

# Dynamic Parameter Estimation with Physics-based Neural Ordinary Differential Equations

Xianghao Kong, Koji Yamashita, Brandon Foggo, and Nanpeng Yu  
*Department of Electrical and Computer Engineering*  
*University of California, Riverside*  
Riverside, California 92507 USA  
nyu@ece.ucr.edu

**Abstract**—Accurate estimation of dynamic parameters of generators is crucial to building a reliable model for dynamical studies and reliable operation of the power system. This paper develops a physics-based neural ordinary differential equations (ODE) approach to learn the parameters of generator dynamic model using phasor measurement units (PMU) data. We design a physics-based neural network to represent the swing equations of the power system dynamics. A loss function is defined as the difference between dynamic simulation results from the physics-based neural networks and pseudo PMU measurements. The parameters of generator dynamic model are iteratively updated using the neural ODEs and the adjoint method. By exploiting the mini-batch scheme in neural ODE training, the parameter estimation performance is significantly improved. Numerical study results on a 3-machine 9-bus system show that the proposed algorithm outperforms state-of-the-art baseline method in both computation time and dynamic parameter estimation accuracy.

**Index Terms**—Dynamic parameter, generator model, neural ordinary differential equations, adjoint method, phasor measurement unit.

## I. INTRODUCTION

High fidelity power system dynamic models are critical to both dynamic studies and reliable operation of the power system. Without accurate parameters, power engineers can not mimic historical disturbances and system events. But the nonlinearities and high dimensionality of the time-varying power system dynamic model make it challenging to estimate the parameters of generator dynamic models with high accuracy.

The advent of phasor measurement unit (PMU) provides system operators with time synchronized voltage and current phasor measurements in real-time [1]. The widespread deployment of PMUs around the world enables the development of data-driven algorithms to estimate the parameters of dynamic generator models in real-time [2]. This paper proposes a physics-based neural ordinary differential equations (ODE) approach to estimate these parameters with PMU data.

The topic of parameter estimation for power system and generator dynamic models has been studied extensively in the past [3], [4]. Given the space limitation, we briefly review a few representative research articles in this area. One of the first works formulates the dynamic parameter estimation problem as a nonlinear least squares problem using the sensitivities of the algebraic state of the system with respect to continuous dynamic state [5]. The parameters of the generator dynamic model are updated iteratively with a Gauss-Newton approach.

The sensitivities are derived with respect to the initial operating condition, which may not be sufficiently accurate for bulk power system disturbances.

In another work [6], a black-box neural network is adopted with input neurons represented by transient stability indices and the output neurons represented by parameters of the generator dynamic model. The lack of power system domain knowledge in the black-box model led to low estimation accuracy and poor sample efficiency. Reference [7] used the Extended Kalman filtering (EKF) for this same task, but the linearization step resulted in similarly low parameter estimation accuracy. To address the shortcomings of the EKF approach, reference [8] applied the unscented Kalman filter (UKF). However, the accuracy of UKF significantly reduces if the signal-to-noise ratio is low. Reference [9] utilized a weighted least squares method by using sensitivities of measured modal frequencies and damping to the parameters. The drawback of this approach is that it relies on estimating dynamic modes of the power system, which may not be sufficiently accurate. A Bayesian approach is proposed in [10], which formulates the dynamic parameter estimation as a maximum a posteriori (MAP) problem. The discrete adjoint method is used to estimate the gradient of the loss function with respect to the dynamic parameters. A local optimization approach called the quasi-Newton method is applied to solve the MAP minimization problem. This approach may lead to local optima when initial dynamic parameters are drastically different from the ground truth, or the posterior distribution is non-Gaussian. To deal with non-Gaussian posterior distribution, a Markov chain Monte Carlo (MCMC) method aimed at finding the global optima for the MAP estimator is proposed [11].

