State Feedback
MAE 433 Spring 2012
Lab 7

Prof. C. Rowley and M. Littman
AIs: Brandt Belson, Jonathan Tu

Princeton University

April 24-27, 2012

1 Overview

This lab addresses the control of an inverted pendulum balanced on a rotary beam. In this lab, we stabilize the pendulum using full state feedback, and explore both pole placement and LQR as techniques for choosing the feedback gains.

2 Background

In the previous lab, we found an approximate linear model of the rotary pendulum system in the form

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ \mathbf{y} &= C\mathbf{x}\end{aligned}$$

where $\mathbf{x} = (\alpha, \theta, \dot{\alpha}, \dot{\theta})$ and u is the voltage to the motor. In this lab, we will design state feedback controllers of the form

$$u = -K\mathbf{x}, \tag{1}$$

where the gain $K = [k_1 \;\; k_2 \;\; k_3 \;\; k_4]$. In this lab we can observe all the states $\alpha, \theta, \dot{\alpha}, \dot{\theta}$, so C is a 4×4 identity matrix.

3 Setup Simulink model

The first step is to set up the simulink model to implement a control law of the form in Equation 1.

1. Open a unmodified template file as you started with in lab 6 (available on blackboard) and save a new copy.

2. Modify the file to reproduce the Simulink diagram given in Figure 1.
In Figure 1, the rectangle on the right is a summing block (not a mux), and you can change a round summing block into a rectangular one by double clicking on it and selecting the

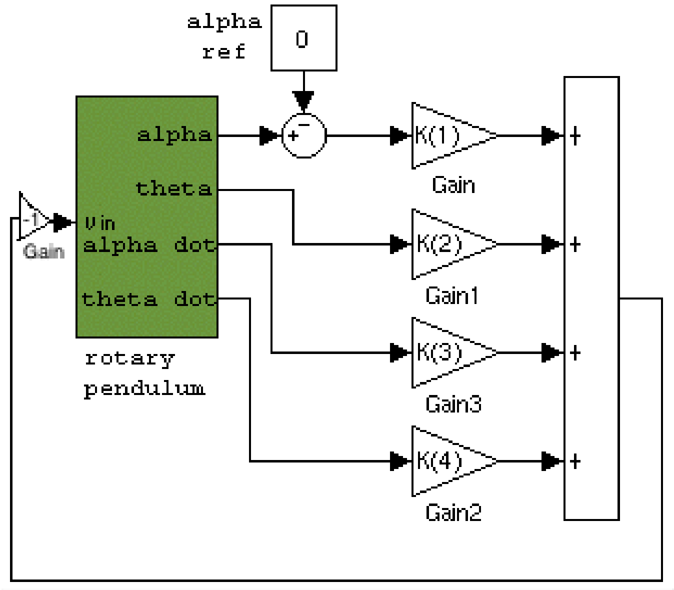

Figure 1: Rotary Pendulum Schematic

appropriate options. Also, note the minus signs in the summing block. We will set K in Matlab later.

3. Build, and bring up scopes to display $\alpha, \dot{\alpha}, \theta, \dot{\theta}$, and V_{in} (Motor input).

4. Initially, assign the matrix gain $K = [1\ 0\ 0\ 0]$ from within Matlab. Then run the experiment. Change α_{ref} while WinCon is running to make the rotary beam move around, as in lab 6.

5. Check that your model from last week is still valid.

   (a) Set $K = [k_p,\ 0,\ k_d,\ 0]$, with your values from last week.

   (b) Perform an experimental step response in α_{ref} (about 0.3 radians), and save your values of α and θ.

   (c) Check that your experimental result looks like your model result from lab 6. (If you do not have your lab 6 results available, then perform a simulated step response, using your state space system from lab 6, and compare.)

Discuss with an AI before proceeding.

4 Pole placement

Next, we stabilize the pendulum, choosing the gains K by pole placement. It is important that you verified your model for the pendulum-down case with the steps above.

The state-space equations linearized about the pendulum up position are

$$\frac{d}{dt}\begin{bmatrix}\alpha\\ \theta\\ \dot{\alpha}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -\frac{mgr}{J} & -\frac{K_2}{J} & 0\\ 0 & \frac{(J+mr^2)g}{Jl} & \frac{rK_2}{Jl} & 0\end{bmatrix}\begin{bmatrix}\alpha\\ \theta\\ \dot{\alpha}\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{K_1}{J}\\ -\frac{rK_1}{Jl}\end{bmatrix}u. \tag{2}$$

1. Create a state space system in Matlab for the pendulum up case by entering

   ```
   C = eye(4);
   sys = ss(A, B, C, 0);
   ```

2. Plot the *open-loop* pole locations of the inverted pendulum.

3. Verify that your system is controllable by using the `ctrb` and `rank` commands in Matlab. If the system is controllable, then you can use pole placement.

4. Choose closed-loop pole locations to be the same as open loop, but mirror image any RHP poles so they are stable.

5. Compute the gains K that achieve these closed-loop pole locations via the `place` command (see Matlab's help, `help place`). Look at the gains, notice anything?

