Problem 20.4.

$$\mathbf{T_o}(s) = \begin{bmatrix} \dfrac{4}{s^2+3s+4} & \dfrac{\alpha s(s+5)}{(s^2+3s+4)(s^2+4s+9)} \\ \\ 0 & \dfrac{9}{s^2+4s+9} \end{bmatrix} \tag{20.20.22}$$

20.4.1 *We first observe that the system poles are the roots of* $(s^2+3s+4)(s^2+4s+9)$ *and that*

$$\det\{\mathbf{T_o}(s)\} = \frac{36}{(s^2+3s+4)(s^2+4s+9)} \tag{20.20.23}$$

Therefore, there are no zeros.

20.4.2 *Since the plant is stable and minimum-phase, an unstable pole-zero cancellation is impossible. Thus the loop is internally stable.*

20.4.3 *If* $D_o(s)$ *is the vector of output disturbances, then the loop response to these disturbances is given by*

$$\begin{aligned} Y(s) &= \mathbf{S_o}(s)D_o(s) \\ &= (\mathbf{I} - \mathbf{T_o}(s))D_o(s) \\ &= \begin{bmatrix} \dfrac{s(s+3)}{s^2+3s+4} & \dfrac{-s(s+5)}{(s^2+3s+4)(s^2+4s+9)} \\ \\ 0 & \dfrac{s(s+4)}{s^2+4s+9} \end{bmatrix} D_o(s) \end{aligned} \tag{20.20.24}$$

We next use SIMULINK to simulate the loop with a unit step output disturbance in channel 1 at time $t = 1$ *and a unit step output disturbance in channel 2 at time* $t = 10$*. The results are shown in Figure 20.2.*

As expected, a disturbance in channel 1 does not produce any effect in the output of channel 2, since the output sensitivity matrix $\mathbf{S_o}(s)$ *is upper triangular. However, for the same structural reason, a disturbance in channel 2 will affect the output of channel 1. This is evident from Figure 20.2.*

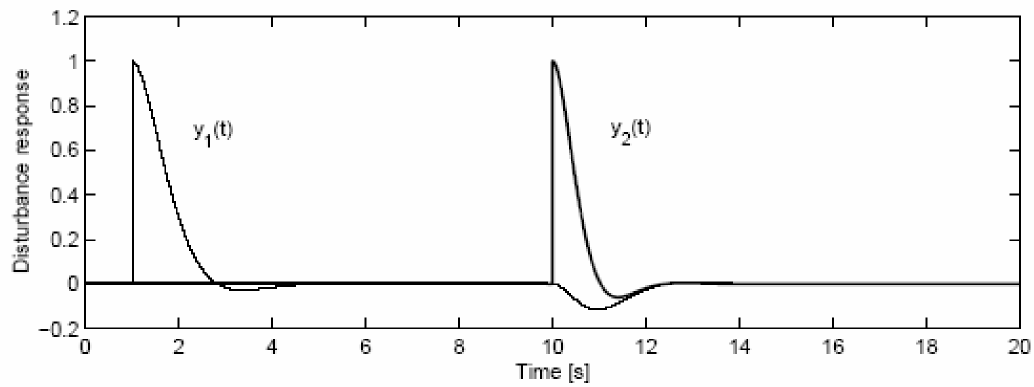

Figure 20.2. Output disturbance response of a triangular 2×2 MIMO system

Problem 20.6. *We first build the transfer function* $\mathbf{T_o}(s)$ *using the MATLAB code*

```
>>h11=tf([1 2],[1 4 0]);h12=tf(0.5,[1 2 0]);
>>h21=h12;h22=tf([2 6],[1 6 8 0]);goc=[h11 h12 ; h21 h22];
>>So=minreal(inv(eye(2,2)+goc));To=minreal(eye(2,2)-So);
```

20.6.1 *We first have to say that there is no way we can know whether an unstable pole zero cancellation has arisen, since we have been given only the product* $\mathbf{G_o}(s)\mathbf{C}(s)$*. We are therefore limited to answer the question whether the complementary sensitivity is stable or unstable. This is easily checked with the MATLAB command* ***pole****, to compute the complementary sensitivity poles. These poles are strictly inside the left-half plane, thus establishing that* $\mathbf{T_o}(s)$ *is stable.*

20.6.2 *The singular values of* $\mathbf{T_o}(s)$ *can be computed using the MATLAB command* ***sigma****. When these singular values are plotted, they appear as in Figure 20.4.*