This paper extends the prior work [10], [11] by converting the forward solver of the ODEs representing power system dynamics into physics-informed neural networks. Then we calculate the loss function over mini-batches of samples based on the difference between dynamic simulation results from the neural networks and pseudo PMU measurements. The gradients of the loss function with respect to parameters are calculated based on the neural ODE technique and the corresponding adjoint method. Finally, the parameters of the dynamic model are updated with a quasi-Newton method.

The main contributions of this work are as follows:

- We adopt neural ODEs and the corresponding adjoint

method to learn the parameters of dynamic generator models online, which provides accurate estimates for parameter gradients.

- By designing physics-based neural networks to represent the forward functions of ODEs, we are able to leverage the parallel computing capabilities of graphics processing units (GPUs) to accelerate the dynamic parameter learning. This advantage becomes more apparent as the grid size increases.
- By leveraging the mini-batch scheme in updating dynamic parameters, the estimation time can be shortened and the buildup of errors in the ODE solver can be reduced.
- Comprehensive numerical studies demonstrate that our proposed method can accurately estimate not only the inertia constant but also mechanical power inputs using PMU data during transmission line events.

The rest of the paper is organized as follows. Section II presents the simplified power system dynamic model. Section III introduces the technical methods, including neural ODE, physics-based neural network design, and adjoint-based gradient calculation method. The numerical study results are provided in Section IV. Section V concludes the paper.

## II. POWER SYSTEM DYNAMIC MODEL

For ease of demonstration, a simplified dynamic model of a multi-machine interconnected power system is adopted. The proposed neural ODE-based parameter estimation technique can be applied to more complex dynamic models. The simplified model assumes that in the short observation period (a few seconds), the mechanical power input,  $P_m$ , is constant, and the classical model represents a generator with a constant voltage source behind a known transient reactance without damper winding. The terminal voltage  $V_i$  and current phasors  $I_i$  of all power plants are assumed to be measured by PMUs.

The differential equation of the classical generator model is represented by the swing equation shown in (1) [12].

$$\frac{M_0}{\omega_R} \ddot{\delta} = P_m - P_e, \quad (1)$$

where,  $M_0$  is a inertia constant,  $\text{MW} \cdot \text{s/MVA}$ ,  $\omega_R$  denotes the rated rotor speed of a generator,  $\delta$  is the angular position of a rotor relative to a synchronously rotating reference.

The algebraic equations coupling the classical generator model to the rest of the power system are represented by the following equations:

$$P_{e_i} = \Re\{E_i I_i^*\} = \Re\{Y_{\text{reduced}}^* E_i^2\}, \quad (2)$$

where  $P_{e_i}$  denotes the active power output of the generator  $i$ , and  $E_i$  is the generator  $i$ 's internal voltage phasor.  $\Re\{\cdot\}$  extracts the real part of a complex number, and  $*$  is the complex conjugate operation.  $Y_{\text{reduced}} = Y_{gg} - Y_{gs} Y_{ss}^{-1} Y_{sg}$  is the reduced Y-bus matrix.  $Y_{gg}$ ,  $Y_{gs}$ ,  $Y_{sg}$ , and  $Y_{ss}$  are submatrices of the admittance matrix of the entire system, where  $g$  and  $s$  correspond to the generator buses and other buses

in the system. Constant impedance loads are assumed to be embedded into the Y-bus matrix.

The internal voltage of the classical generator dynamic model can be calculated with PMU measurements at the terminal as:  $E_i = V_i + jx'_{d_i} I_i$ , where  $x'_{d_i}$  is the D-axis transient reactance of generator  $i$ . Thus, the active power output of unit  $i$  can be calculated as:

$$P_{e_i} = \Re\{Y_{\text{reduced}}^* (V_i + jx'_{d_i} I_i)^2\}. \quad (3)$$

## III. TECHNICAL METHODS

The overall framework of the iterative neural ODE-based dynamic parameter estimation algorithm is shown in Fig. 1. We feed the initial states and start/end timestamps into a physics-informed neural network representing the ODEs to produce estimates of future states and PMU measurements. A loss function that quantifies the difference between the estimated and observed system states is summed over every time stamp, and its gradients (via adjoint) are back-propagated to dynamic parameters. Finally, the dynamic parameters are updated using gradient descent.

