Chapter 12 Quasi-H_∞ Decoupling Control

Quasi-H_∞ Decoupling Control

12.1 Diagonal Factorization for Quasi-H_∞ Control

Procedure of the classical method for decoupling control:

1. Design a decoupler so that the MIMO plant is decomposed into a series of independent SISO plants
2. A SISO method is used to design controllers for these SISO plants

Some unsolved problems in the method:

- It is not applicable to NMP plants and plants with RHP poles
- It is difficult to analyze the effect of the decoupler on the closed-loop performance and robustness

Procedure of the design method in this chapter:

1. It is not applicable to NMP plants and plants with RHP poles
2. It is difficult to analyze the effect of the decoupler on the closed-loop performance and robustness

12.1 Diagonal Factorization for Quasi-H_∞ Control

Procedure of the classical method for decoupling control:

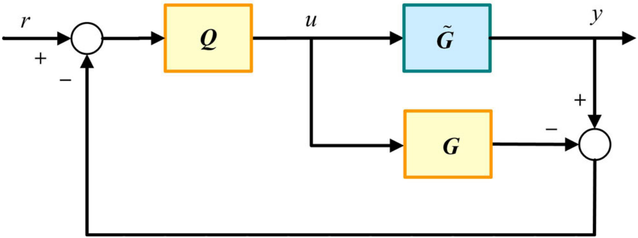
1. Design a decoupler so that the MIMO plant is decomposed into a series of independent SISO plants
2. A SISO method is used to design controllers for these SISO plants

Some unsolved problems in the method:

- It is not applicable to NMP plants and plants with RHP poles
- It is difficult to analyze the effect of the decoupler on the closed-loop performance and robustness

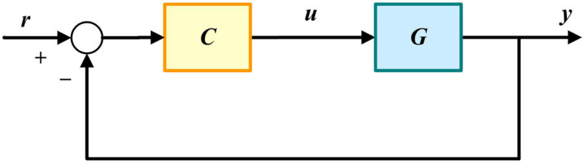
Procedure of the design method in this chapter:

1. It is not applicable to NMP plants and plants with RHP poles
2. It is difficult to analyze the effect of the decoupler on the closed-loop performance and robustness



Consider the IMC structure shown in Figure, where $\tilde{\mathbf{G}}(s)$ is an $n \times n$ plant, $\mathbf{G}(s)$ is the model, and $\mathbf{Q}(s)$ is an $n \times n$ controller. The closed-loop transfer function matrix is $\mathbf{T}(s) = \mathbf{G}(s)\mathbf{Q}(s)$

In quasi-H$_\infty$ decoupling control, the plant can be proper, have time delays, or have poles or zeros on the imaginary axis or in the open RHP



If the plant has poles in the closed RHP, the controller has to be implemented in the unity feedback loop shown in Figure, where $\mathbf{C}(s) = \mathbf{Q}(s)[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]^{-1}$ is an $n \times n$ controller

As stated in Section 10.4, the following assumptions are made for the plant:

1. There is not any unstable hidden mode in $\mathbf{G}(s)$

For quasi-H_∞ control, it is further assumed that

2. $\mathbf{G}(s)$ is of full normal rank, that is, $\text{rank}[\mathbf{G}(s)] = n$

If the second assumption is not satisfied, the closed-loop transfer function matrix $\mathbf{T}(s) = \mathbf{G}(s)\mathbf{C}(s)[\mathbf{I} + \mathbf{G}(s)\mathbf{C}(s)]^{-1}$ must be **identically singular**. When $\mathbf{G}(s)$ is not of full normal rank, a slight perturbation can be introduced in the coefficients of the plant so that the second condition is satisfied

Unlike the classical decoupling control, in quasi-H_∞ control there is not an independent step for decoupler design. The decoupler and the controller are expressed in the form of one transfer function matrix and are designed in **one step**

The key of the design is to define a diagonal factorization, in which the closed RHP zero and the time delay are separated from the plant. In the next section it will be seen that the decoupled response is obtained on the basis of such a factorization

If the second assumption is not satisfied, the closed-loop transfer function matrix $\mathbf{T}(s) = \mathbf{G}(s)\mathbf{C}(s)[\mathbf{I} + \mathbf{G}(s)\mathbf{C}(s)]^{-1}$ must be **identically singular**. When $\mathbf{G}(s)$ is not of full normal rank, a slight perturbation can be introduced in the coefficients of the plant so that the second condition is satisfied

Unlike the classical decoupling control, in quasi-H_∞ control there is not an independent step for decoupler design. The decoupler and the controller are expressed in the form of one transfer function matrix and are designed in **one step**

The key of the design is to define a diagonal factorization, in which the closed RHP zero and the time delay are separated from the plant. In the next section it will be seen that the decoupled response is obtained on the basis of such a factorization

If the second assumption is not satisfied, the closed-loop transfer function matrix $\mathbf{T}(s) = \mathbf{G}(s)\mathbf{C}(s)[\mathbf{I} + \mathbf{G}(s)\mathbf{C}(s)]^{-1}$ must be **identically singular**. When $\mathbf{G}(s)$ is not of full normal rank, a slight perturbation can be introduced in the coefficients of the plant so that the second condition is satisfied

Unlike the classical decoupling control, in quasi-H_∞ control there is not an independent step for decoupler design. The decoupler and the controller are expressed in the form of one transfer function matrix and are designed in **one step**

The key of the design is to define a diagonal factorization, in which the closed RHP zero and the time delay are separated from the plant. In the next section it will be seen that the decoupled response is obtained on the basis of such a factorization

The factorization for the closed RHP zero and the time delay can be obtained easily for a SISO plant, but the extension to the MIMO case should be properly defined

Write the plant $\mathbf{G}(s)$ with multiple time delays in the form of

$$\mathbf{G}(s) = \begin{bmatrix} G_{11}(s)e^{-\theta_{11}s} & \cdots & G_{1n}(s)e^{-\theta_{1n}s} \\ \vdots & \ddots & \vdots \\ G_{n1}(s)e^{-\theta_{n1}s} & \cdots & G_{nn}(s)e^{-\theta_{nn}s} \end{bmatrix}$$

where $G_{ij}(s)(i, j = 1, 2, ..., n)$ are scalar rational transfer functions and $\theta_{ij} \geq 0$ are time delays. Let the inverse of the plant be

$$\mathbf{G}^{-1}(s) = \begin{bmatrix} G^{11}(s)e^{-\theta^{11}s} & \cdots & G^{1n}(s)e^{-\theta^{1n}s} \\ \vdots & \ddots & \vdots \\ G^{n1}(s)e^{-\theta^{n1}s} & \cdots & G^{nn}(s)e^{-\theta^{nn}s} \end{bmatrix}$$

where $G^{ji}(s)e^{-\theta^{ji}s}(j, i = 1, 2, ..., n)$ are the elements of $\mathbf{G}^{-1}(s)$, and θ^{ji} are real numbers denoting the maximum time delays that can be separated from each element. For example, in the following element

$$\frac{e^{-3s}}{2s+1} + \frac{e^{-2s}}{3s+1} = \left(\frac{e^{-s}}{2s+1} + \frac{1}{3s+1}\right) e^{-2s}$$

the maximum time delays that can be separated is 2

Consider the factorization for the time delay first. Some elements of $\mathbf{G}^{-1}(s)$ may contain **predictions (that is, $\theta^{ji} < 0$)**. This implies that the resulting control system is physically irrealizable

To avoid this, these predictions have to be removed. This can be reached by postmultiplying $\mathbf{G}^{-1}(s)$ by a diagonal matrix $\mathbf{G_D}(s)$

Let the matrix without predictions be $\mathbf{G_O}^{-1}(s)$. We have $\mathbf{G_O}(s) = \mathbf{G_D^{-1}}(s)\mathbf{G}(s)$. $\mathbf{G_D}(s)$ should be chosen so that it counteracts those predictions and at the same time no additional time delays are introduced

Definition

Let $\theta_{li}(i = 1, 2, ..., n)$ be the largest prediction of the ith column of $\mathbf{G}^{-1}(s)$, that is, $\theta_{li} = \max_j \theta^{ji}, j = 1, 2, ..., n$. The H_∞ diagonal factorization for the time delay is defined as

$$\mathbf{G_D}(s) = \begin{bmatrix} e^{-\theta_{l1}s} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{-\theta_{ln}s} \end{bmatrix}$$

In particular, for rational plants $\mathbf{G_D}(s) = \mathbf{I}$.

