

**Fontys University of Applied Sciences**

Mechatronics Engineering

Name: Luc Melis, Byron Greene

Student number: 3697452, 3879518

Date: June 18, 2023

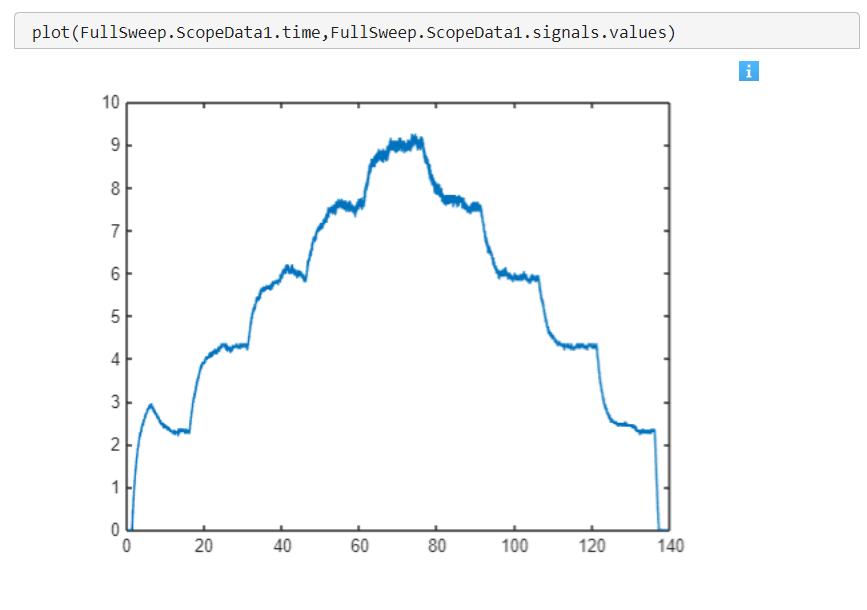
Version: 1

Class: MC2A

Practical Group: B

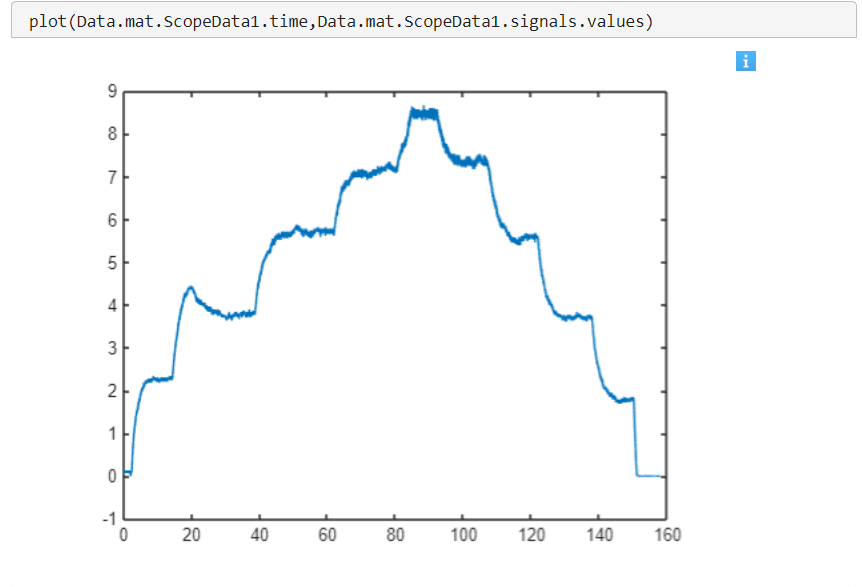
# Lab 1 Assignment

## Graph 1



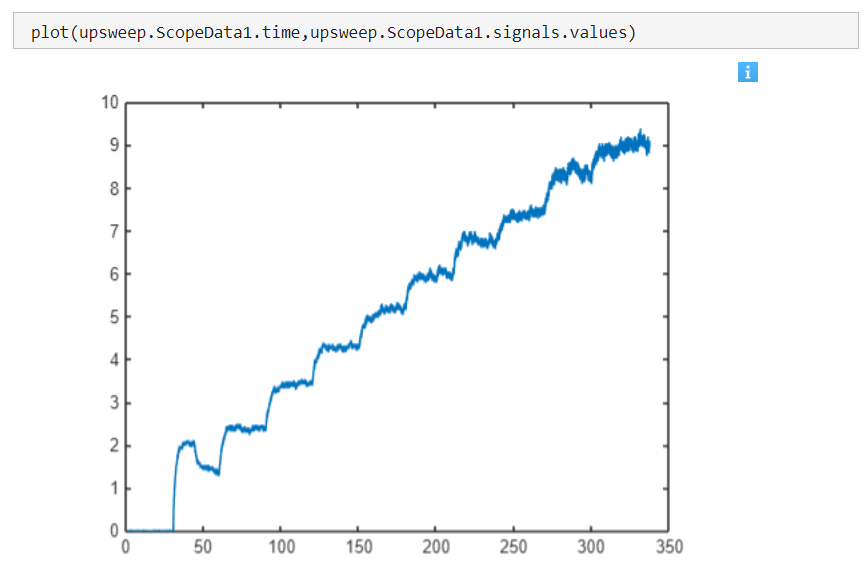
**Graph 1. Full Sweep Done Externally using MyDaq**

## Graph 2



**Graph 2. Full Sweep Done Internally using Control Box.**

## Graph 3 (0-10 mat)



**Graph 3. Full Up Sweep**

## Graph 4

A picture containing text, screenshot, line, plot

Description automatically generated

**Graph 4. Displaying a Full Rise from 0V to 10V.**

## Results from Lab 1.

Make sure to have correctly worked out your measurements *and* obtained the process parameters below:

* The process amplification **Kp**= 1.05
* The time constant **τ** = 2.804 seconds
* The offset on the process line (see the steady state input output graph) 0.115V
* The maximal and minimal values of the output (process limits): **MIN** = 1.206V, **MAX**=8.79V
* The linear work area of the process (not needed for Simulink interpretation): from …… V to ……V

