Electric Power Systems Research 212 (2022) 108562

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Continuous-time echo state networks for predicting power system dynamics ☆

Ciaran Roberts [a,e,*], José Daniel Lara [b], Rodrigo Henriquez-Auba [a], Matthew Bossart [c], Ranjan Anantharaman [d], Chris Rackauckas [d], Bri-Mathias Hodge [c,f], Duncan S. Callaway [b,e]

[a] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, United States
[b] Energy and Resources Group, University of California, Berkeley, United States
[c] Electrical, Computer & Energy Engineering and Renewable and Sustainable Energy Institute, University of Colorado Boulder, United States
[d] Computer Science and AI Laboratory (CSAIL), Massachusetts Institute of Technology, United States
[e] Grid Integration Group, Lawrence Berkeley National Laboratory, United States
[f] National Renewable Energy Laboratory, Golden, CO, United States

ARTICLE INFO

Keywords:
Data-driven modeling techniques
Electro-magnetic transients
Machine learning
Power system dynamics

ABSTRACT

With the growing penetration of converter-interfaced generation in power systems, the dynamical behavior of these systems is rapidly evolving. One of the challenges with converter-interfaced generation is the increased number of equations, as well as the required numerical timestep, involved in simulating these systems. Within this work, we explore the use of continuous-time echo state networks as a means to cheaply, and accurately, predict the dynamic response of power systems subject to a disturbance for varying system parameters. We show an application for predicting frequency dynamics following a loss of generation for varying penetrations of grid-following and grid-forming converters. We demonstrate that, after training on 20 solutions of the full-order system, we achieve a median nadir prediction error of 0.17 mHz with 95% of all nadir prediction errors within ±4 mHz. We conclude with some discussion on how this approach can be used for parameter sensitivity analysis and within optimization algorithms to rapidly predict the dynamical behavior of the system.

1. Introduction

The increasing integration of converter-interfaced generation (CIG) into large-scale synchronous power systems is forcing a re-examination of simulation practices to assess stability and reliability. Historically, large-scale power systems analysis was primarily focused on eletromechanical phenomena, arising from synchronous machines and their associated controls. However, the introduction of CIG, whose control loops act on the timescale of microseconds to milliseconds, is altering the dynamical behavior of power systems and is forcing examination of both electromagnetic and eletromechanical phenomena. These changes are leading to questions about the validity of several simplifications that have enabled computationally tractable large-scale system time domain simulations [1].

One of these key simplifications has been representing high-frequency dynamics, e.g., network dynamics, by either steady-state models or simplified dynamic models [1]. With increasing penetration of CIG, however, these simplifications have been shown to lead to incorrect conclusions about the small-signal stability of an operating point [2,3] and/or lead to incorrect time-domain behavior following a disturbance [4]. This has led to some system operators recently introducing more stringent requirements for models of CIG to enable detailed electromagnetic studies [5].

These electromagnetic studies can be very computationally intensive, and potentially prohibitively time-consuming. The inclusion of both the dynamics of slow-acting synchronous machine and fast-acting CIG results in a *stiff* system. These systems require implicit differential equations solvers and, typically, very small simulation timesteps during a disturbance to accurately solve. This challenge in numerically solving the system is compounded by the increase in the number of equations

☆ This research was supported in part by the Advanced Grid Modeling program within the Office of Electricity of the U.S. Department of Energy, under contract DE-AC02-05CH11231. This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the Department of Energy Office of Electricity Advanced Grid Modeling program. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

* Corresponding author at: Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, United States.
E-mail addresses: ciaran_r@berkeley.edu (C. Roberts), jdlara@berkeley.edu (J.D. Lara), rhenriquez@berkeley.edu (R. Henriquez-Auba), matthew.bossart@colorado.edu (M. Bossart), ranjanan@mit.edu (R. Anantharaman), crackauc@mit.edu (C. Rackauckas), brimathias.hodge@colorado.edu (B.-M. Hodge), dcal@berkeley.edu (D.S. Callaway).

https://doi.org/10.1016/j.epsr.2022.108562
Received 3 October 2021; Received in revised form 17 April 2022; Accepted 2 July 2022
Available online 15 July 2022

to solve as large capacity synchronous generators are replaced by tens, or hundreds, of smaller capacity CIG plants. Therefore, not only are these systems computationally more expensive to simulate for a fixed number of differential equations, but the number of equations is also significantly larger, assuming all dynamics are modeled.

This paper explores the application of scientific machine learning (SciML) to accelerate power systems simulations. SciML is a growing area of research that attempts to blend principles of scientific computing and machine learning [6]. One research area of SciML is computational acceleration using *surrogates*. A surrogate is an accurate approximation of the corresponding physics-based model trained using a data-driven approach. The training data come from recording the output of a full-order physics-based simulation across a subset of operating conditions. Once a surrogate has been trained to fit the training data, it can then be used in place of the full-order physics-based model. This reduces computational burden and can facilitate further analysis, e.g., sensitivity analysis or optimization.

The application of SciML techniques to model the dynamical behavior of power systems has been gaining interest in recent years, particularly the application of physics informed neural networks (PINNs) [7,8]. In this paper, we will explore the application of continuous-time echo state networks (CTESNs) [9] for learning a surrogate to predict the time-domain solution of a power system for varying system parameters.

