Chapter 6 Control of Stable Plants

6.1 The Quasi-H_∞ Smith Predictor

Chapter 4 and Chapter 5: The controller is analytically designed by minimizing the weighted sensitivity function

This section: The controller is analytically designed by specifying the desired closed-loop response
Actually, a simplified version of this method was already used in Sections 5.5 and 5.6

Consider the diagram of the Smith predictor in Figure, where $\tilde{G}(s)$ is the plant, $G(s)$ is its model, and $G_o(s)$ is the delay-free part of $G(s)$. If the closed-loop transfer function $T(s)$ is known, the controller of the Smith predictor is

$$R(s) = \frac{T(s)}{G(s) - T(s)G_o(s)}$$

6.1 The Quasi-H$_\infty$ Smith Predictor

Chapter 4 and Chapter 5: The controller is analytically designed by minimizing the weighted sensitivity function

This section: The controller is analytically designed by specifying the desired closed-loop response
Actually, a simplified version of this method was already used in Sections 5.5 and 5.6

Consider the diagram of the Smith predictor in Figure, where $\tilde{G}(s)$ is the plant, $G(s)$ is its model, and $G_o(s)$ is the delay-free part of $G(s)$. If the closed-loop transfer function $T(s)$ is known, the controller of the Smith predictor is

$$R(s) = \frac{T(s)}{G(s) - T(s)G_o(s)}$$

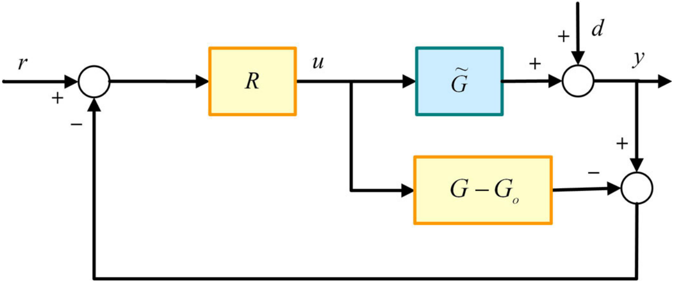

Figure: Diagram of the Smith predictor

Key of the design

How to choose the desired closed-loop transfer function

To introduce the idea clearly, the simplest case is considered first. The general result will be inductively derived

Case 1:

Consider the following stable rational plant of MP:

$$G(s) = \frac{KN_-(s)}{M_-(s)}$$

where K is the gain, $N_-(s)$ and $M_-(s)$ are the polynomials with roots in the LHP, $N_-(0) = M_-(0) = 1$, and $\deg\{N_-\} \le \deg\{M_-\}$. It is easy to control such a plant. For the H_∞ performance index and the weighting function $W(s) = 1/s$ we have

$$\begin{aligned} \|W(s)S(s)\|_\infty &= \|W(s)[1 - G(s)Q(s)]\|_\infty \\ &\ge 0 \end{aligned}$$

The following controller is the optimal one:

$$Q_{opt}(s) = \frac{M_-(s)}{KN_-(s)}$$

Introduce the filter

$$J(s) = \frac{1}{(\lambda s + 1)^{n_j}}$$

where λ is the performance degree. In light of the discussion in Section 5.7, n_j is chosen as follows:

$$n_j = \begin{cases} \deg\{M_-\} - \deg\{N_-\} & \deg\{M_-\} > \deg\{N_-\} \\ 1 & \deg\{M_-\} = \deg\{N_-\} \end{cases}$$

The suboptimal proper controller is

$$Q(s) = \frac{M_-(s)}{KN_-(s)(\lambda s + 1)^{n_j}}$$

The closed-loop transfer function is

$$T(s) = \frac{1}{(\lambda s + 1)^{n_j}}$$

Case 2:

Consider a bit more complex case. Assume that the plant has a zero in the RHP:

$$G(s) = \frac{KN_-(s)(-z_r^{-1}s+1)}{M_-(s)}$$

where $z_r > 0$, $N_-(0) = M_-(0) = 1$, and $\deg\{N_-\} + 1 \leq \deg\{M_-\}$. Solve the weighted sensitivity problem again:

$$\begin{aligned} \|W(s)S(s)\|_\infty &= \|W(s)[1 - G(s)Q(s)]\|_\infty \\ &\geq |W(z_r)| \end{aligned}$$

The optimal controller is obtained as follows:

$$Q_{opt}(s) = \frac{M_-(s)}{KN_-(s)}$$

Introduce the following filter:

$$J(s) = \frac{1}{(\lambda s + 1)^{n_j}}$$

where

$$n_j = \deg\{M_-\} - \deg\{N_-\}$$

The suboptimal proper controller is

$$Q(s) = \frac{M_-(s)}{KN_-(s)(\lambda s + 1)^{n_j}}$$

The closed-loop transfer function can be written as

$$T(s) = \frac{-z_r^{-1} s + 1}{(\lambda s + 1)^{n_j}}$$

Case 3:

Now, consider the general stable rational plant described by

$$G(s) = \frac{KN_+(s)N_-(s)}{M_-(s)}$$

where $N_-(s)$ and $M_-(s)$ are the polynomials with roots in the LHP, $N_+(s)$ is a polynomial with roots in the RHP, $N_+(0) = N_-(0) = M_-(0) = 1$, and $\deg\{N_+\} + \deg\{N_-\} \leq \deg\{M_-\}$. As this is a rational plant, $G_o(s) = G(s)$

Motivated by the foregoing design procedures, the following function is chosen as the desired closed-loop transfer function:

$$T(s) = N_+(s)J(s)$$

where $J(s)$ is a filter

$$J(s) = \frac{1}{(\lambda s + 1)^{n_j}}$$

and

$$n_j = \begin{cases} \deg\{M_-\} - \deg\{N_-\} & \deg\{M_-\} > \deg\{N_-\} \\ 1 & \deg\{M_-\} = \deg\{N_-\} \end{cases}$$

The feature of the closed-loop transfer function is that it has the same RHP zeros as the plant.

Once the desired $T(s)$ is determined, the controller of the Smith predictor can be analytically derived through

$$\begin{aligned} R(s) &= \frac{T(s)}{G(s) - T(s)G_o(s)} \\ &= \frac{1}{K} \frac{M_-(s)}{N_-(s)[(\lambda s + 1)^{n_j} - N_+(s)]} \end{aligned}$$

For rational plants, the unity feedback loop controller $C(s)$ is identical to $R(s)$. The controller has the same order as that of the plant. The corresponding $Q(s)$ is

$$Q(s) = \frac{T(s)}{G(s)} = \frac{M_-(s)}{N_-(s)}$$

Case 4:

When there is a time delay in the plant, the basic idea of designing the Smith predictor is to move the time delay out from the feedback loop, so that the controller can be designed for the rational part of the plant. Along this line, the design procedure for rational plants can be extended to plants with time delays

For rational plants, the unity feedback loop controller $C(s)$ is identical to $R(s)$. The controller has the same order as that of the plant. The corresponding $Q(s)$ is

$$Q(s) = \frac{T(s)}{G(s)} = \frac{M_-(s)}{N_-(s)}$$

Case 4:

When there is a time delay in the plant, the basic idea of designing the Smith predictor is to move the time delay out from the feedback loop, so that the controller can be designed for the rational part of the plant. Along this line, the design procedure for rational plants can be extended to plants with time delays

Assume that the plant with time delay is

$$G(s) = \frac{KN_+(s)N_-(s)}{M_-(s)} e^{-\theta s}$$

where θ is the time delay. The desired closed-loop transfer function can be chosen as

$$T(s) = N_+(s)J(s)e^{-\theta s}$$

where $J(s)$ is identical to (1). The $R(s)$ and $Q(s)$ corresponding to this desired closed-loop transfer function is the same as those in (1) and (1) respectively, but $C(s)$ contains a time delay:

$$C(s) = \frac{Q(s)}{1 - G(s)Q(s)} = \frac{1}{K} \frac{M_-(s)}{N_-(s)[(\lambda s + 1)^{n_j} - N_+(s)e^{-\theta s}]}$$

which is irrational

Stability is a basic requirement for control system design. A question associated with the design is whether the closed-loop system is internally stable.

