Undergraduate Research in Iterative Power Flow using PowerModels and PowerWorld Simulator

Nathan Philipello, Jonathan M. Snodgrass, Thomas J. Overbye
Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX
{philipellon, snodgrass, overbye}@tamu.edu

***Abstract*—This paper introduces an iterative, hybrid framework for solving the AC Optimal Power Flow (OPF) problem by integrating PowerWorld Simulator and PowerModels within a unified Python-driven workflow. The approach establishes a closed-loop optimization pipeline that uses PowerWorld for system modeling, exports solved power flow states to MATPOWER format, solves a nonlinear AC-OPF using PowerModels in Julia, and updates generator setpoints back into PowerWorld for the next iteration. Results demonstrate strong consistency between the simulation and optimization environments, validating the bidirectional data exchange and offering a practical template for hybrid OPF studies in complex grid environments.**

***Index Terms*—AC Optimal Power Flow, PowerWorld Simulator, PowerModels, MATPOWER, iterative optimization, tool integration, power system simulation**

I. Introduction

THE modern electric power grid faces unprecedented challenges due to the increasing integration of renewable energy sources, growing electricity demand, and the need for enhanced grid reliability. Optimal Power Flow (OPF) is a fundamental optimization problem in power systems engineering that aims to determine the most efficient operating conditions for generators, loads, and transmission elements while satisfying physical and operational constraints. Traditional OPF formulations often rely on linearized approximations, such as DC-OPF, which can overlook critical nonlinear aspects such as voltage magnitudes and reactive power flows. To address these limitations, advanced tools and hybrid methodologies are essential for accurate and scalable OPF solutions.

This paper presents an iterative framework that integrates a commercial power flow (CPF) tool and PowerModels to solve AC-OPF problems. Here the particular CPF is PowerWorld Simulator Version 24. CPF serves as the primary simulation environment for modeling detailed power system cases. Solved data are then transferred to MATPOWER [1], which facilitates export in a standardized format. PowerModels leverages Julia's high-performance optimization capabilities to compute an OPF solution, after which the updated system states are fed back into CPF, repeating the loop. This integration enables repeated refinement of system states, converging toward optimal operating points that respect both steady-state power flow equations and economic objectives.

The primary research contribution of this work is the development of a reproducible, automated pipeline that bridges commercial simulation and open-source optimization tools for AC-OPF studies. Specifically, the contributions include: (1) a seamless Python-based pipeline for tool interoperability between PowerWorld Simulator and PowerModels, (2) automated generator mapping to ensure accurate bidirectional data transfer despite differing internal indexing schemes, and (3) empirical validation of model consistency through iterative convergence analysis. While this framework emerged from an undergraduate research project, the methodology itself represents a practical advancement for researchers seeking to leverage the complementary strengths of commercial simulators and open-source optimization libraries.

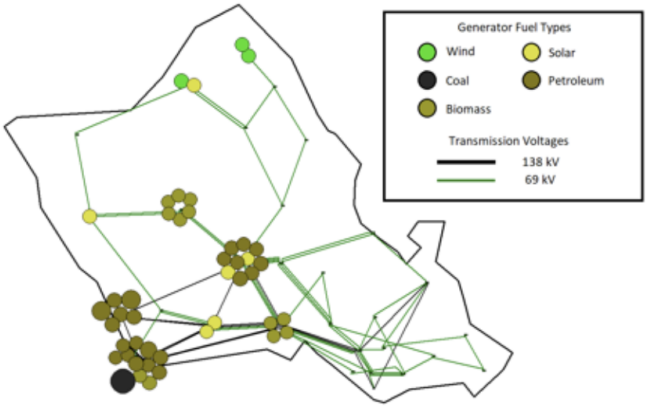

Fig. 1. Hawaii 40-Bus Case Oneline

Our approach is demonstrated on the Hawaii 40-bus test case [2] shown in Figure 1 a realistic representation of an island grid with diverse generation mix. This test case is publicly available in the repository described in [3]. By iterating between simulation, data extraction, OPF resolution, and state updates, the framework achieves validated dispatch consistency and constraint satisfaction over multiple cycles.

The remainder of the paper is organized as follows. Section II reviews related co-simulation and tool integration ap-

proaches in power systems. Section III describes the proposed methodology and tool integration. Section IV presents experimental results and discussions. Finally, Section V concludes the paper and outlines future work.

