Course Name: Control System Design

Course Number and Section: **14:332:417:01**

Project: A Pitch Controller for a BOEING Aircraft

Professor: Zoran Gajic

Date Performed: 12/19/2016 – 12/23/2016

Date Submitted: 12/23/2016

Submitted by: Eun-Sol Kim (133000680)

GRADE: ________________

COMMENTS:

Electrical and Computer Engineering Department
School of Engineering
Rutgers University, Piscataway, NJ 08854
Department of Electrical and Computer Engineering
Rutgers University – College of Engineering
New Brunswick, NJ

Course:	14:332:418
Laboratory:	Project 3: A Pitch Controller for a BOEING Aircraft
Date(s) Performed:	12/19/2016 – 12/23/2016
Date Submitted:	12/23/2016
Submitted by:	Eun-Sol Kim
Teaching Assistant:	Lingyi Xu

PURPOSE

Project # 3 — 332:417 Control System Design
A Pitch Controller for a BOEING Aircraft

Project assigned Dec. 6, 2016. Project due the last day of the final exam by **noon** Dec. 23. 2016.

The linearized equations governing the motion of a BOEING's commercial aircraft are given by (Messner and Tilbury, *Control Tutorials for MATLAB and Simulink*, Addison Wesley, 1998)

$$\begin{aligned}\frac{d\alpha(t)}{dt} &= -0.313\alpha(t) + 56.7q(t) + 0.232\delta_e(t)\\ \frac{dq(t)}{dt} &= -0.0139\alpha(t) - 0.426q(t) + 0.0203\delta_e(t)\\ \frac{d\theta(t)}{dt} &= 56.7q(t)\end{aligned} \tag{1}$$

where $\theta(t)$ represents the pitch angle. The corresponding open-loop transfer function obtained from (1) is given by

$$\frac{\Theta(s)}{\Delta(s)} = \frac{1.151s + 0.1774}{s^3 + 0.739s^2 + 0.921s} = \frac{1.151(s + 0.1541)}{s(s^2 + 0.739s + 0.921)} = 1.151G(s) \tag{2}$$

In this project we design an autopilot that controls the pitch angle $\theta(t)$ of this aircraft. The autopilot is obtained by forming the closed-loop system with a unity feedback and a controller of the form $KG_c(s)$. For simplicity we assume that $K = 1.151K'$ so that the open-loop feedback transfer function is $KG_c(s)G(s)$.

(a) Find the steady state error due to a unit ramp input of the original ($K = 1, G_c(s) = 1$) closed-loop system ($(G(s)/1 + G(s))$). Plot the closed-loop system ramp response and observe (check) the corresponding steady state error. Hint: In order to find the ramp response use the MATLAB function `y=lsim(cnum,cden,t,t)` with `t=0:0.1:30`.

(b) Find the value for the static gain K such that the steady state error due to the unit ramp is at most 10% ($e_{ss}^{ramp} = 0.1$). For the obtained value of K plot the corresponding closed-loop system ramp response and notice the steady state error improvement. Hint: Use the same time range as in part (a).

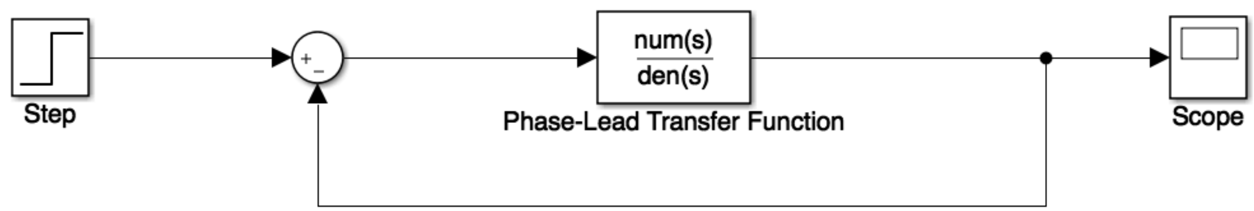
(c) For the obtained value of K find the phase and gain stability margins and observe that the phase margin is pretty pure. Design the *phase-lead* controller $G_c(s)$ to improve the phase stability margin such that the compensated system has the phase stability margin close to 50°. Find the step response of the compensated system and compare it to the step response of the uncompensated system whose static gain K is found in part (b). Comment on the transient response improvement of the compensated system. Hint: In order to be able to estimate the value for ω_{wax} use the following frequency range `w=0.1:0.1:100` with `bode(K*num,den,w)`. MATLAB will produce, for this particular example, the Bode plot in the frequency range up to $10\,\mathrm{rad/s}$. However, in the formulated design problem ω_{wax} is greater than $10\,\mathrm{rad/s}$.

