Table of Contents


```
clear all;close all;clc
```

Part 1

```
%Define given matrices for state space realization of system
A = [0 0 11000 -11000;0 0 0 100000;-0.0066667 0 -8 8; 0.0001 -0.0001
 0.12 -3.12];
B = [0 0; 0 -100000;0.0066667 0;-0.0001 3];
C = [-0.0066667 0 -8 8;0.0001 -0.0001 0.12 -3.12];
D = [0.0066667 0;-0.0001 3];

[V,Deig] = eig(A);

%Eigenspaces don't exist in real plane becuase all eigenvectors are
 complex, so can't
%span subspace R4 that system lives in
% for i = 1:4
%     ES(:,i) = null(A - D(i,i)*eye(4,4));
% end
% set up state space system
%Modal space is the span of the linear combinations of real and
 imaginary
%parts of eigenvecs
sys = ss(A,B,C,D);
t = linspace(0,10,500);
u = zeros(numel(t),2);
T = [real(V(:,1)) imag(V(:,1)) real(V(:,3)) imag(V(:,3))];
x01 = T*(real(V(:,1)) + imag(V(:,1)));
[y1,~,xM1] = lsim(sys,u,t,x01);
% figure
% sgtitle('Timed Response for Initial Conditions, Mode 1')
% subplot(2,2,1)
% plot(t,xM1(:,1),'Linewidth',4)
% grid on
% grid minor
```

```
% xlabel('Time (s)')
% ylabel('Force (N)')
% subplot(2,2,2)
% plot(t,xM1(:,2))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('Force (N)')
% subplot(2,2,3)
% plot(t,xM1(:,3))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('Velocity (m/s)')
% subplot(2,2,4)
% plot(t,xM1(:,4))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('Velocity (m/s)')
%
% fig = gcf;    %or one particular figure whose handle you already
 know, or 0 to affect all figures
% set( findall(fig, '-property', 'fontsize'), 'fontsize', 18)

x02 = T*(real(V(:,3)) + imag(V(:,3)));
[y2,~,xM2] = lsim(sys,u,t,x02);
% figure
% sgtitle('Timed Response For Initial Conditions, Mode 2')
% subplot(2,2,1)
% plot(t,xM2(:,1))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('Force (N)')
% subplot(2,2,2)
% plot(t,xM2(:,2))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('Force (N)')
% subplot(2,2,3)
% plot(t,xM2(:,3))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('Velocity (m/s)')
% subplot(2,2,4)
% plot(t,xM2(:,4))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('Velocity (m/s)')
%
```

```
% fig = gcf;    %or one particular figure whose handle you already
 know, or 0 to affect all figures
% set( findall(fig, '-property', 'fontsize'), 'fontsize', 18)
%
% figure
% sgtitle('System output response for mode 1 and mode 2')
% subplot(2,1,1)
% plot(t,y1(:,1))
% hold on
% plot(t,y2(:,1))
% xlabel('Time (s)')
% ylabel('a_d')
% legend('Mode 1','Mode 2')
% grid on
% grid minor
%
% subplot(2,1,2)
% plot(t,y1(:,2))
% hold on
% plot(t,y2(:,2))
% xlabel('Time (s)')
% ylabel('a_b')
% legend('Mode 1','Mode 2')
% grid on
% grid minor
```

Part 2

make P matrix for reachability

```
P = [B A*B A^2*B A^3*B];
%form orthonormal basis for subspace (P)
[Q,R] = qr(P); % Q results in orthonormal basis of reachable subspace
% compute Qr matrix necessary to find reachability grammian
t0 = 0;
t1 = 0.5;
tcont = linspace(t0,t1,500);
Qr = -(expm(-A*(t1-t0))*B*B'*expm(-A'*(t1-t0)) - B*B');
G = lyap(-A,Qr);
for i = 1:4
    zeta = -(Q(:,i));
    min_energy(i) = zeta'*inv(G)*zeta;
    for j = 1:length(t)
        opencontsig(:,j,i) = B'*expm(A'*-tcont(j))*inv(G)*zeta;
    end
end
```