For each of the 5 upwards and 5 downwards step responses you might get a different amplifications and time constants (thus in total 10 different amplifications and time constants). In fact, you only need 1 good representative amplification and time constant. A good method would be averaging up those amplifications and time constants but first make sure that the numbers that lie far away are not taken into the averaging. Because you find one transfer function that is valid for the response inside the operation region.

**Results for Time constant & Gain**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Time Constants** | **Gain** |
| 0 | 0 |  |  |  |
| 1 | 1.4355 |  |  | 1.4355 |
| 2 | 2.414 | 1.4484 | 2.26 | 1.207 |
| 3 | 3.535 |  |  | 1.178333 |
| 4 | 4.262 | 3.5228 | 3.29 | 1.0655 |
| 5 | 5.15 |  |  | 1.03 |
| 6 | 5.92 | 5.2568 | 2.89 | 0.986667 |
| 7 | 6.53 |  |  | 0.932857 |
| 8 | 7.425 | 6.823 | 3.29 | 0.928125 |
| 9 | 8.129 |  |  | 0.903222 |
| 10 | 8.79 | 8.244 | 2.29 | 0.879 |
|  |  |  |  |  |
|  |  |  | 2.804 | 1.05462 |

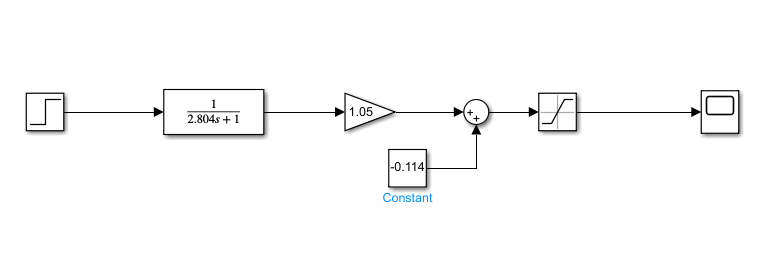
**Avg Time constant (Simulink) =** 2.804 seconds

