6. MORE ABOUT DESIGN

Section 2.5 covered the basics of feedback compensator design, including the work with the user's feedback compensator m-file, that was copied from the file `fbcomp.m` in the Qsyn library, and then renamed and edited, and the use of the command `cdesign`. Prefilter design basics were presented in Section 2.7, including the work with a copy of `prefil.m`, and the command `fdesign`. Controller function m-files and commands working with them was further expounded in Section 3.5.

In this chapter a few more useful hints for and applications of design will be shown.

It is often confusing to have a full set of Horowitz bounds on screen when designing the nominal open loop. One would e.g. like to inspect a smaller set of bounds and the corresponding nominal value(s). One way to do so is to step through the frequencies, displaying all bounds belonging to one frequency at a time together with the nominal open loop, in a sort of "slow motion" film. A script m-file for that purpose is suggested in Section 6.1, using the example of Chapter 2.

Sometimes one would like to find a conventional design for a certain system, maybe as a preliminary step for a full design. That is easily done in the Nichols chart with the `cdesign` command, and in a Bode plot with the `fdesign` command. With a certain plant, one can describe the plant transfer function either asthe nominal plant of a plant description file (see `plant.m`) or as a controller function m-file (see `fbcomp.m`). An example is given in Section 6.2.

The help text of `fbcomp.m` includes hints how a digital or sampled controller can be designed with Qsyn. Section 6.3 contains a digital design for the example in Chapter 2.

```
% film.m   script m-file for Example 2.1

% 9 bound frequencies:
w = [.2 .5 1 2 5 7 10 20 50];

% dummy figure window to avoid erasing previous figures:
figure

% loop over frequencies:
for k = 1:9,
      % close previous figure window:
      close;

      showbnd('ex2 1a',[],w(k),'rsrs','m',w(k),'odsrs','r' );
      hngrid, axis([-360 0 -50 50]), mgrid(12,10);
      h1=cdesign('ex2_1a.m','g1',[], [],[],w,logspace(-2,3));

      % highlight nominal open loop value for  current frequency
      h1=cdesign('ex2 1a.m','g1',[],[],'g',w(k), ...
                                          [w(k)-eps,w(k),w(k)+eps]);
      % Enable zooming during the screening:
      hzoom

      pause;
      % Press Return for next frequency, Press Ctrl-C for exit
end
```

Figure 6.1. Script m-file film.m to step through the Horowitz bounds of Example 2.1 according to frequencies, together with the final nominal open loop C from Figure 2.17a.

6.1 Film

Often it is convenient to display, at one time, one or a few Horowitz bounds only, together with the nominal open loop during the feedback design phase. In Figure 6.1, a script m-file is suggested how to step through the frequencies for the example of Chapter 2.

Consider the final feedback design in Section 2.5 (Figure 2.18 and equation 2.4), with the plant description file `ex2 1a.m`, and the bound file `ex2 1a.bnd` (complemented with bounds for 7 rad/s as in Section 5.4.1). Notice that `ex2_1a.m` does not include 7 rad/s as a template frequency, and therefore the tick mark frequencies must be explictly included in the `cdesign` tick mark frequency argument.

The user may copy the m-file in Figure 6.1, and try it out. It is easily adaptable to other design problems.

6.2 Design for a plant without uncertainty

Assume you would like to design a feedback controller for a plant without uncertainty. In Qsyn such a plant can be either modelled as the nominal plant case in a plant description file, or with the help of controller function file that computes the plant transfer fucntion.

6.2.1 Plant described as nominal case in a plant description file

Let the plant without uncertainty be the nominal case of the plant description file `ex2_1a.m` in Figure 2.2, i.e.

$$P(s) = 2 \cdot \frac{s+3}{1 + 2 \cdot 0.6 \cdot s/4 + s^2/16} \quad . \qquad (6.1)$$

Let a feedback compensator candidate be given by the controller function file `g1.m` in Figure 2.18. Then the open loop is displayed in a Nichols chart (Figure 6.2) with the commands

```
wtick = [.2 .5 1 2 5 7 10 20 50]; % tick mark frequencies
hngrid, axis([-360 0 -50 50]), mgrid(12,10);
h1=cdesign('ex2_1a.m','g1',[], [],[],wtick,logspace(-2,3)); % Fig 6.2
```

and as a Bode plot (Figure 6.3) with the commands

```
h1=fdesign('ex2_1a.m','g1',[],'bode',logspace(-1,2));
subplot(211),grid, mgrid([],10)
subplot(212),axis([.1 100 -180 -90]),grid, mgrid([],9)  % Fig. 6.3.
```

The design proceeds by editing the controller file `g1.m`, and displaying the open loop anew, until the chosen classical criteria, such as bandwidth, phase and gain margin, sensitivity, etc are satisfied. Clearly, the command `fdesign` can be used similarly for feedforward design for a plant without uncertainty.

