years (V. Dinavahi et al. 2009) is to interface two separate environments that exchange data at specific time instants, i.e., *ad-hoc* co-simulation. However, using Modelica, hybrid EMT-TS simulation is possible without the need of two different tools — one can create a single hybrid EMT-TS model that will be solved using variable step-size algorithms (Gibert et al. 2018; de Castro and Vanfretti 2022), such as DASSL for example (Petzold 1982). Meanwhile, if needed, co-simulation can also be achieved if desired, by exporting portions of the proposed interface and models as a Functional Mock-up Unit (FMU) and interfacing them with FMI-compliant tools (Ramirez-Gonzalez et al. 2024).

It is also worth noting that the ability to create a single hybrid EMT-TS model in Modelica provides a unique added value: it allows the user to perform a small signal stability analysis using linearized models, e.g., by using the `Modelica_LinearSystems2` (Baur, Otter, and Thiele 2009). This is simply not possible with existing power system EMT simulators and certainly not feasible with EMT-TS co-simulation frameworks. Observe that even most TS simulators require additional tools for this analysis task (Martins et al. 2000; Nikolaev et al. 2020).

1.2 Objectives

In this paper, the focus will be in presenting a Modelica library that contains an interface that can be used for hybrid EMT-TS simulations. This wave phasor interface is designed to be coupled with the Open Instance Power Systems Library (OpenIPSL) (de Castro, Winkler, et al. 2023) and the Modelica Standard Library (MSL). While the phasor-based portion of a system can be represented by OpenIPSL, the electromagnetic behavior is represented by the appropriate Electrical sub-package within the MSL. The wave-phasor interface present in this library is based on a positive-sequence component filtering approach (Theodoro, Tomim, Lima, et al. 2017; Theodoro, Tomim, Barbosa, Lima, and Santiago 2018; Theodoro, Tomim, Barbosa, Lima, and Barros 2019) originally developed for co-simulation.

1.3 Paper Structure

The remainder of this paper is structured as follows: Section 2 presents the structure of the library and its organization; Section 3 shows the auxiliary models that are used in the library; Section 4 describes the mathematical description of the wave-phasor interface and its implementation in Modelica; Section 5 brings two simulation examples and their results; Final discussion and conclusions are presented in Section 6.

2 Library Overview and Organization

The Open-Instance Phasor-Wave Interface (OpenIWPI) library was developed to help engineers simulate hybrid power system models composed of phasor-based circuits and detailed components represented in the electromagnetic-transient fashion. The main idea is to be able to interface detailed components, using EMT-like representations, with power system grids, which are usually modeled in their phasor-domain positive sequence equivalents. In other words, this library should help engineers in creating hybrid EMT-TS models and simulation using a wave phasor interface (Theodoro, Tomim, Lima, et al. 2017; Theodoro, Tomim, Barbosa, Lima, and Santiago 2018; Theodoro, Tomim, Barbosa, Lima, and Barros 2019) implemented in Modelica (de Castro and Vanfretti 2022).

The interface uses electrical connectors from the OpenIPSL (de Castro, Winkler, et al. 2023) and from the Modelica Standard Library - MSL. Therefore, the library is ideal for interconnecting phasor-based representations built using the OpenIPSL, for TS studies, with EMT-like models that are built using the Electrical library within the MSL. As an outcome, the user will need to have both libraries loaded (OpenIPSL and MSL) before loading the OpenIWPI library. The OpenIWPI library has six sub-packages as shown in Figure 1.

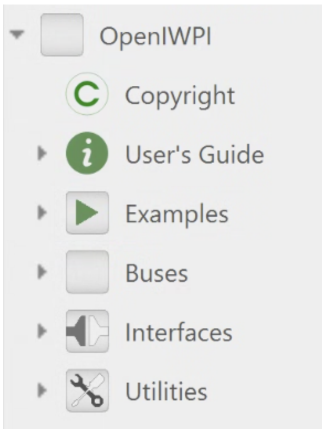

Figure 1. Library sub-packages overview.

The sub-packages are briefly described below:

- **Copyright:** the library's copyright notice informing users that the library is subject to a 3-clause BSD license.
- **User's Guide:** package containing informative items for the users, such as a starting guide and a list of

reference papers used by the library.

- **Examples:** package to demonstrate the application of the interface using simple examples of the power system for simulation.
- **Buses:** contains the electrical bus that uses the hybrid wave-phasor interface to connect phasor-based models with EMT-like models.
- **Interfaces:** contains the filter-based wave-phasor interface that allows the simulation of hybrid wave-phasor (EMT-TS) circuits.
- **Utilities:** contains auxiliary models that are used in the library, such as a Phase-Locked Loop (PLL) model, reference frame transformations, as well as resettable and second-order generalized integrators.

The OpenIWPI has two important sub-packages imported from its dependencies. They are listed in the library's main `package.mo` file as shown below. The first import is related to OpenIPSL's `Types`, which allows OpenIWPI to maintain dimension consistency in calculations made in the library. The second import is related to the mathematical constants listed in MSL and that are used in the calculations performed within the OpenIWPI.

Listing 1. Importing OpenIPSL Types and Modelica Constants.

```
import OpenIPSL.Types;
import C = Modelica.Constants;
```

3 Auxiliary Models and Blocks

The `Utilities` sub-package of the OpenIWPI library is composed of many auxiliary blocks and models that are essential for assembling the wave-phasor interface. The following subsections describe each auxiliary block/model in detail, highlighting its contribution to the proper operation of the interface.

