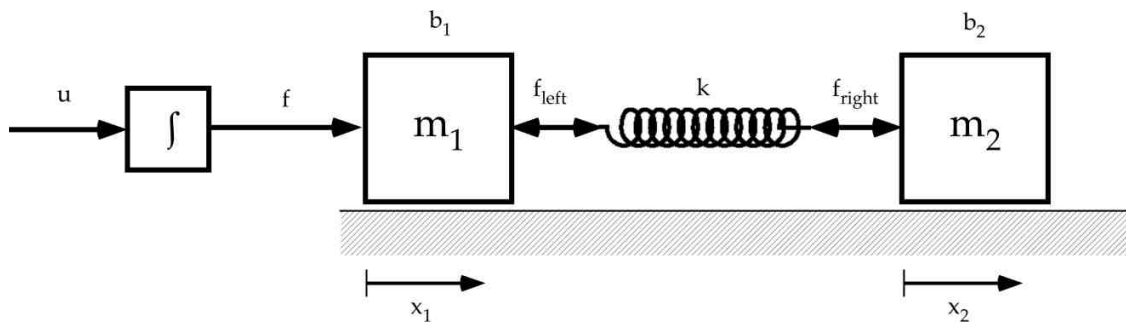


CS545–Introduction to Robotics

Homework Assignment 1 (Due February 28)

In the following problems, you should use MATLAB to compute numerical results and visualize the data, and Simulink for simulations. **IMPORTANT:** In your solutions of the homework, also provide intermediate steps how you derived the solution to a problem.

1) (60 Points) Consider the linear dynamical system



- This system is a model of an actuator with internal springiness, also called a series-elastic actuator. The system naturally decomposes into three subsystems: i) an integrator, ii) a mass system m_1 with viscous friction b_1 due to sliding on the floor, iii) a second mass system m_2 with viscous friction b_2 . Each of these subsystems has its own inputs and outputs, e.g., the input to the integrator is u and its output is a force f — a typical example of such a system is a hydraulic actuator, which has flow control. f_{left} and f are the inputs to the mass system m_1 . f_{right} is the input to the mass system m_2 . The position of the first mass is x_1 and the position of the second mass is x_2 . Write down the differential equation that characterizes the equation of motion of each subsystem. Note: ignore the spring as f_{left} and f_{right} are placeholders for the spring force.
- Transform the three subsystems into the frequency domain, and write down the transfer function of each subsystem.
- Draw a control diagram of the entire system, i.e., with u as input and x_1 and x_2 as output. Assume that the spring force can be calculated as $f_{spring} = (x_2 - x_1)k$. Hint: you will need a feedback loop to make this a proper control diagram.
- Write down the transfer function of the entire system, i.e., the system that receives u as input and has x_1 and x_2 as output.
- Use the provided skeleton Matlab Simulink file “spring_system.mdl” from the web. In this file you find a sine-wave generator that generates a desired position x_d and desired velocity \dot{x}_d . Insert the model of the above dynamic system by using the Transfer Function building blocks, reusing your knowledge of the control diagram from c) and the transfer functions from b). The physical values are $k = 10$, $b_1 = 1$, $b_2 = 10$, $m_1 = 0.1$, $m_2 = 1$. Add a PD controller and tune the gains such that you achieve good tracking of the desired system. Visualize the

tracking using the “Scope” building block of Simulink. Provide a printout of your tracking performance as picture (or screen capture) of the Scope. Provide a print-out of your entire Simulink control system (e.g., screen capture or print from the Simulink window). Provide the numerical values of your P and D gains. Is the tracking performance good? If yes, why? If no, why?

2) (40 Points) Download the files `oculomotor_fb.mdl` and `OculomotorAnimationSimple.m` from the homework webpage. They contain the simulation as described in Lecture 9 (see handouts on the web). In order to run this simulation, your matlab path has to be set to the directory where you downloaded these two files.

- a) To verify that the simulation works, run the simulation one time by setting the amplitude of the sinusoidal block to zero, and one time to $\pi/6$. Tune the gains of the PD controller to obtain the best possible performance. Give the plots of the retinal slip (over 10 seconds simulation time) in both cases. Make sure the plots are scaled to maximal resolution.
- b) The steady state performance (sinusoidal amplitude = 0) can be improved by using an integrator (I) controller. Add this element to the current control system. Tune the gains of the PID control system to get as good as possible performance (i.e., minimal tracking error, or fast convergence to desired position). Provide a plot of the modified Simulink control system and the same two plots as in a). Give a brief comment on the improvements over the original system, why they were achieved, and what is still unsatisfactory.
- c) The tracking performance of the system can be improved by adding a feedforward control system that employs an inverse dynamics model. For this purpose, take the dynamics equations of the eye system as discussed in Lecture V, and simply replaced the state and its derivatives by the desired state and its derivatives:

$$u_{ff} = m\ddot{\theta}_d + b\dot{\theta}_d + k\theta_d$$

The parameter m , b , and k , can be found by double clicking on the “Oculomotor Animation” block in the provided Simulink file. Add this feedforward controller to your Simulink model. Provide a plot of this new system, and its performance in the sinusoidal tracking task (amplitude = $\pi/6$). Briefly comment on the performance of the system. If your tracking error (i.e., retinal slip) has an amplitude greater than 0.01, give reasons why you think that this tracking error remains. (Hints: Inserting the inverse dynamics feedforward element is not quite trivial. Carefully examine which inputs the inverse dynamics model needs, and whether you have these inputs already in the simulation, or whether you need to generate them through additional Simulink blocks.)