ECE 414 - Frequency Response - Josh Andrews

Table of Contents

Use PIDtune to find different controllers

by using pidtuneOptions, various design foci and phase margins can be set

```
clear all; clc;

G = tf(40, [1 30 200 0]);
```

Find different design focus controllers for different PM

Starting with balanced design focus

```
for i = 30:1:90
    j = i-29;

    opts = pidtuneOptions('PhaseMargin',
 i, 'DesignFocus', 'balanced');
    C = pidtune(G, 'PDF', opts);

    % calculate open/closed loop and controller ffort
    L(j) = G*C;
    T(j) = L(j)/(1+L(j));
    U(j) = C/(1+L(j));

    %get info from margins
    info = margins(L(j));
    PMb(j) = info.Pm;
```

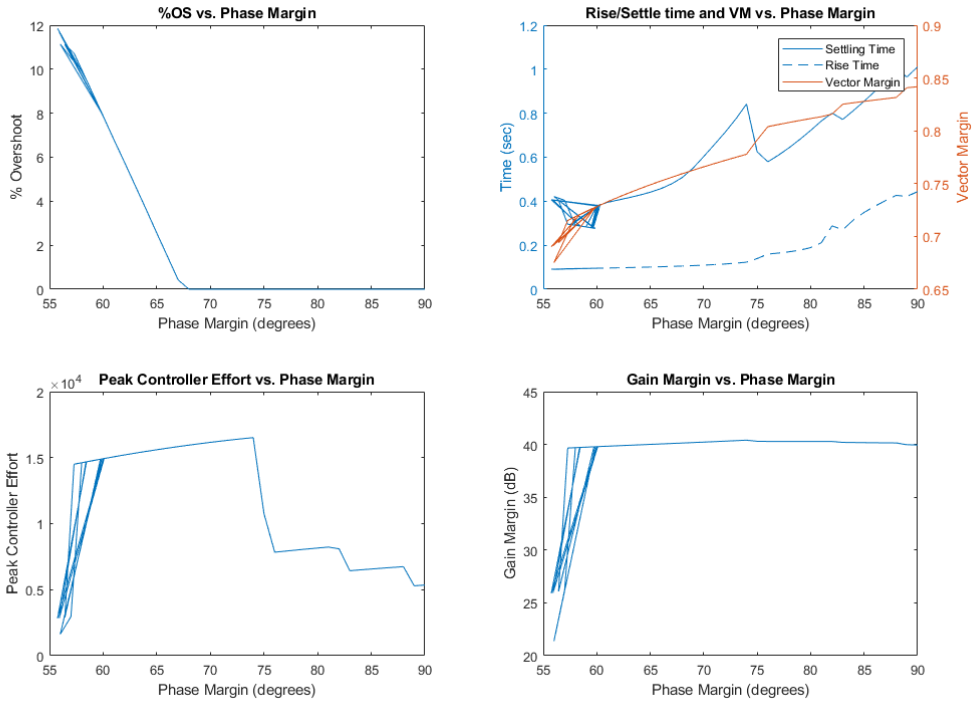
```
    GMb(j) = info.Gm;
    VMb(j) = info.Vm;

    %get info from step response
    info = stepinfo(T(j));
    trb(j)= info.RiseTime;
    OSb(j) = info.Overshoot;
    tsb(j) = info.SettlingTime;

    %get controller effort
    info = stepinfo(U(j));
    CEb(j) = info.Peak;
end
```

Plot the Balanced Controller Data

The data received when trying to set a phase margin below 60 for the balanced design resulted in invalid data as can be seen in the graphs below. The phase margin never dropped below 56 and the values of each parameter flucuated wildly.

