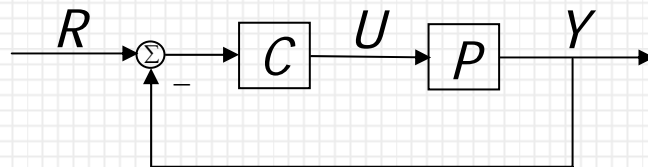


7. Completing a Design: Loop Shaping

Now that we understand how to analyze stability using Nichols plots, recall the design problem from Chapter 5: consider the following feedback system



where

$$\mathcal{P} = \left\{ P(s) = \frac{k}{s(s+a)} : k \in [1, 10], a \in [1, 10] \right\}.$$

The objective is to design a controller C such that it achieves

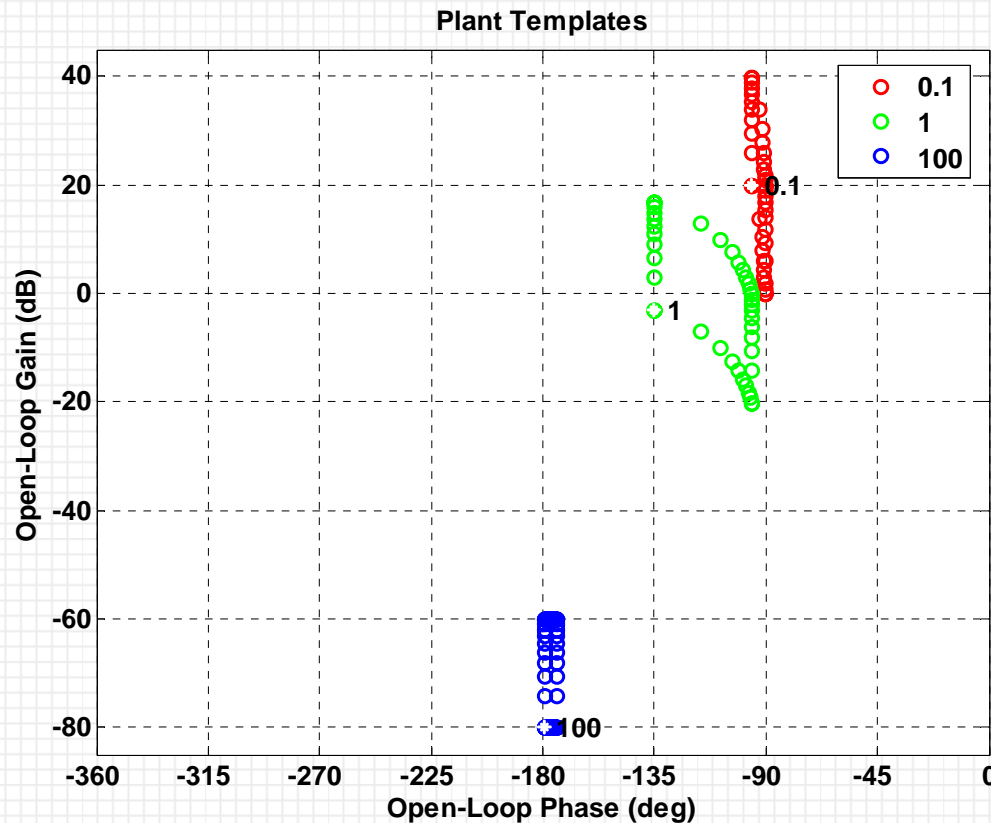
- Robust stability margins of $PM \geq 50^\circ$ and $GM \geq 1.83$.
- $C(0) = 100$.

We saw that a constraint on the peaking in the complimentary sensitivity function T (or in S) is a more reasonable way to deal with stability margins. In this example we obtained:

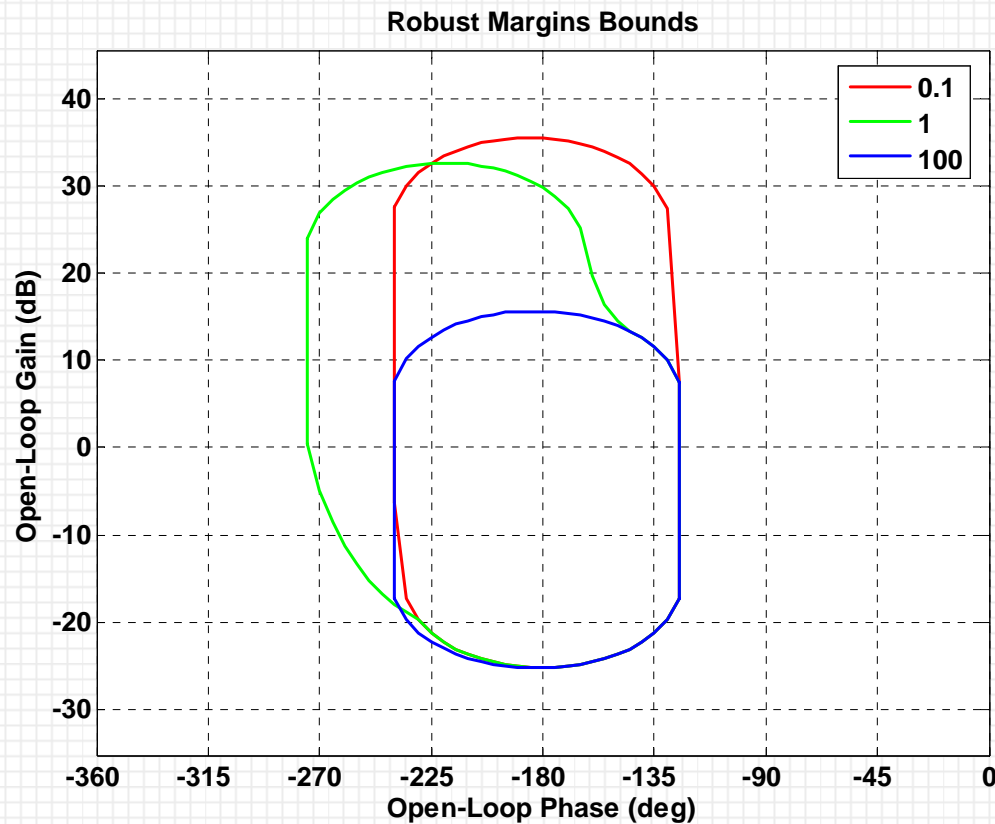
$$|T(j\omega)| = \left| \frac{L(j\omega)}{1+L(j\omega)} \right| \leq 1.2, \quad \omega \geq 0, \quad \forall P \in \mathcal{P}.$$

We are now ready to attempt a design of a stabilizing controller which also meets the above specifications.

1. Run `ch5_t.m` and enter a few frequencies for template generation. As a reminder, the shape of templates is often asymptotic with frequency. So choose one low frequency, one high frequency and a few mid-frequency points to capture shape variations (use different ones from mine!).



2. Run `ch5_b.m` to compute the robust margin bounds. You will need to enter your nominal plant index (I used a 40-point grid and case 1 for nominal plant).



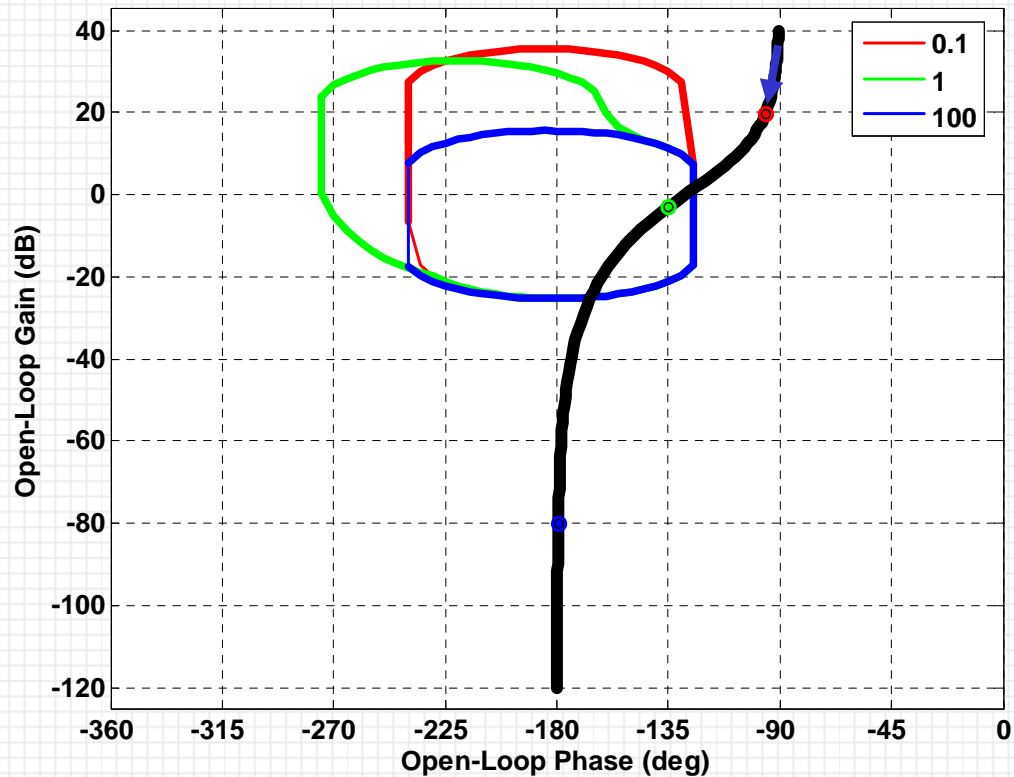
7.1. Loop Shaping

Loop shaping is a procedure where you add and modify the controller elements until the nominal loop is stable and satisfies its bounds.

3. Run `ch7_lp.m` which executes the following commands:

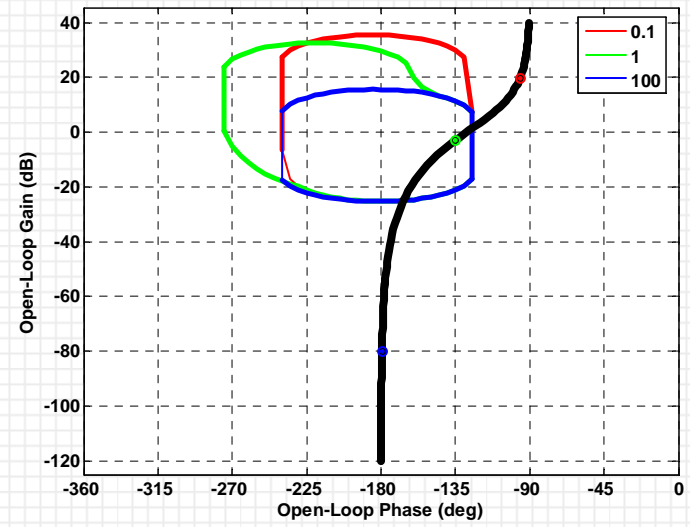
```
wl = logspace(-2,3,100);  
lpshape(wl,bdb1,P(1,1,nompt));
```

You will see something like this (depending on your choice of nominal point and frequencies for computing bounds). We did not pass an initial controller, so at present $C = 1$.



Stability analysis:

What do we know?

