Physics-informed ML for power grids

Vineet Jagadeesan Nair (jvineet9@mit.edu)

May 18, 2023

1 Introduction and background

In this project, I explore the use of physics-informed machine learning for simulating and analyzing transmission grids, to aid planning as well as real-time operations. Scientific machine learning, parallelization, and high-performance computing can help accelerate these steps, especially for large-scale networks with millions of nodes and lines. I will be applying physics-informed learning methods for two applications:

1. Online estimation of key system parameters (that are either unknown or time-varying) from past measurements.

2. Neural network-based approaches for faster solutions of system states, to study transient dynamics as well as perform stability analysis.

Specifically, I will be focusing on methods to accelerate the analysis of fast dynamics and transient stability in high-voltage transmission systems. It is critical to be able to estimate system frequency in real-time since there is only a very limited amount of time available to respond to such grid frequency events as shown in fig. 1. Such deviations in frequency can occur due to major changes in either electricity generation or load. For example, the increasing penetration of clean, renewable energy sources implies that generation is much more variable and intermittent, and there's more uncertainty in forecasts. Furthermore, extreme weather events like heat waves, wildfires, or hurricanes can disrupt the system and/or cause sudden, unpredictable fluctuations in demand. Sustained, continuous drops in frequency can cause cascading failures and even complete collapse of the power system - this almost occurred during the Texas arctic winter storm in February 2021.

There are several resources that can provide frequency support. I will be focusing here on the fastest timescales, i.e. inertial response which occurs immediately after frequency disturbances (within $< 10\ s$). In today's grid, these are generally provided by the conventional synchronous generators - essentially the rotors and turbines associated with conventional fossil fuel-based thermal power plants like coal and natural gas. These machines have high rotational inertia that can dampen the disturbance and thus partially help mitigate frequency changes.

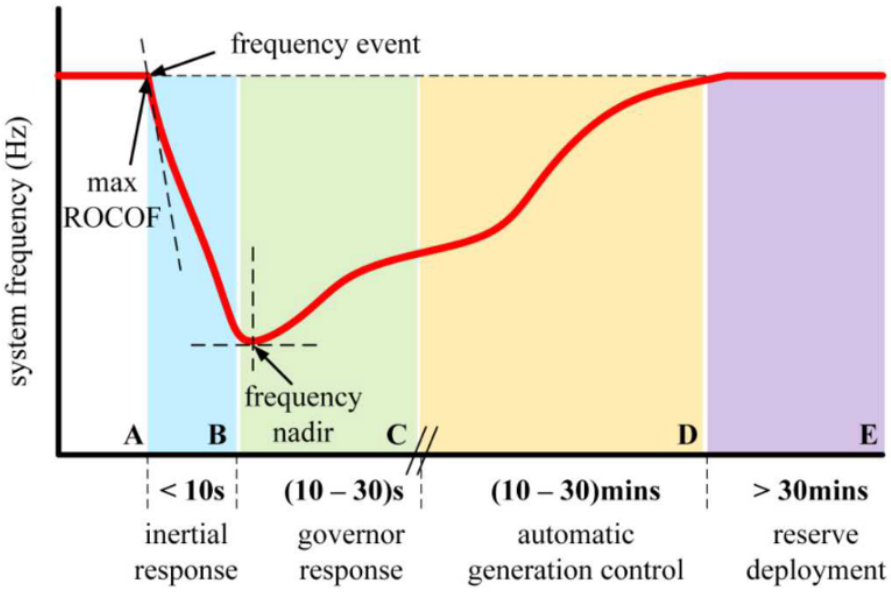


Fig. 1. System frequency response following an event.

Figure 1: Timelines for frequency response in transmission systems.

Grid frequency events are likely to be come more frequent and acute in the future as we move away traditional centralized, high inertia generators to distributed inerter-based energy sources like wind, solar and batteries that are based on faster power electronics and thus have much lower inertia. This increases the risk for potential stability issues, further emphasizing the need for faster transient stability analysis tools.