Since the elements of $\mathbf{G_D}(s)$ are time delays, **no closed RHP zeros and poles are cancelled** when forming $\mathbf{G_O}(s)$

Now consider the factorization for closed RHP zeros. The plant may have closed RHP zeros in addition to the time delays. This implies that $\mathbf{G_O}^{-1}(s)$ is unstable. Then, an internally unstable system will be obtained

To make $\mathbf{G_O}^{-1}(s)$ stable, the unstable poles in each element must be removed. This can be reached by postmultiplying $\mathbf{G_O}^{-1}(s)$ by a diagonal matrix $\mathbf{G_N}(s)$

Let the obtained matrix be $\mathbf{G_{MP}}^{-1}(s)$. We readily obtain that $\mathbf{G_{MP}}(s) = \mathbf{G_N}^{-1}(s)\mathbf{G_O}(s)$. According to the definition in Section 10.1, **$\mathbf{G_{MP}}(s)$ is MP**

Since the elements of $\mathbf{G_D}(s)$ are time delays, **no closed RHP zeros and poles are cancelled** when forming $\mathbf{G_O}(s)$

Now consider the factorization for closed RHP zeros. The plant may have closed RHP zeros in addition to the time delays. This implies that $\mathbf{G_O}^{-1}(s)$ is unstable. Then, an internally unstable system will be obtained

To make $\mathbf{G_O}^{-1}(s)$ stable, the unstable poles in each element must be removed. This can be reached by postmultiplying $\mathbf{G_O}^{-1}(s)$ by a diagonal matrix $\mathbf{G_N}(s)$

Let the obtained matrix be $\mathbf{G_{MP}}^{-1}(s)$. We readily obtain that $\mathbf{G_{MP}}(s) = \mathbf{G_N}^{-1}(s)\mathbf{G_O}(s)$. According to the definition in Section 10.1, **$\mathbf{G_{MP}}(s)$ is MP**

Assume that $z_j(\mathrm{Re}(z_j) \geq 0, j = 1, 2, ..., r_z)$ are unstable poles of $\mathbf{G_O}^{-1}(s)$

Definition

Let $k_{ij}(i = 1, 2, ..., n)$ be the largest multiplicity of the unstable pole $z_j(j = 1, 2, ..., r_z)$ in the ith column of $\mathbf{G_O}^{-1}(s)$. The H_∞ diagonal factorization for closed RHP zeros is

$$\mathbf{G_N}(s) = \begin{bmatrix} \prod_{j=1}^{r_z} (-s/z_j + 1)^{k_{1j}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \prod_{j=1}^{r_z} (-s/z_j + 1)^{k_{nj}} \end{bmatrix}$$

In particular, for MP plants $\mathbf{G_N}(s) = \mathbf{I}$.

The factorization of $\mathbf{G}(s)$ for both the closed RHP zero and the time delay can be written as follows:

$$\mathbf{G}(s) = \mathbf{G_D}(s)\mathbf{G_N}(s)\mathbf{G_{MP}}(s)$$

In the factorization, $\mathbf{G_D}(s)$ denotes the time delay part of the plant, $\mathbf{G_N}(s)$ is related to the closed RHP zeros of the plant, and $\mathbf{G_{MP}}(s)$ is the MP part of the plant

$\mathbf{G_O}(s) = \mathbf{G_N}(s)\mathbf{G_{MP}}(s)$ is the "rational" part of the plant. It should be emphasized that in MIMO systems $\mathbf{G_O}(s)$ is generally not rational. The element of $\mathbf{G_O}(s)$ may have time delays

When the element of the plant involves time delays, it may have infinite RHP zeros, that is, $r_z \to \infty$ or $k_{ij} \to \infty$. The following example illustrates such a case

Example

Consider the plant described by the following transfer function matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 1 \\ 1 & 2e^{-s} \end{bmatrix}$$

$\mathbf{G}(s)$ has zeros at $s = \ln 2 + j2k\pi, k = 0, \pm 1, \pm 2, \ldots$. Following the factorization procedure in this section, we have

$$\begin{aligned} \mathbf{G_D}(s) &= \mathbf{I} \\ \mathbf{G_N}(s) &= (-2e^{-s} + 1)\mathbf{I} \\ \mathbf{G_{MP}}(s) &= \frac{1}{-2e^{-s} + 1} \begin{bmatrix} 1 & 1 \\ 1 & 2e^{-s} \end{bmatrix} \end{aligned}$$

The rational part of the plant is $\mathbf{G_O}(s) = \mathbf{G_N}(s)\mathbf{G_{MP}}(s)$. The element of $\mathbf{G_O}(s)$ has time delays.

Problem: When the numerator or denominator of the element in $\mathbf{G_{MP}}(s)$ involves multiple time delays, the design problem becomes complicated

Solution: In this case, it is recommended to use rational approximations to reduce the order of $\mathbf{G_O}(s)$ or $\mathbf{G_{MP}}(s)$. Rational approximations make the design simpler and easier without loss of too much precision

Tradeoff: The higher the order of the approximated numerator or the denominator, the better the precision. The designer has to trade off between the complexity and the precision

12.2 Quasi-H_∞ Controller Design

The design of MIMO control systems is similar to that of SISO control systems:

1. A desired closed-loop transfer function matrix $\mathbf{T}(s)$ is constructed based on the factorization developed in the last section
2. Utilizing the $\mathbf{T}(s)$, the controller $\mathbf{Q}(s)$ is then derived

Feature of the design: This is a "no-weight" design. The designer is not required to select weighting functions in the procedure

Assume that the controller is designed for step inputs. Factorize the plant: $\mathbf{G}(s) = \mathbf{G}_\mathrm{D}(s)\mathbf{G}_\mathrm{N}(s)\mathbf{G}_\mathrm{MP}(s)$

12.2 Quasi-H_∞ Controller Design

The design of MIMO control systems is similar to that of SISO control systems:

1. A desired closed-loop transfer function matrix $\mathbf{T}(s)$ is constructed based on the factorization developed in the last section
2. Utilizing the $\mathbf{T}(s)$, the controller $\mathbf{Q}(s)$ is then derived

Feature of the design: This is a "no-weight" design. The designer is not required to select weighting functions in the procedure

Assume that the controller is designed for step inputs. Factorize the plant: $\mathbf{G}(s) = \mathbf{G_D}(s)\mathbf{G_N}(s)\mathbf{G_{MP}}(s)$

If the plant is stable, the desired closed-loop transfer function matrix can be chosen as

$$\mathbf{T}(s) = \mathbf{T_{opt}}(s)\mathbf{J}(s)$$

where

$$\begin{aligned} \mathbf{T_{opt}}(s) &= \mathbf{G_D}(s)\mathbf{G_N}(s) \\ \mathbf{J}(s) &= \begin{bmatrix} J_1(s) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_n(s) \end{bmatrix} \end{aligned}$$

and

$$J_i(s) = \frac{1}{(\lambda_i s + 1)^{n_i}}, i = 1, 2, ..., n$$

Here $\lambda_i (i = 1, 2, ..., n)$ are the performance degrees

Let $\mathbf{T}_{\mathbf{opt}}(s) = \mathbf{G}(s)\mathbf{Q}_{\mathbf{opt}}(s)$. Then

$$\mathbf{Q}_{\mathbf{opt}}(s) = \mathbf{G}^{-1}(s)\mathbf{T}_{\mathbf{opt}}(s) = \mathbf{G}_{\mathbf{MP}}{}^{-1}(s)$$

Relative degree: The degree of a transfer function's denominator polynomial-the degree of its numerator polynomial

Denote the largest relative degree of all the elements in the ith column of $\mathbf{Q}_{\mathbf{opt}}(s)$ as α_i. Then
$n_i = -\alpha_i$ for strictly proper columns (that is, at least one element of the column is strictly proper)
$n_i = 1$ for semi-proper columns (that is, every element of the column is semi-proper)

Utilizing the desired closed-loop transfer function matrix $\mathbf{T}(s)$ and the factorization of $\mathbf{G}(s)$, the controller is obtained as follows:

$$\mathbf{Q}(s) = \mathbf{G}^{-1}(s)\mathbf{T}(s) = \mathbf{G_{MP}}^{-1}(s)\mathbf{J}(s)$$

The controller can be implemented in the IMC structure, or in the unity feedback loop:

$$\mathbf{C}(s) = \mathbf{Q}(s)[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]^{-1}$$

When the plant is unstable, the desired closed-loop transfer function matrix can be chosen as

$$\mathbf{T}(s) = \mathbf{T_{opt}}(s)\mathbf{J}(s)$$

where $\mathbf{T_{opt}}(s)$ and $\mathbf{J}(s)$ keep the same form as that for stable plants, but

$$J_i(s) = \frac{N_{xi}(s)}{(\lambda_i s + 1)^{n_i}}, i = 1, 2, ..., n$$

Here $N_{xi}(s)(i = 1, 2, ..., n)$ are polynomials with all roots in the LHP, and $N_{xi}(0) = 1$. $n_i = \deg\{N_{xi}(s)\} + \alpha_i$ for strictly proper columns and $n_i = \deg\{N_{xi}(s)\} + 1$ for semi-proper columns

