Evaluation of a Reduced-Order Model for IBR Fault Response Representation via OEM Blackbox Models

Owen Cornmesser[1], Soham Chakraborty[1], Rasel Mahmud[1], Jing Wang[1], Andy Hoke[1],
Paulo Henrique Pinheiro[2], Brian Johnson[2], Romulo Goncalves Bainy[2],
[1]Power Systems Engineering Center, National Renewable Energy Laboratory, Golden, Colorado, 80401, USA.
[2]Electrical and Computer Engineering, University of Idaho, Moscow, Idaho, 83844, USA.
{owen.cornmesser, rasel.mahmud, soham.chakraborty, jing.wang, andy.hoke}@nrel.gov,
{paulopinheiro, bjohnson, romulo}@uidaho.edu

***Abstract*—Driven by the need to capture the electromagnetic transients of transmission lines, inverter switching behavior, and detailed control systems, electromagnetic transient (EMT) studies have become increasingly important in industry, such as IBR interconnection study and fault study. However, original equipment manufacturer (OEM) inverter models typically include extensive parameters and proprietary settings that are unavailable to protection engineers. This paper introduces a data-driven, reduced-order model (ROM) developed as a PSCAD library component for use in EMT-based fault studies. The ROM replicates key OEM model behaviors without requiring detailed knowledge of control design or parameterization. The accompanying Python automation scripts streamline data generation, parameter fitting, and validation. The ROM's performance is demonstrated through comparison with both IEEE 2800-compliant and non-compliant OEM models in a real-world power system. Relay responses show nearly identical results, while simulation runtime is reduced by an average of 32.8%, highlighting the ROM's practicality for protection engineers.**

***Index Terms*—Fault study, reduced-order model, inverter-based resources, protection design**

I. INTRODUCTION

The increasing penetration of inverter-based resources (IBRs) has introduced new challenges for power system protection, as their fault response is fundamentally different from synchronous machines and strongly dependent on the embedded power electronic controls [1], [2]. Prompted by this, the use of electromagnetic transient (EMT) simulation has become increasingly common in industry, particularly among utilities, for both stability and protection studies [3]. Unlike traditional phasor-domain simulations, EMT tools can capture the fast transients, control interactions, and nonlinear dynamics that are critical for accurately assessing IBR fault behavior and designing reliable protection systems. However, understanding detailed IBR models requires deep expertise in power electronics, control theory, and OEM parameterization—knowledge that many protection engineers may not have [2]. As a result, these models are often impractical for routine protection studies. Moreover, the computational efficiency is very low when using OEM provided detailed IBR models, in an already computationally cumbersome environment of EMT simulation. To address this gap, there is a need for reduced-order, computationally efficient, detailed IBR fault models that can represent essential fault characteristics of OEM models without requiring detailed software and power electronics expertise, and at the same time remove ambiguity regarding the control settings and/or parameterization of OEM models. Limitations regarding the traditional look-up table methods for IBR short circuit representation and issues with incorrect and/or unavailable vendor model settings have been identified within the Power Systems Reliability Committee (PSRC) [1]. Therefore, they have a desire for a mathematical based alternative fault model.

To address this modeling gap, various fault models have been developed and explored. Looking into grid-following inverters (GFLs), [4] outlined a control structure through analysis in the dq reference frame. The authors in [5] proposed a reduced-order model (ROM) based on separating fast and slow acting states from the governing differential equations of the control structure. Meanwhile, [6] described a similar model reduction process based on singular perturbation theory and the dynamic equations. In all these fault models, the parameterization of the previous model was known. However, the parameterization of vendor models is proprietary and therefore typically unknown to the user. Thus, a data-driven approach is becoming more popular, as this allows any parameterization of a vendor model to be learned. References [7] and [8] proposed a data-driven ROM that builds the theoretical basis to develop such a model to capture IBR fault response during transients.