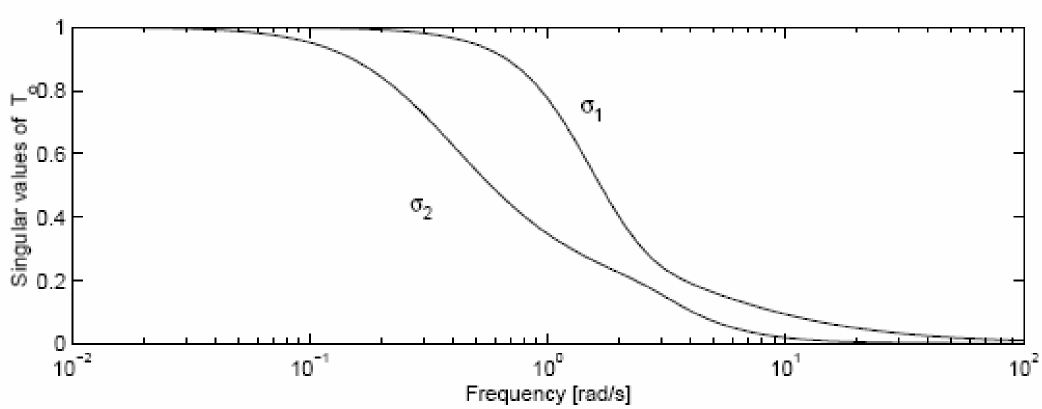

Figure 20.4. Singular values of the complementary sensitivity.

Problem 21.6.

21.6.1 *The diagonalizing precompensator* $\mathbf{P}(s)$ *is computed from*

$$\mathbf{G_o}(s)\mathbf{P}(s) = \mathbf{H}(s) \Longleftrightarrow \mathbf{P}(s) = [\mathbf{G_o}(s)]^{-1}\mathbf{H}(s) \tag{21.21.46}$$

where $\mathbf{H}(s)$ *is chosen as*

$$\mathbf{H}(s) = \begin{bmatrix} \dfrac{1}{s+1} & 0 \\ 0 & \dfrac{1}{s+1} \end{bmatrix} \tag{21.21.47}$$

Then the matrix $\mathbf{P}(s)$ *can be computed using the following MATLAB code*

```
>> g11=tf(8,[1 6 8]);g12=tf(0.5,[1 8]);
>> g21=tf(-2,[1 4]);g22=tf([6 6],[1 5 6]);
>> Go=[g11 g12;g21 g22];
>> h11=tf(1,[1 1]);h12=tf(0,1);h21=h12;h22=tf(1,[1 1]);
>>H=[h11 h12;h21 h22];
>>detGo=minreal(g11*g22-g12*g21);
>>p11=minreal(g22*h11/detGo);p12=-minreal(g12*h22/detGo);
>>p21=-minreal(g21*h11/detGo);p22=minreal(g11*h22/detGo);
>>P=[p11 p12;p21 p22]
```

where, as usual, the function **minreal** *is used to simplify common terms in transfer functions.*

We next need to design a diagonal PI controller for the precompensated plant $\mathbf{H}(s)$*; we choose the controllers so that the complementary sensitivity in every channel is given by*

$$T_{o1,2}(s) = \frac{9}{s^2 + 4s + 9} \tag{21.21.48}$$

This leads to

$$
\mathbf{C}(s) = \begin{bmatrix} \dfrac{3(s+3)}{s} & 0 \\ 0 & \dfrac{3(s+3)}{s} \end{bmatrix} \qquad (21.21.49)
$$

The performance is evaluated using SIMULINK, we obtain the step response shown in Figure 21.4.

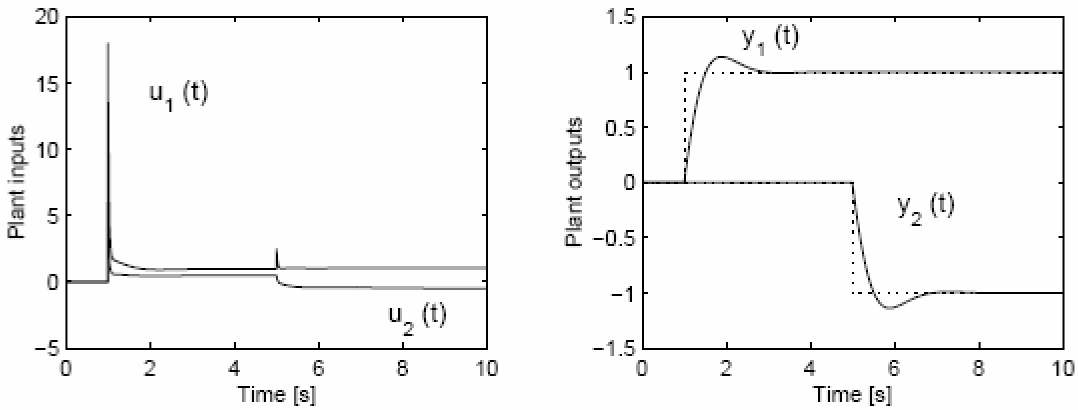

Figure 21.4. Closed loop step response

A step reference has been applied to each channel. Firstly, a positive unit step is applied as reference in channel 1, at time $t = 1$*, then a negative unit step is applied as reference in channel 2, at time* $t = 5$.