3.1 Reference Frame Transformations

In the library, two reference frame transformations are needed: Clarke and Park transformations. Both transformations are implemented as models in the OpenIWPI library. The Clarke transformation model uses three real inputs (v_A, v_B, and v_C) and outputs two real variables (v_α and v_β) (Yazdani and Iravani 2010). In Modelica, this transformation is implemented as shown in Listing 2.

Listing 2. Clarke transformation implementation.

```
equation
    v_alpha = (2/3)*(v_A - (1/2)*v_B -
        (1/2)*v_C);
    v_beta = (2/3)*(sqrt(3)/2)*(v_B - v_C);
```

Similarly, the Park transformation model uses three real inputs (v_α, v_β and δ) and outputs two real variables (v_d and v_q) (Yazdani and Iravani 2010). The Park transformation transforms a reference frame composed of two waves in quadrature ($\alpha\beta$) into a set of two stationary values (dq) using a reference frame that rotates with the grid's frequency. Its Modelica implementation is as depicted in Listing 3.

Listing 3. Park transformation implementation.

```
equation
    v_d = v_alpha*Modelica.Math.cos(delta)
        + v_beta*Modelica.Math.sin(delta);
    v_q = -v_alpha*Modelica.Math.sin(delta)
         + v_beta*Modelica.Math.cos(delta);
```

3.2 Resettable Integrator

The integrator with a reset option was created in order to be used in the PLL present in the library. This integrator is a re-implementation of the regular integrator used in the MSL. The only difference is that it resets its output every time it reaches the value of 2π. The reason behind this reset is that it allows the output to always be within one angular revolution, avoiding an unnecessary large-angle variable to be used in calculations. This reset integrator is common in hardware implementations of PLLs based on the Synchronous Reference Frame (SRF) to avoid an integer overflow issue in the digital signal processor. In Modelica, the resettable integrator equations are implemented as shown in Listing 4. Note that this integrator uses the same parameters as the basic integrator found in MSL. Also note that Listing 4 is built under the assumption that `y_start` is 0, and that `k*u` is greater than 0.

Listing 4. Resettable integrator's equations in Modelica.

```
initial equation
    y = y_start;
equation
    der(y) = k*u;
    when y > 2*C.pi then
        reinit(y,y_start);
    end when;
```

3.3 Second-Order Generalized Integrator

The Second-Order Generalized Integrator (SOGI) (Rodriguez et al. 2006) is a filter with the purpose of generating two signals in quadrature with each other. The input of the filter, v_{in}, is filtered, and its fundamental component is output as v. The second output is a signal in quadrature with the first output and is named qv. If the transfer function between v_{in} and v is called $D(s)$ and the transfer function between v_{in} and qv is $Q(s)$, then it can be described as Equation (1) (Rodriguez et al. 2006; de Castro and Vanfretti 2022).

$$\begin{cases} D(s) & = \dfrac{v}{v_{in}} = \dfrac{k\omega_s s}{s^2 + k\omega_s s + \omega_s^2}, \\ Q(s) & = \dfrac{qv}{v_{in}} = \dfrac{k\omega_s^2}{s^2 + k\omega_s s + \omega_s^2}, \end{cases} \tag{1}$$

where the gain $\omega_s = 2\pi f_s$ is the angular frequency of the wave that must be filtered and k is a gain that can be adjusted to change the characteristics of the filter. These two parameters can be set by the user. In this application, $f_s = 60$ hertz, while the gain is set to $k = \sqrt{2}$ because it provides a dynamic response with a good compromise between the settling time and the overshoot of the filtering process under disturbances (Rodriguez et al. 2006; de Castro and Vanfretti 2022). Note that $D(s)$ describes a band-pass filter, while $Q(s)$ describes a low-pass filter. In OpenIWPI, the SOGI has been implemented as a block diagram. It is shown in Figure 2.

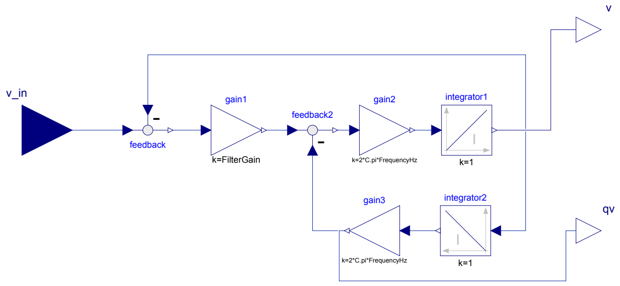

Figure 2. Sencond-order generalized integrator implementation.

3.4 Phase-Locked Loop

The PLL modeled in the library is based on a Synchronous Reference Frame (SRF) loop with double SOGI filters, described below (Rodriguez et al. 2006). The PLL can be divided into two main parts. The first part acts as a positive sequence filter, having three-phase quantities as input (voltage or current), and having as output the positive sequence equivalent for the corresponding $\alpha\beta$ components of the input quantities (Rodriguez et al. 2006; de Castro and Vanfretti 2022).

