Distributed Co-Simulation of Networked Hardware-in-the-Loop Power Systems

Nauman Beg[1], Moiz Ahmed[1], Karen Derendorf[1], Frank Schuldt[1], Stefan Geißendörfer[1]
nauman.beg@dlr.de, moiz.ahmed@dlr.de, karen.derendorf@dlr.de, frank.schuldt@dlr.de, stefan.geissendoerfer@dlr.de

[1] German Aerospace Center (DLR), Institute of Networked Energy Systems, Oldenburg, Germany

Abstract— **The paper presents a method to extend existing co-simulation frameworks to emulate quasi-dynamic behavior of real grid components in a grid simulator using a distributed networked co-simulation platform. The platform uses generic socket communication to exchange data between real grid components and the grid simulator. The framework utilizes event triggered models to enable data exchange between grid components and the platform over user datagram protocol (UDP)/IP interface. This framework has its use-case in smart grids as well as microgrids for analyzing, monitoring and control applications where it is impractical to model each component separately inside the grid simulator. Intrinsic communication delays including jitter is handled using built-in fallback strategies inside the framework itself. As a proof-of-concept, a co-simulation between a simulated low-voltage (LV) grid model and an emulation of a simplified Photovoltaic (PV) model is presented. The behavior of PV emulator is integrated in the grid simulator using socket communication. The primary focus of this work is to validate the extended framework. The effect of network delays on the stability of the distributed co-simulation setup are also investigated.**

Index Terms— **Hardware-in-the-Loop (HiL), distributed systems, co-simulation, Real-time (RT)**

I. INTRODUCTION

The Energy transformation process with its ambitious goal to achieve climate neutral Europe by 2050 through integration of distributed generation (DG) on all voltage levels poses new challenges for system dynamics and operation. Consequently, the existing electrical power grid has to be reinforced accordingly. Analysis of decentralized systems is necessary for synchronous operation of future electrical grids and appropriate investigations are required on individual component as well as on system level.

Simulation studies are an essential tool for engineers, power system planners as well as system developers in the evolution of power systems [1]. Established simulation tools and techniques in [2], [3] are widely available to analyze system behavior and contingencies for several operational scenarios. These tool-sets are widely used for monolithic simulations i.e. one model - one solver configuration. Existing monolithic environments are limited in their application for analyzing multi-domain problems including analysis of coupled grid problems, technology specific modeling of Intelligent Electronic Devices (IEDs) and Information and Communication Technology (ICT) infrastructure. In this regard, co-simulation frameworks allow modeling and coupling of sub-systems in their specific domains [4].

In [5], a co-simulation architecture is introduced with emphasis on analyzing effects of Power-to-X in electrical grids. The approach incorporates physical storage systems with simulation models running in Real-time (RT). The implemented control strategy enables successful co-simulation for such a multi-modal energy system. In [6], MOSAIK platform is presented for synchronized co-simulation of cyber-physical energy systems i.e. power systems embedded with communication network. In [7], it is demonstrated that increasing the number of IEDs including smart sensors in the energy system will increase the observability and hence the intelligence of power systems. Co-simulation frameworks can help analyzing necessary interactions in such systems.

Conventional co-simulations are mostly executed in time-step resolution between [ms–sec] and may not be sufficient for investigating fast dynamics of a power system such as frequency and switching transients. RT X-in-the-loop simulations have played an important role for analyzing transient system stability. Depending on the object under test (power hardware, controller, software, grid) the term X in X-in-the-loop can be appropriately replaced. Various literature describes the application and advantages of conventional RT simulations coupled with hardware [8]–[13]. The application of RT Power Hardware-in-the-Loop (PHiL) also extends to simulation of geographically distributed systems and hence enhances the existing capabilities of smart-grid labs infrastructure [14]. The spatially distributed systems can also interact over Wide Area Network (WAN) interface.

