H_∞ Loop Shaping

J Treurnicht

May 21, 2007

1 Classical Loop Shaping

The idea of shaping the loop response stems from the work of Bode and Horowich. From the Bode integral, at the frequency crossover point ω_0 (where the open-loop gain approached unity or 0 dB), if the slope of the open-loop gain approaches $-20n$ dB/decade, then the phase will be limited to

$$\angle L(j\omega_0) < -n \times 65.3^\circ, \quad \frac{1}{3} \leq \frac{\omega}{\omega_0} \leq 3$$

To get an adequate phase margin, $n \leq 2$ (the slope of the magnitude must be less than -40 dB/decade (preferably -20 dB/decade.

So classical loop shaping design consist of adding controllers and gain to

- Shape the loop function steeply in the low frequency region $\omega \ll \omega_0$ (maximizing the sensitivity function S)
- Limiting the slope to -20 dB/decade around the crossover point $\omega \approx \omega_0$ for stability and transient response. This provides a good phase margin.
- Shape the loop function steeply in the high frequency region $\omega \gg \omega_0$ to minimize the complementary sensitivity function T avoid problems with uncertainties.

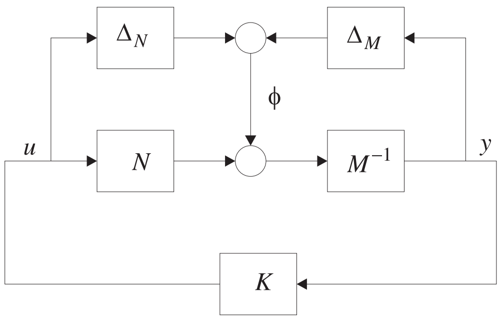
We will use the same basic principle with H_∞ techniques to help us design the controller.

2 Formulation of the H_∞ Loop Shaping Design

Using the left coprime factorization

$$P = M^{-1}N$$

for a perturbed plant



then P becomes

$$P = (M + \Delta_M)^{-1}(N + \Delta_N)$$

The stability is robust (see Glover and McFarlane 1989) provided

$$\gamma = \left\|\left[\begin{array}{c} K \\ I \end{array}\right](I - PK)^{-1}M^{-1}\right\|_\infty \leq \frac{1}{\epsilon}$$

Note that γ is the $||H||_\infty$ norm from ϕ to $\left[\begin{array}{c} u \\ y \end{array}\right]$ and $(I - PK)^{-1}$ is the sensitivity function.

The lowest γ and stability margin ϵ comes from

$$\gamma_{\min} = \epsilon_{\max}^{-1} = \left\{1 - ||[N\ M]||_H^2\right\}^{\frac{1}{2}} = (1 + \rho\, X\, Z)^{\frac{1}{2}}$$

with the two Ricatti equations given by

$$\begin{array}{rcl} (A - B\,S^{-1}\,D^T\,C)Z + Z(A - B\,S^{-1}\,D^T\,C)^T - Z\,C^T\,R^{-1}\,C\,Z + B\,S^{-1}\,B^T & = & 0 \\ (A - B\,S^{-1}\,D^T\,C)X + Z(A - B\,S^{-1}\,D^T\,C)^T - X\,B\,S^{-1}\,B^T\,X + C^T\,R^{-1}\,C & = & 0 \end{array}$$

with $R = I + D\,D^T$ and $S = I + D^T\,D$.

The controller which guarantees that

$$\left\|\left[\begin{array}{c} K \\ I \end{array}\right](I - PK)^{-1}M^{-1}\right\|_\infty \leq \gamma\left(= \frac{1}{\epsilon}\right)$$

is given by

$$\begin{array}{rcl} K & = & \left[\begin{array}{c|c} A + B\,F + \gamma^2(L^T)^{-1}ZC^T(C + DF) & \gamma^2(L^T)^{-1}ZC^T \\ \hline B^T\,X & -D^T \end{array}\right] \\ F & = & -S^{-1}(D^T\,C + B^T\,X) \\ L & = & (1 - \gamma^2)I + X\,Z \end{array}$$

The function `coprimeunc` listed in the appendix can be used to generate such a controller.

