ECE-420: Discrete-Time Control Systems
Homework 7

Due: Friday October 23 in class
Exam 2: Friday October 30

1) For the discrete-time state variable system given by

$$x(k+1) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k)$$

a) Assuming state variable feedback, use Akermann's formula to find a state variable feedback matrix K to place the closed loop poles at 0 and 0.1.

b) Use Ackermann's formula to find a state variable feedback matrix K that will result in deadbeat control.

c) Assuming state variable feedback, use the direct eigenvalue assignment method to find a state variable feedback matrix K to place the closed loop poles at 0 and 0.1.

2) For the discrete-time state variable system given by

$$x(k+1) = \begin{bmatrix} 0.1 & 0 \\ 0.2 & -0.1 \end{bmatrix} x(k) + \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(k)$$

a) Assuming state variable feedback, use Ackermann's formula to find a state variable feedback matrix K to place the closed loop poles at 0 and 0.1.

b) Use Akermann's formula to find a state variable feedback matrix K that will result in deadbeat control.

c) Assuming state variable feedback, use the direct eigenvalue assignment method to find a state variable feedback matrix K to place the closed loop poles at 0 and 0.1.

3) For the discrete-time state variable system given by

$$x(k+1) = \begin{bmatrix} 0.1 & 0.2 \\ 0.2 & 0.1 \end{bmatrix} x(k) + \begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(k)$$

Assuming state variable feedback, use the direct eigenvalue placement method to find two different state variable feedback matrices K to place the closed loop poles at -0.1 and -0.2.

4) In this problem we will explore a form or writing the state equations that makes determining the required state variable feedback coefficients easier to determine. This form is call the *controllable cannonical* form. We will restrict ourselves to a third order system for simplicity. (Note that there is also an *observable canonical* form, and we do similar things with that to place observer poles.

a) Consider the following state variable model,

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(\mathrm{t}) = \begin{bmatrix} b_3 - a_3 b_0 & b_2 - a_2 b_0 & b_1 - a_1 b_0 \end{bmatrix} x(t) + b_0 u(t)$$

Show that (feel free to use Maple on this part) the transfer function for this system is

$$\frac{Y(s)}{U(s)} = G_p(s) = \frac{b_0 s^3 + \mathrm{b}_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3}$$

So the characteristic polynomial is $\Delta(s) = s^3 + a_1 s^2 + a_2 s + a_3$. Note that is it very easy to determine the characteristic polynomial coefficients from the A matrix. Remember that the characteristic polynomial is given by $\det(sI - A) = \Delta(s)$

b) Consider now introducing state variable feedback, so $u(t) = -Kx(t) = -\begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} x(t)$. Show that the characteristic equation is now given by

$$\Delta(s) = s^3 + (a_1 + k_3)s^2 + (a_2 + k_2)s + (a_3 + k_1)$$

If we want the characteristic equation for closed loop poles μ_1, μ_2 and μ_3 to be given by

$$\Delta(s) = s^3 + \beta_1 s^2 + \beta_2 s + \beta_3 = (s - \mu_1)(s - \mu_2)(s - \mu_3)$$

Then it should be pretty easy to determine what we want the feedback gain matrix (vector) K to be.

At this point it should be pretty clear that if a system is written in controllable canonical from than it is pretty easy to determine the state variable feedback gains. However, there are very few systems that can be written in this way that have physically meaningful states, so this looks like something of only academic value. However, with a little work, we can transform a system to this form, and that is what much of the remainder of this problem deals with.

(continued on the next page)

For our (3rd order system) we can write the controllability matrix as $M = \begin{bmatrix} B & AB & A^2B \end{bmatrix}$ and the characteristic polynomial is $\Delta(s) = s^3 + a_1 s^2 + a_2 s + a_3$.

c) Show that we can write $AM = M\begin{bmatrix} 0 & 0 & -a_3 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_1 \end{bmatrix}$

Hint: on the right hand side, use the Cayley-Hamilton Theorem and the characteristic polynomial to write A^3 *in terms of lower powers of* A.

This means that $M^{-1}AM = \begin{bmatrix} 0 & 0 & -a_3 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_1 \end{bmatrix}$

d) Now let's define $W = \begin{bmatrix} a_2 & a_1 & 1 \\ a_1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ and $T = MW$. Show that $T^{-1}AT = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}$

Hint: After a little work you end up needing to show that $\begin{bmatrix} 0 & 0 & -a_3 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_1 \end{bmatrix} W = W \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}$

e) Show that $T^{-1}B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ or, equivalently, $B = T\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

At this point we have most of the tools we need, now we just need to put everything together.

f) Consider the state variable model

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Let's define a new vector $\hat{x}(t)$ where $x(t) = T\hat{x}(t)$. We can rewrite the first equation in the form

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t)$$

What are the $\hat{A}$ and $\hat{B}$ matrices, in terms of other matrices we have defined?