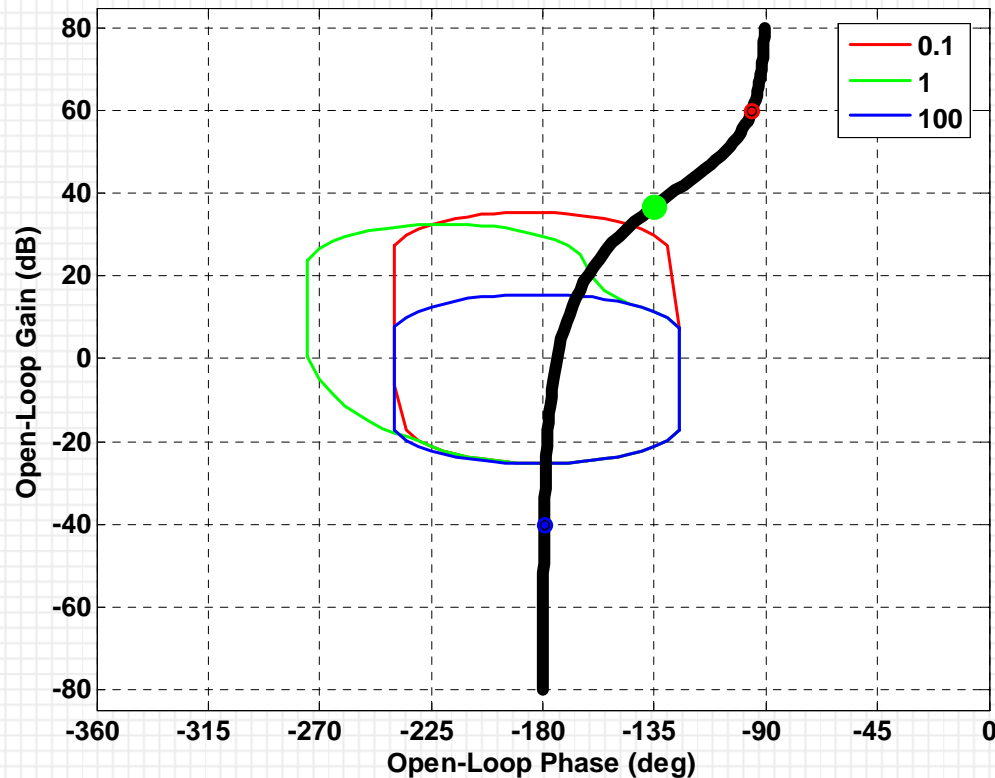


7.1.1 Loop Shaping Elements

The Toolbox supports the following continuous-time elements for nominal loop design ("shaping"):

Element	Form
Real pole	$\frac{1}{s/p+1}$
Real zero	$\frac{s}{z} + 1$
Complex pole	$\frac{1}{s^2/\omega_n^2 + 2\zeta s/\omega_n + 1}$
Complex zero	$s^2/\omega_n^2 + 2\zeta s/\omega_n + 1$
Super 2nd (2/2)	$\frac{a_1 s^2 + a_2 s + 1}{b_1 s^2 + b_2 s + 1}$
Integrator (n>0) or Differentiator (n<0)	$\frac{1}{s^n}$
Lead or Lag	$\frac{s/z+1}{s/p+1}$
Notch	$\frac{s^2/\omega_n^2 + 2\zeta_1 s/\omega_n + 1}{s^2/\omega_n^2 + 2\zeta_2 s/\omega_n + 1}$
Complex lead/lag	$\frac{s^2 + 2abs + a^2}{s^2 + 2bds + b^2} \frac{b^2}{a^2} \quad (d = 0.45)$

The steady-state spec requires that $C(0)=100$. From Select Elements to Tune menu, highlight Gain entry (value of 1 at this point), enter inside the k box the desired value, 100 in this example, then <CR>, then Apply . To optimize axis values, from the pull down menus do View|Full.



The nominal loop response does not exhibit any crossings indicating nominal closed-loop stability (stability can also be checked analytically from Tools|Stability). Since all bounds are satisfied, let us evaluate our design with respect to plant gridding and frequency discretization.

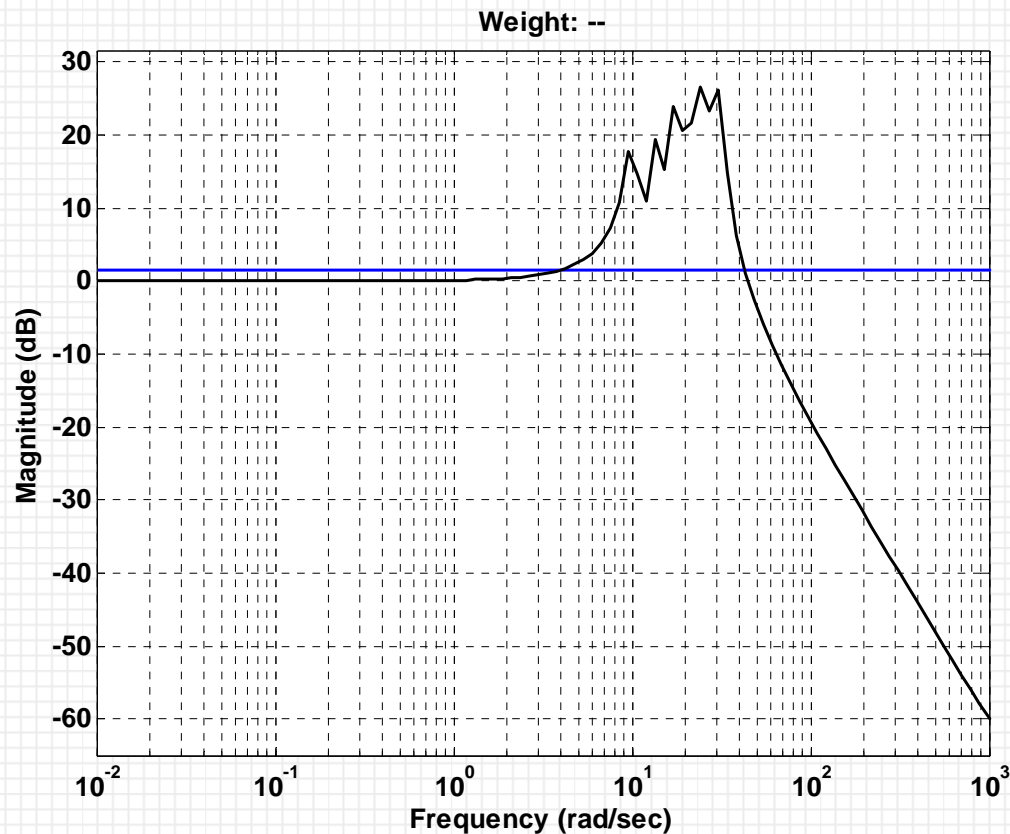
Before proceeding to analysis, you should pass your present design to the workspace via File|Export and the notation C (note the other I/O options). Check the workspace for this C.

7.2. Analysis

4. Run `ch7_a.m` which executes the following commands:

```
chksiso(1,w1,W1,P,[],C);
```

You will see something like this.

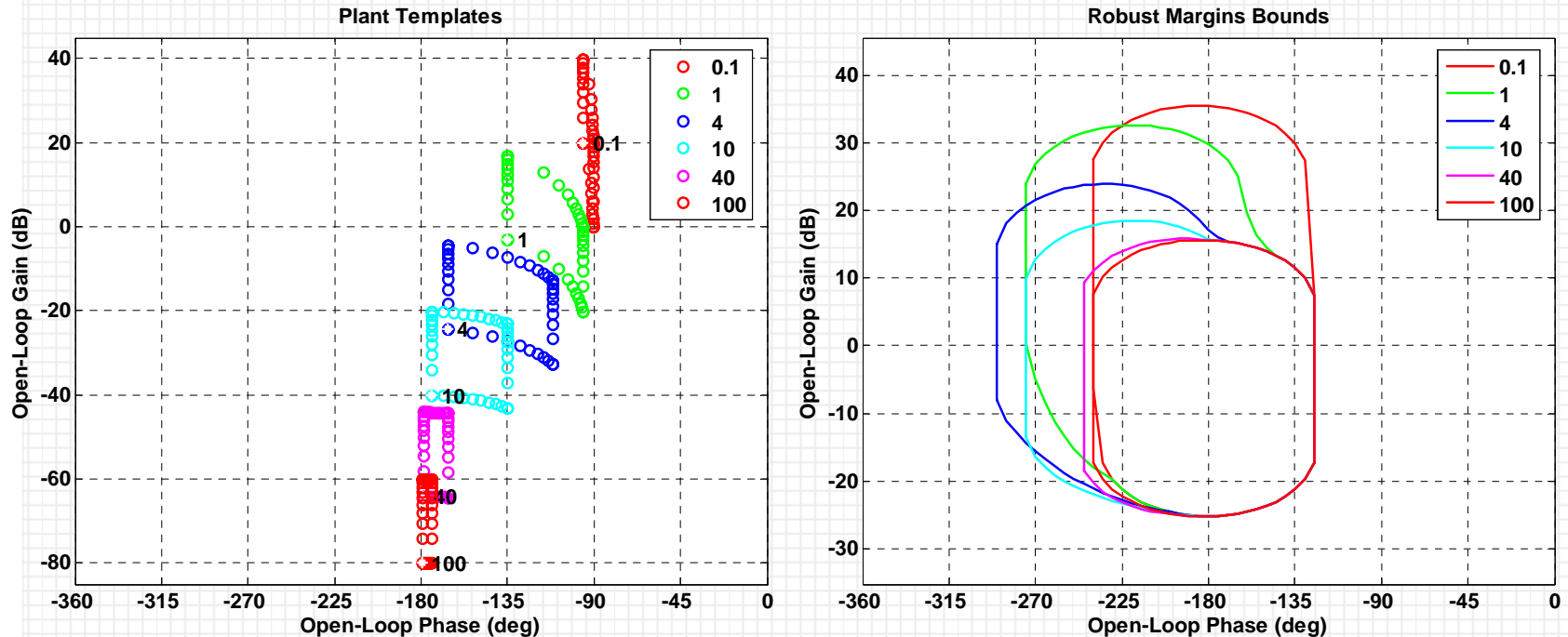


Clearly, this design does not meet the margin spec in the range [4,40] rad/sec. Why? Returning to the LPSHAPE window, we observe a violating of the high-freq (i.e., 100 r/d) bound in the range [1,40]. We could go back and add some lead in this freq range assuming this bound to be effective for $\omega > 4$. This might work at times, but let's look at a more comprehensive approach.

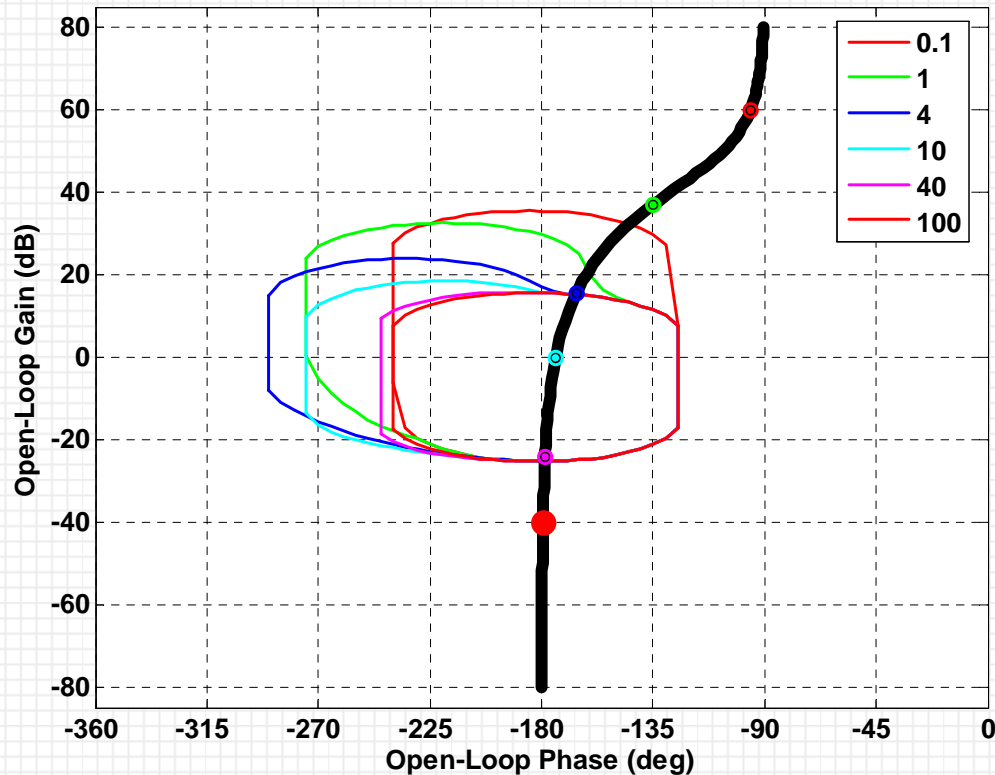
In general, missing a weight is a result of two possible reasons:

- Template discretization and /or
- Frequency discretization (too few bounds).