Define the margin $b_{P,K}$ as

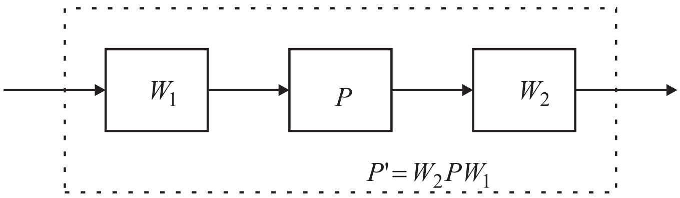
$$b_{P,K} = \begin{cases} \left(\left\| \begin{bmatrix} K \\ I \end{bmatrix} (I-PK)^{-1} \begin{bmatrix} I & P \end{bmatrix} \right\|_\infty \right)^{-1} & \text{if } K \text{ stabilises } P \\ 0 & \text{otherwise} \end{cases}$$

The margin $b_{P,K}$ is related to the SISO gain and phase margins through

$$\begin{aligned} \text{gain margin} &\geq \frac{1+b_{P,K}}{1-b_{P,K}} \\ \text{phase margin} &\geq 2\sin^{-1} b_{P,K} \end{aligned}$$

The classical procedure for designing the loop consists of:

1. Choose a pre-compensator W_1 and post-compensator W_2 to augment the plant response



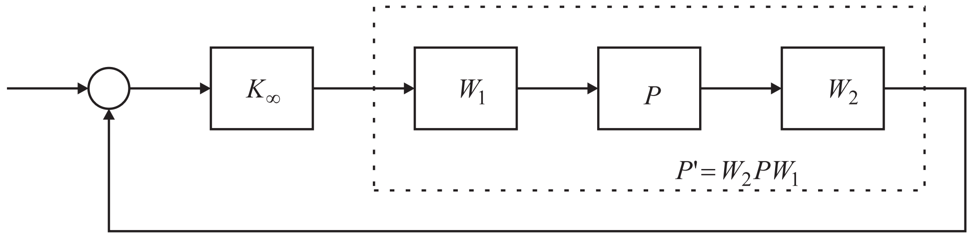
   Then calculate $\epsilon_{\max} = b_{P,K}$ from

$$\begin{aligned} \epsilon_{\max} &= \begin{cases} \left(\left\| \begin{bmatrix} K \\ I \end{bmatrix} (I-PK)^{-1} \begin{bmatrix} I & P \end{bmatrix} \right\|_\infty \right)^{-1} & \text{if } K \text{ stabilises } P \\ 0 & \text{otherwise} \end{cases} \\ &= \sqrt{1 - \left\| \begin{bmatrix} \tilde{N} & \tilde{M} \end{bmatrix} \right\|_H^2} < 1 \end{aligned}$$

   with $\tilde{M}, \tilde{N}$ the normalised coprime factors of P such that $P = \tilde{M}^{-1}\tilde{M}$. The idea is to have $\epsilon_{\max}$ not too small. Adjust W_1 and W_2 to maximize $\epsilon_{\max}$ at the cross-over point.

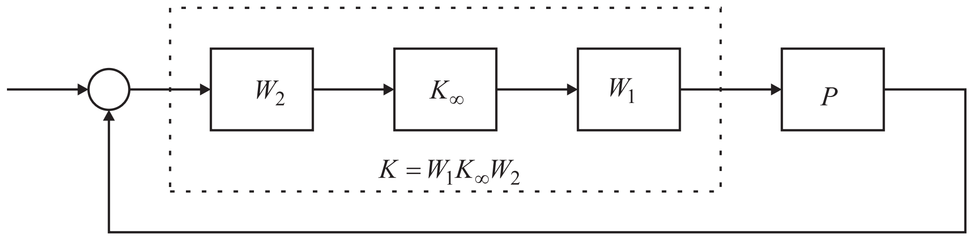
2. Select a $\epsilon \leq \epsilon_{\max}$ and synthesize a controller K_∞ that satisfies

$$\left\| \begin{bmatrix} K \\ I \end{bmatrix} (I-PK)^{-1} M^{-1} \right\|_\infty \leq \frac{1}{\epsilon}$$