Part 3

look at Bn = inv(V)*B to determine controllability

```
Bn = inv(V)*B;
```

```
% becuase this Bn matrix has no zero rows, our systme is completely
% controllable, so every plant mode can be changed by feedback
 control.
% Also can look at rank of reachability (P) -> rank 4, system is
 completely
% reachable -> completely controllable, all plant modes can be changed
 by
% feedback control
%Use real parts of first and third eigen values for starters
eigs = [real(Deig(1,1)) real(Deig(3,3)) -2 -5];
%only real negative part indicates decay with no oscillation, negative
 and
%less than -1 gives time constants shorter than 1 second via Tconst =
%-1/lambda_i
```

Part 4

```
K = place(A,B,eigs);
[Vdes,Ddes] = eig(A - B*K);

newsysA = (A - B*K);
newsys = ss(newsysA,B,C,D);
Tdes = cdf2rdf(Vdes,Ddes);
x0mat = Tdes*eye(4);

[y1des,~,xM1des] = lsim(newsys,u,t,x0mat(:,1));
[y2des,~,xM2des] = lsim(newsys,u,t,x0mat(:,2));
[y3des,~,xM3des] = lsim(newsys,u,t,x0mat(:,3));
[y4des,~,xM4des] = lsim(newsys,u,t,x0mat(:,4));
% create cell arrays of plot and subplots names
% yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
% plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
% xM = {xM1des xM2des xM3des xM4des};
% sptitles = {'Response to Unit Perturbation for f_{k_2}
(0)=1','Response to Unit Perturbation for f_{k_1}(0)=1','Response to
 Unit Perturbation for v_{m_2}(0)=1','Response to Unit Perturbation
 for v_{m_1}(0)=1'};
% for i = 1:4
%     figure
%     for j = 1:4
%         subplot(2,2,j)
%         plot(t,xM{i}(:,j))
%         yl = ylim;
%         ylim([min(-abs(yl)),max(abs(yl))])
%         xlabel('Time (s)')
%         ylabel(yax{j})
%         title(plottitles{j})
%         grid on
%         grid minor
%     end
%     sgtitle(sptitles{i});
%
% end
```

```
% comparison -> talk about eigenvalues resulting in expected
 performance of
% system, e.g. decay and no oscillations, etc.
ucon1 = -K*xM1des';
for i = 1:size(ucon1,2)
    contsig1(i) = norm(ucon1(:,i));
end
ucon2 = -K*xM2des';
for i = 1:size(ucon2,2)
    contsig2(i) = norm(ucon2(:,i));
end
ucon3 = -K*xM3des';
for i = 1:size(ucon3,2)
    contsig3(i) = norm(ucon3(:,i));
end
ucon4 = -K*xM4des';
for i = 1:size(ucon4,2)
    contsig4(i) = norm(ucon4(:,i));
end
contsig = [trapz(t,contsig1) trapz(t,contsig2) trapz(t,contsig3)
 trapz(t,contsig4)];
% figure
% plot(t,ucon1(1,:))
% hold on
% plot(t,ucon1(2,:))
% plot(tcont,opencontsig(1,:,1))
% plot(tcont,opencontsig(2,:,1))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('u')
% title('Minimum energy VS closed-loop control signals, 1st
 orthonormal basis vector direction')
% legend('f_a,closed-loop','v_w,closed-loop','f_a,min.
 energy','v_w,min. energy')
%
% figure
% plot(t,ucon2(1,:))
% hold on
% plot(t,ucon2(2,:))
% plot(tcont,opencontsig(1,:,2))
% plot(tcont,opencontsig(2,:,2))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('u')
% title('Minimum energy VS closed-loop control signals, 2nd
 orthonormal basis vector direction')
% legend('f_a,closed-loop','v_w,closed-loop','f_a,min.
 energy','v_w,min. energy')
%
% figure
% plot(t,ucon3(1,:))
% hold on
```

```
% plot(t,ucon3(2,:))
% plot(tcont,opencontsig(1,:,3))
% plot(tcont,opencontsig(2,:,3))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('u')
% title('Minimum energy VS closed-loop control signals, 3rd
 orthonormal basis vector direction')
% legend('f_a,closed-loop','v_w,closed-loop','f_a,min.
 energy','v_w,min. energy')
%
% figure
% % set(findall(gcf,'-property','FontSize'),'FontSize',12)
% plot(t,ucon4(1,:))
% hold on
% plot(t,ucon4(2,:))
% plot(tcont,opencontsig(1,:,4))
% plot(tcont,opencontsig(2,:,4))
% grid on
% grid minor
% xlabel('Time (s)')
% ylabel('u')
% title('Minimum energy VS closed-loop control signals, 4th
 orthonormal basis vector direction')
% legend('f_a,closed-loop','v_w,closed-loop','f_a,min.
 energy','v_w,min. energy')% figure
%
% fig = gcf;    %or one particular figure whose handle you already
 know, or 0 to affect all figures
% set( findall(fig, '-property', 'fontsize'), 'fontsize', 18)

% plot(t,contsig1)
% xlabel('Time (s)')
% ylabel('u')
% title('Control signal VS Time, 1st orthonormal basis vector
 direction perturbation')
% grid on
% grid minor
% figure
% plot(t,contsig2)
% xlabel('Time (s)')
% ylabel('u')
% title('Control signal VS Time, 2nd orthonormal basis vector
 direction perturbation')
% grid on
% grid minor
% figure
% plot(t,contsig3)
% xlabel('Time (s)')
% ylabel('u')
% title('Control signal VS Time, 3rd orthonormal basis vector
 direction perturbation')
% grid on
```

```
% grid minor
% figure
% plot(t,contsig4)
% xlabel('Time (s)')
% ylabel('u')
% title('Control signal VS Time, 4th orthonormal basis vector
 direction perturbation')
% grid on
% grid minor
```