**Avg Time constant (Real time) =** 16.36 seconds

**Avg Gain =** 1.05462

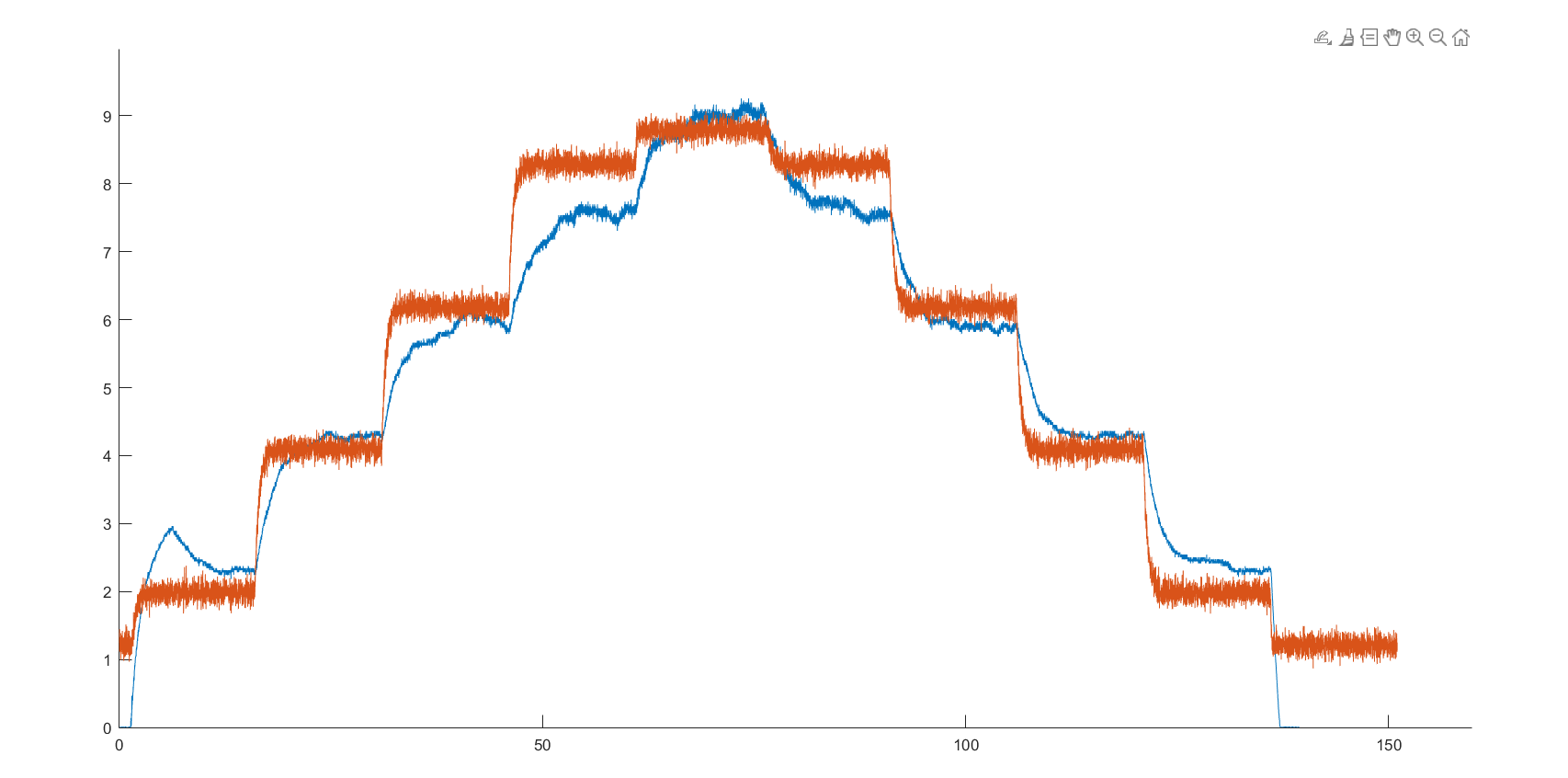
## Graph Linear

## Simulink Block Diagram

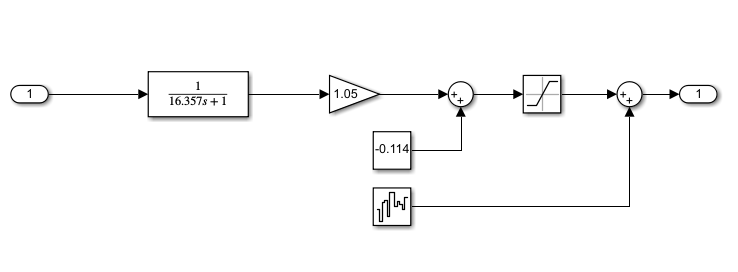


# LAB 2 Assignment

## Put experimental graphs on top of the simulated graphs to show the (mis)match. This might require some time to send all the data to MATLAB Workspace and manipulating the data in such a way that the curves are next to each other visually.



