8. MIMO DESIGN

This chapter covers the 2-input 2-output MIMO design for the basically non-interacting servo case, i.e. it is required that the closed loop design be such that reference signal number 1, $r_1(t)$, commands output number 1, $y_1(t)$, with minimal cross-coupling on output signal number 2, $y_2(t)$, and correspondingly for $r_2(t)$. A block diagram for a 2x2 MIMO servo system in closed loop is shown in Figure 8.1.

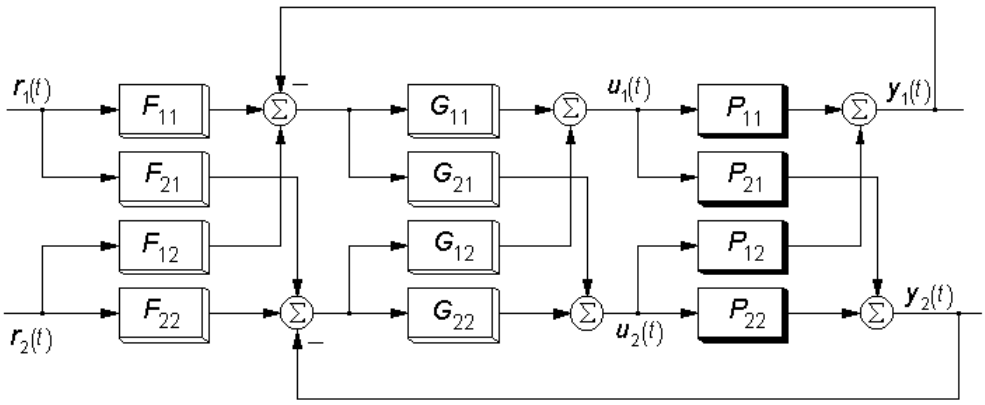


Figure 8.1: Block diagram of 2-input, 2-output MIMO servo system in closed loop.

The QFT design method in Qsyn for the basically non-interacting 2x2 MIMO servo system will be briefly presented, and a detailed example is given. The method is based on Horowitz (1992). The user should note that there exist other QFT approaches to this problem. Other MIMO control problems, e.g. the one degree-of-freedom disturbance rejection problem, requires some different analysis and formulation, see e.g. Horowitz (1992), Horowitz and Oldak (1992), Yaniv (1992), Yaniv (1995), and references given therein. Based on the present Qsyn functions, the user may write her own functions to implement other design approaches.

8.1 Design procedure

8.1.1 Equivalent SISO systems

Referring to Figure 8.1, let the transfer function matrices of the prefilter $F(s)$, the feedback controller $G(s)$, and the uncertain plant $P(s)$ be, with the Laplace arguments s suppressed,

$$F = \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix}, \quad G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}, \quad P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}. \tag{8.1}$$

The Laplace transforms of the reference signal vector $r(t)=\left(r_1(t)\ r_2(t)\right)^T$, and the output vector $y(t)=\left(y_1(t)\ y_2(t)\right)^T$, are denoted, respectively,

$$R=\begin{pmatrix} r_1 \\ r_2 \end{pmatrix},\quad Y=\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}. \tag{8.2}$$

The closed loop transfer function matrix from $R(s)$ to $Y(s)$ is called $T(s)$, i.e. with the Laplace arguments *s* suppressed and *I* denoting the unity matrix,

$$Y=TR \tag{8.3}$$

where

$$T=\begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}=\left(I+PG\right)^{-1}PGF\ . \tag{8.4}$$

Our design method allows for a full matrix feedback compensator $G(s)$ which is given as

$$G=P_0^{-1}L_0 \tag{8.5}$$

where $P_0(s)$ is a user chosen, fixed transfer function matrix, called the *pre-compensator*,

$$P_0=\begin{pmatrix} P_{011} & P_{012} \\ P_{021} & P_{022} \end{pmatrix}, \tag{8.6}$$

and $L_0(s)$ is a diagonal, fixed transfer function matrix which is synthesized by the user during the design process,

$$L_0=\begin{pmatrix} L_{10} & 0 \\ 0 & L_{20} \end{pmatrix}. \tag{8.7}$$

The representation (8.5) of the feedback compesator requires that $P_0(s)$ be invertible for almost all *s*, and is thus not completely general. This draw-back seems to be of minor importance in practice, since one would generally not design a rank-deficient $G(s)$.

The choice of $P_0(s)$ must be done *a priori* before the start of the feedback design process, such that the uncertain plant "seen" by the "diagonal feedback compensator to be designed, $L_0(s)$" is

$$V(s)=P(s)P_0^{-1}(s). \tag{8.8}$$

Ideas how to select $P_0(s)$ can be found in Yaniv (1995). There do not exist, however, general rules how to do this. One could e.g. try to find a pre-compensator that decouples the plant as much as possible, i.e. minimizes cross-coupling between the loops, at least for a critical frequency range. Or one could try to "shift uncertainty" between the loops such that the design task becomes roughly equally difficult with respect to the given specifications. Another idea is to pick the pre-compensator such that the effect of sensor noise is lessened, or that limited actuator power or amplitude is taken into account.

In the example below, $P_0(s)$ is chosen as one "typical" plant case, implying that the apparent plant $V(s)=P(s)P_0^{-1}(s)$ equals *I* for one plant case. This might, or might not, facilitate the

design, and the user is encouraged to redo the design with another choice of $P_0(s)$. Moreover, the arbitrary plant nominal $P_{\text{nom}}(s)$ is, indeed arbitrarily, chosen equal to $P_0(s)$. Such a choice is neither necesarry nor recommended.

If $P_0(s)=I$ is chosen, then one gets a diagonal feedback compensator, and a *decentralized* closed loop control system. Often a diagonal feedback compensator achieves the design objectives, and such a controller is often more practical to implement if the the sensor and actuator of each loop are co-located, while the sensor-actuator pairs of the two loops are distant from each other.

Considering (8.8), define

$$W = \begin{pmatrix} W_{11} & W_{12} \\ w_{21} & W_{22} \end{pmatrix} \stackrel{def}{=} V^{-1} = P_0 P^{-1}, \tag{8.9}$$

and let W be decomposed into one diagonal and one off-diagonal term,

$$W = W_d + W_b, \qquad W_d = \begin{pmatrix} W_{11} & 0 \\ 0 & W_{22} \end{pmatrix}, \qquad W_b = \begin{pmatrix} 0 & W_{12} \\ w_{21} & 0 \end{pmatrix}. \tag{8.10}$$

Note that W is an uncertain transfer function matrix, derived from the uncertain plant and the fixed pre-compensator. Then, from (8.4)-(8.10),

$$T = \left(I + W_d^{-1} L_0\right)^{-1} W_d^{-1} \left(L_0 F - W_b T\right) \tag{8.11}$$

or, equivalently,

$$T_{11} = \frac{\frac{L_{10}}{W_{11}} F_{11} - \frac{W_{12}}{W_{11}} T_{21}}{1 + \frac{L_{10}}{W_{11}}} \tag{8.12}$$

$$T_{12} = \frac{\frac{L_{10}}{W_{11}} F_{12} - \frac{W_{12}}{W_{11}} T_{22}}{1 + \frac{L_{10}}{W_{11}}} \tag{8.13}$$

$$T_{21} = \frac{\frac{L_{20}}{W_{22}} F_{21} - \frac{W_{21}}{W_{22}} T_{11}}{1 + \frac{L_{20}}{W_{22}}} \tag{8.14}$$

$$T_{22} = \frac{\frac{L_{20}}{W_{22}} F_{22} - \frac{W_{21}}{W_{22}} T_{12}}{1 + \frac{L_{20}}{W_{22}}} \tag{8.15}$$

The scalar transfer transfer functions, $T_{ij}(s)$, i=1,2, j=1,2, in (8.12)-(8.15) can each be represented by the SISO block diagram in Figure 8.2, assuming that its reference signal is an impulse (whose Laplace transform equals 1) and that its plant input disturbance has a Laplace transform equalling $-W_{ik}T_{kj}$, k=1,2, $k \neq i$.

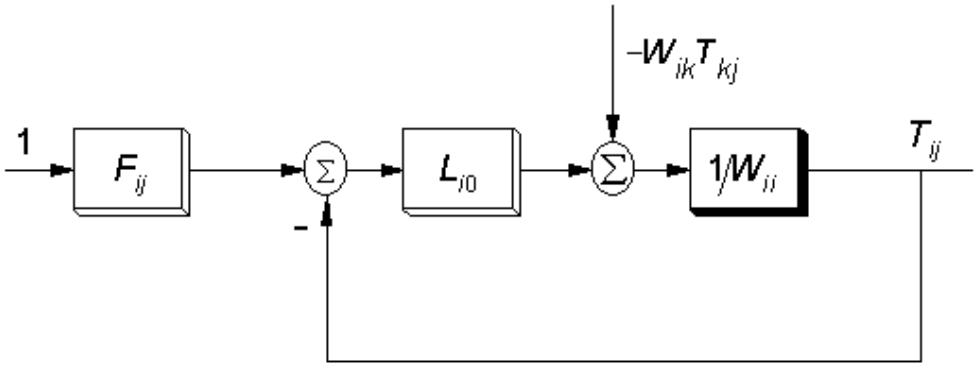


Figure 8.2. Block diagram representation of (8.12)-(8.15), whereby for (8.12), i=1, j=1, and k=2; for (8.13), i=1, j=2, and k=2; for (8.14), i=2, j=1, and k=1; and for (8.15), i=2, j=2, and k=1.

Clearly, in (8.12) and (8.13), the "uncertain plant" is $1/W_{11}$ (which is a function of $P(s)$ and $P_0(s)$ as mentioned above), and the feedback compensator to be designed is $L_{10}(s)$ in a standard SISO problem. Likewise, in (8.14) and (8.15), the "uncertain plant" is $1/W_{22}$, and the feedback compensator is $L_{20}(s)$ in another standard SISO problem. The only difficulty seems to be that the plant input disturbance $-W_{ik}T_{kj}$ depends on not-yet-designed elements of the closed loop transfer function.

Horowitz suggested a clever way to circumvent the above problem by assuming that T_{kj} would satisfy its maximal gain specification when the design would be completed, i.e.

$$\left|T_{kj}\right| \le b_{kj}(\omega), \tag{8.16}$$

where $b_{kj}(\omega)$ is the upper closed loop gain specification as a function of the frequency ω. Then the plant input disturbance $-W_{ik}T_{kj}$ is replaced by a worst case disturbance

$$D_{ij} = \max\left|W_{ik}\right| b_{kj} e^{j\varphi} \tag{8.17}$$

where the phase $\varphi(\omega)$ is implicitly chosen in a worst case sense in each of (8.12)-(8.15) as to attempt to violate as much as possible the specifications for $\left|T_{ij}\right|$, as shown in Section 8.1.4 below. (In (8.17) the argument j in the exponent obviously denotes $\sqrt{-1}$.) With these assumptions we have achieved two standard SISO feedback design problems for $L_{10}(s)$ and $L_{20}(s)$. How the prefilter design is simplified is shown in Section 8.1.3.

