7. CASCADED DESIGN

It has been shown (Horowitz 1992, and works quoted therein) that if it is possible to measure an internal variable (in addition to the controlled output variable) with a not too noisy a sensor and feed it back into the feedback compensater, the controller bandwidth can be reduced in comparison with the controller bandwidth of a system where only the output variable is measured, while retaining the specifications for the same total plant uncertainty. The basic reason is that the burden of uncertainty reduction is carried by two feedback compensator sections.

In this chapter we will illustrate one such situation, a cascaded design with servo and output disturbance specifications, and independent uncertainty in the two plant sections, see Figure 7.1. We will follow one design methodology in Horowitz (1992) which is extendable in a straight-forward way to plants with more than two sections.

Qsyn includes three fully documented functions for this type of cascaded design, `fcasc_r`, `fcasc_s`, and `tplcasc`. For other types of cascaded design, e.g. one including other specifications, or dependent uncertainties, the user will have to write her own functions on the basis of the existing ones.

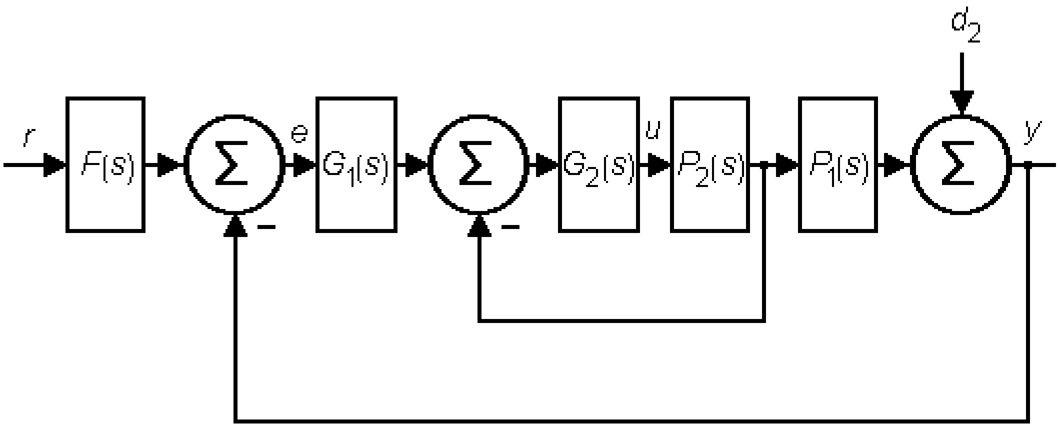

Figure 7.1. Cascaded SISO system with three degree of freedoms, and output disturbance.

7.1 Design procedure

Assume that a cascaded three degree-of-freedom (F, G_2, and G_1) with independent uncertainty in the two plant sections P_1 and P_2 is given according to Figure 7.1. Contrary to conventional cascaded design, the QFT cascaded design is "top-down", or "outside-in", i.e. the compensator for the outer loop, G_1, is designed first.

1. Assume that the inner closed loop, $P_2G_2/(1+P_2G_2)$, is sufficiently well regulated, so that it temporarily can be approximated by 1. Then the design problem for the outer loop is reduced to a conventional SISO design problem (Chapter 1), whereby G_1 is designed as to satisfy the disturbance rejection specifications for the closed loop transmission from d_2 to y, $1/(1+P_1G_1)$, and the tolerance specification emanating from the servo specification for the transmission from r to y.

During the feedback design phase of G_1, it is *essential* not to satisfy the Horowitz bounds in too tight a way, in particular in the cross over frequency region. In general this can be done without paying any large price in bandwidth augmentation of G_1. The reason for this rule is that one has to leave some *free uncertainty* for the inner loop $P_2G_2/(1+P_2G_2)$, so that it will not be necessary to design P_2G_2 with infinite bandwidth.

