Systems Engineering and Process Control

Lecture Notes

2016

Contents

1. Introduction

1.1 Why Automatic Control?

Automatic control is very important in process industry. It is needed to operate the processes such that energy and raw materials are utilized in the most economical and efficient ways. At the same time, it is necessary that the produced products fulfill the specifications, and that the processes operate in a safe way.

In a feedback control system, the available manipulated variables are used to fulfill the specifications on the process output, despite the influence of disturbances, see Figure 1.1. The process variables that are measurable are called *process outputs* or *measurement signals*. The variables that are available for manipulation are called *process inputs*, *manipulated variables* or *control signals*. The desired performance of the system or plant can be specified in terms of the desired value of an output. This is called the *setpoint* or *reference value*. The performance of the system is also influenced by measurable or (most commonly) unmeasurable *disturbances*. The task of the *controller* is to determine the process inputs such that the performance of the total system is as good as possible. The performance can be measured in many different ways. For instance, we may want to minimize the variation in the output or minimize the energy consumption.

In our daily life we are surrounded by control systems. Some examples are:

- Heating and cooling systems in buildings.
- Temperature and concentration control within the human body.
- Frequency and voltage control in power networks.
- Speed and path following when riding bikes or driving cars.

A process control system in a plant typically has many different purposes:

- *Product specifications*. The outgoing product must fulfill the specifications with respect to, for instance, quality or purity.
- *Safety*. The plant must be operated such that the safety regulations for personnel and equipment are met.
- *Operational constraints*. Due to environmental and process considerations it is necessary to keep certain variables within tight bounds.

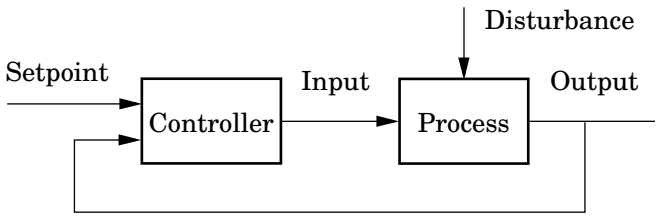


Figure 1.1 A simple feedback control system.

- *Economy*. The control system must ensure that energy and raw materials are used as economically as possible.

These tasks can be seen as plant-wide goals that must be fulfilled. To do so it is necessary to divide the operation of the plant into subtasks and subgoals. These are then further subdivided until we reach the bottom line, which consists of simple control loops with few inputs and outputs and with quite well defined goals. The basic building blocks in a process control system are simple feedback controllers. The purpose of this text is to develop an understanding for how these building blocks can be used to create a control system for a large plant.

1.2 Basic Principles of Automatic Control

Feedback

The main principle of automatic control is *feedback*, which implies that the measured signals are used in real time to calculate the control signals, i.e., that information is continuously fed back into the process. This turns out to be a very important and useful concept. The idea of feedback control is illustrated in an simple example.

EXAMPLE 1.1—THERMOSTAT
A simplified picture of a thermostat is shown in Figure 1.2. The purpose of the thermostat is to keep the temperature of the water constant. The temperature of the water is measured using a thermocouple. The thermocouple gives an electrical signal that is compared with a signal that represents the desired temperature. The comparison is made in a differential amplifier. This gives an error signal that is used to control the power of the heater. The heater can be controlled by a relay or contactor in which case the power is switched on or off. To obtain a better control the heater can be controlled by a thyristor. It is then possible to make a continuous change in the power. The thermostat is an example of a feedback system, where the control signal is determined through feedback from the measured output. The feedback is negative since an increase in the water temperature results in a decrease in the power. □

The introduction of feedback in the control system can often lead to dramatic performance improvements. Unstable systems can be stabilized, the speed of the system

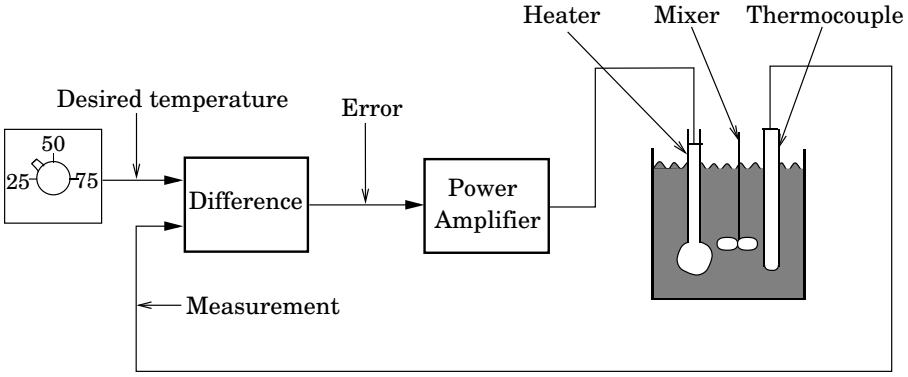


Figure 1.2 A simplified diagram of a thermostat for heating of water.

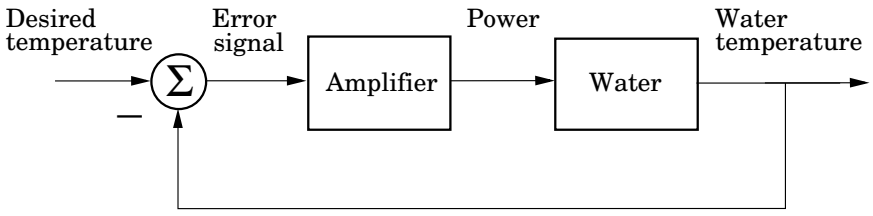


Figure 1.3 Block diagram of the thermostat in Figure 1.2.

can be increased, and the tracking accuracy can often be improved by several orders of magnitude. Since the system operates in *closed loop*, only a rough model of the process is needed. Disturbances can be compensated for once their effect have become visible in the process output.

Block Diagrams

To describe processes from a control point of view we often use *block diagrams*. Figure 1.3 shows a block diagram for the thermostat system. The block diagram is built up by rectangles and circles, which are connected by lines. The lines represents *signals* that carry *information*. The rectangles or blocks show how the signals influence each other. The arrows show the cause-effect direction. The circle with the summation symbol shows how the error signal is obtained by taking the difference between the desired temperature and the measured temperature. Notice that one symbol in the block diagram can represent several parts of the process. This gives a good way to compress information about a process. The block diagram gives an abstract representation of the process and shows the *signal flow* or *information flow* in the process.

Feedforward

Feedforward is a complementary principle to feedback. A very simple controller, like an automatic door opener, can often operate using pure feedforward. The principle is illustrated in in the block diagram in Figure 1.4. The controller calculates the control signal based on the setpoint and a model of the process. The resulting output is not measured, meaning that the system operates in *open loop*. It is obvious that such an approach cannot be used for unstable systems. Even for stable systems, there will always be some error in the output due to model uncertainty and disturbances. If some process disturbances are measurable, they can be taken into account by the controller, as indicated by the dashed arrow. Again, this requires a model of how the disturbances affect the process.

Combined Feedback and Feedforward

A good controller typically uses both feedback and feedforward to achieve optimum performance, see Figure 1.5. Measurable disturbances and known changes in the

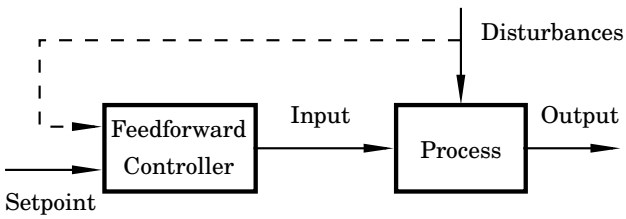


Figure 1.4 A feedforward control system.

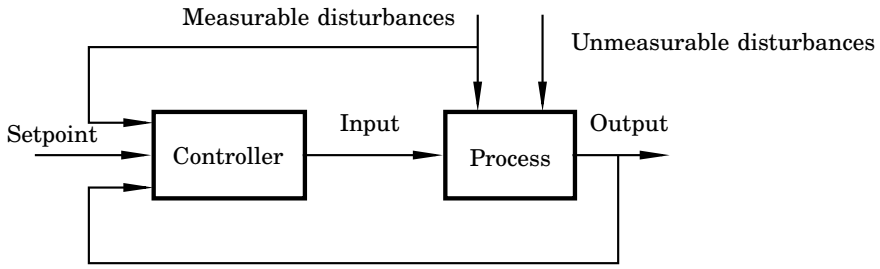


Figure 1.5 A combined feedforward–feedback control system.

setpoint may be compensated for using feedforward, while the unmeasurable disturbances are handled using feedback. Consider for instance the cruise control system in a car. Feedback is used to ensure that the speed of the vehicle is maintained at the desired value, despite the load of the vehicle and the changing road conditions. Feedforward can be added to achieve more comfortable acceleration and deacceleration during setpoint changes and to compensate ahead of time for changing road or traffic conditions using information from, e.g., the GPS system.

1.3 Graphical Process Representations

Process models are central to the analysis and design of feedback control systems. Process control systems can often seem complicated, since they contain a relatively large number of different parts. These parts may represent many different branches of technology, e.g., chemistry, biology, electronics, pneumatics, and hydraulics. To work effectively with control systems it is essential to master several different ways to describe the system and its elements. Graphical representations provide a qualitative impression of the system function. Many different graphical methods have been used within control for process description. We have already seen the block diagram. Three other common representations are described below.

General Process Layouts

A simplified process (or plant) sketch is often an effective means to provide an overview of a process to be controlled. This sketch should show the different process (or plant) components; preferably in a stylized form to provide clarity. A schematic layout of a simple process is shown in Figure 1.6. A layout of a more complicated plant site is shown in Figure 1.7. Note that there are no standards or rules for these drawings. However, good pictures are easily identified.

Process Flow Sheets

Within the process industry a limited number of unit operations and reactor types are used. Therefore a set of almost international standard symbols have been introduced for the different components. Similar standards have been introduced by the power generation and the sanitation industries. A process flow sheet is illustrated in Figure 1.8.

Process and Instrumentation Diagrams

To illustrate the combined process and instrumentation system, a set of standard symbols have been proposed and is applied within the process and control industry. These diagrams are often referred to as P/I (process and instrumentation) diagrams.

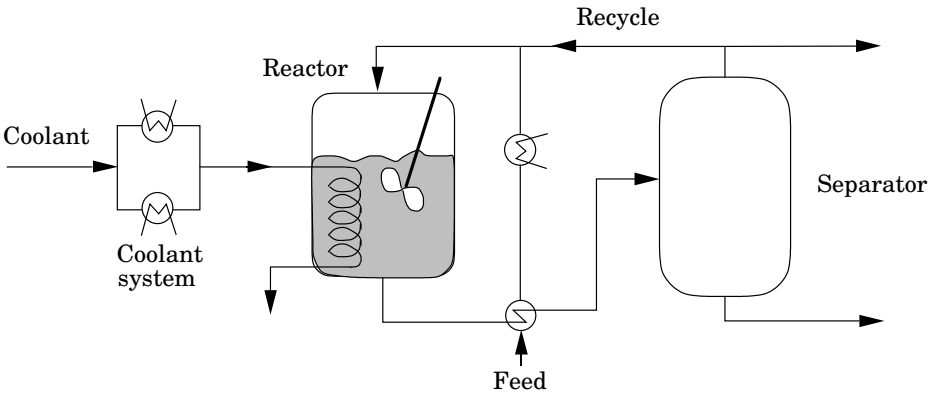


Figure 1.6 Schematic layout of a simple process.

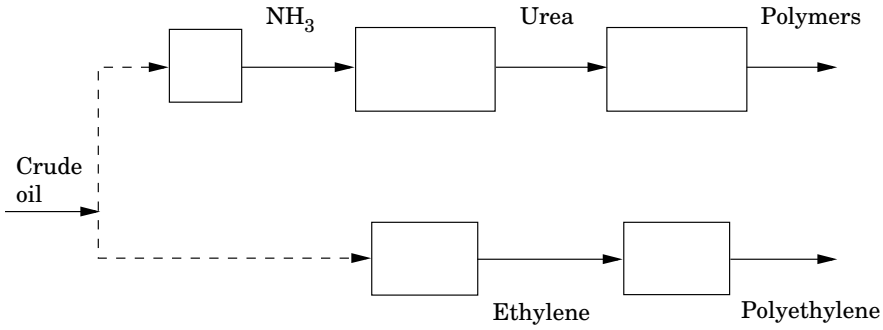


Figure 1.7 Schematic layout of a more complex plant site.

A simple example is shown in Figure 1.9. The P/I diagram is a factual representation of the combined process and instrumentation system hardware.

Instruments are depicted as circles in the P/I diagram. A two or three letter combination is used to denote the purpose of the instrument. The first letter denotes the physical quantity being instrumented. The most common quantities in process industry are T = temperature, L = level, F = flow, and P = pressure. Nonstandard quantities include C (or Q) = concentration and X = power. The last letter(s) denotes the instrument function(s). Here, T = transmitter (sensor) and C = controller are the most common functions. Other common functions are I = indicator, R = recorder, A = alarm, and V = valve.

1.4 Introduction to Dynamical Process Models

In the simple feedback control system depicted in Figure 1.1, it is the task of the controller to determine the process input such that the process output follows the setpoint. Different processes react differently to changes in the control signal. Some processes react very quickly, while other processes are slow. Further, the response may be well damped, oscillatory, or even unstable. These characteristics are called the *process dynamics*.

In order to design a good control system, we must learn the dynamical properties of the process. This can be done in several different ways. For simple control loops in

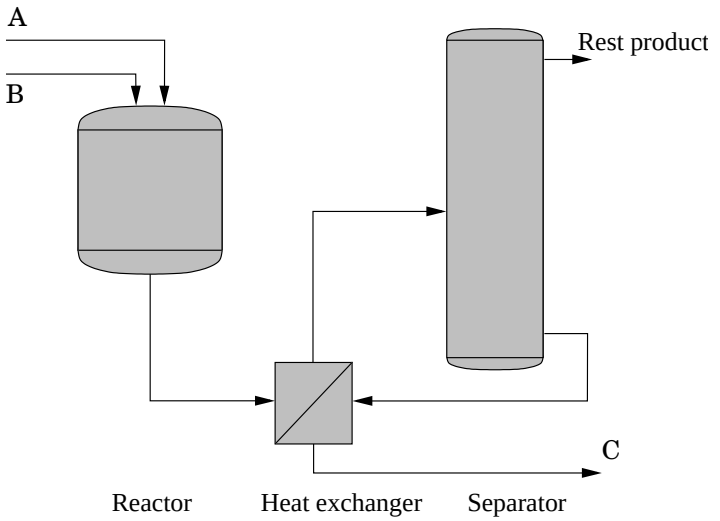


Figure 1.8 Process flowsheet for a simple plant with a reactor, heat exchanger, and a distillation column.

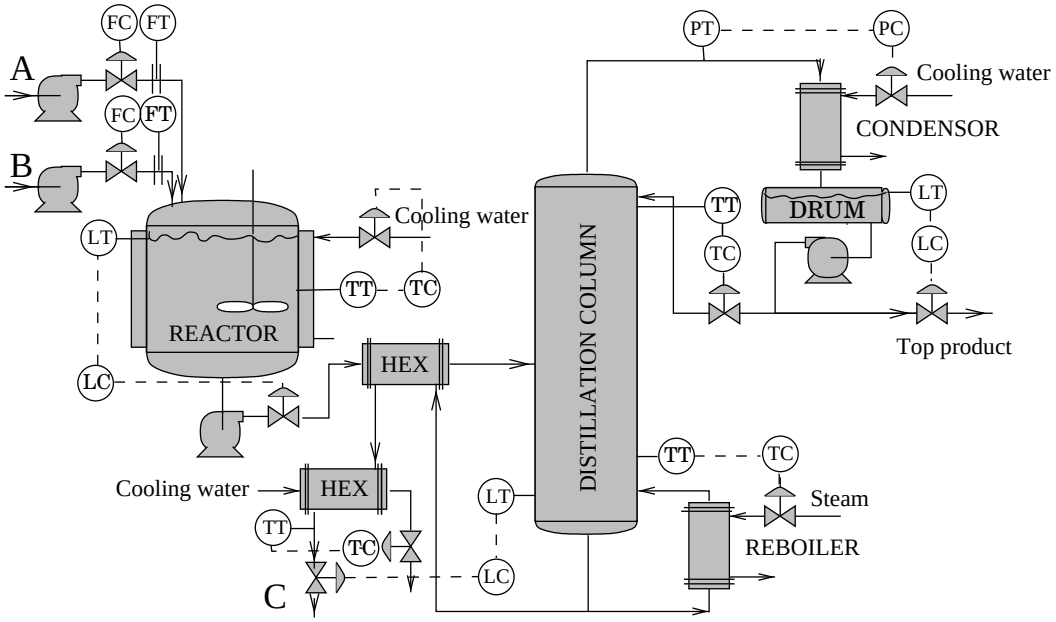


Figure 1.9 Process and instrumentation (P/I) diagram for the process in Figure 1.8. Signal connections are indicated with dashed lines.

process industry, it often suffices to perform some simple experiments. For more complex process parts, or for control loops with higher demands on performance, it may be necessary to derive a mathematical model of the process dynamics. This involves deriving the relationships between the physical quantities using mass balances, energy balances, etc. This method is sometimes the only one possible, for example when the process is not yet in existence.

The two methods outlined above are often combined. First, the structure and complexity of the model are determined from the physics of the plant. Then, the parameters of the dynamical model are found by experiments on the process.

In this course, we will work with several different dynamical process models. We will mainly work with continuous-time models. Simple *step-response models* based on experiments will be introduced in the next lecture. In Lecture 3, the state-space model is introduced as a general mathematical object to describe both linear and nonlinear process dynamics. A state-space model describes how the process inputs, the internal process state, and process outputs are related.

Often, the internal workings of the process are not so important from a control perspective. It can suffice to describe the dynamical relationship between the inputs and the outputs. Such models are called *input-output models*, and they are introduced in Lecture 4. The *transfer function* is a compact mathematical representation of the input-output dynamics for a linear process.

Sometimes it can give important insight to study processes and control systems in the frequency domain. There, the response of the system to sinusoidal input signals of different frequencies is characterized. Such *frequency-domain models* are introduced and analyzed in Lecture 8.

Some of the different dynamical process descriptions that we will encounter in the course are illustrated in Figure 1.10. The exact meaning of the various mathematical symbols will be revealed in the following chapters.

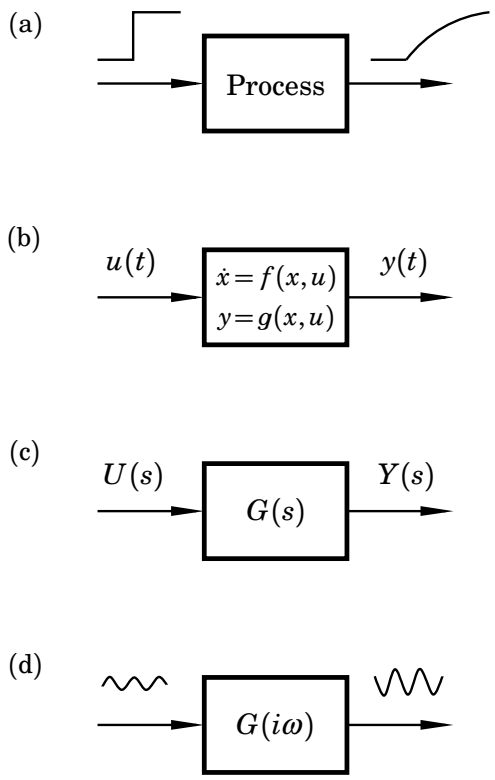


Figure 1.10 Illustration of some different dynamical process models that will be used in the course: (a) experimental step-response model, (b) nonlinear state-space model, (c) linear input-output model in the Laplace domain, (d) frequency-response model.

2. Simple Process Models. PID Control

2.1 Introduction

In this lecture we will study simple control loops; those which consist of a section of a process with feedback through a controller (see Figure 2.1). For the moment we shall disregard the surroundings and treat the simple control loops as our whole control problem. Our process section has a measured signal level or process variable y and a control signal u. The aim of the control process is for the process variable y to follow a setpoint r as closely as possible. In certain circumstances this means that we want the process variable to follow the setpoint when this changes quickly. This is the typical servo problem. It is not common in the field of process control for the setpoint to vary very often. Instead, it is variations in load which cause us the problems in getting y to follow r. Regardless of which particular type of problem we have, we solve it by the use of feedback. Feedback means that we compare the process variable y with the setpoint r, and use this comparison to determine what the control signal u will look like. This decision is taken inside the controller.

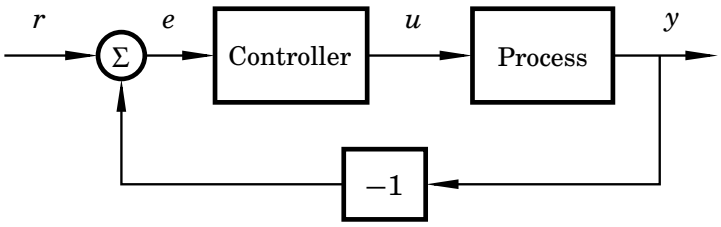


Figure 2.1 A simple feedback loop.

2.2 Step-Response Models

In order to obtain good closed-loop performance, the parameters of the controller should be adjusted in accordance with the dynamics of the process. A simple experimental way to obtain a rough model of the process dynamics is to perform a step response experiment. First, the process should be in steady state. The input u is then momentarily changed as a step of size Δu. The output response y is recorded and analyzed.

A typical step response for a well-behaved industrial process is shown in Figure 2.2. Some important characteristics of the process dynamics can be determined from the response:

- **Dead time**, L, is defined as the time it takes from a change in the control signal to the start of a reaction in the process variable.
- **Time constant**, T, is defined as the time taken for the process variable to reach $1 - e^{-1} \approx 63\%$ of its final value. Note that the dead time is not included

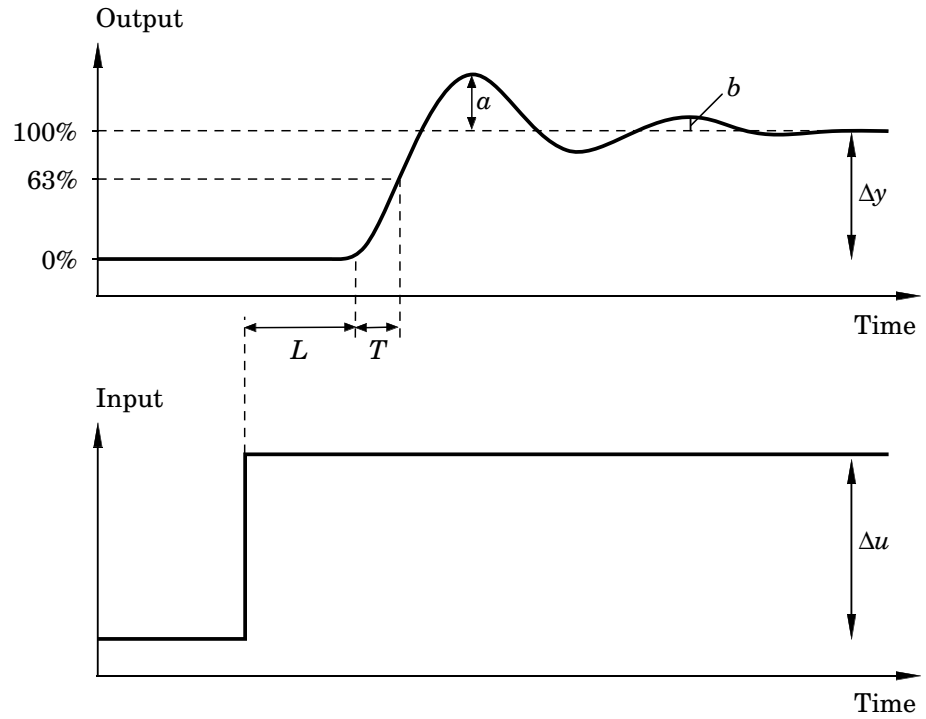


Figure 2.2 Typical step response with definitions of some process characteristics.

when determining this value. A step response usually consists of several time constants because most processes are multi-capacitive. The characteristic we specify in this way is the *dominant* time constant.

- **Static gain**, $K_p = \Delta y/\Delta u$, is a measure of how much the process variable changes in relationship to the change in the control signal when the signals have stabilized.
- **Overshoot**, $a/\Delta y$, is usually given as a percentage of the change in static process variable. Within the process industry, the overshoot is nearly always zero in open-loop, uncontrolled processes.
- **Damping**, here defined as $1 - b/a$, is a measure of the stability of the process. Many industrial processes are well damped before they are subjected to feedback control. In step-response analysis we therefore find that the damping is one in the majority of cases. A slightly different definition of damping will be given in Lecture 5.

A great advantage of step-response analysis is that it is simple to carry out. The input signal is simple to generate by setting the controller in manual mode and adjusting its control signal. In most cases this method requires a chart recorder in order to record the process variable, but it may be sufficient sometimes to just use a clock.

Step-response analysis is a useful aid when we want to tune a controller using a systematic method but cannot spend too much time on building the model. Systematic tuning rules of this type are described in Lecture 9.

If the process is of integrating type, i.e., if the process does not settle at a new stationary level after a step change in the control signal but continues to increase, the step-response analysis as described above is not directly usable. Instead, one usually determines only the dead time L and the *velocity gain*, $K_v = \Delta y/(\Delta u \cdot L)$, see Figure 2.3.

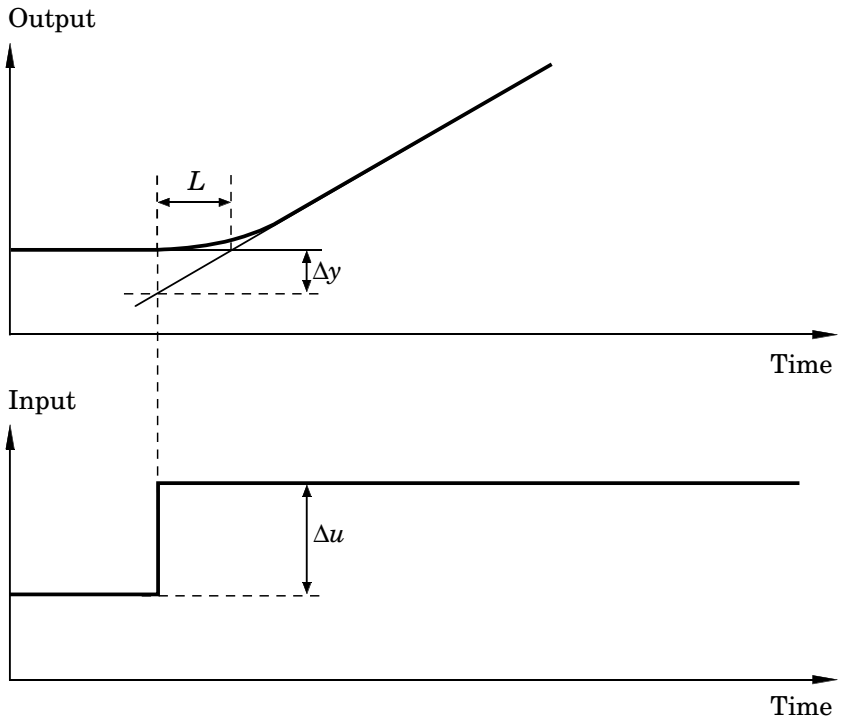


Figure 2.3 Step response for an integrating process.

2.3 Common Process Types

The dynamic characteristics of different sections of the process are often very different. They vary in their speed, amplification, stability, etc. In this section we shall attempt to group them into certain main types. This grouping or cataloging will make future description easier, as we shall define a number of concepts and types of process which later on we shall be using frequently.

It is important to stress what is included in the section of the process in simple control loops. The process section consists of everything apart from the controller, in other words all the dynamic components which sit between the output signal of the controller and its process variable or measured signal. This means that, e.g., transducers, valves, actuators and any module boxes are included in what we have called the process section.

Figure 2.4 shows the step response for six different types of processes. More detailed step-response analysis will be carried out using mathematical tools in Lecture 5. What this figure shows is quite simply the process variable after a step change has occurred in the control signal. We shall now go briefly through each type of process.

Single-Capacitive Processes

The single-capacitive process is called so because its step response has the same appearance as what we get from a simple electrical RC circuit. The time constant, which is defined in the next section, for an RC filter is $T = RC$ (see Figure 2.5).

This is the simplest type of process we come across. Mixing processes often display this behavior. Temperature processes with direct transfer between two volumes can also be described as single-capacitive processes.

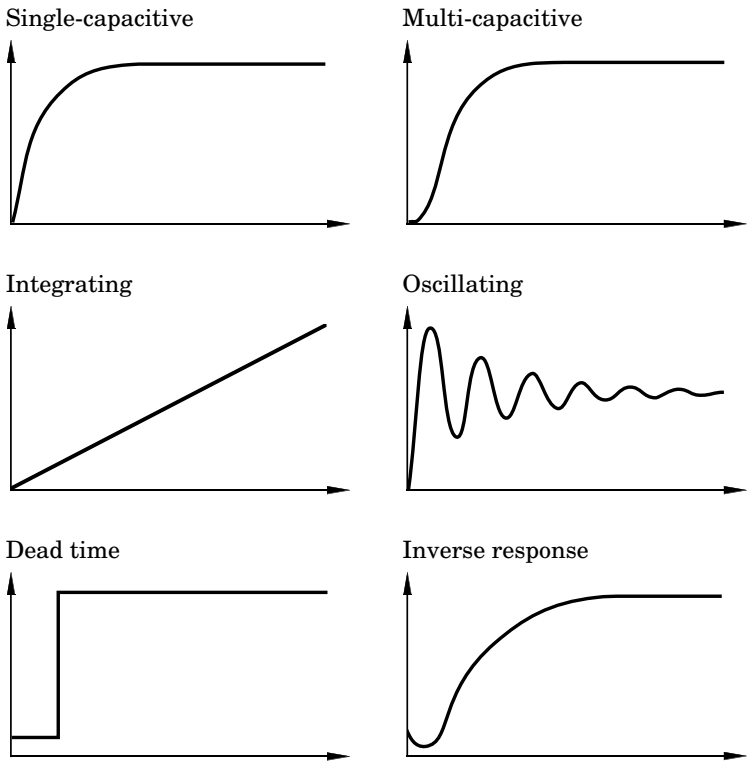


Figure 2.4 Step responses for some different types of processes.

Multi-Capacitive Processes

This is the most common type of process, often combined with the dead-time process described below. The step response is the same as one gets from several series-connected RC filters, hence the name.

Flow and pressure processes, together with certain temperature processes, can often be described in this way.

Integrating Processes

Most processes within industry are input–output stable. This means that if we change the control signal from one value to another, then the process variable will also stabilize at a new level.

There are also certain processes which are of the integrating type. These do not have the above characteristics. Some examples of this are level control, pressure control in a closed vessel, concentration control in batches, and temperature control

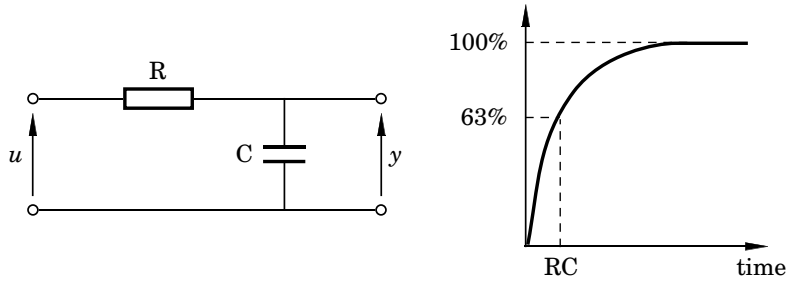


Figure 2.5 A simple RC circuit and its step response.

in batches. If, for example, a valve is opened to allow flow into a tank, the level will rise linearly assuming that the output flow does not change. The common factor in all these processes is that some form of storage occurs in them. In level, pressure and concentration control storage of mass occurs, while in the case of temperature control there is a storage of energy.

We shall see later in this text that there are large differences between stable and integrating types of processes with regard to modeling and control. It is therefore good to have a feeling for which types of processes are of the integrating type.

Oscillating Processes

This type of process is characterized by an oscillation of the step response around its final stationary value. This type of process is not so common among uncontrolled process sections within process industries. (On the other hand, it unfortunately is quite common in controlled processes.) One case where it occurs is in concentration control of recirculating fluids. In mechanical designs, however, it is common for processes to be oscillating where elastic materials are used, e.g. pliable axles in servos, spring constructions, etc.

Dead-Time Processes

This type of process is characterized by the lack of reaction of the process variable to a step change until a certain time (the dead time or delay time) has passed. It is seldom that the process consists only of a dead time response; the dynamics will usually also include one or more of the other process types.

Dead times occur most often with material transport in pipes or on belts. If we measure the pH, for example, of a fluid being transported in a pipe, where the addition of the substance of which we wish to measure the concentration occurs a long way upstream in relation to the transducer, a dead time will occur which corresponds to the time taken for the fluid to be transported between the supply point and the transducer.

This type of process can be difficult to control. In Lecture 10 we will describe a special control structure for control of dead-time processes.

Inverse Response Processes

This last type of process is not particularly common. It is characterized by the response to the step change starting in the “wrong” direction. It is not difficult to understand that such a process could pose great problems for the controller.

The most common inverse response process within the process industry is level control of the water level in boilers.

The six types of process mentioned above will naturally not cover all our processes. There are for instance static processes that do not exhibit any dynamics at all. A few processes are open-loop unstable. Also, we have not taken any form of non-linearity into account. The vast majority of our processes, however, can be described in gross terms by one of the above process types or combinations of them.

Of the processes above, the single capacitive and the integrating processes are the easiest to control. This is because the process variable reacts immediately to a change in the control signal. For the same reason, the dead-time process and the inverse response process are the most difficult, because at the beginning the process variable does not react at all, or reacts in the wrong direction.

2.4 The PID Controller

We shall now introduce the most commonly used controller—the PID controller. The controller operates in closed loop according to the simple feedback loop shown in Figure 2.1. We deduce the structure of the PID controller and show that it is a natural extension of the very simplest controller, namely the On/Off controller. More details about the PID controller will be given in Lecture 9.

On/Off Control

The On/Off controller is the simplest imaginable controller. Its control signal u is given by

$$u(t) = \begin{cases} u_{\max} & e(t) > 0 \\ u_{\min} & e(t) < 0 \end{cases}$$

where e is the control error, i.e. the difference between the setpoint r and the measurement signal y:

$$e = r - y$$

The function of the On/Off controller can also be described graphically, as shown in Figure 2.6.

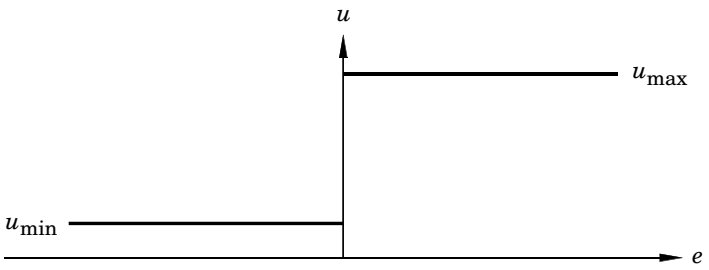


Figure 2.6 The control signal of the On/Off controller.

A drawback with this controller is that it gives rise to oscillations in the control loop. In order for the controller to maintain a small difference between measurement signal and setpoint, it must constantly switch the control signal between the two levels $u_{\max}$ and $u_{\min}$. If we for instance control the speed of a car by means of the gas pedal, while it can only take on the values “no gas” and “full gas”, we will need to switch between these two values in order to keep the average speed at the setpoint. This is one way of driving, but it will not be very comfortable.

Proportional Control

For large control errors it can be reasonable to apply either the maximum or minimum control signal. Consequently, the On/Off controller performs well for large errors. The oscillations appear for small control errors and can be reduced by e.g. decreasing the controller gain for small control errors. This can be achieved by introducing a proportional band, giving a P controller. The control signal of the P controller is given by

$$u(t) = \begin{cases} u_{\max} & e(t) > e_0 \\ Ke(t) + u_0 & -e_0 \le e(t) \le e_0 \\ u_{\min} & e(t) < -e_0 \end{cases}$$

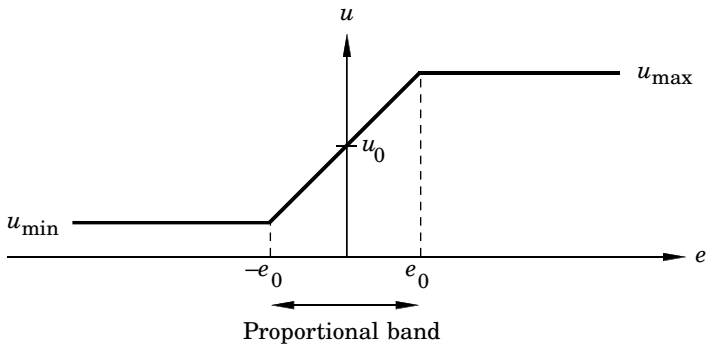


Figure 2.7 The control signal of the P controller.

where u_0 is the control signal corresponding to a zero control error and K is the gain of the controller. The P controller can also be described graphically, as shown in Figure 2.7.

The output of the P controller corresponds to that of the On/Off controller for large control errors. For control errors of magnitude less than e_0, the control signal is, however, proportional to the control error.

The P controller removes the oscillations, which were present during on/ off control. Unfortunately this comes at a price. We are no longer granted a zero stationary error, or in other words, that the setpoint and measurement signal coincide when all signals in the control loop have reached constant values. This is easily realized by studying the control signal. For small control errors, the P controller works within its proportional band. The control error is then given by

$$e = \frac{u - u_0}{K}$$

In stationarity the control error becomes zero if and only if at least one of the following criteria are fulfilled:

1. K is infinitely large
2. $u_0 = u$

Option number one, an infinite controller gain or a zero proportional band, is equivalent to On/Off control. This alternative is therefore not a good solution, since it leaves us with the initial oscillation problem. We are hence referred to option number two, in order to eliminate the stationary control error. Here we can only eliminate the stationary control error if we can find a value of u_0, which makes it equal to the control signal u for all values of the setpoint r.

From the expression for the control error of the P controller we see that a higher controller gain K leads to a smaller control error. We also see that we minimize the maximal stationary control error by choosing u_0 in the center of the working range of the control signal. In most controllers, u_0 is consequently preset to $u_0 = 50\%$. In some controllers, it is possible to adjust the value of u_0. From the above discussion we see that u_0 should be chosen as close to the stationary value of u as possible.

Proportional–Integral Control

Rather than letting u_0 be a constant parameter, one can choose to adjust it automatically in order to achieve $u_0 = u$ when all signals in the control loop have reached

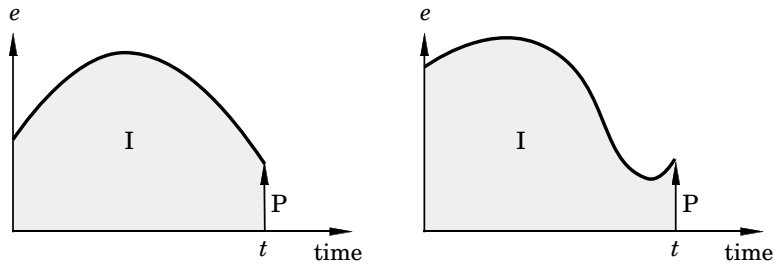


Figure 2.8 Two control cases where the output from a PI controller are equal at time t.

constant values. This would eliminate the residual control error and is exactly what the integral part (I part) of a PI controller does. The control signal of a PI controller is given by

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\right)$$

where T_i is the integral time of the controller. The constant level u_0 of the P controller has thus been replaced by the term

$$u_0(t) = \frac{K}{T_i}\int_0^t e(\tau)d\tau$$

which is proportional to the *integral* of the control error. This is why the term is called the integral term or the integral part of the PID controller.

One can be convinced that the PI controller has the ability to eliminate residual control errors by studying the above control law. Assume that we have a stationary control error $e \neq 0$ despite the use of a PI controller. If the control error e is constant, the proportional part in the PI controller will also hold a constant value Ke. The integral part will, however, not be constant. It will increase or decrease, depending on the sign of the control error. If the control signal is changed, the measurement signal y of the process must sooner or later increase or decrease. Consequently, the error $e = r - y$ cannot be constant. Since this conflicts with the assumption of a stationary error, we have showed that we cannot have a non-zero stationary error, when the controller contains an integral part. The only occasion when all signals internal to the controller can be stationary, is when $e = 0$.

We have now showed that the PI controller solves the problem of a residual stationary error and that of oscillations resulting from On/Off control. The PI controller is therefore a controller without any substantial shortcomings. It is generally sufficient when performance requirements are not extensive. Consequently, the PI controller is the by far most commonly used controller in industrial applications.

Proportional–Integral–Derivative Control

One characteristic which limits the performance of the PI controller is that it only takes past and present control errors into account; it does not try to predict the future evolution of the control error. The problem is illustrated in Figure 2.8.

The two curves in Figure 2.8 show the evolution of the control error in the two cases. The P part of the controller is proportional to the control error at the present time instance t. This control error is equal for both figures. The integral part is proportional to the surface delimited by the control error curve. This implies that a PI controller yields the same control signal at time t for the two cases. An intelligent controller should, however, see that there is a big difference between the cases. In the left

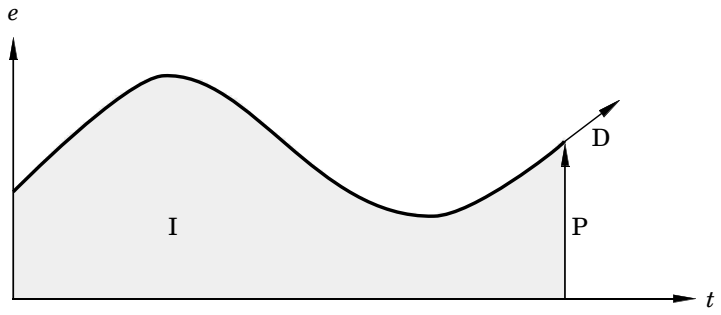


Figure 2.9 Illustration of PID control. The integral part is proportional to the surface under the control error curve, the proportional part is proportional to the current control error and the derivative part is proportional to the change rate of the control error.

curve, the control error decreases rapidly and the control action should be deliberate, in order not to cause an overshoot. In the right curve, a decrease in the control error is followed by a sudden increase. Here, the controller should apply a large control signal in order to curtail the control error. The derivative part of the PID controller accomplishes exactly this type of compensation. It is proportional to the change rate of the control error, i.e. proportional to the time derivative of the error. The equation of the PID controller is given by

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

where T_d is the derivative time of the controller.

The maximal benefit of the D part is obtained in cases where much can be earned by predicting the control error. This is the case for many temperature control applications. Due to the inertia of these systems it is necessary to abort heating in time. Slow heat conduction can otherwise result in rising temperatures, long after the seize of heating.

Everybody who has used a thick-bottomed pan for broiling has experienced this phenomenon. It can take quite a while from the time instance when one turns down the temperature control knob, until the temperature in the pan actually begins to decrease. In the meanwhile the temperature can be subject to a significant overshoot if one is not careful with the temperature control.

The PID controller can be summarized by means of Figure 2.9. The proportional part provides the control signal with a contribution proportional to the current control error. The integral part is the memory of the PID controller. It is proportional to a weighted sum of all past control errors. Lastly, the derivative part tries to predict future control errors using the derivative of the current control error.

2.5 Choice of Controller Type

After having studied the structure of the PID controller and the function of its three parts, we can now decide when to use one or more parts of it.

On/Off Controller

The simplest type of controller, the On/Off controller, has, as we have seen, the great disadvantage that it gives rise to oscillations in the process variable. However, it also

has great advantages. It is cheap to manufacture and does not require the adjustment of any controller parameters. It is therefore useful in processes where oscillations are not too much of a disadvantage, where there is a requirement for an inexpensive design, and where we do not want problems with selecting controller parameters. The result is that we see this type of controller in our simplest household appliances such as ovens, fridges, stoves, irons etc.

P Controller

In certain types of process we can work with a high gain in the controller without having problems of stability. Many single-capacitive and integrating processes fall into this category. We have seen earlier that a high gain in a P controller means that the remaining control error will be small. We therefore do not need any integral part in these control examples, if we can accept the remaining control error. We can often accept this small control error in e.g. level control, because we are often working here with a surge tank where the exact level is not important; what is important is that the level remains within certain limits, such as the working range of the level transducer. Another instance where it often does not matter whether we have a remaining control error is in the inner loop in a cascade control structure (see Lecture 10). Here it is also satisfactory just to use a P controller.

The D part is not required either if our process does not have high levels of inertia with large lag (multi-capacitive processes), or if we do not place high demands on the control performance. Integrating and single-capacitive processes respond immediately to a change in the control signal. This can be seen clearly in Figure 2.4. For these types of processes, therefore, it is not necessary to predict the control error or to make prior compensation for it. The D part is therefore not necessary for these process types. Also, since the high controller gain, despite filtering, results in amplification of the noise which then appears in the control signal, it is usually wise not to use the derivative part for these processes.

PD Controller

A well-insulated thermal process has almost the same response as an integrator. Almost all the energy supplied is used to raise the temperature in the oven because energy losses are negligible. With these types of process, too, we can work with large gains in the controller, and we therefore often do not need any integral part in the controller. In a thermal process, thermal energy is stored instead of mass as in level control. In contrast to level control, thermal processes, in addition to acting as an integrator, often have other difficult dynamics which arise from heat transport within the materials. This means that it is seldom sufficient to use a P controller on its own, but we have to complement it with a derivative part. This derivative part allows us to stop the input of energy in time. In some thermal processes it is not sufficient even to use the derivative part; some of the corresponding second derivative of the temperature may also have to be used. This means that we are not just studying the changes in temperature via the derivative part, but also the acceleration of the temperature changes. The PD controller is sensitive to noise since it has a relatively high gain at high frequencies. One important reason why PD controllers work so well with thermal processes is that we can often obtain measurement signals from them with relatively low noise level.

PI Controller

This is the most common controller type in industry. In the derivation of the PID controller above, we saw that this was the simplest form which did not cause any

particular disadvantage such as oscillations or stationary control error. We can always manage without the D part if we do not have high demands on the speed of the control loop. Another case which was mentioned above was when the lag of the process is small (single-capacitive and integrating processes). Here the D part does not give any great improvement to the control but can rather cause trouble as it amplifies the noise.

Another case which is suitable for PI control is when we have long dead times. It is certainly in this type of process that we have the greatest need for predicting the future control error. However, trying to get a prediction of how the process variable will change in the immediate future from the derivative of the process variable is not a good method. Because of the dead time there is a delay before the effects of any control action can be seen on the process variable. It is therefore considerably better with this type of process to try to predict the future course of the process variable by studying the control signal combined with a model of the process. This is called dead time compensation, and will be studied in Lecture 10. If we do not have access to dead time compensation, it is better to use a PI controller than a PID controller.

A third case where we should disconnect the D part is when the process is disturbed by high levels of noise. As a first attempt, of course, we should try to filter out the noise, but sometimes this is not sufficient. The D part will then give a poor prediction and should be removed.

PID Controller

Adding the D part to the PI controller often gives better performance, especially for lag-dominated processes without long dead times, in other words on multi-capacitive processes. It allows us to increase both the P part and the I part while maintaining the same level of damping.

I Controller

A pure integrating controller is appropriate when the process dynamics can be approximated by a static system. This is the case when the process is stable and has a very short time constant or when the performance demands are very modest. In those cases, the proportional part is not needed to speed up the dynamics, but the integral part is still needed in order to eliminate the stationary error.

3. State-Space Models

3.1 Mathematical Modeling

To describe the dynamic behavior of processes, mathematical models are often derived using the extensive quantities, which, according to the laws of classical physics, should be conserved. The most important extensive quantities in process control are

- Total mass
- Component mass
- Energy
- Momentum

The variables representing the extensive quantities are called *state variables* since they represent the system state. The changes in the system or process state are determined by the conservation equations. The conservation principle for extensive quantity $\mathcal{K}$ may be written for a selected volume element V of a process phase as

$$\frac{\begin{matrix}\text{Accumulation}\\ \text{of } \mathcal{K} \text{ in } V\end{matrix}}{\text{unit time}} = \frac{\begin{matrix}\text{Flow of } \mathcal{K}\\ \text{into } V\end{matrix}}{\text{unit time}} - \frac{\begin{matrix}\text{Flow of } \mathcal{K}\\ \text{out of } V\end{matrix}}{\text{unit time}} + \frac{\begin{matrix}\text{Generation}\\ \text{of } \mathcal{K} \text{ in } V\end{matrix}}{\text{unit time}} - \frac{\begin{matrix}\text{Consumption}\\ \text{of } \mathcal{K} \text{ in } V\end{matrix}}{\text{unit time}} \tag{3.1}$$

The balance must be formulated for each of the relevant phases of the process, and expressions for the quantity transfer rates between the phases must be formulated and included in the flow terms. Note that by convention an inflow of quantity into a phase is positive on the right hand side of (3.1). The type of differential equation resulting from application of the conservation principle depends upon the *a priori* assumptions. If the particular phase can be assumed well mixed such that there are no spatial gradients and the quantity is scalar (e.g. temperature or concentration) then the conservation balance may be formulated for the whole volume occupied by the phase. In that case an ordinary differential equation results. Other types of models, i.e. where spatial gradients are essential, yield partial differential equations.

Some of the most commonly applied conservation balances for the case of ideal mixing are listed below. Note that the momentum balance usually plays no role in this case.