It is evident that although co-simulation offers several advantages in terms of its modular nature and flexibility, still modeling of sub-systems is challenging and requires detail know-how of the underlying platform. In this work, a simplistic approach is proposed to integrate the quasi-dynamic behavior of real grid components in grid simulators by extending the existing co-simulation frameworks with sub-system descriptors and socket handles. This approach has its advantage for integrating the behavior of real components without detailed modeling. As proof-of-concept, a RT PV emulator is coupled with a synthetic grid model in a distributed co-simulation setup to identify the feasibility and effect of network jitter on the stability of the proposed method. For this setup, the architecture of Mosaik co-simulation framework is

979-8-3503-9678-2/23/$31.00 ©2023 IEEE

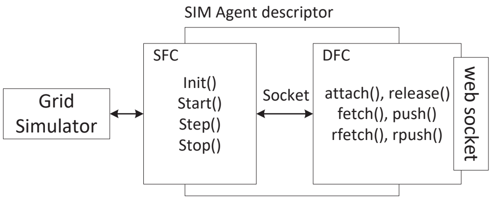

Fig. 1. Implemented methods in SIM agent descriptor

extended with individual type descriptors for the grid simulator and PV emulator. Section II describes the details of the necessary modifications proposed in the co-simulation framework. Section III outlines the test-setup as well as use-case for demonstration. Key performance metrics for the individual scenarios in the presented use-case are summarized in section IV whereas highlights of the proposed method are concluded in section V.

II. METHODOLOGY

To enable co-simulation of networked distributed systems including HiL sub-systems, existing frameworks can be extended by incorporating the following type descriptors in the individual subsystems:

A. *Sim agent descriptor (SIM)*

This descriptor defines control and data communication structure between a grid simulator and the co-simulation platform and has two subsequent core definitions:

1) Simulation flow control (SFC)
2) Data flow control (DFC)

Simulation flow control describes model setup, model initialization as well as simulation step resolution and time-step advance of a grid simulator. In Fig. 1, the model initialization is realized with Init() method whereas Start(), Step() and Stop() methods control the simulation flow of a grid simulator.

Data flow control on the other hand regulates data exchanges between a grid simulator and the co-simulation platform at regular co-simulation time-steps. In Fig. 1, this is implemented with fetch() and push() methods in the DFC descriptor. The attach() and release() methods are implemented to communicate with a remote grid simulator / HiL emulator. The methods rfetch() and rpush() fetches and push data from / to remote simulators respectively.

SIM agent object is a time triggered module in the platform and is invoked at regular co-simulation time-steps.

B. *HiL agent descriptor*

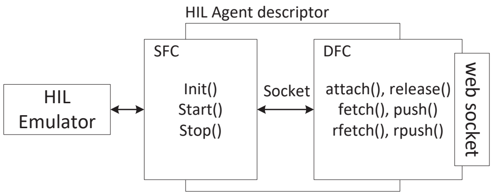

Fig. 2. Implemented methods in HiL agent descriptor

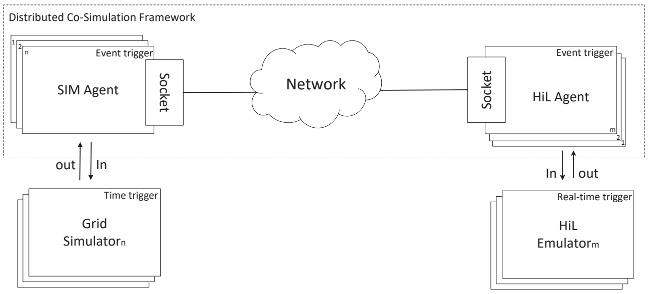

Fig. 3. Overview of distributed co-simulation framework

This descriptor defines control and data communication between a HiL emulator and the co-simulation platform. The HiL descriptor as shown in Fig. 2 has similar structure as for the SIM agent descriptor with only one difference in its SFC implementation where Step() method is not required since HiL emulator runs in RT and does not require time-step advance manually.

HiL agent object is an event triggered module in the platform and is invoked by the SIM agent data request query through the co-simulation platform.

C. *Sock handle*

Sock handles extend the core functionality of a co-simulation platform by enabling communication between individual agents over network. The inter-agent communications are handled with web sockets as shown in Fig. 1 and Fig. 2. Each sock handle relays necessary data and control messages between SIM / HiL agents and the platform over UDP interface. The sock handles are implemented as an inherent feature of individual SIM / HiL descriptors. A global timeout callback function with agent data retention is implemented to compensate communication delays and outages in individual agents.

The individual type descriptors together with the sock handles extend the functionality of existing co-simulation frameworks for spatial distributed systems including HiL subsystems. An overview of the extended co-simulation framework is shown in Fig. 3.