2. With G_1 designed, one turns to the inner loop. Now G_2 has to be designed such that the tolerance specification emanating from the servo specification and the output disturbance rejection specification are satisfied. The equation to find the Horowitz bounds for the tolerence specification is

$$\frac{\max\limits_{m,n}\left|\dfrac{G_1(j\omega_k)\left(\dfrac{G_2(j\omega_k)P_{2n}(j\omega_k)}{1+G_2(j\omega_k)P_{2n}(j\omega_k))}\right)P_{1m}(j\omega_k)}{1+G_1(j\omega_k)\left(\dfrac{G_2(j\omega_k)P_{2n}(j\omega_k)}{1+G_2(j\omega_k)P_{2n}(j\omega_k))}\right)P_{1m}(j\omega_k)}\right|}{\min\limits_{m,n}\left|\dfrac{G_1(j\omega_k)\left(\dfrac{G_2(j\omega_k)P_{2n}(j\omega_k)}{1+G_2(j\omega_k)P_{2n}(j\omega_k))}\right)P_{1m}(j\omega_k)}{1+G_1(j\omega_k)\left(\dfrac{G_2(j\omega_k)P_{2n}(j\omega_k)}{1+G_2(j\omega_k)P_{2n}(j\omega_k))}\right)P_{1m}(j\omega_k)}\right|}\le\frac{b(\omega_k)}{a(\omega_k)} \tag{7.1}$$

where ω_k denotes the template frequencies, P_{1m} the outer plant section cases, P_{2n} the inner plant section cases, and $b(\omega_k)$ and $a(\omega_k)$ are the upper and lower servo specification limits in the frequency domain, compare (1.8), (1.10), and Figure 1.5. The equation to find the Horowitz bounds for the output disturbance rejection specification is

$$\max_{m,n}\left|1+G_1(j\omega_k)\left(\frac{G_2(j\omega_k)P_{2n}(j\omega_k)}{1+G_2(j\omega_k)P_{2n}(j\omega_k))}\right)P_{1m}(j\omega_k)\right|^{-1}\le c(j\omega_k) \tag{7.2}$$

where $c(j\omega_k)$ is the output disturbance rejection (sensitivity) specification in the frequency domain. As usual the Horowitz bounds are found relative to $L_{2\text{nom}}=G_2P_{2\text{nom}}$ rather than for G_2. The design of G_2 is done as usual: an $L_{2\text{nom}}$ candidate and the Horowitz bounds are displayed in a Nichols chart.

3. When the feedback compensators G_1 and G_2 are designed, it is wise to check the noise rejection properties of the transmissions from n_1 to u, and from n_2 to u, (where n_1 is the outer sensor noise, n_2 is the inner sensor noise, and u is the control signal into the plant) in relation to the relative strengths and frequency distributions of n_1 and n_2. One might find that another distribution of bandwidths between G_1 and G_2 would decrease the total noise influence on u, and hence a redesign would be beneficial.

4. Design of the prefilter F to satisfy the closed loop servo specifications.

.

7.2 A cascaded design example

7.2.1 Plant definition and templates

Let the outer plant section be given by

$$P_1(s) = \frac{k_1}{(s+1)^2}, \quad k_1 \in [1,100] \tag{7.3}$$

modelled in the plant definition file `p1.m` (Figure 7.2), with $P_{1nom}(s) = 1/(s+1)^2$. Let the inner plant section be

$$P_2(s) = \frac{k_2}{s}, \quad k_2 \in [1,100] \tag{7.4}$$

modelled in the plant definition file `p2.m` (Figure 7.3), with $P_{2nom}(s) = 1/s$. The templates are calculated and displayed with the commands

```
ctpl('P1',[],'adedge_[5,5]');
hngrid,  showtpl('p1',[],[],'r-',gcf);  hzoom         % Figure 7.4a

ctpl('P2',[],'adedge_[5,5]');
figure,hngrid, showtpl('p2',[],[],'r-',gcf); hzoom    % Figure 7.4b
```

