John Clouse Midterm 2

Table of Contents

Problem 1

```
clear all
A =[ -4.2 2.6 0;
    1.6 -3.2 1.6;
    0 2.6 -4.2];

fprintf('Rank of A: %d\n',rank(A));

% Decompose A into eigenvalue matrix and e'vecs. Easier to calcualte STM
% this way.
[V,D] = eig(A);
V_inv = inv(V);
Phi = zeros(3);
t = sym('t'); % Let t = t1-t0
% Here's the STM
for ii = 1:3
    Phi = Phi + exp(D(ii,ii)*t)*V(:,ii)*V_inv(ii,:);
end
Phi = vpa(Phi,5) %Show STM in a readable form

% Part b
B = [1;0;0];
sys = ss(A,B,eye(3),0);

% Calculate the controllability Grammian:
G = gram(sys, 'c')
% Rank = 3, reachable!
fprintf('Rank of G: %d\n',rank(G))

% Again, let t = t1-t0
u = B'*Phi'*inv(G)*([10;10;10]);
u = vpa(u,5)

% Part C
B = [1;1;1];

sys = ss(A,B,eye(3),0);

G = gram(sys, 'c')
```

```
% Not reachable! Rank < 3
fprintf('Rank of G: %d\n',rank(G))

% Reachable subspace:
[Q,R] = qr(G);
reachable = Q(:,1:2);
fprintf('Reachable subspace:\n')
disp(reachable)

[Q,R] = qr(G');
unreachable = Q(:,3);
fprintf('Unreachable subspace:\n')
disp(unreachable)
```

```
Rank of A: 3

Phi =

[          0.2927*exp(-6.6275*t) + 0.2073*exp(-0.77254*t) + 0.5*exp(-4.2*t
[ 0.27327*exp(-0.77254*t) - 0.27327*exp(-6.6275*t) + 3.0784e-16*exp(-4.2*t
[          0.2927*exp(-6.6275*t) + 0.2073*exp(-0.77254*t) - 0.5*exp(-4.2*t


G =

    0.1492    0.0486    0.0209
    0.0486    0.0338    0.0189
    0.0209    0.0189    0.0117

Rank of G: 3

u =

3219.0*exp(-6.6275*conj(t)) + 353.32*exp(-0.77254*conj(t)) - 3094.2*exp(-4


G =

    0.5115    0.6340    0.5115
    0.6340    0.7902    0.6340
    0.5115    0.6340    0.5115

Rank of G: 2
Reachable subspace:
   -0.5318    0.4661
   -0.6591   -0.7521
   -0.5318    0.4661

Unreachable subspace:
   -0.7071
   -0.0000
    0.7071
```

