5ELEN018W - Tutorial 5 Exercises

1 Checkpoint

Based on the last tutorial session, only a few of the students finished all of the tutorial exercises.

Make sure that you finish them during this week (during the tutorial or on your own free time) as well as finishing all of the exercises since the beginning of the module.

If in doubt, you can show me your solution and get freedback.

Sample Solutions will be provided ONLY for some selected Exercises, not ALL!

2 Bode Plots in Simulink

Consider a dynamic system describing a watertank (Figure 1). Water enters the tank from the top at a rate proportional to the voltage, V, applied to the pump. The water leaves through an opening in the tank base at a rate that is proportional to the square root of the water height, H, in the tank. The presence of the square root in the water flow rate results in a nonlinear plant.

The dynamic system is described by the following differential equation:

$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - \alpha\sqrt{H} \tag{1}$$

where H is the height of the water in the tank, Vol is the volume of the water in the tank, b is a constant related to the flow rate into the tank, A relates to the area of the tank and α is a constant related to the flow rate out of the tank.

The following values should be used: $A = 20, \alpha = 2, b = 5, H_{ref} = 10$, where H_{ref} is the desired water level in the tank. You should add variables A, a, b in the main Matlab window or as part of the model (the latter is preferable).

The initial condition for the H level should be 1 and this should be set in the integrator.

Run the simulation for 200 steps.

1. Build a Simulink model of the system with a PID controller. The PID controller should use the following values for its parameters: $P = 0.208, P_I = 0.031, P_D = 0$.
2. Plot the Bode diagrams and make the appropriate calculations to analyse its stability (see the lecture slides and the lecture recording).

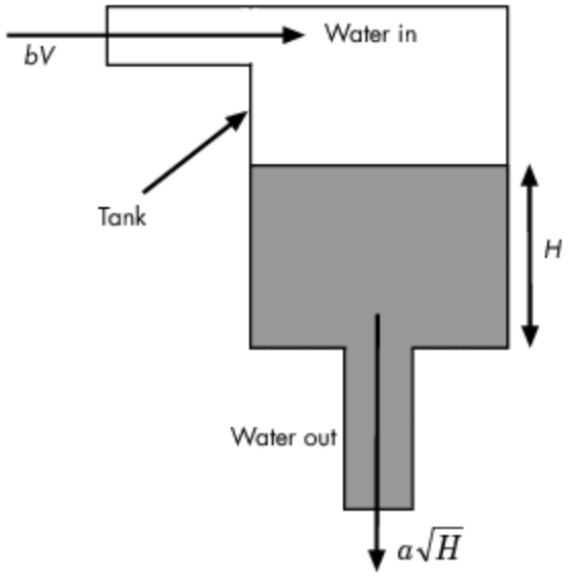


Figure 1: The watertank dynamic system.

3 Creating Transfer Functions in Matlab

The transfer function of a dynamic system can be created in Matlab by using similar code as the one described in the last lecture (Slides 7–8).

Implement in Matlab the following transfer function:

$$\frac{5s + 15}{77s^3 + 4s^2 + 1000s + 10} \tag{2}$$

4 Drawing Bode Plots in Matlab

1. Implement in Matlab the following transfer function:

$$\frac{0.5s + 5}{0.0002s^4 + 0.0064s^3 + 0.512s^2 + s} \tag{3}$$

2. Draw the Bode plot in Matlab by calling the `bode` function and passing it a single argument corresponding to a variable which is assigned to the transfer function in the previous step.

3. Compare your diagram with the one given in slide 11 of the lecture. Do they match?

4. Is the system stable or unstable? Justify your answer.

5 The Inverted Pendulum Control Problem

The inverted pendulum is a classic problem in dynamics and control theory and is used as a benchmark for testing control strategies. It is implemented with the pivot point mounted on a cart that can move horizontally with the usage of a force F produced by a motor.

The objective is to control the pendulum so that it is in its upright position at all times ($\theta = 0$).

The problem is shown in Figure 2.

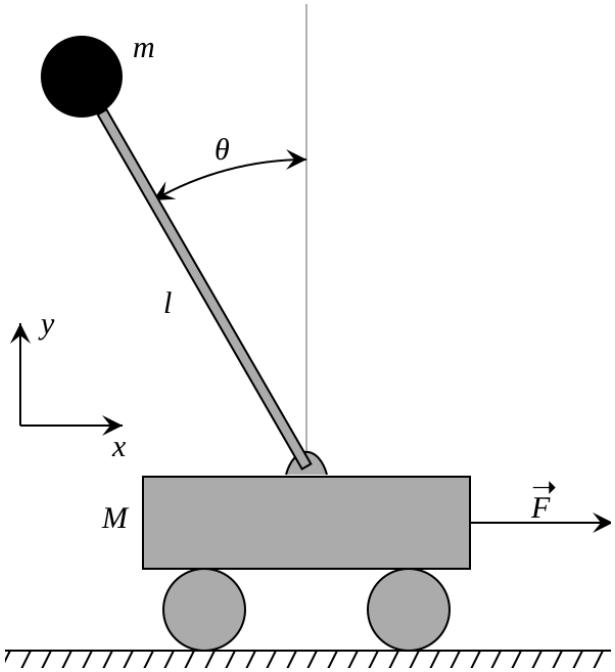


Figure 2: The inverted pendulum control problem.

The equations of motion for the system are:

$$(M+m)\,\ddot{x} + m\ell\cos\theta\,\ddot{\theta} - m\ell\dot{\theta}^2\sin\theta = F - b_c\dot{x}, \tag{4}$$

$$m\ell\cos\theta\,\ddot{x} + m\ell^2\,\ddot{\theta} - mg\ell\sin\theta = -b_p\dot{\theta}. \tag{5}$$

where

- Cart mass: $M = 1kg$
- Pole mass: $m = 0.1kg$
- Pole length: $l = 0.5m$
- Gravity: $g = 9.81m/s^2$
- Actuator (force) limits (typical): $F_{min} = -10N, F_{max} = +10N$
- Viscous friction on cart: $b_c = 0.10N \cdot s/m$
- Viscous friction at the pendulum pivot: $b_p = 0.01N \cdot m \cdot s/rad$

1. Model the dynamic system using Simulink.

2. Attempt to control the system use a PID controller. Can you find a set of parameters which solves the balancing problem?

3. Plot the Bode diagrams for the system based on your PID controller. What can you say about the stability of the system?

6 Implementing a Dynamic System and a Controller in Code (not Simulink)

In the last lecture, we implemented in Python the dynamic system and a PID controller for the car cruise control system.

Consider the robot surgery dynamic system that we studied in the last few weeks. Implement in a programming language of your choice (e.g. Java, Python, etc) the dynamic system and a PID controller with it.