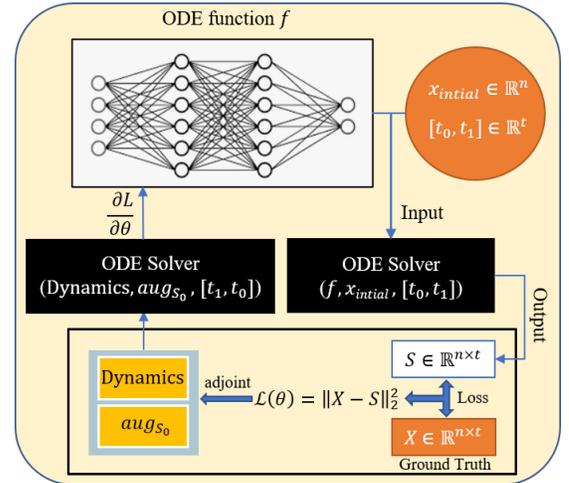


Fig. 1. Neural ODE-based dynamic parameter estimation framework.

### A. Overview of Neural ODEs

Neural ODEs are deep learning models which differ from standard machine learning methods in one key way - while standard machine learning methods map input variables to hidden variables for immediate use, Neural ODEs map input variables to the *derivative* of hidden variables which must first be integrated before their use [13]. Mathematically, We can write a standard neural network via the two equations  $s = f_{\alpha}^{\text{in}}(x), y = f_{\alpha}^{\text{out}}(s)$ . In contrast a neural ODE can be written as the two equations  $\frac{\partial s}{\partial t} = f_{\theta}^{\text{in}}(x), y = f_{\theta}^{\text{out}}(s)$  where now the function  $f_{\theta}^{\text{in}}$  returns time derivatives of the state variables instead of the state variables themselves. Here,  $\alpha$  and  $\theta$  represent the parameters of the two corresponding models.

The advantage of Neural ODEs over more typical models is that the hidden variable,  $s$ , is now actually a *smooth family* of hidden variables parameterized by a new variable,  $t$ . This

variable is typically used to represent a continuous “depth” of the network, but can also represent a time variable when modeling dynamical systems, in which we have the single hidden variable per time instance. Our work will adopt the latter interpretation. Since our goal is to model an existing function of time, this hidden variable can be used directly as our output (i.e.,  $f_{\theta}^{\text{out}}$  is the identity function).

The disadvantage of using Neural ODEs is that hidden variables need to be integrated. In practice, this means they must be sent through an ODE solver. Furthermore, we need to take gradients of the solver regarding the parameters so that the ODE must also be back-propagated through.

The existing ODE solvers can be divided into two groups. One group consists of adaptive-step ODE solvers, such as the Dormand–Prince method [14]; another group consists of fixed-step ODE solvers, such as the Euler method [15] and the Runge-Kutta method [16]. The latter is faster and more widely used in the industry than the former. This paper adopts the explicit fourth-order Runge-Kutta method with the 3/8 rule.

### B. Physics-Informed Neural Network Design

Neural ODEs typically use the expressivity of large neural networks to model the parameters of an ODE as a parametric black box. However, in our case, the ODE we are modeling possesses explicit mathematical expressions - namely, the swing equation. Thus, our neural network is explicitly modeled to the form of this known physical equation.

Let us assume that  $n$  generators and that  $|E_i|$  and  $Y_{bus}$  are given. Denote the element in the  $Y_{reduced}^*$  matrix located at the  $i$ -th row and  $j$ -th column as  $|Y_{ij}|e^{-\angle\phi_{ij}}$ . Then, the dynamic equations of the  $i$ -th generator can be formulated as follows:

$$\begin{cases} \dot{\delta}_i = g_i(\omega_i[\text{pu}]) \\ \dot{\omega}_i[\text{pu}] = h_i(\delta_{i=1,\dots,n}) \\ g_i = \omega_R(\omega_i[\text{pu}] - 1) \\ h_i = \frac{P_{m_i} - \sum_{j=1}^n |E_i||E_j||Y_{ij}|\cos(\delta_i - \delta_j - \phi_{ij})}{M_{0_i}}, \end{cases} \quad (4)$$