Note that we are really just doing a change of basis, and that now our first equation is in controllable canonical form.

g) Let's now assume that we have state variable feedback, so $u(t) = -Kx(t) = -KT\hat{x}(t) = -\hat{K}\hat{x}(t)$ where $\hat{K} = KT = \begin{bmatrix} \delta_3 & \delta_2 & \delta_1 \end{bmatrix}$. Show that the characteristic equation for this system is now $\Delta(s) = s^3 + (a_1 + \delta_1)s^2 + (a_2 + \delta_2)s + (a_3 + \delta_3)$.

h) If we wanted the closed loop characteristic equation to be

$$\Delta(s) = (s-\mu_1)(s-\mu_2)(s-\mu_3) = s^3 + \alpha_1 s^2 + \alpha_2 s + \alpha_3$$

how should we determine the δ_1, δ_2 and δ_3?

i) Show that our state variable feedback gain is then given by

$$K = \begin{bmatrix} \alpha_3 - a_3 & \alpha_2 - a_2 & \alpha_1 - a_1 \end{bmatrix} T^{-1}$$

Note that we do not need to transform the system to determine the a_i *, we only need to determine the characteristic equation of the original system. Now for some practice.*

j) Consider the system described by $\dot{x}(t) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$. Determine the state feedback gain so the closed loop poles are at -1 and -2. *Answer:* $K = \begin{bmatrix} -1 & 7 \end{bmatrix}$.

Hint: In this case $W = \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}$

k) Consider the system described by $\dot{x}(t) = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t)$. Determine the state feedback gain so the closed loop poles are at $-1 \pm 2j$. *Answer:* $K = \begin{bmatrix} 6 & 10 \end{bmatrix}$

Finally, although this method was derived in the continuous-time domain, the same basic approach applies in the discrete-time domain (just replace s *with* z *)*

5) The one degree of freedom discrete –time Simulink model **DT_sv1.mdl** implements a state variable model with state feedback. This model uses the Matlab code **DT_sv1_driver.m** to drive it. Both of these programs are available on the course website. The blue and yellow blocks represent the *model* of the system. Note that the Matlab driver program assumes a one unit delay between the input and output.

a) Load the continuous time one degree of freedom ***torsional*** state variable description, **bobs_1dof_model205.mat.** The program will allow you to change the sampling rate if you need to.

b) Modify the Matlab code (the poles and the **place** command) to place the poles of the closed loop system in such a way so that for a 15 degree step input:

- the settling time for your system is less than 1 s
- the percent overshoot for your system is less than 20%
- the control effort does not hit a limiter (does not saturate)
- the steady state error is zero

c) *Simulate the system and turn in your graphs.* Try at least two different sampling rates (do not make the sampling interval less than 0.01) so you can see how the system responds as the sampling rate is increased, and how the control effort is usually increased as the sampling interval is decreased.

d) Modify the Matlab code so your system has deadbeat response, if possible. You may need to use the **acker** command instead of place to do this. You may have to increase the sampling interval for this to work. *Turn in your graphs.*

6) Copy **DT_svl.mdl** and **DT_sv1_driver.m** to **DT_sv2.mdl** and **DT_sv2_driver.m,** respectively, and then modify the new files to work with a two degree of freedom system. All of the states except the last (delayed input) state must be plotted, as well as the control effort and the output state (y). You should use subplot(3,2) so each state has its own plot. Assume we want to control the position of the second disk (You may have to change the C matrix to do this.) You will need to modify both the initial conditions on the delay block, and you will need 5 states output from the **demux** (x1, x1_dot, x2, x2_dot, and delayed u). Then

a) Load the continuous time two degree of freedom ***torsional*** model, **bobs_2dof_model205.mat.**

b) Modify the Matlab code (the poles and the **place** command) to place the poles of the closed loop system in such a way so that for a 15 degree step input, *for the first disk*:

- the settling time for your system is less than 1 s
- the percent overshoot for your system is less than 20%
- the control effort does not hit a limiter (does not saturate)
- the steady state error is zero

Simulate the system *and turn in your graph*.

c) If you do not like your response, you may change the sampling interval, but don't make the sampling interval smaller than 0.01 seconds.

d) Repeat part b (and c, if necessary) to control the position of the second disk. *Turn in your graph.*

e) Modify the Matlab code so your system has deadbeat response, if possible. You may need to use the **acker** command instead of place to do this. You may control the position of either disk. *Turn in your graph.*