Document Version
Final published version

Citation (APA)
López Torres, C., Cvetkovic, M., van der Meer, A., & Palensky, P. (2019). Co-simulation of Intelligent Power Systems. In P. Palensky, M. Cvetković, & T. Keviczky (Eds.), *Intelligent Integrated Energy Systems: The PowerWeb Program at TU Delft* (pp. 99-122). Springer. https://doi.org/10.1007/978-3-030-00057-8_5

Chapter 5
Co-simulation of Intelligent Power Systems

Claudio David López, Miloš Cvetković, Arjen van der Meer and Peter Palensky

Abstract The complexity of energy systems increases as more renewable generation and energy storage technologies are added to the grid. Diverse energy carriers are becoming interconnected and the grids are getting reliant on communication networks for timely operation. The arising complexity is difficult to model with the existing mathematical models and using existing simulation tools due to confinement of these models and tools to a subset of the interconnected system. To overcome this challenge, combined simulation (co-simulation) methodology is being deployed. In co-simulation, multiple models and tools are being interconnected to truthfully represent reality. In this work, we review several aspects of co-simulation. First, we look at interconnecting transmission and distribution grid simulations in order to enable collaboration between transmission system operators (TSOs) and distribution system operators (DSOs). Next, we investigate co-simulation as means to dynamic model exchange between TSOs. Finally, we analyze co-simulation capabilities for running experiments in remotely connected research labs.

5.1 Coupled Simulation of Multiarea and Transmission/Distribution Systems

The unprecedented complexity of modern power system has called into question the traditional approach to their analysis. In this traditional approach, an area of interest in the system is selected and analyzed in detail, while its surrounding areas are

---

C. D. López (✉) · M. Cvetković · A. van der Meer · P. Palensky
Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, Delft, The Netherlands
e-mail: c.d.lopez@tudelft.nl

M. Cvetković
e-mail: m.cvetkovic@tudelft.nl

A. van der Meer
e-mail: a.a.vandermeer@tudelft.nl

P. Palensky
e-mail: p.palensky@tudelft.nl

P. Palensky et al. (eds.), *Intelligent Integrated Energy Systems*,
https://doi.org/10.1007/978-3-030-00057-8_5

represented by simplified equivalent models. For example, at the transmission level a distribution grid could be represented as a current source and at the distribution level a transmission grid could be represented as a voltage source or a generator with a large inertia constant. While the use of equivalent models certainly simplifies the analysis, technologies like power electronics and ICT have only made it more difficult to come up with equivalents that are truly representative of the grids they stand for.

As the diversity and complexity of the technologies that can be found in a power system increases, so does the need for more detailed analysis methods. These methods should properly account for the uniqueness of each area in the system and the interactions between neighboring areas. Two main approaches to address this need follow: developing better equivalent models and using detailed models instead of their equivalents.

Equivalent models have the inherent advantage of reduced computational burden when compared to detailed models. However, the right assumptions need to be made to ensure representativity. Furthermore, as systems become more complex, the complexity of the methods required for obtaining the equivalent model also increases. One way to eliminate the challenge of creating equivalent models that are indeed representative would be to use full models, but there are some practical obstacles in this case as well.

Oftentimes a full model of a grid is not available, either because it does not exist or because it is owned by a third party that, out of privacy concerns, is unwilling to share it. A similar challenge can be encountered if instead the model is to be developed from scratch; the grid data required to develop the model might be owned by a third party and it might be confidential. Furthermore, even if a full model is already available, simulating a multiarea system in such detail would be very computationally expensive. Additionally, such a large model would be highly labor intensive to maintain.

5.1.1 Co-simulation as an Approach

In a co-simulation several independent simulators, each simulating only a part of a larger system, collaborate at runtime by exchanging data. The data exchange and the synchronization of the local simulation time of each simulator is orchestrated by a so called co-simulation master. Since the data exchange can be over a communication network, the simulators can be geographically distributed. This open up the possibility for different institutions to simulate collaboratively while bypassing any limitations due to confidentiality of information and/or models. Using co-simulation for multiarea system analysis has a set of additional advantages:

- Access to data: The model of each area can be developed independently by the institution that has access to the information needed.

- Tool flexibility: The simulation tool of choice for each area is irrelevant since data is exchanged over a network using a standard protocol.
- Extensibility: New simulators or models can be easily coupled.
- Privacy: The models can remain private if required since only selected simulation variables need to be shared with other simulators at run time.
- Reduced work load: Tasks related to model development and model maintenance are naturally divided among those that have access to grid information.
- Reduced computational load: The distributed nature of the co-simulation allows for the computational load of the co-simulation to be shared.

5.1.2 Challenges

Implementing such a co-simulation does not come without its challenges. The first challenge that must be overcome stems from the need to accommodate closed-source simulators in the co-simulation. In many cases this means that the co-simulation interfaces needed to couple the simulators in a co-simulation environment lack certain functionalities, for example, time roll back [1]. In many cases it becomes necessary to resort to hacks in order to create the interfaces. Another challenge, also related to closed-source simulators, has to do with step size control. The easiest case to manage is when all simulators use the same time step size and this remains constant during the co-simulation. However, there are cases when this cannot be ensured. In such cases the complexity of the synchronization mechanisms increases rapidly, especially when a potentially large number of simulators participate in the co-simulation. Numerical accuracy and stability of co-simulations can be difficult to ensure as well, and the limited control that users have over a simulator often stands in the way of certain co-simulation methods that are useful for addressing numerical challenges [2]. Managing a co-simulation like this is also a challenge, as a large number of simulators can be involved. Tools like mosaik [3] have been developed to address this problem at the software level, but the management challenge goes beyond software when several stakeholders are involved in running a co-simulation collaboratively. The performance of these co-simulations is a concern as well, especially if the number of involved simulators is large or if the frequency of data exchange is high, which is the case of EMT-type co-simulations [4].

5.1.3 Environment

The co-simulation environment developed to research this type of co-simulation is depicted in Fig. 5.1. The environment is composed of a set of Windows Server 2012 virtual servers, each running PowerFactory 15, plus an additional virtual server running the co-simulation master. The co-simulation master was developed in Python specifically for this application. The simulators and the master exchange

