

A Unified Approach for Learning the Dynamics of Power System Generators and Inverter-based Resources

Shaohui Liu, Weiqian Cai, Hao Zhu, Brian Johnson

Chandra Family Department of Electrical and Computer Engineering
The University of Texas at Austin

{shaohui.liu, stratoscwq, haozhu, b.johnson}@utexas.edu

Abstract

The growing prevalence of inverter-based resources (IBRs) for renewable energy integration and electrification greatly challenges power system dynamic analysis. To account for both synchronous generators (SGs) and IBRs, this work presents an approach for learning the model of an individual dynamic component. The recurrent neural network (RNN) model is used to match the recursive structure in predicting the key dynamical states of a component from its terminal bus voltage and set-point input. To deal with the fast transients especially due to IBRs, we develop a Stable Integral (SI)-RNN to mimic high-order integral methods that can enhance the stability and accuracy for the dynamic learning task. We demonstrate that the proposed SI-RNN model not only can successfully predict the component's dynamic behaviors, but also offers the possibility of efficiently computing the dynamic sensitivity relative to a set-point change. These capabilities have been numerically validated based on full-order Electromagnetic Transient (EMT) simulations on a small test system with both SGs and IBRs, particularly for predicting the dynamics of grid-forming inverters.

1. Introduction

The electric power infrastructure is witnessing transformative changes with the rapid electrification of various sectors and the increasing installation of renewable energy sources. These changes have led to growing prevalence of power electronics throughout the power grid [1]. While traditional rotating machines will continue to play a significant role in generation, storage, and motor-driven applications, the growing role of inverter-based resources (IBRs) presents significant challenges for grid operation and stability. In particular, the presence of IBRs increases the presence of fast transients and increases the complexity of power system dynamics [2]. While Electromagnetic Transient (EMT) simulations could offer greater precision over the conventional phasor-domain simulations, they are known to be computationally intensive and lack numerical scalability [3].

Developing effective yet tractable dynamic models for IBRs for system-level simulations is of high importance to facilitate the dynamic studies and stability analysis of power systems.

Recent advances in data-driven approaches, especially deep learning techniques based on the neural network (NN) models, have been increasingly utilized for power system dynamical modeling; see e.g., [4–6]. Enabled by the proliferation of sensing infrastructure, the ability to learn dynamics from real data is now feasible. With this approach, NNs offer high model accuracy and are generalizable to different operating conditions. However, a majority of these approaches aim to model the full system model along with all of its dynamical subsystems. Such methods incur very high model complexity and limit flexibility under various operating conditions seen in practice. A modularized design where each synchronous generator (SG) has its own dynamical model can be more adaptive and scalable [7, 8]. Despite this advancement, many existing models predict only fixed types of SG dynamic states, overlooking the underlying variations in dynamic models, which are not suitable for learning fast dynamics of IBRs. An alternative method involves neural ordinary differential equation (neural-ODE) based approaches [9], which parameterize the state-space model using a neural network, rather than directly predicting the states themselves [10–12]. While this approach is flexible in evaluation points, it necessitates the use of a differential equation solver to generate state predictions. This requirement not only adds a significant computational burden but also complicates system integration if modularized design is considered, particularly with the additional necessity of solving power flow equations.

The goal of our work is to put forth a unified approach for learning the models of dynamical components including both SGs and IBRs. To encompass high-order heterogeneous component models, our method builds upon the actual dynamic models to extract the key state variables that are sufficient for representing the system-level behaviors of these different resources. This

reduced-order model is helpful for lowering data rates and model complexity, and allows for the component to interact with the terminal bus only and decouple from the rest of system. The latter is very useful for modular integration of network-wide resources for attaining a full system model. It also allows us to build the learning model based upon the recurrent NN (RNN) to match the dynamic recursion form of the underlying physics-based model.

Nonetheless, the extremely fast dynamics of IBRs would require the RNN to have a very small time-step and thus high computational complexity. This issue is similar to the stability concern faced by classical numerical integration methods, as the prediction error per time step can get aggravated through the iterative propagation process. To tackle this issue, we put forth an innovative RNN design that mimics the higher-order numerical integral methods which are known to have much better stability guarantees than the first-order Euler's method. We propose the Stable Integral RNN (SI-RNN) model by introducing higher-order blocks of RNNs that are similar to the Runge-Kutta (RK) method [13, Ch. 9.3]. Our proposed SI-RNN can use a reasonable time-step to track IBR transients and greatly improve the stability regions, a crucial consideration in grid dynamic modeling [14]. We have further leveraged the SI-RNN model towards effectively predicting the dynamic sensitivity with respect to (wrt) a change of control set-point. This capability is useful for enhancing the generalizability of the trained SI-RNN model, but more importantly, it opens up new possibilities for applying to sensitivity-based optimization and control.

The main contributions of this research are two-fold. First, we present a unified approach to learn the dynamic model of an individual dynamic component like SGs and IBRs. Through reduced-order modeling and network decoupling, our approach can achieve flexible and efficient integration of all network-wide components in a modular fashion. Second, we develop a novel RNN-based learning method that can effectively capture the fast transients of IBRs. Inspired by the excellent stability guarantees of high-order integral schemes, our proposed SI-RNN can significantly reduce the time-step requirement and computation complexity. We demonstrate that this model is not only accurate in predicting the component's dynamics, but also capable of tracking the dynamic sensitivity to set-point changes, thus providing an effective tool for sensitivity-based optimal control tasks.

The rest of the paper is organized as follows. Section 2 presents the dynamic models of the SGs and grid-forming (GFM) IBRs. Section 3 develops a RNN-based model to learn the component-level dynamics, as

well as to predict the dynamic sensitivity. Section 4 demonstrates the validity and advantages of our proposed model over the regular RNN model, especially for predicting GFM dynamics. Conclusions and ongoing works are summarized in Section 5.