Theorem

The closed-loop system is internally stable

Proof.

Follows directly from the Youla parameterization for stable plants □

The design method here is in fact a pole placement method. Since the method is developed based on special H_∞ solutions, it is named quasi-H_∞ control

Stability is a basic requirement for control system design. A question associated with the design is whether the closed-loop system is internally stable.

Theorem

The closed-loop system is internally stable

Proof.

Follows directly from the Youla parameterization for stable plants □

The design method here is in fact a pole placement method. Since the method is developed based on special H_∞ solutions, it is named quasi-H_∞ control

Stability is a basic requirement for control system design. A question associated with the design is whether the closed-loop system is internally stable.

Theorem

The closed-loop system is internally stable

Proof.

Follows directly from the Youla parameterization for stable plants □

The design method here is in fact a pole placement method. Since the method is developed based on special H_∞ solutions, it is named quasi-H_∞ control

A frequently encountered case is that the plant is MP or has only one zero in the RHP. Then an exact H_∞ controller can be obtained by the method

If the plant has more than one zero in the RHP or the plant contains a time delay, the results of H_∞ control and the quasi-H_∞ control are different

The quasi-H_∞ control is a compromise: The solution may not be an exact H_∞ controller, but the design is significantly simplified

The analytical design formula for the quasi-H_∞ controller has been given. If the nominal plant is known, the quasi-H_∞ controller can be obtained by directly substituting the plant parameters into the formula. One can also design the quasi-H_∞ controller through the following steps:

A frequently encountered case is that the plant is MP or has only one zero in the RHP. Then an exact H_∞ controller can be obtained by the method

If the plant has more than one zero in the RHP or the plant contains a time delay, the results of H_∞ control and the quasi-H_∞ control are different

The quasi-H_∞ control is a compromise: The solution may not be an exact H_∞ controller, but the design is significantly simplified

The analytical design formula for the quasi-H_∞ controller has been given. If the nominal plant is known, the quasi-H_∞ controller can be obtained by directly substituting the plant parameters into the formula. One can also design the quasi-H_∞ controller through the following steps:

1. If the plant does not contain a time delay, turn to 3.
2. If the plant contains a time delay, take the rational part of the plant as the nominal plant.
3. If the nominal plant does not have any zeros in the RHP, take its inverse as $Q_{opt}(s)$ and turn to 5.
4. If the nominal plant has zeros in the RHP, remove the factor that contains these zeros and take the inverse of the reminder as $Q_{opt}(s)$.
5. Introduce a filter to $Q_{opt}(s)$, compute the controller by $R(s) = Q(s)/[1 - G_o(s)Q(s)]$ and $C(s) = Q(s)/[1 - G(s)Q(s)]$.

If it is necessary, the desired closed-loop transfer function of the quasi-H$_\infty$ control can be chosen as complex as desired

6.2 The H_2 Optimal Controller and the Smith Predictor

Consider the general plant used in the last section:

$$G(s) = \frac{KN_+(s)N_-(s)}{M_-(s)}e^{-\theta s}$$

It is assumed that the performance index is $\min \|W(s)S(s)\|_2$, the input is a unit step, and the weighting function is $W(s) = 1/s$. For asymptotic tracking, the following constraint must be satisfied:

$$\lim_{s\to 0}[1 - G(s)Q(s)] = 0$$

The $Q(s)$ that satisfies the condition can be expressed as

$$Q(s) = \frac{1}{K} + sQ_1(s)$$

where $Q_1(s)$ is stable

Therefore,

$$
\begin{aligned}
& \|W(s)S(s)\|_2^2 \\
= & \left\| W(s)\left\{1 - G(s)\left[\frac{1}{K} + sQ_1(s)\right]\right\}\right\|_2^2 \\
= & \left\| \frac{1}{s}\left[1 - \frac{N_+(s)N_-(s)}{M_-(s)}e^{-\theta s} - \frac{KN_+(s)N_-(s)s}{M_-(s)}e^{-\theta s}Q_1(s)\right]\right\|_2^2 \\
= & \left\| \frac{M_-(s) - N_+(s)N_-(s)e^{-\theta s}}{sM_-(s)} - \frac{KN_+(s)N_-(s)}{M_-(s)}e^{-\theta s}Q_1(s)\right\|_2^2 \\
= & \left\| \frac{N_+(s)}{N_+(-s)}e^{-\theta s}\left[\frac{M_-(s)N_+(-s)e^{\theta s} - N_+(s)N_-(s)N_+(-s)}{sM_-(s)N_+(s)} - \frac{KN_+(-s)N_-(s)}{M_-(s)}Q_1(s)\right]\right\|_2^2 \\
= & \left\| \frac{M_-(s)N_+(-s)e^{\theta s} - N_+(s)N_-(s)N_+(-s)}{sM_-(s)N_+(s)} - \frac{KN_+(-s)N_-(s)}{M_-(s)}Q_1(s)\right\|_2^2
\end{aligned}
$$

$$= \left\| \begin{array}{l} \frac{N_+(-s)e^{\theta s} - N_+(s)}{sN_+(s)} + \\ \frac{M_-(s) - N_-(s)N_+(-s)}{sM_-(s)} - \frac{KN_-(s)N_+(-s)}{M_-(s)} Q_1(s) \end{array} \right\|_2^2$$

Since $M_-(0) = N_+(0) = N_-(0) = 1$, s must be a factor of

$$N_+(-s)e^{\theta s} - N_+(s)$$

and

$$M_-(s) - N_-(s)N_+(-s)$$

Then we have

$$\begin{aligned} & \|W(s)S(s)\|_2^2 \\ = \; & \left\| \frac{N_+(-s)e^{\theta s} - N_+(s)}{sN_+(s)} \right\|_2^2 + \\ & \left\| \frac{M_-(s) - N_-(s)N_+(-s)}{sM_-(s)} - \frac{KN_-(s)N_+(-s)}{M_-(s)} Q_1(s) \right\|_2^2 \end{aligned}$$

Minimizing the right-hand side gives the optimal performance:

$$\min \|W(s)S(s)\|_2^2 = \left\| \frac{N_+(-s)e^{\theta s} - N_+(s)}{sN_+(s)} \right\|_2^2$$

There are two important implications with regard to the result:

1. This performance is the limit of the H_2 control for the given index and input, **no matter what design method is used**
2. The optimal performance is obtained by using **only the input-output information**

As the unique optimal $Q_{1opt}(s)$ is

$$Q_{1opt}(s) = \frac{M_-(s) - N_-(s)N_+(-s)}{KsN_-(s)N_+(-s)}$$

The optimal controller is

$$Q_{opt}(s) = \frac{M_-(s)}{KN_-(s)N_+(-s)}$$

Introduce the following filter to roll the optimal controller off:

$$J(s) = \frac{1}{(\lambda s + 1)^{n_j}}$$

where λ is the performance degree,

$$n_j = \begin{cases} \deg\{M_-\} - \{N_+\} - \{N_-\} & \{M_-\} > \{N_+\} + \{N_-\} \\ 1 & \{M_-\} = \{N_+\} + \{N_-\} \end{cases}$$

The suboptimal controller is

$$Q(s) = Q_{opt}(s)J(s) = \frac{M_-(s)}{KN_-(s)N_+(-s)(\lambda s + 1)^{n_j}}$$

The Smith predictor is

$$R(s) = \frac{1}{K}\frac{M_-(s)}{N_-(s)[(\lambda s + 1)^{n_j}N_+(-s) - N_+(s)]}$$

Notice that the order of the controller is identical to that of the rational part of the plant

The unity feedback loop controller is

$$C(s) = \frac{1}{K} \frac{M_-(s)}{N_-(s)[(\lambda s + 1)^{n_j} N_+(-s)e^{-\theta s} - N_+(s)]}$$

One can also design it through the following steps:

1. If the plant does not contain a time delay, turn to 3.
2. If the plant contains a time delay, take the rational part of the plant as the nominal plant.
3. If the nominal plant does not have zeros in the RHP, take its inverse as $Q_{opt}(s)$ and turn to 5.
4. If the nominal plant has zeros in the RHP, construct an all-pass transfer function by using the factor that contains these zeros and then remove the all-pass transfer function. Take the inverse of the remainder as $Q_{opt}(s)$.
5. Introduce a filter to $Q_{opt}(s)$, compute $R(s)$ and $C(s)$.