2 System modeling

The overall dynamics of the power system can be described by high-dimensional nonlinear ordinary differential equations (eq. (1)) for dynamic devices (mainly generators) along with algebraic equations (eq. (2)) for the network as a whole, resulting in a system of differential-algebraic equations (DAE):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{V}; \mathbf{p}) \tag{1}$$

$$\mathbf{g}(\mathbf{x}, \mathbf{V}; \mathbf{p}) = \mathbf{YV} - \mathbf{I}(\mathbf{x}, \mathbf{V}; \mathbf{p}) = \mathbf{0} \tag{2}$$

where $\mathbf{f}$ describes the ODEs, $\mathbf{x}$ represents all the state variables of the power system, $\mathbf{V}$ represents the voltage vector for all buses (or nodes) in the network, $\mathbf{I}$ is the bus current injection vector, $\mathbf{Y}$ is the network admittance matrix and $\mathbf{g}$ represents all the algebraic network equations (e.g. power balance).

2.1 Algebraic network equations

For the algebraic equations, optimal power flow is the key optimization problem that governs the operations of power systems. It describes the physical laws the system needs to obey such as

Kirchoff's laws, Ohm's law, line thermal limits, etc. These are exactly specified by the alternating current (AC) optimal power flow equations, shown below.

$$P_i + \mathbf{j}Q_i = V_i \sum_{k=1}^{n} \overline{\mathbf{Y}}_{ik} \bar{V}_k \tag{3}$$

Splitting these into real and imaginary components, we get:

$$P_i = |V_i| \sum_{j=1}^{n} |V_j| \left(\mathbf{G}_{ij} \cos\left(\delta_i - \delta_j\right) + \mathbf{B}_{ij} \sin\left(\delta_i - \delta_j\right) \right) \tag{4}$$

$$Q_i = |V_i| \sum_{j=1}^{n} |V_j| \left(\mathbf{G}_{ij} \sin\left(\delta_i - \delta_j\right) - \mathbf{B}_{ij} \cos\left(\delta_i - \delta_j\right) \right) \tag{5}$$

where P, Q are the real and reactive power, respectively, Y, G and B are the network admittance, conductance and susceptance matrices, V are the complex voltage phasors (With magntiude $|V|$ and angle δ). Since this is an NP-hard nonconvex problem, a commonly used approach for transmission systems is to instead make several simplifying assumptions that result in the linear direct current (DC) optimal power flow equations instead. This is what I used for the simulations in this project.

$$\min_{P_G, Q_G, |V|, \delta} \sum_{i \in \mathcal{G}} f_i \left(P_{Gi} \right) \tag{6}$$

$$\text{subject to} \tag{7}$$

$$P_{Gi} - P_{Di} = \sum_{k=1}^{n} B_{ik} \left(\delta_i - \delta_k \right), \forall i \in N \tag{8}$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}, \forall i \in G \tag{9}$$

$$-\overline{f_{ik}} \leq B_{ik} \left(\delta_i - \delta_k \right) \leq \overline{f_{ik}}, \forall (i,k) \in L \tag{10}$$

$$\delta_1 = 0 \quad \text{(Slack bus)} \tag{11}$$

$$|\delta_i - \delta_k| \leq \overline{\Delta \delta_{ik}}, \forall (i,k) \in L \tag{12}$$

These only account for power balance (eq. (8)), generator capacities (eq. (9)), line power flow limits or thermal ratings (eq. (10)) and voltage angle constraints. Thus DCOPF model was used to solve the system economic dispatch and determine the optimal power injections (setpoints) at both generator and load buses (or nodes) as well as the line power flows.

2.2 Differential equations

I'll mainly be focusing on the power system frequency dynamics that are important for determining transient stability. This is governed by the swing equation for each generator k:

$$m_k \ddot{\delta}_k + d_k \dot{\delta}_k + \sum_j B_{kj} |V_k| |V_j| \sin\left(\delta_k - \delta\right) - P_k = 0 \tag{13}$$

$$\underline{P_k} \leq P_k \leq \overline{P_k}, \ \mathbf{p}_k = [m_k, d_k, B_k]^\intercal, \ \mathbf{x}_k = [\delta_k, \omega_k]$$

where m_k defines the generator inertia constant, d_k represents the damping coefficient, B_{kj} is the $\{k, j\}$-entry of the bus susceptance matrix, P_k is the mechanical power of the k^{th} generator, k, j and δ_k, δ_j represent the voltage angles behind the transient reactance, for generators this is the rotor angle. $\dot{\delta}_k$ is the angular frequency of generator k, often also denoted as ω_k.