8.1.2 Specifications

Basically non-interacting specifications are of the following type:

$$a_{ij}(\omega) \le \left|T_{ij}(j\omega)\right| \le b_{ij}(\omega), \qquad i = j \tag{8.18}$$

and

$$\left|T_{ij}(j\omega)\right| \le b_{ij}(\omega), \qquad i \ne j \tag{8.19}$$

where (8.18) is a usual servo specification in the frequency domain (1.8) pertaining to the transmission between a reference signal and its assigned plant output, while (8.19) is a *cross coupling specification* in the frequency domain with $b_{ij}(\omega)$ in general chosen small. We will demonstrate in the example that for different Horowitz bound computations, it is convenient to let $b_{ij}(\omega)$ have different high frequency roll-offs, which is a design aid that will not influence actual system performance.

8.1.3 Diagonal prefilter

The design of the prefilter F is in principle an open loop problem, after the complementary sensitivity function

$$\overline{S} = (I + PG)^{-1}PG \tag{8.20}$$

has been designed, see (8.4). For the servo specifications (8.18), it is mandatory that F_{11} and F_{22} are essentially non-zero and in general approximatively one for low frequencies, just like the in a standard servo design in the SISO case, see e.g. (2.5). In general one would design the off-diagonal elements of $\overline{S}$ small so that the cross-coupling transmission satisfies (8.19). The question naturally arises if the off-diagonal elements F_{12} and F_{21} could be designed such that the cross-coupling is reduced, while the specified servo properties are retained.

The answer seems to depend on how much uncertainty remains in $\overline{S}$ after the feedback design of G, since for a certain, full-rank $\overline{S}$, F is simply chosen as $\overline{S}^{-1}$ times a high bandwidth low-pass filter, to ensure that F becomes (strictly) proper. But in a robust QFT design, G is in general designed such that at least one of the specifications is just satisfied. One might surmise that often off-diagonal elements in F would yield no significant benefit.

Therefore we *postulate* that for the basically non-interactive servo problem, we *a priori* select F to be diagonal, i.e.

$$F_{12} = F_{21} = 0. \tag{8.21}$$

This said, we urge the user to carefully study her closed loop solution. If she finds that the servo specifications (8.18) are well satisfied, while one or both cross coupling specifications (8.19) are only marginally satisfied, then she may test if beneficial non-zero F_{12} and F_{21} could be found. If this is the case, she may further contemplate to relax the cross coupling specification(s) in a second feedback design attempt (then the gain and bandwidth of G would be lowerd), and then, after having found $\overline{S}$, try to find a pre-filter with non-zero F_{12} and F_{21} that gets the cross coupling back within specifications.

8.1.4 The first design step

It would be possible to solve simultaneously (8.18), (8.19), (8.21) with (8.12), (8.13), and with (8.14), (8.15) to find $L_{10}(s)$, $F_{11}(s)$ and $L_{20}(s)$, $F_{22}(s)$, respectively. It is however smarter to select *one* of the problems, (8.12), (8.13) *or* (8.14), (8.15), in a first design step, since then, in the second design step, no worst case approximation (8.17) for the cross coupling (plant input disturbance in Figure 8.2) will be needed.

One would typically choose to solve the easiest of the two problems first, since then the more difficult second step problem would have become easier after the worst case cross coupling approximation (8.17) is eliminated.

Assume that the design for (8.18), (8.19), (8.21), (8.12), (8.13) is chosen in the first step. Then we have

$$a_{11}(\omega) \le \left| \frac{\frac{L_{10}(j\omega)}{W_{11}(j\omega)} F_{11}(j\omega) - \frac{W_{12}(j\omega)}{W_{11}(j\omega)} T_{21}(j\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right| \le b_{11}(\omega) \tag{8.22}$$

$$\left| \frac{\frac{W_{12}(j\omega)}{W_{11}(j\omega)} T_{22}(j\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right| \le b_{12}(\omega) \tag{8.23}$$

The worst case cross coupling effect (8.17) in (8.22), with the approximation (8.16) is found by using the triangle and the inverse triangle inequalitites,

$$\left| \frac{\frac{L_{10}(j\omega)}{W_{11}(j\omega)} F_{11}(j\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right| + \left| \frac{\frac{W_{12}(j\omega)}{W_{11}(j\omega)} b_{21}(\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right| \le b_{11}(\omega) \tag{8.24}$$

$$a_{11}(\omega) \le \left\| \frac{\frac{L_{10}(j\omega)}{W_{11}(j\omega)} F_{11}(j\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right| - \left| \frac{\frac{W_{12}(j\omega)}{W_{11}(j\omega)} b_{21}(\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right\| \tag{8.25}$$

To find Horowitz bounds for the nominal open loop $L_{10}/W_{11\text{nom}}$ (see Figure 8.2) from (8.24) and (8.25) is not easy, since they depend on the not-yet-designed F_{11}. With the reasonable assumption (at least for low frequencies) that the left term of the right member of (8.25) has larger modulus than the right term of the right member of (8.25), (8.24) and (8.25) may be reformulated in the following way *without* further approximations,

$$a_{11}(\omega) + b_{d11}(\omega,\cdot) \le \left| \frac{\frac{L_{10}(j\omega)}{W_{11}(j\omega)} F_{11}(j\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right| \le b_{11}(\omega) - b_{d11}(\omega,\cdot) \tag{8.26}$$

$$\left| \frac{\frac{W_{12}(j\omega)}{W_{11}(j\omega)} b_{21}(\omega)}{1 + \frac{L_{10}(j\omega)}{W_{11}(j\omega)}} \right| \le b_{d11}(\omega,\cdot) \tag{8.27}$$

where $b_{d11}(\omega,\cdot)$ satisfying $0 \le b_{d11}(\omega,\cdot) \le \big(b_{11}(\omega) - a_{11}(\omega)\big)/2$ is chosen such that the tolerance bound (1.10) for $L_{10}(j\omega)/W_{11\text{nom}}(j\omega)$ emanating from (8.26) equals the disturbance rejection bound from (8.27). In such a way, there will be no over-design with respect to any of (8.26) or (8.27). The second argument in $b_{d11}(\omega,\cdot)$ reflects that $b_{d11}(\omega,\cdot)$ is chosen in this way.

Equation (8.23) with the worst case approximation (8.16) gives the specification

$$\left|\frac{\frac{W_{12}(j\omega)}{W_{11}(j\omega)}b_{22}(\omega)}{1+\frac{L_{10}(j\omega)}{W_{11}(j\omega)}}\right| \le b_{12}(\omega) \qquad (8.28)$$

from which a third set of bounds for $L_{10}(j\omega)/W_{11nom}(j\omega)$ are easily found.

Notice that to get a high frequency roll-off for the bounds emanating from (8.27), it is in general required that $b_{21}(\omega)$ rolls off faster than $b_{d11}(\omega,\cdot)$, which is achieved if $b_{21}(\omega)$ rolls off faster than $b_{11}(\omega)$. On the other hand, to get a high frequency roll-off for the bounds emanating from (8.28), it is in general required that $b_{22}(\omega)$ rolls off faster than $b_{12}(\omega)$. If one has symmetrical specifications (8.18), (8.19), i.e. $b_{12}(\omega)=b_{21}(\omega)$, and $b_{11}(\omega)= b_{22}(\omega)$, then this roll-off condition is impossible to achieve. Computationally the conflict may be solved either by ignoring the high frequency bounds from (8.27) and (8.28) , or better, by artificially giving $b_{12}(\omega)$ and $b_{21}(\omega)$ the required roll-off when computing bounds from (8.27) and (8.28). If $b_{11}(\omega)= b_{22}(\omega)$ roll off with -40 dB/dec, then let $b_{21}(\omega)$ roll off with e.g. -80 dB/dec, and let $b_{12}(\omega)$ have no high frequency roll off (0 dB/dec). This "trick" to get reasonable bounds will have no influence on the closed loop system behaviour, since $L_{10}(j\omega)/W_{11nom}(j\omega)$ has to be designed with a reasonable high frequency roll-off, whether the bounds impose it or not.

With the three sets of Horowitz bounds, from (8.26), (8.27), and (8.28), where the bounds from (8.26) and (8.27) should be equal, $L_{10}(s)$ is synthesized by loop shaping, see Sections 1.6 and 2.5.

Notice that after $L_{10}(j\omega)$ has been designed, $L_{10}(j\omega)=L_{10}^*(j\omega)$, (8.27) gives the *designed* b_{d11}*-value*, i.e.

$$b_{d11}^*(\omega)=\max\left|\frac{\frac{W_{12}(j\omega)}{W_{11}(j\omega)}b_{21}(\omega)}{1+\frac{L_{10}^*(j\omega)}{W_{11}(j\omega)}}\right| \qquad (8.29)$$

where max$|\ |$ is taken over all plant cases. When designing F_{11}, one must then use the specification (8.26) with the designed b_{d11}-value inserted,

$$a_{11}(\omega)+b_{d11}^*(\omega)\le\left|\frac{\frac{L_{10}^*(j\omega)}{W_{11}(j\omega)}F_{11}(j\omega)}{1+\frac{L_{10}^*(j\omega)}{W_{11}(j\omega)}}\right| \le b_{11}(\omega)-b_{d11}^*(\omega). \qquad (8.30)$$

Note that the notation $L_{10}(j\omega)=L_{10}^*(j\omega)$ for the chosen feedback design is used, for clarity, only in (8.29), (8.30).

The first design step is concluded by synthesizing $F_{11}(s)$ to satisfy (8.30), exactly like in standard SISO design (Section 1 and 2).

8.1.5 The second design step

With $L_{10}(s)$ and $F_{11}(s)$ given from the first design step in the preceding section, (8.14) and (8.15) become, respectively,

$$T_{21} = \frac{\dfrac{L_{20}}{W_{22}^e}F_{21} - \dfrac{W_{21}}{W_{22}^e}\cdot\dfrac{L_{10}}{1+(L_{10}/W_{11})}F_{11}}{1+\dfrac{L_{20}}{W_{22}^e}} \tag{8.31}$$

and

$$T_{22} = \frac{\dfrac{L_{20}}{W_{22}^e}F_{22} - \dfrac{W_{21}}{W_{22}^e}\cdot\dfrac{L_{10}}{1+(L_{10}/W_{11})}F_{12}}{1+\dfrac{L_{20}}{W_{22}^e}} \tag{8.32}$$

where

$$W_{22}^e = W_{22}\left(1-\frac{\gamma}{1+(L_{10}/W_{11})}\right) \tag{8.33}$$

and

$$\gamma = \frac{W_{12}W_{21}}{W_{11}W_{22}}\ . \tag{8.34}$$

With F_{12} = F_{21} = 0 from (8.21), (8.31) and (8.32) together with the specifications (8.18), (8.19) yield the specification equations

$$\left|\frac{\dfrac{W_{21}(j\omega)}{W_{22}^e(j\omega)}\cdot\dfrac{L_{10}(j\omega)}{1+(L_{10}(j\omega)/W_{11}(j\omega))}\cdot F_{11}(j\omega)}{1+\dfrac{L_{20}(j\omega)}{W_{22}^e(j\omega)}}\right| \le b_{21}(\omega) \tag{8.35}$$

and

$$a_{22}(\omega) \le \left|\frac{\dfrac{L_{20}(j\omega)}{W_{22}^e(j\omega)}}{1+\dfrac{L_{20}(j\omega)}{W_{22}^e(j\omega)}}F_{22}(j\omega)\right| \le b_{22}(\omega)\ . \tag{8.36}$$

Clearly, (8.35) and (8.36) represent a standard SISO design problem for designing $L_{20}(s)$ and $F_{22}(s)$, with (8.35) giving disturbance rejection bounds, and (8.36) tolerance bound (see Section 1.5), respectively, for $L_{20}(j\omega)/W_{22\text{nom}}^e(j\omega)$.

