Feedback Control
Homework and Lab-Exercises

EE100: Introduction to Information Science and Technology

Teacher: Prof. Boris Houska

Welcome!

Welcome to the lectures and exercises of Week 12 of ShanghaiTech's EE100 course. During Week 12 we will learn about the basic concepts of feedback control and its applications. This guide is about the course material, software tools and programming requirements, as well as the lab-exercises and homework. Some of the course sections will be easy to understand, others not so much. In order to benefit from the course including the programming exercises, the following sections help you to prepare yourself.

Boris Houska
SIST, ShanghaiTech.

Contents

1 Homework

1.1 Course Material

The lectures on May 18 and May 21 will be about feedback control. As these lectures will use basic linear algebra notation, please make sure that you download and **read the following material before the lecture**:

1. Lecture slides on feedback control for this course:

   Course Slides

2. Introductory course material about vectors, matrices, and matrix multiplication (from Stanford courses, click on the links):

   Vectors

   Matrices

   Matrix-Multiplication

3. Textbook on:

   Introduction to Matrix Methods and Applications

1.2 Software

During the lecture May 21 and for the lab exercises on May 23, we will implement a linear controller for an inverted pendulum. Please make sure that you download and **test the following Python code before the lecture**:

1. Python code for simulating an inverted pendulum

   Inverted Pendulum Code

2 Lab-Exercises

2.1 Simulation of an Inverted Pendulum (for all students)

1. During the in-class exercise on May 21, we will simulate an inverted pendulum. We will discuss the code together during the lecture, but please look into the code before the lecture and think about which lines you don't understand. The details about the model can be found in the lecture slides.

2. The goal of this in-class exercise is to tune the control gain $K = (a, b, c, d)$, which translates the measurement x of current state vector into a control reaction $u = Kx$. Recall that the control u is in our example the force with which we are pulling the trolley. First try to run the code with $K = (0, 10, 0, 0)$. How does the reaction of the pendulum look like? Interpret the result.

3. Try to adjust the other control gains a, c, and d step-by-step. Visualize your results and try to explains what happens if you tune these constants. Can you find gains that bring all system states to 0?

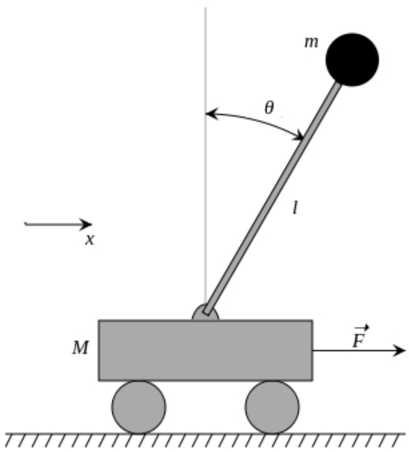

Figure 1: Inverted Pendulum

2.2 Simulation of an Inverted Spring-Pendulum (for SIST students)

On May 23 there will be an additional lab exercise for SIST students. We will learn how to implement a closed-loop system for an inverted spring-pendulum.

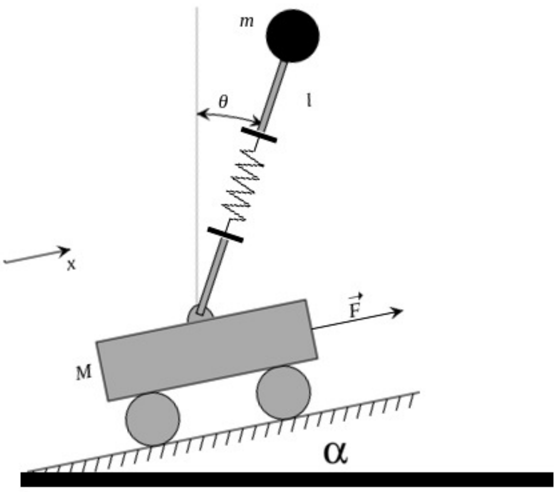

Figure 2: Inverted Spring Pendulum

1. This exercise is about controlling an inverted spring pendulum, which can move up and down a ramp with incline angle $\alpha = 0.2$ [rad]. The model has 6 states, namely,

$$z = \left(x, s, \theta, \dot{x}, \dot{s}, \dot{\theta}\right) \ ,$$

where x is the position of the trolley on the ramp, s the elongation of the spring with respect to equilibrium position, and θ the angle of pendulum with respect to the vertical axis (see Figure 2). For the case that $M = 0.4$, $m = 0.1$ are the mass of the trolley and pendulum, $g = 9.81$ the gravitational constant, $l = 0.2$ the equilibrium length of the pendulum and spring, and $D = 3$ the spring constant, the equations of motion are (for small excitations)

$$\dot{z} = Az + Bu$$

with

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1.5 & -2.4 & 0 & 0 & 0 \\ 0 & -30.3 & 0.5 & 0 & 0 & 0 \\ 0 & -7.3 & 60.8 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2.5 \\ -0.5 \\ -12.3 \end{pmatrix} .$$

Here, the control input $u = F - (M+m)g\sin(\alpha)$ is equal to the force F pulling on the trolley in x-direction—corrected by the constant force $(M+m)g\sin(\alpha)$ that is needed to prevent the trolley from moving down the ramp at equilibrium position. Can you explain why the matrices A and B have this particular structure? Where do the zeros and ones in the first three rows come from? Why are the accelerations independent of the trolley position x?

2. Implement a closed-loop simulation for the above spring pendulum. Start by making a safe copy of the "pendulum.py" file that you've downloaded from the course page. Use the other copy as a basis for the implementation. Which lines of the code do you need to change? What is the dimension of the control gain K? Think about these questions before you modify the code! Try to make a minimum amount of changes to the existing code.

3. The goal of this exercise is to stabilize the horizon position and velocity, the angle and angular velocity of the pendulum, as well as the elongation and elongation velocity of the spring. How is it possible that we can stabilize six states at the same time, although we have only one input, namely, the force F? First solve this puzzle in words and then start implementing the closed loop system. Adjust all 6 coefficient of the control law $u = Kx$ in such a way that all states of the closed-loop system converge to zero after a small excitation.

4. Write a visualization tool to generate an animation (or video) of your results.