where  $\delta_i$  and  $\omega_i$  are the rotor angle and the rotor angle speed of generator  $i$  respectively. The unknown parameters are  $P_{m_i}$  and  $M_{0_i}$ . A neural network structure is strictly derived following (4). An example of this design for two generators is visualized in Fig. 2. In the case of multiple generators, we only need to extend this figure horizontally. It is too difficult for Neural ODE to solve all the unknown noisy parameters in one layer. Therefore, we use two nonlinear layers  $\exp(\cdot)$  and  $\ln(\cdot)$  to separate them, improving the nonlinear fitting ability of neural network. Since the reciprocal of  $M_{0_i}$  may be very small, it is amplified using a factor of  $k$ , and then an extra activation function is added before the output layer, which multiplies its input by a factor of  $\frac{1}{k}$ . The weights and bias are fixed in the first two layers. Thus, only the unknown parameters are updated. The neural network corresponding to  $g_i$  is relatively simple since every weight and bias are fixed.

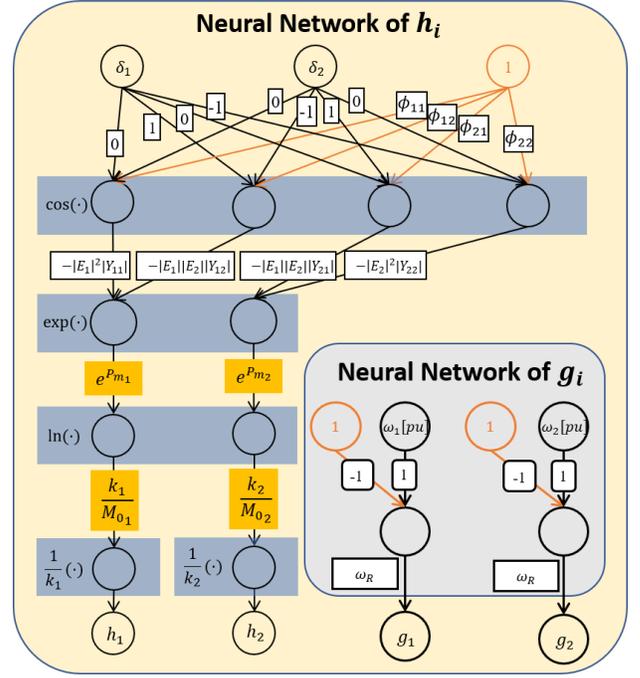


Fig. 2. The diagram of the physics-informed neural network design. Only the parameters in yellow boxes will be updated, others are fixed.

### C. Loss Function and Gradient Descent

The following mean square error loss function is used to train the physics-informed neural network:

$$L(s) = \sum_{t=t_0}^{t_1} \|\mathbf{x}(t) - s(t)\|_2^2, \quad (5)$$

where  $\mathbf{x}(t)$  represents the vector time-series for calculated state variables (i.e.,  $\delta_i$  and  $\omega_i$ ) from the PMU data.  $s(t)$  denotes the vector time-series of state variables estimated from the physics-based neural networks.

To calculate  $\frac{\partial L}{\partial \theta}$ , the gradients of  $L$  with respect to  $s(t)$  need to be computed first. The **adjoint method** is chosen to derive the gradient of  $L$  with respect to the estimated state variables [17]. Specifically, a new time series,  $\mathbf{a}(t) = \frac{\partial L}{\partial s(t)}$ , which we call the *adjoint* of  $s(t)$ , is created. It satisfies the following ODE [13]:

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^T \frac{\partial f(s(t), t, \theta)}{\partial s}, \quad (6)$$

where  $f$  denotes two physics-based neural networks,  $g$  and  $h$ , and  $\theta$  denotes the parameters of the physics-based neural network  $M_{0_i}$  and  $P_{m_i}$ .