With $L_{10}(s)$ and $L_{20}(s)$ given, the feedback compensator $G(s)$ is computed from (8.5).

8.1.6 The third optional design step

The design is actually concluded with the second design step. Since, inevitable, the first design step included a certain over-design due to the worst case approximations (8.17), (8.24), (8.25) of the then unknown closed loop transfer function elements, it could however be clever to re-design $L_{10}(s)$ and $F_{11}(s)$ with $L_{20}(s)$ and $F_{22}(s)$ now known.

The equations for this re-design are exactly those of Section 8.1.5 with the indices 1 and 2 exchanged for one another.

8.1.7 Closed loop stability

The last design step is a regular SISO design problem, and closed loop asymptotic stability should be ascertained by e.g. the Nyquist theorom for each plant case, as in Section 1.7. Then the closed loop MIMO system will be asymptotically stable, provided that it contains no unstable and uncontrollable or unstable and unobservable subsystems, that it is fully connected (i.e. all inputs influence all outputs), and that there are no internal unstable pole-zero cancellations, according to a theorem due to Bode (Horowitz, 1992).

For uncertain plants, it is uncommon that generic pole-zero cancellations occur. In general a sensitivity specification (1.5) is included in the last design step.

8.1.8 Simulations of the closed loop

As always, the closed loop should be simulated in the frequency and time domains for sufficiently many plant cases, see Section 1.9.

8.2 A 2x2 MIMO servo design example

8.2.1 Plant definition and templates

Let the plant be given by

$$P = \begin{pmatrix} k_{11}/s & k_{12}/s \\ k_{21}/s & k_{22}/s \end{pmatrix}, \quad k_{11} \in [2,6],\ k_{12} \in [0.5,15],\ k_{21} \in [0.5,15],\ k_{22} \in [2,6] \tag{8.37}$$

with all uncertainties mutually independent, and with the nominal being

$$P_{\text{nom}} = \begin{pmatrix} 2/s & 1/s \\ 1/s & 2/s \end{pmatrix} \tag{8.38}$$

Each plant element is defined in its own plant definition file, called `p11.m`, `p12.m`, `p21.m`, and `p22.m`, for $P_{11}(s)$, $P_{12}(s)$, $P_{21}(s)$, and $P_{22}(s)$, respectively. See Figures 8.3 - 8.6.

Then compute the four template files, `p11.tpl`, `p12.tpl`, `p21.tpl`, and `p22.tpl`, for $P_{11}(s)$, $P_{12}(s)$, $P_{21}(s)$, and $P_{22}(s)$, respectively.

```
ctpl('p11',[],'grid');
ctpl('p12',[],'grid');
ctpl('p21',[],'grid');
ctpl('p22',[],'grid');
```

The grid method is used, and we will see that this method leaves "holes" in the templates. The user is therefore encouraged to recompute this example with another method.

```
function   [Par,w tpl,w nom,method,P num,P den, ...
            n dif,Uns Par] = p11

% MIMO Plant:  p11.m

% Definition of the parameters
% ====================
   Par = ['k11=[2,6,2,4]'];
   Uns_Par=[];

% Definition of the frequency vectors [rad/sec]
% =====================================
  w tpl = [0.1 0.2 0.5 1 2 5 10 20 50 70 100];
  w nom = logspace(-2,3,200);

% Definition of the template computation method
% ============================================
  method = 'rff_[1,1]';

% Plant definition
% ================
  P num='(gain,k11)';
  P_den='[1  0]';
```

Figure 8.3. The plant definition file `p11.m` *for the plant element* $P_{11}(s)$.

```
function   [Par,w tpl,w nom,method,P num,P den, ...
            n dif,Uns Par] = p12

% MIMO Plant:  p12.m

% Definition of the parameters
% ====================
   Par = ['k12=[0.5,1.5,1,4]'];
   Uns Par=[];

% Definition of the frequency vectors [rad/sec]
% =====================================
  w_tpl = [0.1 0.2 0.5 1 2 5 10 20 50 70 100];
  w_nom = logspace(-2,3,200);

% Definition of the template computation method
% ============================================
  method = 'rff [1,1]';

% Plant definition
% ================
  P_num='(gain,k12)';
  P_den='[1  0]';
```

Figure 8.4. The plant definition file `p12.m` *for the plant element* $P_{12}(s)$.

```
function   [Par,w_tpl,w_nom,method,P_num,P_den, ...
            n dif,Uns Par] = p21

% MIMO Plant:  p21.m

% Definition of the parameters
% ====================
   Par = ['k21=[0.5,1.5,1,4]'];
   Uns_Par=[];

% Definition of the frequency vectors [rad/sec]
% =====================================
  w tpl = [0.1 0.2 0.5 1 2 5 10 20 50 70 100];
  w nom = logspace(-2,3,200);

% Definition of the template computation method
% ============================================
  method = 'rff_[1,1]';

% Plant definition
% ================
  P_num='(gain,k21)';
  P_den='[1  0]';
```

Figure 8.5. The plant definition file `p21.m` *for the plant element* $P_{21}(s)$.

```
function   [Par,w tpl,w nom,method,P num,P den, ...
            n dif,Uns Par] = p22

% MIMO Plant:  p22.m

% Definition of the parameters
% ====================
   Par = ['k22=[2,6,2,4]'];
   Uns_Par=[];

% Definition of the frequency vectors [rad/sec]
% =====================================
  w tpl = [0.1 0.2 0.5 1 2 5 10 20 50 70 100];
  w nom = logspace(-2,3,200);

% Definition of the template computation method
% ============================================
  method = 'rff_[1,1]';

% Plant definition
% ================
  P num='(gain,k22)';
  P_den='[1  0]';
```

Figure 8.6. The plant definition file `p22.m` *for the plant element* $P_{22}(s)$.

The templates of $P_{11}(s)$ in `p11.tpl` are displayed with the command

```
showtpl('p11');
```

shown in Figure 8.7. Quite unsurprisingly, the templates are shown lying along straight line segments with gain but no phase extent, hidden behind the nominal. The other plant element templates look similar.

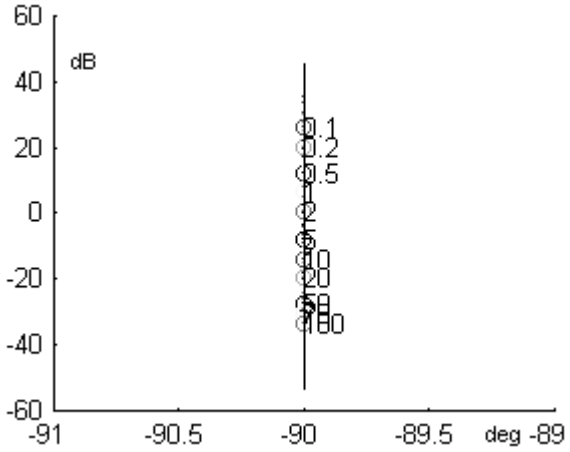


Figure 8.7. The templates of $P_{11}(s)$ *in* `p11.tpl` *displayed with the command* `showtpl('p11');`

8.2.2 The pre-compensator

Next the pre-compensator $P_0(s)$ (8.6) is defined as a fixed transfer function in the controller function m-file `p0.m`, see Figure 8.8. In this example, $P_0(s)$ was set equal to $P_{\text{nom}}(s)$ in (8.38). The user is encouraged to try other choices of $P_0(s)$, as discussed in Section 8.1.1.

```
function [P0] = p0(s)
% p0.m  precompensator transfer function for 2x2 MIMO

P0 = [2./s 1./s
      1./s 2./s];
```

Figure 8.8. Controller function file `p0.m` *defining the pre-compensator* $P_0(s)$.

8.2.3 Specifications

We choose basically non-interactive specifications, (8.18), (8.19), that are symmetric, implying furthermore that

$$\begin{cases} a_{11}(\omega) = a_{22}(\omega) \\ b_{11}(\omega) = b_{22}(\omega) \\ b_{12}(\omega) = b_{21}(\omega) \end{cases} . \qquad (8.39)$$

The servo specification (8.18), (8.39) is defined as usual with the `rsrs`-command, and displayed with `showspc` in Figure 8.9, where the time domain choices can be clearly seen, as well as the resulting frequency domain specification. A cut-off frequency where $a_{11}(\omega)$ fell off sharply was not included, implying that the final design would be allowed to violate $a_{11}(\omega)$ for high frequencies. Notice however that both second and third order closed loop models were included in the `rsrs`-commands, implying that the tolerance $b_{11}(\omega)/a_{11}(\omega)$ increases for increasing high frequencies. The specifation matrix is saved in the file `ex8.spc`.