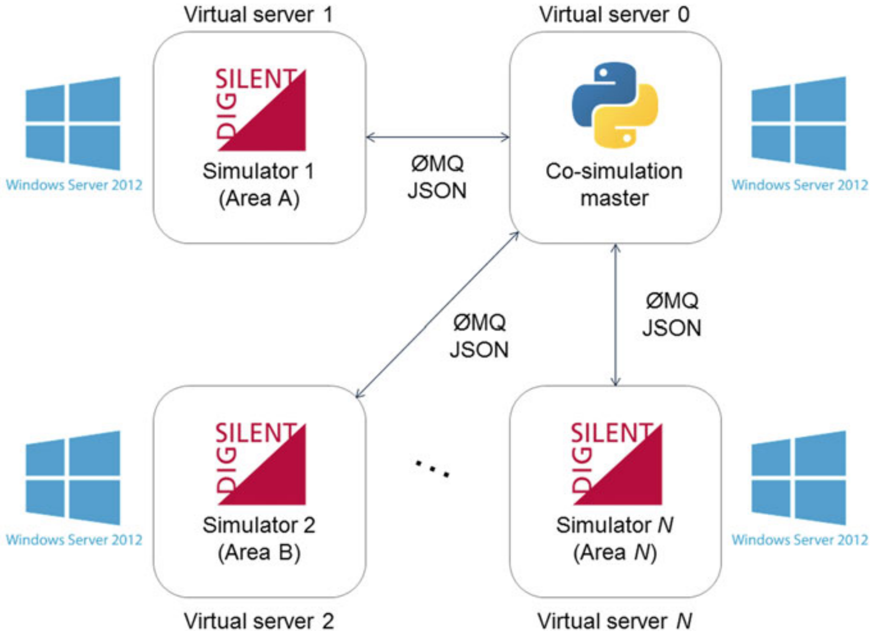


Fig. 5.1 Co-simulation environment for dynamic multiarea power system analysis

JSON-encoded messages over ØMQ sockets. These messages contain simulator inputs/outputs and all the necessary information for time synchronization.

5.1.4 *Example*

As an example of a multiarea co-simulation run with the previously described co-simulation environment, let us consider the transmission grid from Fig. 5.2 and the distribution grid from Fig. 5.3. The distribution grid is connected to Bus 6 of the transmission grid.

Figure. 5.4 shows the co-simulation environment and the results of a co-simulation of the two systems from Figs. 5.2 and 5.3. In this co-simulation a three-phase to ground fault occurs on the low voltage side of the distribution transformer (Fig. 5.3) at 0.05 s. Figure 5.5 shows some of the co-simulation results in more detail. This figure compares the phase voltage of one phase as measured on each side of the co-simulation interface (v_a). The same comparison is made with the current flowing through the interface (i_a) (between the transmission and distribution grids). There is almost no perceptible difference between voltages and currents on each side of the interface, which might give the false impression of high co-simulation accuracy. However, the difference on the power flowing through the co-simulation interface

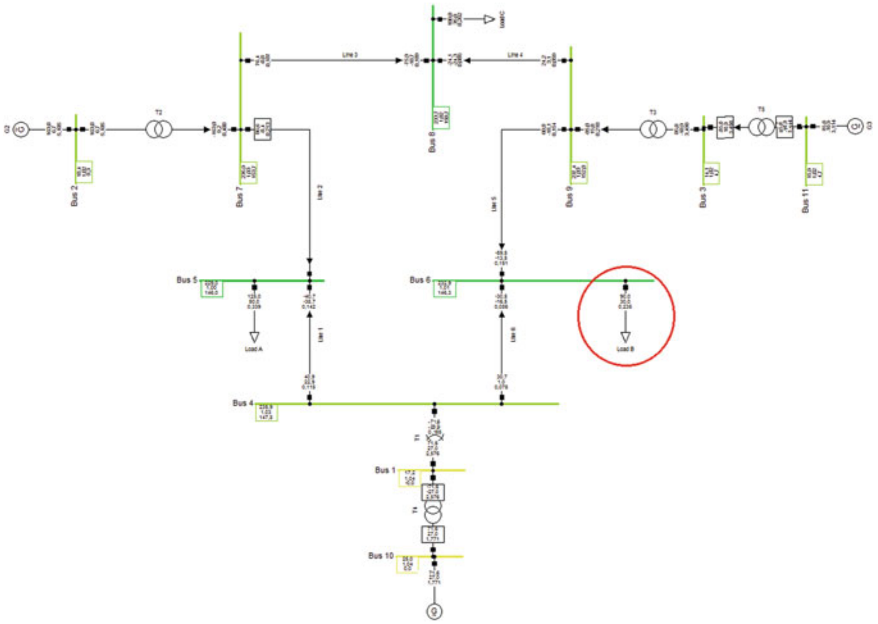


Fig. 5.2 IEEE 9 bus transmission system. The load marked in red represents the distribution grid. This grid is modeled as an ideal current source

(p_a), also shown in Fig. 5.5, is much more pronounced. This means that the power that the transmission grid is sending to the distribution grid is not the same as the power the distribution grids receives from the transmission grid, which is a violation of the law of conservation of energy. The difference between the power flow on each side of the co-simulation interface (Δp) provides a measure of how inaccurate the co-simulation actually is. In this case, while the co-simulation is in steady state the Δp is very small, but as soon as a disturbance occurs in the system, the Δp reaches a value beyond 2 MW. Despite the loss of accuracy that a co-simulation might display, its advantages still stand. Engineering judgment is required to determine whether these advantages outweigh the loss of accuracy or if more sophisticated co-simulation methods that have higher computational burden but deliver more accurate results are needed, for example, iterative methods [1].

In this section, we introduced co-simulation, its advantages and disadvantages, and we illustrated its accuracy in a representative use case. In the following section, we compare co-simulation to other methods for exchange of dynamic models between TSOs.

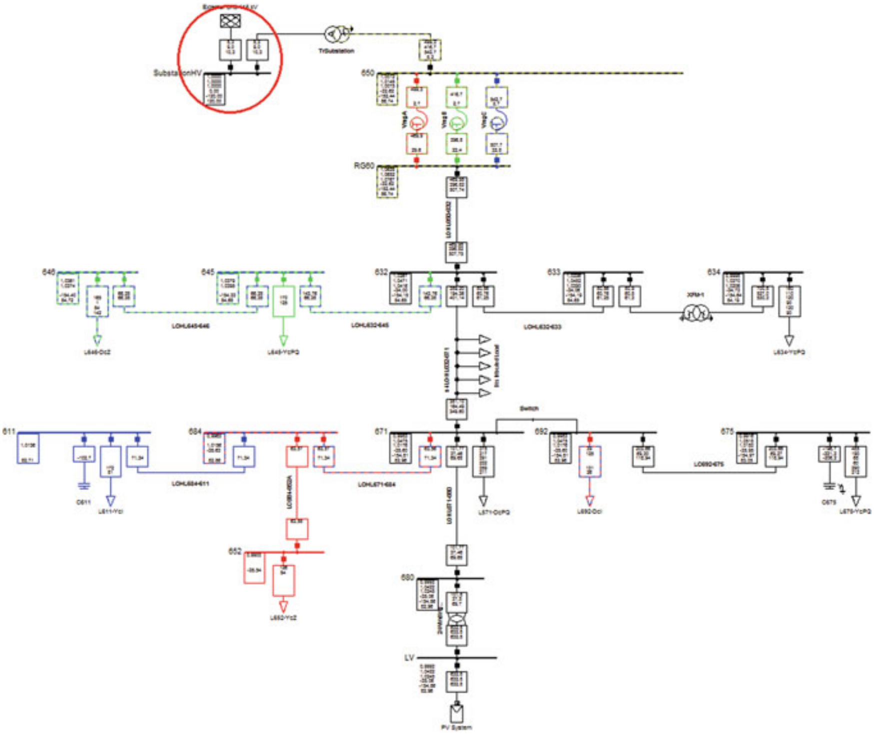


Fig. 5.3 IEEE 13 bus distribution system. The external grid marked in red represents the transmission grid. This grid is modeled as an ideal voltage source

