1

Mathematical and Control Theory Background

1.1 Introduction

This chapter will review some mathematical and control theory background, some of which is actually assumed covered by previous control courses. Both the coverage of topics and their presentation will therefore lack some detail, as the presentation is aiming

- to provide sufficient background knowledge for readers with little exposure to control theory,
- to correct what is this author's impression of what are the most common misconceptions
- to establish some basic concepts and introduce some notation.

1.2 Models for Dynamical Systems

Many different model representations are used for dynamical systems, and a few of the more common ones will be introduced here.

1.2.1 Dynamical Systems in Continuous Time

A rather general way of representing a dynamical system in continuous time is via a set of ordinary differential equations:

$$\dot{x} = f(x, u, d) \tag{1.1}$$

where the variables x are termed as the *system states* and $\dot{x} = \frac{dx}{dt}$ is the time derivative of the state. The variables u and d are both external variables that affect the system. In the context of control, it is common to distinguish between the *manipulated variables* or *(control) inputs* u that can be manipulated by a controller, and the *disturbances* d that are external variables that affect the system but which cannot be set by the controller.

The system states x are generally only a set of variables that are used to describe the system's behavior over time. Whether the individual components of the state

Advanced Chemical Process Control: Putting Theory into Practice, First Edition. Morten Hovd.

vector can be assigned any particular physical interpretation will depend on how the model is derived. For models derived from fundamental physical and chemical relationships (often termed as "rigorous models"), the states will often be quantities like temperatures, concentrations, and velocities. If, in contrast, the model is an empirical model identified from observed data, it will often not be possible to assign any particular interpretation to the states.

Along with the state equation (1.1), one typically also needs a *measurement equation* such as:

$$y = g(x, u, d) \tag{1.2}$$

where the vector y is a vector of *system outputs*, which often correspond to available *physical measurements* from the systems. Control design is usually at its most simple when all states can be measured, i.e. when $y = x$.

Disturbances need not be included in all control problems. If no disturbances are included in the problem formulation, Eqs. (1.1) and (1.2) trivially simplify to $\dot{x} = f(x, u)$ and $y = g(x, u)$, respectively.

Since we are dealing with *dynamical* systems, it is hopefully obvious that the variables x, y, u, d may all vary with time t. In this section, time is considered as a continuous variable, in accordance with our usual notion of time.

Together, Eqs. (1.1) and (1.2) define a system model in continuous time. This type of model is rather general and can deal with any system where it suffices to consider system properties at specific points in space, or where it is acceptable to average/lump system properties over space. Such models where properties are averaged over space are often called *lumped models*.

For some applications, it may be necessary to consider also *spatial distribution* of properties. Rigorous modeling of such systems typically result with a set of partial differential equations (instead of the ordinary differential equations of (1.1)). In addition to derivatives with respect to time, such models also contain derivatives with respect to one or more spatial dimensions. Models described by partial differential equations will not be considered any further here. Although control design based on partial differential equations is an active research area, the more common industrial practice is to convert the set of partial differential equations to a (larger) set of ordinary differential equations through some sort of spatial *discretization*.

1.2.2 Dynamical Systems in Discrete Time

Although time in the "real world" as we know it is a continuous variable, control systems are typically implemented in computer systems, which cyclically execute a set of instructions. Measurements and control actions are therefore executed at discrete points in time, and to describe system progression from one time instant to subsequent instants we will need a discrete-time model. Such models may be represented as:

$$x_{k+1} = f(x_k, u_k, d_k) \tag{1.3}$$

$$y_k = g(x_k, u_k, d_k) \tag{1.4}$$

where x_k, y_k, u_k, and d_k are the discrete-time counterparts to the system states, outputs, inputs, and disturbances introduced above for continuous-time systems, and the subscript (k) identify the timestep (or sampling instant). Thus, x_k is the state x at timestep k, while x_{k+1} is the state at the subsequent timestep. Note that although the same letter f is used to represent the system dynamics for both continuous- and discrete-time systems, these functions will be different for the two different model types. The measurement equation, however, will often be identical for the two model types.

1.2.3 Linear Models and Linearization

Many control design methods are based on *linear* models. It is therefore necessary to be able to convert from a nonlinear model to a linear model which is (hopefully) a close approximation to the nonlinear model. This is called linearization of the nonlinear model.

A systems is linear if the functions f and g (in (1.1) and (1.2) for the case of continuous-time models, or in (1.3) and (1.4) for the case of discrete-time models) are linear in all the variables x, u, and d. Thus, a linear continuous-time model may be expressed as:

$$\dot{x} = Ax + Bu + Ed \tag{1.5}$$

$$y = Cx + Du + Fd \tag{1.6}$$

where A, B, C, D, E, F are matrices of appropriate dimensions, and the matrix elements are independent of the values of x, u, d. Linear models for discrete-time systems follow similarly.

Linearization is based on the Taylor series expansion of a function. Consider a function $h(a)$. We want to approximate the value of $h(a)$ in the vicinity of $a = a^*$. The Taylor series expansion then provides the approximation:

$$h(a) = h(a^* + \delta a) \approx h(a^*) + \left.\frac{\partial h}{\partial a}\right|_{a^*} \delta a + \frac{1}{2}\delta a^T \left.\frac{\partial^2 h}{\partial a^2}\right|_{a=a^*} \delta a + \cdots \tag{1.7}$$

where the notation $|_{a=a^*}$ indicates that the value $a = a^*$ is used when evaluating the derivatives.

1.2.3.1 Linearization at a Given Point

When linearizing a dynamical system model, we terminate the Taylor series expansion after the first-order term. The underlying nonlinear system is therefore naturally assumed to be continuous and have continuous first-order derivatives. Assume that the linearization is performed at the point:

$$a = \begin{bmatrix} x \\ u \\ d \end{bmatrix} = \begin{bmatrix} x^* \\ u^* \\ d^* \end{bmatrix} = a^* \tag{1.8}$$

The terminated Taylor series expansion of (1.1) then becomes

$$\frac{dx}{dt} = \frac{d\delta x}{dt} \approx f(a^*) + \left.\frac{\partial f}{\partial x}\right|_{a=a^*} \delta x + \left.\frac{\partial f}{\partial u}\right|_{a=a^*} \delta u + \left.\frac{\partial f}{\partial d}\right|_{a=a^*} \delta d \tag{1.9}$$

Similarly, we get for (1.2)

$$y = y^* + \delta y \approx g(a^*) + \left.\frac{\partial g}{\partial x}\right|_{a=a^*} \delta x + \left.\frac{\partial g}{\partial u}\right|_{a=a^*} \delta u \left.\frac{\partial g}{\partial d}\right|_{a=a^*} \delta d \tag{1.10}$$

where it is understood that $y^* = g(a^*)$.

Next, define $A = \left.\frac{\partial f}{\partial x}\right|_{a=a^*}$, $B = \left.\frac{\partial f}{\partial u}\right|_{a=a^*}$, $E = \left.\frac{\partial f}{\partial d}\right|_{a=a^*}$, $C = \left.\frac{\partial g}{\partial x}\right|_{a=a^*}$, $D = \left.\frac{\partial g}{\partial u}\right|_{a=a^*}$, $F = \left.\frac{\partial d}{\partial x}\right|_{a=a^*}$

Linearizing at an Equilibrium Point The point a^* used in the linearization is usually an equilibrium point. This means that

$$f(a^*) = 0 \tag{1.11}$$

$$g(a^*) = y^* \tag{1.12}$$

Thus, we get

$$\frac{dx}{dt} = A\delta x + B\delta u + E\delta d \tag{1.13}$$

$$\delta y = C\delta x + D\delta u + F\delta d \tag{1.14}$$

Linearizing a discrete-time model is done in the same way as for continuous-time models. The only slight difference to keep in mind is that for a discrete-time model at steady state $x_{k+1} = x_k$, and therefore $f(a^*) = x_k$ when linearizing at a steady state.

Deviation Variables It is common to express the system variables (x, u, d, and y) in terms of their *deviation* from the linearization point a^*. When doing so the δ's are typically suppressed for ease of notation – *as will be done in the remainder of this book*. It is, however, important to beware that when converting from deviation variables to "real" variables, the linearization point has to be accounted for.

To illustrate: A model for a chemical reactor is linearized at steady-state conditions corresponding to a reactor temperature of 435 K. If the linearized model, expressed in deviation variables, indicates a temperature of −1, the corresponding "real" temperature would be 434 K.

Linear Controllers Are Not Linear! It appears that many students, even after introductory control courses, do not appreciate that *our so-called "linear" controllers are only linear when expressed in deviation variables.* In "natural" variables, the typical "linear" controller is in fact *affine*, i.e. they have a constant term in addition to the linear term. This can lead to many frustrations, until the misunderstanding has been clarified – which might actually take some time, because the importance of this issue will depend on both controller structure and controller type. Consider a simple feedback loop, with a (linear) controller K controlling a system G, as illustrated in Figure 1.1.

