Hanze University of Applied Sciences
Institute of Engineering
Smart Systems Laboratory

Lab Session 1 - Creating a Virtual Model of a Simple Mechanical System

1 | Aim

The aim of this lab is to develop a virtual model of a pendulum and motor system using Python. Figure 1.1 shows the schematic of the inverted pendulum apparatus, a classical problem in control engineering, designed for the development of the Introduction of Smart Systems practical sections.

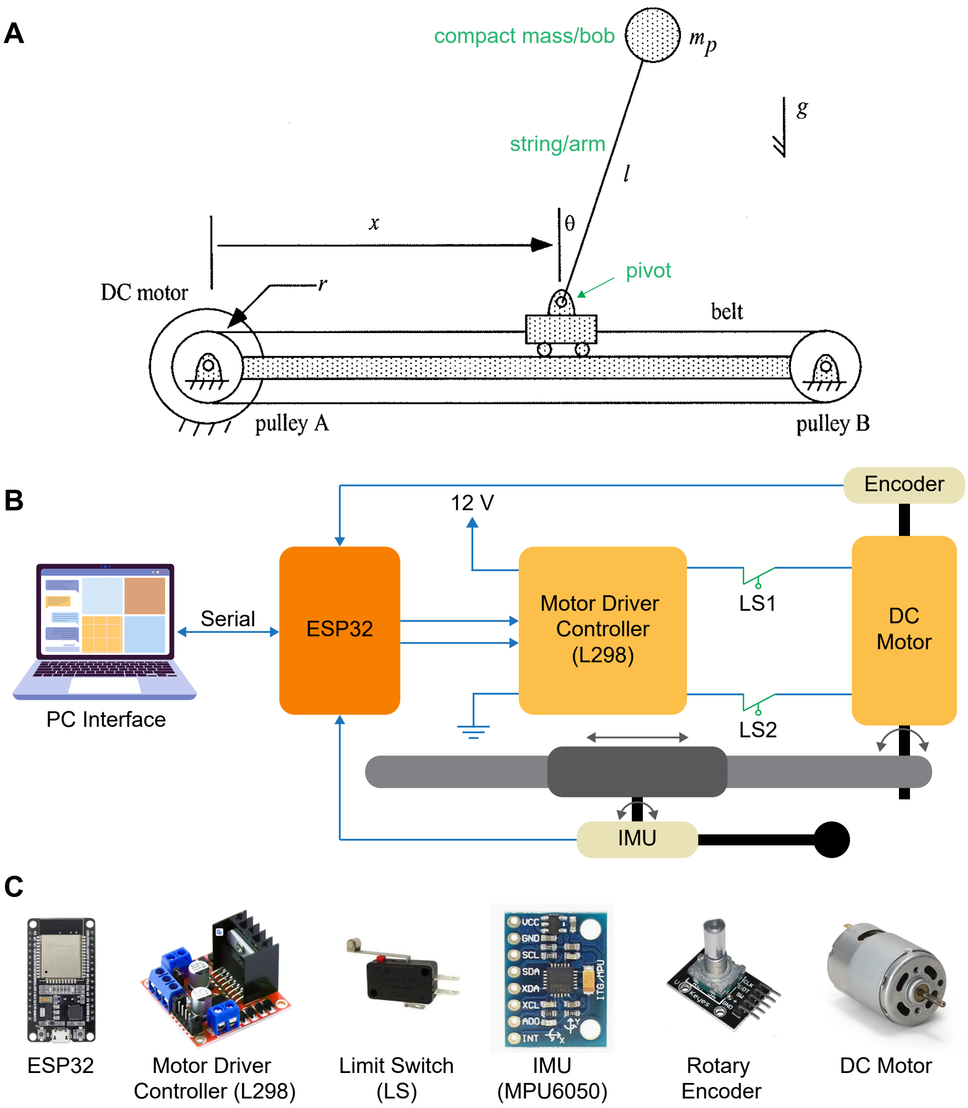

Figure 1.1: (A) Inverted pendulum mechanical schematic. (B) Electrical configuration of the pendulum apparatus. (C) Hardware components used in the system, including ESP32 microcontroller, L298 motor driver controller, limit switch (LS), MPU6050 inertial measurement unit (IMU), rotary encoder, and DC motor.

2 | Preparation

For this experiment, as material, you only need a laptop with Python installed with the necessary dependencies.

- Find them listed in the `requirements.txt` file.
- To install, use `pip install -r requirements.txt`.

Before getting started, make sure to complete the following preparatory steps:

- Watch the following video: Differential equations guide: http://tinyurl.com/525buknk;
- Read the paper [1]: this paper presents valuable mathematical representation of the motor modeling;
- Read the paper[2]: this paper models the inverted pendulum;
- Run the Python examples using a pendulum: to familiarize yourself with PyGame library.

The following Python files are necessary for today (place them in the same folder):

- `Digital_twin.py`: important file, used throughout all the labs; and
- `Lab_1_simulation.py`.

3 | Procedure

3.1. Generate a flowchart of the code `Lab_1_simulation.py`. Use the debug function.

3.2. The goal is to create a model that represents the angular acceleration ($\frac{d^2\theta}{dt^2} = \ddot{\theta}$) of the pendulum. For the first part, you should implement the following function:

Algorithm 1: Function to calculate angular acceleration.

```python
def get_theta_double_dot(self, theta, theta_dot):
    """
    Lab 1: Model the angular acceleration (theta_double_dot)
    as a function of theta, theta_dot and the self.currentmotor_acceleration.
    You should include the following constants as well: c_air, c_c, a_m, l, and g.
    """
    # Implement your model here.
```

This function should return the pendulum angular acceleration, also denoted here as `theta_double_dot`.

Keep in mind that the sampling period `delta_t` of the virtual model should match the sample rate of the sensor (Lab 2). Collect it via Arduino IDE or VSCode (Serial Monitor extension).

There is already a function implemented for the motor response model in

```python
def update_motor_accelerations(self, direction, duration)
```

- *Bonus*: Improve the **motor model** to better align with the actual motor response.

3.3. Once you have implemented the function `get_theta_double_dot(self, theta, theta_dot)`, you can run the `Lab_1_simulation.py` file.

3.4. Try to swing up the pendulum using the keys on your keyboard and create a sequence of 10 actions that successfully swings up the pendulum.

4 | Tasks

Provide a report describing the following aspects of the models (the codes need to be included).

4.1. Pendulum Model:

- Include air friction, mechanical friction, pendulum length, and gravitational effects in the model;
- State space parameters should encompass the angle and angular velocity of the pendulum. Try to set the constants such that the behavior is similar to the real pendulum.

4.2. Motor Model:

- Explain how the motor acceleration relates to the pendulum model.
  - Based on the paper you have read [1], represent the motor in terms of its acceleration effect on the pendulum;
- Ensure the model can simulate the dynamic interaction between the motor and pendulum.

5 | References

[1] V. H. Pinto, J. Gonçalves, and P. Costa, "Modeling and Control of a DC Motor Coupled to a Non-Rigid Joint," *Applied System Innovation*, vol. 3, no. 2, p. 24, 2020. [Online]. Available: https://www.mdpi.com/2571-5577/3/2/24

[2] P. Strakov and J. Tuma, "Mathematical modelling and controller design of inverted pendulum," in *2017 18th International Carpathian Control Conference (ICCC)*. IEEE, 2017, pp. 388–393.