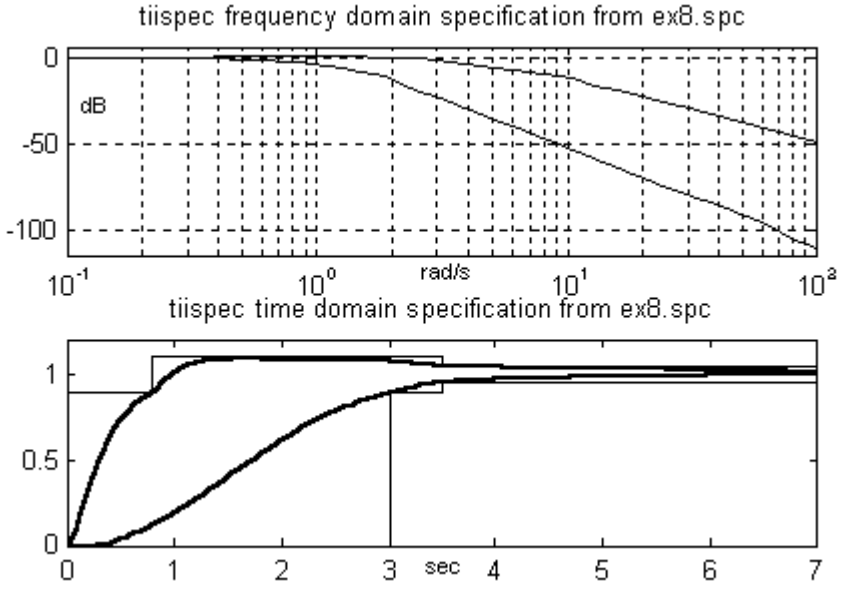


Figure 8.9. Servo specifications in the time domain and resulting frequency domain specification for the closed loop frequency function elements $|T_{11}(j\omega)|$ *and* $|T_{22}(j\omega)|$.

```
rsrs('ex8','tiispec',[3 0.8],10,3.5,[],logspace(-1,2),[],3);
showspc('ex8','tiispec');                                % Figure 8.9
```

In view of the discussion following equation (8.28) two specification vectors, $b_{12}(\omega) = b_{21}(\omega)$ in (8.19), (8.39) for the off-diagonal, cross coupling, closed loop transfer function elements $|T_{12}(j\omega)|$ and $|T_{21}(j\omega)|$ are defined and shown in Figure 8.10: one, `tijspeca`, with no high frequency roll-off, and another, `tijspec`, with -40 dB/dec high frequency roll-off.

```
add2spc('ex8','tijspeca',logspace(-1,2),-20);  % b12 with roll-off=0
add2spc('ex8','tijspec',logspace(-1,2),[.1],[.01 .14 1]);% -40 dB/dec
showspc('ex8','tijspec');showspc('ex8','tijspeca',[],[],gcf));
                                                         % Figure 8.10
```

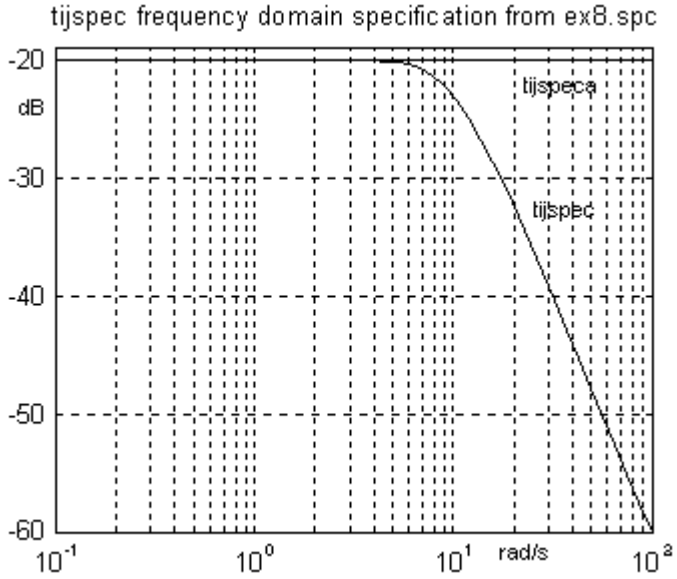


Figure 8.10: Cross coupling closed loop specifications.

Note that `tijspec` has the same roll-off as the upper servo specification $b_{11}(\omega) = b_{22}(\omega)$ in `tiispec`. It might have been a little more suitable to choose the `tijspec` roll-off to be -60 dB/dec or -80 dB/dec, in view of the ensuing bound computation. The user may try this out.

Finally a 6 dB sensitivity specification is included into `ex8.spc`:

```
add2spc('ex8','sens',logspace(-1,2),6);                    % sensitivity
```

Such a specification has no direct significance for the first design step, since there is no requirement that the first loop be independently stable. Sometimes it might even be necessary to design the first loop unstable, to avoid a non-minimum phase, or unstable non-minimum phase problem in the second design step, see (8.33), (8.34). However, if possible, one would design the first loop stable. Therefore the sensitivity specification is included as a design aid.

8.2.4 The first design step

Template computation

We create the template files for necessary for the bound computations of the first design step, (8.12)-(8.15) or (8.26)-(8.28), whether the first step will be a design for the "plant" $1/W_{11}(j\omega)$ with the "disturbance filter" $W_{12}(j\omega)/W_{11}(j\omega)$ in (8.27)-(8.28), or a design for $1/W_{22}(j\omega)$, with the filter $W_{21}(j\omega)/W_{22}(j\omega)$. In the latter case the user might want to exchange the indices 1 and 2, in all equations from (8.22) on. The four template files are created with the command `mtpl1`, where the following intuitive template file names are chosen: `iw11.tpl` for $1/W_{11}(j\omega)$, `iw22.tpl` for $1/W_{22}(j\omega)$, `w12_w11.tpl` for $W_{12}(j\omega)/W_{11}(j\omega)$, and `w21_w22.tpl` for $W_{21}(j\omega)/W_{22}(j\omega)$. The nominals of the output template files emanate from the plant element nominals.

```
mtpl1('iw11','iw22','w12_w11','w21_w22','p11','p12','p21','p22','p0');
showtpl('iw11'); showtpl('w12_w11'); % Figure 8.11
showtpl('w21_w22'); showtpl('iw22'); % Figure 8.11
```

Figure 8.11 reveals that the set of templates and nominals are equal for all frequencies, for each of the four template files. The templates of $1/W_{11}(j\omega)$ equal those of $1/W_{22}(j\omega)$, being a line segment along the real positivie axis. The templates of $W_{12}(j\omega)/W_{11}(j\omega)$ equal those of $W_{21}(j\omega)/W_{22}(j\omega)$, being a line segment on the real axis, crossing the origin. The grid method for the template computation caused the templates to be represented discontinuously, with "holes", the consequence of which will be Horowiths bound "islands" (hidden in Figure 8.12, but apperent in Figure 8.20) that are easily discernable and neglectable by the observant user. Since the templates for the two loops, (8.12), (8.13) versus (8.14), (8.15) are identical, we choose to start designing $L_{10}/W_{11\text{nom}}$.

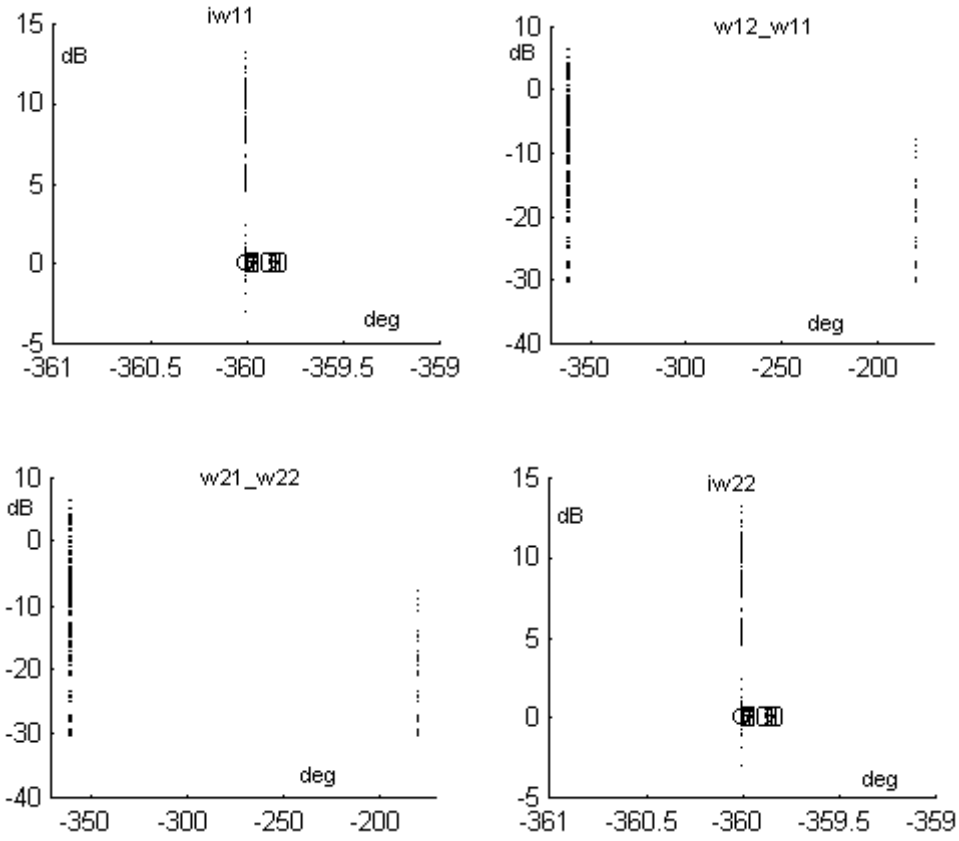


Figure 8.11. Templates for $1/W_{11}(j\omega)$ *from* `iw11.tpl`, $W_{12}(j\omega)/W_{11}(j\omega)$ *from* `w12_w11.tpl`, $W_{21}(j\omega)/W_{22}(j\omega)$ *from* `w21_w22.tpl`, *and for* $1/W_{22}(j\omega)$ *from* `iw22.tpl`. *The nominals of* $1/W_{11}(j\omega)$ *and* $1/W_{22}(j\omega)$ *have all the value* 1. *The nominals of* $W_{12}(j\omega)/W_{11}(j\omega)$ *and* $W_{21}(j\omega)/W_{22}(j\omega)$ *have all the value* 0, *which is represented in Nichols form by* -320 dB, *outside the scale of the figures.*

Bound computation

The common Horowitz bounds for $L_{10}/W_{11\text{nom}}$ from (8.26) and (8.27) are computed with the command `mcbnd11`, in which also is stored, for each frequency, a matrix giving $b_{d11}(\omega,\cdot)$ as a function from **C** [deg, dB] to **R**. The criterion function used is `fbd11.m` (default name) which computes the values of the middle member of (8.26) and the left member of (8.27). The bound file is called `ex8.bnd`, and by default, the name of the resulting bounds is `servo1`:

```
mcbnd11('ex8','tiispec','tijspec',[],[],'ex8','iw11',' w12_w11');
                                        % Default bound name: servo1
```

Notice the order of the two specification vector arguments, `tiispec` with 3 columns, and `tijspec` with 2 columns which signals that the common bounds from (8.26) and (8.27) are to be computed. Notice also that `tijspec` with its -40 dB/dec high frequency roll-off (Figure 8.10) is used here for $b_{21}(\omega)$.

The first step cross coupling bounds from (8.28) are also computed with the command `mcbnd11`, using the criterion function `fcouple1.m` (default name), and placed in `ex8.bnd` under the default name `couple1`:

```
mcbnd11('ex8','tijspeca','tiispec',[],[],'ex8','iw11',' w12 w11');
                                        % Default bound name: couple1
```

Notice the order of the two specification vector arguments, `tijspec` with 2 columns, and `tiispec` with 3 columns which signals that the Horowitz bounds from (8.28) are to be computed, using the criterion function `fcouple1.m` (if no other criterion function name is given as an argument). Notice also that `tijspeca` without high frequency roll-off (Figure 8.10) is used here for $b_{12}(\omega)$.