II. Related Work

Integrating commercial simulation software with flexible, scriptable environments has opened new doors for power system analysis, particularly in the context of competitive electricity markets. The works in [4] and [5] exemplify this approach through their coupling of PowerWorld and MATLAB®. As described in [5], a framework for agent-based modeling is implemented in which the complex behaviors of market participants, including generators and consumers, are executed within MATLAB®. This environment handles agent logic, bidding strategies, and market interactions. In parallel, CPF is used for its robust capabilities to simulate the underlying physical power system and to calculate important market outcomes such as locational marginal prices (LMPs).

A key achievement of this integration is its ability to capture both the economic and the physical dimensions of market simulation, a combination that neither tool accomplishes fully in isolation. By orchestrating communication between MATLAB® and CPF (often via file exchange or API interfacing), the researchers created a loose coupling that allowed MATLAB® to direct market actions and CPF to validate those actions against real power system constraints. The result is a simulation environment that mirrors the complexities of actual power markets, where economic decisions must always contend with the realities of grid physics. Building on this foundation, their 2015 paper extends the methodology to more advanced market operations such as optimal dispatch and unit commitment [5]. In this work, MATLAB® is employed to solve challenging optimization problems, using mathematical programming techniques to generate generator schedules and dispatch instructions. These solutions are then passed to CPF, which applies them within detailed power flow models, verifying that proposed market results are feasible given the electrical properties of the transmission network.

This established integration between CPF and MATLAB® provides a useful blueprint for the type of tool integration we are proposing with CPF and PowerModels. In our case, PowerModels (implemented in Julia) plays a role analogous to MATLAB® in [4], [5]: it offers flexible optimization routines for market clearing, dispatch, and related decision problems, while CPF contributes detailed AC power flow analysis, contingency assessment, and visualization. Adopting a similar loosely coupled approach—where market logic and optimization are handled in PowerModels and physical validation occurs in CPF—promises a rich and flexible research environment. The PowerWorld–MATLAB® works show how careful data exchange and clear separation of responsibilities between tools can create powerful synergies; our work builds directly on this idea, but updates it to an open-source optimization ecosystem and to the modern market.

Beyond these specific examples, the broader power systems community has begun to formalize such multi-tool integrations through general-purpose co-simulation frameworks. A prominent example is the HELICS platform, which is designed to support high-performance co-simulation across transmission, distribution, communication, and market domains [6]. HELICS introduces a standardized, federated architecture in which each simulator (or "federate") operates with its own time steps and numerical methods, while exchanging information through a coordinated messaging and time management layer. Conceptually, this generalizes the loose coupling philosophy seen in [4], [5] from a bespoke CPF–MATLAB® integration to a reusable infrastructure that can orchestrate many different kinds of tools.

A related strand of work focuses on transmission and distribution (T–D) co-simulation to capture the growing impact of distributed energy resources on system dynamics. For example, [7] proposes a T–D dynamic co-simulation framework that couples a transmission system simulator with a distribution system simulator to investigate frequency response contributions from distributed resources. Each simulator retains its own modeling assumptions and numerical solvers while exchanging boundary conditions such as interface power injections and bus voltages. This is conceptually similar to both the PowerWorld–MATLAB® integration and our proposed CPF–PowerModels setup: each subsystem is modeled in the tool best suited to its characteristics, and consistency is maintained through carefully designed data exchange at the interface.

Co-simulation has also been extended to encompass power systems and communication networks. In [8], a framework is introduced for co-simulation of power systems and communication infrastructures, where a power system simulator is coupled with a communications network simulator. The goal is to analyze how communication delays, congestion, or failures affect the performance of protection schemes, control algorithms, and other grid-supporting applications. As in the market and T–D cases, the simulators exchange key variables such as measurement data and control signals using a synchronization strategy that preserves temporal and causal consistency. Although the application domain in [8] is cyber-physical interaction rather than market operation, the underlying co-simulation principle is the same one we adopt: leverage specialized tools for different subsystems and integrate them through a carefully designed interface rather than attempting to capture all phenomena within a single simulator.