We begin with a discussion of the challenges that stiff systems present for some popular SciML surrogates, e.g., PINNs and long short-term memory (LSTM) networks. We then present a detailed introduction to CTESNs and how they bypass these stiffness-related challenges. Following this, we demonstrate an application for accelerating power system dynamic simulations through the use case of predicting frequency dynamics for varying compositions of CIG. We show that, after training on 20 solutions of the full-order system, we achieve a median nadir prediction error of 0.17 mHz with 95% of all nadir prediction errors within ±4 mHz across the parameter space. Finally, we conclude with a discussion on the potential of CTESNs for power systems analysis and propose future research directions.

The main contributions of this paper are:

- An exploration of CTESNs as a new approach to accelerating parameter-sensitivity time domain simulations for power systems.
- An empirical examination of how the accuracy of CTESNs depends on the number of true solutions used during the training phase.
- An applied example where we use CTESNs to predict frequency dynamics following a large loss of generation with varying relative compositions of CIG (i.e., grid-forming vs grid-following).

2. Surrogates for power system simulation

The general process for training a surrogate is shown in Fig. 1. The parameter space of interest is sparsely sampled and the solution of the full-order model is obtained for each of these samples. These solutions form the training data from which the surrogate learns the parameter sensitivity of the solution. Once a surrogate has been trained, it can then be used to probe the parameter space at a much finer granularity with significantly less computational overhead, while maintaining a sufficient level of accuracy.

One surrogate model that has gained significant interest in recent years is a PINN [10]. A PINN is a continuous time function that predicts the solution of a physical system. Its loss function typically has two terms. The first of these terms penalizes the neural network for predicting values that do not match measured, or simulated, data. The second term in the loss function typically encodes known, or approximate, physical equations governing the behavior of the underlying physical system. An example of this in the context of power systems is penalizing the neural network for predicting voltage magnitudes and angles that do not satisfy the algebraic network power flow equations. PINNs have shown promise in predicting the solution of synchronous machines [8].

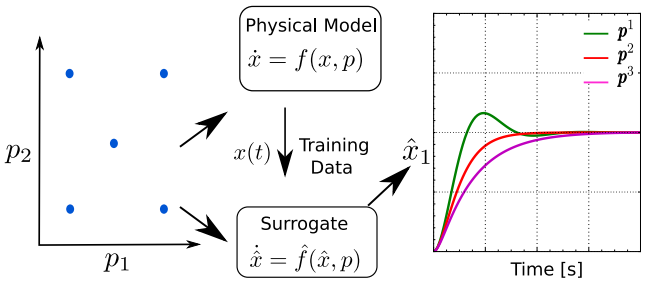

Fig. 1. Surrogate for computational acceleration.

Another data-driven approach to learning the solution to a system of equations are LSTM networks. These neural networks are a type of recurrent neural network (RNN) where connections between nodes form a directed graph along a temporal sequence. This allows RNNs to learn to exhibit temporal dynamic behavior, and consequently, are suitable for learning the solution to a dynamical system. Like PINNs, LSTM networks have been successfully shown to be able to learn the solution of a synchronous machine [11,8].

Both PINNs and LSTM networks have, however, been shown to be very difficult to fit to stiff systems [9,12]. With the continual addition of CIG to our power systems, fast power system dynamics can no longer universally be well approximated by their steady-state algebraic equations [1]. Stiff systems are difficult for gradient-based surrogate training techniques for the same reason they are very difficult for explicit numerical solvers; they require a very small numerical step. For gradient-based optimization techniques, this numerical step is the learning rate of the optimizer.

2.1. CTESNs

Echo state networks (ESNs), like LSTM networks, are a type of RNN. However, they differ in that the weights of both the input matrix, $\boldsymbol{W}_{in}$, and the weights of the reservoir, $\boldsymbol{A}$, in (1) are randomly assigned and fixed throughout the training. Therefore, the *training* of an ESN simplifies to learning the weights of the output layer, $\boldsymbol{W}_{out}$, in (2).

$$\boldsymbol{r}_{n+1} = f(\boldsymbol{A}\boldsymbol{r}_n + \boldsymbol{W}_{in}\boldsymbol{x}_n) \tag{1}$$

$$\hat{\boldsymbol{x}}_n = g(\boldsymbol{W}_{out}\boldsymbol{r}_n) \tag{2}$$

Typically the function $g(\cdot)$ is the identify function. The process for learning the weights of the readout layer then reduces to a least-squares problem, thereby, significantly simplifying the training of these networks. Consequently, ESNs avoid the challenges of gradient-based optimization for learning stiff systems. ESNs typically have a large recurrent layer, as shown in Fig. 2, that results in a high-dimensional state space of rich dynamics. It is this wide spectrum of heterogeneous dynamics that allow ESNs to achieve excellent performance in predicting time series behavior. ESNs have been used to predict chaotic systems [13], energy consumption and wind power generation [14].

