John Clouse, Homework 7

Table of Contents

a)

```
close all

A = [-10 0 -10 0; 0 -0.7 9 0; 0 -1 -0.7 0;1 0 0 0];
B = [20 2.8; 0 -3.13; 0 0; 0 0];
C = [0,0,1,0;0,0,0,1];
C_fake = eye(4); %to capture the state to calculate control input

system = ss(A,B,C,0);

cont_mat = ctrb(system);

rank(cont_mat)
%It is reachable!

[V,d] = eig(A);

lambda = diag(d)
% There is a pole at zero, so the open loop system is unstable.
% The negaitve-real part is assymptotically stable if there are no unstable
% poles.
% The complex-conjugate pair will cause oscillations that assymptotically
% reach stability if there are no unstable poles.
```

```
ans =

     4


lambda =

   0.000000000000000 + 0.000000000000000i
 -10.000000000000000 + 0.000000000000000i
  -0.700000000000000 + 3.000000000000000i
  -0.700000000000000 - 3.000000000000000i
```

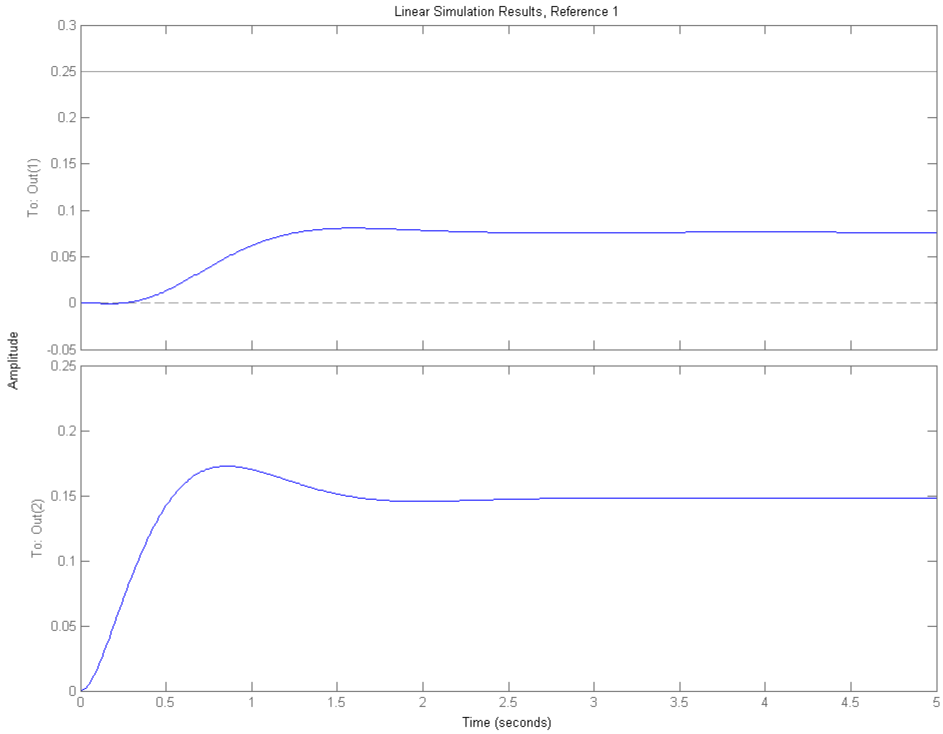
b)