In parallel, open-source optimization frameworks for power systems have matured substantially, offering capabilities that naturally fit into co-simulation architectures. PowerModelsDistribution [9], for instance, extends the PowerModels ecosystem to support a variety of distribution power flow formulations. This framework provides a flexible environment for defining and exploring different models of distribution networks, including unbalanced and multi-phase representations. Although [9] focuses on modeling and solving distribution power flow problems within a single environment, the modular, solver-agnostic design of PowerModels and PowerModels-

Distribution makes them well suited to act as the optimization or analysis engine in a broader co-simulation setup. In such a configuration, PowerModels-based tools can compute optimal dispatch, reconfiguration, or voltage control strategies, which are then validated or applied within a separate power system simulator such as CPF.

Taken together, these strands of literature—from the original PowerWorld–MATLAB® integration [4], [5], through generalized co-simulation platforms like HELICS [6], transmission–distribution dynamic co-simulation [7], power–communication co-simulation [8], and open-source optimization frameworks for distribution systems [9]—build a coherent context for the CPF–PowerModels integration proposed in this work. They demonstrate that separating economic or optimization modules from detailed physical simulators, and then coordinating them through standardized interfaces, is both technically feasible and increasingly common. Within this landscape, our contribution is to instantiate these integration principles in a concrete architecture that couples a commercial AC power system tool with an open, extensible optimization framework (PowerModels), tailored to studies of competitive markets, dispatch, and system operation.

III. Methodology

CPF and PowerModels represent two distinct paradigms in power system analysis. CPF is a commercial, high-fidelity simulation platform widely used in industry for detailed contingency analysis, voltage stability assessment, and operator training. It employs a full Newton-Raphson power flow solver with robust convergence properties and supports complex system models including detailed generator dynamics, transformer tap control, and switched shunt devices. In contrast, PowerModels is an open-source optimization framework built in Julia, designed specifically to formulate and solve a broad class of OPF problems using mathematical programming techniques. It leverages the state-of-the-art nonlinear programming solver Ipopt to minimize generation cost subject to AC power flow equations and operational constraints. While CPF excels in simulation accuracy and visualization, PowerModels provides greater flexibility in objective functions, constraint modeling, and algorithmic customization.

It is important to note that the proposed methodology constitutes an offline, batch-oriented integration rather than a time-coordinated co-simulation. Unlike frameworks such as HELICS [6] that synchronize multiple simulators in real-time with coordinated time-stepping, our approach executes each tool sequentially within an iterative loop. This design choice prioritizes simplicity and reproducibility for steady-state OPF studies, where temporal dynamics are not the primary concern. The trade-off is that time-dependent phenomena such as transient stability or communication latency effects are not captured; however, for economic dispatch and optimal power flow applications, this offline integration provides a practical and effective solution pathway.

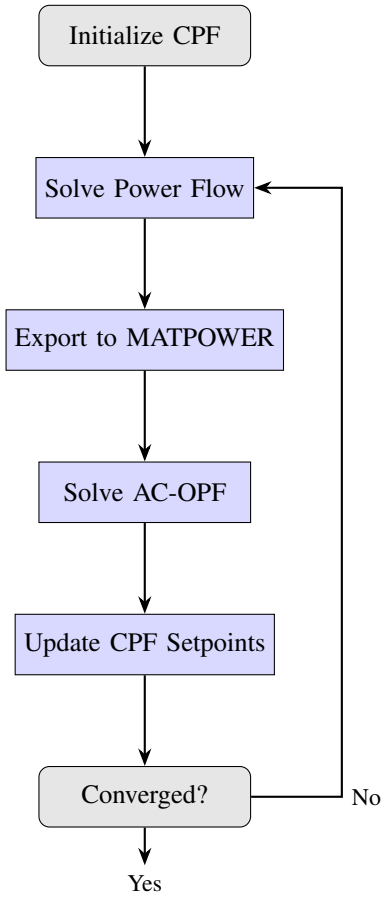

Fig. 2. High-level flowchart of the iterative CPF-PowerModels framework.