2. Dynamics Modeling

We first present the modeling of two types of important power system dynamic components, namely the synchronous generators (SGs) and grid-forming (GFM) inverters. Typically, the system dynamics are represented by a set of differential algebraic equations (DAEs), where the differential equations model the dynamics of the component state variables, and algebraic equations capture the power balance; see e.g., [15, Ch. 6-9]. Note that both SGs and IBRs have very high-order dynamics in practice. Nonetheless, we will focus on the low-complexity models of a reduced order that could well represent their behaviors at the power system level.

2.1 Modeling of Synchronous Generators (SGs)

We consider the the two-axis model of a synchronous machine as described in [16, Ch. 5], which will be used to incorporate the governor/exciter controls later on. There are four dynamic states for the SGs: the rotor angle δ and its derivative (frequency) ω , as well as the d/q-axis internal voltages E'_q and E'_d . Using the d/q-axis current magnitudes, I_d and I_q , per the power flow coupling, we obtain

$$\frac{d\delta}{dt} = \omega - \omega_o, \quad (1a)$$

$$M \frac{d\omega}{dt} = P_M - E'_d I_d - E'_q I_q - (X'_q - X'_d) I_d I_q - D(\omega - \omega_o), \quad (1b)$$

$$T_{do} \frac{dE'_q}{dt} = -E'_q - (X_d - X'_d) I_d + E_{fd}, \quad (1c)$$

$$T'_{qo} \frac{dE'_d}{dt} = -E'_d + (X_q - X'_q) I_q, \quad (1d)$$

where ω_o stands for the nominal frequency (e.g., the synchronous frequency), while the field voltage E_{fd} and the mechanical power P_M are determined by the SG controllers. The other parameters in (1) are mostly fixed constants for each SG, including inertial, damping, and reactance constants. Note that this fourth-order model in (1) is a simplified approximation of the higher-order dynamics including additional state variables like sub-transient voltages and fluxes.

The SG internal controllers include the excitation systems and governors [16, Ch. 8.6]. These controllers typically follow the proportional-integral control rule

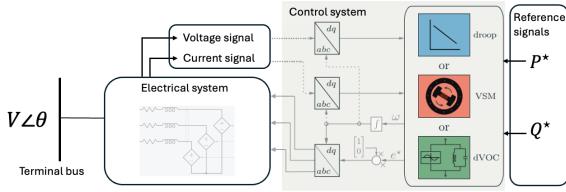


Figure 1: Schematic diagram of the averaged full-order GFM-inverter model with the reference signals, control system, and electrical system [17].

based on their corresponding set-points. Since the controller states are typically unobservable for SGs, we only consider simplified state-space models for SG controller states that have direct impacts on SG internal states. Specifically, the exciter control using the voltage reference set-point signal V^* can be represented by:

$$T_A \frac{dE_{fd}}{dt} = -E_{fd} + K_A (V^* - V), \quad (2)$$

where V stands for the bus voltage magnitude measured at the generator terminal. Similarly for the governor, it can be simplified as a first-order system using the power set-point signal P^* :

$$T_{SV} \frac{dP_M}{dt} = -P_M + P^* - \frac{1}{R_D} \left(\frac{\omega}{\omega_o} - 1 \right). \quad (3)$$

To determine the currents I_d and I_q , we solve the power flow equations between the SG's internal voltages and its terminal bus voltage phasor, $V\angle\theta$, given by

$$RI_d - X_q I_q = E'_d - V \sin(\delta - \theta), \quad (4a)$$

$$RI_q + X_d I_d = E'_q - V \cos(\delta - \theta). \quad (4b)$$

The SG states can be fully determined by the control signals V^* and P^* , as well as the terminal bus voltage $V\angle\theta$. If the control signals and terminal voltages are all given, the SG dynamics is decoupled from the remaining system. This will be used as the basis for component-wise dynamics learning.

2.2 Modeling of Inverter-based Resources (IBRs)

IBRs are broadly categorized into grid-following (GFL) and grid-forming (GFM) inverters based on the control objective design. While the majority of IBRs currently installed into the power grid are in GFL mode, the GFM inverters are envisioned to be more prevalently used to replace SGs thanks to their capabilities of providing voltage and frequency supports [18]. There have also been significant research efforts into developing a unified GFM model with various types of controller designs, as discussed in the recent paper [17]. Thus, this

work focuses on the modeling and implementation of GFM inverters.

Unlike SGs, the dynamics of the GFM inverters are dominated by their primary power control loop instead of physical laws, as depicted in Fig. 1. Typical GFM control strategies, including droop, virtual synchronous machine (VSM), and dispatchable virtual oscillator (dVOC), have been unified as a generic primary control model in [17]. This unification can be further simplified under assumptions of single inductor filter dynamics, pure inductive line impedance, stiff grid, and perfect grid voltage synchronization in the controller. There are seven dynamic state variables in the generic GFM model after the simplification: the phase angle of the inverter internal voltage δ , inverter voltage frequency ω , reference voltage magnitude E^* , filtered real and reactive power P_m and Q_m , and inverter output current magnitude in synchronous reference frame I_d and I_q that rotate at nominal frequency. Specifically, the generic GFM inverter model can be written as

$$\frac{d\delta}{dt} = \omega - \omega_o, \quad (5a)$$

$$\tau_f \frac{d\omega}{dt} = (\kappa_d + 1) (\omega_o - \omega) + \kappa_f (P^* - P_m),$$

$$\tau_v \frac{dE^*}{dt} = f_v(E^*) + \kappa_v (Q^* - Q_m), \quad (5b)$$

$$\tau_p \frac{dP_m}{dt} = \frac{3}{2} V I_d - P_m, \quad (5c)$$

$$\tau_p \frac{dQ_m}{dt} = -\frac{3}{2} V I_q - Q_m, \quad (5d)$$

$$L \frac{dI_d}{dt} = E^* \cos(\delta - \theta) - V + \omega_o L I_q, \quad (5e)$$