As mentioned above, a voluntary 6 db sensitivity bound with respect to $L_{10}/W_{11\text{nom}}$, from $\left|1/\left(1+\left(L_{10}(j\omega)/W_{11}(j\omega)\right)\right)\right| \le 2$, is computed with the standard SISO bound computation function `cbnd` with the usual criterion function `fodsrs.m`. The sensitivity bounds are equal for all frequencies since the templates $1/W_{11}(j\omega)$ are all equal (Figure 8.10), and the specification is equal (6 dB). The sensitivity bounds are placed under the name `sens` in `ex8.bnd`:

```
cbnd('ex8','sens',[],[],'ex8','iw11','sens','fodsrs');
```

The user may study all computed bounds with the command showbnd. We select the dominant bounds, and display them in Figure 8.12.

```
showbnd('ex8',[],[.1 .2],'servo1',[], ...
                  [.5 1 2 5 10 20 ],'couple1',[],[50 70 100],'sens');
axis([-360 0 -20 50 ]),mgrid(12,7)              % Figure 8.12
```

It turns out that the servo bounds are dominant for 0.1 and 0.2 rad/s, the cross coupling bounds are dominant for 0.5 - 20 rad/s, and the single sensitivity bound is dominant for still higher frequencies. We point out again, that the sensitivity bounds are not mandatory. The cross coupling bounds for 50, 70, and 100 rad/s are mandatory. They are found inside the sensitivity bound. The user may display them, and she will then find that they have "islands" due to the "holes" in the computed templates.

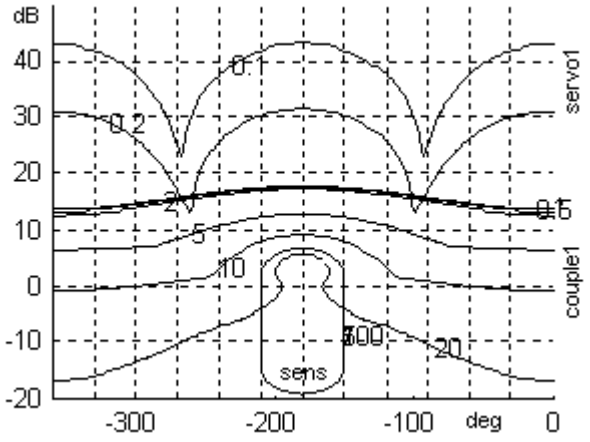


Figure 8.12. Dominant Horowitz bounds for $L_{10}/W_{11\text{nom}}$ in the first design step, displayed with the command `showbnd('ex8',[],[.1 .2],'servo1',[],[.5 1 2 5 10 20], ... 'couple1',[],[50 70 100],'sens');`

Feedback compensator design

Figure 8.11 shows that the nominal $1/W_{11nom}$ in `iw11.tpl` is recorded at -360 deg. In order to have a more convenient design of $L_{10}(j\omega)/W_{11nom}(j\omega)$, we shift the nominal in `iw11.tpl` by +360 deg:

```
[w_nom,nom]=gettpl('iw11','nom');
nom=nom+360;
insert('iw11.tpl',nom,'nom','r');
```

Next display the Horowitz bounds of Figure 8.12. Edit the controller function m-file for $L_{10}(s)$, `L10.m`, whose final form is shown in Figure 8.13. With the command `cdesign`, show the designed nominal open loop $L_{10}(j\omega)/W_{11nom}(j\omega)$ together with the bounds in Figure 8.14. One possible command sequence is:

```
h2=cdesign('iw11.tpl','L10');
axis([-180 -80 -30 50 ])
showbnd('ex8',gcf,[.1 .2],'servo1',[], ...
                  [.5 1 2 5 10 20],'couple1',[],[50],'sens');
hngrid,mgrid(10,8)                          % Figure 8.14
```

```
function [L]=L10(s)
% L10.m, first step feedback compensator transfer function

L = 15*(s/80 + 1)./( s.*(s/100 + 1).*(s/30 + 1) );
```

Figure 8.13. Controller function file `L10.m` *for the feedback controller of the first MIMO design step, realizing the transfer function*

$$L_{10}(s)=\frac{15(s/80+1)}{s(s/30+1)(s/100+1)} \tag{8.40}$$

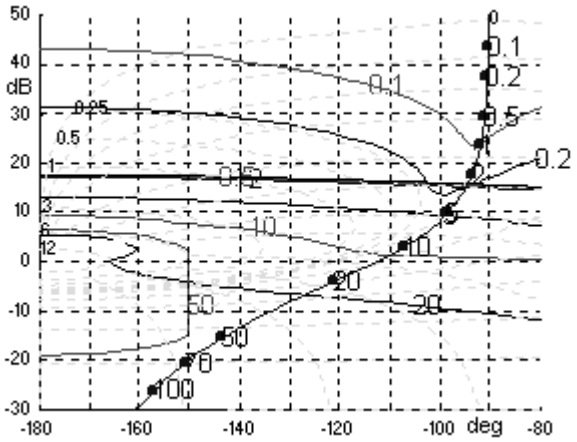


Figure 8.14. The final nominal open loop of the first MIMO design step, $L_{10}(j\omega)/W_{11nom}(j\omega)$, parametrized in rad/s, in a NIchols diagram with the dominant Horowitz bounds from Figure 8.12. $L_{10}(s)$ is given in (8.40), and $W_{11nom}(s)$=1 from Figure 8.11.

Modified servo specification

In contrast to a regular SISO design where one proceeds directly with the prefilter design, we have here to compute here the open loop templates, $L_{10}(j\omega)/W_{11}(j\omega)$, which we place in the template file `L10 w11.tpl`,

```
tplfop('L10_w11','*',[],'iw11',1,'L10');
```

in order to find the true values of $b^*_{d11}(\omega)$ in (8.29), and modify the servo specification to get $\left[a_{11}(\omega)+b^*_{d11}(\omega),\ b_{11}(\omega)-b^*_{d11}(\omega)\right]$ for the prefilter design, cf. (8.26). This is done with the command `bd11spc`, whose arguments include the original servo specification, `tiispec` from `ex8.spc`; the b_{d11}-matrix, called `servo1bd11` in `ex8.bnd`; and the template file, `L10_w11.tpl`, for the designed open loop, $L_{10}(j\omega)/W_{11}(j\omega)$ which includes the nominal $L_{10}(j\omega)/W_{11\text{nom}}(j\omega)$ for which $b^*_{d11}(\omega)$ is found. The new specification $\left[a_{11}(\omega)+b^*_{d11}(\omega),\ b_{11}(\omega)-b^*_{d11}(\omega)\right]$, whose default name becomes `tiispecm`, is placed by default in `ex8.spc`. The original and modified servo specifications are shown in Figure 8.15.

```
bd11spc('ex8','tiispec','servo1bd11',[],'L10_w11');
            % default name for new servo spec: tiispecm in ex8.spc
showspc('ex8','tiispec','freq')
showspc('ex8','tiispecm','freq','cx',gcf)       % Figure 8.15
```

Notice that $\left[a_{11}(\omega)+b^*_{d11}(\omega),\ b_{11}(\omega)-b^*_{d11}(\omega)\right]$ is computed only for those (low) template frequencies for which $b^*_{d11}(\omega)$ satisfies $0\le b^*_{d11}(\omega)\le\left(b_{11}(\omega)-a_{11}(\omega)\right)/2$. If the roll-off of `tijspec` had been -60 dB/dec or -80 dB/dec, as mentioned in the discussion after Figure 8.10, then acceptable $b^*_{d11}(\omega)$-values would have been obtained for all template frequencies. The lower modified servo specification, $a_{11}(\omega)+b^*_{d11}(\omega)$, has however no significance beyond the frequency where normally the lower specification would have had a cut-off, cf Figure 2.10. In our example, the cut-off frequency would be 3 rad/s.

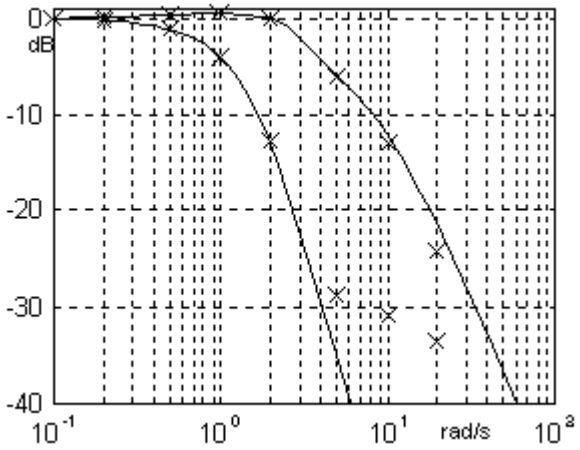


Figure 8.15. The original servo specification $\left[a_{11}(\omega),\ b_{11}(\omega)\right]$ *from Figure 8.9 called* `tiispec` *in the specification file* `ex8.spc` *(solid line), together with the modified specification* $\left[a_{11}(\omega)+b^*_{d11}(\omega),\ b_{11}(\omega)-b^*_{d11}(\omega)\right]$, *called* `tiispecm` in `ex8.spc` *(x). The modified specification was computed from (8.29) and the designed nominal open loop with the* command `bd11spc('ex8','tiispec','servo1bd11',[],'L10_w11');`

Prefilter design

From now on the design proceeds just like in the standard SISO case. The first loop complementary sensitivity function templates

$$\overline{S}_1(j\omega) = \frac{L_{10}(j\omega)/W_{11}(j\omega)}{1+\left(L_{10}(j\omega)/W_{11}(j\omega)\right)} \tag{8.41}$$

are computed, and placed into the template file `cosens1.tpl` by

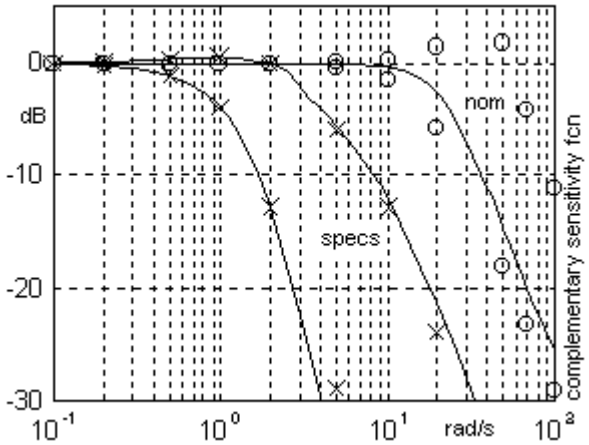
```
tplfop('cosens1','idsrs',[],'L10 w11');
```

or by

```
tplfop('cosens1','iosrs',[],'iw11',1,'L10');
```

The nominal, and gain extents of the complementary sensitivity function templates are shown in a Bode diagram, together with the original and modified servo specifications, in Figure 8.16.

```
h0=fdesign('cosens1.tpl',[],'new');              % Figure 8.16
showspc('ex8','tiispecm','freq','cx',gcf);       % modified spec (x)
showspc('ex8','tiispec','freq',[],gcf);          %  orig spec (-)
```



*Figur 8.16. The gain extent (*o*), and the nominal (solid line) of the complementary sensitivity function (8.41) together with the original servo specifications (solid lines) and the* $b^*_{d11}(\omega)$*-modified servo specifications (*x*).*

```
function [F]=F11(s)
% F11.m  is the prefilter file for the first step

F = (1)./( (s/2 + 1).*(s/10 + 1) );
```