Part 5

create gain matrix F s.t ref. inputs are accurately tracked by corresponding outputs at low freq.

```
F = inv(C*inv(-A+B*K)*B);
Bt = B*F;
Ct = C - D*K;
Dt = D*F;

tracksys = ss(newsysA,Bt,Ct,Dt);
ut = zeros(numel(t),2);
ut((10:end),1) = 1;
[yt,~,xt] = lsim(tracksys,ut,t);


utn = F*ut' - K*xt';
ytn = yt' - D*utn;
ytn = ytn';
% figure
% sgtitle('Unit Step Input Responses, 1st Input')
% subplot(1,2,1)
% plot(t,ytn(:,1))
% xlabel('Time (s)')
% ylabel('a_d')
% title('System output response to unit input VS time')
% grid on
% grid minor
% subplot(1,2,2)
% plot(t,ytn(:,2))
% xlabel('Time (s)')
% ylabel('a_b')
% title('System output response to unit input VS time')
% grid on
% grid minor
%
% figure
% sgtitle('State Response to Unit Step Input, 1st Input')
% subplot(2,2,1)
% plot(t,xt(:,1))
% xlabel('Time (s)')
% ylabel('f_{k_2}')
% title('f_{k_2} response to unit step input for 1st input')
% grid on
```

```
% grid minor
% subplot(2,2,2)
% plot(t,xt(:,2))
% xlabel('Time (s)')
% ylabel('f_{k_1}')
% title('f_{k_1} response to unit step input for 1st input')
% grid on
% grid minor
% subplot(2,2,3)
% plot(t,xt(:,3))
% xlabel('Time (s)')
% ylabel('v_{m_2}')
% title('v_{m_2} response to unit step input for 1st input')
% grid on
% grid minor
% subplot(2,2,4)
% plot(t,xt(:,4))
% xlabel('Time (s)')
% ylabel('v_{m_1}')
% title('v_{m_1} response to unit step input for 1st input')
% grid on
% grid minor

ut2 = zeros(numel(t),2);
ut2((10:end),2) = 1;
[yt2,~,xt2] = lsim(tracksys,ut2,t);
utn2 = F*ut2' - K*xt2';
ytn2 = yt2' - D*utn2;
ytn2 = ytn2';

% figure
% sgtitle('Unit Step Input Responses, 2nd Input')
% subplot(1,2,1)
% plot(t,ytn2(:,1))
% xlabel('Time (s)')
% ylabel('a_d')
% title('System output response to unit input VS time')
% grid on
% grid minor
% subplot(1,2,2)
% plot(t,ytn2(:,2))
% xlabel('Time (s)')
% ylabel('a_b')
% title('System output response to unit input VS time')
% grid on
% grid minor
%
% figure
% sgtitle('State Response to Unit Step Input, 2nd Input')
% subplot(2,2,1)
% plot(t,xt2(:,1))
% xlabel('Time (s)')
% ylabel('f_{k_2}')
% title('f_{k_2} response to unit step input for 2nd input')
```

```
% grid on
% grid minor
% subplot(2,2,2)
% plot(t,xt2(:,2))
% xlabel('Time (s)')
% ylabel('f_{k_1}')
% title('f_{k_1} response to unit step input for 2nd input')
% grid on
% grid minor
% subplot(2,2,3)
% plot(t,xt2(:,3))
% xlabel('Time (s)')
% ylabel('v_{m_2}')
% title('v_{m_2} response to unit step input for 2nd input')
% grid on
% grid minor
% subplot(2,2,4)
% plot(t,xt2(:,4))
% xlabel('Time (s)')
% ylabel('v_{m_1}')
% title('v_{m_1} response to unit step input for 2nd input')
% grid on
% grid minor

polplucon1 = F*ut' - K*xt';
for i = 1:size(polplucon1,2)
    polplcontsig1(i) = norm(polplucon1(:,i));
end
polplucon2 = F*ut2' - K*xt2';
for i = 1:size(polplucon2,2)
    polplcontsig2(i) = norm(polplucon2(:,i));
end
polplcontsig = [trapz(t(1:102),polplcontsig1(1:102))
 trapz(t(1:102),polplcontsig2(1:102))];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
```

