Homework #8: Closed-Loop Estimators and Dynamic Modelling in MATLAB

A. Closed-Loop Estimator Design

Consider the following linear dynamic system:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 3 & 2 \\ -5 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned}$$

1. Design a closed-loop estimator for this system for which the eigenvalues are stable and have time constant ($\tau = 1/\zeta\omega_n$) three times faster than the open-loop poles. Give both:

   – the closed-loop estimator poles

   – the estimator gain matrix L

<u>Solution</u>

To find the open-loop poles, we need to find the eigenvalues of the system. Hence, we can form

$$\begin{aligned} &\det(\lambda I - A) = 0 \\ &\to \det\left(\begin{bmatrix} \lambda - 3 & -2 \\ 5 & \lambda - 2 \end{bmatrix}\right) = 0 \\ &\to (\lambda - 3)(\lambda - 2) - (-2)(5) = 0 \\ &\to \lambda^2 - 5\lambda + 16 = 0 \quad (1) \\ &\to \lambda = \frac{5 \pm \sqrt{5^2 - 4 \times 1 \times 16}}{2 \times 1} = \frac{5 \pm \sqrt{25 - 64}}{2} \\ &\to \begin{cases} \lambda_1 = 2.5000 + \ 3.1225i \\ \lambda_2 = 2.5000 \ - \ 3.1225i \end{cases} \end{aligned}$$

Note that the characteristic equation is of the form

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0 \quad (2)$$

$$\lambda = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

Now comparing (1) and (2) we can see that for open-loop system (1) we have

$$\begin{cases} 2\zeta\omega_n = -5 \to \zeta\omega_n = -2.5 \\ \omega_n^2 = 16 \to \omega_n = \sqrt{16} = 4 \end{cases}$$

Any time "speed" is mentioned with regards to a pole, it's how "fast" or "slow" the pole is, which means how far or close the pole is to the imaginary axis (respectively). A pole that is very close to the imaginary axis will take a long time to settle; one that is very far from the imaginary axis will settle very quickly.

This is how you get into the "dominant pole(s)" or "dominant response" - the "faster" dynamics (poles) settle quickly and the "slower" dynamics (poles) are left, and so at the medium- to long-term, the only observable dynamics are those of the "slower" poles.

The speed is, therefore, represented by a pole's distance from the imaginary axis ($-\xi\omega_n$).

Now to have the closed-loop poles three times faster than the open loop we need to have their $\zeta\omega_n$ to be three times larger. So we would have

$-2.5 \times 3 = -7.5$ as the new $\zeta\omega_n$ for the new characteristic equation we need for the estimator.

So we get the closed-loop estimator poles as

$$\rightarrow \boxed{\begin{cases} \lambda_1 = 7.5000 + \ 3.1225i \\ \lambda_2 = 7.5000 \ - \ 3.1225i \end{cases}}$$

Forming the estimator's characteristic equation with these eigenvalues, we get

$$\big(\lambda - (7.5 + 2.1225i)\big)\big(\lambda - (7.5 - 3.1225i)\big) = 0$$
$$\rightarrow \lambda^2 - 15\lambda + 66 = 0 \ (3)$$

The characteristic equation for the closed-loop estimator with gain L is

$$\det(\lambda I - A + LC) \ (4)$$

Now we need to form the closed-loop estimator equation and find the gains that give the obtained poles for the estimator, which can be done by equaling (3) and (4)

$$\det(\lambda I - A + LC) = \lambda^2 - 15\lambda + 66$$

Where $L = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$

$$\rightarrow \det\left(\begin{bmatrix} \lambda - 3 & -2 \\ 5 & \lambda - 2 \end{bmatrix} + \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix}\right) = \det\left(\begin{bmatrix} \lambda - 3 & -2 \\ 5 & \lambda - 2 \end{bmatrix} + \begin{bmatrix} l_1 & 0 \\ l_2 & 0 \end{bmatrix}\right)$$
$$= \det\left(\begin{bmatrix} \lambda - 3 + l_1 & -2 \\ 5 + l_2 & \lambda - 2 \end{bmatrix}\right) = \lambda^2 + (l_1 - 5)\lambda + (2l_2 - 2l_1 + 16) = \lambda^2 - 15\lambda + 16$$
$$\rightarrow \begin{cases} 1 = 1 \\ l_1 - 5 = -15 \\ 2l_2 - 2l_1 + 16 = 66 \end{cases} \rightarrow \begin{cases} l_1 = -10 \\ 2l_2 = 2l_1 + 50 \end{cases} \rightarrow l_2 = \frac{1}{2}(-20 + 50) = 15$$
$$\rightarrow \boxed{L = \begin{bmatrix} -10 \\ 15 \end{bmatrix}}$$