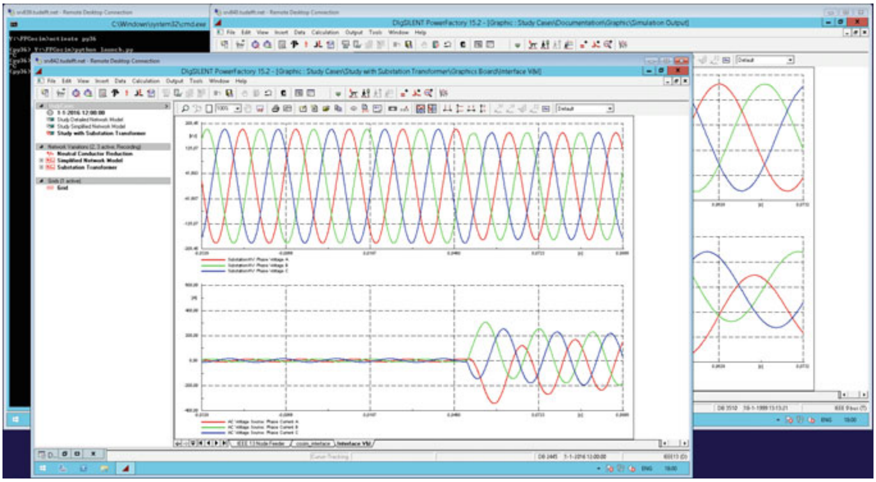


Fig. 5.4 Co-simulation environment running PowerFactory in two servers and an additional server for the co-simulation master. Some co-simulation results can be observed

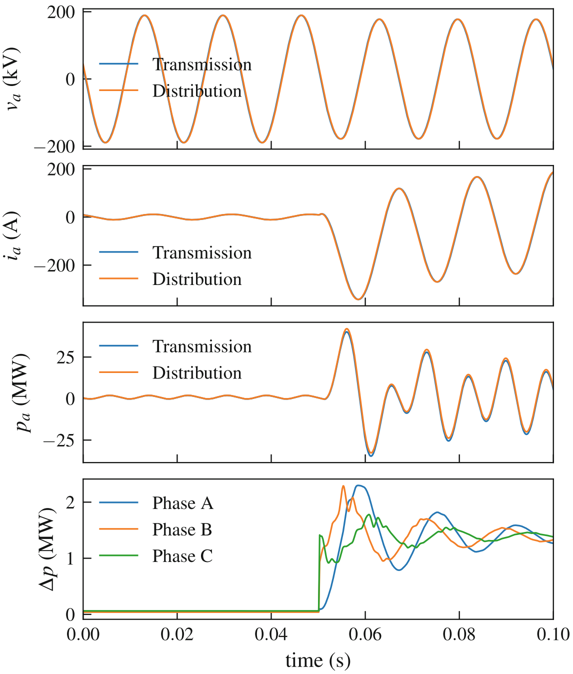


Fig. 5.5 Phase voltage v_a, phase current i_a and phase instantaneous power p_a at and through the interface as measured on the transmission and distribution sides of the co-simulation. The difference between the power flowing through each phase of the interface as measured on the transmission and distribution (Δp) side provides a measure of the inaccuracy of the co-simulation

5.2 Dynamic Model Exchange and Co-simulation

Data and model exchange between TSOs is streamlined in practice using Common Grid Model Exchange Standard (CGMES) [5]. At this point of its development, version 2.5, one of the key features targeted for improvement is model exchange for dynamic studies, particularly in terms of compatibility with user-defined models of novel controllers and prototypes of new equipment.

In addition to the targeted improvements, multiple possibilities for the choice of the toolchain for dynamic model exchange are being considered. The main technical challenge for sharing the models is that TSOs keep and maintain the models within commercial simulation tools of their choice. These models are developed and updated over a span of many years, sometimes even decades, on the occasions of system expansion and component model validation. Since different vendors supply different TSOs, the problem of finding the appropriate means for model exchange becomes the problem of finding an adequate technological solution for sharing the information between simulation tools.

The possibilities to address this challenge are diverse (see Fig. 5.6). The simplest approach that comes to mind is to share a static data format, similarly to already established practice with static power flow models (see CGMES rules on static model exchange [5]). There are two main benefits of using such approach. First, the required technical solutions are relatively simple to design and implement.

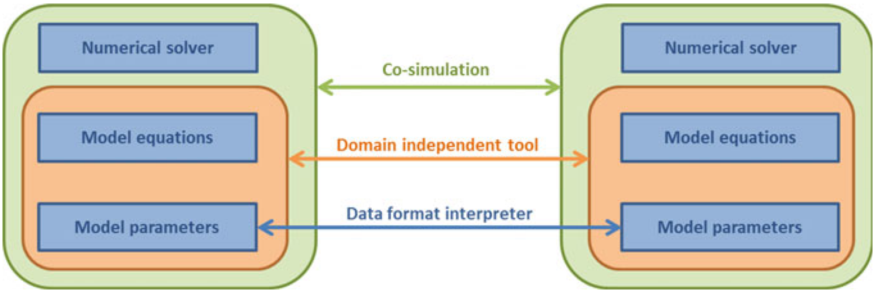


Fig. 5.6 Approaches to dynamic model exchange

Data format interpreters that are capable of exporting model information into the desired format and importing it back from the same format are easy to create using well established parser definitions. Second, the TSOs have been historically open to exchanging information in terms of static datasets [6]. Hence, the adoption barrier is relatively small. However, the accompanying challenges of this approach are far from straightforward to address. The main difficulty is that different simulation tools often use different component models that are not always comparable. For example, one tool might support the model of a synchronous generator with a single damper winding while the other tool might not. In such a case, the time-domain responses of the two simulators would yield different results. To overcome this obstacle, the data format interpreter would need to have a sufficient level of intelligence to handle these model misalignments in a most adequate manner. Defining the KPIs that describe the *most adequate manner* is a challenging task that would require considerable attention.

A slightly more involved approach to model exchange is to use a single target platform (i.e. simulation tool) when exporting the models [7]. If all TSOs export their models to the same target platform, sharing the models among them becomes trivial. The most appropriate choice for a target platform seems to be a domain independent modeling and simulation tool. The domain independent modeling and simulation tools, such as Modelica and Matlab-Simulink, are created for addressing wide range of applications. Thus, they use basic mathematical operations and functions as elementary building blocks (e.g. Modelica uses differential and algebraic equations in acausal manner while Simulink uses basic mathematical operations in causal manner including differentiation and integration and user-defined script functions [8]). In theory, since power system components are physical hardware components, any model encapsulated within proprietary power system simulation tools can be represented in a domain independent tool. Thus, the models can be exported and then simulated in a domain independent tool. This process is referred to as *model migration*. Although exporting models is generally more involved than exporting data, this task is deemed feasible if/when the TSOs develop trust and conformity with the domain independent tool in question.