3. The controller is then formed by combining the design controller K_∞ with the weighting functions W_1 and W_2 using

$$K = W_1 K_\infty W_2$$

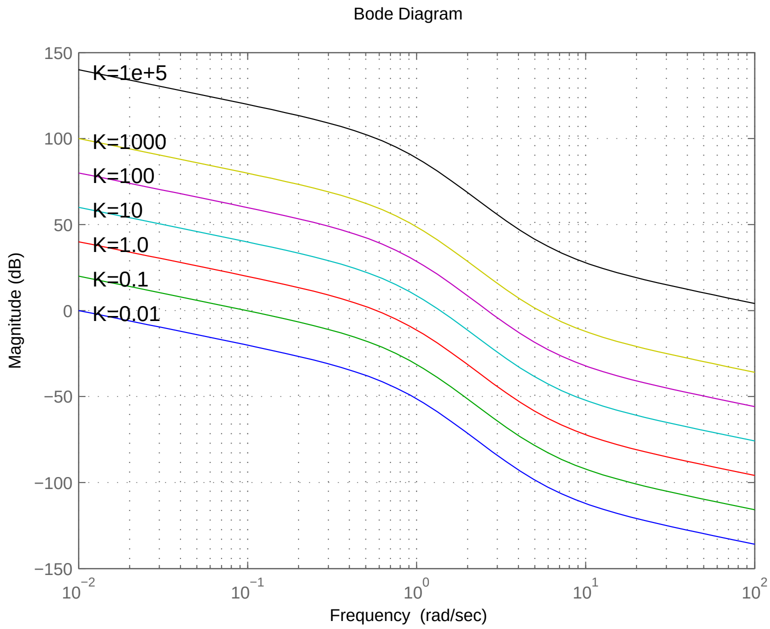


This technique uses the function $b_{P,K}$ to verify that the slope around the cross-over point is not too large (we should aim for -20 dB/decade), therefore allowing us to ignore any phase margin calculations.

Example 1 Consider the transfer function

$$P(s) = \frac{(0.2\,s+1)^4}{s(s+1)^4}$$

The transfer functions for different values of K is given below



The slope of the function between 1 and 10 rad/s is very steep, and not suitable for a loop-shape fit. Calculating the $b_{P,K}$ we can see the effect

K	10^{-5}	10^{-3}	0.1	1	10	10^2	10^4
$b_{P,K}$	0.357	0.094	0.057	0.060	0.077	0.123	0.493

The low values point to the difficulty in forming the solution.

The new functions make the task of fitting a function to a loop-shape significantly simpler as shown by the following example.

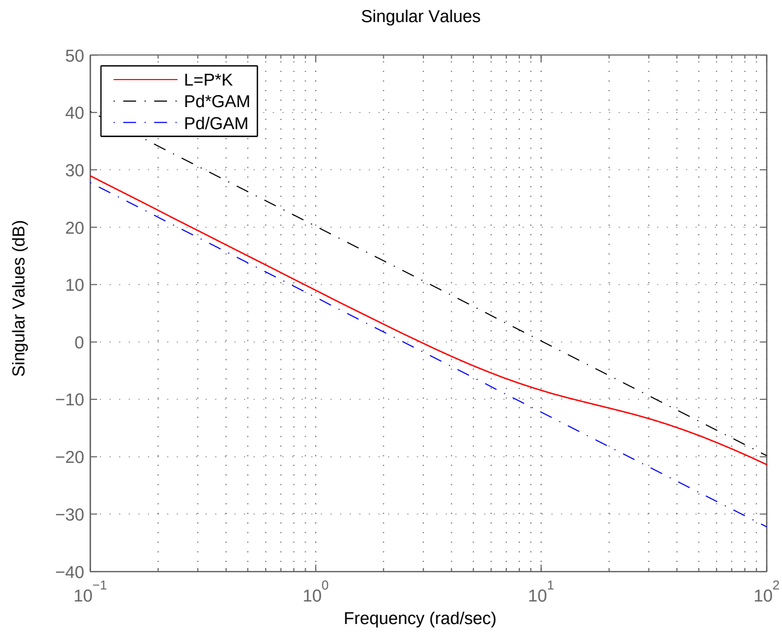
Example 2 Consider the random non-minimum phase plant generated by

```
rand('seed',0);randn('seed',0);
P=((s-10)/(s+100))*rss(3,4,5);     % 4-by-5 non-min-phase plant
```