(d) Design the *phase-lag* controller to satisfy the stability requirement imposed in (c). Find the step response of the system compensated (controlled) by the phase-lag controller and compare it to the step response of the system compensated by the phase-lead controller. Which one has a smaller rise time? Which one do you prefer?

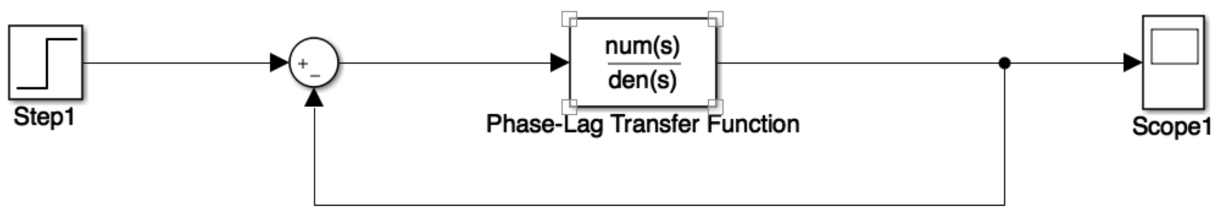
(e) Using the SIMULINK package, build the block diagrams for the system controlled by phase-lead and phase-lag controllers, plot the step responses in both cases, and confirm the results obtained in Parts (c) and (d).

Hint: Use and appropriately modify MATLAB programs for Examples 9.4 and 9.5.

ACTIVITY #b
Phase-Lead Block Diagram:



ACTIVITY #d
Phase-Lag Block Diagram:



THEORY

For this project, we have accompanied the chapter 9 concepts such as Phase-Lag compensation and Phase-Lead compensation. Given the linearized equation that governs the motion of the aircraft, we could design an autopilot that controls the pitch angle of the aircraft. We assumed that it is a unity feedback with controller equation as $K * G_c(s)$. *By assuming that* $K = 1.151 * K'$, *we could find the Bode gain because* $K_p = K_v$ *and* $K_B = K_v$. *We assume that the input is unit ramp; therefore, the steady state error is* $e_{ss} = \frac{1}{K_v}$ *where* $K_v = \lim_{s \to 0} s * G(s)$. *Since* $K_B = \frac{K * z_1 * z_2}{p_1 * p_2}$ *where* $K = 1.151 * K'$. *From found K, we can find phase and gain margin stability of the system by looking at the bode plot of magnitude and phase. By finding* $\Phi \max$ *to be* $5°$ *to* $10°$, *we can calculate value of parameter a:*

$$a = (1 + \sin(\Phi\_\max))/(1 - \sin(\Phi\_\max))$$

Then, we can estimate the compensator's pole with $G_c(s) = \frac{a * s + p_c}{s + p_c}$.

