CS545–Introduction to Robotics

Homework Assignment 1 (Due February 18)

In the following problems, you should use MATLAB to compute numerical results and visualize the data, and Simulink for simulations. A handout about getting started with MATLAB is in

http://www-clmc.usc.edu/Teaching/TeachingIntroductionToRoboticsHomework

This web page also contains all the files needed below. IMPORTANT: In your solutions of the homework, also provide intermediate steps how you derived the solution to a problem.

1) (65 Points) Consider the second order dynamical system:

$$\ddot{x} = b\dot{x} + kx + ru \tag{1}$$

a) Reformulate the system to achieve the canonical representation.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are $n \times n$ and $n \times m$ matrices, respectively, and $\mathbf{x}$, $\dot{\mathbf{x}}$ are $n \times 1$ vectors, while $\mathbf{u}$ is an $m \times 1$ vector. Give the definition of $\mathbf{x}$, $\mathbf{u}$, $\mathbf{A}$, $\mathbf{B}$, and $\dot{\mathbf{x}}$ in this new representation.

b) Graphically sketch a physical system that has the same dynamic equations as Equation (1), and give a physical interpretation of the variables and constants of Equation (1) with respect to the suggested physical system.

c) This dynamical system is to follow a desired trajectory characterized by $\mathbf{x}_d(t)$.Give the Proportional-Derivative (PD) negative feedback control law $\mathbf{u}_{fb}$ for this desired trajectory in vector-matrix notation.

d) Sketch the block diagram of the entire system with PD control

e) How well do you expect this PD-controller to work in general? (Brief answer! Pros & Cons).

f) Better control performance can be achieved by using a feedforward controller in form of an inverse dynamics controller in combination with the PD-controller. For this purpose, it is also necessary to know $\dot{\mathbf{x}}_d(t)$. The inverse dynamics control is derived by solving equation (2) for $\mathbf{u}$ and replacing the state and change-of-state by the desired values. Give the solution for the inverse dynamics control law $\mathbf{u}_{ff}$.

g) Discuss the solution for $\mathbf{u}_{ff}$ in general for the three different cases

1. n=m
2. n>m
3. n<m

by providing one analytical algebraic solution and the technical name for each case, and also keywords of how well you would expect each of the feedforward control systems to perform. (Hint: the matrix inversion is the critical part here, and what does it mean to invert non-square matrices).

h) Add the feedforward part in the graph of c) in a clearly different color (pen-width, etc.)

2) (65 Points) Consider the dynamical system:

$$\begin{aligned} m\dot{x}_1 &= bx_1 + ru \\ \dot{x}_2 &= k\,x_2 + x_1 \end{aligned} \tag{3}$$

where m denotes the mass, b the viscous friction coefficient, k the spring constant, and r the command amplification.

a) Transform the system into the frequency domain and give the individual transfer functions of the two equations and the transfer function of the complete system.

b) Assume m=1, b=-10, k=5, r=10. Use Simulink to build a PD controller for the system, assuming that the system should track a sinusoidal desired trajectory with 1Hz frequency. Use the “Transfer Function” building block in Simulink for this implementation. Attach a “Scope” to the simulation that simultaneously shows the true output of the system, and the desired output (note that “Scopes” accept vector inputs to display multiple trajectories superimposed). Also create a scope that shows the tracking error. Tune the gains of the PD controller to get as good as possible tracking performance. Give plots of i) your Simulink system, and ii) position and desired position as well as the tracking error of the best tracking performance (over 10 seconds simulation time). (Note, all Simulink windows allow copy/paste or direct printing of their contents). Make sure that all your plots are scaled such that they have maximal resolution in the vertical axis!

c) In real physical systems, the state of the controlled systems needs to be measured, usually by means of sensors that add some noise to the measurement. Additionally, the sensory feedback can only be measured at a certain frequency. In order to simulate such a sensory processing, add a “zero order hold” block (from the “Discrete” library) in the feedback path way of your system (i.e., the pathway leading back from the output of the system’s transfer function to the negative feedback controller). Set the “sample time” parameter of this block to 0.005 (i.e., 200Hz). To simulate the noise in the sensor, take the “Random Number” block from the “Sources” library and sum it into the feedback pathway after the “zero order hold”. Set the “sample time” parameter of this block also to 0.005, and the variance parameter to 0.01. Now tune your PD gains to get the best performance for this more realistic control system and provide plots of the best tracking performance and tracking error as in point b-ii above. Briefly describe the difference in performance between the system in b) and the current one, and why this difference exists.

d) In order to deal with sensory noise, signal filtering is usually employed. A very simple filter is of the form:

$$\dot{x} = \alpha(u - x)$$

where α is the time constant of this filter and regulates the amount of filtering: the smaller α, the more filtering is achieved. The danger of too much filtering, however, is that the filtered signal becomes delayed. Convert the filtering equation into a transfer function, and add it in the feedback pathway after the point where the noise was added. Adjust α to improve the performance of your control system, and provide the same plots as in c). Briefly comment on the success of your filtering. Note that you may have to re-tune the PD controller for the filtering.