1 Steady State Error (30 pts)

Given the following continuous time (CT) system

\[
\dot{x} = A_1 x + B_1 u = \begin{bmatrix} 0 & 1 \\ -12 & -7 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad y = \begin{bmatrix} 10 & 0 \end{bmatrix} x \tag{1}
\]

a) Given error \( e(t) = r(t) - y(t) \) where \( r(t) \) is a scalar, find \( \lim_{t \to \infty} e(t) \) for input \( r(t) \) a unit step, and control law \( u(t) = r(t) - y(t) \).

Recall that

\[
e_{ss} = \left( I + C \bar{A}^{-1} \bar{B} \right) r_{ss}
\]

\[
\bar{A} = A - BK C
\]

\[
\bar{B} = BK
\]

when \( K \) is the controller gain. In this case, \( K = 1 \). Therefore:

\[
\bar{A} = \begin{bmatrix} 0 & 1 \\ -12 & -7 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 10 & 0 \end{bmatrix}
\]

\[
= \begin{bmatrix} 0 & 1 \\ -12 & -7 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 10 & 0 \end{bmatrix}
\]

\[
= \begin{bmatrix} 0 & 1 \\ -22 & -7 \end{bmatrix}
\]

\[
\bar{A}^{-1} = \frac{1}{22} \begin{bmatrix} -7 & -1 \\ 22 & 0 \end{bmatrix}
\]

\[
\bar{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\]

\[
I + C \bar{A}^{-1} \bar{B} = 1 + \begin{bmatrix} 10 & 0 \end{bmatrix} \frac{1}{22} \begin{bmatrix} -7 & -1 \\ 22 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\]

\[
= 1 - \frac{5}{11}
\]

\[
e_{ss} = \frac{6}{11} r_{ss} = \frac{6}{11}
\]

b) Evaluate the steady state error \( \lim_{t \to \infty} e(t) \) for input \( r(t) \) a unit step, with state feedback, that is, \( u = -K_1 x + r \), where \( K_1 \) is chosen so that the closed loop poles are at \( s_i = -5, -5 \).

First, find \( K_1 \). We want the eigenvalues of \( A_1 - B_1 K_1 \) to be \( (-5, -5) \). Find the characteristic
equation, then make it equal \((s + 5)(s + 5)\):

\[
A_1 - B_1 K_1 = \begin{bmatrix} 0 & 1 \\ -12 & -7 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \\
= \begin{bmatrix} 0 & 1 \\ -12 - k_1 & -7 - k_2 \end{bmatrix}
\]

\[
\Delta(s) = \begin{bmatrix} -s & 1 \\ -12 - k_1 & -7 - k_2 - s \end{bmatrix} \\
= s(s + 7 + k_2) + k_1 + 12 \\
= s^2 + (7 + k_2)s + (k_1 + 12) \\
= (s + 5)(s + 5) = s^2 + 10s + 25
\]

\[
k_1 + 12 = 25 \\
k_1 = 13 \\
k_2 + 7 = 10 \\
k_2 = 3
\]

Now find the steady state error. Note that the method we used in (a) had \(u = K(r - y)\). This time, \(u = r - K_1 x\). The difference is that \(r\) does not get multiplied by \(K_1\). We need to re-derive the steady state error relationship.

\[
\dot{x} = (A_1 - B_1 K_1)x + B_1 r
\]

Note there’s no \(C_1\), because we’re doing full state feedback. Assume that \(\dot{x} = 0\) in steady state.

\[
0 = (A_1 - B_1 K_1)x_{ss} + B_1 r_{ss} \\
x_{ss} = -(A_1 - B_1 K_1)^{-1} B_1 r_{ss} \\
e_{ss} = r_{ss} - C_1 x_{ss} \\
= \left( I + C_1(A_1 - B_1 K_1)^{-1} B_1 \right) r_{ss}
\]

Now we can evaluate \(e_{ss}\).