Figure 8.17. Controller function file `F11.m` *for the prefilter of the first MIMO design step, realizing the transfer function*

$$F_{11}(s) = \frac{1}{(s/2+1)(s/10+1)} \tag{8.42}$$

Next the controller function m-file, `F11.m`, defining the prefilter is edited, and tested with the command `fdesign`. The finally chosen prefilter is found in Figure 8.17. The nominal and the gain extent of

$$T_1(j\omega) = F_{11}(j\omega)\frac{L_{10}(j\omega)/W_{11}(j\omega)}{1+\left(L_{10}(j\omega)/W_{11}(j\omega)\right)} \tag{8.43}$$

is displayed in Figure 8.18 which is the same diagram as Figure 8.16 from which (8.41) is replaced by (8.43) with the prefilter given in (8.42).

```
h1=fdesign('cosens1.tpl','F11.m', h0);     % Figure 8.18
```

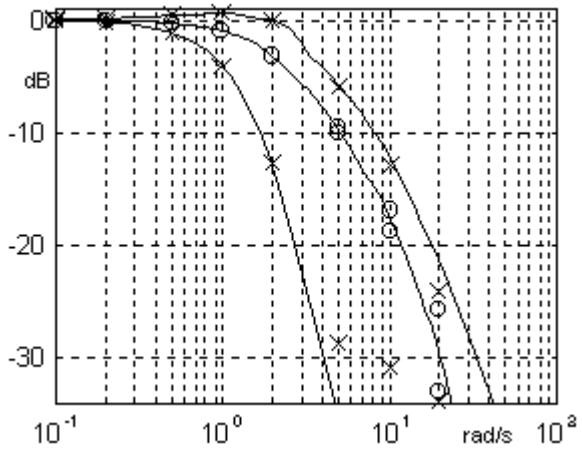


*Figure 8.18. The gain extent (*o*), and the nominal (solid line) of the final first step closed loop (8.42), (8.43) together with the original servo specifications (solid lines) and the* $b^*_{d11}(\omega)$*-modified servo specifications (*x*).*

8.2.5 The second design step

Template computations

After the successful design of the first loop feedback compensator and prefilter in Section 8.2.4, we turn to the second design step, see Section 8.1.5. In addition to the template files computed with the command `mtpl1` in the beginning of Section 8.2.4, one needs the templates of the first step closed loop frequency function $T_1(j\omega)$ in (8.43) and the first sensitivity function

$$S_1(j\omega) = \frac{1}{1+\left(L_{10}(j\omega)/W_{11}(j\omega)\right)} \tag{8.44}$$

in order to compute the templates appearing in (8.33), (8.35), and (8.36). The template files `closed1.tpl` for $T_1(j\omega)$ in (8.43), and `sens1.tpl` for $S_1(j\omega)$ in (8.44) are created with the commands

```
tplfop('closed1','rsrs',[],'iw11',1,'L10','F11');     % T1 in (8.43)
tplfop('sens1','odsrs',[],'iw11',1,'L10');      % S1 in (8.44)
```

Then the templates for the "equivalent plant", $1/W_{22}^{e}(j\omega)$ in (8.33), and the "filtered" first step closed loop from (8.35),

$$T_1^f = \frac{W_{21}(j\omega)}{W_{22}^{e}(j\omega)} \cdot \frac{L_{10}(j\omega)}{1+\left(L_{10}(j\omega)/W_{11}(j\omega)\right)} \cdot F_{11}(j\omega) \qquad (8.45)$$

are computed and placed into the template files `iw22e.tpl`, and `wwc1.tpl`, respectively, with the command

```
mtpl2('iw22e','wwc1','iw22','w12 w11','w21 w22','sens1','closed1');
```

The nominal $1/W_{22nom}^{e}(j\omega)$ in `iw22e.tpl` is shifted by +360 degrees,

```
[w nom,nom]=gettpl('iw22e','nom');
nom=nom+360;
insert('iw22e.tpl',nom,'nom','r');
```

and then the templates of $1/W_{22}^{e}(j\omega)$ are displayed in Figure 8.19a. It is clear that all nominals equal 1, and that all templates are very similar to each other, lying along the the positive real axis. An appreciation of the phase extent is gained in Figure 8.19b which presents a zoomed picture of the template for 5 rad/s. We note that the gain extent is about 20 degrees which could be compared with the templates of $1/W_{22}(j\omega)$ in Figure 8.11 which all have no phase extent at all. Moreover, the templates are not connected, due to the original grid computation method in Section 8.2.1.

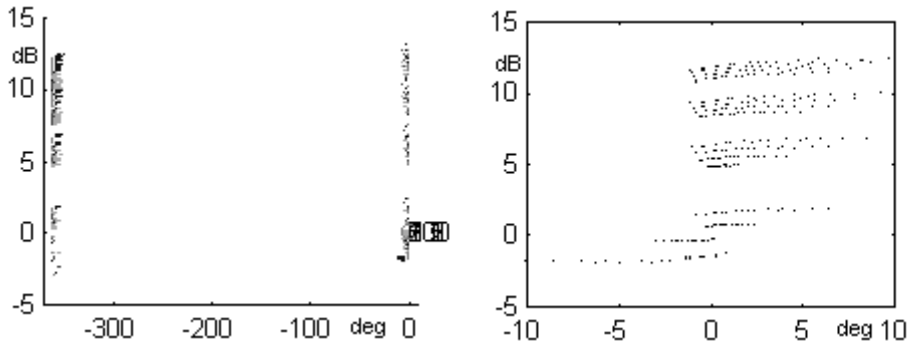


Figure 8.19. a) Left picture: The templates of $1/W_{22}^{e}(j\omega)$, and nominal $1/W_{22nom}^{e}(j\omega)$ generated with the command `showtpl('iw22e');` *b) Right picture: Zoomed and wrapped plot of $1/W_{22}^{e}(5j)$ generated with* `plot(wrap(gettpl('iw22e',50),180),'.');`

Bound computation

The Horowitz bounds for L_{20}/W_{22nom}^{e} from (8.35) which are true second step cross coupling bounds, are computed with the command `cbnd`, with the flag mimo2 as the last argument, and the criterion function `fcouple2.m`. To get a fast high frequency roll-off for the bounds, we choose the specification `tijspeca` for $b_{21}(\omega)$ in (8.35), since `tijspeca` is constant = -20 dB, rather than `tijspec` which includes a high frequency roll off. The bounds are collected in the bound file `ex8 2.bnd` under the name `couple2a`, and shown in Figure 8.20.

```
cbnd('ex8 2','tijspeca',[],[],'ex8','iw22e','couple2a',...
                          'fcouple2',[],[],[],'wwc1','mimo2');
showbnd('ex8 2',[],[],'couple2a','r');  % Figure 8.20
```

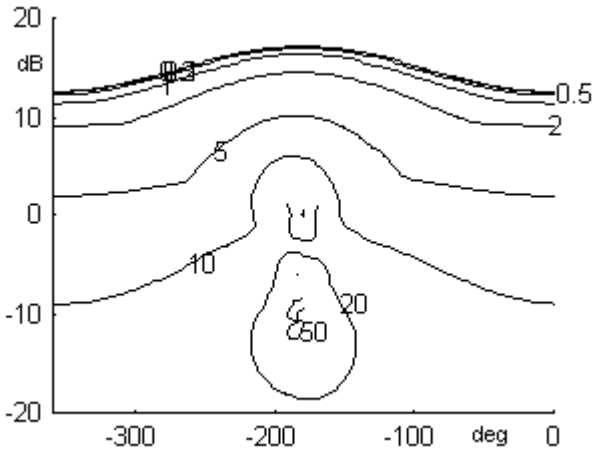


Figure 8.20. The cross coupling bounds from (8.35) generated by the command `cbnd('ex8_2','tijspeca',[],[],'ex8','iw22e','couple2a','fcouple2',...[],[],[],'wwc1','mimo2');` *Notice that the bounds for 20 rad/s and 50 rad/s have "islands" due to the discontinuous representation of the templates, compare e.g Figure 8.19.*

Next comes the computation of the tolerance bounds for $L_{20}/W^{e}_{22\text{nom}}$ from the servo specification (8.36). which is a completely standard SISO rsrs specification. The name given to the new set of bounds is `servo2`, shown in Figure 8.21.

```
cbnd('ex8 2','tiispec',[],[],'ex8','iw22e','servo2','frsrs');
showbnd('ex8_2',[],[],'servo2');    % Figure 8.21
```

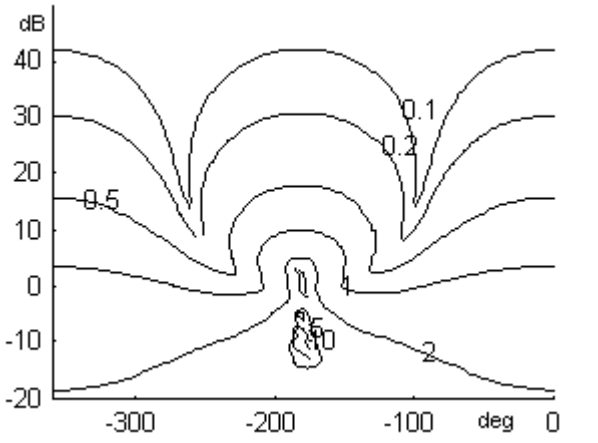


Figure 8.21. The tolerance bounds emanating from the second step servo specification (8.36), computed with the command
`cbnd('ex8_2','tiispec',[],[],'ex8','iw22e','servo2','frsrs');`

And finally, 6 dB sensitivity bounds, named `sens2` are computed from

$$\left|\frac{1}{1+\left(L_{20}(j\omega)/W_{22}^{e}(j\omega)\right)}\right| \le 2 \qquad (8.46)$$

with a standard SISO `cbnd` command. With reference to Section 8.17, the stability of this second loop correctly reflects the closed loop stability. Therefore the sensitivity bounds have to be adhered to. The sensitivity bounds are shown in Figure 8.22.

```
cbnd('ex8 2','sens',[],[],'ex8','iw22e','sens2','fodsrs');
showbnd('ex8 2',[],[],'sens2')      % Figure 8.22
```

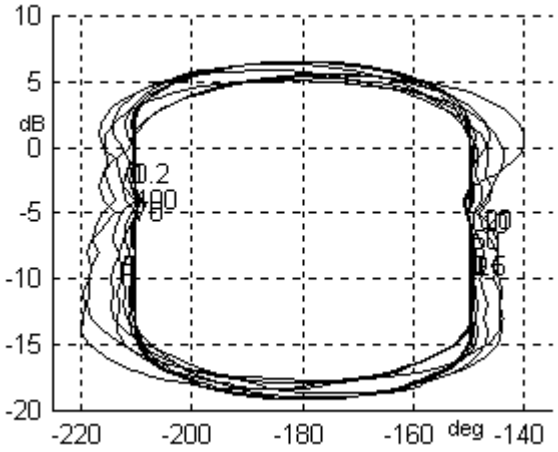


Figure 8.22. The sensitivity bounds, emanating from (8.46), computed with the command `cbnd('ex8_2','sens',[],[],'ex8','iw22e','sens2','fodsrs');`

From Figures 8.20-8.22, the dominant bounds are selected,

```
showbnd('ex8_2',[],[.2 .5 1 2 5 10],'couple2a',[], ...
            [.1 .2 ],'servo2',[],[10 20 50 70 100],'sens2',[]);
```

and shown in Figure 8.23. We note that the servo bounds are dominant for the lowest frequencies, the cross coupling bounds are dominant between 0.2 and 10 rad/s, and the sensitivity bounds dominate for frequencies higher than 10 rad/s.