The swing equation is a heterogeneous nonlinear second-order differential equation with multiple variables. Thus, there is no known exact method to solve these analytically. Currently, these are most commonly solved via discretization and numerical schemes such as Euler or Runge-Kutta methods. However, these are computationally expensive, especially for large systems, which makes it challenging to solve these in real-time and quickly respond to disturbance. They also involve truncation errors and need very precise estimates for the parameters - this reduces their accuracy in practice and they may not be able to capture important effects [1]. Rapidly changing system conditions imply that these equations need to be solved very frequently (sub-second) and the move to distributed energy resources (DER) further increase the problem's dimensionality in terms of the number of nodes we need to study.

This motivates a data-driven approach for estimating parameters and solving eqs. (1), (2) and (13). For a realistic network with many generators, we may not know all the inertias m and damping coefficients d. In addition to large synchronous machines (conventional generators), renewable energy sources like wind turbines and other inverter-based resources like solar PV also have non-synchronous inertia called synthetic or virtual inertia. We can derive similar swing-equation-like models for these types of resources as well [2]. For example, grid-following virtual inertia inverter devices are described by:

$$\dot{\hat{\delta}}_k = \hat{\omega}_k \tag{14}$$

$$\tau_k \dot{\hat{\omega}}_k = -\hat{\omega}_k - K_{\mathrm{P},k} V_{\mathrm{q},k} - K_{\mathrm{I},k} \int v_{\mathrm{q},k} \tag{15}$$

where $V_{q,k}$ is the q-axis component of the bus voltage V_k (in a $d-q$ reference frame) with angle $\angle\delta_k$, $\tau_k, K_{\mathrm{P},k},$ and $K_{\mathrm{I},k}$ are the filter timeconstant, proportional gain, and integral synchronization gain. Grid-forming inverters are modeled as:

$$\dot{\delta}_{\mathrm{VI},k} = \omega_{\mathrm{VI},k} \tag{16}$$

$$\tilde{m}_k \dot{\omega}_{\mathrm{VI},k} = -\tilde{d}_k \omega_{\mathrm{VI},k} - P_{\mathrm{VI},k} \tag{17}$$

where $\tilde{m}_k$ and $\tilde{d}_k$ are the virtual inertia and damping constants. In addition to generators, we can also model certain frequency-dependent loads as:

$$d_k \dot{\delta}_k + \sum_j B_{kj} V_k V_j \sin\left(\delta_k - \delta_j\right) - P_k = 0 \tag{18}$$

In this project, I mainly focused on the ODE models for synchronous generators (eq. (13)), grid-forming inverters (eqs. (16) and (17)) and frequency dependent loads (eq. (18)).

Our grid is rapidly changing as we transition away from synchronous generators and increase the penetration of renewables. This reduces overall inertia and damping, making frequencies more susceptible to power deviations and potentially destabilizing the grid:

$$G(s) = \frac{\Delta f}{\Delta P_e} = -\frac{1}{2Hs + D}$$

Thus, estimating the time-varying overall inertia H and damping D characteristics of the network is crucial, especially as the generation mix becomes cleaner with increasing renewable penetration in the future

3 Methodolody, simulations and preliminary results

I considered and experimented with few different physics-informed learning approaches for ODE parameter estimation and solutions. Here, we focus mainly on the transient or dynamic stability of the transmission system, and not the steady state. I used the steady state DC OPF eq. (6) only to solve the dispatch (i.e. generator setpoints), and then used it to understand effects on generator angles and frequencies immediately following a disturbance, i.e when the generation suddenly deviates from these setpoints. The response of the system depends on the overall system inertia. Note that in all of these simulations, we're only interested in solving for the next few seconds ($<$ 5-10 s) since this is the timeframe for inertial frequency response.

3.1 Online estimation of ODE parameters

I first focused on estimating the best fit inertia and damping parameters, i.e. solving the inverse problem for ODE using an optimization-based approach. The form of the nonlinear ode $\ddot{\delta}(t)$ is known but the inertia and damping parameters are unknown. I started by considering a simple single machine infinite bus (SMIB) system as a proof of concept, which consists of a single generator connected to the grid.

$$m_1\ddot{\delta} + d_1\dot{\delta} + B_{12}V_1V_2\sin(\delta) - P_1 = 0$$

Considering our state $\mathbf{x}(t) = [\delta(t), \omega(t)]$, this results in the system of nonlinear 2nd order ODEs:

$$\begin{aligned}
\dot{\delta}(t) &= \omega(t) = x_1(t) \\
\dot{\omega}(t) &= \frac{1}{m_1}\left(P_1 - d_1\dot{\delta} - B_{12}V_1V_2 sin(\delta(t))\right) \qquad (19)\\
&= \frac{1}{m_1}\left(P_1 - d_1x_2(t) - B_{12}V_1V_2 sin(x_1(t))\right) \qquad (20)
\end{aligned}$$

V_1, V_2	B_{12}	P_1	m_1	d_1
1 p.u.	0.2 p.u.	[0.08,0.18] p.u.	[0.1,0.4] p.u.	[0.05,0.15] p.u.