Assume that $\mathbf{G}(s)$ has r_p unstable poles; The multiplicity of the unstable pole p_j ($\mathrm{Re}(p_j) \geq 0, j = 1, 2, ..., r_p$) is l_j; l_{ij} is the largest multiplicity of p_j in the ith row of $\mathbf{G}(s)$; The ith elements of $\mathbf{G_D}(s)$ and $\mathbf{G_N}(s)$ are $G_{Di}(s)$ and $G_{Ni}(s)$, respectively. Then $N_{xi}(s)$ is determined by

$$\lim_{s \to p_j} \frac{d^k}{ds^k}[I - G_{Di}(s)G_{Ni}(s)J_i(s)] = 0$$

$$i = 1, 2, ..., n; j = 1, 2, ..., r_p; k = 0, 1, ..., l_{ij} - 1$$

with $\deg\{N_{xi}(s)\} = \sum_{j=1}^{r_p} l_{ij}$

For unstable plants, the controller must be implemented in the unity feedback loop:

$$\mathbf{C}(s) = \mathbf{Q}(s)[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]^{-1}$$

Furthermore, it is required that all the RHP zero-pole cancellations in $[I - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ are removed. Recall the discussion in Section 8.1. This can be achieved by employing rational approximations

It should be pointed out that the following condition

$$\lim_{s \to p_j} \frac{d^k}{ds^k} \det[I - \mathbf{G}(s)\mathbf{Q}(s)] = 0, j = 1, 2, ..., r_p; k = 0, 1, ..., l_{ij} - 1$$

is not sufficient for internal stability with $\deg\{N_{xi}(s)\} = \sum_{j=1}^{r_p} l_{ij}$

For unstable plants, the controller must be implemented in the unity feedback loop:

$$\mathbf{C}(s) = \mathbf{Q}(s)[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]^{-1}$$

Furthermore, it is required that all the RHP zero-pole cancellations in $[I - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ are removed. Recall the discussion in Section 8.1. This can be achieved by employing rational approximations

It should be pointed out that the following condition

$$\lim_{s \to p_j} \frac{d^k}{ds^k} \det[I - \mathbf{G}(s)\mathbf{Q}(s)] = 0, j = 1, 2, ..., r_p; k = 0, 1, ..., l_{ij} - 1$$

is not sufficient for internal stability

By comparing the aforementioned two conditions, it can be found that more closed RHP zeros have to be introduced in $I - \mathbf{G}(s)\mathbf{Q}(s)$ to guarantee the internal stability. This is the price for decoupling

Example

Consider the plant described by the following transfer function matrix:

$$\mathbf{G}(s) = \begin{bmatrix} \frac{1}{s+3} & \frac{1}{s-2} \\ \frac{2}{s+3} & \frac{s-1}{s-2} \end{bmatrix}$$

The plant is NMP. It has one pole at $s = -3$, one pole at $s = 2$, and one zero at $s = 3$. Assume that the controller is

$$\mathbf{Q}(s) = \begin{bmatrix} \frac{-(s+3)(s-1)}{3(s+1)^2} & \frac{(s+3)(13s+1)}{3(s+1)^2} \\ \frac{2(s-2)}{3(s+1)^2} & \frac{-(s-2)(13s+1)}{3(s+1)^2} \end{bmatrix}$$

Example (ctd.1)

We have

$$\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s) = \begin{bmatrix} \frac{s(s+7/3)}{(s+1)^2} & 0 \\ 0 & \frac{16s(s-2)}{3(s+1)^2} \end{bmatrix}$$

It has a zero at $s = 2$. No more zeros are introduced by the controller. Since

$$[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s) = \begin{bmatrix} \frac{s(s+7/3)}{(s+3)(s+1)^2} & \frac{s(s+7/3)}{(s-2)(s+1)^2} \\ \frac{32s(s-2)}{3(s+3)(s+1)^2} & \frac{16s(s-1)}{3(s+1)^2} \end{bmatrix}$$

$[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ is not stable. Hence, the closed-loop system is not internally stable.

Example (ctd.2)

Now consider another controller described by

$$\mathbf{Q}(s) = \begin{bmatrix} \frac{-(s+3)(s-1)(40s+1)}{3(s+1)^3} & \frac{(s+3)(13s+1)}{3(s+1)^2} \\ \frac{2(s-2)(40s+1)}{3(s+1)^3} & \frac{-(s-2)(13s+1)}{3(s+1)^2} \end{bmatrix}$$

One readily obtains

$$\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s) = \begin{bmatrix} \frac{s(s-2)(s+55/3)}{(s+1)^3} & 0 \\ 0 & \frac{16s(s-2)}{3(s+1)^2} \end{bmatrix}$$

It can be seen that one more zero is introduced at $s = 2$ by the controller.

Example (ctd.3)

Since

$$[\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)=\begin{bmatrix} \frac{s(s-2)(s+55/3)}{(s+3)(s+1)^3} & \frac{s(s+55/3)}{(s+1)^3} \\ \frac{32s(s-2)}{3(s+3)(s+1)^2} & \frac{16s}{3(s+1)^2} \end{bmatrix}$$

$[\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ is stable.

Question: When more RHP zeros have to be introduced for internal stability

Answer: This has to be done if the closed RHP poles and their multiplicities in the elements of at least one row of $\mathbf{G}(s)$ are not the same

Example

Consider the following plant:

$$\mathbf{G}(s) = \begin{bmatrix} \frac{1}{s+3} & \frac{2}{s+3} \\ \frac{1}{s-2} & \frac{s-1}{s-2} \end{bmatrix}$$

The plant has one pole at $s = -3$, one pole at $s = 2$, and one zero at $s = 3$. The multiplicities of the closed RHP poles in every row are the same. If the following controller is taken:

$$\mathbf{Q}(s) = \begin{bmatrix} \frac{(s+3)(s-1)}{-3(s+1)^2} & \frac{2(s-2)(13s+1)}{3(s+1)^2} \\ \frac{s+3}{3(s+1)^2} & -\frac{(s-2)(13s+1)}{3(s+1)^2} \end{bmatrix}$$

the closed-loop response is decoupled.

Example

ctd.1 It is evident that both

$$\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)=\begin{bmatrix} \frac{s(s+7/3)}{(s+1)^2} & 0 \\ 0 & \frac{16s(s-2)}{3(s+1)^2} \end{bmatrix}$$

and

$$[\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)=\begin{bmatrix} \frac{s(s+7/3)}{(s+3)(s+1)^2} & \frac{2s(s+7/3)}{(s+3)(s+1)^2} \\ \frac{16s}{3(s+1)^2} & \frac{16s(s-1)}{3(s+1)^2} \end{bmatrix}$$

are stable. There is only one closed RHP zero in $\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)$.

The quasi-H_∞ controller can also be designed with the following procedure:

1. If the plant does not contain time delays (that is, $\mathbf{G_A}(s) = \mathbf{I}$), turn to 3.
2. If the plant contains time delays, take the rational part $\mathbf{G_O}(s)$ as the nominal plant.
3. If $\mathbf{G_O}(s)$ does not have zeros in the closed RHP (that is, $\mathbf{G_N}(s) = \mathbf{I}$), take its inverse as $\mathbf{Q_{opt}}(s)$ and turn to 5.
4. If $\mathbf{G_O}(s)$ has zeros in the closed RHP, remove the factor that contains the zeros (that is, $\mathbf{G_N}(s)$) and take the inverse of the reminder as $\mathbf{Q_{opt}}(s)$.
5. Introduce a filter to $\mathbf{Q_{opt}}(s)$, compute $\mathbf{C}(s)$ and remove the RHP zero-pole cancellation in $\mathbf{C}(s)$.

12.3 Analysis for Quasi-H_∞ Control Systems

Nominal stability: According to the discussion in Section 10.4, the closed-loop system is internally stable if and only if all the elements in the following matrix are stable:

$$\mathbf{H}(s) = \begin{bmatrix} \mathbf{G}(s)\mathbf{Q}(s) & [\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s) \\ \mathbf{Q}(s) & -\mathbf{Q}(s)\mathbf{G}(s) \end{bmatrix}$$

The following theorems provide sufficient and necessary conditions for the internal stability of the system designed in the last section

Theorem

The unity feedback control system is internally stable for the plant with time delays if and only if

1. $\mathbf{Q}(s)$ *is stable*
2. $[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ *is stable*

Proof.