Project Part 2

```
%part 1
opt = gramOptions('TimeIntervals',[0,.5]);
Go = gram(sys,'o',opt);
% Wc_other = [C;C*A;C*A^2;C*A^3];
% rank n observability matrix found using 'gram' function -> system is
% completely observable.

% Calculate enerergies using slide 8, lec 37 with inish condishs and
 obsv.
% gram.
% [T,M] = cdf2rdf(V,Deig);
% x01 = T*(real(V(:,1)) + imag(V(:,1)));
x01 = x01./norm(x01);
x02 = x02./norm(x02);
```

```
en_out1 = x01'*Go*x01;
en_out2 = x02'*Go*x02;
% Energy seems very small, need to look at energy or eigenvalues of
% observability mat. to determine degree of observability.
```

Part 2

Design a Luenberger observer for you system for threee different performance objectives

```
eigsp2 = (1/5)*eigs;        %eigenvalues for slow observer
Lp2 = place(A',C',eigsp2);
Lp2 = Lp2';

eigsone = eigs;             %eigenvalues for equal observer
Lone = place(A',C',eigsone);
Lone = Lone';

eigsfive = 5*eigs;          %eigenvalues for fast observer
Lfive = place(A',C',eigsfive);
Lfive = Lfive';
```

Part 3

For each case in part 2, sim. closed loop observer/controller response to a step at each ref. input, one at at a time, w/ 0 ICs in both plant and observer state.

```
%For slow observer
Aprime = [(A - B*K) B*K;zeros(4) (A - Lp2*C)];
Bprime = [(B*F);zeros(4,2)];
Cprime = [(C - D*K) ones(2,4)];
Dprime = D*F;

sysp2 = ss(Aprime,Bprime,Cprime,Dprime);
[Yp2,~,Xp2] = lsim(sysp2,ut,t);

utn = F*ut' - K*Xp2(:,[1:4])';
Yp2n = Yp2' - D*utn;
Yp2n = Yp2n';

figure
sgtitle('Slow Observer System Output, Unit Step Input in u1')
subplot(2,1,1)
plot(t,Yp2n(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Yp2n(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
```

```
title('a_b')
grid on
grid minor

saveas(gca,'P3out_slow_u1.eps',eps)

[Yp2_2,~,Xp2_2] = lsim(sysp2,ut2,t);

utn2 = F*ut2' - K*Xp2_2(:,[1:4])';
Yp2n_2 = Yp2_2' - D*utn2;
Yp2n_2 = Yp2n_2';

figure
sgtitle('Slow Observer System Output, Unit Step Input in u2')
subplot(2,1,1)
plot(t,Yp2n_2(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Yp2n_2(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
title('a_b')
grid on
grid minor

saveas(gca,'P3out_slow_u2.eps',eps)

yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
xM = {Xp2 Xp2_2};
sptitles = {'Slow Observer State Response to Unit Step Input in
 u_1','Slow Observer State Response to Unit Step Input in u_2'};
for i = 1:2
    figure
    for j = 1:4
        subplot(2,2,j)
        plot(t,xM{i}(:,j))
        yl = ylim;
        ylim([min(-abs(yl)),max(abs(yl))])
        xlabel('Time (s)')
        ylabel(yax{j})
        title(plottitles{j})
        grid on
        grid minor
    end
    sgtitle(sptitles{i});
    saveas(gca,['P3state_slow_u',num2str(i),'.eps'])

end
```

```
% equal observer
Aprime = [(A - B*K) B*K;zeros(4) (A - Lone*C)];
sysnorm = ss(Aprime,Bprime,Cprime,Dprime);
[Ynorm,~,Xnorm] = lsim(sysnorm,ut,t);