Typically, ESNs are trained against fixed timestep interval data. This, however, can present challenges for learning stiff systems where a high number of samples may be required during disturbances to appropriately sample the fast dynamics. To overcome this issue, the authors in [9] proposed the use of continuous-time echo state networks (CTESNs). These networks sample the true solution at the same non-uniform time intervals as was required to accurately solve the system using an adaptive implicit ODE solver.

To train a CTESN, we first begin with a sample solution of the true physical non-linear system that we term a *nominal* solution. We denote this nominal solution with parameters $\boldsymbol{p}^\star$, and pre-defined disturbance $w^\star$, as $\boldsymbol{x}(\boldsymbol{p}^\star, w^\star, t)$. We then use this nominal solution to drive the dynamics of the high dimensional reservoir

$$\dot{\boldsymbol{r}}(t) = f(\boldsymbol{A}\boldsymbol{r}(t) + \boldsymbol{W}_{in}\boldsymbol{x}(\boldsymbol{p}^\star, w^\star, t)) \tag{3}$$

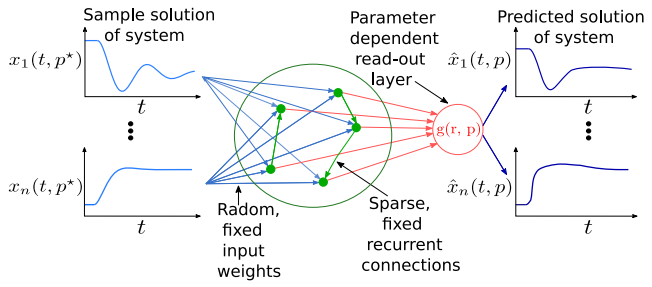

Fig. 2. General structure of ESNs.

where $\boldsymbol{A}$ is a fixed sparse random matrix $\in \mathbb{R}^{N\times N}$, $\boldsymbol{W}_{in}$ is a fixed dense random matrix $\in \mathbb{R}^{N\times M}$, $\boldsymbol{x}(\boldsymbol{p}^\star, w^\star, t) \in \mathbb{R}^M$ and $f(\cdot)$ is a non-linear function, e.g., $\tanh(\cdot)$. Because the reservoir in (3) is a non-stiff system, it is computationally cheap to simulate.

Once we have simulated the reservoir, we then want to learn a parameter-dependent mapping from this solution, $\boldsymbol{r}(t)$, to a predicted solution of the true system, $\hat{\boldsymbol{x}}(\boldsymbol{p}, w^\star, t)$. Typically, this is a linear-projection estimated by least squares. In our work however, we adopt a non-linear projection, the radial basis function (RBF), from [15]. RBFs are a method of interpolating unstructured data in high-dimensional spaces. The interpolant takes the form of a weighted sum of radial basis functions $\phi||.||$, e.g., Gaussian, linear or cubic, and frequently has some low order polynomials. Given n pairs of training data, $(\boldsymbol{x}, \boldsymbol{y})$, over which we fit an RBF, we estimate $\hat{\boldsymbol{y}}$ at a test datapoint, $\boldsymbol{x}^\star$, by (4)

$$\hat{\boldsymbol{y}} = rbf(\boldsymbol{x}^\star) = \sum_{i=1}^{n} \lambda_i \phi(||\boldsymbol{x}^\star - \boldsymbol{x}_i||) + \sum_{j=1}^{m} \gamma_j p_j(\boldsymbol{x}^\star) \tag{4}$$

where the weights $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ are learned during training. We use two RBF functions in this work. The first of these, rbf_β in (5a), takes as input the parameters we want to vary within our simulation, $\boldsymbol{p}$, and outputs a vector of weights, $\boldsymbol{\beta}(\boldsymbol{p})$. These weights then parameterize a second RBF, rbf_x in (5b), that maps from the pre-simulated reservoir, $\boldsymbol{r}(t)$, to the physical time series we want to predict, $\hat{\boldsymbol{x}}(t)$.

$$\boldsymbol{\beta}(\boldsymbol{p}) = rbf_\beta(\boldsymbol{p}) \tag{5a}$$

$$\hat{\boldsymbol{x}}(\boldsymbol{p}, w^\star, t) = rbf_x(\boldsymbol{\beta}(\boldsymbol{p}), \boldsymbol{r}(t)) \tag{5b}$$

We train this model by first sampling the boundary points of our multi-dimensional parameter space of interest. We then sample the remaining training points using Latin hypercube sampling, a low-discrepancy sequence, to generate a set of sample parameter vectors, $\boldsymbol{P} = [\boldsymbol{p}^1\ \boldsymbol{p}^2\ \dots\ \boldsymbol{p}^n]$. Then, $\forall \boldsymbol{p}^i \in \boldsymbol{P}$, we simulate the full order model of the power system and calculate the parameter dependent rbf_x weights, $\boldsymbol{\beta}(\boldsymbol{p})$, that map the reservoir states, $\boldsymbol{r}(t)$, to states of the physical system, $\boldsymbol{x}(t)$. Once we have a set of training weights, $\boldsymbol{\beta}(\boldsymbol{p})\ \forall \boldsymbol{p}^i \in \boldsymbol{P}$, we fit the radial basis function interpolation in (5a) to estimate these weights for parameters outside our training set. All RBF functions are fit using the Julia package `Surrogates.jl`.[1]

After a CTESN has been trained, predicting system behavior for different parameters consists of a simple evaluation of (5a) to determine the parameter dependent weights, $\boldsymbol{\beta}(\boldsymbol{p})$, followed by a matrix multiplication of these weights by the precomputed interpolation matrix of rbf_x. The dominate operation in this workflow is a simple matrix multiplication that scales $\mathcal{O}(N)$. This is significantly cheaper that the cost of a general implicit ODE solver that generally scales up to $\mathcal{O}(N^3)$.