Necessity is obvious. Consider sufficiency.
Assume that $[\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ is stable but $\mathbf{G}(s)\mathbf{Q}(s)$ is not stable. AS $\mathbf{Q}(s)$ is stable, the unstable pole of $\mathbf{G}(s)\mathbf{Q}(s)$ should be at the unstable poles of $\mathbf{G}(s)$. This implies that $\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)$ is unbounded at the unstable poles of $\mathbf{G}(s)$, which contradicts the assumption. Hence, if $[\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ is stable, $\mathbf{G}(s)\mathbf{Q}(s)$ is stable.
Since

$$[\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)=\mathbf{G}(s)[\mathbf{I}-\mathbf{Q}(s)\mathbf{G}(s)]$$

if $[\mathbf{I}-\mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ is stable, $\mathbf{Q}(s)\mathbf{G}(s)$ is also stable. $\square$

Theorem

Assume that

1. $\lim_{s \to p_j} \frac{d^k}{ds^k}[I - G_{Di}(s)G_{Ni}(s)J_i(s)] = 0, i = 1, 2, ..., n; j = 1, 2, ..., r_p; k = 0, 1, ..., l_{ij} - 1$
2. *All the RHP zero-pole cancellations in* $[I - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ *are removed*

Then, the unity feedback control system with a quasi-H_∞ *controller is internally stable.*

Proof.

It is sufficient to prove that the two conditions in the last Theorem hold.

It is evident that $\mathbf{Q}(s) = \mathbf{G_{MP}}^{-1}(s)\mathbf{J}(s)$ is stable, because $\mathbf{G_{MP}}(s)$ is MP and $\mathbf{J}(s)$ is stable.

Proof ctd.1.

Now consider the stability of $[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$. Since

$$\lim_{s \to p_j} \frac{d^k}{ds^k}[I - G_{Di}(s)G_{Ni}(s)J_i(s)] = 0,$$

$$i = 1, 2, ..., n; j = 1, 2, ..., r_p; k = 0, 1, ..., l_{ij} - 1,$$

all the unstable poles of $\mathbf{G}(s)$ are cancelled by $\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)$ when all the RHP zero-pole cancellations in $[I - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ are removed. Therefore, $[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ must be stable. □

Nominal performance: It is required that the system should have a zero steady-state error. The quasi-H_∞ controller is designed for step inputs. In view of the discussion in Section 10.4, the system should be of Type 1 for asymptotically tracking step inputs

Proof ctd.1.

Now consider the stability of $[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$. Since

$$\lim_{s \to p_j} \frac{d^k}{ds^k}[I - G_{Di}(s)G_{Ni}(s)J_i(s)] = 0,$$

$$i = 1, 2, ..., n; j = 1, 2, ..., r_p; k = 0, 1, ..., l_{ij} - 1,$$

all the unstable poles of $\mathbf{G}(s)$ are cancelled by $\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)$ when all the RHP zero-pole cancellations in $[I - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ are removed. Therefore, $[\mathbf{I} - \mathbf{G}(s)\mathbf{Q}(s)]\mathbf{G}(s)$ must be stable. □

Nominal performance: It is required that the system should have a zero steady-state error. The quasi-H_∞ controller is designed for step inputs. In view of the discussion in Section 10.4, the system should be of Type 1 for asymptotically tracking step inputs

In other words, the closed-loop transfer function matrix should satisfy the following condition:

$$\lim_{s\to 0} \mathbf{T}(s) = \mathbf{I}$$

Evidently, the quasi-H_∞ controller satisfies the condition

Robustness: The robustness of the closed-loop system can be tested by rigorous criteria introduced in Chapter 10.
As an alternative, the engineering tuning method for SISO systems can be directly extended to MIMO systems; that is, **increase the performance degrees monotonically until the required response is obtained**

The advantage of this tuning procedure is that it is quantitative and very simple. It can be used for the tuning of both nominal performance and robustness.

In other words, the closed-loop transfer function matrix should satisfy the following condition:

$$\lim_{s \to 0} \mathbf{T}(s) = \mathbf{I}$$

Evidently, the quasi-H_∞ controller satisfies the condition

Robustness: The robustness of the closed-loop system can be tested by rigorous criteria introduced in Chapter 10.
As an alternative, the engineering tuning method for SISO systems can be directly extended to MIMO systems; that is, **increase the performance degrees monotonically until the required response is obtained**

The advantage of this tuning procedure is that it is quantitative and very simple. It can be used for the tuning of both nominal performance and robustness.

One may find that some time delays or closed RHP zeros influence other channels while some others not. To distinguish them, two definitions are given

Definition

A time delay is canonical if at least one element of $\mathbf{G}_\mathrm{D}(s)$ contains the time delay, provided that the greatest common time delay of all elements of $\mathbf{G}_\mathrm{D}(s)$ has been removed

Definition

A RHP zero is canonical if at least one element of $\mathbf{G}_\mathrm{N}(s)$ has the zero, provided that the greatest common factor of all elements of $\mathbf{G}_\mathrm{N}(s)$ has been removed

One may find that some time delays or closed RHP zeros influence other channels while some others not. To distinguish them, two definitions are given

Definition

A time delay is canonical if at least one element of $\mathbf{G_D}(s)$ contains the time delay, provided that the greatest common time delay of all elements of $\mathbf{G_D}(s)$ has been removed

Definition

A RHP zero is canonical if at least one element of $\mathbf{G_N}(s)$ has the zero, provided that the greatest common factor of all elements of $\mathbf{G_N}(s)$ has been removed

A canonical time delay or a canonical RHP zero will not spread its influence over all channels, whereas a non-canonical time delay or a non-canonical RHP zero will affect all channels

Example

Consider the plant with the following transfer function matrix:

$$\mathbf{G}(s) = \frac{1}{(s+3)(s-1)} \begin{bmatrix} s-2 & 2(s-2) \\ 1 & s-1 \end{bmatrix}$$

The plant has a zero at $s = 2$ and a zero at $s = 3$. Since

$$\mathbf{G_N}(s) = \begin{bmatrix} \frac{-(s-2)}{s+2} & 0 \\ 0 & 1 \end{bmatrix} \frac{-(s-3)}{s+3}$$

The zero at s=2 is canonical and the zero at s=3 is non-canonical

A canonical time delay or a canonical RHP zero will not spread its influence over all channels, whereas a non-canonical time delay or a non-canonical RHP zero will affect all channels

Example

Consider the plant with the following transfer function matrix:

$$\mathbf{G}(s) = \frac{1}{(s+3)(s-1)} \begin{bmatrix} s-2 & 2(s-2) \\ 1 & s-1 \end{bmatrix}$$

The plant has a zero at $s = 2$ and a zero at $s = 3$. Since

$$\mathbf{G_N}(s) = \begin{bmatrix} \frac{-(s-2)}{s+2} & 0 \\ 0 & 1 \end{bmatrix} \frac{-(s-3)}{s+3}$$

The zero at s=2 is canonical and the zero at s=3 is non-canonical

Example

This example illustrates the effect of the non-canonical zeros. Consider the plant in the forgoing example:

$$\mathbf{G}(s) = \begin{bmatrix} \frac{1}{s+3} & \frac{1}{s-2} \\ \frac{2}{s+3} & \frac{s-1}{s-2} \end{bmatrix}$$

If the controller is

$$\mathbf{Q}(s) = \begin{bmatrix} \frac{-(s+3)(s-1)(40s+1)}{3(s+1)^3} & \frac{(s+3)(13s+1)}{3(s+1)^2} \\ \frac{2(s-2)(40s+1)}{3(s+1)^3} & \frac{-(s-2)(13s+1)}{3(s+1)^2} \end{bmatrix}$$

the closed-loop transfer function matrix is

$$\mathbf{T}(s) = \begin{bmatrix} \frac{40s+1}{(s+1)^3} & 0 \\ 0 & \frac{13s+1}{(s+1)^2} \end{bmatrix} \left(-\frac{s}{3} + 1\right)$$

The zero at $s = 3$ affects all channels

12.4 Increase Time Delays for Improving Performance

Two classes of methods in literature for enhancing the closed-loop performance:

1. Develop control schemes that have a better ability for time delay compensation
2. Modify the plant to reduce the effect of NMP factors

The discussion in Section 12.2 falls into the first class, while the second class will be discussed in this section.

A fact in MIMO systems: Decreasing or increasing the time delay may result in improved performance

Usually, it is impossible to decrease the time delay in a plant, but in many cases it is possible to increase it. For example, the time delay can be increased by simply increasing the length of the pipe that connects the process units

This section considers the strategy enhancing performance by increasing the time delay

Let $\theta_{si} (i = 1, 2, ..., n)$ be the smallest time delay of the ith row of $\mathbf{G}(s)$, that is, $\theta_{si} = \min_j \theta_{ij}, j = 1, 2, ..., n$. Then

The minimum time necessary for any inputs to affect the output i is θ_{si}

Theorem

The element with the largest prediction in each column of $\mathbf{G}^{-1}(s)$ *is on the diagonal if and only if the rows and columns of* $\mathbf{G}(s)$ *are rearranged so that the smallest time delay of* $\mathbf{G}(s)$ *in each row is at the diagonal. If this is true, the largest prediction is* $e^{\theta_{si}}$

Proof.