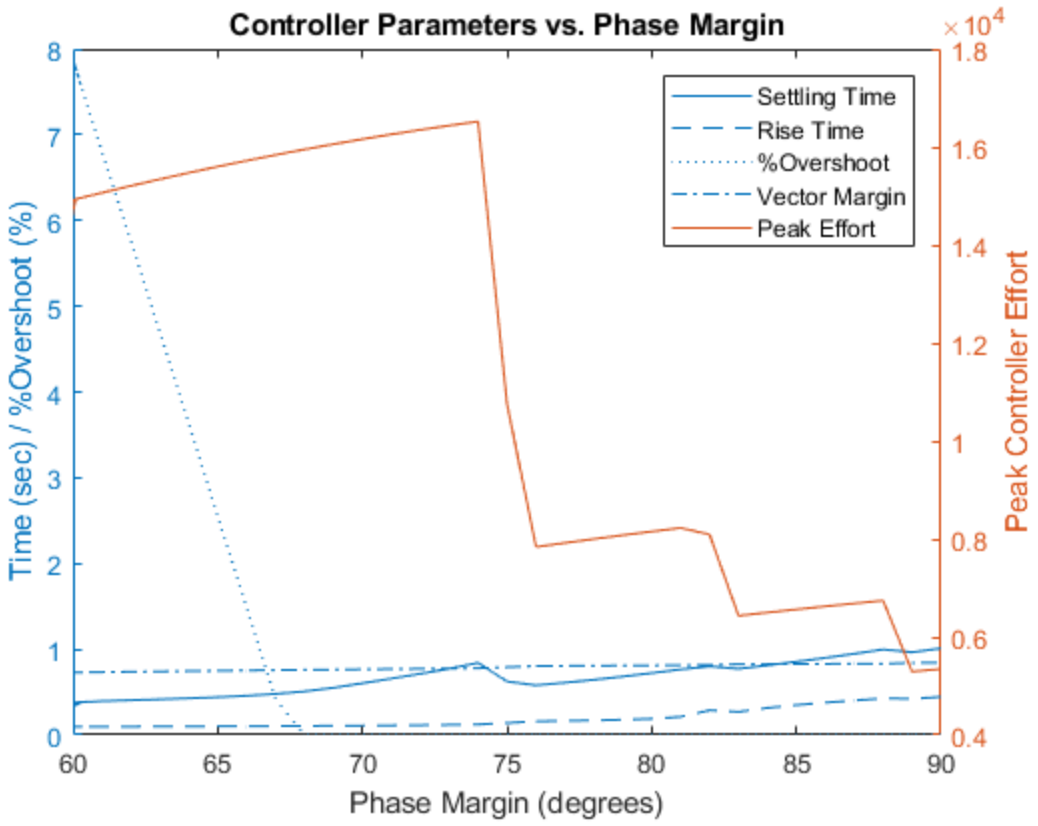
```
figure(1); clf;

subplot(2,2,1)
plot(PMb, OSb)
xlim([55 90])
xlabel('Phase Margin (degrees)')
ylabel('% Overshoot')
title('%OS vs. Phase Margin')

subplot(2,2,2)
yyaxis left
plot(PMb, tsb, PMb, trb)
xlim([55 90])
xlabel('Phase Margin (degrees)')
ylabel('Time (sec)')
yyaxis right
plot(PMb, VMb)
ylabel('Vector Margin')
title('Rise/Settle time and VM vs. Phase Margin')
legend('Settling Time', 'Rise Time', 'Vector Margin')

subplot(2,2,3)
plot(PMb, CEb)
xlim([55 90])
xlabel('Phase Margin (degrees)')
ylabel('Peak Controller Effort')
title('Peak Controller Effort vs. Phase Margin')

subplot(2,2,4)
plot(PMb, GMb)
xlim([55 90])
xlabel('Phase Margin (degrees)')
ylabel('Gain Margin (dB)')
```

```
title('Gain Margin vs. Phase Margin')
```



The gain margin stays fairly constant around 40dB for a phase margin from 60-90 degrees, so lets look at the others. Putting the rest on the same plot will allow for easier comparison. Will also disregard data for phase margin below 60 degrees because of the invalid data (it shouldn't jump like that)

```
figure(2); clf;
yyaxis left
plot(PMb, tsb, PMb, trb, PMb, OSb, PMb, VMb)
xlim([60 90])
xlabel('Phase Margin (degrees)')
ylabel('Time (sec) / %Overshoot (%)')
yyaxis right
plot(PMb, CEb)
ylabel('Peak Controller Effort')
title('Controller Parameters vs. Phase Margin')
legend('Settling Time', 'Rise Time', '%Overshoot','Vector
 Margin', 'Peak Effort')
```

