Lab 5: Summary

Prerequisites

1. Work in the same group from Lab 4.
2. Lab 4: Task 1A (read temperature from the Arduino), Task 1B (send a duty cycle input to the Arduino).
3. Lab 4: Task 1C - Use the same board and its estimated plant transfer function
4. Lab 4: Task 2A/2B – Use the same reference signal, PI controller structure and controller parameters for Lab 5, Task 1.

Tasks for Lab 5

First, complete the Tasks for Lab 4 if you have not already done so.

1A: Implement a discretized PI controller with anti-windup in simulation (Similar to Lab 3).

1B: Using the controller from 1A, complete the hardware experiment (Similar to Lab 4).

2A-2B: Design controller to track a reference signal containing a ramp. Simulate the closed-loop system. (Similar to: Tutorial 3, Lab 2, and Lab 4)

2C: Using the controller from 2A-2B, complete the hardware experiment (Similar to Lab 4).

Outcomes

Building on your knowledge from Lab 4, you will: plan, design, simulate and implement two more controller types on hardware.

Report Requirements:

1. Labs 4 and 5 will be combined into a single report worth 15% of your ACS grade.

2. Hardware will not be available outside of scheduled ACS lab hours. Failure to complete the tasks will result in a substantial number of report marks being inaccessible.

3. Maximum of three students per group. All students must attend the same lab session. Groups must be fixed between Lab 4 and Lab 5. Due to the work required to complete the lab, students who do not attend both labs with their group will not receive a mark for the report.

Lab 5 will be tight on time. To support your group members, you must be punctual.

Lab 5: Pre-Lab

Prior to Lab 5, there are a number of tasks that you can work on:

- Task 1:
  - Review your work from Lab 3: Discretized PID controllers. How would you implement the discretized PI controller for Lab 5? Using the estimated plant model from Lab 4, you can complete the simulation.
- Task 2:
  - You can create the ramp reference signal by adding two ramp signals together in Simulink.
  - The controller can also be designed using the estimated model from Lab 4.
  - Complete the simulation.

This pre-lab will not be marked, but it will help you complete Lab 5 on time.

Task 1

Discretized PI controller with anti-windup in simulation and, on hardware.

Task 1A

1. New Simulink file.
2. Add the plant transfer function estimated in Lab 4, Task 1C.
3. Choose the same reference signal in Lab 4, Task 2A/2B.
4. Choose the same PI controller structure in Lab 4, Task 2A/2B.
5. Create a discretized PI controller (Similar to Lab 3). Use the same controller parameters from Lab 4, Task 2A/2B.
6. Set the anti-windup limits for the control signal $U_{act}(t_i) \in [0,1]$. There are no limits for the derivative of U_{act}.
7. Implement the closed-loop system in a simulation.
8. Plot reference, control and output signals.
9. Save your MATLAB files.

Task 1B

1. New Simulink file.
2. Implement the discretized PI controller on hardware.
3. Plot your reference, control, and output signals.
4. Save your plots and MATLAB files.

Turn off the power at the wall – let the board cool back to room temperature.

Task 2

For some applications it is also important that the ramp up/down of the output to the steady-state is done in a controlled, predictable manner. For example, to correctly make some plastics, the ramp up in temperature must follow a specific temperature profile to produce the correct chemical reaction.