$$L \frac{dI_q}{dt} = E^* \sin(\delta - \theta) - \omega_o L I_d, \quad (5f)$$

where P^* and Q^* are respectively the active and reactive power reference set-points dispatched by the secondary controllers. Additionally, τ_f , τ_v , τ_p , κ_d , κ_f , and κ_v are fixed constant parameters and take different forms based on different GFM control strategies, as detailed in Table 1. Note that $f_v(E^*)$ also takes different forms per primary control strategies. While Droop and VSM consider the voltage deviation $V_o - E^*$ as $f_v(E^*)$ from nominal voltage V_o , dVOC considers a nonlinear function where

$$f_v(E^*) = E^* (V_o^2 - E^{*2}).$$

In general, both the component-level dynamics for SGs (1) and GFMs (5) can be represented as the first-order ODE system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u}), \quad (6)$$

	τ_f	τ_v	τ_p	κ_d	κ_f	κ_v
Droop	0	0	$\frac{1}{\omega_c}$	0	$\frac{1}{d_f}$	$\frac{1}{d_v}$
VSM	$\frac{m_f}{d_f}$	0	$\frac{1}{\omega_c}$	$\frac{d_d}{d_f}$	$\frac{1}{d_f}$	$\frac{1}{d_v}$
dVOC	0	$\frac{1}{\omega_o \kappa_2}$	0	0	$\frac{\omega_o \kappa_1}{(E^*)^2}$	$\frac{\kappa_1}{\kappa_2 E^*}$

Table 1: Parametric assumptions under which droop, VSM, and dVOC are transformed into generic primary control model. The definition of these parameter follows [17].

where the vector \mathbf{x} includes the dynamic state variables, vector \mathbf{y} contains the algebraic variables, while vector \mathbf{u} has the control set-points as the input. The vector $\mathbf{y} = [V, \theta]^\top$ essentially represents the voltage phasor of the terminal bus, which allows to decouple the dynamic component from the rest of network. Note that the current variables are also present in the dynamics, they can be solved from exactly using the internal voltage and terminal voltage, as in (4a). In addition, the vector \mathbf{u} comes from the reference set-point signals determined by the secondary controls. Furthermore, from the perspective of system-level behaviors, certain dynamic states in \mathbf{x} can be simplified, as well. Specifically, if we assume GFM currents remain relatively stable, (5e)-(5f) are equivalent to (4a). Additionally, E'_d can be eliminated in (1), as T'_{qo} is usually sufficiently small. If we further keep the internal controllers as implicit functions, the dynamic states \mathbf{x} in (6) are the trio: internal angle δ , internal frequency ω , and internal voltage magnitude E . These simplifications allow to develop a general approach for learning the model of component dynamics as follows.

Remark 1 (Model simplification). *The simplification of the dynamic states \mathbf{x} is justified by the negligible impact of high-frequency oscillation modes at the system level, as evidenced by our empirical findings from full-state Electromagnetic Transient (EMT) simulations for GFMs. These simulations demonstrate that the high-frequency modes of the internal states are effectively filtered out before influencing system-level dynamics, and confirming the low variability of GFM currents, whose transient disturbances are attenuated by the LCL filters [19]. Following this filtering process, our methods consider only state variables in $\mathbf{x} = [\delta, \omega, E]^\top$ without other dynamics. Notably, the datasets generated for dynamic learning reflect this smoothed, system-level behavior, enabling simplifications that avoid the unnecessary complexity of transient high-frequency modes in the modeling process, as detailed in Section 4.*

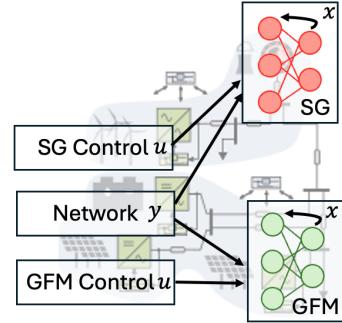


Figure 2: Modular architecture for modeling grid dynamics with each component represented as individual NNs. Each component has independent dynamic states, control inputs, and electrical input from the network [20].

3. Learning the Component Dynamics

We consider learning the dynamics of the component system in (6) by directly predicting the state trajectory $\mathbf{x}(t)$. Our design falls under the general framework for dynamics prediction in [7, 21]. As discussed later, by building the component-level predictive models, we could flexibly incorporate the system-wide SGs and IBRs into an integrated model, as depicted in Fig. 2.

To this end, we propose a recurrent neural network (RNN) model with a recursive structure that is nicely suited for predicting the dynamic evolution in (6). Using a sufficiently small time-step Δt , a simple numerical integration method like the Euler forward method converts (6) into the discrete-time system

$$\mathbf{x}_{t+1} \approx \mathbf{x}_t + \Delta t \cdot f(\mathbf{x}_t, \mathbf{y}_t, \mathbf{u}_t) \quad (7)$$

where the approximation accuracy depends on the time-step Δt , as discussed soon. Without knowing the underlying model (6), the integrator in (7) can be approximated using a multi-layer Elman RNN [22], given by

$$\mathbf{h}_t = \sigma(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{yh}\mathbf{y}_t + \mathbf{W}_{uh}\mathbf{u}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_1), \quad (8a)$$

$$\mathbf{x}_{t+1} = \sigma(\mathbf{W}_h\mathbf{h}_t + \mathbf{b}_h), \quad (8b)$$

where \mathbf{h}_t 's are the hidden states in the RNN model. The RNN has trainable parameters denoted by \mathbf{W}_h , \mathbf{b}_h which are of proper dimensions. For the activation function $\sigma(\cdot)$, typical choices include `relu` and `tanh`. RNNs are efficient in capturing temporal dependencies through the introduction of the hidden states. Basically, the hidden states act as a form of memory by retaining the important information on the past inputs, allowing the network generates outputs based on both the current input and the information it has captured in the past.