Total mass balance:

$$\frac{d\rho V}{dt} = \sum_{\substack{i=\text{all}\\ \text{inlets}}} \rho_i q_i - \sum_{\substack{i=\text{all}\\ \text{outlets}}} \rho_i q_i$$

Component j mass balance:

$$\frac{dc_j V}{dt} = \sum_{\substack{i=\text{all}\\ \text{inlets}}} c_{j,i} q_i - \sum_{\substack{i=\text{all}\\ \text{outlets}}} c_{j,i} q_i + r_j V$$

Total energy balance:

$$\frac{dE}{dt} = \sum_{\substack{i=\text{all}\\ \text{inlets}}} \rho_i V_i H_i - \sum_{\substack{i=\text{all}\\ \text{outlets}}} \rho_i V_i H_i + \sum_{\substack{\text{all phase}\\ \text{boundaries}}} Q_k + W$$

Here,

c = molar concentration (moles/unit volume)
E = total energy
q = volume flow rate
H = enthalpy in per unit mass
Q = net heat received from adjacent phase
r = net reaction rate (production)
ρ = mass density
V = volume
W = net work done on the phase

Mathematical modeling of process dynamics will now be illustrated in an example.

EXAMPLE 3.1—CONTINUOUS STIRRED TANK REACTOR
Consider the continuous stirred tank reactor (CSTR) system in Figure 3.1. An exothermic reaction A → B takes place in the reactor. A cooling coil is used to maintain the reaction mixture at the desired operating temperature by removing heat that is released in the exothermic reaction. The fundamental dependent quantities for the reactor are:

1. The total mass of the reacting mixture in the tank
2. Mass of chemical A in the reacting mixture
3. Total energy of the reacting mixture in the tank

We will assume that the CSTR is perfectly mixed, that the mass densities of the feed and the product streams are both equal to ρ, and that the reaction rate is given by the Arrhenius expression

$$r = k_0 e^{-E/RT} c_A$$

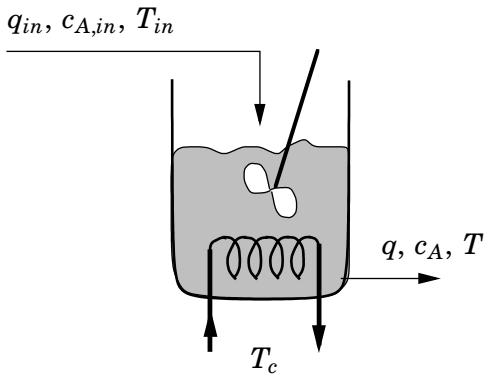


Figure 3.1 Continuous stirred tank reactor with cooling coil

where E is the activation energy, R is the ideal gas constant, and k_0 is a kinetic constant.

Further, we will assume that C_p is the specific heat capacity of the reacting mixture, $(-\Delta H_r)$ is the heat of reaction, U is the overall heat transfer coefficient, A is the total area of heat transfer, and T_c is the temperature of the cooling fluid. We can then state the following balance equations:

1. Total mass balance:
$$\frac{d(\rho V)}{dt} = \rho q_{in} - \rho q$$

2. Mass balance on component A:
$$\frac{d(c_A V)}{dt} = c_{A,in} q_{in} - c_A q - rV$$

3. Total energy balance:
$$\rho V C_p \frac{dT}{dt} = \rho C_p q_{in}(T_{in} - T) + (-\Delta H_r) rV + UA(T_c - T)$$

Simplifying these expressions, the differential equations for the state variables V, c_A, and T become

$$\begin{aligned}
\frac{dV}{dt} &= q_{in} - q \\
\frac{dc_A}{dt} &= \frac{q_{in}}{V}(c_{A,in} - c_A) - k_0 e^{-E/RT} c_A \\
\frac{dT}{dt} &= \frac{q_{in}}{V}(T_{in} - T) + \frac{(-\Delta H_r) k_0}{\rho C_p} e^{-E/RT} c_A + \frac{UA}{V \rho C_p}(T_c - T)
\end{aligned} \tag{3.2}$$

The model is nonlinear and of third order since there are three state variables. In the model, q_{in}, q, $c_{A,in}$, T_{in}, and T_c can be viewed as input signals (manipulated variables or disturbances), while ρ, C_p, $(-\Delta H_r)$, k_0, E, R, U, and A are constants. □

3.2 State-Space Models

Modeling the dynamic process behavior using balance equations typically yields a set of ordinary differential equations. The resulting model may be written as a higher-order differential equation relating the input u and the output y, e.g.,

$$\frac{d^n y}{dt^n} + p_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + p_n y = q_0 \frac{d^m u}{dt^m} + q_1 \frac{d^{m-1} u}{dt^{m-1}} + \cdots + q_m u$$

The above model is linear, but we may also have nonlinear differential equations. This type of model is called an *input-output model* and will be the focus of the next lecture.

In a *state-space model*, we use a set of first-order differential equations to explicitly model how the n internal state variables $x_1, \ldots, x_n$ of the process evolve as functions

of the m process inputs $u_1, \ldots, u_m$ and of the state variables themselves:

$$\begin{aligned}
\frac{dx_1}{dt} &= f_1(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
\frac{dx_2}{dt} &= f_2(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
&\vdots \quad \vdots \\
\frac{dx_n}{dt} &= f_n(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m)
\end{aligned} \tag{3.3}$$

The parameter n is called the order of the system. The state-space model may be efficiently represented using vector notation, with column vectors for each of the process state variables x, the process inputs u, and for the right hand sides f as follows,

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} \quad f = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

thus giving the vector differential equation

$$\frac{dx}{dt} = f(x,u)$$

This equation is called the *state equation* of the system. Given the initial values of the state variables and the input signals it is in principle possible to predict the future values of the state by integrating the state equation.

The models for the p process outputs, which are assumed to be algebraic, may similarly be collected in a measurement vector

$$\begin{aligned}
y_1 &= h_1(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \\
&\vdots \quad \vdots \\
y_p &= h_p(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m)
\end{aligned}$$

Introducing vectors

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix} \quad h = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_p \end{pmatrix}$$

allows us to compactly write the *measurement equation* as

$$y = h(x,u)$$

Many of the processes that we are interested in are modeled by nonlinear differential equations. Such nonlinear state-space models can be simulated by computing its solutions numerically, using e.g. a simulation package such as MATLAB/Simulink. It is however well known that there is no general mathematical theory for the analytical solution of nonlinear equations. Only for linear differential equations are closed-form, analytic solutions available.

In a few rare cases, it is possible to transform a nonlinear system into a linear one by an appropriate change of variables. This is in general not possible. The standard approach in process control is rather to develop a linear model that approximates the dynamic behavior of a nonlinear system in a neighborhood of the specified operating point. *Linearization* is the procedure by which we approximate nonlinear systems with linear ones. This procedure will be described in detail in Lecture 6.

Linear State-Space Models

A linear time-invariant state-space model with n state variables, m inputs, and p outputs is given by

$$\begin{aligned}
\frac{dx_1}{dt} &= a_{11}x_1 + \ldots + a_{1n}x_n + b_{11}u_1 + \ldots + b_{1m}u_m \\
&\vdots \\
\frac{dx_n}{dt} &= a_{n1}x_1 + \ldots + a_{nn}x_n + b_{n1}u_1 + \ldots + b_{nm}u_m \\
y_1 &= c_{11}x_1 + \ldots + c_{1n}x_n + d_{11}u_1 + \ldots + d_{1m}u_m \\
&\vdots \\
y_p &= c_{p1}x_1 + \ldots + c_{pn}x_n + d_{p1}u_1 + \ldots + d_{pm}u_m
\end{aligned}$$

Note that only linear expressions of the state variables and of the inputs appear in the right-hand side of the equations. In a linear model, the state variables x_1, …, x_n usually describe deviations from the desired operating point.

Introducing vectors x, u and y as before and matrices A, B, C, and D of appropriate sizes, (3.2) can compactly be written as

$$\frac{dx}{dt} = Ax + Bu \tag{3.4}$$

$$y = Cx + Du \tag{3.5}$$

Note that $(x,u) = (0,0)$ is always a steady-state solution to (3.4).

Under enough simplifying assumptions, mathematical modeling can sometimes directly yield linear dynamical models. One such modeling example is given below.

Example 3.2—Compartment Model

A *compartment model* can be used to model the transport of substances between interconnected volumes, such as the flow of drugs and hormones in the human body. It is assumed that there is perfect mixing so that the substance concentration is constant in each compartment. The complex transport processes are approximated by assuming that the flow rates between the compartments are proportional to the concentration differences in the compartments. This results in simple, linear models, which are amenable to analysis.

Erik Widmark pioneered the study of ethanol pharmacokinetics in the 1920s and was the first to describe a one-compartment model. Torsten Teorell later introduced the two-compartment model. Compartment models are still an important analytical tool in the development of new drugs.

A two-compartment model is shown in Figure 3.2. We assume that the transport between the connected compartments is driven by concentration differences. We further

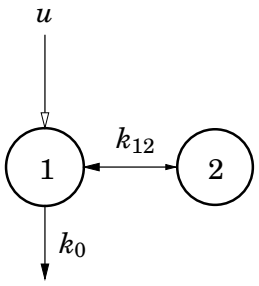


Figure 3.2 A two-compartment model

assume that a drug with concentration c_0 is injected in compartment 1 at a volume flow rate of u, and that the concentration in compartment 2 is the measured output. Let c_1 and c_2 be the concentrations of the drug in the compartments and let V_1 and V_2 be the volumes of the compartments. The dynamics of the system can be obtained by keeping track of the flow rates into and out of each compartment. The mass balances for the compartments are

$$\begin{aligned} V_1 \frac{dc_1}{dt} &= k_{12}(c_2 - c_1) - k_0 c_1 + c_0 u \\ V_2 \frac{dc_2}{dt} &= k_{12}(c_1 - c_2) \\ y &= c_2 \end{aligned}$$

Clearly, this model is not valid for all possible values of the variables. In particular, the concentrations can never be negative in a real system. □

3.3 Solution of the State Equation

For a linear state-space model, (3.4)–(3.5), it is possible to analytically compute how the system state will behave as a function of the initial state and of the control inputs. We will see that the eigenvalues of the system matrix A play a key role in the solution. This will allow us to reason about stability of a system in the following section. We will first look at the scalar case (i.e., one state variable) before treating the general case.

The Scalar Case

Consider a first-order linear system with one input:

$$\frac{dx}{dt} = ax + bu$$

The solution to this differential equation is given by

$$x(t) = e^{at}x(0) + \int_0^t e^{a(t-\tau)}bu(\tau)\,d\tau \tag{3.6}$$

The first term on the right hand side represents the influence of the initial value and the second term is the influence of the input signal during the time interval from 0 to t. If $a < 0$ then the influence of the initial value will vanish as t increases.

The General Case

The linear n:th order state equation (3.4) has the solution

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau \tag{3.7}$$

where e^{At} is the *matrix exponential*, defined as

$$e^{At} = I + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots$$

As in the scalar case, the solution (3.7) consists of two parts. The first part depends on the initial value of the state vector $x(0)$. This part is also called the solution to the *free system* or *homogeneous system*. The second part depends on the input signal u over the time interval from 0 to t.

The exponential matrix is an essential part of the solution. The characteristics of the solution is determined by the matrix A. The matrix e^{At} is called the *fundamental matrix* or the *state transition matrix* of the linear system (3.4). The influence of the B matrix is essentially a weighting of the input signals.

The Role of the Eigenvalues of A

We will now show that the eigenvalues of the matrix A play an important role for the solution of (3.4). The eigenvalues of the matrix A are given by the roots of the *characteristic equation*

$$\det(\lambda I - A) = 0 \tag{3.8}$$

The polynomial $\det(\lambda I - A) = P(\lambda)$ is known as the *characteristic polynomial* of the system. The number of roots of (3.8) is the same as the order of A.

Let us first assume that the A matrix in (3.4) is diagonal, i.e.

$$A = \begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_n \end{pmatrix}$$

For a diagonal matrix the diagonal elements λ_i are the eigenvalues of A. Note that we allow λ_i to be a complex number. The matrix exponential is then

$$e^{At} = \begin{pmatrix} e^{\lambda_1 t} & 0 & 0 & \dots & 0 \\ 0 & e^{\lambda_2 t} & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & e^{\lambda_n t} \end{pmatrix}$$

The eigenvalues of the A matrix will thus determine the time functions that build up the solution. Eigenvalues with positive real part give solutions that increase with time, while eigenvalues with negative real part give solutions that decay with time.

To interpret the solution (3.7) for a general matrix A we need to introduce the concept of multiplicity of an eigenvalue. The *multiplicity* of an eigenvalue is defined as the number of eigenvalues with the same value. If the matrix A has an eigenvalue λ_i with multiplicity m_i then it can be shown that the corresponding time function in e^{At} becomes

$$P_{m_i-1}(t)e^{\lambda_i t}$$

where $P_{m_i-1}(t)$ is a polynomial of t with a maximum degree of $m_i - 1$.

EXAMPLE 3.3—EIGENVALUE WITH MULTIPLICITY TWO
Assume that

$$A = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix}$$

This matrix has the eigenvalue -1 with multiplicity 2. The matrix exponential is

$$e^{At} = \begin{pmatrix} e^{-t} & te^{-t} \\ 0 & e^{-t} \end{pmatrix}$$

□

To summarize, the free system response is a sum of functions $P_{m_i-1}(t)e^{\lambda_i t}$, where λ_i are the eigenvalues of the matrix A. Real eigenvalues correspond to real exponential functions. The characteristic equation (3.8) can also have complex roots

$$\lambda = \sigma + i\omega$$

Such a root corresponds to the oscillatory solution

$$e^{\lambda t} = e^{\sigma t + i\omega t} = e^{\sigma t}(\cos \omega t + i \sin \omega t)$$

3.4 Stability

We saw in (3.7) that the solution to the linear state equation has two parts: one depending on the initial state and one depending the input signal. When discussing stability, we disregard the input signal and only look at the free system

$$\frac{dx}{dt} = Ax$$

which has the solution

$$x(t) = e^{At}x(0)$$

We give the following stability definitions:

DEFINITION 3.1—ASYMPTOTIC STABILITY
A linear system is *asymptotically stable* if $x(t) \to 0$ when $t \to \infty$ for all initial values $x(0)$ when $u(t) = 0$. □

DEFINITION 3.2—STABILITY
A linear system is *stable* if $x(t)$ is bounded for all initial values $x(0)$ when $u(t) = 0$.
□

DEFINITION 3.3—INSTABILITY
A linear system is *unstable* if there is any initial value $x(0)$ that gives an unbounded state $x(t)$ when $u(t) = 0$. □

In the previous section, we found that each eigenvalue λ_i gives rise to a solution term of the form $P_{m_i-1}(t)e^{\lambda_i t}$, where m_i is the multiplicity of the eigenvalue. From this, we can conclude that the system is asymptotically stable if the real part of all

eigenvalues are negative. If any eigenvalue has positive real part, then the system is unstable. Finally, if all eigenvalues have negative or zero real part then the system may be stable or unstable; if the eigenvalues with zero real part have multiplicity one, then the system is stable.

To summarize we have the following stability criteria for linear dynamical systems:

- A linear system is asymptotically stable if and only if all eigenvalues lie in the left half-plane, i.e., if and only if all roots λ_i of the characteristic equation satisfy $\mathrm{Re}\,\lambda_i < 0$.
- A linear system is unstable if any eigenvalue lies in the right half-plane, i.e., if any root λ_i of the characteristic equation satisfies $\mathrm{Re}\,\lambda_i > 0$.
- A linear system is stable if all eigenvalues lie in the left half-plane or on the imaginary axis and any eigenvalues on the imaginary axis are single, i.e., if all roots λ_i of the characteristic equation satisfy $\mathrm{Re}\,\lambda_i \leq 0$ and any roots satisfying $\mathrm{Re}\,\lambda_i = 0$ have multiplicity one.

Note that asymptotic stability implies stability, but the opposite is not true. A stable system that is not asymptotically stable is referred to as a *marginally stable* system.

Example 3.4—Stability of a First-Order System
Consider the scalar linear system

$$\frac{dx}{dt} = ax$$

The eigenvalue is a, and the free system response is $x(t) = e^{at}x(0)$. The relation between the eigenvalue location in the complex plane and the stability of the system is illustrated in Figure 3.3.

□

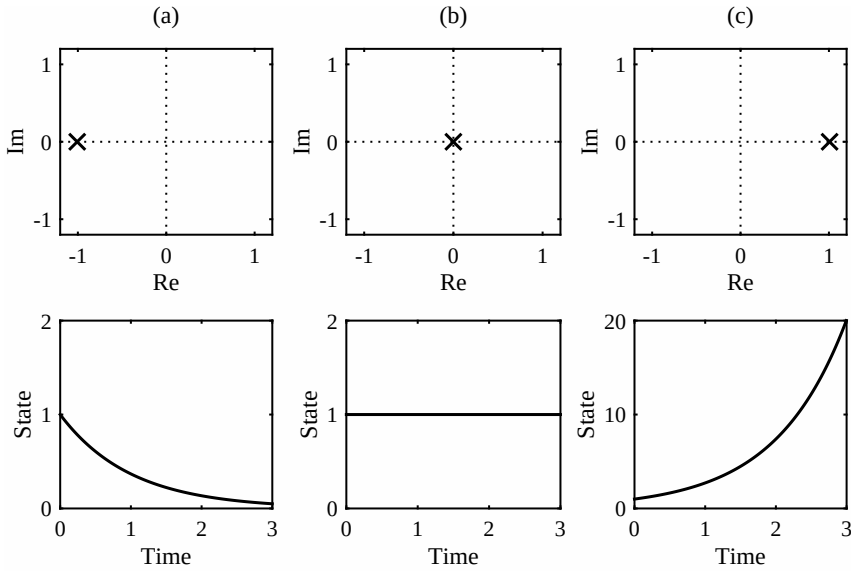


Figure 3.3 Relationship between eigenvalue location (top) and free system response (bottom) for a first-order system: (a) asymptotic stability, (b) marginal stability, (c) instability. The initial state is $x(0) = 1$.

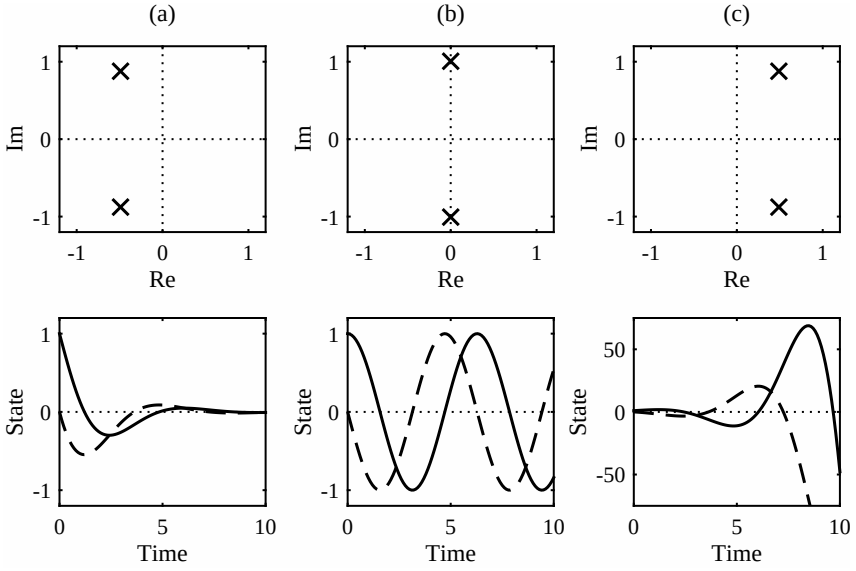


Figure 3.4 Relationship between eigenvalue locations (top) and free system response (bottom) for a second-order system: (a) asymptotic stability, (b) marginal stability, (c) instability. The initial state is $x(0) = [1\ 0]^T$.

EXAMPLE 3.5—STABILITY OF A SECOND-ORDER SYSTEM
Consider a second-order system with complex eigenvalues. The imaginary part of the eigenvalues causes the free response to oscillate, cf. (3.3). The relation between the eigenvalue locations in the complex plane and the stability of the system is illustrated in Figure 3.4.

□

Routh–Hurwitz Stability Criteria

It turns out that it is not necessary to compute the eigenvalues of A to check stability; it is sufficient to study the coefficients of the characteristic polynomial. Assume that the characteristic polynomial is given by a monic polynomial

$$P(\lambda) = \det(\lambda I - A) = \lambda^n + p_1\lambda^{n-1} + p_2\lambda^{n-2} + \ldots + p_n$$

A *necessary* condition for asymptotic stability is that all coefficients $p_1, \ldots, p_n$ are positive.

Exact conditions for asymptotic stability were derived by E. J. Routh and A. Hurwitz in the 19th century. The resulting stability tests are known as the Routh–Hurwitz stability criteria. It is useful to know the Routh–Hurwitz criteria for second and third order systems by heart. For a second-order characteristic polynomial,

$$P(\lambda) = \det(\lambda I - A) = \lambda^2 + p_1\lambda + p_2$$

the system is asymptotically stable if and only if $p_1 > 0$ and $p_2 > 0$. For a third-order characteristic polynomial,

$$P(\lambda) = \det(\lambda I - A) = \lambda^3 + p_1\lambda^2 + p_2\lambda + p_3$$

the system is asymptotically stable if and only if $p_1 > 0$, $p_2 > 0$, $p_3 > 0$, and $p_1p_2 > p_3$. Criteria for higher-order systems can be derived using the so called Routh–Hurwitz' algorithm.

4. Input-Output Models

In this lecture we will focus on linear dynamical processes and analyze the input-output behavior of such systems. We will disregard the internal states of the systems. The concept of a *transfer function* will allow us to treat linear dynamical systems as abstract mathematical objects. These objects can be connected in series, parallel or feedback fashion by means of simple algebraic manipulations. The use of transfer functions and the *Laplace transform* makes it straightforward to calculate the output response to arbitrary input signals.

4.1 The Laplace Transform

Solving the state equation (3.4) for a particular input signal can be very cumbersome and usually gives little insight into the general properties of the system. To simplify the study of linear differential equations we therefore introduce a new tool: the *Laplace transform*. This is a method that is often used in applied mathematics. The Laplace transform gives a way to get a feel for the behavior of the system without solving the differential equations in detail. The Laplace transform will thus make it possible to qualitatively determine how the processes will react to various types of input signals and disturbances.

Definition

The Laplace transform is a transformation from a real-valued function of a scalar variable t to a complex-valued function of a complex variable s. In process control t is the time and s can be interpreted as a (complex) frequency. The transformation implies that both time functions and differential equations are transformed into functions of a complex variable. The analysis of the systems can then be done by investigating the transformed variables. The solution of differential equations is reduced to algebraic manipulations of the transformed system and the transformed time functions.

Definition 4.1—Laplace transform

The Laplace transform of the function $f(t)$ is denoted $F(s)$ and is obtained through

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t)\, dt$$

□

Direct application of the definition can be used to derive the Laplace transform of some common time functions:

Example 4.1—Laplace transform of a step function

Let $f(t)$ be a unit step function

$$f(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

The Laplace transform is

$$F(s) = \int_0^\infty e^{-st}\,dt = \left[-\frac{e^{-st}}{s}\right]_0^\infty = \frac{1}{s}$$

□

EXAMPLE 4.2—LAPLACE TRANSFORM OF A RAMP FUNCTION
Assume that $f(t)$ is a ramp

$$f(t) = \begin{cases} 0 & t < 0 \\ at & t \geq 0 \end{cases}$$

The Laplace transform is

$$F(s) = \int_0^\infty ate^{-st}\,dt = \frac{a}{s^2}$$

□

Often the definition is not used directly. A table of Laplace transforms of common time functions is found in the Collection of Formulae.

Properties of the Laplace Transform

From the definition of the Laplace transform it follows that it is a linear operator, i.e.

$$\mathcal{L}\{a_1 f_1(t) + a_2 f_2(t)\} = a_1\mathcal{L}\{f_1(t)\} + a_2\mathcal{L}\{f_2(t)\} = a_1F_1(s) + a_2F_2(s)$$

Since the single-sided Laplace transform is used, $F(s)$ does not contain any information about $f(t)$ for $t < 0$. This is usually not a drawback in process control since we can define that the systems are in steady state for $t < 0$ and let the inputs start to influence the systems at $t = 0$.

To obtain the Laplace transform of a differential equation it is necessary to derive the transform for a time derivative of a signal.

THEOREM 4.1—LAPLACE TRANSFORM OF TIME DERIVATIVE
Let $F(s)$ be the Laplace transform of $f(t)$. Then

$$\mathcal{L}\left\{\frac{df(t)}{dt}\right\} = sF(s) - f(0)$$

where $f(0)$ is the initial value of the function $f(t)$. □

If the initial values are zero then taking the time derivative corresponds to multiplication by s. If the differential equation has initial values the expressions become more complex.

The following theorem is useful for analysing dead time processes.

THEOREM 4.2—LAPLACE TRANSFORM OF TIME DELAY
Let $F(s)$ be the Laplace transform of $f(t)$, and let $L > 0$ be a time delay. Then

$$\mathcal{L}\{f(t-L)\} = F(s)e^{-sL}$$

□

The following two theorems can be used to obtain steady state values and starting of solutions to differential equations. They will prove useful when analyzing the step responses of dynamical systems.

THEOREM 4.3—FINAL VALUE THEOREM
Let $F(s)$ be the Laplace transform of $f(t)$. Then

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$$

if the limit on the left hand side exists. □

THEOREM 4.4—INITIAL VALUE THEOREM
Let $F(s)$ be the Laplace transform of $f(t)$. Then

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s)$$

if the limit on the left hand side exists. □

Further properties of the Laplace transform can be found in the Collection of Formulae.

Solution of Linear Differential Equations Using the Laplace Transform

Using Theorem 4.1 together with a table of Laplace transforms makes it very easy to solve ordinary differential equations. Sometimes, it is necessary to use partial fraction decomposition before applying the inverse transform. Two examples are given below.

EXAMPLE 4.3—FIRST-ORDER INHOMOGENEOUS ODE WITH INITIAL VALUE
Determine the solution to the first-order linear differential equation

$$\frac{dy}{dt} + ay = t$$

with the initial value $y(0) = y_0$.

Taking the Laplace transform of each term in the equation gives

$$\begin{aligned} sY(s) - y_0 + aY(s) &= \frac{1}{s^2} \\ Y(s) &= \frac{y_0}{s+a} + \frac{1}{s^2(s+a)} \end{aligned}$$

Applying the inverse Laplace transform (using the table of transforms) yields

$$y(t) = e^{-at}y_0 + \frac{t}{a} - \frac{1}{a^2}\left(1 - e^{-at}\right)$$

The influence of the initial value will vanish as $t \to \infty$ if $a > 0$. □

EXAMPLE 4.4—SECOND-ORDER INHOMOGENEOUS ODE
Determine the solution to the second-order linear differential equation

$$\frac{d^2y}{dt^2} + 4\frac{dy}{dt} + 3y = 5\sin 2t$$

with the initial conditions

$$y(0) = 0 \qquad \frac{dy(0)}{dt} = 0$$

Taking the Laplace transform of the equation gives

$$(s^2 + 4s + 3)Y(s) = \frac{10}{s^2 + 4}$$

The transform of the output is

$$\begin{aligned} Y(s) &= \frac{10}{(s^2 + 4s + 3)(s^2 + 4)} = \frac{10}{(s+1)(s+3)(s^2+4)} \\ &= \frac{1}{s+1} - \frac{5}{13}\frac{1}{s+3} - \frac{8}{13}\frac{s}{s^2+4} - \frac{1}{13}\frac{2}{s^2+4} \end{aligned}$$

The table of transforms now gives

$$y(t) = e^{-t} - \frac{5}{13}e^{-3t} - \frac{8}{13}\cos 2t - \frac{1}{13}\sin 2t$$

□

4.2 The Transfer Function

Let the input–output relationship of a linear system be described by an n:th order differential equation

$$\frac{d^n y}{dt^n} + p_1\frac{d^{n-1}y}{dt^{n-1}} + \cdots + p_n y = q_0\frac{d^m u}{dt^m} + q_1\frac{d^{m-1}u}{dt^{m-1}} + \cdots + q_m u \tag{4.1}$$

Assuming that all initial values are zero and taking the Laplace transform gives

$$(s^n + p_1 s^{n-1} + \cdots + p_n)Y(s) = (q_0 s^m + q_1 s^{m-1} + \cdots + q_m)U(s)$$

Introducing the polynomials

$$\begin{aligned} P(s) &= s^n + p_1 s^{n-1} + \cdots + p_n \\ Q(s) &= q_0 s^m + q_1 s^{m-1} + \cdots + q_m \end{aligned}$$

this can be compactly written as

$$P(s)Y(s) = Q(s)U(s)$$

Solving for $Y(s)$ gives

$$Y(s) = \frac{Q(s)}{P(s)}U(s) = G(s)U(s)$$

$G(s)$ is called the *transfer function* of the system. It is a compact representation of the input-output relationship (4.1). Using the transfer function, the output response $Y(s)$ to any input signal $U(s)$ can be found by means of the simple algebraic relationship

$$Y(s) = G(s)U(s) \tag{4.2}$$

In general, this procedure is much simpler than directly solving the differential equation (4.1) in the time domain.

Solution of the State Equation Using the Laplace Transform

We will now calculate the response of a state-space system by using the Laplace transform. Starting from a linear state-space model

$$\begin{aligned}\frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

and taking the Laplace transform of each term yields

$$\begin{aligned}sX(s) - x(0) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s)\end{aligned}$$

Note that we include the initial value of x. Solving for $X(s)$ gives

$$\begin{aligned}(sI - A)X(s) &= x(0) + BU(s) \\ X(s) &= (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s)\end{aligned}$$

The output signal is given by

$$Y(s) = C(sI - A)^{-1}x(0) + \Big(C(sI - A)^{-1}B + D\Big)\,U(s) \tag{4.3}$$

Just as we saw in Equation (3.7) in the previous lecture, we have one part that depends on the initial state $x(0)$ and one part that depends on the input signal $u(t)$. Since the system is linear, we can study the response to the initial state and the input separately. Assuming $x(0) = 0$, we obtain the input-output relationship

$$Y(s) = \Big(C(sI - A)^{-1}B + D\Big)\,U(s) = G(s)U(s)$$

Again, $G(s)$ is the called *transfer function* of the system.

Poles and Zeros

The transfer function $G(s)$ gives the properties of the system and can be used to determine its dynamical behavior without solving the differential equations. The transfer function can often be written as a rational function

$$G(s) = \frac{Q(s)}{P(s)}$$

where $Q(s)$ and $P(s)$ are polynomials in the Laplace variable s. $P(s)$ is called the *characteristic polynomial* of the system. The degree of $P(s)$ is n, where n is the order of the system. The degree of $Q(s)$ is less than or equal to n.

One important exception to the above is when the system contains a time delay. A process dead time of L seconds corresponds to the inclusion of a factor e^{-sL} in the transfer function. This factor cannot be written as a rational function, and hence cannot be described in terms of a finite number of poles and zeros.

Definition 4.2—Poles

The *poles* of the system are given by the roots of the characteristic polynomial, i.e., by the n solutions to

$$P(s) = 0$$

This is known as the *characteristic equation*. □

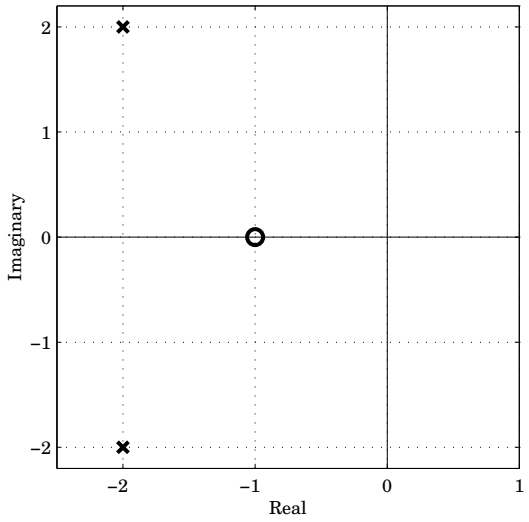


Figure 4.1 Singularity diagram for the system $G(s) = k(s+1)/(s^2+4s+8)$.

DEFINITION 4.3—ZEROS
The *zeros* of the system are given by the roots of $Q(s)$, i.e., by the solutions to

$$Q(s) = 0$$

□

The poles and zeros for a given transfer function $G(s)$ can be illustrated in a *singularity diagram* (also known as a *pole/zero map*). The poles are represented by crosses and the zeros by circles in the complex s-plane, see the example in Figure 4.1. Note that to fully specify the transfer function from the singularity diagram it is necessary to know the gain parameter k in the transfer function.

Relationship Between Eigenvalues and Poles

For a linear state-space system

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

we have shown above that the transfer function is given by

$$G(s) = C(sI - A)^{-1}B + D$$

By noting that

$$(sI - A)^{-1} = \frac{\text{adj}(sI - A)}{\det(sI - A)}$$

we see that the poles of $G(s)$ are given by the roots of the characteristic equation

$$\det(sI - A) = 0$$

This is the same equation as (3.8) (with s as the independent variable instead of λ). We conclude that the poles of $G(s)$ are identical to the eigenvalues of A. The stability criteria in Section 3.4 can hence be used also to check the stability of systems written in transfer function form. For instance, for a system to be asymptotically stable, all poles should lie in the left half plane.

Translation Between Different System Representations

So far, we have encountered three different ways to represent a linear dynamical system:

- Linear state-space model
- Higher-order linear differential equation
- Transfer function

The three representations are equivalent in the sense that they can all capture the input–output relationship of a linear system. The state-space model is however more expressive since it also keeps track of the internal state variables of the system.

It is straightforward to translate between the last two representations using Theorem 4.1. The translation from state-space model to transfer function is given by (4.2). The opposite translation is non-unique, since the state variables can be chosen in infinitely many ways. Translation charts for various canonical state-space forms are given in the Collection of Formulae.

4.3 Block Diagram Algebra

A control system typically consists of several subsystems. These may be different parts of the process, sensors, controllers, and actuators. It important to be able to describe the dynamics of the full system. Block diagrams and transfer functions are good tools in this respect. For linear systems each subsystem (block) can be described by a transfer function. The subsystems can be connected in different ways. There are three basic couplings, see Figure 4.2. Let each the subsystem have the transfer functions G_1 and G_2 and let the total transfer function from the input to the output be G. We have the following relationships:

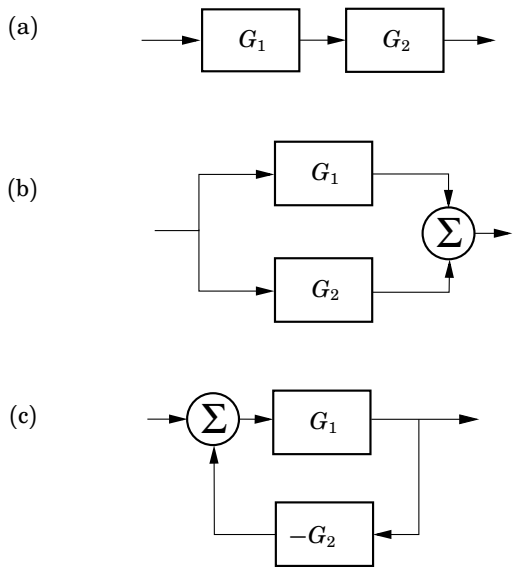


Figure 4.2 Three basic connections of subsystems: (a) series connection, (b) parallel connection, (c) feedback connection.

Series connection:

$$G(s) = G_1(s)G_2(s)$$

Parallel connection:

$$G(s) = G_1(s) + G_2(s)$$

Feedback connection:

$$G(s) = \frac{G_1(s)}{1 + G_1(s)G_2(s)}$$

The three basic couplings can be used to simplify a complex system to derive the total transfer function from the input to the output. A straightforward way to derive the total transfer function from inputs to the output is to introduce notations for the internal signal and write down the relations between the signals. The extra signals are then eliminated one by one until the final expression is obtained. Another way to simplify the derivation is the so called *backward method*. The method can be used on systems without inner feedback loops and is illustrated in the following example.

EXAMPLE 4.5
We consider the system in Figure 4.3 and derive the transfer functions from $U(s)$ and $V(s)$ to $Y(s)$. Start from the output end of the block diagram and write down the expression for the Laplace transform of the output. This is expressed in signals coming from the left. These signals are then expressed in terms of other signals, and so on. For the example we have

$$\begin{aligned} Y &= G_2(V + Y_1) \\ Y_1 &= G_1 E \\ E &= H_1 U - H_2 Y \end{aligned}$$

Combining these, we have

$$Y = G_2(V + G_1(H_1 U - H_2 Y))$$

After some training it is possible to write down the last expression directly, without introducing the internal variables Y_1 and E. Solving for Y, The expression for the Laplace transform of the output is given as

$$Y = \frac{G_1 G_2 H_1}{1 + G_1 G_2 H_2} U + \frac{G_2}{1 + G_1 G_2 H_2} V$$

□

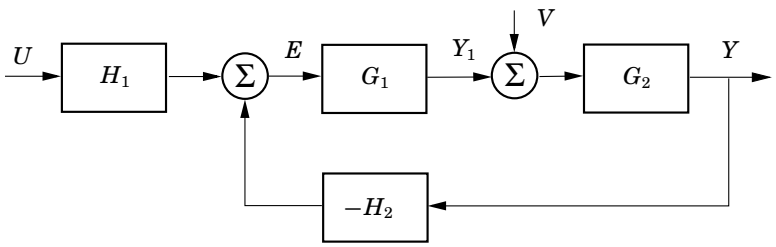


Figure 4.3 The system in Example 4.5

.

5. Transient Analysis. Nonlinear Systems

In the first part of this lecture, we will finish the analysis of linear dynamical system by studying their transient response—their impulse response and their step response. In the second part, we will have a look at typical nonlinearities in process control.

5.1 Impulse and Step Response

In the previous lecture, we learned how to calculate the output response of a linear system to an arbitrary input signal using the Laplace transform. Given the systems's transfer function $G(s)$, the input-output relationship in the Laplace domain is given by

$$Y(s) = G(s)U(s)$$

Below we study two important special cases: the response to an impulse function and the response to a step function.

The Impulse Response

The impulse response describes how the output signal responds when the input signal is an impulse function, i.e.,

$$u(t) = \delta(t)$$

Impulse response analysis is common in medicine and biology. One can for instance inject a substance into the blood circulation and study the resulting uptake and secretion. An example of an impulse response experiment is shown Figure 5.1.

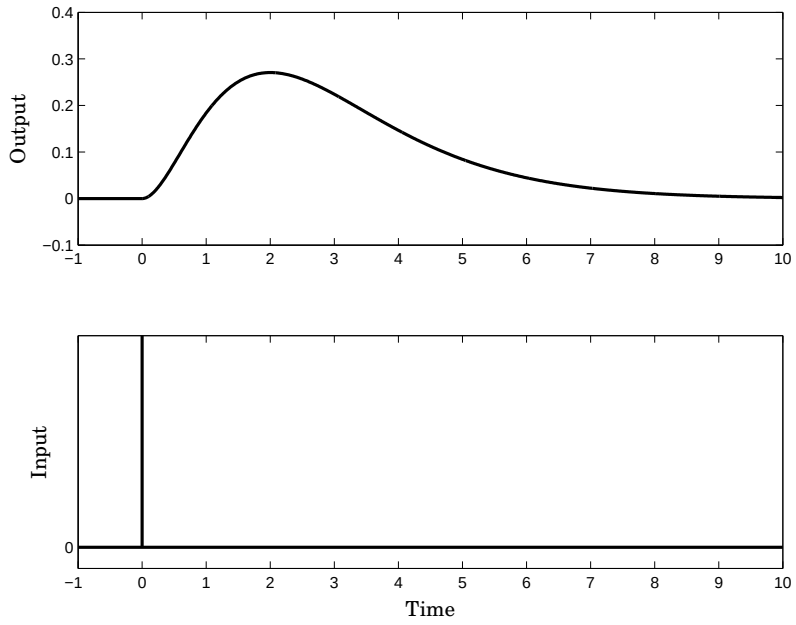


Figure 5.1 Impulse response experiment.

For a linear system, the impulse response, denoted $h(t)$, is easily computed from a state-space model or a transfer function model. With $x(0) = 0$ and $u(t) = \delta(t)$, the solution (3.7) to the state-space equation simplifies to

$$x(t) = e^{At}B$$

and the output is given by

$$y(t) = Ce^{At}B + D\delta(t) \equiv h(t)$$

Once again it is seen that the matrix exponential e^{At} plays an important role. The impulse response will approach zero when $t \to \infty$ if the system is asymptotically stable, and it will be bounded (except, possibly, at $t = 0$) if the system is stable.

For a system given in transfer function form, the output is computed as

$$Y(s) = G(s)U(s) \tag{5.1}$$

The Laplace transform of the input $u(t) = \delta(t)$ is $U(s) = 1$, so we have

$$Y(s) = G(s)$$

The transfer function is hence equal to the Laplace transform of the impulse response. The impulse response can be obtained through inverse transformation:

$$h(t) = \mathcal{L}^{-1}\Big\{G(s)\Big\}$$

$h(t)$ is also known as the *weighting function* of the system. The reason is the following. The response to a general input $u(t)$ can be computed by (5.1). Applying the inverse transform to we obtain

$$y(t) = \int_0^t h(t-\tau)u(\tau)\,d\tau \tag{5.2}$$

The output at time t is hence a weighted sum of old inputs up to time t, where $h(t)$ determines how old inputs are weighted.

The Step Response

The step response describes how the output signal responds when the input signal is a unit step function, i.e.,

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

Step response analysis is very common in chemical process industry, where it can be used to identify process dynamics. An example of a step response experiment is shown in Figure 5.2

For a linear system in state-space form, assuming $\det A \neq 0$, the step response can be calculated as

$$y(t) = CA^{-1}(e^{At} - I)B + D \qquad t \geq 0$$

If the transfer function is given, the system output can be calculated using

$$Y(s) = G(s)U(s)$$

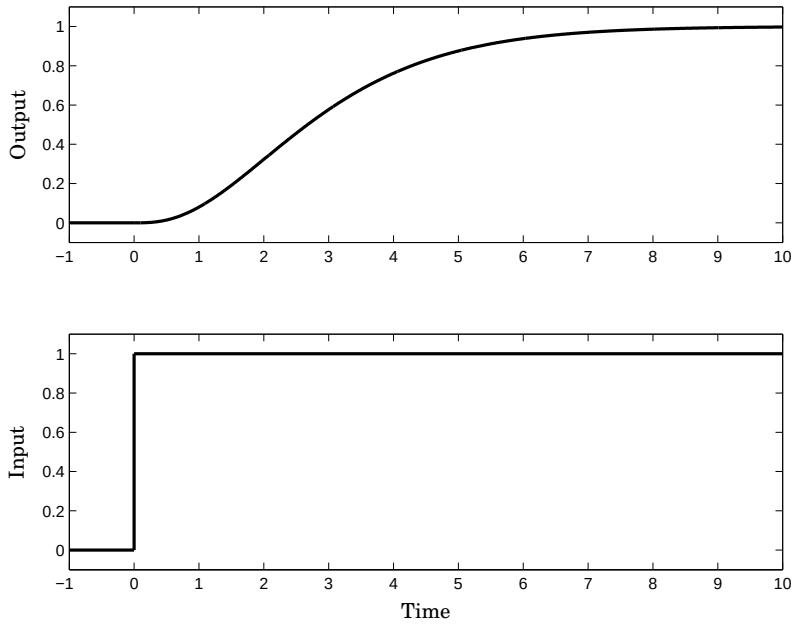


Figure 5.2 Step response experiment.

where the Laplace transform of the step input is given by $U(s) = 1/s$. We have

$$Y(s) = G(s)\frac{1}{s}$$

Applying the inverse Laplace transformation yields

$$y(t) = \mathcal{L}^{-1}\left\{\frac{1}{s}G(s)\right\} = \int_0^t h(\tau)\,d\tau$$

The step response is thus given by the integral of the impulse response.

Static Gain

In Section 2.2 is was shown how the static gain of a process can be determined from a step-response experiment as $K_p = \Delta y/\Delta u$. If the process is linear and its transfer function $G(s)$ known, the static gain can be directly obtained as $K_p = G(0)$.

To prove this fact, let the input signal be a step of size Δu: $U(s) = \Delta u/s$. The output signal then becomes

$$Y(s) = G(s)\frac{\Delta u}{s}$$

Applying the final value theorem yields

$$\Delta y = \lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} sG(s)\frac{\Delta u}{s} = G(0)\Delta u$$

and we have the static gain

$$K_p = \frac{\Delta y}{\Delta u} = G(0)$$

Note that the final value—and hence the static gain—exists if and only if $G(s)$ is asymptotically stable.

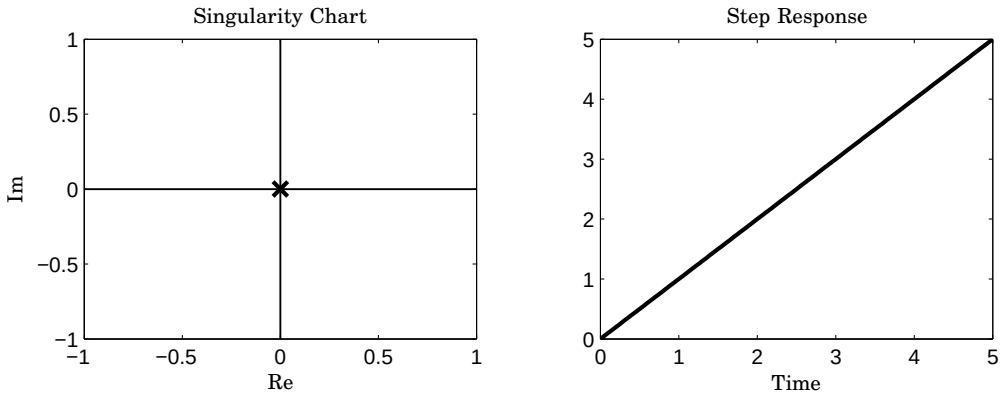


Figure 5.3 Singularity chart and step response for the process $G(s) = 1/s$.

5.2 Relationship Between Transfer Function and Step Response

In many situations it is useful to be able to see the characteristics of the step response directly from the transfer function, without the need of calculations. In this section we will see how the poles of the system affect the properties of the step response for some simple processes. The influence of a process zero and a dead time in the transfer function will also be examined.

Integrating System

Assume that the process is an integrator, described by the transfer function

$$G(s) = \frac{K}{s}$$

This process has a pole in $s = 0$. If the input of the process is a unit step, the Laplace transform of the process output becomes

$$Y(s) = G(s)\frac{1}{s} = \frac{K}{s^2}$$

The output of the process is obtained by inverse transformation:

$$y(t) = Kt$$

It is seen that the output is unbounded and that the steady-state value does not exist. The process can be said to have infinite static gain. Figure 5.3 shows the step response when $K = 1$.

First-Order System

Assume that the transfer function of the process is given by

$$G(s) = \frac{K}{1 + sT}$$

This process has a pole in $s = -1/T$. If the input of the process is a unit step, the Laplace transform of the process output becomes

$$Y(s) = G(s)\frac{1}{s} = \frac{K}{s(1 + sT)}$$

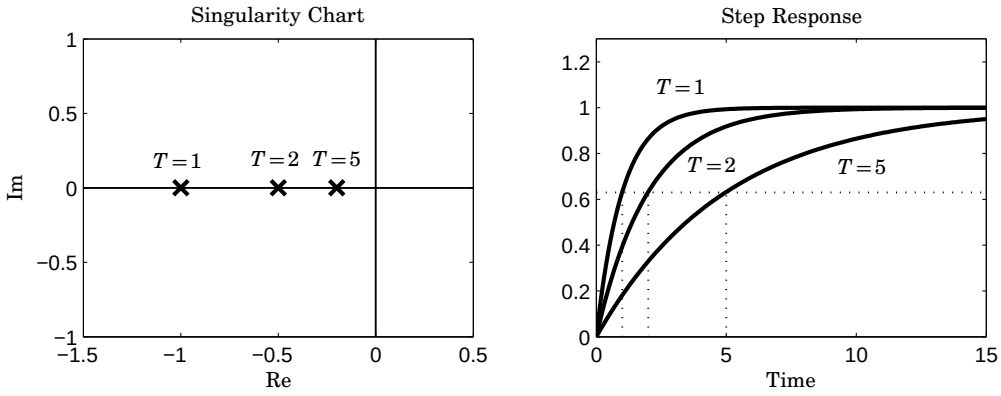


Figure 5.4 Singularity chart and step response for the process $G(s) = 1/(1 + sT)$ when $T = 1, 2, 5$.

The output of the process is obtained by inverse transformation

$$y(t) = K\left(1 - e^{-t/T}\right)$$

If $T < 0$, i.e. if the pole lies in the right half-plane, the process is unstable and $y(t)$ is unbounded. From now on we assume that $T > 0$, i.e. that the pole lies in the left half-plane.

Figure 5.4 shows the step responses corresponding to three different values of T. We recognize this as the single-capacitive process type from Section 2.3. The figure shows that a smaller T yields a faster step response. It also shows the pole of the process. The further into the left half-plane the pole lies, the faster the step response becomes.

The static gain of the system is

$$G(0) = K$$

This tell us that the process output approaches $y = K$ as $t \to \infty$.

The parameter T is called the *time constant* of the process. At time $t = T$ the process output is given by

$$y(T) = K\left(1 - e^{-T/T}\right) = K\left(1 - e^{-1}\right) \approx 0.63K$$

Time constant T is thus the time it takes for the step response to reach 63% of its final value. This is also seen in Figure 5.4.

Applying the initial value theorem on $y(t)$ tells us that $y(0) = 0$. It is more interesting to study at which rate the process output changes in the initial phase, i.e. to study $\dot{y}(0)$:

$$\lim_{t\to 0} \dot{y}(t) = \lim_{s\to\infty} s \cdot sY(s) = \lim_{s\to\infty} \frac{s^2 K}{s(1+sT)} = \frac{K}{T}$$

The shorter the time constant, i.e. the further into the left half-plane the pole lies, the faster the initial change of the process output becomes. This is also visible in Figure 5.4.