Prove sufficiency first. Assume that $\mathbf{G}(s)$ has been rearranged so that the smallest time delay in each row is at the diagonal, that is, $\theta_{si} = \theta_{jj}, j = 1, 2, ..., n$. It is known that

$$\mathbf{G}^{-1}(s) = \frac{\mathrm{adj}[\mathbf{G}(s)]}{\det[\mathbf{G}(s)]}$$

Proof ctd.1.

The largest time delay that can be separated from det[$\mathbf{G}(s)$] is

$$\theta_{det} = \theta_{11} + \theta_{22} + \ldots + \theta_{nn}$$

Consider the jth column of adj[$\mathbf{G}(s)$]. Compare the maximum time delays that can be separated from the diagonal element (denoted by θ_{ajj}) and the non-diagonal element (denoted by $\theta_{aij}, i \neq j$):

$$\begin{aligned}\theta_{ajj} &= \theta_{11} + \ldots + \theta_{(i-1)(i-1)} + \theta_{ii} + \theta_{(i+1)(i+1)} + \\ &\quad \ldots + \theta_{(j-1)(j-1)} + \theta_{(j+1)(j+1)} + \ldots + \theta_{nn}. \\ \theta_{aij} &= \theta_{11} + \ldots + \theta_{(i-1)(i-1)} + \theta_{ij} + \theta_{(i+1)(i+1)} + \\ &\quad \ldots + \theta_{(j-1)(j-1)} + \theta_{(j+1)(j+1)} + \ldots + \theta_{nn}.\end{aligned}$$

Proof ctd.2.

Since

$$\theta_{jj} \le \theta_{ij}, \forall i, j = 1, 2, ..., n$$

the result is

$$\theta_{ajj} \le \theta_{aij}$$

Subtracting the two sides of the inequality from θ_{det} yields

$$\theta_{det} - \theta_{ajj} \ge \theta_{det} - \theta_{aij}$$

$\theta_{det} - \theta_{ajj} = \theta_{jj} = \theta_{si}$ is the prediction of the diagonal element in the jth column of $\mathbf{G}^{-1}(s)$, while $\theta_{det} - \theta_{aij}$ is the prediction of the non-diagonal element in the jth column of $\mathbf{G}^{-1}(s)$.
The sufficiency proof is reversible □

It may not be feasible to make the smallest time delay in each row at the diagonal by rearranging rows and columns of a plant

The idea here: Increase individual delays θ_{ij} by the minimum amount, so as to make the minimum time delays occur at the diagonal

Key: Formulate this problem mathematically

Let $b_{ij}(i, j = 1, 2, ..., n)$ be a continuous variable that represents the modified time delay. The binary variable y_{ij} is introduced, which takes values of $0 - 1$ and is associated with the element b_{ij}. When $y_{ij} = 1$, the corresponding element has the smallest time delay. **The minimum time delay necessary to improve the closed-loop response** is given by the solution of the following optimization problem:

It may not be feasible to make the smallest time delay in each row at the diagonal by rearranging rows and columns of a plant

The idea here: Increase individual delays θ_{ij} by the minimum amount, so as to make the minimum time delays occur at the diagonal

Key: Formulate this problem mathematically

Let $b_{ij}(i, j = 1, 2, ..., n)$ be a continuous variable that represents the modified time delay. The binary variable y_{ij} is introduced, which takes values of $0 - 1$ and is associated with the element b_{ij}. When $y_{ij} = 1$, the corresponding element has the smallest time delay. **The minimum time delay necessary to improve the closed-loop response** is given by the solution of the following optimization problem:

$$\mathrm{Min} \sum_{\mathrm{i}} \sum_{j} b_{ij}$$

subject to

$$\begin{aligned}
\sum_{i} y_{ij} &= 1, j = 1, 2, ..., n && (1)\\
\sum_{j} y_{ij} &= 1, i = 1, 2, ..., n && (2)\\
\sum_{j} (y_{ij} b_{ij}) - b_{ij} &\leq 0, i, j = 1, 2, ..., n && (3)\\
\theta_{ij} &\leq b_{ij}, i, j = 1, 2, ..., n && (4)\\
b_{ij} &\leq \max_{j} \theta_{ij}, i, j = 1, 2, ..., n && (5)\\
y_{ij} &= 0, 1, i, j = 1, 2, ..., n && (6)
\end{aligned}$$

Constraints (1) and (2) imply that only one element is picked up as the element with the smallest time delay

Constraint (3) states that the selected element indeed has the smallest time delay. The first term in its left-hand side picks up one element from each row, and compares it with the other elements in this row (that is, the second term in its left-hand side)

Constraints (4) and (5) provide lower and upper bounds for the continuous variable

This mathematical formulation is a mixed-integer nonlinear programming problem, since the constraint (3) involves the bilinearity of the form $y_{ij}b_{ij}$. The constraint (3) is not convex. This implies that it is hard to obtain the global optimal solution

Constraints (1) and (2) imply that only one element is picked up as the element with the smallest time delay

Constraint (3) states that the selected element indeed has the smallest time delay. The first term in its left-hand side picks up one element from each row, and compares it with the other elements in this row (that is, the second term in its left-hand side)

Constraints (4) and (5) provide lower and upper bounds for the continuous variable

This mathematical formulation is a mixed-integer nonlinear programming problem, since the constraint (3) involves the bilinearity of the form $y_{ij}b_{ij}$. The constraint (3) is not convex. This implies that it is hard to obtain the global optimal solution

The problem can be overcome by converting the nonlinearity programming into a linear programming. The bilinearity of the form $y_{ij}b_{ij}$, where y_{ij} is an integer and b_{ij} is a continuous variable, can be substituted by a continuous variable h_{ij} with

$$\begin{aligned}
h_{ij} &= y_{ij}b_{ij} \\
b_{ij} - U(1-y_{ij}) &\leq h_{ij} \\
Ly_{ij} &\geq h_{ij} \\
b_{ij} - L(1-y_{ij}) &\leq h_{ij} \\
Uy_{ij} &\geq h_{ij} \\
i, j = 1, 2, ..., n &
\end{aligned}$$

where the scalars L and U satisfy

$$L \leq b_{ij} \leq U, i, j = 1, 2, ..., n$$

In the above formulation, one can take $L = 0$ and $U = \max_{ij}(\theta_{ij})$

Utilizing the conversion, the formulation becomes

$$\text{Min} \sum_{\text{i}} \sum_{j} b_{ij}$$

subject to

$$\begin{aligned}
\sum_{i} y_{ij} &= 1, j = 1, 2, ..., n \\
\sum_{j} y_{ij} &= 1, i = 1, 2, ..., n \\
\sum_{j} h_{ij} - b_{ij} &\leq 0, i, j = 1, 2, ..., n
\end{aligned}$$

$$
\begin{aligned}
b_{ij} - (\max_{ij} \theta_{ij})(1 - y_{ij}) &\leq h_{ij}, i, j = 1, 2, ..., n \\
0 &\leq h_{ij}, i, j = 1, 2, ..., n \\
\theta_{ij} &\leq b_{ij}, i, j = 1, 2, ..., n \\
b_{ij} &\leq \max_{j} \theta_{ij}, i, j = 1, 2, ..., n \\
y_{ij} &= 0, 1, i, j = 1, 2, ..., n
\end{aligned}
$$

This is a mixed-integer linear programming problem. It can be solved with the help of computer softwares. The b_{ij} that corresponds to the y_{ij} with its value being 1 provides improved performance

The mathematical tool in this section is thoroughly different from those introduced in foregoing chapters. For a detailed discussion, please refer to monographs addressing this topic

12.5 A Design Example for Quasi-H_∞ Control

In this section, an example is given to illustrate the design procedure of the quasi-H_∞ controller. It is shown that the performance is significantly improved by employing the optimization technique introduced in the last section

Example

Consider the model of a pilot scale ethanol and water distillation column:

$$\boxed{\mathbf{G}(s) = \begin{bmatrix} \frac{0.66e^{-6s}}{6.7s+1} & \frac{-0.005e^{-s}}{9.1s+1} \\ \frac{-34.7e^{-9.2s}}{8.1s+1} & \frac{0.87(11.6s+1)e^{-s}}{(3.9s+1)(18.8s+1)} \end{bmatrix}}$$

Example (ctd.1)

In this system the outputs are

- Overhead ethanol mole fraction.
- Bottom composition temperature (°C).

The inputs are

- Reflux flow rate (gpm).
- Reboiler stream pressure (psig).