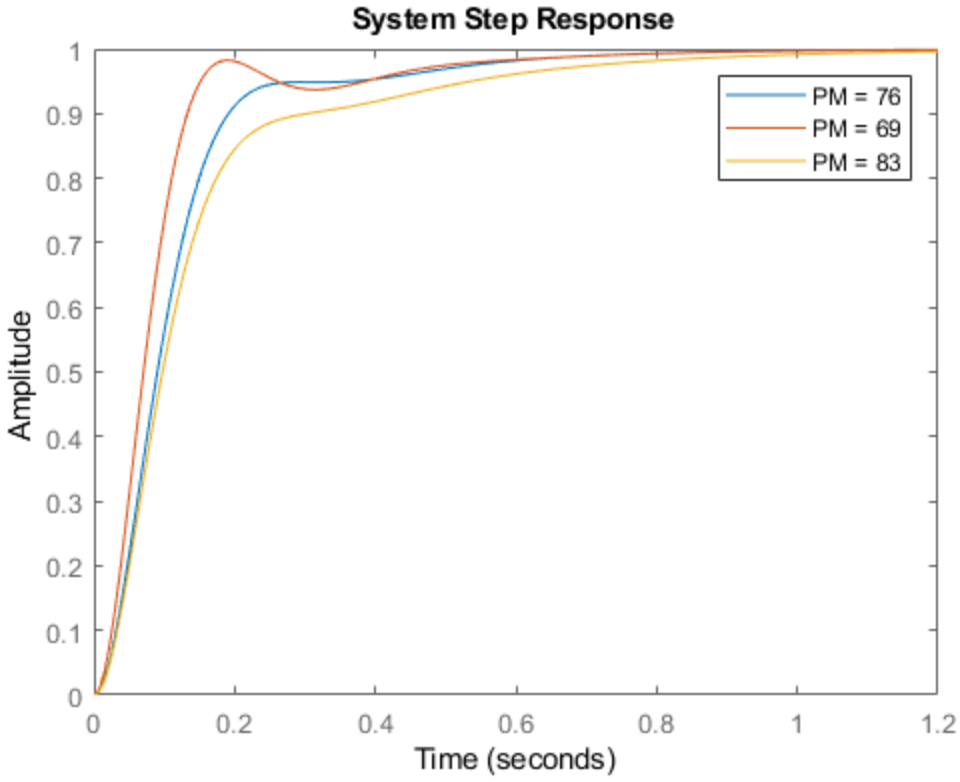


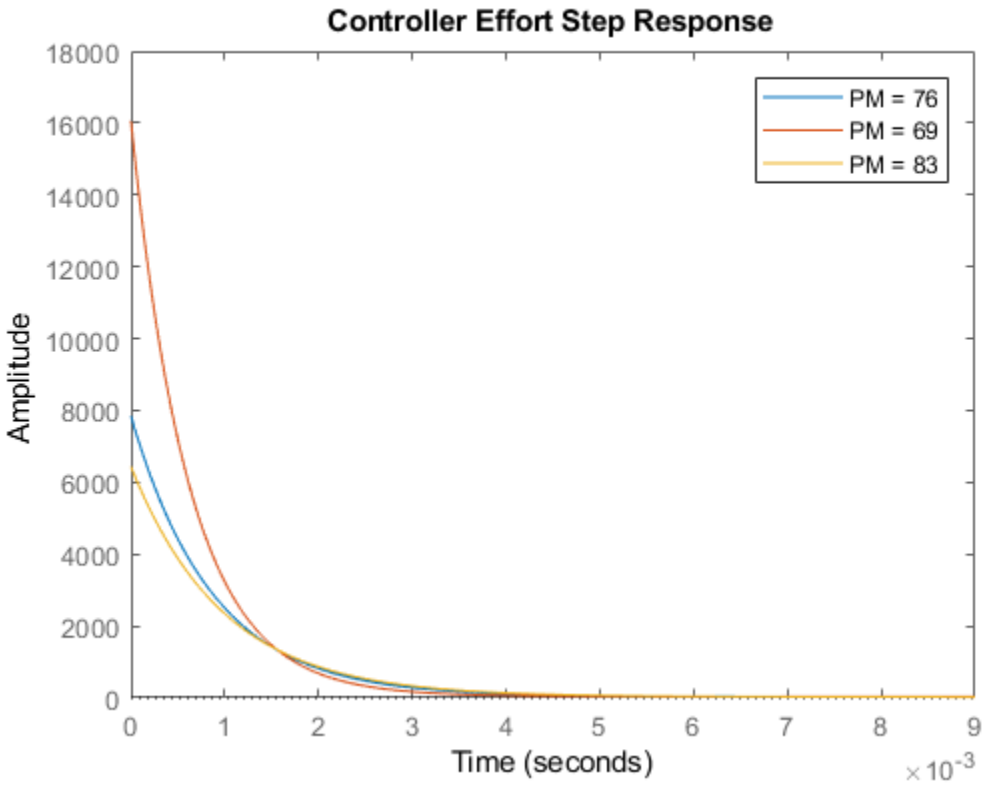
Looking at the figure above, the phase margin for which all parameters can be minimized with respect to one another is 76 degrees

Step Response Confirmation

Testing a few step responses around 76 degrees should show how well it performs.

```
figure(3); clf;
step(T(47), T(40), T(54))
legend('PM = 76', 'PM = 69', 'PM = 83')
title('System Step Response')

figure(4); clf;
step(U(47), U(40), U(54))
legend('PM = 76', 'PM = 69', 'PM = 83')
title('Controller Effort Step Response')
```


System Step Response
Amplitude
Time (seconds)
PM = 76
PM = 69
PM = 83
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.4
0.6
0.8
1
1.2

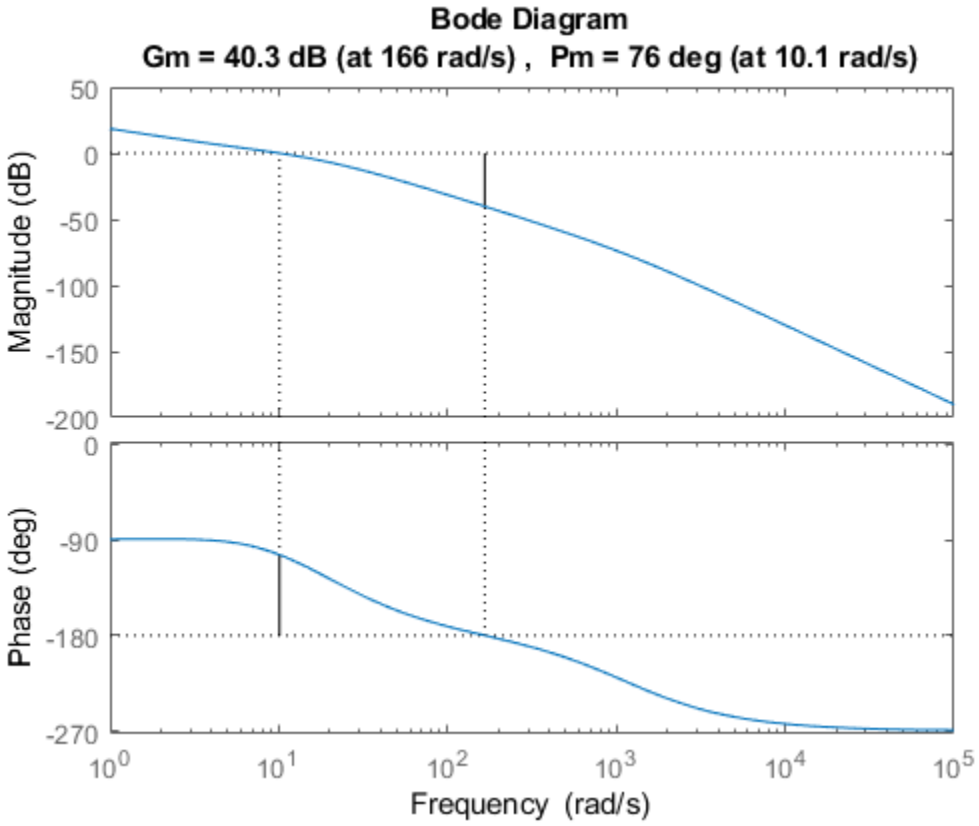

Controller Effort Step Response
Amplitude
Time (seconds)
PM = 76
PM = 69
PM = 83
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
1
2
3
4
5
6
7
8
9
×10-3

We can see that compared to some of the other controllers we have designed, this one takes a fairly long time to get the output up to the desired value of 1. All three phase margins have similar rise and settle times. The controller effort has to be considered as well, and therefore because rise and settle time only differ only slighly, picking the lowest controller effort in the range, PM = 76, is the best choice.