whereby we remark that the resolution of the template computation will turn out to be too low to give smooth Horowitz bounds later on. Notice that the templates in Figure 7.4b are all vertical with gain extent only, lying along the nominal 1/*s*.

```
function   [Par,w tpl,w nom,method,P num,P den, ...
            n_dif,Uns_Par] = p1

% The P1 plant in the cascaded loop design example.

% Definition of the parameters
% =====================
   Par = [
         'k1=[1,100,1]' , ... % uncertain parameters
         ];
% Definition of the frequency vectors [rad/sec]
% =====================================
  w_tpl=[0.5 1 2 5 10 20 30 50 70 100 200];
                                    % Template frequency vector.
  w_nom=logspace(-1,4,200);         % Nominal frequency vector.

% Definition of the template computation method
% ==========================================
  method = 'adedge_[2,2]';

% Polynomial Structure
% ================
  P_num ='k1';            % numerator
  P_den='s^2+2*s+1';      % denominator
```

Figure 7.2. The plant definition file `p1.m` *for the outer plant section (7.3).*

```
function    [Par,w_tpl,w_nom,method,P_num,P_den, ...
             n_dif,Uns_Par] = p2

% The P2 plant in the cascade loop design example.

% Definition of the parameters
% ====================
   Par = [
         'k2=[1,100,1]' , ... % uncertain parameters
         ];
% Definition of the frequency vectors [rad/sec]
% ========================================
  w_tpl=[0.5 1 2 5 10 20 30 50 70 100 200];
                                 % Template frequency vector.
  w_nom=logspace(-1,4,200);      % Nominal frequency vector.

% Definition of the template computation method
% ===========================================
  method = 'adedge_[2,2]';

% Polynomial Structure
% ================
  P_num = 'k2';          % numerator
  P_den = 's';           % denominator
```

Figure 7.3. The plant definition file `p2.m` *for the inner plant section (7.4).*

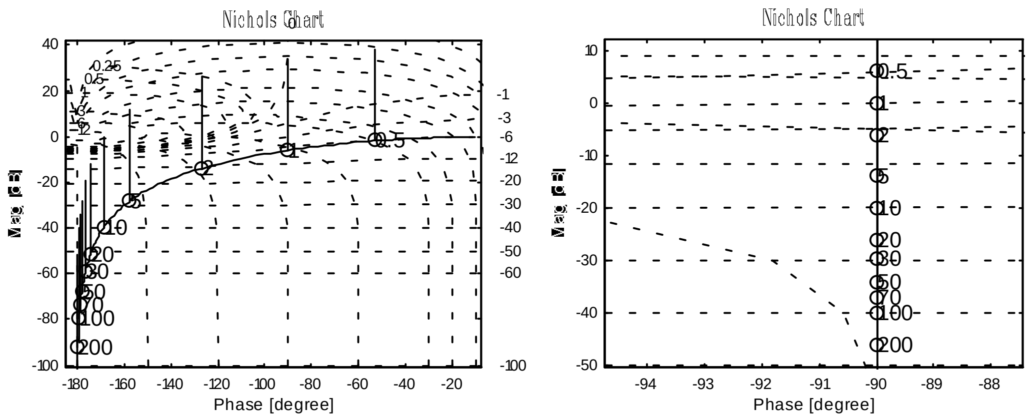

Figure 7.4. a) templates for (7.3); b) templates for (7.4)

7.2.2 Specifications

Next comes the specifications. The servo specifications are constructed in the usual way with the command `rsrs`, while a constant 6 dB limit is chosen as an output disturbance rejection specification in the frequency domain. The specifications are shown in Figures 7.5 and 7.6.

```
w_nom=gettpl('p1','nom');      % extract nominal frequency vector

% Stepresponse specification, 3:rd order model:
rsrs('p1','rsrs',[1 0.5],10,[1.5 10],[],w_nom,5,3.1);
showspc('p1','rsrs'), subplot(211), axis([1 100 -40 10]) % Figure 7.5

% 6 dB output disturbance rejection or sensitivity specification:
insert('p1.spc',[w_nom(:),6*ones(size(w_nom(:)))],'odsrs_w','r')
                               % add2spc could have been used instead
subplot(211), showspc('p1','odsrs','freq','r-',gcf)       % Figure 7.5
```