A critical challenge in integrating these tools lies in their differing internal representations of network components. CPF identifies generators using a composite key of bus number and generator ID (e.g., Bus 12, ID "1"), whereas PowerModels assigns sequential integer indices to generators during data parsing. These indexing schemes are not inherently aligned, and the order of generator enumeration can vary depending on how each tool traverses the network topology. This discrepancy prevents direct transfer of data values between the two environments. To resolve this, a generator map is constructed at the beginning of the workflow. This mapping is generated by querying both systems for generator attributes—bus number, ID, and rated capacity—and creating a lookup table that associates each CPF generator with its corresponding PowerModels index. This mapping ensures consistent data exchange throughout the iterative process.

The proposed methodology establishes a closed-loop iterative framework that couples the simulation capabilities of CPF with the optimization capability of PowerModels, as illustrated in Figure 2. The process begins in CPF, where an initial power flow solution is computed using the Newton-Raphson method. This step establishes a feasible operating point that respects voltage limits, line flow constraints, and generator reactive power capabilities. The resulting bus voltages, angles, generator outputs, and branch flows are extracted via CPF's SimAuto

interface using the GetParametersMultipleElement function. This data is then used to construct a standardized system representation compatible with the PowerModels solver.

Once the power flow converges in CPF, the state variables are exported to create a MATPOWER case file—a widely adopted format for power system data exchange. A custom Python function, writematpowercase, assembles the bus, generator, branch, and load data into the required .m file structure. During this conversion, generator mapping is applied to ensure that active and reactive power setpoints, and operating limits of each generator are correctly assigned to their corresponding indices in the MATPOWER model. The base MVA is set to 100, consistent with standard test cases, and all units are converted to per-unit where necessary. This step transforms the detailed CPF model into a compact, solver-ready format without loss of engineering constraints.

The MATPOWER case file is then passed to PowerModels, where an AC Optimal Power Flow (AC-OPF) is solved. Within the Julia environment, accessed via PyJulia, the case is parsed and reformulated as a nonlinear optimization problem using the PowerModels abstract modeling layer. The objective is to minimize total generation cost subject to full AC power flow equations, voltage magnitude limits (typically 0.95–1.05 p.u.), thermal line limits, and generator active/reactive power bounds. The Ipopt solver, a primal-dual interior point method, is employed due to its robustness with non-convex problems. Upon convergence, PowerModels returns a complete solution that includes optimal generator dispatches, bus voltages, and branch flows, along with diagnostic information such as termination status and objective value.

The OPF solution is written as a JSON file, which is then read back into the Python environment. Using the precomputed generator mapping, the optimal active power setpoints from PowerModels are translated into CPF-compatible (bus, ID) pairs. These values are injected into the CPF case via the SetParameter function, updating only the generator MW outputs while preserving all other model details. The case is saved, and the loop returns to the initial step: solving power flow in CPF with the new dispatch. This iterative process continues for a fixed number of cycles set by the user, allowing the system state to progressively align with the globally optimal dispatch computed by PowerModels while remaining grounded in the detailed simulation model of CPF.

Convergence is monitored by comparing generator MW outputs between CPF and PowerModels at each iteration. Data from both tools are logged in CSV format, and a convergence plot is generated to visualize the trajectory of selected generator setpoints. The framework terminates after the prescribed iterations or upon satisfaction of a convergence criterion (e.g., maximum dispatch difference < 0.1 MW). This hybrid approach combines the strengths of simulation-based validation and optimization-driven efficiency, offering a practical pathway for deploying OPF solutions in real-world systems using readily available commercial simulators.

Regarding scalability and practical deployment, the methodology itself is computationally scalable and fully automated once configured—no manual intervention is required during iterative execution. However, practitioners should be aware of software dependency constraints. The framework requires access to PowerWorld Simulator, a commercial product with associated licensing requirements, as well as a Julia installation with the PowerModels package and its dependencies (including the Ipopt solver). These prerequisites may limit adoption in environments where commercial software licenses are unavailable. Nevertheless, the core algorithmic approach could be adapted to other commercial or open-source power flow tools that provide scripting interfaces, potentially broadening accessibility in future implementations.

IV. Results

In evaluating the convergence behavior of the proposed hybrid CPF-PowerModels iterative OPF framework, we monitored the evolution of key control variables—specifically, the active power output (MW) of a selected generator and the voltage magnitude (p.u.) at a critical bus—across multiple iterations. For the base case configuration of the test system, the trajectories of these variables appeared as nearly flat lines, with negligible variation after the first iteration, as seen in Figures 3 and 4. At first glance, this behavior might suggest that the system is either already at optimality or that the test case is too small to demonstrate the value of iterative coordination. However, a deeper analysis reveals a more significant insight: the flat convergence profile is evidence of strong consistency between the CPF simulation model and the PowerModels optimization model.