Similarly, we could use the Matlab code to obtain the same results for L

```
A = [3 2; -5 2];
B = [0; 1];
C = [1, 0];
```

```
P = [7.5000+ 3.1225i 7.5000- 3.1225i];
L = (place(A',C',P)')
```

B. Optimal Estimators

This problem is roughly based on problem 2 of homework assignment 6 from MIT OCW 16.30.

Consider the system

$$\begin{aligned}\dot{\vec{x}} &= \begin{bmatrix}-11 & -10\\ 1 & 0\end{bmatrix}\vec{x} + \begin{bmatrix}1\\0\end{bmatrix}u + \begin{bmatrix}1\\0\end{bmatrix}w\\ y &= [1 \quad 8]\vec{x} + v\end{aligned}$$

where $R_{ww} = 1$ and $R_{vv} = \rho_e$.

The use of a good calculator or MATLAB/Octave is strongly recommended for this problem.

Design an estimator for this system using LQE. Use $\rho_e = 0.0025$. Determine the resulting value of the feedback gain L, and the closed-loop estimator pole locations. Which does this estimator design "trust" more: the model or the measurements?

Solution

we have

$A = \begin{bmatrix}-11 & -10\\ 1 & 0\end{bmatrix}$

$B = \begin{bmatrix}1\\0\end{bmatrix}$

$C = [1 \quad 8]$

$R_{ww} = 1$

$R_{vv} = \rho_e = 0.0025$

$0 = AQ + QA^T + BR_{ww}B^T - QC^TR_{vv}^{-1}CQ$

$$\rightarrow \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} = \begin{bmatrix}-11 & -10\\ 1 & 0\end{bmatrix}\begin{bmatrix}q_1 & q_3\\ q_3 & q_2\end{bmatrix} + \begin{bmatrix}q_1 & q_3\\ q_3 & q_2\end{bmatrix}\begin{bmatrix}-11 & 1\\ -10 & 0\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix}[1][1 \quad 0] - \begin{bmatrix}q_1 & q_3\\ q_3 & q_2\end{bmatrix}\begin{bmatrix}1\\8\end{bmatrix}\frac{1}{\rho_e}[1 \quad 8]\begin{bmatrix}q_1 & q_3\\ q_3 & q_2\end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} -11q_1 - 10q_3 & -11q_3 - 10q_2 \\ q_1 & q_3 \end{bmatrix} + \begin{bmatrix} -11q_1 - 10q_3 & q_1 \\ -11q_3 - 10q_2 & q_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$
$$- \frac{1}{\rho_e} \begin{bmatrix} q_1^2 + 16q_1q_3 + 64q_3^2 & 8q_1q_2 + q_1q_3 + 64q_2q_3 + 8q_3^2 \\ 8q_1q_2 + q_1q_3 + 64q_2q_3 + 8q_3^2 & 64q_2^2 + 16q_2q_3 + q_3^2 \end{bmatrix}$$

$$\rightarrow \begin{cases} 0 = -11q_1 - 10q_3 - 11q_1 - 10q_3 + 1 - \left(\frac{1}{\rho_e}\right) q_1^2 + 16q_1q_3 + 64q_3^2 \\ 0 = -11q_3 - 10q_2 + q_1 + 0 - \left(\frac{1}{\rho_e}\right) 8q_1q_2 + q_1q_3 + 64q_2q_3 + 8q_3^2 \\ 0 = 2q_3 - \left(\frac{1}{\rho_e}\right) 64q_2^2 + 16q_2q_3 + q_3^2 \end{cases}$$

$$\rightarrow \begin{cases} 0 = -22q_1 - 20q_3 + 1 - \left(\frac{1}{\rho_e}\right) q_1^2 + 16q_1q_3 + 64q_3^2 \\ 0 = -11q_3 - 10q_2 + q_1 - \left(\frac{1}{\rho_e}\right) 8q_1q_2 + q_1q_3 + 64q_2q_3 + 8q_3^2 \\ 0 = 2q_3 - \left(\frac{1}{\rho_e}\right) 64q_2^2 + 16q_2q_3 + q_3^2 \end{cases}$$

