Chapter 8

Loop Design

8.1 Introduction

This is the first Chapter that deals with design and we will therefore start by some general aspects on design of engineering systems. Design is complicated because there are many issues that have to be considered. Much research and development has been devoted to development of design procedures and there have been numerous attempts to formalize the design problem. It is useful to think about a design problem in terms of specifications, trade-offs, limitations, design parameters. In additions it is useful to be aware of the fact that because of the richness of design problem there are some properties that are captured explicitly by the design method and other properties that must be investigated when the design is completed.

Specifications are an attempt to express the requirements formally. For control systems they typically include: attenuation of load disturbances, measurement noise, process uncertainty and command signal following. Specifications are typically expressed by quantities that capture these features. For control systems the fundamental quantities are typically transfer functions or time responses and specifications are some representative features of these functions. A design fundamentally involves trade-offs. It is also extremely important to be aware of fundamental limitations to avoid unrealistic specifications and impossible trade-offs. Sometimes it is attempted to capture the trade-offs in a single optimization criterion, but such a criterion always includes parameters that have to be chosen. It is often useful to make the trade-offs explicit and to have design parameters that gives the designer control of them. A design method typically focuses on some aspects of a design problem but it is often difficult to capture all aspects of a design problem formally. There will frequently be several properties that must be investigated separately when the design is completed. In control system design it is common practice to verify a design by extensive simulation that may include hardware in the loop. When discussing design in this and the following chapters we have made an attempt to emphasize the fundamental tradeoffs and the design parameters.

In this Chapter we will present a design method which is focused on the loop transfer function of the system. The method is focused on design a feedback controller that satisfies can deal with disturbances and process uncertainty. The response to reference signals can later be dealt with using feedforward which is discussed in the next chapter. The idea of loop shaping is outlined in Section 8.2 which present classical results that were developed when control emerged in the early 1940s. This section also contains some simple compenstators like lead-lag compensation, which is closely related to PID control. Section **??** presents another classical result due to Bode, who developed the concept of an ideal loop transfer function for electronic amplifiers. In the following sections it is shown how the loop transfer function relates to important properties of the closed loop system. Section **??** treats robustness properties. It is shown that the estimates developed earlier can be refined to give more precise information. Two graphical tools, the Hall diagram and the Nichols chart are also introduced. Section **??** deals with load disturbances. An estimate the norm of the transfer function from load disturbances to process output based on the crossover frequency is derived. The result shows that it is typically advantageous to have a high crossover frequency. Section **??** deals with measurement noise. An estimate of the norm of the transfer function from measurement noise to the control signal is developed. It is shown that a high crossover frequency is typically associated with high controller gains at frequencies above the crossover frequency. In Section **??** it is shown that non-minimum phase dynamics imposes severe constraints on the admissible crossover frequencies. Summarizing the findings of this section with the results of Sections **??**, **??**, and **??** we are in a good position to discuss the major trade-offs in the choice of gain crossover frequency. First the choice is severely restricted if process dynamics is not minimum phase. The choice is then governed by a compromise between robustness, attenuation of load disturbances and injection of measurement noise. It turns out that the compromise can be captured in a single diagram. This diagram also indicates the complexity of the controller that is required for different choices. This is shown in Section **??**.

8.2 Compensation

The idea of loop shaping is to find a controller so that the loop transfer function has desired properties. Since the loop transfer function $L = PC$ is the product of the transfer functions of the process and the controller it is easy to see how the loop transfer function is influenced by the controller. Compensation is typically done by successive modifications of the loop transfer function starting with proportional control. If the desired properties cannot be obtained by proportional control the controller is modified by multiplying the controller transfer function by compensating networks having simple transfer functions. The design is conveniently visualized using bode diagrams. Since the Bode diagram is logarithmic multiplication by a compensating network corresponds to additions in the Bode diagram. We start by discussing a desired properties of a loop transfer function, then we present some simple compensators and we end by a few examples. In the following sections we will present a more systematic treatment where we discuss how th different properties of the loop are influenced by the loop transfer function.

Properties of the Loop Transfer Function

The Bode diagram of a typical loop transfer function is shown in Figure 8.1. Recall that the lowest frequency where the loop transfer function has gain 1 is called gain crossover frequency, see Section **??**. This parameter is an important design parameter in loop shaping. When we talk about high and low frequencies in the following we are using the crossover frequency as a reference. The gain curve of a typical loop transfer function is decreasing as shown in Figure 8.1. The behavior at low frequencies determine attenuation of load disturbances and behavior of tracking of low frequency reference signals. The behavior around the crossover frequency determines robustness and sensitivity to modeling errors. The gain margin g_m and the phase margin φ_m are easily visualized in the Bode plot. The behavior at high frequency determines sensitivity to measurement noise.