Suppose the three-phase quantities v_A, v_B, and v_C. They are used as inputs for a Clarke block. Each output, say $v_{\alpha,in}$ and $v_{\beta,in}$, is used as the input of a SOGI filter. The SOGI filter that receives the α component, there will be two outputs: v_α and qv_α. Similarly, in the case of the SOGI that filters the β component, the outputs are v_β and qv_β. The four variables can be used to calculate the positive sequence component in the $\alpha\beta$ reference frame as shown in Equation (2) (Rodriguez et al. 2006; de Castro and Vanfretti 2022). The output of this positive sequence calculation can then be used in a SRF loop designed to track one of the rotating axes. The DSOGI-based SRF-PLL (DSOGI-PLL) block diagram is shown in Figure 3.

$$\begin{cases} v_\alpha^+ = \frac{1}{2}\left(v_\alpha - qv_\beta\right) \\ v_\beta^+ = \frac{1}{2}\left(v_\beta + qv_\alpha\right) \end{cases} \tag{2}$$

The SRF-PLL (Kaura and Blasco 1997) uses a Park model and PI control to synchronize the rotating frame's and the $\alpha\beta$ positive sequence component's frequencies, such that the Park model's outputs are stationary. The SRF-PLL is described in Figure 4.

The compensator $C(s)$ is usually a PI controller with Kp and Ki that can be defined as

$$\begin{cases} K_p = & \frac{2\omega_n \xi}{\hat{X}} \\ K_i = & \frac{\omega_n^2}{\hat{X}}, \end{cases} \tag{3}$$

where $\hat{X}$ is the nominal peak value of the filtered wave, while ξ is the damping factor and ω_n is the cut-off frequency. These parameters can be tuned by the user and, for 60 hertz waves, $\xi = 0.7$ and $\omega_n = 100$ rad/s result in good dynamic performance (Rodriguez et al. 2006).

The implementation of DSOGI-PLL is carried out using block diagrams in Modelica, as shown in Figure 5. Note that the integrator located at the bottom right in the block diagram is the resettable integrator described in previous sections. This means that the output `delta` is always in the interval between 0 and 2π.

4 Wave-Phasor Interface Model

The integration between models created under different paradigms, that is, EMT and TS-like models, requires a component capable of adequately interfacing both representations. In this approach (Theodoro, Tomim, Lima, et al. 2017; Theodoro, Tomim, Barbosa, Lima, and Santiago 2018; Theodoro, Tomim, Barbosa, Lima, and Barros 2019), the TS-type power system with phasor-based models acts similarly to a variable voltage source on the interface bus. Transformation of a voltage phasor into a wave-like voltage source requires a phasor-to-wave transformation. In contrast, the EMT-type power system model will act as a variable current injection on the interface bus. This phasor-type current injection must be calculated from the waveforms obtained from the detailed EMT-type model, and thus a wave-to-phasor transformation is needed.

4.1 Voltage Phasor-to-Wave

This interface takes all necessary information from a phasor and translates it into waveform equations. The phasor information is carried out through OpenIPSL's `connector` with a Cartesian representation. The phasor $\tilde{V}$ is represented in OpenIPSL as

$$\tilde{V} = v_{re} + jv_{im}. \tag{4}$$

Hence, given the nominal voltage of the bus V_b and the base frequency of the system f_s, it is possible to calculate the wave information (amplitude A, frequency ω_s, and angle ϕ_v) as

$$\begin{cases} A = \frac{V_b\sqrt{2}}{\sqrt{3}}\left(\sqrt{v_{re}^2 + v_{im}^2}\right), \\ \omega_s = \omega_0 + \dfrac{d\phi_v}{dt}, \\ \phi_v = tg^{-1}\left(\dfrac{v_{im}}{v_{re}}\right), \end{cases} \tag{5}$$

where $\omega_0 = 2\pi f_s$. The calculated angle ϕ_v is assumed to be the phase from phase A's voltage, and the angles for

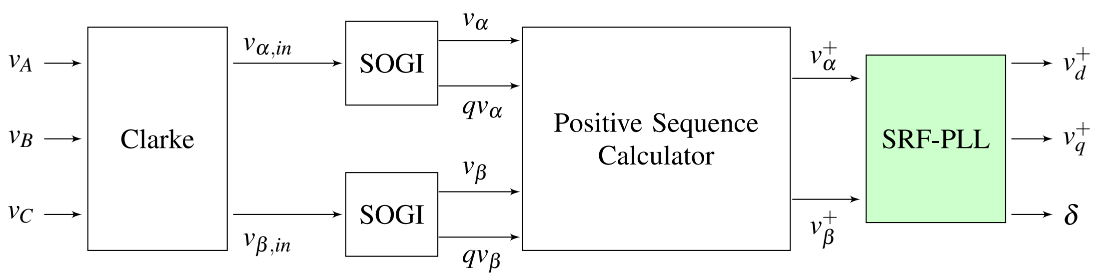

Figure 3. Diagram for the DSOGI-PLL.

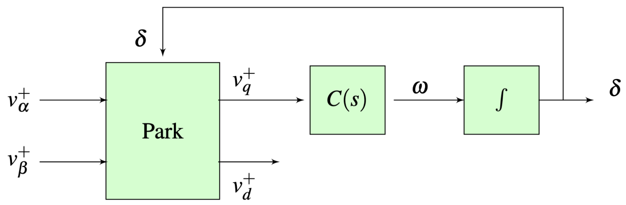

Figure 4. Diagram representing a PLL based on Synchronous Reference Frame (SRF).