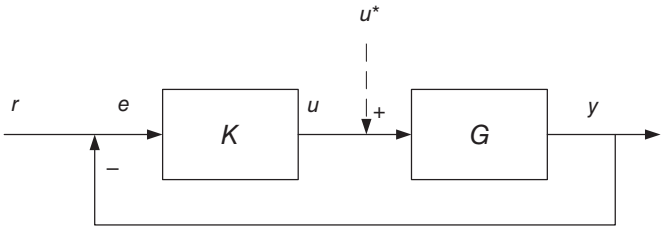

Figure 1.1 A simple feedback loop with a one-degree-of-freedom controller and possible "output bias."

This type of controller is called a "one-degree-of-freedom controller," since it has only one input, the control offset $e = r - y$. We can make the following observations:

- Clearly, it does not matter whether the reference r and measurement y are expressed in "physical" variables or deviation variables, as long as the same scale is used for both. This is because the controller input is the difference between these two variables.
- Consider the case when the controller K is a pure proportional controller, i.e. $u = K(r - y)$ with K constant. It is then necessary to add u^* as an "output bias"[1] to the controller output, as indicated by the dashed arrow in the figure.
- Consider next the case when the controller K contains integral action. In this case, the "output bias" is not strictly necessary, since the value of the integrating state will adjust for this when the system reaches steady state. However, an output bias may improve transient response significantly when putting the controller into operation.[2]

Consider next a loop where the controller has separate entry port for the reference and the measurement, as shown in Figure 1.2. This type of controller is used

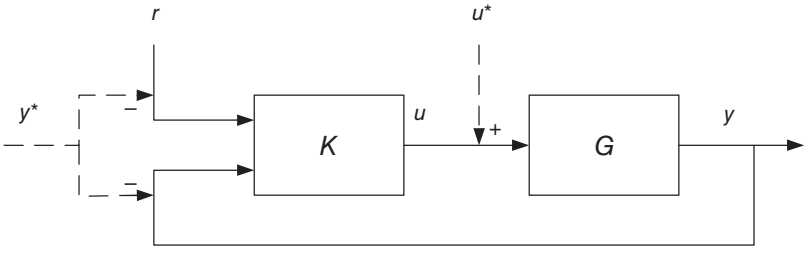

Figure 1.2 A simple feedback loop with a two-degree-of-freedom controller and possible "bias" on both controller inputs and controller output.

1 Some system vendors may use different terminology.
2 See also Section 6.4 on Bumpless Transfer.

when one wants to treat the measurement and reference signals differently in the controller. We note that

- In this case, we need to subtract the value of the measurement at the linearization point, y^*, from both the reference and the measurement.
- Whether to add u^* to the controller output is determined by the same considerations as for the one-degree-of-freedom controller.

1.2.3.2 Linearizing Around a Trajectory

It was noted above that it is most common to linearize around a steady state. However, in some cases, one may want to linearize around a trajectory, i.e. around a series of consistent future values of x, u, and d. This most commonly occurs in nonlinear model predictive control (MPC). Each time an MPC controller executes, it solves an optimization problem that optimizes system behavior over a "prediction horizon." However, for some strongly nonlinear problems, using the same linearized model for the entire prediction horizon may not give sufficient accuracy. In such cases, one may choose to linearize around a trajectory instead.

Given the present state, a prediction of the future manipulated variables (typically obtained from the previous execution of the MPC), and predicted values for future disturbances, the nonlinear model can be used to simulate the system in the future. This gives predicted future states that are consistent with the present state and the predicted future manipulated variables and disturbances.

For each timestep in the future, the linearization is performed around the predicted state, manipulated variable, and disturbance values. This will give different matrices A, B, CD, E, F for each timestep. In this way, a nonlinear system is approximated by a linear, *time-varying* model.

Linearizing around a trajectory clearly complicates the model. In addition to the added complexity of having to ensure that the right model matrices are used at the right timestep in the future, one also has to remember that the linearization point varies from timestep to timestep (resulting from $f(a^*) \neq x_k$ in the discrete-time equivalent of (1.9)). This adds additional complexity when converting between physical variables and deviation variables.

1.2.4 Converting Between Continuous- and Discrete-Time Models

It will often be necessary to convert from continuous- to discrete-time models (and less frequently necessary to convert the other way). Process models based on first principles modeling will typically result in continuous-time models. Often, control design is performed with a continuous-time model. The continuous-time controller is thereafter converted to a discrete-time controller for implementation in a computer. There are also controller types that are more conveniently designed using discrete-time models. The most notable example of such controllers are the so-called MPC controllers which will be described in some detail later in the book.

To convert from continuous to discrete time, we need to

- choose a numerical integration method for the system dynamics, and
- determine (assume) how the external variables (u and d) change *between* the time instants for the discrete-time model.

It is common to assume so-called "zero-order hold,"[3] i.e. that the external variables are constant at the value of the previous time instant until the next time instant is reached. This agrees with what is common practice for control inputs in control systems.

Most control design software will have functions for converting between continuous- and discrete-time linear models. It is also included in most basic control textbooks. We will nevertheless give a short introduction here, primarily in order to discuss the handling of time delay when converting from a continuous to a discrete-time model. The presentation is inspired by that of Åström and Wittenmark [1].

Consider a continuous-time linear model:

$$\dot{x} = A_c x(t) + B_c u(t) \tag{1.15}$$

Assuming zero-order hold and a timestep of length h, integration over one timestep (from $t = kh$ to $t = kh + h$) gives

$$x(kh + h) = e^{A_c h} x(kh) + \int_{kh}^{kh+h} e^{A_c(kh+h-r)} B_c u(r) dr \tag{1.16}$$

This is commonly expressed as the discrete-time model:

$$x_{k+1} = A_d x_k + B_d u_k \tag{1.17}$$

where the sampling interval h is assumed known and therefore not explicitly stated.[4] The matrices A_d and B_d are given by:

$$A_d = e^{A_c h}$$
$$B_d = \int_{kh}^{kh+h} e^{A_c(kh+h-r)} B_c u(r) dr = A_c^{-1} \left(e^{A_c h} - I \right) B_c$$

1.2.4.1 Time Delay in the Manipulated Variables

Consider next the case when the manipulated variables u do not affect the state derivative $\dot{x}$ directly, but only after a time delay τ. The model (1.15) thus becomes

$$\dot{x} = A_c x(t) + B_c u(t - \tau) \tag{1.18}$$

Note that there is no exact representation of a pure time delay using ordinary differential equations – this would require an infinite number of states. Therefore, the time delay is instead introduced explicitly in the argument when representing the manipulated variable u as a function of time. While there is an extensive literature on how to account for time delays in continuous-time systems, this will not be covered here. Instead, we note that

---

3 An nth *order hold* means that the nth time derivative is held constant between the sample instants of the discrete-time model.

4 Note also that the subscript d refers to *discrete time* rather than "disturbance." Elsewhere in this note B_d is sometimes used as "the B-matrix for the disturbance."

- Time delays in linear continuous-time systems are easily handled in the frequency domain. For a system such as (1.18) with a measurement model (1.6), the transfer function from input u to output y is simply given by:

$$y(s) = G(s)e^{-\tau s}u(s)$$

where $G(s) = C(sI - A_c)^{-1}B_c + D$ is the transfer function for the delay-free system.[5]

- Whenever a state-space model is needed, e.g. for controller synthesis, a low-order approximation of the time delay is typically used. The most common such approximation is the *Padé approximation*. The nth order Padé approximation is given by:

$$e^{-\tau s} \approx \frac{\left(1 - \frac{\tau}{2n}s\right)^n}{\left(1 + \frac{\tau}{2n}s\right)^n}$$

A state-space model for the right-hand side of the approximation above can easily be found using, e.g. the Control Systems Toolbox in Matlab. Often, the second- or third-order approximation will suffice.

Multiple Timestep Time Delays If the time delay is an integer number of sampling intervals, this is easily captured in a discrete-time model. Let $u_\Delta(k) = u(k-n)$. This can be expressed as:

$$\begin{aligned} x_\Delta(k+1) &= A_\Delta x_\Delta(k) + B_\Delta u(k) \\ &= \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \vdots & 0 \\ 0 & \vdots & \vdots & \vdots & 0 \\ 0 & \vdots & \vdots & 0 & I \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} x_\Delta(k) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{bmatrix} u(k) \\ u_\Delta(k) &= C_\Delta x_\Delta(k) = \left[\underbrace{I\ 0\ 0\ \cdots\ 0}_{n} \right] x_\Delta(k) \end{aligned} \tag{1.19}$$

The overall model then results from the series interconnection of the delay-free model and the model for the time delay above.