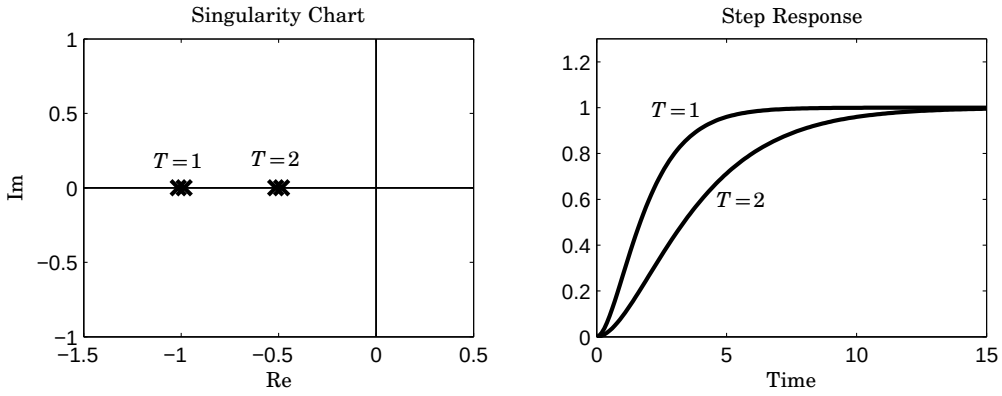


Figure 5.5 Singularity chart and step response of the process $G(s) = 1/(1+sT)^2$ when $T = 1, 2$.

Second-Order System with Real Poles

Assume that the transfer function of the process is given by

$$G(s) = \frac{K}{(1+sT_1)(1+sT_2)}$$

This process has two real poles, in $s = -1/T_1$ and $s = -1/T_2$, respectively. T_1 and T_2 are the time constants of the process. Compared to a first-order system, the second-order system has an *equivalent time constant* $T_{eq} = T_1 + T_2$.

If the input to the process is a unit step, the Laplace transform of the process output is given by

$$Y(s) = G(s)\frac{1}{s} = \frac{K}{s(1+sT_1)(1+sT_2)}$$

By inverse transformation of this expression, we obtain the output of the process

$$y(t) = \begin{cases} K\left(1 - \dfrac{T_1 e^{-t/T_1} - T_2 e^{-t/T_2}}{T_1 - T_2}\right) & T_1 \neq T_2 \\ K\left(1 - e^{-t/T} - \dfrac{t}{T}e^{-t/T}\right) & T_1 = T_2 = T \end{cases}$$

From the first expression we see that, when one of the time constants is significantly smaller than the other, the step response will approach that of a first-order process, as in Example 5.2. If any pole lies in the right half-plane, the process is unstable and $y(t)$ grows out of bounds. We therefore assume that both poles lie in the left half-plane, i.e. that $T_1 > 0$ and $T_2 > 0$.

Figure 5.5 shows the step response of the process as $T_1 = T_2 = T$. We recognize this as a multi-capacitive process (see Section 2.3). Just as in the previous example, we observe that the step response becomes faster as the poles move further into the left half-plane. The static gain is

$$G(0) = K$$

It tells us that the output approaches $y = K$ as $t \to \infty$.

The initial value theorem applied to $\dot{y}(t)$ yields

$$\lim_{t\to 0} \dot{y}(t) = \lim_{s\to\infty} s \cdot sY(s) = \lim_{s\to\infty} \frac{s^2 K}{s(1+sT_1)(1+sT_2)} = 0$$

Hence the time derivative is zero, which is also evident from Figure 5.5.

It is easy to verify that the initial time derivative is zero for all systems where the number of poles minus the number of zeros is greater than one.

Second-Order System with Complex Poles

For a second order system with complex poles it is often convenient to write the transfer function in the form

$$G(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \qquad 0 < \zeta < 1$$

Interpretations of the parameters ω_0 and ζ are given in Figure 5.6. The parameter ω_0 is called the *undamped frequency* and corresponds to the distance between the poles and the origin. The parameter ζ is called the *relative damping* and is related to the angle φ as in Figure 5.6 through

$$\zeta = \cos\varphi$$

The relative damping ζ gives the relation between the real and imaginary parts of the poles.

If the process input is a step, the Laplace transform of the process output becomes

$$Y(s) = G(s)\frac{1}{s} = \frac{K\omega_0^2}{s(s^2 + 2\zeta\omega_0 s + \omega_0^2)}$$

By inverse transformation of this expression the following process output is obtained.

$$y(t) = K\left(1 - \frac{1}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_0 t}\sin\left(\omega_0\sqrt{1-\zeta^2}t + \arccos\zeta\right)\right) \qquad 0 < \zeta < 1$$

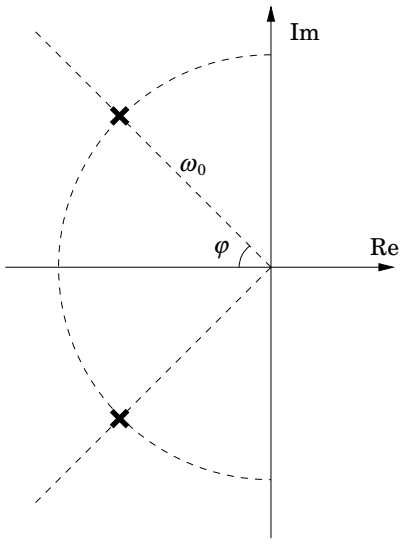


Figure 5.6 Interpretation of the parameters in the characteristic polynomial $s^2 + 2\zeta\omega_0 s + \omega_0^2$. The distance between the poles and the origin is the undamped frequency ω_0, whereas $\zeta = \cos\varphi$ is the relative damping of the system.

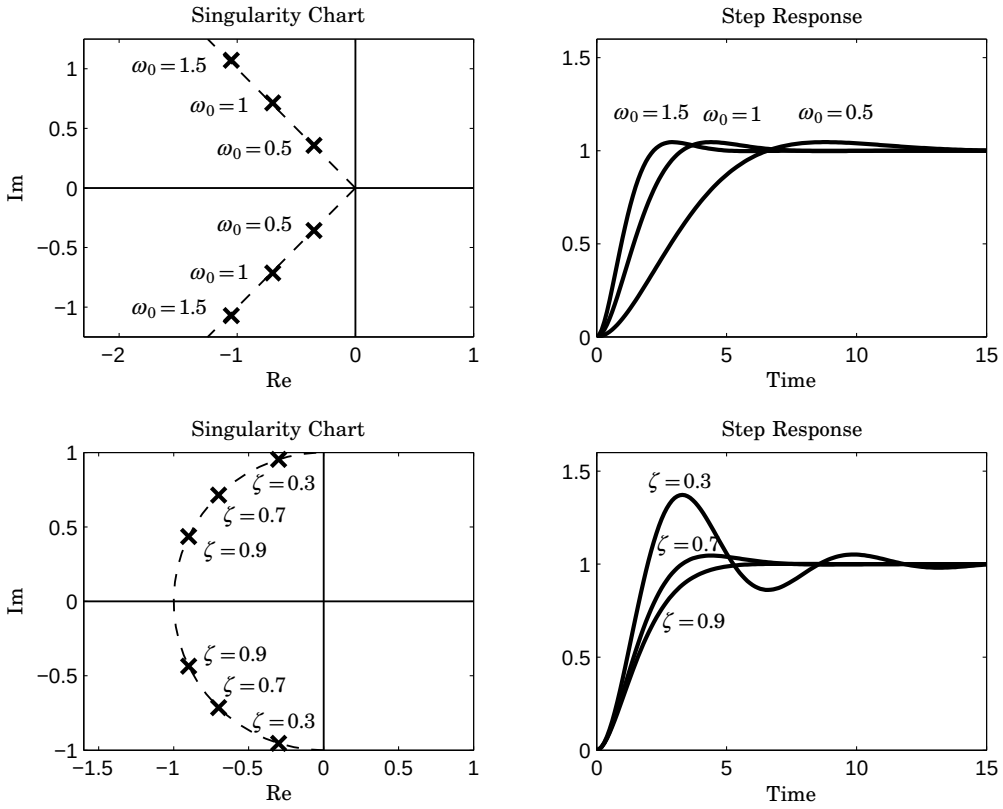


Figure 5.7 Singularity chart and step response of the process $G(s) = 1/(s^2 + 2\zeta\omega_0 s + \omega_0^2)$. The two upper plots show the cases $\zeta = 0.7$ and $\omega_0 = 0.5, 1, 1.5$. The two lower plots show the cases $\omega_0 = 1$ and $\zeta = 0.3, 0.7, 0.9$.

The expression contains a term consisting of a sinusoid with decaying amplitude.

Figure 5.7 shows the step response of the process for some different values of ζ and ω_0. The parameter ω_0 determines the distance between the poles and the origin. Just as in the previous examples, the step response becomes faster as the poles are moved further into the left half-plane. We also see that the shape of the step response does not change as long as ζ is held constant.

The parameter ζ determines the ratio between the real and imaginary part of the poles. Figure 5.7 shows that the smaller ζ is, the less damped the step response becomes. We also see that the initial part of the step response is fairly consistent as long as ω_0 is held constant.

Second-Order System with a Zero

We will now examine the effect of a zero in the process transfer function. Assume that the transfer function is given by

$$G(s) = (1 + sT_z)G_0(s)$$

where $G_0(s)$ is a stable transfer function without zeros. The influence of T_z will now be determined. The step response is given by

$$y(t) = \mathcal{L}^{-1}\left\{G_0(s)\frac{1}{s}\right\} + T_z\mathcal{L}^{-1}\{G_0(s)\}$$

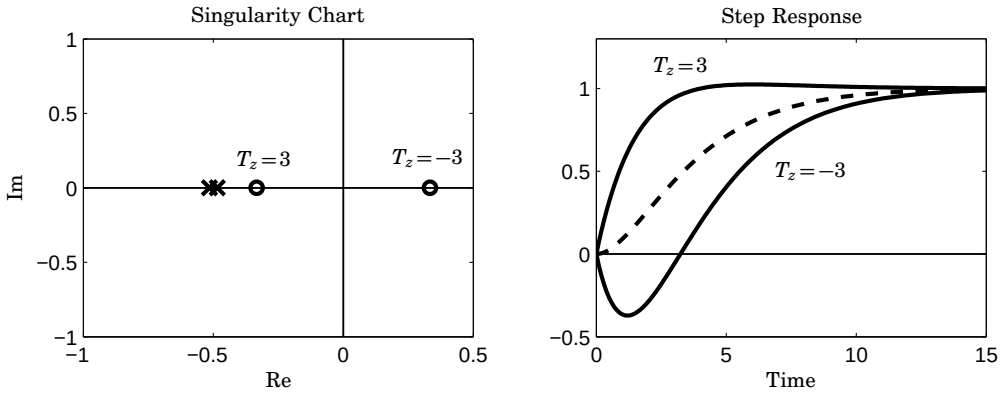


Figure 5.8 Singularity chart and step response of the process $G(s) = (1 + sT_z)/(1 + sT)^2$ when $T_z = 3, -3$. The dashed step response corresponds to $G_0(s) = 1/(1+sT)^2$, i.e., the process without a zero.

The response is a weighted sum of the step and impulse responses of the system $G_0(s)$. If T_z is small, the step response is dominated by the step response of $G_0(s)$. If T_z is large, the shape of the step at the beginning will be dominated by the impulse response of $G_0(s)$.

Figure 5.8 shows the step response of the process

$$G(s) = \frac{1 + sT_z}{(1 + sT)^2} \qquad T = 2$$

for different values of T_z. When T_z is negative, the step response starts in the "wrong" direction. We recognize this as the inverse response process type from Section 2.3.

In the process industry, a well known example of a process with inverse response arises when controlling the dome level in steam boilers. If one, for instance, would like to increase the dome level by increasing the flow of water into the dome, the first reaction will be that the water in the dome is cooled. This results in less steam bubbles in the water and consequently the dome level decreases. Only after a while, when the water has been heated anew, will the level increase.

Dead-Time System

In the last example, we study the effect of a dead time (time delay) in the transfer function. Assume that the transfer function is given by

$$G(s) = G_0(s)e^{-sL}, \quad L > 0$$

where $G_0(s)$ is the delay-free part of the process dynamics and L is the dead time. Let $y_0(t)$ be the delay-free step response, i.e.,

$$y_0(t) = \mathcal{L}^{-1}\left\{G_0(s)\frac{1}{s}\right\}$$

Then the step response for the dead time process will simply be delayed by L time units:

$$y(t) = y_0(t - L)$$

The dead time can be combined with any of the transfer functions examples studied above.

5.3 Nonlinear Systems

As a simplifying assumption, we often say that the process we wish to control is linear. By this we mean that the reaction of the process to changes in the control signal is the same regardless of which part of the working range of the process we are lying in. As long as the process is linear we can use one and the same controller setting over the whole of the working range.

In practice, however, we always have nonlinearities in our control loops. In many cases these may be disregarded, but sometimes they cause problems that we have to remedy. This is particularly true if there are high demands on performance. It is important to remember that we use the term *process* to mean everything which lies outside the controller; this therefore includes transducers, control devices, etc.

Nonlinear Actuators—Valves

Valves are probably the most common cause of problems with nonlinearities in process control. This is partly because they are nonlinear in their design, and partly because they often cause problems because they are not in good condition.

A valve often has a different gain in different working areas. This can be determined by drawing out the characteristics of the valve, i.e. the relationship between the signal input to the valve and the flow through it. Figure 5.9 shows some common types of valve characteristic. The linear characteristic is naturally the one the controller would most like to see. If we have linear characteristics we can use the same controller parameters over the whole of the working range. The quick opening characteristic quickly gives a fast flow when the valve is opened. The equal percentage valve gives a constant relative accuracy. The throttle valve is by nature a quick-opening valve. However, using actuators, cams etc., the characteristic can be changed to linear or equal percentage.

The above reasoning will hold as long as we do not have any large pressure-reducing element in our line apart from the valve. If we have any other constrictions which cause a pressure reduction in the line, the valve characteristics may be completely different. In order to understand this, let us look at the two cases illustrated in Figure 5.10. The flow through the valve is proportional to the square root of the pressure difference, i.e. the difference in pressure before and after the valve. In the first case, valve *A* is the only pressure-reducing element, and the characteristic will

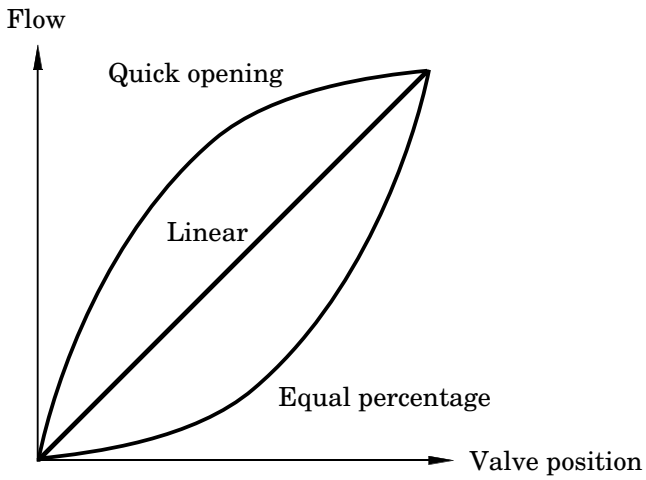


Figure 5.9 Some different types of valve characteristics

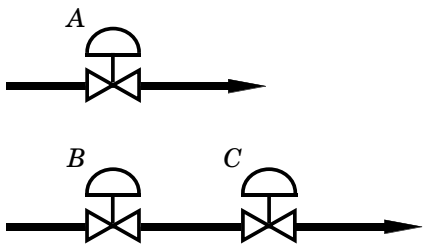


Figure 5.10 Two cases which will result in different valve characteristics

be that which was expected. In the second case, apart from valve B we have valve C which reduces the pressure in the line. An increase in the valve position of valve B, therefore, will not give as large an increase in flow as a corresponding opening of valve A would give.

It is not difficult to understand the technical problems of control when the valve is nonlinear. Assume for a moment that we have a quick-opening valve. At a small valve position the relative change in flow is large; in other words the valve has a high gain at small valve openings. On the other hand, the quick-opening valve has a low gain at large valve openings. If we are to tune a controller which is to span the whole control range of the valve, we would have to give it a low gain so that we don't get stability problems at small valve openings. This, however, will result in slow control at large valve apertures as we then have low gain in both the process and the controller.

One solution to the problem is to tell the controller what the valve characteristics look like, so that it can vary its gain at different working positions. This is called *gain scheduling* and will be discussed at the end of the lecture.

Nonlinear Sensors

Transducers (sensors) are often nonlinear. The reason for this is that the transducers are often not measuring directly the quantity we are actually interested in, but another quantity which is related to it. Temperature transducers made of thermocouples are one example. In the thermocouple, a voltage is measured which is dependent on the temperature. The relationship between voltage and temperature is not linear. Many controllers are indeed fitted with a calibration table so that one can linearize these signals. In this way, the controller algorithm really does get a signal which is proportional to temperature.

We shall look more closely at two other cases of nonlinear transducer where we often have to remember to compensate for the nonlinearities ourselves.

pH Control. pH meters are often used for measuring concentrations in a fluid. The pH meter is really measuring the concentration of hydrogen ions in the fluid. The relationship between hydrogen ion concentration and the concentration we actually wish to control is often nonlinear. This relationship often looks similar to that shown in Figure 5.11. Here we can see clearly how the gain varies at different working points. If we want to control our process just using a fixed controller, we should tune it according to the worst case. This means that we should give the controller a low enough gain so that we do not get stability problems where the process has its highest gain. The result is that we have to accept slow control in the area where the process has low gain. One solution to this problem is to tell the controller what the nonlinearity looks like, and let the controller compensate for it accordingly.

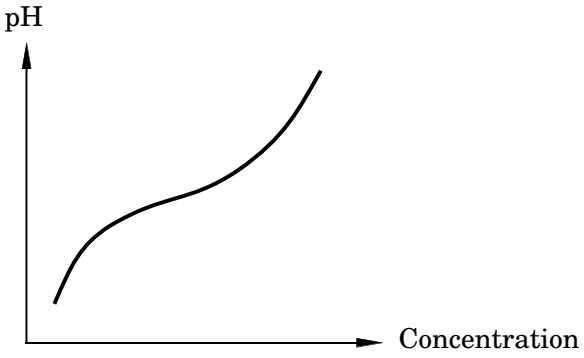


Figure 5.11 Example of the relationship between concentration and pH value

Flow Measurement. Flow is commonly measured by means of a differential pressure sensor. The pressure difference is not a linear function of flow, but the pressure is proportional to the square of the flow rate. In order to get a linear process, therefore, we have to take the square root of the signal before it reaches the PID algorithm (see Figure 5.12). The transducers themselves are sometimes provided with a square root algorithm, so that the signals from the device are already linear. In order to cover those cases where the transducers themselves do not linearize the signal, most process controllers have the ability to take the square root of the process variable.

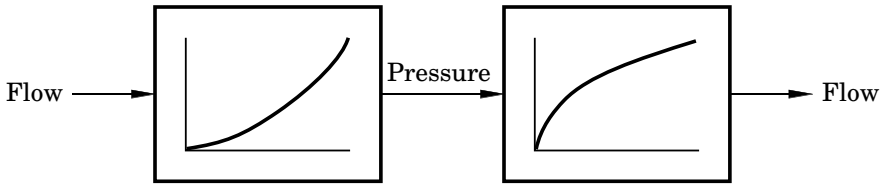


Figure 5.12 Compensation of the signal from a differential pressure sensor in order to give a flow signal

Nonlinear Process Dynamics

Process dynamics are often nonlinear. If the nonlinearity is smooth, it is possible to approximate the dynamics using a linear model around a stationary operating point. This method is known as *linearization* and will be described in the next lecture. Below we give two examples of nonlinear process dynamics.

Heating Processes. One example of a process where the dynamics vary depending on whether the control signal is rising or falling is in heating processes, e.g. furnaces. In these cases we can often increase the temperature in the controlled medium relatively quickly by raising the temperature of the furnace. Lowering of the temperature usually occurs without any active cooling, i.e. the medium is just allowed to cool naturally. In automatic control terms, this means that the process has a high gain (and a short time constant) when the control signal is increasing, and a low process gain (and a long time constant) when the control signal reduces.

Population Dynamics. Another example of processes with nonlinear dynamics is population dynamics. The dynamics of populations are interesting and important in many different areas of social and environmental policy. There are examples where new species have been introduced into new habitats, sometimes with disastrous results.

Let x be the population of a species at time t. A simple model is to assume that the birth rates and mortality rates are proportional to the total population. This gives the linear model

$$\frac{dx}{dt} = bx - dx = (b-d)x = rx, \quad x \geq 0 \tag{5.3}$$

where birth rate b and mortality rate d are parameters. The model gives an exponential increase if $b > d$ or an exponential decrease if $b < d$. A more realistic model is to assume that the birth rate decreases when the population is large. The following modification of the model has this property:

$$\frac{dx}{dt} = rx\left(1 - \frac{x}{x_c}\right), \quad x \geq 0, \tag{5.4}$$

where x_c is the carrying capacity of the environment. This model is called the logistic growth model.

A simulation of the logistic growth model is shown in Figure 5.13. The parameters are $r = 1$ and $x_c = 100$. It is seen that close to the (unstable) equilibrium point $x = 0$, the population grows exponentially. Close to the (stable) equilibrium point $x = 100$, however, the population saturates.

A more sophisticated model of population dynamics includes the effects of competing populations, where one species may feed on another. This is known as a *predator-prey model*.

Let $H(t)$ represent the number of hares (prey) and let $L(t)$ represent the number of lynxes (predator). The dynamics of the system are modeled as

$$\begin{aligned} \frac{dH}{dt} &= rH\left(1 - \frac{H}{k}\right) - \frac{aHL}{c+H}, \quad H \geq 0, \\ \frac{dL}{dt} &= b\frac{aHL}{c+H} - dL, \quad L \geq 0. \end{aligned} \tag{5.5}$$

In the first equation, r represents the growth rate of the hares, k represents the maximum population of the hares (in the absence of lynxes), a represents the interaction term that describes how the hares are diminished as a function of the lynx population and c controls the prey consumption rate for low hare population. In the second equation, b represents the growth coefficient of the lynxes and d represents

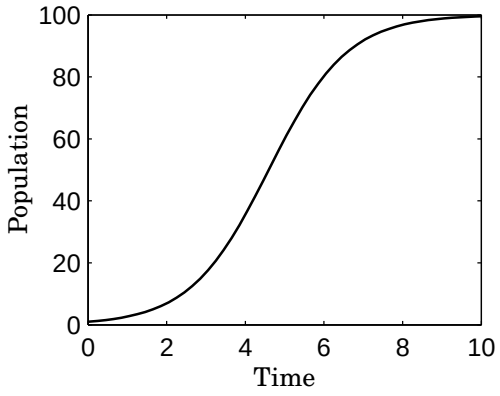


Figure 5.13 Simulation of the logistic growth model

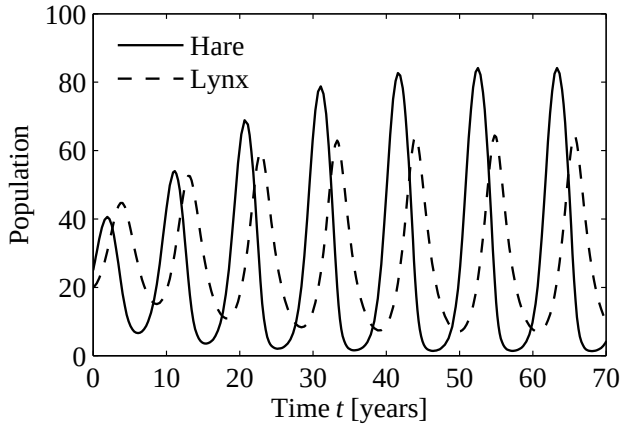


Figure 5.14 Simulation of the predator-prey model

the mortality rate of the lynxes. Note that the hare dynamics include a term that resembles the logistic growth model (5.4).

A simulation of the predator-prey model is shown in Figure 5.14. For the given parameters and initial conditions, the populations do not converge to stationary values. Rather, they oscillate and converge to a stable so called *limit cycle*. Limit cycles are a common feature of nonlinear dynamical systems.

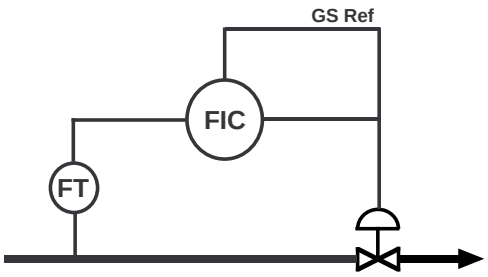
Gain scheduling

Above we have seen different types of nonlinearities—in the actuator, in the sensor, and in the process dynamics. If we want to able to operate the process over a wide range of values, it is often necessary to compensate for the nonlinearity. Sometimes, when we know the form of the nonlinearity exactly, we can do this very accurately. Examples of this are the compensation for thermocouples and taking the square root of the signal from a flow meter. In most cases, however, the compensation is not so simple. Also, it is often unnecessary for the compensation to be so accurate. In this section we shall show how we can compensate for nonlinearities by the use of gain scheduling.

In principle, gain scheduling is a table with a number of sets of controller parameters which are each used under different operating conditions. We could imagine, for example, that instead of one controller we had three, each of which worked within its own part of the overall working range, and a switch which selected the correct controller. Let us illustrate the principle with an example.

Example 5.1—Gain Scheduling with a Nonlinear Valve
Figure 5.15 shows the principle of gain scheduling in the control of a nonlinear valve. In the table we have filled in the PID controller parameters which correspond to the different working points. For small valve positions the lowest parameters are used; for medium valve openings the middle set of parameters is used, and the upper set of parameters is used for the large valve openings. One could naturally have more than three sets if a more accurate distribution was required.

The output signal from the controller is used in this case to determine when we should change controller parameters. It is of course the control signal which determines where on the valve characteristic we find ourselves, i.e. the gain of the process. □



K_c	T_i	T_d	GS Ref %
0.20	1.71	0.43	100
0.28	1.89	0.47	67
0.43	1.81	0.45	33
			0

Figure 5.15 Gain scheduling to compensate for a nonlinear valve

In the example above we allowed the control signal to determine which controller parameters were to be used. In other types of nonlinearity, other types of signal are used to determine this. Here are some examples of reference signals selected for gain scheduling:

1. Nonlinear valves, actuators: The control signal is used to form the reference signal for gain scheduling.
2. Nonlinear transducers or nonlinear dynamics: The process variable is used to produce the reference signal for gain scheduling
3. Production-dependent variations: Some external signal is used to form the gain scheduling reference signal. In the case of heating it is, for example, the change in the control signal which determines which parameters should be used.

We end by showing an example of the second choice of reference signal for gain scheduling.

Example 5.2—Gain Scheduling with a Nonlinear Tank
Figure 5.16 shows the principles of gain scheduling in the level control of a tank with a variable cross-sectional area. Because of the shape of the tank, the gain of the process varies depending on the level in the tank. In this case, at low levels we have a small cross-sectional area in the tank. This means that the level varies rapidly if we vary the flow into the tank. In other words, the process has a high gain when the level in the tank is low. At high tank levels we have a relatively large cross-sectional area in the tank. The process therefore has a low gain when the level in the tank is high.

In this example is would be desirable to have different controller parameters at different tank levels. The gain scheduling should therefore be controlled by the level measuring signal. The control signal, which determines the controlled flow into the tank, will not function as a reference signal in this case because it is not at all related to the tank level and therefore to the nonlinearity. □

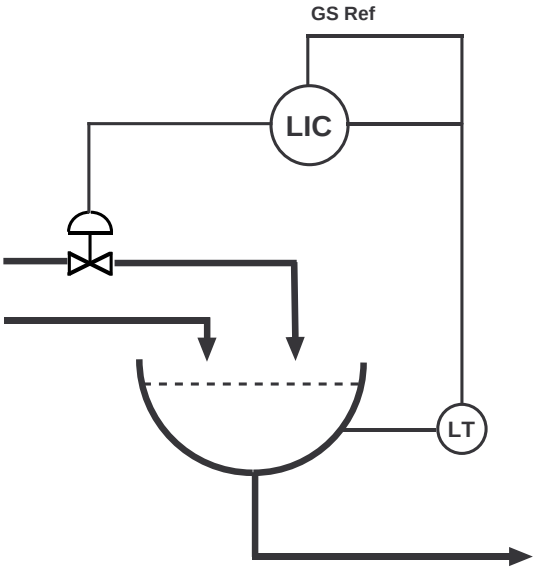


Figure 5.16 Gain scheduling in the level control of a nonlinear tank

6. Linearization. Feedback Systems—An Example

6.1 Linearization

For processes with nonlinear dynamics, it is possible to *linearize* the nonlinear state-space model around a stationary point. The resulting linear approximation is valid in a small region around the stationary point. The linearized model is useful for the analysis and design of feedback control loops.

Stationary Points

Consider a nonlinear dynamical system described by a state-space model,

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned}$$

The *stationary points* or *equilibria* of the system are all points (x^0, u^0) where

$$f(x^0, u^0) = 0$$

that is, the points where the time derivatives of all state variables are zero.

Linearization—The Scalar Case

To explain the idea of linearization, we start with a simple scalar case. Assume a nonlinear dynamical system in one variable,

$$\frac{dx}{dt} = f(x)$$

We perform the linearization in four steps:

1. Find a stationary point x^0, where $f(x^0) = 0$.
2. Approximate $f(x)$ with a straight line through x^0:

$$f(x) \approx \underbrace{f(x^0)}_{=0} + \underbrace{\frac{df}{dx}(x^0)}_{=a}(x - x^0)$$

3. Introduce a new state variable that measures the deviation from the stationary point:

$$\Delta x = x - x^0$$

4. The linearized system can then be written

$$\frac{d\Delta x}{dt} = a\Delta x$$

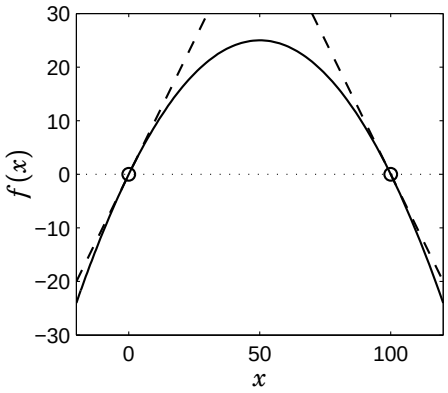


Figure 6.1 Linearization of the logistic growth model.

EXAMPLE 6.1—LOGISTIC GROWTH MODEL
Consider the logistic growth model from Section 5.2 with birth rate $r = 1$ and carrying capacity $x_c = 100$:

$$\frac{dx}{dt} = x\left(1 - \frac{x}{100}\right) = f(x)$$

The function $f(x)$ is illustrated in Figure 6.1. There are two stationary points: $x^0 = 0$ and $x^0 = 100$. In either of these points, we can approximate $f(x)$ by a straight line as shown in the figure. For $x^0 = 0$ we get the slope $a = 1$, and for $x^0 = 100$ we get the slope $a = -1$. Introducing the new state variable $\Delta x = x - x^0$ the linearized system can for the two different cases be written as

$$\frac{d\Delta x}{dt} = \begin{cases} \Delta x, & x^0 = 0 \\ -\Delta x, & x^0 = 100 \end{cases}$$

From the analysis, we can draw the conclusion that $x^0 = 0$ represents an unstable equilibrium, while $x^0 = 100$ represents a stable equilibrium.

□

Linearization—The General Case

We will now demonstrate how to linearize a general nonlinear state-space model

$$\dot{x} = f(x,u)$$
$$y = g(x,u)$$

where f and g are smooth nonlinear functions of x and u. The procedure consists of the following four steps:

1. Determine a stationary point (x^0, u^0) around which we shall approximate the system.
2. Make Taylor series expansions of f and g around (x^0, u^0). Keep only the first order terms.

$$f(x,u) \approx f(x^0,u^0) + \frac{\partial}{\partial x} f(x^0,u^0)(x - x^0) + \frac{\partial}{\partial u} f(x^0,u^0)(u - u^0)$$
$$g(x,u) \approx g(x^0,u^0) + \frac{\partial}{\partial x} g(x^0,u^0)(x - x^0) + \frac{\partial}{\partial u} g(x^0,u^0)(u - u^0)$$

   Note that $f(x^0,u^0) = 0$ and introduce the notation $y^0 = g(x^0,u^0)$.

3. Introduce new variables

$$\Delta x = x - x^0$$
$$\Delta u = u - u^0$$
$$\Delta y = y - y^0$$

4. The state space equations in the new variables are given by

$$\Delta\dot{x} = \dot{x} - \dot{x}^0 = \dot{x} = f(x,u) \approx \frac{\partial}{\partial x} f(x^0,u^0)\Delta x + \frac{\partial}{\partial u} f(x^0,u^0)\Delta u = A\Delta x + B\Delta u$$
$$\Delta y = y - y^0 = g(x,u) - y^0 \approx \frac{\partial}{\partial x} g(x^0,u^0)\Delta x + \frac{\partial}{\partial u} g(x^0,u^0)\Delta u = C\Delta x + D\Delta u$$

Note that f and g are in general vector functions. If we for instance deal with a system with the two state variables, one measurement signal y, and one control signal, it follows that

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},\ f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix},\ \frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix},\ \frac{\partial f}{\partial u} = \begin{pmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{pmatrix},\ \frac{\partial g}{\partial x} = \begin{pmatrix} \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} \end{pmatrix}$$

EXAMPLE 6.2—LINEARIZATION OF A NONLINEAR TANK

Figure 6.2 shows a conical tank process with nonlinear dynamics. Assuming that the tank walls form an angle that makes the diameter of the fluid surface equal to the height of the fluid, the fluid volume becomes

$$V = \frac{\pi h^3}{12}$$

A mass balance for the tank shows that the change in volume equals the inflow minus the outflow

$$\frac{dV}{dt} = q_{in} - q_{out}$$

We are interested in changes of the fluid level. These are obtained through

$$\frac{dV}{dt} = \frac{dV}{dh}\frac{dh}{dt} = \frac{\pi h^2}{4}\frac{dh}{dt}$$

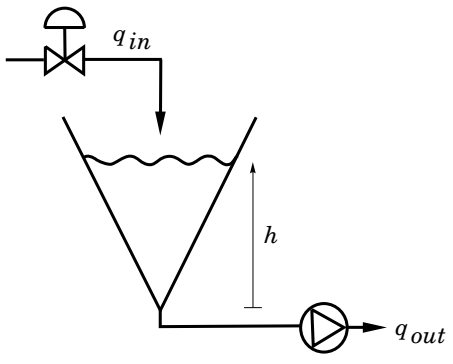


Figure 6.2 The nonlinear tank in Example 6.2.

i.e.

$$\frac{dh}{dt} = \frac{4}{\pi h^2}\left(q_{in} - q_{out}\right)$$

Introduce the familiar notation

$$x = y = h, \quad u = q_{in}$$

and assume that q_{out} is constant. The resulting state space description is given by

$$\begin{aligned}\dot{x} &= \frac{4}{\pi x^2}\left(u - q_{out}\right) = f(x,u)\\ y &= x = g(x,u)\end{aligned}$$

Now we linearize this system according to the procedure outlined above.

1. Stationary points.

   $$f(x^0,u^0) = 0 \Leftrightarrow u^0 = q_{out}$$

   This criterion simply means that the inflow must equal the outflow in stationarity. We obtain no restrictions on x^0, i.e. we can linearize around any given level.

2. Taylor series expansion.

   $$\begin{aligned}f(x,u) &\approx -\frac{8}{\pi(x^0)^3}(u^0 - q_{out})(x - x^0) + \frac{4}{\pi(x^0)^2}(u - u^0) = \frac{4}{\pi(x^0)^2}(u - u^0)\\ g(x,u) &\approx y^0 + 1\cdot(x - x^0) + 0\cdot(u - u^0) = y^0 + (x - x^0)\end{aligned}$$

3. New variables.

   $$\begin{aligned}\Delta x &= x - x^0\\ \Delta u &= u - u^0\\ \Delta y &= y - y^0\end{aligned}$$

4. State space equations in the new variables.

   $$\begin{aligned}\dot{\Delta x} &= f(x,u) \approx \frac{4}{\pi(x^0)^2}\Delta u\\ \Delta y &= g(x,u) - y^0 = \Delta x\end{aligned}$$

We observe that the linearization in this case meant replacing the division by h^2 in the nominal case by a division by $(x^0)^2$, i.e. the level around which the system was linearized. The approximation will be accurate as long as the deviation Δx is small.

□

6.2 Feedback Systems—An Example

The general idea of feedback was introduced in Lecture 1. The concept will now be elaborated upon and properties of feedback systems will be investigated. We start by analyzing the effects of feedback on a simple example—a heated stirred tank. The process is shown in Figure 6.3. Assuming constant heat capacity and mass density we obtain the model

$$V\rho C_p \frac{dT(t)}{dt} = q\rho C_p(T_{in}(t) - T(t)) + Q(t)$$

where ρ is the density, V the volume of the tank, C_p specific heat constant of the liquid, and q the volume flow rate in and out the tank. Let the input be the heat Q into the system and the output the liquid temperature T in the tank. The disturbance is the input temperature T_{in}. It is convenient to regard all variables as deviations from stationary values. For instance, $T_{in} = 0$ then implies that T_{in} is at its normal value. The system can also be written as

$$T_1 \frac{dT(t)}{dt} + T(t) = K_1 Q(t) + T_{in}(t) \tag{6.1}$$

The process gain K_1 and the time constant T_1 of the process are given by

$$K_1 = \frac{1}{q\rho C_p} \qquad T_1 = \frac{V}{q}$$

Notice that the time constant T_1 has dimension time. Taking the Laplace transform of the differential equation (6.1) gives

$$T(s) = \frac{1}{1 + sT_1}\Big(K_1 Q(s) + T_{in}(s)\Big) \tag{6.2}$$

The system can be represented by the block diagram in Figure 6.4. We will assume that the parameters and the time scale are such that $K_1 = 1$ and $T_1 = 1$. The purpose is to use the input $Q(t)$ to keep the output $T(t)$ close to the reference value $T_{ref}(t)$ despite the influence of the disturbance T_{in}.

We will now investigate the dynamic behavior of the output temperature, when the reference value T_{ref} and the disturbance T_{in} are changed.

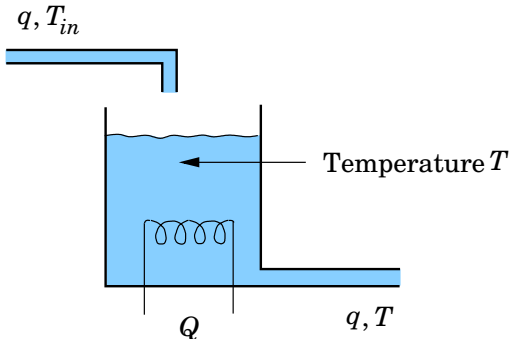


Figure 6.3 Stirred tank process.

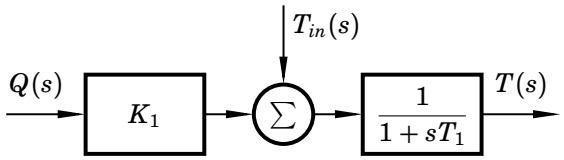


Figure 6.4 Block diagram of the stirred tank process.

Open-Loop Response

Assume that the input Q is zero, i.e. no control, and that the disturbance is a unit step, i.e., the inlet temperature is suddenly increased. Notice that the desired value in this case is $T_{ref} = 0$. We do not want T to deviate from this value.

The Laplace transform of the output is

$$T(s) = \frac{1}{1+sT_1} \cdot \frac{1}{s}$$

and the time response is

$$T(t) = 1 - e^{-t/T_1}$$

The temperature in the tank will approach the inlet temperature as a first order system. The change in the input temperature will cause a change in the tank temperature. The new steady state value will be the same as the input temperature. Physically this is concluded since all the liquid will eventually by replaced by liquid of the higher temperature. This is also found mathematically from (6.1) by putting the derivative equal to zero and solving for T. The time to reach the new steady state value is determined by the time constant of the open loop system T_1. It takes 3—5 time constants for the transient to disappear.

Proportional Control

A first attempt to control the system is to let the input heat Q be proportional to the error between the liquid temperature and the reference temperature, i.e., to use the P controller

$$Q(t) = K\big(T_{ref}(t) - T(t)\big) = Ke(t)$$

In the Laplace domain, the controller is given by

$$Q(s) = K\big(T_{ref}(s) - T(s)\big) = KE(s) \tag{6.3}$$

If the temperature is too low then the heat is increased in proportion to the deviation from the desired temperature. The parameter K is the (proportional) gain of the controller. Combining (6.2) and (6.3) we have

$$T(s) = \frac{1}{1+sT_1}\Big(K_1K\big(T_{ref}(s) - T(s)\big) + T_{in}(s)\Big)$$

Solving for $T(s)$, we obtain the input–output relationship

$$T(s) = \frac{K_1K}{1+sT_1+K_1K}T_{ref}(s) + \frac{1}{1+sT_1+K_1K}T_{in}(s)$$

The closed-loop system is of first order with a pole located in

$$s = -\frac{1+K_1K}{T_1}$$

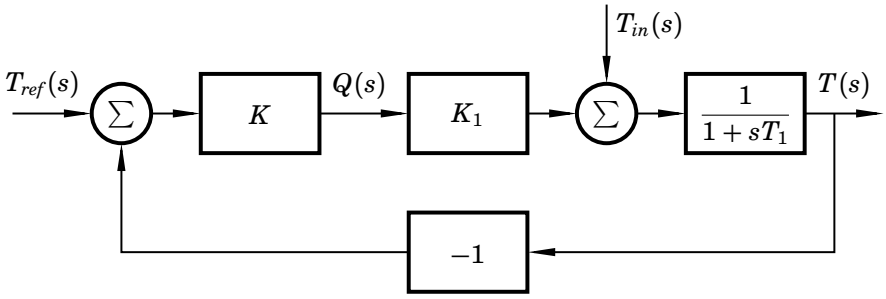


Figure 6.5 Proportional control of the stirred tank process.

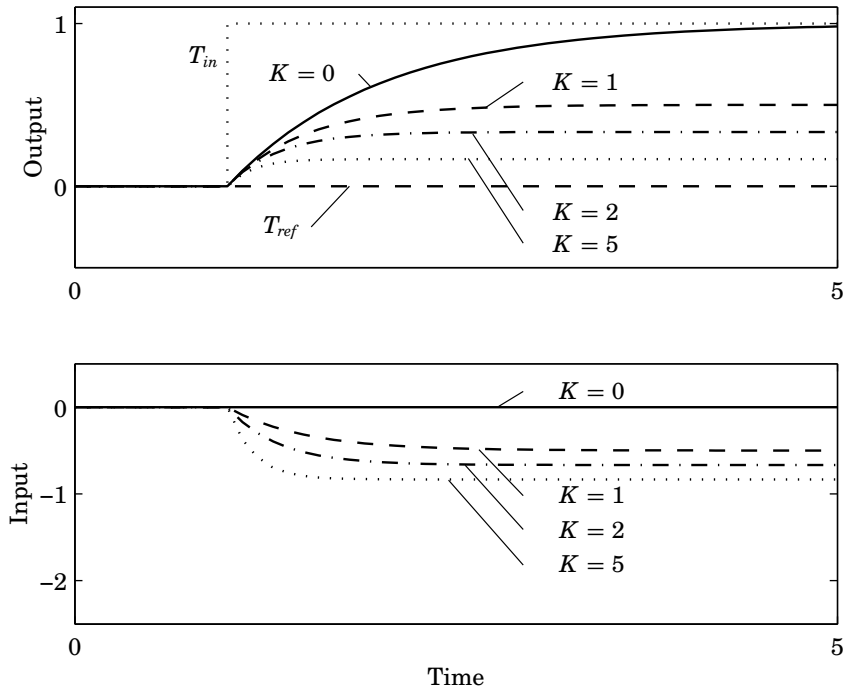


Figure 6.6 Simulation of the stirred tank under P control with $K = 0$, 1, 2, and 5. The desired reference is $T_{ref} = 0$, and the disturbance is a unit step at $t = 0$. The case $K = 0$ gives the open loop response.

By proper selection of the controller parameter K, the pole can be placed anywhere on the negative real axis. This design method is known as *pole placement*. A block diagram of the closed-loop system is shown in Figure 6.5.

Figure 6.6 shows the measurement signal and the control signal when using a P controller for different values of the controller gain. The steady state error decreases with increasing K. We can also note that the response becomes faster when the gain is increased.

Proportional and Integral Control

To eliminate the steady state error we have to change the controller type. One possibility is to introduce a PI controller. The controller can be written as

$$Q(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau\right) \tag{6.4}$$

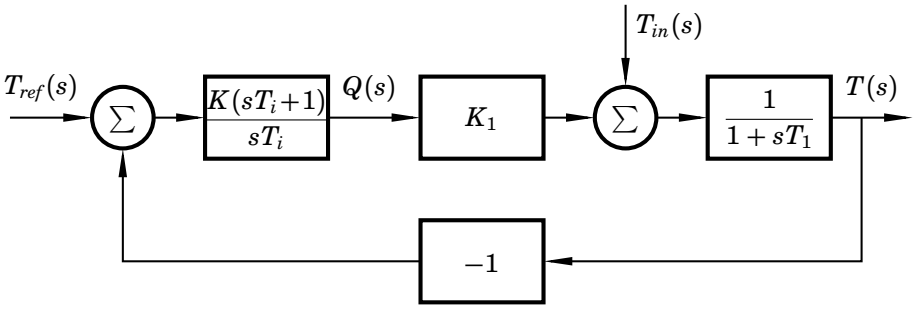


Figure 6.7 Proportional–integral control of the stirred tank process.

where $e(t) = T_{ref}(t) - T(t)$. In the Laplace domain, the PI controller is given by

$$Q(s) = K\left(1 + \frac{1}{sT_i}\right)E(s)$$

The Laplace transform of the output is given by

$$T(s) = \frac{1}{1+sT_1}\left(\frac{K_1K(sT_i+1)}{sT_i}\Big(T_{ref}(s) - T(s)\Big) + T_{in}(s)\right)$$

Solving for $T(s)$, we obtain the input–output relationship

$$T(s) = \frac{K_1K(sT_i+1)}{s^2T_1T_i + s(K_1K+1)T_i + K_1K}T_{ref}(s) + \frac{sT_i}{s^2T_1T_i + s(K_1K+1)T_i + K_1K}T_{in}(s)$$

The block diagram of the closed-loop system is shown in Figure 6.7. The closed-loop system is now of second order. By proper selection of the controller parameters K and T_i, we can place the poles according to any characteristic polynomial. Figure 6.8 shows the response to disturbances when $K = 1$ and for different values of T_i.

When the integral term is active ($T_i \neq \infty$) the steady state error becomes zero. A too small value of T_i leads to an oscillatory response. Figure 6.9 shows the closed loop performance when the reference value is changed from 0 to 1 at $t = 0$ and when the disturbance T_{in} is a unit step at $t = 5$. The regulator is a PI controller with $K = 1.8$ and $T_i = 0.45$. The response is fast and well damped, without excessive control signals.

Sensitivity Reduction and Robustness

Feedback will decrease the sensitivity of the closed loop system to changes in the dynamics of the process. Assume that the process is controlled by a proportional controller with gain K. The closed-loop static gain is then given by

$$K_{cl} = \frac{K_1K}{1+K_1K}$$

Assume that $K_1K = 5$. A $\pm 10\%$ change in K_1K will then give $K_{cl} = [0.82, 0.85]$. The sensitivity will be less when K_1K is increased. For instance $K_1K = 10$ gives $K_{cl} = [0.9, 0.92]$ after a $\pm 10\%$ change in K_1K.

The process gain and the time constant are inversely proportional to the flow q. Figure 6.10 shows the closed loop performance after a step in the reference value when a PI-controller is used and when the flow through the tank q is changed $\pm 20\%$ compared to the nominal value. Feedback brings robustness towards process variations and disturbances in the closed-loop system.

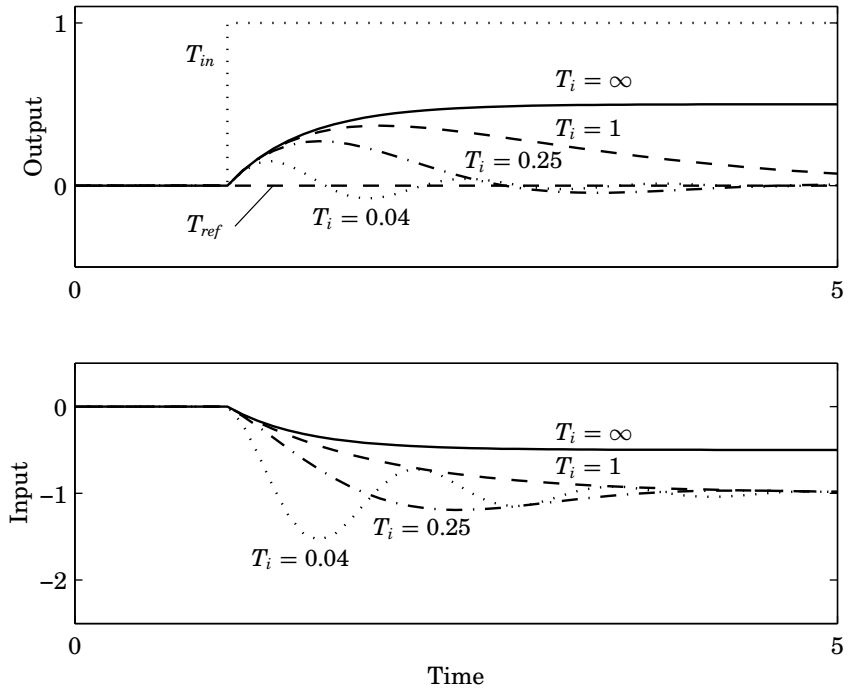


Figure 6.8 Simulation of the stirred tank under PI control with $K = 1$ and $T_i = \infty$, 1, 0.25, and 0.04. The desired reference value is $T_{ref} = 0$ and the disturbance is a unit step at $t = 1$.