Bode Plot for balanced design

With the desired phase margin found, lets look at the bode plot for it

```
figure(5); clf;
margin(L(47))
```

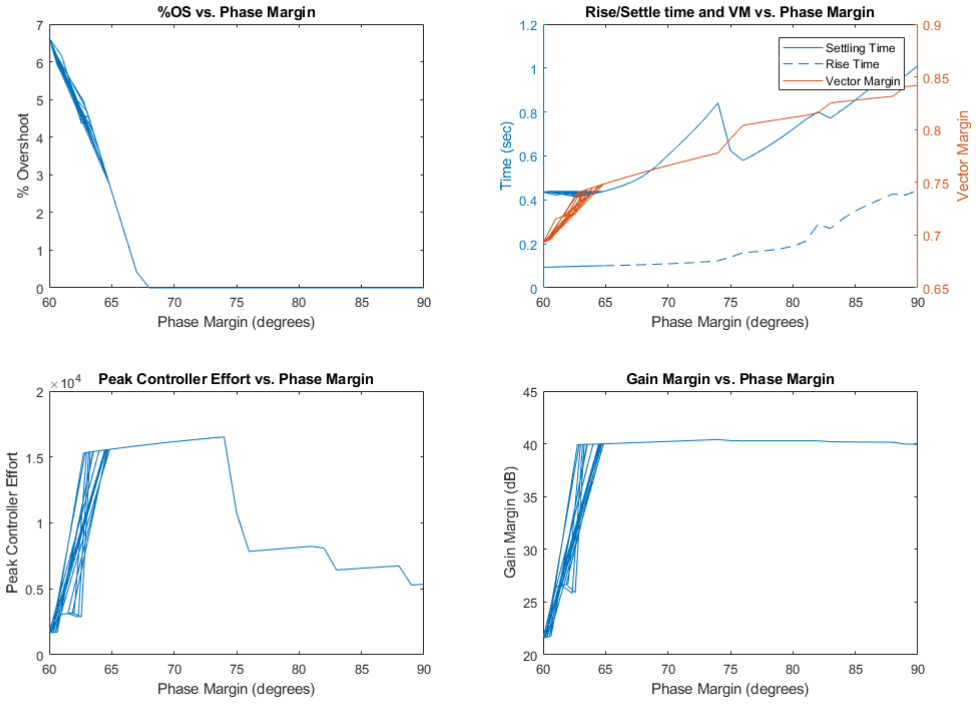


Now lets try reference tracking

repeat the same steps as above but for a reference tracking design focus

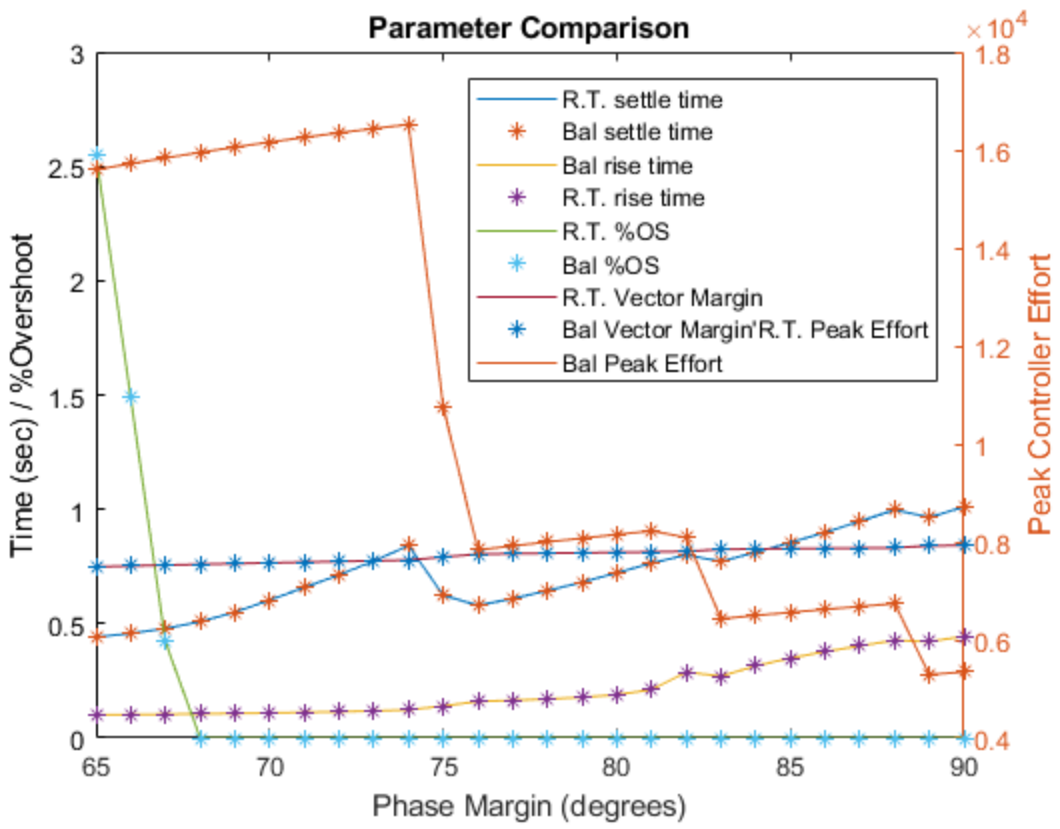
```
for i = 30:1:90
    j = i-29;

    opts = pidtuneOptions('PhaseMargin', i, 'DesignFocus', 'reference-
tracking');
    C = pidtune(G, 'PDF', opts);

    % calculate open/closed loop and controller ffort
    L(j) = G*C;
```

```
    T(j) = L(j)/(1+L(j));
    U(j) = C/(1+L(j));

    %get info from margins
    info = margins(L(j));
    PMr(j) = info.Pm;
    GMr(j) = info.Gm;
    VMr(j) = info.Vm;

    %get info from step response
    info = stepinfo(T(j));
    trr(j)= info.RiseTime;
    OSr(j) = info.Overshoot;
    tsr(j) = info.SettlingTime;

    %get controller effort
    info = stepinfo(U(j));
    CEr(j) = info.Peak;
end

figure(6); clf;

subplot(2,2,1)
plot(PMr, OSr)
xlim([60 90])
xlabel('Phase Margin (degrees)')
ylabel('% Overshoot')
title('%OS vs. Phase Margin')

subplot(2,2,2)
yyaxis left
plot(PMr, tsr, PMr, trr)
xlim([60 90])
xlabel('Phase Margin (degrees)')
ylabel('Time (sec)')
yyaxis right
plot(PMr, VMr)
ylabel('Vector Margin')
title('Rise/Settle time and VM vs. Phase Margin')
legend('Settling Time', 'Rise Time', 'Vector Margin')

subplot(2,2,3)
plot(PMr, CEr)
xlim([60 90])
xlabel('Phase Margin (degrees)')
ylabel('Peak Controller Effort')
title('Peak Controller Effort vs. Phase Margin')

subplot(2,2,4)
plot(PMr, GMr)
xlim([60 90])
xlabel('Phase Margin (degrees)')
ylabel('Gain Margin (dB)')
title('Gain Margin vs. Phase Margin')
```