Table 1: Parameters used for SMIB system [3].

From fig. 3a, we see that this system is very sensitive to the choice of parameters since the solution values for both the rotor angles $\delta(t)$ and frequencies $\omega(t)$ diverge quickly after the 1st few seconds. Here $\delta(t), \omega(t)$ corresponds to parameter values $[m, d] = [0.2, 0.1]$ $p.u$ while $\delta'(t), \omega'(t)$ correspond to $[m\prime, d\prime] = [0.1, 0.05]$ $p.u$.

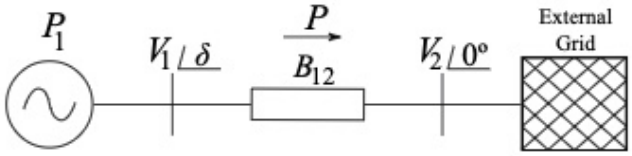
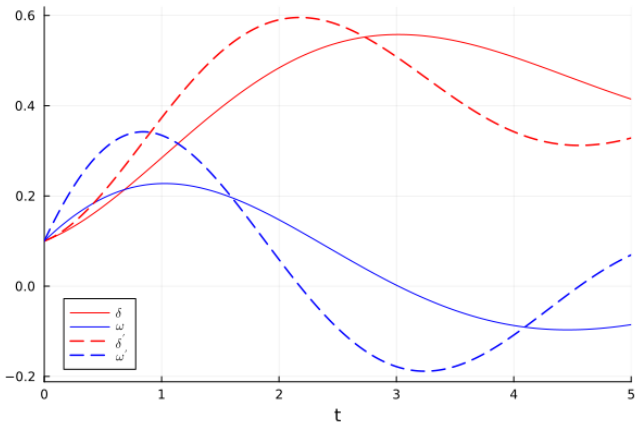
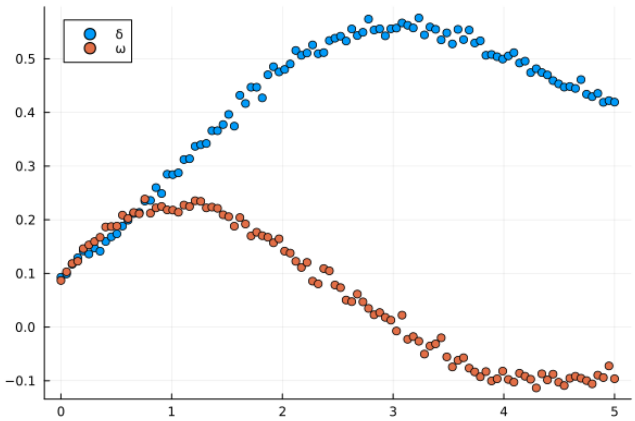


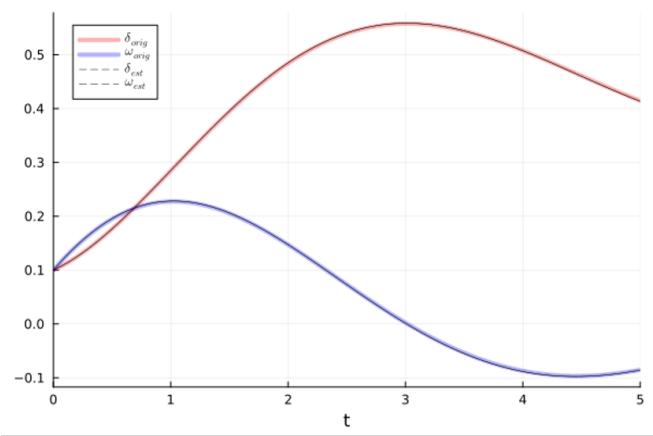
Figure 2: Single machine infinite bus system.