Fractional Timestep Time Delays If the time delay τ is only a fraction of the sampling interval h, we must account for the fact that the value of the manipulated variable which affects $\dot{x}$ in (1.15) from time kh to time $kh + \tau$ is actually $u(kh - h)$. Thus, the integral in (1.16) must be split in two, and we get

5 The reader is expected to have some knowledge of transfer functions and frequency responses – but for those for which this is unfamiliar, a brief introduction will be given shortly.

$$
\begin{aligned}
x(kh+h) &= e^{A_c h}x(kh) + \int_{kh}^{kh+\tau} e^{A_c(kh+h-r)}B_c dr u(kh-h) \\
&\quad + \int_{kh+\tau}^{kh+h} e^{A_c(kh+h-r)}B_c dr u(kh) \\
&= A_d x(kh) + B_{d0}u(kh) + B_{d1}u(kh-h) \qquad (1.20)\\
B_{d1} &= e^{A_c(h-\tau)}A_c^{-1}\left[e^{A_c\tau} - I\right]B_c = e^{A_c(h-\tau)}\int_0^{\tau} e^{A_c r}dr B_c \\
B_{d0} &= A_c^{-1}\left[e^{A_c(h-\tau)} - I\right]B_c = \int_0^{h-\tau} e^{A_c r}dr B_c
\end{aligned}
$$

This can be expressed in state-space form as:

$$
\begin{bmatrix} x(kh+h) \\ u(kh) \end{bmatrix} = \begin{bmatrix} A_d & B_{d1} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x(kh) \\ u(kh-h) \end{bmatrix} + \begin{bmatrix} B_{d0} \\ I \end{bmatrix} u(kh) \qquad (1.21)
$$

For time delays lasting more than one timestep, but a non-integer number of timesteps, the overall model is found by the series interconnection of the multiple timestep delay model in (1.19) and the system dynamics + fractional timestep delay model in (1.21).

Some modern control techniques like MPC are computationally intensive and may induce a computational time delay. If the computational time is significant compared to the sampling interval, it may be necessary to include a fractional time delay in the model even for plants that by themselves have no time delay.

1.2.4.2 Time Delay in the Measurements

Time delays in measurements may occur both due to the characteristics of the sensor equipment (e.g. delays in analyzers such as online gas chromatographs) or due to transportation delays (long pipes or conveyor belts from the plant to the sensor).

For linear, time-invariant systems, it does not matter whether the time delay is modeled at the input or the output of the plant. However, for multivariable systems, the time delay may be different for different measurements. In such cases, the time delay must be modeled at the output, since it cannot be moved to the input.

Also, a measurement is often dependent on multiple states. The number of discrete-time states used to model the time delay can then be reduced by delaying the measurement in the model instead of delaying the states and calculating the measurement from the delayed states [2].

Time delays in the measurements can be handled in much the same way as that explained above for time delay in the manipulated variables. The details are therefore left to the reader.

1.2.5 Laplace Transform

The Laplace transform should be familiar to all readers from introductory control courses, and no attempt is made here at providing a complete or self-contained introduction to the topic. It is merely introduced here as a minimal introduction to its use later in this book.

Restating first the linear(ized) ordinary differential equation model, we have

$$\dot{x} = Ax + Bu + Ed \tag{1.22}$$

$$y = Cx + Du + Fd \tag{1.23}$$

where the δ's are suppressed for notational simplicity. We should nevertheless keep in mind that the linear model is expressed in deviation variables. The model described by (1.22) and (1.23) is called a (linear) *state-space model* of a system.

Using standard rules for the Laplace transformation (available in standard undergraduate mathematics textbooks), we have

$$sx(s) + x(t = 0) = Ax(s) + Bu(s) + Ed(s) \tag{1.24}$$

$$y(s) = Cx(s) + Du(s) + Fd(s) \tag{1.25}$$

where s is a complex-valued scalar. The effect of the initial conditions (the term $x(t = 0)$ above) is usually ignored, since stability and common measures of performance do not depend on initial conditions (for linear systems). Nevertheless, one should be aware that the initial response will depend on initial conditions. If the closed-loop system contains modes that are poorly damped, the effects of the initial conditions may be felt for a significant time.

Ignoring the term involving the initial conditions (or assuming the initial conditions equal to zero in deviation variables), we obtain by simple manipulations:

$$\begin{aligned} y(s) &= \left[C(sI - A)^{-1}B + D\right] u(s) + \left[C(sI - A)^{-1}E + F\right] d(s) \\ &= G(s)u(s) + G_d(s)d(s) \end{aligned} \tag{1.26}$$

where $G(s)$ and $G_d(s)$ are the (monovariable or multivariable) transfer functions from the manipulated variable and the disturbance, respectively, to the system output.

1.2.6 The *z* Transform

Whereas the Laplace transform using the Laplace variable s is used for transfer functions for continuous-time systems, the *forward shift operator* z is used to define transfer functions for discrete-time systems. The forward shift operator shifts a time series one step forward in time, i.e.

$$w_{k+1} = zw_k \tag{1.27}$$

For the discrete-time linear system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ed_k \\ y_k &= Cx_k + Du_k + Fd_k \end{aligned} \tag{1.28}$$

Noting that $x_{k+1} = zx_k$, we arrive at the discrete-time transfer functions:

$$\begin{aligned} y(z) &= \left[C(zI - A)^{-1}B + D\right] u(z) + \left[C(zI - A)^{-1}E + F\right] d(z) \\ &= G(z)u(z) + G_d(z)d(z) \end{aligned} \tag{1.29}$$

When writing out discrete-time transfer functions, it is common to use z^{-1} (known as the *backward shift operator*) when expressing how the present value of a variable depends on previous values of the same variable and/or previous inputs.

1.2.7 Similarity Transformations

Whereas the transfer function is unique for a given input–output behavior, there is an infinite number of different state-space models that describe the same dynamics.

Given a state-space model such as (1.22) and (1.23) and consider the case where we instead of the original states x want to use the alternative states $\tilde{x}$. The state vectors x and $\tilde{x}$ must be related through

$$x = T\tilde{x} \tag{1.30}$$

where T is an invertible matrix. This ensures that when specifying the state in one set of state variables, we also uniquely specify the states in the other set of state variables. Trivial manipulations then yield

$$\dot{\tilde{x}} = T^{-1}AT\tilde{x} + T^{-1}Bu + T^{-1}Ed \tag{1.31}$$

$$y = CT\tilde{x} + Du + Fd \tag{1.32}$$

from which the state-space matrices for the transformed state-space model are easily identifiable. This reveals the fact that the state-space representation of a dynamical system is not unique – via similarity transforms the exact same dynamics can be represented by "different" state-space models. In addition, a state-space model may contain "redundant" states, as discussed next. In contrast, the frequency response of a model in the Laplace domain (such as (1.26)) is unique. Furthermore, the transfer function model $G(s)$ itself is unique, provided any redundant states have been removed, i.e. provided cancelation of common terms in the numerator and denominator has been performed, or it is obtained from the Laplace transformation of a *minimal* model.

1.2.8 Minimal Representation

A state-space model may contain states that either cannot be affected by the inputs (an uncontrollable state) or cannot affect any of the outputs of the system (an unobservable state). Such states do not contribute to the input–output behavior of the system. The model then contains more states than the minimal number of states

required to represent the input–output behavior of the system. Therefore, such models are called non-minimal.

Many control calculations assume that the model supplied is minimal, and numerical problems may occur if this is not the case. It is therefore common practice to remove uncontrollable or unobservable states, and standard control software have functions for doing this (such as *minreal* in Matlab).

However, one should bear in mind that the uncontrollable or unobservable system states may represent important quantities for the overall system. Whether it is advisable to remove uncontrollable or unobservable states can depend on several factors:

- How was the model obtained? If the model is the result of rigorous modeling based on physical and chemical principles, the states will typically represent physical/chemical quantities in the system.
- Empirical models identified from experiments will typically result in models containing only observable and controllable states – although not all states need to be recognizable as a distinct physical quantity in the system.
- When assembling a system model from models of parts of the system, states representing the same physical quantity may be represented in several of the smaller models. This can easily lead to a non-minimal model when assembling the overall system model. Such "duplicate states" can safely be removed.
- It is usually considered safe to delete *stable* uncontrollable and unobservable modes.
  1. If a stable mode is uncontrollable, its effect on the output will die out over time – unless it is excited by some disturbance. A state may be "controllable" from a disturbance even if it is uncontrollable from the manipulated variables. This is the situation in many disturbance attenuation problems. Although such states may be removed from the plant model (from manipulated to controlled variables), it cannot be removed from the disturbance model (from disturbances to controlled variables).
  2. A controllable but unobservable mode will be excited by the manipulated variables, and even if it is stable will not necessarily decay to zero if the state is continuously excited by the manipulated variables or disturbances. If the state represents some quantity of little importance, this situation would appear acceptable. It may, however, be the case that the state represents some important quantity, and the fact that it is unobservable merely reflects an inappropriate set of measurements.

When discovering unobservable or uncontrollable states, the engineer should therefore reflect on how and why these states are introduced in the model. It may be that such states can safely be removed from the model. It may also be the case that one should install new measurements or new actuators to make the states observable and controllable.