\[
A_1 - B_1 K_1 = \begin{bmatrix} 0 & 1 \\ -12 & -7 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 13 & 3 \end{bmatrix} \\
= \begin{bmatrix} 0 & 1 \\ -12 & -7 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 13 & 3 \end{bmatrix} \\
= \begin{bmatrix} 0 & 1 \\ -25 & -10 \end{bmatrix}
\]
(actually, we could have jumped right to this, since \(A_1, B_1\) are in controllable canonical form, so \(A_1 - B_1K_1\) will have the coefficients of the characteristic equation in the bottom row.)

\[
(A_1 - B_1K_1)^{-1} = \frac{1}{25} \begin{bmatrix} -10 & -1 \\ 25 & 0 \end{bmatrix}
\]

\[
I + C_1(A_1 - B_1K_1)^{-1}B_1 = 1 + \begin{bmatrix} 10 & 0 \end{bmatrix} \frac{1}{25} \begin{bmatrix} -10 & -1 \\ 25 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\]

\[
= 1 - \frac{2}{5}
\]

\[
e_{ss} = \frac{3}{5}r_{ss} = \frac{3}{5}
\]

For parts c) and d) below, we want to add an integral controller to the system (1) to get zero steady state error, but still have state feedback to allow arbitrary pole placement.

**c)** Let \(u = -K_2x + k_ew \int (r - Cx)\,d\tau\). Write the augmented state and output equations for this system with integral control, for example, \(\dot{z} = A_2z + B_2r\), where \(A_2\) is \(3 \times 3\). Find gains \(K_2, k_e\) such that the system has closed-loop poles at \(s = -4, -5, -10\), and evaluate the steady state error for a step input \(r(t)\).

We are adding a new state variable to represent the integrator. To avoid confusing it with the original \(x\) state variables, we’ll call it \(w\).

\[
z = \begin{bmatrix} w \\ x_1 \\ x_2 \end{bmatrix}
\]

Manipulate the definition of \(u\) to get dynamics for \(w\).

\[
u = -K_2x + k_ew \int (r - Cx)\,d\tau
\]

\[
u = -K_2x + k_e w
\]

\[
w = \int (r - Cx)\,d\tau
\]

\[
\dot{w} = (r - Cx)
\]

Notice that the new form, \(\dot{z} = A_2z + B_2r\), doesn’t have \(u\). We need to make \(A_2\) capture the “inside loop” of the full state feedback. Manipulate the original dynamics.

\[
\dot{x} = A_1x + B_1u
\]

\[
\dot{x} = A_1x + B_1(-K_2x + k_e w)
\]

\[
\dot{x} = (A_1 - B_1K_2)x + B_1k_e w
\]
Now we can write the state and output equations.

\[
\dot{z} = A_2 z + B_2 r \\
y = C_2 z
\]

\[
\dot{z} = \begin{bmatrix} 0 & -C_1 \\ B_1 k_e & A_1 - B_1 K_2 \end{bmatrix} \begin{bmatrix} w \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r \\
y = \begin{bmatrix} 0 & C_1 \end{bmatrix} \begin{bmatrix} w \\ x \end{bmatrix}
\]

\[
\dot{z} = \begin{bmatrix} 0 & -10 & 0 \\ 0 & 0 & 1 \\ k_e & -12 - k_1 & -7 - k_2 \end{bmatrix} \begin{bmatrix} w \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r \\
y = \begin{bmatrix} 0 & 10 & 0 \end{bmatrix} \begin{bmatrix} w \\ x_1 \\ x_2 \end{bmatrix}
\]

To find the gains \( K_2, k_e \), find the characteristic equation for \( A_2 \), and make it equal \((s + 4)(s + 5)(s + 10)\):

\[
\Delta(s) = \begin{vmatrix} -s & -10 & 0 \\ 0 & -s & 1 \\ k_e & -12 - k_1 & -s - 7 - k_2 \end{vmatrix}
\]

\[
= -s \begin{vmatrix} -s & 1 \\ -12 - k_1 & -s - 7 - k_2 \end{vmatrix} - 10 \begin{vmatrix} 1 \\ -s - 7 - k_2 \\ k_e \end{vmatrix} + 0 \begin{vmatrix} 0 \\ k_e \\ -12 - k_1 \end{vmatrix}
\]

\[
= -s(s(s + 7 + k_2) + 12 + k_1) - 10(k_e)
\]

\[
= -s^3 - (7 + k_2)s^2 - (12 + k_1)s - 10k_e
\]

