Assignment - IV

CHRISTOPHER OHARA (31459079)
cao36@njit.edu

May 8, 2019

Motor-Driven Pendulum - PEN 5

Pole Placement: s = -6 +/- 4j; Theta = 0

```
Transfer Function with Gain:
            1
-------------------------
 2
s  + (G2 + 4) s + G1 - 12
```

```
Transfer Function:
          1
  ---------------
  s^2 + 12 s + 52
```

```
Gain =
   64.0000    8.0000
```

Listing 1: PEN 5; s @ -6 +/- 4j; Theta @ 0

```
%% Theta = 0; s = -6+/-4j
syms s G1 G2;
A = [0 1; 12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-6+4*j, -6-4*j];
P = place(A, B, poles)

G = [G1 G2]
Ac = A-B*G
Det = inv(s*eye(2)-Ac);
deltaC = det(Det);
DeltaC = collect(deltaC, s)
pretty(DeltaC)
Ac2 = A-B*P
[a,b] = ss2tf(Ac2, B, C, D)
H = tf(a,b)

%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid
```

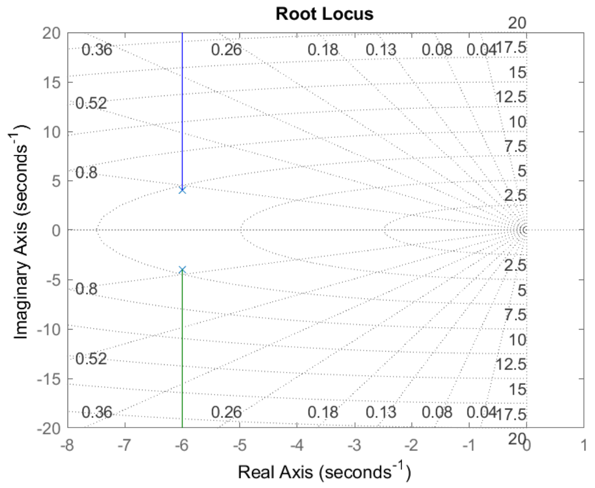

Figure 1: PEN5 - Root Locus - s = -6 +/- 4j; Theta = 0

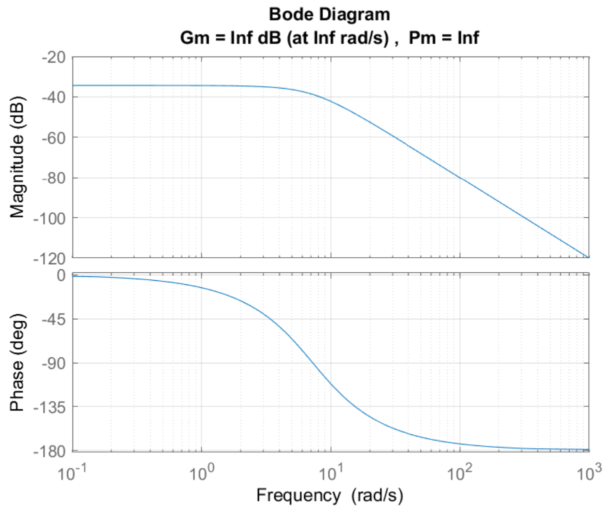

Figure 2: PEN5 - Bode - s = -6 +/- 4j; Theta = 0

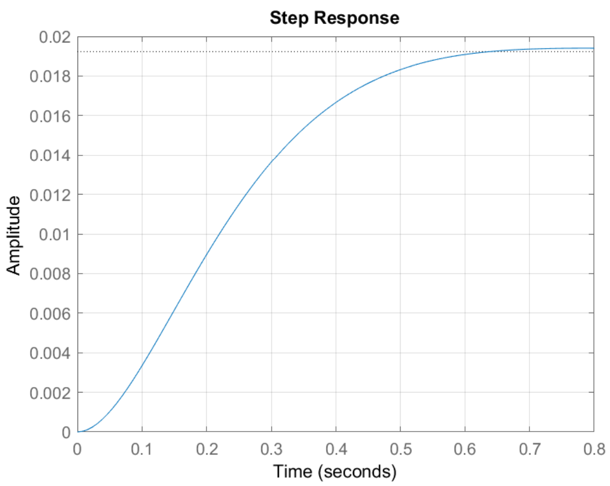

Figure 3: PEN5 - Step Response - s = -6 +/- 4j; Theta = 0

Pole Placement: s = -8,-10; Theta = 0

```
Transfer Function with Gain:
            1
-------------------------
 2
s  + (G2 + 4) s + G1 - 12
```

```
Transfer Function:
         1
  ---------------
  s^2 + 18 s + 80
```

```
Gain =
   92.0000   14.0000
```

Listing 2: PEN 5; s @ -8-10 ; Theta @ 0

```
%% Theta = 0; s = -8,-10
syms s G1 G2;
A = [0 1; 12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-8, -10];
P = place(A, B, poles)

G = [G1 G2]
Ac = A-B*G
Det = inv(s*eye(2)-Ac);
deltaC = det(Det);
DeltaC = collect(deltaC, s)
pretty(DeltaC)
Ac2 = A-B*P
[a,b] = ss2tf(Ac2, B, C, D)
H = tf(a,b)

%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid
```