[1] https://github.com/Surrogates.jl.

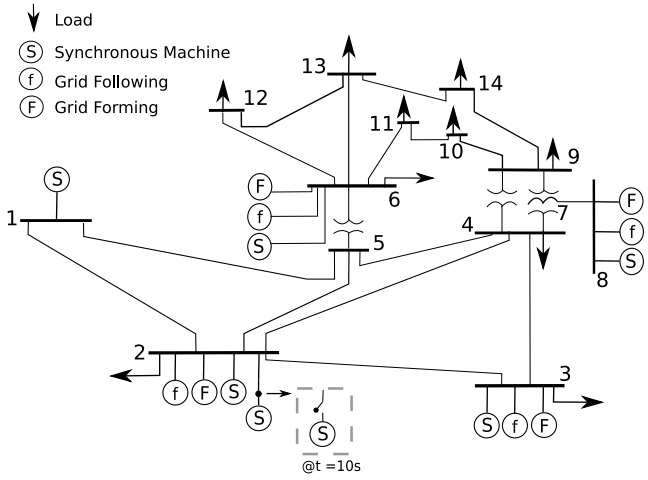

Fig. 3. Modified IEEE 14 bus model.

3. Methodology

The use case we consider within this work is predicting the frequency of the system following a large loss of generation for varying penetrations of both grid-following and grid-forming converters. The inclusion of these devices, and their relative penetrations, impacts key metrics of interest following large disturbances, e.g., the frequency nadir and rate of change of frequency (RoCoF).

3.1. Power system models

The main power system considered for analysis is a modified version of the IEEE 14 bus system, shown in Fig. 3. Generation capacity and active power set-points are adjusted to distribute active power generation more evenly among the units. An additional thermal unit is added to bus two that provides 4% of the systems total active power generation. The disturbance considered for all experiments is a trip of this generating unit.

To understand the impact of increasing penetration of CIG, we add both a grid-forming and grid-following converter to bus 2, 3, 6 and 8, as shown in Fig. 3. All generation resources have an active power set-point of 0.8 p.u. with respect to their own capacity. For each simulation we change the penetration of CIG by changing the installed capacity for every generation resource on the network. Therefore, in addition to changing the active power injection from synchronous units, we also change the total system inertia for each simulation.

All synchronous machines are thermal units with a Type I turbine governor, with an active power droop of 5%, and a Type II AVR [16]. The grid-forming converters, from [2], are operating in droop mode, with an active droop of 5%. The grid-following inverters, taken from [17], are injecting fixed P and Q into the network and we model both network current dynamics and voltage dynamics for nodes with non-zero capacitance. This results in a system with $x \in \mathbb{R}^{243}$ with an associated stiffness ratio of the system on the order of 10^6.

Additionally, to demonstrate the scalability of the proposed approach, we will also examine how both the CTESN relative computation time and accuracy scale for increasing system size in Section 4.3. These systems are constructed by connecting copies of the IEEE-9 bus system. All simulations are carried out using the Julia package `PowerSimulationDynamics.jl` [18,19] using `Differentialequations.jl` [20] and all code for the experiments described here can be found at [21].

3.2. Training the CTESN

To generate training data, we simulate the network subject to a loss of generation at $t = 10$ s. We vary the system operating conditions by changing (1) the % of generation that are CIG $\in [10, 80]$ and (2) the % of the CIG that are operating in grid-forming mode $\in [10, 40]$. The system is solved using the adaptive-stepping implicit IDA solver in Sundials [22]. The timesteps at which IDA solves the system determines the non-uniform sampling interval for training a CTESN. The input matrix, $\boldsymbol{W}_{in}$, is a random matrix whose entries are drawn from $x \sim \mathcal{N}(0, 1)$. The reservoir matrix, $\boldsymbol{A}$, is a Erdős–Renyi random graph with the total number of connections equaling the dimensional of the matrix. For rbf_x we use a linear RBF function while for rbf_β we use a cubic RBF function.

For this application, we train a CTESN to predict all line currents, nodal voltages for nodes with non-zero capacitance and the frequency of all synchronous machines on the network. To ensure that the reservoir is sufficiently excited by the solution of the true system, we normalize all state variables of the *nominal* solution, by subtracting their respective mean and dividing by the standard deviation, prior to feeding it as an input into the differential equations describing the evolution of the reservoir.

Within this work, we sample the parameter space and simulate the full-order physical system for the primary purpose of generate training data. In practice, depending on the application, the user/operator may follow such an approach. Alternatively, a CTESN can viewed as a valuable secondary use of the enormous amount of data that operators are generating daily through contingency analysis. In such cases, a CTESN can be trained in parallel until it is deemed to have met some accuracy requirements, from which time onward it can accelerate the exploration of contingencies.