(a) Solutions for perturbed damping and inertia.



(b) Synthetic noisy data.



(c) True vs solutions using estimated parameters.

Figure 3: SMIB parameter estimation.

I generated synthetic data for angles and frequencies by 1st solving the forward problem and then adding some Gaussian noise to mimic measured data as shown in fig. 3b. I used initial conditions $u_0 = [\delta_0, \omega_0]$ with $\delta_0 = 0.1\ rad$ and $\omega_0 = 0.1\ rad/s$. I then minimized the $L2$ squared norm between the true values and estimated solutions when solving the ODE in eq. (20) with estimated parameters:

$$\hat{m}, \hat{d} = \arg\min_{m,d} \|(\delta^*(t) - \hat{\delta}(\hat{m}, \hat{d})\|^2 + \|(\hat{\omega}^*(t) - \omega(\hat{m}, \hat{d})\|^2 \tag{21}$$

$$s.t. \underline{\delta} \leq \delta \leq \overline{\delta}, \underline{\omega} \leq \delta \leq \overline{\omega} \tag{22}$$

We see from fig. 3c that we're able to recover the true parameters and the resulting forward solution almost exactly matches the true values.

3.2 Neural ordinary differential equations

Following a similar approach to [4], we can use a neural approach to estimate the gradients of the ODE and then use an integrator to solve the system:

$$\dot{x}(t) = \mathcal{NN}(\mathbf{x}(t), t; \hat{\mathbf{w}}) \tag{23}$$

where $\mathbf{x}(t) = [\delta(t), \omega(t)]$ and $\hat{\mathbf{w}}$ refer to the neural network parameters (weights and biases). Need to use conventional numerical methods to integrate/solve the resulting neural ODEs to get the NN predicted solution. This approach can be useful when the parameters are unknown or if there are sources of uncertainty or noise in any of the ODE terms. We can even apply this if we don't have any information on the structure of the ODE beforehand. For example, in the power systems case, this could occur when the grid operator does not know what types of devices are connected to each bus, i.e. the dynamics could be any of eq. (13)-(18). In this case, we use the same loss function as in section 3.1:

$$C(\hat{\boldsymbol{x}}(t), \hat{\mathbf{p}}) = \sum_{t=t_0}^{T} \|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t, \hat{\boldsymbol{w}})\|_2^2$$

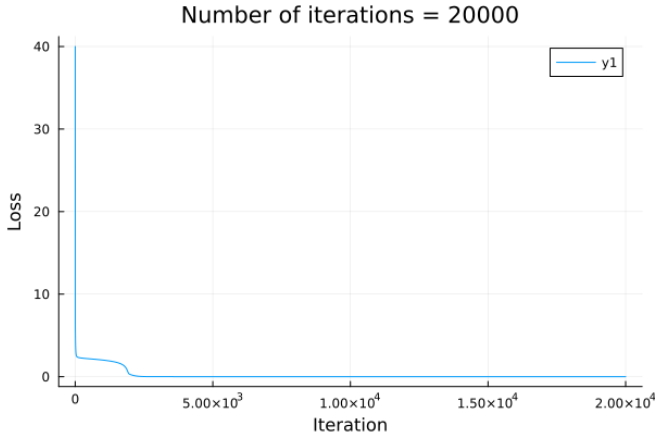
and derive the gradients of this loss w.r.t neural (NN) network parameters using the adjoint sensitivity approach. The adjoint of the ODE is defined by the system of equations:

$$\dot{\lambda} = -\lambda^* \frac{\partial \mathrm{f}}{\partial \hat{\mathbf{x}}} \tag{24}$$

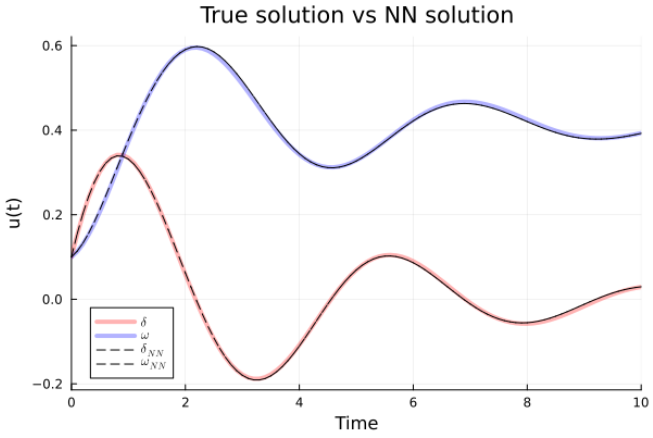
$$\dot{\mu} = -\lambda^* \frac{\partial \mathrm{f}}{\partial \hat{\mathbf{w}}} \tag{25}$$