The absence of oscillation or drift indicates that the initial power flow solution in CPF already lies within the feasible region defined by the AC-OPF constraints in PowerModels, and that the optimization layer does not identify economically or technically superior setpoints under the given objective and constraints. This alignment is not a limitation but a validation of data fidelity in the bidirectional interface: generator limits, voltage setpoints, branch parameters, and control modes (e.g., PV vs. PQ buses) are accurately synchronized between the two environments. When the cost coefficients, reactive limits, and network topology are identical, both solvers converge to the same operating point, confirming that the MATPOWER case export and SimAuto parameter updates preserve the full system model without distortion.

This observed consistency is particularly noteworthy given the complexities often encountered when bridging different simulation and optimization platforms. In practice, discrepancies may arise due to differences in how each tool interprets model parameters, enforces limits, or handles numerical tolerances. The seamless agreement seen here demonstrates that the hybrid framework can reliably transfer states and constraints without introducing artifacts or misalignments. This lays an essential foundation for future studies involving larger systems or cases with deliberately introduced data inconsistencies, contingencies, or non-standard operating scenarios—where the strength of iterative coordination and diagnostic value of the

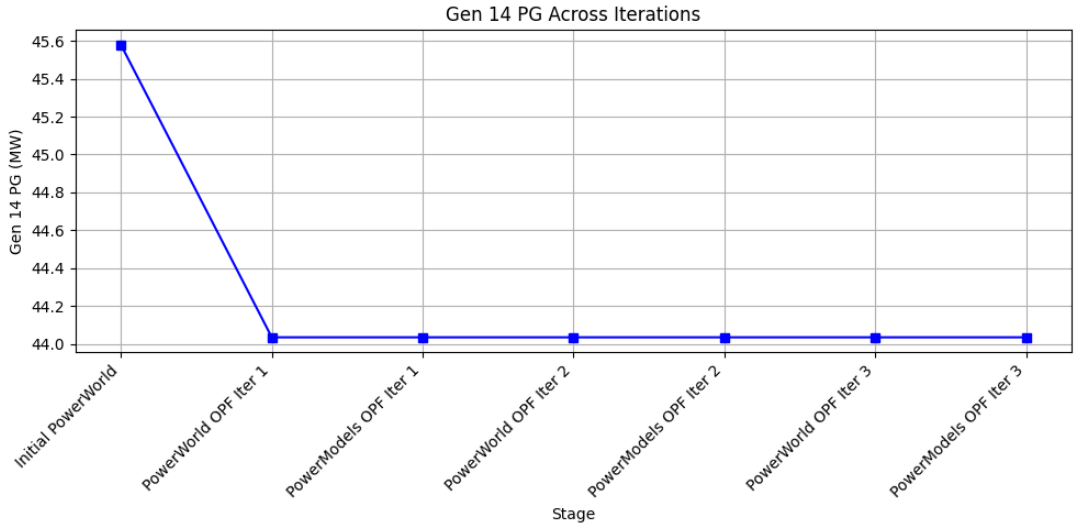

Fig. 3. Generator 14 MW Value Across Iterations

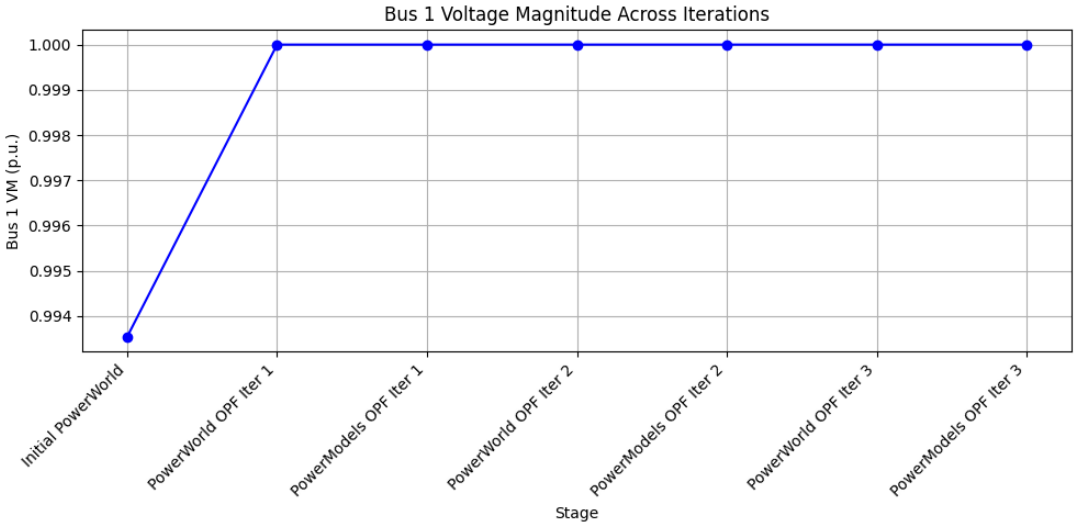

Fig. 4. Bus 1 Voltage Magnitude Value Across Iterations

framework will become more pronounced. Thus, the flat convergence profiles serve not only as validation in this instance, but also as a baseline for gauging system-wide impacts as complexity increases.

V. Conclusion

In conclusion, the proposed hybrid CPF-PowerModels iterative framework successfully demonstrates bidirectional consistency and convergence under controlled conditions. The results on the Hawaii 40-bus test case validate the accuracy of the data exchange pipeline, with flat convergence trajectories confirming model alignment between the commercial simulator and the open-source optimization tool. While the small-scale case is valuable for validation and debugging, it exhibits minimal dispatch adjustment in the base configuration, reflecting a well-tuned initial operating point rather than algorithmic necessity. Furthermore, certain test scenarios revealed practical challenges, including solver divergence in CPF following OPF updates, often triggered by reactive power limit violations or voltage control mismatches. These failures highlight a critical reality: even mathematically optimal setpoints from PowerModels may be physically unenforceable in a full-featured simulator due to discrete controls, islanding constraints, or external limit enforcement mechanisms not fully represented in standard OPF formulations.