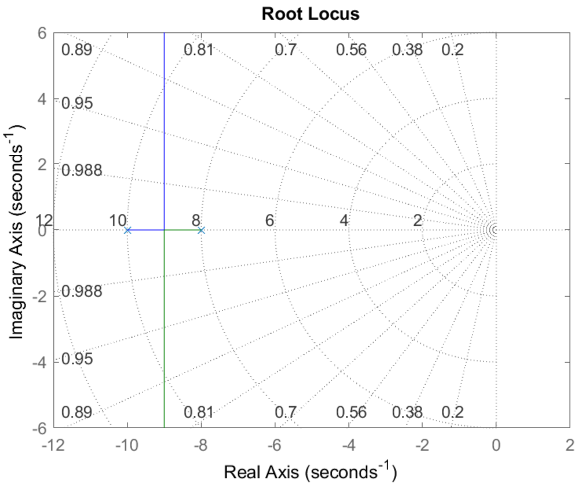

Figure 4: PEN5 - Root Locus - s = -8,-10; Theta = 0

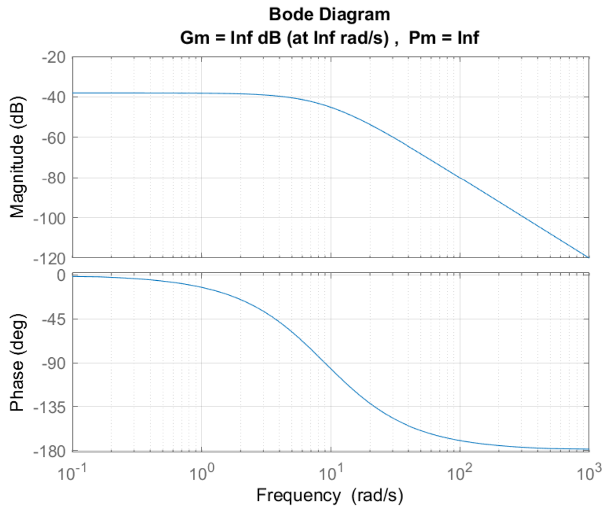

Figure 5: PEN5 - Bode - s = -8,-10; Theta = 0

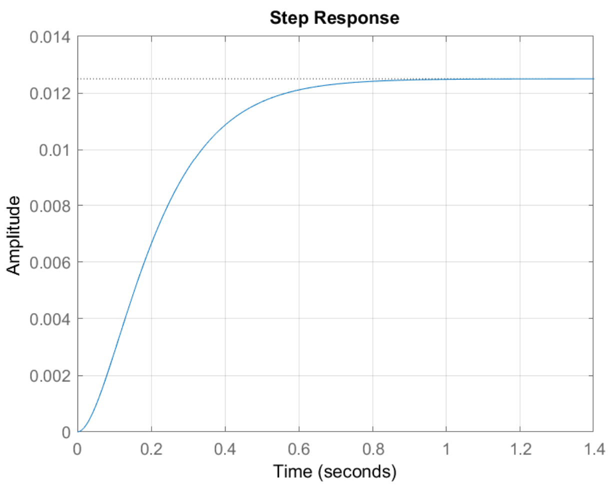

Figure 6: PEN5 - Step Response - s = -8,-10; Theta = 0

Pole Placement: s = -6 +/- 4j; Theta = 180

```
Transfer Function:
         1
  ---------------
  s^2 + 12 s + 52
```

```
Gain =
   40.0000    8.0000
```

Listing 3: PEN 5; s @ -6 +/- 4j; Theta @ 180

```
%% Theta = 180; s = -6+/-4j
syms s G1 G2;
A = [0 1; -12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-6+4*j, -6-4*j];
G = place(A, B, poles)

Ac = A-B*G
[a,b] = ss2tf(Ac, B, C, D)
H = tf(a,b)

%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid
```

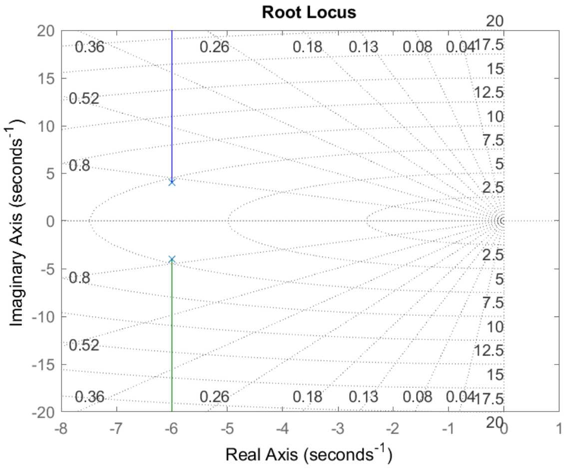

Figure 7: PEN5 - Root Locus - s = -6 +/- 4j; Theta = 180

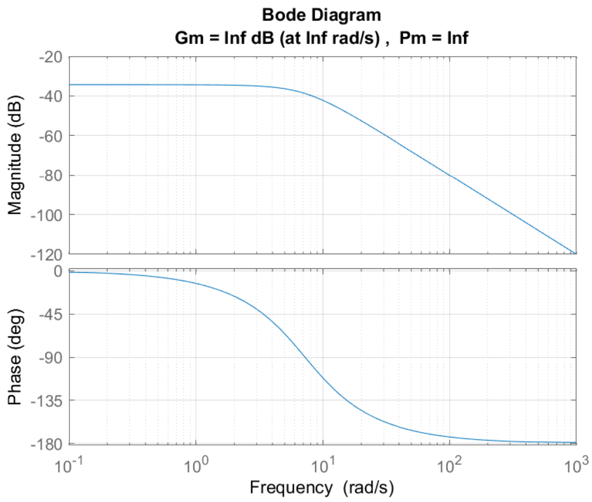

Figure 8: PEN5 - Bode - s = -6 +/- 4j; Theta = 180

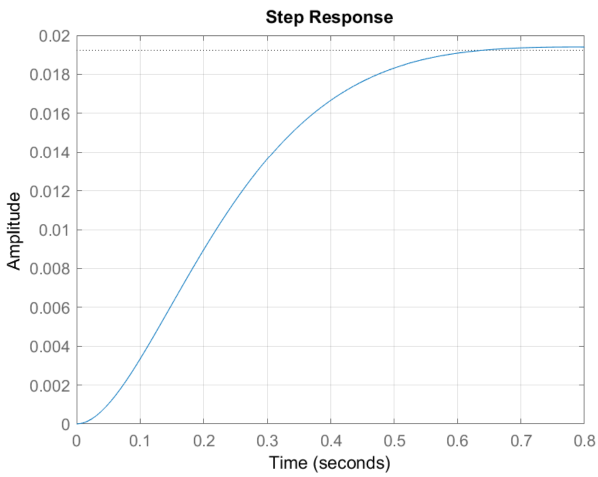

Figure 9: PEN5 - Step Response - s = -6 +/- 4j; Theta = 180

Pole Placement: s = -8,-10; Theta = 180

```
Transfer Function:
         1
  ---------------
  s^2 + 18 s + 80
```

```
Gain =
   68.0000   14.0000
```

Listing 4: PEN 5; s @ -8-10; Theta @ 180

```
%% Theta = 180; s = -8,-10
syms s G1 G2;
A = [0 1; -12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-8, -10];
G = place(A, B, poles)

Ac = A-B*G
[a,b] = ss2tf(Ac, B, C, D)
H = tf(a,b)

%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid
```