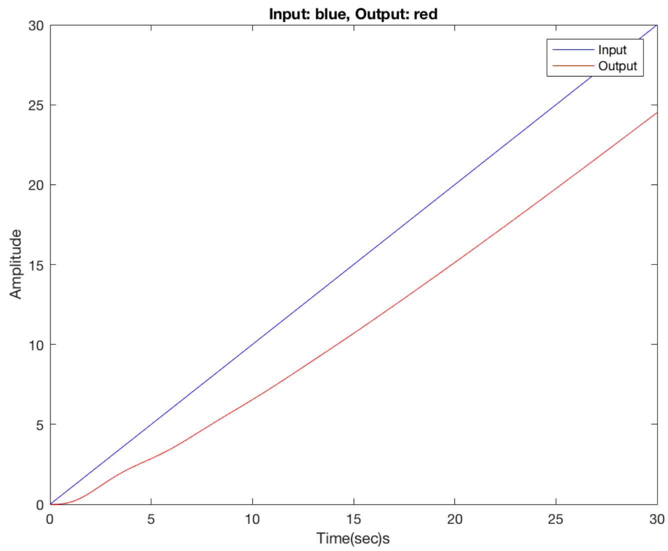
The Phase-Lag Controller and Phase-Lead Controller is formulated as follows:

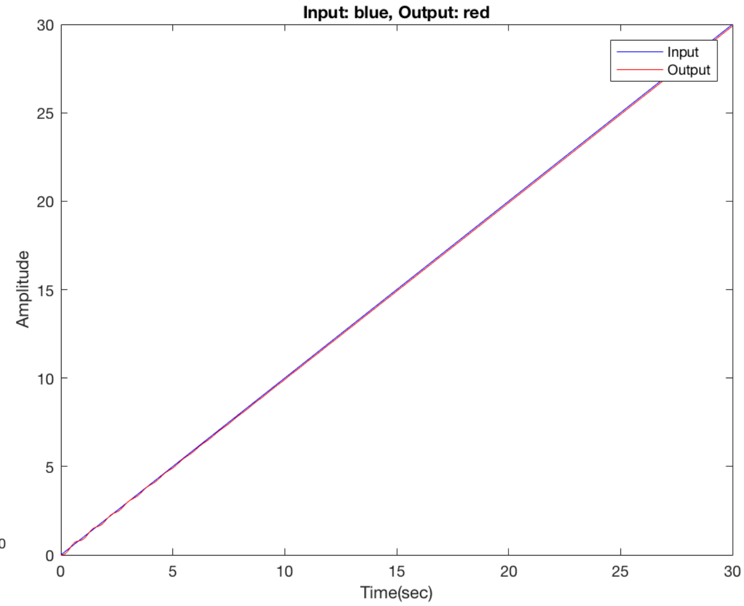
$$Phase - Lead\ Compensator: \quad G_{lead}(jw) = \frac{1 + j\left(\frac{w}{z_2}\right)}{1 + j\left(\frac{w}{p_2}\right)}$$

$$Phase - Lag\ Compensator: G_{lag}(jw) = \frac{\left(1 + j\left(\frac{w}{z_1}\right)\right)}{\left(1 + j\left(\frac{w}{p_2}\right)\right)}$$

Phase-Lead can be used in case pole 1 is larger than zero 1 and Phase-Lag can be used in case when pole1 is smaller than zero1. Phase Lead compensator is PD type wherea phase-Lag is PI type. Phase- Lag reducs the system bandwidh and steady-state error wherea Phase-Lead increases the bandwidth and increase the phase.