Mini-tutorial 1.1 Illustrating the importance of minimal models

Consider the simple level control problem as shown in Figure 1.3. The output of the level controller is the reference for the flow control loop. With the flow control loop being fast and accurate, a simple model of the level of the tank can be obtained by simply integrating the flow in minus the flow out, giving the simple model:

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} -k_u \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ k_d \end{bmatrix} d$$

$$y = \begin{bmatrix} 1 & 1 \end{bmatrix} x$$

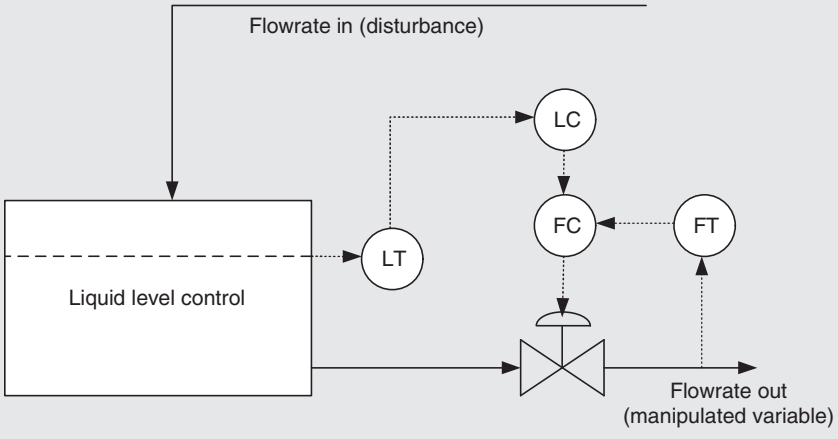

Figure 1.3 Simple level control problem.

With appropriate values of the constants k_u and k_d, this model does capture the input–output dynamics of the system. However, it is clearly the case that only one state is controllable from the input – and the uncontrollable state is integrating – meaning that it will need to be stabilized by feedback. To make matters worse, the individual states are not observable – only the sum of the states is. At first glance, it might therefore appear that this system cannot be stabilized by feedback control.

However, note that there is only one place where mass (in this case liquid) is accumulated in the system – and hence a single state should suffice to model the system. This leads to the model:

$$\dot{x} = 0x + \begin{bmatrix} -k_u & k_d \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix}$$

$$y = x$$

This one-state model describes exactly the same input–output behavior as the two-state model above! Furthermore, the integrating state is both controllable and

(Continued)

Mini-tutorial 1.1 (Continued)

observable, making feedback control possible. The (erroneous) conclusion above about the inability to stabilize by feedback was thus due only to inappropriate modeling.

In practice, to be able to control the system, the range of manipulation for the outlet flow must be at least as large as the range of variation in the inlet flowrate – but this is related to the constraints of the system and does not come out in a linear analysis.

For *diagonalizable* systems, i.e. systems for which the A-matrix has a full rank eigenvector matrix, it is straightforward to perform a similarity transform to identify the uncontrollable or unobservable states. Let M be the eigenvector matrix of the matrix A in (1.22) and Λ be the corresponding (diagonal) eigenvalue matrix. Choosing $T = M^{-1}$ in (1.30), then yields

$$\dot{\tilde{x}} = \Lambda\tilde{x} + MBu + MEd \tag{1.33}$$

$$y = CM^{-1}\tilde{x} + Du + Fd \tag{1.34}$$

Uncontrollable states (in terms of the states $\tilde{x}$) can then be identified from rows that are equal to zero in MB, whereas unobservable states are identified from columns in CM^{-1} equal to zero.

1.2.9 Scaling

An appropriate scaling of inputs and outputs will greatly simplify the interpretation of many of the analyses described in this book. For the system

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

we will assume throughout the book that:

- $y(s)$ is scaled such that the largest acceptable deviation from the reference value is equal to 1 in the scaled variable. If the largest acceptable deviation from the reference value is different in the positive and negative direction, the smaller of the two (in magnitude) is used.
- $u(s)$ is scaled such that the value 1 (in the scaled variable) corresponds to the largest available input value. If the largest available $u(s)$ is different in the positive and negative direction, the smaller of the two (in magnitude) is used.
- $d(s)$ is scaled such that the value of 1 (in the scaled variable) corresponds to the largest expected disturbance. If the largest available $d(s)$ is different in the positive and negative direction, the larger of the two (in magnitude) is used.

Note that the description above refers to $y(s), u(s)$, and $d(s)$ as *deviation variables*. The scaling is easily performed using diagonal matrices S_y, S_u, and S_d with positive elements along the diagonal. That is,

$$S_y = \text{diag}\{s_{yi}\} \tag{1.35}$$

where s_{yi} is the largest acceptable deviation from the reference value for output i. The matrices S_u and S_d are defined similarly. Using the subscript s to denote the scaled variable, we then get

$$\begin{aligned} S_y y_s(s) &= G(s)S_u u_s(s) + G_d(s)S_d d_s(s) \\ &\Updownarrow \\ y_s(s) &= S_y^{-1}G(s)S_u u_s(s) + S_y^{-1}G_d S_d d_s(s) \end{aligned}$$

where the scaled $G(s)$ is easily identifiable as $S_y^{-1}G(s)S_u$ and the scaled $G_d(s)$ as $S_y^{-1}G_d S_d$. Unless otherwise stated, we will throughout this book assume that the transfer function matrices $G(s)$ and $G_d(s)$ have been thus scaled, and we will not use the subscript s on the input and output variables (even though the variables are assumed to be scaled).

1.3 Analyzing Linear Dynamical Systems

1.3.1 Transfer Functions of Composite Systems

In this section, simple rules for finding transfer functions of composite systems will be provided, and thereafter some closed-loop transfer functions that will be defined that are frequently encountered in this book. The presentation in this section assumes all transfer functions to be multivariable, i.e. described by transfer function *matrices*. For monovariable systems, the transfer functions are scalar, which simplifies their calculation, since scalars do commute.

1.3.1.1 Series Interconnection

Consider the series interconnection of two transfer function matrices, as illustrated in Figure 1.4. The transfer function $L(s)$ from $r(s)$ to $y(s)$ can be found by starting at the output $y(s)$ and writing down the transfer function matrices as we trace the path back to the input $r(s)$. Thus, we find

$$y(s) = L(s)r(s) = G(s)K(s)r(s)$$

This technique is readily applied also to more than two transfer function matrices in series. We emphasize once again that the order of the transfer function matrices is important, $GK \neq KG$.

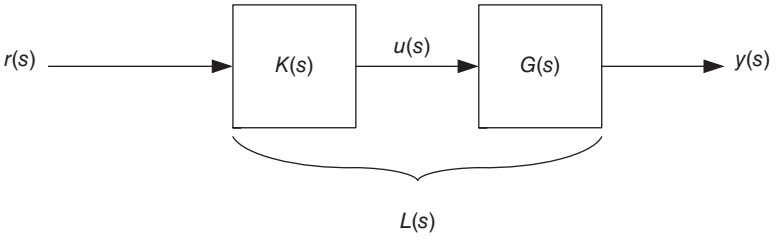

Figure 1.4 Series interconnection of two transfer function matrices.

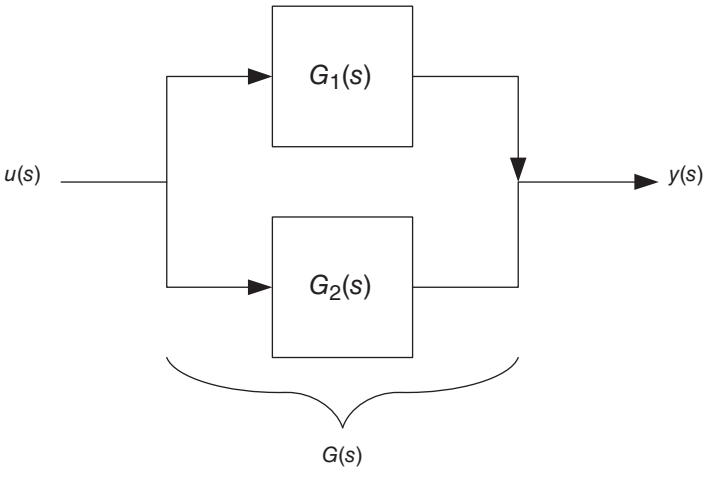

Figure 1.5 Two transfer function matrices in parallel.

1.3.1.2 Parallel Systems

For systems in parallel, the overall transfer function from input to output is obtained by simply adding the transfer functions of the individual paths.

Thus, in Figure 1.5, the transfer function $G(s)$ from $u(s)$ to $y(s)$ is given by:

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

1.3.1.3 Feedback Connection

When finding transfer functions involving feedback loops, we start as before at the output, go toward the input, and apply as appropriate the rules for series and parallel interconnections above. Then, at the point of leaving the feedback loop, multiply by $(I - L(s))^{-1}$, where $L(s)$ is the loop gain at the point of exiting the loop, going "countercurrent" to the direction of signal transmission around the loop.