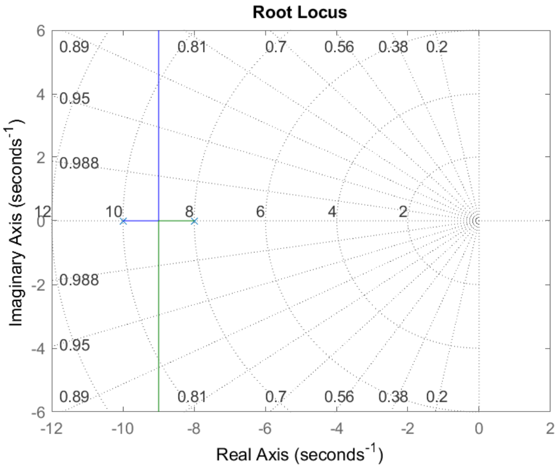

Figure 10: PEN5 - Root Locus - s = -8,-10; Theta = 180

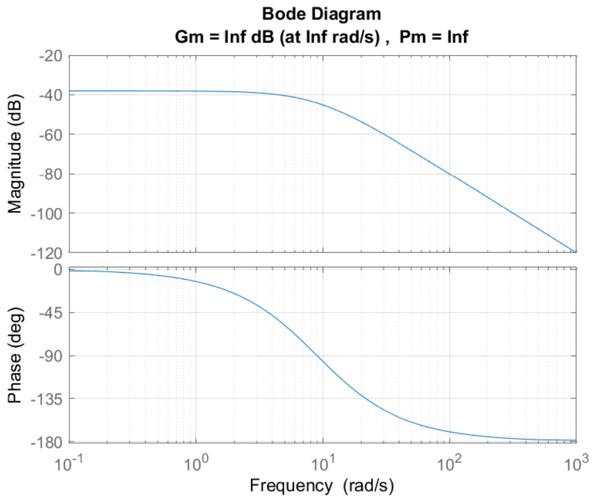

Figure 11: PEN5 - Bode - s = -8,-10; Theta = 180

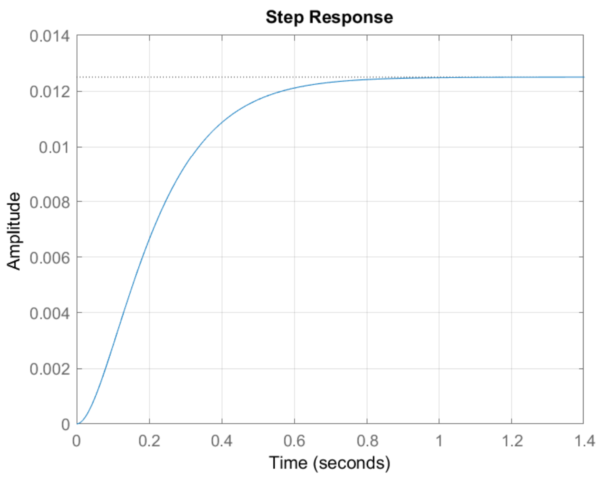

Figure 12: PEN5 - Step Response - s = -8,-10; Theta = 180

Pole Placement: s = -6 +/- 4j; Theta = 0, r_theta = 0.5

```
Transfer Function (with values equated):
      1
-------------
 2
s  + 68 s - 4
```

```
Gain =
   64.0000    8.0000
```

Listing 5: PEN 6; s @ -6 +/- 4j; Theta @ 0; r_theta @ 0.5

```
%% Theta = 0; s = -6+/-4j, ref_theta = 0.5
syms s G1 G2;
ref_theta = 0.5;
A = [0 1; 12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-6+4*j, -6-4*j];
P = place(A, B, poles);

Ac0 = A-B*P;
Det0 = inv(s*eye(2)-Ac0);
deltaC0 = det(Det0);
DeltaC0 = collect(deltaC0, s);
pretty(DeltaC0)

%% Exo
G = [G1 G2];

Ac = A-B*G;
Det = inv(s*eye(2)-Ac);
deltaC = det(Det);
DeltaC = collect(deltaC, s);
pretty(DeltaC)

%% Equating Values
G2 = [8 64]; %Analytical Compared

Ac2 = A-B*G2;
Det2 = inv(s*eye(2)-Ac2);
deltaC2 = det(Det2);
DeltaC2 = collect(deltaC2, s);
pretty(DeltaC2)

%% Control Theory Version
%G0 = inv((C*inv(A)*B))*C;
%G0 = 12
%Ac2 = A-B*G0;

%Det2 = inv(s*eye(2)-Ac2);
%deltaC2 = det(Det2);
%DeltaC2 = collect(deltaC2, s);
%pretty(DeltaC2)
%Ac2 = A-B*P

%% Transfer Function
[a,b] = ss2tf(Ac2, B, C, D);
H = tf(a,b); % Get Denominator for sisotool

%% Simulation
stateSpace = ss(Ac2, B, C, D);
t = 0:0.05:100;
u = ref_theta*ones(size(t));

lsim(stateSpace,u,t)
axis([0 10 0 .1])

```

```
%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid

%% Stable Gain Range
%s=tf('s');
%Gp= 1/(s^2 + 68*s - 4);
%sisotool(Gp)
% Unstable
```