It is desirable to have high gain at low frequency and rapidly decreasing gain after the gain crossover frequency. Since we require that the closed loop system should be stable, the slope of the gain curve at crossover n_{gc} cannot be too steep at the crossover. However, the phase of a system is related to its gain, and hence it is not possible to independently specify these quantities. In the remainder of this chapter we shall explore some of the limits of this tradeoff between gain and phase through a variety of standard compensator

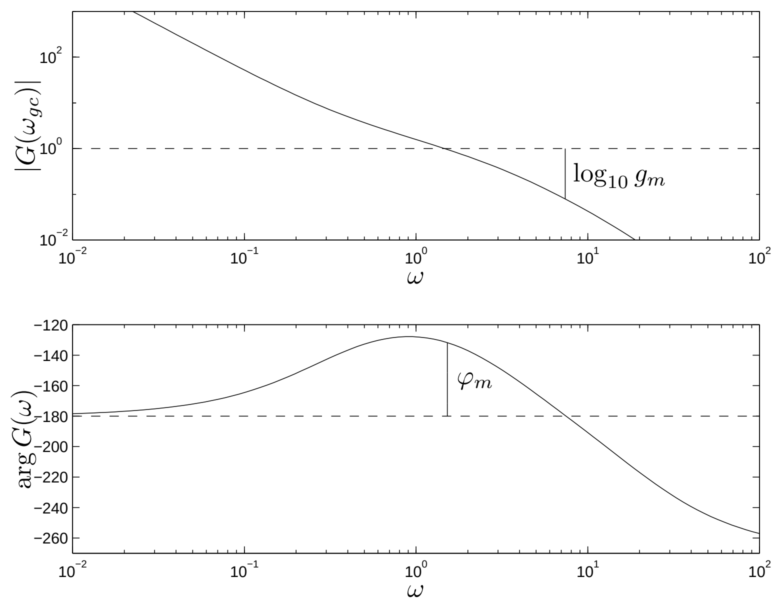

Figure 8.1: Bode plot of the loop transfer function $L(s) = \frac{100(s+0.316)}{s^2(s+3.16)(s+20)}$.

designs.

Proportional Compensation

Lead Compensation

Lead compensation is used to improve the stability margin of a system or to increase the crossover frequency. A lead compensator has the transfer function

$$G(s) = \frac{s+b}{s/N+b} \tag{8.1}$$

where $N > 1$. This transfer function has unit gain for low frequencies, $\omega << b$, and the gain N at high frequencies $\omega >> b$. This compensator has its largest gain, $\max_\omega |G(i\omega)| = N$, at high frequencies. The largest phase lead is given by

$$\varphi = \max_\omega \arg G(i\omega) = \arctan \frac{N-1}{2\sqrt{N}},$$

which is obtained for $\omega = \omega_m = a\sqrt{N}$. The gain at ω_m is $|G(i\omega_m)| = \sqrt{N}$. The achievable phase lead increases with N but it is less than $90°$. Phase

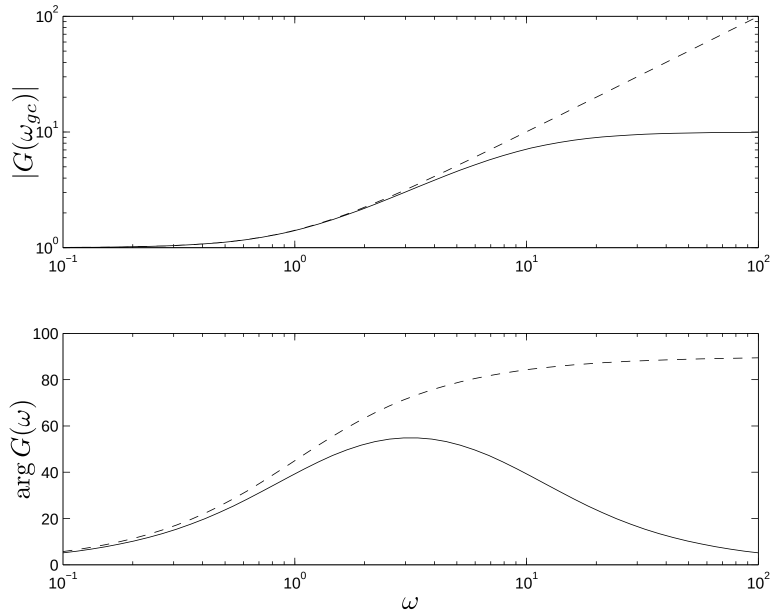

Figure 8.2: Bode plot of the transfer function $G(s) = \frac{s+b}{s/N+b}$ of a lead compensator with $b = 1$ and $N = 10$ (full lines) and $N = \infty$ (dashed lines) which is a PD controller.