So let us add a few more bounds in that range, say $[.1, 1, 4, 10, 40, 100]$. Run `ch5_t.m`, followed by `ch5_b.m`. This is what you should see.



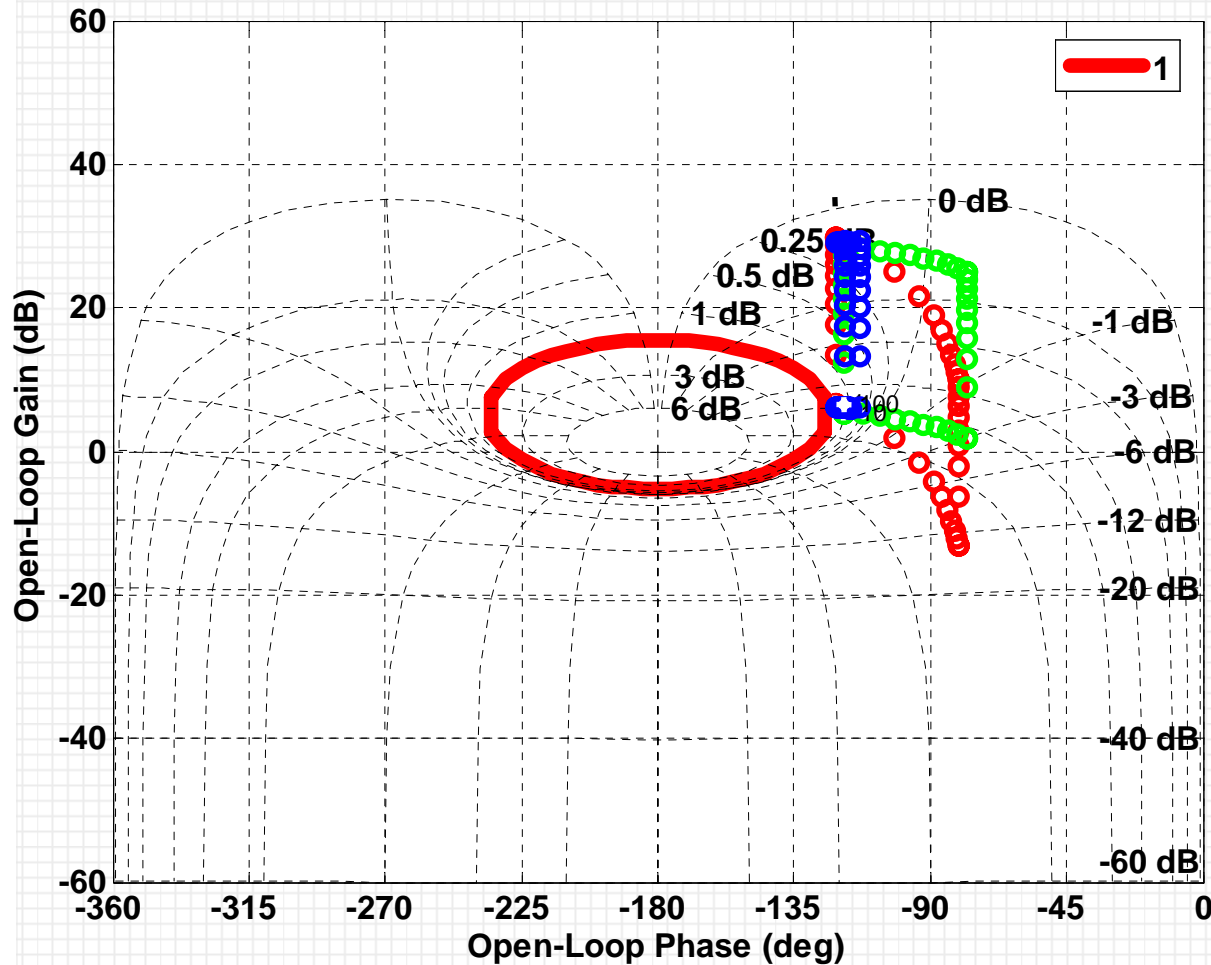
Now start again loop shaping using `ch7.lpm`. Tune the gain to 100 as before. Now you can clearly observe the bound violation at $w=[4, 10, 40]$ which resulted in missing the margin spec.



Note that the right-hand side of the bounds for $\omega > 1$ is fixed. Hence, in this example, we didn't need to compute the additional bounds. What we needed was a shift of the nominal loop to the right of the high-frequency bound (lead).

Why did the bounds conveniently “stretch” to the left allowing us to get away with only a few bounds?

Plant Templates



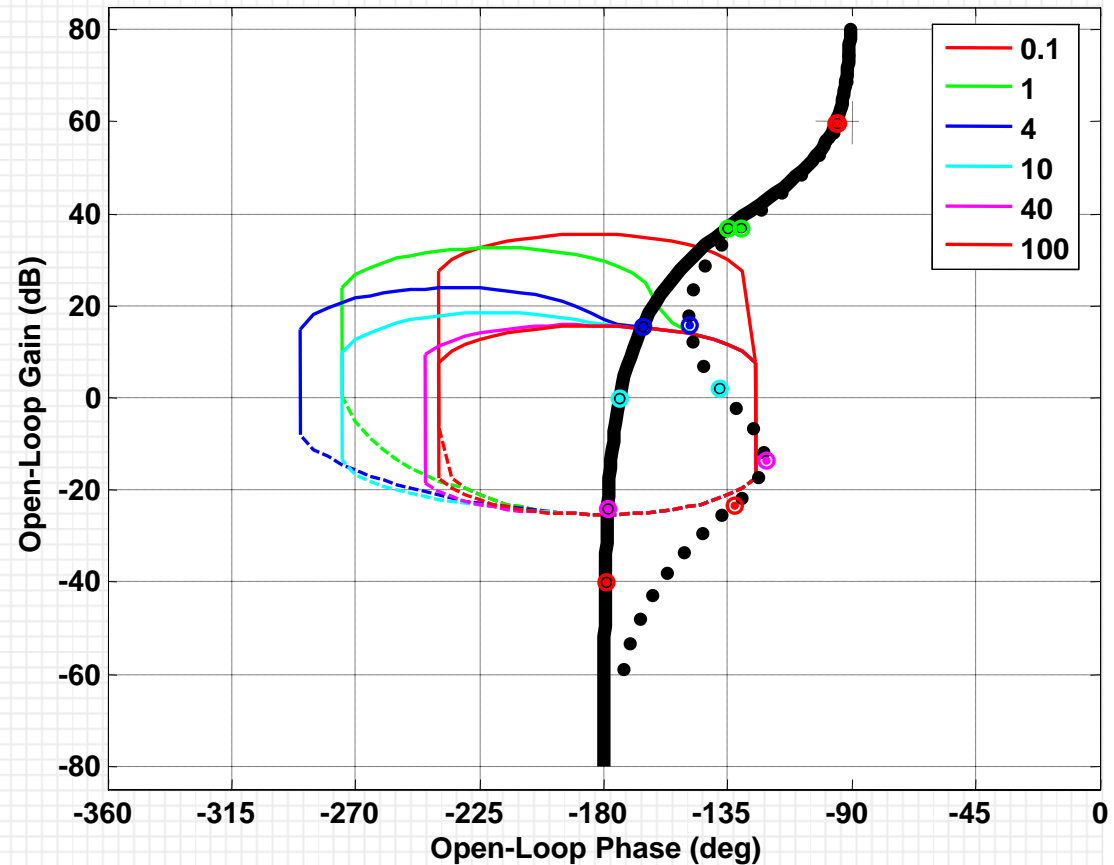
7.3. Final Design Steps

You can shift the loop to the right interactively or by entering numerical values for specific elements. Adding phase can be accomplished using several elements: real zero, lead, complex zero and 2/2. Adding a new element is done by horizontally dragging the loop to the left (1st or 2nd order pole, lag and 2/2) or right (1st or 2nd order zero, lead and 2/2). The program will match the desired phase change (but not the mag change, except for 2/2) and implement ONLY stable elements. Also, the maximum change is limited, e.g., a real zero/pole element can achieve $< \pm 90^\circ$ phase change. Once you add an element you can see its numerical value interactively displayed to the right.

Let's try it. For example, select a Lead/Lag element, then point the cursor on the loop at $\omega = 40$. You should notice the cursor type changing when you are on "top" of the loop. Now drag the loop to the right until it clears the bound as shown below.

Observations:

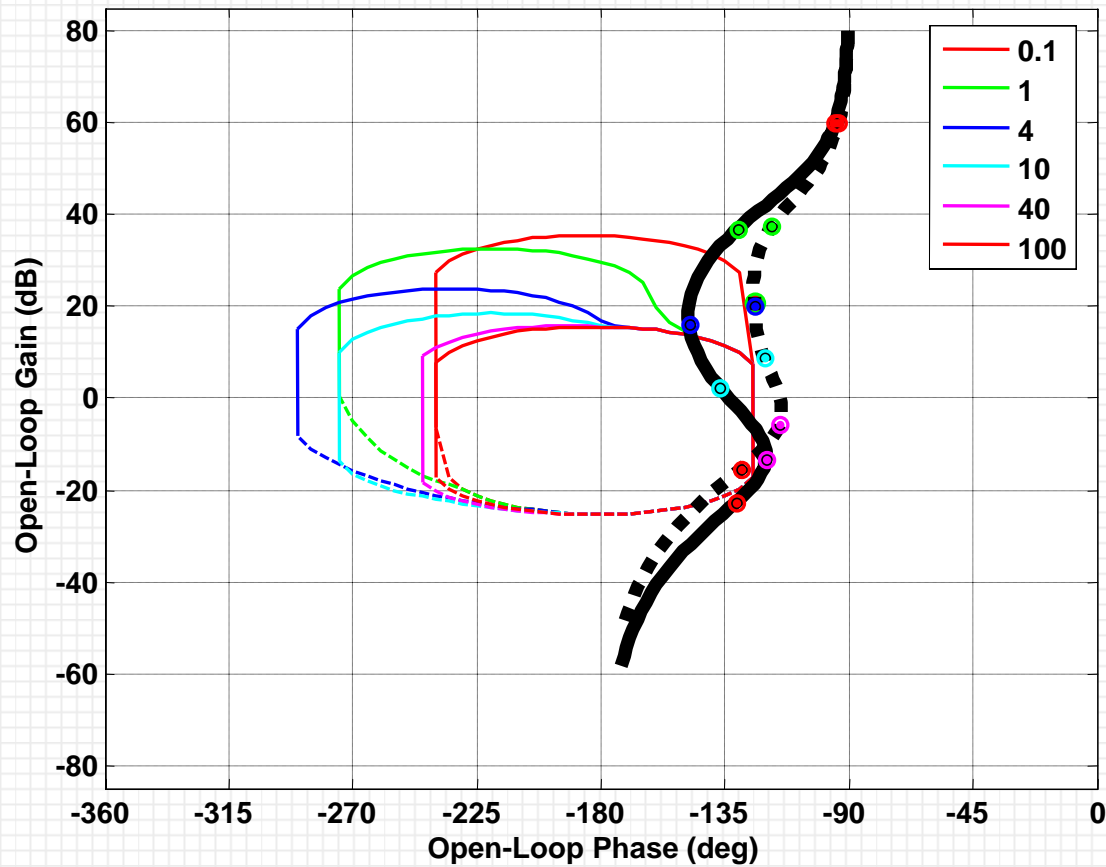
- The dashed response is the new one. Clicking Apply will formally replace the previous design with this new one.