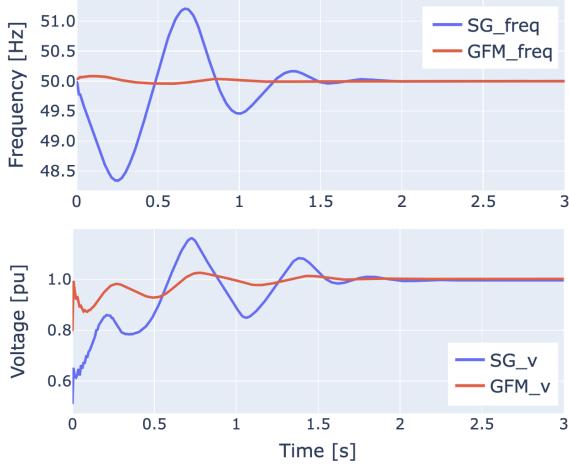


Figure 3: Comparison of SG and GFM state responses to a three-phase grounded fault of the 5-bus system in Fig. 7.

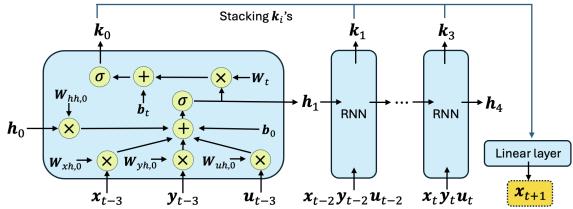


Figure 4: Diagram of the proposed SI-RNN dynamics learning framework.

While the RNN structure nicely fits with the dynamic learning task, the approximation error introduced by the discretization in (7) can adversely affect the accuracy of the trained RNN model. This is related to the well-known stability issue for numerical integration methods based upon (7). In order to satisfy the so-called CFL condition for the stability [13, Ch. 11], the step size Δt could be extremely small if (6) is stiff, or equivalently, having very fast dynamics. In general, the high-stiffness arises from systems with a high IBR penetration due to the fast dynamics and nonlinear behaviors of IBRs, as shown in Fig. 3. For dynamical systems, the approximation error of $f(\cdot)$ would accumulate across time, leading to severe error propagation. Thus, it is imperative to improve the RNN design to address the pronounced stability issue with IBR-rich power systems.

Inspired by higher-order numerical integration methods, we develop a new Stable Integral RNN (SI-RNN) design by imitating the fourth-order explicit Runge-Kutta (RK) method [13, Ch. 9.3]. In general, the explicit RK methods use the intermediate steps to reduce the truncation error. For example, consider the

fourth-order RK method given by

$$\mathbf{x}_{t+1} \approx \mathbf{x}_t + \Delta t \cdot \sum_{i=0}^3 c_i \mathbf{k}_i, \quad (9a)$$

$$\mathbf{k}_i = f(t + \alpha_i \Delta t, \mathbf{x}_t + \Delta t \sum_{j=0}^{i-1} \beta_{i,j} \mathbf{k}_j, \mathbf{y}_t, \mathbf{u}_t) \quad (9b)$$

where $\{c_i, \alpha_i, \beta_{i,j}\}$ and \mathbf{k}_i 's are respectively the weight coefficients and functional values for the intermediate steps. Compared with the Euler method with an accumulated approximation error in the order of $O(\Delta t)$, the fourth-order RK in (9) has the total error in the order of $O(\Delta t^4)$. This way, the latter can conveniently handle stiff problems by significantly expanding the stability region for the same time-step.

To imitate the design (9), our proposed SI-RNN leverages multiple historical data points by stacking the recurrent structure in (8a). Fig. 4 illustrates this idea by stacking the RNNs for the current input and three previous ones. Each one of the four \mathbf{k}_i 's is predicted by an individual RNN, with the hidden states \mathbf{h}_i 's also connected. Thus, our SI-RNN can be described as

$$\begin{aligned} \mathbf{h}_{i+1} = & \sigma(\mathbf{W}_{xh,i} \mathbf{x}_{t-3+i} + \mathbf{W}_{yh,i} \mathbf{y}_{t-3+i} \\ & + \mathbf{W}_{uh,i} \mathbf{u}_{t-3+i} + \mathbf{W}_{hh,i} \mathbf{h}_i + \mathbf{b}_{1,i}) \end{aligned} \quad (10a)$$

$$\mathbf{k}_i = \sigma(\mathbf{W}_{h,i} \mathbf{h}_{i+1} + \mathbf{b}_i), i = 0, \dots, 3 \quad (10b)$$

$$\mathbf{x}_{t+1} = \sigma(\mathbf{W}_\ell [\mathbf{k}_0, \dots, \mathbf{k}_3] + \mathbf{b}_\ell). \quad (10c)$$

Basically, each of the four RNNs is used to predict the corresponding \mathbf{k}_i in the RK method. Note that unlike the typical Elman RNN (8), the hidden states \mathbf{h}_i in (10) only track temporal dependencies among the output and four input steps. This special design helps in learning intricate dynamic patterns that might be neglected by the model simplification and state reduction steps without suffering from additional training difficulties, as detailed later in Section 4. Additionally, the last step in (10c) is a linear layer that matches the linear combination in (9). All the weight coefficients \mathbf{W} 's and \mathbf{b} 's are trainable parameters for the SI-RNN.

