
Operator Learning for Power Systems Simulation

Matthew Schlegel
Schulich School of Engineering
University of Calgary
matthew.schlegel@ucalgary.ca

Matthew E. Taylor
Department of Computing Science
University of Alberta
mtaylor3@ualberta.ca

Mostafa Farrokhhabadi
Schulich School of Engineering
University of Calgary
mostafa.farrokhhabadi@ucalgary.ca

Abstract

Time domain simulation, i.e., modeling the system’s evolution over time, is a crucial tool for studying and enhancing power system stability and dynamic performance. However, these simulations become computationally intractable for renewable-penetrated grids, due to the small simulation time step required to capture renewable energy resources’ ultra-fast dynamic phenomena in the range of 1-50 microseconds. This creates a critical need for solutions that are both fast and scalable, posing a major barrier for the stable integration of renewable energy resources and thus climate change mitigation. This paper explores operator learning, a family of machine learning methods that learn mappings between functions, as a surrogate model for these costly simulations. The paper investigates, for the first time, the fundamental concept of simulation time step-invariance, which enables models trained on coarse time steps to generalize to fine-resolution dynamics. Three operator learning methods are benchmarked on a simple test system that, while not incorporating practical complexities of renewable-penetrated grids, serves as a first proof-of-concept to demonstrate the viability of time step-invariance. Models are evaluated on (i) zero-shot super-resolution, where training is performed on a coarse simulation time step and inference is performed at super-resolution, and (ii) generalization between stable and unstable dynamic regimes. This work addresses a key challenge in the integration of renewable energy for the mitigation of climate change by benchmarking operator learning methods to model physical systems.

1 Introduction

The increasing penetration of renewable and distributed energy resources has introduced novel dynamic phenomena to the power grid, creating new challenges for system stability analysis. Time domain simulation has been used as a reliable tool for studying system stability. In general, power systems time domain simulation involves the numerical integration of nonlinear differential-algebraic equations [12], which are computationally intensive. This bottleneck has rendered system-wide simulation computationally intractable as renewable generation grows, given the need for fine-grained integration time steps to capture novel dynamics associated with inverters [15, 19]. Recent grid incidents, such as the April 2025 massive blackout in Spain and Portugal, underscore these limitations and the need for scalable, real-time analysis tools. Overcoming this computational barrier is crucial to the stable integration of renewable energy sources as mandated by climate change mitigation policies.

Prior works on machine learning (ML)-based modeling for power system simulation can be broadly categorized into two approaches. First are methods that use classifiers to determine system stability

status and/or stability margin post-disturbance [8, 9, 22, 24, 25]. These are not in the scope of this paper due to their inability to be used as surrogate models for time domain simulation. While helpful for stability prediction, they cannot be used for control design and stability enhancement. Second are approaches that have used surrogate models to produce trajectories from initial conditions [4, 6, 13, 18, 23]. Surrogate modeling can be further subdivided into two approaches. The first replaces a component in the full simulation with a model that approximates its differential equations [4, 23]. This has the advantage of being incorporated directly into existing simulation frameworks, but necessitates significant complexity to create an interface to integrate the machine learning-based component into the differential equations for the rest of the system [4]. Another approach, which is the focus of this paper, is to devise a surrogate data-driven model for the entire system. However, a surrogate model of the full system requires extensive training, verification, and generalization tests.

In this paper, a surrogate model is used to learn a function, in this case an operator, that maps from an initial disturbed trajectory of a test system. Operator learning has emerged in scientific ML offering discretization invariance, and thus the potential to generalize across resolutions. The immediate contributions of this study are: (i) a comparison of three operator learning methods on zero-shot super-resolution generalization, and (ii) an analysis of the generalization between stable and unstable regimes. The long-term contributions of the paper are: (i) providing a path for stable integration of renewable energies through a scalable and computationally tractable analysis tool, and (ii) tackling machine learning for modeling complex physical systems.