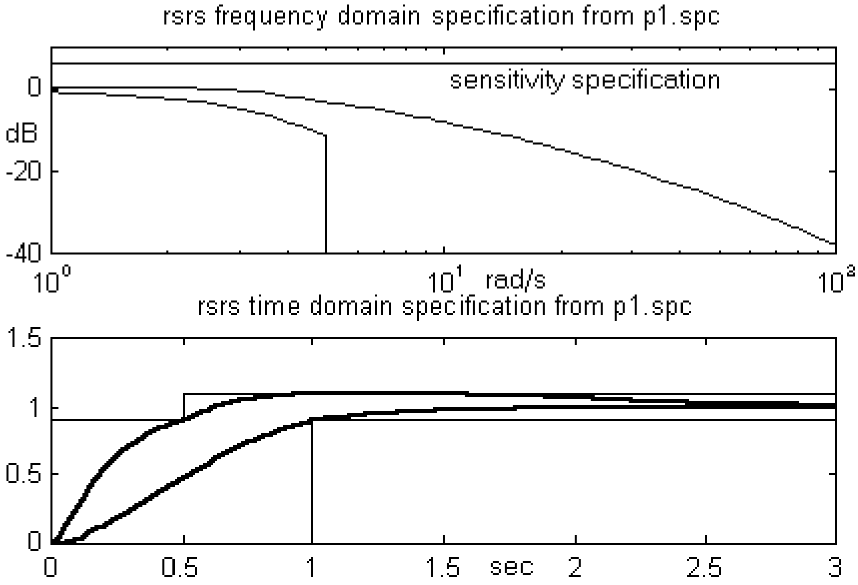

Figure 7.5. Servo and sensitivity (output disturbance rejection) specifications.

7.2.3 Outer loop feedback design

The outer loop design is identical to a usual SISO design (Chapter 2), and the two sets of outer Horowitz bounds are calculated accordingly:

```
% outer rsrs (tolerance) bounds
cbnd('P1','rsrs');
figure, hngrid, showbnd('P1',gcf,[],'rsrs','roll-'); hold on
                                                    % Figure 7.6

% outer output disturbance rejection (sensitivity) bounds
cbnd('p1','odsrs',[],[],'p1','p1');
showbnd('p1',gcf,[],'odsrs','roll-');               % Figure 7.6
```

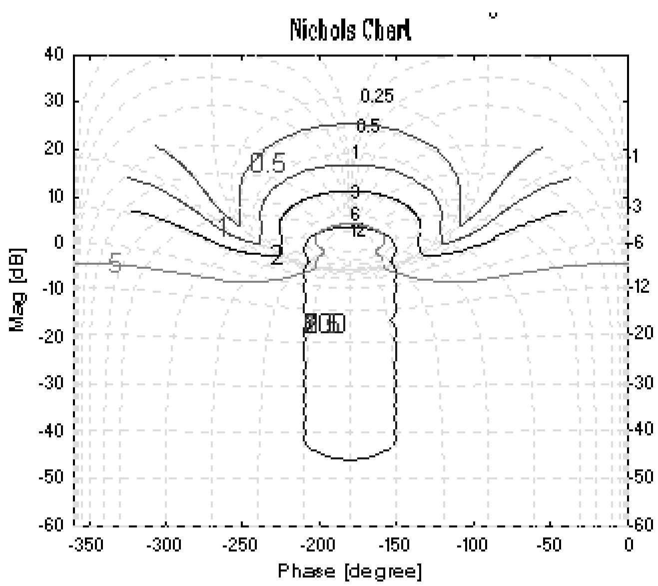

Figure 7.6. Tolerance bounds (hat shaped) and sensitivity bounds (loaf shaped) for the design of the outer loop. Notice that all sensitivity bounds are equal, since the templates of (7.3) are equal, and the sensitivity specification is a constant (6 dB).

```
function[G]=g1(s);
% outer loop feedback
G=3*(s.^2+2*s+1).*(s/50+1).*(s/350+1)./...
  ( s.*(s/10+1).*(s/200+1).*(s.^2/350^2+s/350+1));
```