The procedures for designing the quasi-H_∞ controller and the H_2 controller is illustrated in the following example.

Example

Consider the control system of the maglev gap described in the last chapter. The dynamic model of the gap is

$$G(s) = \frac{s-4}{(s+2)^2}$$

Normalize the plant so that the constant terms of all factors are 1:

$$G(s) = -\frac{-s/4+1}{(s/2+1)^2}$$

First, the quasi-H_∞ controller is designed. There is no time delay in the plant, but there is a RHP zero. Remove the factor containing the zero and take the inverse of the reminder as $Q_{opt}(s)$:

Example (ctd.1)

$$Q_{opt}(s) = \frac{(s/2+1)^2}{-1}$$

A proper $Q(s)$ can be obtained by introducing a filter. When $Q(s)$ is known, it is trivial to compute $R(s)$ and $C(s)$.

Now design the H_2 controller. First, an all-pass transfer function has to be constructed by utilizing the factor that contains the RHP zero:

$$G(s) = -\frac{s/4+1}{(s/2+1)^2}\frac{-s/4+1}{s/4+1}$$

Second, remove the all-pass transfer function and take the inverse of the remainder as $Q_{opt}(s)$:

Example (ctd.2)

$$Q_{opt}(s) = -\frac{(s/2+1)^2}{s/4+1}$$

Finally, introduce a filter to Q_{opt}(s)

Construction of the all-pass transfer function is very simple for SISO plants. Assume that an open RHP zero of the plant is $z_r = a + bi, a > 0$. The all-pass transfer function can be constructed as follows:

$$\frac{-s+z_r}{s+\bar{z}_r}$$

Example (ctd.2)

$$Q_{opt}(s) = -\frac{(s/2+1)^2}{s/4+1}$$

Finally, introduce a filter to Q_{opt}(s)

Construction of the all-pass transfer function is very simple for SISO plants. Assume that an open RHP zero of the plant is $z_r = \mathrm{a} + \mathrm{b}i, \mathrm{a} > 0$. The all-pass transfer function can be constructed as follows:

$$\frac{-s+z_r}{s+\bar{z}_r}$$

6.3 Equivalents of the Optimal Controller

Explanation for the Optimal Controller

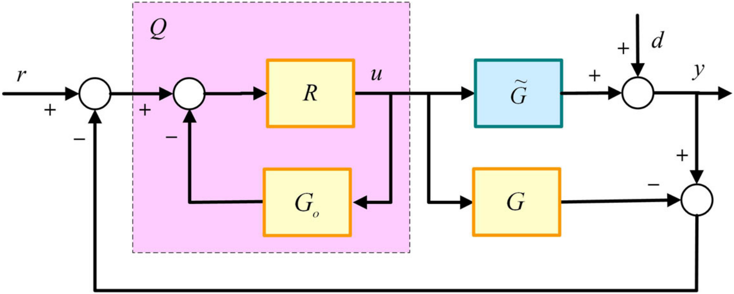

Figure: Rearrangement of the Smith predictor

Rearrange the diagram of the Smith predictor. An equivalent is obtained, which is in fact the IMC structure

Assume that the model is exact (that is, $\widetilde{G}(s) = G(s)$) and there is no disturbance. Then the feedback signal is zero. A natural idea is to take

$$Q(s) = G(s)^{-1}$$

as the controller. Then the closed-loop transfer function is

$$T(s) = G(s)Q(s) = 1$$

This implies that the output can track the reference instantaneously without any error. This situation, referred to as perfect control, is impossible in a real system, since the inverse of the time delay is non-casual. A non-casual transfer function is not physically realizable

An alternative is to take

$$Q(s) = G_o(s)^{-1}$$

as the controller. The closed-loop transfer function becomes

$$T(s) = G(s)Q(s) = e^{-\theta s}$$

which implies that the output can track the reference perfectly after the time delay θ. Such a result is reasonable. Imagine a shower control system. Assume that the temperature of outlet water is controlled by adjusting the flow rate of inlet hot water. When the valve of hot water is increased by a small percentage (so that the change on pressure can be omitted), the increased temperature can only be detected at the outlet after a period of time. **No matter what control method is used, it is impossible to eliminate the time delay**

$G_o(s)^{-1}$ is improper. Since $G_o(s)^{-1}$ is rational, it can be arbitrarily approximated by a proper transfer function of finite order:

$$Q(s) = \frac{G_o(s)^{-1}}{(\lambda s + 1)^{n_j}}$$

where λ is the performance degree and n_j is the positive integer that makes $Q(s)$ bi-proper. Obviously, **this is exactly the result obtained in the optimal design**

Norminal case: λ can be any positive real number. There is no overshoot and the rise time can be arbitrarily fast as $\lambda -> 0$

Uncertain case: λ can be calculated. However, in practice it is difficult to obtain an uncertainty profile with high precision, and the profile may vary. In this case, the tuning method introduced in the last two chapters can be used: **Increase the performance degree monotonically until the required response is obtained**

$G_o(s)^{-1}$ is improper. Since $G_o(s)^{-1}$ is rational, it can be arbitrarily approximated by a proper transfer function of finite order:

$$Q(s) = \frac{G_o(s)^{-1}}{(\lambda s + 1)^{n_j}}$$

where λ is the performance degree and n_j is the positive integer that makes $Q(s)$ bi-proper. Obviously, **this is exactly the result obtained in the optimal design**

Norminal case: λ can be any positive real number. There is no overshoot and the rise time can be arbitrarily fast as $\lambda - > 0$
Uncertain case: λ can be calculated. However, in practice it is difficult to obtain an uncertainty profile with high precision, and the profile may vary. In this case, the tuning method introduced in the last two chapters can be used: **Increase the performance degree monotonically until the required response is obtained**

Relationship Among Several Controllers

Status quo: Many different design methods have been developed in the past decades. Some of these methods have been applied to real systems and provide satisfied performances

Question: The optimal solution is unique in mathematics. Are these methods really independent?

The quasi-H_∞ controller and the H_2 controller:
If the plant has a stable rational part of MP, they are identical

Dahlin controller: The attractiveness of this technique comes from the fact that it is easy to use and can provide good performance

Relationship Among Several Controllers

Status quo: Many different design methods have been developed in the past decades. Some of these methods have been applied to real systems and provide satisfied performances

Question: The optimal solution is unique in mathematics. Are these methods really independent?

The quasi-H_∞ controller and the H_2 controller:
If the plant has a stable rational part of MP, they are identical

Dahlin controller: The attractiveness of this technique comes from the fact that it is easy to use and can provide good performance

Relationship Among Several Controllers

Status quo: Many different design methods have been developed in the past decades. Some of these methods have been applied to real systems and provide satisfied performances

Question: The optimal solution is unique in mathematics. Are these methods really independent?