However, one of the main technical challenges still remains. The commercial tools for simulation of power system dynamics often deploy proprietary modifications of numerical solvers in order to improve the performance of the tool and keep the competitive edge. At times, the proprietary modifications are also applied to the models. If/when this happens, no guarantees exist that the domain independent tool and the proprietary software tool will yield the same time domain response.

Simulations of power electronically enriched systems pose additional challenges for domain independent tool approach. The manufacturers of converters often keep their models encapsulated as black box models. Such models would be impossible to export to a domain independent modeling and simulation tool without the involvement of the vendor. In addition, some domain independent tools, such as Modelica, are created with open source policy in mind and their support for protecting proprietary models is limited (although possible using Functional Mock-up Units - FMUs [9]), making this approach to model exchange more involved.

Another difficulty with using domain independent tool appears in simulations of any large expansion project. Each large expansion project comes with specific design requirements and components that are often unique for that particular project. In such a case, user defined models must be created to capture the relevant dynamics. Migrating user defined models to a domain independent tool represents a challenge and an open problem [5]. PowerFactory for example uses its own proprietary scripting language (DSL) and its own proprietary block-diagram modeling tool (DPL). A user could, in all their freedom, develop a component so complex, that the migration from PowerFactory to Modelica becomes extremely difficult. The same challenge appears in the case of heterogeneous components at the distribution system level and in simulations of new technologies.

Finally, the co-simulation can be considered as the third and most comprehensive approach to dynamic model exchange. In fact, it is more appropriate to consider this approach as an alternative to model exchange, since it does not involve any model exchange per se. Co-simulation is created by exchanging the data between the proprietary simulators in run-time. Thus, there is no need to export the models and the parameters. Instead, the values of the so-called coupling variables are being exchanged as the simulations are executed. For example, if two TSOs wish to run transient stability simulations, they would couple their simulation tools. Each simulation tool would run the models and solvers that it typically runs with one addition, the boundary condition information would be exchanged between the tools in run-time. These boundary condition information are also referred to as exogenous information for each of the tools, or coupling variables on the interface between the tools. This approach guarantees that even the vendor-made adjustments to the model-solver combination together with the complex user-defined models would be accurately included in the resulting simulation response. At the same time, the TSOs would have high confidence in correctness of the response.

It should be noted that verification of the models is a challenge that accompanies all three of these approaches, although to a slightly different extent and in a different form. When using a data format interpreter or domain independent tool, the burden of verification is placed on the TSO that is exporting the models. The TSO would have to ensure that the imported model produces the same response as the model that is being exported. Since the TSO has the complete trust in its operating models, the verification would typically be done against the original model and the original simulation tool. In the case of the data format interpreter, this might have to be done for many different target platforms (PowerFactory, PSS/E, PSS/Netomac, PSCAD, etc.) which significantly increases the effort and expertise of the involved personnel. In the case of a specific domain independent modeling tool, the effort would be reduced to a single tool.

In the case of co-simulation, verification of time responses is more difficult to perform. Two alternatives exist: verification against a simulation tool, and verification against the field data. If verification is done against another (commercial) simulation tool, one would have to create exactly the same reference model in this commercial simulation tool since such reference model does not exist. In doing this, one would repeat the same work that would have to be done if data format interpreter or domain independent modeling tool are used for model exchange. Thus, this verification process defeats the purpose. Verification using field data is another option. Since existing TSO models are anyhow verified using signatures from the field data, extending this approach to verify co-simulation seems feasible in practice (albeit not without difficulties since the data belongs to several organizations).

In the rest of this section, we compare model migration to a domain independent modeling and simulation tool (Modelica in particular) with the co-simulation approach for dynamic model exchange. First, we introduce the case study used for comparison. Next, we analyze the results and identify similarities and differences between the obtained time responses. Finally, we outline possible technical reasons for the discrepancy between the signatures.

Case Study: Comparison of Model Migration and Co-simulation

The case study for comparison is chosen with several requirements in mind. First, the case study must be simple enough to minimize possible factors for potential discrepancies between two approaches. Yet, it has to provide sufficiently accurate representation of typical power system dynamics. Second, commonly used and industry adopted tool must be deployed as a benchmark for verification purposes.

To satisfy these two requirements we choose the same case study as in [10]. In this case study, the power system model is a three bus system. A synchronous generator is connected to Bus 1 and two identical loads are connected to Buses 2 and 3. One transmission line connects Buses 1 and 2 and another one connects Buses 1 and 3.

The benchmark simulation is conducted in PowerFactory 15.2. To illustrate model migration, the same power system has been modeled in Modelica using models developed in [7]. To illustrate co-simulation, the PowerFactory model is divided into two. The load at Bus 3 and transmission line connecting buses 1 and 3 are simulated in one instance of PowerFactory while the rest of the system runs in another instance of PowerFactory. The exchange of data among the simulators and synchronization of simulation execution is achieved by using a light weight co-simulation master algorithm from [4].

It is important to mention one detail regarding the following comparison. Model migration is benchmarked against Root Mean Square (RMS) simulation while co-simulation is benchmarked against Electromagnetic Transient (EMT) simulation. This is a non-ideal situation since the use of two different modeling paradigms allows only indirect comparison of model migration and co-simulation. The only reason to take this approach is of practical nature. Currently, model migration of RMS models is much easier to perform. At the same time, co-simulating RMS models is much more difficult than co-simulating EMT models. Thus, the difference in the two taken approaches.

The comparison is performed under two characteristic events. The first simulated event is a short circuit at Bus 2 in duration of 0.2 s created after 5 s of simulation time. The second event is a complete loss of load at Bus 2 that occurs at the 5 s mark and lasts until the end of simulation. Even though the simulations are performed from 0 to 10 s, only the most interesting time period of simulation is shown in the plots that follow.

We compare percentage difference error for some characteristic variables of this test case. We look at electrical torque and rotational speed of the synchronous generator and voltage at the generator terminal (Bus 1). The percentage difference error is computed according to the following equation

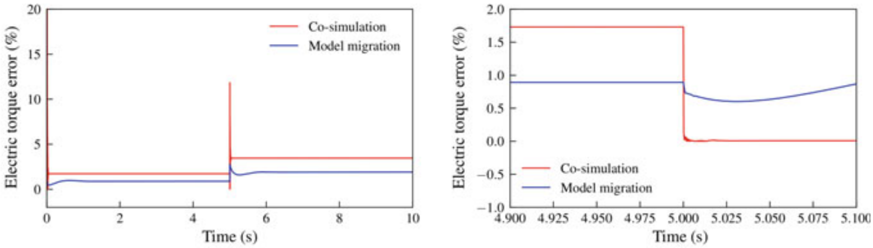
$$err(t) = \frac{x(t) - \bar{x}(t)}{\bar{x}(t)} \cdot 100\% \tag{5.1}$$

where $x(t)$ is the variable under scrutiny obtained as a result of model migration or co-simulation while $\bar{x}(t)$ is the same variable obtained from the benchmark simulation.