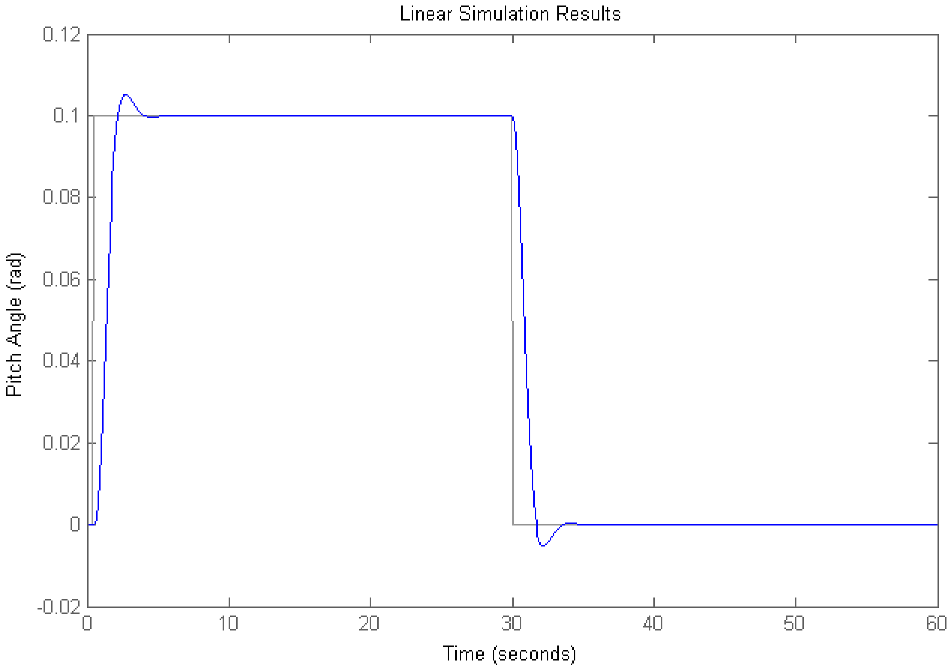
Problem 2

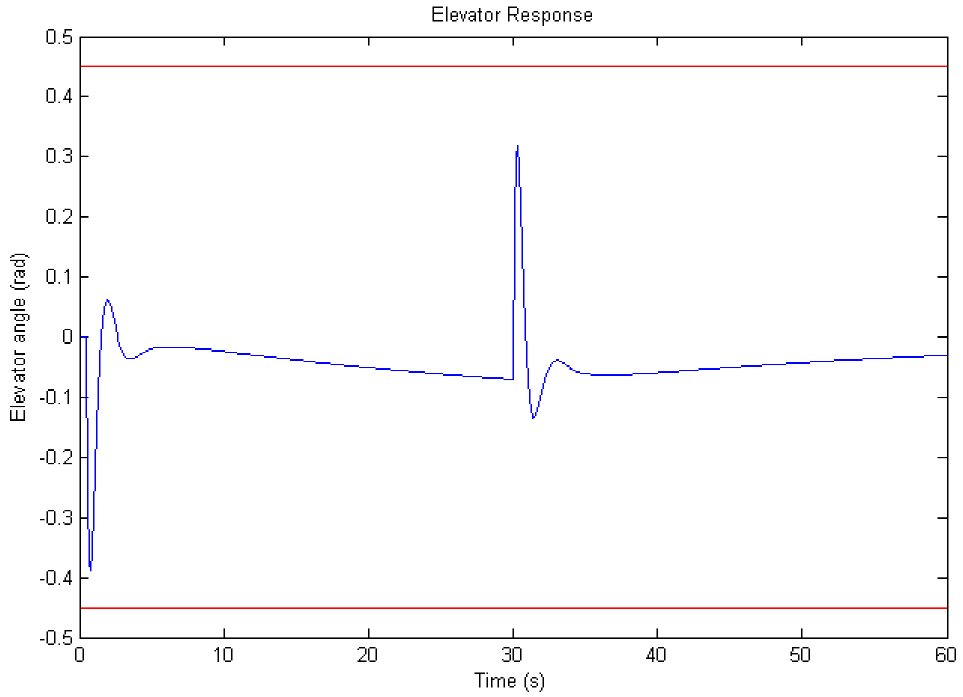
Part a)