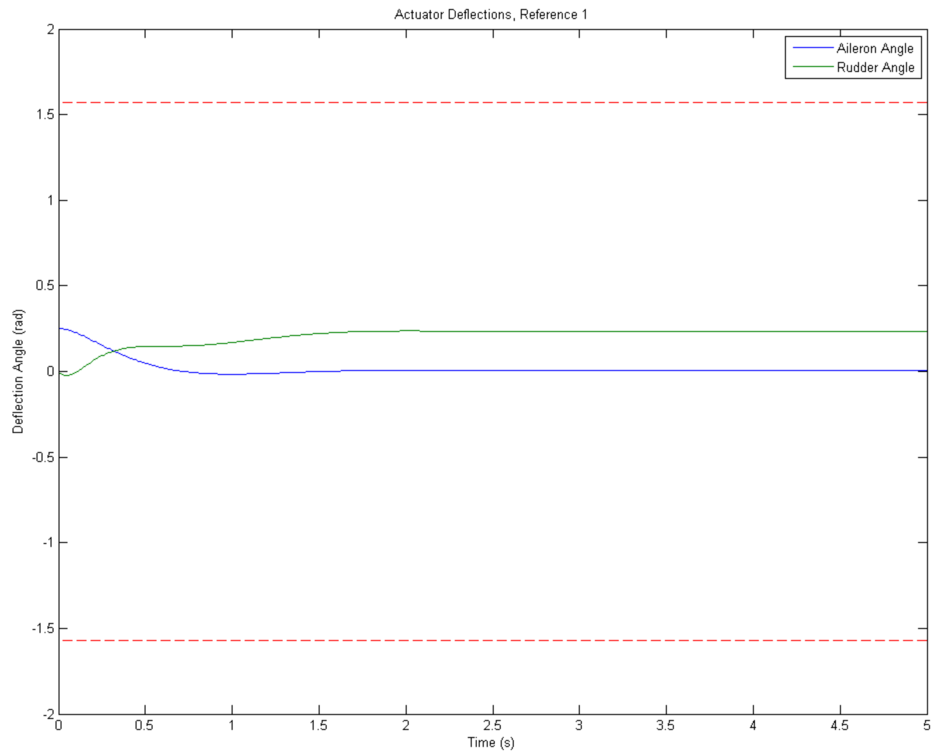
The design parameters

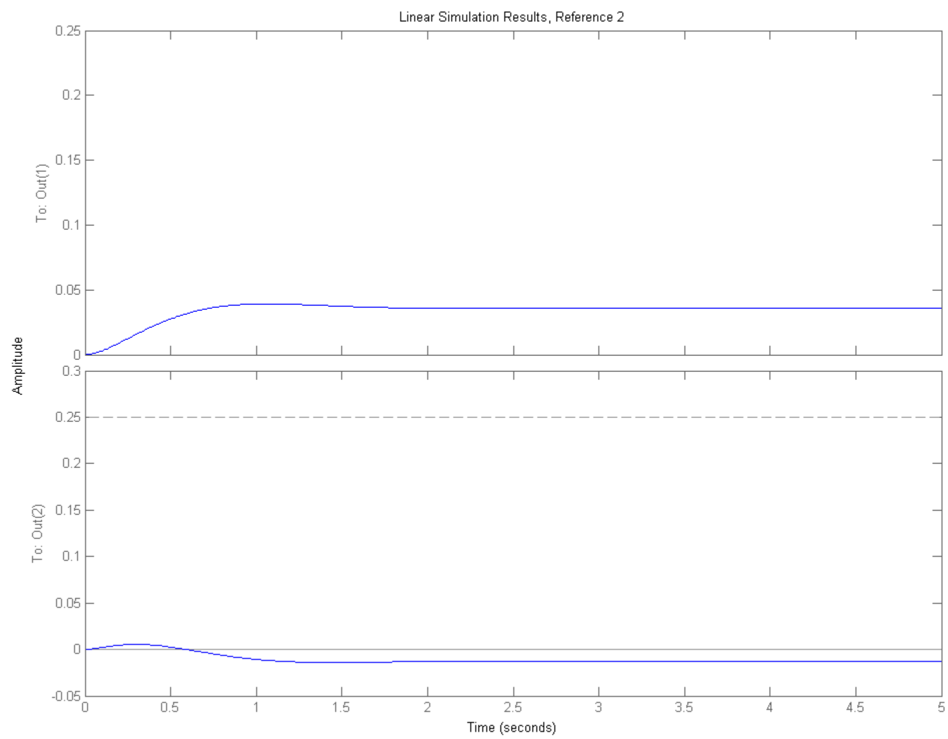
```
PO_desired = 10/100;
PS_desired = 5/100;
PO = 9/100;
PS = 4/100; %Settle percentage
Ts = 1.5;

% Get the desired dominant poles
damp_times_wn = -log(PS)/Ts
damping_ratio = -log(PO)/sqrt(pi*pi+(log(PO))^2);
wn = damp_times_wn/damping_ratio;
wd = wn*sqrt(1-damping_ratio^2)

P = [-5 -10 complex(-damp_times_wn, wd) complex(-damp_times_wn, -wd)];
K = place(A,B,P);
F = eye(2);

A_CL = A-B*K;
B_CL = B*F;

CL_system = ss(A_CL, B_CL, C,0);
CL_system_Fake = ss(A_CL, B_CL, C_fake,0);
t = 0:0.01:5;
r1 = [.25;0];
r2 = [0;0.25];

% No amount of tuning can eliminate the steady-state error for this system
% since F != inv[C*inv[A-BK]*B]. The feedforward gain is chosen to produce
% zero steady-state error in this manner.

% Reference 1
figure('Position',[0 0 hw_pub.figWidth hw_pub.figHeight])
lsim(CL_system,repmat(r1,1,length(t)),t)
title('Linear Simulation Results, Reference 1')
y1 = lsim(CL_system_Fake,repmat(r1,1,length(t)),t);
plot_CL_ctrl_outputs(r1,F,K,y1,t);
title('Actuator Deflections, Reference 1')
print_design_params(y1,t,PO_desired,PS_desired,Ts,3)

% Reference 2
figure('Position',[0 0 hw_pub.figWidth hw_pub.figHeight])
lsim(CL_system,repmat(r2,1,length(t)),t)
title('Linear Simulation Results, Reference 2')
y2 = lsim(CL_system_Fake,repmat(r2,1,length(t)),t);
plot_CL_ctrl_outputs(r2,F,K,y2,t);
title('Actuator Deflections, Reference 2')
print_design_params(y2,t,PO_desired,PS_desired,Ts,4)
```

