Applying Physic Informed Neural Network (PINNs) to Swing Equation

Hyung-Jo Kwon (s210377) and **Bekarys Gabdrakhimov (s210127)**

Department of Electrical Engineering, Technical University of Denmark

Introduction

In this project we will be approximating the swing equation's solution using PINNs. PINNs is a scientific machine learning technique that utilizes the ordinary/partial differential (ODE/PDE) problems without using much data to train it. Loss functions are defined using these problems with boundary and/or initial value conditions [1]. The swing equation's parameters are unknown, making it an unsupervised learning problem. In addition, data are difficult to acquire. Thus, PINNs provide an effective method to train for the swing equation as it requires little data and reduces the computational time greatly [2].

Mathematical Modelling

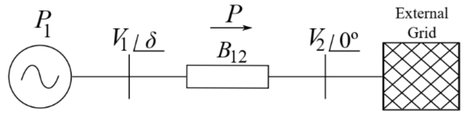


Figure 1: Single Machine Infinite Bus system [2]

We modeled our power system as a single machine infinite bus (SMIB) for simple analysis. The swing equation is defined in Equation 1:

$$m\ddot{\delta} + d\dot{\delta} + B_{12}V_1V_2\sin(\delta_1 - \delta_2) - P_1 = 0 \tag{1}$$

where: m : Inertia constant
d : Damping coefficient
B_{12} : Bus susceptance matrix of bus 1 to 2
V_1, V_2 : Voltage magnitudes of bus 1 and 2
$\dot{\delta}$: Angular velocity
δ : Rotor angle
P_1 : Mechanical power of generator 1

We define the outputs of first order non-linear differential equations using it as our true values in Equation 2:

$$F(x) = Ax + Bu + FCx \tag{2}$$

where: $x = \begin{bmatrix}\delta_0 \\ \omega_0\end{bmatrix}$: Initial state variables of δ and ω
A, B, C, F : Parameters of swing equation
u : Power disturbance in the system

We defined our parameters for our input data $g(P,t)$ as in Equation 6:

$$t \in [0, 20]s \tag{3}$$
$$P \in [0.1, 0.2]p.u \tag{4}$$
$$\delta_0 = 0 \tag{5}$$
$$\omega_0 = 0 \tag{6}$$

Loss Functions

We defined our initial value loss functions for δ, ω as shown in Equation 9:

$$M^{\delta}_{IC}(\theta) = \frac{1}{N_i}\sum_{i=1}^{N_i}(N(t_0^i,\theta) - F_{\delta}(\delta_0,\omega_0,t_0,u))^2 \tag{7}$$
$$M^{\omega}_{IC}(\theta) = \frac{1}{N_i}\sum_{i=1}^{N_i}(N'(t_0^i,\theta) - F_{\omega}(\delta_0,\omega_0,t_0,u))^2 \tag{8}$$
$$M_{IC} = M^{\delta}_{IC}(\theta) + M^{\omega}_{IC}(\theta) \tag{9}$$

where: $N(t_0^i,\theta)$: Defined neural network
θ : Neural network weights and biases

We then define M_{ODE} in Equation 12 using the swing equation for all t:

$$M^{f}_{ODE} = \frac{1}{N_i}\sum_{i=1}^{N_i}\left(0.4N''(t,\theta) + 0.15N'_{\omega=\frac{\partial\delta}{\partial t}} + 0.2\sin(N(t,\theta)) - g_p(P,t)\right)^2 \tag{10}$$

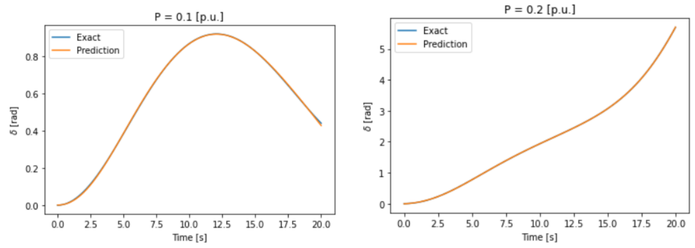
$$M^{collocation}_{ODE} = \frac{1}{N_i}\sum_{i=1}^{N_i}\left(N(t,\theta) - y_{true}\right)^2 \tag{11}$$
$$M_{ODE} = M^{f}_{ODE} + M^{collocation}_{ODE} \tag{12}$$

where: $N'_{\omega=\frac{\partial\delta}{\partial t}}$: `autograd()` of $N(t,\theta)$ wrt to t
$g_p(P,t)$: Input data $g(P,t)$, using only P

Optimization

In this section we will outline steps we performed in order to optimize our model:

- Used LBFGS instead of Adam [3].
- Decreased the learning rate (0.01)
- Increased number of collocation points (8,000 to 10,000)
- Calculated loss function for initial values of ω and derivative of model output
- Calculated loss function for exact values at collocation points
- Added more neurons for 20 seconds simulations (10 to 20 neurons)



(a) P = 0.1. (b) P = 0.2.

Figure 2: Comparison of predicted and exact values of δ.

Results

In order to create the training dataset, we used *solve_ivp* function from SciPy library to solve our ODE with initial conditions Equation 12. The time interval chosen is [0, 20s] with step of 0.1s resulting in total of 201 time values. The voltages are $V_1 = V_2 = 1p.u.$ and $B_{12} = 0.2p.u.$. The power of generator 1 is within the range of [0.1, 0.2 p.u.] and initial conditions are $\delta_0 = \omega_0 = 0$. In total we have 51 trajectories and 10,251 values. For training we use 40 points at initial conditions and 10,000 points at collocation points. We are using a neural network consisting of 5 hidden layers with 20 neurons in each layer. The Fig. 2 shows the comparison of predicted and exact values proving effectiveness of PINN's in calculating the rotor angle and increasing the computation time by 7 times.

References

[1] A. Engsig-Karup. "Project: Physics-Informed Neural Networks (PINNs)". In: *Deep Learning* (2021).

[2] S. Chatzivasileiadis J. Stiasny G. Misyris. "Physics-Informed Neural Networks for Non-linear System Identification for Power System Dynamics". In: *Technical University of Denmark* (Apr. 2021).

[3] StackExchange. *The reason of superiority of Limited-memory BFGS over ADAM solver*. URL: `https://stats.stackexchange.com/questions/315626/the-reason-of-superiority-of-limited-memory-bfgs-over-adam-solver`. (accessed: 05.12.2021).