Figure 7.7. Controller function file for (7.5)

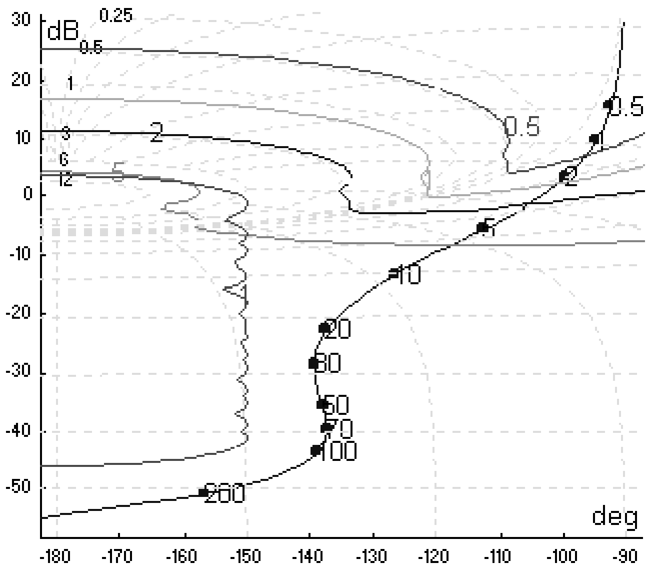

Figure 7.8. Nominal open outer loop $L_{1nom} = G_1 P_{1nom}$ with bounds from Figure 7.6. Notice that the sensitivity bound is not smooth due to low template resolution.

After a few attempts we settled for the outer feedback controller

$$G_1(s) = 3\frac{\left(s^2 + 2s + 1\right)\left(s/50 + 1\right)\left(s/350 + 1\right)}{s\left(s/10 + 1\right)\left(s/200 + 1\right)\left(s^2/350^2 + s/350 + 1\right)}, \tag{7.5}$$

realized in the controller function file `g1.m`, reprinted in Figure 7.7. The nominal open outer loop $L_{1nom} = G_1 P_{1nom}$ is presented in Figure 7.8, as a result of the commands

```
% nominal open outer loop
cdesign('p1.tpl','g1');
showbnd('p1',gcf,[],'rsrs','roll-',[1],'odsrs',':');  % Figure 7.8
```

We notice in Figure 7.8 that all bounds are well satisfied, with some "air" between the nominal points and their bounds.

7.2.4 Inner loop feedback design

In order to design the inner loop, we must compute the Horowitz bounds from (7.1) and (7.2), which requires that the templates "seen" by G_2 (or $L_{2nom} = G_2 P_{2nom}$) in the left hand side of (7.1) and (7.2) be computed. A special command, `tplcasc`, computes the necessary templates. Notice that these templates are stored in complex form for computational speed and cannot be displayed by the command `showtpl`, without first converting them with the operation `c2n`, e.g. with the `tplfop` command.

The user is urged to study that function `tplcasc` carefully, in order to be able to modify it for other cascaded designs. Two special criteria functions, `fcasc r` and `fcasc s` are especially adapted to the template file produced by `tplcasc`, in order to compute the bounds for L_{2nom} from (7.1) and (7.2), respectively. The command sequence is

```
% Combine P1,P2,G1 into a template suitable for the  calculatation
% of cascaded bounds for inner loop design:
%
tplcasc('Pcasc','P1','P2','g1');

% Compute tolerance bounds, and outer disturbance rejection bounds:
cbnd('Pcasc','rsrs',[.5 1 2 5],[],'p1','Pcasc','casc r','fcasc r',...
            [10 2],[-10 30]);
cbnd('Pcasc','odsrs',[ 5 10 20 30 50 70 100 200],[],'p1','Pcasc',...
            'casc_s','fcasc_s',[10 2],[-80 20]);
```