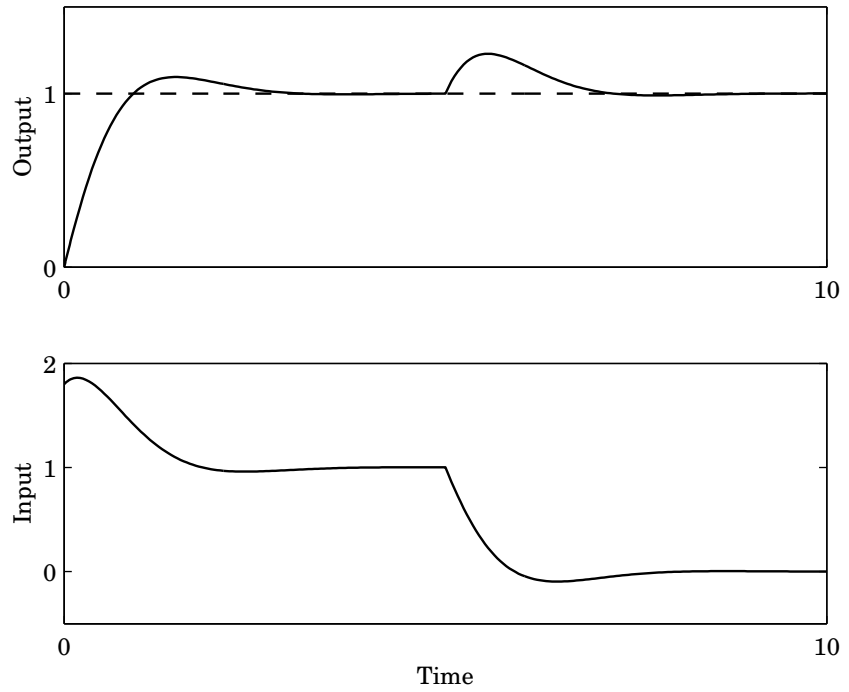


Figure 6.9 Reference and load disturbance for the temperature control system using a PI-controller $K = 1.8$ and $T_i = 0.45$. The liquid temperature and the control signal are shown.

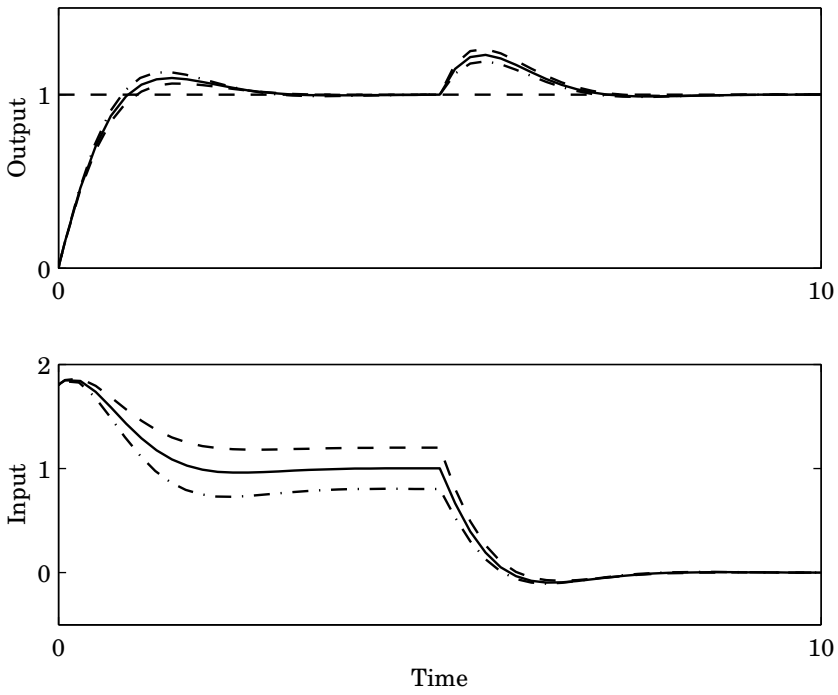


Figure 6.10 The sensitivity to changes in the flow through the tank, when the system is controlled with a PI-controller with $K = 1.8$ and $T_i = 0.45$. Increase of 20% (dashed), decrease of 20% (dash-dotted), and the nominal case (full). Compare Fig 6.9.

Unmodeled Dynamics

We will now increase the complexity of the system model by assuming that the temperature is measured using a sensor with dynamics. It is assumed that the measured value T_m is related to the true temperature by the transfer function

$$T_m(s) = \frac{1}{1 + T_s s} T(s)$$

The sensor has the time constant T_s, and the gain is one. Figure 6.11 shows the temperature and the control signal when the PI-controller (6.4) is used with T_m instead of T. The closed loop system will be unstable for some parameter combinations. The stability can be analyzed by studying the coefficients of the characteristic polynomial of the closed-loop system, see Section 3.4.

Summary

We have seen that feedback can be used to change the transient and steady state properties of a closed loop system. Also the influence of the disturbances are reduced through feedback. To obtain a steady state error that is zero it is necessary to have an integrator in the controller. Finally, it was found that the closed loop system may become unstable due to unmodeled dynamics.

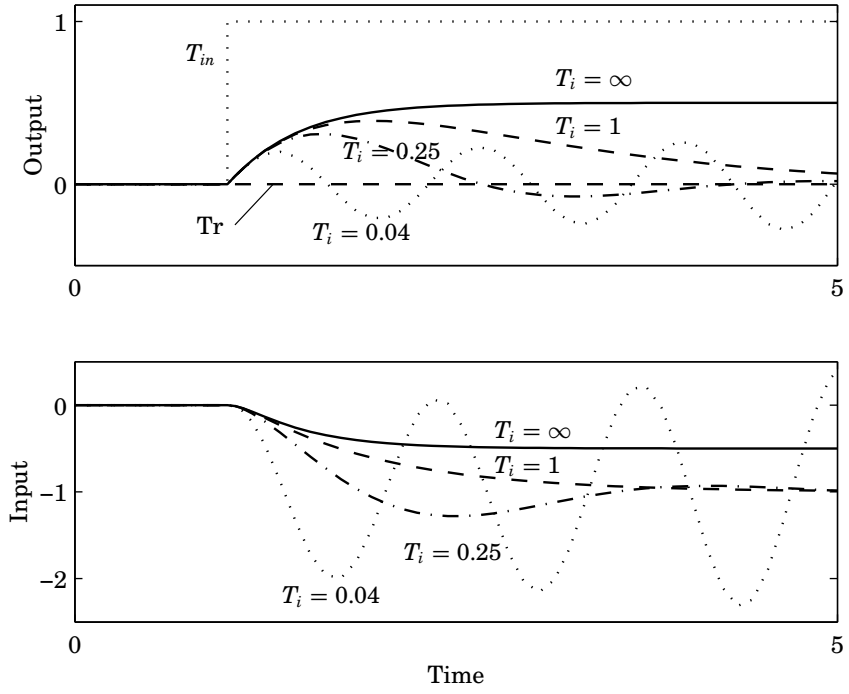


Figure 6.11 Temperature control using PI-controller and sensor dynamics. The liquid temperature and the control signal are shown when T_{in} is a unit step, $T_s = 0.1$, $K = 1$ and $T_i = \infty$, 1, 0.25, and 0.04. Compare Figure 6.8

7. Feedback Systems

7.1 Analysis of Stationary Errors

Two important tasks for a feedback control system are to make the output follow the reference signal (the servo problem) and to eliminate disturbances (the regulator problem). The reference value and the disturbance can be modeled as piecewise constant signals, ramps, etc. An important question is how large the stationary errors will be for different kinds of reference signals and disturbances. We know from experience that the integral part in a PID controller can eliminate stationary errors due to a constant reference value or a constant disturbance. The general situation will now be analyzed.

Consider the feedback system in Figure 7.1. The Laplace transform of the control error is

$$E(s) = \frac{1}{1 + G_0(s)} R(s) - \frac{G_p(s)}{1 + G_0(s)} V(s)$$

where $G_0 = G_p G_c$. Applying the final value theorem then gives the stationary error

$$e(\infty) = \lim_{s \to 0} sE(s)$$

This calculation is valid as long as $sE(s)$ is asymptotically stable.

To gain more insight we will assume that

$$G_0(s) = \frac{K(1 + q_1 s + \cdots + q_{m-1} s^{m-1})}{s^n (1 + p_1 s + \cdots + p_m s^m)} = \frac{KQ(s)}{s^n P(s)}$$

The open loop system is parameterized to show the number of integrators, (i.e., poles in the origin) and a gain K. The open loop system has n integrators. Notice that $Q(0) = P(0) = 1$. The parameter K can be interpreted as the static gain when the integrators have been removed. Further we let

$$G_c(s) = \frac{K_1 Q_1(s)}{s^m P_1(s)}$$

$$G_p(s) = \frac{K_2 Q_2(s)}{s^{n-m} P_2(s)}$$

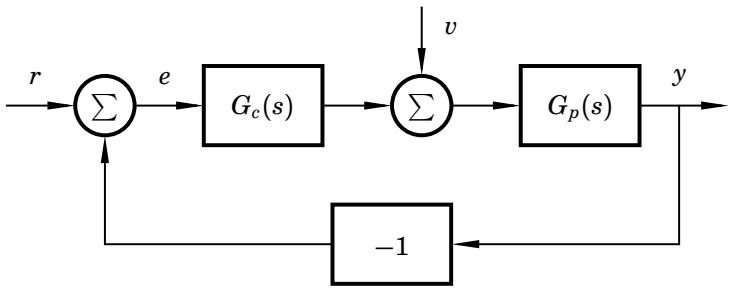


Figure 7.1 Simple feedback control system with reference signal r and disturbance signal v.

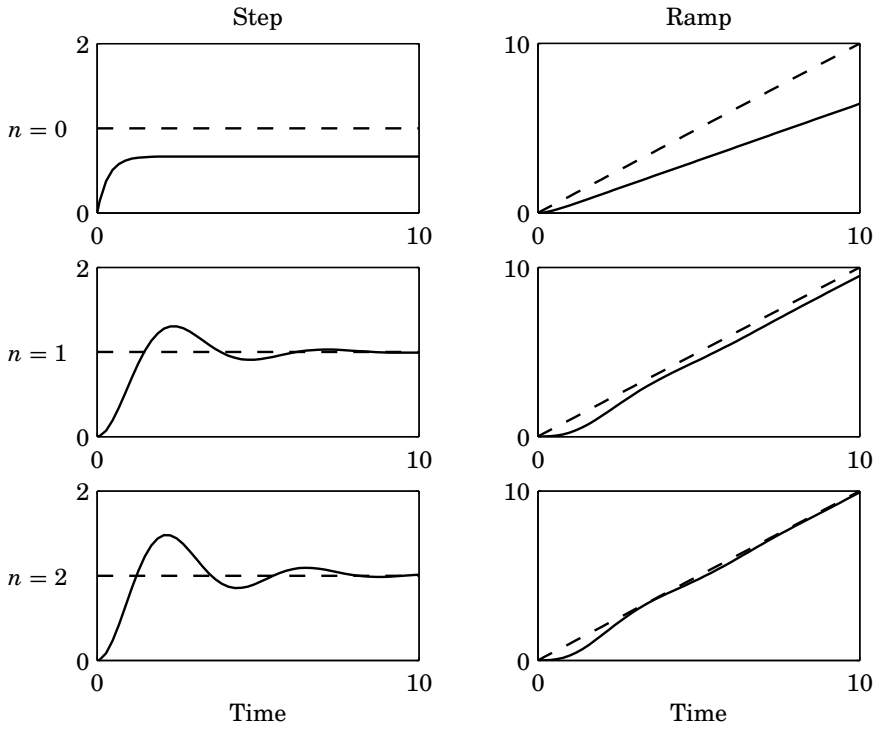


Figure 7.2 The output and reference values as functions of time for different number of integrators, n, when the reference signal is a step or a ramp.

where $Q_1(0) = Q_2(0) = P_1(0) = P_2(0) = 1$ and $n - m \geq 0$. We first investigate the influence of a reference value (and set $v = 0$). Let $r(t)$ be a step of size a, i.e. $R(s) = a/s$. The final value theorem then gives

$$e(\infty) = \lim_{s\to 0} \frac{s^n P(s) a}{s^n P(s) + KQ(s)} = \begin{cases} a/(1+K) & n = 0 \\ 0 & n \geq 1 \end{cases}$$

When $r(t)$ is a ramp, i.e. $R(s) = b/s^2$, we get

$$e(\infty) = \lim_{s\to 0} \frac{s^n P(s)}{s^n P(s) + KQ(s)} \cdot \frac{b}{s} = \begin{cases} \infty & n = 0 \\ b/K & n = 1 \\ 0 & n \geq 2 \end{cases}$$

It is possible to continue for more complex reference signal, but the pattern is obvious. The more complex the signal is the more integrators are needed to get zero steady state error. The calculations are summarized in Figure 7.2. Let us now make the same investigation for disturbances when the reference value is zero. Now

$$\begin{aligned} E(s) &= -\frac{G_p(s)}{1 + G_0(s)} V(s) \\ &= -\frac{s^n P(s)}{s^n P(s) + KQ(s)} \cdot \frac{K_2 Q_2(s)}{s^{n-m} P_2(s)} V(s) \\ &= -\frac{s^m K_2 P_1(s) Q_2(s)}{s^n P(s) + KQ(s)} \cdot V(s) \end{aligned}$$

Let the disturbance be a step, i.e. $V(s) = a/s$. The final value theorem gives

$$e(\infty) = \lim_{s\to 0} -\frac{s^m K_2 P_1(s) Q_2(s) a}{s^n P(s) + KQ(s)} = \begin{cases} -aK_2/(1+K) & m = 0, n = 0 \\ -a/K_1 & m = 0, n \geq 1 \\ 0 & m \geq 1 \end{cases}$$

When $v(t)$ is a ramp, i.e. $V(s) = b/s^2$, we get

$$e(\infty) = \lim_{s\to 0} -\frac{s^m K_2 P_1(s) Q_2(s)}{s^n P(s) + KQ(s)} \cdot \frac{b}{s} = \begin{cases} -\infty & m = 0 \\ -b/K_1 & m = 1 \\ 0 & m \geq 2 \end{cases}$$

Once again the number of integrators and the gain determine the value of the stationary errors.

Where Should the Integrators be Located?

It was shown above that it is important to have integrators in the system to eliminate stationary errors. If the process does not have any integrator we can introduce integrators in the regulator. Consider the system in Figure 7.1. If r is a step then e will be zero in steady state if there is an integrator in either the controller $G_c(s)$ or in the process $G_p(s)$. If v is a step it is necessary to have an integrator in $G_c(s)$ to obtain zero steady state. It is not sufficient with an integrator in $G_p(s)$. To eliminate load disturbances it is necessary to introduce integrators in the controller even if the process contains integrators.

7.2 Sensitivity

Above it has been shown how feedback can be used to reduce the influence of disturbances entering into the system. We will now show how feedback can be used to make the system less sensitive to variations in parameters in the process.

EXAMPLE 7.1—NONLINEAR VALVE

Control of flows is very common in chemical process control. Control valves have, however, often a nonlinear characteristic relating the flow to the opening of the valve. For simplicity, we assume that the valve is described by static nonlinear relation.

$$y = g(u) = u^2 \qquad 0 \leq u \leq 1$$

where u is the valve opening and y is the flow through the valve. Small changes in the opening Δu will give small changes in the flow Δy. The change is proportional to the derivative of the valve characteristic, i.e.

$$\Delta y = g'(u)\Delta u = 2u\Delta u$$

The gain thus changes drastically depending on the opening. When $u = 0.1$ the gain is 0.2 and when $u = 1$ it is 2. This nonlinearity can be reduced by measuring the flow and controlling the valve position, see Figure 7.3. Assume that there are no dynamics in the system and that the controller is a proportional controller with gain K. The system is then described by

$$\begin{aligned} e &= r - y \\ y &= g(Ke) \end{aligned}$$

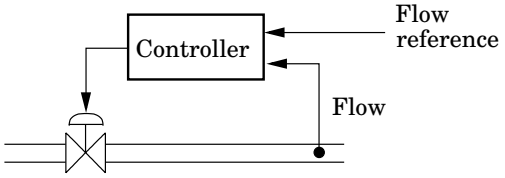


Figure 7.3 Flow control.

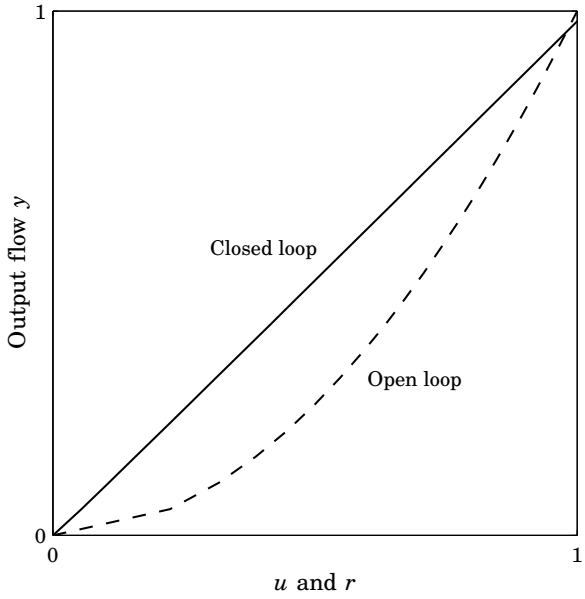


Figure 7.4 Input-output relations for the open loop system and the closed loop system when $K = 50$. Notice that the input for the open loop system is r for the closed loop system.

This gives the relation

$$r = y + \frac{1}{K} g^{-1}(y) = y + \frac{1}{K}\sqrt{y} = f(y)$$

where g^{-1} is the inverse function. The gain of the closed loop system is given through

$$\Delta r = f'(y)\Delta y$$

or

$$\Delta y = \frac{1}{f'(y)}\Delta r = \frac{2K\sqrt{y}}{1 + 2K\sqrt{y}}\Delta r = \frac{2Ku}{1 + 2Ku}\Delta r$$

If K is sufficiently high the gain of the closed loop system will be close to 1 and almost independent of u. Figure 7.4 shows the input output relations for the open loop and the closed loop systems. □

We will now treat the more general case shown in Figure 7.5. The transfer function from r to y is given by

$$G_{cl}(s) = \frac{G_f(s)G_o(s)}{1 + G_o(s)G_y(s)} \tag{7.1}$$

We will first investigate how variations in the transfer functions influence the re-

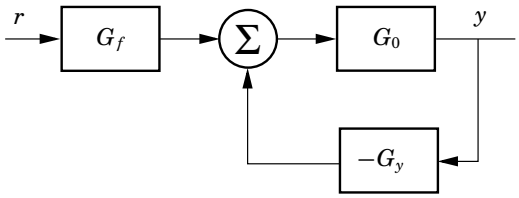


Figure 7.5 Closed-loop system.

sponse from the reference value to the output. One way to do this is to determine the sensitivity when there is a small change in the transfer function. Let a transfer function G depend on a transfer function H, i.e. $G = G(H)$. We define the *relative sensitivity* of G with respect to H as

$$S_H = \frac{dG}{dH} \cdot \frac{H}{G}$$

This can be interpreted as the relative change in G, dG/G, divided by the relative change in H, dH/H. For the transfer function G_{cl} in (7.1) we have

$$\begin{aligned} S_{G_f} &= 1 \\ S_{G_o} &= \frac{1}{1 + G_o G_y} \\ S_{G_y} &= -\frac{G_o G_y}{1 + G_o G_y} \end{aligned}$$

It is seen that a relative change in G_f will give the same relative change in G_{cl}. As long as the loop gain $G_o G_y$ is large then S_{G_o} will be small. This will, however, cause S_{G_y} to become close to one. From a sensitivity point of view it is crucial that the transfer functions G_f and G_y are accurate. These transfer functions are determined by the designer and can be implemented using accurate components.

8. Analysis in the Frequency Domain

8.1 Frequency Response

Periodic or oscillatory signals are not very common in chemical engineering applications. For electrical and mechanical processes it is more natural to discuss responses of systems to sinusoidal inputs. The frequency domain approach is, however, a very useful way to get an additional way to analyze a dynamical system. The frequency response that will be introduced below can be interpreted as a frequency dependent "gain" of the process. The frequency response makes it possible to determine which frequencies that will be amplified or attenuated. Based on the frequency response of an open loop system it is easy to determine if a *closed loop* system is stable or not. It is also possible to determine suitable controllers based on the frequency response. Finally, the method gives a new interpretation of the transfer function of the system.

Figure 8.1 shows a block digram of a process with disturbances. The input $u(t)$ and the disturbance $v(t)$ act on the process via the transfer functions $G_p(s)$ and $G_d(s)$ respectively. The disturbance can for instance be a step, an impulse, or a periodic signal. The steady state influence of a step disturbance is given by the steady state gain $G_d(0)$. The disturbance may also be periodic. Examples of periodic or almost periodic disturbances are

- Measurement noise due to hum from the power supply.
- Unbalances in rotating mechanical parts.
- Influence of daily variation in outdoor temperature.
- Variation in feed concentration to a unit process.

In Section 5.1 we showed how a linear system could be characterized by giving its step or impulse response. In this section we will investigate how periodic inputs or disturbances in steady state are influencing the output of the process.

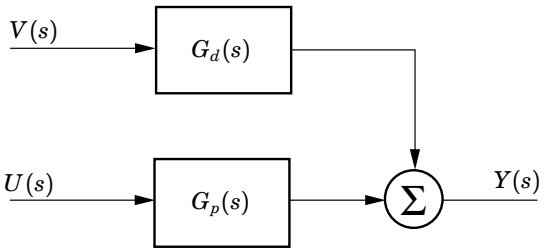


Figure 8.1 Block diagram of process with input and disturbances.

EXAMPLE 8.1—FREQUENCY RESPONSE OF A FIRST ORDER SYSTEM
Assume that the process is described by

$$Y(s) = G(s)U(s) = \frac{b}{s+a}U(s) \tag{8.1}$$

where $a > 0$ and assume that the input signal is a sinusoidal signal $\sin \omega t$. The Laplace transform of the input is

$$U(s) = \frac{\omega}{s^2 + \omega^2}$$

This gives

$$\begin{aligned} Y(s) &= \frac{b}{s+a}\frac{\omega}{s^2+\omega^2} \\ &= \frac{b\omega}{\omega^2+a^2}\left(\frac{1}{s+a} + \frac{a-s}{s^2+\omega^2}\right) \end{aligned}$$

The first part on the right hand side has an inverse transform which is a decaying exponential since $a > 0$. This part will thus vanish as t increases. The second term represents sinusoidal signals with frequency ω. Asymptotically we get

$$\begin{aligned} y_a(t) &= b\left(\frac{a}{\omega^2+a^2}\sin \omega t - \frac{\omega}{\omega^2+a^2}\cos \omega t\right) \\ &= A\sin(\omega t + \varphi) \end{aligned}$$

where

$$A = \frac{b}{\sqrt{\omega^2+a^2}}$$

$$\varphi = -\arctan\frac{\omega}{a}$$

The derivation above shows that asymptotically the response to a sinusoidal input is a sinusoidal output with the *same* frequency but with a change in amplitude and phase. The excitation with a single frequency forces the output to vary with the same frequency. To derive the result we used that the system is linear and stable. We can thus regard the sinusoidal response determined by A and φ as a frequency dependent gain and phase shift. The responses of the system (8.1) to sinusoidal inputs of different frequencies are shown in Figure 8.2.

□

Frequency Response of a Linear System

Assume that the input signal is given by a sinusoid of frequency ω, i.e.

$$u(t) = \sin \omega t$$

Let $G(s)$ be the transfer function and $h(t)$ the impulse response of the studied system. Since the Laplace transform of an impulse is simply 1, it holds that

$$G(s) = \mathcal{L}(h(t)) = \int_0^\infty e^{-st}h(t)dt$$

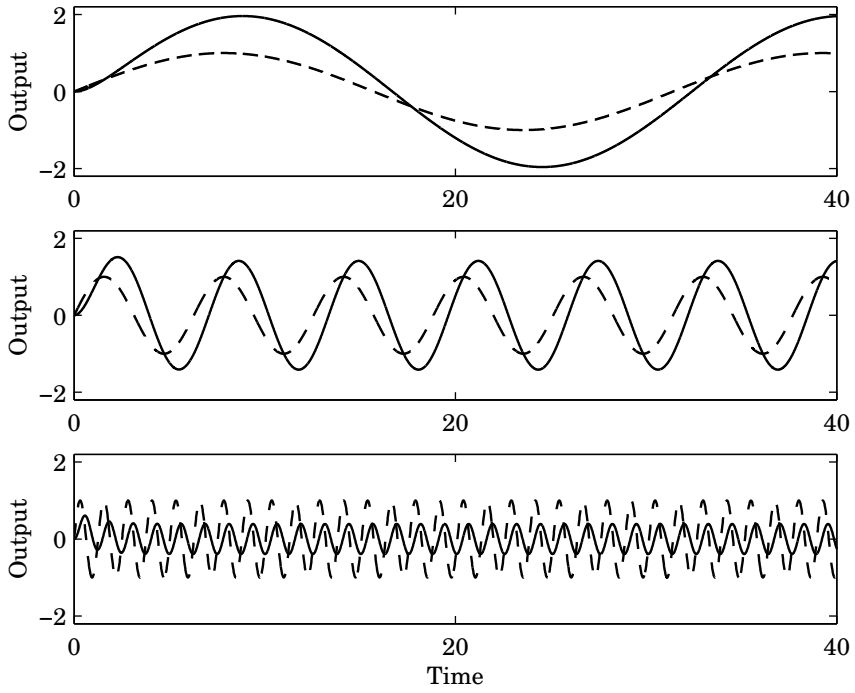


Figure 8.2 The input (dashed) and output (full) when the signal $\sin \omega t$ is applied to the system (8.1) with zero initial value, when $a = 1$, $b = 2$, and (a) $\omega = 0.2$; (b) $\omega = 1$; (c) $\omega = 5$.

We exploit this in order to find which output we obtain when the input is given by a sinusoid. When the transient has died out, i.e. when $t \to \infty$, Equation (5.2) yields

$$\begin{aligned} y(t) &= \int_0^t h(t-\tau')u(\tau')d\tau' = [\tau = t-\tau'] = \int_0^t h(\tau)u(t-\tau)d\tau \\ &= \int_0^t h(\tau)\sin\omega(t-\tau)d\tau = \operatorname{Im}\int_0^t h(\tau)e^{i\omega(t-\tau)}d\tau \\ &= \operatorname{Im}\int_0^t h(\tau)e^{-i\omega\tau}d\tau e^{i\omega t} = [t\to\infty] = \operatorname{Im} G(i\omega)e^{i\omega t} \\ &= |G(i\omega)|\operatorname{Im} e^{i\arg G(i\omega)}e^{i\omega t} = |G(i\omega)|\sin(\omega t + \arg G(i\omega)) \end{aligned}$$

This means that when the input signal is given by $u(t) = \sin(\omega t)$, the output signal becomes

$$y(t) = A\sin(\omega t + \varphi) \tag{8.2}$$

where

$$\begin{aligned} A &= |G(i\omega)| \\ \varphi &= \arg G(i\omega) \end{aligned}$$

If we carry out a frequency analysis, i.e. let the input signal be a sinusoid with varying frequency, and measure the amplitude and phase shift of the output signal, we can thus determine the value of the transfer function for these frequencies. We obtain a table containing frequencies and their corresponding amplitudes and phase shifts. A table is, however, an inconvenient representation of the process dynamics. Therefore the table is usually represented graphically. This is done mainly in either of two ways; the Nyquist curve and the Bode plot.

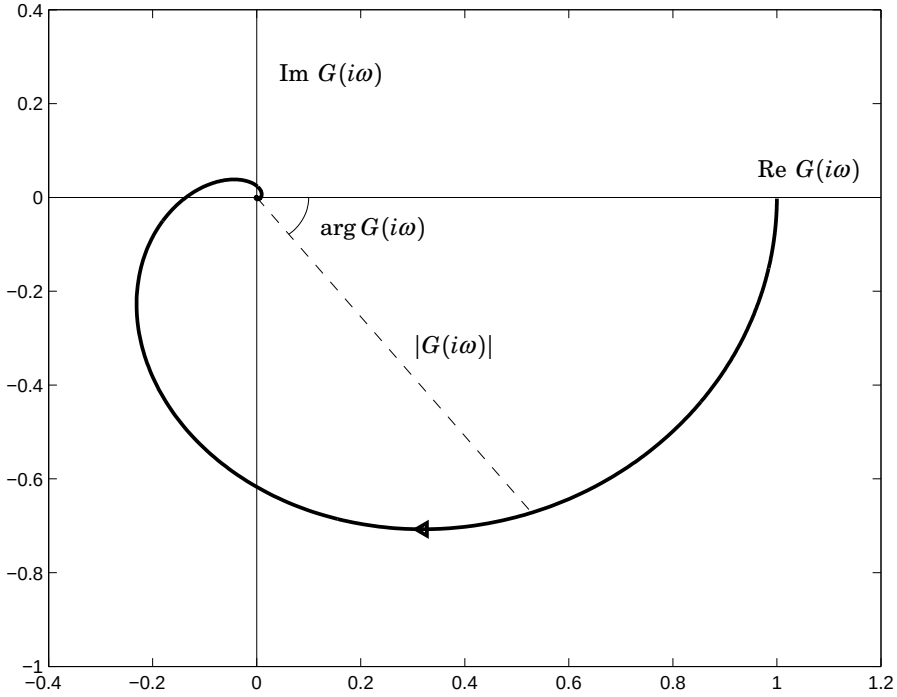


Figure 8.3 Nyquist curve.

8.2 The Nyquist Curve

The Nyquist curve is constituted of the complex number $G(i\omega)$ drawn in the complex plane for ω in $[0, \infty]$. Figure 8.3 shows a typical Nyquist curve.

Most processes have low-pass characteristics. This means that the output signal of the process is affected by low frequency inputs, whereas high frequency signals are damped out. Since the distance between the origin and points on the Nyquist curve describes the gain of process, it is normal that the Nyquist curve approaches the origin for high frequencies, as shown in Figure 8.3. The phase shift between in- and output does usually increase with the frequency. This is the explanation to the spiral shape of the curve in Figure 8.3.

The following example shows how one can compute the shape of the Nyquist curve, given the transfer function.

EXAMPLE 8.2—NYQUIST CURVE DRAWING
Assume that the process is described by the transfer function

$$G(s) = \frac{1}{s+1}$$

We compute $G(i\omega)$ and separate into its real- and imaginary parts

$$G(i\omega) = \frac{1}{1+i\omega} = \frac{1-i\omega}{1+\omega^2} = \frac{1}{1+\omega^2} - i\frac{\omega}{1+\omega^2}$$

We see that the real part is positive, whereas the imaginary part is negative for all ω. In other words, the Nyquist curve will be contained in the fourth quadrant. Further, we see that $G(i\omega) \approx 1$ for small ω and $G(i\omega) \to 0$ as $\omega \to \infty$. The Nyquist curve is shown in Figure 8.4 □

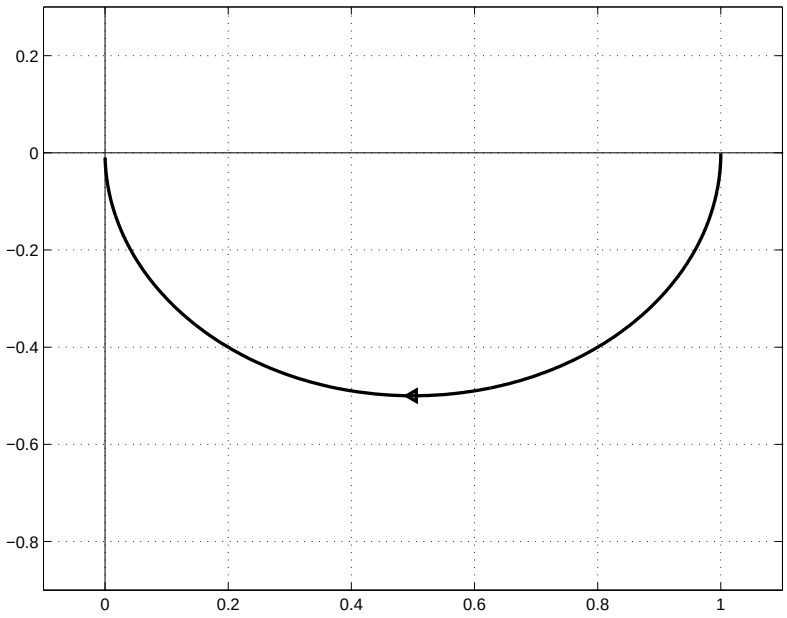


Figure 8.4 The Nyquist curve in Example 8.2.

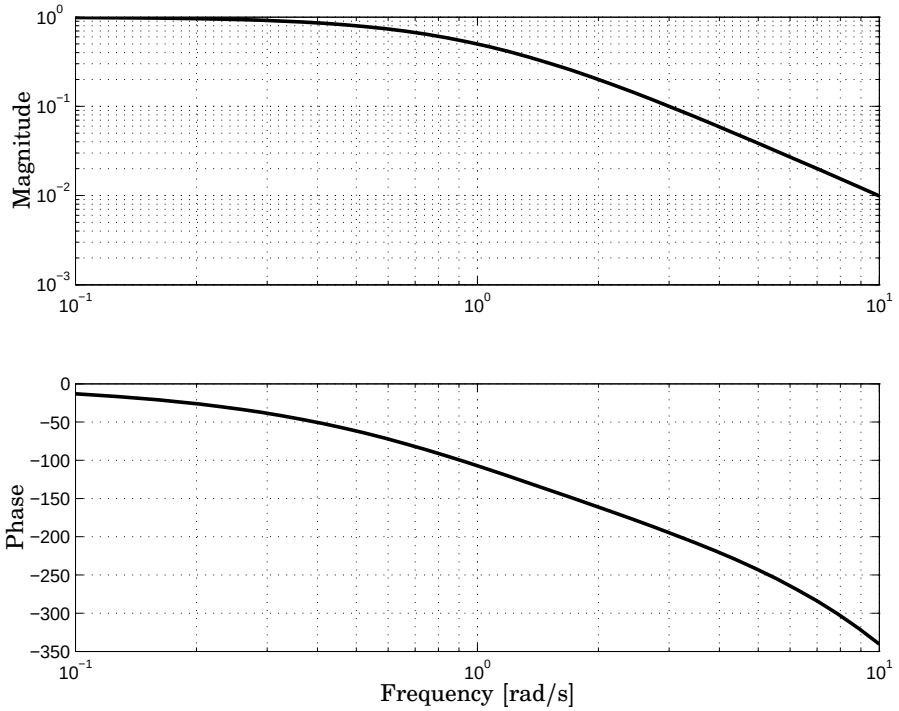


Figure 8.5 Bode plot.

8.3 The Bode Plot

The Bode plot features two curves, $|G(i\omega)|$ and $\arg G(i\omega)$ as functions of ω. Figure 8.5 shows the Bode plot of a typical process. The magnitude plot is drawn in a logarithmic scale, whereas the argument is drawn in a linear scale. The frequency axis is logarithmic.

The Bode plot of a process often looks similar to the one shown in Figure 8.5. The low

frequency gain is often constant and corresponds to the static gain of the process. As the frequency increases, the gain and phase shift also increase. In other words, the process has low pass characteristics.

The Bode plot has some properties which makes it easier to draw than the Nyquist plot. Assume that we can factor the transfer function, e.g., as

$$G(s) = G_1(s)G_2(s)G_3(s)$$

The logarithms of the magnitude and argument, respectively, are given by

$$\begin{aligned}\log|G(i\omega)| &= \log|G_1(i\omega)| + \log|G_2(i\omega)| + \log|G_3(i\omega)| \\ \arg G(i\omega) &= \arg G_1(i\omega) + \arg G_2(i\omega) + \arg G_3(i\omega)\end{aligned}$$

This means that the Bode plot of a transfer function is given by the sum of the Bode plots of its factors. This, in terms, enables us to draw all Bode plots which correspond to products of less complex transfer functions, for which we have already drawn the Bode plots. We shall study five sample transfer functions, into which all transfer functions in this course can be factored. These sample transfer functions are

1. K
2. s^n
3. $(1+sT)^n$
4. $\left(1 + 2\zeta s/\omega_0 + (s/\omega_0)^2\right)^n$
5. e^{-sL}

1. Bode Plot of $G(s) = K$

The magnitude and argument of the transfer function $G(s) = K$ are given by

$$\begin{aligned}\log|G(i\omega)| &= \log K \\ \arg G(i\omega) &= 0\end{aligned}$$

Both the gain and argument are independent of ω. The Bode plot is thus made up by two horizontal lines. This is shown in Figure 8.6 where Bode plots corresponding to three values of K are shown.

2. Bode Plot of $G(s) = s^n$

The magnitude and argument of the transfer function $G(s) = s^n$ are given by

$$\begin{aligned}\log|G(i\omega)| &= \log|i\omega|^n = n\log\omega \\ \arg G(i\omega) &= n\arg(i\omega) = n\frac{\pi}{2}\end{aligned}$$

The magnitude plot is a straight line with slope n, due to the logarithmic scales. The argument is independent of ω and thus forms a horizontal line. Figure 8.7 shows three Bode plots corresponding to different values of n.

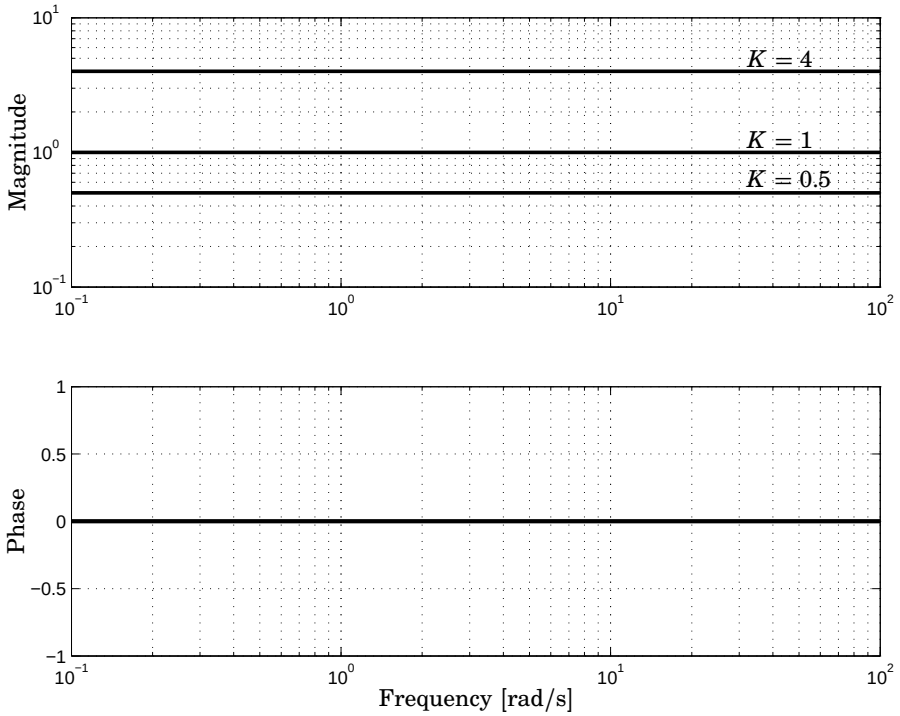


Figure 8.6 Bode plot of $G(s) = K$, where $K = 0.5, 1$ and 4.

3. Bode Plot of $G(s) = (1 + sT)^n$

The magnitude and argument of the transfer function $G(s) = (1 + sT)^n$ are given by

$$\log|G(i\omega)| = n \log\sqrt{1 + \omega^2 T^2}$$
$$\arg G(i\omega) = n \arg(1 + i\omega T) = n \arctan(\omega T)$$

For small values of ω the functions are given by

$$\log|G(i\omega)| \to 0$$
$$\arg G(i\omega) \to 0$$

For large values of ω the functions are given by

$$\log|G(i\omega)| \to n \log \omega T$$
$$\arg G(i\omega) \to n\frac{\pi}{2}$$

These two asymptotes, the low-frequency and high-frequency asymptotes, are shown in Figure 8.8 together with the Bode plots corresponding to some different values of n. The intersection between the low- and high frequency asymptotes is given by

$$\log \omega T = 0$$

This frequency is called the corner frequency and is given by $\omega = 1/T$.

4. Bode Plot of $G(s) = \left(1 + 2\zeta s/\omega_0 + (s/\omega_0)^2\right)^n$

The low-frequency asymptote of this transfer function is given by $G(i\omega) \approx 1$, i.e.

$$\log|G(i\omega)| \to 0$$
$$\arg G(i\omega) \to 0$$

For large ω the high-frequency asymptote is given by

$$G(i\omega) \approx (i\omega/\omega_0)^{2n} = (-1)^n(\omega/\omega_0)^{2n}$$

so that

$$\log|G(i\omega)| \to 2n \log \frac{\omega}{\omega_0}$$
$$\arg G(i\omega) \to n\pi$$

Figure 8.9 shows the Bode plots for some different values of the parameter ζ. The figure shows that this transfer function results in a resonance peak at the frequency ω_0. The peak increases in magnitude when ζ is decreased.

5. Bode Plot of $G(s) = e^{-sL}$

This transfer function describes a pure time delay. This means that the output is identical to the input, except that it has been delayed by a time L, $y(t) = u(t - L)$. If one sends a sinusoid through such a process, it outputs a sinusoid with the same amplitude, but with a phase shift which is larger for higher frequencies. For the transfer function $G(s) = e^{-sL}$ the magnitude and argument become

$$\log|G(i\omega)| = \log|e^{-i\omega L}| = 0$$
$$\arg G(i\omega) = \arg e^{-i\omega L} = -\omega L$$

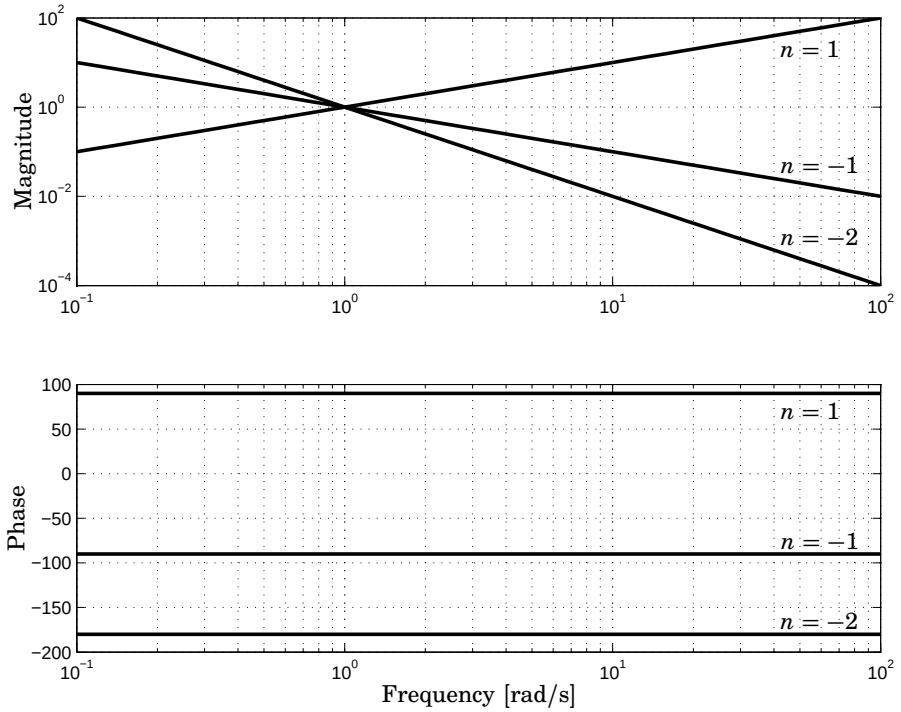


Figure 8.7 Bode plots of $G(s) = s^n$, where $n = 1, -1$ and -2.

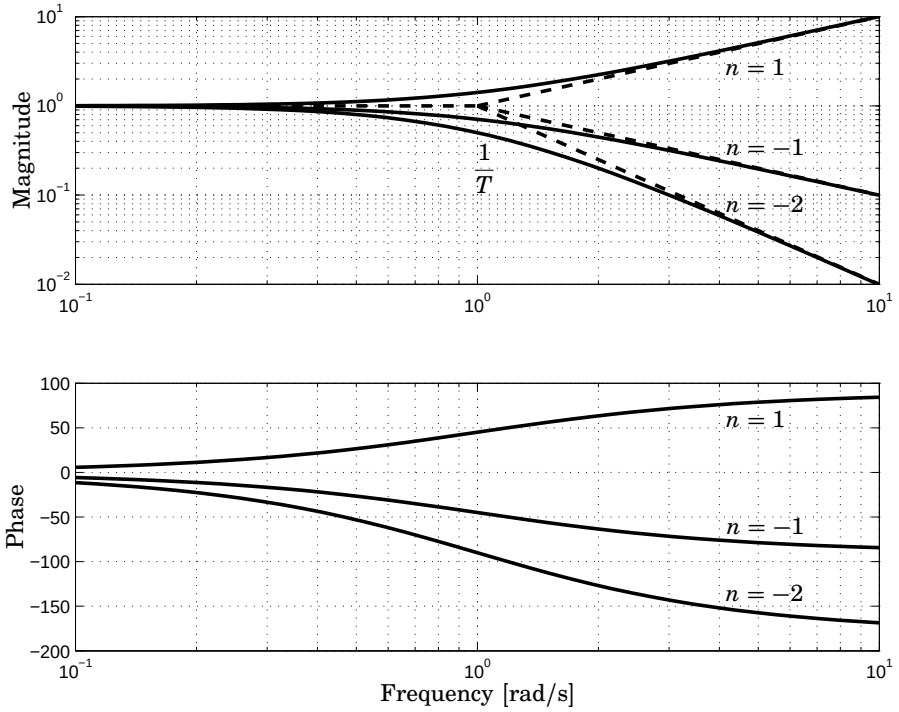


Figure 8.8 Bode plot of $G(s) = (1 + sT)^n$, where $T = 1$ and $n = 1, -1$ och -2.

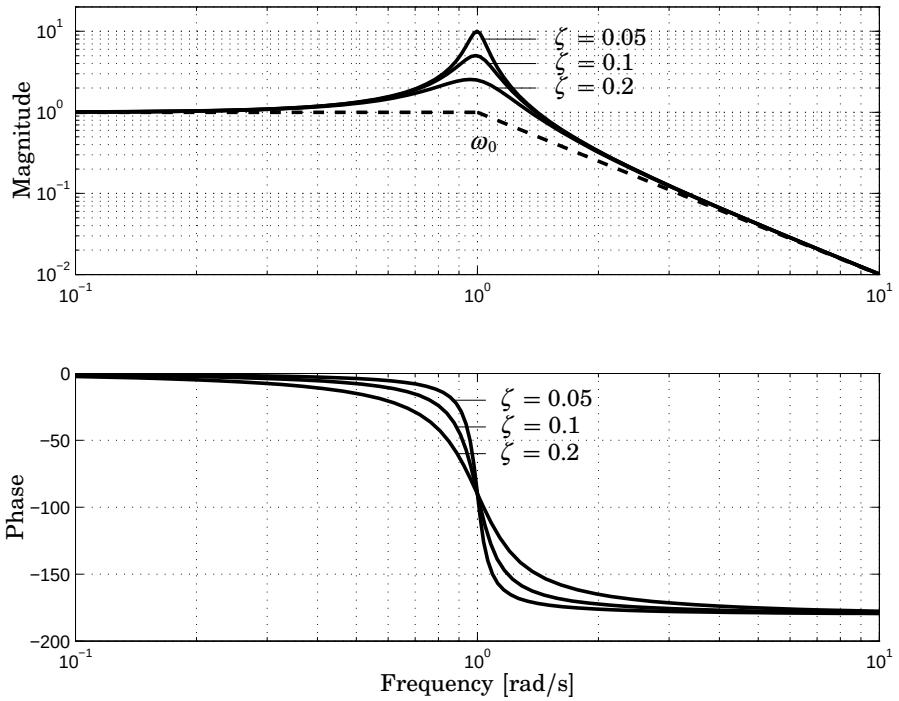


Figure 8.9 Bode plot of $G(s) = \omega_0^2/(s^2 + 2\zeta\omega_0 s + \omega_0^2)$, where $\omega_0 = 1$ and $\zeta = 0.05, 0.1, 0.2$.

Figure 8.10 shows the Bode plot for some different choices of the delay L.

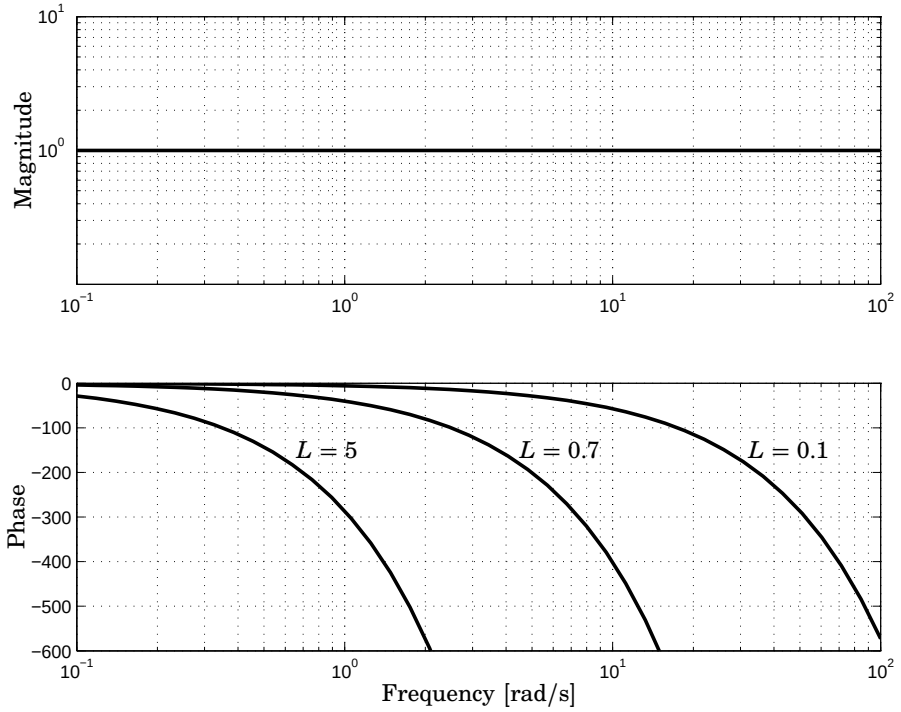


Figure 8.10 Bode plot of $G(s) = e^{-Ls}$ for $L = 5, 0.7$ and 0.1

8.4 Nyquist's Stability Criterion

Figure 8.11 shows the block diagram of a simple feedback loop where $G_0 = G_c G_p$ is the open-loop transfer function, i.e. the product of the process transfer function G_p and the controller transfer function G_c. There is a switch in the figure, which enables us to cut the feedback. When the switch is in position 1, the control loop functions normally. However, when the switch is in position 2, the feedback is broken and a sinusoid is applied to the open-loop transfer function.