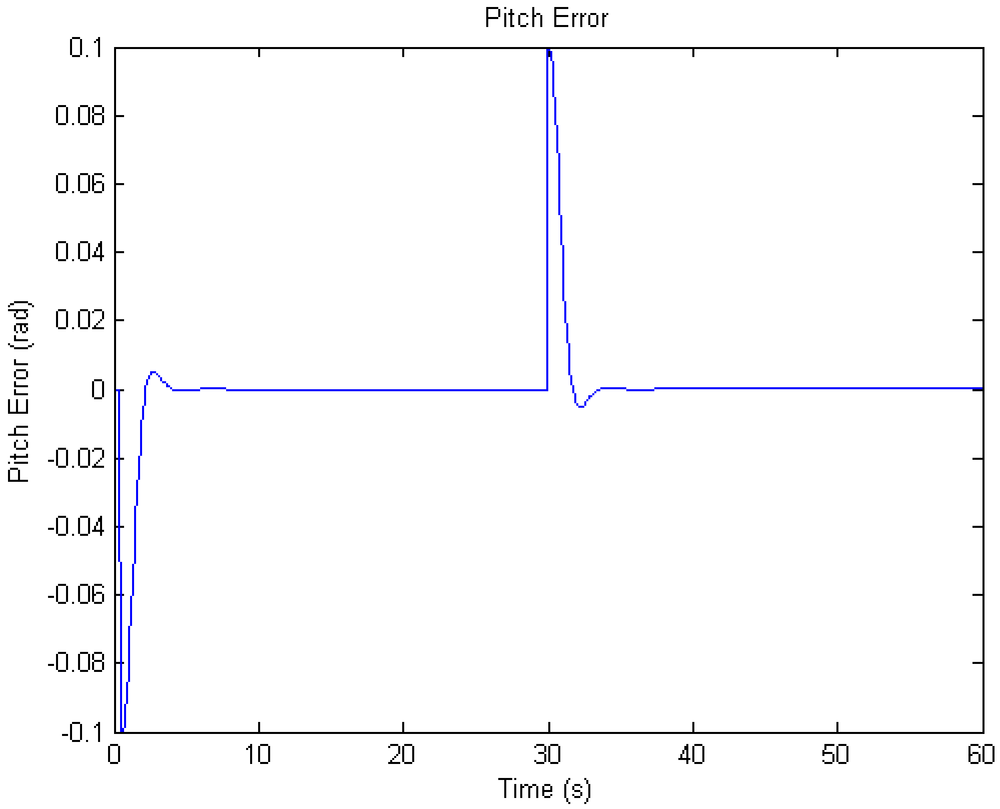
```
clear all
close all
A = [-0.0188 11.5959 0 -32.2;
    -0.0007 -0.5357 1 0;
    0.000048 -0.4944 -0.4935 0;
    0 0 1 0];
fprintf('Rank of A: %d\n',rank(A));

B = [0;0;-0.5632;0];
C = [0 0 0 1];

sys = ss(A,B,eye(4),0);
G = gram(sys, 'c');
%Reachable!
fprintf('Rank of G: %d\n',rank(G));

% part b
fprintf('Eigenvalues of A: \n')
disp(eig(A))
% The eigenvalues are 2 pairs of complex conjugates, with the real part
% being negative. The system is underdamped and will asymptotically
% approach equilibrium, meaning it is stable.

% Part c
% Using LQR control with integral error control as the last term in the
% state.
% x = [ du
%       angle of attack
%       pitch rate
%       pitch
%       e];

A_OL_Aug =[A,zeros(4,1);-C, zeros(1)];
B_OL_Aug = [B;zeros(1)];
% Initial set of weights: all equal
Q_wts = [1,1,1,1,1];
% Final weights: bump up the integral term's weight to reduce ss error, but
% not so much that it will make the elevator angle response exceed the
% limit.
Q_wts = [1,1,1,1,5];
Q_wts = Q_wts/sum(Q_wts);
% These were chosen to not violate any obvious max state issues
state_max = [10, 20*pi/180, .01, 30*pi/180, 0.01];
Q = diag(Q_wts.*Q_wts./(state_max.*state_max));
% Initial rho
rho = 1;
% Use the design requirement for the max.
u_max = .45;
R = rho/u_max;
```

```
[K_LQR, W, E] = lqr(A_OL_Aug,B_OL_Aug,Q,R);

A_LQR = [A-B*K_LQR(1:4),-B*K_LQR(5);-C,0];
B_LQR = [zeros(size(B));1];
C_Obs_LQR = [C, 0];
C_Obs_LQRFake = [1 0 0 0 0 0 0 0 0;...
    0 0 1 0 0 0 0 0 0;...
    0 0 0 0 0 1 0 0 0;...
    0 0 0 0 0 0 0 1 0];
LQR_system = ss(A_LQR, B_LQR, [C 0], 0);

t1 = 0.5;
t2 = 30;
tf = 60;
dt = 0.1;
seg1 = zeros(1,length(0:dt:(t1-dt)));
seg2 = 0.1*ones(1,length(t1:dt:(t2-dt)));
seg3 = zeros(1,length(t2:dt:(tf)));

r = [seg1 seg2 seg3];

outputPlot = figure('Position', [95   447   746   492]);
lsim(LQR_system, r, 0:dt:tf)
ylabel('Pitch Angle (rad)')

y = lsim(ss(A_LQR, B_LQR, eye(5), 0), r, 0:dt:tf);
u = -y*K_LQR';
respPlot = figure('Position', [862   449   745   490]);
plot(0:dt:tf,u)
hold on
plot(0:dt:tf,.45,'r--')
plot(0:dt:tf,-.45,'r--')
title('Elevator Response')
xlabel('Time (s)')
ylabel('Elevator angle (rad)')

errorPlot = figure;
plot(0:dt:tf,y(:,4)'-r)
title('Pitch Error')
xlabel('Time (s)')
ylabel('Pitch Error (rad)')

% Part d)
% The controller was pretty good for the initial iteration, and only
% tweaked slightly to achieve the presented results. The closed loop poles
% are:
%   -4.2621 + 0.0000i
%   -1.4627 + 1.6685i
%   -1.4627 - 1.6685i
%   -0.5196 + 0.0000i
%   -0.0350 + 0.0000i
% The CL poles are all on the negative side of the real axis, and there is
% only one complex conjugate pair. This ensures that the system is
% asymptotically stable. The actuator response makes sense: the deflection
% is highest during the rise/drop stages, and it settles to its final value
% similarly to the pitch angle. The direction also makes sense, given the A
% and B matrices as well as the included picture.
```

```
Rank of A: 4

Rank of G: 4
Eigenvalues of A:
  -0.5221 + 0.7029i
  -0.5221 - 0.7029i
  -0.0019 + 0.1250i
  -0.0019 - 0.1250i
```