2 Operator Learning

Operator learning approximates a mapping between two infinite-dimensional spaces¹ [2, 11, 16]. Methods in operator learning enable discretization-invariant predictions through algorithmic and architectural designs. This property allows models trained on coarse time domain simulations ($\Delta t = 100$ ms) to potentially generalize to finer simulation time steps ($\Delta t = 50 \mu s$), reducing the need for computationally-intensive high-resolution simulation to generate training data for a surrogate model. In this paper, we learn an operator G that maps from an initial trajectory $t \in [0, \tau]$ to a future expected trajectory generated from an ordinary differential equation $t \in [\tau_i, T]$.

Deep Operator Networks (DeepONets) [16] use two neural network branches: a branch network processes the input trajectory, and a trunk network encodes the output query time. Their outputs are combined via an inner product to produce the predicted trajectory value.

Fourier Neural Operators (FNOs) [10, 11, 14] project inputs into a Fourier basis, apply learned spectral multipliers, and transform back to the time domain. Tensorization reduces parameter count while preserving accuracy compared to standard FNOs. This spectral representation can capture smooth temporal structures and support resolution changes.

Latent Neural ODEs (LNODEs) [5] encode the input trajectory into a latent state, evolve it through a neural ODE, and decode it back to the original space. We use an autoencoder variation rather than a variational autoencoder for a fairer comparison. The fixed-step (Adams) and adaptive-step (Dormand–Prince–Shampine) solvers are evaluated.

The rationale for selecting the above methods is as follows: neural ODEs have been used in the prior art to model independent components of a power system to integrate into a traditional simulation [4, 23]. DeepONets have been applied to post-disturbance power system modeling [18]. FNOs, while suitable for the task, have not been used for power system modeling. This paper presents the first step towards the application of FNOs to power system modeling.

3 Experimental Setup

3.1 SMIB Test System

This study focuses on a single-machine infinite-bus (SMIB) test system, which is a single synchronous machine connected to a stiff, i.e., fixed voltage and frequency, source via a transmission line [20]. Without the loss of generality, the generator angle dynamics are used for this study, associated with a classic stability phenomenon: if a generator’s mismatch between input mechanical and output

¹The interested reader should refer to [16] and [11] for more details on the operator learning problem setting.

electrical powers exceeds a critical threshold after a disturbance, i.e., any change that alters the steady-state or normal dynamic behavior of the system, the machine can no longer maintain a stable exchange of power with the grid. This instability is a well-known real-world phenomenon in power systems and continues to pose practical challenges. As such, the SMIB, despite its simplicity, serves as a suitable candidate to explore new proof-of-concept simulation approaches; it is simple enough to isolate the fundamental dynamics, yet rich enough to reflect practical stability issues.

A synchronous machine angle dynamics are governed by the following equation:

$$\frac{\partial^2 \delta}{\partial t^2} = \frac{\pi f_0}{H} \left(P_m - D \frac{\partial \delta}{\partial t} - \frac{|E||V|}{X} \sin \delta \right), \quad (1)$$

where $\delta(t)$ is the rotor (power) angle in electrical radians, P_m is the mechanical power input per-unitized on the machine base, D is the damping coefficient in per unit power per radians/s, E and V are the machine and stiff source voltages per unitized on a common base, X is the reactance of the transmission line between the two sources in per unit, H is the inertia constant in seconds, and f_0 is the nominal frequency in Hz. At steady state, the pre-disturbance rotor angle is $\delta_0 = \sin^{-1}(P_m/P_m^{\max})$, where $P_m^{\max} = |E||V|/X$ is the maximum transferable electrical power. A disturbance at $t = 0$ changes the mechanical input from P_m to P_{m1} . In the undamped case ($D = 0$), the maximum post-disturbance mechanical input preserving stability is:

$$P_{m1}^{\max} = P_m^{\max} \sin(\pi - \delta_{\max}) \quad (2)$$

$$0 = (\delta_{\max} - \delta_0) \sin(\delta_{\max}) + \cos(\delta_{\max}) - \cos(\delta_0). \quad (3)$$

where $\delta_{\max}$ is determined by Equation (3). Integration is performed using Runge-Kutta of order 5(4).