where $f(\cdot) = \mathcal{NN}(\mathbf{x}(t), t; \hat{\mathbf{w}})$ represents the neural net approximating the derivative. The adjoints or vector-Jacobian products for the derivative of the NN w.r.t to each of the inputs (at any $\mathbf{y}$) can be calculated using its pullback $B^{\hat{\mathbf{w}}}_{NN}(\mathbf{y})$. Thus, $B^{\hat{\mathbf{w}}}_{f}(\lambda) = \lambda^* \frac{\partial f}{\partial \hat{\mathbf{w}}}$. We also know that $\mu(T) = 0$ and $\lambda(T) = \frac{\partial C}{\partial \hat{\mathbf{x}}}$. While solving the reverse ODE adjoint problem, we also need to add jumps $\frac{\partial C}{\partial \mathbf{x}(t_i)} = 2(\mathbf{x}_i - \hat{\mathbf{x}}(t_i, \hat{\mathbf{w}})$ to $\lambda(t_i)$ for each data point $(t_i, \mathbf{x}_i)$. After solving the reverse problem, the gradient of the loss function is given by $\mu(0) = \frac{\partial C}{\partial \hat{\mathbf{w}}}$ which we can use to update the neural network parameters I was able to get good convergence by just using simple gradient descent with a learning rate $\eta = 0.001$ (as seen in fig. 4a but we could also use any other stochastic gradient descent variants like `ADAM`. The neural ODE performs very well and is able to accurately predict the outputs even farther out into the future. I also tested the pre-trained neural network on a different initial condition than the one used during training and it still does a great job at accurate

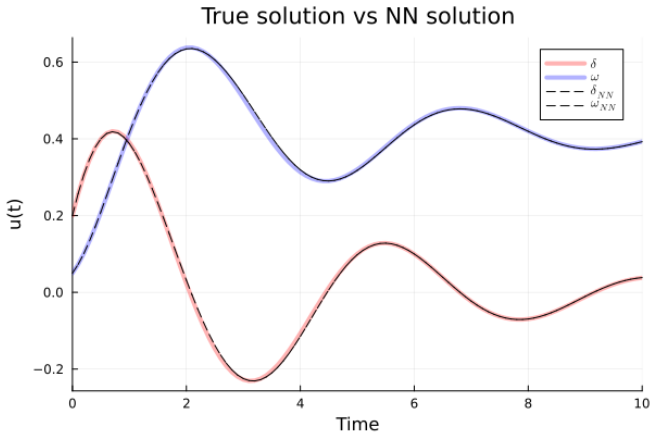
predictions as seen in fig. 4c, indicating good generalization rather than overfitting to the training data.



(a) Neural ODE training loss.



(b) Comparison of true vs Neural ODE solutions.



(c) Comparison of true vs Neural ODE solutions for new initial conditions.

Figure 4: Neural ODE results.

3.3 Physics-informed neural networks

Finally, I tried to also use a Physics-informed neural network (PINN) approach to simultaneously estimate the parameters and predict solutions [5]. As opposed to the neural ODE approach where the NN computes the gradients, the PINN attempts to directly predict the states by exploiting prior knowledge about the underlying physics and structure of the ODE. Here, I trained the PINN to approximate the temporal evolution of the rotor angle and then applied automatic differentiation to calculate the frequencies, given the initial conditions. These can be compared against the true $\delta, \dot{\delta}$ values to optimize the neural network parameters via supervised learning.