We observe the results of the simulation runs in Fig. 5.7. By inspection, we see that the error is always the largest at the moment of event. This is expected since the switching events excite dynamics on all time scales and are sometimes handled differently by different tools. With all but one studied variable, the size of error is within reasonable range. The only variable with the higher error value, terminal voltage in the short circuit case in Fig. 5.7f, is the one that is most sensitive to the events in the grid. The high error value can be explained by the small values of benchmark voltage during the short circuit event (see Eq. (5.1)).

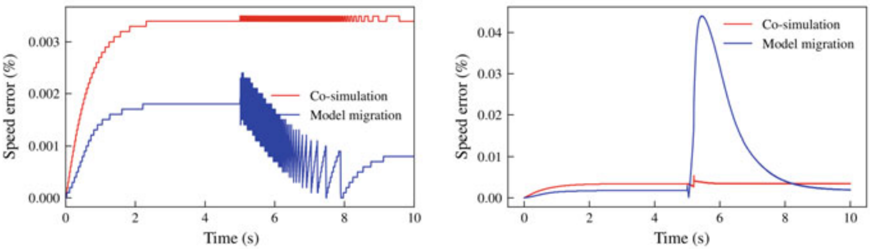
We also observe that co-simulation is more sensitive to events than model migration (see Fig. 5.7e). This is in part due to the modeling differences in RMS and EMT (EMT is more detailed, and thus, more sensitive to the events). The second reason, which is of higher interest for this work, is that the co-simulation master is less optimized to handle events than the internal solver of a monolithic simulation tool. For example, our co-simulation master does not have access to the system Jacobian while the internal solver does. This is a characteristic of the co-simulation approach that generally results in less accurate responses immediately after the event.

Another difference between model migration and co-simulation is that error in the case of model migration typically takes longer to settle (see Fig. 5.7d). This is mostly attributed to the differences between RMS and EMT. At the same time, the value of error in steady-state is sometimes larger in the case of co-simulation (see



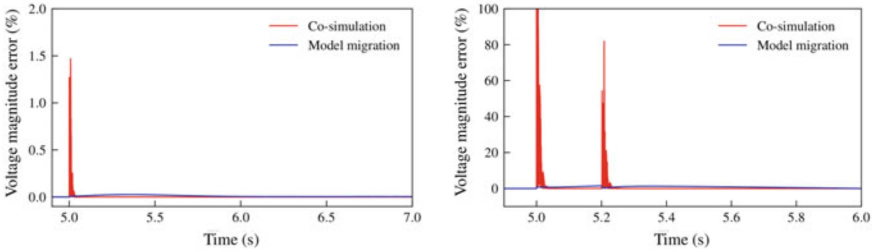
(a) El. torque error in the loss of load case.

(b) El. torque error in the short circuit case.



(c) Rot. speed error in the loss of load case.

(d) Rot. speed error in the short circuit case.



(e) Gen. voltage error in the loss of load case.

(f) Gen. voltage error in the short circuit case.

Fig. 5.7 Comparison of errors in model migration and co-simulation

Fig. 5.7a) and sometimes in the case of model migration (see Fig. 5.7b). The steady-state error depends on the models and also on the initialization approach and the choice of interfaces for co-simulation. These are different from one tool to another and can, to some extent, impact the dynamic response as well.

In this section, we compared two approaches for dynamic model exchange, model migration and co-simulation. As illustrated in the case study, both approaches have potential to achieve high accuracy. In the next section, we look at the role of co-simulation in the holistic testing and validation of smart grid experiments.

5.3 Holistic Testing and Validation of Cyber-Physical Energy Systems

Assets in power systems commonly have a life span of decades, being the primary equipment such as transformers, cables, and switch gear, or the secondary equipment like protective relaying and communication infrastructure. Such high durability puts a burden on the associated control algorithms and the energy management system, which as a platform have to maintain semantic compatibility across devices and subsystems for a long time. This used to hamper the deployment of innovative concepts and the operation of the power system is therefore very conventional (centralised control based on the physical properties of the connected units).

Successive technological developments have led to more rapid deployment of new primary and secondary equipment. For instance, over the past decade TSOs deployed well-controllable power electronics at high voltage levels, potentially offering devices the capability to behave as conventional power plants at the grid interface. This makes the nature of the primary source of a lesser concern and fosters the coupling of multi-energy to the electricity grid. The rapid digitalisation of our society on the other hand enables faster, widespread communication, and massive data acquisition, which is not left unnoticed in the power system. Decentralised control, automatic coordinated control, fast supervisory interventions by a centralised entity are striking examples that are current practice, i.e., smart grids. Above all, this makes interactions inside the power system faster and largely based on controls and its associated ICT infrastructure rather than physical response.

The entanglement of the multi-energy power system with ICT infrastructure leads to an unprecedented level of heterogeneity: the cyber-physical energy system (CPES). To maintain the same level of comfort and reliance on the electrical energy in our lives, measures for ensuring reliability and security of supply must be followed in the CPES. It is therefore significant to chart how

- domains such as electricity, ICT, and heat interact with each other, and
- test and validate new concepts for smart grids.

Testing new concepts is a challenge because laboratories are traditionally specialised into one particular domain such as ICT security, high-voltage electrical equipment, high-power electrical devices, etc. Moreover, a categorisation can be made in terms of experimental focus, such as laboratories focusing on pure hardware experiments, pure software experiments, or a combination like hardware in the loop (HIL) assessment. Testing a smart grid concept hence commonly yields drastic simplifications of the interconnected domains or a non-ideal experimental setup. For the conventional power system this was not an issue and abstracting out the boundaries of the system was common practice. Nowadays, the heterogeneity is more prominent and holistic system validation spanning multiple domains and utilising various experimental concepts is considered paramount.

The complexity of the CPES hence calls for a system-wide, cross-disciplinary procedure to test, validate, and roll-out smart grid concepts. Under the umbrella of the ERIGrid research project various smart-grid research infrastructures across Europe have set out a formal testing and validation procedure to cover these needs [11]. It combines the merits of established (quasi-)standards (e.g., smart grid architecture model, common information model, and unified modelling language [12]) with the testing expertise of laboratories. Besides subsystem-level validation the procedure aims to make tests transferable (and even partitionable) among research infrastructures and facilitate reproduction of experimental results.