utn = F*ut' - K*Xnorm(:,[1:4])';
Ynorm_n = Ynorm' - D*utn;
Ynorm_n = Ynorm_n';

figure
sgtitle('Equal Observer System Output, Unit Step Input in u1')
subplot(2,1,1)
plot(t,Ynorm_n(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Ynorm_n(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
title('a_b')
grid on
grid minor

saveas(gca,'P3out_norm_u1.eps',eps)

[Ynorm_2,~,Xnorm_2] = lsim(sysnorm,ut2,t);

utn2 = F*ut2' - K*Xnorm_2(:,[1:4])';
Ynormn_2 = Ynorm_2' - D*utn2;
Ynormn_2 = Ynormn_2';

figure
sgtitle('Equal Observer System Output, Unit Step Input in u2')
subplot(2,1,1)
plot(t,Ynormn_2(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Ynormn_2(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
title('a_b')
grid on
grid minor

saveas(gca,'P3out_norm_u2.eps',eps)

yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
```

```
plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
xM = {Xnorm Xnorm_2};
sptitles = {'Equal Observer State Response to Unit Step Input in
 u_1','Equal Observer State Response to Unit Step Input in u_2'};
for i = 1:2
    figure
    for j = 1:4
        subplot(2,2,j)
        plot(t,xM{i}(:,j))
        yl = ylim;
        ylim([min(-abs(yl)),max(abs(yl))])
        xlabel('Time (s)')
        ylabel(yax{j})
        title(plottitles{j})
        grid on
        grid minor
    end
    sgtitle(sptitles{i});
    saveas(gca,['P3state_norm_u',num2str(i),'.eps'])
end

%Fast observer
Aprime = [(A - B*K) B*K;zeros(4) (A - Lfive*C)];
sys5 = ss(Aprime,Bprime,Cprime,Dprime);
[Y5,~,X5] = lsim(sys5,ut,t);

utn = F*ut' - K*X5(:,[1:4])';
Y5n = Y5' - D*utn;
Y5n = Y5n';

figure
sgtitle('Fast Observer System Output, Unit Step Input in u1')
subplot(2,1,1)
plot(t,Y5n(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Y5n(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
title('a_b')
grid on
grid minor

saveas(gca,'P3out_fast_u1.eps',eps)

[Y5_2,~,X5_2] = lsim(sys5,ut2,t);

utn2 = F*ut2' - K*X5_2(:,[1:4])';
Y5n_2 = Y5_2' - D*utn2;
Y5n_2 = Y5n_2';
```

```
figure
sgtitle('Fast Observer System Output, Unit Step Input in u2')
subplot(2,1,1)
plot(t,Y5n_2(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Y5n_2(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
title('a_b')
grid on
grid minor

saveas(gca,'P3out_fast_u2.eps',eps)

yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
xM = {X5 X5_2};
sptitles = {'Fast Observer State Response to Unit Step Input in
 u_1','Fast Observer State Response to Unit Step Input in u_2'};
for i = 1:2
    figure
    for j = 1:4
        subplot(2,2,j)
        plot(t,xM{i}(:,j))
        yl = ylim;
        ylim([min(-abs(yl)),max(abs(yl))])
        xlabel('Time (s)')
        ylabel(yax{j})
        title(plottitles{j})
        grid on
        grid minor
    end
    sgtitle(sptitles{i});
    saveas(gca,['P3state_fast_u',num2str(i),'.eps'],eps)

end
```