3.2 Data Generation and Model Training

SMIB trajectories are generated in stable and unstable regimes by sampling the $P_m \sim \mathcal{U}[0, 2]$, and damping $D \sim [0.0, 0.135]$. For stable regimes, we sample $P_{m1} \sim \mathcal{U}[0.0, P_{m1}^{\max}]$ assuming $D = 0$ to avoid mislabeled trajectories. For unstable trajectories, the lower bound of the unstable range is approximated using a 100-iteration bisection search over the range $[P_{m1}^{\max}, P_{\max}]$ [7]. SMIB parameters are resampled if $P_{\max}$ does not result in an unstable trajectory. Other parameters are taken from an example in [20]: $E = 1.35$, $V = 1.0$, $X = 0.65$, $H = 9.94$, and $f_0 = 60$. These may not reflect a practical power system but their values are unlikely to change the conclusions. Each dataset had 8000 training trajectories and 1000 validation trajectories. The input sequence covers $t \in [0, 0.2]$ s and the target covers $t \in [0.3, 3.1]$ s. The input features δ is clipped to a max value of π and $\mathbf{x} = [\delta, \omega]$ are normalized such that $\delta \in [0, 1]$ and $\omega \in [-1, 1]$ using ranges in the training set.

Each model is tuned through 120 trials using a Bayesian hyperparameter optimization method [1, 3] with a small number of epochs (20 for the LNODE and FNO and 200 for the DeepONets). All the models have approximately $\sim 700,000$ parameters to ensure a fair comparison. Using the best hyperparameter settings, each model was trained to obtain a reasonable validation error, or until the validation error stopped decreasing. The LNODEs and FNOs took 60 epochs, while the DeepONets took 600 epochs. All methods used the H1 Sobolev norm loss for training (see [11] for details).

4 Results

Zero-Shot Super-Resolution: Models are trained on data generated on a coarse time step (100 ms) and compared on their generalization to data generated on a fine time step ($50 \mu\text{s}$) without retraining. Results are reported in Table 1 with example trajectories in Figure 1. Overall, the LNODE outperformed both DeepONets and FNOs, with the LNODE achieving the lowest change in performance between the resolutions. While the FNOs outperformed the other approaches in absolute RMSE on the original training resolution, they had the highest drop in performance for the higher resolution.

Generalization across different dynamic regimes: The models are trained on two datasets: one with 0% unstable trajectories and one containing 20% unstable trajectories for training. The models are compared on their performance for a single SMIB system, $P_m = 0.4$ and $D = 0.05$, and a sweep over the post-disturbance mechanical power P_{m1} . The system loses stability at a critical threshold marked by the red dashed line in Figure 2. As seen in Figure 2, the methods struggle to generalize to dynamical regimes not covered by the training data; this is consistent with prior observations [21].

Model Name	$\Delta t = 100ms$ (RMSE)	$\Delta t = 50\mu s$ (RMSE)	Percent Difference
DeepONet	0.0220 ± 0.0001	0.0348 ± 0.0002	45.2(39.0, 51.31)
FNO	0.0186 ± 0.0001	0.0302 ± 0.0001	47.5(39.9, 54.8)
LNODE (Fixed)	0.0280 ± 0.0006	0.0305 ± 0.0006	8.6(0.5, 33.6)
LNODE (Adaptive)	0.0275 ± 0.0003	0.0296 ± 0.0003	7.3(0.4, 19.7)

Table 1: **Zero-Shot Super-Resolution:** Values are reported over 200 test trajectories and 20 independent runs. Both resolution datasets use the same SMIB parameters. Bootstrap confidence intervals are used for the percent difference and standard errors are reported for RMSE values.