leads 20°, 30°, 40°, 45°, 50° and 60° correspond to $N = 2$, 3, 4.6, 5.8 7.5 and 13.9. A Bode plot of the compensator is given in Figure 8.2 The phase of the compensator is always positive, which explains the name lead compensator. We illustrate the used of lead compensation by an example.

Example 30 *(Lead Compensation).* To illustrate lead compensation we consider a process with the transfer function

$$\bar{P}(s) = \frac{250000}{s^2 + 2500s + 250000}$$

which represents the scaled transfer function of a MEMS accelerometer. The process has a bandwidth of 500 rad/s and very low damping $\zeta = 0.005$. It is desired to have a closed loop system with a bandwidth of 5 krad/s and good damping. To simplify the writing we first change the frequency scale to krad/s instead of rad/s and the process transfer function then becomes

$$\bar{P}(s) = \frac{0.25}{s^2 + 0.0025s + 0.25}$$

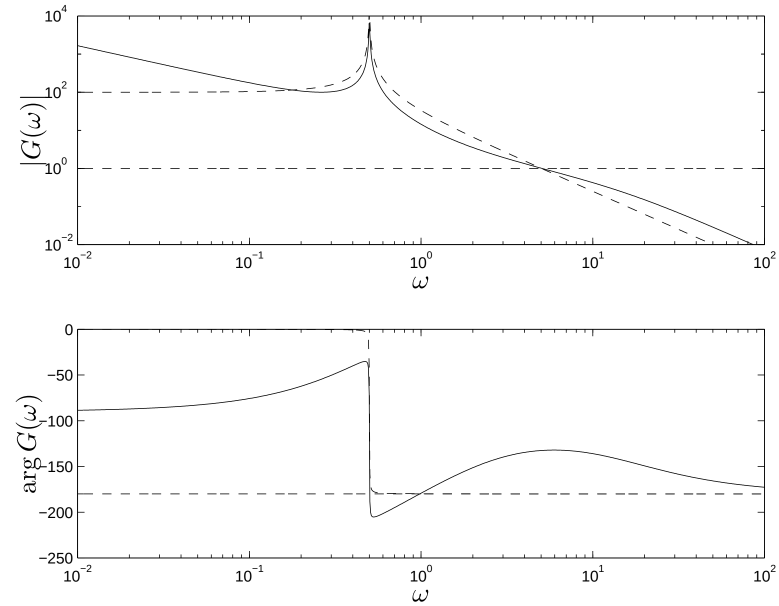

Figure 8.3: Bode plot of the loop transfer function $L(s) = \frac{25}{s^2+0.025s+0.25}$ (dashed lines) and the compensated loop transfer function $L_c(s) = \frac{25(3s+5)}{(s+15)(s^2+0.0025s+0.25)}$ (full lines).

To obtain the desired bandwidth the crossover frequency should be around 5 which means that the process gain has to be about 25. The Bode plot of the loop transfer function with a gain of 25 is shown in Figure 8.3. The figure shows that the phase margin is very small. To improve the phase margin we introduce a lead compensator. To have reasonable damping we choose $N = 9$ which gives a phase lead of 53°. Adjusting the gain so that the the gain of the compensator is 1 at the crossover frequency $\omega_{gc} = 5$ we find that the compensator is

$$G(s) = \frac{1}{3}\frac{s+5/3}{s/9+5/3} = \frac{3s+5}{s+15}$$

and the compensated loop transfer function becomes'

$$L(s) = \frac{25(3s+5)}{(s+15)(s^2+0.0025s+0.25)}$$

A Bode plot of the loop transfer function is given in Figure **??**.

Additional compensation mechanisms*

8.3 PID Compensation

The PID controller is the most common form of feedback. It was an essential element of early governors and it became the standard tool when process control emerged in the 1940s. In process control today, more than 95% of the control loops are of PID type, most loops are actually PI control. PID controllers are today found in all areas where control is used. The controllers come in many different forms. There are stand-alone systems in boxes for one or a few loops, which are manufactured by the hundred thousands yearly. PID control is an important ingredient of a distributed control system. The controllers are also embedded in many special-purpose control systems. PID control is often combined with logic, sequential functions, selectors, and simple function blocks to build the complicated automation systems used for energy production, transportation, and manufacturing. Many sophisticated control strategies, such as model predictive control, are also organized hierarchically. PID control is used at the lowest level; the multivariable controller gives the setpoints to the controllers at the lower level. The PID controller can thus be said to be the "bread and butter of control engineering. It is an important component in every control engineer's tool box.