Part 4

```
%For each case in part 2, simulate the closed loop system response to
 non
%zero ICs (w/ zero ref. input). Use zero ICs in observer and non-zero
 ICs
%in plant, one state vector component at a time, with unit size.
X0 = [eye(4);eye(4)];
%slow observer
Aprime = [(A - B*K) B*K;zeros(4) (A - Lp2*C)];
```

```
[Vclosed,Dclosed] = eig(Aprime);
[T,~] = cdf2rdf(Vclosed,Dclosed);

ut = zeros(500,2);
ut2 = zeros(500,2);
sysp2 = ss(Aprime,Bprime,Cprime,Dprime);

X01 = T*X0(:,1);

[Yp2,~,Xp21] = lsim(sysp2,ut,t,X0(:,1));

utn = F*ut' - K*Xp2(:,[1:4])';
Yp2n1 = Yp2' - D*utn;
Yp2n1 = Yp2n1';

X02 = T*X0(:,2);

[Yp2,~,Xp22] = lsim(sysp2,ut,t,X0(:,2));

utn = F*ut' - K*Xp2(:,[1:4])';
Yp2n2 = Yp2' - D*utn;
Yp2n2 = Yp2n2';

X03 = T*X0(:,3);

[Yp2,~,Xp23] = lsim(sysp2,ut,t,X0(:,3));

utn = F*ut' - K*Xp2(:,[1:4])';
Yp2n3 = Yp2' - D*utn;
Yp2n3 = Yp2n3';

X04 = T*X0(:,4);

[Yp2,~,Xp24] = lsim(sysp2,ut,t,X0(:,4));

utn = F*ut' - K*Xp2(:,[1:4])';
Yp2n4 = Yp2' - D*utn;
Yp2n4 = Yp2n4';

yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
xM = {Xp21 Xp22 Xp23 Xp24};
sptitles = {'Slow Observer Response to Unit Initial Condition f_{k_2}
(0)=1','Slow Observer Response to Unit Initial Condition f_{k_1}
(0)=1','Slow Observer Response to Unit Initial Condition v_{m_2}
(0)=1','Slow Observer Response to Unit Initial Condition v_{m_1}
(0)=1'};
for i = 1:4
    figure
    for j = 1:4
        subplot(2,2,j)
        plot(t,xM{i}(:,j))
        yl = ylim;
        ylim([min(-abs(yl)),max(abs(yl))])
```

```
        xlabel('Time (s)')
        ylabel(yax{j})
        title(plottitles{j})
        grid on
        grid minor
    end
    sgtitle(sptitles{i});
    saveas(gca,['P4state_slow_IC',num2str(i),'.eps'],eps)
end

%equal observer
Aprime = [(A - B*K) B*K;zeros(4) (A - Lone*C)];
[Vclosed,Dclosed] = eig(Aprime);
T = cdf2rdf(Vclosed,Dclosed);
sysnorm = ss(Aprime,Bprime,Cprime,Dprime);

X01 = T*X0(:,1);

[Ynorm,~,Xnorm1] = lsim(sysnorm,ut,t,X0(:,1));

utn = F*ut' - K*Xnorm(:,[1:4])';
Ynorm_n1 = Ynorm' - D*utn;
Ynorm_n1 = Ynorm_n1';

X02 = T*X0(:,2);

[Ynorm,~,Xnorm2] = lsim(sysnorm,ut,t,X0(:,2));

utn = F*ut' - K*Xnorm(:,[1:4])';
Ynorm_n2 = Ynorm' - D*utn;
Ynorm_n2 = Ynorm_n2';

X03 = T*X0(:,3);

[Ynorm,~,Xnorm3] = lsim(sysnorm,ut,t,X0(:,3));

utn = F*ut' - K*Xnorm(:,[1:4])';
Ynorm_n3 = Ynorm' - D*utn;
