5ELEN018W - Tutorial 7 Exercises

1 Car Cruise Control

Car cruise control can be found in most of the modern cars. The cruise control system keeps the car at a constant speed despite any external disturbances, such as the road surface and incline.

The car's mass m is controlled by a force u. To simplify the situation, we assume that the force u that our controller applies is not affected by other parameters such as tires, etc.

The equation of the system is described by the following:

$$m\dot{v} + bv = u \tag{1}$$

where v is the speed of the car, u is the control action and b is the damping coefficient due to friction.

For the purposes of this exercise, the following values should be used for the system: $m = 1000, b = 50, u = 500$.

1.1 Modelling of Cruise Control

Implement a Simulink modelling of the above system. For the purposes of the control force u, use a *Step* block (`Simulink->Sources`) that has a value equal to 500 from time $t = 0$.

Use a simulation time of 120 secs (set *Stop time* to 120 in the main Simulink tab).

1.2 Control of the System

1. Add a PID controller to the system, following the guidelines we have covered in the last lecture.

   The same reference signal as in the previous section should be used (step with value equal to 500) as the desired response of the system.

   Use a PI controller with some arbitrary K_p and K_i values.

   Is the response of the cruise control system satisfactory when considering the desired speed response?

2. Open the PID block by double clicking on it and click on the *Tune* button. Choose different values for *Response Time* and *Transient Behavior* so as to try to match the desired response for the speed of the system. Click on the *Update Block* button to save your chosen values to the PID controller and open a Scope connected to the speed signal of your implemented system to check the response.

3. Set the values of the PID controller to $K_p = 800, K_i = 40$. Check the desired speed response of the system by running it and opening the Scope connected to the speed signal. Is that a better response from what you have achieved before? Can you do better than that, by re-opening the PID controller tuner and modify its parameters using the sliders?

2 Two Tank System Modelling and Control

A robot tries to control the level of the liquid contained in 2 tanks which are connected to each other. The system is shown in Figure 1. The robot controls the inflow Q_{in} for tank 1 by turning a valve. It is assumed that the time to turn the valve is negligible therefore the robot controls the inflow Q_{in} instantaneously.

Tank 1 receives an inflow $Q_{in}(t)$. Tank 2 receives an outflow from Tank 1 and drains liquid from its own outlet. Both tanks have outlets through which liquid exits at a rate proportional to the height of liquid in each tank. The objective for the robot is to control the liquid (e.g. water) levels in both tanks using a PID controller, by adjusting $Q_{in}(t)$ at every time step.

The dynamic system is described by the following coupled system of differential equations derived from the principle of mass conservation (rate of change of volume = inflow - outflow):

$$\begin{aligned} A_1 \frac{\mathrm{d}h_1(t)}{\mathrm{d}t} &= Q_{in}(t) - Q_{out,1}(t) \\ A_2 \frac{\mathrm{d}h_2(t)}{\mathrm{d}t} &= Q_{out,1}(t) - Q_{out,2}(t) \end{aligned} \tag{2}$$

where the outflow from Tank 1 $Q_{out,1}(t)$ is proportional to its height h_1:

$$Q_{out,1}(t) = k_1\sqrt{h_1(t)} \tag{3}$$

and the outflow from Tank 2 $Q_{out,2}(t)$ is proportional to its height h_2:

$$Q_{out,2}(t) = k_2\sqrt{h_2(t)} \tag{4}$$

In the above equations, the following variables are used:

- $h_1(t)$ is the liquid level in Tank 1 (m units)
- $h_2(t)$ is the liquid level in Tank 2 (m)
- Q_{in} is the inflow to Tank 1 (m^3/s), i.e. the control variable
- $Q_{out,1}(t)$ is the outflow from Tank 1 to Tank 2 (m^3/s)
- $Q_{out,2}(t)$ is the outflow from Tank 2 (m^3/s)
- A_1 is the cross-sectional area of Tank 1 (m^2)
- A_2 is the cross-sectional area of Tank 2 (m^2)
- k_1 is the valve constant for outflow from Tank 1 (m^2/s)