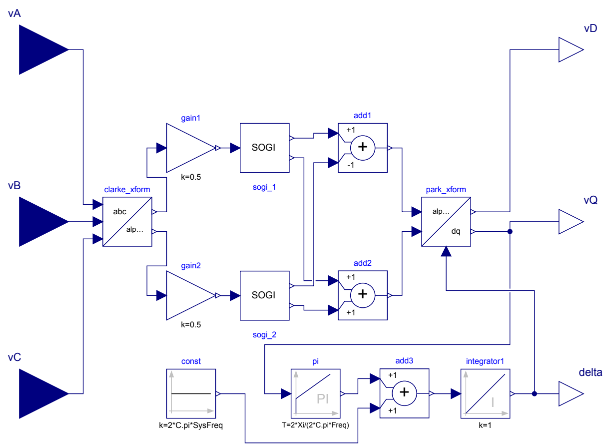

Figure 5. Modelica implementation of the DSOGI-PLL.

the other phase voltages are calculated displacing phase B and C of $\frac{2\pi}{3}$ radians. Hence, the three-phase waves that will be apparent for the EMT portion of the interface can be described as

$$\begin{cases} v_A = A\cos\left(\omega_s t + \phi_v\right), \\ v_B = A\cos\left(\omega_s t + \phi_v - \dfrac{2\pi}{3}\right), \\ v_C = A\cos\left(\omega_s t + \phi_v + \dfrac{2\pi}{3}\right). \end{cases} \quad (6)$$

4.2 Current Wave-to-Phasor

This part of the interface uses a filter to find the current phasor that will be shown at the phasor-domain portion of the interface. This can be done by using two PLLs: one DSOGI-PLL to be coupled with voltage waves described in Equation (6), and one modified PLL coupled with the currents that flow in the EMT portion of the interface. The DSOGI-PLL, coupled with voltage waves, is used to find the δ that varies with the appropriate speed ω_s. The second PLL can be considered as a modified DSOGI-PLL and it uses the current waves as inputs. The first modification is that the PLL can use a DSOGI filter as it is shown in Figure 3, but it could also use other filters to obtain i_α^+ and i_β^+. The second modification comes from the fact that the SRF-PLL part, as depicted in Figure 4 is reduced to a simple Park block that has inputs i_α^+ and i_β^+ as well as δ which is the output from the DSOGI-PLL coupled to the voltage waves. This procedure yields a current phasor

$$\tilde{I} = i_d^+ + j i_q^+. \quad (7)$$

where i_d^+ and i_q^+ are the outputs of the Park block of the modified PLL coupled to current waves. However, the angle δ is such that

$$\left(v_{re} + j v_{im}\right) e^{-j\delta} = v_d + j v_q = v_d. \quad (8)$$

Therefore, the current phasor $i_d^+ + j i_q^+$ produces an angle difference between the voltage and the current phasors. Consequently, to calculate the appropriate positive sequence current phasor value, it must be displaced from the voltage angle ϕ_v (see Equation (6)).

4.3 Modelica Implementation

In Modelica, the model contains one `connector` from the OpenIPSL and four from the MSL (three connectors for the voltage phases and one connector for the ground).

Listing 5. Connector instantiation.

```
// pins for wave signal:
Modelica.Electrical.Analog.Interfaces.
    PositivePin PinA;
Modelica.Electrical.Analog.Interfaces.
    PositivePin PinB;
Modelica.Electrical.Analog.Interfaces.
    PositivePin PinC;
```

```
Modelica.Electrical.Analog.Interfaces.
    NegativePin PinGrd;
// pin for phasor signal
OpenIPSL.Interfaces.PwPin PSPin;
```

Three system parameters are necessary for the calculations and they are entered as parameters for the interface as shown below.

Listing 6. System parameter declaration.

```
parameter Types.ApparentPower S_b = 100e6 "
    System base power";
parameter Types.Voltage V_b = 230e3 "Bus
    base voltage";
parameter Types.Frequency fn = 60 "System
    frequency";
```

The DSOGI-PLL from Figure 3 as well as a derivative block from MSL are instantiated in the model as presented below.

Listing 7. Instantiation of PLL and Derivative block.

```
OpenIWPI.Utilities.PhaseLockedLoop.Dsogi
    pll (SysFreq = fn, VLine = V_b);
Modelica.Blocks.Continuous.Derivative derA(
    T=0.1, y_start=0);
```

In addition, variables that are used in internal calculations are declared as protected. Note that the variables `type` are assigned attributes from those in `OpenIPSL.Types`.

Listing 8. Protected variables.

```
protected
    Types.AngularVelocity w
    // voltage variables
    Types.Voltage MagV;
    Types.Angle AngV;
    // current variables
    Types.PerUnit ir;
    Types.PerUnit ii;
    Types.PerUnit MagI;
    Types.Angle AngI
```

In the **`equation`** environment, the voltage phasor-to-wave transformation can be implemented in Modelica as

Listing 9. Voltage phasor-to-wave implementation

```
equation
    // voltage information from phasors
    MagV = V_b*sqrt(2/3)*sqrt(PSPin.vr^2+
        PSPin.vi^2);
    AngV = Modelica.Math.atan2(PSPin.vi,
        PSPin.vr);
    derA.u = AngV;
    w = 2*C.pi*fn + derA.y;
    // voltage waves
    PinA.v-PinGrd.v = MagV*Modelica.Math.
        sin(w*time+AngV);
    PinB.v-PinGrd.v = MagV*Modelica.Math.
        sin(w*time+AngV-2*C.pi/3);
    PinC.v-PinGrd.v = MagV*Modelica.Math.
        sin(w*time+AngV+2*C.pi/3);
```

The implementation of the current equations is described using Listing 10. The Clarke transformation result, `ialpha` and `ibeta`, is instantiated as `RealOutput`, while the inputs for the Park transformation, `iapos` and `ibpos`, are instantiated as `RealInput`. This allows the user to create other hybrid buses by using other filters and methods to extract the positive sequence of the $\alpha\beta$ frame.