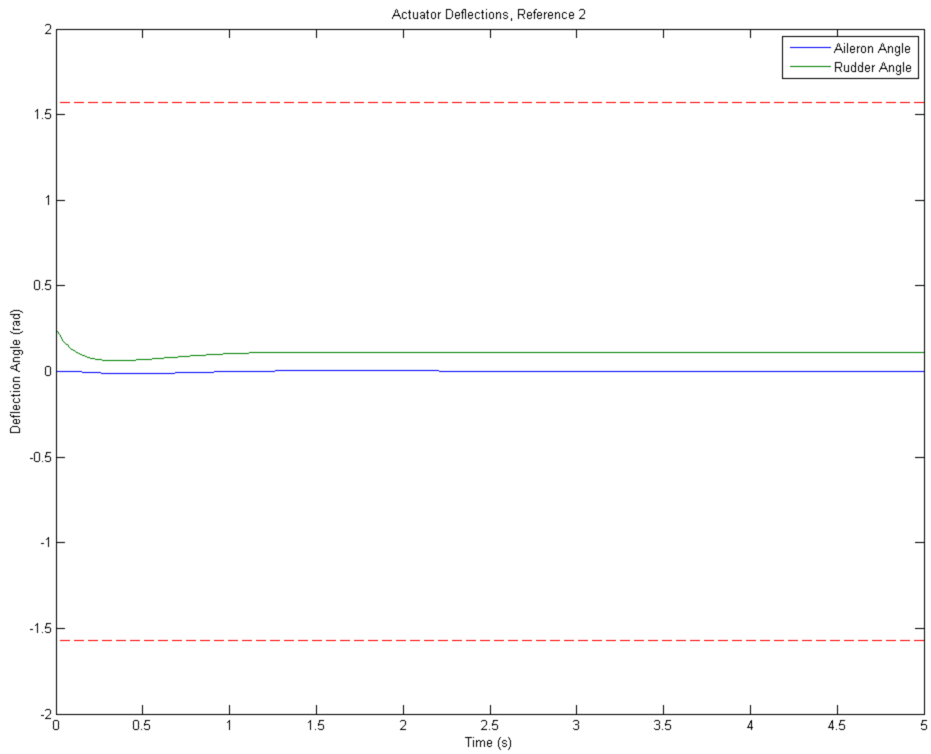
```
        damp_times_wn =
```

```
   2.145917216578801

wd =

   2.799730084680033

Max overshoot within 10%: True (0.081)
Settled within 1.5 sec: True

Max overshoot within 10%: True (0.005)
Settled within 1.5 sec: False
```




Actuator Deflections, Reference 1
Aileron Angle
Rudder Angle
Deflection Angle (rad)
Time (s)


Linear Simulation Results, Reference 2
To: Out(1)
To: Out(2)
Amplitude
Time (seconds)



Part c)

