**Lab Session 2: Incorporating Data Connectivity into the Digital Twin**

**Objective**: Enhance the virtual model with real-time data from sensors.

For this lab session, you will need the following files: **test\_data.csv**, **plot\_data.py**, **Make\_recording.py**, **Run\_recording.py**, and **Digital\_twin.py**. You should already possess the last file and have a functional model implemented.

We begin by examining the data produced by our sensor system. The **test\_data.csv** file contains a recording of the pendulum swinging. Execute the **plot\_data.py** file to analyse the data:

* *What type of noise is present?*
* *Which filtering technique would be most appropriate?*
* *What additional transformations are required to incorporate this data into your virtual model?*

Reflect on our system dynamics lecture, where we used the phase portrait to analyse significant aspects of the dynamic system. To enhance our understanding from the data we collect, consider the following questions:

* *What type of data is critical to collect?*
* *How can the collected data provide information on the energy dissipation within the system?*

Once you've determined the data you wish to collect, proceed with running the **Make\_recording.py** file. You can modify the actions in the **actions** array, with available actions ranging from 0 to 8, as defined in your digital twin file (**action\_map**). Note: If for some reason you don’t get any data via the connected device, you might want to try to add the following in the digital twin file: ser.flush() after reading and writing the serial port. If it does not work, please continue the lab with the provided data or ask a classmate to share collected data.

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

When initiating a new recording, ensure to rename the file in the **digital\_twin.start\_recording("test")** function. In this instance, **test.csv** will be created and saved in the directory containing the **Make\_recording.py** file.

Next, we will filter the data from the recording by executing the **Run\_recording.py** file. This script implements three different filtering methods: the Kalman filter, a Median filter, and an Exponential Moving Average (EMA) filter.

* *Adjust the different filters to achieve optimal signal filtration.*
* *Discuss the advantages and disadvantages of each filtering technique.*
* *Identify your preferred filtering method and explain your choice.*
* *Calibrate and scale the signal so that it can be accurately represented in the virtual model.*
* *Uncomment the section that enables the digital twin's visualization and display your chosen filter with the optimal parameters.*
* *Finally, integrate this filter and transformation into the* ***Digital\_twin.py*** *file, applying a transformation to the* ***x\_pivot*** *as well.*