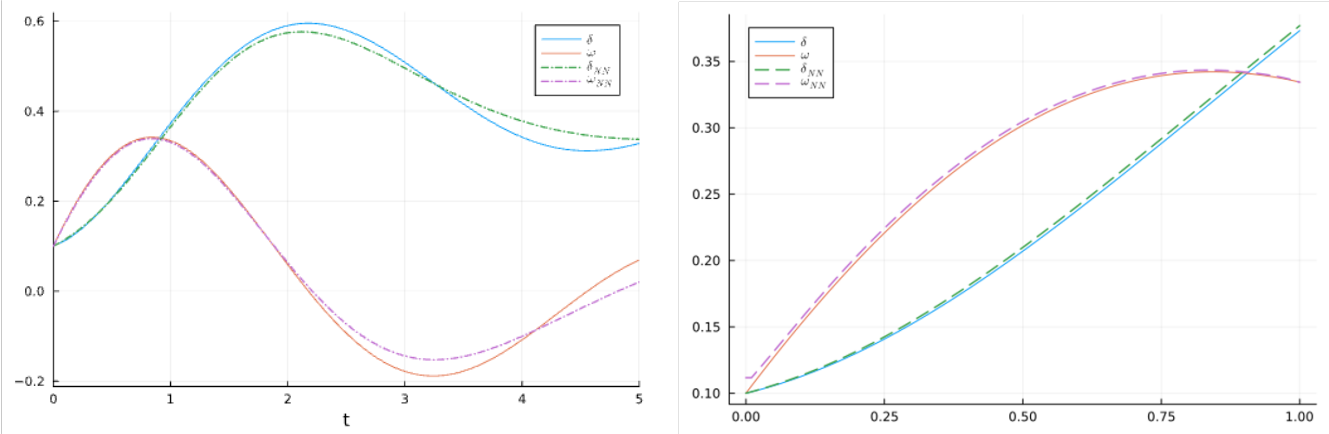
$$\hat{\delta}(t) = NN(t\ \delta_0, \omega_0, m, d),\ \hat{\omega} = \dot{\hat{\delta}}(t) = AD(\hat{\delta}(t))$$

Simultaneously, if needed, we can also learn the true parameters $\mathbf{p} = [m, d]$ by adding a physics regularization term to the loss function to enforce the dynamics. This allows us to learn the true

system parameters in an unsupervised manner. Thus, we combine the ODE output prediction error $\mathcal{L}_{\mathbf{x}}$ with the parameter estimation error $\mathcal{L}_{\mathbf{p}}$:

$$\mathcal{L}_{\mathbf{x}} = \left\| \delta_k(t) - \hat{\delta}_k(t) \right\|^2 + \left\| \dot{\delta}_k(t) - \dot{\hat{\delta}}_k(t) \right\|^2 \tag{26}$$

$$\mathcal{L}_{\mathbf{p}} = \left\| \hat{m}_k \ddot{\hat{\delta}}_k(t) + \hat{d}_k \dot{\hat{\delta}}_k(t) + \sum_j B_{kj} V_k V_j \sin\left(\hat{\delta}_k(t) - \hat{\delta}_j(t)\right) - P_k \right\|^2 \tag{27}$$



(a) Predictions for 1 second into the future.

(b) Predictions over 5 seconds.

Figure 5: True solutions versus predictions by the trained PINN.

It was more challenging to obtain good results using the PINN approach. As seen in fig. 5b, the PINN is able to accurately for the next few timesteps fairly accurately ($\approx$ 0-2 s) but the errors increase quickly for predictions farther ahead as seen in fig. 5a. I tried tuning the neural network hyperparameters (number of layer and neurons) extensively and also experimented with other methods like batch normalization, dropout, weight decay and learning rate schedules. The final results shown in fig. 5 were obtained with a network with 3 hidden layers with 16 neurons each and sigmoidal activation functions, using the ADAM optimizer. I'm still working on investigating why the PINN is not able to converge well but it may be because this is a stiff nonlinear ODE and the ODE error stagnates after a certain preventing the network from learning further.

4 Conclusions and future work

Overall, I found that the neural ODE approach (in section 3.2) performs better than the PINN (in section 3.3) at predicting the ODE solutions. Thus, this can be combined with the optimization based approach (from section 3.1) to also simultaneously estimate the ODE parameters. However, an important caveat for both of these methods is that training them may be more challenging in practice due to limited availability of good quality data on angle/frequency measurements. Thus, we may need to leverage other methods to deal with these potential data sparsity issues.