The quasi-H_∞ controller and the H_2 controller:
If the plant has a stable rational part of MP, they are identical

Dahlin controller: The attractiveness of this technique comes from the fact that it is easy to use and can provide good performance

The Dahlin algorithm was presented for the first-order plant with time delay:

$$G(s) = \frac{Ke^{-\theta s}}{\tau s + 1}$$

The basic idea is to specify the desired closed-loop transfer function $T(s)$ as a first-order transfer function with its time delay equal to that of the plant $G(s)$; that is,

$$T(s) = \frac{e^{-\theta s}}{\lambda s + 1}$$

from which a unity feedback loop controller $C(s)$ can be derived. Since the time delay is difficult to treat in Laplace domain, the design procedure is performed in discrete domain. The time delay is a finite dimension function in discrete domain

Evidently, the Dahlin algorithm is a discrete domain version of the quasi-H_∞ controller and the H_2 controller for the first-order plant with time delay

Deadbeat control:

When $\lambda \to 0$, the Dahlin controller reduces to the deadbeat controller (also referred to as minimal prototype controller). Therefore, the deadbeat controller is a special case of the quasi-H_∞ controller and the H_2 controller as well

Inferential control and IMC:

Consider the diagram of the inferential control in Figure. A plant is given to the right-hand side of the dotted line, with one unmeasured output $\widetilde{y}(s)$ and one secondary measured auxiliary output $y(s)$. The manipulated variable $u(s)$ and the disturbance $d(s)$ affect both outputs

Evidently, the Dahlin algorithm is a discrete domain version of the quasi-H_∞ controller and the H_2 controller for the first-order plant with time delay

Deadbeat control:

When $\lambda \to 0$, the Dahlin controller reduces to the deadbeat controller (also referred to as minimal prototype controller). Therefore, the deadbeat controller is a special case of the quasi-H_∞ controller and the H_2 controller as well

Inferential control and IMC:

Consider the diagram of the inferential control in Figure. A plant is given to the right-hand side of the dotted line, with one unmeasured output $\widetilde{y}(s)$ and one secondary measured auxiliary output $y(s)$. The manipulated variable $u(s)$ and the disturbance $d(s)$ affect both outputs

Evidently, the Dahlin algorithm is a discrete domain version of the quasi-H_∞ controller and the H_2 controller for the first-order plant with time delay

Deadbeat control:

When $\lambda \to 0$, the Dahlin controller reduces to the deadbeat controller (also referred to as minimal prototype controller). Therefore, the deadbeat controller is a special case of the quasi-H_∞ controller and the H_2 controller as well

Inferential control and IMC:

Consider the diagram of the inferential control in Figure. A plant is given to the right-hand side of the dotted line, with one unmeasured output $\widetilde{y}(s)$ and one secondary measured auxiliary output $y(s)$. The manipulated variable $u(s)$ and the disturbance $d(s)$ affect both outputs

The disturbance is considered to be unmeasured. $Q(s)$ is the controller and $G(s)$ is the model of a stable MP plant

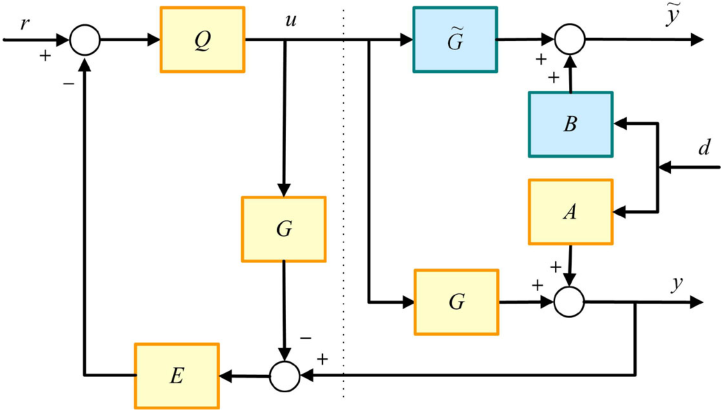

Figure: Inferential control system

Since

$$y(s) = G(s)u(s) + A(s)d(s)$$

the disturbance can be written as

$$d(s) = \frac{y(s)}{A(s)} - \frac{G(s)}{A(s)}u(s)$$

Define an estimator

$$E(s) := \frac{B(s)}{A(s)}$$

The estimated value of the unmeasured output is

$$\begin{aligned} \widetilde{y}(s) &= \widetilde{G}(s)u(s) + B(s)d(s) \\ &= \widetilde{G}(s)u(s) + E(s)[y(s) - G(s)u(s)] \end{aligned}$$

The function of $E(s)$ is to combine inputs and predict the effect of the unmeasured disturbance on the plant output

Now assume that the output $\widetilde{y}(s)$ can be measured. If the model is exact: $\widetilde{G}(s) = G(s)$, $\widetilde{y}(s) = y(s)$, $A(s) = B(s)$, then $E(s) = 1$. The inferential control structure reduces to the IMC structure

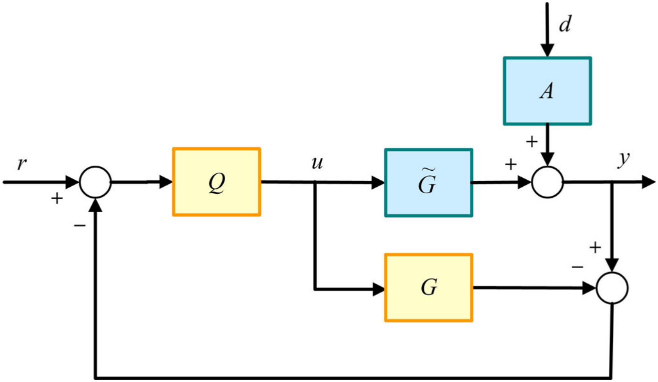

Figure: Reduced inferential control system

The signal entering the estimator is $d(s)A(s)$. To reject its effect on the plant output, the control effort should be

$$u(s) = -d(s)A(s)Q(s)$$

Cancellation is perfect when $Q(s) = 1/G(s)$.
For the general plant

$$G(s) = \frac{KN_+(s)N_-(s)}{M_-(s)}$$

the controller is

$$Q(s) = \frac{M_-(s)}{KN_-(s)N_+(-s)}$$

This controller contains the element that is not physically realizable. The problem can be solved by introducing a filter to the controller. Then, the result is identical to the H_2 controller

Consequently, the quasi-H_∞ control, the H_2 control, the inferential control with measured output, and the IMC are equivalent for the plant whose rational part is stable MP

The H_2 control, the inferential control scheme with measured output, and the IMC are equivalent for the plant whose rational part is stable

Model predictive control:

The model predictive control is a general designation of a variety of control algorithms developed for computer control systems, rather than one single control algorithm

Assume that the plant is rational stable MP, and T_s denotes the sampling time

Consequently, the quasi-H_∞ control, the H_2 control, the inferential control with measured output, and the IMC are equivalent for the plant whose rational part is stable MP

The H_2 control, the inferential control scheme with measured output, and the IMC are equivalent for the plant whose rational part is stable

Model predictive control:

The model predictive control is a general designation of a variety of control algorithms developed for computer control systems, rather than one single control algorithm

Assume that the plant is rational stable MP, and T_s denotes the sampling time

Model: The value of a unit step response at every sampling instant $t = T_s, 2T_s, \ldots, NT_s$: $a_1, a_2, \ldots, a_N$

Control objective: The predicted output $y_p(k)$ on the considered horizon L follows the desired output trajectory $y_r(k)$

Desired output trajectory:

$$y_r(k+1) = \alpha y(k) + (1-\alpha)r(k)$$

Here $\alpha = e^{-T_s/\lambda}$, λ is the time constant of the desired output trajectory, and $y(s)$ is the real output of the plant

Objective function:

$$\min \sum_{i=1}^{P} [y_p(k+i) - y_r(k+i)]^2$$

P control variables, $u(k)$s, can be calculated by minimizing it

Comparison:

The predictive model **vs** The step response model in the form of a transfer function

The desired output trajectory **vs** The desired closed-loop transfer function

The objective function **vs** The optimal performance index

It is seen that the design idea of the model predictive control is very similar to that of the quasi-H_∞ Smith predictor and the H_2 Smith predictor.