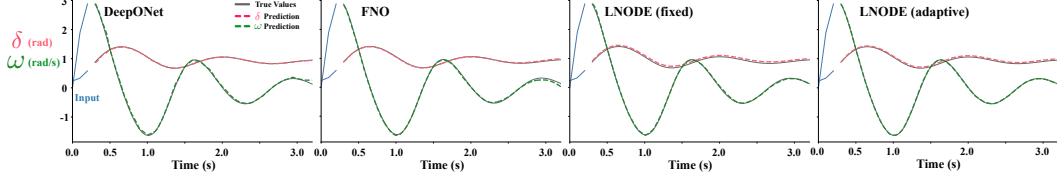


Figure 1: **Zero-Shot Super-Resolution:** This figure illustrates the true time series of the system angle (δ) and frequency (ω) dynamics in solid lines and the operators' predicted trajectories in dotted lines. The plots start with the segment of the true trajectory in a solid blue line used as the input.

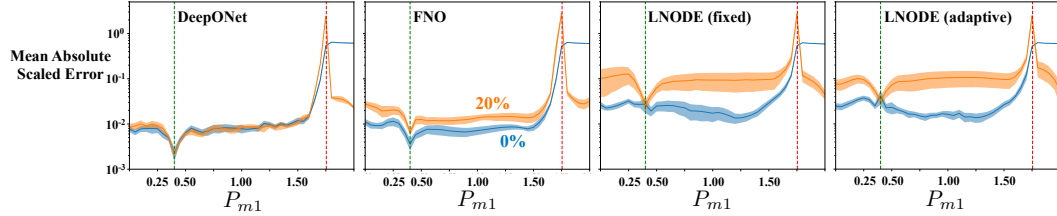


Figure 2: **Generalization across different dynamical regimes:** Training data has 0% (blue) and 20% (orange) unstable trajectories. The green dotted line is where $P_{m1} = P_m$, i.e., no disturbance, and the red dotted line is the critical point where trajectories become unstable. The mean absolute scaled error is calculated using a baseline-prediction of the previous time step over 20 runs.

When trained with a mixture of stable and unstable trajectories, all the methods are able to capture the unstable region's dynamics, while the DeepONet is the only method to perform similarly in the stable regime when the training dataset includes both stable and unstable trajectories.

5 Discussion

The main limitation of this work is the simplicity of the SMIB, which does not capture the complex dynamics of renewable-penetrated grids. Although all methods performed well, comprehensive modeling and testing on practical test systems are needed to draw generalizable conclusions. For example, on this simple test system, the FNO was shown to outperform alternatives in absolute error in the two evaluation tasks, but the DeepONets and LNODE approaches are likely to be competitive with different inductive biases [17]; more tests are needed to prove this. This study provides a controlled setting to establish the feasibility of time step-invariant operator learning. This capability addresses a major bottleneck in renewable-penetrated power system analysis through time domain simulation, ultimately supporting the transition to low-carbon grids needed for climate change mitigation.

GitHub Repository: <https://github.com/mkschleg/NOForPSSim.git>

Acknowledgments and Disclosure of Funding

This research was done in the Data+Grid Research and Innovation (DGRI) Lab at the University of Calgary and the Intelligent Robot Learning (IRL) Lab at the University of Alberta. This research was funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada Grant RGPIN-2024-03955 and supported in part by Alberta Innovates; Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Digital Research Alliance of Canada; and Mitacs.