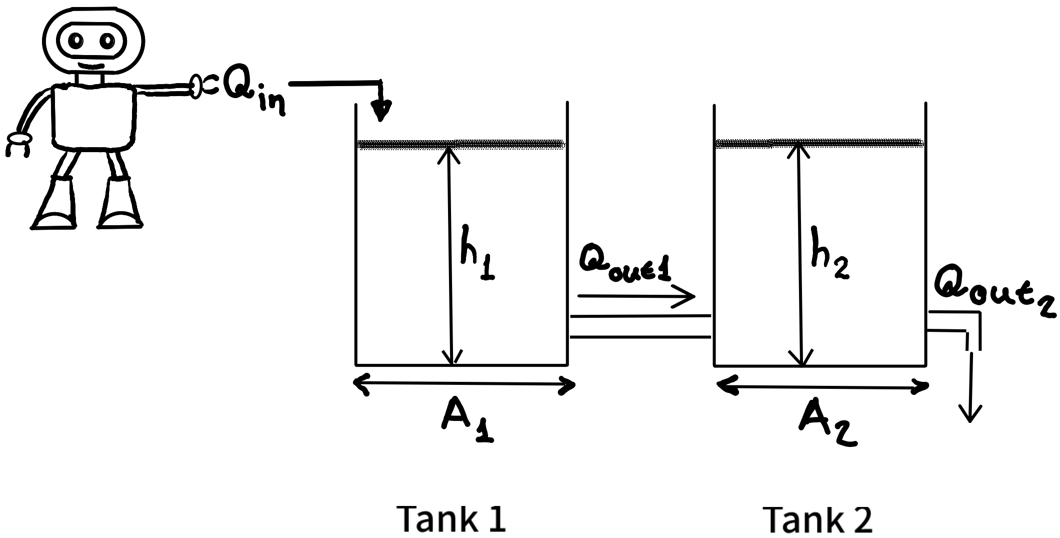

Figure 1: The 2 tank system controlled by the robot.

- k_2 is the valve constant for outflow from Tank 2 (m^2/s)

For the purposes of all simulations in this problem, the following constant values should be used:

- $A_1 = 2\ m^2$ (Tank 1 cross-sectional area)
- $A_2 = 1.5\ m^2$ (Tank 2 cross-sectional area)
- $k_1 = 0.4\ m^2/s$ (Tank 1 outflow constant)
- $k_2 = 0.3\ m^2/s$ (Tank 2 outflow constant)

2.1 Simulink Model

Implement a Simulink block diagram model of the 2-tank system without controlling the level of liquid in the two tanks. The dynamic system is described in equations (2).

2.2 Control of the Simulink Model

Extend the Simulink model you developed in Section 2.1 with a PID controller which the robot applies.

The goal of the robot equipped with a PID controller is to maintain the liquid levels in both tanks at desired levels $h_1^{desired}$ and $h_2^{desired}$:

- $h_1^{desired} = 1.5m$
- $h_2^{desired} = 2.667m$

The total simulation time is 1000 secs.

3 Transfer Functions

1. Using a manual calculation on paper prove that equation (8) in the lecture slides is correct for the Feedback Controller Transfer Function shown in the Figure in the same slide.
2. Using Python's symbolic computation (Sympy) prove that equation (8) in the lecture slides is correct for the Feedback Controller Transfer Function shown in the Figure in the same slide.

   Hint: Use the solve() function in Sympy. Study the relevant documentation at:

   `https://docs.sympy.org/latest/guides/solving/solve-equation-algebraically.html`

4 Mock In-Class Test Questions

Without looking at the solutions provided, attempt the following mock in-class test questions found under the Assessments tab on Blackboard.

You can look up at the lecture slides and tutorials but NOT at the solutions provided for the mock test.