Listing 10. Current wave-to-phasor implementation

```
equation
    // voltage PLL synchronization
    pll.vA = PinA.v-PinGrd.v;
    pll.vB = PinB.v-PinGrd.v;
    pll.vC = PinC.v-PinGrd.v;
    // clarke transformation
    ialpha = (2/3)*(PinA.i - (1/2)*PinB.i -
        (1/2)*PinC.i);
    ibeta = (2/3)*((sqrt(3)/2)*PinB.i - (
        sqrt(3)/2)*PinC.i);
    // Real and imaginary parts of the
        current phasor using Park transf.
    ir = -sqrt(3/2)*(V_b/S_b)*(iapos*cos(
        pll.delta)+ibpos*sin(pll.delta));
    ii = sqrt(3/2)*(V_b/S_b)*(iapos*sin(pll
        .delta)-ibpos*cos(pll.delta));
    // current magnitude and angle
    MagI = sqrt(ii^2+ir^2);
    AngIV = Modelica.Math.atan2(ii,ir);
    // grounding
    PinGrd.i = - (PinA.i+PinB.i+PinC.i);
    // current phasor waves
    PSPin.ir = MagI*Modelica.Math.cos(AngIV
        +AngV);
    PSPin.ii = MagI*Modelica.Math.sin(AngIV
        +AngV);
```

4.4 Hybrid Bus

Finally, the hybrid bus model is assembled using the double SOGI strategy, followed by the use of Equation (2), similar to what is used in the DSOGI-PLL (Theodoro, Tomim, Lima, et al. 2017; Theodoro, Tomim, Barbosa, Lima, and Santiago 2018; Theodoro, Tomim, Barbosa, Lima, and Barros 2019). The hybrid bus model is implemented in the library as shown in Figure 6. Note that the `pwPin` on the left hand side is that of OpenIPSL; while those on the right hand side (`pin_A, pin_B, pin_C`) and the bottom (`pin_grd`) are those from the MSL.

5 Simulation Examples

The OpenIWPI library has two simple examples that are used to show the appropriate behavior of the wave-phasor interface.

5.1 Simple Circuit

This first example shows a comparison between a power system completely represented in phasor-domain models and a circuit that is partially represented in phasor-domain and partially in EMT models. Figure 7 shows the two circuits that are present in the example: the upper circuit is

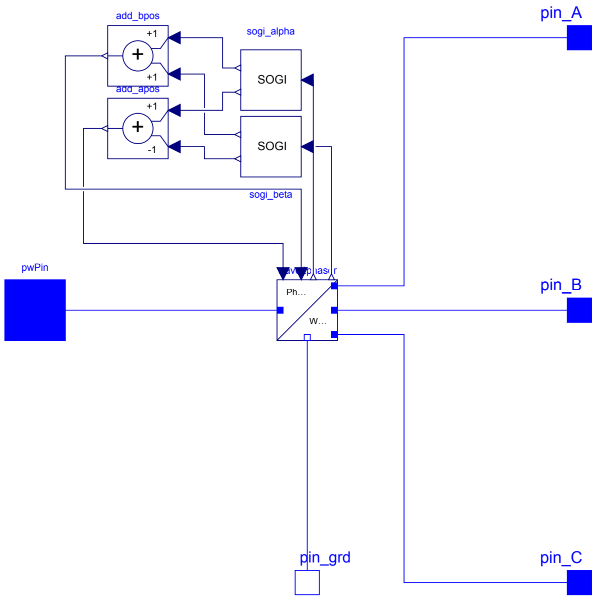

Figure 6. Modelica implementation of the hybrid bus.

the positive sequence (phasor-based model), while the bottom circuit is the hybrid TS-EMT model.

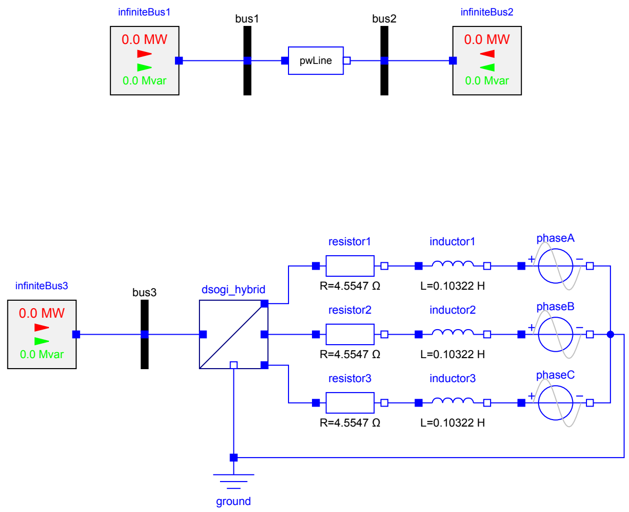

Figure 7. Simple circuit example in Modelica.