Ynorm_n3 = Ynorm_n3';

X04 = T*X0(:,4);

[Ynorm,~,Xnorm4] = lsim(sysnorm,ut,t,X0(:,4));

utn = F*ut' - K*Xnorm(:,[1:4])';
Ynorm_n4 = Ynorm' - D*utn;
Ynorm_n4 = Ynorm_n4';

yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
xM = {Xnorm1 Xnorm2 Xnorm3 Xnorm4};
sptitles = {'Equal Observer Response to Unit Initial Condition f_{k_2}
(0)=1','Equal Observer Response to Unit Initial Condition f_{k_1}
(0)=1','Equal Observer Response to Unit Initial Condition v_{m_2}
```

```
(0)=1','Equal Observer Response to Unit Initial Condition v_{m_1}
(0)=1'};
for i = 1:4
    figure
    for j = 1:4
        subplot(2,2,j)
        plot(t,xM{i}(:,j))
        yl = ylim;
        ylim([min(-abs(yl)),max(abs(yl))])
        xlabel('Time (s)')
        ylabel(yax{j})
        title(plottitles{j})
        grid on
        grid minor
    end
    sgtitle(sptitles{i});
    saveas(gca,['P4state_norm_IC',num2str(i),'.eps'],eps)
end
%Fast observer
Aprime = [(A - B*K) B*K;zeros(4) (A - Lfive*C)];
[Vclosed,Dclosed] = eig(Aprime);
T = cdf2rdf(Vclosed,Dclosed);
sys5 = ss(Aprime,Bprime,Cprime,Dprime);

X01 = T*X0(:,1);

[Y5,~,X51] = lsim(sys5,ut,t,X0(:,1));

utn = F*ut' - K*X5(:,[1:4])';
Y5n1 = Y5' - D*utn;
Y5n1 = Y5n1';

X02 = T*X0(:,2);

[Y5,~,X52] = lsim(sys5,ut,t,X0(:,2));

utn = F*ut' - K*X5(:,[1:4])';
Y5n2 = Y5' - D*utn;
Y5n2 = Y5n2';

X03 = T*X0(:,3);

[Y5,~,X53] = lsim(sys5,ut,t,X0(:,3));

utn = F*ut' - K*X5(:,[1:4])';
Y5n3 = Y5' - D*utn;
Y5n3 = Y5n3';

X04 = T*X0(:,4);

[Y5,~,X54] = lsim(sys5,ut,t,X0(:,4));

utn = F*ut' - K*X5(:,[1:4])';
Y5n4 = Y5' - D*utn;
```

```
Y5n4 = Y5n4';

yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
xM = {X51 X52 X53 X54};
sptitles = {'Fast Observer Response to Unit Initial Condition f_{k_2}
(0)=1','Fast Observer Response to Unit Initial Condition f_{k_1}
(0)=1','Fast Observer Response to Unit Initial Condition v_{m_2}
(0)=1','Fast Observer Response to Unit Initial Condition v_{m_1}
(0)=1'};
for i = 1:4
    figure
    for j = 1:4
        subplot(2,2,j)
        plot(t,xM{i}(:,j))
        yl = ylim;
        ylim([min(-abs(yl)),max(abs(yl))])
        xlabel('Time (s)')
        ylabel(yax{j})
        title(plottitles{j})
        grid on
        grid minor
    end
    sgtitle(sptitles{i});
    saveas(gca,['P4state_fast_IC',num2str(i),'.eps'],eps)
end
```

