Swansea University  
School of Engineering

This Matlab *M-book* presents an analytical procedure for PID compensator design. It is based on Section 7.11 of Phillips and Harbor “Feedback *Control Systems,*” Prentice Hall, 1988[[1]](#footnote-1). The compensator transfer function is assumed to be

 (1)

where  is the proportinal gain,  is the derivative gain and  is the initegral gain. In this procedure we choose the PID gain parameters such that, given a desired location for one of the closed-loop poles , the equation

 (2)

is satisfied; that is we are designing a compensator that places a root of the closed-loop characteristic equation at .

The design proceeds as follows. First we express the desired closed loop pole position

 ()

and

 ()

Then the design equations (derived in Appendix B of Phillips and Harbor, 1988) are

 (5)

 (6)

Since there are three unknowns and only two relationships that must be satisfied, one of the gains may be chosen to satisfy a different design specification, such as choosing  to achieve a certain steady-state response. These equations can also be used for PI and P+D controllers by setting the appropriate gain to zero. We now illustrate the design procedure with an example.

## Example

### Definitions (change these to change design)

The plant transfer function is :

G = tf([1],conv([1 1],[5 1]));

The feedback transfer function is  :

H=tf(1,1);

So *G*(*s*)*H*(*s*) is:

GH=G\*H

Transfer function:

1

---------------

5 s^2 + 6 s + 1

The root locus of the uncompensated system is:

clf, sgrid(1/sqrt(2),[0.25:0.25:2]), hold on, rlocus(GH),hold off

(The picture works better if you use Matlab for its display. Select "*Notebook-Notebook Options*" and turn off the "*Embed Figures in Document*" switch).



From the root locus diagram it is clear that for ideal damping the natural frequency of the closed-loop poles would be about 0.9 rad/s with a settling time of:

 s

Suppose we wish to half the settling time then we need to double the natural frequency to **** rad/s. That is:

zeta = 1/sqrt(2); wn=2;

s1 = -zeta\*wn+j\*wn\*sqrt(1-zeta^2)

s1 =

-1.4142 + 1.4142i

The steady state error of the uncompensated type 0 system is:



For the compensated system, which is type 1:

 

So if we want a steady-state *velocity* error of 20% we need

Ki=20;

### Calculations (shouldn’t need to change these commands)

Polar form of 

m\_s1=abs(s1), p\_s1 = angle(s1)\*180/pi % degrees

m\_s1 =

2

p\_s1 =

135

Transfer function evaluated at in polar form:

#### [numGH,denGH] = tfdata(GH,'v');

#### GHs1=polyval(numGH,s1)/polyval(denGH,s1)

GHs1 =

-0.0397 + 0.0610i

Magnitude:

mGHs1=abs(GHs1)

mGHs1 =

0.0728

Phase:

pGHs1=-angle(GHs1)\*180/pi - 90% degrees  **[[2]](#footnote-2)**

pGHs1 =

-213.0264

Hence:

beta = p\_s1\*pi/180; psi = pGHs1\*pi/180;  **% radians**

From (5) and (6)

Kprop = (-sin(beta+psi))/(mGHs1\*sin(beta)) - (2\*Ki\*cos(beta)/m\_s1)

Kprop =

33.1421

Kd = (sin(psi)/(m\_s1\*mGHs1\*sin(beta))) + Ki/(m\_s1^2)

Kd =

10.2929

Compensator is therefore given by

D = tf([Kd, Kprop, Ki],[1, 0])

Transfer function:

10.29 s^2 + 33.14 s + 20

------------------------

s

## Evaluation of Design

### Open loop transfer function:

Go=D\*GH

Transfer function:

10.29 s^2 + 33.14 s + 20

------------------------

5 s^3 + 6 s^2 + s

### Root locus:

rlocus(Go)



### Closed-loop transfer function:

DG = D\*G

Transfer function:

10.29 s^2 + 33.14 s + 20

------------------------

5 s^3 + 6 s^2 + s

Gc = feedback(DG,H)

Transfer function:

10.29 s^2 + 33.14 s + 20

--------------------------------

5 s^3 + 16.29 s^2 + 34.14 s + 20

### Step response:

step(Gc)



1. The proofs of the formulae given are derived in Appendix B of this text. [↑](#footnote-ref-1)
2. You must be careful with angles when using packages like Matlab, and indeed pocket calculators. It is nearly always beneficial to have a sketch so that you can correct the results. In this case a correction of 90° was needed. [↑](#footnote-ref-2)