4. Results

4.1. Accuracy of the surrogate

We first begin by examining the accuracy of a CTESN. We sample from the parameter space using a Sobol sequence to generate a test size of 200. Fig. 4 shows a histogram of the errors between the true frequency nadir from the full-order system and the predicted frequency nadir from the CTESN for each individual test for different training sizes. In this context, training size refers to the number of time-domain solutions of the full-order we use for training the CTESN.

For the case of a training size of 20, 95% of all the errors are $\leq \pm$ 4 mHz. This empirical level of accuracy in Fig. 4 supports the idea that these surrogate approaches can be used to understand parameter sensitivities and/or within optimization algorithms to ensure satisfactory system frequency bounds. Furthermore, as previously noted, these training samples were randomly generated using Latin hypercube sampling. With more intelligent sampling of the parameters space there exists the possibility for equivalent, or improved, accuracy with reduced training samples.

Another metric receiving increasing interest is the maximum RoCoF following a loss of generation. Fig. 5 shows a histogram of the errors between the largest negative RoCoF from the full-order system and the predicted largest negative RoCoF from the CTESN for each individual test. For both cases, the RoCoF was not a state variable but was estimated by calculating the maximum change in frequency over a time period of 0.1 s. Given this, the error distribution is less sensitive to training size, as expected.

Fig. 6 compares three frequency traces from the synchronous machine at Bus 1 for three different generation composition mixes. In the first subplot, the CTESN predicted frequency traces are overlaid with the true solution, the dashed black line, and the prediction error for each case is shown in the second subplot. Across all cases, the CTESN accurately predicts the dynamical behavior of the system, with a maximum prediction error of ≈ 10 mHz. This plot shows the capability of the CTESN to predict metrics such as the nadir and RoCoF, while also providing an estimation of the settling time and how damped the system response is.

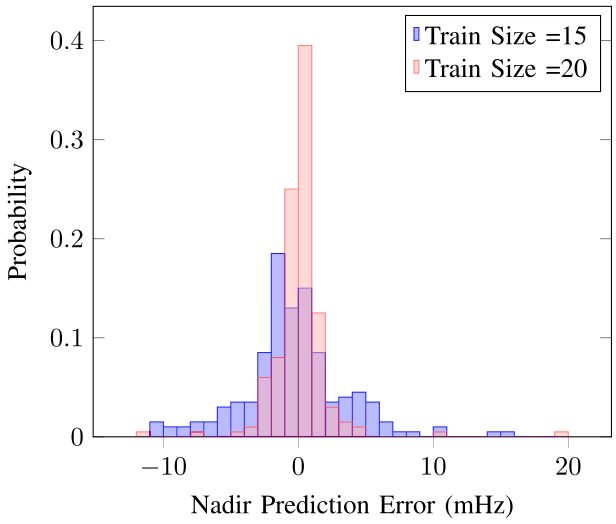

Fig. 4. Empirical probability distribution of worse-case CTESN frequency nadir prediction error for varying training size.

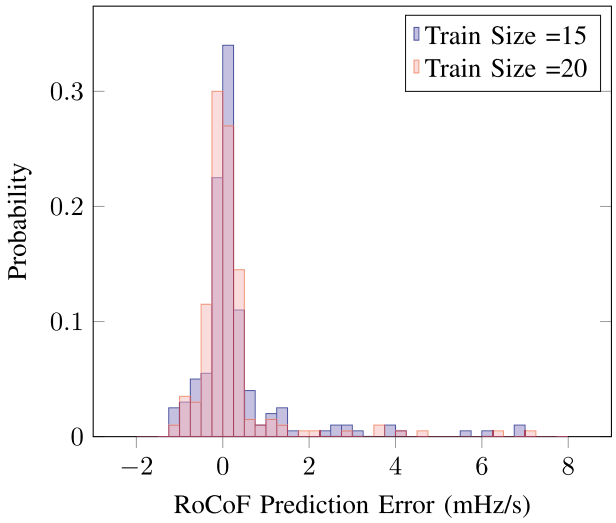

Fig. 5. Empirical probability distribution of worse-case CTESN RoCoF prediction error for varying training size.

4.2. Power system dynamic behavior

Once a CTESN is trained we can use it to rapidly predict the time series response for any parameter set within the upper- and lower-bounds specified during training. Fig. 7 shows the predicted frequency nadir by conducting a parameter sweep across the parameter range with a parameter granularity of 0.5%. This type of analysis can help system operators understand the required relative composition of grid-forming CIG to satisfy frequency containment requirements following the largest loss of generation. As expected, we see that for a fixed % of CIG, increasing the % of these resources that are grid-forming leads to a better frequency nadir.

