

# **Practical Implementation of a Transmission– Distribution Co–Simulation Framework Using FMU and OpenDSS**

Emre Can Kaya

## **Introduction**

The increasing integration of distributed energy resources (DERs) into electrical grids has significantly altered the operational characteristics of modern power systems.

Understanding the dynamic interactions between different parts of these systems has therefore become increasingly important. In particular, transmission systems (TS) and distribution systems (DS), which have traditionally been studied separately, are now more tightly coupled due to high levels of DERs, power-electronic interfaces, and advanced control strategies. Analyzing these interactions requires simulation approaches capable of representing heterogeneous subsystems while preserving the specific modeling assumptions and numerical methods best suited for each domain.

Co-simulation has emerged as a practical approach to address this need. In a co-simulation framework, different subsystems are simulated using separate tools or models that exchange information during runtime. This allows each subsystem to be modeled using specialized software while still enabling the study of system-level interactions. Such an approach is particularly useful when combining models with different time scales, levels of detail, or physical domains.

However, co-simulation of electrical systems presents significant challenges. Electrical networks often involve tightly coupled variables such as voltages and currents that must satisfy instantaneous physical constraints. When a network is partitioned into subsystems simulated by different tools, the boundary between subsystems must be placed at a specific physical location. At such interfaces, quantities like voltage and current must remain consistent across the coupled simulations. As a result, the variables computed in one simulator depend directly on those produced by the other, creating algebraic loops across subsystem boundaries. Resolving these loops typically requires iterative coordination between simulators at each time step, which can significantly increase computational complexity and reduce robustness.

One possible approach to overcome this difficulty is to separate the subsystems by introducing a transmission line between them. In this case, the subsystem boundaries no longer coincide at the same physical point; instead, the two simulators are connected through the transmission line, which introduces a propagation delay. This delay breaks

the instantaneous dependency between voltages and currents at the interface, allowing each simulator to advance independently while exchanging delayed signals. The transmission line may represent a real physical element present in the system, or it may be introduced artificially to facilitate numerical coupling.

The work referenced in [1] proposes such an approach by introducing a fictitious Bergeron transmission line between the transmission and distribution models. The artificial delay created by the transmission line removes the algebraic loop between the subsystems while maintaining a physically meaningful representation of power exchange. However, an important question is under what conditions such artificial delays remain acceptable and how they influence the accuracy and stability of the resulting co-simulation.

Motivated by these considerations, this paper evaluates the proposed transmission-distribution co-simulation approach in [1] and demonstrates its practical implementation. The framework interfaces a dynamic transmission-system Functional Mock-up Unit (FMU) with a quasi-static distribution solver using a fictitious Bergeron transmission line. Our goal is to reproduce the method in [1], investigate practical implementation aspects, and assess how the approach can be used for studying interconnected electrical networks such as those found in ship power systems or other complex grid architectures. In addition to validating the original concept, we present a reproducible implementation, a GUI-based parameter exploration tool, and a discussion of modeling assumptions that influence the behavior of the co-simulation. The implementation is openly available at: [https://github.com/Novia-RDI-Seafaring/fmu\\_dss\\_cosim/](https://github.com/Novia-RDI-Seafaring/fmu_dss_cosim/).

## **Dynamic behavior in co-simulation**

In the proposed co-simulation framework, dynamic behavior is a result of the time-stepped interaction between a dynamic FMU model and the quasi-static OpenDSS network solver, both represented in the phasor domain. The FMU, generated from Modelica, contains differential-algebraic models of the transmission system (e.g., generators and controls) and advances its internal states using its embedded numerical solver at each communication step. In contrast, OpenDSS solves a steady-state power flow of the distribution system and provides the resulting currents and voltages at the interface bus. During each time step, the master algorithm exchanges these interface quantities between the FMU and OpenDSS: the FMU provides updated voltages or currents based on its dynamic states, OpenDSS recomputes the network solution, and the resulting electrical quantities are fed back to the FMU to influence the next integration step. To enable independent solution of the subsystems, the interface is

modeled as a fictitious transmission line based on the Bergeron model, which introduces a small propagation delay through historical terms of voltages and currents. This delay decouples the subsystems numerically, allowing each simulator to advance independently while the repeated exchange of phasor quantities at discrete time steps causes the overall transmission-distribution system to evolve dynamically.