Certainly, model predictive control involves many algorithms. Each predictive algorithm possesses its specific form. Not every predictive algorithm is exactly equivalent to the quasi-H_∞ Smith predictor and the H_2 Smith predictor

Consider the one step MAC, that is, $P = L = 1$. Let the output of the model is $y_m(k)$. The predicted output of the plant, $y_p(k)$, is

$$y_p(k+1) = y_m(k+1) + [y(k) - y_m(k)]$$

When the system is optimal, we have $y_p(k+1) = y_r(k+1)$ and $y(k) = y_r(k)$. Then

$$\alpha y_r(k) + (1-\alpha)r(k) = y_r(k) + y_m(k+1) - y_m(k)$$

Taking the Z-transform yields

$$(1-\alpha)r(z) - (1-\alpha)y_r(z) = (z-1)y_m(z)$$

This equation, together with the Z-transforms of the model output and the desired output trajectory

$$y_m(z) = u(z)G(z), y_r(z) = \frac{1-\alpha}{z-\alpha}r(z)$$

shows that

$$\frac{u(z)}{r(z)} = \frac{1-\alpha}{(z-\alpha)G(z)}$$

Computing the Laplace domain version of this controller, one can find it is identical to that of the quasi-H_∞ controller and the H_2 controller.

Similarly, it can be proved that the DMC with $P = L$ is identical to the quasi-H_∞ controller and the H_2 controller.

6.4 The PID Controller and High-Order Controllers

Section 4.3 An approximate Smith predictor was derived by utilizing the PID controller
This section: How is a PID controller derived by utilizing the quasi-H_∞ Smith predictor or the H_2 Smith predictor

For simplicity of presentation, it might as well let the rational part of the plant be stable MP and have no zeros. In this case, the quasi-H_∞ control and the H_2 control results in the same controller

Rearranging the Smith predictor, one can obtain an equivalent unity feedback loop with the controller

$$C(s) = \frac{R(s)}{1 + [G_o(s) - G(s)]R(s)}$$

6.4 The PID Controller and High-Order Controllers

Section 4.3 An approximate Smith predictor was derived by utilizing the PID controller
This section: How is a PID controller derived by utilizing the quasi-H_∞ Smith predictor or the H_2 Smith predictor

For simplicity of presentation, it might as well let the rational part of the plant be stable MP and have no zeros. In this case, the quasi-H_∞ control and the H_2 control results in the same controller

Rearranging the Smith predictor, one can obtain an equivalent unity feedback loop with the controller

$$C(s) = \frac{R(s)}{1 + [G_o(s) - G(s)]R(s)}$$

Substitute the nominal plant

$$G(s) = \frac{Ke^{-\theta s}}{M_{-}(s)}$$

and the Smith predictor

$$R(s) = \frac{M_{-}(s)}{K(\lambda s + 1)^{n_j} - K}$$

into the controller. The obtained controller is

$$C(s) = \frac{1}{K} \frac{M_{-}(s)}{(\lambda s + 1)^{n_j} - e^{-\theta s}}$$

When $\lambda \to 0$, the $C(s)$ tends to be optimal. The optimal controller is unique

Assume that there is a first-order plant with time delay, that is, $M_{-}(s) = \tau s + 1$ and $n_j = 1$. The controller $C(s)$ is

$$C(s) = \frac{1}{K}\frac{\tau s + 1}{\lambda s + 1 - e^{-\theta s}}$$

The open-loop transfer function of the system is

$$\begin{aligned} L(s) &= G(s)C(s) \\ &= \frac{e^{-\theta s}}{\lambda s + 1 - e^{-\theta s}} \end{aligned}$$

The Nyquist plot and the Bode plot are given in Figures

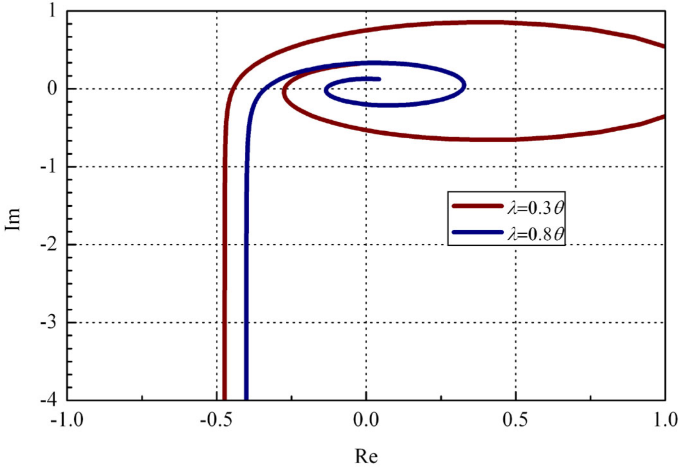

Figure: Nyquist plot of the system with the first-order plant

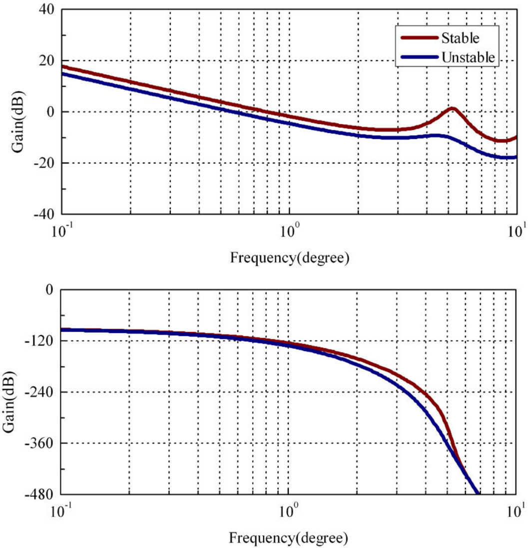

Figure: Bode plot of the system with the first-order plant

With this controller, the closed-loop response can be computed analytically. The reference response is

$$y(t) = \begin{cases} 0 & 0 < t < \theta \\ 1 - e^{-(t-\theta)/\lambda} & t \geq \theta \end{cases}$$

The response for the input disturbance can be written as

$$y(t) = \begin{cases} 0 & 0 < t < \theta \\ K\left[1 - e^{-(t-\theta)/\tau}\right] & 0 \leq t < 2\theta \\ K\left[\begin{array}{l} \frac{\lambda}{\lambda-\tau} e^{-(t-2\theta)/\lambda} - \\ \frac{\tau}{\lambda-\tau} e^{-(t-2\theta)/\tau} - e^{-(t-\theta)/\tau} \end{array}\right] & t \geq 2\theta \end{cases}$$

Since the feedback acts after $t \geq 2\theta$, the error appearing during $t < 2\theta$ can never be overcome. This error will not be less than $K(1 - e^{-\theta/\tau})$. The larger the θ/τ, the larger the error

Evidently, the PID controller can not be used as an exact substitute for the optimal controller. There are two reasons:

1. When the time delay equals zero, if the order of the plant is greater than 3, $C(s)$ is of high order. A PID controller is not able to reproduce the dynamics of $C(s)$ exactly
2. When the time delay is not zero, $C(s)$ involves a time delay and thus is of infinite dimension. Since the PID is of finite dimension, it cannot reproduce the dynamics of $C(s)$

Two ways to derive a PID controller from high-order controllers:

1. expand $e^{-\theta s}$ by using the Pade approximation. The resulting controller is the H_2 PID controller developed in Section 5.1
2. Approximate the overall controller by a low-order rational function. One obtains the Maclaurin PID controller (Section 5.5) or the PID controller with best achievable performance (Sections 5.6)

Evidently, the PID controller can not be used as an exact substitute for the optimal controller. There are two reasons:

1. When the time delay equals zero, if the order of the plant is greater than 3, $C(s)$ is of high order. A PID controller is not able to reproduce the dynamics of $C(s)$ exactly
2. When the time delay is not zero, $C(s)$ involves a time delay and thus is of infinite dimension. Since the PID is of finite dimension, it cannot reproduce the dynamics of $C(s)$

Two ways to derive a PID controller from high-order controllers:

1. expand $e^{-\theta s}$ by using the Pade approximation. The resulting controller is the H_2 PID controller developed in Section 5.1
2. Approximate the overall controller by a low-order rational function. One obtains the Maclaurin PID controller (Section 5.5) or the PID controller with best achievable performance (Sections 5.6)

Besides these analytical design methods, one can also utilize the pole-placement method or other numerical algorithms to design the PID controller. This is not recommended because

1. Numerical methods are tedious.
2. The controller has fixed parameters. It cannot be tuned for quantitative responses.
3. Almost no performance improvement can be obtained.

The following example is used to compare the responses of different controllers.

Example

The primary control loop of a nuclear power plant is shown in Figure. The goal is to control the temperature of water by adjusting the speed of the reaction, which is determined by the depth of the control rods in the reactor.

Besides these analytical design methods, one can also utilize the pole-placement method or other numerical algorithms to design the PID controller. This is not recommended because

1. Numerical methods are tedious.
2. The controller has fixed parameters. It cannot be tuned for quantitative responses.
3. Almost no performance improvement can be obtained.

The following example is used to compare the responses of different controllers.

Example

The primary control loop of a nuclear power plant is shown in Figure. The goal is to control the temperature of water by adjusting the speed of the reaction, which is determined by the depth of the control rods in the reactor.

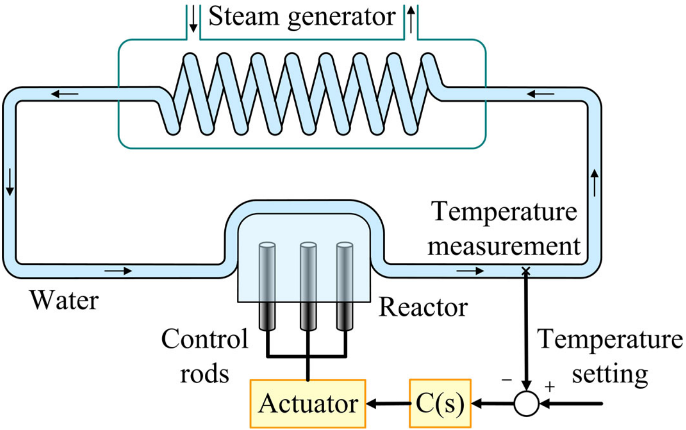

Figure: Control system of a nuclear reactor

Example (ctd.1)

Since the water has to be transported from the reactor to the measurement point, there is a time delay in the plant. The transfer function of the temperature plant is obtained by carrying out experiments:

$$\boxed{G(s) = \frac{e^{-0.4s}}{0.2s+1}.}$$

In this example,
$K = 1, \theta = 0.4, N_+(s) = 1, N_-(s) = 1, M_-(s) = 0.2s + 1$, and $n_j = 1$. The Smith predictor given by (1) or (1) is

$$\boxed{R(s) = \frac{0.2s+1}{\lambda s}.}$$

Example (ctd.2)

The H_2 PID controller is

$$\boxed{C(s) = \frac{(0.2s+1)^2}{s(0.2\lambda s + \lambda + 0.4)}}$$

The Maclaurin PID controller is

$$\boxed{C(s) = \frac{T_I}{\lambda + 0.4}\left[1 + \frac{1}{T_I s} + \frac{0.08(3T_I - 0.4)}{3T_I(\lambda + 0.4)}\right]}$$

where

$$T_I = 0.2 + \frac{0.08}{\lambda + 0.4}$$

Example (ctd.3)

The PID controller with best achievable performance is

$$\boxed{C(s) = a_1 \left[1 + \frac{a_0}{a_1 s} + \frac{a_2}{a_1} s \right] \frac{1}{b_1 s + 1}}$$

where

$$\begin{aligned}
a_0 &= \frac{1}{2(\lambda + 0.4)}, \\
a_1 &= \frac{0.224 + 0.16\lambda}{0.64 + 0.4\lambda}, \\
a_2 &= 0.0333 \frac{0.64 - 0.48\lambda}{0.64 + 0.4\lambda}, \\
b_1 &= -0.2 \frac{0.064 - 0.48\lambda - 0.4\lambda^2}{0.64 + 0.4\lambda}
\end{aligned}$$

Example (ctd.4)

Take $\lambda = 0.41\theta$ for these controllers. A unit step reference is added at $t = 0$ and a unit step load is added at $t = 100$. The nominal responses of the closed-loop system are shown in Figure

In practice, the reference has a limited bandwidth. If the bandwidth of reference is restricted to be 5 rad/s, the responses of these methods are closer than those without the restriction.

It is assumed that there are 10% uncertainties on the three parameters of the plant, respectively. The worst case is that the gain and the time delay become the maximum and the time constant becomes the minimum. In this case, the closed-loop responses are given in Figure. The responses of these controller are still steady with respect to this uncertainty