This paper moves further from [7] and [8] to develop a data-driven ROM as a PSCAD library component for protection engineers to capture accurate IBR fault response in EMT simulations. It allows for streamlined and simple implementation into their studies, removing the barriers of technical knowledge and unknown OEM model settings/parameters. In summary, the main contributions of this paper are to (1) introduce a data-driven ROM embedded in a fully developed PSCAD library component, enabling protection en-

This work was authored in part by NREL for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This material is based upon work supported by the U.S Department of Energy Office of Energy Efficiency and Renewable Energy under Solar Energy Technologies Office Agreement Number 41215. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government.

979-8-3315-8138-1

gineers to integrate an accurate IBR fault model without software/control/parameterization knowledge; (2) outline the streamlined process to utilize the PSCAD ROM component, which is entirely automated in Python (the full process including data generation, parameter fitting, uploading parameters into the ROM component, and comprehensive verification is fully automated); and (3) validate the PSCAD ROM component against two separate OEM models through the use of a protection relay model. The results show that the relay model sees the same fault responses between the ROM and OEM models, all while reducing simulation runtime by 32.8% on average.

II. Description of the ROM and Python Automation Script

The data-driven ROM shown in this paper originated from [7], which explains the methodology and framework of developing a GFL IBR model for IBR fault representation. Note the goal of this paper is to focus on the usability and accuracy of the PSCAD library component for use by protection engineers (e.g., software implementation/usage); further information on the mathematics and principles of the ROM will be detailed in a forthcoming paper. Nonetheless, we provide a high-level overview of the ROM as follows.

A. Overview of the ROM

The ROM is data-driven; after a training process, it utilizes a set of pre-calculated parameters during simulation. A pre-determined, nonlinear, parameterized equation uses this parameter set to emulate the fault current injection from a detailed GFL model. In order to obtain these parameters, a timeseries dataset of fault current and voltage is required; with this data, the best-fit parameters in the nonlinear equation are found for each fault's current injection. Here, parameters are found to mirror the injected current in the phasor domain (positive sequence magnitude and phase, and negative sequence magnitude and phase, referenced from here on as $|I_1|$, $\angle I_1$, $|I_2|$, $\angle I_2$, respectively).

After data collection, the machine-learning task is to process the relationship between deviations in parameters from fault to fault and deviations in terminal voltage from fault to fault. The relationship is encoded for each parameter by a set of sub-parameters. These sub-parameters are the final parameters used by the PSCAD component during simulation, labeled "Static Parameters" in Fig. 1.

The ROM measures the terminal voltage during a simulation and decomposes it into its phasor-domain quantities ($|V_1|$, $\angle V_1$, $|V_2|$, and $\angle V_2$). The sub-parameters are then used to back-calculate the proper parameters to use in the parameterized equation via the parameter/voltage relationship. Subsequently, the proper current injection is calculated; this process is depicted in Fig. 1. The commanded current is transformed to the abc frame and directly injected into the system from the ROM component. Prior to the fault, the ROM component injects current via PQ dispatch; it detects the fault by measuring the lowest phase RMS voltage. Once this falls below some threshold (0.88 pu is used in the results of this paper), the model immediately enters the operating state depicted in Fig. 1.

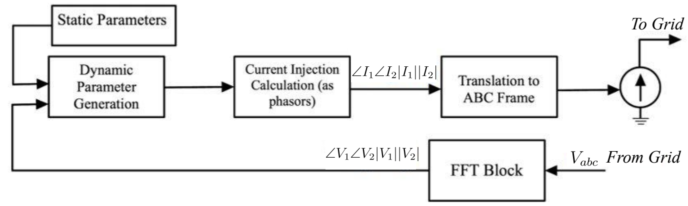


Fig. 1: The methodology of the PSCAD ROM component in a simulation, during the fault.

B. Generalizability of the ROM

By measuring the terminal voltage during a simulation, the model adapts appropriately by injecting the corresponding current based off what was seen during training. From here, the next question is how much each voltage phasor quantity should influence the model, and how the relationship should be (e.g., linear, quadratic). Yet, the exact relationship between terminal voltage and the current injection is uncertain. When any relationship is assumed, the model may diverge from accurate responses during unseen voltage conditions. To avoid this, the current version of the model uses a process most similar to clustering, so it is forced to inject current only similar to what it has been trained on. This removes the ability of the model to adapt to extremely different voltage conditions as compared to training. Therefore, it should be noted that the current implementation of the ROM in untrained conditions can cause inaccurate behaviors; the use of this version lies in repeated similar studies and in situations where internal vendor model parameters and control settings are unknown. Extension of the ROM into a verified, fully generalizable model is a necessary future research direction.