Similarly, we can look at both the maximum RoCoF and frequency settling time, shown in Figs. 8 and 9 respectively, across the parameter range. We define the settling time as the time when the mean frequency

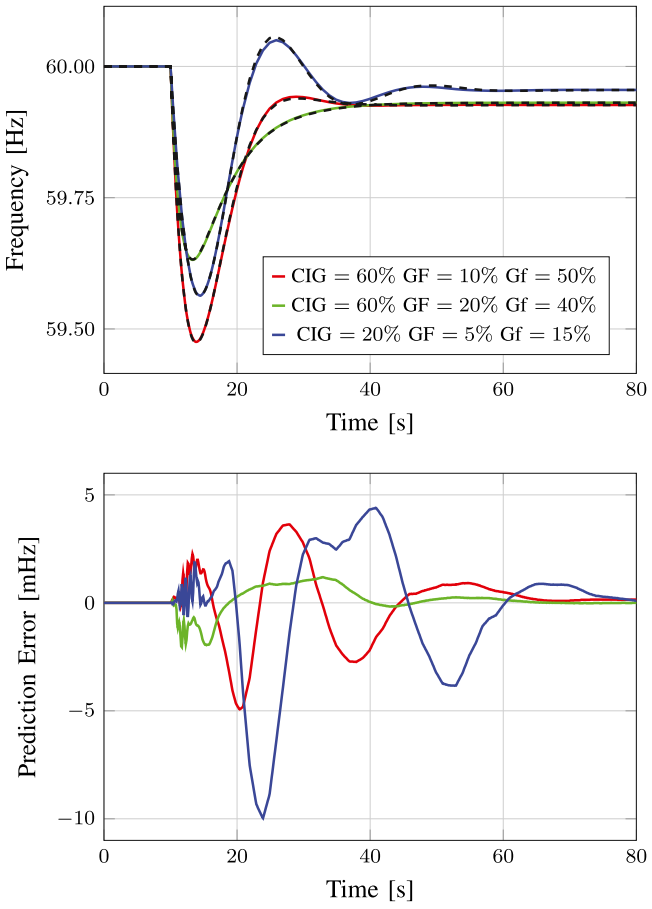

Fig. 6. CTESN predicted frequency (solid lines) vs true solution (dashed black lines) and prediction error.

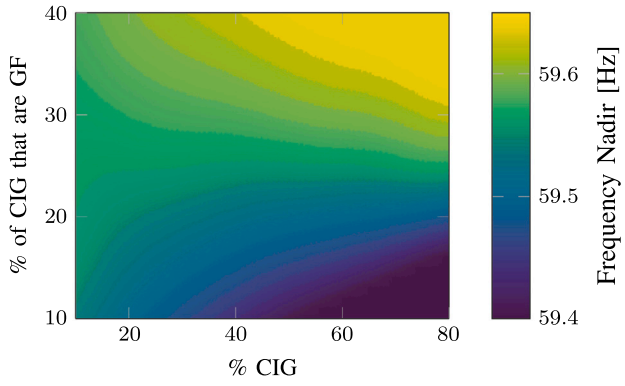

Fig. 7. CTESN predicted frequency nadir.

across all machines enters, and stays within, a band of ±20 mHz Hz of its final settling frequency. We see in Fig. 9 that the % of CIG that is grid-forming has a significant impact on the frequency settling time.

To further examine how the response of the system changes we consider three distinct operating conditions, and their associated frequency dynamics, in Fig. 6. We see that as we vary the % of CIG, the shape of the response can move from a 2nd-order response to more of a 1st-order response, in agreement with recent work [23]. Understanding the shifting behavior of the system with increased deployment of CIG will be critical for understanding and optimizing the deployment of these resources in large-scale networks.

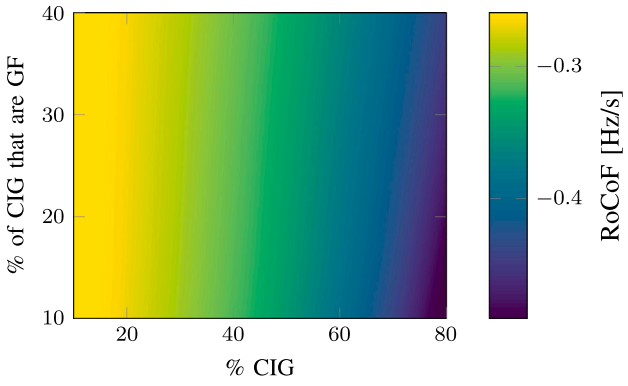

Fig. 8. CTESN predicted largest RoCoF.

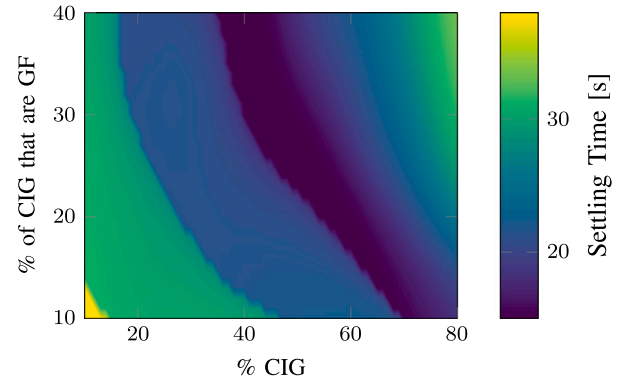

Fig. 9. CTESN predicted frequency settling time.

Table 1
CTESN training times.

	18 Bus	36 Bus	72 Bus	144 Bus
Train time[a]	52 s	186 s	201 s	626 s

[a]Excluding generation of training data.