Looking at the plot above, it is very similar to the balanced design ones, except the phase margin does not go below 60 this time, so lets check it out. If we only look from a phase margin of 65-90, due to the garbage data below that, and plot the parameters for each design, we can see if there is any difference

```
figure(7); clf;
plot(PMr, tsr, PMb, tsb, '*')
hold on
plot(PMr, trr, PMb, trb, '*')
hold on
plot(PMr, OSr, PMb, OSb, '*')
hold on
plot(PMr, VMr, PMb, VMb, '*')
xlabel('Phase Margin (degrees)')
ylabel('Time (sec) / %Overshoot')
yyaxis right
plot(PMr, CEr, PMb, CEb, '*');
ylabel('Peak Controller Effort')
legend('R.T. settle time', 'Bal settle time', 'Bal rise time', 'R.T.
 rise time', 'R.T. %OS', 'Bal %OS','R.T. Vector Margin', 'Bal Vector
 Margin''R.T. Peak Effort', 'Bal Peak Effort')
title('Parameter Comparison')
xlim([65 90])
```



Its the same

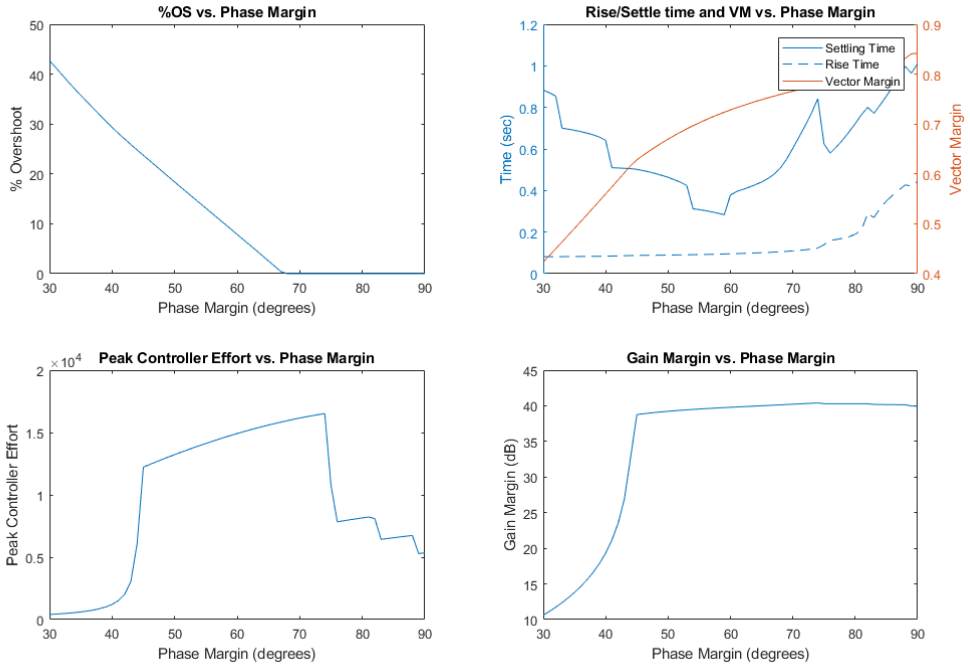
We can see that all the parameters are the same and so no matter if you choose a balanced or reference tracking design focus, the response will still be the same for a PM > 60, which means setting the PM to 76 is still the best choice. The bode plot and step response will look identical to those generated for the balanced design and are therefore not repeated.