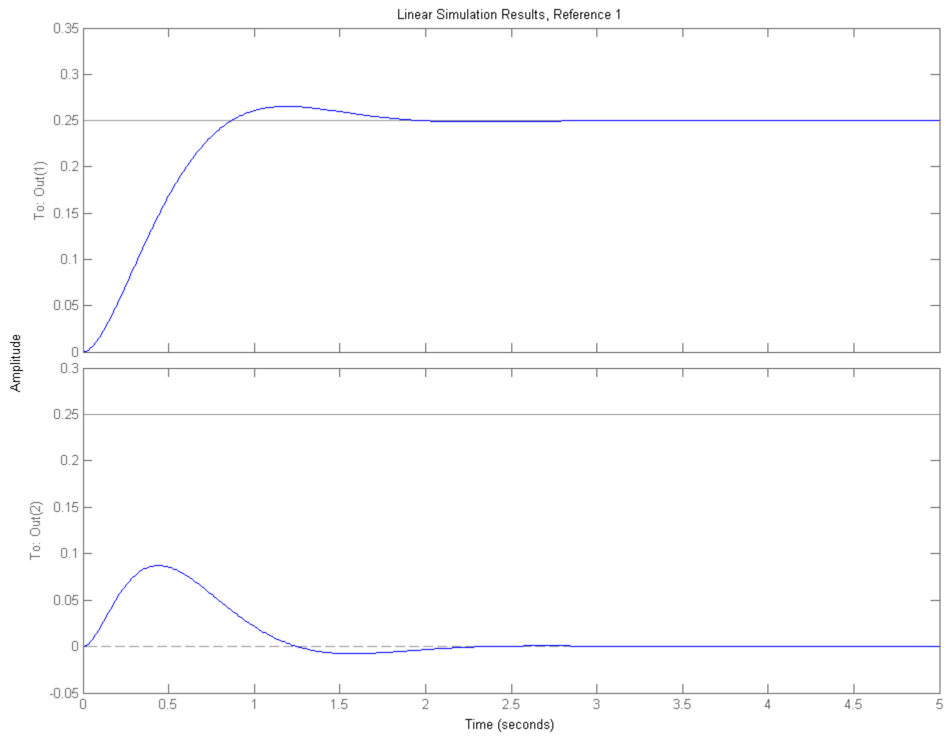
```
F = inv(C*inv(-A+B*K)*B);
B_CL = B*F;
CL_system_FF = ss(A_CL, B_CL, C,0);
CL_system_FF_Fake = ss(A_CL, B_CL, C_fake,0);

% To tune K, complex conj. poles were picked s.t. the natural frequency of
% damping ratio of a second-order system would adhere to the design
% parameters. The design parameters were given some margin to more easily
% place the remaining poles. The remaining poles were picked to be negative
% real numbers beyond (damping ratio)*(natural frequency). Their exact
% values were adjusted until the desired design was achieved.

% Reference 1
figure('Position',[0 0 hw_pub.figWidth hw_pub.figHeight])
lsim(CL_system_FF,repmat(r1,1,length(t)),t);
title('Linear Simulation Results, Reference 1')
y1 = lsim(CL_system_FF_Fake,repmat(r1,1,length(t)),t);
plot_CL_ctrl_outputs(r1,F,K,y1,t); % The control outputs
title('Actuator Deflections, Reference 1')
print_design_params(y1,t,PO_desired,PS_desired,Ts,3) % The g

% Reference 2
figure('Position',[0 0 hw_pub.figWidth hw_pub.figHeight])
```

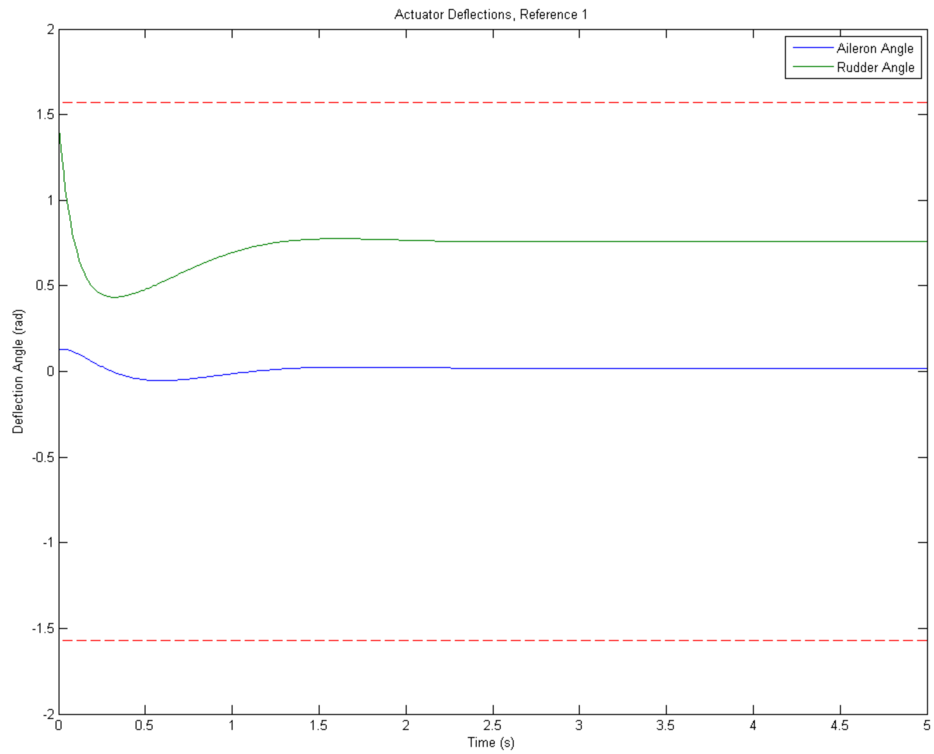
```
lsim(CL_system_FF,repmat(r2,1,length(t)),t)
title('Linear Simulation Results, Reference 2')
y2 = lsim(CL_system_FF_Fake,repmat(r2,1,length(t)),t);
plot_CL_ctrl_outputs(r2,F,K,y2,t);
title('Actuator Deflections, Reference 2')
print_design_params(y2,t,PO_desired,PS_desired,Ts,4)
```