If we desire a 5 rad/s bandwidth we can generate the controller by simply

```
Pd=5/s;             % desired bandwith w0=5
[K,CL,GAM,INFO]=loopsyn(P,Pd);
```

giving the following loop-shape fit

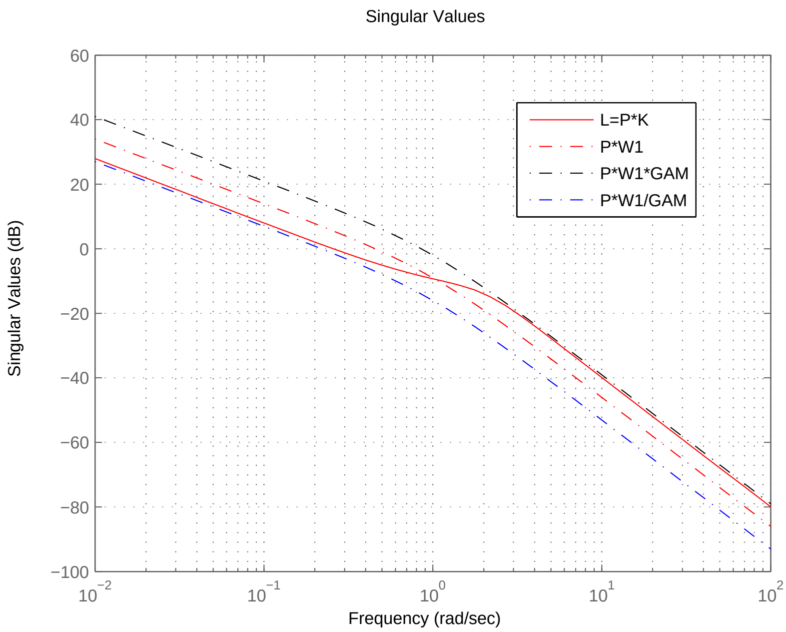


Example 3 Consider the transfer function

$$P(s) = \frac{(s-1)}{(s+1)^2}$$

The procedure to solve this using the original McFarlane & Glover procedure as per Zhou is

```
s=zpk('s');
P=(s-1)/(s+1)^2;  % Plant
W1=0.5/s;         % Ideal Shaped plant
[K,CL,GAM]=ncfsyn(P,W1);
sigma(P*K,'r',P*W1,'r-.',P*W1*GAM,'k-.',P*W1/GAM,'b-.')
grid
legend('L=P*K','P*W1', 'P*W1*GAM','P*W1/GAM',2)
```



Singular Values
Singular Values (dB)
Frequency (rad/sec)
60
40
20
0
−20
−40
−60
−80
−100
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
L=P*K
P*W1
P*W1*GAM
P*W1/GAM

A Function coprimeunc

```
% Uses the Robust Control Toolbox
function [Ac,Bc,Cc,Dc,gammin] = coprimeunc(a,b,c,d,gamrel)
%
% Finds the controller which optimally robustify a given shaped plant
% in terms of tolerating maximum coprime uncertainty
%
% Inputs:
%   a,b,c,d:    State-space description of (shaped) plant
%   gamrel:     Gamma is used as gamrel*gammin (typical gamrel = 1.1)
%
% Outputs:
%   Ac,Bc,Cc,Dc:    Robustifying controller (positive feedback)
%

S = eye(size(d'*d))+d'*d;
R = eye(size(d*d'))+d*d';
A1 = a-b*inv(S)*d'*c;
Q1 = c'*inv(R)*c;
R1 = b*inv(S)*b';
[x1,x2,eig,xerr,wellposed,X]=aresolv(A1,Q1,R1);
[x1,x2,eig,xerr,wellposed,Z]=aresolv(A1',R1,Q1);
gammin = sqrt(1+max(eig(X*Z)));
gam = gamrel*gammin;
L = (1-gam*gam)*eye(size(X*Z))+X*Z;
F = -inv(S)*(d'*c+b'*X);
Ac = a+b*F+gam*gam*inv(L')*Z*c'*(c+d*F);
Bc = gam*gam*inv(L')*Z*c';
Cc = b'*X;
Dc = -d';
```