We can flip the sign of \( \Delta(s) \) for convenience, since we only care about its roots

\[
= s^3 + (7 + k_2)s^2 + (12 + k_1)s + 10k_e
\]

\[
= (s + 4)(s + 5)(s + 10) = s^3 + 19s^2 + 110s + 200
\]

\[
7 + k_2 = 19 \quad \Rightarrow \quad k_2 = 12
\]

\[
12 + k_1 = 110 \quad \Rightarrow \quad k_1 = 98
\]

\[
10k_e = 200 \quad \Rightarrow \quad k_e = 20
\]

Now we can evaluate the steady state error.

\[
\dot{z} = A_2 z + B_2 r \\
0 = A_2 z_{ss} + B_2 r_{ss}
\]

\[
z_{ss} = -A_2^{-1} B_2 r_{ss}
\]

\[
y_{ss} = -C_2 A_2^{-1} B_2 r_{ss}
\]

\[
e_{ss} = r_{ss} - y_{ss}
\]

\[
= \left( I + C_2 A_2^{-1} B_2 \right) r_{ss}
\]

\[
= \left( 1 + \begin{bmatrix} 0 & 10 & 0 \\ 0 & 0 & 1 \\ 20 & -110 & -19 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}
\]

\[
= 0
\]
d) Augment the system matrix $A_1$ to include an integrator (keeping the phase variable form), and then choose $u = K_3(r \cdot [1 \ 0 \ 0]^T - x)$. Write the augmented state and output equations for this system for example, $\dot{z} = A_3 z + B_3 r$, where $A_3$ is $3 \times 3$. Find gains $K_3$ such that the system has closed-loop poles at $s = -4, -5, -10$, and evaluate the steady state error for a step input $r(t)$.

Here is the signal flow diagram of the original system. Notice that we start in phase variable form.

There is only one way to add an integrator to this system that keeps the phase variable form and that does not modify the dynamics of the original system: after $x_1$.

Notice that this new integrator doesn’t actually affect anything (both gains leading away from $x_{\text{INT}}$ are zero). That’s OK; when we do full state feedback, this new state will start affecting the rest of the system.

We can write the pre-state-feedback system equations based on this diagram:

$$ z = \begin{bmatrix} x_{\text{INT}} \\ x_1 \\ x_2 \end{bmatrix} $$

$$ \dot{z} = A_3' z + B_3' u = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -12 & -7 \end{bmatrix} z + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u $$

$$ y = C_3 z = \begin{bmatrix} 0 & 10 & 0 \end{bmatrix} z $$

Next, we have to bring the “inside loop” into $A_3$, like we did in part (c). To do this, we write
out the dynamics, then get rid of \( u \) by substitution.

\[
\dot{z} = A_3'z + B_3'K_3(r \cdot [1 \ 0 \ 0]^T - x)
\]

\[
= (A_3' - B_3'K_3)z + \begin{bmatrix} 0 & k_1 & k_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} r
\]

\[
y = \begin{bmatrix} 0 & 10 & 0 \end{bmatrix} z
\]

Now we have \( A_3, B_3, \) and \( C_3 \). Next, we need to find the \( K_3 \) values. This time it’s easier, because \( A_3, B_3 \) are in phase variable form. We want the characteristic equation to equal \( (s + 4)(s + 5)(s + 10) = s^3 + 19s^2 + 110s + 200 \), so the bottom row of \( A_3 \) must be \((-200, -110, -19)\).