You can solve this using Matlab

```
clc; clear; close all;
A =[-11 -10; 1 0];
B = [1; 0];
C =[1 8];
D = 0;

rho = 0.0025;
R_ww = 1;
R_vv = rho;

syms q_1 q_2 q_3 rho

Q = [q_1 q_3; q_3 q_2];

f = A*Q+Q*A'+B*R_ww*B'-Q*C'*R_vv^(-1)*C*Q;

sol = solve(f);

double(sol.q_1)
double(sol.q_2)
double(sol.q_3)
```

solving this, gives 4 solutions for each of q_1, q_2, q_3 as follows

$$q_1 = \begin{cases} 0.0272 \\ -0.1292 \\ -1.5599 \\ -11.9239 \end{cases} \qquad q_2 = \begin{cases} 0.0001 \\ 0.0001 \\ -0.0329 \\ -0.1639 \end{cases} \qquad q_3 = \begin{cases} 0.0005 \\ 0.0040 \\ 0.2296 \\ 1.3946 \end{cases}$$

$$Q = \begin{bmatrix} q_1 & q_3 \\ q_3 & q_2 \end{bmatrix}$$

$$\rightarrow L = QC^T R_{vv}^{-1} = \begin{bmatrix} q_1 & q_3 \\ q_3 & q_2 \end{bmatrix} \begin{bmatrix} 1 \\ 8 \end{bmatrix} \frac{1}{\rho_e}$$

Now we have many pairs of answers for which we have to loop through and find the proper Q matrix and proper L gains. To find the poles for each case we can use

$$\det(\lambda I - A + LC) = 0$$

Using the following code, we can find all desired values through this loop

```
for n=1:4

    q1=double(sol.q_1(n));
    q2=double(sol.q_2(n));
    q3=double(sol.q_3(n));

    Q=[q1 q3; q3 q2]
    L=Q*C'*R_vv^(-1)

    syms lambda
    lambda_cl = double(solve(det(lambda*eye(2)-A+L*C)))

end
```

which gives us the following 4 Q matrices and L and λ_{cl} vectors:

Set	Q	L	λ_{cl}
1	0.0272 0.0005 0.0005 0.0001	12.5210 0.6429	-21.0472 -7.6168
2	-0.1292 0.0040 0.0040 0.0001	-38.7910 1.7951	-7.6168 21.0472
3	-1.5599 0.2296 0.2296 -0.0329	110.8620 -13.5540	-21.0472 7.6168
4	-11.9239 1.3946 1.3946 -0.1639	-306.8778 33.4017	7.6168 21.0472

We can see where $Q > R$ where $R = diag(R_{vv})$, the estimator is penalizing the states more heavily, meaning it is relying on the measurements more. For the case where $Q < R$ the opposite holds.

We can see that only the poles of the estimator for the first set of answers are in the left-hand plane (LHP) for the imaginary plane. This means that only the first set would have a stable estimation and other sets would not converge to the actual system states.

C. Dynamic Modelling in MATLAB: Dynamics of a two DOF spring-mass-damper system

Consider the system shown in Figure 1, with mass $m = 1$ kg, $k_1 = k_2 = k_3 = k_4 = 1$ N/m, and $c = 1$ N $\cdot$ s/m. Also $L = 1.0$ m. This is a "top-down" view of the system so that gravity acts normal to the page. The mass slides on a frictionless surface so that there is no effect of gravity on the dynamics in the x and y directions.