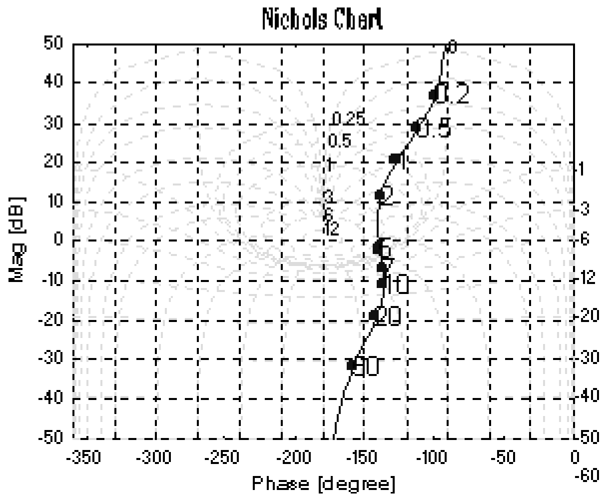

Figure 6.2. Feedback design for a plant without uncertainty in a Nichols chart.

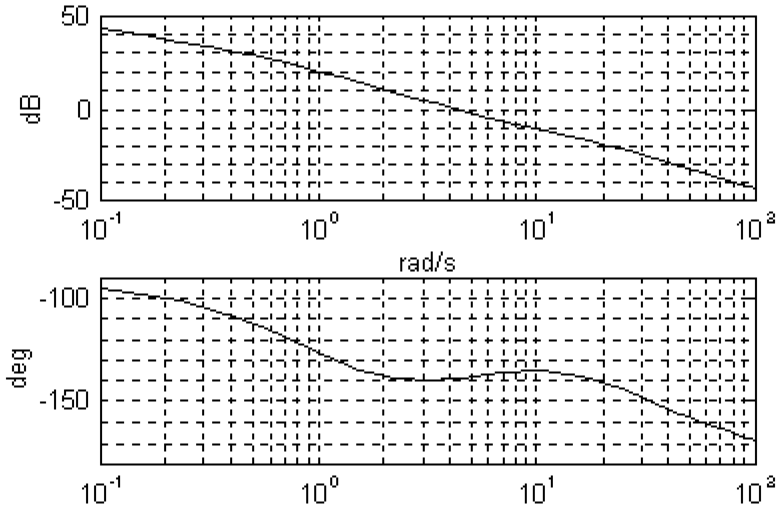

Figure 6.3. Feedback design for a plant without uncertainty in a Bode diagram.

6.2.2 Plant described with a controller function m-file

A plant without uncertainty can also be defined with the help of a controller function m-file, a file type that was defined in Section 3.5. Let the plant be given by (6.1). The controller function m-file `ex6_2.m` in Figure 6.4 computes its frequency function.

```
function [P] = ex6_2(s)

% ex6_2.m Controller function description of plant (6.1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  P = 2*(s+3)./(1 + 2*0.6*s/4 + s.*s/16);
```

Figure 6.4. A controller function realizaton of the certain plant (6.1).

Let a feedback compensator candidate be given by the controller function file `g1.m` in Figure 2.18. Then the open loop is displayed in a Nichols chart (Figure 6.2) with the commands

```
wtick = [.2 .5 1 2 5 7 10 20 50]; % tick mark frequencies
hngrid, axis([-360 0 -50 50]), mgrid(12,10);
h1=cdesign('ex6_2.m','g1',[], [],[],wtick,logspace(-2,3)); % Fig 6.2
```

and as a Bode plot (Figure 6.3) with the commands

```
h1=fdesign('ex6 2.m','g1',[],'bode',logspace(-1,2));
subplot(211),grid, mgrid([],10)
subplot(212),axis([.1 100 -180 -90]),grid, mgrid([],9)  % Fig. 6.3.
```

The design proceeds by editing the controller file `g1.m`, and displaying the open loop anew, until the chosen classical criteria, such as bandwidth, phase and gain margin, sensitivity, etc are satisfied.

