Bayesian Neural ODEs for Microgrid Dynamics and Control

Samarth SciML Project Roadmap

July 2025

1 Introduction

This project presents a scientifically grounded approach to modeling microgrid dynamics using a system of ordinary differential equations (ODEs). The goal is to learn and recover hidden mechanisms using Scientific Machine Learning (SciML), particularly Bayesian Neural ODEs and Universal Differential Equations (UDEs), in the context of energy storage and grid stabilization systems.

2 Microgrid Control ODE Model

We consider the following system of ordinary differential equations to model a microgrid with energy storage and power flow dynamics:

Energy Storage Dynamics

$$\frac{dx_1(t)}{dt} = \eta_{\text{in}} \cdot u(t) \cdot 1_{\{u(t)>0\}} - \frac{1}{\eta_{\text{out}}} \cdot u(t) \cdot 1_{\{u(t)<0\}} - d(t) \tag{1}$$

Where:

- $x_1(t)$: Energy stored in the battery at time t
- $u(t)$: Control input for charging (+) or discharging (–)
- η_{in}: Charging efficiency
- η_{out}: Discharging efficiency
- $1_{\{\cdot\}}$: Indicator function (1 if condition is true, 0 otherwise)
- $d(t)$: Power demand from the storage

Grid Power Flow Dynamics

$$\frac{dx_2(t)}{dt} = -\alpha x_2(t) + \beta \cdot (P_{\text{gen}}(t) - P_{\text{load}}(t)) + \gamma \cdot x_1(t) \tag{2}$$

Where:

- $x_2(t)$: Net power flow through the grid at time t
- α: Damping factor in the grid
- β: Gain on power mismatch (generation - load)
- $P_{\text{gen}}(t)$: Power generated at time t
- $P_{\text{load}}(t)$: Power load (consumed) at time t
- γ: Coupling coefficient between storage and grid flow

3 Objectives

1. Replace the full ODE with a Bayesian Neural ODE and perform prediction and forecasting.
2. Replace only the nonlinear term $\beta \cdot P_{\text{gen}}(t)$ with a neural network, forming a Universal Differential Equation (UDE), and recover hidden system dynamics.
3. Extract the symbolic form of the recovered neural network to interpret the underlying reaction dynamics.

4 Immediate Action Items

1. Implement and run the given ODE in Julia.
2. Generate underlying data, including adding noise to simulate real-life conditions.
3. Implement and run Bayesian Neural ODEs to get prediction results.
4. Implement and run UDEs to get prediction results.
5. Start a GitHub repository to upload code files and figures.
6. The manuscript for this project is being written in Overleaf and can be accessed here