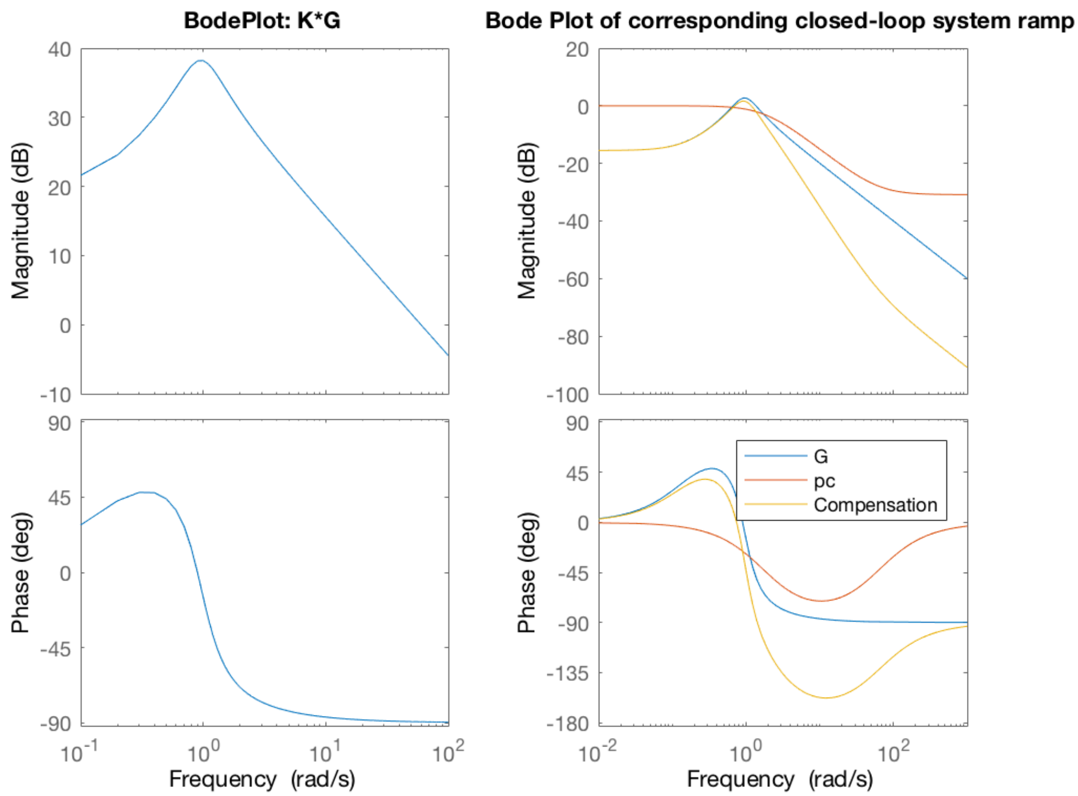
DATA SECTION

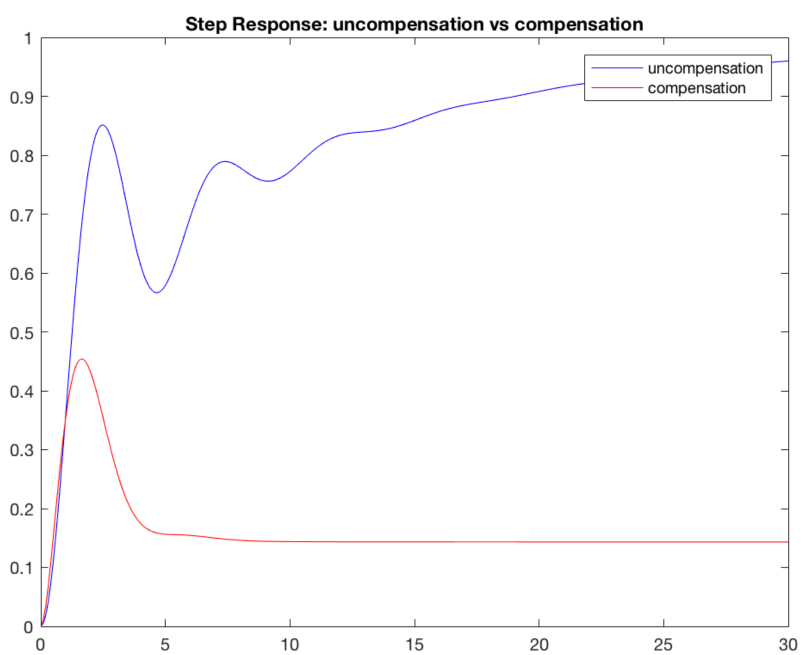


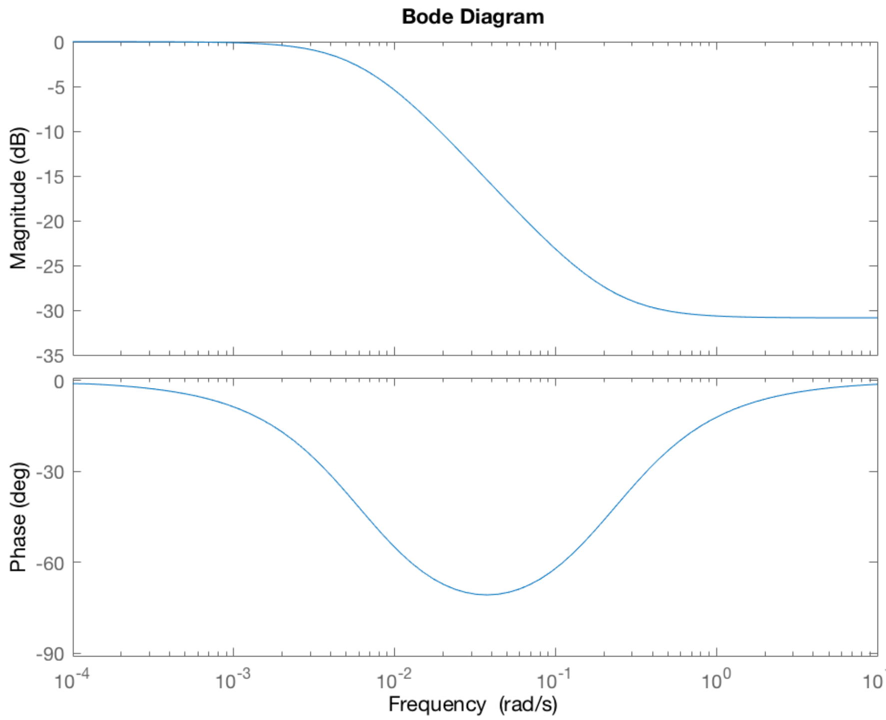