6.2.3 Bounds for plants without uncertainty

It should be pointed out that also for plants without uncertainty, some Horowitz bounds can be computed, such as e.g. sensitivity bounds, disturbance rejection bounds, one degree-of-freedom servo bounds, and error and stiffness coefficient bounds. Tolerance bounds have no meaning.

When computing bounds for a plant without uncertainty, then the plant must be defined in a plant definition file, with some infinitesimal dummy uncertainty, e.g. gain uncertainty, see `plant.m` (use the command `type plant.m` to see its contents).

6.3 Digital control

Qsyn allows the design of digital controllers, as outlined in the standard controller function file `fbcomp.m` (use the command `type fbcomp.m` or `help fbcomp.m` to see its contents). In this section we will attempt to design a digital feedback controller for Example 2.1 (plant definition file `ex2 1a.m`, template file `ex2 1a.tpl`, and specification file `ex2_1a.spc`) such that the Horowitz bounds computed in Section 2.4 and Section 5.4.1 and collected in the bound file `ex2 1a.bnd` are satisfied.

Assume that a digital controller with sampling period T_s = 0.020 seconds is to be used, i.e. the sampling frequency is $f_s = 50$ Hz or $\omega_s = 100\pi$ rad/s ≈ 314 rad/s. Therefore an analog anti-aliasing pre-sampling filter, $A(s)$, with a bandwidth of 100 rad/s, is introduced as part of the controller transfer function. We choose

$$A(s) = \frac{1}{1 + 2 \cdot 0.7 \cdot s/100 + s^2/10000} \tag{6.2}$$

defined in the controller function file `d1.m` (Figure 6.5) and whose Bode diagram is displayed in Figure 6.6.

The sampling and hold operation of the digital feedback compensator has the analog transfer function

$$H(s) = \frac{1 - e^{-T_s s}}{T_s s} \tag{6.3}$$

```
function [G] = d1(s)

% d1.m Controller function description of anti-aliasing filter (6.2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  G = [1]./(1 + 2*0.7*s/100 + s.*s/10000);
```

Figure 6.5. A controller function realizaton of the anti-aliasing filter (6.2).

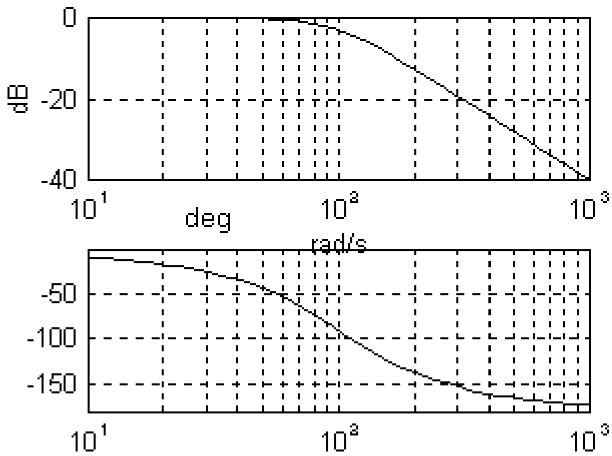

Figure 6.6. Bode diagram of the anti-aliasing filter (6.2) generated by the command `h1=fdesign([],'d1',[],'bode',logspace(1,3));`

which is realized in series with the anti-aliasing filter in the controller function file `d2.m` in Figure 6.7. The Bode diagram of $A(s)H(s)$ is shown in Figure 6.8, but notice that it from now on is meaningful to display the frequency function up to the Nyquist frequency of $50\pi \approx 157$ rad/s, only.

```
function [G] = d2(s)

% d2.m Controller function description of anti-aliasing filter (6.2)
%      in series with the sample and hold transfer function (6.3)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  A = [1]./(1 + 2*0.7*s/100 + s.*s/10000);% (6.2)
  Ts = 0.020;                             % seconds, sampling period
  H = (1 - exp(-Ts*s))./(Ts*s);           % (6.3)
  G = A.*H;
```

Figure 6.7. A controller function realizaton of the anti-aliasing filter (6.2) in series with the sample and hold transfer function (6.3).