The inverse of the plant is

$$\mathbf{G}^{-1}(s) = \frac{\mathrm{adj}[\mathbf{G}(s)]}{\det[\mathbf{G}(s)]}$$

where

Example (ctd.2)

$$\mathrm{adj}[\mathbf{G}(s)] = \begin{bmatrix} \frac{0.87(11.6s+1)e^{-s}}{(3.9s+1)(18.8s+1)} & \frac{0.005e^{-s}}{9.1s+1} \\ \frac{34.7e^{-9.2s}}{8.1s+1} & \frac{0.66e^{-6s}}{6.7s+1} \end{bmatrix}$$

$$\det[\mathbf{G}(s)] = \left[\frac{0.66 * 0.87(11.6s+1)}{(6.7s+1)(3.9s+1)(18.8s+1)} - \frac{34.7 * 0.005e^{-3.2s}}{(8.1s+1)(9.1s+1)}\right]$$

$\mathbf{G}(s)$ is factorized into

$$\mathbf{G}(s) = \mathbf{G_D}(s)\mathbf{G_O}(s)$$

The maximum predictions of both the first and the second columns of $\mathbf{G}^{-1}(s)$ are 6. According to Definition 1, $\mathbf{G_D}(s)$ is given by

$$\mathbf{G_D}(s) = \begin{bmatrix} e^{-6s} & 0 \\ 0 & e^{-6s} \end{bmatrix}$$

Example (ctd.3)

$\mathbf{G_O}(s) = \mathbf{G_D}^{-1}(s)\mathbf{G}(s)$ does not have any RHP zeros, which implies that $\mathbf{G_N}(s) = \mathbf{I}$ and $\mathbf{G_{MP}}(s) = \mathbf{G_O}(s)$. In light of (1), the quasi-H_∞ controller is

$$\mathbf{Q_{opt}}(s) = \frac{\begin{bmatrix} \frac{0.87(11.6s+1)}{(3.9s+1)(18.8s+1)} & \frac{0.005}{9.1s+1} \\ \frac{34.7e^{-8.2s}}{8.1s+1} & \frac{0.66e^{-5s}}{6.7s+1} \end{bmatrix}}{\frac{0.66*0.87(11.6s+1)}{(6.7s+1)(3.9s+1)(18.8s+1)} - \frac{34.7*0.005e^{-3.2s}}{(8.1s+1)(9.1s+1)}}$$

This exact controller is too complex. Model reduction techniques are used here to simplify the design task

Suppose that the relative degree of the approximate denominator is chosen as 2. The following result can be obtained by using fitting techniques:

Example (ctd.4)

$$\frac{0.66 * 0.87(11.6s+1)}{(6.7s+1)(3.9s+1)(18.8s+1)} - \frac{34.7 * 0.005e^{-3.2s}}{(8.1s+1)(9.1s+1)} \approx \frac{0.4007}{3s^2+16s+1}$$

With the approximate denominator, we have

$$\mathbf{Q}_{\mathbf{opt}}(s) = \begin{bmatrix} \frac{0.87(11.6s+1)}{(3.9s+1)(18.8s+1)} & \frac{0.005}{9.1s+1} \\ \frac{34.7e^{-8.2s}}{8.1s+1} & \frac{0.66e^{-5s}}{6.7s+1} \end{bmatrix} \frac{3s^2+16s+1}{0.4007}$$

Since the plant is stable, the filter can be chosen as

$$\mathbf{J}(s) = \begin{bmatrix} \frac{1}{\lambda_1 s+1} & 0 \\ 0 & \frac{1}{\lambda_2 s+1} \end{bmatrix}$$

The suboptimal controller is

Example (ctd.5)

$$\mathbf{Q}(s) = \begin{bmatrix} \frac{0.87(11.6s+1)}{(3.9s+1)(18.8s+1)(\lambda_1 s+1)} & \frac{0.005}{(9.1s+1)(\lambda_2 s+1)} \\ \frac{34.7e^{-8.2s}}{(8.1s+1)(\lambda_1 s+1)} & \frac{0.66e^{-5s}}{(6.7s+1)(\lambda_2 s+1)} \end{bmatrix} \frac{3s^2+16s+1}{0.4007}$$

If there is not any requirement on the closed-loop response, λ_1 and λ_2 can be selected freely.

Suppose there are some design requirements on the closed-loop response, for example, 10% overshoot for each channel. Using the engineering tuning method, it is easy to obtain that $\lambda_1 = 3.2$ and $\lambda_2 = 4$. The closed-loop responses are shown in Figure.

In the last section, an optimization design method was introduced for improving the performance. Using the method, one gets

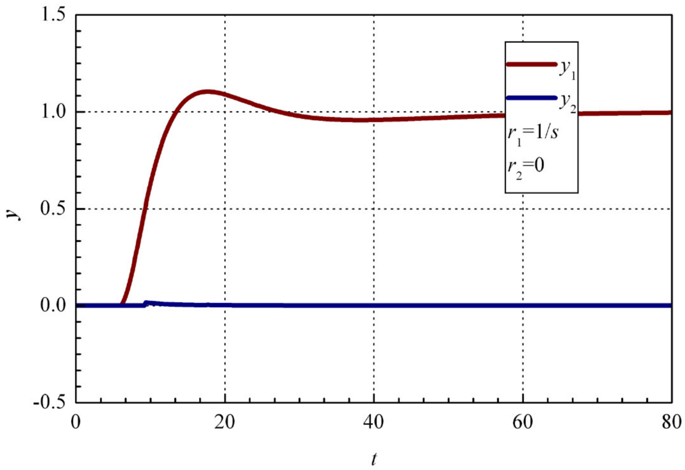


Figure: Closed-loop response with $\lambda_1 = 3.2$ and $\lambda_2 = 4$-1

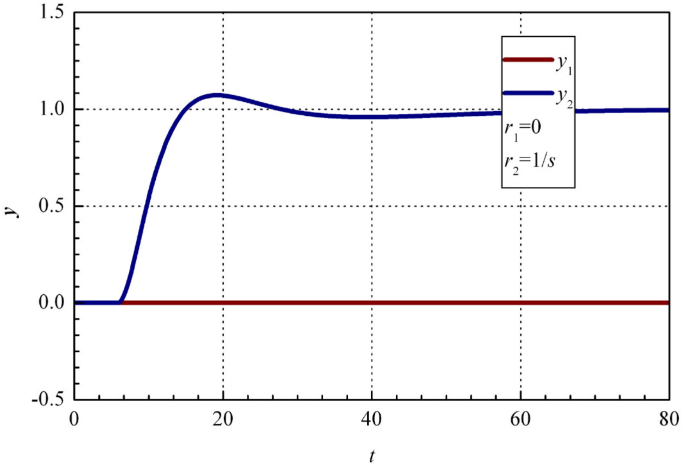


Figure: Closed-loop response with $\lambda_1 = 3.2$ and $\lambda_2 = 4\text{-}2$

Example (ctd.6)

$$\mathrm{Min} \sum_{\mathrm{i}} \sum_{j} b_{ij} = 22.2,$$

$$b_{11} = 6, b_{12} = 6, b_{21} = 9.2, b_{22} = 1,$$

$$y_{11} = 1, y_{12} = 0, y_{21} = 0, y_{22} = 1$$

This implies that the closed-loop performance could be improved by increasing θ_{12} in $G(s)$ to 6 min while keeping other time delays. In this case,

$$\mathbf{G_D}(s) = \begin{bmatrix} e^{-6s} & 0 \\ 0 & e^{-s} \end{bmatrix}$$

and the controller is

Example (ctd.7)

$$\mathbf{Q}(s) = \begin{bmatrix} \frac{0.87(11.6s+1)}{(3.9s+1)(18.8s+1)(\lambda_1 s+1)} & \frac{0.005}{(9.1s+1)(\lambda_2 s+1)} \\ \frac{34.7e^{-8.2s}}{(8.1s+1)(\lambda_1 s+1)} & \frac{0.66}{(6.7s+1)(\lambda_2 s+1)} \end{bmatrix} \frac{3s^2+16s+1}{0.4007}$$

For the sake of comparison, the same controller parameters as that in the system with unmodified time delays are taken. It can be seen from Figure that the response of the second output is significantly improved

