
Project 3Due date: Wednesday 26 June 2024, 23:59

The main objective of the project is to apply the concepts learned in class related to Neural Differential Equations and Hybrid Workflows.

1. Inverted pendulum

The objective of this task is to control the inverted pendulum problem using a Neural Network. The inverted pendulum is a classic problem in control theory, where the goal is to balance a pendulum in an upright position by applying an external force F to the cart that holds the pendulum. The system is described by the following differential equation:

$$(M + m)\ddot{x} + ml\ddot{\theta}\cos(\theta) - ml\dot{\theta}^2\sin(\theta) = F \quad (1)$$

$$l\ddot{\theta} + g\sin(\theta) + \ddot{x}\cos(\theta) = 0 \quad (2)$$

where θ is the angle of the pendulum with respect to the vertical, $g = 9.81 \text{ m/s}^2$ is the acceleration due to gravity, $l = 1.0 \text{ m}$ is the length of the pendulum, $m = 0.1 \text{ kg}$ is the mass of the pendulum, $M = 1.0 \text{ kg}$ is the mass of the cart and F is the force applied to the cart in Newtons.

Solving the coupled ODE system

First, we need to solve the coupled ODE system to simulate the dynamics of the inverted pendulum. The idea is to use an autodifferentiation library to code the solver in order to use it later to train a Neural Network to control the pendulum. In this case, I used torch due to its familiarity to me. I used Runge-Kutta 4th order method to solve the system. To test the implementations, I simulated the system with a sinusoidal force $F(t) = 10\sin(t)$. The results are shown in Figure 1.

Learning to balance the pendulum

Next, we will use a Neural Network to learn to balance the pendulum. The input to the network will be just the time t and the output will be the force F applied to the cart. We will train the network using the whole trajectory of the pendulum. The network will be trained using the Adam optimizer and a custom loss function that penalizes the deviation of the pendulum from the vertical position and large angular velocities. The network is a simple feedforward network with 1 hidden layer with 32 neurons and SiLU activation function. The output layer has a single neuron with linear activation function. The network is trained for 500 epochs with an initial learning rate of 10^{-2} and a scheduler that reduces the learning rate by a factor of 0.5 every 75 epochs. The plot of the loss function during training is shown in Figure 2. The results of the simulation using the Neural Network to control the force applied to the cart are shown in Figure 3.

Attached to the report you will find the code used to solve the ODE system and train the Neural Network, as well as the simulation results in a gif format.

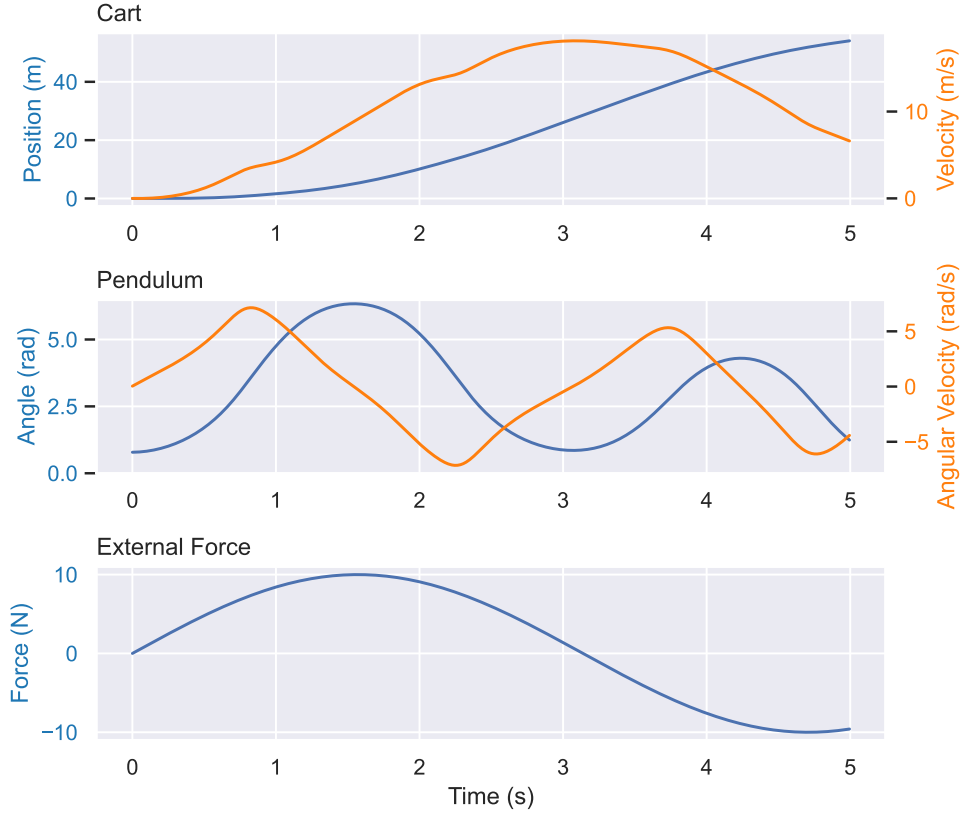


Figure 1: Simulation of the inverted pendulum with a sinusoidal force.

2. PDE-FIND

The objective of this task is to use the PDE-FIND [1] to discover the differential equation that governs the dynamics of the some unknown system using only data files.