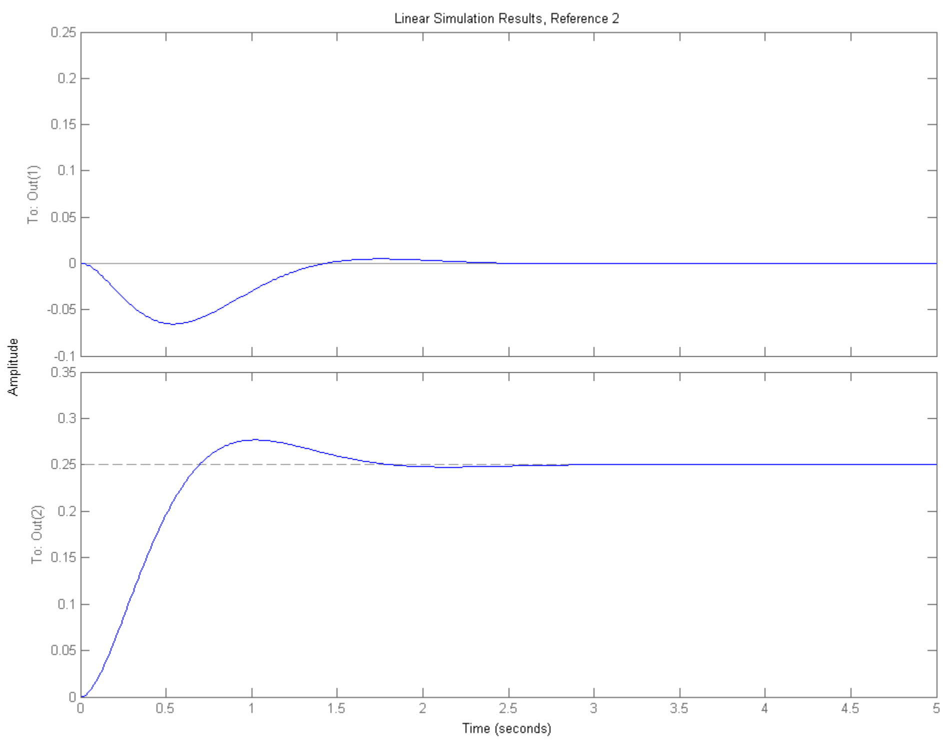
Max overshoot within 10%: True (0.265)