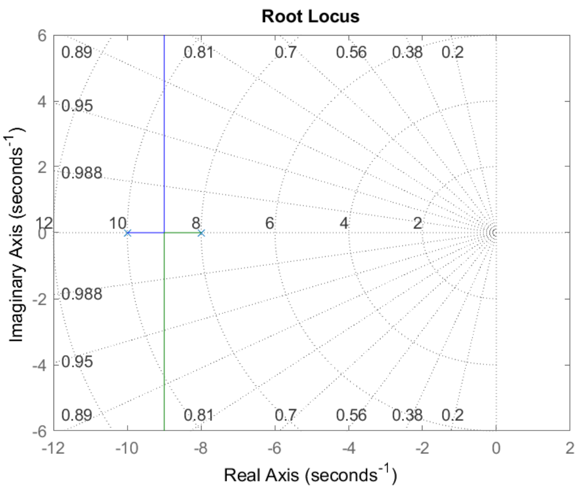

Figure 13: PEN6 - Root Locus - s = -6 +/- 4j; Theta = 0

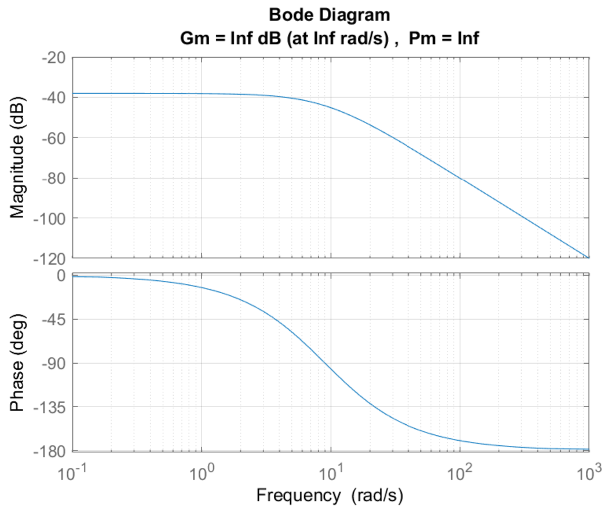

Figure 14: PEN6 - Bode - s = -6 +/- 4j; Theta = 0

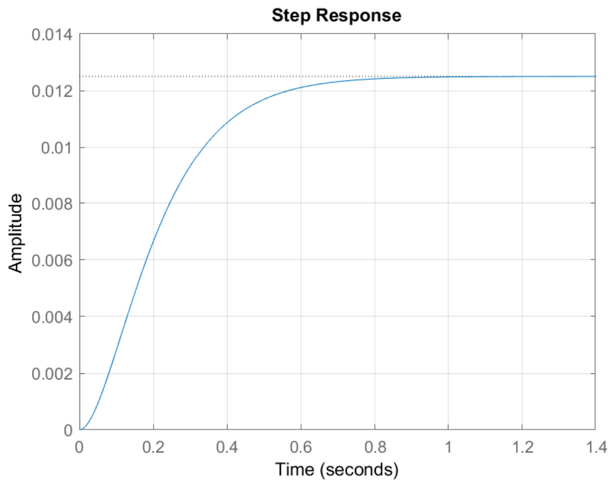

Figure 15: PEN6 - Step Response - s = -6 +/- 4j; Theta = 0

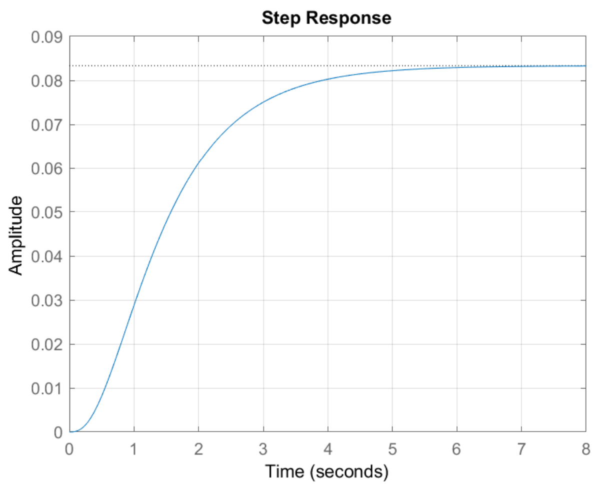

Figure 16: PEN6 - Linear Simulation - s = -6 +/- 4j; Theta = 0

Pole Placement: s = -8,-10; Theta = 0, r_theta = 0.5

```
Transfer Function:
      1
-------------
 2
s  + 96 s + 2
```

```
Gain =
   92.0000   14.0000
```

Listing 6: PEN 6; s @ -8-10; Theta @ 0

```
%% Theta = 0; s = -8,-10, ref_theta = 0.5
syms s G1 G2;
ref_theta = 0.5;
A = [0 1; 12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-8, -10];
P = place(A, B, poles);

Ac0 = A-B*P;
Det0 = inv(s*eye(2)-Ac0);
deltaC0 = det(Det0);
DeltaC0 = collect(deltaC0, s);
pretty(DeltaC0)

%% Exo
G = [G1 G2];

Ac = A-B*G;
Det = inv(s*eye(2)-Ac);
deltaC = det(Det);
DeltaC = collect(deltaC, s);
pretty(DeltaC)

%% Equating Values
G = [14 92]; %Analytical Compared

Ac2 = A-B*G2;
Det2 = inv(s*eye(2)-Ac2);
deltaC2 = det(Det2);
DeltaC2 = collect(deltaC2, s);
pretty(DeltaC2)

%% Control Theory Version
%G0 = inv((C*inv(A)*B))*C;
%G0 = -12
%Ac2 = A-B*G0;

%Det2 = inv(s*eye(2)-Ac2);
%deltaC2 = det(Det2);
%DeltaC2 = collect(deltaC2, s);
%pretty(DeltaC2)
%Ac2 = A-B*P

%% Transfer Function
[a,b] = ss2tf(Ac2, B, C, D);
H = tf(a,b); % Get Denominator for sisotool

%% Simulation
stateSpace = ss(Ac2, B, C, D);
t = 0:0.05:100;
u = ref_theta*ones(size(t));

lsim(stateSpace,u,t)
axis([0 10 0 .1])

```

```
%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid

%% Stable Gain Range
%s=tf('s');
%Gp= 1/(s^2 + 96*s + 2);
%sisotool(Gp)
%Stable
```