And finally disturbance rejection

check the last pidtune design focus option and do the same as above

```
for i = 30:1:90
    j = i-29;

    opts = pidtuneOptions('PhaseMargin',
 i, 'DesignFocus', 'disturbance-rejection');
    C = pidtune(G, 'PDF', opts);

    % calculate open/closed loop and controller ffort
    L(j) = G*C;
    T(j) = L(j)/(1+L(j));
    U(j) = C/(1+L(j));

    %get info from margins
    info = margins(L(j));
```

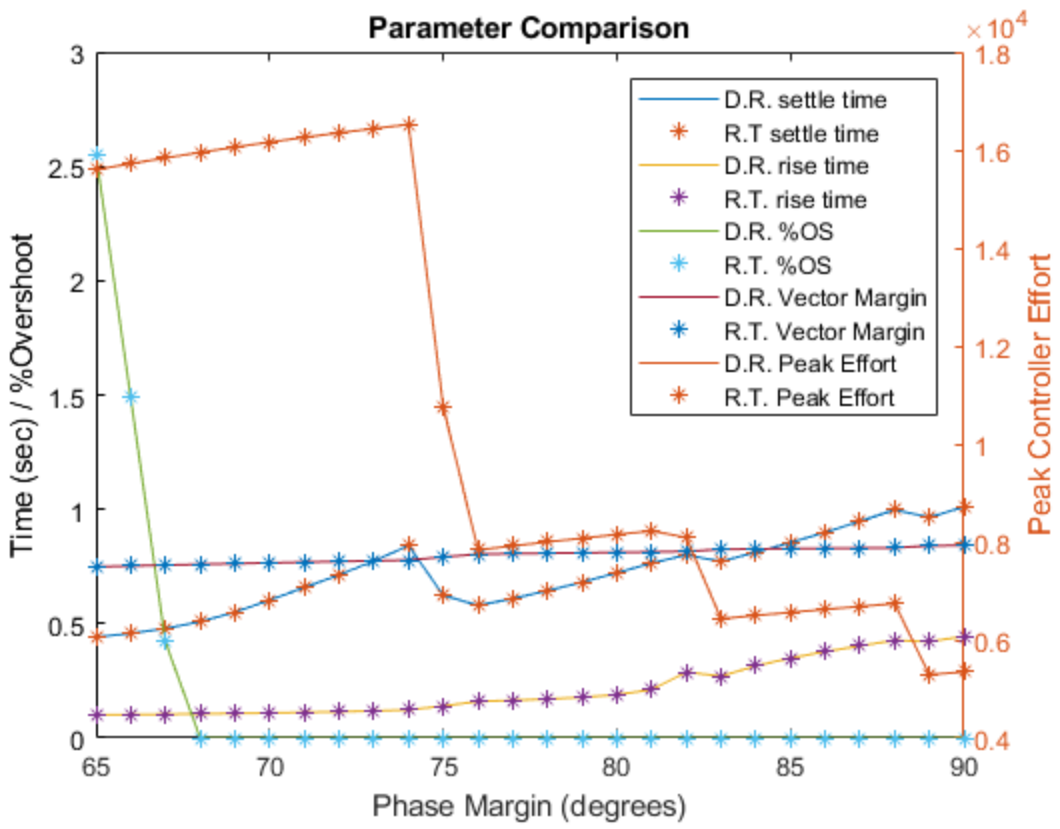
```
    PMd(j) = info.Pm;
    GMd(j) = info.Gm;
    VMd(j) = info.Vm;

    %get info from step response
    info = stepinfo(T(j));
    trd(j)= info.RiseTime;
    OSd(j) = info.Overshoot;
    tsd(j) = info.SettlingTime;

    %get controller effort
    info = stepinfo(U(j));
    CEd(j) = info.Peak;
end

figure(8); clf;

subplot(2,2,1)
plot(PMd, OSd)
xlim([30 90])
xlabel('Phase Margin (degrees)')
ylabel('% Overshoot')
title('%OS vs. Phase Margin')

subplot(2,2,2)
yyaxis left
plot(PMd, tsd, PMd, trd)
xlim([30 90])
xlabel('Phase Margin (degrees)')
ylabel('Time (sec)')
yyaxis right
plot(PMd, VMd)
ylabel('Vector Margin')
title('Rise/Settle time and VM vs. Phase Margin')
legend('Settling Time', 'Rise Time', 'Vector Margin')

subplot(2,2,3)
plot(PMd, CEd)
xlim([30 90])
xlabel('Phase Margin (degrees)')
ylabel('Peak Controller Effort')
title('Peak Controller Effort vs. Phase Margin')

subplot(2,2,4)
plot(PMd, GMd)
xlim([30 90])
xlabel('Phase Margin (degrees)')
ylabel('Gain Margin (dB)')
title('Gain Margin vs. Phase Margin')
```



The disturbance rejection design focus allowed valid data to be collected from the entire range of 30-90 for PM. Using this design focus it would be apparently possible to lower the peak controller effort considerably by setting the phase margin to 30. This however would cause %OS to be large, settling time fairly high, and a very low gain margin

So if we look at the range of 65-90 again, and compare it to the reference tracking design, we can see that it is again the same.

```
figure(9); clf;
plot(PMd, tsd, PMr, tsr, '*')
hold on
plot(PMd, trd, PMr, trr, '*')
hold on
plot(PMd, OSd, PMr, OSr, '*')
hold on
plot(PMd, VMd, PMr, VMr, '*')
xlabel('Phase Margin (degrees)')
ylabel('Time (sec) / %Overshoot')
yyaxis right
plot(PMd, CEd, PMr, CEr, '*');
ylabel('Peak Controller Effort')
legend('D.R. settle time', 'R.T settle time', 'D.R. rise time', 'R.T.
 rise time', 'D.R. %OS', 'R.T. %OS','D.R. Vector Margin', 'R.T. Vector
 Margin', 'D.R. Peak Effort', 'R.T. Peak Effort')
title('Parameter Comparison')
xlim([65 90])
```



Which is Best?

