**Lab Session 3: Optimizing the pendulum model through Parameter Tuning**

**Objective**: The goal of this lab is for you to optimizing the model you constructed in lab 1 by tuning its parameters. You will attempt to find the optimal set of parameters that minimize the difference between the observed data and the model's predictions.

For this lab session, you will need the following files: **Lab\_3\_model\_optimization.py** and **SOME\_data.csv** (your recording). You will also need to place the lab\_3 python file in the same directory as the **Digital\_Twin.py** as the digital twin class will be imported.

We begin by pre-processing the data produced by our sensor system. The **SOME\_data.csv** file contains a recording of the pendulum swinging. Before we start optimizing the model we need to make sure the following things are already in place (from lab 2):

* *The sensor data is transformed to radians.*
  + *Scaled down in value*
  + *Include logic to transform the data correctly*
  + *Calibrate as there is some offset in the data*
* *Optionally a filter should already be in place*

Ones the data is transformed into radians, the following tasks should be performed:

* *Find the initial conditions in your recording. \*Be aware: If you recorded actions, you will also need to optimize the motor model, which is a bonus point. If you do not want to do this, use a recoding without action for the time being. You could estimate the effect of the action if you have an optimal model for the pendulum, come up with a strategy to do so.*
* *Specify a range of values for the parameters. Why do you choose this range and why the specific resolution? Resolution here says something about what discrete parameter values you are going to test.*

Once you have determined the initial condition and the ranges/resolutions for the parameters it is time to simulate and compare the outcomes to the real data by performing a simple grid search.

* *How would adding the filtering effect the optimization?*
* *How could the grid search be improved? Try to implement a better/more advanced method*
* *Can you change the code so that difference in sampling in the recoding and simulation is minimized?*
* *What are the optimal parameters? And is the error acceptable? If not, could you include additional parameters in your model? What would they be and why do you think so?*

Optimizing a model can be computationally expensive. Discuss how you could further minimize the computation you would need for the optimization. Could you reduce the data you use? Could you evaluate in a sparse way, e.g. at random points? Could you automatically estimate the upper and lower range of the parameter values?