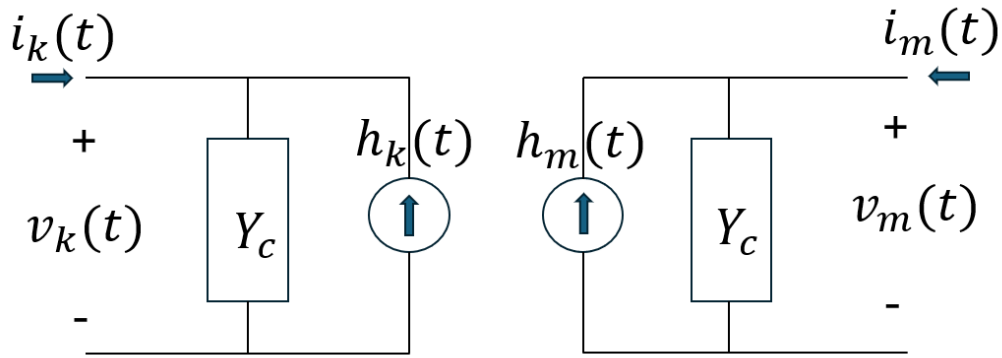


Figure 1: Bergeron Model of an ideal transmission line. (adapted from [1])

Bergeron Model (Fig. 1) can be described by the following set of equations, where  $h_k(t)$  and  $h_m(t)$  are the historical currents at both sides and depend on the past values of the voltages and currents.  $\tau$  is the fictitious propagation delay which is equal to the fixed time step used in the simulations.

$$i_k(t) = Y_c v_k(t) - h_k(t)$$

$$i_m(t) = Y_c v_m(t) - h_m(t)$$

$$h_k(t) = Y_c v_m(t - \tau) + i_m(t - \tau)$$

$$h_m(t) = Y_c v_k(t - \tau) + i_k(t - \tau)$$

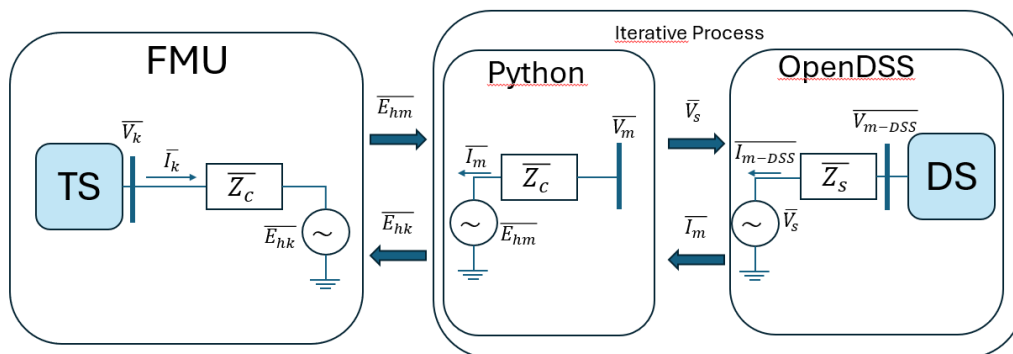


Figure 2: Co-simulation interfacing an FMU with OpenDSS. (adapted from the original paper [1])

## Ideal transmission line and characteristic impedance selection

Bergeron model for an ideal transmission line is depicted in Fig. 1. The  $k$  terminal connects to TS and the  $m$  terminal connects to DS. In other words, the point of common coupling between the two sides is modeled as an ideal transmission line. When the co-simulation is stable, the voltages at the  $k$  and  $m$  terminals remain nearly identical, indicating consistent coupling between the subsystems. The circuit model of the interface and the direction of information flow during co-simulation are depicted in Fig. 2 (Fig. 8 in the original paper). Note that having  $\overline{E}_{hk}$ , an ideal voltage source and a serial impedance  $\overline{Z}_c = Z_c e^{j\theta_c}$  is equivalent to having an ideal current source  $h_k(t)$  with an admittance  $\overline{Y}_c = 1/\overline{Z}_c$  connected in parallel to the current source where  $h_k(t) = \overline{E}_{hk}/\overline{Z}_c$ .