- \(-k_1 = -200 \Rightarrow k_1 = 200\)
- \(-12 - k_2 = 1 - 10 \Rightarrow k_2 = 98\)
- \(-7 - k_3 = -19 \Rightarrow k_3 = 12\)

Finally, we evaluate the steady state error. Because the system in (c) and (d) have the same form: \( \dot{z} = A_3z + B_3r \); we can re-use the derivation of the steady state error from (c).

\[
e_{ss} = \left( I + C_3A_3^{-1}B_3 \right) r_{ss}
\]

\[
= 1 + \begin{bmatrix} 0 & 10 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -200 & -110 & -19 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 200 \end{bmatrix} 1
\]

\[
= 1
\]

What’s going on here? The open loop system (defined by \( A_3', B_3' \)) clearly has a pole at the origin (one of the elements of the bottom row of \( A_3' \) is zero). Doesn’t that mean it’s a Type 1 system, and therefore zero steady state error when driven by a unit step?

There are a couple of ways of looking at this. As mentioned earlier, that the new integrator isn’t actually doing anything in the open loop form. It’s integrating \( x_1 \), but the result of the integration doesn’t affect the other states or the output.

Another way to look at it is to convert the open loop state space system into a transfer function. (See section 3.6 of Nise). When you do this, you find that the TF representation of the open loop system doesn’t have a pole at the origin.

2 Linear Quadratic Regulator (35 pts)
Consider a pair of cars in a “platoon” regulated to a stopping position, as illustrated above. The dynamics of car 1 are \(\ddot{x}_1 = u_1\) and car 2 has a plant model \(\dot{x}_3 = 0.5u_2\) where \(u_1\) and \(u_2\) are the car’s thrust due to engine and braking. (The offset of \(x_1\) from car 1’s rear bumper prevents the cars from colliding when \(x_3 = x_1\).) The outputs of the system are \(y_1 = x_1\) and \(y_2 = x_3 - x_1\). Note that if \(y_2 > 0\) then car 2 has intruded on the safety zone of car 1.

Initial conditions are car 1 at -100 m, 20 m/s, and car 2 at -102 m, 25 m/s.

a) Write the system equations in state space form.

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
\dot{x}_4
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix} +
\begin{bmatrix}
0 & 0 \\
1 & 0 \\
0 & 0 \\
0 & 0.5
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
y_1 \\
y_2
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix}
\]

\[
x(0) =
\begin{bmatrix}
-100 \\
20 \\
-102 \\
25
\end{bmatrix}
\]

b) Use the LQR method (from Matlab function `lqr(sys,Q,R)`, with \(Q=\text{diag}([2.5,10,2.5,10])\) and \(R=\text{diag}([1,1])\) to find an optimal \(K\) for the state feedback control \(u = -K_b x\). Plot \(x(t)\) and \(u(t)\) for the given initial condition (Matlab `initial`) and state feedback with gain \(K_b\).

<table>
<thead>
<tr>
<th>Time (seconds)</th>
<th>x(t)</th>
<th>u(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-100</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>-90</td>
<td>1.58</td>
</tr>
<tr>
<td>2</td>
<td>-80</td>
<td>3.62</td>
</tr>
<tr>
<td>3</td>
<td>-70</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-60</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>-50</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>-40</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>-30</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>-20</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>-10</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
K_b = \begin{bmatrix}
1.5811 & 3.6280 & 0 & 0 \\
0 & 0 & 1.5811 & 4.0404
\end{bmatrix}
\]

Notice that the safety zone is violated by about 5 cm at \(t = 7\) s.

c) In part b) both cars get within 1 m of the stop sign at a speed less than 1 m/s at 10 seconds. However, car 2 lags more than 4 meters behind car 1 during the stopping trajectory. Find a new
cost function $Q = \text{diag}([q_1, q_2, q_3, q_4])$ which keeps car 2 within 2 m of car 1 for the whole trajectory, while maintaining $y_2 < 0$ to prevent a collision, and distance from stop sign less than 1 m at 10 sec with velocity less than 1 m/s. (Suggestion: let $q_1 = 2.5$ and $q_2 = 10$ and search for new $q_3$ and $q_4$.) Plot $x(t)$ and $u(t)$ for the given initial condition (Matlab initial) and state feedback with new gain $K_c$.