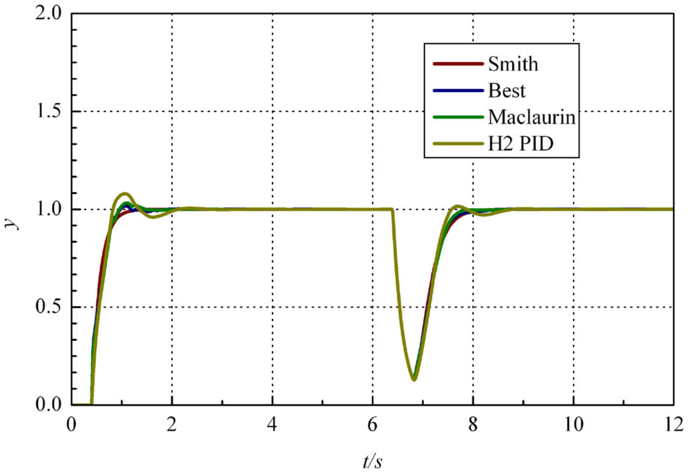

Figure: Responses for full frequency range

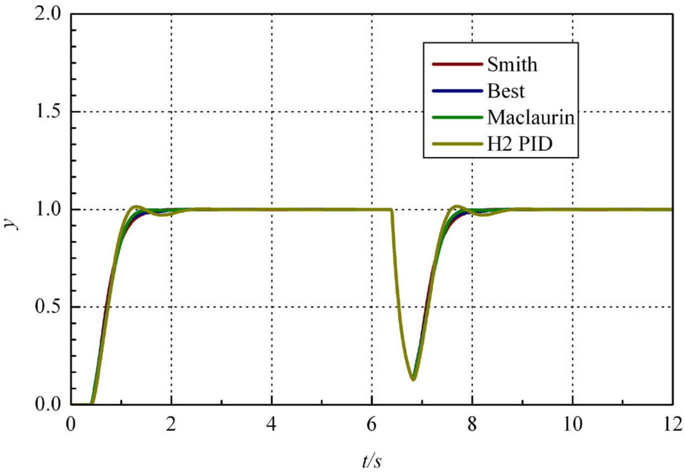

Figure: Responses for limited frequency range

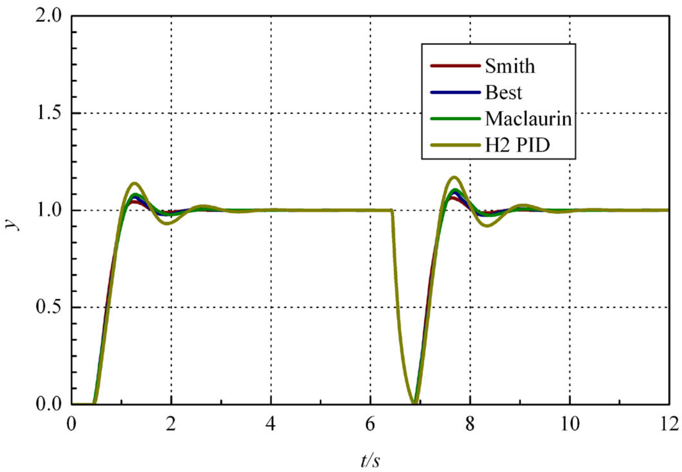

Figure: Worst-case responses with 10% uncertainties

6.5 Choice of Weighting Functions

Two problems that closely relate to the design of optimal controllers:

1. The choice of the filter (The subject of Section 5.7)
2. The choice of the weighting function (The subject of this section)

Basis of the design in this book: System gains

Constraints imposed on the weighting function by the system gain: The weighting function should be chosen in accordance with the system input

H_2 optimal control:

The controller is designed for an impulse

6.5 Choice of Weighting Functions

Two problems that closely relate to the design of optimal controllers:

1. The choice of the filter (The subject of Section 5.7)
2. The choice of the weighting function (The subject of this section)

Basis of the design in this book: System gains

Constraints imposed on the weighting function by the system gain: The weighting function should be chosen in accordance with the system input

H_2 optimal control:

The controller is designed for an impulse

The weighting function should be chosen so that

$$\left\| \frac{r(s)}{W(s)} \right\|_2 = 1$$

or equivalently, the weighting function should equal the system input

H_∞ optimal control:

Require that

$$\left\| \frac{r(s)}{W(s)} \right\|_2 \leq 1$$

To cover all energy-bounded inputs, a reasonable choice for the weighting function is to take it to be equal to the input

Now consider the choice problem of the weighting functions for several frequently encountered cases

The weighting function should be chosen so that

$$\left\| \frac{r(s)}{W(s)} \right\|_2 = 1$$

or equivalently, the weighting function should equal the system input

H_∞ optimal control:

Require that

$$\left\| \frac{r(s)}{W(s)} \right\|_2 \leq 1$$

To cover all energy-bounded inputs, a reasonable choice for the weighting function is to take it to be equal to the input

Now consider the choice problem of the weighting functions for several frequently encountered cases

Case 1:

In most methods, the control system is designed for ideal step inputs or step-like signals (that is, an ideal step signal with a lag). In this case, the transfer function of the input signal can directly be chosen as the weighting function for both the H_2 optimal control and the H_∞ optimal control

Case 2:

In statistic control, the input is a random signal. Normally, the statistics feature or the spectrum of the input is known. Then the transfer function of the equivalent input can be obtained. The weighting function $W(s)$ can also be taken as the transfer function of the equivalent input

Case 3:

In some applications, designers have acquired through experience the desired shape for the Bode magnitude plot of $S(s)$

Case 1:

In most methods, the control system is designed for ideal step inputs or step-like signals (that is, an ideal step signal with a lag). In this case, the transfer function of the input signal can directly be chosen as the weighting function for both the H_2 optimal control and the H_∞ optimal control

Case 2:

In statistic control, the input is a random signal. Normally, the statistics feature or the spectrum of the input is known. Then the transfer function of the equivalent input can be obtained. The weighting function $W(s)$ can also be taken as the transfer function of the equivalent input

Case 3:

In some applications, designers have acquired through experience the desired shape for the Bode magnitude plot of $S(s)$

Case 1:

In most methods, the control system is designed for ideal step inputs or step-like signals (that is, an ideal step signal with a lag). In this case, the transfer function of the input signal can directly be chosen as the weighting function for both the H_2 optimal control and the H_∞ optimal control

Case 2:

In statistic control, the input is a random signal. Normally, the statistics feature or the spectrum of the input is known. Then the transfer function of the equivalent input can be obtained. The weighting function $W(s)$ can also be taken as the transfer function of the equivalent input

Case 3:

In some applications, designers have acquired through experience the desired shape for the Bode magnitude plot of $S(s)$

In particular, a good performance is known to be achieved if the plot of $S(s)$ lies under some curve. In this case, the weighting function $W(s)$ can be chosen as the transfer function corresponding to the curve. A known $S(s)$ has two implications:

1. Since $S(s) + T(s) = 1$, the desired shape of $T(s)$ is determined.
2. The bandwidth of the system is restricted.

Further consideration:

The above method may not be a good one for the design problem. In the design problem, it is desirable the weighting function is as simple as possible. In some applications, the system input is complex. Consequently, the weighting function $W(s)$ is in a complex form. For example, the weighting function might be

$$W(s) = \frac{1}{s(\tau_1 s + 1)(\tau_2 s + 1)}.$$

In particular, a good performance is known to be achieved if the plot of $S(s)$ lies under some curve. In this case, the weighting function $W(s)$ can be chosen as the transfer function corresponding to the curve. A known $S(s)$ has two implications:

1. Since $S(s) + T(s) = 1$, the desired shape of $T(s)$ is determined.
2. The bandwidth of the system is restricted.

Further consideration:

The above method may not be a good one for the design problem. In the design problem, it is desirable the weighting function is as simple as possible. In some applications, the system input is complex. Consequently, the weighting function $W(s)$ is in a complex form. For example, the weighting function might be

$$W(s) = \frac{1}{s(\tau_1 s + 1)(\tau_2 s + 1)}.$$

The design of an optimal controller for a complex weighting function is tedious.

Simplified design: Take $W(s) = 1/s$
This works because the controller designed for an ideal step has a potential to work well for step-like signals

Design procedure:

1. Design the controller for the ideal step
2. Choose an appropriate filter according to the design requirement

Since $W(s)$ is fixed, the obtained closed-loop response may not reflect the required feature. The problem can simply be solved by tuning the filter (Figure)

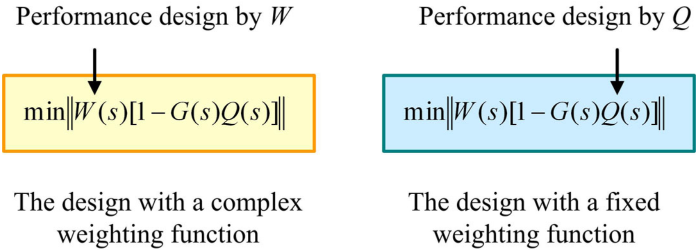

Figure: Designs with different procedures

Difference Between the Two Methods

The question of interest: Whether can the function of the weighting function be thoroughly substituted by a filter

Analysis: Suppose that the complex weighting function and the associated controller are $W_{p1}(s)$ and $Q_1(s)$ respectively, and the fixed weighting function and the associated controller are $W_{p2}(s)$ and $Q_2(s)$ respectively. In most cases, both of the two weighting functions are MP and do not have poles in the open RHP. For example, the weighting functions may be

$$W_{p1}(s) = \frac{1}{s(\tau_1 s + 1)(\tau_2 s + 1)} \text{ and } W_{p2}(s) = \frac{1}{s}$$

Difference Between the Two Methods

The question of interest: Whether can the function of the weighting function be thoroughly substituted by a filter