PID controllers have survived many changes in technology, from mechanics and pneumatics to microprocessors via electronic tubes, transistors, integrated circuits. The microprocessor has had a dramatic influence on the PID controller. Practically all PID controllers made today are based on microprocessors. This has given opportunities to provide additional features like automatic tuning, gain scheduling, and continuous adaptation.

In this section we give a brief introduction to this common class of compensator. A more detailed description of the design of PID compensators can be found in (Astrom, 2003).

We will start by summarizing the key features of the PID controller. The "textbook" version of the PID algorithm is described by:

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\,\frac{de(t)}{dt}\right) \tag{8.2}$$

where y is the measured process variable, r the reference variable, u is the control signal and e is the control error ($e = y_{sp} - y$). The reference variable is often called the set point. The control signal is thus a sum of three terms: the P-term (which is proportional to the error), the I-term

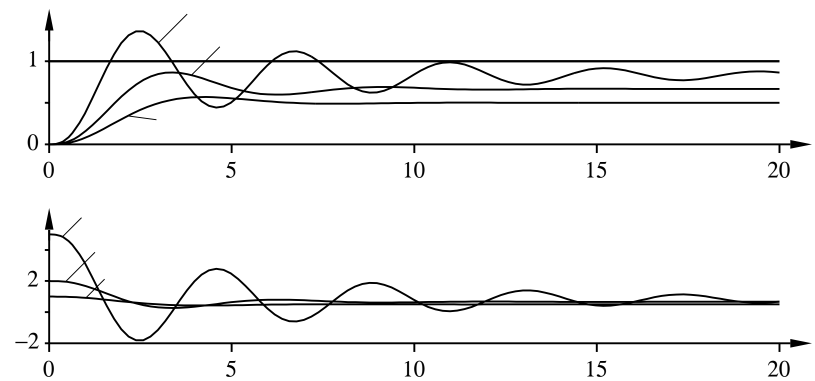

Figure 8.4: Simulation of a closed-loop system with proportional control. The process transfer function is $P(s) = 1/(s+1)^3$.

(which is proportional to the integral of the error), and the D-term (which is proportional to the derivative of the error). The controller parameters are proportional gain K, integral time T_i, and derivative time T_d. The integral, proportional and derivative part can be interpreted as control actions based on the past, the present and the future as is illustrated in Figure **??**. The derivative part can also be interpreted as prediction by linear extrapolation as is illustrated in Figure **??**. The action of the different terms can be illustrated by the following figures which show the response to step changes in the reference value in a typical case.

Proportional control is illustrated in Figure 8.4. The controller is given by (8.2) with $T_i = \infty$ and $T_d = 0$. The figure shows that there is always a steady state error in proportional control. The error will decrease with increasing gain, but the tendency towards oscillation will also increase.

Figure 8.5 illustrates the effects of adding integral. It follows from (8.2) that the strength of integral action increases with decreasing integral time T_i. The figure shows that the steady state error disappears when integral action is used. Compare with the discussion of the "magic of integral action" in Section Section **??**. The tendency for oscillation also increases with decreasing T_i.

Figure 8.6 illustrates the effects of adding derivative action. The parameters K and T_i are chosen so that the closed-loop system is oscillatory. Damping increases with increasing derivative time, but decreases again when derivative time becomes too large. Recall that derivative action can be in-

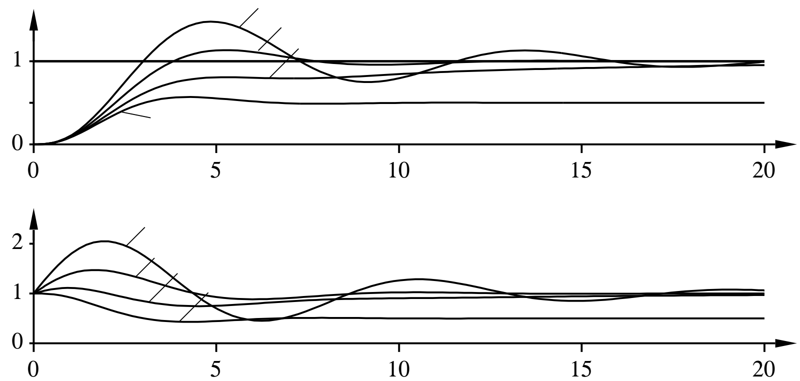

Figure 8.5: Simulation of a closed-loop system with proportional and integral control. The process transfer function is $P(s) = 1/(s+1)^3$, and the controller gain is $K = 1$.