4.3. Scalability and computation time

To benchmark the relative computation times, we consider systems of increasing sizes, all constructed using the WSCC 9-bus system as a modular building block and compare the computation time for both the CTESN and the full-order physical model. For each system size in Table 2, a CTESN was trained using 20 time-domain solutions of the full-order power system model with different CIG compositions. All computational benchmarking was carried out on a rack server with an Intel Xeon Processor E5-2623 v3.

Table 1 shows the total training time for each system. This time excludes the data generation, i.e., simulating the full-order physical model, as this is dependent on the specific simulation environment used, (e.g. PSCAD, etc.). For a given training size, the primary factors influencing the training time are the number of time-steps required by the adaptive ODE solver to solve the system as well the number of variables that we want to predict. For larger systems, parallelization of components of the training process could reduce training times.

The mean execution times, which included initializing and solving/predicting the time-domain solution across 20 simulations of each system, are shown in Table 2.

We see that, as expected, for increasing system size the computation acceleration achieved by the CTESN continues to improve. In each case the CTESN predicted the frequency of all synchronous machines, as well as nodal voltage with non-zero capacitance and line currents. Fig. 10

Table 2
Mean model execution times.

System size	CTESN	Full-order model	Improvement (×times)
18 Bus	0.312 s	0.849 s	2.71
36 Bus	0.797 s	4.37 s	5.47
72 Bus	1.492 s	29.19 s	19.56
144 Bus	2.9 s	109.83 s	37.83

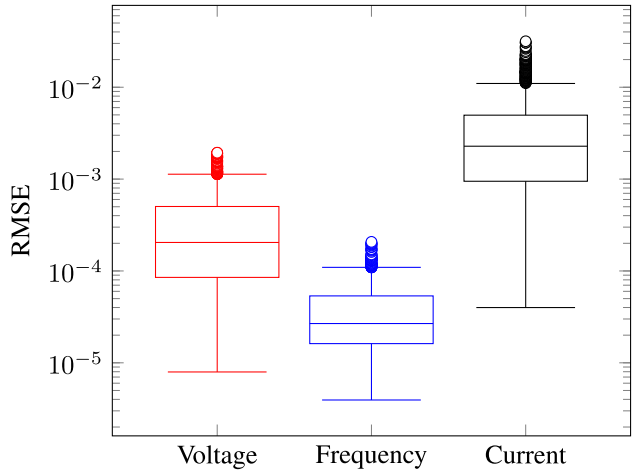

Fig. 10. Box plot of prediction RMSE trained on 25 solutions for the 144 bus test case system.

shows a box plot of the CTESN prediction RMSE across 50 runs of the 144 Bus system, each for varying CIG penetration levels. For each prediction of the 144 Bus system, the CTESN is predicting a total of 467 state variables. Fig. 10 shows that, even as we increase system size, the CTESN can predict all variables of interest with low error.

5. Conclusions

This work examines the application of CTESNs for accelerating parameter sensitivity analysis for power system time domain simulations. The results suggest that there may be suitable candidate use cases for CTESNs, depending on the accuracy requirements in question. One avenue that the current implementation of CTESNs may open is rapid predictions of system dynamics subject to a pre-defined set of disturbances. This can allow for inclusion of frequency and/or voltage constraints within market optimization algorithms e.g., procuring reactive power support and/or frequency containment reserves. Furthermore, CTESNs can be used for planning purposes, e.g., understanding shifting system dynamics, and controller gain tuning. The ability to sparsely sample parameter spaces and build accurate approximate models can lead to more targeted exploration using the full-order model.

The use of CTESNs for developing surrogates for *stiff*-systems is a nascent area of research [9] and there is a significant amount of further work required to characterize the suitability of this approach for power system analysis. While the use case presented here was focused on learning the solution at the transmission grid level, there is no reason why CTESNs cannot be used for many other applications in power systems, e.g., for learning the solution of distribution systems and/or networked microgrids. Additionally, there is recent work that attempts to develop reusable component level surrogates based on CTESNs [24]. For power system applications, this could open further acceleration techniques beyond those presented within this paper. Most notably, this could include using surrogates as modular building blocks that can be mixed with physics-based models to understand the response of the system to discontinuities not seen during the training phase, e.g., network topology changes and/or tripping of different resources.

Future works includes more intelligent sampling of the parameter space to minimize training requirements as well as exploring the idea of surrogates as modular building blocks for dynamical simulations. Additionally, some way of characterizing the accuracy of the surrogate, without the explicit use of a test set, would give users a greater confidence in integrating these surrogates into their workflow.

CRediT authorship contribution statement

Ciaran Roberts: Methodology, Investigation, Software, Writing – original draft, Validation. **José Daniel Lara:** Data curation, Methodology, Software, Writing – review & editing. **Rodrigo Henriquez-Auba:** Data curation, Software, Writing – review & editing. **Matthew Bossart:** Methodology, Writing – review & editing. **Ranjan Anantharaman:** Conceptualization, Software, Methodology, Writing – review & editing. **Chris Rackauckas:** Conceptualization, Methodology, Writing – review & editing. **Bri-Mathias Hodge:** Motivation, Writing – review & editing. **Duncan S. Callaway:** Motivation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] N. Hatziargyriou, J. Milanovic, C. Rahmann, V. Ajjarapu, C. Canizares, I. Erlich, D. Hill, I. Hiskens, I. Kamwa, B. Pal, P. Pourbeik, J. Sanchez-Gasca, A. Stankovic, T. Van Cutsem, V. Vittal, C. Vournas, Definition and classification of power system stability — Revisited & extended, IEEE Trans. Power Syst. 36 (4) (2021) 3271–3281, http://dx.doi.org/10.1109/TPWRS.2020.3041774.

