ECE-320: Linear Control Systems
Homework 6

Due: Tuesday April 14, 2011 at the beginning of class

If matrix P is given as

$$P = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

then

$$P^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

and the determinant of P is given by $ad-bc$. This determinant will also give us the characteristic polynomial of the system.

1) For each of the systems below:

- determine the transfer function when there is state variable feedback
- determine if k_1 and k_2 exist ($k = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$) to allow us to place the closed loop poles anywhere. That is, can we make the denominator look like $s^2 + a_1 s + a_0$ for any a_1 and any a_0. If this is true, the system is said to be ***controllable***.

a) Show that for

$$\dot{q} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} q + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} q + [0]u$$

the closed loop transfer function with state variable feedback is

$$G(s) = \frac{(s-1)G_{pf}}{(s-1)(s-1+k_2)}$$

b) Show that for

$$\dot{q} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} q + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} q + [0]u$$

the closed loop transfer function with state variable feedback is

$$G(s) = \frac{sG_{pf}}{s^2 + (k_2 - 1)s + k_1}$$

c) Show that for

$$\dot{q} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} q + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} q + [0]u$$

the closed loop transfer function with state variable feedback is

$$G(s) = \frac{G_{pf}}{s^2 + (k_2 - 1)s + (k_1 - 1)}$$

2) For one of the rectilinear systems in lab, I found the following state variable representations:

$$\dot{q} = \begin{bmatrix} 0 & 1 \\ -174.8205 & -2.6469 \end{bmatrix} q + \begin{bmatrix} 0 \\ 5050 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} q + [0]u$$

The closed loop transfer function with state variable feedback is

$$G_o(s) = \frac{5050 G_{pf}}{s^2 + (2.6469 + 5050k_2)s + (174.8205 + 5050k_1)}$$

a) ***<u>Design I</u>***

- Determine feedback gains k_1 and k_2 so the resulting system has real poles and a bandwidth of 2 Hz. Set the second pole twice as far from the origin as the first pole.
- Use the final value Theorem to determine the prefilter gain so the steady state error for a step input is zero.
- Estimate the settling time for the step response based on the system bandwidth.
- Use Matlab to plot the Bode plot of the closed loop system (to verify the bandwidth) and plot the step response (to verify the estimated settling time).

You should get numbers like 0.028, 0.0069, 0.06, and 0.32. When you look at the Bode plot, the bandwidth will not be 2 Hz, primarily because the second pole is too close to the first pole.

b) ***<u>Design II</u>***

- Determine feedback gains k_1 and k_2 so the resulting (ideal second order) system has a percent overshoot of 15% and a settling time of 0.5 seconds.
- Use the final value Theorem to determine the prefilter gain so the steady state error for a step input is zero.
- Use Matlab to plot the step response (to verify the percent overshoot and settling time).
- Determine (analytically) the steady state output if the input to the closed loop system is r(t) = 1 cos(4t) cm and determine the time delay between the input and the output.

You should get numbers like 0.0128, 0.0026, 0.047, and a time delay of 0.07 seconds. Your step response should be fairly close to the requirements.

3) For one of the rotational systems in lab, I found the following state variable representations:

$$\dot{q} = \begin{bmatrix} 0 & 1 \\ -329.8427 & -1.2501 \end{bmatrix} q + \begin{bmatrix} 0 \\ 938.3581 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} q + [0]u$$

The closed loop transfer function with state variable feedback is

$$G_o(s) = \frac{938.3581 G_{pf}}{s^2 + (1.2501 + 938.3581 k_2)s + (329.8427 + 938.3581 k_1)}$$

a) ***<u>Design I</u>***

- Determine feedback gains k_1 and k_2 so the resulting system has real poles and a bandwidth of 3 Hz. Set the second pole twice as far from the origin as the first pole.
- Use the final value Theorem to determine the prefilter gain so the steady state error for a step input is zero.
- Estimate the settling time for the step response based on the system bandwidth.
- Use Matlab to plot the Bode plot of the closed loop system (to verify the bandwidth) and plot the step response (to verify the estimated settling time).