Applying this to the system in Figure 1.6, we start at $y(s)$ and have noted $G(s)K(s)$ when we arrive at the point of exciting the feedback loop (in front of $K(s)$). The loop gain as seen from that point, going "countercurrent" around the loop, is $-F(s)G(s)K(s)$, remembering to account for the negative feedback. The overall transfer function from $r(s)$ to $y(s)$ is therefore given by:

$$y(s) = G(s)K(s)(I + F(s)G(s)K(s))^{-1}r(s)$$

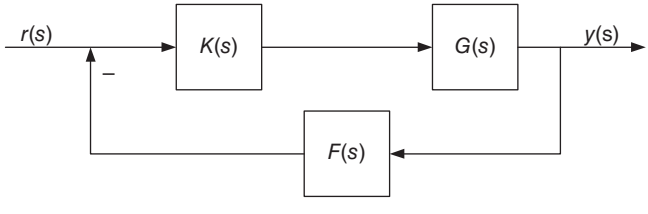

Figure 1.6 Feedback interconnection of systems.

1.3.1.4 Commonly Used Closed-Loop Transfer Functions

A simple feedback loop excited by disturbances d, reference changes r, and measurement noise n is illustrated in Figure 1.7.

Applying the rules for finding transfer functions above, we get

$$y = (I + GK)^{-1}G_d d + GK(I + GK)^{-1}r - GK(I + GK)^{-1}n \tag{1.36}$$

$$u = -K(I + GK)^{-1}G_d d + K(I + GK)^{-1}r - K(I + GK)^{-1}n \tag{1.37}$$

Two terms that appear repeatedly above are

$$(I + GK)^{-1} = S \quad \text{the sensitivity function}$$

$$GK(I + GK)^{-1} = T \quad \text{the complementary sensitivity function}$$

We will frequently refer to S and T, both by symbol and by name, but the origin of the names will be of little importance for our use of the terms.

1.3.1.5 The Push-Through Rule

The push-through rule says that

$$(I + M_1M_2)^{-1}M_1 = M_1(I + M_2M_1)^{-1} \tag{1.38}$$

The proof is left for the reader as an exercise. Note that the push-through rule holds also if M_1 and M_2 do not commute. If M_1 and M_2 are not square (but of compatible dimension), the identity matrices on each side of the equality above will have to be of different dimensions. Note also that the order of occurrence of M_1 and M_2 is the same on both sides of the equality sign above (ignoring all other symbols, we have $M_1 - M_2 - M_1$ on both sides). The push-through rule is sometimes a useful tool for simplifying transfer functions. Note that it implies

$$GK(I + GK)^{-1} = G(I + KG)^{-1}K = (I + GK)^{-1}GK$$

The matrix $S_I = (I + KG)^{-1}$ is sometimes called the *sensitivity function at the plant input*, and correspondingly $T_I = KG(I + KG)^{-1}$ is sometimes called the *complementary sensitivity function at the plant input*. S_I and T_I will not be used extensively in this book, but it is worth noting that for multivariable systems, the properties of a feedback loop depends on the location in the feedback loop.

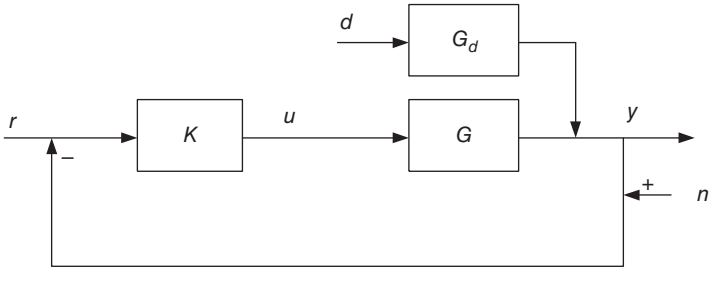

Figure 1.7 Basic feedback loop excited by disturbances d, reference changes r, and measurement noise n.

1.4 Poles and Zeros of Transfer Functions

Consider a scalar transfer function, that can be factored as:

$$G(s) = k\frac{(s+z_1)(s+z_2)\cdots(s+z_n)e^{-Ts}}{(s+p_1)(s+p_2)\cdots(s+p_m)} \tag{1.39}$$

where $m \geq n$, as otherwise there would be no state-space model that represent the transfer function dynamics. The parameters z_i are known as the *zeros* of the transfer function, whereas the p_i are termed *poles*. The term e^{-Ts} represents a pure time delay (transportation delay) of T time units. Zeros and poles can be either strictly real- or complex-valued. However, complex-valued zeros or poles always appear in complex conjugate pairs, since both the numerator and denominator of the transfer function have only real-valued coefficients (for transfer functions corresponding to a model described by ordinary differential equations). Remember that the time delay term e^{-Ts} cannot be described (exactly) by ordinary differential equations.

For a minimal representation of a system, the poles may also be defined as the roots of the characteristic polynomial (also called the *pole polynomial*):

$$\phi(s) = \det(sI - A) \tag{1.40}$$

Zeros and poles are often classified according to whether their real parts are positive or negative. Poles and zeros whose real part are strictly negative are called *left half-plane* (LHP) poles and zeros, respectively. Similarly, poles and zeros whose real parts are positive are called *right half-plane* (RHP) poles and zeros. RHP poles (for continuous-time systems) means that the system is unstable. If the open-loop system has an RHP pole, it will therefore be necessary to stabilize the system using feedback control. RHP poles for the closed-loop system is unacceptable. Poles in the LHP cause no fundamental problem.[6] LHP zeros also pose no particular problem for linear systems – although zeros close to the imaginary axis may indicate that the effect of the input is weak in the corresponding frequency range, and therefore there is a risk that the input magnitude required is larger than what is available.[7]

The problem with RHP zeros is that for high loop gain (corresponding to fast control), the closed-loop poles approach the open-loop zeros. Consider a simple feedback loop, such as Figure 1.10, and let the (open)-loop transfer function be composed of a controller[8] k and the plant transfer function $g(s) = n(s)/d(s)$. Thus $L(s) = k\frac{n(s)}{d(s)}$. The closed-loop transfer function from r to y is given by:

$$L(s)/(1+L(s)) = \frac{n(s)}{\left(\frac{d(s)}{k} + n(s)\right)}$$

We see that the closed-loop transfer function approaches 1 (the measurement tracks the reference signal) as $k \to \infty$. The closed-loop poles are given by the roots of the

6 Although they may also need to be moved by feedback, if they result in too slow responses for the application at hand.

7 Note that this problem does not show up in linear analysis, since magnitude bounds on inputs is a nonlinear effect.

8 A static controller is used for simplicity of exposition.

denominator polynomial of the closed-loop transfer function, and as $k \to \infty$ the denominator polynomial approaches the open-loop numerator polynomial. This means that the closed-loop poles will approach the open-loop zeros – resulting in poles in the RHP if the open-loop numerator polynomial has zeros in the RHP. Thus, open-loop zeros in the RHP are inconsistent with perfect control. The performance limitations arising from RHP zeros will be further elaborated in Chapter 4.

1.4.1 Poles of Multivariable Systems

For multivariable systems, the pole polynomial can be found from (1.40) just as for monovariable system. The pole polynomial can also be calculated from the transfer function matrix. All multivariable poles will appear as a pole of one or more transfer function elements, the only difficulty arises in knowing *how many* poles are needed, i.e. it is easy to find out that the system has a pole at p_i, but less obvious *how many* poles are at p_i (also known as the *multiplicity* of the pole at p_i). That issue is resolved by the following result from [7]:

Theorem 1.1 *The pole polynomial $\phi(s)$ for a system with transfer function $G(s)$ is the least common denominator of all not-identically-zero minors of all orders of $G(s)$.*

Recall that a *minor* of $G(s)$ is the determinant of a submatrix obtained by deleting rows and columns of $G(s)$. Minors of all orders include the individual elements, as well as the determinant of the overall matrix (or of the largest possible submatrixes, if $G(s)$ is not square). When calculating the minors, pole-zero cancelations of common terms in the numerator and denominator should be carried out whenever possible.

1.4.2 Pole Directions

The input and output pole directions, denoted u_{pi} and y_{pi}, respectively, capture the input direction with infinite gain and the corresponding output direction, for the system $G(s)$ evaluated at the pole $s = p_i$. That is, with some abuse of notation we may say that

$$G(p_i)u_{pi} = \infty \tag{1.41}$$

$$y_{pi}^H G(p_i) = \infty \tag{1.42}$$

The input and output pole directions could conceptually be found from the input and output singular vectors corresponding to the infinite singular value of $G(p_i)$. However, this is a numerically ill-conditioned calculation. Instead, the pole directions can be found starting from the right and left eigenvalue decomposition of the matrix A:

$$\begin{aligned} At_i &= p_i t_i \\ q_i^H A &= p_i q_i^H \\ u_{pi} &= B^H q_i \\ y_{pi} &= Ct_i \end{aligned}$$

We will throughout this note assume that the input and output pole directions have been normalized to have unit length. For single input single output (SISO) transfer functions, we trivially have $u_{pi} = y_{pi} = 1$.