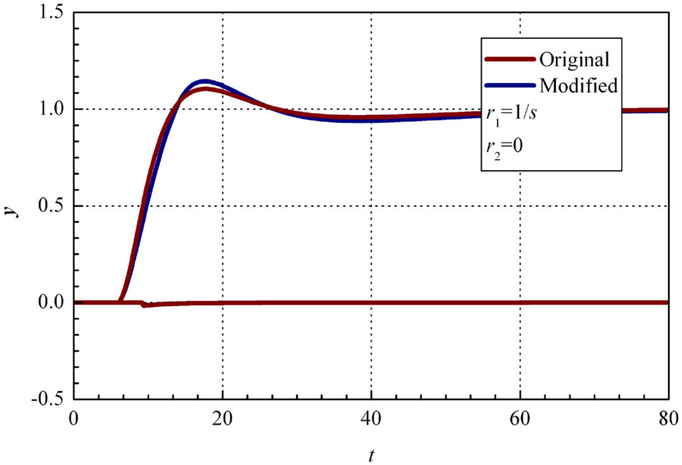


Figure: Performance improvement by increasing the time delay-1

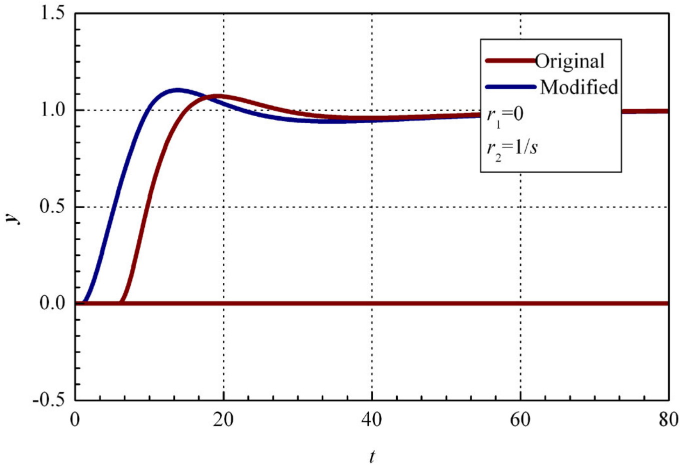


Figure: Performance improvement by increasing the time delay-2

12.6 Multivariable PID Controller Design

Open question: The PID controller is widely used in practice. Design techniques of SISO PID controllers have been well developed. However, the design of multivariable PID controllers remains a subject of study, since the MIMO case is much more intricate than the SISO case

Design idea in this section:
Section 12.2 introduced a simple design method for multivariable controllers, of which one main feature is that the controller is analytical

Once an analytical result is obtained, one can directly use the design procedures in Section 5.5 and Section 5.6 to reduce every element of the multivariable controller $\mathbf{C}(s)$ into a SISO PID controller

When the desired PID controllers have the same form (that is, all controllers are in the form of PI, or all controllers are in the form of PID), the design procedure can be expressed by matrix and vector notations. As an example, the design procedure for Maclaurin PID controllers is provided here

Consider the quasi-H_∞ control. First, the plant is factorized as

$$\mathbf{G}(s) = \mathbf{G}_D(s)\mathbf{G}_\mathbf{N}(s)\mathbf{G}_{MP}(s)$$

The desired closed-loop transfer function matrix is chosen as

$$\mathbf{T}(s) = \mathbf{G}_\mathbf{D}(s)\mathbf{G}_\mathbf{N}(s)\mathbf{J}(s)$$

With the closed-loop transfer function matrix, the obtained unity feedback loop controller can be expressed as

$$\mathbf{C}(s) = \mathbf{G}_\mathbf{MP}{}^{-1}(s)\mathbf{J}(s)[\mathbf{I} - \mathbf{G}_D(s)\mathbf{G}_\mathbf{N}(s)\mathbf{J}(s)]^{-1}$$

When the desired PID controllers have the same form (that is, all controllers are in the form of PI, or all controllers are in the form of PID), the design procedure can be expressed by matrix and vector notations. As an example, the design procedure for Maclaurin PID controllers is provided here

Consider the quasi-H_∞ control. First, the plant is factorized as

$$\mathbf{G}(s) = \mathbf{G}_D(s)\mathbf{G}_\mathbf{N}(s)\mathbf{G}_{MP}(s)$$

The desired closed-loop transfer function matrix is chosen as

$$\mathbf{T}(s) = \mathbf{G}_\mathbf{D}(s)\mathbf{G}_\mathbf{N}(s)\mathbf{J}(s)$$

With the closed-loop transfer function matrix, the obtained unity feedback loop controller can be expressed as

$$\mathbf{C}(s) = \mathbf{G}_\mathbf{MP}{}^{-1}(s)\mathbf{J}(s)[\mathbf{I} - \mathbf{G}_D(s)\mathbf{G}_\mathbf{N}(s)\mathbf{J}(s)]^{-1}$$

Rewrite the controller as

$$\mathbf{C}(s) = s^{-1}\mathbf{f}(s)$$

where

$$\mathbf{f}(s) = s\mathbf{G_{MP}}^{-1}(s)\mathbf{J}(s)[\mathbf{I} - \mathbf{G}_D(s)\mathbf{G_N}(s)\mathbf{J}(s)]^{-1}$$

Expand the controller in a Maclaurin series:

$$\mathbf{C}(s) = s^{-1}\left[\mathbf{f}(0) + \mathbf{f}'(0)s + \mathbf{f}''(0)s^2/2! + \mathbf{f}^{(3)}(0)s^3/3! + \ldots\right]$$

The first three terms form a standard PID controller:

$$\mathbf{C}(s) = \mathbf{K_C} + \mathbf{T_I}s^{-1} + \mathbf{T_D}s$$

whose parameters are

$$\mathbf{K_C} = \mathbf{f}'(0), \mathbf{T_I} = \mathbf{f}(0), \mathbf{T_D} = \mathbf{f}''(0)/2$$

To simplify the representation, let

$$\begin{aligned}\mathbf{N}(s) &= s^{-1}[\mathbf{I} - \mathbf{G}_D(s)\mathbf{G}_\mathbf{N}(s)\mathbf{J}(s)]\mathbf{J}^{-1}(s) \\ &= s^{-1}[\mathbf{J}^{-1}(s) - \mathbf{G}_D(s)\mathbf{G}_\mathbf{N}(s)]\end{aligned}$$

Then we have

$$\mathbf{f}(s) = \mathbf{G_{MP}}^{-1}(s)\mathbf{N}^{-1}(s)$$

The values of $\mathbf{f}(s)$ and its derivatives at the origin are

$$\begin{aligned}\mathbf{f}(0) &= \mathbf{G_{MP}}^{-1}(0)\mathbf{N}^{-1}(0), \\ \mathbf{f}'(0) &= [\mathbf{G_{MP}}^{-1}(0)]'\mathbf{N}^{-1}(0) + \mathbf{G}_{MP}^{-1}(0)[\mathbf{N}^{-1}(0)]' \\ \mathbf{f}''(0) &= [\mathbf{G_{MP}}^{-1}(0)]''\mathbf{N}^{-1}(0) + \\ &\quad 2[\mathbf{G_{MP}}^{-1}(0)]'[\mathbf{N}^{-1}(0)]' + \mathbf{G_{MP}}^{-1}(0)[\mathbf{N}^{-1}(0)]''\end{aligned}$$

where

$$
\begin{aligned}
\mathbf{N}(0) &= \left[\mathbf{J}^{-1}(0)\right]' - [\mathbf{G_N}^{-1}(0)\mathbf{G_D}^{-1}(0)]' \\
[\mathbf{N}(0)]' &= \{[\mathbf{J}^{-1}(0)]'' - [\mathbf{G_N}^{-1}(0)\mathbf{G_D}^{-1}(0)]''\}/2! \\
[\mathbf{N}(0)]'' &= \{[\mathbf{J}^{-1}(0)]^{(3)} - [\mathbf{G_N}^{-1}(0)\mathbf{G_D}^{-1}(0)]^{(3)}\}/3! \\
\left[\mathbf{N}^{-1}(0)\right]' &= -\mathbf{N}^{-1}(0)[\mathbf{N}(0)]'\mathbf{N}^{-1}(0) \\
\left[\mathbf{N}^{-1}(0)\right]'' &= -[\mathbf{N}^{-1}(0)]'[\mathbf{N}(0)]'\mathbf{N}^{-1}(0) - \\
&\quad \mathbf{N}^{-1}(0)[\mathbf{N}(0)]''\mathbf{N}^{-1}(0) - \\
&\quad \mathbf{N}^{-1}(0)[\mathbf{N}(0)]'[\mathbf{N}^{-1}(0)]' \\
\left[\mathbf{G}^{-1}(0)\right]' &= -\mathbf{G}_{MP}^{-1}(0)[\mathbf{G}_{MP}(0)]'\mathbf{G_{MP}}^{-1}(0) \\
\left[\mathbf{G}^{-1}(0)\right]'' &= -[\mathbf{G_{MP}}^{-1}(0)]'[\mathbf{G}_{MP}(0)]'\mathbf{G_{MP}}^{-1}(0) - \\
&\quad \mathbf{G_{MP}}^{-1}(0)[\mathbf{G}_{MP}(0)]''\mathbf{G_{MP}}^{-1}(0) - \\
&\quad \mathbf{G_{MP}}^{-1}(0)[\mathbf{G}_{MP}(0)]'[\mathbf{G_{MP}}^{-1}(0)]'
\end{aligned}
$$