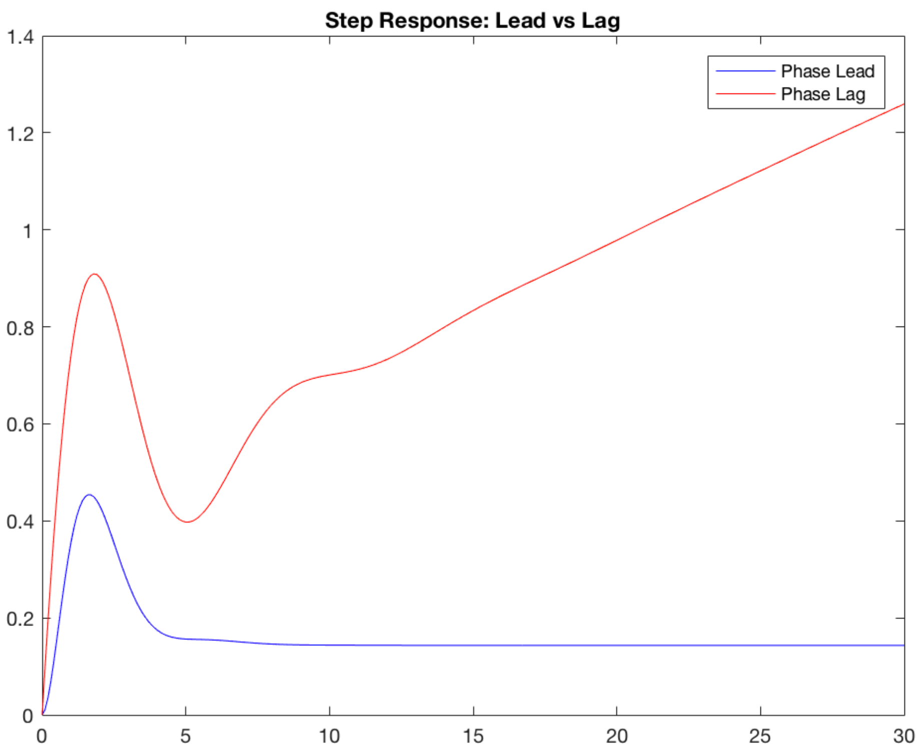

```
Gcs_lead =

  0.02865 s + 1.777
  -----------------
      s + 1.777

Continuous-time transfer function.
```