There are many solutions that satisfy the constraints. The gray region in the following graph shows $q_3, q_4$ values that work.

Taking $Q=\text{diag}([2.5 10 10 40])$ gives the following response:

$$K_c = \begin{bmatrix} 1.5811 & 3.6280 & 0 & 0 \\ 0 & 0 & 3.1623 & 7.2560 \end{bmatrix}$$

d) Find the solution to the Riccati equation $P$ using Matlab function `care(A,B,Q,R)` and estimate the cost $J = (x^T P x)(0)$ for each of b) and c).
\[
\begin{bmatrix}
5.7363 & 1.5811 & 0 & 0 \\
1.5811 & 3.6280 & 0 & 0 \\
0 & 0 & 6.3884 & 3.1623 \\
0 & 0 & 3.1623 & 8.0807
\end{bmatrix}
\]

\[
x^T(0)P_b x(0) = 107880
\]

\[
\begin{bmatrix}
5.7363 & 1.5811 & 0 & 0 \\
1.5811 & 3.6280 & 0 & 0 \\
0 & 0 & 22.9454 & 6.3246 \\
0 & 0 & 6.3246 & 14.5119
\end{bmatrix}
\]

\[
x^T(0)P_c x(0) = 268030
\]

Answers for \(P_c\) will vary depending on the choice of \(Q\). Notice that the top left corner of \(P_c\) is the same, since \(q_1\) and \(q_2\) were not changed.

e) Briefly compare the tradeoffs between control effort and time response between the two cases.

Notice that the time response, control effort, and \(K\) entries associated with the first car are identical in (b) and (c). This will be true as long as the suggestion to keep \(q_1\) and \(q_2\) the same is followed.

Car 2’s response, control effort, and \(K\) entries are different between the two cases. The control effort used in (c) is bigger in magnitude than in (b). Correspondingly, the time response of \(y_2\) (the distance between the two cars) is much different as well: in (c), it seems to be decaying to zero much faster than in (b). \(y_2\) in (b) shows evidence of several modes with similar time constants at work (the back-and-forth motion). In (c), \(y_2\) seems to be dominated by a single, faster mode.

In general, we get better performance (faster convergence to zero) in (c) compared to (b), at the cost of greater control effort.

3 Discrete Time Control (35 pts)

*Given the following continuous time (CT) system*

\[
\dot{x} = Ax + Bu = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -12 & -5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t), \quad y = [2 \ 1 \ 0] x
\]

*the corresponding discrete time (DT) system is*

\[
x[n+1] = Gx[n] + Hu[n], \quad y[n] = Cx[n]
\]

*which can be found using the Matlab function c2d(sys,T,’zoh’).*

* a) With initial condition \(x_0 = [1 \ 0 \ 0]'\), plot the ZIR using Matlab function initial() for the CT system and the DT system (with \(T = 0.4\) sec).*

\[
\begin{align*}
A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -12 & -5 \end{bmatrix} \\
B &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
C &= [2 \ 1 \ 0] \\
D &= 0 \\
csys = ss(A,B,C,D);
\end{align*}
\]
Continuous time response:

Zero-Order Hold Discrete time response:

\[ u = k(r - y) \]

b) For output feedback control \( u = k(r - y) \), sketch the root locus for the equivalent transfer function for the continuous time (CT) system.

\[ [\text{NUM, DEN}] = \text{ss2tf}(A, B, C, D) \]

\[ \text{NUM} = \]
\[
\begin{array}{cccc}
0 & -0.0000 & 1.0000 & 2.0000 \\
\end{array}
\]

DEN =
\[
\begin{array}{cccc}
1.0000 & 5.0000 & 12.0000 & 8.0000 \\
\end{array}
\]

Root locus for TF:
\[
\frac{s + 2}{s^3 + 5s^2 + 12s + 8} = \frac{s + 2}{(s + 1)(s + 2 + 2j)(s + 2 - 2j)}
\]

c) Determine the closed loop pole locations for the CT system for \( k = 5 \) and plot the closed-loop step response using Matlab.