C. The PSCAD Component's Python Automation Script

Theoretical IBR fault models and an easy-to-use software product for industry rarely come together. We bridge this gap by creating a customizable Python automation script for the PSCAD component, allowing for easy implementation in place of a protection engineer's OEM model. The main Python script takes in a list of test cases (Excel spreadsheet), a PSCAD project containing the OEM model in the test system, a PSCAD project containing the ROM component in the test system (where it replaces an OEM model), and a list of flags to customize the process. From there, everything runs all in one go, in a hands-off manner. This includes creating the test data or retrieving it from another source (for example, if field data is the desired training data), computing the best-fit parameters, uploading these parameters into the ROM component, and testing the ROM's response against the OEM model data. From the perspective of a protection engineer, the workflow visualized in Fig. 2 and can be outlined as follows.

First, the user should decide if they want the script to generate new data from an existing OEM model PSCAD project, or instead read COMTRADE/CSV data from a field-study or elsewhere. If they would like to retrieve stored results, the script will read the data within a designated folder. Otherwise, it will load a PSCAD project containing the OEM model, and run it for all selected fault scenarios. This is performed via an API script. With this data, the Python script will either create new parameters for the ROM or read previously defined parameters. If new parameters are created, they are loaded

979-8-3315-8138-1

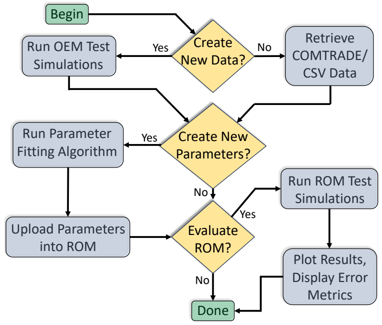


Fig. 2: Workflow of the Python Automation Script

into the PSCAD ROM component (which exists within a pre-defined project). At this point, the user has the option to evaluate the ROM against the same fault scenarios. If so, phasor plots are produced comparing the ROM to the OEM model, and error metrics are saved. The data is also stored via COMTRADE format to evaluate relay response. This entire process is documented for the protection engineer within the open-source platform [9], from creating their own API scripts for their project, which flags to set based off their desired use, etc. All in all, the use of the ROM component is streamlined and simple to use, eliminating the common barriers that would otherwise prevent them from implementing it in their studies.

III. Example Results in a Test System

For verification and demonstration of the whole process, the Python framework is used to create and verify a ROM for two separate OEM vendor models. One OEM model is IEEE2800 compliant, and another is not. The injection of a negative sequence current from an IBR is critical for proper relay operation, so the ability of the ROM to generalize to both models is crucial. In general, it should be noted that the ROM should adapt to any inverter/OEM model, given the data-driven methodology. However, multiple OEM models beyond the initial two are not analyzed in this study and should be verified in future work.

The test system depicted in Fig. 3 is utilized to perform the example, which was developed from a real-world grid [10]. Several locations along Line 2 are used to apply the fault, from the bus closest to the IBR model to the bus furthest from it. This example simulates single-line-to-ground (SLG) and line-to-line (LL) faults to initially examine the applicability between fault types. The initial power flow from the IBR is also adjusted for further comprehensiveness. Through the use of the Python workflow, these test cases are all run in the system with the OEM model, the parameters for the ROM are fit appropriately, the parameters are uploaded into the ROM component, and the subsequent fault analysis is simulated for the ROM as verification. This whole process runs in a single click by the user, with zero human intervention along the way.