Analysis: Suppose that the complex weighting function and the associated controller are $W_{p1}(s)$ and $Q_1(s)$ respectively, and the fixed weighting function and the associated controller are $W_{p2}(s)$ and $Q_2(s)$ respectively. In most cases, both of the two weighting functions are MP and do not have poles in the open RHP. For example, the weighting functions may be

$$W_{p1}(s) = \frac{1}{s(\tau_1 s + 1)(\tau_2 s + 1)} \text{ and } W_{p2}(s) = \frac{1}{s}$$

Let

$$W_{p1}(s)[1 - G(s)Q_1(s)] = W_{p2}(s)[1 - G(s)Q_2(s)]$$

Then

$$Q_2(s) = G^{-1}(s)[1 - W_{p2}^{-1}(s)W_{p1}(s)] + W_{p2}^{-1}(s)W_{p1}(s)Q_1(s)$$

In light of the feature of $W_{p1}(s)$ and $W_{p2}(s)$, $W_{p2}^{-1}(s)W_{p1}(s)$ is stable

$G(s)$ **is MP:** For a given $W_{p1}(s)$ one can always find a stable $Q_2(s)$, which can reach the same performance as $W_{p1}(s)$

$G(s)$ **is NMP:** If $G^{-1}(s)[1 - W_{p2}^{-1}(s)W_{p1}(s)]$ is stable, one can find an equivalent $Q_2(s)$ to $W_{p1}(s)$. For other cases, the function of the weighting function cannot be substituted by a filter

Since $T(s) = G(s)Q_{opt}(s)J(s)$, the filter provides an alternative to design the controller for the required closed-loop response

The design method of using a filter has several advantages:

1. In many advanced design methods, the determination of weighting functions is a difficult problem. The "no-weight" design procedure given in this section does not require the designer to choose a weight function. The design task is significantly simplified.
2. In some methods, the weighting function is determined by empirical methods, which implies that different designers will obtain different controllers, even when the same method is used. With the design procedure in this section, different designers will obtain the same controller.
3. The filter is closely related to the closed-loop response. As compared with the method using weighting functions for performance design, the new method is provides a direct and simple means for adjusting the closed-loop response.

6.6 Simplified Tuning for Quantitative Robustness

Many methods: The controller is designed based on both the nominal plant and the uncertainty profile. If the uncertainty profile varies, the controller has to be re-designed

Problems: Such methods are inconvenient in practice.

A solution: The design methods introduced in this book do not depend on the prior information about the uncertainty profile

If the uncertainty profile is known, quantitative performance and robustness can be obtained by computation or tuning
When the uncertainty profile is not known, quantitative performance and robustness can roughly be achieved by tuning

If the uncertainty profile varies, it is not necessary to re-design the controller. The robust performance can be obtained by tuning the performance degree

The tuning procedure is simple: **Increase the performance degree monotonically until the required response is obtained**

Further consideration: The procedure is still a bit inconvenient. In many applications, the requirement on the dynamic performance is not very strict, while the convenience is very important. Sometimes, the dynamic performance is even sacrificed for the convenience

Goal of this section: Simplify the tuning procedure and to provide an engineering tuning method that can be applied in control software or hardware

Simplified Tuning

Assume that the plant model is

$$G(s) = \frac{Ke^{-\theta s}}{\tau s + 1}$$

and the uncertainties of the three parameters have the same amplitude. Their varying profile, e_{pu}, is expressed in the percentage of the nominal value. The unstructured uncertainty of the system can be expressed by

$$\Delta_m(j\omega) = \begin{cases} \left| \frac{|K| + e_{pu}K}{|K|} \frac{j\tau\omega + 1}{j(\tau - e_{pu}\tau)\omega + 1} e^{je_{pu}\theta\omega} - 1 \right| & \omega < \omega^* \\ \left| \frac{|K| + e_{pu}K}{|K|} \frac{j\tau\omega + 1}{j(\tau - e_{pu}\tau)\omega + 1} \right| + 1 & \omega \geq \omega^* \end{cases}$$

Here ω^* is determined by

$$e_{pu}\theta\omega^* + \arctan\frac{e_{pu}\tau\omega^*}{1+\tau(\tau - e_{pu}\tau)\omega^{*2}} = \pi, \frac{\pi}{2} \le e_{pu}\theta\omega^* \le \pi$$

It should be pointed out that the above assumption does not require the three parameters to change simultaneously, or that the three parameters must have the same level of uncertainty. The real case could be that the uncertainty of one parameter is larger than that of another one or the other two; nevertheless, the unstructured uncertainty profile is still within the same scope

Now let us see how to simplify the tuning procedure. Split the overall uncertainty scope e_{pu} into 3 ranges: small (10%), middle (20%), and large (30%). Certainly, it can also be split into more ranges for finer tuning. The three ranges are marked on a knob, which is set up on the panel of a regulator (Figure)

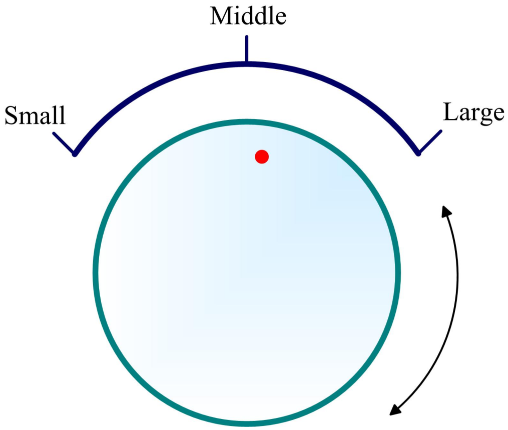

Figure: Three range knob of the performance degree

How to use:

1. Make a rough estimation of the uncertainty
2. Set the performance degree knob at a proper range
3. If the performance requirement changes or the uncertainty profile varies, the performance and the robustness can be adjusted by re-setting the performance degree knob

Two cases encountered:

1. The closed-loop system is unstable or the response oscillates intensely. This implies that the actual uncertainty is larger than the estimated. The performance degree knob should be set at a larger range
2. The response of the closed-loop system is always slow. This implies that the actual uncertainty is smaller than the estimated. The performance degree knob should be set at a smaller range

How to use:

1. Make a rough estimation of the uncertainty
2. Set the performance degree knob at a proper range
3. If the performance requirement changes or the uncertainty profile varies, the performance and the robustness can be adjusted by re-setting the performance degree knob

Two cases encountered:

1. The closed-loop system is unstable or the response oscillates intensely. This implies that the actual uncertainty is larger than the estimated. The performance degree knob should be set at a larger range
2. The response of the closed-loop system is always slow. This implies that the actual uncertainty is smaller than the estimated. The performance degree knob should be set at a smaller range

Performance Degree Somputation

Problem formulation: After the parameter uncertainty scope, e_{pu}, is given, how to determine the relationship between e_{pu} and λ: $\lambda = f(e_{pu})$

A simple method:

$$\lambda = (\alpha e_{pu} + \beta)\theta$$

where α and β are listed in Table. **This is a rough formula**, as its form is very simple

Table: Tuning parameters for three range knob

	10%	20%	30%
α	1.32	0.47	0.76
β	0.95	0.39	0.32

Scope: As indicated in the preceding section, many widely applied methods are equivalent to each other. Hence, the split-range tuning method can be applied not only to the design method in this book, but also to many other methods, for example, the Dahlin controller and the model predictive control

Comparison with the automatic camera:
Automatic camera: The user has to roughly estimate the distance from the objective to the camera. There are usually three choices: flower (which denotes near), portrait (which denotes middle), and mountain (which denotes far). The focus range is then determined. Other works are automatically finished by the camera after the shutter button is pressed

Split-range method: The only work to do is to roughly estimate the uncertainty profile. Other works are automatically finished by the regulator

End of Chapter 6