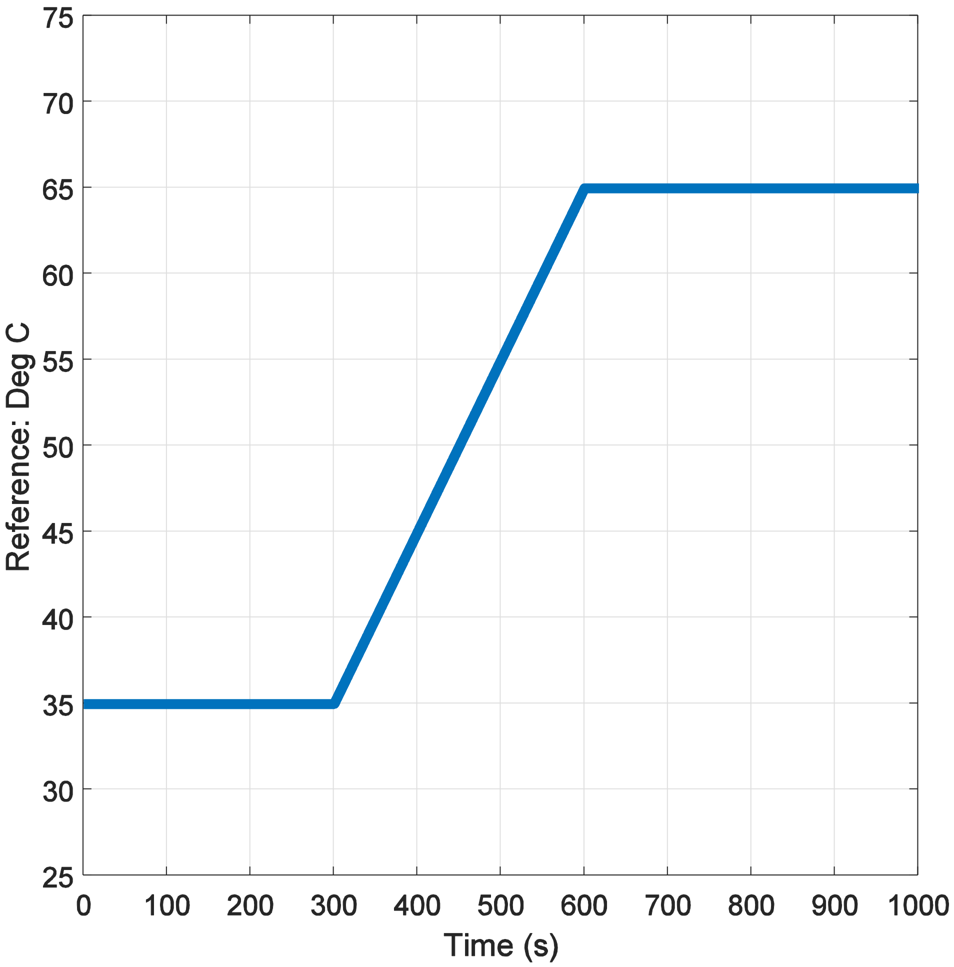
A controller that is able to track a reference signal containing a ramp, has the structure:

$$C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s^2}$$

Implement in simulation and hardware.

Task 2A

1. New Simulink file.
2. By adding two ramp signals together, create a reference signal as illustrated:



Confirm that the values of your reference signal as well as the time when the signal changes are the same as the diagram.

Task 2B

1. From Lab 4, Task 1C: $G(s) = \frac{a}{\tau s+1}$
2. Design the controller: $C(s) = \frac{c_2 s^2 + c_1 s + c_0}{s^2}$ with three poles at -λ.
3. Add the reference signal designed in Task 2A.
4. Add the plant transfer function found in Lab 4: Task 1C.
5. Simulate the closed-loop system
6. Tune λ to meet the performance specifications:
    a. Overshoot <= 20% at all times.
    b. Output is settled prior to the start of the ramp at t = 300 seconds.
    c. Subjectively, high performance tracking of the ramp.
    d. Output has settled at the final temperature prior to t = 900 seconds.
7. Confirm your solution with a lab demonstrator before proceeding.

Task 2C

1. New Simulink file.
2. Implement the designed controller on hardware.
3. Confirm your solution with a lab demonstrator before proceeding.
4. Plot your reference, control, and output signals.
5. Save your plots.
6. **Save your MATLAB files, Simulink files, and plot pictures on the network drive.**
7. Once you have completed the lab, in the Arduino IDE, select the Arduino Mega 2560 board and upload the blank sketch.