The bounds are displayed in Figure 7.10 together with our suggested inner nominal open inner loop, $L_{2nom} = G_2 P_{2nom}$, where the inner feedback compensator,

$$G_2(s) = 2\frac{\left(s^2 + 2s + 1\right)(s/200 + 1)(s/800 + 1)}{(s/5 + 1)(s/20 + 1)(s/3500 + 1)\left(s^2/200^2 + s/200 + 1\right)}, \qquad (7.6)$$

has its frequency function computed in the controller function file g2.m, see Figure 7.9.

```
function[G]=g2(s);
% inner loop feedback
G=2*(s.^2+2*s+1).*(s/200+1).*(s/800+1)./...
  ( (s/5+1).*(s/20+1).*(s/3500+1).*(s.^2/200^2+s/200+1));
```

Figure 7.9. Controller function file for (7.6)

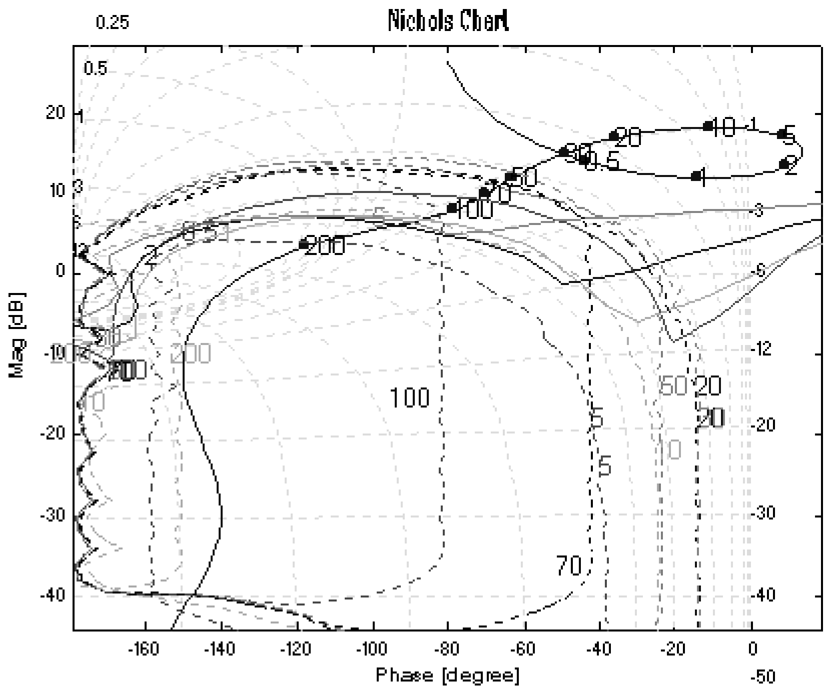

Figure 7.10. The nominal inner open loop L_{2nom}, together with the tolerance and output disturbance rejection bounds. Notice the violation of the latter bound for 70 rad/s.

The commands leading to the display of the inner nominal open loop and its bounds are

```
figure; hngrid; cdesign('P2.tpl','g2');
showbnd('Pcasc',gcf,[0.5 1 2 5],'casc_r','roll-', ...
      [5 10 20 30 50 70 100 200],'casc_s','roll:')  % Figure 7.10
```

We notice in Figure 7.10 that the output sensitivity bound of 70 rad/s is violated. Moreover, the cross over frequency of the inner loop is more than 200 rad/s while the crossover frequency of the outer loop is about 20 rad/s. This may be acceptable or not, but has to be considered in relation to the sensor noise characteristics. An eventual redesign of the outer loop should leave a larger phase gap to the sensitivity bound at 50 rad/s and higher.

We would also like to point out that G_1 has an integrator and is thus of PI type, while G_2 is of PD type (both with an additional number of loop shaping filters), exactly as conventional wisdom and experience has it for cascaded design.