You should get numbers like 0.4058, 0.058939, 0.76, and 0.2. When you look at the Bode plot, the bandwidth will not be 2 Hz, primarily because the second pole is too close to the first pole.

b) ***<u>Design II</u>***

- Determine feedback gains k_1 and k_2 so the resulting (ideal second order) system has a percent overshoot of 10% and a settling time of 0.8 seconds.
- Use the final value Theorem to determine the prefilter gain so the steady state error for a step input is zero.
- Use Matlab to plot the step response (to verify the percent overshoot and settling time).
- Determine (analytically) the steady state output if the input to the closed loop system is $r(t) = 15^o \cos(10t)$ and determine the time delay between the input and the output.

You should get numbers like -0.27, 0.009, 0.076, and a time delay of 0.18 seconds. Your step response should be fairly close to the requirements.

Preparation for Lab 7

4) *You will be using this code and the following designs in Lab 7, so come prepared! This prelab is really pretty mindless, so just follow along*

The two degree of freedom Simulink model (**Basic_2dof_State_Variable_Model.mdl**) implements a state variable model for a two degree of freedom system. This model uses the Matlab code **Basic_2dof_State_Variable_Model_Driver.m** to drive it. Both of these programs are available on the course website (you downloaded them last week)

a) Design a state variable feedback system using **pole placement** for either your torsional or your rectilinear system. We want the output to be the position of the **first cart** (this determines the C matrix). For this method, we basically guess the pole locations and simulate the system. To set the location of the closed loop poles, find the part of the code that assigns poles to the variable p, and change the elements of p. You will need to choose the closed loop pole locations (This is a guess and check sort of thing. The biggest problem is making sure the control effort is not too large.) Your resulting design must have a settling time of 0.5 seconds or less and must have a percent overshoot of 10% or less. Your design should not saturate the system (control effort) and you should use a **constant prefilter**.

b) Run your simulation for 2.0 seconds. Plot both the system output (from 0 to 2 seconds) and the control effort (from 0 to 0.2 seconds). Plot the control effort only out to 0.2 seconds since the control effort is usually largest near the initial time. If your control effort reaches its limits, you need to go back and modify your design. Turn in your plot with your closed loop poles and your gains (you can just write these on your plot).

c) An alternative method for determining the feedback gains is based on what is called a **linear quadratic regulator**. The linear quadratic regulator finds the gain K to minimize

$$J = \int_0^\infty \left[x^T(t)Qx(t) + u(t)^T Ru(t) \right] dt$$

where

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ u(t) &= -Kx(t) \end{aligned}$$

For our two degree of freedom systems, Q is a 4x4 positive definite matrix, and R is a scalar. Since we will use a diagonal matrix for Q and for our system $u(t)$ is a scalar, we can rewrite J as

$$J = \int_0^\infty \left[q_1 x_1^2(t) + q_2 \dot{x}_1^2(t) + q_3 x_2^2(t) + q_4 \dot{x}_2^2(t) + Ru^2(t) \right] dt$$

A large value of R penalizes a large control signal, a large value of q_1 will penalize the position of the first cart, while a large value of q_2 will penalize a large value of the velocity of the first cart. All of the q_i should be zero or positive.

It's easiest to find K using the following command in Matlab: $K = lqr(A, B, diag([q_1 \quad q_2 \quad q_3 \quad q_4]), R)$

Try different values of the q_i to find an acceptable controller. Again you must have a percent overshoot less than 10% and a settling time of less than 0.5 seconds. You should also try to produce a controller with a smooth trajectory (not a lot of unnecessary motion). Trajectories can be made more smooth by penalizing slopes. Turn in your plot with your closed loop poles and your gains (you can just write these on your plot).

d) Repeat parts a-c for a system where the output is the position of the **second cart**

Turn in your plots and your code. Be sure your plots are accurately labeled!