6. Create the closed-loop system in Matlab. `sysCL = feedback(sys, K, -1)`

7. Simulate an initial condition response (equivalent to a step response in the reference) in α and plot it.

   ```
   t = 0:0.01:5;
   xInitial = [0.3, 0, 0, 0];
   [y, t, x] = initial(sysCL, xInitial, t);
   figure
   hold on
   plot(t, y)
   plot(t, x*K')
   legend('alpha','theta','alphadot','thetadot','u=Vin')
   ```

   Discuss with an AI before proceeding.

8. If the voltages look reasonable, try this out on the experiment. **REMEMBER TO START THE PENDULUM IN THE BALANCED, INVERTED POSITION BEFORE RUNNING THE MODEL!**
   It's important to plot the simulated response before using a controller on the experiment. If the predicted voltages are too large, do not try this controller on the experiment.
   Discuss with an AI before proceeding.

9. Does this stabilize the pendulum in the up position? What happens if you change α_{ref}?

10. There should be one closed loop pole at the origin. Choose the gains to move this pole slightly into the left half plane (again with `place`).

11. Adjust the pole locations and simulate your system to observe the effect on the initial condition response.

12. Design the "best" controller you can using pole placement, trying the controllers in simulations and experiments. Save a plot of your best controller's response (both simulation and experiment), and specify your choice of poles and the resulting K. *It might be good to gently nudge the pendulum to see how it responds to disturbances.* **Discuss with an AI before proceeding.**

5 LQR

In this section we will use LQR to choose the poles and gains for us, such that the cost function is minimized and the closed-loop system is stable.

The Matlab command `lqr` is useful, `[K,S,E] = lqr(A, B, Q, R)`.

1. Start with the following weights for LQR: $Q = \mathrm{diag}([1\ \ 1\ \ 0\ \ 0])$, $R = 1$. Compute the optimal gains, simulate, and plot the closed-loop step responses and pole locations.

2. If the voltages look reasonable, try your controller out on the experiment.
   Gently nudge the pendulum to see how it responds to disturbances.

3. Add weights on derivatives: $Q = \mathrm{diag}([1\ \ 1\ \ 1\ \ 1])$. Note the effects in simulations and experiments.

4. Try increasing the weight on α. Note how the values of the gains change, how the closed-loop poles change, and how response changes (simulations first, then experiments).

5. Try eliminating θ weight, does the result match your expectation?
   Remember: LQR gives a stabilizing controller.
   Discuss with an AI before proceeding.

6. Find the "best" controller you can by choosing Q, first simulating, then trying in experiments. What are your criteria are for evaluating how "good" a controller is? Save step response in α_{ref} from an experiment only, plotting all four state variables, stating the Q, R, and resulting K.
   Discuss with an AI before proceeding.

7. Save your simulink model file.

5.1 The Crane Problem

In this section of the lab we will deal with the non-inverted, or standard planar pendulum. We imagine the rotary beam as a crane with the pendulum in the down position as a load to be displaced. We wish to move the load as quickly as a possible but minimize the amount of swinging. First you will use a proportional controller to bring the rotary beam to the desired location, and then use an LQR controller to do so.

Recall that the state-space equations linearized about the pendulum down position are

$$\frac{d}{dt}\begin{bmatrix}\alpha\\ \theta\\ \dot{\alpha}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -\frac{mgr}{J} & -\frac{K_2}{J} & 0\\ 0 & -\frac{(J+mr^2)g}{Jl} & -\frac{rK_2}{Jl} & 0\end{bmatrix}\begin{bmatrix}\alpha\\ \theta\\ \dot{\alpha}\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{K_1}{J}\\ \frac{rK_1}{Jl}\end{bmatrix}u. \tag{3}$$

1. Use proportional control by setting

```
kp=8;
K=[kp 0 0 0];
```

2. Add a α reference to the signal α.

3. Run the experiment with the pendulum in the down position. Note the oscillations in the pendulum, θ.
4. Design an LQR controller that penalizes non-zero pendulum angles and non-zero rotary beam locations. Use similar Q matrices to the previous ones, but try different combinations to find the 'best' controller.
5. Plot all state variables for a step response in α_{ref}. Again, specify the Q, R, and K matrices and the closed-loop poles.

5.2 Sensor noise (optional)

You probably noticed that the rotary beam is usually quite jittery. In particular, the signals for the velocities $\dot{\alpha}$ and $\dot{\theta}$ are very noisy, and so the control voltage V_{in} is noisy as well, especially when gains K_3 and K_4 are large.
OPTIONAL: Adjust the low-pass filters in the `Rotary pendulum` block in the Simulink diagram, and see how these affect the noise level, and the closed-loop response.

6 Deliverables

1. Plots of a simulated and experimental step-response for the 'best' controller you can design using pole placement for the inverted pendulum. Specify the gains and poles.
2. Plots of a simulated and experimental step-response for the 'best' controller you can design using LQR for the inverted pendulum. Specify the weights, gains, and poles.
3. Plots of only the experimental step-response for the 'best' controller you can design using LQR for the crane problem. Specify the weights, gains, and poles.