III. SETUP OF DISTRIBUTED CO-SIMULATION FRAMEWORK

The agent descriptors described in section II are implemented as abstract derived classes in Mosaik Smart grid co-simulation framework in its high-level Application Programmable Interface (API) in Python [15]. To study the effect of network jitter on the co-simulation step resolution, a use-case is defined where a distributed co-simulation between a synthetic Root-Mean-Square (RMS) grid model developed in

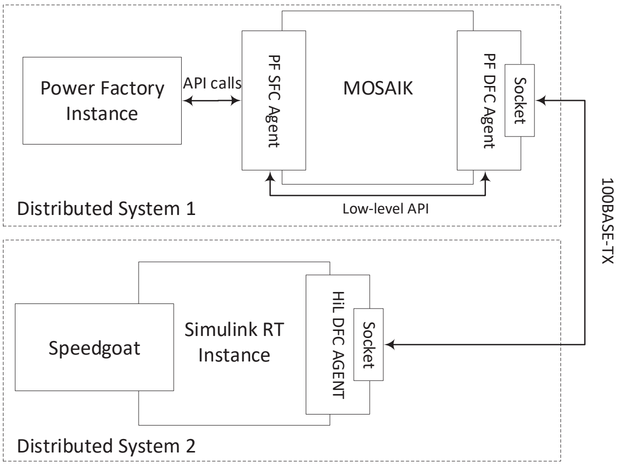

Fig. 4. Model setup for distributed co-simulation

TABLE I
DESCRIPTION OF SIMULATED SCENARIOS

Scenario Nr.	Δt_{co-sim} (ms)	Δt_{sim}	Setup
0	NA	$1\ (ms)$	monolithic
1	100		distributed co-simulation
2	80		
3	40		
4	20		

DIgSILENT PowerFactory (PF) [16] and PV emulator developed on Speedgoat in Simulink Real-Time is implemented. The model setup for Distributed systems (DSs) is presented in Fig. 4

A PF SFC (SIM descriptor) agent is initialized to handle communications between PowerFactory instance and the platform whereas PF DFC (SIM descriptor) and remote HiL DFC (HiL descriptor) agents are initiated to co-ordinate communication between PV emulator and the PF simulator at co-simulation step intervals. The distributed systems are linked with 100BASE-TX 802.3u network interface.

IV. SIMULATIONS AND RESULTS

To benchmark the distributed co-simulation framework, a reference model in PowerFactory as shown in Fig. 5 is simulated in both monolithic and in distributed co-simulation environment where a DG unit with a $P-f$ droop constant $s_{DG} = 5\%$ as shown in Fig. 6 and grid in-feed P_{DG} is connected to a distribution grid. The droop characteristic of DG simulates the simplified PV model. The grid has p.u. voltage and frequency values given by v_{pu} and f_{pu} respectively. The DG power is measured for both topologies: a). monolithic and b). distributed co-simulation environment. In topology a, the control model for DG is programmed locally inside a PowerFactory DIgSILENT Simulation Language (DSL) model and is taken as the reference case for benchmark. In topology b, the control model is programmed in a distributed instance of Simulink Real-Time and the models are synchronized through Mosaik framework. A frequency disturbance is introduced at time $t_{RT} = 2s$ where the grid frequency is increased to 1.03 p.u. The total RT simulation is fixed to $3s$ whereas the simulation time-step (Δt_{sim}) in PF is fixed to $1ms$. In topology b, co-simulation for 4 distinct scenarios as defined in Table I with different co-simulation step-size are investigated. Scenario 0 represents the monolithic case (topology a) which is defined as base case for benchmark. The co-simulation initialization is handled by the SFC descriptor instances as shown in the co-simulation flow sequence diagram in Fig. 7. The communication between both PF and Simulink-Real-Time instances are handled by DFC descriptor instances respectively.

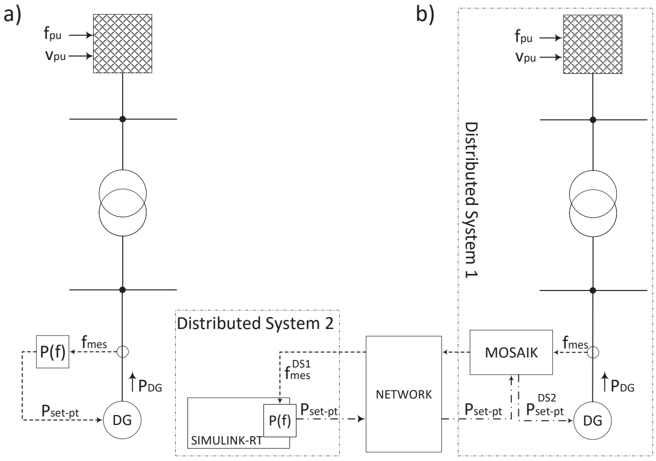

Fig. 5. Simulation topology : a). Monolithic b). distributed co-simulation

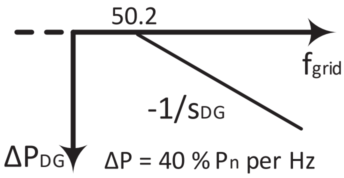

Fig. 6. DG $P-f$ droop characteristic curve [17]