The goal of this example is to demonstrate the filtering of the three-phase current wave, showing that both circuits will yield the same result. The example is set to start with zero current on all inductors, which challenges the filter to reach the steady-state response and allows one to observe its behavior. The results presented in Figure 8 show that even under that condition, the filter is able to track the correct real and imaginary values of the current phasor.

5.2 Converter Circuit

The second example describes an application of the hybrid interface, in which a power-electronics-based device

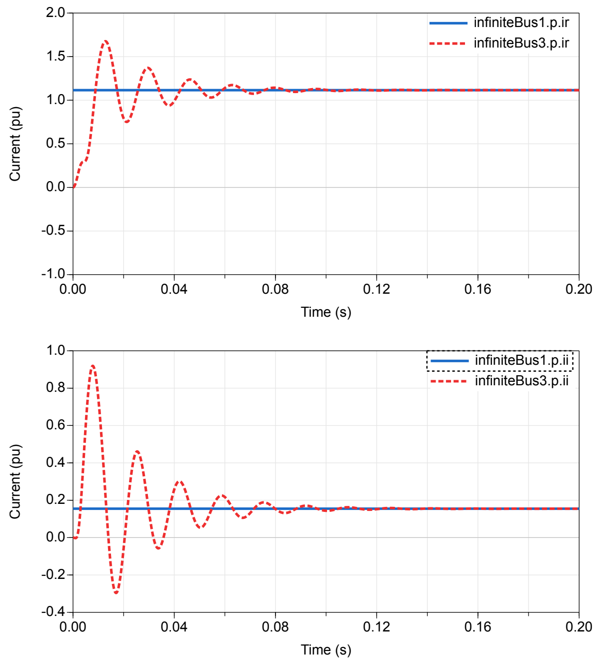

Figure 8. Current filtering simulation result for real (upper plot) and imaginary (bottom plot) components of the positive sequence current phasor.

is represented in detail using its EMT model, while the rest of the grid is built using the phasor-domain (TS) models. The Static Compensator (STATCOM) is an inverter-based resource that can be used to provide reactive power support to the power grid (Hingorani and Gyugyi 1999). The example circuit is presented in Figure 9 showing the interconnection of STATCOM to bus `B2`.

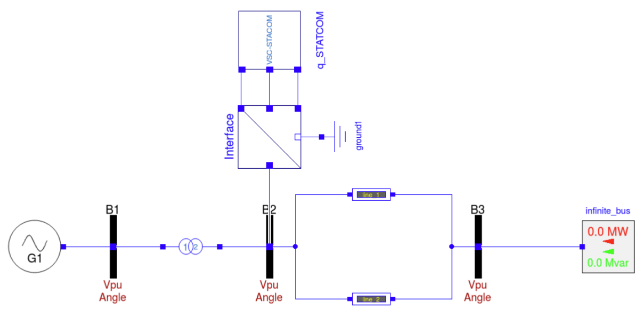

Figure 9. Diagram for converter circuit example.

This test system was developed to show the operation of a STATCOM and its controllers modeled in detail using the EMT approach, while the rest of the system is modeled in phasor-domain for a TS study. The STATCOM controllers used in this example are explained in more detail in (de Castro and Vanfretti 2022). To assess this behavior, the reference to the STATCOM's reactive power injection changes from 0 to 90 Mvar at $t = 1$ second, as shown in Figure 10. The three-phase modulation indices for each phase are shown in Figure 11. Note that the modulation indices change almost instantaneously at $t = 1.0$

sec., showing the appropriate control action for the desired reactive power injection.

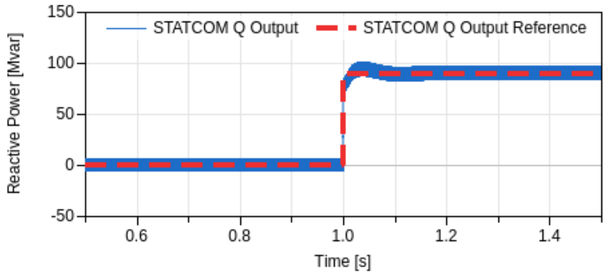

Figure 10. Reference value and reactive power injection by STATCOM.

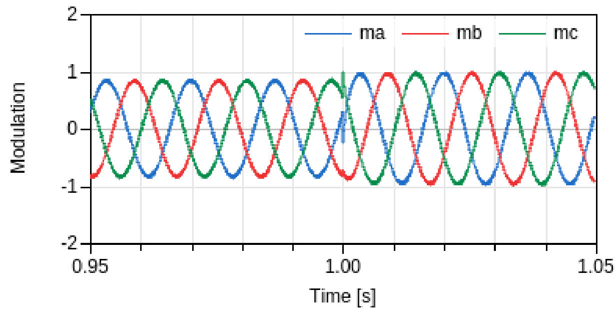

Figure 11. Three-phase modulation indices for the STATCOM's controller.

The voltage behavior on the STATCOM's DC bus is presented in Figure 12. The DC voltage is controlled throughout the simulation at its reference value of 60 kV. Note how the voltage suffers a disturbance exactly at $t = 1$ sec. Another disturbance in the voltage occurs at $t = 0.05$ sec., which is the time instant when the STATCOM connects to the grid. The bus `B2` voltage is depicted in Figure 13. Note that the bus voltage increases its magnitude as the reactive support offered by the STATCOM increases.