Assume that the switch is in position 2. Under the assumption that the loop transfer function is stable, the signal e will also be a sinusoidal. Using (8.2), we can compute the control error as

$$\begin{aligned} e(t) &= -|G_0(i\omega)| \sin(\omega t + \arg G_0(i\omega)) \\ &= |G_0(i\omega)| \sin(\omega t + \arg G_0(i\omega) + \pi) \end{aligned}$$

Let us choose the frequency of the sinusoid such that $\arg G_0(i\omega) = -\pi$ and denote

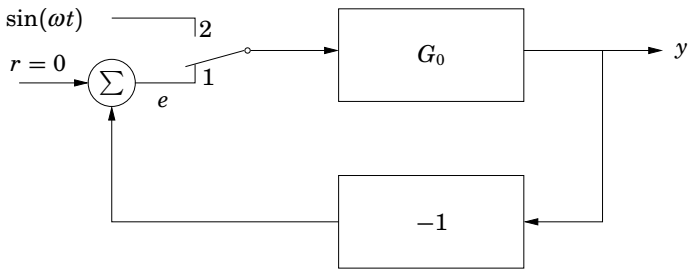


Figure 8.11 The simple feedback loop, being analyzed by means of the Nyquist criterion.

this frequency ω_0. We then obtain

$$e(t) = |G_0(i\omega_0)| \sin(\omega_0 t)$$

Let us further assume that $|G_0(i\omega_0)| = 1$. Then we will obtain the signal

$$e(t) = \sin(\omega_0 t)$$

i.e. the same signal which is sent into the system. If the switch is now toggled from position 2 to position 1, the signals in the control circuit will not be affected. The control loop is caught in a self-induced oscillation. In other words we lie on the stability boundary.

Correspondingly, one can imagine what will happen if $|G_0(i\omega_0)| \neq 1$. Assume that $|G_0(i\omega_0)| > 1$. Then the signal $e(t)$ will have the same frequency and phase as the input, but the amplitude will be larger. If one toggles the switch from position 2 to position 1 under these circumstances, the amplitude in the control loop will grow and an unstable loop is obtained. Analogously, a gain $|G_0(i\omega_0)| < 1$ will imply a decreasing amplitude and a stable control loop.

We can summarize the stability investigation in the following way: Investigate the magnitude of the loop transfer function at the frequency ω_0 for which $\arg G_0(i\omega) = -\pi$. Depending on the magnitude we obtain one of the following cases.

$\|G_0(i\omega_0)\| < 1$	Stable.
$\|G_0(i\omega_0)\| = 1$	Stability boundary.
$\|G_0(i\omega_0)\| > 1$	Unstable.

This intuitive reasoning is unfortunately not always true. It assumes, e.g., that signal components with other freuqencies than ω_0 are damped out, which is not always the truth. It was Nyquist that showed the shortcomings in this reasoning, and therafter he formulated his criterion:

THEOREM 8.1—NYQUIST'S STABILITY CRITERION
Assume that all poles of the open-loop transfer function lie in the left half plane or on the imaginary axis. Given this, the system is stable if the point -1 lies to the left of the Nyquist curve as it is traversed from $\omega = 0$ to $\omega = \infty$. □

Figure 8.12 shows Nyquist curves for some different loop transfer functions and the Nyquist criterion interpretation of these.

One advantage of Nyquist's stabiliy criterion, as opposed to studying the characteristic equation of the closed-loop system, is that it is applicable also when the system contains a delay. The Nyquist criterion, as formulated here, can however not be used when the open-loop system contains poles in the right half-plane.

8.5 Stability Margins

We have now discussed stability concepts and different methods to determine stability. In practice it is not sufficient to determine whether a process is stable. In addition one wants to know the margins towards the stability limit. We will now introduce three common margins, namely the gain margin, the phase margin and the delay margin. The margins measure the *robustness* of the closed-loop system, i.e., how far the system is from instability.

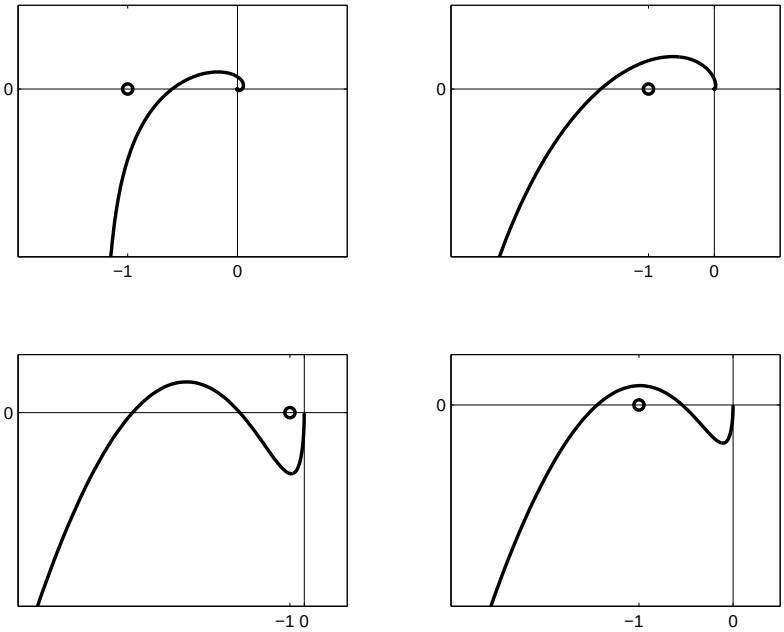


Figure 8.12 Nyquist curves for four different loop transfer functions. According to the Nyquist criterion the two leftmost systems are stable, whereas the two systems to the right are unstable.

Gain and Phase Margin

The gain and phase margins are easily defined using the Nyquist plot, see Figure 8.13.

For simplicity we assume that the Nyquist curve of the open loop transfer function G_0 is strictly decreasing, both in magnitude and argument. The gain margin is denoted A_m and determines by how much the gain can be be increased without reaching

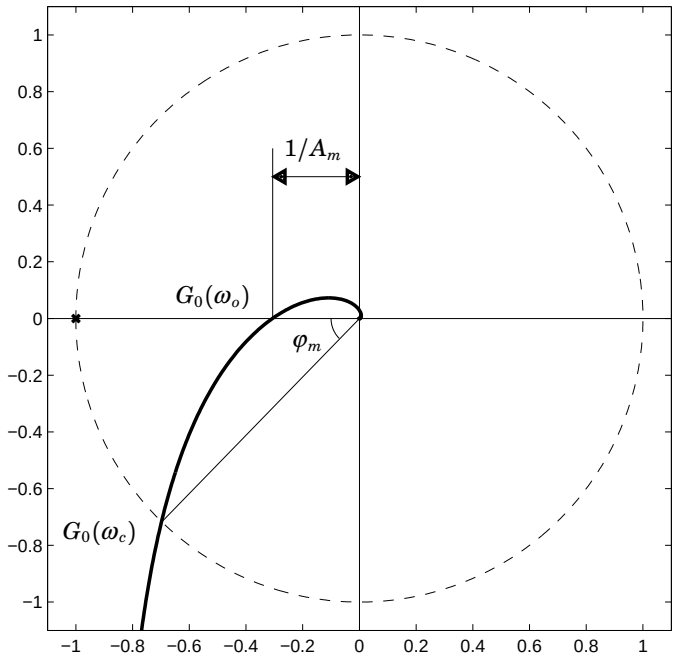


Figure 8.13 Phase margin φ_m and gain margin A_m in the Nyquist plot.

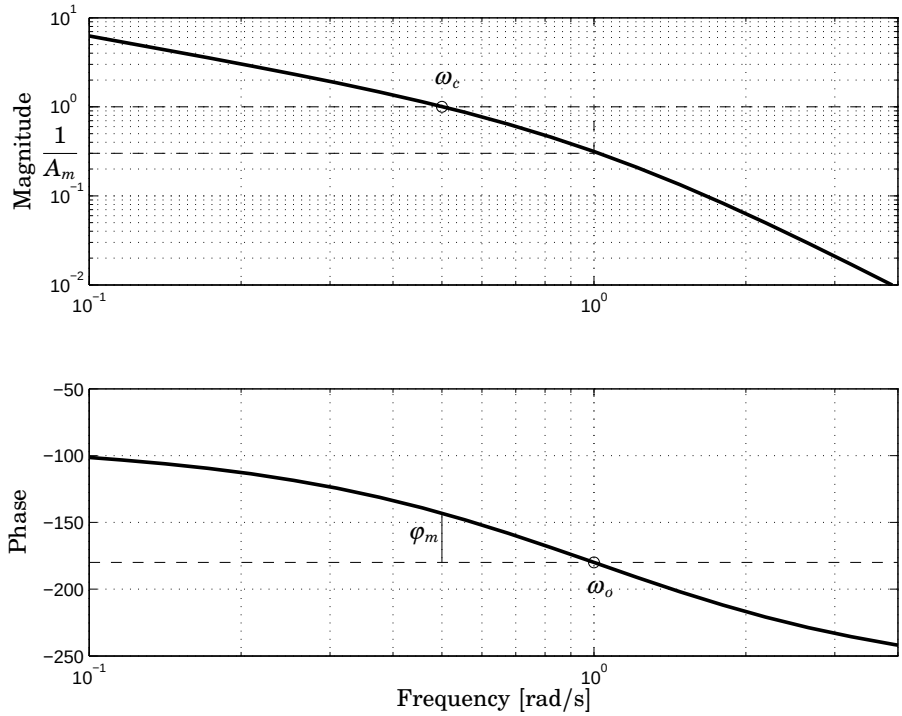


Figure 8.14 Phase margin φ_m and gain margin A_m in the Bode plot.

instability. This margin is read at the frequency ω_0, where the phase shift is π, i.e. $\arg G_0(i\omega_0) = -\pi$. The gain margin is thus given by

$$A_m = 1/\left|G_0(i\omega_0)\right|$$

The phase margin is denoted φ_m and determines by how much the phase shift can be decreased without passing the stability limit. The phase margin can be determined by observing the phase shift in the Nyquist curve at the frequency ω_c, where the magnitude is unity, i.e. $|G_0(i\omega_c)| = 1$. The frequency ω_c is known as the cross-over frequency. The phase margin is given by

$$\varphi_m = \pi + \arg G_0(i\omega_c)$$

The cross-over frequency can be viewed as a measure of the speed of the closed-loop system.

The gain and phase margins can also be read from the Bode plot, see Figure 8.14. The critical point -1 in the Nyquist plot corresponds to two horizontal lines in the Bode plot. One line corresponds to the magnitude $|G_0(i\omega)| = 1$ while the other is corresponding to the argument $\arg G_0(i\omega) = -\pi$. The gain margin is obtained as the distance between the line $|G_0(i\omega)| = 1$ and the magnitude curve at the frequency ω_0. The phase margin is obtained as the distance between the line $\arg G_0(i\omega) = -\pi$ and the phase curve.

It is important to maintain reasonable stability margins, since it allows for some variations in process dynamics. This is, however, not the only reason. The stability margins and the distance to the critical point -1 are also decisive for the control performance. If the stability margins are inadequate, jerky and poorly damped control is obtained. On the other hand, large stability margins result in slow control.

Customary rules of thumb prescribe gain and phase margins within the intervals $A_m \in [2, 6]$ and $\varphi_m \in [45°, 60°]$, respectively.

Delay Margin

The delay margin determines the length of an added delay required to drive the control loop unstable. The delay margin has no interpretations as a distance in the Nyquist plot, as we have seen for the gain and phase margins.

Assume that the open-loop transfer function $G_0(s)$ is augmented with a delay. The new loop transfer function thus becomes

$$G_0^{new}(s) = e^{-sL} G_0(s)$$

where L is the delay. The gain and phase shift of the new transfer function are given by

$$|G_0^{new}(i\omega)| = |G_0(i\omega)|$$
$$\arg G_0^{new}(i\omega) = \arg G_0(i\omega) - \omega L$$

The gain is thus not affected by the delay, while the phase shift decreases. Assume that the nominal loop transfer function G_0 has cross-over frequency ω_c, i.e. that $|G_0(i\omega_c)| = 1$, and that the corresponding phase margin is denoted φ_m. Since G_0^{new} has the same gain as G_0, the cross-over frequency of G_0^{new} will also be ω_c. The phase margin will, however, decrease since the phase shift has decreased. The new phase margin becomes

$$\varphi_m^{new} = \varphi_m - \omega_c L$$

If the delay is excessive, the phase margin vanishes and the closed-loop system becomes unstable. This occurs when

$$\omega_c L = \varphi_m$$

This gives us the following bound on how long delays can be before causing an unstable system

$$L_m = \frac{\varphi_m}{\omega_c}$$

The delay L_m is known as the delay margin and is a robustness margin in the same way as the gain margin A_m and the phase margin φ_m.

Obviously, we cannot allow delays close to L_m. The limit

$$\omega_c L < 0.2$$

is a good rule of thumb which guarantees a phase margin decrease of at most 12°. The equation also reveals how this criterion can be met. Either the delay L must be kept sufficiently short or one has to limit the cross-over frequency ω_c, i.e. limit the speed of the system.

9. The PID Controller

9.1 Introduction

Many control problems can be solved using a PID-controller. This controller is named after its function which can be described as

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int^{t} e(\tau)\,d\tau + T_d\frac{de(t)}{dt}\right] = P + I + D \tag{9.1}$$

where u is the controller output, and e is the error, i.e. the difference between the reference value r (the set point) and process output y (the measured variable). The control action is thus composed of three terms, one part (P) is proportional to the error, another (I) is proportional to the integral of the error, and a third (D) is proportional to the derivative of the error. The transfer function of the PID controller is given by

$$G_c(s) = K\left(1 + \frac{1}{sTi} + sT_d\right) \tag{9.2}$$

Special cases are obtained by only using some of the terms i.e. P, I, PI, or PD controllers. The PI controller is most common. It is also possible to have more complicated controllers e.g. an additional derivative term, which gives a PIDD or a DPID controller. The name PID controller is often used as a generic name for all these controllers.

The PID controller is very common. It is used to solve many control problems. The controller can be implemented in many different ways. For control of large industrial processes it was very common to have control rooms filled with several hundred PID controllers. The algorithm can also be programmed into a computer system that can control many loops. This is the standard approach today to control of large industrial processes. Many special purpose control systems also use PID control as the basic algorithm.

The PID controller was originally implemented using analog techniques. The technology has developed through many different stages, pneumatic, relay and motors, transistors and integrated circuits. In this development much know-how was accumulated that was embedded into the analog design. In this process several useful modifications to the “textbook” algorithm given by (9.1) were made. Many of these modifications were not published, but kept as proprietary techniques by the manufacturers.

Today virtually all PID controllers are implemented digitally. Early implementations of digital PID controllers were often a pure translation of the “textbook” algorithm which left out many of the good extra features of the analog design. The failures renewed the interest in PID control.

It is essential for any user of control systems to master PID control, to understand how it works and to have the ability to use, implement, and tune PID controllers. This chapter is intended to provide that knowledge.

9.2 The PID Algorithm

In this section the PID algorithm will be discussed in more detail. Properties of proportional, integral, and derivative action will be explained. Based on the insight gained we will also make several modifications of the small-signal behavior of the algorithm. This will lead to an improved algorithm which is significantly better than the basic algorithm given by (9.1).

Proportional Action

A proportional controller can be described by

$$u = K(r - y) + u_0 = Ke + u_0 \tag{9.3}$$

The control signal is thus proportional to the error. Notice that there is also a reset or a bias term u_0. The purpose of this term is to provide an adjustment so that the desired steady state value can be obtained. Without the reset (integrating) term it is necessary to have an error to generate a control signal that it different from zero. The reset term can be adjusted manually to give the correct control signal and zero error at a desired operating point.

Equation (9.1) holds for a limited range only because the output of a controller is always limited (see Figure 2.7). The range of input signals where the controller is linear is called the *proportional band*. Let p_B denote the proportional band and u_{min} and u_{max} the limits of the control variable. The following relation then holds

$$K = \frac{u_{\max} - u_{\min}}{p_B}$$

The proportional band is given in terms of the units of the measured value or in relative units. It is in practice often used instead of the controller gain. A proportional band of 50% thus implies that the controller has a gain of 2.

The properties of a proportional control applied to a multi-capacitive process are illustrated in Figure 9.1. This figure shows the step response of the closed-loop system with proportional control. The figure shows clearly that there is a steady state error. The error decreases when the controller gain is increased, but the system then becomes oscillatory. It is easy to calculate the stationary error. The Laplace transforms of the error $e = r - y$ is given by

$$E(s) = \frac{1}{1 + G_p(s)G_c(s)}R(s)$$

where R is the Laplace transforms of the reference value. With $G_p(0) = 1$ and $G_c(0) = K = 1$ we find that the error due to a setpoint signal is 50% as is seen in the figure.

Integral Action

Early controllers for process control had proportional action only. The reset adjustment u_0 in (9.3) was used to ensure that the desired steady state value was obtained. Since it was tedious to adjust the reset manually there was a strong incentive to automate the reset. One way to do this is illustrated in Figure 9.2. The idea is to low pass filter the controller output to find the bias and add this signal to the controller output. It is straight-forward to analyze the system in Figure 9.2. We get

$$U(s) = KE(s) + \frac{1}{1 + sT_i}U(s)$$

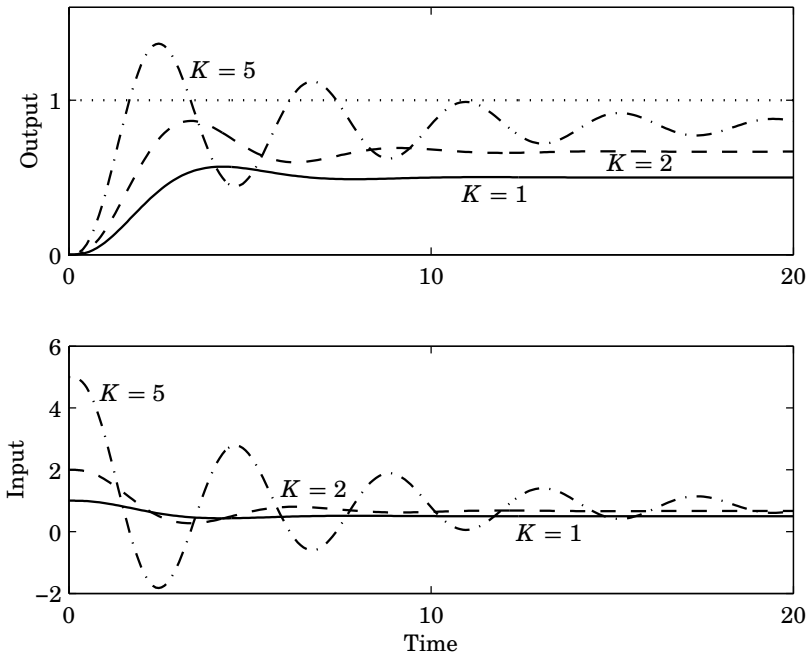


Figure 9.1 Illustration of proportional control. The process has the transfer function $G_p(s) = (s+1)^{-3}$.

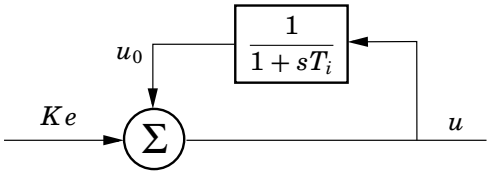


Figure 9.2 Controller with automatic reset.

Solving for U we get

$$U(s) = K\left(1 + \frac{1}{sT_i}\right)E(s) \tag{9.4}$$

Conversion to the time domain gives

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau\right) = P + I$$

which is the input-output relation for a PI controller. Parameter T_i, which has dimension time, is called *integral time* or *reset time*. The properties of a PI controller are illustrated in Figure 9.3. The figure illustrates that the idea of automatic reset or PI control works very well in the specific case.

It is straight-forward to show that a controller with integral action will always give a control error that is zero. To do this assume that there exist an equilibrium (steady state). The process input, the output, and the error must then be constant. Let e^0 denote the error and u^0 the process input. Taking the inverse Laplace transform of (9.4) gives the following relation

$$u^0 = K\left(e^0 + t\frac{e^0}{T_i}\right)$$

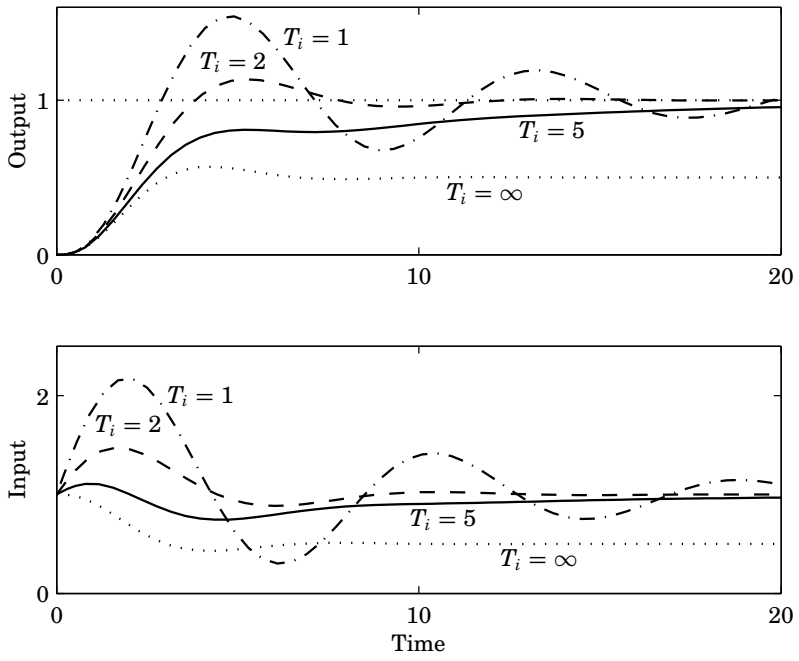


Figure 9.3 Illustration of PI control. The process has the transfer function $G_p(s) = (s+1)^{-3}$. The controller has the gain $K = 1$.

This implies that u^0 is constant only if $e^0 = 0$, i.e. the control error is zero. The argument will obviously hold for any controller with integral action. Notice, however, that a stationary solution may not necessarily exist. For instance the signals may oscillate.

Another intuitive argument that also gives insight into the benefits of integral control is to observe that with integral action a small control error that has the same sign over a long time period may generate a large control signal.

Sometimes a controller of the form

$$u(t) = K_i \int_0^t e(\tau)\, d\tau = I$$

is used. This is called an I controller or a *floating controller*. The name floating relates to the fact that with integral control there is not a direct correspondence between the error and the control signal.

Derivative Action

A controller with proportional action has a significant disadvantage because it does not anticipate what is happening in the future. This is illustrated in Figure 9.4 which shows two error curves, A and B. At time t a proportional controller will give the same control action for both error curves. A significant improvement can be obtained by introducing prediction.

A simple way to predict is to extrapolate the error curve along its tangent. This means that control action is based on the predicted error T_d time units ahead, i.e.

$$e_p(t + T_d) = e(t) + T_d \frac{de(t)}{dt}$$

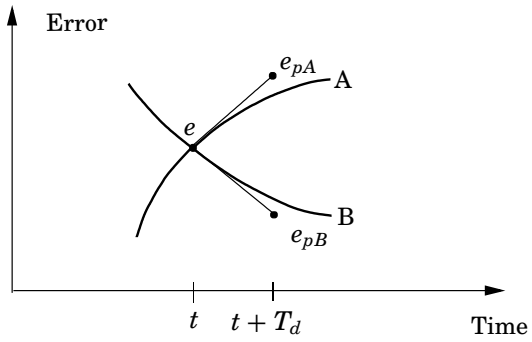


Figure 9.4 Illustrates the predictive nature of proportional and derivative control.

This gives the control law

$$u(t) = K\left(e(t) + T_d\frac{de(t)}{dt}\right) = P + D$$

which is a PD controller. With such a controller the control actions for the curves A and B in Figure 9.4 will be quite different. Parameter T_d, which has dimension time, is called *derivative time*. It can be interpreted as a prediction horizon.

The fact that control is based on the predicted output implies that it is possible to improve the damping of an oscillatory system. The properties of a controller with derivative action are illustrated in Figure 9.5. This figure shows that the oscillations are more damped when derivative action is used.

Notice in Figure 9.5 that the output approaches an exponential curve for large values of T_d. This can easily be understood from the following intuitive discussion. If the derivative time is longer than the other time constants of the system the feedback

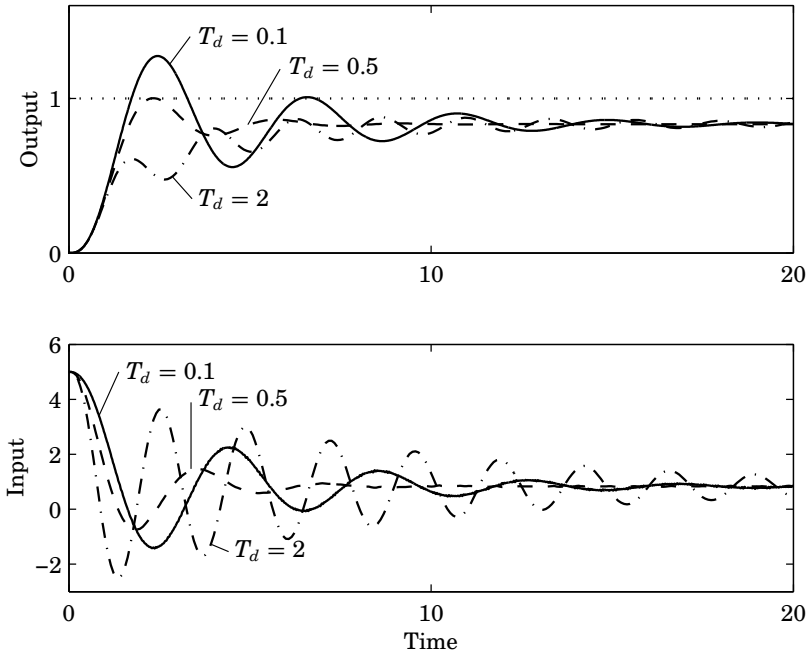


Figure 9.5 Illustration of the damping properties of derivative action. The process has the transfer function $G_p(s) = (s + 1)^{-3}$. The gain is $K = 5$ and T_d is varied.

loop can be interpreted as a feedback system that tries to make predicted error e_p small. This implies that

$$e_p = e + T_d \frac{de}{dt} = 0$$

This differential equation has the solution $e(t) = e(0)e^{-t/T_d}$. For large T_d the error thus goes to zero exponentially with time constant T_d.

A drawback with derivative action is that parameter T_d has to be chosen carefully. Industrial PID controllers often have potentiometers to set the parameters K, T_i, and T_d. Because of the difficulty in adjusting derivative time T_d the potentiometer for T_d is made so that derivative action can be switched off. In practical industrial installations we often find that derivative action is switched off.

9.3 The Series Form of the PID Controller

Thus far we have assumed that the PID controller is described by the equation

$$u = K\left(e + \frac{1}{T_i}\int e(t)dt + T_d\frac{de}{dt}\right)$$

with corresponding transfer function

$$G_c(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

This form is known as the *parallel form*, since the control error e is treated in parallel in the P, I and D parts. An equally common form in industrial applications is illustrated by the transfer function

$$G_c'(s) = K'\left(1 + \frac{1}{sT_i'}\right)(1 + sT_d')$$

This is known as the *series form*, since it can be described as a series connection of a PI and a PD controller. The difference between these two forms is not as large as it might appear. If we multiply the factors in the series form we arrive at

$$G_c'(s) = K'\left(1 + \frac{1}{sT_i'}\right)(1 + sT_d') = K'\left(1 + \frac{T_d'}{T_i'} + \frac{1}{sT_i'} + sT_d'\right)$$

The controller thus contains P, I and D parts. The only difference between the two forms is hence their parameterizations. The relations are given by the following equations:

$$K = K'\frac{T_i' + T_d'}{T_i'} \qquad K' = \frac{K}{2}\left(1 + \sqrt{1 - \frac{4T_d}{T_i}}\right)$$

$$T_i = T_i' + T_d' \qquad T_i' = \frac{T_i}{2}\left(1 + \sqrt{1 - \frac{4T_d}{T_i}}\right)$$

$$T_d = \frac{T_i'T_d'}{T_i' + T_d'} \qquad T_d' = \frac{T_i}{2}\left(1 - \sqrt{1 - \frac{4T_d}{T_i}}\right)$$

If one switches from one controller form to the other and simultaneously translates the controller parameters appropriately, the functionality of the controller remains unchanged.

Two interesting observations can be made concerning the structures. The first is that the two representations are identical when the controllers are used as either P, PI or PD controllers. It is only when all three terms are used that we have a difference in parameterization between the series and parallel forms.

The second observations is that the parallel form is more general than its series counterpart. We can always translate a controller from series to parallel form, but the contrary is true only for the special case

$$T_i \geq 4T_d$$

This can be seen by comparing the transfer functions. The PID controller has a pole in the origin and two zeros. In the parallel form the two zeros can be complex, while the series form only allows for real zeros.

When the PID controller was implemented with pneumatic technology back in the 30s and 40s, it was done in series form, for practical reasons. The explanation to why there still exist so many controllers in series form is that many manufacturers have kept the controller structure, though the technology used to implement the controllers have changed.

9.4 The Bode Plot of the PID Controller

We shall now study the Bode plot of the PID controller and choose to do so for its series form. This form is easier to draw due to its real zeros. Also, the interpretation of the controller parameters is more straightforward for the series form. The drawn conclusions will, however, be valid also for the parallel form.

The transfer function G_c' can be written

$$G_c'(s) = K'\left(1 + \frac{1}{sT_i'}\right)(1 + sT_d') = \frac{K'}{sT_i'}(1 + sT_i')(1 + sT_d')$$

The low-frequency asymptote of the Bode plot is $K'/(sT_i')$, while its high-frequency asymptote is given by $K'T_d's$. The Bode plot shows two corner frequencies. Under the assumption that $T_i' > T_d'$ the first corner frequency will appear at $\omega = 1/T_i'$ while the second lies at $\omega = 1/T_d'$. Both the corner frequencies correspond to zeros, causing the Bode plot to break upwards.

The Bode plot of the PID controller is shown in Figure 9.6. The Bode plot clearly shows the role of the three parameters. The gain K' determines the level of the magnitude curve whereas the integral and derivative times T_i' and T_d' determine its two corner frequencies.

The integral part increases the low-frequency gain of the controller, which helps reduce stationary errors. At the same, the phase is decreased, which may in turn decrease the phase margin and hence give worse stability margins.

The derivative part of the controller increases the high-frequency gain of the controller, which may amplify noise in the system. At the same time, the phase is increased, which typically increases the phase margin and gives better stability margins.

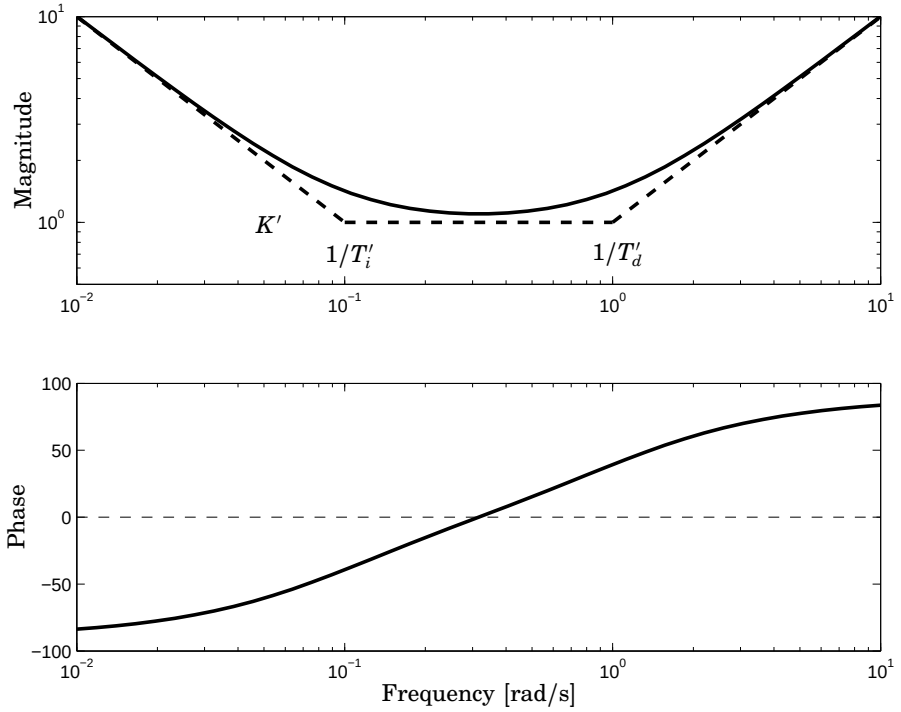


Figure 9.6 Bode plot of the PID controller with parameters $K' = 1$, $T_i' = 10$ and $T_d' = 1$.

9.5 Practical Modifications

The control law of an ideal PID is given by (9.1) or (9.2). We here discuss some modifications that are needed to make the controller work well in practice.

Limitation of Derivative Gain

A pure derivative can and should not be implemented, because it will give a very large amplification of measurement noise. The gain of the derivative must thus be limited. This can be done by approximating the derivative term $D(s) = sKT_d$ as follows

$$D(s) \approx \frac{sKT_d}{1 + sT_d/N}$$

The transfer function on the right approximates the ideal derivative term well at low frequencies but the gain is limited to KN at high frequencies. The parameter N is called the *maximum derivative gain*. Typical values of N are in the range 10–20.

Modification of Setpoint Response

In the basic algorithm given by (9.1) the control action is based on error feedback. This means that the control signal is obtained by filtering the control error. Since the error is the difference between the set point and the measured variable it means that the set point and the measured variable are treated in the same way. There are several advantages in providing separate signal treatments of those signals.

It was observed empirically that it is often advantageous to not let the derivative act on the reference value or to let it act on a fraction of the reference value only. The reason for this is that a step change in the reference signal will make drive the output of the control signal to its limits. This may result in large overshoots in the

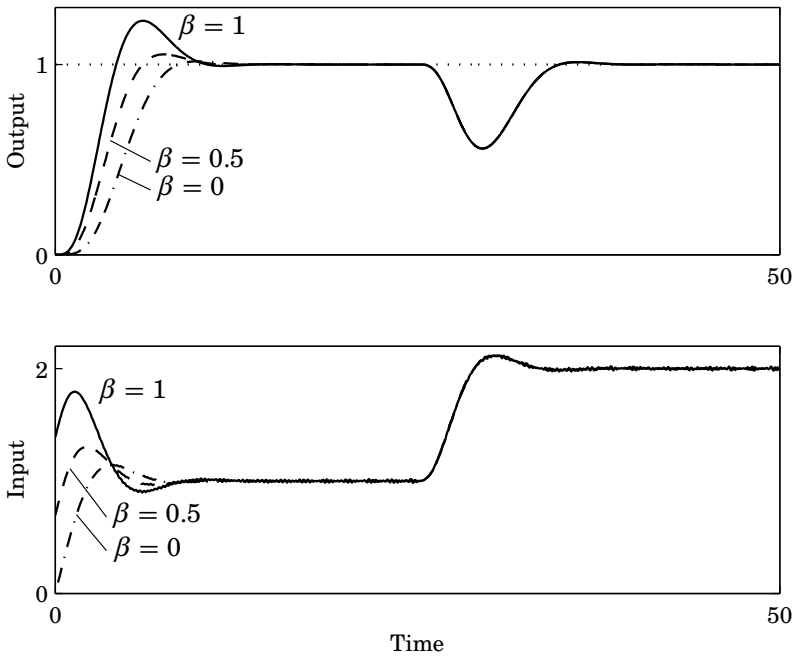


Figure 9.7 Response of system to setpoint changes and load disturbances for controller with different values of parameter β.

step response. To avoid this the derivative term can be modified to

$$D(s) = \frac{sKT_d}{1 + sT_d/N}(\gamma R(s) - Y(s))$$

If the parameter γ is zero, which is the most common case, the derivative action does not operate on the set point.

It has also been found suitable to let only a fraction β of the reference signal act on the proportional part. The PID algorithm obtained then becomes

$$U(s) = K\left[\beta R(s) - Y(s) + \frac{1}{sT_i}(R(s) - Y(s)) + \frac{sT_d}{1 + sT_d/N}(\gamma R(s) - Y(s))\right]$$

where U, R, and Y denote the Laplace transforms of u, r, and y. The idea to provide different signal paths for the process output and the reference signal is a good way to separate reference signal response from the response to load disturbances. Alternatively it may be viewed as a way to position the closed-loop zeros.

The advantages of a simple way to separately adjust responses to load disturbances and set points are illustrated in Figure 9.7. In this case parameters K, T_i, and T_d are chosen to give a good response to load disturbances. With $\beta = 1$ the response to set point changes has a large overshoot, which can be adjusted by changing parameter β.

There are also several other variations of the PID algorithm that are used in commercial systems. An extra first order lag may be used in series with the controller to obtain a high frequency roll-off. In some applications it has also been useful to include nonlinearities. The proportional term Ke can thus be replaced by $Ke|e|$ or by Ke^3. Analogous modifications of the derivative term have also been used.

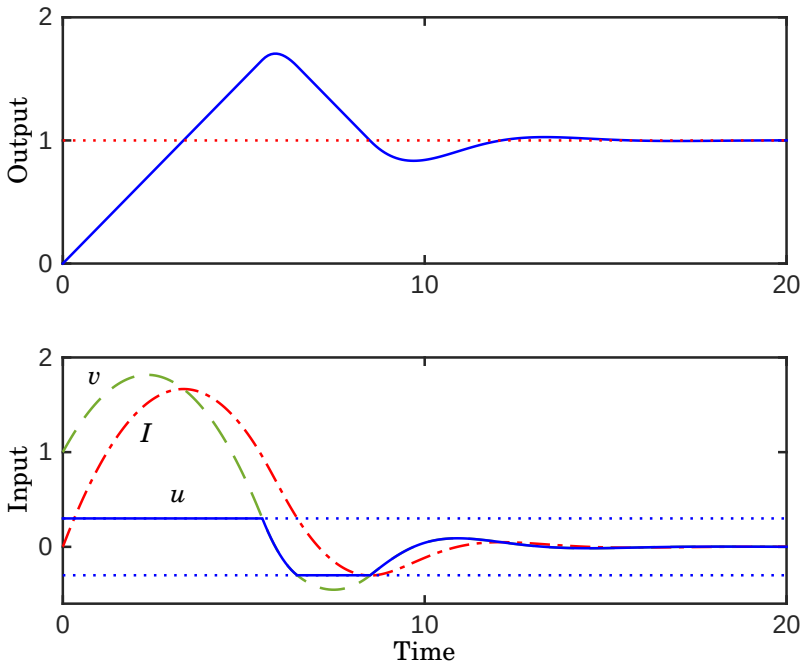


Figure 9.8 Illustration of integrator windup.

Integrator Windup

The combination of a saturating actuator and a controller with integral action gives rise to a phenomena called *integrator windup*. If the control error is so large that the integrator saturates the feedback path will be broken, because the actuator will remain saturated even if the process output changes. The integrator, being an unstable system, may then integrate up to a very large value. When the error changes sign the integral may be so large that it takes considerable time until the integral assumes a normal value again. The phenomena is also called *reset windup*. It is illustrated in Figure 9.8, which shows a simulation of a process with a PI controller. The process dynamics can be described as an integrator and the process input is limited to the range $-0.1 \leq u \leq 0.1$. The controller parameters are $K = 1$ and $T_i = 1$. When a reference signal in the form of a unit step is applied the computed control signal is so large that the process actuator saturates immediately at its high limit. Since the process dynamics is an integrator the process output increases linearly with rate 0.1 and the error also decreases linearly. The control signal will, however, remain saturated even when the error becomes zero because the control signal is given by

$$u(t) = Ke(t) + I$$

The integral has obtained a large value during the transient. The value is proportional to the dashed area in the figure. The control signal does not leave the saturation until the error has been negative for a sufficiently long time to reduce the value of the integral. The net effect is a large overshoot. When the control signal finally leaves the saturation it changes rapidly and saturates again at the lower actuator limit.

Anti-Windup. In a good PID controller it is necessary to avoid integrator windup. There are several ways to avoid integrator windup. One possibility is to stop updating the integral when the actuator saturates. This is called *conditional integration*. Another method is illustrated in the block diagram in Figure 9.9. In this method an

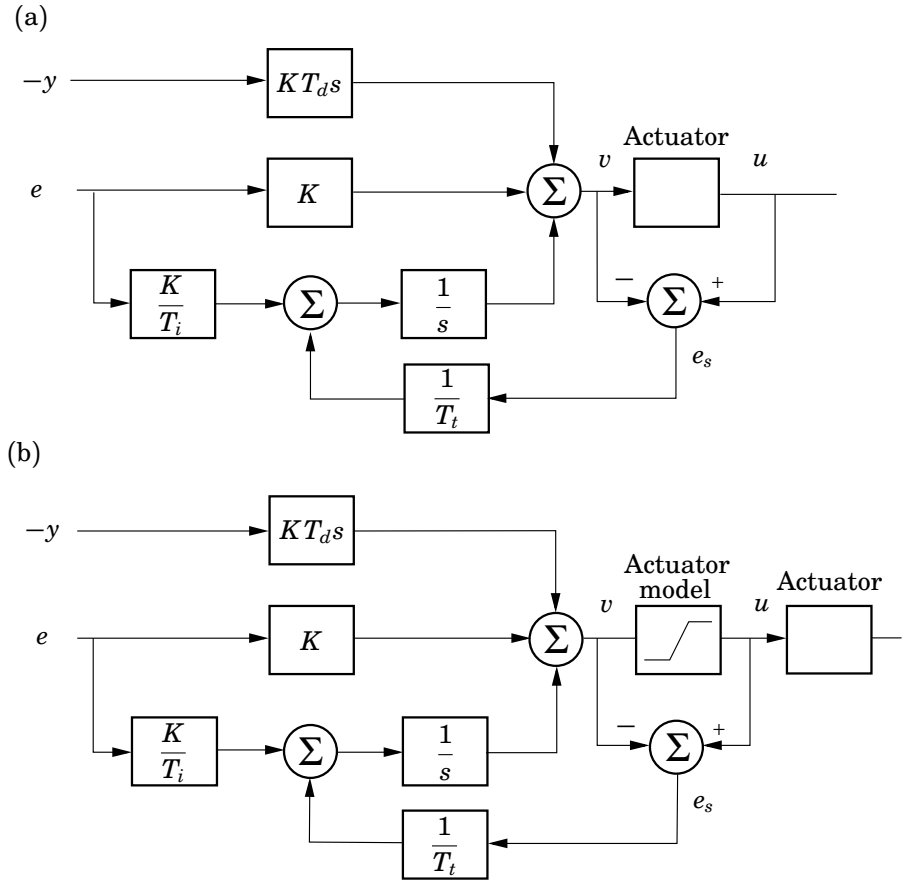


Figure 9.9 Controller with anti-windup. A system where the actuator output is measured is shown in (a) and a system where the actuator output is estimated from a mathematical model is shown in (b).

extra feedback path is provided by measuring the actuator output and forming an error signal (e_s) as the difference between the actuator output (u) and the controller output (v). This error is fed back to the integrator through the gain $1/T_t$. The error signal e_s is zero when the actuator does not saturate. When the actuator saturates the extra feedback path tries to make the error signal e_s equal zero. This means that the integrator is reset so that the controller output tracks the saturation limits. The method is therefore called *tracking*. The integrator is reset to the saturation limits at a rate corresponding to the time constant T_t which is called the tracking time constant. The advantage of this scheme for anti-windup is that it can be applied to any actuator as long as the actuator output is measured. If the actuator output is not measured the actuator can be modeled and an equivalent signal can be generated from a mathematical model as shown in Figure 9.9(b). It is thus useful for actuators having a dead-zone or an hysteresis.

Figure 9.10 shows the improved behavior obtained with a controller having anti-windup based on tracking. The system simulated is the same as in Figure 9.8. Notice the drastic improvement in performance.

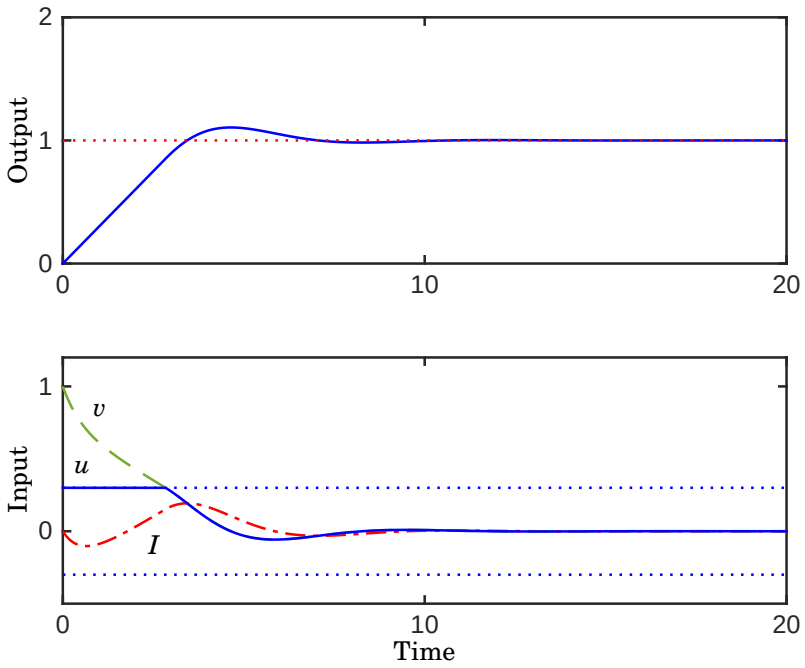


Figure 9.10 Illustration of controller with anti-windup using tracking. Compare with Figure 9.8.

9.6 Simple Tuning Rules

The PID controller is an elementary controller, often used in simple applications where one lacks the time and knowledge needed to further analyze the control problem. Consequently, there is a demand for easily applied rules of thumb, which can be used to tune the controller parameters to an acceptable performance level. The tuning should preferably be based on simple experiments conducted on the process.

Many tuning methods have been suggested since the PID controller was first introduced. The by far most well known are, however, Ziegler–Nichols' methods. They are not the best methods, but among the easiest to apply. It is worth noting that these methods are only to be considered as rules of thumb. They yield often acceptable controller parameters. If one has higher demands of controller performance, more elaborate methods may be needed.

The minimal amount of information needed to control a process is a process gain in order to determine K and a process time for the determination of T_i and T_d. Ziegler and Nichols presented two methods in the 40s, which can be used to obtain these parameters, a step response method and a frequency method (also known as the ultimate-sensitivity method).

The most popular method in industry today is the lambda method. It was derived in the sixties, and it provides acceptable control performance for a large class of processes.

Ziegler–Nichols' Step Response Method

Ziegler–Nichols' step response method is based on manual control of the process, i.e. that the control signal u is manually governed and no controller is present.

When the process has reached an equilibrium a step is issued in the control signal. We assume the step size to be 1. Steps of different magnitude will require normalization

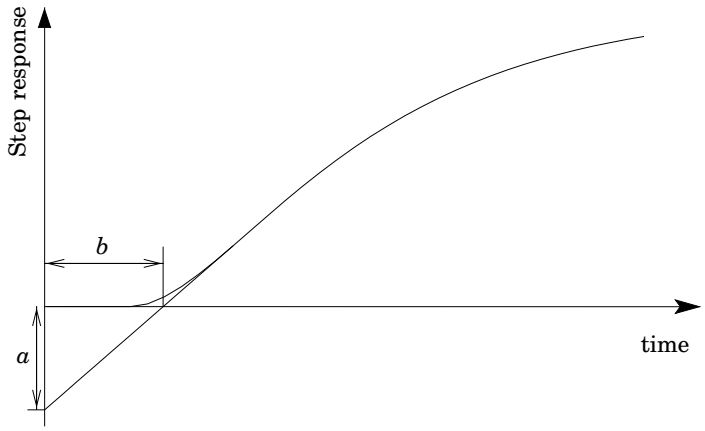


Figure 9.11 Evaluation of the gain a and time b from a process step response.

Controller	K	T_i	T_d
P	$1/a$		
PI	$0.9/a$	$3b$	
PID	$1.2/a$	$2b$	$0.5b$

Table 9.1 Recommended controller tuning according to Ziegler–Nichols' step response method.

of the response in the measurement signal.

Figure 9.11 shows a typical step response. A tangent is drawn through the point where the incline of the step response attains a maximum. The gain a and time b are then obtained from the intersection between this tangent and the coordinate axis. Ziegler and Nichols suggested that the obtained parameters should yield the PID parameters according to Table 9.1.

From the table we see that the controller gain K is chosen to be inversely proportional to the process gain a and that the controller times T_i and T_d are chosen proportional to the process time b.

Ziegler–Nichols' Ultimate-Sensitivity Method

Ziegler and Nichols ultimate-sensitivity method is based on incorporating a P controller in the loop and thereafter conducting the following steps:

1. Successively adjust K until the process oscillates with a constant amplitude. The corresponding gain is denoted K_0.
2. Measure the period T_0 of the oscillation.
3. Choose the controller parameters from Table 9.2.