A. OEM Vendor Model 1 (Std. 2800-2022 Compliant)

We begin by examining the effectiveness of the IEEE Std. 2800-2022 compliant OEM model in terms of its phasor domain current injection. A visual comparison of the differences between the OEM model and the ROM component in a particular scenario (0.25 pu active and 0 pu reactive power initial injections, fault exactly at the midpoint of Line 2) are

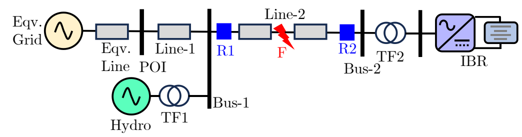
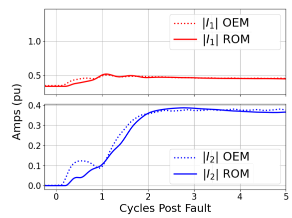
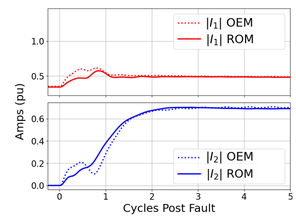


Fig. 3: The test system created in PSCAD, with a detailed GFL model on the right. Faults are placed on Line 2 for testing.

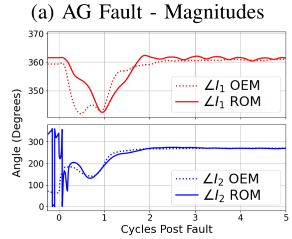
shown for both the SLG (phase-A to ground) and LL (phase-B to phase-C) faults in Fig. 4. Shown here are the ROM current injections mirroring the injection of the OEM model within a half cycle after the fault in both positive and negative sequence. Most notably is the performance of $\angle I_2$; its accuracy is crucial for proper relay operation. The similarity displayed here was seen to persist across all the testing scenarios. For the initial and pre-fault $\angle I_2$ difference, note the magnitude of negative sequence current is still essentially 0 pu. In this context, note the angle is meaningless until the magnitude becomes non-negligible.



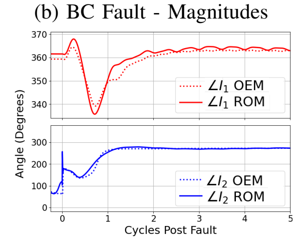
(a) AG Fault - Magnitudes



(b) BC Fault - Magnitudes



(c) AG Fault - Angles



(d) BC Fault - Angles

Fig. 4: Sequence phasor quantities following AG and BC faults

The root mean square error (RMSE) is used as an error metric, which is computed and averaged across all scenarios. The OEM model is considered as the target response. The RMSE is shown in Table I ('PF' stands for post-fault) and is computed for 0-1 cycle after the fault, 1-2 cycles after the fault, and 2-3 cycles after the fault. These time periods are most crucial for protection studies. When formatted this way, the reader can see how the accuracy improves as the time post-fault increases. This is because the ROM bases its current injection based off terminal voltage; it takes some time for the measured terminal voltage to reach its steady state value, where it is distinct from other fault types/locations. Yet, the ability to converge to the desired OEM response quickly post-fault is highlighted in these results, which is crucial for proper relay response. Note, once again, the initial large error in $\angle I_2$ is due to the low initial $|I_2|$; it does not imply the model is misbehaving.

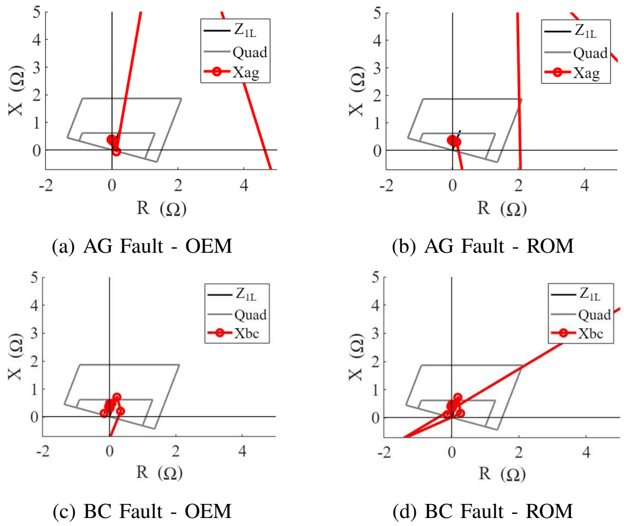
To verify proper relay response, the voltage and current phase waveform data at Bus 2 in Fig. 3 are passed through a relay model (the relay model developed in [11]). By comparing the response of the relay between the simulation with the