A fictitious transmission line is introduced to avoid numerical and modeling issues that arise when directly enforcing  $V_{PCC,TS} = V_{PCC,DS}$ . Imposing this equality directly creates an algebraic loop, since both solvers attempt to enforce boundary conditions at the same point simultaneously. As a result, the coupled system may experience divergence or numerical oscillations during simulation. The fictitious transmission line alleviates this problem by providing a structured interface between the two sides, allowing the solvers to exchange information without directly imposing conflicting constraints. Similar ideas are used in EMT-RMS hybrid simulation and wave-relaxation methods. An ideal transmission line normally introduces both a propagation delay and a characteristic impedance  $\overline{Z}_c$ ; however, because the line used here is fictitious, the propagation delay is simply chosen equal to the simulation communication step. As a result, the main design parameter of the interface becomes the characteristic impedance  $\overline{Z}_c$ . This parameter determines the stiffness of the coupling, numerical damping, convergence speed, and stability margins. If  $\overline{Z}_c$  is too small, the coupling becomes stiff, currents may grow excessively, and the simulation may diverge. If  $\overline{Z}_c$  is too large, the coupling becomes weak and the voltages of the transmission and distribution subsystems may drift apart. In practice,  $\overline{Z}_c$  should be chosen close to the Thevenin impedance seen from the interface (typically the transmission-system Thevenin impedance) so that the interface impedance is consistent with the electrical characteristics of the coupled subsystems.

For studying the effects of  $\overline{Z}_c$  selection, following the methodology in the paper, we have created a GUI where the user can set the  $\overline{Z}_c$ , which is a complex-valued parameter, where  $Z_c$  is the amplitude and  $\theta_c$  is the angle, by simply clicking a point in a 2D polar plot. A screenshot of this GUI is shown on Fig. 3. On the polar plot, the colors indicate  $f(Z_c, \theta_c)$ , *i. e.*, the magnitude of the eigenvalues of the overall system for a given  $\overline{Z}_c$ , which is a measure of the stability of co-simulation where  $\overline{Z}_k$  and  $\overline{Z}_m$  are the TS and DS Thevenin equivalent impedances as seen from k and m terminals of the ideal transmission line, respectively. When the polar plot is clicked and  $\overline{Z}_c$  is set (the red dot), the tool runs the simulation of a TS and a DS which are both modeled as FMUs, and plots the Point of Common Coupling (PCC) voltage at both TS and DS sides with respect to time.