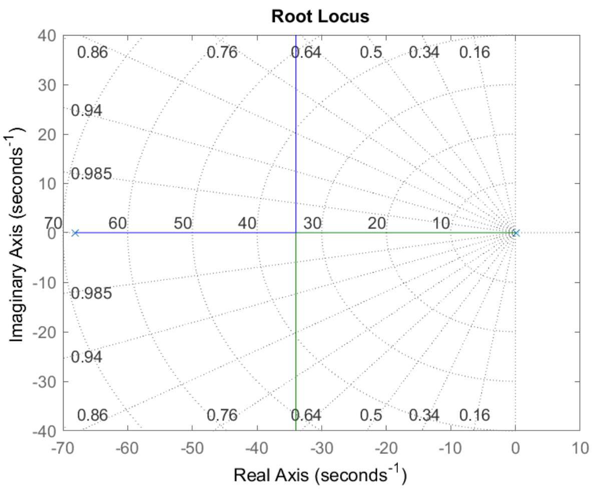

Figure 17: PEN6 - Root Locus - s = -8,-10; Theta = 0, r_theta = 0.5

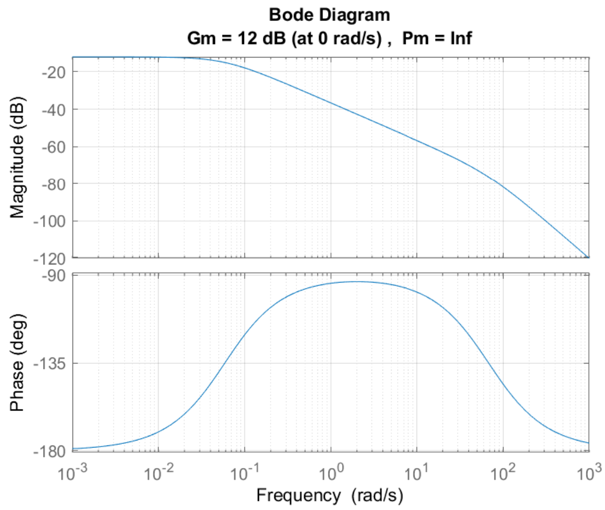

Figure 18: PEN6 - Bode - s = -8,-10; Theta = 0, r_theta = 0.5

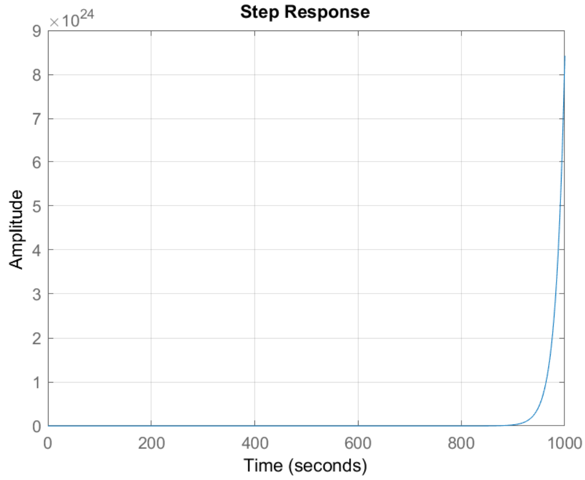

Figure 19: PEN6 - Step Response - s = -8,-10; Theta = 0, r_theta = 0.5

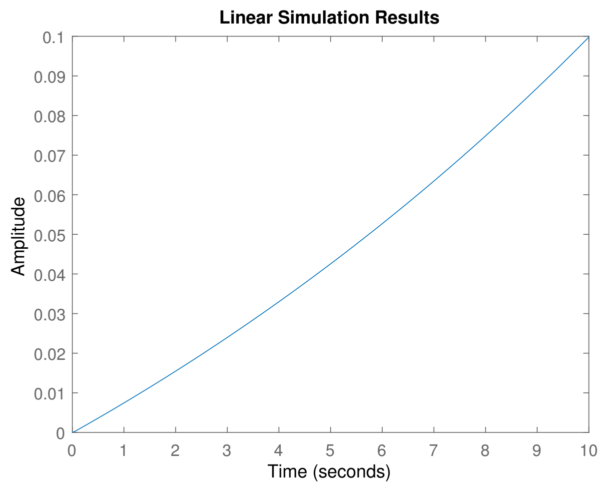

Figure 20: PEN6 - Linear Simulation - s = -8,-10; Theta = 0, r_theta = 0.5

Pole Placement: s = -6 +/- 4j; Theta = 180, r_theta = 0.5

```
Transfer Function (with values equated):
       1
--------------
 2
s  + 44 s + 20
```

```
Gain =
   40.0000    8.0000
```

Listing 7: PEN 6; s @ -6 +/- 4j; Theta @ 180

```
%% Theta = 0; s = -6+/-4j, ref_theta = 0.5
syms s G1 G2;
ref_theta = 0.5;
A = [0 1; -12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-6+4*j, -6-4*j];
P = place(A, B, poles);

Ac0 = A-B*P;
Det0 = inv(s*eye(2)-Ac0);
deltaC0 = det(Det0);
DeltaC0 = collect(deltaC0, s);
pretty(DeltaC0)

%% Exo
G = [G1 G2];

Ac = A-B*G;
Det = inv(s*eye(2)-Ac);
deltaC = det(Det);
DeltaC = collect(deltaC, s);
pretty(DeltaC)

%% Equating Values
G2 = [8 40]; %Analytical Compared

Ac2 = A-B*G2;
Det2 = inv(s*eye(2)-Ac2);
deltaC2 = det(Det2);
DeltaC2 = collect(deltaC2, s);
pretty(DeltaC2)

%% Control Theory Version
%G0 = inv((C*inv(A)*B))*C;
%G0 = 12
%Ac2 = A-B*G0;

%Det2 = inv(s*eye(2)-Ac2);
%deltaC2 = det(Det2);
%DeltaC2 = collect(deltaC2, s);
%pretty(DeltaC2)
%Ac2 = A-B*P

%% Transfer Function
[a,b] = ss2tf(Ac2, B, C, D);
H = tf(a,b); % Get Denominator for sisotool

%% Simulation
stateSpace = ss(Ac2, B, C, D);
t = 0:0.05:100;
u = ref_theta*ones(size(t));

lsim(stateSpace,u,t)
axis([0 10 0 .1])

```

```
%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid

%% Stable Gain Range
%s=tf('s');
%Gp= 1/(s^2 + 44*s + 20);
%sisotool(Gp)
%Stable
```