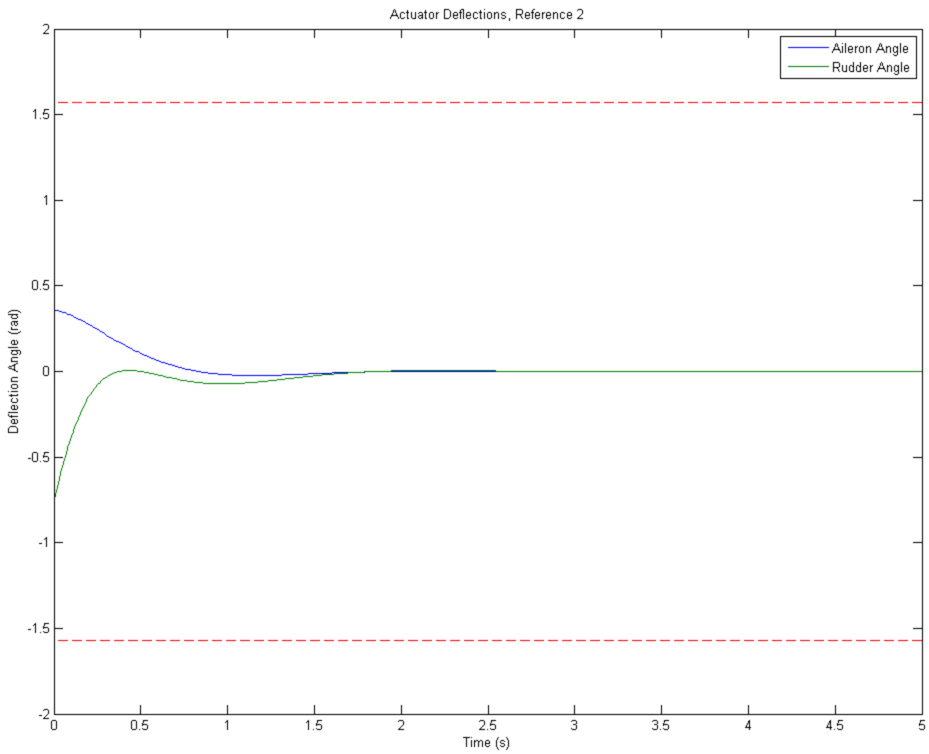
Settled within 1.5 sec: True

Max overshoot within 10%: True (0.277)
Settled within 1.5 sec: True




Actuator Deflections, Reference 1
Aileron Angle
Rudder Angle
Deflection Angle (rad)
Time (s)


Linear Simulation Results, Reference 2
Amplitude
To: Out(1)
To: Out(2)
Time (seconds)