Measured grid transient frequency (f_{mes}), transmitted grid frequency (f_{mes}^{DS1}) from DS1 to DS2 and the respective DG in-feed (P_{DG}) in Fig. 5 is plotted for RT and presented in Fig. 8. The step-size for co-simulation is parameterized according to the scenario definition in Table I to measure the Round-trip delay (RTD) and to identify the limiting case for the network jitter.

In Fig. 8, reducing co-simulation step-size in higher index scenarios increases system latency (average elapsed RT between two simulation time-steps in PF) which delays both measured (f_{mes}) and command (P_{set-pt}) response in DG w.r.t the base case (Scenario 0). The reason for increased system latency is the frequent data exchanges over network between the DS1 and DS2 in Fig. 5. It is also worth mentioning that the system latency is strongly related to the quality of the underlying network link and the number of co-dependent participating agents in the co-simulation setup.

Although lower co-simulation step-size increases system latency, it improves data resolution between DSs that can be a critical factor for high bandwidth distributed control applications. This can be observed in Fig. 9 where higher number of f_{mes} data points are communicated between PF and Simulink-RT instances in higher index scenarios that improves the controllability of the DG in-feed. Hence, there always exists a trade-off between the system latency and the observability of co-simulation sub-systems in a distributed environment and is a key design parameter for a given application.

Scenario 4 in Table I describes the limiting case for the investigated test-setup where the required network latency is lower than the underlying network jitter. This makes the communicated frequency (f_{mes}^{DS1}) from DS1 to DS2 as well as the command signal received for DG from DS2 in DS1 unpredictable. In such cases, the retention fallback strategy incorporated in the extended framework relays the last valid value of signals between DS instances. Hence, the DG holds its P_{set-pt} in the absence of command signal in Fig. 8. The network jitter is specific to the test-setup and hence must be estimated for individual application.

RTD between monolithic and the co-simulation environment is defined as a real-world performance metric that quantifies the overall latency of a co-simulation setup in milliseconds. The RTD is evaluated by calculating the time-value pair offset between a monolithic simulation and its equivalent co-simulation setup. 0ms RTD represents no performance degradation of co-simulation setup in comparison to a monolithic setup in terms of real-world time. The RTD values for the discussed use-case are summarized in Table II. An inverse relationship exists between the co-simulation step-size and RTD as shown in Fig. 10. A smaller co-simulation step-size results in larger RTD and vice-versa due to frequent data exchanges over platform. A larger RTD also represents higher system latency of RT sub-systems in a distributed co-simulation environment. The optimum co-simulation step-size (Δt_{co-sim}) can be selected from the evaluated RTD values and the required latency for a specific application. RTD values describe aggregated overhead introduced by the PF SFC and DFC descriptors including latency of the low level API of the co-simulation platform in each scenario.