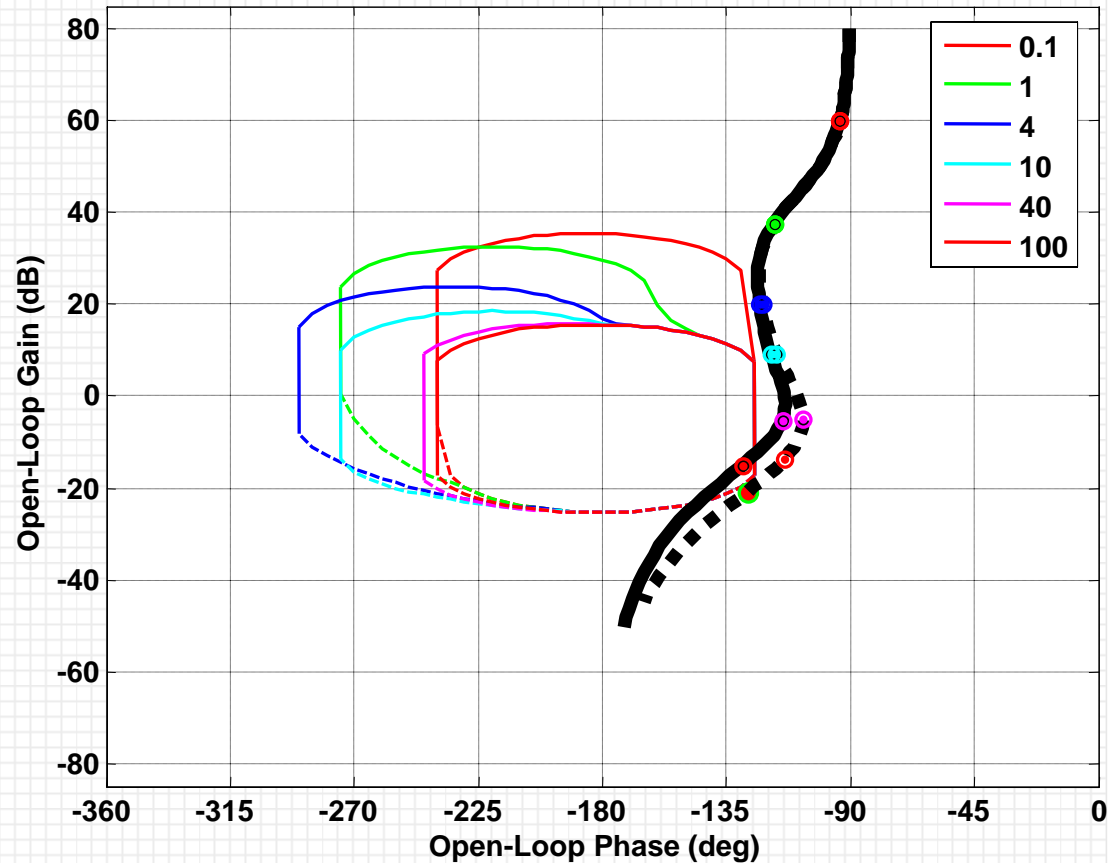
- The loop at $w = 40$ has been shifted to the right due to the lead element. The list on the right shows a new Lead/Lag element $[57.93, 40]$ $[z=11.5, p=139.2]$. This means a lead element with max phase increase of 57.93° at 40 rad/sec achieved with a zero at 11.5 and a pole at 139.2 (see notation on page 7-8).

- We're still violating the bounds in the range $[1, 10]$.

Select Apply and further shift the response to avoid the bounds using a Lead/lag element drag again the loop at a point, say half way in the range $\omega = [1, 40]$. After Apply I got this.



One additional Lead element in a higher frequency range is what we need as shown below. One additional lead does the trick. *Also, you can highlight any specific element already in your design, then fine tune it using the small sliders next to each of its parameters.*

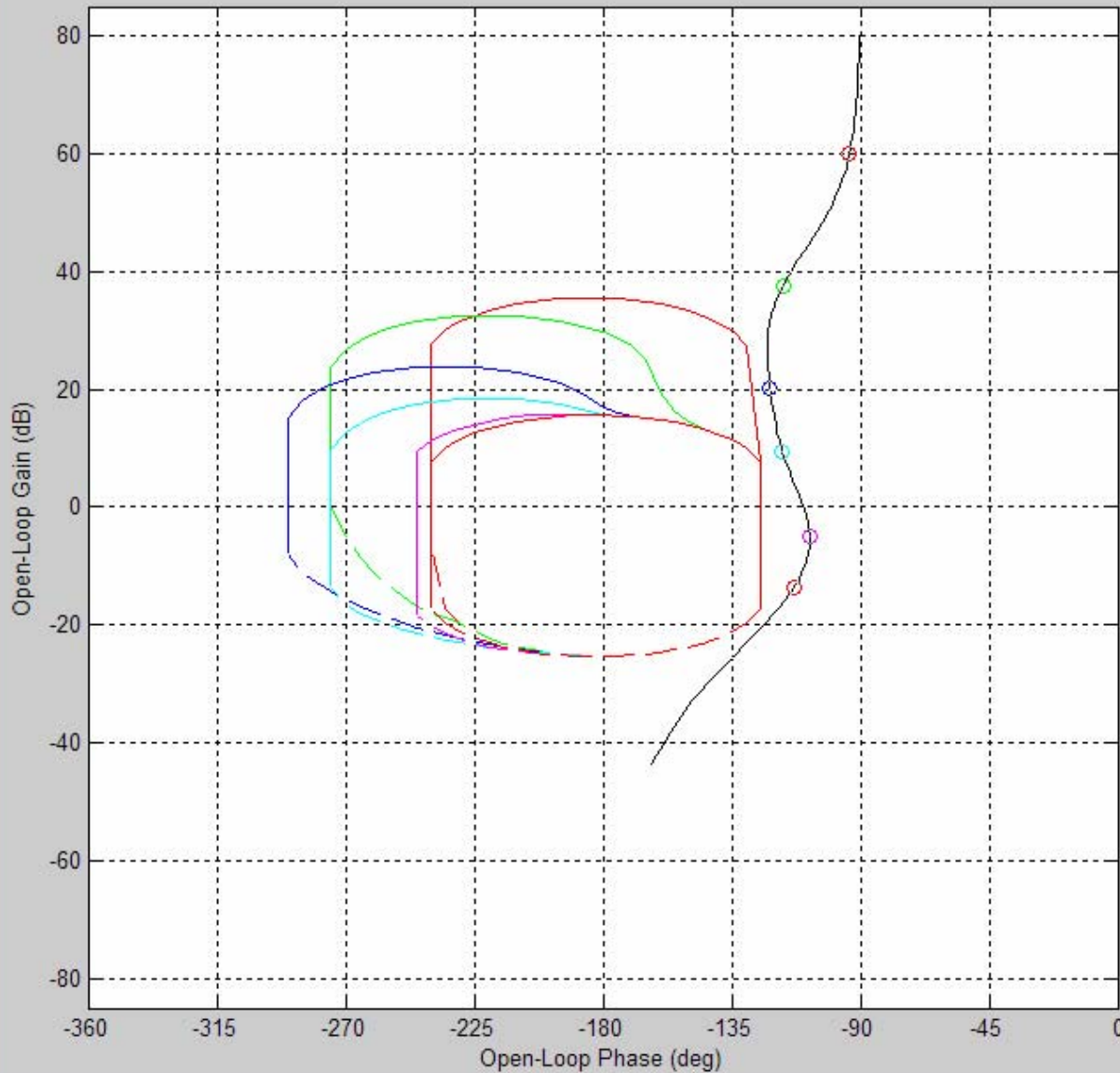


$$C = 100 \frac{\left(\frac{s}{150.7} + 1\right) \left(\frac{s}{11} + 1\right) \left(\frac{s}{2.7} + 1\right)}{\left(\frac{s}{255.8} + 1\right) \left(\frac{s}{145.7} + 1\right) \left(\frac{s}{6} + 1\right)}$$

This 3rd-order controller appear to do the job, but we need some analysis.

Continuous-time Loop Shaping

File View Tools



Pointer Info

Open-Loop: -22.50 deg, -80.06 dB

Closed-Loop: -22.50 deg, -80.06 dB

Frequency: n/a rad/sec

Rad/Sec

Hertz

Controller Elements

Select elements to Add.

Lead or Lag

phase

22.38

w

220.5131

Add using Input Fields

Apply

Cancel

Select elements to Tune.

Gain: [100]

Lead/Lag: [22.38, 220.5] [z=147.7, p=329]

Lead/Lag: [57, 40] [z=11.85, p=135]

Lead/Lag: [26.64, 4] [z=2.468, p=6.482]

Delete

Reduction

Bounds

0.1,r (on)

1,g (on)

4,b (on)

10,c (on)

40,m (on)

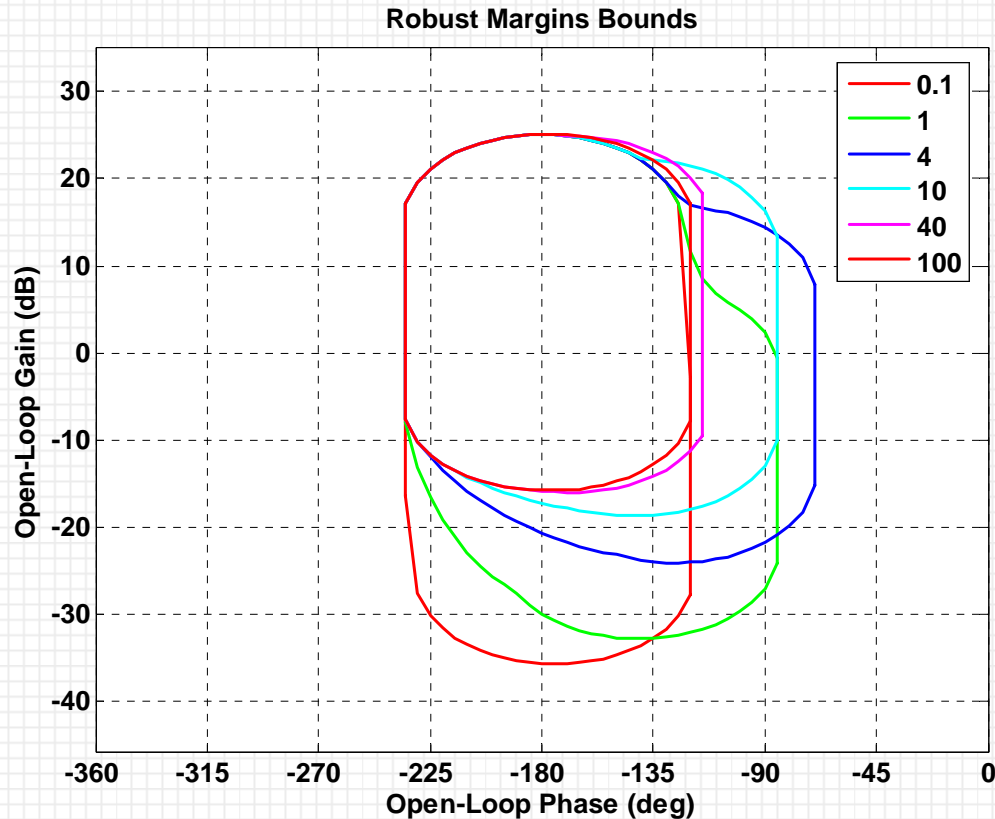
100,r (on)