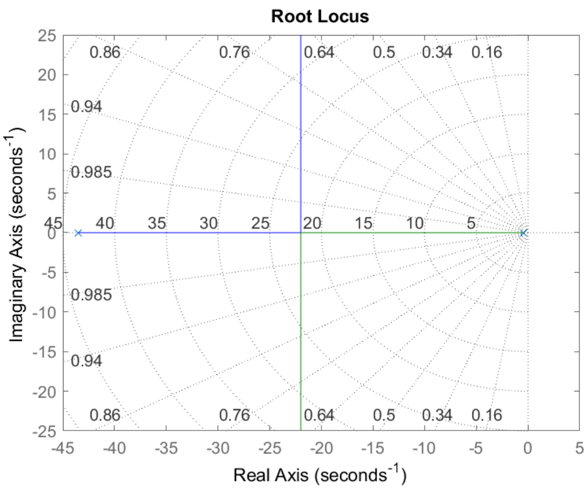

Figure 21: PEN6 - Root Locus - s = -6 +/- 4j; Theta = 180

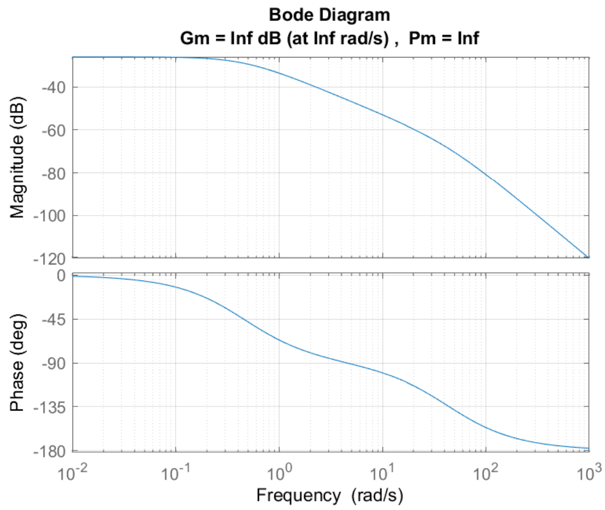

Figure 22: PEN6 - Bode - s = -6 +/- 4j; Theta = 180

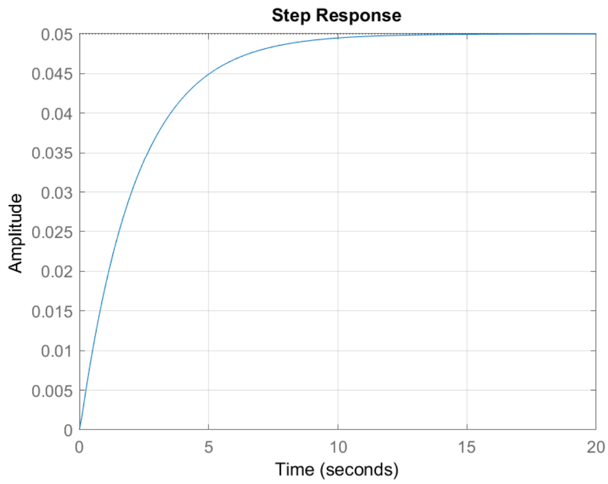

Figure 23: PEN6 - Step Response - s = -6 +/- 4j; Theta = 180

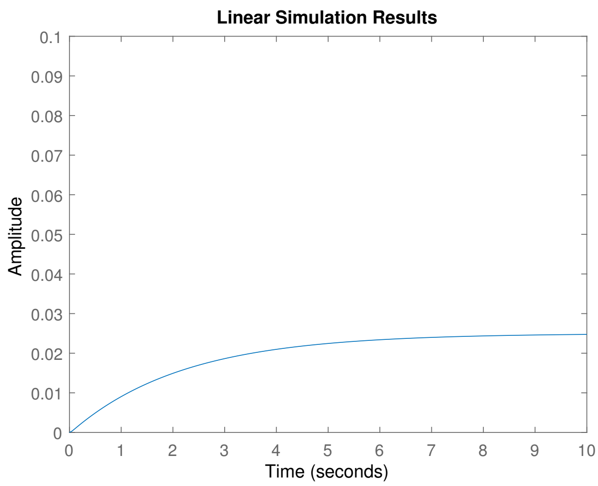

Figure 24: PEN6 - Linear Simulation - s = -6 +/- 4j; Theta = 180

Pole Placement: s = -8,-10; Theta = 180, r_theta = 0.5

```
Transfer Function (with values equated):
       1
-------------
 2
s  + 96 s + 2
```

```
Gain =
   92.0000   14.0000
```

Listing 8: PEN 6; s @ -8-10; Theta @ 180

```
%% Theta = 180; s = -8,-10, ref_theta = 0.5
syms s G1 G2;
ref_theta = 0.5;
A = [0 1; -12 -4];
B = [0; 1];
C = [1 0];
D = 0;

%% Place Poles and Gain
poles = [-8, -10];
P = place(A, B, poles);

Ac0 = A-B*P;
Det0 = inv(s*eye(2)-Ac0);
deltaC0 = det(Det0);
DeltaC0 = collect(deltaC0, s);
pretty(DeltaC0)

%% Exo
G = [G1 G2];

Ac = A-B*G;
Det = inv(s*eye(2)-Ac);
deltaC = det(Det);
DeltaC = collect(deltaC, s);
pretty(DeltaC)

%% Equating Values
G2 = [14 68]; %Analytical Compared

Ac2 = A-B*G2;
Det2 = inv(s*eye(2)-Ac2);
deltaC2 = det(Det2);
DeltaC2 = collect(deltaC2, s);
pretty(DeltaC2)

%% Control Theory Version
%G0 = inv((C*inv(A)*B))*C;
%G0 = 12
%Ac2 = A-B*G0;

%Det2 = inv(s*eye(2)-Ac2);
%deltaC2 = det(Det2);
%DeltaC2 = collect(deltaC2, s);
%pretty(DeltaC2)
%Ac2 = A-B*P

%% Transfer Function
[a,b] = ss2tf(Ac2, B, C, D);
H = tf(a,b); % Get Denominator for sisotool

%% Simulation
stateSpace = ss(Ac2, B, C, D);
t = 0:0.05:100;
u = ref_theta*ones(size(t));

lsim(stateSpace,u,t)
axis([0 10 0 .1])
```

```
%% Root Locus
figure(1)
plot(1,2)
rlocus(H), grid

%% Nyquist
%figure(2)
%plot(1,2)
%nyquist(H), grid

%% Bode
figure(3)
plot(1,2)
margin(H), grid

%% Step
figure(4)
plot(1,2)
step(H), grid

%% Stable Gain Range
%s=tf('s');
%Gp= 1/(s^2 + 72*s + 26);
%sisotool(Gp)
% Stable
```

