ECE-520: Discrete-Time Control Systems
Homework 6

Due: Monday January 25 in class
Exam 2 Thursday January 28

1) For the following matrix

$$A = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 1 & 1 & 0 & 2 \end{bmatrix}$$

a) Find a set of vectors that form a basis for the null space of A.

b) Is the vector $\underline{n} = \begin{bmatrix} 2 & 2 & -2 & 2 \end{bmatrix}^T$ in the null space of A? That is, can you represent this vector as a linear combination of your basis vectors?

2) For the following matrix

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 2 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

a) Find a set of vectors that form a basis for the null space of A.

b) Is the vector $\underline{n} = \begin{bmatrix} 2 & 6 & -2 & -1 \end{bmatrix}^T$ in the null space of A? That is, can you represent this vector as a linear combination of your basis vectors?

3) For the following matrix

$$A = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 2 & 1 & 3 & 5 \end{bmatrix}$$

a) Find the rank of A (the number of linearly independent rows or columns).

b) Determine two vectors that span the null space of A.

4) 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.

d) Use the direct eigenvalue assignment method to find a state variable feedback matrix K that will result in deadbeat control.

e) Using the value of K determined in part **b**, determine the correct value of prefilter gain G_{pf} so the steady state error for a unit step is zero. *Feel free to use a computer on this part.*

5) 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.

d) Use the direct eigenvalue assignment method to find a state variable feedback matrix K that will result in deadbeat control.

e) Using the value of K determined in part **b**, determine the correct value of prefilter gain G_{pf} so the steady state error for a unit step is zero. *Feel free to use a computer on this part.*

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

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

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

a) 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.

b) Use the direct eigenvalue placement method to find a state variable feedback matrix K that will result in deadbeat control.

8) Pre-Lab: To be done individually. (*Servo System)* Copy your program **DT_sv1_driver.m** to a new file **DT_sv1_servo_driver.m,** then copy the Simulink file **DT_sv1.mdl** to **DT_sv1_servo.mdl .** Modify these ***new*** files to implement both state variable feedback and an integrator control . Your programs should be able to use either **place** or **acker** to place the closed loop poles directly. Note that if our system has an integrator in it, there should be no prefilter.

Using this code and the LQR algorithm, show that for your one degree of freedom ***rectilinear*** system you can meet the following design constraints:

For a 1 cm 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

Turn in your graph.

Using this code and the **acker** command, design and simulate a deadbeat response for your one degree of freedom rectilinear system. Don't worry if your system saturates the control effort, you have no control over this.

Turn in your graph.

Copy your program **DT_sv1_servo_driver.m** to a new file **DT_sv2_servo_driver.m,** then copy the Simulink file **DT_sv1_servo.mdl** to **DT_sv2_servo.mdl .** Modify these ***new*** files to implement both state variable feedback and an integrator control. Your programs should be able to use either **place** or **acker** to place the closed loop poles directly. Note that if our system has an integrator in it, there should be no prefilter.

Using this code and the LQR algorithm, show that for your two degree of freedom ***rectilinear*** system you can meet the following design constraints:

For a 1 cm step input (assuming the second cart is the output):

- 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

Turn in your graph.

Using this code and the **acker** command, design and simulate a deadbeat response for your two degree of freedom rectilinear system. Don’t worry if your system saturates the control effort, you have no control over this.

Turn in your graph.

Finally, turn in your code for the 2 dof system and a copy of the corresponding Simulink model file.