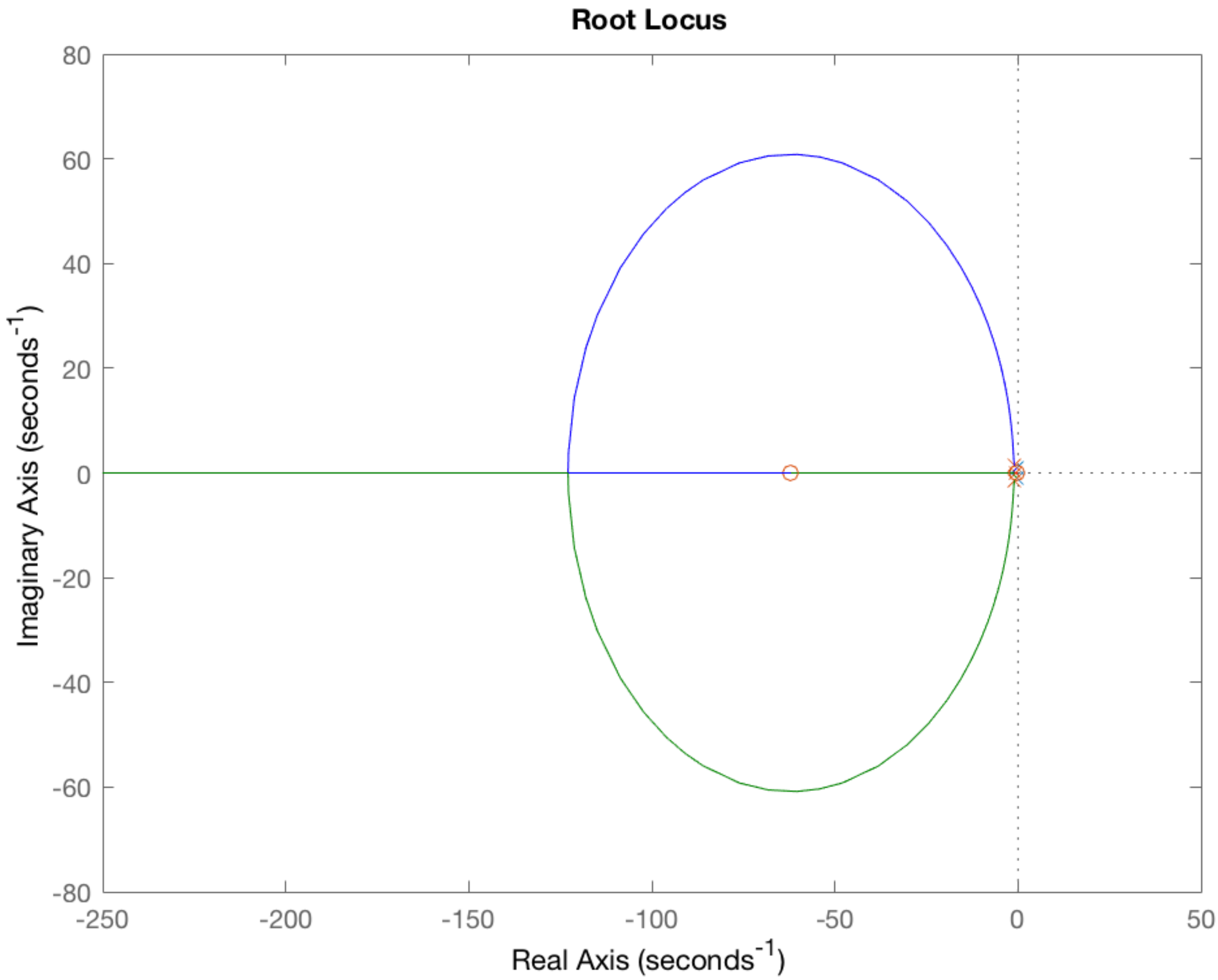
```
Gcs_lag =

  0.02865 s + 0.006342
  --------------------
      s + 0.006342

Continuous-time transfer function.
```