5.3.1 Holistic Testing Procedure

In [13] the concept of *holistic testing* was introduced as being *"The process and methodology for evaluation of a concrete function, system or component within its relevant operational context with reference to a given test objective."*. Especially the system and their components exhibit functions that are cross-domain. Take as an example a centralised controller inside a wind power plant: physical quantities like voltages and currents are measured at the terminals of the wind turbines and transported to the park controller across dedicated communication channels over for instance IEC61850. At the connection point of the wind park the voltage and reactive power exchange with the transmission system shall be maintained within strict boundaries. This can be achieved by dispatching reactive power setpoints to the individual wind turbines. The communication channels can, however, cause latencies inhibiting the operation of the wind park as such. This cross-coupling of the physical, control, and ICT domain is very prominent here and needs to be carefully considered during specialized component and system testing.

To facilitate a bit of structure in such complex systems it is therefore important to specify which components and systems interact and how this relates to actual test objective. In ERIGrid this led to the holistic testing approach, first proposed in [14] and outlined in Fig. 5.8. It formalises the separation of the general test case description and specification (i.e., *what* needs to be tested and *why*?) from the actual experimental implementation accordingly (i.e., *how* will the system or component test be carried out?).

The test case description (i.e., step ①) is the most abstract and contains predominantly the following attributes:

- The *test objective*, the purpose for carrying out the test or sequence of tests;
- The *Use Case*, which is a high-level description of the functionality of the considered cyber-physical system [15]; and
- the *generic system configuration*, which hierarchically specifies the type of and relation between components and domains.

Then the system under test delineating the system boundaries of the generic system configuration is to be formulated. It encompasses all relevant interactions and function that require investigation. Central in these interactions is the object under investigation, which is the component or sub-system inside the system under test to

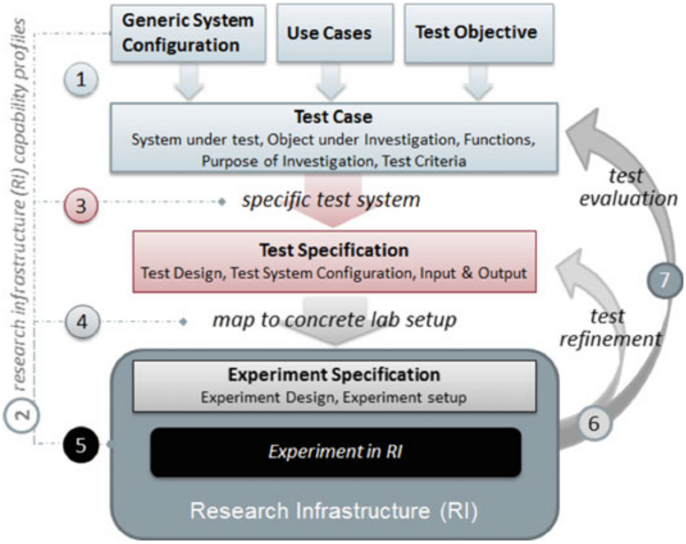


Fig. 5.8 The main specification steps for the holistic testing approach in the ERIGrid research project

which the eventual test criteria shall be evaluated. Test criteria can have a characterisation, verification, or validation nature.

The next step is to project these metrics on a specific test system (i.e., step ③), yielding a description of inputs and outputs, quantitative test metrics to make a test pass or fail, and all further design specifications of the test that are independent of the implementation of the experiment. Some research infrastructures are more tailored for a particular experiment than others. A potentially destructive experiment can only be realised under well-evaluated conditions in specialised laboratories. (Co-)simulation experiments tremendously sustain the qualification of such test boundaries and assumptions. It is hence significant to carefully map a (part of a) particular test on the capabilities of laboratories and institutes (i.e., steps ② and ④). Eventually, the experiment itself needs to be specified. At this level (i.e., ⑤), relevant connection diagrams of lab components, data acquisition, safety regulations, data type conversions (so the experiment setup) and the execution, repetition, and treatment variation (so the experiment design) are significant.

So far the approach has been pretty much feed forward: the test case has been specified at various levels of granularity, starting at a conceptual and abstract dimension to a more detailed, implementation driven level. To account for reproducibility of tests and their statistical relevance, the overall test design is evaluated (for instance by screening or sensitivity analysis) and adjusted accordingly [16] (i.e., steps ⑥ and ⑦).

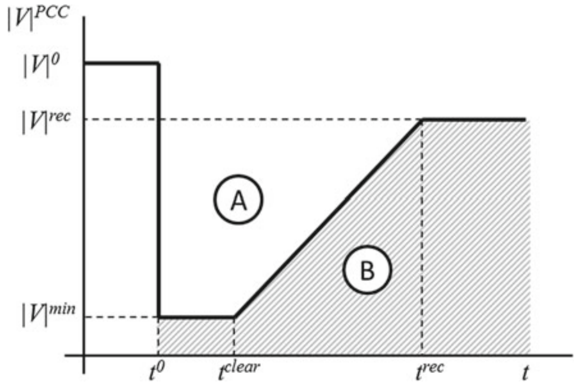


Fig. 5.9 Illustrative fault ride-through voltage versus time curve for wind parks

5.3.2 *Holistic Test Case Example*

The approach outlined in the previous section is illustrated with the test design that revolves around the use case the so-called fault ride through compliance of an onshore wind park. As seen from the transmission system, such parks are considered one single generation entity that must legally comply to various requirements at the point of common coupling, which is often the high voltage side of the transformer linking the collection grid with the transmission system. Fault ride-through entails the ability of the wind power plant to stay connected during voltage dips experienced at the point of common coupling. This is challenging for various reasons but most importantly because of the vulnerability of the power electronic converters inside the individual wind turbines.

The fault ride-through requirement is often expressed by a voltage versus time envelope as illustrated in Fig. 5.9. Starting at fault ignition at t^0 the wind park is only permitted to disconnect if the per unit voltage amplitude at the point of common coupling drops below the indicated voltage profile and enters the dashed gray zone. Otherwise it should stay connected. It consists of mainly four parts. The prefault part in which the terminal voltages are around $|V|^0$, the faulted part during which the (remote) fault causes a severe voltage dip (Ⓐ), the recovery part after fault clearance (the skew area, Ⓑ), and the post-fault ride through part at which the system and hence voltage is expected to behave normally again.

In terms of illustrating holistic testing, fault ride through is an attractive option as

- fault ride through involves interactions between multiple domains like ICT, physics, control.
- This domain coupling is rigid and these interactions are fast so abstracting away phenomena comes at a severe risk of false positives. System Configurations and corresponding functions must hence be carefully chosen.
- Though the fault ride through objective is global (at the point of common coupling), the implementation is done locally by the individual wind turbines. This needs to be reflected into the test metrics (criteria).