Elevator Response
Elevator angle (rad)
Time (s)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
10
20
30
40
50
60


Pitch Error
Pitch Error (rad)
Time (s)
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
10
20
30
40
50
60

Problem 3

```
w0 =  7.29219108e-5; % rad/s
r0 = 4.218709065 ; % m
% w0 =  2*pi; % rad/s
% r0 = 1 ; % m

A = [0 1 0 0;
    2*w0^2 0 0 2*r0*w0;
    0 0 0 1;
    0 -2*w0/r0 0 0];

B = [0 0;1 0;0 0;0 1/r0];

C1 = [1 0 0 1];
C2 = [1 0 1 0];

% Part a)
% The rank of A is 3, this can be determined by observation: the third
% column is all zero, so max possible rank is 3. The second column cannot
% be linearly combined with another column except with the trivial case
% of being multiplied by 0, so it is LI. The fourth column cannot be
% linearly manipulated to produce the first column due to its 3rd element.
% Thus there are three LI columns, and rank is 3.
% Row rank: 2nd row is LI because it's the only one with a value in column
% 1. First and fourth rows are LD. None of these rows can be combined to
% make row 3, so rank is 3.

% Since the controlability matrices are poorly conditioned, use rref to
% determine rank
fprintf('u1:\n')
disp(rref(ctrb(A,B(:,1))))
fprintf('u2:\n')
disp(rref(ctrb(A,B(:,2))))

% Rank of controllability matrix for u1-only is 3
% Rank of controllability matrix for u2-only is 4
% Keeping u2 means the system is controllable (but not necessarily
% reachable), since the controllability matrix has rank = 4.

% Part b)
% Since the controlability matrices are poorly conditioned, use rref to
% determine rank
fprintf('y1:\n')
disp(rref(obsv(A,C1)))
fprintf('y2:\n')
disp(rref(obsv(A,C2)))

% Rank of observability matrix for y1 is 3
% Rank of observability matrix for y2-only is 4
% Using y2 means the system is observable since the observability matrix
% has rank = 4.
```

```
u1:
    1.0000         0         0         0
         0    1.0000         0   -0.0000
         0         0    1.0000         0
         0         0         0         0

u2:
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

y1:
     1     0     0     0
     0     1     0     0
     0     0     0     1
     0     0     0     0

y2:
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

Published with MATLAB® R2013b