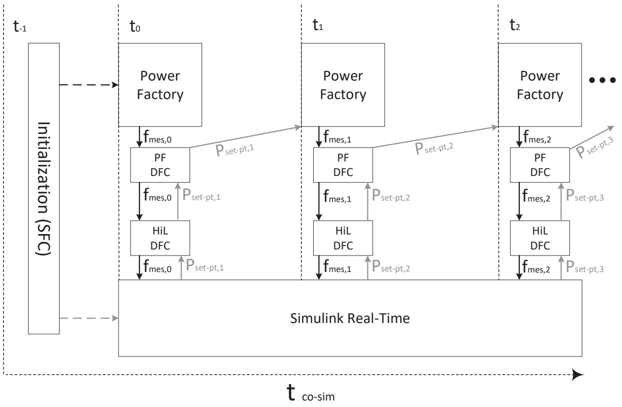

Fig. 7. Sequence flow diagram for distributed co-simulation use-case

TABLE II
RTD IN CO-SIMULATION SCENARIOS

Scenario Nr.	RTD (ms)
1	4
2	7
3	44
4	79

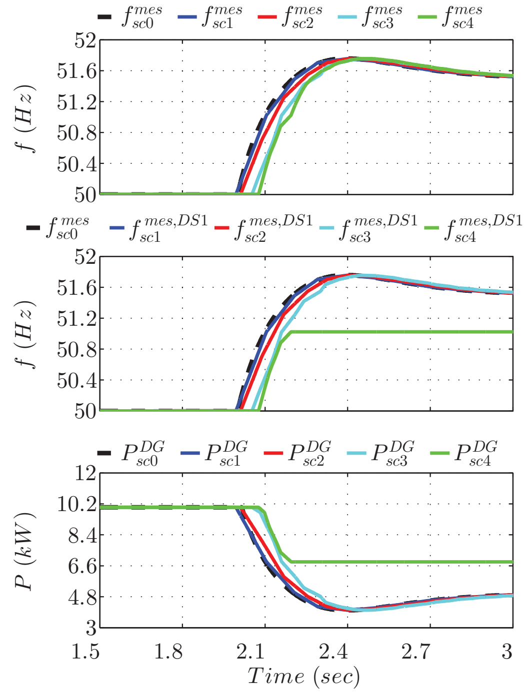

Fig. 8. Results for distributed co-simulation

V. SUMMARY AND CONCLUSIONS

In this work, a method to extend existing co-simulation frameworks with socket communication is presented to incorporate the quasi-dynamic behavior of spatially distributed sub-systems in grid simulators. The extended architecture relies on UDP based agent descriptors to handle communication between the co-simulation platform and the individual sub-systems. The functionality of the proposed architecture is verified in a test-setup with a simplified grid simulation model and a RT instance of a PV emulator. From the evidence of the co-simulated use-case, conclusions on the feasibility and application of the proposed method are presented as follows:

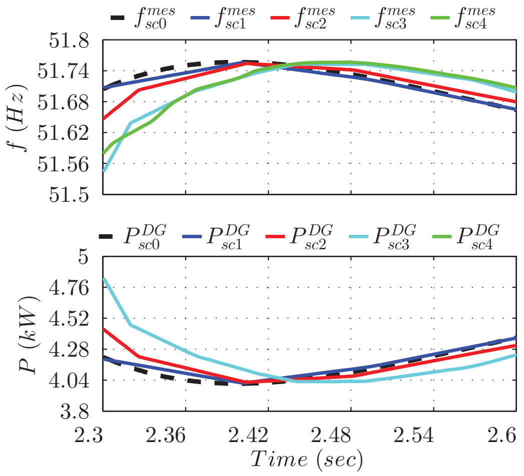

Fig. 9. Results for distributed co-simulation (peak values)

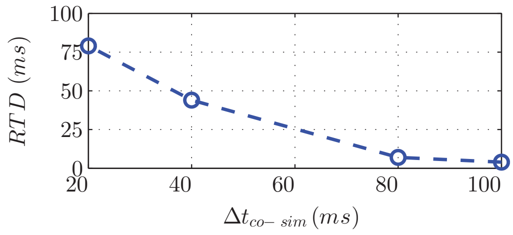

Fig. 10. RTD vs co-simulation step-size

- The proposed architecture is feasible to incorporate slow time varying dynamics of spatially distributed real systems in the grid simulators. The framework is prone to network delays and hence is practical for limited bandwidth distributed control applications with RT subsystems.
- A trade-off exist between the observability and latency of the processes that is identified with benchmarking the distributed co-simulation setup against a reference monolithic environment. The co-dependency between participating sub-systems effects it further.
- A simple fallback strategy is successfully tested against unreliable communication between distributed systems to improve the robustness and stability of the co-simulation setup. The focus of any such strategy should be to improve the stability of individual sub-systems in absence of communication infrastructure.
- The network delays are very specific to test-setup topology that requires thorough investigations for individual use-case and application.

Study under network congestions are beyond the current scope of this work and requires further investigations with statistical considerations.

REFERENCES

[1] R. Kuffel, P. Forsyth, and C. Peters, “The Role and Importance of Real Time Digital Simulation in the Development and Testing of Power System Control and Protection Equipment,” IFAC-PapersOnLine, Volume 49, Issue 27, 2016, Pages 178-182. doi: 10.1016/j.ifacol.2016.10.739

[2] Y. Kumar, V. K. Devabhaktuni, and S. Vemuru, ”Comparison of power system simulation tools with load flow study cases,” 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, USA, 2015, pp. 290-294. doi: 10.1109/EIT.2015.7293355.