To simplify the notations, we represent the SI-RNN model as

$$f_R(\mathbf{x}, \mathbf{y}, \mathbf{u}; \Theta), \quad (11)$$

where Θ contains all trainable parameters in (10). To generate the training datasets, the trajectories of the component states and algebraic variables, along with the set-point signals, are dissected into tuples:

$$\{ \text{'Input': } \mathbf{X}_t, \text{'Output': } \mathbf{x}_{t+1} \}$$

with the RNN input

$$\mathbf{X}_t \triangleq \left\{ \{\mathbf{x}_{t-i}, \mathbf{y}_{t-i}, \mathbf{u}_{t-i}\}_{i=0, \dots, 3} \right\}.$$

As for the loss function used by the SI-RNN training, one can compare the predicted $\hat{\mathbf{x}}_{t+1} = f_R(\mathbf{X}_t; \Theta)$ with actual states to form

$$\mathcal{L}(\mathbf{x}; \Theta) \triangleq \frac{1}{T} \sum_{t=1}^T \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2. \quad (12)$$

We can use the SI-RNN model developed for an individual dynamic component to come up with an integrated system model of multiple components.

Remark 2 (Network integration). *The SI-RNN design is convenient for performing an integrated network analysis with multiple, heterogeneous dynamic components. First, the development of the SI-RNN model is applicable to any type of dynamic resources, either SGs or IBRs. Through the setup of \mathbf{y} and \mathbf{u} in Section 2, the component-level model only needs to interface with the rest of power network at the terminal bus, as shown in Fig. 2. This allows for separately training the SI-RNN for any dynamic component, followed up an integration step based on static power flow coupling only. This way, the network model can conveniently adapt to the connectivity of each component, as well as the network connectivity in terms of line status. This flexible integration of network-wide resources is unlike the neural-ODE method [10, 23] that predicts $f(\cdot)$ in (6) as a function of time t and cannot be used in the terminal coupling directly. We will further explore this network integration step in future.*

3.1 Dynamic Sensitivity

Our SI-RNN model allows for quickly predicting the dynamic sensitivity with respect to (wrt) the change of control set-points. This is important for a variety of power system dynamic analysis tasks. For example, dynamic sensitivity factors can facilitate the development of measurement-based system identification [24] and fast contingency screening [25]. In addition, the predicted state trajectories could be used in the design of optimal control strategies [26, 27].

While various kinds of dynamic sensitivity can be considered for the general model (6), we take the example of the $P^* - \omega$ sensitivity. This stands for the sensitivity in the frequency $\omega(t)$ wrt a small change in the active power set-point P^* . If one has the access to the numerical simulation data under different power set-points, the trajectory of this sensitivity can be approximated numerically. Using a small perturbation of ϵ on the set-point, the finite difference gives

$$J(\omega(t); \Delta P^*) \triangleq \frac{\Delta \omega(t)}{\Delta P^*} \approx \frac{\omega_\epsilon(t) - \omega_0(t)}{\epsilon}, \quad (13)$$

where $\omega_\epsilon(t)$ and $\omega_0(t)$ are the frequency trajectories under the set-points $P^* + \epsilon$ and P^* , respectively. Note

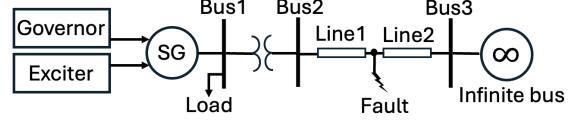


Figure 5: One-line diagram of the SMIB system for learning the SG dynamics.

that when generating the $\omega(t)$ and $\omega_0(t)$ trajectories, the same initial conditions and set-points are used, except for the change in the set-point P^* .

Since the SI-RNN model has been trained to match the underlying dynamics, it can be used for directly predicting the dynamic sensitivity for a variety of operating conditions. In fact, it conveniently facilitates this computation thanks to the fast gradient calculation via back-propagation. In addition to predicting the trajectory of $\omega(t)$ with given initial conditions and other inputs, the SI-RNN can efficiently produce the sensitivity by computing the partial derivative of $\omega(t)$ wrt P^* , as follows:

$$\begin{aligned} \hat{J}(\omega(t); \Delta P^*) &\triangleq \frac{\partial \omega(t)}{\partial P^*} \\ &= \frac{\partial \omega(t)}{\partial \mathbf{k}_0} \frac{\partial \mathbf{k}_0}{\partial \mathbf{h}_1} \frac{\partial \mathbf{h}_1}{\partial P^*} \\ &+ \frac{\partial \omega(t)}{\partial \mathbf{k}_1} \frac{\partial \mathbf{k}_1}{\partial \mathbf{h}_2} \left(\frac{\partial \mathbf{h}_2}{\partial P^*} + \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} \frac{\partial \mathbf{h}_1}{\partial P^*} \right) \\ &+ \frac{\partial \omega(t)}{\partial \mathbf{k}_2} \frac{\partial \mathbf{k}_2}{\partial \mathbf{h}_3} \left(\frac{\partial \mathbf{h}_3}{\partial P^*} + \frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_2} \left(\frac{\partial \mathbf{h}_2}{\partial P^*} + \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} \frac{\partial \mathbf{h}_1}{\partial P^*} \right) \right) \\ &+ \frac{\partial \omega(t)}{\partial \mathbf{k}_3} \frac{\partial \mathbf{k}_3}{\partial \mathbf{h}_4} \left(\frac{\partial \mathbf{h}_4}{\partial P^*} + \frac{\partial \mathbf{h}_4}{\partial \mathbf{h}_3} \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_2} \left(\frac{\partial \mathbf{h}_2}{\partial P^*} + \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} \frac{\partial \mathbf{h}_1}{\partial P^*} \right) \right) \right) \end{aligned} \quad (14)$$

where the second step follows from the chain rule, and can be efficiently calculated through recursion.