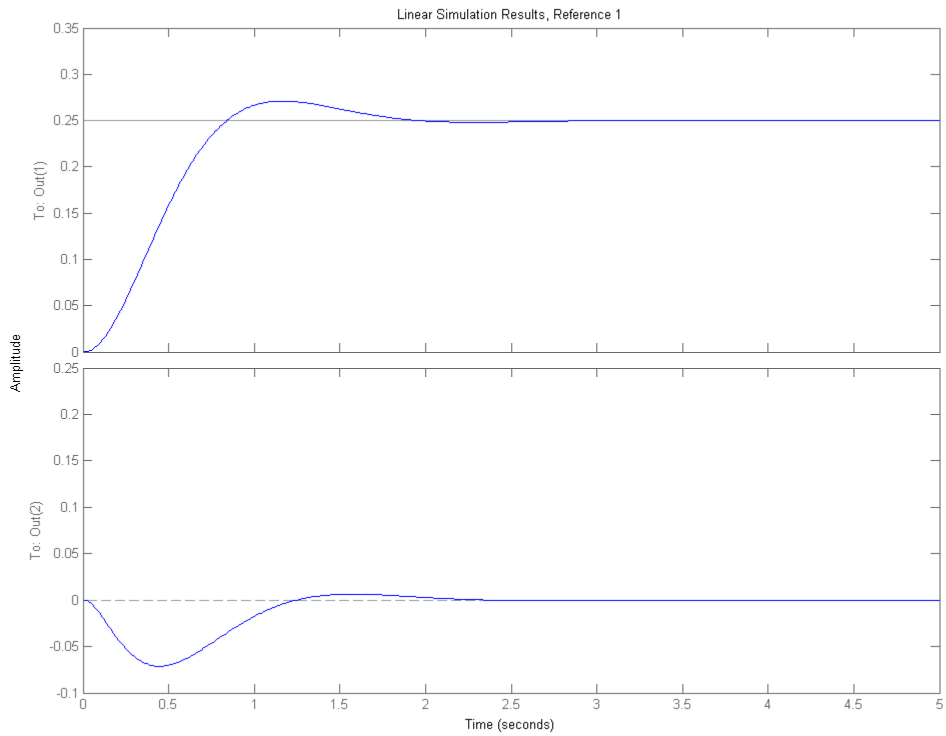


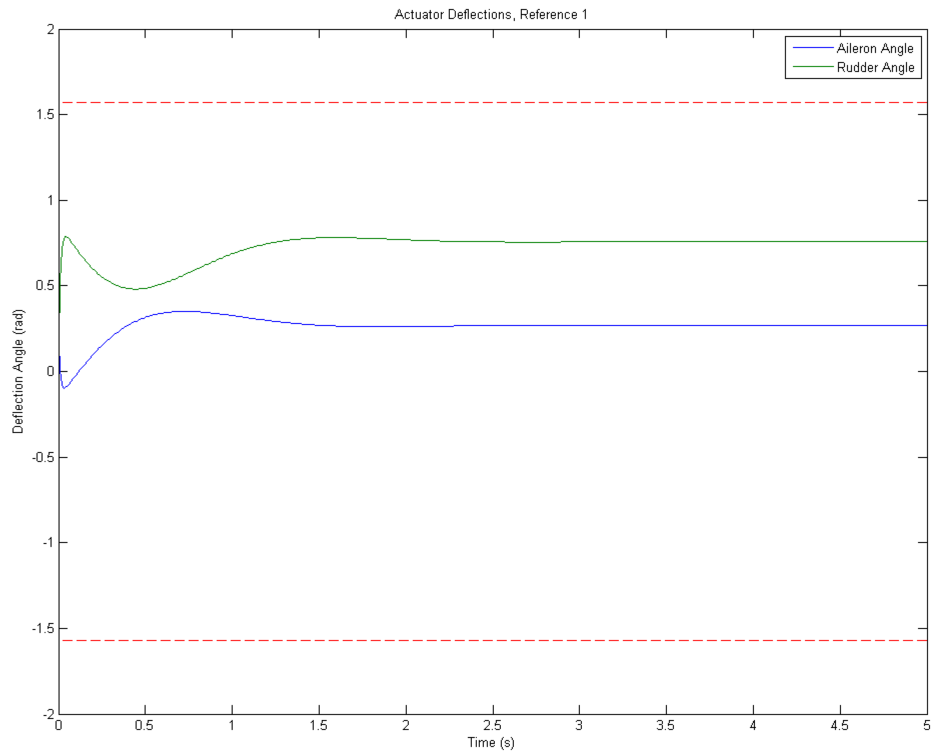
Part d)

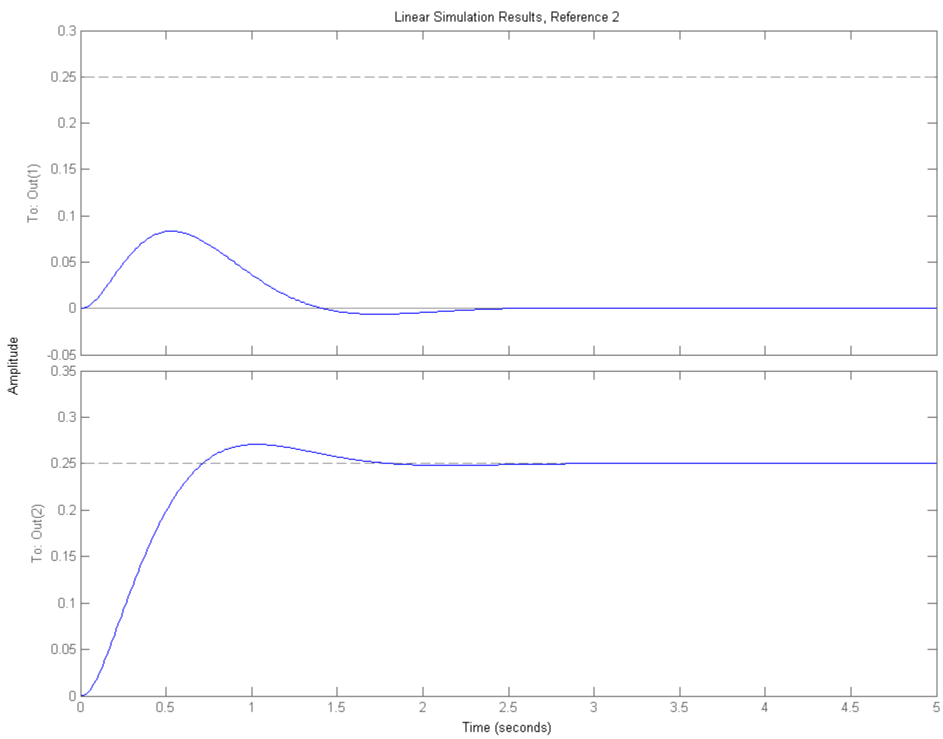
```
F = eye(2);
A_OL_Aug = [A,zeros(4,2);-C, zeros(2,2)];
B_OL_Aug = [B;zeros(2,2)];
P_Aug = [-100,-100.1,P];
K_Aug = place(A_OL_Aug,B_OL_Aug,P_Aug);
K = K_Aug(1:2,1:4); % gain for the nominal states
KI = K_Aug(1:2,5:6); % Integral gain
A_CL_Aug = [A-B*K, -B*KI; -C, zeros(2,2)];
B_CL_Aug = [zeros(4,2);eye(2)];

CL_system_Integral = ss(A_CL_Aug, B_CL_Aug,[C zeros(2)],0);
CL_system_Integral_Fake = ss(A_CL_Aug, B_CL_Aug,eye(6),0);

% Tuning K was done by placing the poles just like part c, with the
% additional poles being very far to the negative side of the Real axis.
% This controller will be better at rejecting unexpected disturbances or
% model errors than c), since it actively controls the error to zero.
% However, this controller requires an additional state for each desired
% output to be driven to a reference.

% Reference 1
figure('Position',[0 0 hw_pub.figWidth hw_pub.figHeight])
lsim(CL_system_Integral,repmat(r1,1,length(t)),t)
```