In Ziegler–Nichols' ultimate-sensitivity method we identify the point $G_p(i\omega_{180})$, being the point where the process phase shift is $-180°$. If a P controller is installed in the loop and its gain is gradually increased, one will eventually reach the gain K_0 for which $|K_0 G_p(i\omega_{180})| = 1$. This corresponds to the limit of stability. The gain K_0 thus holds adequate information to calculate $G_p(i\omega_{180})$ while the frequency ω_{180} is given by the period T_0, $\omega_{180} = 2\pi/T_0$.

Now that we have realized how the ultimate-sensitivity method works, we can observe that in the case of P control it suggests a controller with gain margin $A_m = 2$.

Controller	K	T_i	T_d
P	$0.5K_0$		
PI	$0.45K_0$	$T_0/1.2$	
PID	$0.6K_0$	$T_0/2$	$T_0/8$

Table 9.2 Recommended controller tuning according to Ziegler–Nichols' ultimate-sensitivity method.

The Lambda Method

The lambda method is based on a step response experiment, where static process gain K_p, time delay L, and time constant T are determined. The experiment is shown in Figure 9.12. The point where the process output has the largest derivative is first determined, and the tangent to the curve at this point is drawn. The intersection between this tangent and the line representing the level of the process output before the step change is then determined. The time from the step change to this intersection point gives an estimate of time delay L. Time constant T is then determined from the time it takes to reach 63 % of the final value. Static gain K_p is finally determined by dividing the change in process output with the magnitude of the control signal step:

$$K_p = \frac{\Delta y}{\Delta u}$$

The lambda method has one parameter that can be set by the user, namely the desired time constant of the closed-loop system. This time constant is called λ (lambda).

The original lambda method only considered PI control. The simple tuning rule is

$$\begin{aligned} K &= \frac{1}{K_p}\frac{T}{L+\lambda} \\ T_i &= T \end{aligned} \tag{9.5}$$

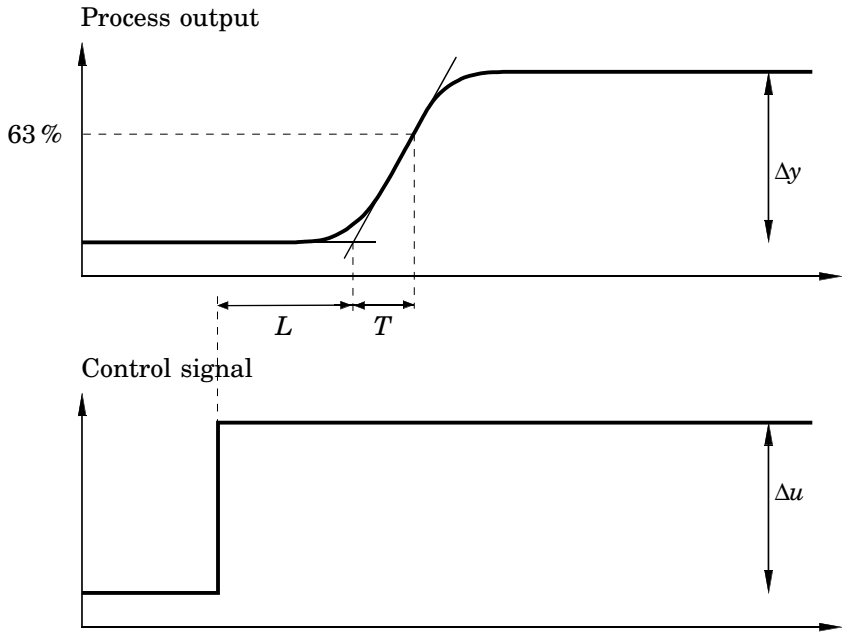


Figure 9.12 Determination of K_p, L, and T from a step response experiment.

The integral time is chosen equal to the process time constant T. The gain, however, is dependent on the choice of λ. A common choice is $\lambda = T$, which means that it is desired to give the closed-loop system the same time constant as the open-loop system.

The controller parameters are derived in the following way. The process and the controller transfer functions are

$$G_p(s) = \frac{K_p e^{-sL}}{1 + sT} \qquad G_c(s) = K\frac{1 + sT_i}{sT_i}$$

Since $T_i = T$, the loop transfer function is

$$G_0(s) = \frac{K_p K e^{-sL}}{sT}$$

The closed-loop transfer function between the setpoint and the process output becomes

$$G(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{K_p K e^{-sL}}{sT + K_p K e^{-sL}} \approx \frac{K_p K e^{-sL}}{sT + K_p K(1 - sL)} = \frac{e^{-sL}}{1 + s(T/(K_p K) - L)}$$

where the approximation is that the time delay in the denominator is replaced by the first terms in its Taylor series expansion. Therefore, the closed-loop transfer function is a first-order system with the same time delay as for the process, and with the time constant $\lambda = T/(K_p K) - L$. By specifying λ, K is given by Equation (9.5).

It is possible to derive tuning rules for PID controllers with a similar approach as for the PI controller. In this case, the time delay in the denominator of the closed-loop transfer function is not approximated by a Taylor series expansion, but with $e^{-sL} \approx (1-sL/2)/(1+sL/2)$. The resulting PID parameters are given in the Collection of Formulae.

10. Control Structures

10.1 Introduction

The simple feedback controller we have discussed so far is a powerful tool, which can provide efficient control and thus ensure stable operation of many simple processes. The simple feedback controller suffers, however, from a shortcoming which originates from the necessity of the measurement of the controlled variable. A deviation has to be registered between the controlled variable and the setpoint before any corrective action is taken by the controller.

Methods to deal with some aspects of this limitation are given in this chapter. The fact that a deviation must be registered before any corrective action can be taken implies that simple feedback is not very useful for processes with large delays relative to the dominating time constants. In many processes it may be possible to measure secondary process variables, which show the effect of disturbances before it is seen in the controlled variable. This leads to the idea of using a secondary loop to aid the primary control loop. This usage of multiple loops, called *cascade control*, is very widespread in the process industries to improve the performance of simple feedback. The ultimate case of this first basic limitation, occur when the desired controlled variable cannot be measured, then simple feedback cannot be applied, and one must resort to using secondary process measurements to estimate the desired controlled variable. Such methods require also process models.

In summary this chapter deals with improving the behavior of feedback control using linear functions, which are built around the simple feedback loop. The main application of these techniques stems from operating the processes under varying conditions. The implementation of these functions have been simplified tremendously by the digital computer. Therefore the techniques described in this chapter have reached widespread usage in the process industries.

10.2 Cascade Control

Cascade control is a strategy where two controllers are combined so that the output of the first controller forms the setpoint of the other. This is illustrated by the following example.

Example 10.1—Control of a Heat Exchanger
We want to control the temperature on the secondary side of a heat exchanger by controlling the steam valve on its primary side. This can be achieved by letting the temperature controller actuate the steam valve directly as shown in Figure 10.1. What actually affects the temperature is not the position of the valve, but rather the steam flow. If the valve is linear and the steam flow does not vary, there is a constant relation between the valve position and the steam flow. Usually, however, valves exhibit some form of nonlinearity and the steam pressure varies over time.

E.g. assume that the steam pressure on the primary side suddenly starts to decrease. As a consequence the steam flow will decrease, leading to slower heating of the

water on the secondary side. The temperature controller will issue a control signal corresponding to a more open valve position and after a while the steam flow will anew stabilize at a correct level. Consequently this strategy works, but the price is rather large disturbances in the temperature.

If one can measure the steam flow it is possible to incorporate a flow controller according to Figure 10.2. We form an inner control loop, which controls the steam flow. The setpoint of the flow controller G_{c_2} is given by the control signal of the temperature controller G_{c_1}. This is an example of a cascade control.

Cascading the controllers leaves the master controller G_{c_1} with a simpler task. Rather than letting G_{c_1} bear the entire control burden, part of the assignment is reassigned to the controller G_{c_2}. The controller G_{c_1} now only needs to produce a flow setpoint. Subsequently, it is up to the flow controller to maintain a flow close to this setpoint. A pressure variation will efficiently be eliminated by the flow controller, leading to a decreased disturbances in the temperature, as compared to the case with only one PID controller. □

The general principle of cascaded control is shown in Figure 10.3. The primary goal is to control the signal y_1 by means of the controller G_{c_1}. This could be achieved by using merely the controller G_{c_1} and letting its control signal enter the process directly. During cascade control one exploits the availability of an additional measurement signal, y_2. By locally establishing a feedback connection from y_2 by means of the controller G_{c_2} one can achieve more efficient control, than would be possible with only one controller. The controller G_{c_1} is often referred to as the primary or master controller, whereas G_{c_2} is known as the secondary or slave controller.

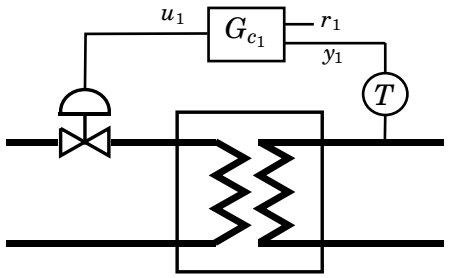


Figure 10.1 Temperature control of a heat exchanger.

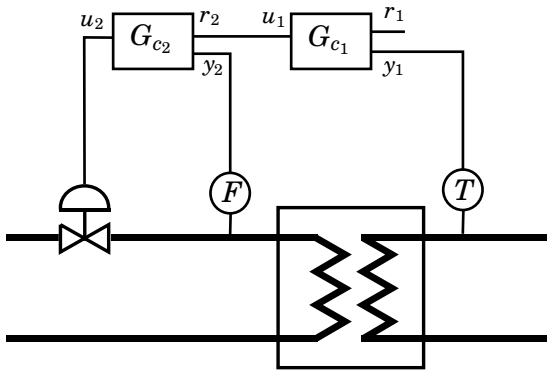


Figure 10.2 Cascade control of a heat exchanger.

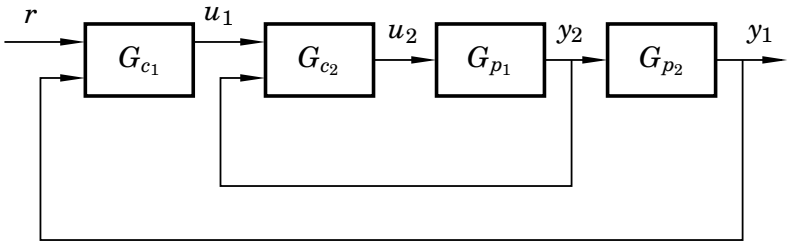


Figure 10.3 The principle of cascade control.

Cascade control is a typical example of how one can achieve more advanced control solutions, despite the simple structure of the PID controller, by combining several controllers. The foremost reason to use cascade control is to handle disturbances which enter the process at G_{p_2}, before they give rise to disturbances in the primary measurement signal y_1. An example of this are the pressure variations in the above example. A prerequisite is obviously that the inner control loop is significantly faster than the outer one. Another advantage with cascade control is that the dynamics which the primary controller is to control can be simplified. Without cascade control, the controller G_{c_1} works on a process consisting of the two section G_{p_1} and G_{p_2}. When cascading is implemented the process sections are changed to a combination of G_{p_1} and G_{p_2} in a feedback connection with G_{c_2}.

10.3 Mid-Range Control

The problem treated by mid-range control is illustrated in Figure 10.4. The figure illustrates an example where two valves are used to control a flow. One valve, v_1, is small but has a high resolution. The other valve, v_2, is large but has a low resolution.

Suppose that the small valve v_1 is in the middle of its operating range and that only small disturbances are acting on the system. In this case, one controller that manipulates valve v_1 is able to take care of the control problem. However, when larger disturbances occur, valve v_1 will saturate. In this case, the larger valve v_2 must also be manipulated.

The mid-range control strategy is illustrated in Figure 10.5. Controller G_{c_1} takes the set point r and flow signal y as inputs and manipulates the small valve v_1. A second controller, G_{c_2}, takes the control signal from G_{c_1} as input and tries to control it to a set point r_{u_1} in the middle of its operating range by manipulating the large valve v_2. If both controllers have integral action, the flow will be at the set point r and the valve v_1 will be at the set point r_{u_1} in steady state.

A block diagram of the mid-range control strategy is given in Figure 10.6. Process G_{p_1} and controller G_{c_1} together form a fast feedback loop. The mid-ranging controller

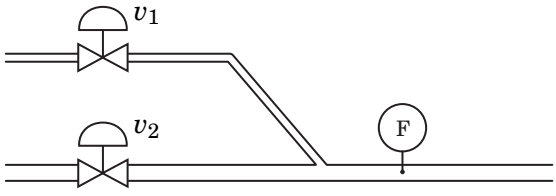


Figure 10.4 Two valves are used to control the flow.

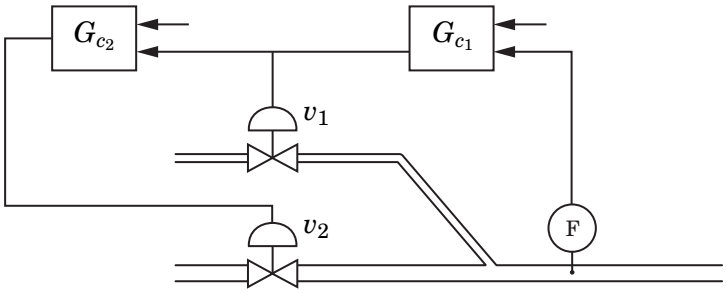


Figure 10.5 Mid-range control

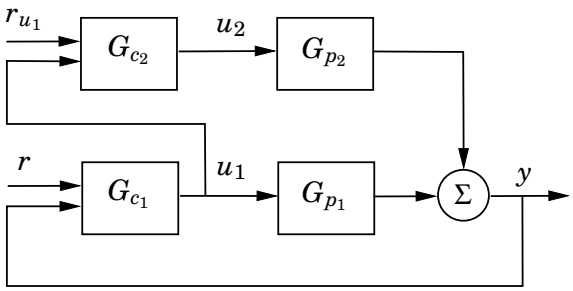


Figure 10.6 Block diagram of a system with mid-range control.

G_{c_2} controls the valve position of controller G_{c_1} via the process output y. This means that the output of controller G_{c_1} is controlled by driving the process output y away from the set point. If this is done slowly, the deviation from the set point can be kept small.

10.4 Ratio Control

In the process industries it is often desirable to maintain a fixed ratio between two variables. One example is two feed streams to a chemical reactor where it is desirable to maintain the molar ratio in order to ensure the proper stoichiometric mixture for the reaction. Another example is two flows in a separation process, such as the reflux ratio, i.e. ratio of reflux flow to distillate flow, in a distillation column. Ratio control is extensively used in the process industries. Below two implementation methods for *ratio control* will be discussed and one will be illustrated upon an air/fuel ratio control of a combustion process.

In Figure 10.7 the two ratio control schemes are shown. In scheme a) the ratio of the two measured variables is calculated and compared to the desired ratio a. The

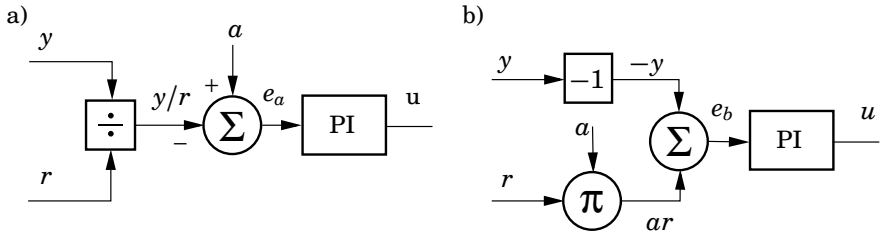


Figure 10.7 Block diagram for two implementations of ratio PI control. The desired ratio of y/r is a. The symbol π designates multiplication, $\div$ designates division.

formed deviation is

$$e_a = a - \frac{y}{r}$$

In scheme b) the denominator variable r is multiplied with the desired ratio a, thus forming the desired value for y. The deviation is formed as

$$e_b = ar - y$$

In both cases the controller output is generated from the error signal with a conventional PI (or PID) controller function. It follows from the two error equations that if the error is zero then

$$a = \frac{y}{r}$$

Thus both schemes fulfill their purpose of keeping the ratio constant at steady state.

The main difference between the two schemes is seen by finding the gain of the two error signals as r varies. These gains may be determined by finding the partial derivative of the error with respect to the varying variable r

$$k_a = \left.\frac{\partial e_a}{\partial r}\right|_y = \frac{y}{r^2}$$

$$k_b = \left.\frac{\partial e_b}{\partial r}\right|_y = a$$

Thus the gain is variable in the a) scheme but constant in the b) scheme. Both schemes are commonly used in practice, but the b) scheme is applied most often due to the constant gain.

A ratio controller can be built from a conventional PI or PID controller combined with a summation and multiplier. These are often supplied as one unit, since this control function is so common, where the controller can be switched from a ratio to a normal controller.

EXAMPLE 10.2—AIR/FUEL RATIO TO A BOILER OR A FURNACE

Controlling a boiler or a furnace it is desirable to maintain a constant ratio between air and fuel. Air is supplied in excess to ensure complete combustion of the fuel. The greater the air excess the larger is the energy loss in the stack gases. Therefore maintaining an optimal air excess is important for economical and environmental reasons. A configuration to achieve this goal using ratio control is shown in Figure 10.8. The fuel flow is controlled by a PI controller. The air flow is controlled by a ratio controller where fuel flow is the ratio variable. This configuration includes also a bias b, which for safety reasons ensures an air flow even when no fuel flow is available. Evaluation of the control error gives

$$e = aF_f + b - F_{air}$$

Thus the static flow characteristic is: $F_{air} = aF_f + b$. □

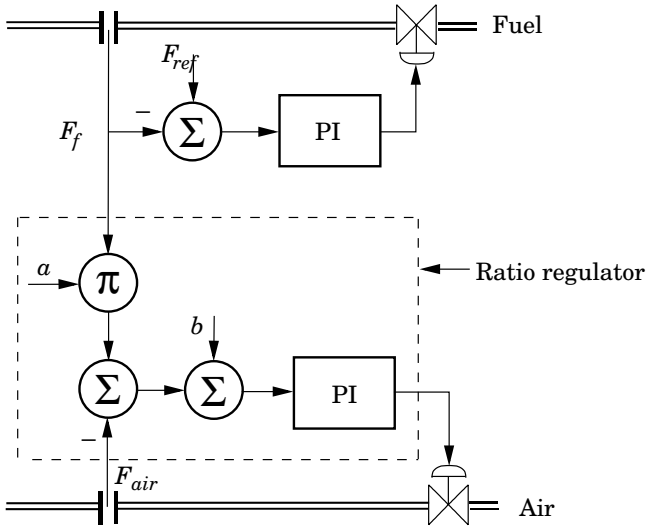


Figure 10.8 Configuration for air/fuel ratio control to a boiler or a furnace.

10.5 Feedforward

One obvious limitation of feedback is that corrective actions of course can not be taken before the influence is seen at the output of the system. In many process control applications it is possible to measure the disturbances coming into the systems. Typical examples are concentration or temperature changes in the feed to a chemical reactor or distillation column. Measurements of these disturbances can be used to make control actions before anything is noticed in the output of the process. It is then possible to make the system respond more quickly than if only feedback is used.

Consider the process in Figure 10.9. Assume that the disturbance can be measured. The objective is to determine the transfer function G_{ff} such that it is possible to reduce the influence of the measurable disturbance v. This is called *feedforward control*. The system is described by

$$Y = G_p U + G_v V = (G_p G_{ff} + G_v) V$$

The disturbance is *totally* eliminated from the output if

$$G_{ff}(s) = -\frac{G_v(s)}{G_p(s)}$$

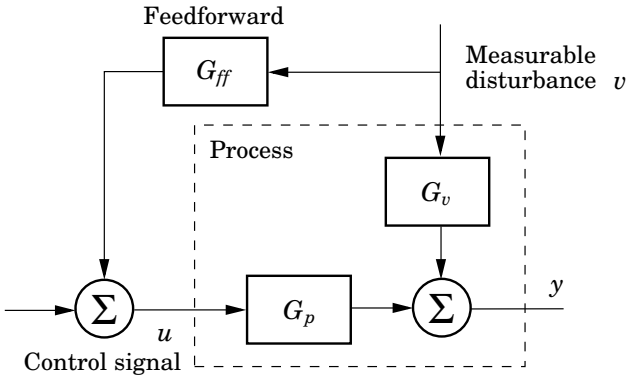


Figure 10.9 System with measurable disturbance.

Feedforward is in many cases very effective. The great advantage is that rapid disturbances can be eliminated by making corrections before their influences are seen in the process output. Feedforward can be used on linear as well as nonlinear systems. Feedforward can be regarded as a built-in process model, which will increase the performance of the system. The main drawback with feedforward is that it is necessary to have good process models. The feedforward controller is an open loop compensation and there is no feedback to compensate for errors in the process model. It is therefore common to combine feedforward and feedback. This usually gives a very effective way to solve a control problem. Rapid disturbances are eliminated by the feedforward part. The feedback takes care of unmeasurable disturbances and possible errors in the feedforward term.

To make an effective feedforward it is necessary that the control signal can be sufficiently large, i.e. that the controller has sufficiently large control authority. If the ratio of the time constants in G_p and G_v is too large then the control signals become large. To illustrate this assume that the transfer functions in Figure 10.9 are

$$G_p = \frac{k_p}{1 + T_p s} \qquad G_v = \frac{k_v}{1 + T_v s}$$

The feedforward controller given by (10.5) becomes

$$G_{ff} = -\frac{k_v}{k_p} \cdot \frac{1 + T_p s}{1 + T_v s}$$

The high frequency gain is $k_v T_p/(k_p T_v)$. If this ratio is large then the feedforward controller has high gain for high frequencies. This will lead to difficulties if the measurement is corrupted by high frequency noise. Also if the pole excess (I.e. the difference between the number of poles and zeros in the transfer function.) is larger in G_p than in G_v the feedforward controller will result in derivatives of the measured signal. One way to avoid the high frequency problem with feedforward is to only use a static feedforward controller, i.e. to use $G_{ff}(0)$. This will also reduce the need for accurate models.

EXAMPLE 10.3—FEEDFORWARD
Consider the tank system in Figure 10.10. The level is controlled using the input flow. The output flow is a measurable disturbance. It is assumed that there is a calibration error (bias) in the measurement of the output flow. The tank is described by an integrator, i.e. $G_p = 1/s$, and the valve as a first order system

$$G_{valve}(s) = \frac{1}{s + 1}$$

The feedforward controller is in this case given by

$$G_{ff}(s) = \frac{1}{G_{valve}(s)} = s + 1$$

The controller will thus contain a derivation. The derivative can be approximated by the high pass filter

$$s \approx \frac{s}{1 + T_{ff} s}$$

The feedforward is implemented as

$$G_{ff}(s) = \frac{s}{1 + T_{ff} s} + 1 \tag{10.1}$$

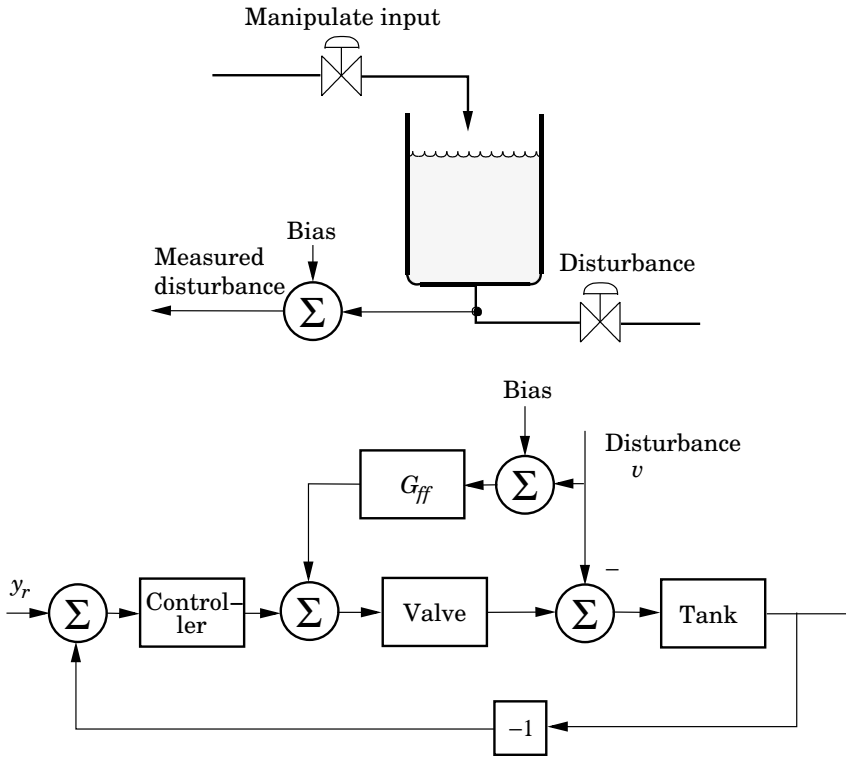


Figure 10.10 Tank system with measurable disturbance.

or as the static feedforward controller

$$G_{ff}(0) = 1 \tag{10.2}$$

Figure 10.11 shows the response to a step disturbance when $T_{ff} = 0.1$ and when (10.1) and (10.2) are used. The reference value is zero. Also the response without feedforward is shown. The response is greatly improved by including the dynamics in the feedforward controller. The bias in the measurement will cause a drift in the output, since the controller does not have any feedback from the output level.

Assume that the controller is extended with a feedback part. Figure 10.12 shows the output and input when both feedforward and feedback are used. The example shows the advantage of combining feedforward with feedback. □

10.6 Deadtime Compensation

The derivative part of the PID controller is used to predict future values of the measurement signal. This is done by studying its derivative. Evidently this method works poorly if the process involves a long delay. In this case one could of course use a PI controller, with the loss of the prediction provided by the D part. This is, however, a significant disadvantage, since the prediction of future control errors is especially useful when we have long process delays. As a consequence special controllers have been developed, which can predict future control errors also in processes with long delays. These predictions are, however, not based on deriving the measurement signal.

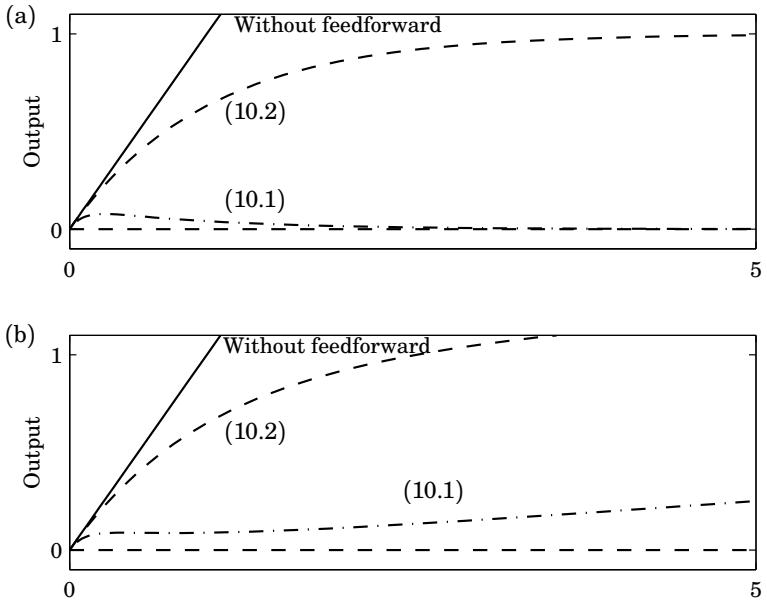


Figure 10.11 Feedforward control of the process in Figure 10.10 using (10.1) and (10.2) with $T_{ff} = 0.1$, when the disturbance is a unit step. The response without feedforward, i.e. the open loop response, is also shown. (a) No measurement bias; (b) Measurement bias 0.05.

The principle of deadtime compensation controllers is to construct the control error prediction based on the control signal, rather than the measurement signal. By devising a model of the process to be controlled and letting the control signal drive both the real process and the model, one can obtain near-future values of the measurement signal of the process by studying the measurement signal of its simulation. The most common delay compensation strategy is implemented in the Smith predictor. A schematic sketch, illustrating the working principle of this strategy is shown in Figure 10.13.

Apart from the usual controller parameters, the Smith predictor needs a model of the process. Especially, it needs to know the length of the process delay. The figure shows how the control signal enters the process as well as models of the process, one with and one without the estimated process delay.

Let us assume that the model is an accurate deception of the process. The two signals y and y_1 will then be identical and therefore cancel each other. The remaining signal entering the controller is y_2, i.e. the signal we would have obtained if there was no delay in the process. This way the controller works against a simulated process, identical to the real process, with its delay removed. The control performance becomes as good as it would have been without the delay, except the fact that the measurement signal is obviously still delayed.

In reality the model is, however, not a perfect description of the process. Consequently, the signal going back into the controller is not identical to y_2. This usually calls for a more conservative tuning of the controller than one would have chosen if the delay was not present.

We shall now give an example of control using the Smith predictor.

EXAMPLE 10.4—THE SMITH PREDICTOR

Figure 10.14 shows the control of a first order process with a delay. The figure shows the control performance of both the PI controller and the Smith counterpart.

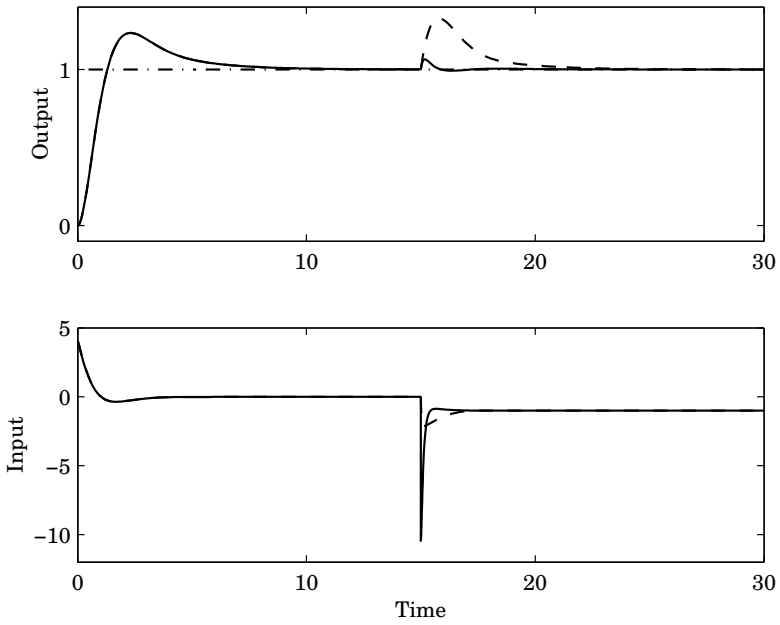


Figure 10.12 Feedforward and feedback control of the process in Figure 10.10 using (10.1) with $T_{ff} = 0.1$ and bias=–0.05. The feedback controller is a PID controller with $K_c = 4$, $T_i = 3$, and $T_d = 0.5$. (a) Output with (full) and without (dashed) feedforward; (b) Input with (full) and without (dashed) feedforward. The reference signal is a unit step at $t = 0$ and the load disturbance is a unit step at $t = 15$.

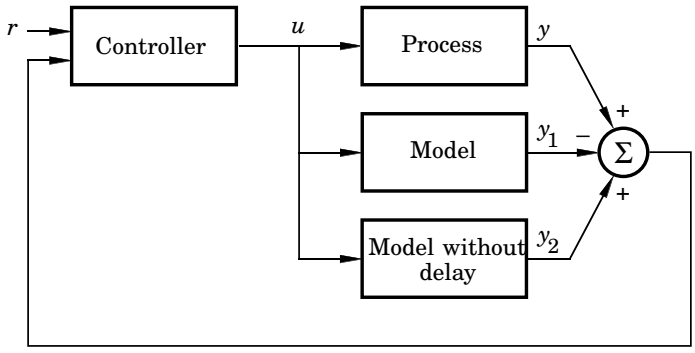


Figure 10.13 The principle of the Smith predictor.

The PI controller is tuned to provide fast control action, without overshoots. At the setpoint change the control signal steadily starts to integrate the control error. The integration is slow enough not to cause any overshoot.

The delay compensation controller is also tuned not to cause any overshoot. It responds to setpoint changes and load disturbances significantly faster than the PI controller. We can especially notice a crucial improvement at setpoint changes, as compared to the PI case. This is because the Smith predictor issues a control signal change before the setpoint change is visible in the measurement signal.

At load disturbances none of the controllers can react before the disturbance is visible in the measurement signal. The only remedy here would be to introduce a feedforward link from the disturbance. □

We shall now conduct a closer study of the Smith predictor. The controller described by Figure 10.13 is equivalent to the block diagram shown in Figure 10.15.

The transfer function of the process is given by

$$G_p(s) = G_{p0}(s)e^{-sL}$$

The estimated process is correspondingly described by the transfer function

$$\hat{G}_p(s) = \hat{G}_{p0}(s)e^{-s\hat{L}}$$

The controller G_{c0}, which is part of the Smith controller, is often a PI controller.

From the block diagram in Figure 10.15 the control signal of the Smith controller can be computed as

$$\begin{aligned} U &= G_{c0}(R - Y + Y_1 - Y_2) \\ &= G_{c0}(R - Y + \hat{G}_{p0}e^{-s\hat{L}}U - \hat{G}_{p0}U) \\ &= G_{c0}(R - Y) + G_{c0}\hat{G}_{p0}(e^{-s\hat{L}} - 1)U \end{aligned}$$

If G_{c0} is a PI controller, we see that the Smith controller can be considered as a PI controller with an added term, which is driven by the control signal u. This is the

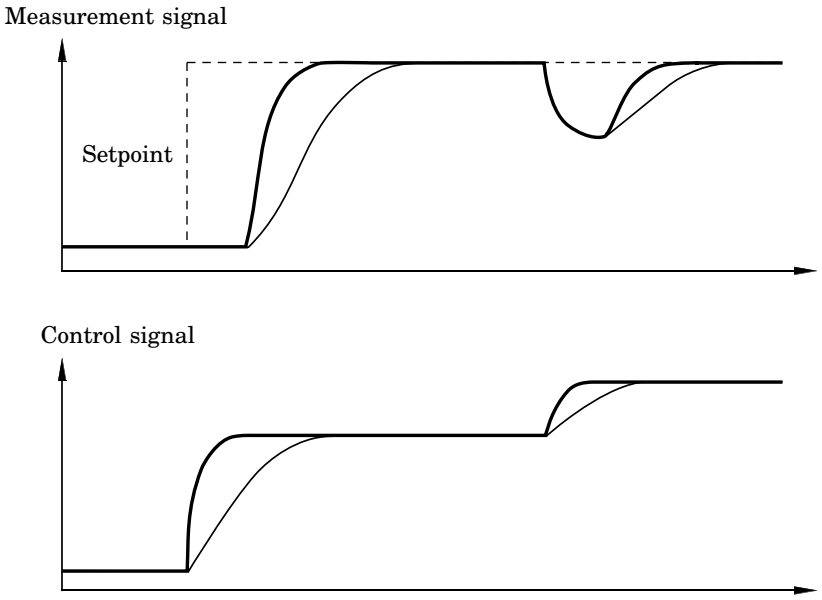


Figure 10.14 Comparison between the Smith controller (thick lines) and a PI controller (thin lines). The figure shows the control performance when a setpoint change is followed by a load disturbance.

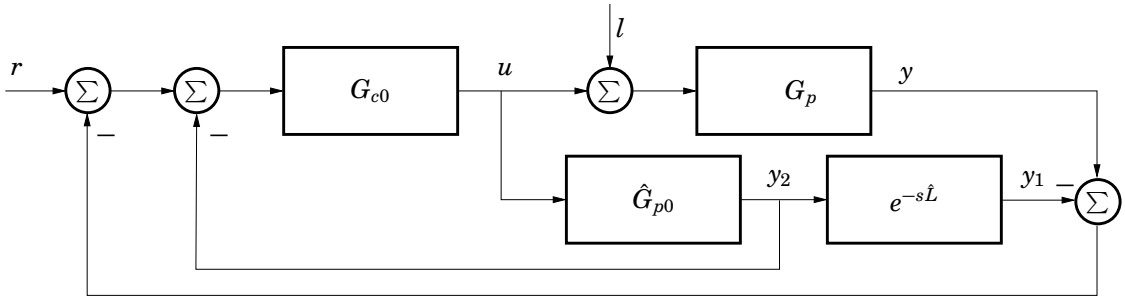


Figure 10.15 Equivalent structure of the Smith predictor.

term responsible for the prediction. In other words we have replaced the measurement signal derivative based prediction in the PID controller with a prediction based on the process model $\hat{G}_p$ and the control signal.

If the model is identical to the real process, i.e. if $\hat{G}_p = G_p$ and if we do not have any process disturbances, the signals y and y_1 will be identical. The only signal re-entering the controller G_{c0} will then be y_2. This yields the control signal

$$U = G_{c0}(R - Y_2) = G_{c0}(R - G_{p0}U) = \frac{G_{c0}}{1 + G_{c0}G_{p0}}R$$

Subsequently, the relation between the setpoint and measurement signal is given by

$$\frac{Y}{R} = \frac{G_p G_{c0}}{1 + G_{p0}G_{c0}} = \frac{G_{p0}G_{c0}}{1 + G_{p0}G_{c0}}e^{-sL}$$

This means that we have a transfer function identical to the case of a process without delay, $G_p = G_{p0}$, except from the fact that the measurement signal is delayed by a time L. Ideally this means that the controller G_{c0} can be determined as if there was no process delay. In reality, however, modelling errors and robustness margins force us to tune the controller more conservatively.

X. Discrete-Time and Sequence Control

X.1 Introduction

In this chapter we will give an overview of sampled-data systems and how computers can be used for implementation of controllers. Standard controllers such as PID-controllers have over the years been implemented using mechanical, pneumatic, hydraulic, electronic, and digital equipment. Most manufactured PID-controllers are today based on microprocessors. This is done because of the cost benefits. If the sampling frequency is high the controllers can be regarded as continuous time controllers. The user can make the tuning and installation without bothering about how the control algorithm is implemented. This has the advantage that the operators do not have to be re-educated when equipment based on new technology is installed. The disadvantage is that the full capacity of sampled-data systems is not used.

Some advantages of using computers to control a process are:

- Increased production
- Better quality of the product
- Improved security
- Better documentation and statistics
- Improved flexibility

One of the most important points above is the increased flexibility. It is, in principle, very simple to change a computer program and introduce a new controller scheme. This is important in industrial processes, where revisions always are made. New equipment is installed and new piping is done etc.

Some Historical Remarks

The development of process control using computers can be divided into five phases:

Pioneering period	$\approx$ 1955
Direct-digital-control period	$\approx$ 1962
Minicomputer period	$\approx$ 1967
Microcomputer period	$\approx$ 1972
General use of digital control	$\approx$ 1980

The years above give the approximate time when different ideas appeared.

The first computer application in the process industry was in March 1959 at the Port Arthur, Texas, refinery. The project was a cooperation between Texaco and the computer company Thomson Ramo Woodridge (TRW). The controlled process was a polymerization unit. The system controlled 26 flows, 72 temperatures, 3 pressures, and 3 compositions. The essential function of the computer was to make an optimization of the feeds and recirculations of the process. During the pioneering period

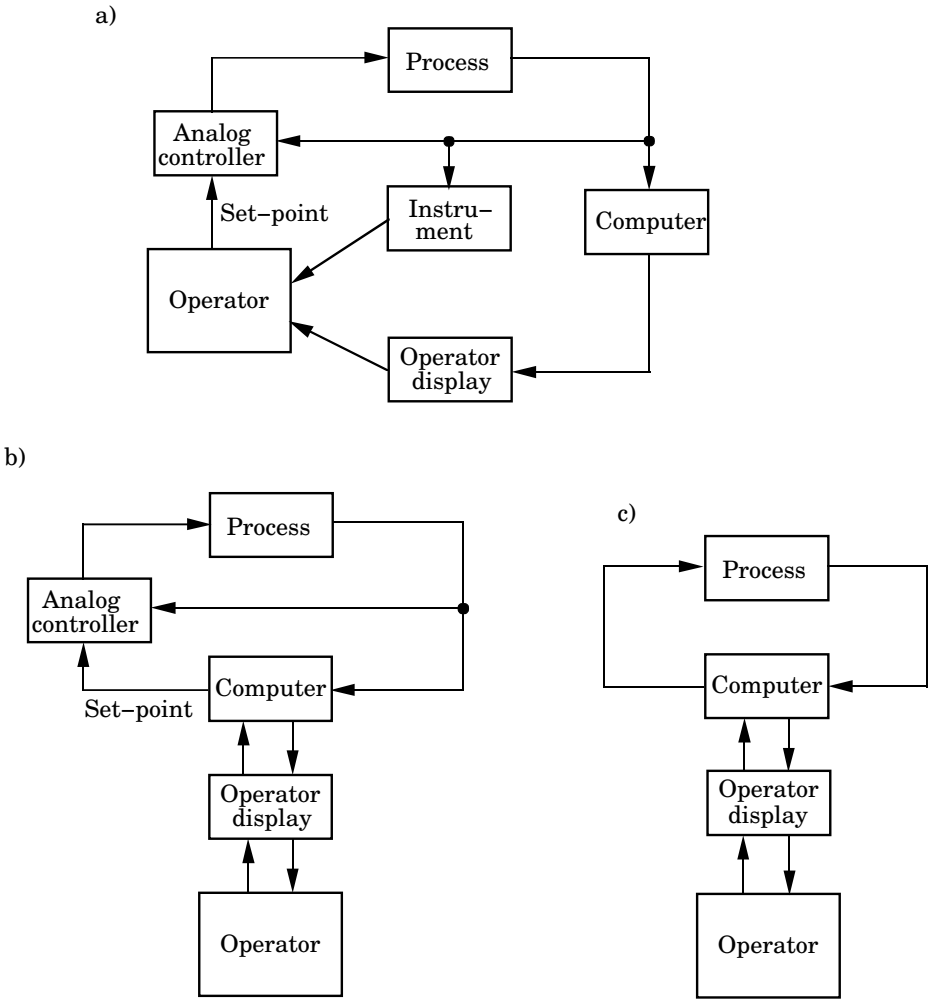


Figure X.1 Different ways to use computers. a) Operator guide; b) Set-point control; c) Direct Digital Control (DDC).

the hardware reliability of the computers was very poor. The Mean Time Between Failures (MTBF) for the central processing unit could be in the range 50–100 hours. The task of the computer was instead to compute and suggest the set-point values to the conventional analog controllers, see Figure X.1 a). The operator then changed the set-points manually. This is called *operator guide*. With increased reliability it became feasible to let the computers change the set points directly, see Figure X.1 b). This is called *set-point control*. The major tasks of the computers were optimization, reporting, and scheduling. The basic theory for sampled-data system was developed during this period.

With increasing reliability of the computers it became possible to replace the conventional controllers with algorithms in the computers. This is called *Direct Digital Control, (DDC)*, see Figure X.1 c). The first installation using DDC was made at Imperial Chemical Industries (ICI) in England in 1962. A complete analog instrumentation was replaced by one computer, a Ferranti Argus. The computer measured 224 variables and controlled directly 129 valves. At this time a typical central processing unit had a MTBF of about 1000 hours. Using DDC it also became more important to consider the operator interfaces. In conventional control rooms the measured values and the controllers are spread out over a large panel, usually covering a whole

Figure X.2 A standard microprocessor-based single-loop controller for process control. (By courtesy of Satt Control Instruments, Stockholm, Sweden.)

wall in a room. Using video displays it became possible to improve the information flow to the process operators.

Integrated circuits and the development of electronics and computers during the end of the sixties lead to the concept of minicomputers. As the name indicates the computers became smaller, faster, cheaper, and more reliable. Now it became cost-effective to use computers in many applications. A typical computer from this time is CDC 1700 with an addition time of 2 μs and a multiplication time of 7 μs. The primary memory was in the range 8–124k words. The MTFB for the central processing unit was about 20 000 hours. The computers were now equipped with hardware that made it easy to connect them to the processes. Special real-time operating systems were developed which made it easier to program the computers for process control applications.

The development with very large scale integration (VLSI) electronics made it possible to make a computer on one integrated circuit. Intel developed the first microprocessors in the beginning of the seventies. The microprocessors made it possible to have computing power in almost any equipment such as sensors, analyzers, and controllers. A typical standard single loop controller based on a microprocessor is shown in Figure X.2. This type of standard controllers are very common in the industry. Using a microprocessor makes it possible to incorporate more features in the controller. For instance, auto-tuning and adaptivity can be introduced.

From the beginning of the eighties the computers have come into more general use and the development have continued both on the hardware and the software. Today most control systems are built up as distributed systems. The system can be easily expanded and the reliability has increased with the distributed computation and

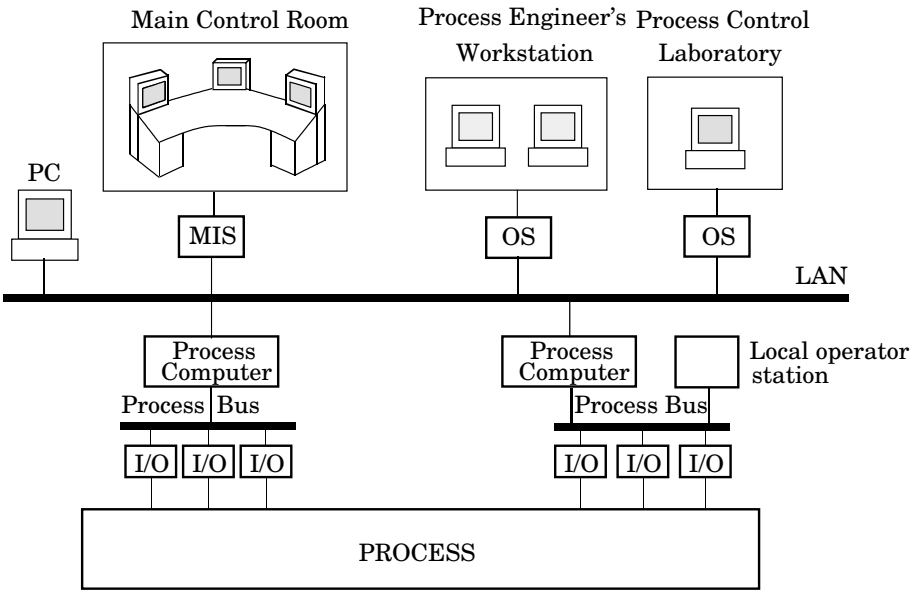


Figure X.3 Architecture of a distributed control system.

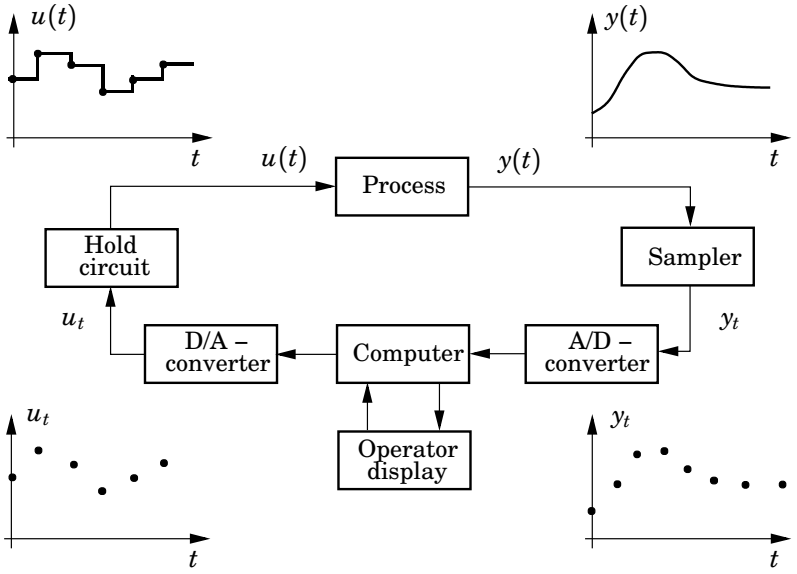


Figure X.4 Computer control of a process.

communication. A typical architecture of a distributed control system is shown in Figure X.3.

X.2 Sampled-Data Systems

In the context of control and communication, sampling means that a continuous time signal is replaced by a sequence of numbers. Figure X.4 shows the signals in the different parts of the system when a process is controlled using a computer. The output of the process is sampled and converted into a number using a analog-to-digital (A-D) converter. The sampler and converter are usually built into one unit. The sampling times are determined by the real-time clock in the computer. The sampled

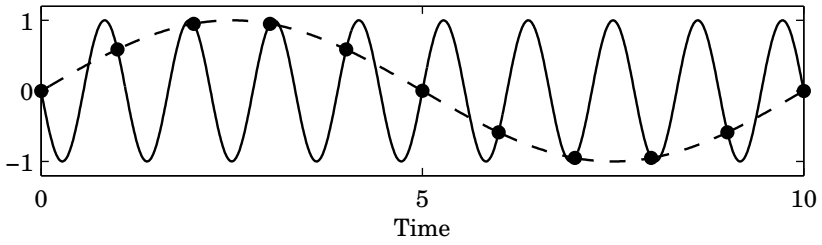


Figure X.5 Illustration of aliasing. The sinusoidal signals have the frequencies 0.1 and 0.9 Hz respectively. The sampling frequency is 1 Hz.

signal is represented as a digital number. Typically the converter has a resolution of 8–16 bits. This implies that the sampled signal is quantized both in time and amplitude. The time between the sampling instances is called the *sampling period*. The sampling is usually periodic, i.e. the times between the samples are equal. The sampling frequency ω_s and the sampling period h are related through

$$\omega_s = \frac{2\pi}{h}$$

The controller is an algorithm or program in the computer that determines the sequence of control signals to the process. The control signals are converted into an analog signal by using a digital-to-analog (D-A) converter and a hold-circuit. The hold circuit converts the sequence of numbers to a time-continuous signal. The most common hold circuit is the *zero-order-hold*, which keeps the signal constant between the D-A conversions. The end result is that the process is controlled by a piece-wise constant signal. This implies that the output of the system can be regarded as a sequence of step responses for the open loop system. Due to the periodicity of the control signal there will be a periodicity also in the closed loop system. This makes it difficult to analyze sampled-data systems. The analysis will, however, be considerably simplified if we only consider the behavior of the system at the sampling points. We then only investigate how the output or the states of the system is changed from sampling time to sampling time. The system will then be described by *recursive* or *difference equations*.