Remark 3 (Sensitivity-aware learning). *The SI-RNN model-based sensitivity (14) can be efficiently computed using the automatic differentiation and reverse accumulation in Pytorch, as implemented by the widely adopted backpropagation method in NN training [28]. This approach not only alleviates the need of simulating multiple trajectories in the numerical estimation approach based on (13), but also can be utilized for training the SI-RNN model to match a given sensitivity trajectory. By regularizing the SI-RNN loss function in (12) with the mismatch between the predicted and actual sensitivity, we can further improve the model accuracy and generalizability performance.*

4. Numerical Results

We present the numerical results for the proposed SI-RNN-based dynamic learning approach, by consid-

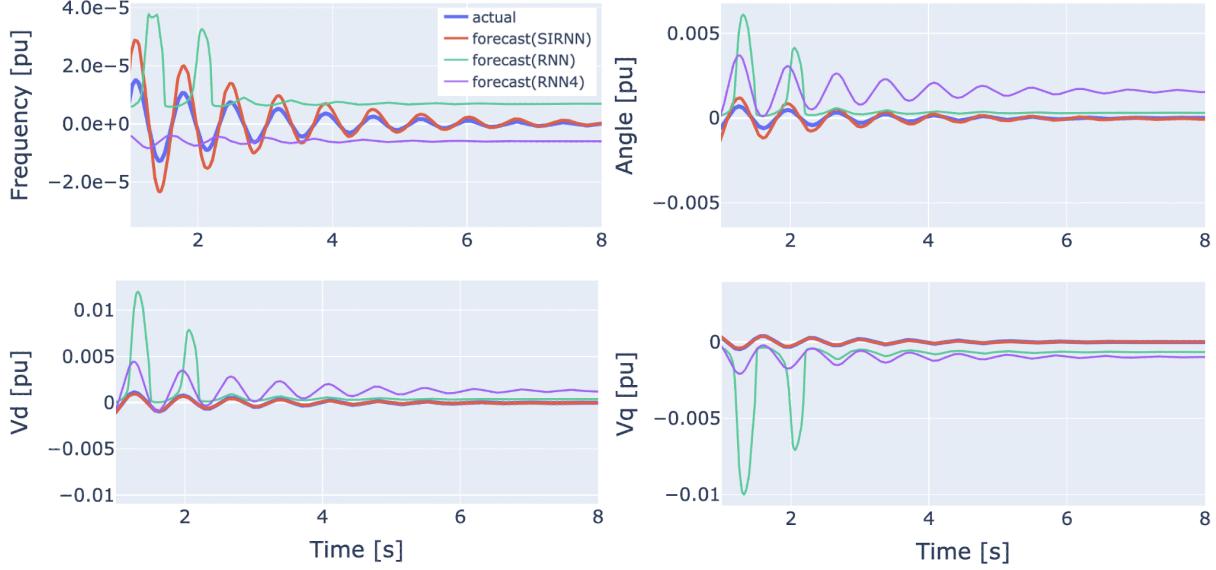


Figure 6: Comparisons of the proposed SI-RNN with RNN models on the SMIB test case in predicting the SG states for one fault scenario from the test set.

ering both SGs and GFMs. To generate the datasets of dynamic trajectories, we use the time-domain (TD) simulations based on the Simulink® in Matlab®. Both SGs and GFM have been represented using the three-phase full-order models, which are even more complex than those in Sec. 2. Particularly, the GFM model follows from [29], and has used the Electromagnetic Transient (EMT) simulation mode. To obtain the measurement data, we focused on educated-order states as in Remark 1, and a smoothing and down-sampling process has been used to filter the internal states of dynamic components. Additionally, the terminal bus voltage phasor data have been synthesized using the virtual Phasor Measurement Units (PMUs).

To evaluate the proposed SI-RNN’s performance in predicting dynamic responses to disturbances, we simulate different fault scenarios as detailed in specific tests later. The dynamic sensitivity is numerically computed by using a step change in P^* after the fault clearance. The simulated trajectories for all the scenarios are split into the training/test datasets with a 90%/10% division. To better compare the performance, we consider two error metrics to evaluate the trained models on the scenarios in the test dataset. The validation error measures the deviations of NN outputs of one-step predictions from ground truth, while the prediction error is with respect to the trajectories of iterative predictions using only initial conditions. Hence, the latter should be slightly higher as it is prone to error propagation. As for the NN model setups, the proposed SI-RNN uses a

total of four RNN blocks where the number of hidden features is 50 for each block, along with a final linear output layer. We also implemented the vanilla RNNs with the same parameter setup for comparisons. Both learning approaches have been implemented using the PyTorch library in Python. They are trained by the standard AdamW algorithm with the same convergence criteria. The NVIDIA Quadro® RTX 5000 is utilized for computation acceleration¹.

4.1 Learning the SG model

We first validate the performance of the proposed SI-RNN for learning the full-order SG and controller models. A sixth-order SG model with a hydraulic turbine and governor, as well as an IEEE Type1 excitation system, is connected to an infinite bus through some transmission components to form a single machine infinite bus (SMIB) system, as depicted in Fig. 5. A total of 400 fault scenarios are generated by changing the location of a three-phase grounded fault on the transmission line, and duration from 1 to 10 cycles. For each scenario, the post-fault responses are simulated for 14 seconds with a 100Hz sampling rate. We compare the test performance of the proposed SI-RNN model with the vanilla RNN one in predicting the state trajectory. We use the performance metric of the normalized mean

¹The simulation files, code, and results are available at: https://github.com/ShaoHuiLiu/IBR_Dyn_Learn

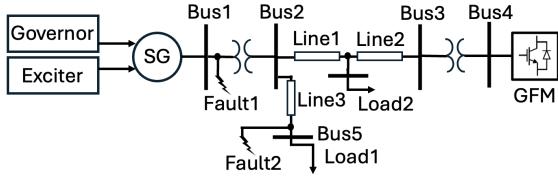


Figure 7: One-line diagram of the 5-bus test system with one SG and one GFM.