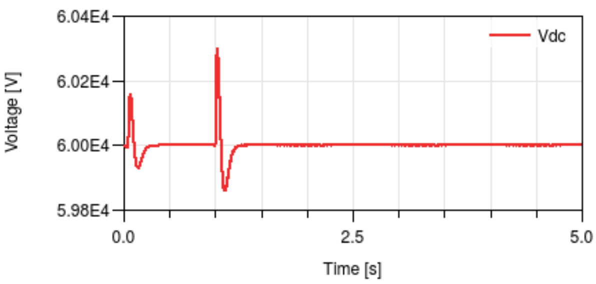

Figure 12. STATCOM DC bus voltage during simulation.

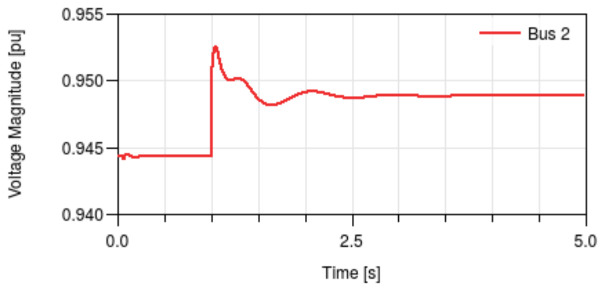

Figure 13. Bus `B2` voltage level during simulation.

6 Conclusion

This work presented the Open-Instance Wave-Phasor Interface, or OpenIWPI, a Modelica library designed to allow coalesced simulation of hybrid EMT-TS models, using a wave-phasor interface. The interface is built using SOGI-based filters originally proposed in (Theodoro, Tomim, Barbosa, Lima, and Barros 2019). Although the library is small, it provides all the essential components to successfully link EMT and TS models, providing auxiliary models and reference frame transformations that are used by the interface. The interface bus presented in this paper uses connectors from the OpenIPSL, which has power system models for phasor-domain simulations, and it also uses connectors from the Electrical package within the MSL, which allows part of the power system to be modeled for EMT.

Furthermore, this paper describes two example models that are part of the OpenIWPI library and presents their simulation results. The results should provide the user with details on how the library is expected to work and how to assemble larger models using the interface. The wave-phasor interface could be used as a magnifying glass for parts of the system to be modeled in detail (EMT), while other parts of the power grid can continue to have their phasor-based positive-sequence equivalent model (TS). This can be particularly useful when there is not enough information to model the whole power grid in EMT, or when one is interested in a particular plant's response when faced with system-wide dynamics (Boersma et al. 2021) that can only be captured with large-scale simulation models that are typically only represented for TS.

The OpenIWPI library has been tested with Dymola 2025x and requires OpenIPSL release v.3.1.0, which can be found online at: `http://openipsl.org/`. The OpenIWPI library is released as open source, and is available here: `https://github.com/ALSETLab/OpenIWPI`.

References

Adapa, Rambabu and John Reeve (1988). "A new approach to dynamic analysis of AC networks incorporating detailed modeling of DC systems. II. Application to interaction of DC and weak AC systems". In: *IEEE Trans. Power Del.* 3.4, pp. 2012–2019.

Baur, Marcus, Martin Otter, and Bernhard Thiele (2009-09). "Modelica Libraries for Linear Control Systems". In: *7th International Modelica Conference*. 43. Como, Italy:

DOI 10.3384/ecp218223

Linköping Electronic Conference Proceedings, pp. 593–602. DOI: 10.3384/ecp09430068. URL: https://ep.liu.se/en/conference-article.aspx?series=ecp&issue=43&Article_No=68.

Boersma, Sjoerd et al. (2021). "Probing signal design for enhanced damping estimation in power networks". In: *International Journal of Electrical Power & Energy Systems* 129, p. 106640. ISSN: 0142-0615. DOI: 10.1016/j.ijepes.2020.106640. URL: https://www.sciencedirect.com/science/article/pii/S0142061520341855.

Chang, Hao and Luigi Vanfretti (2024). "Power Hardware-in-the-Loop Smart Inverter Testing with Distributed Energy Resource Management Systems". In: *Electronics* 13.10. ISSN: 2079-9292. DOI: 10.3390/electronics13101866.

Chow, Joe H and Juan J Sanchez-Gasca (2020). *Power system modeling, computation, and control*. John Wiley & Sons.

de Castro, Marcelo and Luigi Vanfretti (2022). "Multi Time-Scale Modeling of a STATCOM and Power Grid for Stability Studies using Modelica". In: *2022 Open Source Model. Simul. of Energy Syst. (OSMSES)*, pp. 1–7.

de Castro, Marcelo, Dietmar Winkler, et al. (2023). "Version [OpenIPSL 2.0.0] - [iTesla Power Systems Library (iPSL): A Modelica library for phasor time-domain simulations]"". In: *SoftwareX* 21, pp. 1–3.

Dennetiere, Sebastien et al. (2019). "Supporting Energy Transition in Transmission Systems: An Operator's Experience Using Electromagnetic Transient Simulation". In: *IEEE Power and Energy Magazine* 17.3, pp. 48–60. DOI: 10.1109/MPE.2019.2897179.

Dinavahi, V. et al. (2009). "Interfacing techniques for simulation tools". In: *2009 IEEE Power & Energy Society General Meeting*, pp. 1–6. DOI: 10.1109/PES.2009.5275553.