On/Off

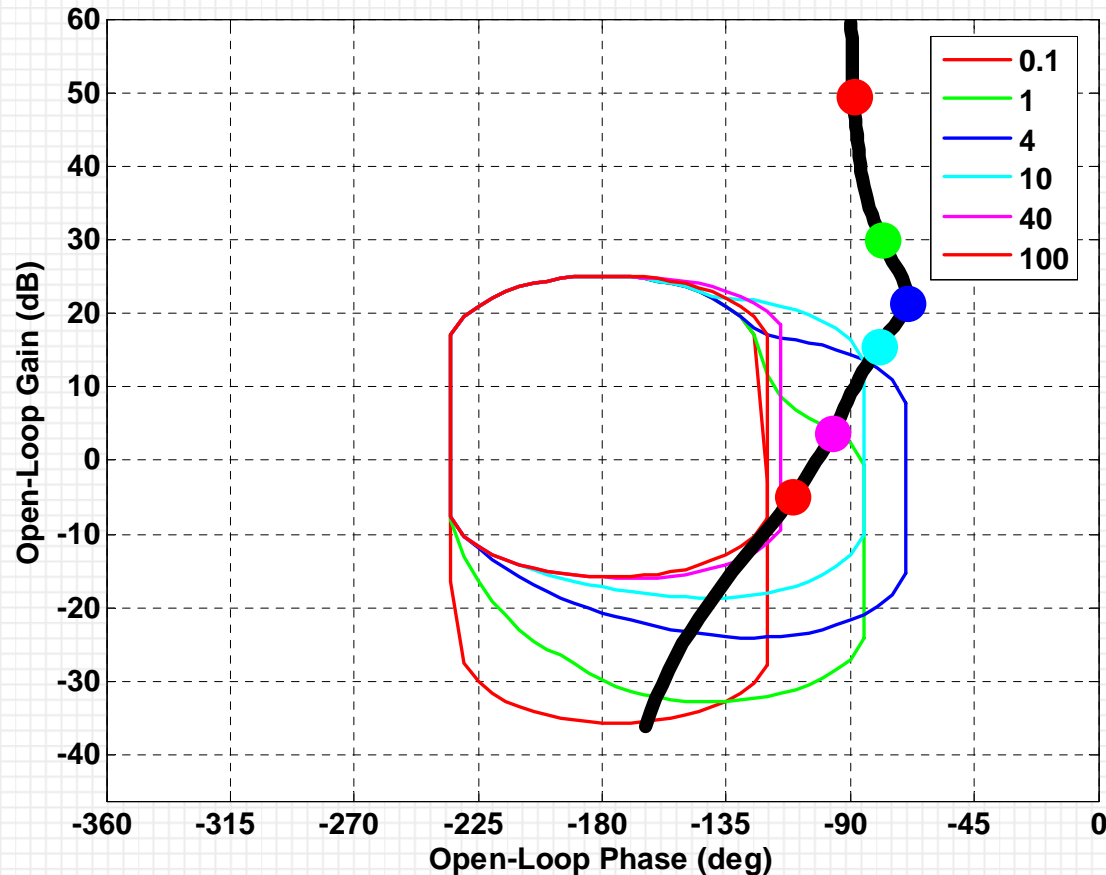
All On

All Off

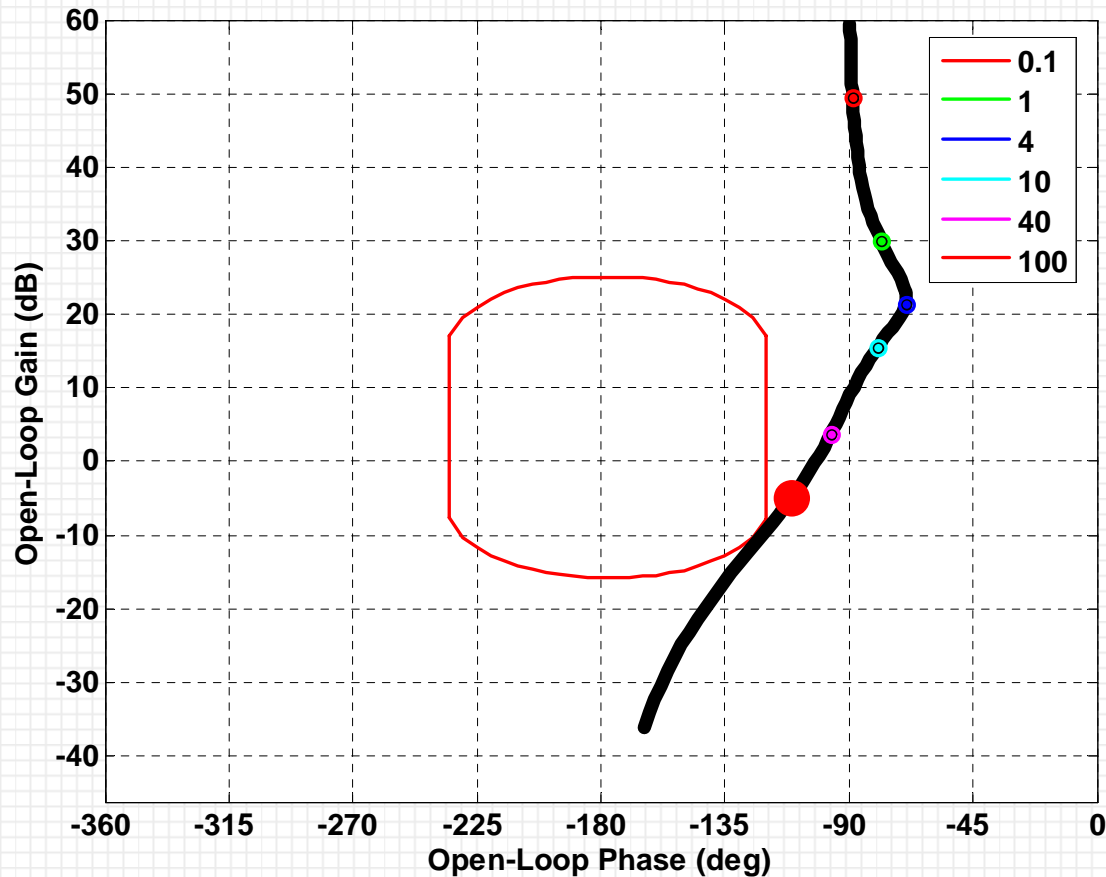
Before we proceed to evaluate the design, you need to become convinced that the choice of nominal plant is not important. Save the controller in a QFT Toolbox style File|Save. Run `ch5_b.m` again but enter a different nominal plant index. Originally I used 1 and now 33. Note that the bounds have same shape but different orientation.



Now run again the loop shaping function `ch7_lp.m` and retrieve our design File|Open.

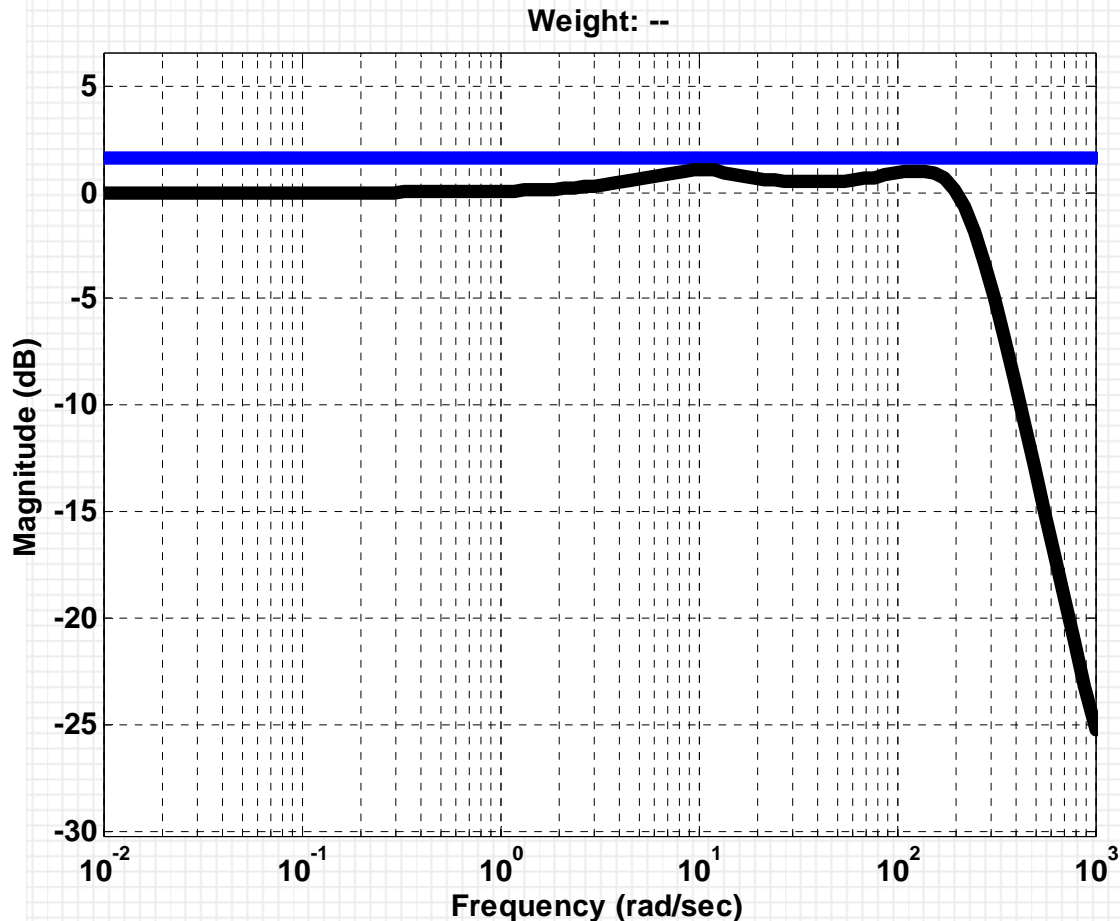


Toggle on/off the various bounds to convince yourself that indeed the bounds are satisfied. For example



Conclusion: if the nominal loop satisfies its bounds, the same controller will do the job with any other nominal plant (with appropriately computed bounds).

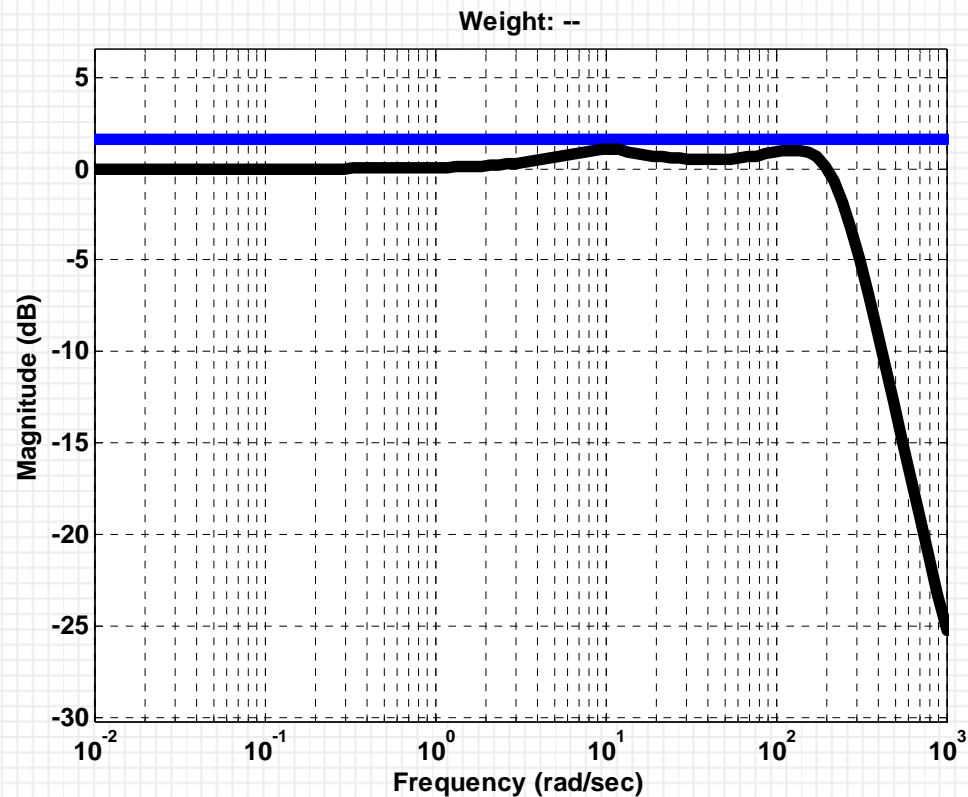
Back to analysis. Run `ch7_a.m` once more. Make sure the controller is sent to the workspace as `C` using `File|Export`