[2] U. Markovic, O. Stanojev, P. Aristidou, E. Vrettos, D. Callaway, G. Hug, Understanding small-signal stability of low-inertia systems, IEEE Trans. Power Syst. 36 (5) (2021) 3997–4017, http://dx.doi.org/10.1109/TPWRS.2021.3061434.

[3] R. Henriquez-Auba, J.D. Lara, C. Roberts, D.S. Callaway, Grid forming inverter small signal stability: Examining role of line and voltage dynamics, in: IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, 2020, pp. 4063–4068, http://dx.doi.org/10.1109/IECON43393.2020.9255030.

[4] Y. Cheng, M. Podlaski, J. Schmall, S.-H.F. Huang, M. Khan, ERCOT PSCAD model review platform development and performance comparison with PSS/E model, in: 2020 IEEE Power & Energy Society General Meeting, PESGM, IEEE, 2020, pp. 1–5.

[5] Electromagnetic Transient Modeling Requirements, Tech. Rep., California ISO, 2021.

[6] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, et al., Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence, Tech. Rep., USDOE Office of Science (SC), Washington, DC (United States), 2019.

[7] G.S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-informed neural networks for power systems, in: 2020 IEEE Power & Energy Society General Meeting, PESGM, 2020, pp. 1–5, http://dx.doi.org/10.1109/PESGM41954.2020.9282004.

[8] N. Stulov, D.J. Sobajic, Y. Maximov, D. Deka, M. Chertkov, Learning model of generator from terminal data, Electr. Power Syst. Res. 189 (2020) 106742.

[9] R. Anantharaman, Y. Ma, S. Gowda, C. Laughman, V. Shah, A. Edelman, C. Rackauckas, Accelerating simulation of stiff nonlinear systems using continuous-time echo state networks, 2020, arXiv preprint arXiv:2010.04004.

[10] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[11] J. Li, M. Yue, Y. Zhao, 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, IEEE, 2020, pp. 1–6.

[12] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient pathologies in physics-informed neural networks, 2020, arXiv preprint arXiv:2001.04536.

[13] D. Li, M. Han, J. Wang, Chaotic time series prediction based on a novel robust echo state network, IEEE Trans. Neural Netw. Learn. Syst. 23 (5) (2012) 787–799.

[14] H. Hu, L. Wang, S.-X. Lv, Forecasting energy consumption and wind power generation using deep echo state network, Renew. Energy 154 (2020) 598–613.

[15] C. Rackauckas, R. Anantharaman, A. Edelman, S. Gowda, M. Gwozdz, A. Jain, C. Laughman, Y. Ma, F. Martinuzzi, A. Pal, et al., Composing modeling and simulation with machine learning in Julia, 2021, arXiv preprint arXiv:2105.05946.

[16] F. Milano, Power System Modelling and Scripting, Springer Science & Business Media, 2010.

[17] R.W. Kenyon, A. Sajadi, A. Hoke, B.-M. Hodge, Open-source PSCAD grid-following and grid-forming inverters and a benchmark for zero-inertia power system simulations, in: 2021 IEEE Kansas Power and Energy Conference, KPEC, IEEE, 2021, pp. 1–6.

[18] J.D. Lara, R. Henríquez-Auba, M. Bossart, D. Krishnamurthy, A. Plietzsch, C. Roberts, NREL-SIIP/PowerSimulationsDynamics.jl: V0.7.0, 2021, http://dx.doi.org/10.5281/zenodo.5525487.

[19] J.D. Lara, C. Barrows, D. Thom, D. Krishnamurthy, D. Callaway, PowerSystems.Jl — A power system data management package for large scale modeling, SoftwareX 15 (2021) http://dx.doi.org/10.1016/j.softx.2021.100747.

[20] C. Rackauckas, Q. Nie, Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in Julia, J. Open Res. Softw. 5 (1) (2017).

[21] Energy Modeling, Analysis and Control Group, GitHub, URL https://github.com/Energy-MAC.

[22] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Trans. Math. Softw. 31 (3) (2005) 363–396.

[23] R.W. Kenyon, A. Sajadi, B.-M. Hodge, Frequency dynamics with grid forming inverters: A new stability paradigm, 2021, arXiv preprint arXiv:2102.12332.

[24] R. Anantharaman, A. Abdelrehim, F. Martinuzzi, S. Yalburgi, E. Saba, K. Fischer, G. Hertz, P. de Vos, C. Laughman, Y. Ma, et al., Composable and reusable neural surrogates to predict system response of causal model components, in: AAAI 2022 Workshop on AI for Design and Manufacturing, ADAM, 2021.