terpreted as providing prediction by linear extrapolation over the time T_d. Using this interpretation it is easy to understand that derivative action does not help if the prediction time T_d is too large. In Figure 8.6 the period of oscillation is about 6 s for the system without derivative action. Derivative actions ceases to be effective when T_d is larger than a 1 s (one sixth of the period). Also notice that the period of oscillation increases when derivative time is increased.

8.4 Design Considerations

Reference tracking

Consider the standard unity gain feedback loop, shown in Figure **??**. The transfer function between the reference input r and the output error e is given by

$$H_{er}(s) = \frac{1}{1 + PC}.$$

This transfer function is known as the *sensitivity function*:

$$S = \frac{1}{1 + PC} = \frac{1}{1 + L}. \tag{8.3}$$

This transfer function will be very useful for specifying performance and so we investigate some of its properties.

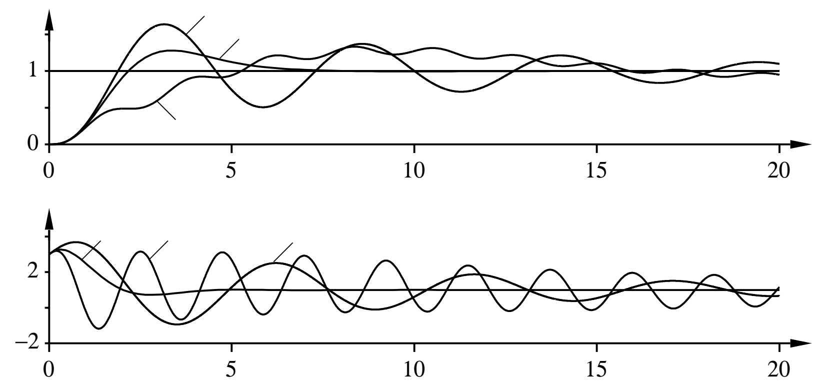

Figure 8.6: Simulation of a closed-loop system with proportional, integral and derivative control. The process transfer function is $P(s) = 1/(s+1)^3$, the controller gain is $K = 3$, and the integral time is $T_i = 2$.

A good control design should make the sensitivity function small, so that the error is close to zero. However, for most plants $P(j\omega) \to 0$ as $\omega \to 0$ and if our controller has finite gain at high frequencies, then $S(j\omega) \to 0$ as well. Hence it is generally not possible to have small tracking error over all possible frequencies.

Noise rejection

Another transfer function of common interest is the transfer function between the sensor noise n and the output y:

$$H_{yn} = \frac{PC}{1+PC}.$$

This transfer function is known as the *complementary sensitivity function*:

$$T = \frac{PC}{1+PC} = \frac{L}{1+L} = 1 - S. \tag{8.4}$$

The term complementary sensitivity function comes from the last relationship in the above equation.

A good control design should make the complementary sensitivity function small, so that measurement noise down not affect the output. However, we note that since $S + T = 1$, it is not possible to simultaneously make the tracking error small and reject sensor noise at the same frequency. The reason for this is simple: we cannot distinguish sensor noise from changes in the reference signal (they are added to together). Hence, it is important that we design our system so that the sensor noise is small in the frequency range in which we wish to track our reference signal.

The complementary sensitivity function is typically used to shape the response of the system at high frequency, where we cannot achieve tracking of the input. We note that if $L \ll 1$ is small at high frequency, then $T \ll 1$ and hence sensor noise is attenuated.

The Gang of Four

We have seen that the sensitivity function and the complementary sensitivity function naturally arise from looking at selected input/output pairs in the basic feedback loop. The two other transfer functions of interest relate the measurement noise to the control signal, H_{un}, and the process disturbances to the output error, H_{yd}. These transfer functions can be written in terms of the process and controller transfer functions as

$$H_{un} = \frac{C}{1+PC} = CS \qquad H_{yd} = \frac{P}{1+PC} = PS.$$

The transfer functions have many interesting properties that will be discussed in then following. A good insight into these properties are essential for understanding feedback systems. The load disturbance sensitivity function is sometimes called the input sensitivity function and the noise sensitivity function is sometimes called the output sensitivity function.

8.5 Further Reading