Table 2: NMSE of predicting the SG states for the SMIB test system.

State	Validation error			Prediction error		
	RNN	RNN4	SI	RNN	RNN4	SI
ω	1.00	0.99	0.03	1.04	1.00	0.07
δ	0.24	0.81	0.04	0.52	1.54	0.11
E_d	0.32	0.67	0.04	0.50	1.28	0.11
E_q	0.38	0.80	0.06	0.66	1.00	0.12

squared error (NMSE) given by

$$\text{NMSE} = \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \quad (15)$$

for predicting the ground-truth trajectory vector $\mathbf{x} = \{\mathbf{x}_t\}$. Examples of the test trajectories are presented in Fig. 6, and the detailed comparison results are listed in Table 2.

Fig. 6 and Table 2 show that both learning approaches can achieve good accuracy in predicting all the states in the validation test, while the SI-RNN slightly outperforms the RNN model. Nonetheless, when considering the iterative prediction using only the initialization, the proposed SI-RNN model exhibits much higher accuracy and numerical stability over the RNN model. Although adding additional historical trajectories will improve the accuracy of predicting oscillation modes as shown in Fig. 6, the deviations caused by the instability will lead to larger prediction error as shown in the column ‘RNN4’ in Table 2. This comparison speaks of the SI-RNN’s improved capability of tracking longer-term dependency and maintaining the stability.

4.2 Learning the GFM model

We further test the proposed SI-RNN for predicting the fast dynamics of IBRs, by using a 5-bus sub-system taken from the WECC 9-bus system. As shown in Fig. 7, there are 1 GFM and 1 SG, respectively connected to Bus4 and Bus1. The SG here has the same setup as in the previous SMIB system in Sec. 4.1, while a GFM with dVOC as primary control has replaced the SG con-

Table 3: NMSE of predicting the states for the SG, and the GFM along with its trajectory sensitivity.

	Validation error		Prediction error		
	SG	RNN	SI-RNN	RNN	SI-RNN
ω	0.78	0.05	0.81	0.17	
E	0.76	0.05	0.83	0.11	
GFM	RNN	SI-RNN	RNN	SI-RNN	
ω	0.04	0.04	0.13	0.11	
E	0.12	0.05	0.63	0.31	
Sensitivity	1.42	1.31	2.03	1.16	

nected to Bus4 in the original 9-bus system. A total of 200 fault scenarios have been simulated by using 2 different fault locations for a three-phase grounded fault of various fault durations from 1 to 10 cycles. For each scenario, the post-fault trajectories are generated by the EMT simulations for a time window of 3 seconds at a rate of 10kHz. Note that the simulation time here is significantly shorter than that of the previous SMIB system with the single SG, and this is because the inclusion of the GFM has led to much faster damping of the system oscillations. From the 10kHz EMT simulations, we further perform a filtering and down-sampling process to generate the measurement data at a rate of 100Hz for the states after reduction, as detailed in Remark 1.

The validation test results on GFM state predictions are similar to the SG test case in the SMIB system, with SI-RNN outperforms the vanilla RNN, as shown in Fig. 8 and Table 3. Interestingly, the SI-RNN model have shown much higher accuracy and stability in iterative predictions over the vanilla RNN model for both SG and GFM states in the reduced-order model. This shows the capability of the proposed SI-RNN model in predicting nonlinear and fast dynamics.

Using the GFM-based 5-bus system, we further compare the dynamic sensitivity predicted by the SI-RNN model with the numerical estimation method based on (13). Fig. 8 plots the predicted versus the estimated sensitivity trajectories for a change of active power set-point P^* at the GFM right after the fault clearance. Interestingly, the sensitivity in all scenarios from the test set are very similar to the numerical sensitivity calculated using simulated data. Nonetheless, the SI-RNN-based sensitivity trajectories are not smooth, which is consistent with the fact that the sensitivity is not time-dependent like states.

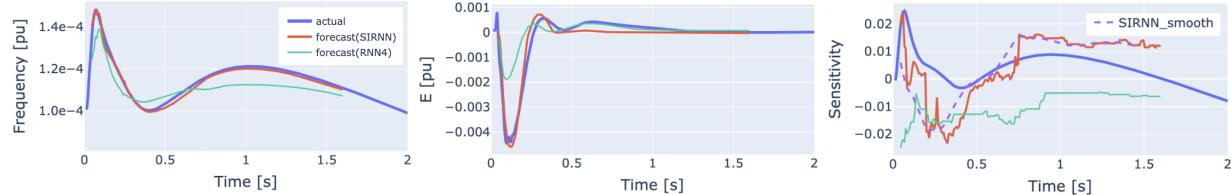


Figure 8: Comparisons of the proposed SI-RNN with RNN models on the 5-bus test case in predicting the GFM states and $\omega - P_c$ dynamic sensitivity for one fault scenario from the test set.

5. Conclusions

This paper develops a modularized learning framework to predict power system dynamics. We have effectively demonstrated a reduced-order modeling approach to learn the dynamics of both SGs and IBRs as individual dynamic components, through a novel RNN-based framework that integrates high-order integral schemes to enhance stability and predictability. Our proposed SI-RNN model not only copes with the rapid dynamics characteristic of modern power systems but also ensures robust state sensitivity tracking in response to control signal changes. Numerical validations on datasets generated by full-order EMT simulations from the SMIB system and a 5-bus system demonstrate the validity and advantages of our proposed model over the regular RNN model, especially for predicting the GFM dynamics and the dynamic sensitivity. Numerical results demonstrate our approach paves the way for a flexible and efficient integration of all network-wide components in a modular fashion. Our ongoing works include coupling different types of modules and learn system-wide dynamics. Furthermore, we are actively investigating using predicted dynamic sensitivity to design learning-based control systems [26, 30].