The gradient of the loss function with respect to the neural network parameters,  $\theta$ , is a reverse integral over  $[t_0, t_1]$  [13]:

$$\frac{dL}{d\theta} = - \int_{t_1}^{t_0} \mathbf{a}(t)^T \frac{\partial f(s(t), t, \theta)}{\partial \theta} dt \quad (7)$$

The gradient calculation steps are summarized in Algorithm 1. Finally, we update the dynamic parameters  $\theta$  with the

**Algorithm 1** Calculate gradient w.r.t.  $\theta$  by adjoint method.

**Input:** parameters in the neural network  $\theta$ , time span  $[t_0, t_1]$ , final state  $s(t_1)$ , adjoint  $\frac{\partial L}{\partial s(t_1)}$

- 1:  $\text{augS}_0 = [s(t_1), \frac{\partial L}{\partial s(t_1)}, 0]$ ;
- 2: Calculate dynamics of augmented state:  
Dynamics =  $[f(s(t), t, \theta), -\mathbf{a}(t)^T \frac{\partial f}{\partial s}, -\mathbf{a}(t)^T \frac{\partial f}{\partial \theta}]$ ;
- 3: ODESolver(Dynamics,  $\text{augS}_0, t_0, t_1, \theta$ )  $\Rightarrow$   
 $[s(t_0), \frac{\partial L}{\partial s(t_0)}, \frac{\partial L}{\partial \theta}]$ ;
- 4: **return**  $\frac{\partial L}{\partial \theta}$ ;

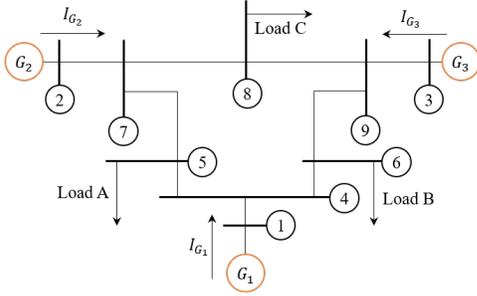


Fig. 3. WECC 3-machine-9-bus system [18].

limited-memory Broyden-Fletcher-Goldfarb-Shanno method, which updates each weight with its own squared gradient.

#### IV. NUMERICAL STUDY

##### A. Simulation Set up

We generate pseudo PMU measurements from dynamic simulation data from a 3-machine 9-bus system (see Fig. 3). The power system parameters and initial operating conditions are given in Tables I and II. A single transmission line is disconnected at 5s as a disturbance, and the simulation time is set at 10s. The initial values of six state variables are given in [18] as  $\delta_{i=1,2,3} = [0.0396, 0.3447, 0.2304]$ ,  $\omega_{i=1,2,3}[\text{pu}] = 1$ .  $V_i$  and  $I_i$  are calculated from  $\delta_i$  and  $\omega_i$ , and are considered to be our input data. Then, random Gaussian noise signal,  $G \sim N(0, 0.001)$  is added to the magnitude and angle of  $V_i$  and  $I_i$ , which is consistent with the estimated noise shown in a standard [19]. Finally,  $\delta_i = \angle E_i$  can be indirectly derived from  $E_i = V_i + jx_{di}I_i$ . Then,  $\omega_i$  is calculated by  $\omega_i[\text{pu}] = \frac{\delta_i[t+1] - \delta_i[t-1]}{4\pi\Delta T} + \omega_0$  [20] where  $\omega_0 = 1$ . Ground truth values of six parameters are:  $P_{mi} = \{0.7141, 1.6300, 0.8508\}$ ,  $M_{0i} = \{9.5515, 3.3333, 2.3516\}$ .

Two disturbance scenarios are studied. The first one is a single line tripping between nodes 5 and 7. The dynamic response for this disturbance is the same as the one illustrated in [18]. The second one disconnects the line between nodes 8 and 9. Note that the rotor angles of the generators continuously grow following the line tripping events because damper components are not modeled. The dynamic simulation time step is set as 1/120 s for both events. The PMU measurements of the generators are then downsampled to 30 Hz.