Nevertheless, these observations do not diminish the framework's potential; they illuminate the path forward. Future work should focus on scaling the approach to larger, industrially representative systems (e.g., synthetic 118-bus or 2000-bus grids) where economic dispatch gradients and network

congestion create meaningful optimization margins. Robust handling of infeasibility—through warm-start refinement or integrated contingency-aware OPF—will be essential to ensure reliable convergence. Ultimately, this hybrid methodology lays a foundational bridge between power system simulation and optimization, paving the way for co-optimized, simulator-validated operational planning in complex, real-world grids.

REFERENCES

[1] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[2] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3258–3265, July 2017.

[3] S. Kunkolienkar, F. Safdarian, J. Snodgrass, A. Birchfield, and T. Overbye, "A description of the texas A&M university electric grid test case repository for power system studies," in *2024 IEEE Texas Power and Energy Conference (TPEC)*, College Station, TX, Feb. 2024.

[4] D. Zhang and S. Li, "Integrating powerworld and matlab for agent-based modeling and simulation of competitive electric power markets," in *2014 IEEE PES T&D Conference and Exposition*, 2014, pp. 1–5.

[5] S. Li and D. Zhang, "Integrating powerworld and matlab for optimal dispatch and unit commitment study of competitive electric power markets," *American Journal of Engineering and Applied Sciences*, vol. 8, pp. 291–301, 03 2015.

[6] B. Palmintier, D. Krishnamurthy, P. Top, S. Smith, J. Daily, and J. Fuller, "Design of the helics high-performance transmission-distribution-communication-market co-simulation framework," 04 2017, pp. 1–6.

[7] W. Wang, X. Fang, H. Cui, F. Li, Y. Liu, and T. J. Overbye, "Transmission-and-distribution dynamic co-simulation framework for distributed energy resource frequency response," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 482–495, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9569771

[8] W. Zhong and F. Milano, "A co-simulation framework for power systems and communication networks," 06 2019, pp. 1–6.

[9] D. Fobes, S. Claeys, F. Geth, and C. Coffrin, "Powermodelsdistribution.jl: An open-source framework for exploring distribution power flow formulations," 04 2020.