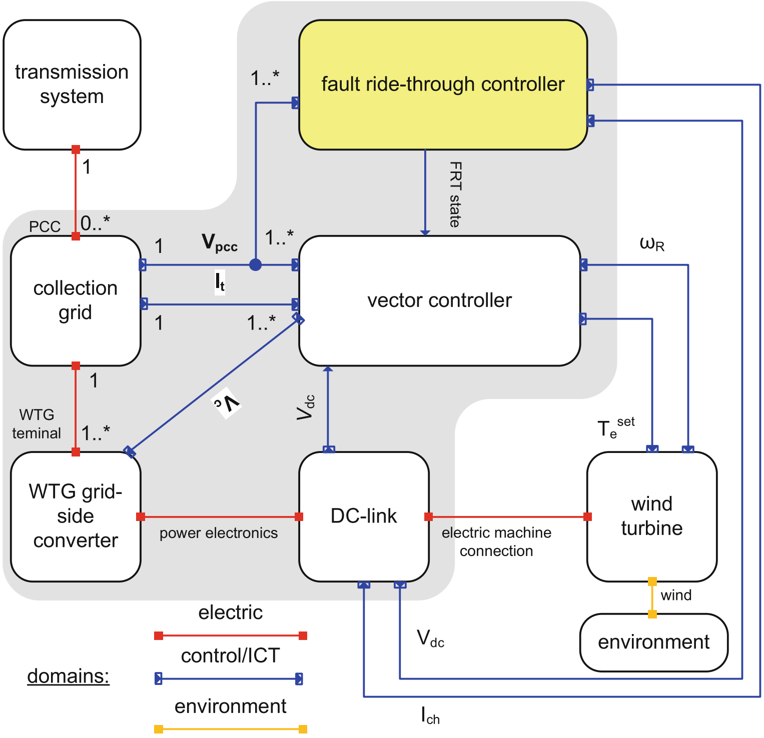


Fig. 5.10 Generic system configuration of the fault ride-through operation of a wind power plant; gray: system under test, yellow: object under investigation

- Testing fault ride through can be destructive to components, which needs a smart test and experiment design.

The objective of the test is to verify the ability of wind park as a generating plant to withstand the external voltage dip and remain connected during and after the fault. The implementation is done locally by the wind turbines, so their controls are the object under investigation. The system functions that are *assumed* dominant to fault ride through are the transient voltage and frequency response of the power system, the active and reactive power controls of the wind turbine, and the power electronics protection devices. The involved (sub-)systems and components of these functions make up the generic system configuration of the test case, which is shown in Fig. 5.10. At this stage only the types and multiplicity of the components and their relational setup is given, all in terms of the functions relevant for fault ride-through. Therefore, this diagram is aimed to be valid for any fault ride-through capable wind park consisting of so-called full-converter generator wind turbines.

The test is considered successful when the wind park is able to stay in operation during the fault and the system quantities like voltage and frequency return dynamically to a stable operating region (i.e., nominal).

Now the generics of the test have been specified we can have a look into the actual design of the experiment. The metrics to assess the test criteria are the output current ($\mathbf{I}_t$) or power to show that the turbines remain connected and the amplitude of the voltage and frequency at the point of common coupling (i.e., $|V|^{\text{PCC}}$ and ω_{G} respectively) to track whether the system returns to a plausible (stable) operating point after the fault. The relevant component input/output variables are shown in Fig. 5.10. As the functionality to be assessed is triggered by short circuits in the transmission system, the design of the test procedure is deterministic and based on the behaviour of the system under test after faults, and can be summarised by:

1. achieve consistent operating point throughout the system under test
2. determine the short circuit location x such that the depth of the voltage dip at the point of common coupling approaches $|V|^{\text{ret}}$ in Fig. 5.9.
3. initiate fault at t^0 and start obtaining time-stamped component and system measurements
4. clear the fault at $t^{\text{clear}} = y$
5. assess the test criteria
6. vary x and y to cover both short close faults and longer remote faults
7. repeat the experiment.

As field tests are infeasible here for obvious reasons, we need to look for alternative ways to implement the experiment. Ideally speaking, the above test design shall be implemented in a laboratory that at least implements components or subsystems of the wind park in hardware. This could lead to controller hardware-in-the-loop in which for instance the real park controller is assessed whereas the remainder of the system is simulated in real time. Both are then coupled through the I/O interface of the respective real-time simulator. Alternatively, the electrical part of one of the wind turbines could be interfaced with a real time simulator by coupling it through a controllable grid interface (i.e., power hardware-in-the-loop).

Computer simulation is, however, the most optimal option to conduct the experiment. Optimal in the sense of costs, safety (no experiments harmful to humans or hardware) but also flexibility (parameter and model adjustment, determination of initial operating point, model validation, and reproduction). The downside of simulations is that, especially with a rigid cross-domain coupling, the validity of the system response as a whole is determined by the validy of the individual models. To gain a system model with a well defined level of detail for each specialised domain is challenging and usually scale badly (model size, simulation time).

As discussed in previous sections, these issues can largely be overcome with co-simulations, which allow the corresponding subsystems and components to be considered individually by specialised simulators. A master process then keeps track of the interfaces between models over time and event handling. This co-simulation approach is also used with the current example of the wind park.

Figure 5.11 shows a layout of the experimental implementation by co-simulation. It depicts the master algorithm on top and the coupled simutors with their respective subsystems below. The co-simulation employs the functional mockup (FMI) interface

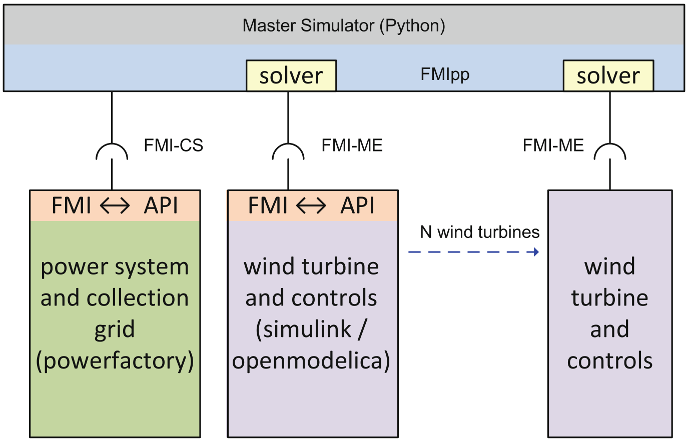


Fig. 5.11 Experimental setup of the co-simulation for fault ride-through testing. FMI: functional mockup interface, CS: co-simulation, ME: model exchange, API: application program interface

standard for interfacing continuous and discrete simulations [17]. FMI defines a set of functions, attributes, and a specification format to which a simulation (FMI for co-simulation) or model (FMI for model exchange) should comply to in order to be able to cooperate/interface with other simulations or the master. A simulator or model that is encapsulated such that it can be programatically linked to the master simulator is called a functional mockup unit. The simulator blocks in Fig. 5.11 each contain an adapter that fulfils this and are hence functional mockup units. The master algorithm is implemented in the Python programming language and uses the Python bindings of the FMI++ library [18], which highly facilitates the adaptation of the functional mockup standard for power system studies.

In this case it was decided to model the power system in Powerfactory whereas the component models are modelled either in Matlab/Simulink or in Openmodelica. Several reasons can be found to have a split like this. First, wind turbine models are commonly vendor-specific and are commonly delivered to customers as black or gray boxed models for a particular simulator. Second, a tool like Powerfactory is well known for its hierarchical scenario and case variation possibilities but is less respected for its ease of model development, which is on its turn the unique feature that makes Simulink and Modelica popular. Third, the static generator model inside Powerfactory allows a flexible and powerful interface to dedicated models (either in its own DSL language or externally developed models).