Dommel, Hermann W (1986). *Electromagnetic Transients Program Reference Manual: EMTP Theory Book*. Portland, OR, USA: Bonneville Power Administration.

Fachini, Fernando et al. (2025-05). "Customized open source renewable energy models validated through PHIL lab experiments". In: *Renewable Energy* 244, p. 122627. ISSN: 0960-1481. DOI: 10.1016/j.renene.2025.122627. (Visited on 2025-04-09).

Gibert, Pierre-Marie et al. (2018). "Use of the sinusoidal predictor method within a fully separated modeler/solver framework for fast and flexible EMT simulations". In: *SIMULTECH*, pp. 34–43.

Heffernan, MD et al. (1981). "Computation of ac-dc system disturbances-Part I. Interactive coordination of generator and convertor transient models". In: *IEEE Trans. Power App. Syst.* PAS-100.11, pp. 4341–4348.

Hingorani, Narain G and Laszlo Gyugyi (1999). *Understanding FACTS: Concepts and Technology of Flexible AC Transmission Systems*. Piscataway, NJ, USA: IEEE Press.

Kaura, Vikram and Vladimir Blasco (1997). "Operation of a phase locked loop system under distorted utility conditions". In: *IEEE Trans. Ind. Appl.* 33.1, pp. 58–63.

Kundur, Prabha S, Neal J Balu, and Mark G Lauby (1994). *Power System Dynamics and Stability*. 1st ed. New York, NY, USA: McGraw Hill Inc.

Mahseredjian, Jean, Venkata Dinavahi, and Juan A. Martinez (2009-07). "Simulation Tools for Electromagnetic Transients in Power Systems: Overview and Challenges". In: *IEEE Transactions on Power Delivery* 24.3, pp. 1657–1669. ISSN: 1937-4208. DOI: 10.1109/TPWRD.2008.2008480. URL: https://ieeexplore.ieee.org/document/4907257 (visited on 2025-05-01).

Martins, Nelson et al. (2000). "A small-signal stability program incorporating advanced graphical user interface". In: *Proceedings of the VII SEPOPE*.

Masoom, Alireza, Jean Mahseredjian, et al. (2021). "MSEMT: An advanced Modelica library for power system electromagnetic transient studies". In: *IEEE Trans. Power Del.* 37.4, pp. 2453–2463.

Masoom, Alireza, Tarek Ould-Bachir, et al. (2020). "Simulation of electromagnetic transients with Modelica, accuracy and performance assessment for transmission line models". In: *Electr. Power Syst. Res.* 189. Art. no. 106799.

Modelica Association (2025). *Modelica language*. https://modelica.org/language/ (accessed Apr. 28, 2025).

Nikolaev, Nikolay et al. (2020). "PSS/E Based Power System Stabilizer Tuning Tool". In: *2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA)*. IEEE, pp. 1–6.

Petzold, Linda R (1982). *Description of DASSL: a differential/algebraic system solver*. Tech. rep. Sandia Nat. Labs., Livermore, CA, USA.

Ramirez-Gonzalez, M. et al. (2024). "Real-Time Simulation of the Initial Dynamic Model of CE in ePHASORsim: Issues and Troubleshooting". In: *2024 IEEE 65th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pp. 1–6. DOI: 10.1109/RTUCON62997.2024.10830830.

Rodriguez, Pedro et al. (2006). "New positive-sequence voltage detector for grid synchronization of power converters under faulty grid conditions". In: *2006 37th IEEE Power Electron. Specialists Conf.* Pp. 1–7.

Schweiger, Gerald et al. (2019). "An empirical survey on co-simulation: Promising standards, challenges and research needs". In: *Simulation modelling practice and theory* 95, pp. 148–163.

Thapa, Jitendra et al. (2023). "Plug-and-Play Regional Models for Real-Time Electromagnetic Transient Simulations of Large-Scale Power Grids: A Case Study of New York State Power Grid". In: *IEEE Access* 11, pp. 87600–87614. DOI: 10.1109/ACCESS.2023.3305394.

Theodoro, Thainan, Marcelo Tomim, Pedro Barbosa, Antonio Lima, and Maria de Barros (2019). "A flexible co-simulation framework for penetration studies of power electronics based renewable sources: A new algorithm for phasor extraction". In: *Int. J. Elect. Power Energy Syst.* 113, pp. 419–435.

Theodoro, Thainan, Marcelo Tomim, Pedro Barbosa, Antonio Lima, and Juan Santiago (2018). "A hybrid simulation tool for distributed generation integration studies". In: *2018 Power Syst. Comput. Conf. (PSCC)*, pp. 1–8.

Theodoro, Thainan, Marcelo Tomim, Antonio de Lima, et al. (2017). "A hybrid simulation tool for penetration studies of distributed generation in smartgrids". In: *2017 Brazilian Power Electron. Conf. (COBEP)*, pp. 1–7.

Watson, Neville and Jos Arrillaga (2003). *Power Systems Electromagnetic Transients Simulation*. Vol. 39. London, UK: IET.

Winkler, Dietmar (2017). "Electrical power system modelling in modelica–comparing open-source library options". In: *Proc. 58th Conf. Simul. Model. (SIMS 58)*, pp. 263–270.

Yazdani, Amirnaser and Reza Iravani (2010). *Voltage-Sourced Converters in Power Systems: Modeling, Control, and Applications*. Hoboken, NJ, USA: John Wiley & Sons.