7.2.5 Closing the loops

We are now in the position to close the inner loop, to get $P_2G_2/(1+P_2G_2)$, by the command

```
tplfop('P2close','iosrs',[],'P2',1,'G2',1);  %closed inner loop
```

The outer uncompensated open loop, with the closed inner loop as a factor, $P_1P_2G_2/(1+P_2G_2)$, is computed as follows, whereby we can show its templates in Figure 7.11,

```
% P1*(closed inner loop)
tplfop('Popen','*',[],'P1','P2close')
hngrid, showtpl('Popen',[],[],[],gcf)      % Figure 7.11
```

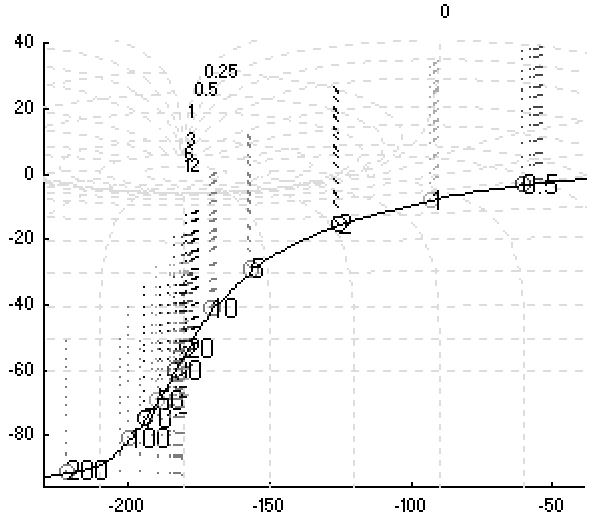

Figure 7.11. The nominal and the templates of $P_1P_2G_2/(1+P_2G_2)$, i.e. the outer plant with the closed open loop in series. Compare with Figure 7.4a and notice the phase spread of the high frequency templates, and the faster high frequency roll off in this figure.

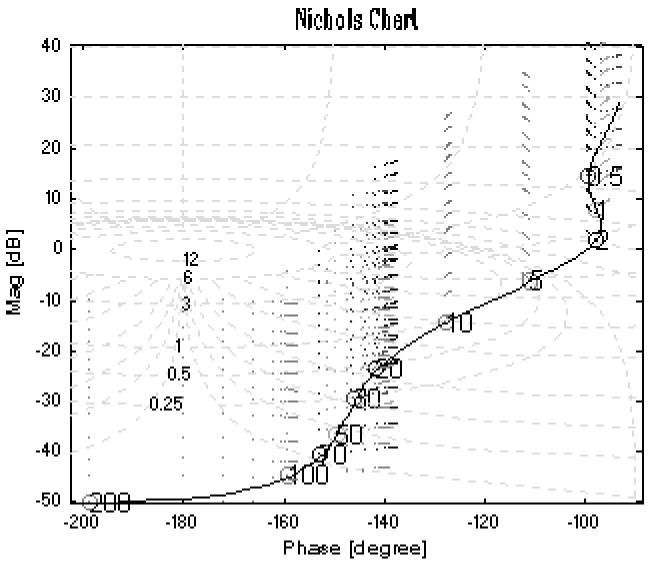

Figure 7.12. The nominal and templates of the outer compensated open loop, with the closed inner loop as a factor, $P_1(P_2G_2/(1+P_2G_2))G_1$ *in an inverse Nichhols chart. Notice the sensitivity violation at 70 rad/s, i.e. the template enters the interior of the* $|1/(1+L)|$=6dB *sensitivity locus*.

The outer compensated open loop, with the closed inner loop as a factor, $P_1(P_2G_2/(1+P_2G_2))G_1$, is easily be computed, with its templates and nominal in Figure 7.12,

```
% P1*(closed inner loop)*G1
tplfop('PGopen','*',[],'Popen',1,'G1')
figure, hngrid([],[],1),showtpl('PGopen',[],[],[],gcf),hzoom
                                                    % Figure 7.12
```

With the outer plant in series with the closed inner loop, $P_1 P_2 G_2 / (1+P_2 G_2)$, computed in the template file `Popen.tpl` one can easily recompute the outer Horowitz bounds for the "true" outer plant, and display them together with the nominal outer compensated open loop, $P_{1nom}\left(P_{2nom} G_2 / (1+P_{2nom} G_2)\right) G_1$, with the commands

```
cbnd('Popen','rsrs',[],[],'P1');            % tolerance bounds
cbnd('Popen','odsrs',[],[],'P1');           % sensitivity bounds

%P1nom*(closed nominal inner loop)*G1 + bounds
figure, cdesign('Popen.tpl','g1');showbnd('Popen',gcf,[],'rsrs', ...
                                          'roll-',[1],'odsrs','r');
```

It is left as an exercise to the user to do this, check the sensitivity bound violation at 70 rad/s and possibly redesign G_1.

