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
```

Published with MATLAB® R2019b