Design of Embedded and Intelligent Systems

Robot Control - Lab Assignments

Asst. Prof. Eren Erdal Aksoy

eren.aksoy@hh.se

The main aim of these exercises is to give a gentle introduction to robot control. There exist three different exercises each has an accompanying Matlab script. Students are supposed to have basic programming skills and a good understanding of the theoretical concept explained during the lecture. Please read each exercise described below and try to code the missing parts in the corresponding Matlab script.

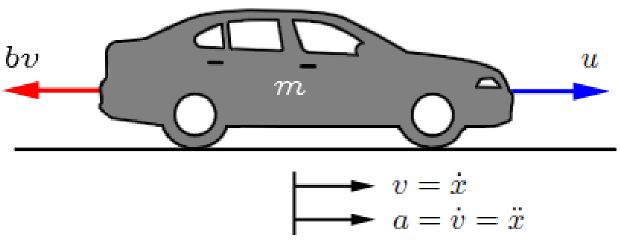
Exercise 01: Automatic cruise control

The main aim of this exercise is to design a feedback control system to keep the speed of the vehicle constant against external disturbances such as wind or road grade. To be able to maintain a constant vehicle speed, you have to implement a PID controller that measures the vehicle speed and adjusts it accordingly by comparing with the desired speed. Assume that the vehicle has a simple free-body diagram as shown below. We have a control force $\boldsymbol{u}$ on the vehicle. The term $\boldsymbol{m}$ represents the vehicle mass. In the direction opposite the vehicle's motion, there exist the linearly varying resistive forces $\boldsymbol{bv}$ introduced by rolling resistance and wind drag. With the assumption of having a first-order mass-damper system, the system equation of the vehicle can be derived by summing all forces in the x-direction and by referring to Newton's 2nd law as:

$$F = ma$$

$$u - bv = m\,\dot{v}$$

$$\dot{v} = \frac{1}{m}u - \frac{b}{m}v$$



Please have a look at the provided Matlab script (Lab_01.m) and write a PID controller by first computing the error, derivative, and integral parts separately. Next, you are supposed to write a set of code to alter the vehicle speed. Note that since we are trying to control the speed, the output equation can simply be written as follows:

$$y = v$$

Finally, to update the vehicle speed at each time step, you can use the following formula:

$$v(t) = v(t-1) + \left(\frac{1}{m}u(t) - \frac{b}{m}v(t)\right)dt$$

Hint: You can use the trapezoid rule to approximate the integral part of the controller.

Exercise 02: Automatic cruise control with Matlab state space representation

In this exercise, you are supposed to model the same system described in the previous exercise by using the Matlab state space representation. For this purpose, you have to take the *Laplace transform* of the system equation, which can be written as:

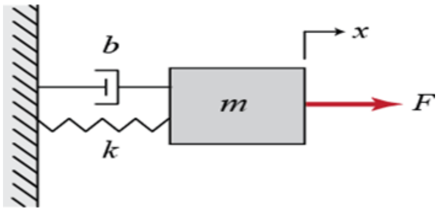
$$u = ms + b$$
$$y = 1$$

Please have a look at the provided Matlab script (Lab_02.m) and compute the transfer function of the system by referring to the Laplace transform. By using the ***pid*** function in Matlab, please compute the step response of the system for different P, I, and D values. Tune the PID gains until having a stable response.

Hint: Use the help function in Matlab to get more information on how to call the ***pid*** function.

Exercise 03: Spring-Mass-Damper System

Please have a look at the provided Matlab script (Lab_03.m) and compute the transfer function of the Spring-Mass-Damper system given below. You can use the same Matlab state space approach introduced in Exercise 02.



$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = F$$