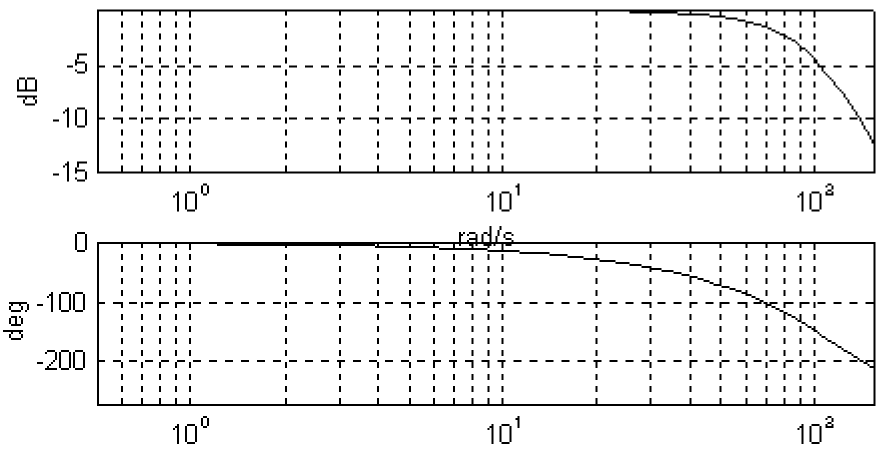

Figure 6.8. Bode diagram of A(s)H(s) where A(s) is the anti-aliasing filter (6.2) and H(s) is the sample and hold transfer function (6.3), generated by the command
`h1=fdesign([],'d2',[],'bode',50*logspace(-2,pi));`

Now comes the stage to design the digital feedback compensator pulse transfer function as a function of *z*. With $z = e^{sT_s}$ one gets its frequency function. It is of course possible to design directly in the *z*-domain with first, second or higher order polynomials in *z* for the numerator and denominator.

It is however often conveniant to "think in continuous time" or in the *s*-domain and translate the compensator to the *z*-domain. There exist many methods to do this, some of them very bad. We recommend the so called modal or matched pole-zero translation (see e.g. the Matlab Control System Toolbox command `c2dm`), which means that the numerator and denominator polynomials are factorized, and each factor in the *s*-domain is translated according to

$$(s-a) \quad \rightarrow \quad \left(z - e^{aT_s}\right) \qquad (6.4)$$

whether *a* is complex or real, with a final adjustment of the compensator function DC-gain. The translation (6.4) means that controller poles and zeros are translated to their corresponding sampled locations. Note, that the imaginary part of *a* is restricted to within $\pm\omega_s/2$.

Our first design attempt will be a modal translation of the analog compensator in Figure 2.16,

$$G(s) = \frac{\left(1 + 2\cdot 0.6s/4 + s^2/16\right)}{s(1+s/3)} \quad \rightarrow$$

$$\rightarrow \quad C(z) = g\frac{\left(z - e^{(-2.4+3.2j)T_s}\right)\left(z - e^{(-2.4-3.2j)T_s}\right)}{(z-1)\left(z - e^{-3T_s}\right)} =$$

$$= g\frac{\left(z^2 - 2e^{-2.4T_s}\cos(3.2T_s)z + e^{-4.8T_s}\right)}{(z-1)\left(z - e^{-3T_s}\right)} \qquad (6.5)$$

with the adjustment of the gain *g* to be done in the Nichols chart with respect to the Horowitz bounds. $C(z)H(s)A(s)$ is realized in the controller function file `d3.m`, see Figure 6.9.

```
function [G] = d3(s)

% d3.m Controller function description of anti-aliasing filter (6.2)
%      in series with the sample and hold transfer function (6.3)
%      and the digital feedback compensator (6.5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  A = [1]./(1 + 2*0.7*s/100 + s.*s/10000);% (6.2)
  Ts = 0.020;                             % seconds, sampling period
  H = (1 - exp(-Ts*s))./(Ts*s);           % (6.3)
  z = exp(Ts*s);
  C = 0.4*(z.*z - 2*exp(-2.4*Ts)*cos(3.2*Ts)*z + exp(-4.8*Ts))./ ...
        ((z - 1).*(z - exp(-3*Ts)));

  G = A.*H.*C;
```

Figure 6.9. A controller function realizaton of the anti-aliasing filter (6.2) in series with the sample and hold transfer function (6.3) and the digital feedback compensator (6.5).