References

- [1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-generation Hyper-parameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA, July 2019. Association for Computing Machinery.
- [2] K. Azizzadenesheli, N. Kovachki, Z. Li, M. Liu-Schiaffini, J. Kossaifi, and A. Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, May 2024.
- [3] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.
- [4] M. Bossart, J. D. Lara, C. Roberts, R. Henriquez-Auba, D. Callaway, and B.-M. Hodge. Acceleration of Power System Dynamic Simulations Using a Deep Equilibrium Layer and Neural ODE Surrogate. *IEEE Transactions on Energy Conversion*, pages 1–12, 2025.
- [5] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [6] T. Cheng, R. Chen, N. Lin, T. Liang, and V. Dinavahi. Machine-Learning-Reinforced Massively Parallel Transient Simulation for Large-Scale Renewable-Energy-Integrated Power Systems. *IEEE Transactions on Power Systems*, 40(1):970–981, Jan. 2025.
- [7] J. F. Epperson. *An introduction to numerical methods and analysis*. John Wiley & Sons, 2013.
- [8] A. Gupta, G. Gurralla, and P. S. Sastry. An Online Power System Stability Monitoring System Using Convolutional Neural Networks. *IEEE Transactions on Power Systems*, 34(2):864–872, Mar. 2019.
- [9] M. He, J. Zhang, and V. Vittal. Robust Online Dynamic Security Assessment Using Adaptive Ensemble Decision-Tree Learning. *IEEE Transactions on Power Systems*, 28(4):4089–4098, Nov. 2013.
- [10] J. Kossaifi, N. B. Kovachki, K. Azizzadenesheli, and A. Anandkumar. Multi-Grid Tensorized Fourier Neural Operator for High-Resolution PDEs. *Transactions on Machine Learning Research*, Apr. 2024.
- [11] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [12] P. Kundur. Power system stability. *Power system stability and control*, 10(1), 2007.
- [13] J. Li, M. Yue, Y. Zhao, and G. Lin. Machine-Learning-Based Online Transient Analysis via Iterative Computation of Generator Dynamics. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, Nov. 2020.
- [14] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, Oct. 2020.
- [15] P. Lopion, P. Markewitz, M. Robinius, and D. Stolten. A review of current challenges and trends in energy systems modeling. *Renewable and Sustainable Energy Reviews*, 96:156–166, Nov. 2018.
- [16] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, Mar. 2021.

- [17] L. Lu, X. Meng, S. Cai, Z. Mao, S. Goswami, Z. Zhang, and G. E. Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, Apr. 2022.
- [18] C. Moya, S. Zhang, G. Lin, and M. Yue. DeepONet-grid-UQ: A trustworthy deep operator framework for predicting the power grid’s post-fault trajectories. *Neurocomputing*, 535:166–182, May 2023.
- [19] S. Pfenninger, A. Hawkes, and J. Keirstead. Energy systems modeling for twenty-first century energy challenges. *Renewable and Sustainable Energy Reviews*, 33:74–86, May 2014.
- [20] H. Saadat. *Power System Analysis*, volume 2. McGraw-hill, 1999.
- [21] S. Subramanian, P. Harrington, K. Keutzer, W. Bhimji, D. Morozov, M. W. Mahoney, and A. Gholami. Towards Foundation Models for Scientific Machine Learning: Characterizing Scaling and Transfer Behavior. *Advances in Neural Information Processing Systems*, 36:71242–71262, Dec. 2023.
- [22] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella. An artificial intelligence framework for online transient stability assessment of power systems. *IEEE Transactions on Power Systems*, 4(2):789–800, May 1989.
- [23] T. Xiao, Y. Chen, S. Huang, T. He, and H. Guan. Feasibility Study of Neural ODE and DAE Modules for Power System Dynamic Component Modeling. *IEEE Transactions on Power Systems*, 38(3):2666–2678, May 2023.
- [24] J. J. Q. Yu, D. J. Hill, A. Y. S. Lam, J. Gu, and V. O. K. Li. Intelligent Time-Adaptive Transient Stability Assessment System. *IEEE Transactions on Power Systems*, 33(1):1049–1058, Jan. 2018.
- [25] L. Zhu, D. J. Hill, and C. Lu. Hierarchical Deep Learning Machine for Power System Online Transient Stability Prediction. *IEEE Transactions on Power Systems*, 35(3):2399–2411, May 2020.