BodePlot: K*G
Magnitude (dB)
Phase (deg)
Frequency (rad/s)
Bode Plot of corresponding closed-loop system ramp
Magnitude (dB)
Phase (deg)
G
pc
Compensation
Frequency (rad/s)


Step Response: uncompensation vs compensation
uncompensation
compensation


Bode Diagram
Magnitude (dB)
0
-5
-10
-15
-20
-25
-30
-35
Phase (deg)
0
-30
-60
-90
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
Frequency (rad/s)


Step Response: Lead vs Lag
Phase Lead
Phase Lag
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30


Root Locus
80
60
40
20
0
-20
-40
-60
-80
-250
-200
-150
-100
-50
0
50
Real Axis (seconds^-1)
Imaginary Axis (seconds^-1)

ANALYSIS

1)
$$\frac{d\alpha(t)}{dt} = -0.313\alpha(t) + 56.7\,q(t) + 0.232\,\delta_e(t)$$
$$\frac{dq(t)}{dt} = -0.0139\alpha(t) + 0.426\,q(t) + 0.0203\,\delta_e(t)$$
$$\frac{d\theta(t)}{dt} = 56.7\,q(t)$$
motion.

Goal: Design an autopilot that controls pitch angle ($\theta(t)$) on the aircraft.

- Autopilot ÷ form a closed-loop system w/ a unity feedback & controller of the form $KG_c(S)$.
- $K = 1.151K'$
- openloop feedback transfer function $= KG_c(S)G(S)$
- openloop transfer function obtained from motion
  - $$\frac{\theta(s)}{\Delta(s)} = \frac{1.151s + 0.1774}{s^3 + 0.739s^2 + 0.921s} = \frac{1.151(s + 0.1541)}{s(s^2 + 0.739s + 0.921)} = 1.151\,G(s)$$
  - $$\Rightarrow G(s) = \frac{s + 0.1541}{s(s^2 + 0.739s + 0.921)}$$
- overshoot less than 10%

- closed loop $$= \frac{s^4 + 0.8931s^3 + 1.035s^2 + 0.1419s}{s^6 + 1.478s^5 + 3.388s^4 + 2.254s^3 + 1.8835s^2 + 0.1419s} = \frac{G(s)}{1 + G(s)}$$

steady state error

$$K_v = \lim_{s\to 0}\{sG(s)\} = \lim_{s\to 0}\left\{\frac{s(s+0.1541)K}{s(s^2+0.739s+0.921)}\right\} = \lim_{s\to 0}\left\{\frac{(s+0.1541)K}{s^2+0.739s+0.921}\right\} = \frac{0.1541K}{0.921}$$

$$K_v = \frac{0.1541\,K}{0.921}$$
$$e_{ss} = 10\% = 0.1 \rightarrow K_v = \frac{1}{0.1} = 10$$
$$10 = \frac{0.1541\,K}{0.921} \Rightarrow K = 59.7664$$

CONCLUSIONS

The main purpose of this project is to go further explore with different types of controller: Phase-Lag Controller and Phase-Lead Controller. By doing this, I could visualize what the differences between two controllers by applying them into same model. As expected, the Phase-Lead has less rising time and settling time. Phase-Lag tend to go up ward after fall but Lead does not. Therefore, I prefer the Lead because it is stable and it shows that it is well controllable the model.