Aliasing

When sampling the output of the process we must choose the sampling period. Intuitively it is clear that we may loose information if the sampling is too sparse. This is illustrated in Figure X.5, where two sinusoidal signals are sampled. The sinusoidal signals have the frequencies 0.1 and 0.9 Hz respectively. When sampling with a frequency of 1 Hz the signals have the same value *at each sampling instant*. This implies that we can't distinguish between the signals after the sampling and information is lost. The high frequency signal can be interpreted as a low frequency signal. This is called *aliasing* or *frequency folding*. The aliasing problem may become serious in control systems when the measured signal has a high frequency component. The sampled signal will then contain a low frequency alias signal, which the controller may try to compensate for.

EXAMPLE X.1—ALIASING

Figure X.6 is a process diagram of feedwater heating in a boiler of a ship. A valve controls the flow of water. Unintentionally there is a backlash in the valve positioner

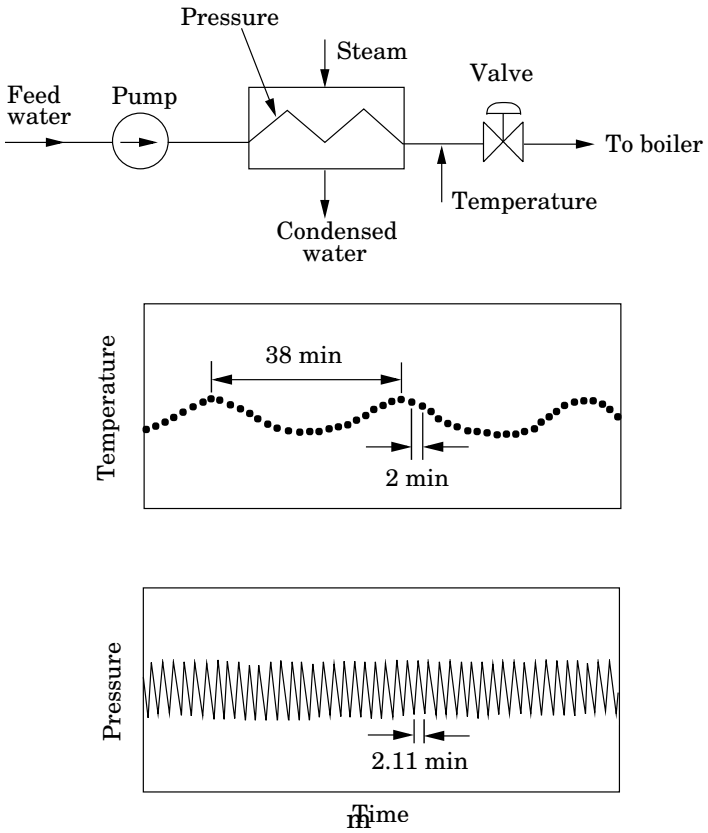


Figure X.6 Process diagram for a feedwater heating system of a boiler.

due to wear. This causes the temperature to oscillate. Figure X.6 shows a sampled recording of the temperature and a continuous time recording of the pressure. The sampling period of the temperature is 2 min. From the temperature recording one might believe that there is an oscillation with a period of 38 min. Due to the physical coupling between the temperature and the pressure we can conclude that also the temperature must have an oscillation of 2.11 min. The 38 min oscillation is the alias frequency. □

Can aliasing be avoided? The sampling theorem by Shannon answers this question. To be able to reconstruct the continuous time signal from the sampled signal it is necessary that the sampling frequency is at least twice as high as the highest frequency in the signal. This implies that there must be at least two samples per period. To avoid the aliasing problem it is necessary to filter all the signals before the sampling. All frequencies above the *Nyquist frequency*

$$\omega_N = \frac{\omega_s}{2} = \frac{\pi}{h}$$

should ideally be removed.

EXAMPLE X.2—PREFILTERING
Figure X.7 a) shows a signal (the square wave) disturbed by high frequency sinusoidal measurement noise. Figure X.7 c) shows the sample and hold of the signal. The aliasing is clearly visible. The signal is then prefiltered before the sampling giving the signal in Figure X.7 b). The measurement noise is eliminated, but some of the

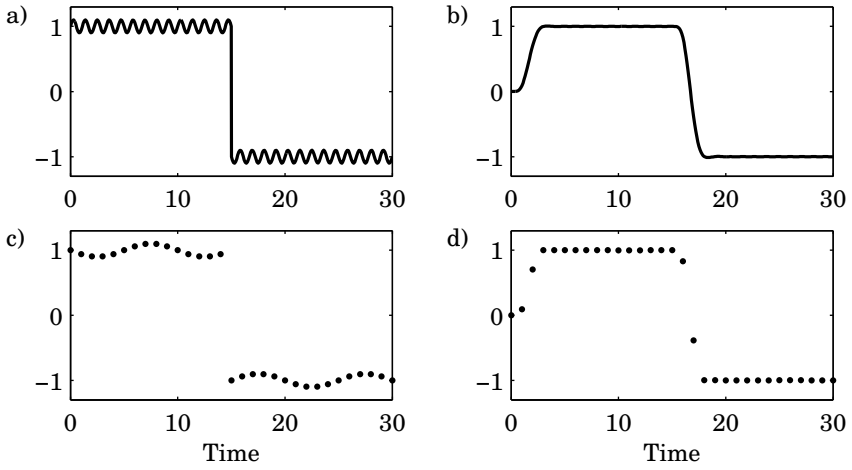


Figure X.7 Usefulness of a prefilter. a) Signal plus sinusoidal disturbance; b) The signal filtered through a sixth-order Bessel filter; c) Sample and hold of the signal in a); d) Sample and hold of the signal in b).

high frequency components in the desired signal are also removed. Sample and hold of the filtered signal is shown in Figure X.7 d). The example shows that the prefiltering is necessary and that we must compromise between the elimination of the noise and how high frequencies that are left after the filtering. It important that the bandwidth of the prefilter is adjusted to the sampling frequency. □

X.3 Description of Sampled-Data Systems

In this section we will briefly discuss how to describe sampled-data systems. The idea is to look at the signals only at the sampling points. Further, we will only discuss the problem when the sampling interval h is constant. Let the process be described by the continuous time state-space model

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{X.1}$$

Assume that the initial value is given and that the input signal is constant over the sampling interval. The solution of (X.1) is now given by (3.7). Let t and $t+h$ be two consecutive sampling times.

$$\begin{aligned} x(t+h) &= e^{Ah}x(t) + \int_t^{t+h} e^{A(t-\tau)}Bu(\tau)\,d\tau \\ &= e^{Ah}x(t) + \int_t^{t+h} e^{A(t-\tau)}B\,d\tau\,u(t) \\ &= e^{Ah}x(t) + \int_0^{h} e^{A\tau'}B\,d\tau'\,u(t) \\ &= \Phi x(t) + \Gamma u(t) \end{aligned}$$

In the second equality we have used the fact that the input is constant over the interval $[t,\, t+h]$. The third equality is obtained by change of integration variable

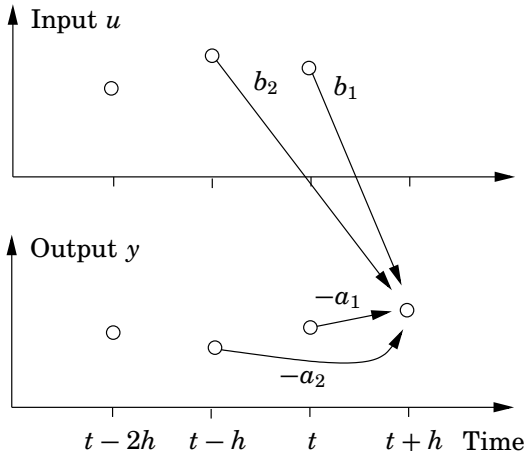


Figure X.8 Illustration of the difference equation $y(t) + a_1 y(t-h) + a_2 y(t-2h) = b_1 u(t-h) + b_2 u(t-2h)$.

$\tau' = t - \tau$. A difference equation is now obtained, which determines how the state is changed from time t to time $t+h$.

$$\begin{aligned} x(t+h) &= \Phi x(t) + \Gamma u(t) \\ y(t) &= Cx(t) \end{aligned} \tag{X.2}$$

The sampled data system is thus described by a difference equation. The solution of the difference equation is obtained recursively through iteration. In the same way as for continuous time systems it is possible to eliminate the states and derive an input output model of the system. Transform theory plays the same role for sampled data systems as the Laplace transform for continuous time systems. For sampled-data systems the z-transform is used. We can also introduce the *shift operator* q, which is defined as

$$qy(t) = y(t+h)$$

i.e. multiplication by the operator q implies shift of the time argument one sampling period ahead. Multiplication by q^{-1} is the same as backward shift. Using the shift operator on (X.2) gives

$$qx(t) = \Phi x(t) + \Gamma u(t)$$

Solving for $x(t)$ gives

$$x(t) = (qI - \Phi)^{-1} \Gamma u(t)$$

Compare (4.2). The input-output model is thus

$$y(t) = C\,(qI - \Phi)^{-1} \Gamma u(t) = H(q)u(t) \tag{X.3}$$

where $H(q)$ is called the *pulse transfer function*. Equation (X.3) corresponds to a higher order difference equation of the form

$$\begin{aligned} y(t) + a_1 y(t-h) + a_2 y(t-2h) + \cdots + a_n y(t-nh) \\ = b_1 u(t-h) + \cdots + b_n u(t-nh) \end{aligned} \tag{X.4}$$

The output of the difference equation (X.4) is thus a weighted sum of previous inputs and outputs. See Figure X.8.

EXAMPLE X.3—RECURSIVE EQUATION
Assume that a sampled-data system is described by the recursive equation

$$y(t+1) - 0.6y(t) = 0.5u(t) \tag{X.5}$$

where $y(0) = 0$ and $u(t) = 1$ when $t \geq 0$. We can now compute the output of (X.5).

t	$u(t)$	$y(t)$
0	1	0
1	1	0.5
2	1	0.8
3	1	0.98
4	1	1.088
5	1	1.153
6	1	1.192
7	1	1.215
8	1	1.229
9	1	1.237
10	1	1.242
⋮	⋮	⋮
∞	1	1.25

The stationary value can be obtained by assuming that $y(t+1) = y(t) = y(\infty)$ in (X.5). This gives

$$y(\infty) = \frac{0.5}{1 - 0.6} = 1.25$$

□

Given a continuous time system and assuming that the control signal is constant over the sampling period is is thus easy to get a representation that describes how the system behaves at the sampling instants. Stability, controllability, observability, etc can then be investigated very similar to continuous time systems. There are also design methods that are the same as for continuous time systems. This is, however, outside the scope of this book.

X.4 Discrete-Time Approximations

It is sometimes of interest to make a transformation of a continuous time controller or a transfer function into a discrete time implementation. This can be done using approximations of the same kind as when making numerical integration. This can be of interest when a good continuous time controller is available and we only want to replace the continuous time implementation by a sampled-data implementation. See Figure X.9.

The simplest ways are to approximate the continuous time derivative using simple difference approximations. We can use *forward difference (Euler's approximation)*

$$\frac{dy}{dt} \approx \frac{y(t+h) - y(t)}{h}$$

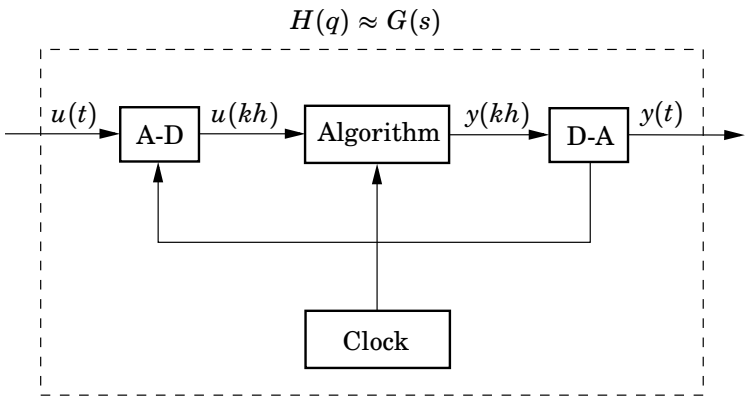


Figure X.9 Approximating a continuous time transfer function, $G(s)$, using a computer.

or *backward difference*

$$\frac{dy}{dt} \approx \frac{y(t) - y(t-h)}{h}$$

Higher order derivatives are obtained by taking the difference several times. These types of approximations are good only if the sampling period is short. The forward difference approximation corresponds to replacing each s in the transfer function by $(q-1)/h$. This gives a discrete time pulse transfer function $H(q)$ that can be implemented as a computer program. Using backward difference we instead replace all s by $(1-q^{-1})/h$.

EXAMPLE X.4—DIFFERENCE APPROXIMATION

Assume that we want to make a difference approximation of a transfer function

$$G(s) = \frac{2}{s+3}$$

The system is thus described by the differential equation

$$\frac{dy(t)}{dt} = -3y(t) + 2u(t)$$

Using forward difference approximation we get

$$\frac{dy}{dt} \approx \frac{y(t+h) - y(t)}{h} = -3y(t) + 2u(t)$$

Rearrangement of the terms gives the difference equation

$$y(t+h) = (1-3h)y(t) + 2hu(t)$$

Using backward difference approximation we instead get

$$\frac{dy}{dt} \approx \frac{y(t) - y(t-h)}{h} = -3y(t) + 2u(t)$$

or

$$y(t) = \frac{1}{1+3h}\left(y(t-h) + 2hu(t)\right)$$

□

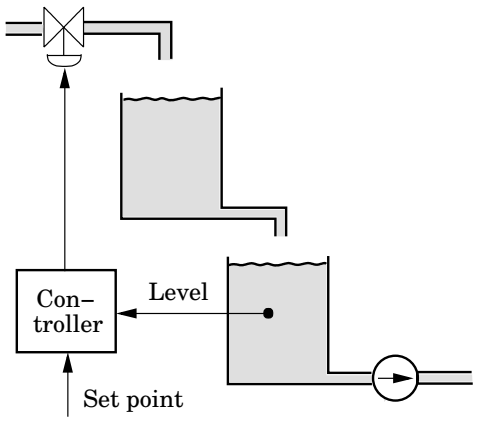


Figure X.10 Level control of a two-tank system.

The choice of the sampling interval depends on many factors. For the approximations above one rule of thumb is to choose

$$h\omega_c \approx 0.15 - 0.5$$

where ω_c is the crossover frequency (in rad/s) of the continuous time system. This rule gives quite short sampling periods. The Nyquist frequency will be about 5–20 time larger than the crossover frequency.

EXAMPLE X.5—APPROXIMATION OF A CONTINUOUS TIME CONTROLLER
Consider the process in Figure X.10. There are two tanks in the process and the outlet of the second tank is kept constant by a pump. The control signal is the inflow to the first tank. The process can be assumed to have the transfer function

$$G_p(s) = \frac{1}{s(s+1)} \tag{X.6}$$

i.e. there is an integrator in the process. Assume that the system is satisfactorily controlled by a continuous time controller

$$G_c(s) = 4\,\frac{s+0.8}{s+3.2} \tag{X.7}$$

We now want to implement the controller using a computer. Using Euler approximation gives the approximation

$$H(q) = 4\,\frac{\dfrac{q-1}{h} + 0.8}{\dfrac{q-1}{h} + 3.2} = 4\,\frac{q-1+0.8h}{q-1+3.2h} \tag{X.8}$$

The crossover frequency of the continuous time system (X.6) in cascade with (X.7) is $\omega_c = 1.7$ rad/s. The rule of thumb above gives a sampling period of about 0.1 – 0.3.

Figure X.11 shows the output and the input of the system when it is controlled by the continuous time controller (X.7) and the sampled-data controller (X.8) for different sampling times. □

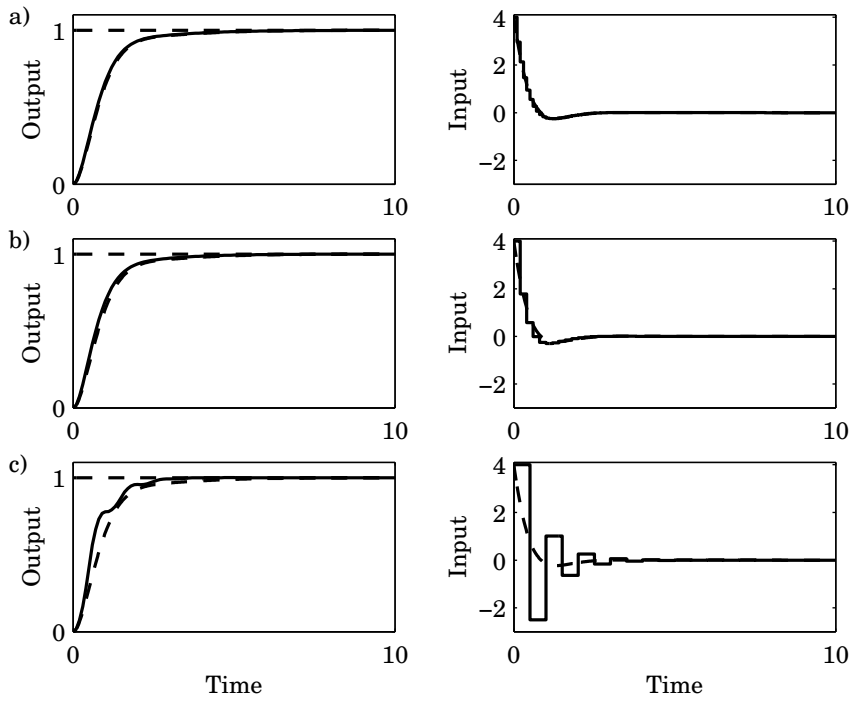


Figure X.11 Process output and input when the process (X.6) is controlled by (X.7) and the sampled-data controller (X.8) when a) $h = 0.1$; b) $h = 0.2$; c) $h = 0.5$. The dashed signals are the output and input when the continuous time controller is used.

X.5 Sampled-Data PID Controllers

A digital computer can neither take derivatives nor compute integrals exactly. To implement a PID controller using a digital computer, it is therefore necessary to make some approximations. As a starting point, we take the modified PID algorithm derived in Section 9.5 (with derivative set-point weighting $\gamma = 0$):

$$U(s) = K\left(\beta R(s) - Y(s) + \frac{1}{sT_i}\left(R(s) - Y(s)\right) - \frac{sT_d}{1 + sT_d/N}Y(s)\right)$$

The proportional part

$$P(t) = K\left(\beta r(t) - y(t)\right) \tag{X.9}$$

requires no approximation since it is a purely static relation. The integral term

$$I(t) = \frac{K}{T_i}\int^{t} e(\tau)\,d\tau$$

is approximated by a rectangular approximation, i.e.

$$I(kh + h) = I(kh) + \frac{Kh}{T_i}e(kh) \tag{X.10}$$

The derivative part given by

$$\frac{T_d}{N}\frac{dD}{dt} + D = -KT_d\frac{dy}{dt}$$

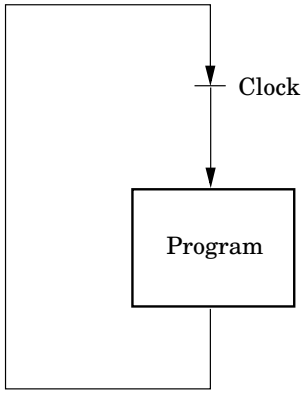


Figure X.12 A simple operating system.

is approximated by taking backward differences. This gives

$$D(kh) = \frac{T_d}{T_d + Nh} D(kh - h) - \frac{KT_dN}{T_d + Nh} (y(kh) - y(kh - h)) \tag{X.11}$$

This approximation has the advantage that it is always stable and that the sampled data pole goes to zero when T_d goes to zero. The control signal is given as

$$u(kh) = P(kh) + I(kh) + D(kh)$$

This approximation has the pedagogical advantage that the proportional, integral, and derivative terms are obtained separately. There are many other approximations, which are described in detail in textbooks on digital control.

To introduce a digital version of the anti-windup scheme, we simply compute a signal

$$v(kh) = P(kh) + I(kh) + D(kh)$$

The controller output is then given by

$$u = \text{sat}(v, u_{min}, u_{max}) = \begin{cases} u_{min} & v < u_{min} \\ v & u_{min} \le v \le u_{max} \\ u_{max} & u > u_{max} \end{cases} \tag{X.12}$$

and the updating of the integral term given by Equation (X.10) is replaced by

$$I(kh + h) = I(kh) + \frac{Kh}{T_i} e(kh) + \frac{h}{T_r} (u(kh) - v(kh)) \tag{X.13}$$

To implement the controller using a digital computer, it is also necessary to have analog to digital converters that convert the set point r and the measure value y to a digital number. It is also necessary to have a digital to analog converter that converts the computed output u to an analog signal that can be applied to the process. To ensure that the control algorithm gets synchronized, it is also necessary to have a clock so that the control algorithm is executed once every h time units. This is handled by an operating system. A simple form of such a system is illustrated in Figure X.12.

The system works like this. The clock gives an interrupt signal each sampling instant. When the interrupt occurs, the following program is executed:

Analog to digital (AD) conversion of r and y

Compute P from (X.9)

Compute D from (X.11)

Compute $v = P + I + D$

Compute u from (X.12)

Digital to analog (DA) conversion of u

Compute I from (X.13)

Wait for next clock pulse

When the interrupt occurs, digital representations of set point r and measured value y are obtained from the analog to digital conversion. The control signal u is computed using the approximations described earlier. The numerical representation of u is converted to an analog signal using the DA converter. The program then waits for the next clock signal.

Selection of Sampling Interval and Word Length

The sampling interval is an important parameter in a digital control system. The parameter must be chosen sufficiently small so that the approximations used are accurate, but not so small that there will be numerical difficulties.

Several rules of thumb for choosing the sampling period for a digital PID controller are given in the literature. There is a significant difference between PI and PID controllers. For PI controllers the sampling period is related to the integration time. A typical rule of thumb is

$$\frac{h}{T_i} \approx 0.1 - 0.3$$

when Ziegler-Nichols tuning is used this implies

$$\frac{h}{L} \approx 0.3 - 1$$

where L is the apparent dead-time or equivalently

$$\frac{h}{T_u} \approx 0.1 - 0.3$$

where T_u is the ultimate period.

With PID control the critical issue is that the sampling period must be so short that the phase lead is not adversely affected by the sampling. This implies that the sampling period should be chosen so that the number hN/T_d is in the range of 0.2 to 0.6. With $N = 10$ this means that for Ziegler-Nichols tuning we have

$$\frac{h}{L} \approx 0.01 - 0.03$$

or

$$\frac{h}{T_u} \approx 0.0025 - 0.0075$$

Controllers with derivative action thus require significantly shorter sampling periods than PI controllers.

Commercial digital controllers for few loops often have a short fixed sampling interval on the order of 200 ms. This implies that PI control can be used for processes with ultimate periods larger than 0.6 s but that PID controllers can be used for processes with ultimate periods larger than 25 s.

From the above discussion it may appear advantageous to select the sampling interval as short as possible. There are, however, also drawbacks by choosing a very short sampling period. Consider calculation of the integral term. Computational problems, such as *integration offset* may occur due to the finite precision in the number representation used in the computer. Assume that there is an error, $e(kh)$. The integrator term is then increased at each sampling time with

$$\frac{Kh}{T_i}e(kh)$$

Assume that the gain is small or that the reset time is large compared to the sampling time. The change in the output may then be smaller than the quantization step in the DA-converter. For instance, a 12-bit DA converter (i.e., a resolution of 1/4096) should give sufficiently good resolution for control purposes. Yet if $K = h = 1$ and $T_i = 3600$, then any error less than 90% of the span of the DA converter gives a calculated change in the integral part less than the quantization step. There will be an offset in the output if the integral part is stored with the same number of digits as used in the DA converter. One way to avoid this is to use higher precision in the internal calculations that are less than the quantization level of the output. Frequently at least 24 bits are used to implement the integral part in a computer, in order to avoid integration offset. It is also useful to use a longer sampling interval when computing the integral term.

X.6 Logical Nets and Sequence Control

The history of logical nets and sequential processes starts with batch processing, which is very important in many process industries. This implies that control actions are done in a sequence where the next step depends on some conditions. Simple examples are recipes for cooking and instructions or manuals for equipment. Washing machines, dish washers, and batch manufacturing of chemicals are other examples. The recipes or instructions can be divided into a sequence of steps. The transition from one step to the next can be determined by a logical expression. In this chapter logical and sequence controllers are presented.

Sequence control is an integrated part of many control systems, and can be used for batch control, start-up and shut-down procedures, interlooks, and alarm supervision. When implementing such systems it is useful to be aware of the notations and traditions in sequence control as it developed before it started to be integrated with automatic control.

Logical nets can be implemented in many different ways, for instance, using clever mechanical constructions, relays, transistors, or computers. The computer implementations are often called PLC (Programmable Logical Controller) systems. An earlier notation was PC (Programmable Controller) systems. This was used before PC became synonymous with personal computer. Logical systems are built up by variables that can be true or false, i.e. the variables can only take one of two values. For instance, a relay may be drawn or not, an alarm may be set or not, a temperature may

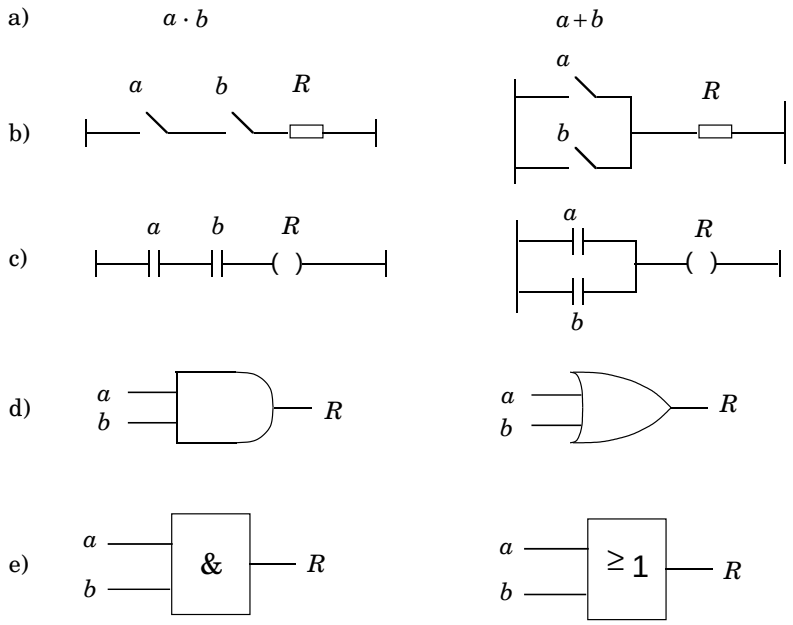


Figure X.13 Different ways to represent the logical expressions $a \cdot b$ and $a + b$: (a) Boolean algebra; (b) Relay symbols; (c) Ladder symbols; (d) Logical circuit symbols (American standard); (e) Logical circuit symbols (Swedish standard).

be over a limit or not. The output of a logical net is also a two-valued variable, a motor may be started or not, a lamp is turned on or off, a contactor is activated or not. The mathematical basis for handling this type of systems is Boolean algebra. This algebra was developed by the English mathematician George Boole in the 1850's. It was, however, not until after about a century that it became a widely used tool to analyze and simplify logical circuits.

Logical variables can take the values true or false. These values are often also denoted by 1 (true) or 0 (false). Boolean algebra contains three operations `or`, `and`, and `not`. We have the following notations:

`and:`	$a \cdot b$	a `and` b	$a \wedge b$
`or:`	$a + b$	a `or` b	$a \vee b$
`not:`	$\bar{a}$	`not` a	$\neg a$

The expression $a \cdot b$ is true only if both a and b are true at the same time. The expression $a+b$ is true if either a or b or both a and b are true. Finally the expression $\bar{a}$ is true only if a is false. In the logical circuit symbols a ring on the input denotes negation of the signal and a ring on the output denotes negation of the computed expression. The `and` and `or` expressions can also be interpreted using relay symbols as in Figures X.13 and X.14. The `and` operator is the same as series connection of two relays. There is only a connection if both relays are drawn. The `or` operator is the same as parallel connection of two relays. There is a connection whenever at least one of the relays are drawn. The relay or ladder representation of logical nets is often used for documentation and programming of PLCs. We will use the more programming or computer oriented approach with $\cdot$ and $+$. This usually gives more compact expressions and is also more suited for algebraic manipulations. Exactly as in ordinary algebra we may simplify the writing by omitting the `and`-operator, i.e. to write ab instead of $a \cdot b$.

To make the algebra complete we have to define the *unit* and *zero elements*. These

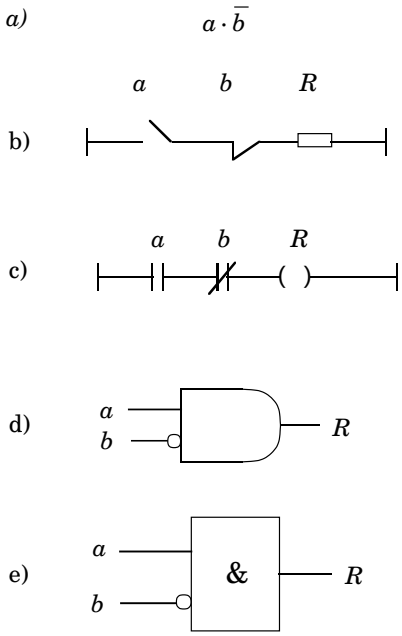


Figure X.14 Different ways to represent the logical expression $a \cdot \bar{b}$: (a) Boolean algebra; (b) Relay symbols; (c) Ladder symbols; (d) Logical circuit symbols (American standard); (e) Logical circuit symbols (Swedish standard).

are denoted by 1 and 0 respectively. We have the following axioms for the Boolean algebra:

$$\begin{aligned}
&\bar{0} = 1 \\
&\bar{1} = 0 \\
&1 + a = 1 \\
&0 + a = a \\
&1 \cdot a = a \\
&0 \cdot a = 0 \\
&a + a = a \\
&a + \bar{a} = 1 \\
&a \cdot \bar{a} = 0 \\
&a \cdot a = a \\
&\bar{\bar{a}} = a
\end{aligned}$$

We further have the following rules for calculation:

$a + b = b + a$	Commutative law
$a \cdot b = b \cdot a$	Commutative law
$a \cdot (b + c) = a \cdot b + a \cdot c$	Distributive law
$a \cdot (b \cdot c) = (a \cdot b) \cdot c$	Associative law
$a + (b + c) = (a + b) + c$	Associative law
$\overline{a + b} = \bar{a} \cdot \bar{b}$	de Morgan's law
$\overline{a \cdot b} = \bar{a} + \bar{b}$	de Morgan's law

A logical net can be regarded as a static system. For each combination of the input signals there is only one output value that can be obtained.

In many applications it can be easy to write down the logical expressions for the system. In other applications the expressions can be quite complicated and it can be desirable to simplify the expressions. One reason for making the simplification is that the simplified expressions give a clearer understanding of the operation of the network. The rules above can be used to simplify logical expressions. One very useful rule is the following

$$a + a \cdot b = a \cdot 1 + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a \tag{X.14}$$

One way to test equality between two logical expressions is a *truth table*. The truth table consists of all combinations of the input variables and the evaluation of the two expressions. Since the inputs only can take two values there will be 2^n combinations, where n is the number of inputs. The expressions are equal if they have the same value for all combinations.

EXAMPLE X.6—TRUTH TABLE
For instance (X.14) is proved by using the table

a	b	$a + ab$	a
0	0	0	0
0	1	0	0
1	0	1	1
1	1	1	1

To the left we write all possible combinations of the input variables. To the right we write the value of the expressions of the left and right hand sides of (X.14). The last two columns are the same for all possible combinations of a and b, which proves the equality. □

There are systematic methods to make an automatic reduction of a logical expression. The methodologies will only be illustrated by an example.

EXAMPLE X.7—SYSTEMATIC SIMPLIFICATION OF A LOGICAL NETWORK
Consider a logical network that has three inputs a, b, and c and one output y. The network is defined by the following truth table:

	a	b	c	y
v_0	0	0	0	0
v_1	0	0	1	0
v_2	0	1	0	0
v_3	0	1	1	1
v_4	1	0	0	1
v_5	1	0	1	1
v_6	1	1	0	1
v_7	1	1	1	1

The different combinations of the inputs are denoted v_i, where the index i corresponds to the evaluation of the binary number abc. I.e. the combination $abc = 101$ corresponds to the number $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$. The expression for the output y can be expressed in two ways: either as the logical `or` of the combinations when the output is true or as the negation of the logical `or` of the combinations when the output is false. Using the first representation we can write

$$\begin{aligned} y &= v_3 + v_4 + v_5 + v_6 + v_7 \\ &= \bar{a}bc + a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c} + abc \\ &= bc(\bar{a} + a) + a\bar{b}(\bar{c} + c) + ab(\bar{c} + c) \\ &= bc + a\bar{b} + ab = bc + a(\bar{b} + b) \\ &= a + bc \end{aligned}$$

The first equality is obtained from the truth table. The second equality is obtained by combining the terms v_3 with v_7, v_4 with v_5, and v_6 with v_7. It is possible to use v_7 two times since $v_7 + v_7 = v_7$. The simplifications are then given from the computational rules listed above.

The second way to do the simplification is to write

$$\begin{aligned} \bar{y} &= v_0 + v_1 + v_2 \\ &= \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} \\ &= \bar{a}\bar{b}(\bar{c} + c) + \bar{a}\bar{c}(\bar{b} + b) \\ &= \bar{a}\bar{b} + \bar{a}\bar{c} = \bar{a}(\bar{b} + \bar{c}) \end{aligned}$$

This gives

$$y = \bar{\bar{y}} = \overline{\bar{a}(\bar{b} + \bar{c})} = a + bc$$

which is the same as before. □

Using the methodology described in the example above it is possible to reduce a complicated logical expression into its simplest form. A more formal presentation of the methods are outside the scope of this book.

PLC Implementation

Most PLC units are implemented using microprocessors. This implies that the logical inputs must be scanned periodically. A typical block diagram is shown in Figure X.15. The execution of the PLC program can be done in the following way:

1. Input-copying. Read all logical input variables and store them in I/O memory.
2. Scan through the program for the logical net and store the computed values of the outputs in the I/O memory.
3. Output-copying. Send the values of output signals from the I/O memory to the process.
4. Repeat from 1.

The code for the logical net is executed as fast as possible. The time for execution will, however, depend on the length of the code. The I/O-copying in Step 1 is done to prevent the logical signals to change value during the execution of the code. Finally all outputs are changed at the same time. The programming of the PLC-unit can be

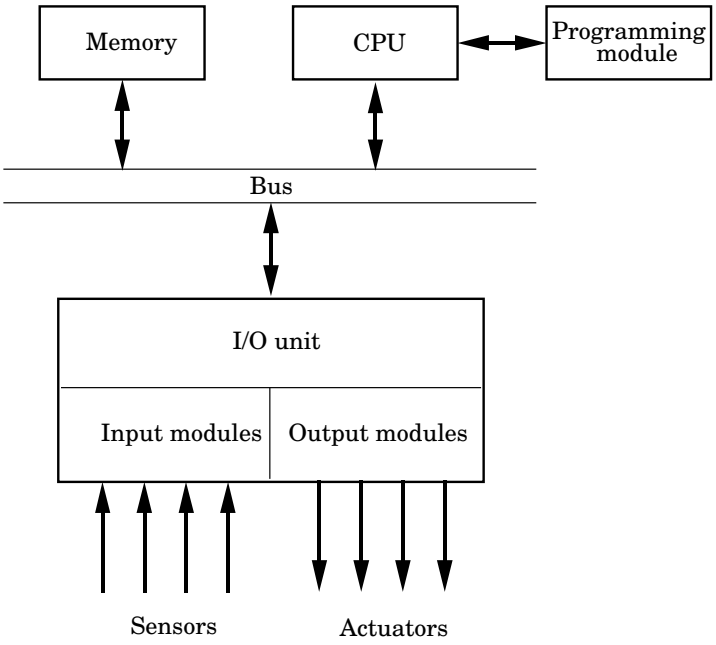


Figure X.15 Block diagram for a PLC system.

done from a small programming module or by using a larger computer with a more effective editor.

The programming is done based on operations such as logical `and`, logical `or`, and logical `not`. Also there are typically combinations such as `nand` and `nor`, which are defined as

$$a \,\texttt{nand}\, b = \overline{a \cdot b} = \bar{a} + \bar{b}$$

$$a \,\texttt{nor}\, b = \overline{a + b} = \bar{a} \cdot \bar{b}$$

Further there are operations to set timers, to make conditional jumps, to increase and decrease counters etc. The specific details differ from manufacturer to manufacturer.

X.7 Sequence Controllers

The logical nets in the previous section are static systems in the sense that the same values of the input signals all the time give the same output signals. There are, however, situations where the outputs should depend on the previous history of the input signals. To describe such systems we introduce *memory* or *states*. We then obtain a *sequential net* or a *sequential process*. Sequences can either be serial or parallel. As the name indicates the serial sequences are run through one step at a time in a series. In parallel sequences we allow several things to happen in parallel with occasional synchronization between the different lines of action. A typical example of a parallel sequence is the assembly of a car. Different subparts of the car, for instance, motor and gearbox, can be assembled in parallel, but the final assembly cannot be done before all subparts are ready.

A sequence can be driven in different ways, by time or by events. A typical time driven sequence is a piece of music. Time driven sequences can be regarded as feedforward control. The chain of events is triggered by the clock and not by the state of the system. An event-driven sequence is a feedback system, where the state of the system determines what will happen next.

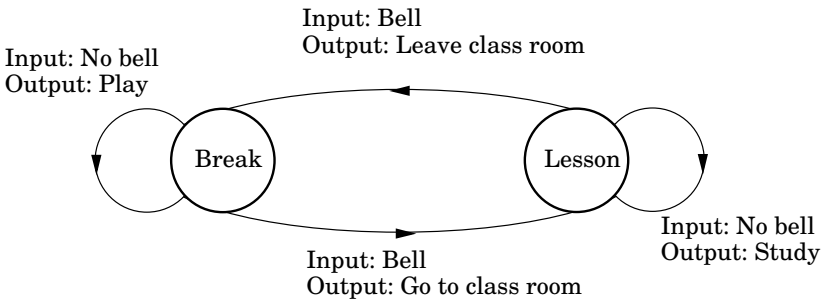


Figure X.16 Simple sequential net for describing a school day.

EXAMPLE X.8—SIMPLE EVENT-DRIVEN SEQUENTIAL NET
Consider the system described in Figure X.16. It can describe a school day. It has two states 'break' and 'lesson'. When the bell rings we get a transition from one state to the other. Which output we get depends on the current state and the input signal. If starting in the state 'break' and there is no bell ringing we stay in the state break. When the bell calls we get transition to the state 'lesson'. □

A sequential net can be described by a *state graph* such as in Figure X.16. The state graph shows the transitions and the outputs for different input signals. The sequential net can also be described by a truth table, which must include the states and also the conditions for transitions. The sequential net is thus described by the maps

$$\begin{aligned} \text{new state} &= f(\text{state, inputs}) \\ \text{output} &= g(\text{state, inputs}) \end{aligned}$$

Notice the similarity with the state equations for continuous time and discrete time systems. The difference is that the states, inputs, and outputs only can take a finite number of values. Sequential nets can be divided into *synchronous* and *asynchronous* nets. In synchronous nets a transition from one state to another is synchronized by a clock pulse, which is the case when the nets are implemented in a computer. A synchronous sequential net can be implemented as shown in Figure X.17. In asynchronous nets the system goes from one state to the other as soon as the conditions for transition are satisfied. The asynchronous nets are more sensitive to the timing when the inputs are changing. In the sequel we will only discuss the synchronous nets.

There are many ways to describe sequences and sequential nets. A standard is now developing based on GRAFCET, developed in France. GRAFCET stands for "Graphe de Commande Etape-Transition" (Graph for Step-Transition Control). GRAFCET with minor modifications is passed as an IEC (International Electrotechnical Commission) standard, IEC 848. The way to describe sequential nets is called *function charts*. GRAFCET is a formalized way of describing sequences and functional specifications. This can be done without any consideration of how to make the hardware implementation. The functional specifications are easy to interpret and understand. Computerized aids to program and present sequences have also been developed.

Another way to describe sequences and parallel actions is *Petri nets*. Petri nets are directed graphs that can handle sequential as well as parallel sequences. Sometimes the formalism for Petri nets makes it possible to investigate for instance reachability. It is then possible to find out which states that can be reached by legitimate transitions. This knowledge can be used to test the logic and to implement alarms.

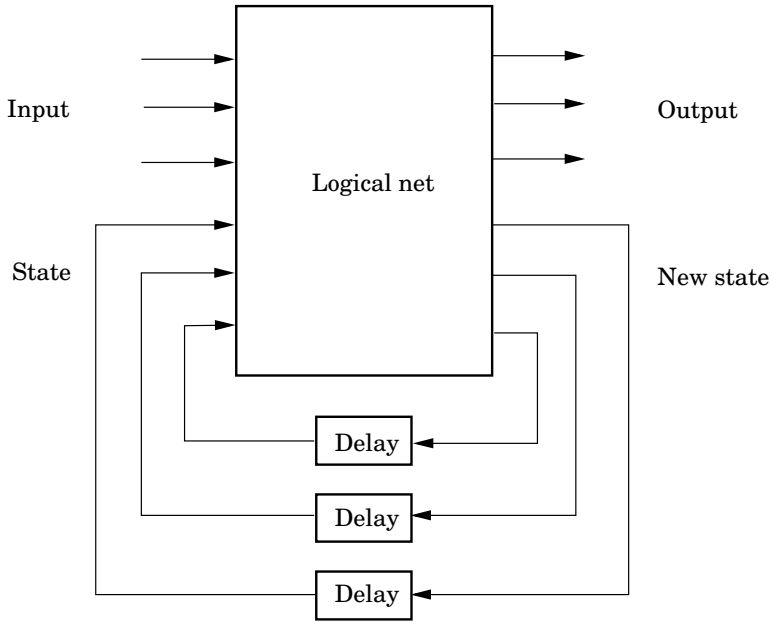


Figure X.17 Synchronous sequential net as combination of logical net and delay or memory elements.

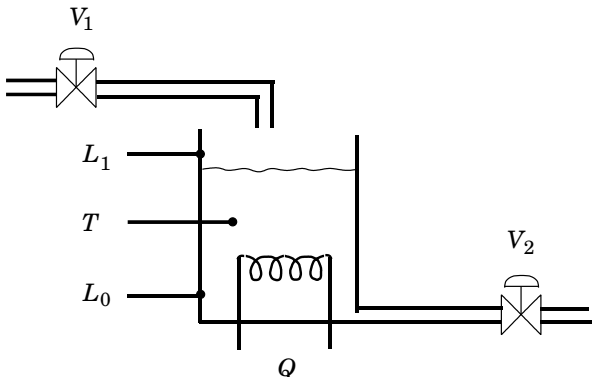


Figure X.18 Process for batch water heating.

X.8 GRAFCET

The objective of GRAFCET is to give a tool for modeling and specification of sequences. The main functions and properties of GRAFCET will be described in this section. A simple example is used to illustrate the concepts.

EXAMPLE X.9—HEATING OF WATER
Consider the process in Figure X.18. It consists of a water tank with two level indicators, a heater, and two valves. Assume that we want to perform the following sequence:

0. Start the sequence by pressing the button B. (Not shown in Figure X.18.)

1. Fill water by opening the valve V_1 until the upper level L_1 is reached.

2. Heat the water until the temperature is greater than T. The heating can start as soon as the water is above the level L_0.

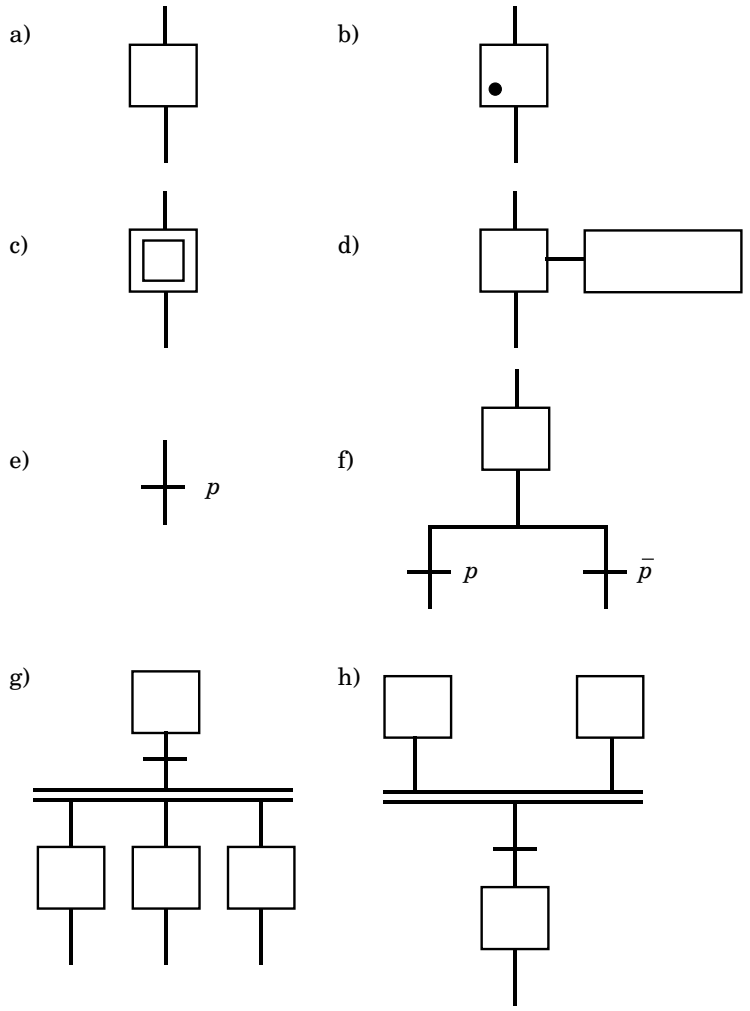


Figure X.19 GRAFCET symbols. (a) Step (inactive); (b) Step (active); (c) Initial step; (d) Step with action; (e) Transition; (f) Branching with mutually exclusive alternatives; (g) Branching into parallel paths; (h) Synchronization.

3. Empty the water by opening the valve V_2 until the lower level L_0 is reached.

4. Close the valves and go to Step 0 and wait for a new sequence to start.

From the natural language description we find that there is a sequence of waiting, filling, heating, and emptying. Also notice that the filling and heating must be done in parallel and then synchronized, since we don't know which will be finished first. □

GRAFCET Specifications

A function chart in GRAFCET consists of *steps* and *transitions*. A step corresponds to a state and can be *inactive*, *active*, or *initial*. See Figure X.19(a)–(c). Actions associated with a step can also be indicated, see Figure X.19(d). A transition is denoted by a bar and a condition when the transition can take place, see Figure X.19(e). A step is followed by a transition, branching with mutually exclusive alternatives, or branching into parallel sequences. Parallel sequences can be synchronized, see Figure X.19(h). The synchronization takes place when all the preceding steps are active and when the transition condition is fulfilled. The function chart in GRAFCET for

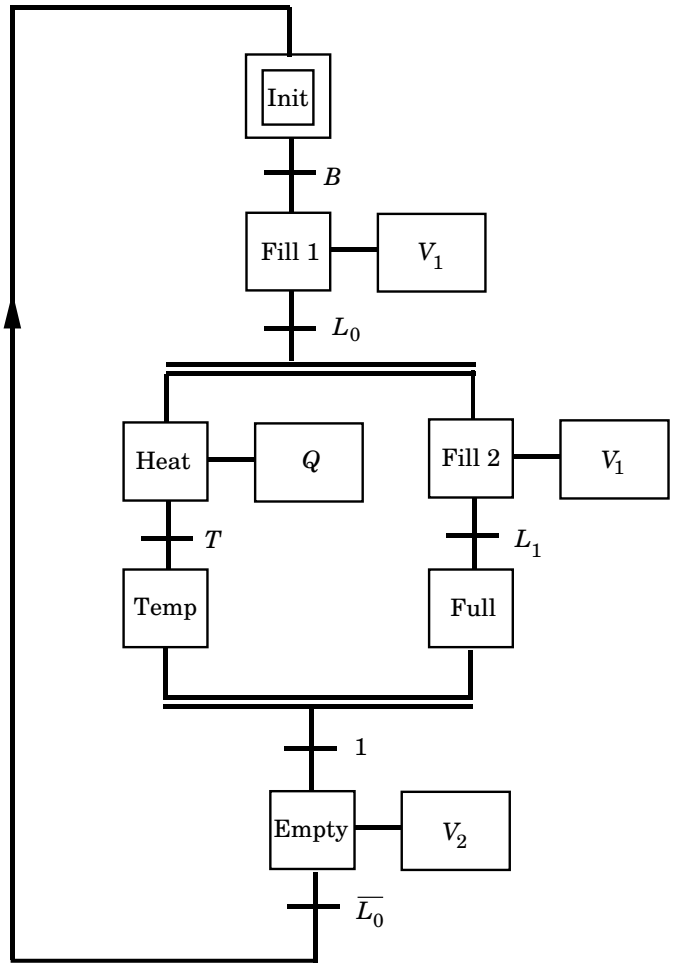


Figure X.20 GRAFCET for the process and sequences in Example X.9.

the process in Example X.9 is shown in Figure X.20. The sequence starts in the step Initial. When $B = 1$ we get a transition to Fill 1, where the valve V_1 is opened until the level L_0 is reached. Now two parallel sequences starts. First the heating starts and we get a transition to Temp when the correct temperature is reached. At this stage the other branch may be finished or not and we must wait for synchronization before the sequence can be continued. In the other branch the filling continues until level L_1 is reached. After the synchronization the tank is emptied until level L_0 is reached thereafter we go back to the initial state and wait for a new sequence to start.