\[
\begin{align*}
\Theta_i &= \pi - \left( \pi - \tan^{-1} \left( \frac{2}{1} \right) \right) \\
&\approx 63^\circ
\end{align*}
\]

\[
\begin{align*}
[x, p, k] &= \text{zpkdata}(\text{feedback}(5*\text{ss}(A,B,C,D),1)); \\
\{p\} &= \text{p(1)} \\
\text{ans} &= \\
-1.7285 + 2.9459i \\
-1.7285 - 2.9459i \\
-1.5430 \\
\text{step}(\text{feedback}(5*\text{ss}(A,B,C,D),1),[0:.1:4]);
\end{align*}
\]

Continuous step response for system with feedback:
d) The closed loop DT system has state equation
\[ x[n + 1] = (G - kHC)x[n] + kHr[n], \quad y[n] = Cx[n] \]
(which can be found using the Matlab feedback function). Using Matlab, determine the closed loop pole locations for the DT system for \( k = 5 \) and sampling period \( T = 0.4 \) sec and plot the step response.

\[
\begin{align*}
\gg [z,p,k] &= \text{zpkdata(feedback(5*dtsys,1))}; \\
\gg p(1) \\
\end{align*}
\]
\[
\text{ans} = \\
0.5209 \\
0.2579 + 0.5838i \\
0.2579 - 0.5838i
\]

\[
\gg \text{step(feedback(5*dtsys,1),[0:.4:4])};
\]

Discrete step response for system with feedback:
e) Use Matlab (iteratively if necessary) to find a sampling period $T$ which gives a closed-loop step response that is “reasonably close” to the CT closed-loop step response. Determine closed-loop pole locations, and plot the DT step response.

```matlab
>> step(feedback(5*c2d(ctsys,0.4,'zoh'),1))
>> step(feedback(5*c2d(ctsys,0.3,'zoh'),1))
>> step(feedback(5*c2d(ctsys,0.2,'zoh'),1))
>> step(feedback(5*c2d(ctsys,0.1,'zoh'),1),[0:.1:4])
>> [z,p,k] = zpkdata(feedback(5*c2d(ctsys,0.1,'zoh'),1));
>> p{1}

ans =

0.8553
0.8159 + 0.2510i
0.8159 - 0.2510i
```

With sampling period of 0.1s, there is at least 8 samples for the first “hump” and the tail settles quickly to the equilibrium value without jittering around.

Discrete step response for system with feedback (Sampling Period changed to $T = 0.1s$)
f) Briefly explain why the CT and DT ZIR responses from a) above are reasonably close, but the closed loop responses from c) and d) (with $T = 0.4\text{ sec}$) do not agree at all. (Hint, consider $e^{AT}$.)

We know there will be some error in the open-loop responses due to discretization error. Intuitively, controlling the system based on an erroneous output would cause even more error in the responses.

Let's look at a quantitative analysis of the differences:

For the discrete system:

$$G = e^{A\cdot0.4} = \begin{bmatrix} 0.9491 & 0.3180 & 0.0392 \\ -0.3138 & 0.4785 & 0.1219 \\ -0.9755 & -1.7771 & -0.1312 \end{bmatrix}$$

and, in feedback,

$$G - 5HC = \begin{bmatrix} 0.8856 & 0.2863 & 0.0392 \\ -0.7060 & 0.2824 & 0.1219 \\ -2.1945 & -2.3866 & -0.1312 \end{bmatrix}$$

Suppose we discretized the closed loop continuous system instead so that the sampler is outside the controller:

$$G_{cl} = e^{(A - 5BC)\cdot0.4} = \begin{bmatrix} 0.8899 & 0.2836 & 0.0366 \\ -0.6585 & 0.2680 & 0.1006 \\ -1.8117 & -2.3696 & -0.2353 \end{bmatrix}$$

Note how these two matrices are similar, but there are a few entries that are off by a few tenths. In the first case, the controller is acting on observations containing discretization error, whereas in the second case, the controller is acting continuously, and then the system is discretized. The difference in the matrices show that a controller can, in effect, amplify discretization error when modelling a continuous system.