Instead we turn to our design, and close the outer loop to get the template file for $\left(P_1\left(P_2 G_2 / (1+P_2 G_2)\right) G_1\right) / \left(1+P_1\left(P_2 G_2 / (1+P_2 G_2)\right) G_1\right)$:

```
%Closed outer loop around  PGopen:
tplfop('Pclosed','iosrs',[],'PGopen',1,1,1);
```

7.2.6 Prefilter design

Exactly as in the two degree-of-freedom SISO design we can now synthesize the prefilter

$$F(s) = \frac{1}{(s/5+1)(s/20+1)\left(s^2/15^2 + 2\cdot 0.8\cdot s/15+1\right)} \tag{7.7}$$

(controller function file `f.m` in Figure 7.13), and show the complete closed loop, $F\left(P_1\left(P_2 G_2 / (1+P_2 G_2)\right) G_1\right) / \left(1+P_1\left(P_2 G_2 / (1+P_2 G_2)\right) G_1\right)$, in a Bode gain diagram together with the servo specification in Figure 7.14:

```
%Prefilter design
figure, fdesign('Pclosed.tpl','F',[]);
showspc('P1','rsrs','freq',[],gcf);               % Figure 7.14
```

An alternative way to design the prefilter, and show Figure 14 is with the following sequenc of commands that closes the outer loop and multiplies the prefilter in one `tplfop` operation:

```
tplfop('Prsrs','rsrs',[],'Popen',1,'G1','F');
figure;fdesign('Prsrs.tpl',[],[]);hold on;
showspc('P1','rsrs','freq',[],gcf);                     % Figure 7.14
```

```
function[F]=f(s);
% prefilter
F=1./( (s/5+1).*(s/20+1).*(s.^2/15^2+2*0.8*s/15+1));
```

Figure 7.13. Controller function file f.m for the prefilter (7.7)

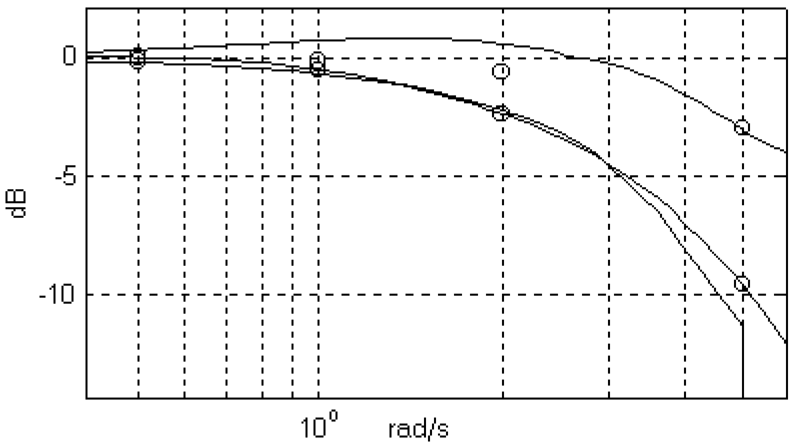

Figure 7.14. The servo specification together with the gain extent and nominal of the complete closed loop from reference r to controlled output y, $F\left(P_1\left(P_2G_2/(1+P_2G_2)\right)G_1\right)/\left(1+P_1\left(P_2G_2/(1+P_2G_2)\right)G_1\right)$.

This concludes our exposé of cascaded design. The user should of course simulate selected plant cases in the frequency and time domains, as was done in the example of Chapter 2.