979-8-3315-8138-1

TABLE I: Average Accuracy (RMSE) of the ROM - IEEE2800 Compliant OEM

Component	0–1 Cycles PF	1–2 Cycles PF	2–3 Cycles PF
$\|I_1\|$ (pu)	6.21e–2	1.90e–2	8.51e–3
$\|I_2\|$ (pu)	6.76e–2	2.87e–2	2.96e–3
$\angle I_1$ (deg)	5.07	3.64	1.27
$\angle I_2$ (deg)	30.24	7.79	3.80

OEM model vs. the ROM, we realistically verify the ROM's use in actual protection studies. For the same particular faults as before, representative results, such as the quad plots as seen by the relay shown in Fig. 5 and negative sequence current polarized directional element $32Q$ shown in Fig. 6 are presented. Additionally, the relay binary outputs for those studies are summarized in Table II.



(a) AG Fault - OEM (b) AG Fault - ROM

(c) BC Fault - OEM (d) BC Fault - ROM

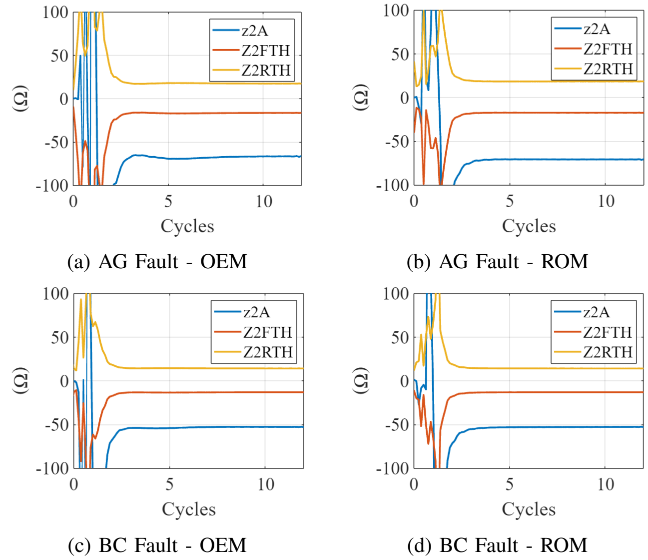
Fig. 5: Quad plots as seen by the relay during a specific AG and BC fault

As seen from Fig. 5, the ROM shows different trajectory initially when compared to the OEM model. This is because of the various transients during the first cycle. However, the trajectory of quad element for both ROM and OEM model look similar after this first-cycle difference, . The relay binary output also shows the timing difference of quad element is less than a quarter of a cycle. For the $32Q$ element response, the ROM mirrors the OEM model closely in terms of shape, trend and timing (less than a quarter of a cycle). Both representative results showcase the close-match of the ROM with the OEM model.

Overall, the relay operation timing is displayed in Table II. The data is presented in cycles post-fault taken to operate, and shown for each individual element of the relay. The response is shown for the same specific SLG (AG) and LL (BC) faults as before. As the relays operated in a similar manner regardless of using the OEM model or the ROM, the ROM would be a reliable alternative for protection engineers in this test setup. More test grids, faults, and OEM models should be verified in future research, yet its accuracy is verified in this initial study.

B. OEM Vendor Model 2: IEEE Std. 2800-2022 Non-Compliant

The IEEE Std. 2800-2022 non-compliant OEM model does not actively control its negative sequence current magni-



(a) AG Fault - OEM (b) AG Fault - ROM

(c) BC Fault - OEM (d) BC Fault - ROM

Fig. 6: Directional 32Q element plots as seen by the relay during a specific AG and BC fault

TABLE II: Amount of Time Post-Fault for Relay Element to Trigger (Cycles) - IEEE2800 Compliant OEM