## To make your model represent the reality better, add some noise at the output and use saturation blocks where necessary because the actuation and measurement has limits.



## Make a simulation of the P-controlled process. Comment on how realistic the simulation is with respect to the reality. Make sure your simulation graphs contain the reference, process output and the regulator output as shown below.

A diagram of a diagram

Description automatically generated with low confidence

A screen shot of a graph

Description automatically generated with low confidence

The P controller has a positive effect on the plant as without it, it would only be able to reach 2,5V output. However, the effect of P is small here as it is not able to reach the wanted 5V as adding more Gain would result in a lot of overshoots while never be able to achieve the wanted settling value. This means the P controller is not enough to control this model, and we need to implement more control elements to this system. On the simulated model, we notice that it starts at 1.2V when it is off, this is due to the minimum possible output value, and is a limiting factor the model.

# LAB 3 Assignment

## PID- Controller Simulink Setup

A picture containing diagram, technical drawing, plan, line

Description automatically generated

The top part of the Simulink model is the PID controller, The bottom part of the model is adding the wanted value, the upper and lower boundaries of 1% of that value, and implementation of a 10% overshoot line. This was done to be able to quickly analyse if the PID controller reached above 10% overshot, or if it settled at the right location.

## Simulated results of Constructed PID controller.

Build and experiment with PI, PD, and PID controllers. What are the roles of P, I and D controllers in this system? What are the best values you tuned for and ? How do the experiment results compare to simulated results?

1. **Roles of the P, I, D controllers.**

**P-controller**

The P-controller function in this system helped in reducing the steady- state error by adjusting the magnitude of the error from the input signal.

**I-controller**

The I-controller was responsible for eliminating the steady state error that could not be corrected by the P-controller alone. Once the error was detected the I- controller slowed the rate at which the voltage value approached the reference value. This trade of slowed the system down but resulted in non-oscillating reference value with no initial overshoot.

**D-controller**

The D- controller in this system focused on the rate of change of the error. It calculated the derivative of the error with respect to time and then generate a signal based on the rate of change. However, this was causing a significant overshoot and the D- controller was not turned off but instead limited to a very minuscule value of either zero (0) or **1x.**

1. **Best tune Values for and .**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tune Combination** |  |  |  |
| **1** | **2** | **0.25** | **0** |
| **2** | **2.5** | **0.5** | **1x** |
| **3** | **2.5** | **2** | **1x** |
| **4** | **2.5** | **1.8** | **1x** |

**Combination 1 Tune Result**

-Reaches close to the selected 5V input. However, falls slight short below the 1% error.

**Combination 2 Tune Result**

-Slowly Reaches the 5V selected reference with no overshoot.

**Combination 3 Tune Result**

-Reaches the reference value with a slight overshoot below 10%

-Oscillates for a period of 1 Simulink second.

-However, Reaches the reference faster than previous tunes.

**Combination 4 Tune Result**

-Reaches the reference value the fastest.

-No overshooting above 10%.

-Consistent results for multiple reference Voltage values.

**C) How do the experiment results compare to simulated results?**

The experimental results are not working perfectly as predicted to the real simulated values, but this means we were already very close to the values we wanted. This can be due to a lot of factors in the real world, as there is lag between the starting of the motor and the rising of the pressure in the water tower etc… This means the values of the gains of the PID controller only needed to be changed slightly to work for the experimental values.

## Experimental performance summary.

Write down the experimental performance summary of the closed loop system. In other words, what are the achieved settling-time, rise-time, overshoot, steady state error, peak time?

We did multiple different experiments to see if the experimental model works at for different inputs:

1. Reference at 6V

A picture containing text, diagram, plot, screenshot

Description automatically generated

1. Reference at 7V

A picture containing text, screenshot, line, plot

Description automatically generated

1. Reference at 8V

A picture containing text, screenshot, plot, line

Description automatically generated

We notice that the Rising time is very a lot faster than the model without the PID controller. This is also reflected in the steady state error, as it almost always less than half of 1% of the reference value, meaning we can consider there is no steady state error. We notice that the higher the reference, the higher the overshoot. This is due that there is more error at that point and the more error, the more the integrator action of the PID controller acts on the system. This overshoot is specifically made to never exceed the 10% overshoot margin we set.