Using the 3 different design foci for pid tune options and varying the pahse margin yielded the same controller for phase margins above 65. A discussed earlier the best choice would be a phase margin of 76 with any of the design foci. This results in an almost 2 fold drop of the peak controller effort from the base pidtune controller for the plant. Ultimately each controller depends on the desired response and setting a PM = 76 resulted in the lowest value for each parameter with respect to each other (in my eyes a balanced design).

Other Noteables

If controller effort was not a factor, a PM of 68 would be the best choice to minimize the other parameters. For the disturbance rejection, a PM of 30 would result in the lowest peak control effort, at the expense of %Overshoot

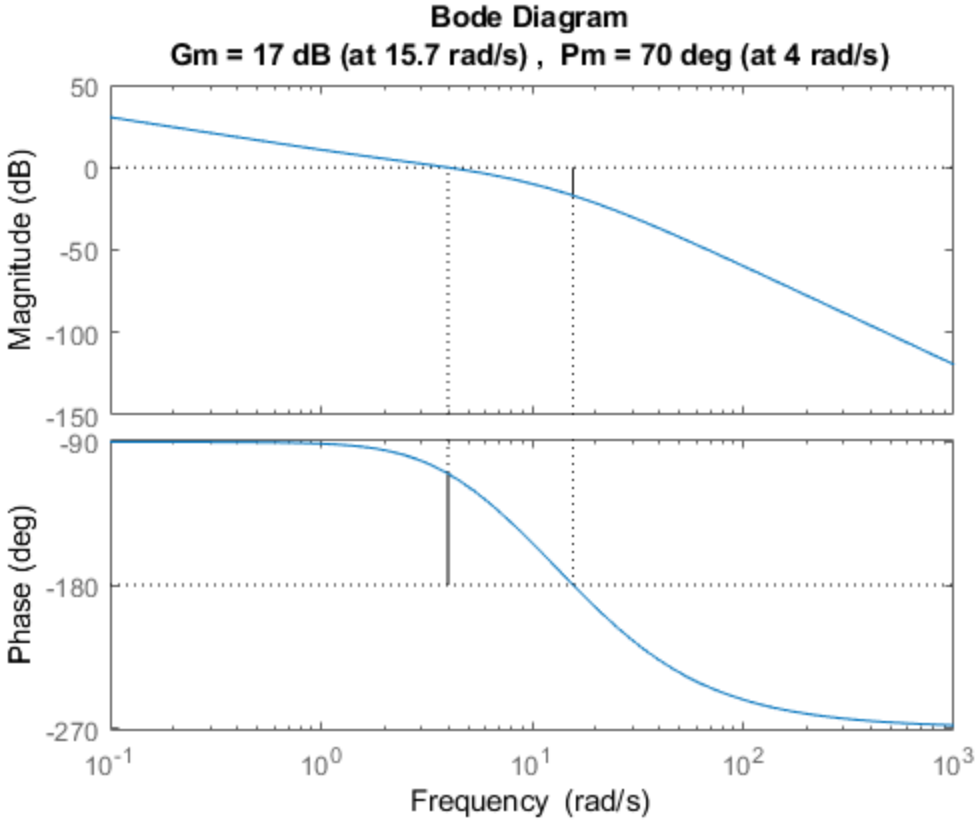
LL Design

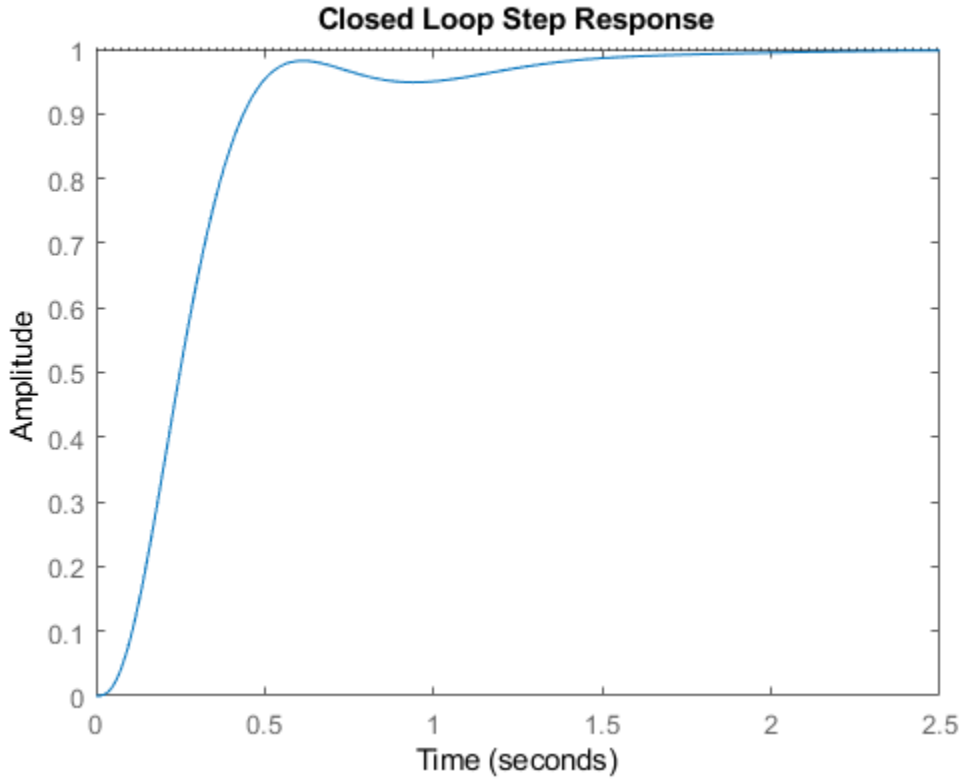
Design a lead and/or lag controller for this system using lldesign. Will start with the values found above (PM = 76, Wg = 10, and default of Ess = 0). I first lowered Wg in order to minimize settling time and then lower the PM to limit the overshoot. Setting Ess to 0.3 was done to decrease both times while limiting the peak controller effort caused by setting it higher.