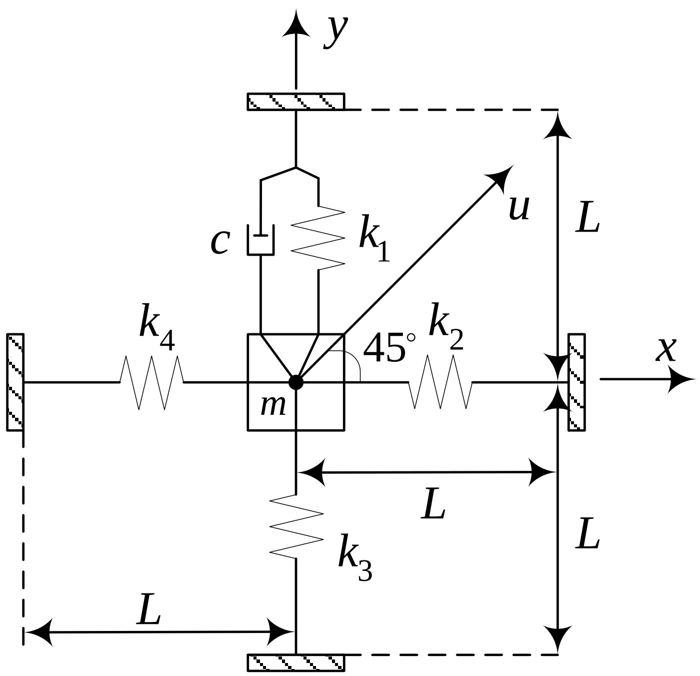

Figure 1 Two DOF spring-mass-damper system

The nonlinear equations of motion for this system are:

$$
\begin{aligned}
m\ddot{x} - k_2 &\left[(L^2 - 2xL + x^2 + y^2)^{\frac{1}{2}} - L\right] \\
&+ k_4 \left[(L^2 + 2xL + x^2 + y^2)^{\frac{1}{2}} - L\right] = U\cos\frac{\pi}{4}
\end{aligned}
$$

$$
\begin{aligned}
m\ddot{y} - c &\left[(L^2 - 2yL + x^2 + y^2)^{-\frac{1}{2}}\right] [(y - L)\dot{y} + x\dot{x}] \\
&- k_1 \left[(L^2 - 2yL + x^2 + y^2)^{\frac{1}{2}} - L\right]
\end{aligned}
$$

$$+k_3\left[(L^2+2yL+x^2+y^2)^{\frac{1}{2}}-L\right]=U\sin\frac{\pi}{4}$$

Let the states be

$$\begin{aligned} x_1 &= x \\ x_2 &= y \\ x_3 &= \dot{x} \Rightarrow \dot{x}_1 = x_3 \\ x_4 &= \dot{y} \Rightarrow \dot{x}_2 = x_4 \end{aligned}$$

so that

$$\vec{x}=[x_1 \quad x_2 \quad x_3 \quad x_4]^T$$

Then the equations of motion in terms of the state variables are

$$\begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \dot{x}_3 &= \frac{k_2}{m}\left[(L^2-2x_1L+x_1^2+x_2^2)^{1/2}-L\right]-\frac{k_4}{m}\left[(L^2+2x_1L+x_1^2+x_2^2)^{1/2}-L\right]+\frac{U}{m}\cos\frac{\pi}{4} \\ \dot{x}_4 &= \frac{c}{m}\left[(L^2-2x_2L+x_1^2+x_2^2)^{-1/2}\right]\left[(x_2-L)x_4+x_1x_3\right] \\ &\quad +\frac{k_1}{m}\left[(L^2-2x_2L+x_1^2+x_2^2)^{1/2}-L\right] \\ &\quad -k_3\left[(L^2+2x_2L+x_1^2+x_2^2)^{1/2}-L\right]+\frac{U}{m}\sin\frac{\pi}{4} \end{aligned}$$

1. Where does each term in the equations of motion come from?

2. Convince yourself that the linearized equations of motion about the equilibrium $\vec{x}=0$ are

$$\dot{\vec{x}}=A\vec{x}+Bu$$

where

$$A=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_2+k_4}{m} & 0 & 0 & 0 \\ 0 & -\frac{k_1+k_3}{m} & 0 & -\frac{c}{m} \end{bmatrix} \quad \text{and} \quad B=\begin{bmatrix} 0 \\ 0 \\ \frac{1}{m}\cos\frac{\pi}{4} \\ \frac{1}{m}\sin\frac{\pi}{4} \end{bmatrix}$$

3. The output is the position of the mass along the x and y axes:

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Check the stability, controllability, observability, stabilizability, and detectability of this system.

4. Check the response of the open-loop dynamics to an initial condition of $x = 1$ m, $y = 1$ m.

5. Use pole placement to design a full-state feedback controller with the following features:

   - Poles (eigenvalues) corresponding to decaying modes are unchanged

   - Poles (eigenvalues) corresponding to non-decaying modes do not have altered **frequencies**, but have real parts the same as those of the decaying modes for the open-loop dynamics

6. Check the response of the closed-loop dynamics to both a step input and the same initial condition as specified in part 4.