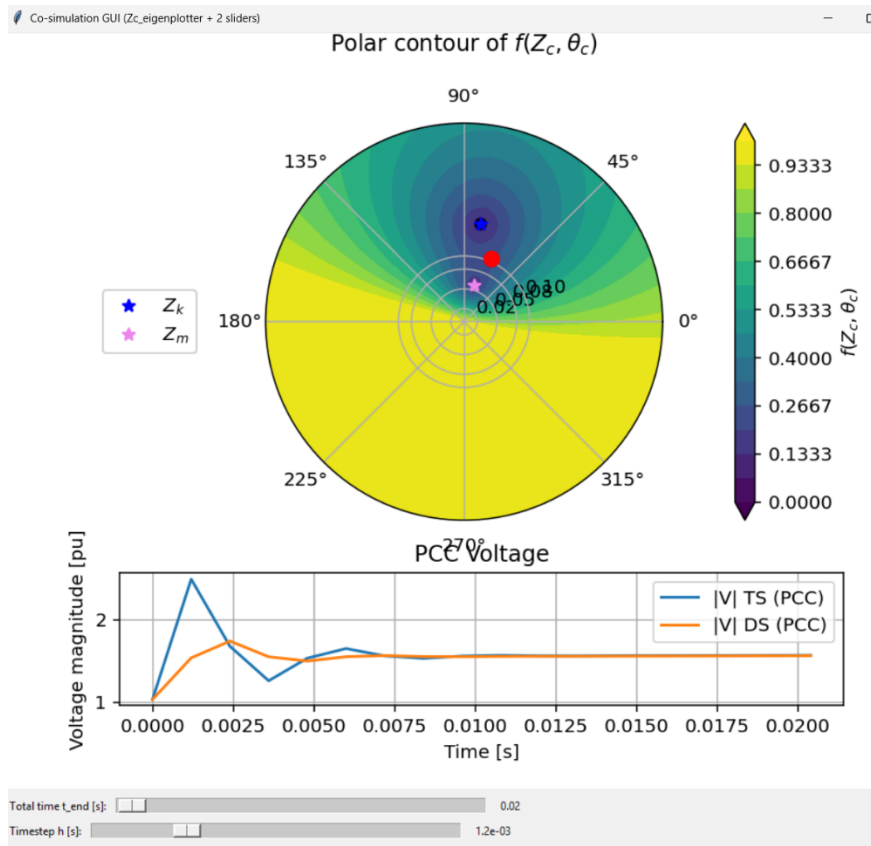


Figure 3: The GUI that we created for studying the system behavior for different  $Z_c$ .

## Some practical points to consider

When implementing the main co-simulation algorithm in the paper (see Fig. 4 in the original paper), one practical issue that caught our attention is the fact that one needs to make certain voltage-current conversions which are not explicitly stated in the paper. We find it helpful to mention this. Specifically, the phasor term  $\overline{E}_{hk}$  denotes voltage, hence it should be converted to a current value by dividing by the characteristic

impedance, to obtain the current value  $h_{in}$ . Similarly,  $\overline{E_{hm}}$  is obtained by multiplying the  $h_{out}$  with the characteristic impedance.

Another practical point that one should note is that, when solving the power flow with OpenDSS, one should disable the internal controls of OpenDSS. Regulators, capacitors, and controllers introduce internal iteration loops. These loops conflict with the macro-step co-simulation. This may lead to solver failures or warnings. Hence, a stable implementation requires disabling controls for example by using the `SolveNoControl()` command. When coupling the two electrical systems together, it is essential to make sure that the Point of Common Coupling has the same base voltage level on both sides. In our implementation, we have ensured that both sides have 400 V base voltage.

## **System under study**

Before describing the implementation details of the co-simulation interface and the FMU generation process, it is useful to first describe the electrical systems that are being coupled. The overall system considered in this work consists of a transmission network (TS) dynamically simulated using Modelica and a distribution network (DS) solved using OpenDSS. These two subsystems are connected at a point of common coupling (PCC) through the fictitious Bergeron transmission line described earlier.

The transmission side is based on the classical two-area power system example from Kundur's *Power System Stability and Control*. In the OpenIPSL library this system is implemented as the model `Two_Areas_PSSE_AVR`, which corresponds to the 11-bus transmission network used in the reference paper. The model contains four synchronous generators with automatic voltage regulators and represents typical electromechanical dynamics used in transient stability studies. During simulation, a three-phase disturbance is applied at one of the transmission buses (Bus 8), producing a temporary voltage dip followed by recovery. This disturbance is used to excite the system dynamics and observe how the coupled transmission–distribution system responds.

The distribution side is modeled using the IEEE 34-bus test feeder, which is included as an example case in the OpenDSS library. This feeder represents a realistic radial distribution network often used to evaluate distribution-level algorithms under unbalanced and high-impedance conditions. In the co-simulation setup, the source bus of the distribution system is renamed PCC so that its voltage can be externally controlled by the co-simulation algorithm.

The two subsystems are therefore connected as follows:

Transmission system (OpenModelica / FMU)

- fictitious Bergeron transmission line
- Distribution system (OpenDSS)

At every simulation step, the transmission FMU provides the PCC voltage and interface currents, while the distribution solver computes the resulting power-flow solution based on these boundary conditions. This structure allows the dynamic behavior of the transmission network and the quasi-static response of the distribution feeder to be studied simultaneously.

## Transmission Side FMU

We have created the FMU for the transmission side, using the built-in FMU exporter of OpenModelica. We have used the `Two_Areas_PSSE_AVR` model (see Fig. 2) inside the third-party public OpenModelica library called `OpenIPSL`. `OpenIPSL` is a library of power system component models that can be used for power system dynamic analysis, such as phasor time-domain simulations. `Two_Areas_PSSE_AVR` implements the Two-Area System example from "Power System Stability and Control", Example 12.6, page 813 by Prabha Kundur. This corresponds to the 11-bus transmission side system in [1]. During simulation, a fault happens at bus 8 of the transmission side. This will cause a sudden crash and recovery of the voltages.