Part 5

```
%Use LQR to design optimal state feedback gain K for plant s.t closed
 loop
%evals have time contants at least as short as slowest mode designed
 in
%project 1, and redesign input matrix F if necessary to preserve
 accurate
%DC tracking of ref. inputs by plant outputs
Q = C'*C;
R = eye(2);
[K,S,CLP] = lqr(A,B,Q,R);

F = inv(C*inv(-A+B*K)*B);

ut = zeros(numel(t),2);
ut((10:end),1) = 1;

Aprime = [(A - B*K) B*K;zeros(4) (A - Lfive*C)];
Bprime = [(B*F);zeros(4,2)];
Cprime = [(C - D*K) ones(2,4)];
Dprime = D*F;

[V5,D5] = eig(Aprime);
sys5 = ss(Aprime,Bprime,Cprime,Dprime);
[Y5,~,X5] = lsim(sys5,ut,t);
```

```
utn = F*ut' - K*X5(:,[1:4])';
Y5n = Y5' - D*utn;
Y5n = Y5n';

figure
sgtitle('Fast Observer System Output, Step Input in u1')
subplot(2,1,1)
plot(t,Y5n(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Y5n(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
title('a_b')
grid on
grid minor

saveas(gca,'P5out_fast_u1.eps',eps)

ut2 = zeros(numel(t),2);
ut2((10:end),2) = 1;

[Y5_2,~,X5_2] = lsim(sys5,ut2,t);

utn2 = F*ut2' - K*X5_2(:,[1:4])';
Y5n_2 = Y5_2' - D*utn2;
Y5n_2 = Y5n_2';

figure
sgtitle('Fast Observer System Output, Step Input in u2')
subplot(2,1,1)
plot(t,Y5n_2(:,1))
xlabel('Time (s)')
ylabel('Output a_d (N)')
title('a_d')
grid on
grid minor
subplot(2,1,2)
plot(t,Y5n_2(:,2))
xlabel('Time (s)')
ylabel('Output a_b (N)')
title('a_b')
grid on
grid minor

saveas(gca,'P5out_fast_u2.eps',eps)

yax = {'Force (N)','Force (N)','Velocity (m/s)','Velocity (m/s)'};
plottitles = {'f_{k_2}','f_{k_1}','v_{m_2}','v_{m_1}'};
```

```
xM = {X5 X5_2};
sptitles = {'Fast Observer State Response to Unit Step Input in
 u_1','Fast Observer State Response to Unit Step Input in u_2'};
for i = 1:2
    figure
    for j = 1:4
        subplot(2,2,j)
        plot(t,xM{i}(:,j))
        yl = ylim;
        ylim([min(-abs(yl)),max(abs(yl))])
        xlabel('Time (s)')
        ylabel(yax{j})
        title(plottitles{j})
        grid on
        grid minor
    end
    sgtitle(sptitles{i});
    saveas(gca,['P5state_fast_u',num2str(i),'.eps'],eps)
end

lqrucon1 = F*ut' - K*X5(:,[1:4])';
for i = 1:size(lqrucon1,2)
    lqrcontsig1(i) = norm(lqrucon1(:,i));
end
lqrucon2 = F*ut2' - K*X5_2(:,[1:4])';
for i = 1:size(lqrucon2,2)
    lqrcontsig2(i) = norm(lqrucon2(:,i));
end

lqrcontsig = [trapz(t(1:71),lqrcontsig1(1:71))
 trapz(t(1:71),lqrcontsig2(1:71))];

figure
subplot(2,1,1)
plot(t,polplucon1(1,:))
hold on
plot(t,lqrucon1(1,:))
xlabel('Time (s)')
ylabel('Control Signal u')
title('Pole Placement VS LQR Control Signal u1')
legend('Pole Placement Control','LQR Control')
grid on
grid minor
subplot(2,1,2)
plot(t,polplucon2(2,:))
hold on
plot(t,lqrucon2(2,:))
xlabel('Time (s)')
ylabel('Control Signal u')
title('Pole Placement VS LQR Control Signal u2')
legend('Pole Placement Control','LQR Control')
grid on
grid minor
sgtitle('LQR VS Pole Placement control signals')
```

```
saveas(gca,'P5Contens.eps',eps)
```

Published with MATLAB® R2020b