Acknowledgement

This work is partially supported by the NSF under Grants 2130706 and 2150571. This work is partially supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Award Number 38637.

References

- [1] L. Tolbert, T. King, B. Ozpineci, J. Campbell, G. Muralidharan, D. Rizy, A. S. Sabau, H. Zhang, W. Zhang, X. Yu, H. F. Huq, and H. Liu, “Power electronics for distributed energy systems and transmission and distribution applications: Assessing the technical needs for utility applications,” Oak Ridge National Lab, Oak Ridge, TN (USA), Tech. Rep., 2005.
- [2] J. A. Taylor, S. V. Dhople, and D. S. Callaway, “Power systems without fuel,” *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 1322–1336, 2016.
- [3] NERC, “Electromagnetic Transient Modeling for BPS-Connected Inverter-Based Resources — Recommended Model Requirements and Verification Practices,” *NERC: Atlanta, GA, USA*, Mar. 2023.
- [4] S. Zhao, F. Blaabjerg, and H. Wang, “An overview of artificial intelligence applications for power electronics,” *IEEE Transactions on Power Electronics*, vol. 36, no. 4, pp. 4633–4658, 2020.
- [5] H. Huang, Y. Lin, Y. Zhou, Y. Zhao, P. Zhang, and L. Fan, “Data-driven modeling of power system dynamics: Challenges, state of the art, and future work,” *iEnergy*, vol. 2, no. 3, pp. 200–221, 2023.
- [6] C. Roberts, J. D. Lara, R. Henriquez-Auba, M. Bossart, R. Anantharaman, C. Rackauckas, B.-M. Hodge, and D. S. Callaway, “Continuous-time echo state networks for predicting power system dynamics,” *Electric Power Systems Research*, vol. 212, p. 108562, 2022.
- [7] J. Li, Y. Zhao, and M. Yue, “Integrating learning and physics based computation for fast online transient analysis,” in *2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2023, pp. 1–5.
- [8] 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*, 2022.
- [9] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” *Advances in neural information processing systems*, vol. 31, 2018.
- [10] 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*, vol. 535, pp. 166–182, 2023.

[11] S. Zhang, K. Yamashita, and N. Yu, "Learning power system dynamics with noisy data using neural ordinary differential equations," in *2024 IEEE PES General Meeting*. IEEE, 2024, pp. 1–5.

[12] S. Chevalier, J. Stiasny, and S. Chatzivasileiadis, "Accelerating dynamical system simulations with contracting and physics-projected neural-newton solvers," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 803–816.

[13] M. T. Heath, *Scientific computing: an introductory survey, revised second edition*. SIAM, 2018.

[14] P. Ellinas, R. Nellikath, I. Ventura, J. Stiasny, and S. Chatzivasileiadis, "Correctness verification of neural networks approximating differential equations," *arXiv preprint arXiv:2402.07621*, 2024.

[15] R. B. Arthur and V. Vittal, *Power system analysis*. Pearson, 2000.

[16] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*. Prentice Hall, 1998.

[17] B. B. Johnson, T. Roberts, O. Ajala, A. D. Domínguez-García, S. V. Dhople, D. Ramasubramanian, A. Tuohy, D. Divan, and B. Kroposki, "A generic primary-control model for grid-forming inverters: Towards interoperable operation & control," in *HICSS*, 2022, pp. 1–10.

[18] Y. Lin, J. H. Eto, B. B. Johnson, J. D. Flicker, R. H. Lasseter, H. N. Villegas Pico, G.-S. Seo, B. J. Pierre, and A. Ellis, "Research roadmap on grid-forming inverters," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2020.

[19] A. Reznik, M. G. Simões, A. Al-Durra, and S. M. Muyeen, "lcl filter design and performance analysis for grid-interconnected systems," *IEEE Transactions on Industry Applications*, vol. 50, no. 2, pp. 1225–1232, 2014.

[20] D. Venkatramanan, M. K. Singh, O. Ajala, A. Dominguez-Garcia, and S. Dhople, "Integrated system models for networks with generators & inverters," *arXiv preprint arXiv:2203.08253*, 2022.

[21] 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)*. IEEE, 2020, pp. 1–6.

[22] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[23] C. Moya, G. Lin, T. Zhao, and M. Yue, "On approximating the dynamic response of synchronous generators via operator learning: A step towards building deep operator-based power grid simulators," *arXiv preprint arXiv:2301.12538*, 2023.

[24] K. Zhang, H. Zhu, and S. Guo, "Dependency analysis and improved parameter estimation for dynamic composite load modeling," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3287–3297, 2016.

[25] A. Al-Digs, S. V. Dhople, and Y. C. Chen, "Dynamic distribution factors," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4974–4983, 2019.

[26] Y. Gao, S. Wang, H. Hussaini, T. Yang, T. Dragičević, S. Bozhko, P. Wheeler, and S. Vazquez, "Inverse application of artificial intelligence for the control of power converters," *IEEE transactions on power electronics*, vol. 38, no. 2, pp. 1535–1548, 2022.

[27] M. Novak and T. Dragicevic, "Supervised imitation learning of finite-set model predictive control systems for power electronics," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1717–1723, 2020.

[28] P. Werbos, "Applications of advances in nonlinear sensitivity analysis," *System Modeling and Optimization*, pp. 762–770, 1982.

[29] A. Tayyebi, D. Groß, A. Anta, F. Kupzog, and F. Dörfler, "Frequency stability of synchronous machines and grid-forming power converters," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 8, no. 2, pp. 1004–1018, 2020.

[30] H. Issa, V. Debusschere, L. Garbuio, P. Lalanda, and N. Hadjsaid, "Artificial intelligence-based controller for grid-forming inverter-based generators," in *2022 IEEE PES innovative smart grid technologies conference Europe (ISGT-Europe)*. IEEE, 2022, pp. 1–6.