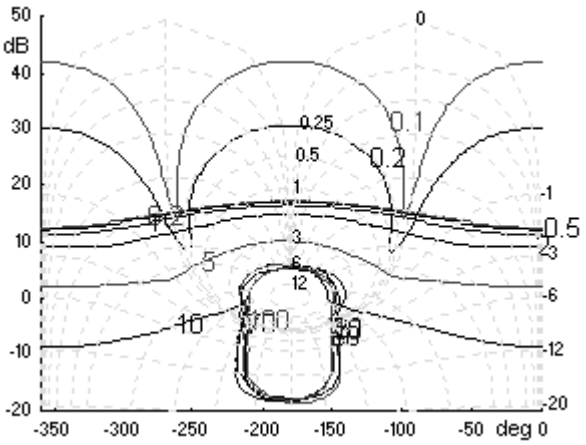


Figure 8.23. Dominant Horowitz bounds from Figures 8.20-8.22.

Feedback compensator design

The controller function m-file, `L20.m`, representing the second loop feedback compensator $L_{20}(s)$ is edited. The final version of `L20.m` is found in Figure 8.24. The nominal open loop $L_{20}/W_{22\text{nom}}^{e}$ is tested against the Horowitz bounds of Figure 8.23, with the command

```
h2=cdesign('iw22e.tpl','L20');   % show  L20*(1/W22enom) in Fig 8.25
```

We notice that $L_{20}/W_{22\text{nom}}^{e}$ sits right on the bounds for 2, 5, 10 rad/s.

```
function [L]=L20(s)
% L20.m.  Second step feedback compensator

L = (7)*./( s.*(s/30 + 1) );
```

Figure 8.24. Controller function file `L20.m` *for the feedback compensator of the second MIMO design step, realizing the transfer function*

$$L_{20}(s)=\frac{7}{s(s/30+1)} \tag{8.47}$$

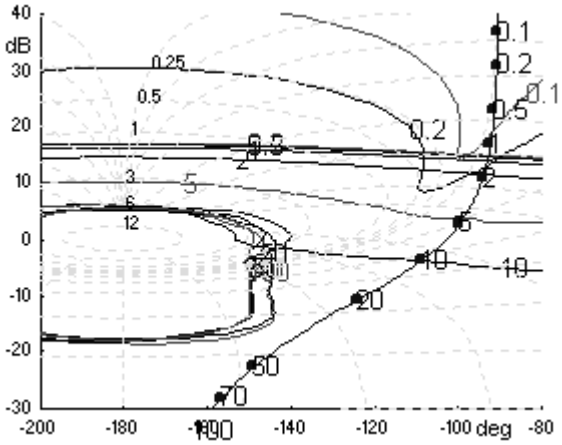


Figure 8.25. The final nominal open loop of the first MIMO design step, $L_{20}(j\omega)/W^{e}_{22\text{nom}}(j\omega)$, parametrized in rad/s, in a NIchols diagram with the dominant Horowitz bounds from Figure 8.23. $L_{20}(s)$ is given in (8.47), and $W^{e}_{22\text{nom}}(s)$=1 from Figure 8.19.

Prefilter design

The second loop is closed, and the complemantary sensitivity function templates

$$\overline{S}_2(j\omega)=\frac{L_{20}(j\omega)/W^{e}_{22}(j\omega)}{1+\left(L_{20}(j\omega)/W^{e}_{22}(j\omega)\right)} \tag{8.48}$$

are computed with the command

```
tplfop('cosens2','iosrs',[],'iw22e',1,'L20');
```

The adherence of $\overline{S}_2(j\omega)$ to the servo specifications pertaining to $T_{22}(j\omega)$, (8.18) and Figure 8.9, are checked before the design of the prefilter $F_{22}(s)$. A suitable prefilter is found in Figure 8.26. The final closed loop design, $\overline{S}_2(j\omega)F_{22}(j\omega)$, is displayed in Figure 8.27.

```
h0=fdesign('cosens2.tpl');
showspc('ex8','tiispec','freq',[],gcf);
h1=fdesign('cosens2.tpl','F22.m');
```

```
function [F]=F22(s)
% F22.m is the prefilter for the second step

F = (1)./( (s/2 + 1).*(s/5 + 1).*(s.*s/100 +1.4*s/10 +1) );
```

Figure 8.26. Controller function file `F22.m` *for the prefilter of the second MIMO design step, realizing the transfer function*

$$F_{22}(s)=\frac{1}{(s/2+1)(s/5+1)\left(s^2/100+14s/10+1\right)} \tag{8.49}$$

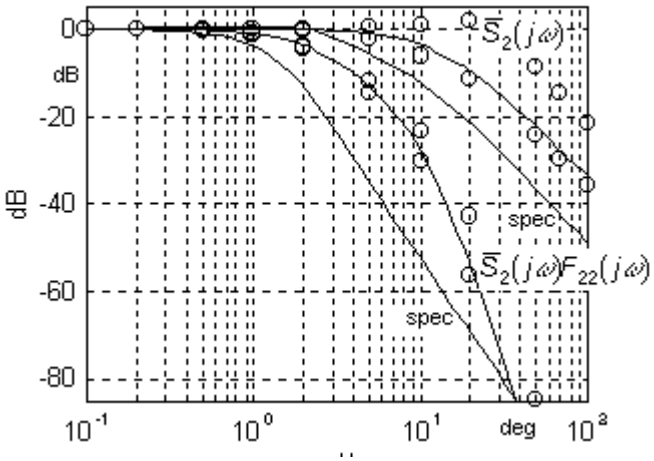


Figure 8.27. The gain extent (o), and the nominal (solid lines) of the second step complementary sensitivity function $\bar{S}_2(j\omega)$ in (8.48), *and the final second step closed loop* $\bar{S}_2(j\omega)F_{22}(s)$, *where* $F_{22}(s)$ *is defined in (8.49), together with the original servo specifications* `tiispec` *from Figure 8.9*.

We notice in Figure 8.27 that the servo specifications are well satisfied, which is not surprising since the cross coupling specifications were the dominant in the mid frequency range. The closed loop seems to violate the lower specification for high frequencies, but we remind ourselves that we should have cut off the lower specification at about 3 rad/s, and hence no violation occurs.

8.2.6 The feedback compensator and prefilter in matrix form

With $L_{10}(s)$ defined in (8.40), $L_{20}(s)$ in (8.47), $F_{11}(s)$ in (8.42), $F_{22}(s)$ in (8.49), and $P_0(s)$ given in Figure 8.8 and (8.38), we use (8.5) to compute $G(s)$.

$$G(s) = \begin{bmatrix} \dfrac{10(s/80+1)}{(s/30+1)(s/100+1)} & -\dfrac{7/3}{(s/30+1)} \\ -\dfrac{5(s/80+1)}{(s/30+1)(s/100+1)} & \dfrac{14/3}{(s/30+1)} \end{bmatrix} \tag{8.50}$$

$$F(s) = \begin{bmatrix} \dfrac{1}{(s/2+1)(s/10+1)} & 0 \\ 0 & \dfrac{1}{(s/2+1)(s/5+1)\left(s^2/100+14s/10+1\right)} \end{bmatrix} \tag{8.51}$$

8.2.7 Simulations

There are various ways to compute the closed loop transfer functions for a large number of cases and simulate them in the frequency and time domains. We chose to use the symbolic mathematics program Maple, and translate the result back into Matlab. The frequency domain simulation, for 625 cases, are done with the m-file `simfreq.m` which is reproduced in Figure 8.28. The time domain simulations are done with the m-file `simtime.m` (Figure 8.29) that uses the Matlab Control Systems Toolbox function `step`, for the step response simulations. The graphs are found in Figures 8.30-8.xx.

```
w = logspace(-1,2); s = sqrt(-1)*w;

T11f = zeros(625,50); T12f = zeros(625,50);
T21f = zeros(625,50); T22f = zeros(625,50);

n = 0;                    %counter

for a = 2:1:6,       % k11 in [2,6]
  for b = 0.5:0.25:1.5,     % k12 in [.5,1.5]
    for c = 0.5:0.25:1.5;     % k21 in [.5, 1.5]
      for d = 2:1:6;       % k22 in [2,6]

n = n+1;

e = (13800*s.^3+180000*s.^2+ ...
78750*a*s*d-78750*b*s*c+2*s.^5+320*s.^4+1800000*a*s-900000* ...
b*s-140*s.^3*c+280*s.^3*d-18200*s.^2*c+36400*s.^2*d-420000*s* ...
c+840000*s*d+750*a*s.^3+82500*a*s.^2+6300000*a*d-375*b*s.^3- ...
41250*b*s.^2-6300000*b*c);

T11 = 7500*(s+80).*(2*a*s.^2-b* ...
s.^2+60*a*s-30*b*s-210*b*c+210*a*d)./e./(s+2)./(s+10);

T12 = -1400*(s+100).*(s+30).*s.*(a-2*b)./ ...
(0.01*s.^2+0.14*s+1)./(s+5)./(s+2)./e;

T21 = 7500*(2*c-d)*(s+80).*s.*(s+30)./e./(s+2)./(s+10);

T22 = -700*(-1125*a*s*d-90000*a*d+1125*b*s*c+90000*b*c+ ...
2*s.^3*c-4*s.^3*d+260*s.^2*c-520*s.^2*d+6000*s*c-12000*s*d)./ ...
(0.01*s.^2+0.14*s+1)./(s+5)./(s+2)./e;

T11f(n,:) = T11; T12f(n,:) = T12;
T21f(n,:) = T21; T22f(n,:) = T22;

     end
    end
  end
end

figure,showspc('ex8','tiispec','freq',[],gcf),hold on,
                                    semilogx(w,20*log10(abs(T11f)));
figure,  showspc('ex8','tijspeca','freq',[],gcf); hold on,
                                    semilogx(w,20*log10(abs(T12f)));
figure,  showspc('ex8','tiispec','freq',[],gcf), hold on,
                                    semilogx(w,20*log10(abs(T21f)));
figure,  showspc('ex8','tijspeca','freq',[],gcf); hold on,
                                    semilogx(w,20*log10(abs(T22f)));
```