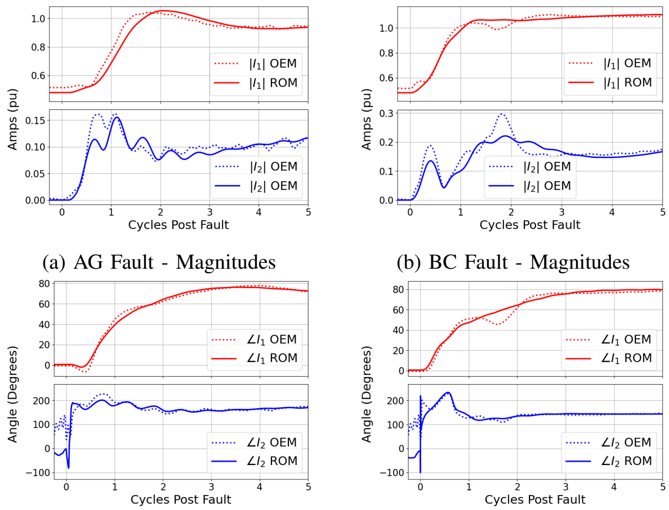
Relay Element	**AG Fault Scenario**		**BC Fault Scenario**	
	OEM	**ROM**	**OEM**	**ROM**
TRIP	2.25	2.375	2.25	2
Z1G / ZIP	2.25	2.25	2.25	2
XAG1 / XBC1	2.25	2.0	2.25	2
XAG2 / XBC2	2.125	2.375	2.25	2
MAG1 / MBC1	2.25	2.25	2.25	2
MAG2 / MBC2	2.25	2.25	2.25	1.875
F32V / F32Q	0.625	0.75	2.25	2
FSA	2.25	2.25	N/A	N/A
FIDEN	2.25	2.375	N/A	N/A

tude/angle, both of which are critical for proper relay operation. Therefore, the ROM should be able to adapt to these variations, especially given the majority of legacy IBRs are not IEEE Std. 2800-2022 compliant. A visual comparison of the differences between the OEM model and the ROM component in a particular scenario (0.5 pu active and 0 pu reactive power initial injections, fault exactly at the midpoint of Line 2) are shown for both the SLG (phase-A to ground) and LL (phase-B to phase-C) faults in Fig. 7. It can be seen that certain peaks of the negative sequence magnitudes are not fully captured, yet, these are brief and the ROM is able to encapsulate the majority of the negative sequence uniqueness, which is particularly difficult with an IEEE2800 non-compliant device.

The RMSE is shown in Table III. Even though the OEM model is not IEEE Std. 2800-2022 compliant, its performance is similar to the IEEE Std. 2800-2022 compliant model. Again, the ability to match the desired OEM response quickly post-fault is highlighted in these results. As time increases post-fault, the error tends to decrease.

The relay model was not able to trip with this OEM model because it injects a high level of harmonic content. The focus of this paper is not the robustness of the relay model used; therefore, the metrics and visualization shown above for the fundamental component phasors are displayed. While

979-8-3315-8138-1



(a) AG Fault - Magnitudes (b) BC Fault - Magnitudes

(c) AG Fault - Angles (d) BC Fault - Angles

Fig. 7: Sequence phasor quantities following AG and BC faults

TABLE III: Average Accuracy (RMSE) of the ROM – IEEE Std. 2800-2022 Non-Compliant OEM

Component	0–1 Cycles PF	1–2 Cycles PF	2–3 Cycles PF
$\|I_1\|$ (pu)	3.23e–2	5.85e–2	1.16e–2
$\|I_2\|$ (pu)	2.49e–3	1.84e–2	7.96e–3
$\angle I_1$ (deg)	3.83	2.31	1.65
$\angle I_2$ (deg)	15.7	4.48	2.96

the ROM is not able to encapsulate harmonic content in it's current injection, it accurately mirrors the current injection for the fundamental component as shown above. This should be noted by users; the current state of the ROM will not inject any harmonics similar to it's OEM equivalent; if this behavior is desired in their study, it should not be used.

Lastly, the ROM was able to achieve this accuracy with, on average, 32.8% less simulation runtime across all scenarios, from both fault models. For example, during one six second simulation, the OEM model test case took 29.77 seconds to run as opposed to 17.32 seconds for the ROM test case. These trials were held under as similar of background operating processes as possible. This factor can be expected to increase in systems with many GFLs/OEM models, as each would contribute to the reduced-order of the system model.