Example

Consider a binary distillation column for separating a mix of methanol and water (the feed) into a bottom product (mostly water) and a methanol distillate (Figure). Schematically, the distillation process functions as follows:

- Steam flows into the reboiler and vaporizes the bottom liquid. This vapor is reinjected into the column and mixes with the feed
- Methanol, being more volatile than water, tends to concentrate in the vapor moving upward. Meanwhile, water tends to flow downward and accumulate as the bottom liquid
- The vapor exiting at the top of the column is condensed by a flow of cooling water. Part of this condensed vapor is extracted as the distillate, and the rest of the condensate (the reflux) is sent back to the column
- Part of the bottom liquid is collected as bottom products

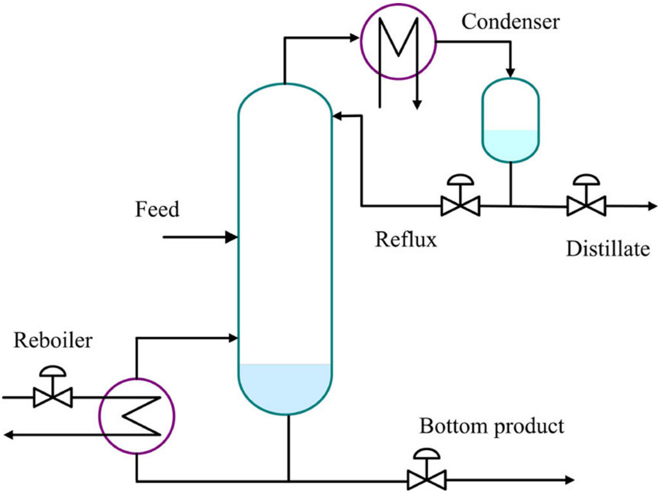


Figure: Binary distillation column

Example (ctd.1)

In this application, the objective is to control the amount of the bottom and top methanol by manipulating the steam flow rate and the reflux flow rate, respectively. Since a change in either steam flow rate or reflux flow rate upsets both methanols, we have an interacting system

The model of the distillation column is described by

$$\boxed{\mathbf{G}(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix}}$$

The inverse of the plant is

$$\mathbf{G}^{-1}(s) = \frac{\mathrm{adj}[\mathbf{G}(s)]}{\det[\mathbf{G}(s)]}$$

Example (ctd.2)

where

$$\mathrm{adj}[\mathbf{G}(s)] = \begin{bmatrix} \frac{-19.4e^{-3s}}{14.4s+1} & \frac{18.9e^{-3s}}{21s+1} \\ \frac{-6.6e^{-7s}}{10.9s+1} & \frac{12.8e^{-s}}{16.7s+1} \end{bmatrix}$$

$$\det[\mathbf{G}(s)] = \left(\frac{-248.3}{240.5s^2+31.1s+1} - \frac{-124.7e^{-6s}}{228.9s^2+31.9s+1}\right)e^{-4s}$$

$\mathbf{G}(s)$ is factorized into

$$\mathbf{G}(s) = \mathbf{G_D}(s)\mathbf{G_O}(s)$$

Since the largest predictions of the first and the second columns of $\mathbf{G}^{-1}(s)$ are 1 and 3 respectively, $\mathbf{G_D}(s)$ is given by

$$\mathbf{G_D}(s) = \begin{bmatrix} e^{-1s} & 0 \\ 0 & e^{-3s} \end{bmatrix}$$

Example (ctd.3)

$\mathbf{G_O}(s)$ is MP, $\mathbf{G_N}(s) = \mathbf{I}$. The desired closed-loop transfer function matrix is chosen as

$$\mathbf{T}(s) = \begin{bmatrix} \dfrac{e^{-s}}{\lambda_1 s + 1} & 0 \\ 0 & \dfrac{e^{-3s}}{\lambda_2 s + 1} \end{bmatrix}$$

Then

$$\mathbf{Q}(s) = \frac{\begin{bmatrix} \frac{-19.4}{(14.4s+1)(\lambda_1 s+1)} & \frac{18.9e^{-2s}}{(21s+1)(\lambda_2 s+1)} \\ \frac{-6.6e^{-4s}}{(10.9s+1)(\lambda_1 s+1)} & \frac{12.8}{(16.7s+1)(\lambda_2 s+1)} \end{bmatrix}}{\frac{124.7e^{-6s}}{228.9s^2+31.9s+1} - \frac{248.3}{240.5s^2+31.1s+1}}$$

Example (ctd.4)

and the unity feedback loop controller is

$$\mathbf{C}(s) = \frac{\begin{bmatrix} \frac{-19.4}{(14.4s+1)(\lambda_1 s+1-e^{-s})} & \frac{18.9e^{-2s}}{(21s+1)(\lambda_2 s+1-e^{-3s})} \\ \frac{-6.6e^{-4s}}{(10.9s+1)(\lambda_1 s+1-e^{-s})} & \frac{12.8}{(16.7s+1)(\lambda_2 s+1-e^{-3s})} \end{bmatrix}}{\frac{124.7e^{-6s}}{228.9s^2+31.9s+1} - \frac{248.3}{240.5s^2+31.1s+1}}$$

This is a rigorously analytical result

Suppose a multivariable PI controller is desired. Take $\lambda_1 = 4.5$ and $\lambda_2 = 4$. The controller parameters are

$$\boxed{\mathbf{K_C} = \begin{bmatrix} 0.2833 & -0.0411 \\ 0.0915 & -0.1210 \end{bmatrix}, \ \mathbf{T_I} = \begin{bmatrix} 0.0285 & -0.0219 \\ 0.0097 & -0.0148 \end{bmatrix}}$$

Example (ctd.4)

If there is not any constraint on the structure of PID controller, one can choose the PID controller with best achievable performance. Nevertheless, the resulting controller is unstable. To overcome this problem, the first column of $\mathbf{C}(s)$ is chosen in the form of the PI controller, while the second column is chosen in the form of the PID controller. Take $\lambda_1 = 3.8$ and $\lambda_2 = 3.5$. The multivariable PID controller is

$$\boxed{\mathbf{C}(s) = \begin{bmatrix} \frac{0.3251s+0.0327}{s} & -\frac{3.0901s^2+0.8804s+0.0235}{s(35.4901s+1)} \\ \frac{1.0506s+0.1112}{10s} & -\frac{8.6640s^2+1.1570s+0.0159}{s(64.3898s+1)} \end{bmatrix}}$$

The closed-loop responses are shown in Figure. Because of the use of rational approximations, the interaction among different channels cannot be thoroughly decoupled in the systems with PI or PID controllers

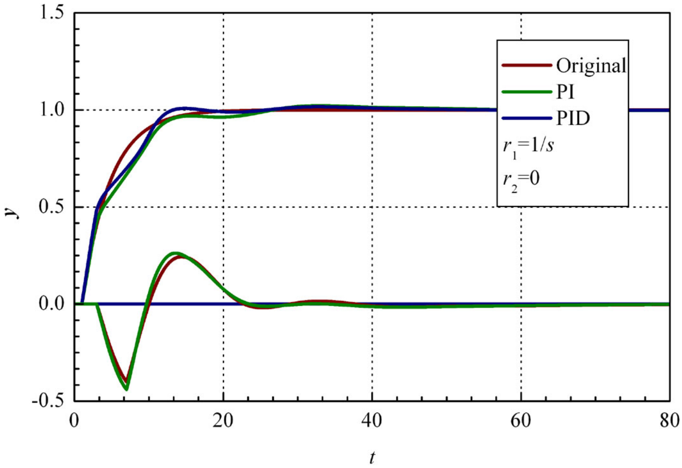


Figure: Closed-loop responses of multivariable PID controllers-1

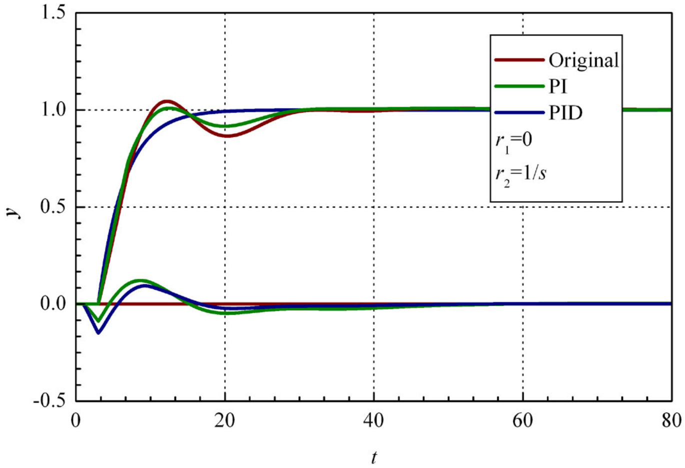


Figure: Closed-loop responses of multivariable PID controllers-2

End of Chapter 12