Figure 8.28. The m-file `simfreq.m` *for the frequency domain simulation of 625 cases of the final closed loop transfer function T(s) in (8.4).*

```
t = 0:0.1:5;

T11f = zeros(625,51); T12f = zeros(625,51);
T21f = zeros(625,51); T22f = zeros(625,51);

n = 0;                  %counter

for a = 2:1:6,          % k11 in [2,6]
  for b = 0.5:0.25:1.5,    % k12 in [.5,1.5]
    for c = 0.5:0.25:1.5;    % k21 in [.5, 1.5]
      for d = 2:1:6;          % k22 in [2,6]

n = n+1;

%e = (13800*s.^3+180000*s.^2+ ...
%78750*a*s*d-78750*b*s*c+2*s.^5+320*s.^4+1800000*a*s-900000* ...
%b*s-140*s.^3*c+280*s.^3*d-18200*s.^2*c+36400*s.^2*d-420000*s* ...
%c+840000*s*d+750*a*s.^3+82500*a*s.^2+6300000*a*d-375*b*s.^3- ...
%41250*b*s.^2-6300000*b*c);

e = [ ...
+2 ...                                          %*s.^5
+320 ...                                        %*s.^4
(+13800-140*c+280*d+750*a-375*b) ...                        %*s.^3
(+180000-18200*c+36400*d+82500*a-41250*b) ...         %*s.^2
(+78750*a*d-78750*b*c+1800000*a-900000*b-420000*c+840000*d) ...   %*s
(+6300000*a*d-6300000*b*c) ...
];

T11num = 7500*conv([1 80],[(2*a-b) (+60*a-30*b) (-210*b*c+210*a*d)]);
T11den = conv( conv(e,[1 2]),[1 10] );

T12num = -1400*conv([1 130 3000],[(a-2*b) 0]);
T12den = conv( conv([0.01 0.14 1],[1 5]), conv([1 2],e) );

T21num = 7500*(2*c-d)*[1 110 2400 0];
T21den = conv(e,[1 12 20]);

T22num = -700*[(+2*c-4*d)  (+260*c-520*d) ...
      (-1125*a*d+1125*b*c+6000*c-12000*d) (-90000*a*d+90000*b*c)];
T22den = conv( conv([0.01 0.14 1],[1 5]), conv([1 2],e) );

T11 = step(T11num,T11den,t)'; T12 = step(T12num,T12den,t)';
T21 = step(T21num,T21den,t)'; T22 = step(T22num,T22den,t)';

T11f(n,:) = T11; T12f(n,:) = T12; T21f(n,:) = T21; T22f(n,:) = T22;

     end
    end
  end
end

figure, plot(t,T11f);title('T11'); figure, plot(t,T12f);title('T12');
figure, plot(t,T21f);title('T21'); figure, plot(t,T22f);title('T22');
```

Figure 8.29. The m-file `simtime.m` *for the time domain simulation of 625 cases of the final closed loop transfer function T(s) in (8.4).*

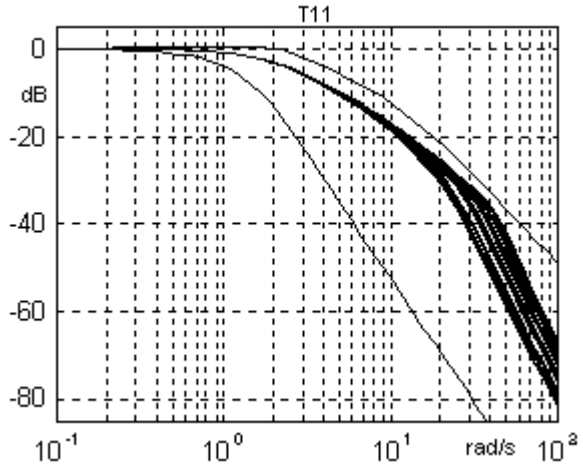


Figure 8.30. Simulation of 625 cases of the final transfer function $T_{11}(s)$ in the frequency domain, together with the servo specification from Figure 8.9.

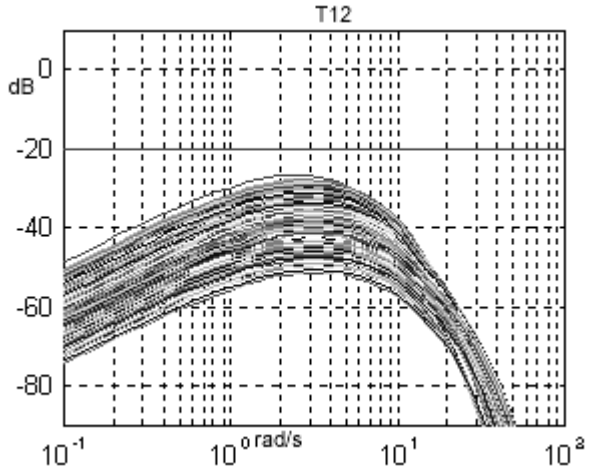


Figure 8.31. Simulation of 625 cases of the final transfer function $T_{12}(s)$ in the frequency domain, together with the cross coupling specification from Figure 8.10.

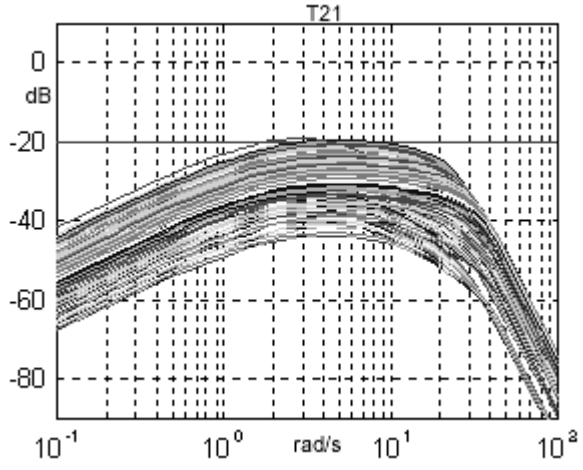


Figure 8.32. Simulation of 625 cases of the final transfer function $T_{21}(s)$ in the frequency domain, together with the cross coupling specification from Figure 8.10.

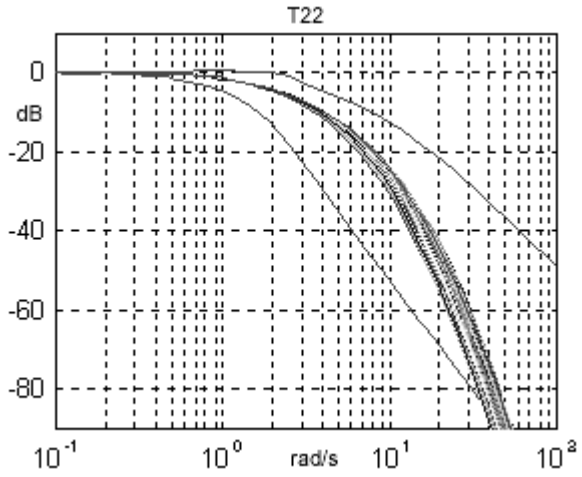


Figure 8.33. Simulation of 625 cases of the final transfer function $T_{22}(s)$ in the frequency domain, together with the servo specification from Figure 8.9.

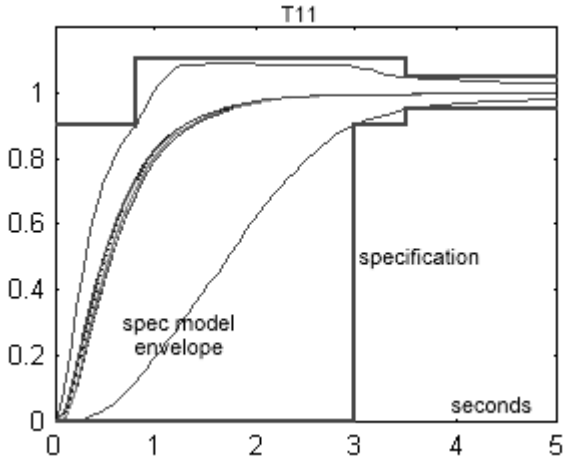


Figure 8.34. Simulation of 625 step response cases for the final transfer function $T_{11}(s)$ in the time domain, together with the servo specification from Figure 8.9.

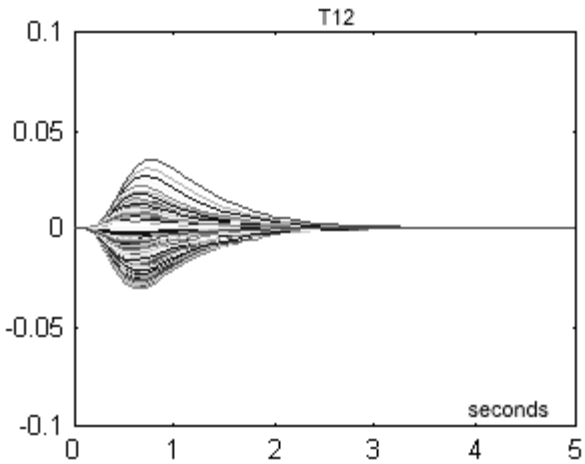


Figure 8.35. Simulation of 625 step response cases for the final transfer function $T_{12}(s)$ in the time domain.

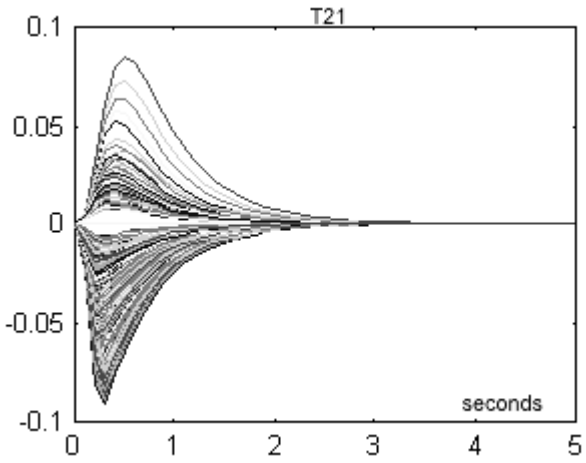


Figure 8.36. Simulation of 625 step response cases for the final transfer function $T_{21}(s)$ in the time domain.

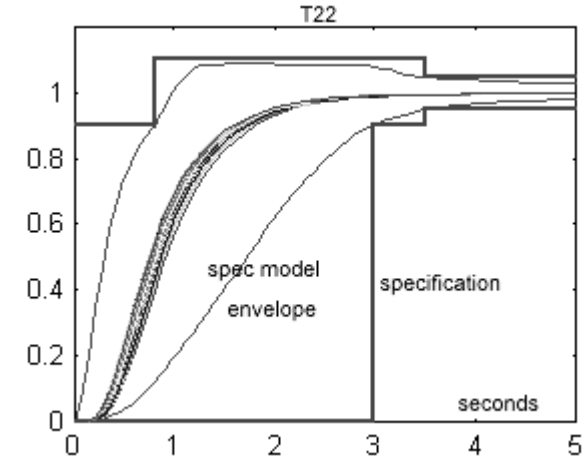


Figure 8.37. Simulation of 625 step response cases for the final transfer function $T_{22}(s)$ in the time domain, together with the servo specification from Figure 8.9.

The figures show that the servo specification are well satisfied in the time and frequency domains. If, presumably, the time domain specification for the cross coupling is that the maximal response be less than 10% of the step amplitude, then the cross coupling responses are also very satisfactory.

One could proceed with a third design step to redesign the first loop in view of the fact that the second is now designed. It might be possible to lower the bandwidth. This is, however, left for the user to do!