The example can be elaborated in different ways. For instance, it may happen that the temperature is reached before the upper level is reached. The left branch is then in step Temp. The water may, however, become too cool before the tank is full. This situation can be taken into account making it possible to jump to the step Heat if the temperature is low. In many applications we need to separate between the normal situation, and emergency situations. In emergency situations the sequence should be stopped at a hazardous situation and started again when the hazard is removed. In simple sequential nets it can be possible to combine all these situations into a single function chart. To maintain simplicity and readability it is usually better to divide the system into several function charts.

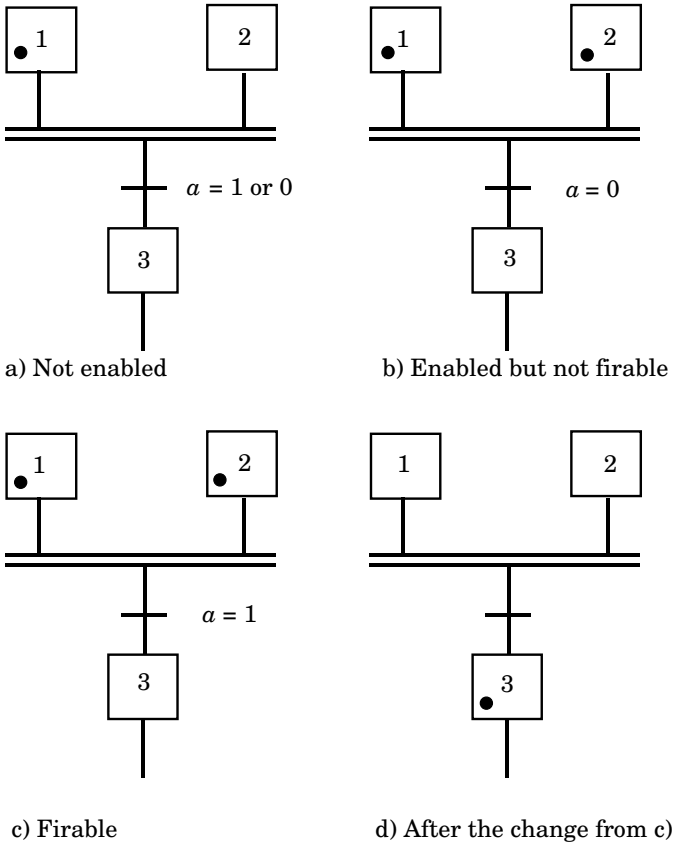


Figure X.21 Illustration of Rule 2.

GRAFCET Rules

To formalize the behavior of a function chart we need a set of rules how steps are activated and deactivated etc. We have the following rules:

Rule 1: The initialization defines the active step at the start.

Rule 2: A transition is *firable* if:

i: All steps preceding the transition are active (enabled).

ii: The corresponding receptivity (transition condition) is true.
A firable transition must be fired.

Rule 3: All the steps preceding the transition are deactivated and all the steps following the transition are activated.

Rule 4: All firable transitions are fired simultaneously.

Rule 5: When a step must be both deactivated and activated it remains active without interrupt.

For instance, Rule 2 is illustrated in Figure X.21. One way to facilitate the understanding of a functional specification is to introduce *macro steps*. The macro step can represent a new functional specification, see Figure X.22. The macro steps make it natural to use a top-down approach in the construction of a sequential procedure. The overall description is first broken down into macro steps and each macro step can then be expanded. This gives well structured programs and a clearer illustration of the function of a complex process.

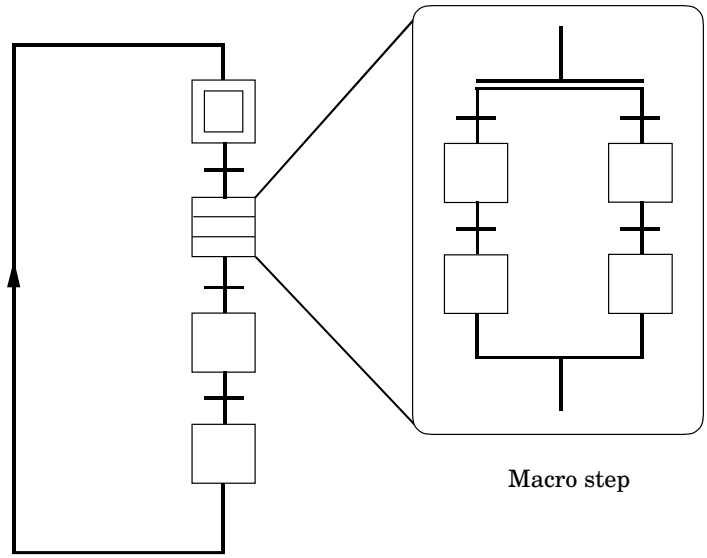


Figure X.22 Zooming of a macro step in GRAFCET.

Y. Control of Multivariable Processes

Y.1 Introduction

So far we have mainly discussed how to analyze simple control loops. I.e., control loops with one input and one output (SISO systems). Most chemical processes contain, however, many control loops. Several hundred control loops are common in larger processes. Fortunately, most of the loops can be designed from a single-input single-output

point of view. This is because there is no or weak interaction between the different parts of the process. Multiloop controllers, such as cascade control, were discussed in Chapter 6. In cascade control there are several measurements, but there is still only one input signal to the process.

In many practical cases it is necessary to consider several control loops and actuators at the same time. This is the case when there is an interaction between the different control loops in a process. A change in one input may influence several outputs in a complex way. If the couplings or interactions are strong it may be necessary to make the design of several loops at the same time. This leads to the concept of multi-input multi-output systems (MIMO systems). We will in this chapter generalize the multiloop control into *multivariable control*. The distinction is not well defined, but we will with multivariable control mean systems with two or more actuators and where all the control loops are design at the same time. A fundamental problem in multivariable control is how the different measurements should be used by the different actuators. There are many possible combinations. We have so far only discussed simple systems where the use of the measurements has been "obvious". The controllers have typically been PID controllers with simple modifications.

Multivariable systems will be introduced using a couple of examples.

EXAMPLE Y.1—SHOWER
A typical example of a coupled system is a shower. The system has two inputs, flow of hot and cold water, and two outputs, total flow and temperature. Changing either of the flows will change the total flow as well as the temperature. In this case the coupling is "strong" between the two input signals and the two output signals. In the daily life we have also seen that the control of flow and temperature in a shower can be considerably simplified by using a thermostat mixer. This will reduce the coupling and make it easier to make the control. □

EXAMPLE Y.2—LEVEL AND TEMPERATURE CONTROL IN A TANK
Consider the heated tank in Figure Y.1. The flow and temperature of the feed are the disturbances of the process. The level is controlled by the outlet valve. The temperature in the tank is controlled by the steam flow through the heating coil. A change in feed temperature T_i or the temperature setpoint T_r will change the steam flow, but this will not influence the level in the tank. A change in the feed flow F_i or the level setpoint L_r will change the output flow F and thus the content in the tank. This will

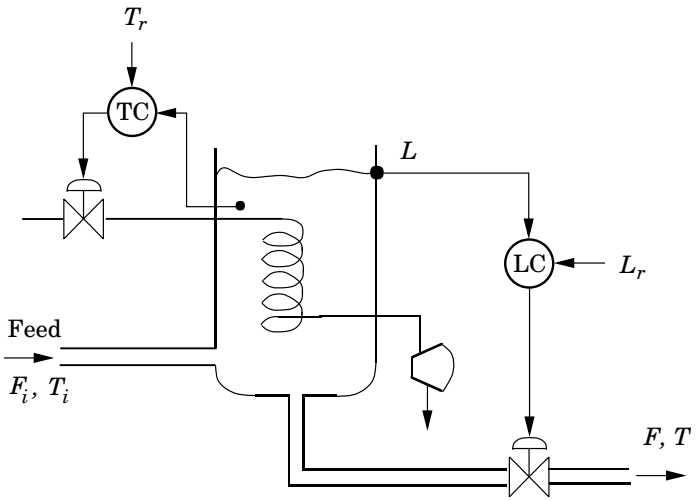


Figure Y.1 Level and temperature control in a tank.

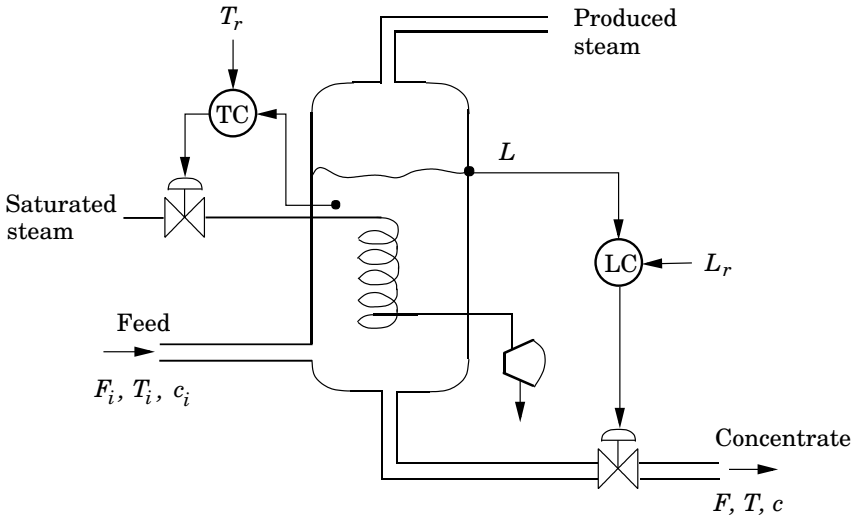


Figure Y.2 Level and temperature control in an evaporator.

also influence the temperature controller that has to adjust the steam flow. There is thus a coupling between the level control and the temperature control, but there is no coupling from the temperature loop to the level loop. □

EXAMPLE Y.3—LEVEL AND TEMPERATURE CONTROL IN AN EVAPORATOR

Consider the evaporator in Figure Y.2. In this process there is an interaction between the two loops. The temperature control loop will change the steam flow to the coil. This will influence both the produced steam and the level. In the same way a change in the output flow will change both the level and the temperature. In the evaporator there is an interaction between both loops. □

One-way interaction has in previous chapters been handled by using feedforward. In this chapter we will discuss how more complex interactions can be handled. Design of MIMO systems can be quite complex and is outside the main theme of this course. It is, however, of great importance to be able to judge if there is a strong coupling

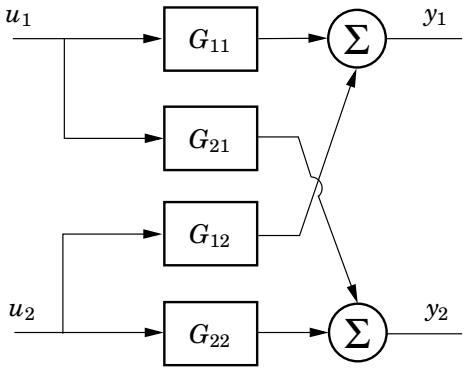


Figure Y.3 Coupled system with two inputs and two outputs.

between different parts of the process. Also it is important to have a method to pair input and output signals. In this chapter we will discuss three aspects of coupled systems:

- How to judge if the coupling in the process will cause problems in the control of the process?
- How to determine the pairing of the inputs and outputs in order to avoid the coupling?
- How to eliminate the coupling in the process?

An example of a coupled system is given in Figure Y.3. The system has two input signals and two output signals. Let G_{ij} be the transfer function from input j to output i and introduce the vector notations for the Laplace transforms of the outputs and inputs

$$Y(s) = \begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} \qquad U(s) = \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix}$$

then

$$Y(s) = \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix} U(s) = G(s)U(s)$$

$G(s)$ is called the *transfer function matrix* of the system. This representation can be generalized to more inputs and outputs. Another way of describing a multivariable system is by using the state space model

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

where the input u and output y now are vectors.

Y.2 Stability and Interaction

To understand some of the problems with coupled systems we will discuss the process in Figure Y.4. This can be a block diagram of the evaporator in Figure Y.2. The process

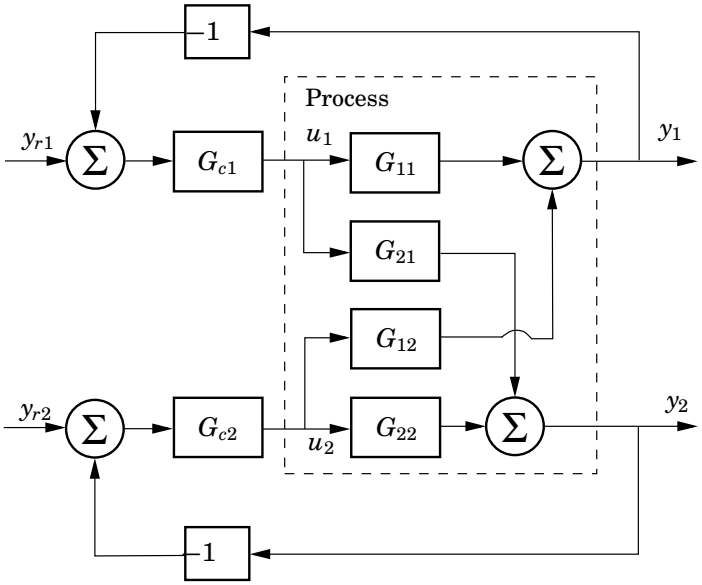


Figure Y.4 Block diagram of a system with two inputs and outputs. The system is controlled by two simple controllers G_{c1} and G_{c2}.

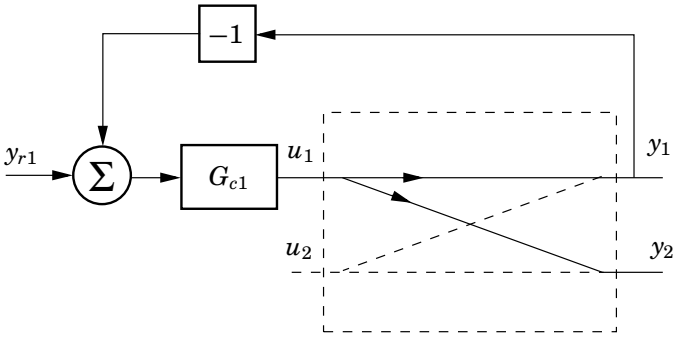


Figure Y.5 The system in Figure Y.4 when only the first loop is closed.

is described by

$$\begin{aligned} Y_1(s) &= G_{11}U_1(s) + G_{12}U_2(s) \\ Y_2(s) &= G_{21}U_1(s) + G_{22}U_2(s) \end{aligned} \tag{Y.1}$$

and the controllers by

$$\begin{aligned} U_1(s) &= G_{c1}\,(Y_{r1}(s) - Y_1(s)) \qquad \text{(Loop 1)} \\ U_2(s) &= G_{c2}\,(Y_{r2}(s) - Y_2(s)) \qquad \text{(Loop 2)} \end{aligned} \tag{Y.2}$$

The situation with only Loop 1 closed is shown in Figure Y.5. The closed loop system is now described by

$$Y_1 = \frac{G_{11}G_{c1}}{1 + G_{11}G_{c1}}Y_{r1}$$

$$Y_2 = G_{21}U_1 = \frac{G_{21}G_{c1}}{1 + G_{11}G_{c1}}Y_{r1}$$

With obvious changes we can also write down the expressions when Loop 2 is closed

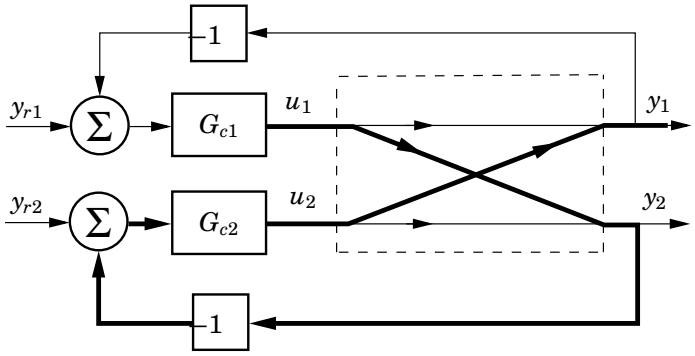


Figure Y.6 Illustration of the influence of u_1 on y_1 through the second loop.

and Loop 1 open. The characteristic equations for the two cases are

$$1 + G_{11}G_{c1} = 0 \tag{Y.3}$$

and

$$1 + G_{22}G_{c2} = 0 \tag{Y.4}$$

Now consider the case in Figure Y.4 with both loops closed. Using (Y.2) to eliminate U_1 and U_2 from (Y.1) gives

$$\begin{aligned}(1 + G_{11}G_{c1})Y_1 + G_{12}G_{c2}Y_2 &= G_{11}G_{c1}Y_{r1} + G_{12}G_{c2}Y_{r2}\\ G_{21}G_{c1}Y_1 + (1 + G_{22}G_{c2})Y_2 &= G_{21}G_{c1}Y_{r1} + G_{22}G_{c2}Y_{r2}\end{aligned}$$

Using Cramer's rule to solve for Y_1 and Y_2 gives

$$\begin{aligned}Y_1 &= \frac{G_{11}G_{c1} + G_{c1}G_{c2}(G_{11}G_{22} - G_{12}G_{21})}{A}Y_{r1} + \frac{G_{12}G_{c2}}{A}Y_{r2}\\ Y_2 &= \frac{G_{21}G_{c1}}{A}Y_{r1} + \frac{G_{22}G_{c2} + G_{c1}G_{c2}(G_{11}G_{22} - G_{12}G_{21})}{A}Y_{r2}\end{aligned}$$

where the denominator is

$$A(s) = (1 + G_{11}G_{c1})(1 + G_{22}G_{c2}) - G_{12}G_{21}G_{c1}G_{c2} \tag{Y.5}$$

If $G_{12} = G_{21} = 0$ then there is no interaction and the closed loop system is described by

$$\begin{aligned}Y_1 &= \frac{G_{11}G_{c1}}{1 + G_{11}G_{c1}}Y_{1r}\\ Y_2 &= \frac{G_{22}G_{c2}}{1 + G_{22}G_{c2}}Y_{2r}\end{aligned}$$

The closed loop system is in this case stable if each loop is stable, i.e. if all the roots of (Y.3) and (Y.4) are in the left half plane. With interaction the stability of the closed loop system is determined by the polynomial $A(s)$ in (Y.5). Notice that the stability of the closed loop system depends on both the controllers and the four blocks in G. The interaction through the second loop is illustrated in Figure Y.6. The bold lines indicate the influence of u_1 on y_1 through the second loop.

The controllers must be tuned such that the total system is stable. Since any of the loops may be switched into manual control, i.e. the loop is open, it is also necessary

that each loop separately is stable. The closed loop system may be unstable due to the interaction even if each loop separately is stable. The interaction in the system may destabilize the closed loop system and this makes it more difficult to do the tuning. The following example points out some of the difficulties of tuning multivariable controllers.

EXAMPLE Y.4—STABILITY OF MULTIVARIABLE SYSTEM
Assume that the process in Figure Y.4 has the transfer functions

$$G_{11} = \frac{1}{0.1s+1} \qquad G_{12} = \frac{5}{s+1}$$
$$G_{21} = \frac{1}{0.5s+1} \qquad G_{22} = \frac{2}{0.4s+1}$$

and the proportional controllers $G_{c1} = K_{c1}$ and $G_{c2} = K_{c2}$. Tuning each loop separately each loop will be stable for any positive gain. The characteristic equation for the total system is

$$\begin{aligned} &0.02s^4 + 0.1(3.1 + 2K_{c1} + K_{c2})s^3 \\ &\quad + (1.29 + 1.1K_{c1} + 1.3K_{c2} + 0.8K_{c1}K_{c2})s^2 \\ &\quad + (2 + 1.9K_{c1} + 3.2K_{c2} + 0.5K_{c1}K_{c2})s \\ &\quad + (1 + K_{c1} + 2K_{c2} - 3K_{c2}K_{c2}) = 0 \end{aligned}$$

A *necessary* condition for stability is that all coefficients are positive. The only one that may become negative is the last one. This implies that a necessary condition for stability of the closed loop system is

$$1 + K_{c1} + 2K_{c2} - 3K_{c1}K_{c2} > 0$$

This condition is violated when the gains are sufficiently high. □

Generalization

The computations above were done for a 2×2 system with simple proportional controllers. We will now consider a system with a general transfer function matrix and a general controller matrix. Let the system be described by

$$Y(s) = G_o(s)U(s)$$

and the controller by

$$U(s) = G_r(s)\left(Y_r(s) - Y(s)\right)$$

Both G_o and G_r are transfer function matrices of appropriate orders. Eliminating $U(s)$ gives

$$Y(s) = G_o(s)G_r(s)(Y_r(s) - Y(s))$$

Solving for $Y(s)$ gives

$$Y(s) = (I + G_o(s)G_r(s))^{-1} G_o(s)G_r(s)Y_r(s) = G_{cl}(s)Y_r(s) \tag{Y.6}$$

where I is the unit matrix. The closed loop transfer function matrix is $G_{cl}(s)$. When evaluating the inverse in (Y.6) we get the denominator

$$A(s) = \det\left(I + G_o(s)G_r(s)\right)$$

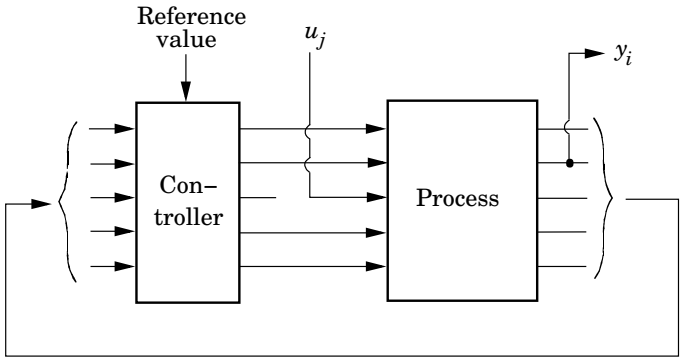


Figure Y.7 A controlled multivariable system.

The characteristic equation of the closed loop system is thus

$$A(s) = \det\left(I + G_o(s)G_r(s)\right) = 0 \tag{Y.7}$$

This is in analogy with the SISO case.

The closed loop system is thus stable if the roots of (Y.7) are in the left hand plane.

Y.3 Relative Gain Array

There are many ways to determine the coupling or interaction in a MIMO system. The *Relative Gain Array* (RGA) was introduced by Bristol in 1966 as a way to determine the *static* coupling in a system. The RGA can be used to find an appropriate pairing of inputs and outputs. The drawback of RGA is that only the coupling for reference value changes is investigated. In process control it is often more important to consider load disturbances. The Relative Disturbance Gain (RDG) introduced by McAvoy is one way to change the focus to the disturbances. The idea behind the RGA is to study the stationary gain $K = G(0)$ of the process and to introduce a normalization. The normalization is necessary to eliminate the effect of scaling of the variables in the process.

Consider Figure Y.7, which is a multivariable system with the same number of inputs as outputs. All inputs except u_j are controlled. We will now investigate the stationary changes in the outputs y_i, when u_j is changed. I.e. we will study the stationary value of

$$\frac{\Delta y_i}{\Delta u_j}$$

where the Δ denotes the change in the variable. Introduce

$$k_{ij} = G_{ij}(0) = \frac{\Delta y_i}{\Delta u_j} \quad \text{when} \quad \Delta u_k = 0 \quad k \neq j$$

The steady state gain $K = G(0)$ is a matrix with the elements k_{ij}.

To obtain a normalization we also study a second situation. Consider the situation in Figure Y.7. Assume that the control of the system is so good that only y_i is changed when u_j is changed. Introduce

$$l_{ij} = \frac{\Delta y_i}{\Delta u_j} \quad \text{when} \quad \Delta y_k = 0 \quad k \neq i$$

The coupling through the static gain $G(0)$ is easy to obtain, when the transfer function matrix is known. The normalization through l_{ij} is more difficult to determine. We will derive the normalization for a system with two inputs and outputs and then give the general expression.

EXAMPLE Y.5—THE NORMALIZATION FOR A SYSTEM WITH TWO INPUTS AND OUTPUTS
Assume that the steady state behavior of the system is described by

$$\begin{aligned}\Delta y_1 &= k_{11}\Delta u_1 + k_{12}\Delta u_2\\ \Delta y_2 &= k_{21}\Delta u_1 + k_{22}\Delta u_2\end{aligned}$$

We first determine l_{21}. In this case $\Delta y_1 = 0$, which gives

$$\Delta u_2 = -\frac{k_{11}}{k_{12}}\Delta u_1$$

Using this in the second equation above gives

$$\Delta y_2 = \left(k_{21} - \frac{k_{22}k_{11}}{k_{12}}\right)\Delta u_1$$

or

$$l_{21} = \frac{\Delta y_2}{\Delta u_1} = -\frac{k_{11}k_{22} - k_{12}k_{21}}{k_{12}} = -\frac{\det K}{k_{12}}$$

where $\det K$ is the determinant of the matrix with the elements k_{ij}. In the same way we can determine

$$\begin{aligned}l_{22} &= \frac{\det K}{k_{11}}\\ l_{11} &= \frac{\det K}{k_{22}}\\ l_{12} &= -\frac{\det K}{k_{21}}\end{aligned}$$

The matrix with the elements $1/l_{ij}$ is thus given by

$$\frac{1}{\det K}\begin{pmatrix} k_{22} & -k_{21}\\ -k_{12} & k_{11}\end{pmatrix} = \left(K^{-1}\right)^T$$

The relative gain array (RGA) is now defined as a matrix Λ with the elements

$$\lambda_{ij} = \frac{k_{ij}}{l_{ij}}$$

□

The example above can be generalized. The *relative gain array* is defined as

$$\Lambda = K \;.*\left(K^{-1}\right)^T \tag{Y.8}$$

where $.*$ denotes the *element-by-element* (Schur) product of the elements in K and $(K^{-1})^T$. Note that it is not a conventional matrix multiplication that is performed.

From (Y.8) it follows that it is sufficient to determine the stationary gain $K = G(0)$ of the open loop system. The relative gain array matrix Λ is the given through (Y.8) and an element by element multiplication of two matrices. The RGA has the properties

$$\sum_{i=1}^{n} \lambda_{ij} = \sum_{j=1}^{n} \lambda_{ij} = 1$$

I.e. the row and the column sums are equal to unity. This implies that for a 2×2 system only one element has to be computed. The rest of the elements are uniquely determined. For a 3×3 system four elements are needed to determine the full matrix.

A system is easy to control using single-loop controllers if Λ is a unit matrix or at least diagonal dominant after possible permutations of rows and/or columns.

EXAMPLE Y.6—NON-INTERACTING SYSTEM
Assume that

$$K = \begin{pmatrix} 0 & a \\ b & c \end{pmatrix}$$

then

$$\left(K^{-1}\right)^T = -\frac{1}{ab} \begin{pmatrix} c & -b \\ -a & 0 \end{pmatrix}$$

and

$$\Lambda = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

By interchanging the rows or the columns we get a unit matrix. The system is thus easy to control using two non-interacting controllers. □

A system has difficult couplings if Λ has elements that are larger than 1. This implies that some elements must be negative since the row and column sums must be unity.

Pairing of Inputs and Outputs

By determining the relative gain array Λ is is possible to solve the first problem stated in the Section Y.1. The RGA matrix can also be used to solve the second problem. I.e. it can be used to pair inputs and outputs. *The inputs and outputs should be paired so that the corresponding relative gains are positive and as close to one as possible.*

EXAMPLE Y.7—PAIRING 1
Let the RGA be

$$\Lambda = \begin{pmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{pmatrix}$$

The RGA and the pairing for different values of λ are shown in Table Y.1. The interaction when $\lambda = 2$ is severe and the system will be difficult to control with two single loops. □

EXAMPLE Y.8—PAIRING 2
Assume that a system has the transfer function matrix

$$G(s) = \begin{pmatrix} \dfrac{1}{s+1} & \dfrac{2}{s+3} \\ \dfrac{1}{s+1} & \dfrac{1}{s+1} \end{pmatrix}$$

Table Y.1 The RGA for a 2×2 system.

λ	Λ	Pairing	Remark
1	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{matrix} u_1 - y_1 \\ u_2 - y_2 \end{matrix}$	No interaction
0	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{matrix} u_1 - y_2 \\ u_2 - y_1 \end{matrix}$	No interaction
0.85	$\begin{pmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{pmatrix}$	$\begin{matrix} u_1 - y_1 \\ u_2 - y_2 \end{matrix}$	Weak interaction
2	$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$	$\begin{matrix} u_1 - y_1 \\ u_2 - y_2 \end{matrix}$	Difficult interaction

The static gain is given by

$$K = \begin{pmatrix} 1 & 2/3 \\ 1 & 1 \end{pmatrix}$$

and we get

$$\left(K^{-1}\right)^T = \begin{pmatrix} 3 & -3 \\ -2 & 3 \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} 3 & -2 \\ -2 & 3 \end{pmatrix}$$

Since Λ has elements that are larger than one we can expect difficulties when controlling the system using single-input single-output controllers. □

EXAMPLE Y.9—PAIRING 3
Assume that a system has the transfer function matrix

$$G(s) = \frac{1}{(s+1)(s+2)} \begin{pmatrix} s-1 & s \\ -6 & s-2 \end{pmatrix}$$

The static gain is given by

$$K = \begin{pmatrix} -0.5 & 0 \\ -3 & -1 \end{pmatrix}$$

and we get

$$\left(K^{-1}\right)^T = \begin{pmatrix} -2 & 6 \\ 0 & -1 \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

This system should be possible to control using two simple controllers, since Λ is a unit matrix. □

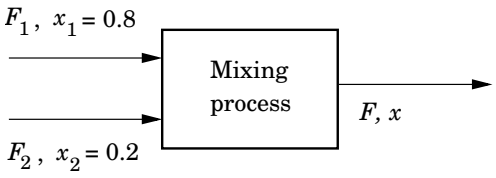


Figure Y.8 Mixing process, where total flow and mixture should be controlled.

EXAMPLE Y.10—MIXING PROCESS
Consider the mixing process in Figure Y.8. The purpose is to control the total flow F and the composition x at the outlet. The inputs are the flows F_1 and F_2. The desired equilibrium point is $F = 200$ and $x = 0.6$. The input compositions are $x_1 = 0.8$ and $x_2 = 0.2$. Which inputs should be used to control F and x respectively?

The mass balances give

$$F = F_1 + F_2$$
$$Fx = F_1x_1 + F_2x_2$$

Solving for the unknown flows give $F_1 = 133.33$ and $F_2 = 66.67$. The system is nonlinear and we can't directly determine the gain matrix at the desired equilibrium point. One way is to calculate the RGA using perturbation. Assume that F_1 is changed one unit and assume that F_2 is kept constant. This gives $F = 201$ and $x = 0.6009$ and

$$\left(\frac{\Delta F}{\Delta F_1}\right)_{F_2} = \frac{1}{1} = 1$$

In the same way we change F_1 by one unit but is keeping x constant. This gives

$$\left(\frac{\Delta F}{\Delta F_1}\right)_{x} = \frac{1.50}{1} = 1.50$$

Using the definition of the RGA we get

$$\lambda_{11} = \frac{\left(\frac{\Delta F}{\Delta F_1}\right)_{F_2}}{\left(\frac{\Delta F}{\Delta F_1}\right)_{x}} = \frac{1}{1.50} = \frac{2}{3}$$

and the full relative gain matrix becomes

$$\Lambda = \begin{pmatrix} 0.69 & 0.31 \\ 0.31 & 0.69 \end{pmatrix}$$

where we have assumed that the inputs are in the order F_1, F_2 and the outputs F, x. The interaction will be minimized if we choose the pairing F–F_1 and x–F_2. There will, however, be a noticeable coupling since the elements are close in size. □

The relative gain array method only considers the static coupling in a system. There may very well be difficult dynamic couplings in the process that is not detected through the RGA methods. Different ways to circumvent this problem have been discussed in the literature. The Relative Dynamic Array (RDA) method is one way to also consider the dynamic coupling in some frequency ranges.

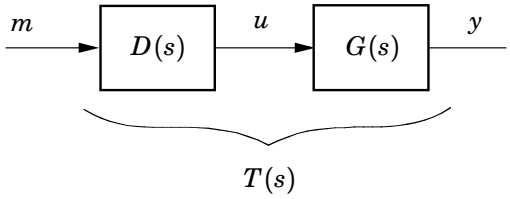


Figure Y.9 A system with decoupling matrix D.

Y.4 Decoupling

In this section we will give a short discussion of how to improve the control of a multivariable system with coupling. One way is to use theory for design of multivariable systems. This is, however, outside the scope of this course. A second way that can be effective is to introduce decoupling in a MIMO system. This is done by introducing new signals that are static or dynamic combinations of the original control signals. After the introduction of a decoupling matrix it can be possible to design the controllers from a single-input single-output point of view. Consider the system in Figure Y.9. The matrix $D(s)$ is a transfer function matrix that will be used to give decoupling by introducing new control signals $m(t)$. We have

$$Y(s) = G(s)U(s) = G(s)D(s)M(s) = T(s)M(s)$$

The idea is to determine $D(s)$ such that the total transfer function matrix $T(s) = G(s)D(s)$ has a decoupled structure. Ideally $T(s)$ should be a diagonal matrix. By specifying T we may solve for D provided the matrix G is invertible. We will make the computations for a system with two inputs and two outputs. Assume that T and G are given as

$$T = \begin{pmatrix} T_{11} & 0 \\ 0 & T_{22} \end{pmatrix} \quad G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}$$

The decoupling matrix is now given by

$$D = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} = \frac{1}{\det G} \begin{pmatrix} G_{22}T_{11} & -G_{12}T_{22} \\ -G_{21}T_{11} & G_{11}T_{22} \end{pmatrix} \tag{Y.9}$$

Equation (Y.9) can give quite complicated expressions for D. *One* choice is obtained by considering the diagonal elements D_{11} and D_{22} as parts of the controller and interpret D_{12} and D_{21} as feedforward terms in the controller. We may then choose

$$D = \begin{pmatrix} 1 & -G_{12}/G_{11} \\ -G_{21}/G_{22} & 1 \end{pmatrix}$$

This gives

$$T = GD = \begin{pmatrix} G_{11} - G_{12}G_{21}/G_{22} & 0 \\ 0 & G_{22} - G_{21}G_{12}/G_{11} \end{pmatrix}$$

With this special choice we have obtained a complete decoupling of the system. There may, however, be difficulties to make a realization of the decoupling. There may be pure derivations in D.

An other solution to the decoupling problem is obtained by the choice

$$D(0) = G^{-1}(0)T(0)$$

This results in a decoupling in steady state.

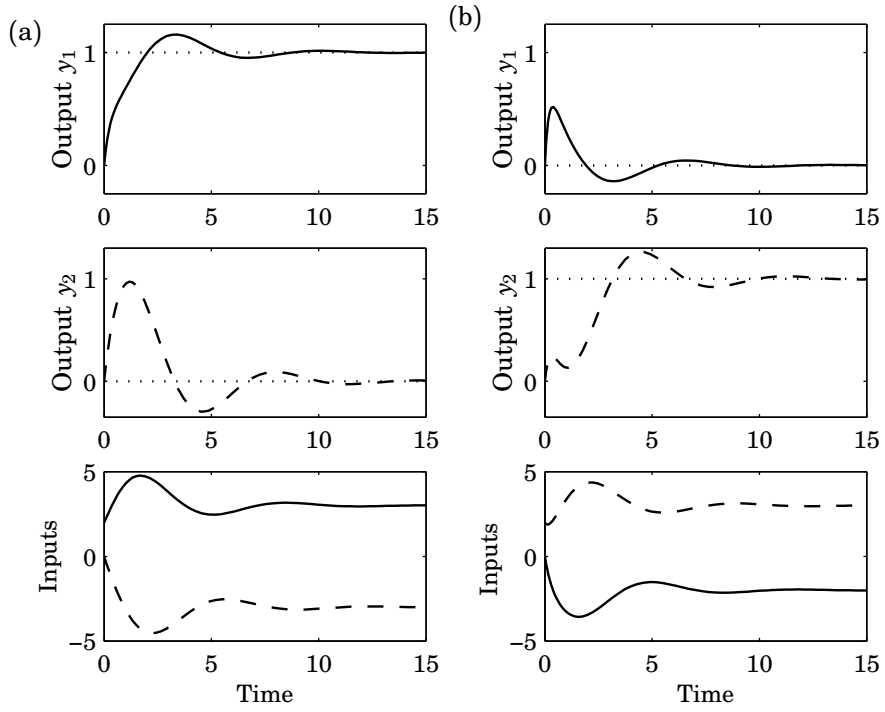


Figure Y.10 The process in Example Y.8 controlled by two PI controllers. (a) y_1, y_2 and the control signals u_1 and u_2, when the reference value of y_1 is changed. (b) Same as (a) when the reference value of y_2 is changed.

EXAMPLE Y.11—DECOUPLING

Consider the system in Example Y.8. Figure Y.10 shows the outputs and control signals when the system is controlled by two PI controllers without any decoupling. The output y_2 changes much, when the reference value to y_1 is changed. Figure Y.11 shows the same experiment as in the previous figure, but when a stationary decoupling is done using

$$D = \begin{pmatrix} 1 & -\frac{2}{3} \\ -1 & 1 \end{pmatrix}$$

This gives

$$T(s) = \begin{pmatrix} \dfrac{1-s}{(s+1)(s+3)} & \dfrac{4s}{3(s+1)(s+3)} \\ 0 & \dfrac{1}{3(s+1)} \end{pmatrix}$$

The RGA matrix of the decoupled system is

$$\Lambda = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The system should now be possible to control using two separate controllers. This is also verified in the simulation shown in Figure Y.11. There is, however, still a dynamic coupling between the two loops. A complete dynamic decoupling is shown in Figure Y.12. In this case we use

$$D = \begin{pmatrix} 1 & -2\dfrac{s+1}{s+3} \\ -1 & 1 \end{pmatrix}$$

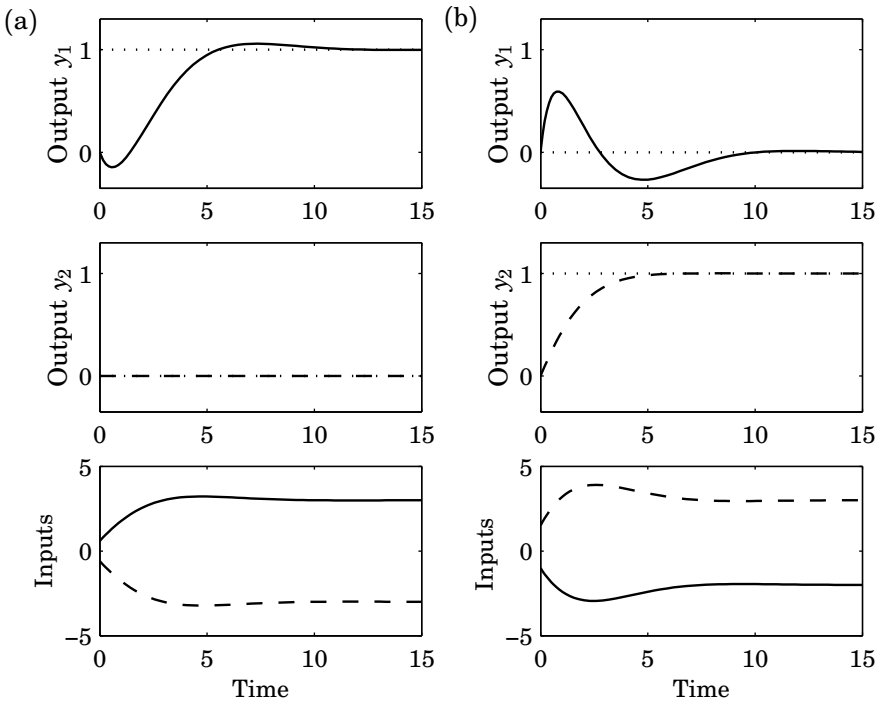


Figure Y.11 Same as Figure Y.10 but with stationary decoupling.

which gives

$$T(s) = \begin{pmatrix} \dfrac{1-s}{(s+1)(s+3)} & 0 \\ 0 & \dfrac{1-s}{(s+1)(s+3)} \end{pmatrix}$$

□

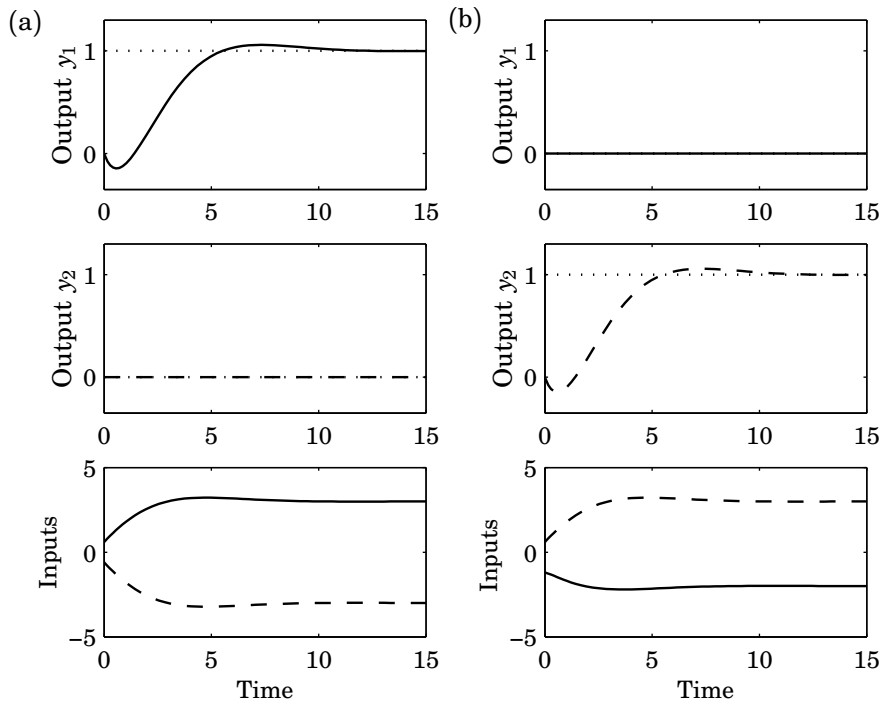


Figure Y.12 Same as Figure Y.11 but with dynamic decoupling.

Dictionary

A
actual value ärvärde
actuator ställdon, aktuator
A-D converter A-D-omvandlare
adaptive control adaptiv reglering
aliasing vikning
amplitude function amplitudfunktion
amplitude margin amplitudmarginal
argument function argumentfunktion
asymptotic stability asymptotisk stabilitet
asynchronous net asynkronnät

B
backward difference bakåtdifferens
bandwidth bandbredd
basis bas
batch process satsvis process
bias avvikelse, nollpunktsförskjutning
block diagram blockdiagram, blockschema
block-diagram algebra blockdiagramalgebra
Bode plot Bodediagram
Boolean algebra Boolesk algebra
break frequency brytfrekvens
bumpless transfer stötfri övergång

C
canonical form kanonisk form
cascade control kaskadreglering
causality kausalitet
characteristic equation karakteristisk ekvation
characteristic polynomial karateristiskt polynom
chattering knatter
closed-loop system slutet system
combinatory network kombinatoriskt nät
complementary sensitivity function komplementär känslighetsfunktion
computer control datorstyrning
computer-controlled system datorstyrt system
control styrning, reglering
control error reglerfel
control signal styrsignal
controllability styrbarhet
controller regulator
controller gain regulatorförstärkning
controller structure regulatorstruktur
coupled systems kopplade system
cross-over frequency skärfrekvens

D
D-A converter D-A omvandlare
damped frequency dämpad frekvens
damping dämpning
dead time dödtid
decade dekad
decoupling särkoppling
delay tidsfördröjning, dödtid
delay margin dödtidsmarginal
derivative term derivataterm
derivative time derivatatid
determinant determinant
difference equation differensekvation
differential equation differentialekvation
differential operator differentialoperator
direct-digital control direkt digital styrning
discrete-time system tidsdiskret system
disturbance störning
dynamic relation dynamiskt samband
dynamic system dynamiskt system

E
eigenvalue egenvärde
eigenvector egenvektor
equilibrium jämvikt
error fel
error signal felsignal

F
feedback återkoppling
feedback control reglering, sluten styrning
feedback system återkopplat system
feedforward framkoppling
filtering filtrering
final-value theorem slutvärdessatsen
floating control flytande reglering
forward difference framåtdifferens
free system fritt system
frequency analysis frekvensanalys
frequency-domain model frekvensdomänmodell
frequency function frekvensfunktion
function chart funktionsschema
fundamental matrix fundamentalmatrix

G
gain förstärkning
gain function förstärkningsfunktion
gain margin amplitudmarginal
gain scheduling parameterstyrning

H
high-frequency asymptote högfrekvensasymptot
hold circuit hållkrets
homogeneous system homogent system
hysterises hysteres

I
identification identifiering

implementation implementering, förverkligande
impulse impuls
impulse response impulssvar
initial value initialvärde, begynnelsevärde
initial-value theorem begynnelsevärdessatsen
input insignal
input-output model insignal-utsignal-modell
instability instabilitet
integral control integrerande reglering
integral term integralterm
integral time integraltid
integrating controller integrerande regulator
integrator integrator
integrator windup integratoruppvridning
interaction växelverkan
interlock förregling
internal model intern modell
inverse invers
inverse-response system system med omvänt svar

L
lag compensation fasretarderande kompensering
Laplace transform Laplace-transform
lead compensation fasavancerade kompensering
limiter begränsare, mättningsfunktion
linear dependent linjärt beroende
linear indedendent linjärt oberoende
linear-quadratic controller linjärkvadratisk regulator
linear system linjärt system
linerarization linjärisering
load disturbance belastningsstörning
logical control logikstyrning
logical expression logiskt uttryck
loop gain kretsförstärkning
low-frequency asymptote lågfrekvensasymptot

M
manipulated variable styrd variabel
manual control manuell styrning
marginal stability marginell stabilitet
matrix matris
measurement noise mätbrus
measurement signal mätsignal
microcontroller styrkrets
mid-range control mitthållningsreglering
minimum-phase system minimumfassystem
mode mod
model-based control modellbaserad reglering
modeling modellering, modellbygge
multi-capacitive process flerkapacitiv process
multiplicity multiplicitet
multivariable system flervariabelt system

N
natural frequency naturlig frekvens
noise brus
nonlinear coupling olinjär koppling
nonlinear system olinjärt system
non-minimum-phase system icke-minimumfassystem
Nyquist plot Nyquistdiagram
the Nyquist stability theorem Nyquists stabilitetssats

O
observability observerbarhet
observer observerare
on-off control till-från-reglering, tvålägesreglering
open-loop control öppen styrning
open-loop system öppet system
operational amplifier operationsförstärkare
operator guide operatörshjälp
ordinary differential equation ordinär differentialekvation
oscillation svängning
output utsignal, ärvärde
overshoot översläng

P
paralell connection parallellkoppling
Petri net Petrinät
phase function fasfunktion
phase margin fasmarginal
PI control PI-reglering
PID control PID-reglering
PLC programmerbart logiksystem, styrdator
pole pol
pole placement polplacering
pole-zero plot pol-nollställe-diagram, singularitetsdiagram
polynomial polynom
position algorithm positionsalgoritm
practical stability praktisk stabilitet
prediction prediktion
prediction horizon prediktionshorisont
prediction time prediktionstid
prefiltering förfiltrering
process process
proportional band proportionalband
proportional gain proportionell förstärkning
proportional control proportionell reglering

pulse-transfer function pulsöverförings-funktion

Q

quantization kvantisering

R

ramp ramp
ramp response rampsvar
rank rang
ratio control kvotreglering
reachability uppnåelighet
real-time programming realtidsprogram-mering
recursive equation rekursiv ekvation
reference value referensvärde
relative damping relativ dämpning
relative gain array relativa förstärknings-matrisen
relay relä
reset time integraltid
reset windup integratoruppvridning
return difference återföringsdifferens
return ratio återföringskvot
rise time stigtid
robustness robusthet
root locus rotort
Routh's algorithm Rouths algoritm

S

sampled-data system samplat system
sampler samplare
sampling sampling
sampling frequency samplingsfrekvens
sampling interval samplingsintervall
sampling period samplingsperiod
saturation mättning
selector väljare
sensitivity känslighet
sensitivity function känslighetsfunktion
sensor mätgivare, sensor
sequential control sekvensreglering
sequential net sekvensnät
series connection seriekoppling
setpoint börvärde
setpoint control börvärdesreglering
settling time lösningstid
shift operator skiftoperator
single-capacitive process enkelkapacitiv process
single-loop controller enloopsregulator
singular value singulärt värde
singularity diagram singularitetsdiagram
sinusoidal sinusformad
smoothing utjämning
solution time lösningstid
split-range control uppdelat utstyrning-sområde
stability stabilitet
stability criteria stabilitetskriterium
state tillstånd
state graph tillståndsgraf
state-space model tillståndsmodell
state-transition matrix tillståndsöverförings-matris
state variable tillståndsvariabel
static gain statisk förstärkning
static system statiskt system
stationary error stationärt fel
steady state jämvikt
steady-state error stationärt fel
steady-state gain stationär förstärkning
steady-state value stationärvärde
step steg
step response stegsvar
superposition principle superpositionsprincipen
synchronous net synkonnät
synthesis syntes, dimensionering
system matrix systemmatris

T

time constant tidskonstant
time delay tidsfördröjning
time-domain approach tidsdomänmetod
time-invariant system tidsinvariant sys-tem
trace spår
tracking följning
transfer function överföringsfunktion
transient transient
transient analysis transientanalys
translation principle translationsprincip
transpose transponat
truth table sanningstabell
tuning inställning

U

ultimate-sensitivity method självsvängn-ingsmetoden
undamped frequency naturlig frekvens
unit element enhetselement
unit step enhetssteg
unmodeled dynamics omodellerad dynamik
unstable system instabilt system

V

vector vektor
velocity algorithm hastighetsalgoritm

W

weigting function viktfunktion
windup uppvridning

Z

zero nollställe
zero element nollelement
zero-order hold nollte ordningens hållkrets

Index