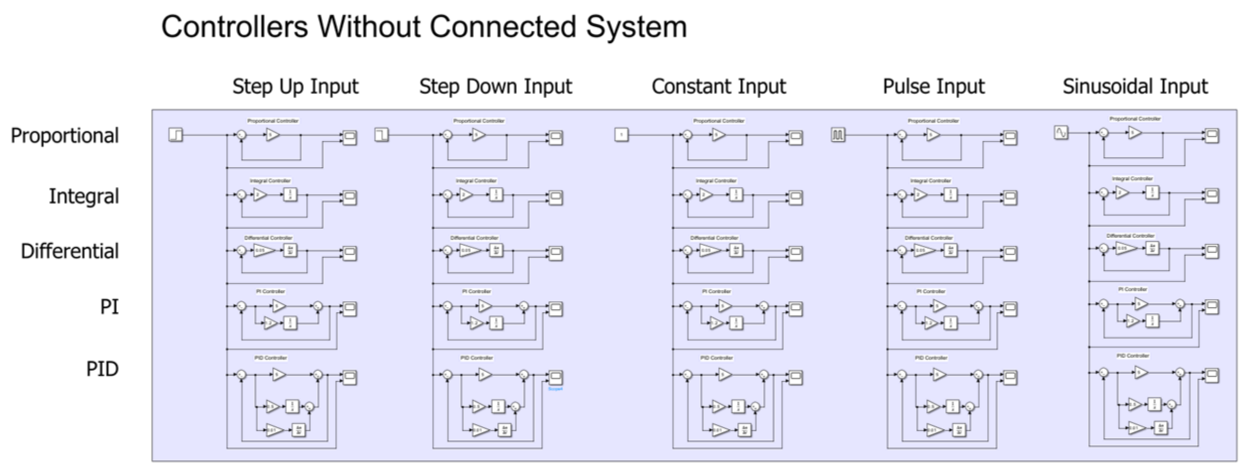
Group Project Part1: Varying Controller Types

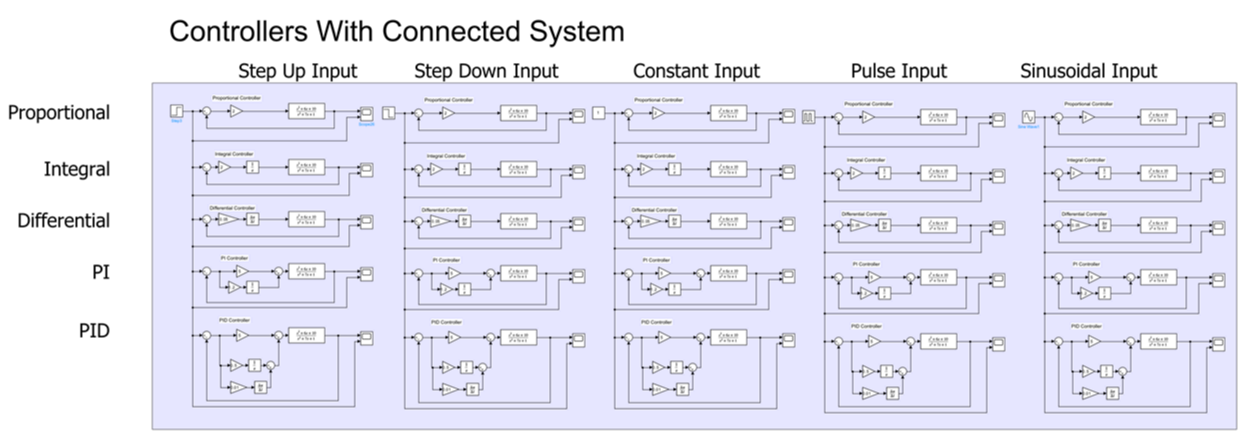
Group 6: Christian Chu Yang, Nafis Isfar

In our Simulink document, we sectioned off each controller into separate sections:

The first section was the controllers on their own,

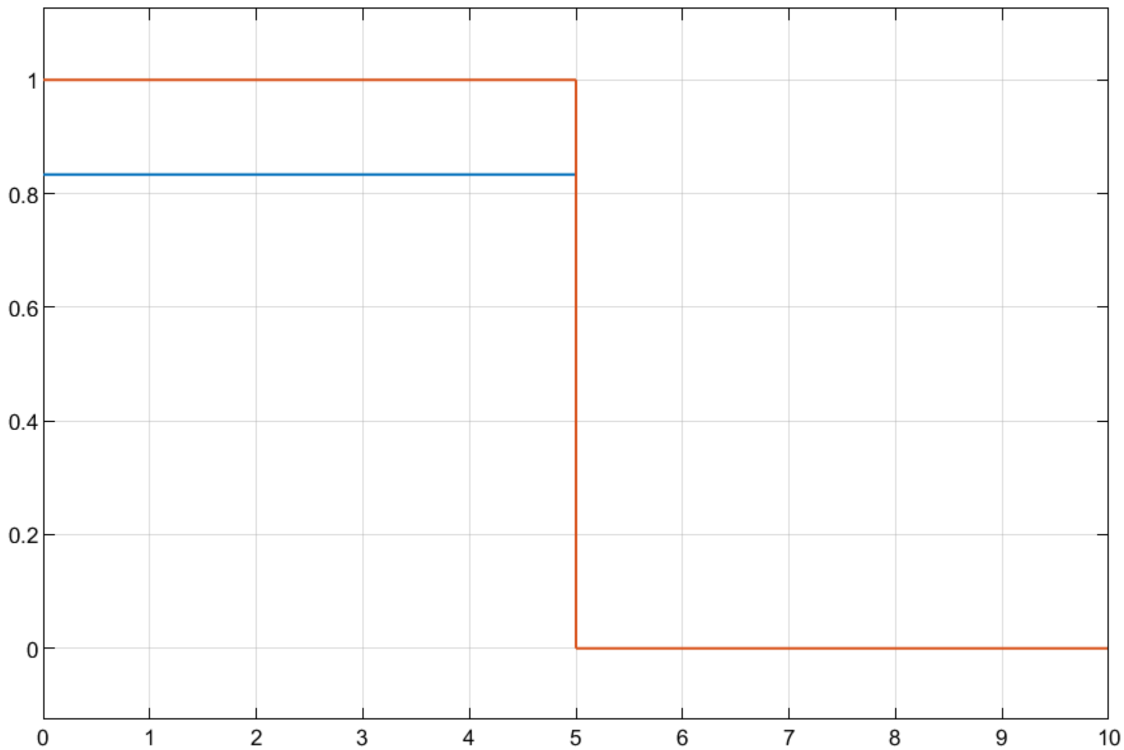


and the second section was the controllers connected to a hypothetical system/process.



Each one of these sections are separated by a grid, following the pattern of different inputs across the horizontal axis, and different controllers by the vertical axis. This way, the simulation needs to be run only once to view all of the differing controller behaviours.

Below is the behaviour for the Proportional Controller:

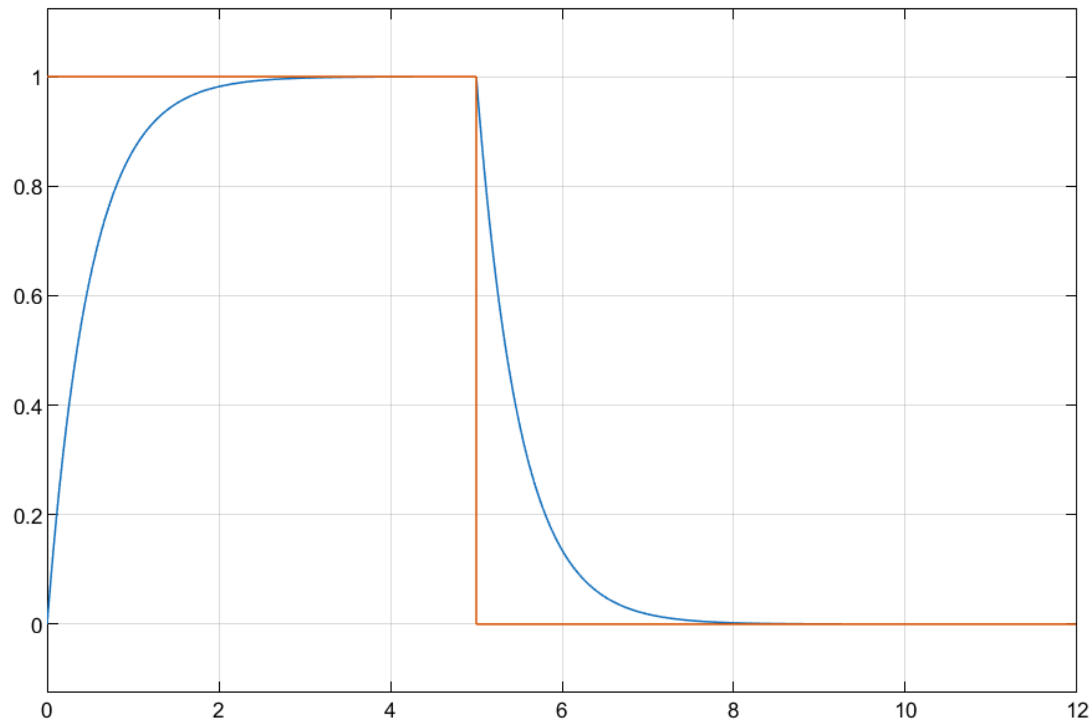


The proportional controller solely scales the error, as the input signal represented by the orange line experiences a step decrease to 0 at t=5, whereas the controller output (blue line) stabilizes at approximately 0.825. The effects of this can be intuitively seen from the transfer function of the proportional controller with no system to control:

$$\frac{Y}{X} = \frac{K_p}{1 + K_p}$$

This proportional controller results in a steady state error as although it responds quickly, the process will not always arrive at its desired position, set by the reference value.