While the PINN offers the potential to simultaneously estimate both solutions and parameters, it currently under performs compared to the other methods in terms of accuracy. This will be one

of my focus areas for future work. Some ideas I have to improve the PINNs are: (i) improving initializations of the parameters $\hat{\mathbf{p}}$ to achieve faster convergence, (ii) optimizing code, incorporating parallelization, and leveraging GPUs, as well as (iii) fusing the PINN with other nonlinear filtering-based approaches for online estimation (e.g. Unscented Kalman filters). Such a hybrid approach could be used to generate better initial estimates, and the PINN can then be used to fine-tune these parameters while also predicting the system states. I will also consider how to use feedback to improve the neural network's weights in real-time based on incoming measurements from devices like phasor measurement units (PMU). Finally, I aim to extend all 3 of the studied methods to validate on much larger, realistic networks. Eventually, fig. 6 shows the overall workflow I propose to implement for using physics informed learning methods for studying power system dynamics.

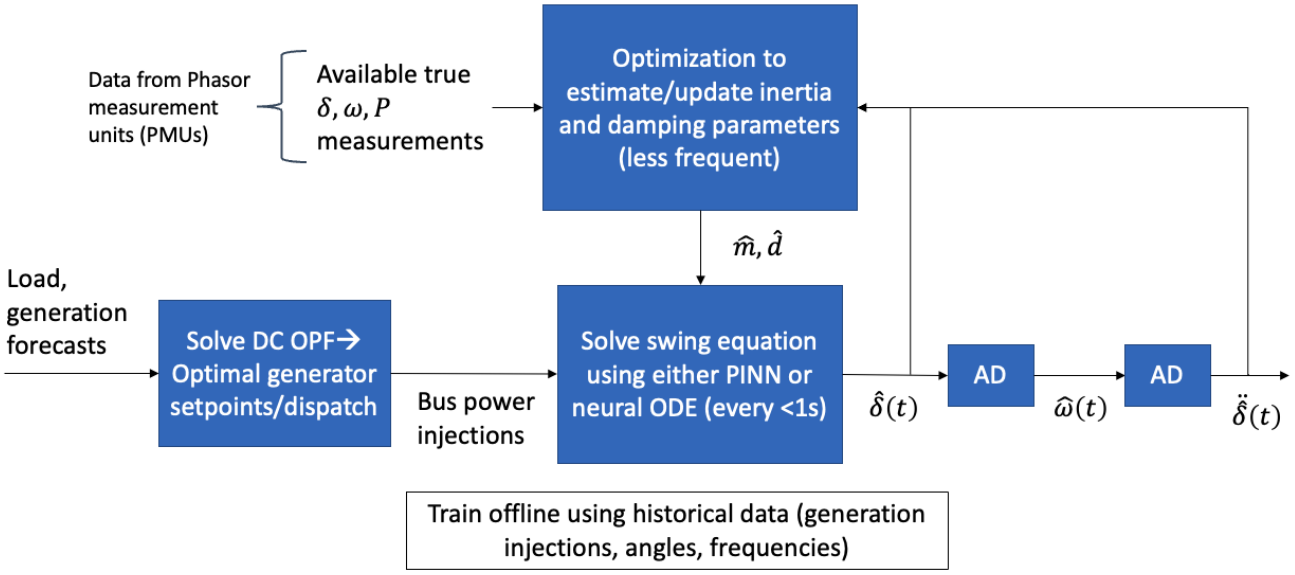


Figure 6: Overall proposed workflow.

PRIVATE CODE:
`https://gitfront.io/r/user-8651281/xLmLmor8o9Tm/18.337-S23-proj-Vineet/` (Still a work in progress - currently working on debugging the 'multi_gen' code files).

References

[1] H. S. Oh, "Analytical solution to swing equations in power grids," *PLOS ONE*, vol. 14, no. 11, p. e0225097, 11 2019. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0225097

[2] B. K. Poolla, D. Groß, and F. Dörfler, "Placement and Implementation of Grid-Forming and Grid-Following Virtual Inertia and Fast Frequency Response," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3035–3046, 7 2019.

[3] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," *IEEE Power and Energy Society General Meeting*, vol. 2020-August, 8 2020.

[4] X. Kong, K. Yamashita, B. Foggo, and N. Yu, "Dynamic Parameter Estimation with Physics-based Neural Ordinary Differential Equations," *IEEE Power and Energy Society General Meeting*, vol. 2022-July, 2022.

[5] J. Stiasny, G. S. Misyris, and S. Chatzivasileiadis, "Physics-Informed Neural Networks for Non-linear System Identification for Power System Dynamics," *2021 IEEE Madrid PowerTech, PowerTech 2021 - Conference Proceedings*, 6 2021.