[3] B. Badrzadeh, Z. Emin, E. Hillberg, D.A. Jacobson, L.H. Kocewiak, G. Lietz, F. F. da Silva, and M.V. Escudero. “The Need or Enhanced Power System Modelling Techniques and Simulation Tools.” (2020).

[4] I .B. O. Chagas, and M. A. Tomim, “Co-simulation applied to power systems with high penetration of distributed energy resources,” Electric Power Systems Research, Volume 212, 2022. doi: 10.1016/j.epsr.2022.108413

[5] H. Çakmak, A. Erdmann, M. Kyesswa, U. Kühnapfel, and V. Hagenmeyer,”A new distributed co-simulation architecture for multi-physics based energy systems integration: Analysis of multimodal energy systems” at - Automatisierungstechnik 67, no. 11 (2019): 972-983. doi: 10.1515/auto-2019-0081

[6] C. Steinbrink, M. Blank-Babazadeh, A. El-Ama, S. Holly, B. Lüers, M. Nebel-Wenner, R. P. R. Acosta, T. Raub, J.S. Schwarz, S. Stark, A. Nieße, and S. Lehnhoff. 2019. ”CPES Testing with mosaik: Co-Simulation Planning, Execution and Analysis” Applied Sciences 9, no. 5: 923. doi: 10.3390/app9050923

[7] P. Palensky, A. A. Van Der Meer, C. D. Lopez, A. Joseph, and K. Pan, ”Cosimulation of Intelligent Power Systems: Fundamentals, Software Architecture, Numerics, and Coupling,” in IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 34-50, March 2017. doi: 10.1109/MIE.2016.2639825.

[8] P. Kotsampopoulos, V. Kleftakis, G. Messinis, and N. Hatziargyriou, ”Design, development and operation of a PHIL environment for Distributed Energy Resources,” IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, Canada, 2012, pp. 4765-4770. doi: 10.1109/IECON.2012.6389005.

[9] C. Dufour, and J. Bélanger, ”On the Use of Real-Time Simulation Technology in Smart Grid Research and Development,” in IEEE Transactions on Industry Applications, vol. 50, no. 6, pp. 3963-3970, Nov.-Dec. 2014. doi: 10.1109/TIA.2014.2315507.

[10] X. Guillaud et al., ”Applications of Real-Time Simulation Technologies in Power and Energy Systems,” in IEEE Power and Energy Technology Systems Journal, vol. 2, no. 3, pp. 103-115, Sept. 2015. doi: 10.1109/JPETS.2015.2445296.

[11] M. Lemaire, P. Sicard, and J. Belanger, ”Prototyping and Testing Power Electronics Systems Using Controller Hardware-In-the-Loop (HIL) and Power Hardware-In-the-Loop (PHIL) Simulations,” 2015 IEEE Vehicle Power and Propulsion Conference (VPPC), Montreal, QC, Canada, 2015, pp. 1-6. doi: 10.1109/VPPC.2015.7353000.

[12] P. Kotsampopoulos et al., ”A Benchmark System for Hardware-in-the-Loop Testing of Distributed Energy Resources,” in IEEE Power and Energy Technology Systems Journal, vol. 5, no. 3, pp. 94-103, Sept. 2018. doi: 10.1109/JPETS.2018.2861559.

[13] M. D. Omar Faruque et al., ”Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis,” in IEEE Power and Energy Technology Systems Journal, vol. 2, no. 2, pp. 63-73, June 2015. doi: 10.1109/JPETS.2015.2427370.

[14] S. Vogel, T.H. Nguyen, M. Stevic, T. Jensen, K. Heussen, V. Rajkumar, and A. Monti, (2020). Distributed Power Hardware-in-the-Loop Testing Using a Grid-Forming Converter as Power Interface. Energies. 13. 3770. doi: 10.3390/en13153770.

[15] S. Schutte, S. Scherfke, and M. Troschel, ”Mosaik: A framework for modular simulation of active components in Smart Grids,” in IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS), 2011, Brussels, Belgium, Oct. 2011. doi: 10.1109/SGMS.2011.6089027.

[16] “PowerFactory – Power System Software & Engineering”, Digsilent. [Online]. Available: https://www.digsilent.de/en/powerfactory.html

[17] ”Technische Richtlinie für Anschluss und Parallel-betrieb von Erzeugungsanlagen am Mittelspannungsnetz”, Ausgabe Juni 2008, Bundesverband der Energie- und Wasserwirtschaft e.V. (BDEW), Berlin, 2008.