Below is the behaviour for the Integral Controller:

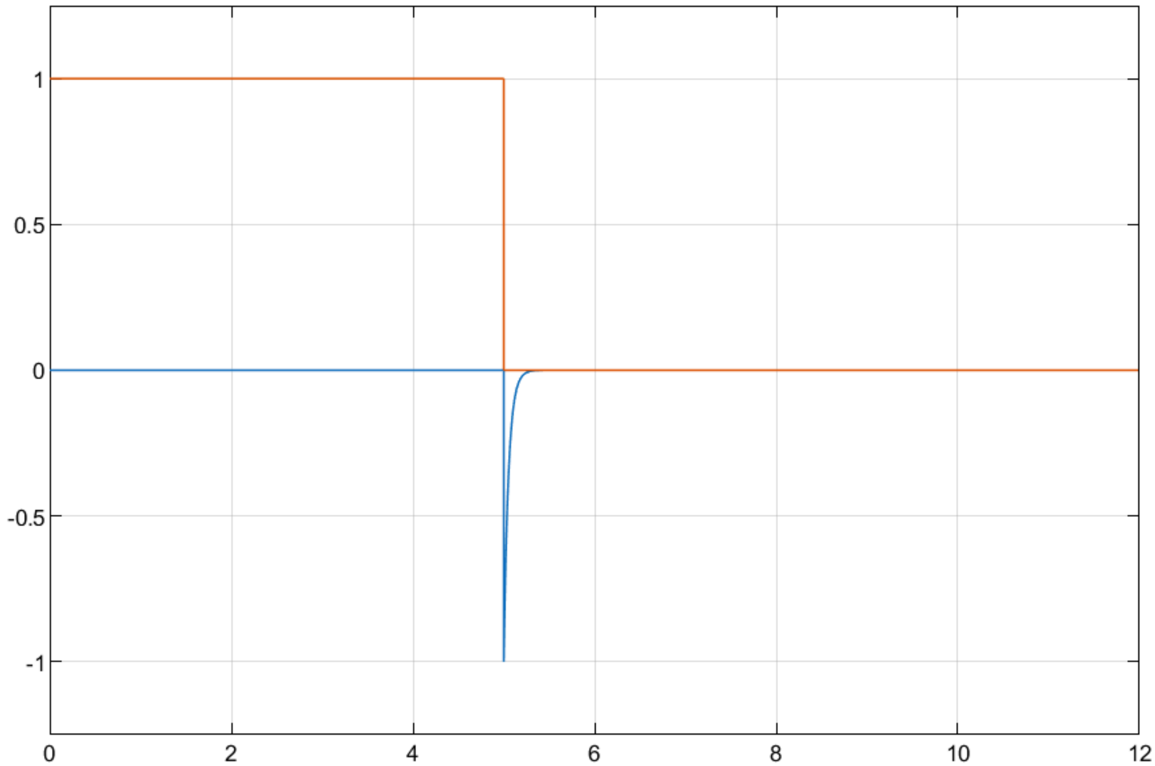


This is the integral controller, of which the controller output represents the area of the error. The brown line represents a step input which jumps from 0 to 1 at time t=0, and then drops back to 0 at around t=5. The blue curve represents the controller's response under integral control.

$$\frac{Y}{X} = \frac{K_i}{s + K_i}$$

The integral controller results in a smooth response with much less steady-state error. However, the result shows a larger transient and thus the integral controller has a longer response time.

Below is the behaviour for the Differential Controller:

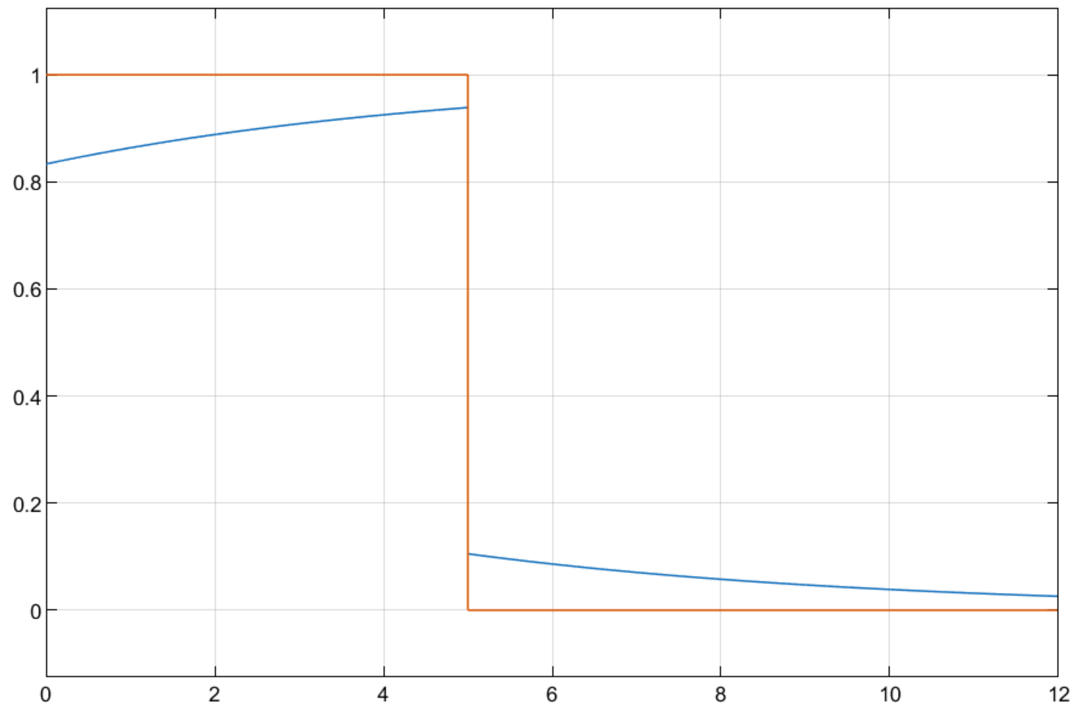


This plot above for the Differential Controller displays the step input as the brown line, which jumps from 1 to 0 at time t=5. The blue curve represents the controller response as the controller reacts to the changes in the input.

$$\frac{Y}{X} = \frac{s \cdot K_i}{1 + s \cdot K_i}$$

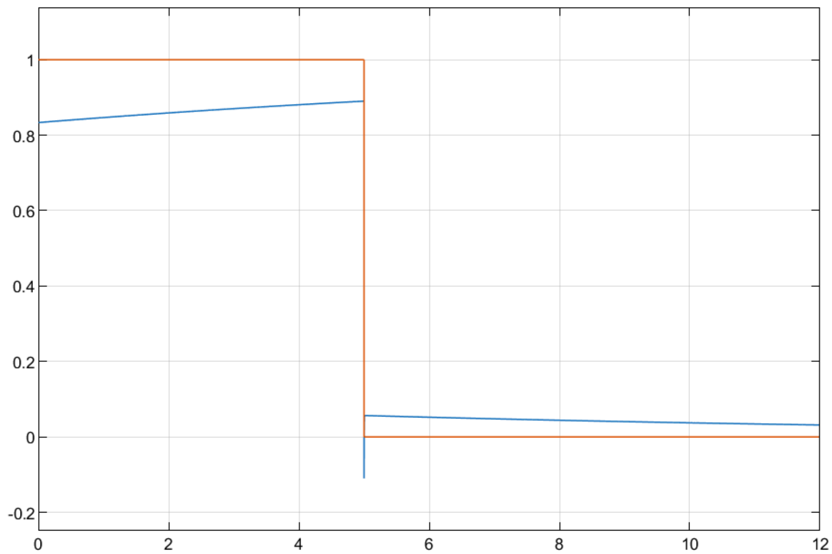
When the step input drops from 1 to 0, the response shows a sharp negative spike, which shows that the differential controller responds solely to changes in the error value, which is useful in applications where the system's output or input changes rapidly.

Below is the behaviour for the PI Controller:



The plot above for the PI controller displays the step input as the orange line, where a sharp change at t=5 indicates a transition from 1 to 0. The controller response of the PI controller is shown as the blue line, which follows the input initially with a gradual rise due to the integral. After the step change, the output decreases smoothly. This showcases the ability of the PI controller to not only return to setpoint/reference value, but also to quickly act in event of a disturbance or desired change, that is, quicker than solely the I controller.

Below is the behaviour for the PID Controller:



The plot above for the PID controller displays the step input as the orange line, which starts at value 1 and drops to 0 at t=5. The controller response (blue line) starts at a value of around 0.8 and increases slightly. After the sudden step change at t=5, the controller output briefly dips below zero and returns to the setpoint. This controller has all the elements of the P, I, and D controllers combined into one, such as a faster response time, reaching the reference value, and improved operation for faster changing situations. However, careful tuning of all the parameters is required for desired operation.