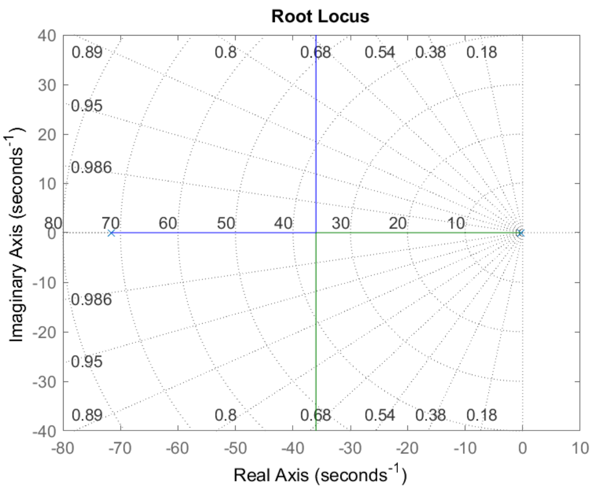

Figure 25: PEN6 - Root Locus - s = -8,-10; Theta = 180, r_theta = 0.5

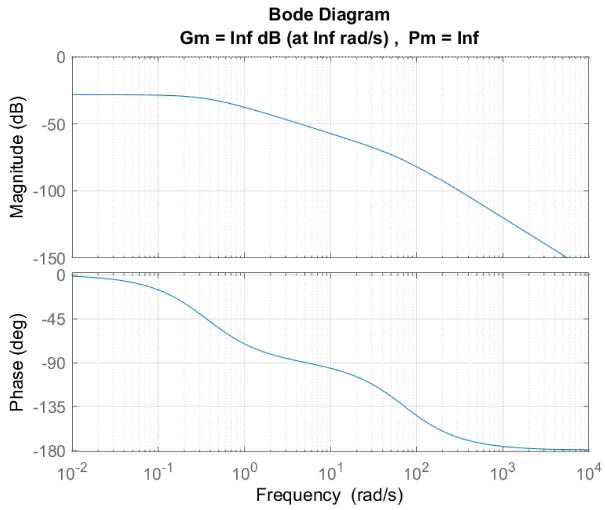

Figure 26: PEN6 - Bode - s = -8,-10; Theta = 180, r_theta = 0.5

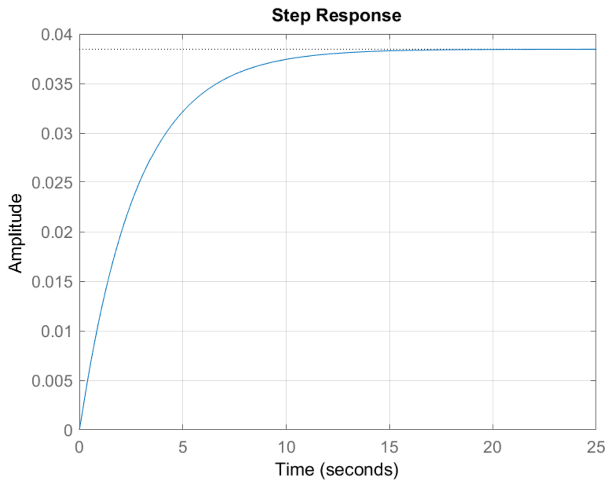

Figure 27: PEN6 - Step Response - s = -8,-10; Theta = 180, r_theta = 0.5

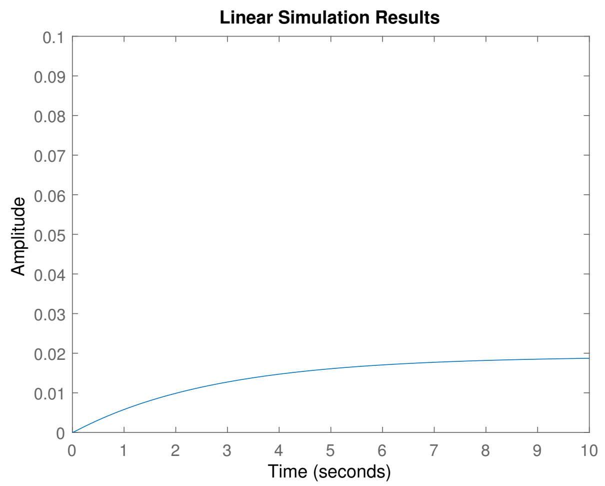

Figure 28: PEN6 - Linear Simulation - s = -8,-10; Theta = 180, r_theta = 0.5

Third-Order Heat Conduction - TH3-4

Using Rowth-Hurwitz Criterion $\rightarrow \frac{(G_P s+G_I)}{s^4+8s^3+19s^2+(12+G_p)s+G_I} = 0$

S^4	1	19	G_I
S^3	8	$12+G_P$	0
S^2	$\frac{140-G_P}{8}$	G_I	0
S^1	$\frac{(140-G_P)(12+G_P)+64G_I}{G_P-140}$		
S^0	G_I		

As previously derived with *sisotool* in the previous application: $\rightarrow G_P < 140 \wedge G_I > 0$
Using *sisotool* to plot various gain values:

```
Transfer Function (with values equated):
             1
  -----------------------
  s^3 + 8 s^2 + 19 s + 12
```