TABLE I  
POWER SYSTEM PARAMETERS [18]

Branch Impedance (pu)		Capacitance (pu)	Load $P_L$ & $Q_L$ (pu)
$z_{14} = 0.0576i$	$z_{45} = 0.0100 + 0.0850i$	$\frac{B_{38}}{2} = 0.1045i$	$P_{L5} = 1.25$
$x'_{d1} = 0.0608i$	$z_{46} = 0.0170 + 0.0920i$	$\frac{B_{44}}{2} = 0.0880i$	$Q_{L5} = 0.50$
$z_{27} = 0.0625i$	$z_{57} = 0.0320 + 0.1610i$	$\frac{B_{46}}{2} = 0.0790i$	$P_{L6} = 0.90$
$x'_{d2} = 0.1198i$	$z_{69} = 0.0390 + 0.1700i$	$\frac{B_{57}}{2} = 0.1530i$	$Q_{L6} = 0.30$
$z_{39} = 0.0586i$	$z_{78} = 0.0085 + 0.0720i$	$\frac{B_{69}}{2} = 0.1790i$	$P_{L8} = 1.00$
$x'_{d3} = 0.1813i$	$z_{89} = 0.0119 + 0.1008i$	$\frac{B_{78}}{2} = 0.0745i$	$Q_{L8} = 0.35$
		$\frac{B_{89}}{2} = 0.1045i$	

Note: Per unit values are calculated with 100 MVA base (and nominal voltage).

TABLE II  
INITIAL CONDITION OF THREE GENERATORS [18]

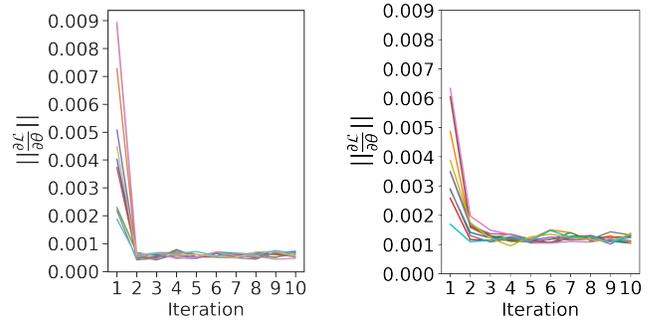
	$G_1$	$G_2$	$G_3$
Active power (pu)	0.7160	1.6300	0.8500
Reactive power (pu)	0.2700	0.0670	-0.1090
Terminal voltage (pu)	1.0400	1.0250	1.0250

##### B. Dynamic Parameter Estimation Results

The neural ODE fitting framework is tested with random  $[-10\%, 10\%]$  errors on the initial value of six parameters. The dataset for each disturbance contains 5 s of data, i.e., 150 data points. We set  $k$  for the generators as  $k_i = \{20, 7, 5\}$  to make sure the ratio  $\frac{k_i}{M_{0i}}$  is almost the same. In the training process, we set the batch size and batch time as 20 and 10 steps, respectively. This means that we randomly choose 20 data points as the initial state in the range of  $[t_0, t_1 - \frac{1}{3}]$ . Then, starting from each initial state, we step forward-in-time through the ODE solver for 10 contiguous timestamps and back-propagate against each of these segments separately.

1) *Parameter Estimation for Two Disturbances:* We display the norm of gradient  $\|\frac{\partial \mathcal{L}}{\partial \theta}\|_2$  against the training iterations in Fig. 4. As shown in the figure, our proposed algorithm converges within only two iterations for both disturbances.

We further quantify the accuracy and computation efficiency of our proposed neural ODE-based dynamic parameter estimation algorithm using the first disturbance. The estimation errors are calculated for the six unknown parameters with three different data lengths (1s, 3s, 5s). The relative estimation error (REE) =  $\frac{|Para-Para|}{Para} \times 100\%$  is used as the accuracy metric. We treat the algorithm in [10] as the baseline for comparison.