Although one could in fact represent each wind turbine as an functional mockup unit (shown in Fig. 5.11) and in such a way run a very detailed co-simulation, we limit ourselves to an aggregated wind turbine for illustration purposes. The wind power plant is modelled as a wind turbine aggregate and the collection grid is abstracted to a series impedance, which represents the step-up transformer to the transmission network.

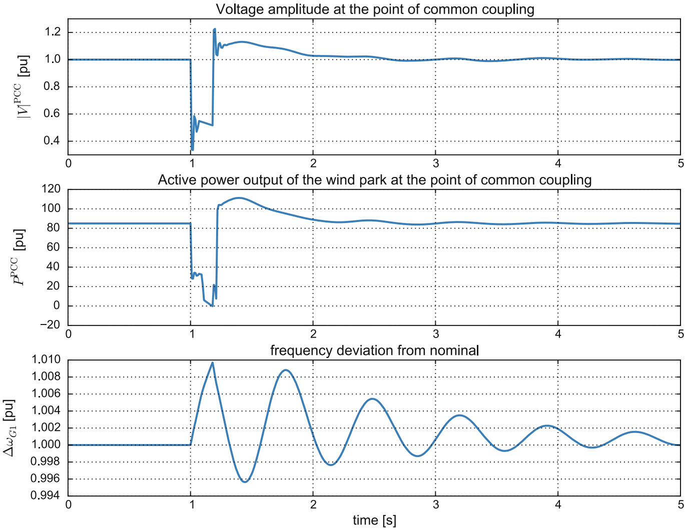


Fig. 5.12 FMI-based co-simulation of a transmission system and full converter based wind turbines; time domain response of a wind park riding through a remote short circuit

The bookkeeping of the interface variables is also accounted for by the master algorithm. Each synchronisation step the grid simulator needs the d–q projection of the static generator reference currents as an input from the wind turbine models, whereas the wind turbine models need the voltage at the point of common coupling ($|V|^{\mathrm{PCC}}$) as an input from the grid simulator.

For one of the tests from outlined the design (nearby fault, interruption after 180 ms) the time domain simulation results are shown in Fig. 5.12. It can be seen that during the fault, the power output of the wind park blocks and the frequency starts to increase as a consequence. After fault clearance the voltages, frequency, and power oscillate and eventually restore to their nominal values. The wind power plant remains connected during the entire simulation yielding a positive test result. During the fault it can be noticed that the voltage amplitude exhibits spikes immediately after fault ignition and clearance, which is probably of numerical nature and caused by the co-simulation experimental setup. This can be resolved by adjusting the synchronisation intervals and/or interface variable handling, and corresponds to step ⑥ in Fig. 5.8. This also holds for testing assumptions made earlier about the aggregation of the wind turbine model at plant level. The outlined holistic testing approach very flexibly enables amendments of the tests and experiments while conserving specifications and definitions at a higher (conceptual) layer.

References

1. P. Palensky, A.A. van der Meer, C.D. López, A. Joseph, K. Pan, Cosimulation of intelligent power systems: fundamentals, software architecture, numerics, and coupling. IEEE Indust. Electron. Mag. **11**(1) (2017)
2. P. Palensky, A.A. van der Meer, C.D. López, A. Joseph, K. Pan, Applied cosimulation of intelligent power systems: implementing hybrid simulators for complex power systems. IEEE Indust. Electron. Mag. **11**(1) (2017)
3. S. Scherfke, S. SchÃtte, *Mosaik-Architecture Whitepaper* (2012), https://mosaik.offis.de/publications/
4. C.D. López, A.A. van der Meer, M. Cvetković, P. Palensky, A variable-rate co-simulation environment for the dynamic analysis of multi-area power systems, in *2017 IEEE Manchester PowerTech* (2017), pp. 1–6
5. ENTSO-E, Common Grid Model Exchange Specification (CGMES), Version 2.5, Draft IEC 61970-600 Part 1, 2nd edn. (2016)
6. ENTSO-E Operational Data Quality Taskforce. Quality of Datasets and Calculations for System Operations, 3rd edn. (2015). Accessed https://docstore.entsoe.eu/ June 2018
7. H. Krishnappa, *Model Validation and Feasibility Analysis of Modelica based Dynamic Simulations using OpenIPSL and CGMES MSc* (Delft University of Technology, Thesis, 2017)
8. K.J. Aström, H. Elmqvist, S.E. Mattsson, Evolution of continuous-time modeling and simulation, in *The 12th European Simulation Multiconference, ESM'98* (Manchester, UK, 1998), pp. 16–19
9. https://fmi-standard.org/
10. M. Cvetković, H. Krishnappa, C.D. López, R. Bhandia, J. Rueda Torres, P. Palensky, Co-simulation and dynamic model exchange with consideration for wind projects, in *Berlin Wind Integration Workshop* (2017), pp. 1–6
11. ERIGrid Project, https://erigrid.eu/
12. J. Trefke, S. Rohjans, M. Uslar, S. Lehnhoff, L. Nordstrom, A. Saleem, Smart grid architecture model use case management in a large European smart grid project, in *Innovative Smart Grid Technologies Europe (ISGT EUROPE)* (Copenhagen, Denmark, 2013), pp. 6–9
13. A.A. van der Meer, P. Palensky, K. Heussen, D.E. Morales Bondy, O. Gehrke, C. Steinbrink, M. Blank, S. Lehnhoff, E. Widl, C. Moyo, T.I. Strasser, V.H. Nguyen, N. Akroud, M.H. Syed, A. Emhemed, S. Rohjans, R. Brandl, A.M. Rohjans, Cyber-physical energy systems modeling, test specification, and co-simulation based testing, in *Workshop on Modeling and Simulation of Cyber-Physical Energy Systems* (Pittsburgh, 2017)
14. M. Blank, S. Lehnhoff, K. Heussen, D.E.M. Bondy, C. Moyo, T. Strasser, Towards a foundation for holistic power system validation and testing, in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)* (2016), pp. 1–4
15. International Electrotechnical Comission, *IEC 62559-2:2015 - Use Case Methodology*, https://webstore.iec.ch/publication/22349
16. A.A. van der Meer, C. Steinbrink, K. Heussen, D.E. Morales Bondy, M.Z. Degefa, F. Pröstl, T. Strasser, S. Lehnhoff, P. Palensky, Design of experiments aided holistic testing of cyber-physical energy systems, in *Proceedings of the Modeling and Simulation of Cyber-Physical Energy Systems* (Porto, Portugal, 2018)
17. Modelisar, Functional mock-up interface for co-simulation, MODELISAR (ITEA 2 - 07006), Technical report (2010)
18. The FMI++ Library, http://sourceforge.net/projects/fmipp/, Accessed Feb 2018