21.6.2 *When* $\beta = 2$*, the system has a NMP zero, which can be computed with the MATLAB command* **zero**. *This is a distributed zero, located at* $s = 49.55$. *Thus, if we were to compute* $\mathbf{P}(s)$ *for the same* $\mathbf{H}(s)$ *as in the previous case, we would obtain an unstable* $\mathbf{P}(s)$ *and a pole zero cancellation would arise. We thus modify our choice so that the NMP zero is preserved in* $\mathbf{H}(s)$. *The* *new choice of* $\mathbf{H}(s)$ *is*

$$\mathbf{H}(s) = \begin{bmatrix} \dfrac{-s+49.55}{(s+1)^2} & 0 \\ 0 & \dfrac{-s+49.55}{(s+1)^2} \end{bmatrix} \tag{21.21.50}$$

Note that the NMP zero must be included in **both** *diagonal entries in* $\mathbf{H}(s)$*, since otherwise, an unstable* $\mathbf{P}(s)$ *will result.*

We next choose the PI controllers to achieve a bandwidth of approximately 3 [rad/s]. This choice allows us to ignore the NMP zero in the computation of the controllers. We thus aim to have a complementary sensitivity (the same in both channels) approximately given by

$$T_{o1,2}(s) \approx \frac{4}{s^2+2s+4} \tag{21.21.51}$$

We then obtain

$$C(s) = \begin{bmatrix} 0.08\dfrac{(s+1)^2}{s(s+2)} & 0 \\ 0 & 0.08\dfrac{(s+1)^2}{s(s+2)} \end{bmatrix} \tag{21.21.52}$$

The performance of the control loop is evaluated via simulation with SIMULINK. The results are shown in Figure 21.5.

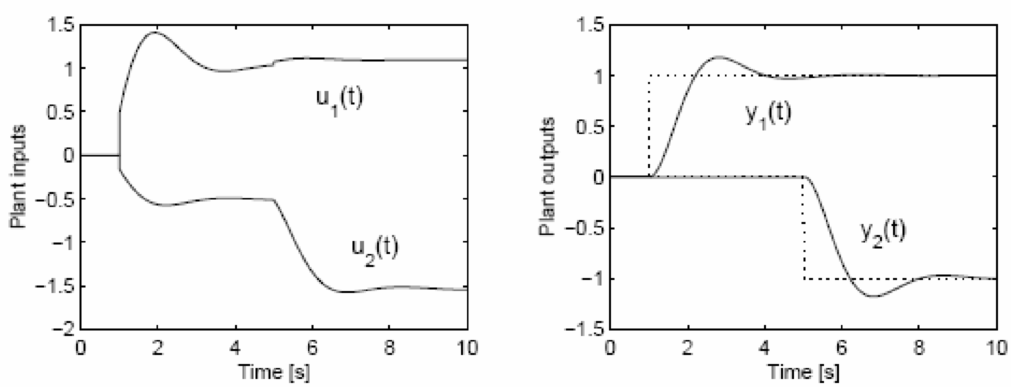

Figure 21.5. Closed loop step response

A step reference has been applied to each channel. Firstly, a positive unit step is applied as reference in channel 1, at time $t = 1$*, then a negative unit* *step is applied as reference in channel 2, at time $t = 5$. Note that the NMP behavior present in both channels is barely noticeable, since the NMP zero is much larger than the closed loop bandwidth (see section §8.6).*

Problem 21.7. *We observe that the system is stable. However it has a zero in the right half plane, at $s = 8$.*

21.7.1 *Assume that we want to use a precompensation matrix $\mathbf{M}(s)$ such that*

$$\mathbf{G_o}(s)\mathbf{M}(s) = \mathbf{H}(s) \tag{21.21.53}$$

where $\mathbf{H}(s)$ is an arbitrary stable diagonal transfer function matrix. Then

$$\mathbf{M}(s) = [\mathbf{G_o}(s)]^{-1}\mathbf{H}(s) \tag{21.21.54}$$

We then observe that, unless $\mathbf{H}(s)$ is carefully chosen, $\mathbf{M}(s)$ will become unstable and an unstable pole zero cancellation will arise. A second potential problem has to do with the possibility that $\mathbf{M}(s)$ turns out to be improper. Both problems will be systematically addressed in Chapter 25, using the idea of interactors. For the time being we will only tackle these issues in an intuitive way.