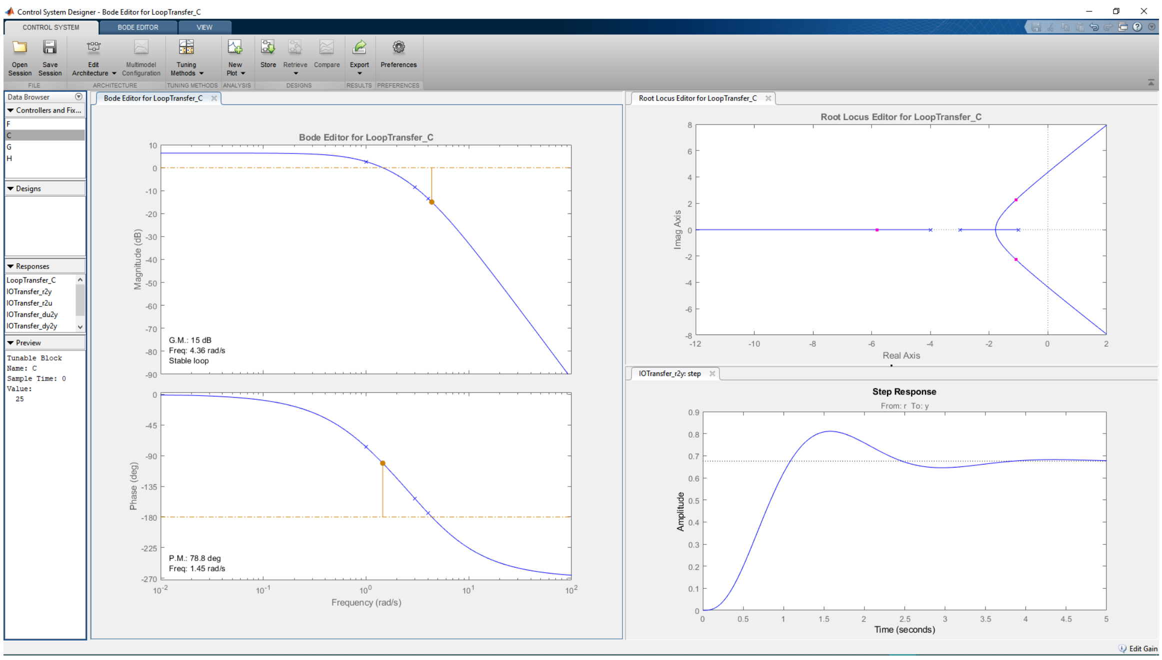

Figure 29: TH3 4 - $G_P, G_I = 25$

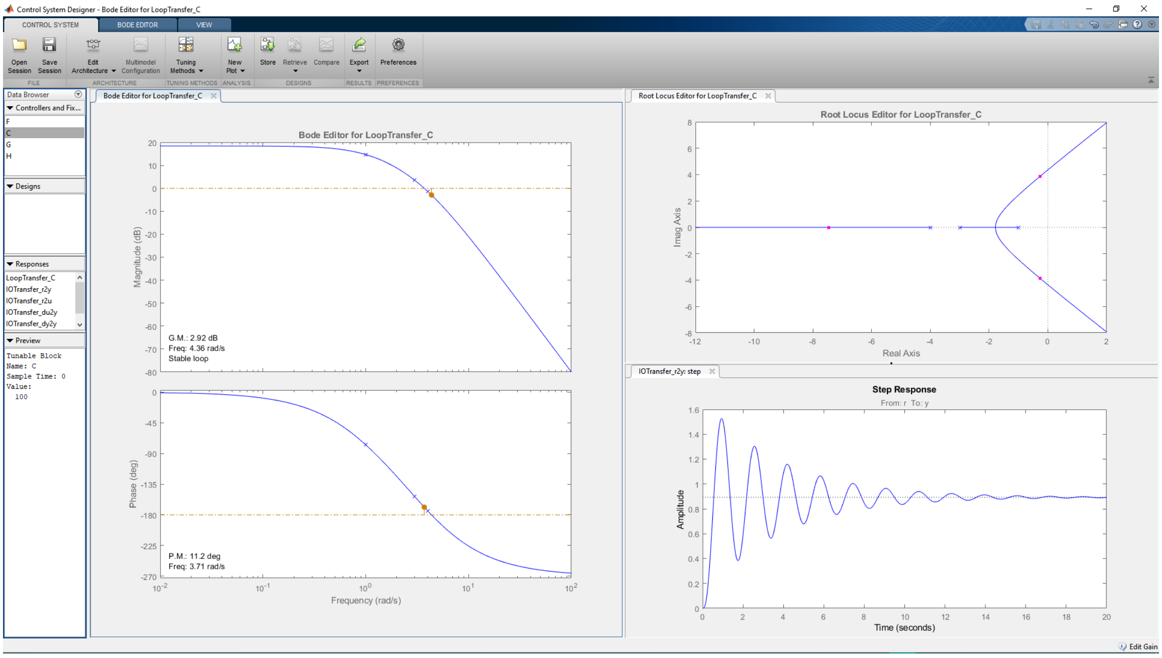

Figure 30: TH3 4 - $G_P, G_I = 100$

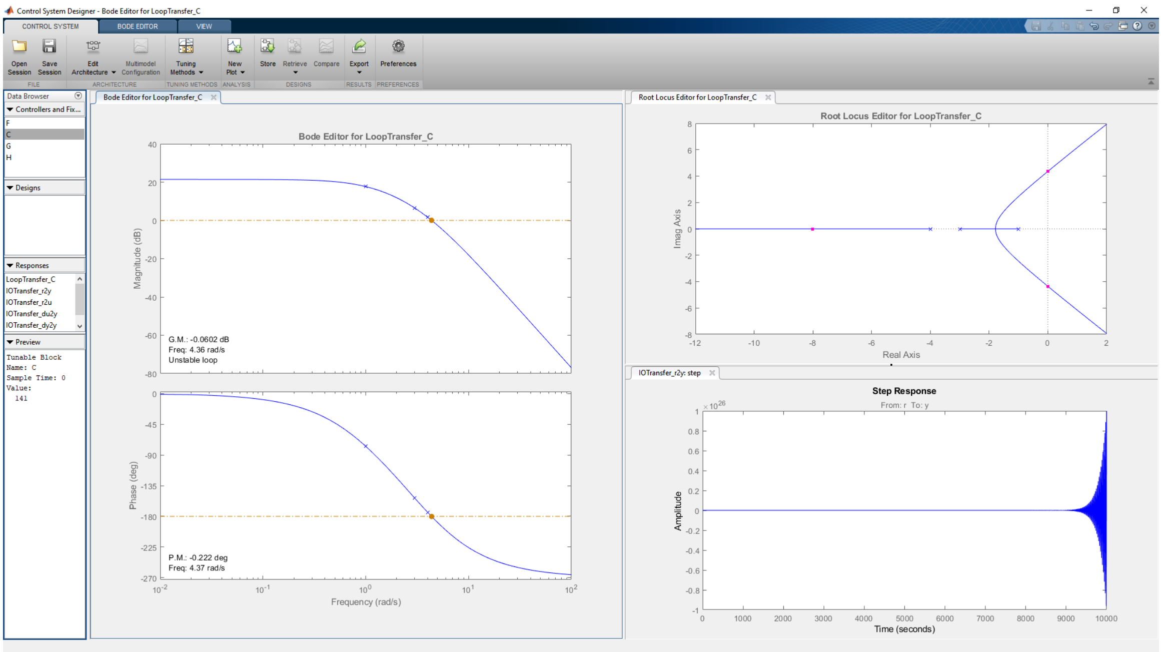

Figure 31: TH3 4 - $G_P, G_I = 141$

```
%% Stable Gain Range
s=tf('s')
Gp= 1/(s^3 + 8*s^2 + 19*s + 12)
sisotool(Gp)
% Stability Analysis from Root Locus:
% Scanning the imag axis when the real axis is 0 and -8, values between 0 and 140 are
    stable
```

Listing 9: TH3

```
%% Control Law
% U(s) = -(Gp+Gi/s)*(Y(s)-Yr(s))

A = [-3 1 0;1 -2 1;0 1 -3];
B = [1; 0; 0];
C = [0 0 1];
D = 0;
eig(A)

H = ss(A,B,C,D)
tf(H)
%Gm = margin(H)

%% Stable Gain Range
s=tf('s')
Gp= 1/(s^3 + 8*s^2 + 19*s + 12)
sisotool(Gp)
% Stability Analysis from Root Locus:
% Scanning the imag axis when the real axis is 0 and -8, values between 0 and 140 are
   stable
```

Pendulum on Cart - PCA 5

Control Law: $u = -G_1 x_1 - G_2 x_2$

Listing 10: PCA

```
syms s a b m M g L G1 G2
%A = [0 0 1 0; 0 0 0 1; 0 -m*g/M -a 0; 0 (M+m)*g/(M*L) a/L 0];
%B = [0;0;b;-b/L];
%a = 4; b = 1; M = 1; m = 0.4; g = 9.81; L = 0.25*M;

%% Control Law
% u = -G1*x1-G2*x2

%% Previously calculated ss values
A = [0 0 1 0; 0 0 0 1; 0 -3.92 -4 0; 0 54.88 16 0];
B = [0 0 1 -4]';

G = [G1 G2 0 0];
Ac = A-B*G;

Delta_c = det(s*eye(4)-Ac);
detDelta_C = collect(Delta_c,s);
pretty(detDelta_C)
```

```

 4       3    /               1372 \   2    784 s    196 G1
s  + 4 s  + | G1 - 4 G2 - ---- | s  - ----- - ------
            \                 25   /         5          5
```

$$\Rightarrow s^4 + 4s^3 + (G_1 - 4G_2 - \frac{1372}{25})s^2 + \frac{784}{5}s - \frac{196G_1}{5}$$

Since the lead coefficients are alternate, this control law cannot stabilize the system (regardless of arbitrary assignments to $G_1 \wedge G_2$).

References

[1] B. Friedland, Observer-Based Control System Design Lecture Notes for ECE660.

[2] B. Friedland, Control System Design: An Introduction to State Space Methods, McGraw-Hill, 1985. ISBN:0070224412 (Reprinted by Dover Publications May 2005, ISBN: 0-486-44278-0.)