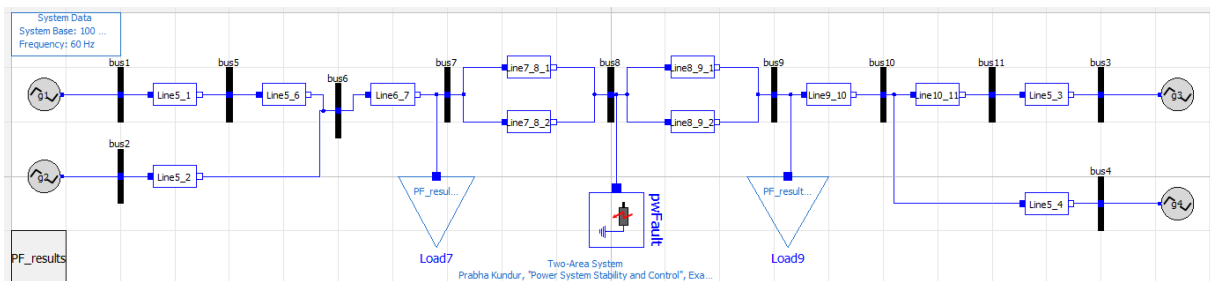


Figure 4: OpenModelica visualization of the `Two_Areas_PSSE_AVR` from `OpenIPSL`.

Before exporting the model as an FMU, we write a wrapper where we define the inputs and outputs of the FMU. The properties of the FMU are shown on Fig. 5.

Here are the description of the input and output variables:

FMU Inputs:

`Sds_im`, `Sds_re`: Imaginary and real parts of the `Sds`, the complex power demand of the distribution system as seen from the transmission-system FMU. These pins are used during the co-simulation initialization routine described in the paper.

hin\_im, hin\_re: Imaginary and real parts of  $H_{in}$ , the historical current at the transmission side ( $h_k(t)$  in the Bergeron model).

FMU Outputs:

hout\_im, hout\_re: Imaginary and real parts of  $H_{out}$ , the historical current at the transmission side ( $h_m(t)$  in the Bergeron model).

Im\_im, Im\_re: Imaginary and real parts of the current at the m side of the interface.

Vpcc\_im, Vpcc\_re: Imaginary and real parts of the voltage at the point of common coupling at the transmission side.

**Model Info**

FMI Version	2.0
FMI Type	Co-Simulation, Model Exchange
Model Name	Two_Areas_TS_FMU
Platforms	c-code, win64
Continuous States	38
Event Indicators	61
Variables	1538
Generation Date	2025-11-28T18:24:44Z
Generation Tool	OpenModelica Compiler OpenModelica v1.22.3 (64-bit)
Description	TS FMU: Sds + h_in inputs, Vpcc + Im + h_out outputs (bus7 is PCC)

**Simulation**

Solver: Variable-step

Step Size: 1e-3

Relative Tolerance: 1e-06

Output Interval: 0.002

Max. Samples: 500

Input: No input file selected

Apply default start values

Log FMI calls

Debug Logging

**Ports**

Sds_im	Im_im
Sds_re	Im_re
hin_im	Vpcc_im
hin_re	Vpcc_re
	hout_im
	hout_re

Figure 5: FMU description for the Transmission Side FMU model as displayed on FMPy GUI.

## Distribution side

Let us next focus on the Distribution Side (DS), which is modeled in the OpenDSS environment. For this purpose, we have used the "34Bus/ieee34Mod2.dss" example found in the IEEE Test Cases folder inside the OpenDSS library. IEEE 34 Bus Model is a realistic benchmark distribution feeder used to test and validate distribution-level power system algorithms under unbalanced and high-impedance conditions.