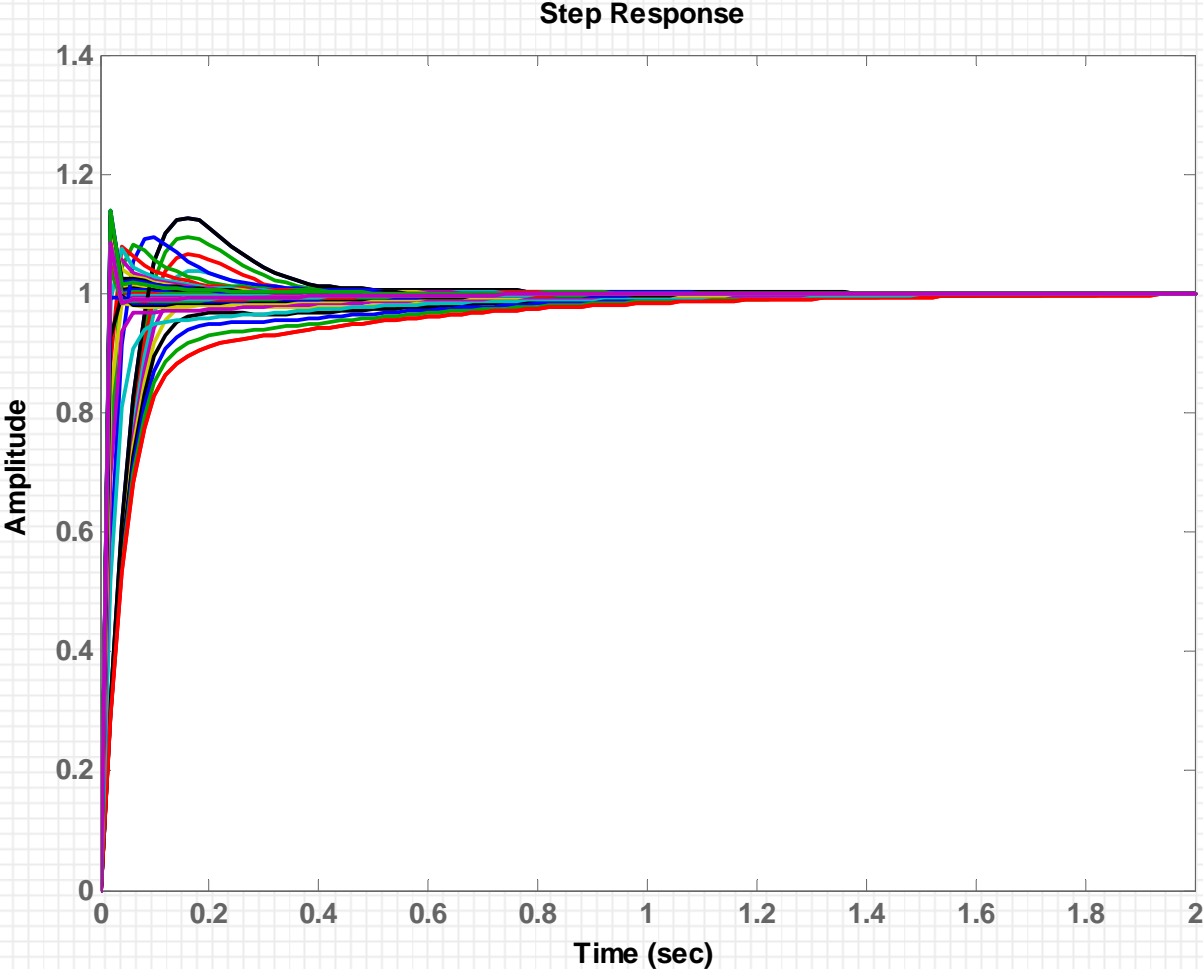
As expected, this design satisfied the robust margin spec. In fact, we have an over-design: the nominal loop is away from its bounds.

What about errors due to plant gridding? Let's take a closer look. Set `ipt=1000` in `ch5_t.m` to define an LTI array with 4000 plants (compared with the initial 40).

You can re-compute the bounds, then check if they are visibly different from the previous set (they are not). Or simply evaluate closed-loop response by running `ch7_a.m`.



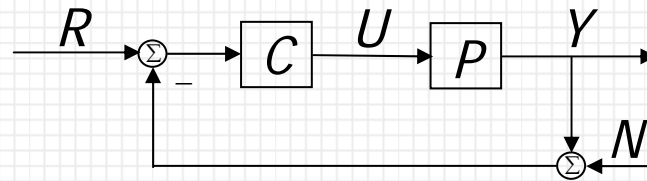
Step reference responses of the closed-loop system reveal that we have both over- and under-damped dynamics.



You think you are done? Maybe so if this was a homework problem. As it turns out, there are many (actually infinite) controllers that solve this problem. Let us try to learn what make one design better than another with respect to a specific figure of merit.

7.4. Sensor Noise Problem

Many feedback systems are sensitive to high-frequency sensor noise and un-modeled dynamics. Consider the block diagram is shown below which includes sensor noise



We are particularly interested in the effect of sensor noise at the plant input, U , since plant inputs are always limited. We have this relation

However, at the same freq range

To compute closed-loop responses, execute the following commands. To compute tracking responses

```
Try=feedback(C*P,tf(1,1));  
% this is less efficient: Try = P*C/(1+P*C);  
w = logspace(-1,5);  
Tryfr = squeeze(freqresp(Try,w));
```

Note:

```
>> size(Try)  
40x1 array of zero-pole-gain models  
Each model has 1 output and 1 input.
```

```
>> size(freqresp(Try, w))  
ans =  
     1     1    50    40
```

```
>> size(Tryfr)  
ans =  
    50    40
```

To compute noise effect at plant input u

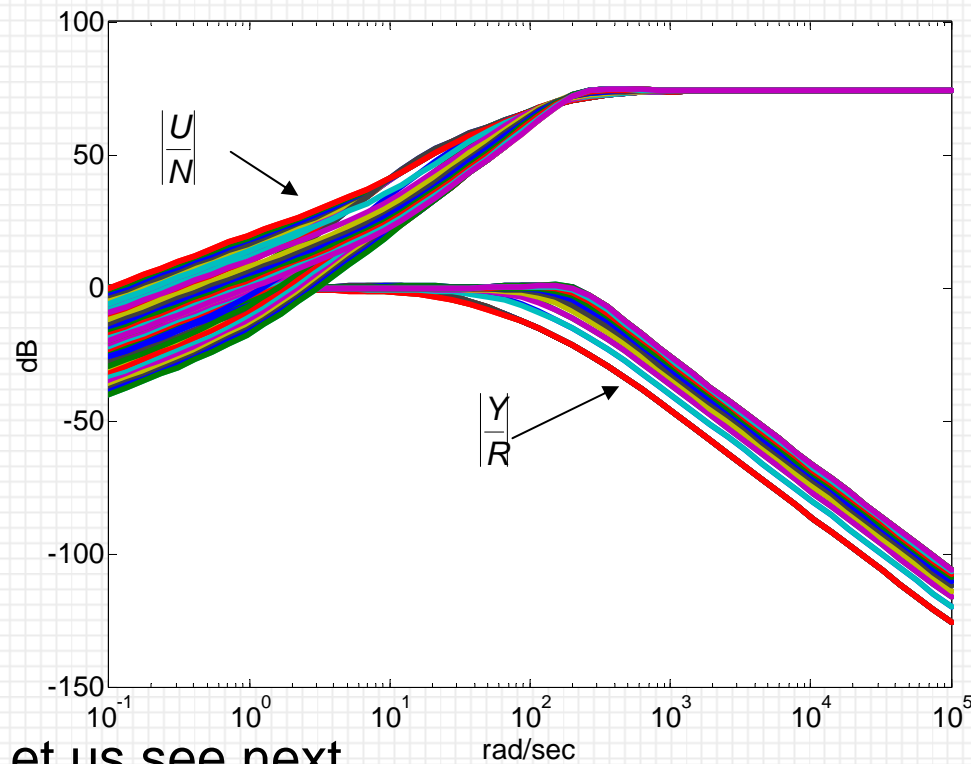
```
Tnu = -feedback(C,P);
```

```
Tnufr = squeeze(freqresp(Tnu,w));
```

```
semilogx(w1,20*log10(abs(Tnyfr)));
```

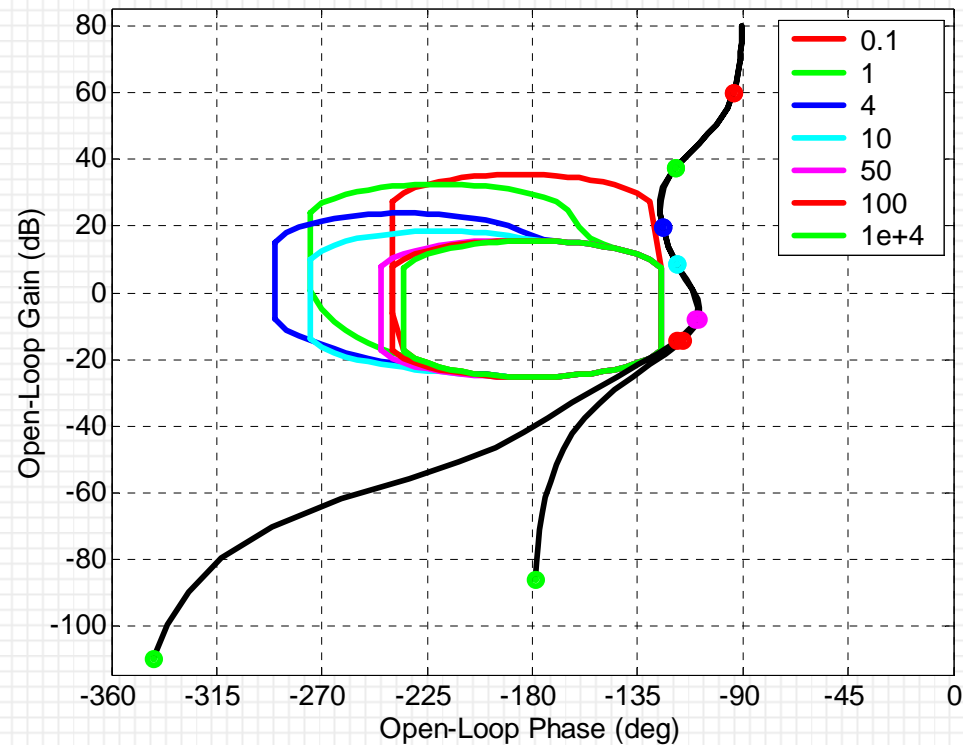
hold on

```
semilogx(w1,20*log10(abs(Tnufr)));
```