We are now in the position to display the compensated nominal open loop like in Section 2.5, or above in Section 6.2,

```
wtick = [.2 .5 1 2 5 7 10 20 50]; % tick mark frequencies
hngrid, axis([-360 0 -50 50]), mgrid(12,10);
showbnd('ex2_1a',gcf,[.2 .5 1 2],'rsrs','roll', ...
  [5 7 10 20 50],'odsrs','roll' );  % roll refers to rolling colors
h1=cdesign('ex2_1a.m','d3',[], [],[],wtick,50*logspace(-3,pi));
```

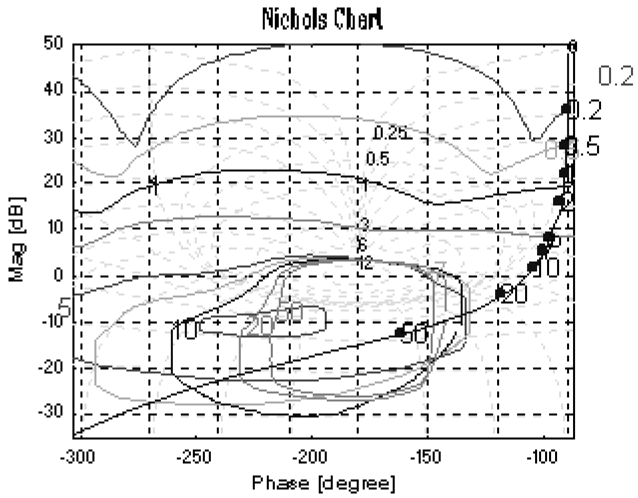

Figure 6.10. Compensated open loop with nominal plant from `ex2_1a.m`*, bounds from* `ex2_1a.bnd`*, and digital compensator function* `d3.m`*.*

The result in Figure 6.10 is not that impressive: another lag, like in (2.4) seems to be necessary. The revised controller function is stored as `d4.m`, see Figure 6.11, with

$$C(z) = g\frac{\left(z-e^{-5T_s}\right)\left(z^2-2e^{-2.4T_s}\cos(3.2T_s)z+e^{-4.8T_s}\right)}{(z-1)\left(z-e^{-3T_s}\right)\left(z-e^{-T_s}\right)} \qquad (6.6)$$

```
function [G] = d4(s)

% d4.m Controller function description of anti-aliasing filter (6.2)
%      in series with the sample and hold transfer function (6.3)
%      and the digital feedback compensator (6.6)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  A = [1]./(1 + 2*0.7*s/100 + s.*s/10000);% (6.2)
  Ts = 0.020;                              % seconds, sampling period
  H = (1 - exp(-Ts*s))./(Ts*s);            % (6.3)
  z = exp(Ts*s);
  C = 0.09* ...
        ((z - exp(-5*Ts)).* ...
         (z.*z - 2*exp(-2.4*Ts)*cos(3.2*Ts)*z + exp(-4.8*Ts)))./ ...
        ((z - 1).*(z - exp(-3*Ts)).*(z - exp(-Ts)));

  G = A.*H.*C;
```

Figure 6.11. A controller function realizaton of the anti-aliasing filter (6.2) in series with the sample and hold transfer function (6.3) and the digital feedback compensator (6.6).

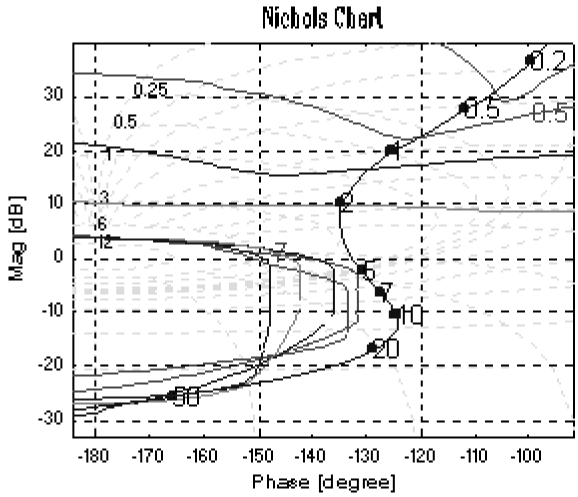

Figure 6.12. Compensated open loop with nominal plant from `ex2_1a.m`*, bounds from* `ex2_1a.bnd`*, and digital compensator function* `d4.m`*.*

The command

```
h1=cdesign('ex2_1a.m','d4',h1, [],[],wtick,50*logspace(-3,pi));
```

then yields Figure 6.12. The design seems to be satisfactory with respect to the bounds; however the prefilter design remains, and then the design must be checked in the time domain as in Chapter 2. Notice that the digital design in (6.6) is very similar to the analog design in (2.4). Notice also the bandwidth limitation in the digital design, visible in Figure 6.12 as a strong phase decrease for frequencies above 20 rad/s, due to the equivalence between sampling and time delay.

When designing a digital prefilter, it is not necessary to include a presampling filter, and a sampling and hold device, if the reference signal is generated in the computer, or received as a sampled signal.