1.4.3 Zeros of Multivariable Systems

We will first address multivariable zeros by considering a simple 2×2 example. Consider the plant

$$y(s) = G(s)u(s) = \frac{1}{s+1}\begin{bmatrix} 1 & s+1 \\ 2 & s+4 \end{bmatrix} u(s) \tag{1.43}$$

The system is open-loop stable. None of the elements of $G(s)$ have zeros in the RHP. Controlling output y_1 with the controller $u_1(s) = k_1(r_1(s) - y_1(s))$, we get

$$y_1 = \frac{g_{11}k_1}{1+g_{11}k_1}r_1 + \frac{g_{12}}{1+g_{11}k_1}u_2$$
$$y_2 = \frac{g_{21}k_1}{1+g_{11}k_1}r_1 + \left(g_{22} + \frac{g_{21}g_{12}k_1}{1+g_{11}k_1}\right)u_2$$

where the term inside the brackets is the transfer function from u_2 to y_2 when y_1 is controlled by u_1, in the following this is denoted $\tilde{g}_2$. Assume that a simple proportional controller is used, i.e. $k_1(s) = k$ (constant). Some tedious but straightforward algebra then results in

$$\tilde{g}_2(s) = \frac{1}{(s+1)(s+1+k)}\left[(s+4)(s+1+k) - 2k(s+1)\right]$$

We can then easily see that the system is stable provided $k > -1$ (clearly, a positive value for k would be used). For small values of k, $\tilde{g}_2$ has two real zeros in the LHP. For $k = 9 - 3\sqrt{8}$, the zeros become a complex conjugate pair, and the zeros move into the RHP for $k > 5$. For $k = 9 + 3\sqrt{8}$, both zeros again become real (but positive), and if k is increased further, one zero approaches $+\infty$ whereas the other zero approaches $+2$. Now, a zero of $\tilde{g}_2(s)$ far into the RHP will not significantly affect the achievable bandwidth for loop 2, but the zero which at high values of k approaches $+2$ certainly will.

Note that it will not be possible to avoid the zero in $\tilde{g}_2(s)$ by using a more complex controller in loop 1. The transfer function $\tilde{g}_2(s)$ will have a zero in the vicinity of $s = 2$ whenever high bandwidth control is used in loop 1.

If we instead were to close loop 2 first, we would get similar problems with loop 1 as we have just seen with loop 2. That is, if loop 2 were controlled fast, the transfer function from u_1 to y_1 would have a zero in the vicinity of $s = 2$.

We therefore conclude that it is a property of the plant that all directions cannot be controlled fast, as we saw above that high gain control of a system with an RHP zero leads to instability.

Looking at the term inside the square bracket in (1.43), we see that the determinant of $G(s)$ looses rank at $s = 2$ (its normal rank is 2, but at $s = 2$ it has rank 1). In terms of systems theory, the plant $G(s)$ has a *multivariable (transmission) zero at* $s = 2$.

There is no direct relationship between monovariable and multivariable zeros, a zero in an individual transfer function element may be at the same location as a multivariable zero, but often that will not be the case. However, as we have seen above, if a multivariable system with n outputs has a zero, and $n-1$ outputs are perfectly controlled using feedback, the zero will appear in any transfer function from the remaining manipulated variable to the remaining controlled variable (if the transfer function takes account of the fact that the other outputs are controlled).

RHP zeros in individual elements of a transfer function matrix need not imply a control performance limitation (they may become serious limitations, however, if parts of the control system is taken out of service, leaving only the loop with the monovariable RHP zero in service).

There are several definitions of zeros in multivariable systems, we will be concerned with the so-called *transmission* zeros[9] of multivariable systems, which occur when competing transmission paths within the system combine to give zero effect on the output, even though the inputs and states are nonzero. As for monovariable zeros, implications on achievable control performance arise mainly when the (transmission) zero is in the RHP.

As alluded to above, zeros of the system $G(s)$ are defined [7] as points z_i in the complex plane where the rank of $G(s)$ is lower than its normal rank. The corresponding *zero polynomial* is defined as:

$$\Theta(s) = \prod_{i=1}^{n_z}(s - z_i) \tag{1.44}$$

where n_z is the number of zeros.[10]

Zeros may be calculated from the transfer function matrix $G(s)$ according to Theorem 1.2. Note that the $G(s)$ will only contain zeros corresponding to a minimal state-space realization of the system.

Theorem 1.2 *[7] Let r be the normal rank of $G(s)$. Calculate all order-r minors of $G(s)$ and adjust these minors to have the pole polynomial $\phi(s)$ in the denominator. Then the zero polynomial $\Theta(s)$ is the greatest common divisor of the numerators of all these order-r minors.*

It is worth reflecting a little over the definition of a zero as a point where $G(s)$ loses rank, and the way zeros are calculated from Theorem 1.2. Consider a non-square system $G(s)$ of dimension $n \times m$:

- If $m < n$, zeros are relatively rare, because it is somewhat unlikely that all order-n minors will share the same zero. The exception is when there is a zero associated with a specific sensor, in which case all elements of the corresponding row of $G(s)$ will share the same zero, which will therefore also appear in all order-n minors.

9 The term *transmission* will frequently be dropped.
10 Disregarding any zeros at infinity, which have no particular implication for control performance.

- If $n > m$, it is also somewhat unlikely that all order-m minors will share the same zero, unless the zero is associated with a specific input, in which case all elements of the corresponding column of $G(s)$ will share the same zero. However, if there is a zero associated with a specific sensor,[11] there will still be a limitation to achievable control performance for the corresponding output – it just will not appear in the zero polynomial.

More commonly than using Theorem 1.2, multivariable zeros are calculated from the state-space description, solving the following generalized eigenvalue problem:

$$\begin{aligned}(z_i I_g - M)\begin{bmatrix} x_{z_i} \\ u_{zi} \end{bmatrix} &= 0 \\ M &= \begin{bmatrix} A & B \\ C & D \end{bmatrix} \\ I_g &= \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\end{aligned} \tag{1.45}$$

The solution to the above problem will give the zero z_i, the initial condition x_{zi} for the "transmission blocking" property and the input direction u_{zi} for the transmission blocking.

Multivariable zeros, like monovariable ones, are invariant to feedback and to input/output scaling.

1.4.4 Zero Directions

Zero input and output directions (denoted u_{zi} and y_{zi}, respectively) corresponding to a multivariable zero at $s = z_i$ contain information on the input and output directions with zero gain for $G(z_i)$. That is,

$$G(z_i)u_{zi} = 0 \tag{1.46}$$

$$y_{zi}^H G(z_i) = 0 \tag{1.47}$$

With knowledge of a multivariable zero of $G(s)$ at $s = z_i$ may be calculated from a singular value decomposition of $G(z)$. Alternatively, the input direction u_{zi} is found from (1.45). Likewise, a zero output direction can be calculated by solving (1.45) using M^T.

Whichever way u_{zi} and y_{zi} are calculated, we will assume that they have been normalized to have unit length. For our uses, the output direction y_{zi} of RHP zeros will be of most interest, as it provides information about how severely the different outputs are affected by the zero. Although the zero is invariant to scaling, the zero directions are not.

---

11 Or otherwise, it occurs that all order-m minors containing a specific row of $G(s)$ share a zero.

1.5 Stability

Assuming that we have a minimal representation of a linear system in continuous time. The system is then stable if

$$Re(\lambda_i(A)) < 0 \ \forall i \tag{1.48}$$

where $\lambda_i(A)$ denotes an eigenvalue of the matrix A in the state-space model. It follows from (1.26) that the eigenvalues of the A matrix also appear as poles of the transfer function. Stable systems thus have their poles strictly in the LHP (as already stated above).

Control textbooks may differ somewhat on whether systems with poles on the imaginary axis are considered stable. In some cases (as a result of a strict mathematical definition of stability), systems with *single* poles on the imaginary axis are classified as stable or "marginally stable", whereas systems with two or more poles in the same place on the imaginary axis are called unstable.

In most practical situations, systems with poles on the imaginary axis will need to be "stabilized" by feedback, irrespective of whether these poles are "single" or "multiple" poles. We will therefore classify all systems with poles on the imaginary axis as unstable.

Note that the eigenvalues of the A matrix correspond to the roots of the characteristic polynomial, which again (for a minimal representation) correspond to the poles of the transfer function. Clearly, these poles/roots/eigenvalues can be used equivalently (under the assumption of a minimal representation) to determine stability.

1.5.1 Poles and Zeros of Discrete-Time Transfer Functions

The forward shift operator z enters discrete-time transfer function in the same way as the Laplace variable s enters continuous-time transfer functions. Poles and zeros may therefore be calculated for discrete-time transfer functions in the same way as for continuous-time transfer functions, and also for discrete-time transfer functions it holds that the poles equal the eigenvalues of the A matrix.

However, the interpretation of the values of the poles and zeros differ for discrete-time and continuous-time transfer functions.