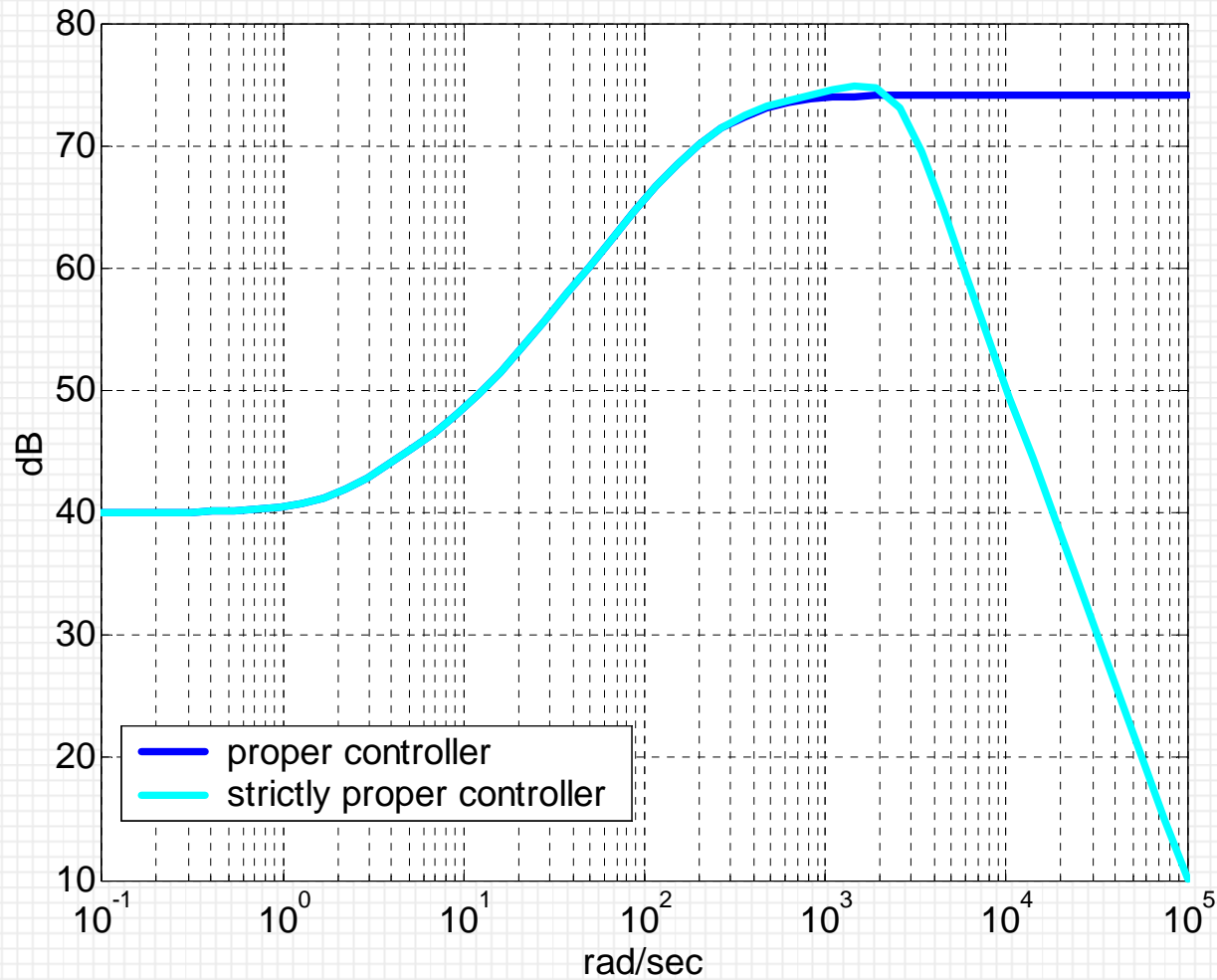
So what? Let us see next.

Recall that our controller was proper. Let us modify it such that while all the specs remain satisfied, we force $|C(j\omega)|$ to approach zero at high frequencies. One way to do that is to add a complex pole with .55 damping and 2500 rad/sec natural frequency.

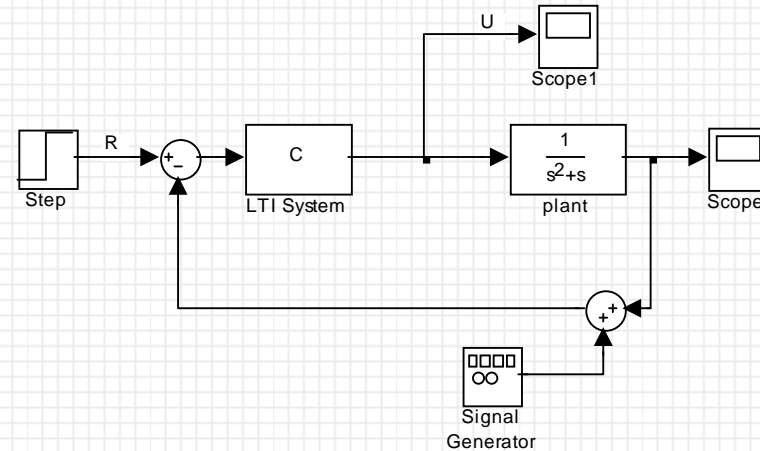


$$C = 100 \frac{\left(\frac{s}{150.7} + 1\right) \left(\frac{s}{11} + 1\right) \left(\frac{s}{2.7} + 1\right)}{\left(\frac{s}{255.8} + 1\right) \left(\frac{s}{145.7} + 1\right) \left(\frac{s}{6} + 1\right) \left(\frac{s^2}{2500^2} + \frac{2 \cdot 0.55 \cdot s}{2500} + 1\right)}$$

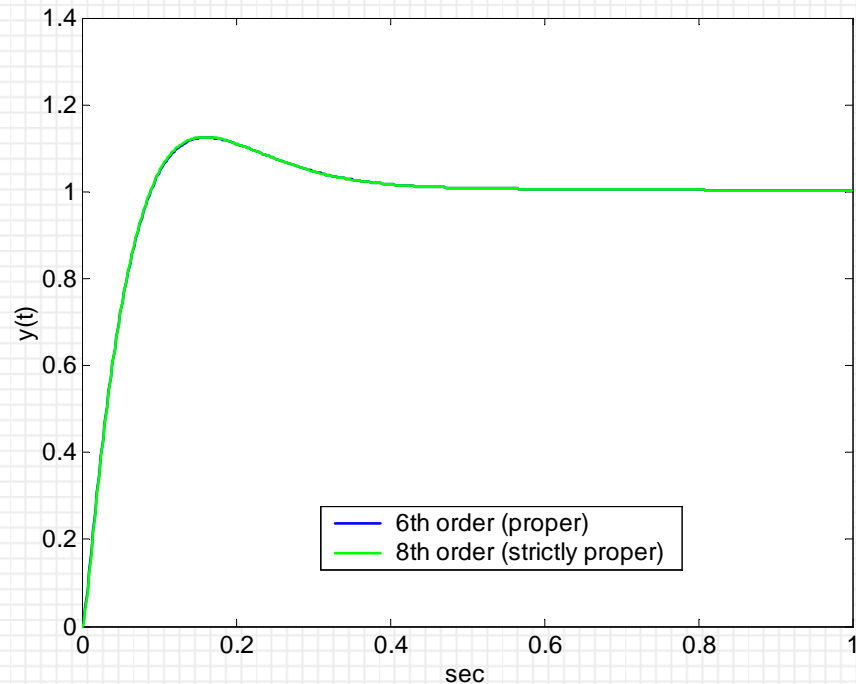
Now compare both controllers (ch7_C6.shp and ch7_C8.shp).



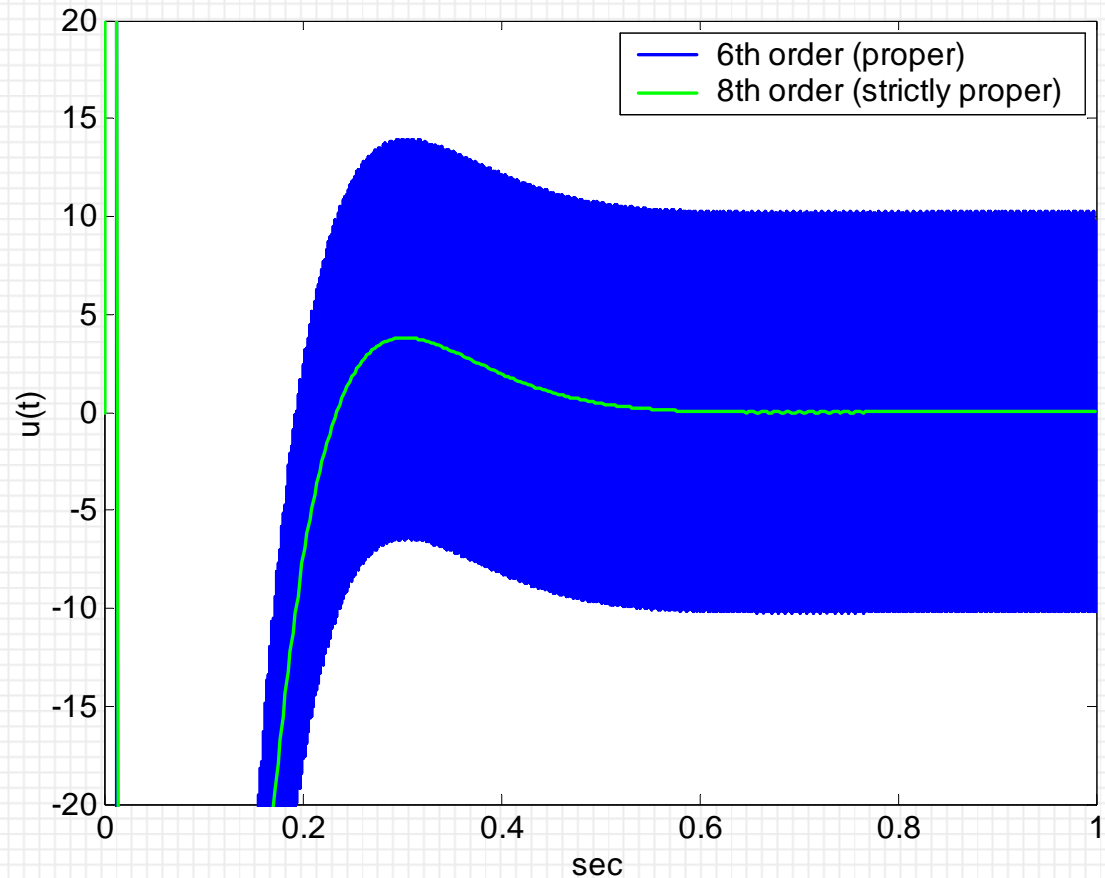
Consider the step response of this system with a 2 mV sinusoidal noise at 10000 rad/sec (ch7_noise.mdl).



As expected, both designs have similar bandwidths which indicate similar transients. Indeed, this is seen from the indistinguishable step responses.



However, the 2 mV high-frequency noise is very noticeable at the plant input when the proper design is used. Most often, this is not a good design - we should always consider high-frequency sensor noise and un-modeled dynamics (e.g., resonances). And saturation at the plant input can further exasperate the situation.



If this was a motor driving a robot arm or an arm holding the lens in a DVD drive, we would see nothing at the plant output (robot arm's tip or lens), while the motor will be humming like crazy, possibly overheating and generating audible noise.

Note: with $|L| < 0$ dB, $T \rightarrow L$ and $S \rightarrow 1$, so there's little benefit from feedback and there's no need to maintain the high loop gain beyond the $\omega = 4$ range. This important observation is the topic of a homework problem in the next chapter.