7. Choose new poles for the controller which give a response time for both outputs of ~ 5 s.

<u>Solution</u>

Use the following Matlab code to obtain results for each section

```
%% symbolic
%% part 2
clc
syms x_1 x_2 x_3 x_4 m L c U k_1 k_2 k_3 k_4
f1 = x_3;
f2 = x_4;
f3 = (k_2/m) * ( (L^2 - (2*x_1*L) + x_1^2 + x_2^2)^(1/2) - L) - ...
      (k_4/m) * ( (L^2 + (2*x_1*L) + x_1^2 + x_2^2)^(1/2) - L) + ...
      (U/m)  * cos(pi/4);
f4 = (c/m)*((L^2-2*x_2*L+x_1^2+x_2^2)^(-1/2))*((x_2-L)*x_4+x_1*x_3) + ...
      (k_1/m)*((L^2-2*x_2*L+x_1^2+x_2^2)^(1/2)-L) - ...
       k_3*((L^2+2*x_2*L+x_1^2+x_2^2)^(1/2)-L)+(U/m)*sin(pi/4);

A1 = [diff(f1,x_1) diff(f1,x_2) diff(f1,x_3) diff(f1,x_4)];
A2 = [diff(f2,x_1) diff(f2,x_2) diff(f2,x_3) diff(f2,x_4)];
A3 = [diff(f3,x_1) diff(f3,x_2) diff(f3,x_3) diff(f3,x_4)];
A4 = [diff(f4,x_1) diff(f4,x_2) diff(f4,x_3) diff(f4,x_4)];
A = [A1; A2; A3; A4];
assume(L>0)
simplify(subs(A,[x_1,x_2,x_3,x_4],[0,0,0,0]))
%% numerical
clear all;
m = 1;
k_1 = 1; k_2 =1;  k_3 =1; k_4 = 1;
c = 1;
L = 1;
A=[0,0,1,0;
    0,0,0,1;
    -(k_2+k_4)/m,0,0,0;
    0,-(k_1+k_3)/m,0,-c/m];
```

```
B=[0;0;(1/m)*cos(pi/4);(1/m)*sin(pi/4)];
C = [1,0,0,0; 0,1,0,0];
%% part 3
[T,Lambda,Tinv] = eig(A);
Lambda
Mo = [B A*B A^2*B A^3*B];
rank(Mo)
Mc = [C; C*A; C*A^2; C*A^3];
rank (Mc)

%% part 4
S = ss(A,B,C,0);
[y,t,x] = initial(S,[1 1 0 0]',20);
figure(2)
clf
plot(t,y(:,1),'r',t,y(:,2),'b')
legend('x_1=x','x_2=y')
title('initial response to open-loop system')
%% part 5
lambda = diag(Lambda)
P = [real(lambda(3:4))+i*imag(lambda(1:2)) lambda(3:4)];
K = acker(A,B,P)

%% part 6
S2 = ss(A-B*K,B,C,0);
[y2,t2,x2] = initial(S2,[1 1 0 0]',20);
figure(3)
clf
plot(t2,y2(:,1),'r',t2,y2(:,2),'b')
legend('x_1=x','x_2=y')
title('initial response to closed-loop system')
[y3,t3,x3] = step(S2,20);
figure(4)
clf
plot(t3,y3(:,1),'r',t3,y3(:,2),'b')
legend('x_1=x','x_2=y')
title('step response to closed-loop system')

%% part 7
P2 = [-2+1i,-2-1i,-2+2i,-2-2i];
K2 = acker(A,B,P2);
S3 = ss(A-B*K2,B,C,0);
[y4,t4,x4] = initial(S3,[1 1 0 0]',20);
figure(5)
clf
plot(t4,y4(:,1),'r',t4,y4(:,2),'b')
legend('x_1=x','x_2=y')
title('initial response to closed-loop system 5 sec settling')
[y5,t5,x5] = step(S3,20);
figure(6)
clf
plot(t5,y5(:,1),'r',t5,y5(:,2),'b')
legend('x_1=x','x_2=y')
title('step response to closed-loop system 5 sec settling')
```