(a) Disturbance 1, learning rate= 0.1 (b) Disturbance 2, learning rate= 0.9

Fig. 4. The norm of six parameters' gradient of the neural ODE estimation framework on 2 disturbances. For each case, we run the experiment 10 times with different random seed. Each colored line represents an experiment.

TABLE III  
RELATIVE ESTIMATION ERROR (%) OF BASELINE AND  
NEURAL ODE-BASED METHOD

		Data Length			Initial REE
		1s	3s	5s	
Parameters	$P_{m1}$	2.36 / <b>1.50</b>	7.70 / <b>1.49</b>	7.47 / <b>1.26</b>	4.21
	$P_{m2}$	0.76 / <b>0.01</b>	5.05 / <b>0.12</b>	5.01 / <b>0.18</b>	5.77
	$P_{m3}$	1.05 / <b>0.41</b>	6.27 / <b>0.32</b>	6.52 / <b>0.46</b>	4.76
	$M_{01}$	5.80 / <b>2.86</b>	5.52 / <b>2.06</b>	6.31 / <b>3.09</b>	4.80
	$M_{02}$	4.36 / <b>3.19</b>	5.92 / <b>2.18</b>	5.88 / <b>3.67</b>	6.10
	$M_{03}$	<b>4.58</b> / 8.37	5.50 / <b>4.82</b>	<b>5.49</b> / 10.51	5.33
Average		3.15 / <b>2.72</b>	5.99 / <b>1.83</b>	6.11 / <b>3.20</b>	5.16

Note: Baseline / Physics-based Neural ODE Algorithm.

The training stops when the change of  $\|\frac{\partial \mathcal{L}}{\partial \theta}\|_2$  is less than a threshold value of 0.001. A large learning rate could lead to divergence, while a small one requires long computation time. For a fair comparison, the best learning rate is selected among 0.5, 0.05, and 0.005 to train our proposed and baseline model. The same random seeds are used in both methods.

2) *Parameter Estimation Accuracy*: The REEs are calculated and shown in Table III. The first and second number in each cell represent the REE of the baseline and the proposed algorithm respectively. When the PMU data length is 3s, our proposed algorithm achieves the lowest REE for the six unknown parameters. We can also observe that the estimation of mechanical power input is more accurate than that of generator inertia constant. Overall, the physics-based neural ODE algorithm outperforms the baseline algorithm in terms of estimation accuracy for most of the unknown parameters.

TABLE IV  
RUNNING TIME (S) OF BASELINE AND NEURAL ODE-BASED METHOD.

Data Length	Running Time (second)		Learning Rate
	Baseline	Neural ODE-based	
1s	8.38	3.78	0.5 / 0.5
3s	38.55	4.82	0.05 / 0.5
5s	100.25	4.67	0.05 / 0.5

Note: Baseline / Physics-based Neural ODE Algorithm.

3) *Computation Time & Scalability*: The computation time of the proposed and baseline algorithms for estimating dynamic parameters from different length of PMU data are reported in Table IV. Note that the learning rate of a scenario in an algorithm is selected such that divergence behavior is avoided. As shown in the Table IV, our proposed physics-based neural ODE algorithm has much shorter computation time than the baseline algorithm. When the data length is 3s, the running time of our model is just 4.82 seconds, which is nearly 8 times faster than the baseline model. The mini-batch scheme of neural network training is leveraged in the proposed algorithm, which greatly shortens the model running time and makes the algorithm more scalable in handling longer training dataset.

## V. CONCLUSION

This paper develops an online physics-based neural ODE algorithm to estimate the parameters of the generator dynamic model. By synergistically combining the swing equation and

neural ODE model, our proposed algorithm outperforms the state-of-the-art baseline algorithm in terms of estimation accuracy and computation time. Numerical studies on a 3-machine 9-bus power system show that our proposed model is capable of accurately estimating the dynamic parameters using just 3 seconds of noisy PMU data with 30 Hz sampling frequency. Furthermore, the entire dynamic parameter estimation procedure takes less than 5 seconds of computation time. In the future, we plan to further extend the proposed algorithm to consider more severe power system events and more realistic power system dynamic models by including damping factor, automatic voltage regulator module, and primary frequency controller of synchronous generators.