IV. Ending Notes and Future Research

The open-source software for both the PSCAD library component and the Python automation script is fully implemented. Complete documentation is also developed to aid the protection engineer, from creating their test system to troubleshooting. The process of creating their own Python API scripts to interact with their specific system under study is outlined, ensuring the user can easily implement the ROM. The software package is currently in the publication review process within NREL; it can be found at [9] upon approval.

While this paper introduced the ROM for engineer use, future papers are under development and will detail the internal mathematics and principles of the ROM in depth. Following versions of this ROM will also include more complexities, which should allow for better generalization of the model, particularly to untrained fault conditions. However, even with the simplistic nature of the present version, its performance is validated by relay response in this real-grid test system against two different OEM models. Lastly, it should be noted the principle of the ROM can be built into any EMT software, but PSCAD is used in this initial realization to reach the widest audience; for example, its extensive/benchmark usage within ISO New England [12].

V. Conclusion

This paper introduced a fully implemented ROM within PSCAD for EMT-based fault studies and protection designs, tailored for protection engineers. It provides a comprehensive ROM overview, implementation guidelines, and an accompanying PSCAD automation script. Validation using a real-world power system model and OEM vendor IBR models demonstrates the ROM's accuracy, usability and practicability, with sequence current matching within less than half a cycle. Additionally, relay validation confirms its ability to replicate OEM vendor model fault responses effectively with less than a quarter-of a cycle timing difference, ensuring reliable relay element operation in protection studies. Accuracy was validated with IEEE Std. 2800-2022 compliant and non-compliant models, a crucial generalization for real-world application. Python automation streamlines the entire workflow, enabling protection engineers to efficiently apply this novel ROM for fast, reliable, and enhanced EMT studies.

References

[1] A. Haddadi, E. Farantatos, I. Kocar, and U. Karaagac, "Impact of inverter based resources on system protection," *Energies*, vol. 14, no. 4, p. 1050, 2021.

[2] X. Lyu, J. Xie, and T. E. McDermott, "Impact of inverter-based resources on utility transmission system protection," Pacific Northwest National Laboratory, Richland, WA, USA, Tech. Rep. PNNL-36069, 2024.

[3] North American Electric Reliability Corporation (NERC), "Electromagnetic transient modeling and simulation for BPS-connected inverter-based resources," NERC, Atlanta, GA, USA, Tech. Rep., Mar. 2023.

[4] M. Abubakar, H. Renner, and R. Schürhuber, "Development of a novel control scheme for grid-following converter under asymmetrical faults," *Energies*, vol. 16, no. 3, 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/3/1276

[5] M. Rasheduzzaman, J. A. Mueller, and J. W. Kimball, "Reduced-order small-signal model of microgrid systems," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1292–1305, Oct 2015.

[6] L. Luo and S. V. Dhople, "Spatiotemporal model reduction of inverter-based islanded microgrids," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 823–832, Dec 2014.

[7] R. Mahumd *et al.*, "A parametric reduced-order model for inverter short-circuit response in protection studies," in *2026 IEEE PES T&D*. IEEE, 2025, pp. 1–5.

[8] Mahmud, Rasel and others, *A Scenario-Driven Framework for Fault Response Simulation, Feature Extraction, and Reduced-Order Model Tuning of Inverter-Based Resources*, 2025, nREL Software Record SWR-25-141.

[9] National Renewable Energy Laboratory. (2025) Software. [Online]. Available: https://www.nrel.gov/research/software.

[10] S. Chakraborty *et al.*, "Studying the impact of IBR modeling on the commonly applied transmission line protective elements," in *2024 IEEE Energy Conversion Conference and Expo (ECCE)*. IEEE, 2024, pp. 1–5.

[11] Wang, Jing and others. (2025) How Inverter-based Resources (IBRs) Affect Protection Relay Elements. [Online]. Available: https://docs.nrel.gov/docs/fy25osti/93804.pdf.

[12] ISO New England, "ISO New England planning procedure no. 5-6: Interconnection planning procedure for generation and elective transmission upgrades," ISO New England, Tech. Rep., October 2025, effective date: October 11, 2025. [Online]. Available: https://www.iso-ne.com/static-assets/documents/planning_procedures/PP5-6.pdf

979-8-3315-8138-1