The idea is that we can assume that the dynamics of the system can be described by a partial differential equation of the form:

$$u_t = F(u, u_x, u_{xx}, u_y, u_{yy}, u_{xy}, \dots, x, y, \dots, t) \quad (3)$$

So we can create a library of possible functions that can appear in the right hand side of the equation and use the data to find the coefficients of the functions that best describe the dynamics of the system.

We are given three files with data of differents system for which we want to find the PDE that governs the dynamics.

Implementation of PDE-FIND

We will use the library PySINDy to use the PDE-FIND algorithm.

File 1

The first file is a 1+1D dimensional system u , with x and t coordinates.

File 2

The first file is also a 1+1D dimensional system u , with x and t coordinates.

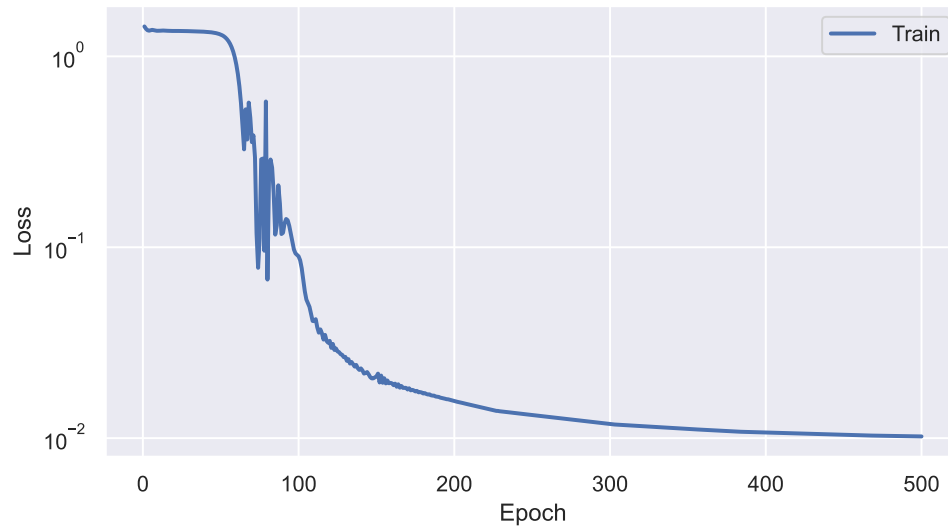


Figure 2: Loss function during training of the Neural Network.

File 3

The third file is a 2+1D dimensional system, with x , y and t coordinates.

3. Bonus Task

References

- [1] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

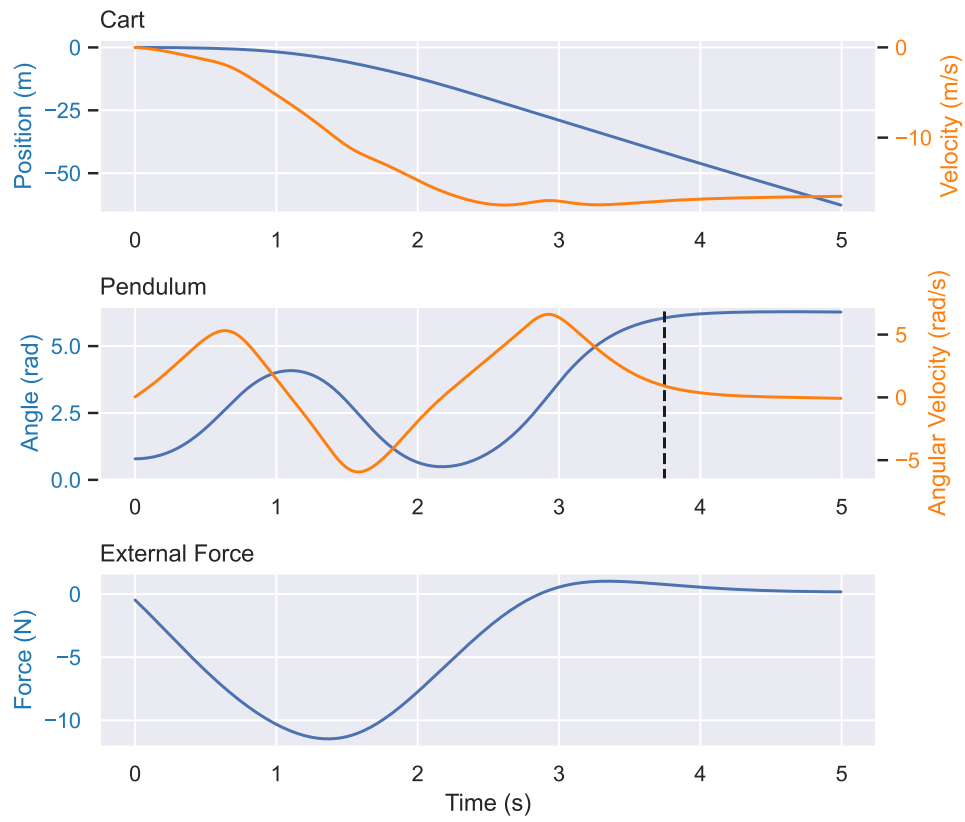


Figure 3: Simulation of the inverted pendulum using a Neural Network to control the force applied to the cart.