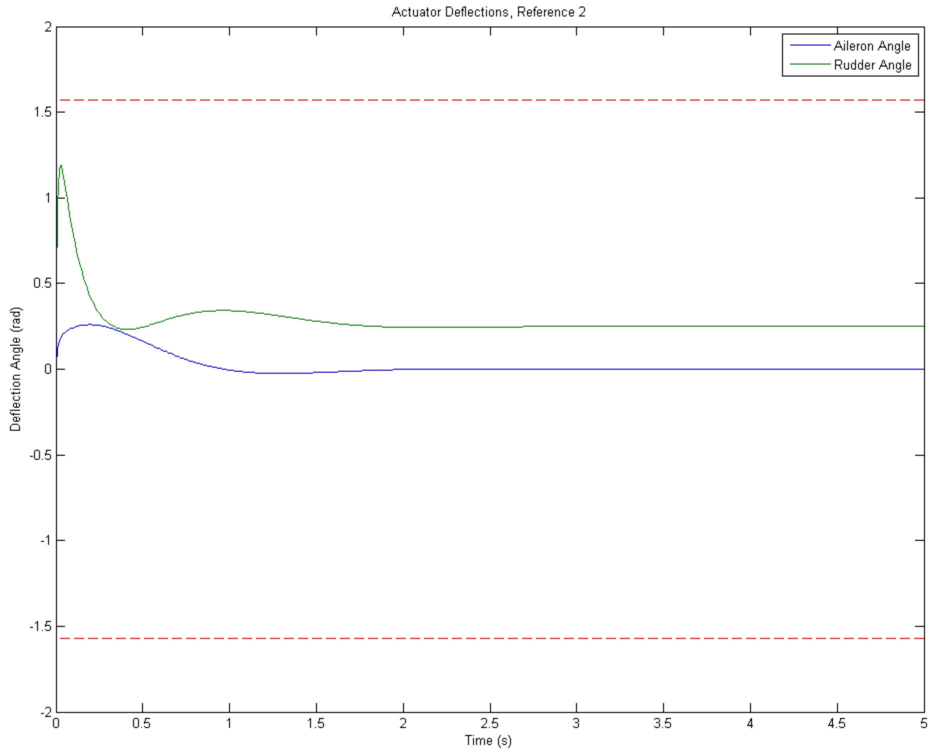
```
title('Linear Simulation Results, Reference 1')
y1 = lsim(CL_system_Integral_Fake,repmat(r1,1,length(t)),t);
plot_CL_ctrl_outputs(r1,F,K_Aug,y1,t);
title('Actuator Deflections, Reference 1')
print_design_params(y1,t,PO_desired,PS_desired,Ts,3)

% Reference 2
figure('Position',[0 0 hw_pub.figWidth hw_pub.figHeight])
lsim(CL_system_Integral,repmat(r2,1,length(t)),t)
title('Linear Simulation Results, Reference 2')
y2 = lsim(CL_system_Integral_Fake,repmat(r2,1,length(t)),t);
plot_CL_ctrl_outputs(r2,F,K_Aug,y2,t);
title('Actuator Deflections, Reference 2')
print_design_params(y2,t,PO_desired,PS_desired,Ts,4)
```

```
Max overshoot within 10%: True (0.271)

Settled within 1.5 sec: True

Max overshoot within 10%: True (0.271)
Settled within 1.5 sec: True
```




Actuator Deflections, Reference 1
Aileron Angle
Rudder Angle
Deflection Angle (rad)
Time (s)


Linear Simulation Results, Reference 2
Amplitude
To: Out(1)
To: Out(2)
Time (seconds)



Published with MATLAB® R2013b