We interface this model by naming the source bus in the circuit definition in the dss code as PCC as follows:

```
New Circuit.ieee34-2 Bus1=PCC BasekV=400 pu=1.0 angle=30 mvasc3=200000
```

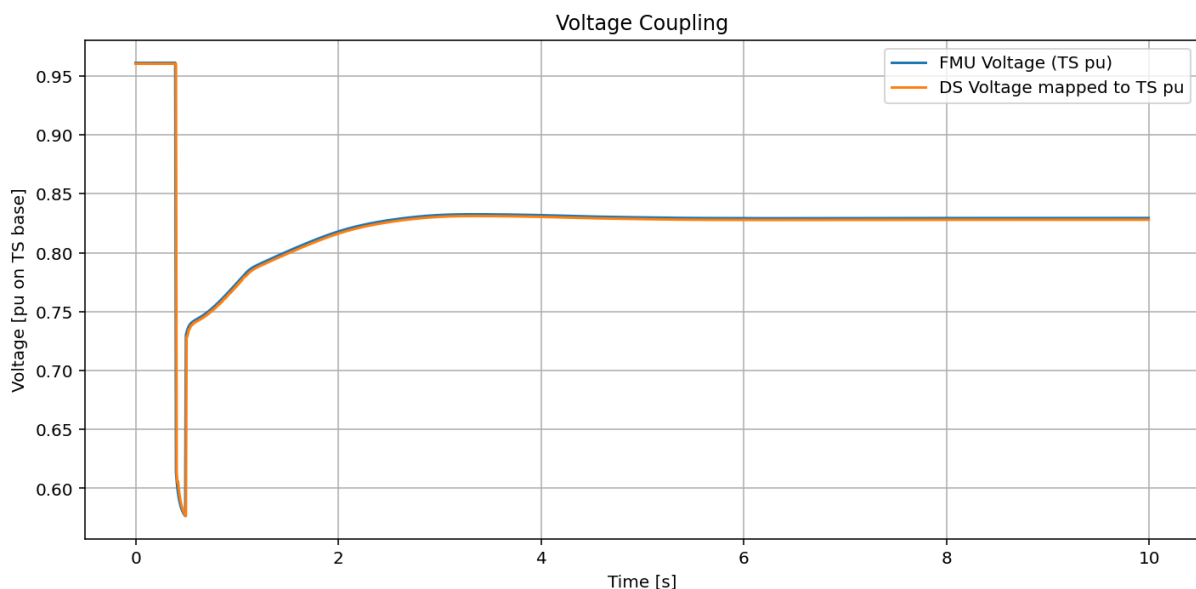
In the Python code that orchestrates the co-simulation, we set the PCC voltage at the DS by running the following comment:

```
dss.Text.Command(f"Edit Vsource.Source pu={mag_ds} angle={ang_deg}"),
```

where dss is the opendssdirect module which lets us run dss commands from python.

## Results and Conclusion

As a result of the co-simulation, we have obtained the PCC voltage at both sides very similar to each other, as seen from Fig. 6, validating the method. Here we have set  $\bar{Z}_c =$



$\bar{Z}_{\text{Thevenin, TS}}$

Figure 6: PCC voltage readings at both sides. Between times 0.4 sec and 0.5 sec, a fault is applied at Bus 8 in the transmission side which results in a temporary voltage drop followed by recovery.

In this work, we have successfully implemented a transmission–distribution co-simulation framework by coupling a dynamic transmission-system FMU with a quasi-static distribution model in OpenDSS through a fictitious Bergeron transmission line. The close agreement of PCC voltage responses on both sides confirms the accuracy and numerical stability of the approach, even under dynamic disturbances such as faults.

Beyond reproducing the original method, this study highlights practical implementation insights, including interface variable handling, solver configuration, and the critical role of characteristic impedance selection. The developed GUI further improves accessibility by allowing intuitive exploration of system behavior.

More broadly, this work demonstrates how complex, interconnected power systems can be analyzed using modular co-simulation techniques. Transmission and distribution systems often require different analysis and simulation techniques and this method brings them together by enabling co-simulation of transmission and distribution systems while complying with the requirements of both sides. As modern energy systems evolve with increasing penetration of distributed energy resources, such approaches become essential for ensuring stability, flexibility, and scalability.

While demonstrated on a benchmark system, the framework is applicable to a wide range of real-world scenarios, including smart grids, microgrids, and electrified transportation systems.



## **Acknowledgements**

This work was conducted within the scope of the Virtual Sea Trial project and was supported by Business Finland. The author acknowledges Novia for providing the research environment and infrastructure, and thanks all project partners for their collaboration and insights.

## References

[1] de Oliveira Chagas, I. B., & Tomim, M. A. (2022). Co-simulation applied to power systems with high penetration of distributed energy resources. *Electric Power Systems Research*, 212, 108413.

[2] [Github repository of the unofficial implementation](#)