Whereas continuous-time transfer functions are stable if the poles are in the (open) LHP, discrete-time transfer functions are stable if the poles are inside the unit circle. That is, a discrete-time system is stable if

$$|\lambda_i| < 1 \ \forall i \tag{1.49}$$

Furthermore, a continuous-time pole at $s = 0$ indicates integrating dynamics. For discrete-time systems, the corresponding pole for an integrating system will be at $z = 1$. For continuous-time systems, complex conjugate poles indicate oscillatory dynamics. For discrete-time systems, it also holds that any poles away from the real axis must occur in complex conjugate pairs, but any poles away from the *positive real* axis indicate oscillatory dynamics.[12]

12 That is, poles on the negative real axis are also oscillatory.

1.5.2 Frequency Analysis

In recent years, frequency analysis has been given less room in process control education. This seems to be a particularly prominent trend in chemical engineering departments in the United States, where control seems to be squeezed by the wish to include "newer" topics such as materials/nano-/bio. Although many esteemed colleagues argue that control can be taught just as well entirely with time-domain concepts, it is this author's opinion that the same colleagues are making the mistake of elevating a necessity to a virtue.

Despite this worrisome trend, the presentation of frequency analysis in this book will be sketchy, assuming that the reader has had a basic introduction to the topic in other courses.

This author agrees with the arguments expressed by Skogestad and Postlethwaite [9] on the advantages of frequency analysis. While those arguments will not be repeated here, but we will note that many control-relevant insights are easily available with a working understanding of frequency analysis.

In this chapter, the frequency response will be used to describe a systems response to sinusoidal inputs of varying frequency. Although other interpretations of the frequency response are possible (see, again, [9]), the chosen interpretation has the advantage of providing a clear physical interpretation and a clear link between the frequency and time domain.

The frequency response of a system with transfer function $G(s)$ at the frequency ω is obtained by evaluating $G(s)$ at $s = j\omega$. The result is a complex-valued number (or a complex-valued *matrix*, for multivariable systems). It should be noted that the frequency ω is measured in *radians/time*,[13] and thus the oscillation period corresponding to the frequency ω is $t_p = 2\pi/\omega$.

The complex-valued frequency response is commonly presented in polar coordinates in the complex plane, with the length being termed the *gain* (or sometimes the *magnitude*) and the angle being termed the *phase*. Anticlockwise rotation denotes positive phase.

That is, consider $G(j\omega) = a + jb$. The gain is then $|G(jw)| = \sqrt{a^2 + b^2}$, whereas the phase is given by $\angle G(j\omega) = \tan^{-1}(b/a)$. Thus, assume that a sinusoidal input is applied:

$$u(t) = u_0 \sin(\omega t + \alpha) \tag{1.50}$$

Once the effect of any initial conditions have died out (or, we might make the "technical" assumption that the input has been applied "forever," since $t = -\infty$), the output will also oscillate sinusoidally at the same frequency:

$$y(t) = y_0 \sin(\omega t + \beta) \tag{1.51}$$

We will then observe that $|G(j\omega)| = y_0/u_0$ and $\angle G(j\omega) = \beta - \alpha$. For multivariable systems, the response of each individual output can be calculated as the sum of the

13 Usually, time is measured in seconds, but minutes are also sometimes used for slow process units such as large distillation towers.

responses to each of the individual inputs. This property holds for all linear systems – both in the time domain and in the frequency domain.

For $G(s)$ in (1.39), we have

$$|G(j\omega)| = |k| \cdot \frac{\prod_{i=1}^{n} |(j\omega + z_i)|}{\prod_{i=1}^{m} |(j\omega + p_i)|} \cdot 1 \tag{1.52}$$

$$\angle G(j\omega) = \angle(k) + \sum_{i=1}^{n} \angle(j\omega + z_i) - \sum_{i=1}^{m} \angle(j\omega + p_i) - \omega T \tag{1.53}$$

The phase and gain of a single term $(s + a)$ is illustrated in Figure 1.8.

Thus, we multiply the gains of k and the numerator terms in the transfer function and divide by the gains of the denominator terms. For the phase, we add the phases of the numerator terms and the (negative) phase from the time delay and subtract the phase contribution from the denominator terms.

In the previous section, we have used Euler's formula to determine the phase and gain of the time delay term:

$$e^{ja} = \cos a + j \sin a \tag{1.54}$$

from which we find that $|e^{-j\omega T}| = 1 \ \forall \omega$ and $\angle e^{-j\omega T} = -\omega T(\text{rad}) = -\frac{\omega T}{\pi} \cdot 180°$.

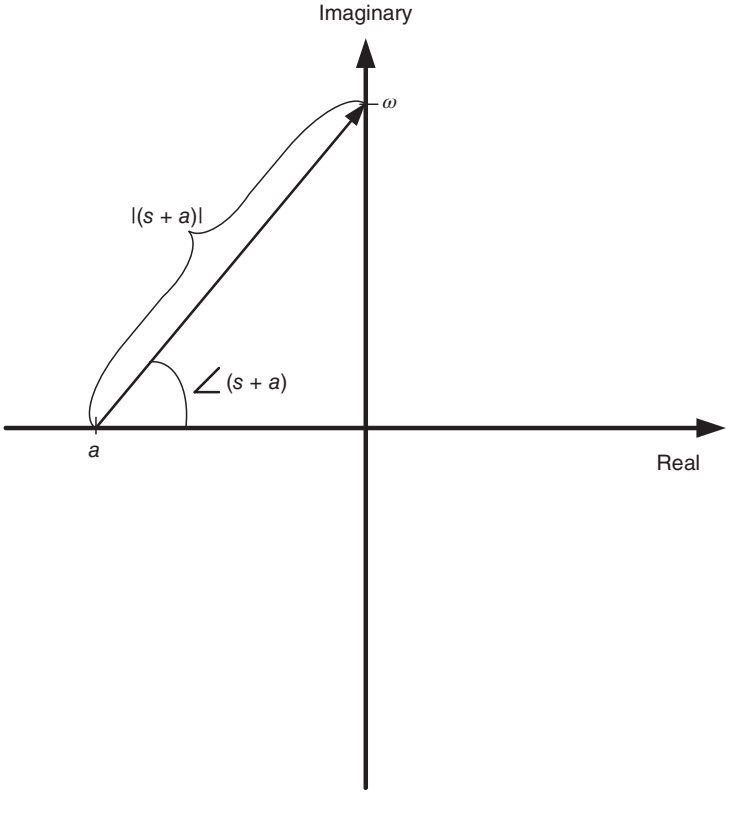

Figure 1.8 The phase and gain of a simple term $(s + a)$ for $a > 0$.

Mathematically, k will have a phase of zero if $k > 0$ and a phase of $-\pi = -180°$ if $k < 0$. However, for stability analysis, this term is of no consequence – in practice, if the plant has a negative gain we simply reverse the sign of the gain in the controller – see the paragraph on steady-state phase adjustment below. That is, for stability assessment using the Bode stability criterion (to be described in the following section), we set the phase contribution from k to zero.

1.5.2.1 Steady-State Phase Adjustment

The steady-state value of the transfer function is obtained by evaluating the transfer function at $s = 0$. Provided there are no poles or zeros at the origin ($z_i \neq 0\ \forall i$, $p_j \neq 0\ \forall j$ in (1.39)), at $s = 0$ the transfer function takes a real value and thus must have a phase of $n \times 180°$, where n is some integer.

Clearly, a purely imaginary term (for $s = j\omega$) contributes 90° to the phase at all nonzero frequencies. For zeros at the origin the phase contribution is positive, and for poles the phase contribution is negative (for positive frequencies).

It is customary to adjust or "correct" the phase such that the phase contribution for the constant k is zero. Similarly, the phase contribution of any RHP zero in (1.39) is adjusted such that its phase at steady state is zero.

This phase adjustment is necessary to be able to assess closed-loop stability from the open-loop frequency response. For *open-loop stable* systems without zeros or poles at the origin, this corresponds to setting the steady-state phase to zero or assuming a positive steady-state gain. If the real steady-state gain is negative (if the output decreases when the input increases), this in corrected for by simply reversing the sign of the gain of the controller – often this is done by specifying that the controller should be "direct acting." See Section 2.4.3.12 for an explanation of direct and reverse acting controllers.

The phase adjustment described above is done irrespective of whether the system is stable in open loop. Note, however, that the phase of any unstable (RHP) poles are *not* adjusted in this way. This may appear inconsistent but is possibly most easily understood by noting that one cannot "normalize the steady-state phase" for a RHP pole. An RHP pole represents an instability in the system, the output will grow exponentially without bounds as a response to a change in the input, and thus there *is no (stable) steady state* for an RHP pole.

After steady-state phase adjustment, the phase of $G(j0)$ should therefore be

$$\angle(G(j0)) = -180° n_p - 90° n_i + 90° n_{z0} \tag{1.55}$$

where n_p is the number of poles in the RHP (unstable poles), n_i is the number of poles at the origin (integrating poles),[14] and n_{z0} is the number of zeros at the origin.[15]

---

14 Strictly speaking, the angle at steady state ($s = j0$) is not well defined if the plant has poles at the origin. In this case, the aforementioned equation should be regarded as representing $\lim_{\omega \to 0^+} \angle(G(j\omega))$.

15 Note that poles and zeros in the same location should be canceled in the transfer function so that at least one of n_i and n_{z0} should be zero.