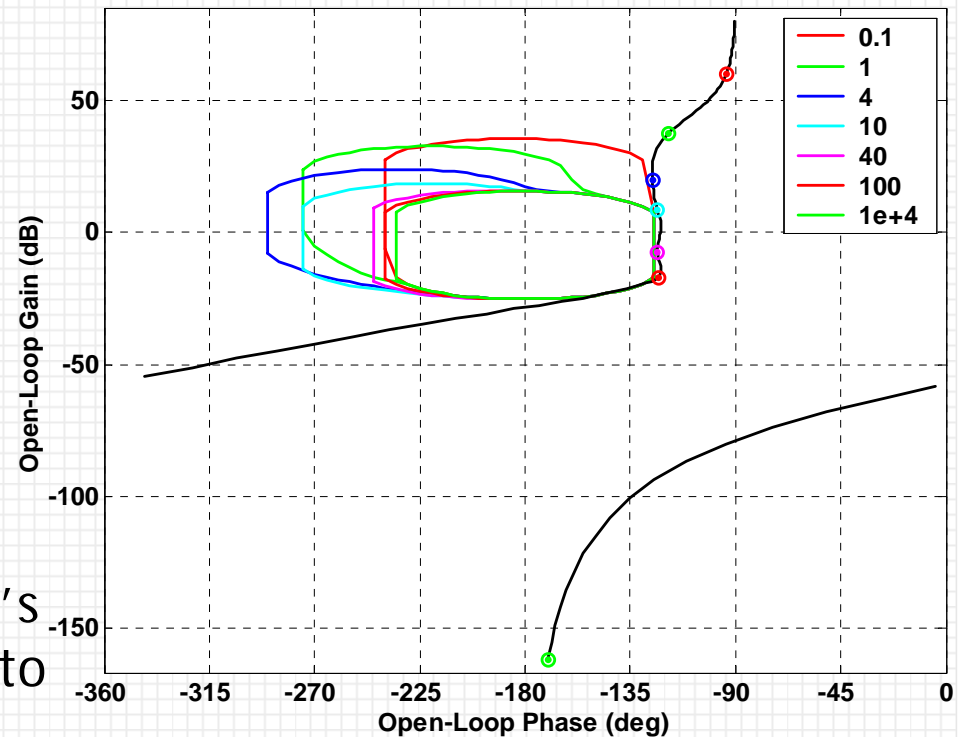
7.5. Reducing the High-Frequency Loop Gain

In QFT, the optimal controller is one that solves the following optimization problem

$$\begin{array}{ll} \min & k_c > 0 \\ \text{subject to} & L_0 \text{ stabilizes the system and} \\ & \text{satisfies its bounds} \end{array}$$

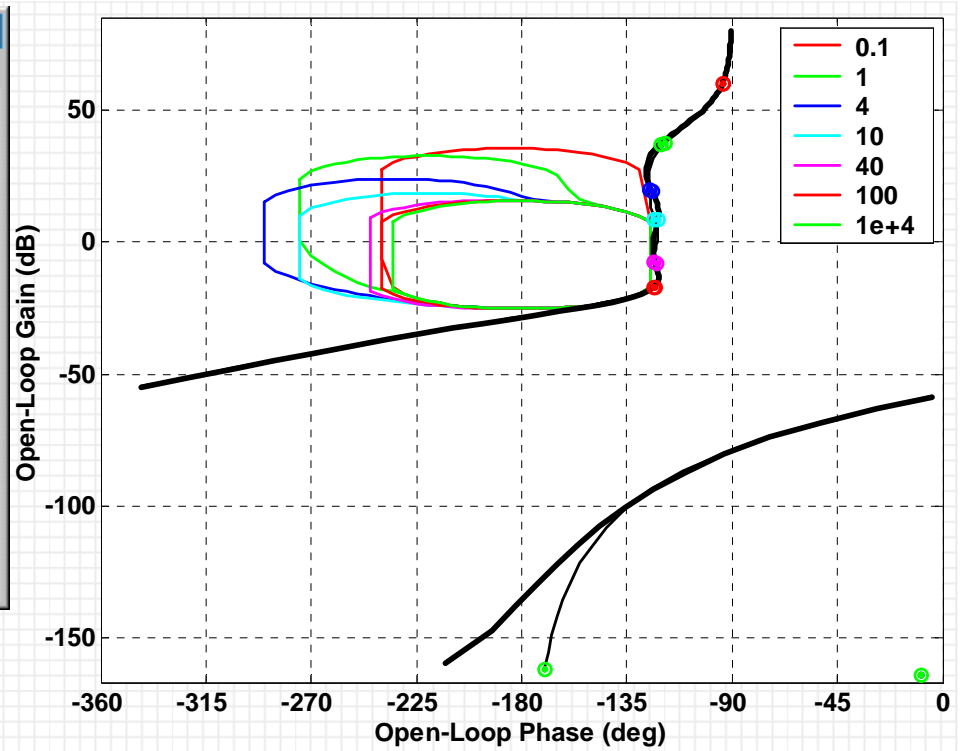
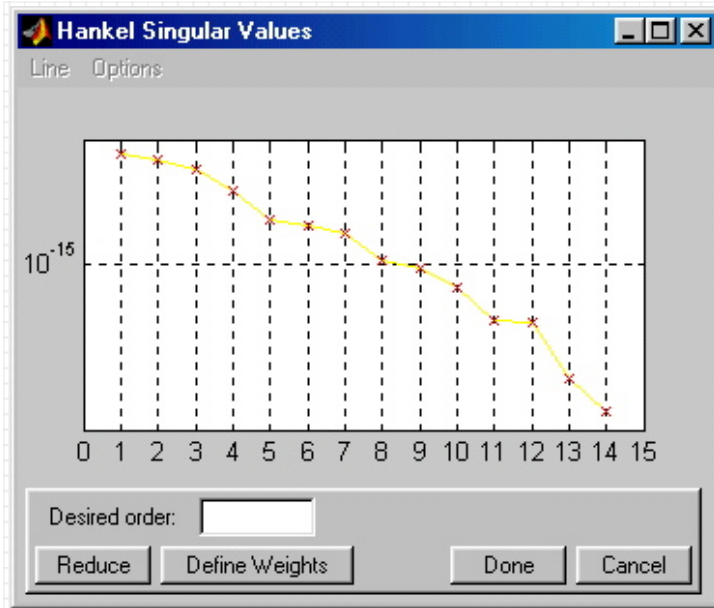
where the controller's high-frequency gain k_c is defined by

Roughly speaking, this optimal loop will lie right on its bounds at each frequency (we have infinite number of these). A practical design will attempt to lie on the few bounds we are working with and then “hug” the high-frequency margin bounds until the frequency where loop phase has no effect on stability. From that frequency and higher, we can drop the loop’s gain arbitrarily fast (subject to maintaining the “hugging” thing). This is done below by adding and tuning elements. The result is shown below (ch7_highorder.shp)

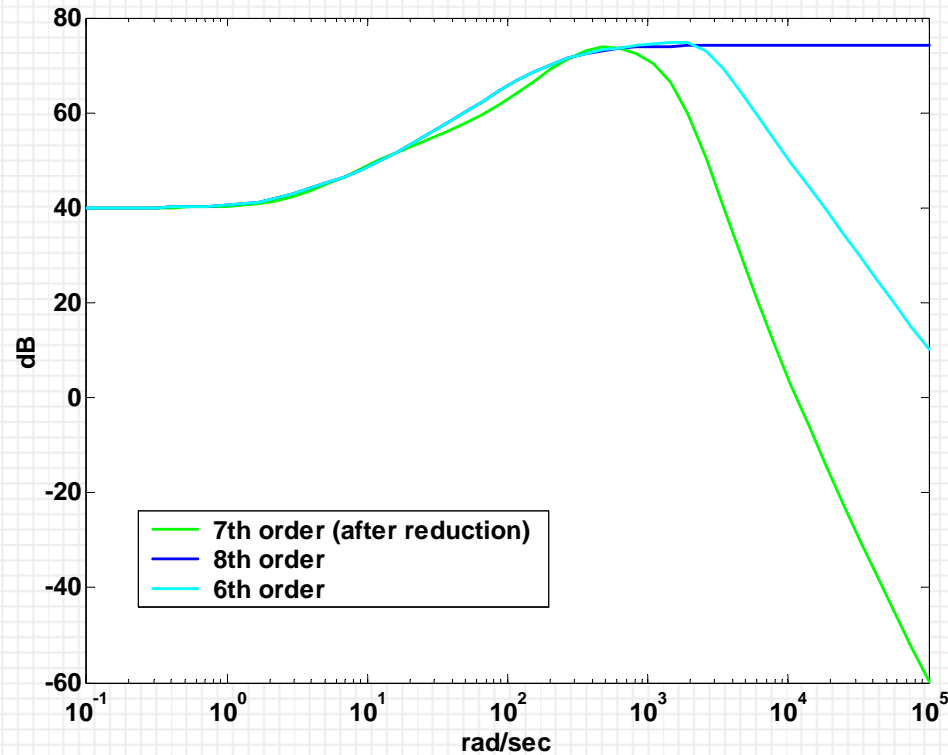


However, we now have a 16th order design. Fortunately, a very efficient order reduction algorithm is available within the loop shaping environment: highlight the controller elements for reduction (here all of them; in general, you can reduce only stable and proper controller). Then click on the Reduction button. It will open a new window titled *Hankel Singular Values*. The plot displays the relative contribution of the poles with respect to the the response (note: near perfect pole/zero cancellations are automatically detected and removed).

You can quickly test the feasibility of lower orders by entering a number, say 7, then hit Reduce. The program will compute the resulting 7th order controller and plot the reduced-order loop response (indicated by a dashed line) on top of the full-order version. In this case, there's a very good agreement in term of the frequency response where it counts, that is, relative to the bounds. If this ok, hit Done (see `ch7_highorder_red.shp`). For more information and available options on order reduction read the manual.



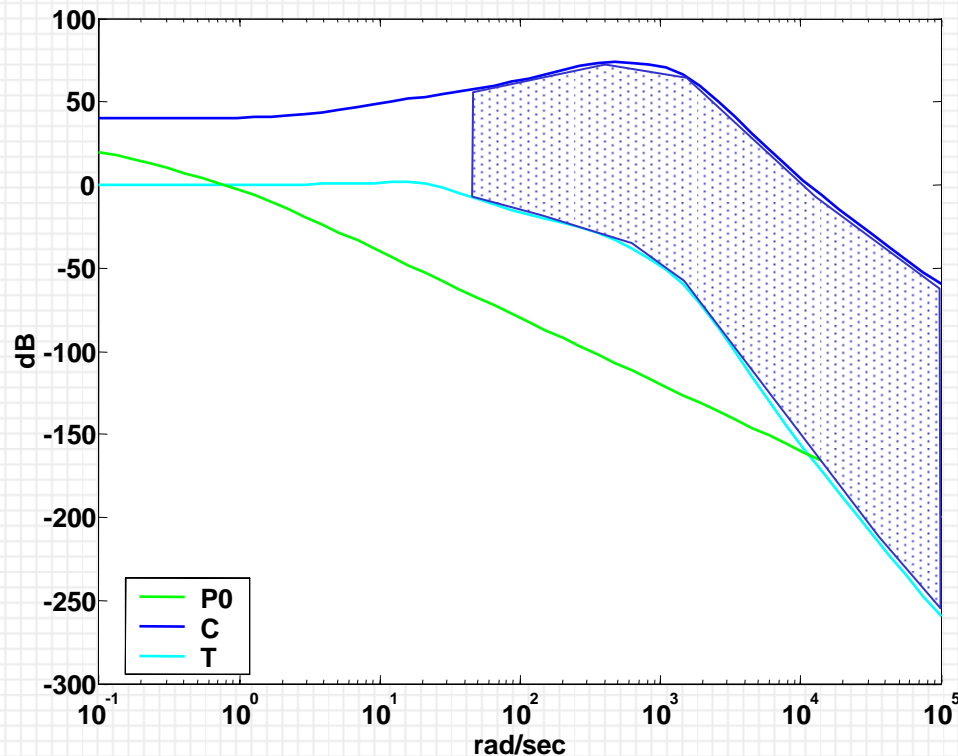
A comparison of the three controllers is shown below.



I.H.: "This is almost always the situation in a competent feedback design: tradeoff between bandwidth and complexity."

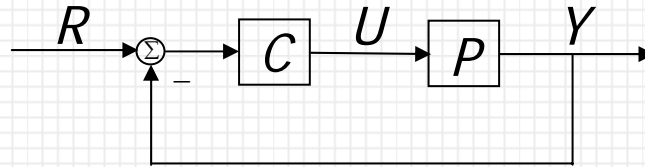
7.6. Cost of Feedback

Horowitz defines the *cost of feedback* as the excessive loop gain beyond the useful system bandwidth. This is nicely illustrated below using our nominal plant and “best” controller.



7.7. Homework

Consider the following feedback system



$$\mathcal{P} = \left\{ P(s) = \frac{k}{(s+a)(s+b)} : k \in [1,7], a \in [.2,3], b \in [9,44] \right\}.$$

Design a controller C such that it achieves:

- Robust stability margins of $PM = 50^\circ$ and $GM = 2.2$.
- $C(0) = 250$.
- Zero steady-state errors for step references R .
- “Minimal” bandwidth design.