21.7.2 *To avoid the unstable pole zero cancellation, a plant factorization is carried out, to isolate the NMP zero. This is done as in (21.10.4). Note that $\mathbf{G_{o2}}(s)$ is stable and minimum phase (use the MATLAB command* **zero** *on $\mathbf{G_{o2}}(s)$ to verify that this is true). We next choose a precompensator $\mathbf{P}(s)$ to diagonalize $\mathbf{G_{o2}}(s)$, i.e.*

$$\mathbf{G_{o2}}(s)\mathbf{P}(s) = \mathbf{H}(s) \tag{21.21.55}$$

where $\mathbf{H}(s)$ is the diagonal matrix given by

$$\mathbf{H}(s) = \begin{bmatrix} \dfrac{1}{s+1} & 0 \\ 0 & \dfrac{1}{(s+1)^2} \end{bmatrix} \tag{21.21.56}$$

The matrix $\mathbf{P}(s)$ can be computed using the following MATLAB code

```
>> g11=tf([1 8],[1 3 2]);g12=tf([1 8],[1 6 5]);
>> g21=tf(1,[1 2 1]);g22=tf(6,[1 5 6]);
>> Go2=[g11 g12;g21 g22];
>> h11=tf(1,[1 1]);h12=tf(0,1);h21=h12;h22=tf(1,[1 2 1]);
>>H=[h11 h12;h21 h22];
```

```
>>detGo2=minreal(g11*g22-g12*g21);
>>p11=minreal(g22*h11/detGo2);p12=-minreal(g12*h22/detGo2);
>>p21=-minreal(g21*h11/detGo2);p22=minreal(g11*h22/detGo2);
>>P=[p11 p12;p21 p22]
```

The reader might like to verify that if the relative degree of $h_{22}(s)$ is chosen equal to 1, then the precompensator turns out to be improper.

21.7.3 *To design the controller we consider the precompensated plant $\mathbf{G_p}(s) = \mathbf{G_{o1}}(s)\mathbf{G_{o2}}(s)\mathbf{P}(s) = \mathbf{G_{o1}}(s)\mathbf{H}(s)$, given by*

$$\mathbf{G_p}(s) = \begin{bmatrix} \frac{-s+8}{(s+8)(s+1)} & 0 \\ 0 & \frac{1}{(s+1)^2} \end{bmatrix} \tag{21.21.57}$$

Based on the arguments presented in Chapter 8, in order that large undershoot is avoided, me must choose a bandwidth much smaller than 8 in both channels. We thus choose these bandwidths approximately equal to 2; this can be achieved with a PI controller in channel 1 and a PID controller in channel 2,

$$\mathbf{C}(s) = \begin{bmatrix} \frac{2s+2}{s} & 0 \\ 0 & \frac{4(s+1)^2}{s(s+3)} \end{bmatrix} \tag{21.21.58}$$

Note that for the design we have ignored the all pass factor $(-s+8)/(s+8)$. The performance of the loop is evaluated using SIMULINK. The results are shown in 21.6. A step reference has been applied to each channel. Firstly, a positive unit step is applied as reference in channel 1, at time $t = 1$, then a negative unit step is applied as reference in channel 2, at time $t = 5$. The NMP behavior shows, as expected, only in channel 1.

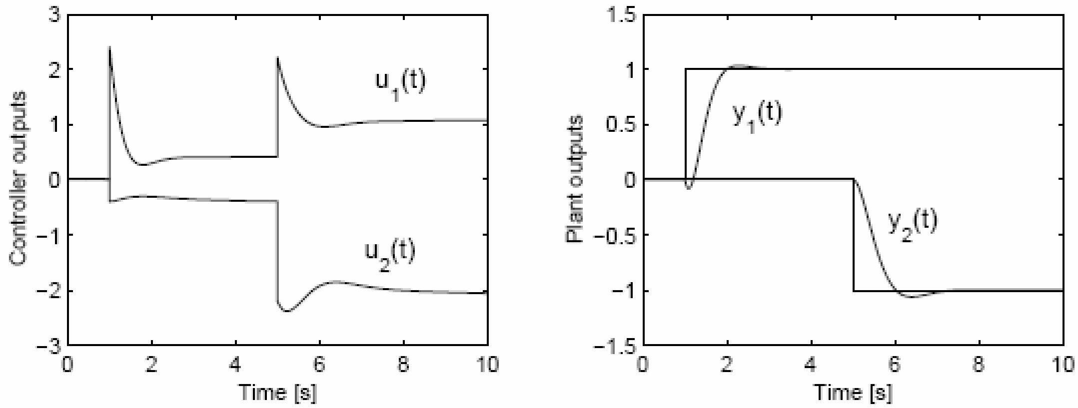

Figure 21.6. Closed loop step response