Appendix: Matlab Code

```
close all;
clear;

s=tf('s');
G = (s + 0.1541) / (s*((s^2) + 0.739*s + 0.921));
opentf = 1.151*G;
oriClosedLoop = G/(1+G);

t = 0:0.1:30;
```

a

```
%steady state error
syms s
a_G = (s + 0.1541) / (s*((s^2) + 0.739*s + 0.921));
a_Kv = limit((s*a_G));
a_ess = 1/a_Kv;

%ramp response
a_y = lsim(oriClosedLoop.Numerator{1},oriClosedLoop.Denominator{1},t,t);
figure(1),plot(t,t,'b',t,a_y,'r')
xlabel('Time(sec)s');
ylabel('Amplitude');
legend('Input','Output');
title('Input: blue, Output: red');


%Refer :http://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_Ess
%Refer :http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1452&context=theses
```

b

```
b_ess = 0.1;

b_Kv = 1/b_ess;

b_zeros = roots(G.Numerator{1});

b_poles = roots(G.Denominator{1});

b_K  = (-(10*prod(b_poles(2:3)))/(b_zeros));
```

```
%corresponding closed-loop system
b_Closed = b_K*G / (1 + (b_K*G));

%ramp response
b_y = lsim(b_Closed.Numerator{1}, b_Closed.Denominator{1},t,t);
figure(2),plot(t,t,'b',t,b_y,'r')
xlabel('Time(sec)');
ylabel('Amplitude');
legend('Input','Output');
title('Input: blue, Output: red');
```

c

```
w = 0.1:0.1:100;

num = [1 0.1541];

den = [1 0.739 0.921];

figure(3);

subplot(1,2,1);

bode(b_K*num,den,w);

title('BodePlot: K*G');
[Gm_act,Pm_act,Wcg_act,Wcp_act] = margin(num,den);
phiMax = 50 - Pm_act + 10;
phiRad = (phiMax/180)*pi;
a = (1 + sin(phiRad))/(1 - sin(phiRad));
delta_G = 20 * log10(a);
subplot(1,2,2);
hold on
bode(num,den);
title('Bode Plot of corresponding closed-loop system ramp');
hold off

%constant from the book.
w_max = 10.5;
pc = w_max*sqrt(a);
numc = [a pc];
denc = [1 pc];
hold on;
bode(numc,denc);
hold off;
compNum = conv(num, numc);
compDen = conv(den, denc);
hold on;
bode(compNum, compDen);
```

```
hold off;
legend('G','pc','Compensation');

%finding closed-loop transfer function
[closedNum, closedDen] = feedback(G.Numerator{1}, G.Denominator{1},1,1,-1);
[closedNumComp, closedDenComp] = feedback(compNum,compDen,1,1,-1);

%%comments: in step response, the compensation, rising time and settling time
%%are reduced significantly.
figure(4);
ystep=step(closedNum,closedDen,t);
ystepComp_Lead = step(closedNumComp, closedDenComp,t);
plot(t,ystep,'b',t,ystepComp_Lead,'r');
%axis([0 2 0 0.5])
title('Step Response: uncompensation vs compensation');
legend('uncompensation','compensation');
s=tf('s');
Gcs_lead= (a*s + pc)/(s+pc)
```

d

```
Wcg_new= 2.1;




% dl=b_K;
% g1 = abs(j*Wcg_new);
% g2 = abs(j*Wcg_new + b_poles(2));
% g3 = abs(j*Wcg_new + b_poles(3));
% dG = dl/(g1*g2*g3)
% mag(dG)
delta_G_new = 30.4;
a_new= 10^(delta_G_new/(-20));
zc_new = Wcg_new/10;
pc_new = a_new*zc_new;
numlag = [1 zc_new];
denlag = [1 pc_new];
newnum= conv(num,numlag);
newden = conv(den,denlag);
Gcs_lag = (a*s +pc_new)/(s+pc_new)
figure(5)
bode(Gcs_lag.Numerator{1},Gcs_lag.Denominator{1})
ystepComp_Lag = step(newnum,newden,t);
figure(6)
plot(t,ystepComp_Lead,'b',t,ystepComp_Lag,'r');
%axis([0 2 0 0.5])
title('Step Response: Lead vs Lag');
legend('Phase Lead','Phase Lag');

[closedLagNum, closedLagDen] = feedback(newnum,newden,1,1,1);
```

```
%http://ece.gmu.edu/~gbeale/ece_421/comp_freq_lag.pdf
figure(7)
rlocus(num,den);
hold on
rlocus(closedNumComp,closedDenComp);
```