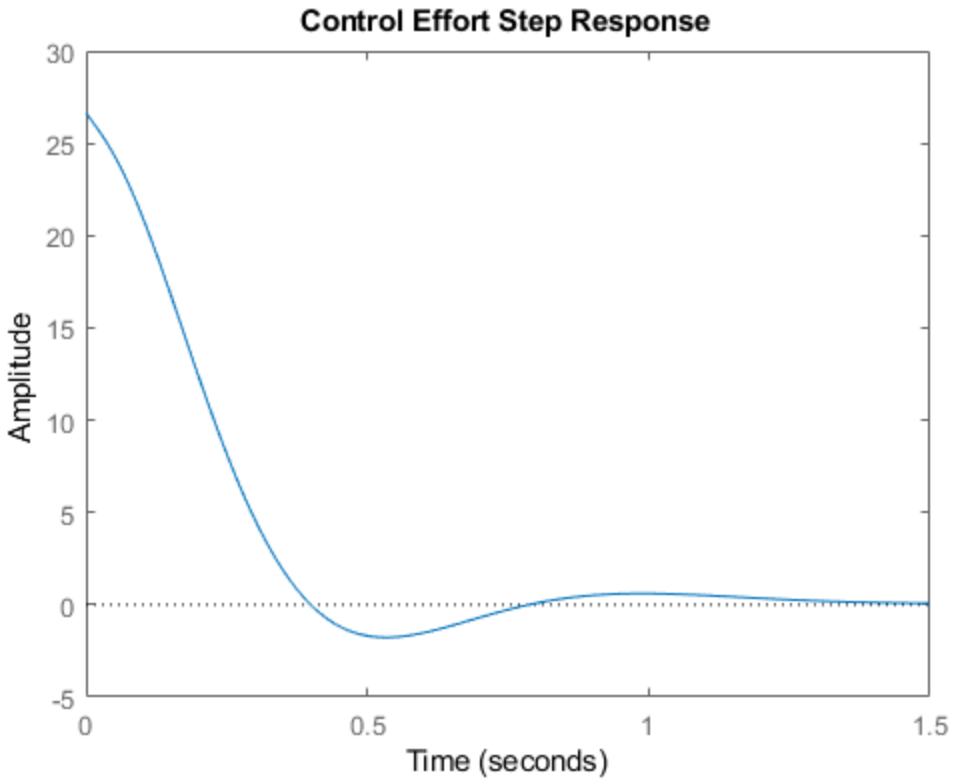
```
PM = 70;
Wg = 4;
```

```
Ess = 0.3;
[D,L,T] = lldesign(G, PM, Wg, Ess);


% Show bode plot with labeled margins and display margin data
figure(10); clf;
margin(L);
margindatall = margins(L);
disp(margindatall);

% Plot the step response of the closed loop t.f.
figure(11); clf;
step(T);
info = stepinfo(T);
OS = info.Overshoot;
RT = info.RiseTime;
ST = info.SettlingTime;
disp('Percent Overshoot')
disp(OS)
disp('Rise Time')
disp(RT)
disp('Settling Time')
disp(ST)
title('Closed Loop Step Response');

% Plot the controller effort step response
figure(12); clf;
step(D/(1+D*G));
title('Control Effort Step Response');
```

```
    Gm: 17.0075
    Wg: 15.6644
    Pm: 70.0085
    Wp: 3.9990
    Vm: 0.7076
    Wv: 8.4465

Percent Overshoot
     0

Rise Time
    0.3300

Settling Time
    1.3719
```


Bode Diagram
Gm = 17 dB (at 15.7 rad/s) , Pm = 70 deg (at 4 rad/s)
Magnitude (dB)
50
0
-50
-100
-150
Phase (deg)
-90
-180
-270
10^-1
10^0
10^1
10^2
10^3
Frequency (rad/s)


Closed Loop Step Response
Amplitude
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
Time (seconds)



The Designed Controller

```
disp('The controller transfer function')
D
```

```
The controller transfer function

D =

  26.647 (s+2.637)
  ----------------
     (s+4.216)

Continuous-time zero/pole/gain model.
```

Using lldesign resulted in a pretty decent controller. While it is much slower compared to the ones generated in the first part of this assignment, the controller effort is considerably lower and makes it much more possible to implement in a real life situation.

Published with MATLAB® R2017b