## REFERENCES

- [1] CIGRE C4.34, "Application of phasor measurement units for monitoring power system dynamic performance," technical report, CIGRE, 2017.
- [2] Z. Wang, L. Fan, and Z. Miao, "Data-driven dynamic model identification for synchronous generators," in *Proc. Annual North Am. Power Symp.*, Oct. 2019.
- [3] A. Monticelli, "Electric power system state estimation," vol. 88, pp. 262–282, Feb. 2000.
- [4] P. Zarco and A. Exposito, "Power system parameter estimation: a survey," vol. 15, pp. 216–222, Feb. 2000.
- [5] I. A. Hiskens and A. Koeman, "Power system parameter estimation," *J. Electr. Electron. Eng. Australia*, vol. 19, pp. 1–8, 1999.
- [6] A. Rahim and A. Al-Ramadhan, "Dynamic equivalent of external power system and its parameter estimation through artificial neural networks," *Int. J. Electr. Power Energy Syst.*, vol. 24, no. 2, pp. 113–120, 2002.
- [7] L. Fan and Y. Wehbe, "Extended Kalman filtering based real-time dynamic state and parameter estimation using PMU data," *Electr. Power Syst. Res.*, vol. 103, pp. 168–177, 2013.
- [8] M. González-Cagigal, J. Rosendo-Macías, and A. Gómez-Expósito, "Parameter estimation of fully regulated synchronous generators using unscented Kalman filters," *Electr. Power Syst. Res.*, vol. 168, pp. 210–217, 2019.
- [9] S. Guo, S. Norris, and J. Bialek, "Adaptive parameter estimation of power system dynamic model using modal information," vol. 29, no. 6, pp. 2854–2861, 2014.
- [10] N. Petra, C. G. Petra, Z. Zhang, E. M. Constantinescu, and M. Anitescu, "A Bayesian approach for parameter estimation with uncertainty for dynamic power systems," vol. 32, no. 4, pp. 2735–2743, 2016.
- [11] Y. Xu, C. Huang, X. Chen, L. Mili, C. H. Tong, M. Korkali, and L. Min, "Response-surface-based Bayesian inference for power system dynamic parameter estimation," vol. 10, no. 6, pp. 5899–5909, 2019.
- [12] P. Kundur, *Power Systems Stability and Control*. McGraw-Hill Education, 1st ed., Jan. 1994.
- [13] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," *arXiv preprint arXiv:1806.07366*, 2018.
- [14] M. Calvo, J. Montijano, and L. Randez, "A fifth-order interpolant for the Dormand and Prince Runge-Kutta method," *J. Comput. Appl. Math.*, vol. 29, no. 1, pp. 91–100, 1990.
- [15] J. Jacod and P. Protter, "Asymptotic error distributions for the Euler method for stochastic differential equations," *Ann. Probab.*, vol. 26, no. 1, pp. 267–307, 1998.
- [16] D. W. Zingg and T. T. Chisholm, "Runge-Kutta methods for linear ordinary differential equations," *Appl. Numer. Math.*, vol. 31, no. 2, pp. 227–238, 1999.
- [17] L. S. Pontryagin, *Mathematical theory of optimal processes*. CRC press, 1987.
- [18] V. Vittal, J. D. McCalley, P. M. Anderson, and A. A. Fouad, *Power System Control and Stability*. Wiley-IEEE Press, 3 ed., 2019.
- [19] A. S. Rana, N. Parveen, S. Rasheed, and M. S. Thomas, "Exploring IEEE standard for synchrophasor C37.118 with practical implementation," in *2015 Annual IEEE India Conf. (INDICON)*, pp. 1–6, IEEE, 2015.
- [20] J. Zhao, L. Zhan, H. Yin, F. Li, W. Yao, and Y. Liu, "Recent development of frequency estimation methods for future smart grid," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 354–365, 2020.