Mathematical software (such as Matlab) may produce a phase that is off by $n \cdot 180°$ compared with what is explained above. The phase calculated by such software is mathematically correct – in the sense that a negative real number has a phase of $-180°$ and a rotation of $n \cdot 360°$ describes the same point in the complex plane. Also, one can expect that any controller tuning tool based on such software correctly indicates closed-loop stability. However, when using frequency analysis for controller tuning (as addressed in Section 2.4.3.2), ignoring the steady-state phase adjustment confuses what undesired phase has to be corrected by dynamic compensation and what is handled by setting the controller to be direct or reverse acting.

1.5.3 Bode Diagrams

The frequency response of a scalar system is often presented in a *Bode diagram* (sometimes also called amplitude-phase-frequency diagram). The Bode diagram consists of two plots, the *magnitude plot* and the *phase plot*.

In the magnitude plot, the transfer function magnitude (or gain) is plotted vs. frequency. Both the magnitude and the frequency axes are logarithmic (to the base 10).

Remark: Note that the magnitude scale used for the Bode magnitude plot in this book is the conventional logarithmic scale (to the base 10). In some books, one can still see the decibel (dB) scale used in the Bode magnitude plot, where

$$|G(jw)|(\text{dB}) = 20\log_{10}|G(j\omega)| \tag{1.56}$$

We repeat that the decibel scale is *not* used in this book.

In the Bode phase plot, the phase is plotted against frequency. The phase is usually plotted in degrees using a linear scale (radians are seldom used), whereas a logarithmic scale is used for the frequency axis. A Bode diagram of the simple system $g(s) = \frac{s+0.1}{(s+0.01)(s+1)}$ is shown in solid lines in Figure 1.9.

Control software that plots Bode diagrams are now easily available, and manual procedures for drawing Bode diagrams are therefore obsolete. One should, however, take a little care to ensure that the steady-state phase is correctly adjusted, as outlined in Section 1.5.3.1. Otherwise, the steady-state phase can easily be off by some multiple of 180°.

1.5.3.1 Bode Diagram Asymptotes

Although procedures for manually drawing Bode diagrams are now obsolete, it is useful to be able to quickly visualize the phase–gain relationships of the Bode diagram – possibly without drawing any diagram at all. For this purpose, knowledge about the Bode diagram asymptotes are useful. This is particularly useful when considering changes to controller parameters for proportional integral (PI)/proportional integral derivative (PID) controllers, since it can give an intuitive understanding of the effects of such changes and thereby simplify the search for appropriate controller parameters. These asymptotes are rather inaccurate approximations to the

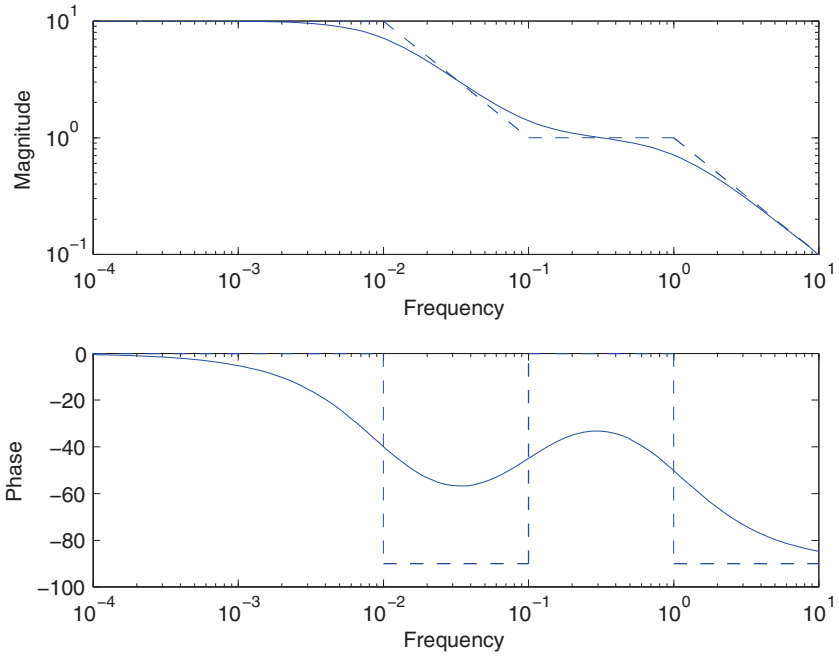

Figure 1.9 The Bode diagram for the simple system $g(s) = 10\frac{(10s+1)}{(100s+1)(s+1)} = \frac{s+0.1}{(s+0.01)(s+1)}$.

exact diagram in the frequency range near a pole or zero, but good approximations at frequencies removed from poles and zeros.

To obtain the asymptotes for the Bode magnitude plot,

- Start from the steady-state gain of the system, $|G(0)|$. If the system has "pure integrators" (poles at $s = 0$), evaluate the transfer function instead at some very low frequency, several decades below any other pole or zero.
- The gradient of the magnitude asymptote (in the log–log scale used in the magnitude plot) at low frequencies is $n_{z0} - n_i$, where n_{z0} is the number of zeros at the origin and n_i is the number of poles at the origin.
- Increase frequency ω. Whenever $\omega = z_i$, *increase* the gradient of the asymptote by 1. Whenever $\omega = p_i$, *decrease* the gradient of the asymptote by 1.

The asymptotes for the Bode phase plot are obtained as follows:

- If the transfer function contains n_i poles at the origin, they contribute a total of $-90^\circ \cdot n_i$ of phase at (very) low frequencies. Similarly, if the transfer function contains n_{z0} zeros at the origin, these contribute a total of $90^\circ \cdot n_{z0}$ of phase at (very) low frequencies.
- Poles in the LHP (the closed LHP except the origin) do not contribute to the phase at steady state. The zeros (anywhere except at $s = 0$) also do not contribute the phase at steady state.
- Poles in the open *RHP* each contribute -180° to phase at steady state.

- Add the phase contributions at steady state. This gives the value of the low-frequency phase asymptote.
- Gradually increase frequency ω. If $\omega = z_i$ (a zero in the LHP), *increase* the asymptote phase by 90°. If $\omega = -z_i$ (a zero in the RHP), *decrease* the asymptote phase by 90°. If $\omega = p_i$ (a pole in the LHP), *decrease* the asymptote phase by 90°. If $\omega = -p_i$ (a pole in the RHP), *increase* the asymptote phase by 90°.

The phase asymptote thus changes in steps of (multiples of) 90°. Note that this way of finding the phase asymptote does not include the time delay. The phase contribution of any time delay therefore has to be added separately afterward, as described earlier. With the logarithmic frequency axis used in the Bode diagram, the time delay contributes little to the phase at $\omega \ll 1/T$, but adds a lot of negative phase at higher frequencies.

To use the above description to account for the phase and magnitude contributions of complex-valued poles or zeros (which have to appear in complex conjugate pairs), the absolute value of the poles or zeros is used instead of the complex-valued p_i or z_i. In this case, the phase and gradient changes must be multiplied by a factor of 2, since the frequency corresponding to two poles/zeros are passed simultaneously. Note that if the system has complex conjugate poles close to the imaginary axis, the magnitude plot may have a large "spike" that is not captured by the asymptote.

Note from the above description that the phase contribution at low frequencies of a zero in the RHP is essentially the same as that of the zero's "mirror image" in the LHP, whereas at high frequencies the phase contribution of the two differ by 180°.

In contrast, the phase contribution at low frequencies of a pole in the RHP is 180° different from that of its "mirror image" in the LHP, but at high frequencies the phase contribution of the two are essentially the same.

The asymptotes are shown with dashed lines in Figure 1.9. The system $g(s) = \frac{s+0.1}{(s+0.01)(s+1)}$ has a steady-state gain of 10, no pure integrators or differentiators. The magnitude asymptote therefore starts with a gradient of 0, while the phase asymptote starts with a phase of 0°. The first pole is at $p_i = 0.01$. At $\omega = 0.01$, the gradient of the magnitude asymptote therefore changes to −1, whereas the phase asymptote goes to −90°. At $\omega = 0.1$ we encounter the (LHP) zero, and thus the gradient of the magnitude asymptote increases to 0, and the phase asymptote goes to 0° again. Finally at $\omega = 1$, we encounter the second pole, changing the gradient of the magnitude asymptote to −1 and the phase asymptote to −90°.

1.5.3.2 Minimum Phase Systems

It should be clear from the above that whether a pole or a zero is in the right or LHP does not affect the Bode magnitude plot, whereas it does affect the phase plot. It turns out that for any system with a given magnitude plot,[16] there is a minimum possible (negative) phase that the system can have. This minimum possible phase can be quantified in terms of the Bode phase–gain relationship, which from which the minimum possible phase can be calculated from an integral over all frequencies

16 Assuming that this magnitude plot makes physical sense, i.e. that it can correspond to a state-space model.

of an expression involving the magnitude. The precise form of this expression is of little importance in our context, the interested reader may consult [9] or other textbooks on linear systems theory. One can, however, find from the expression that the local phase depends strongly on the local gradient of the magnitude in the log–log plot (the Bode magnitude plot). Thus, the minimum possible phase is approximately given by:

$$\angle G(j\omega)_{min} \approx -90^\circ \cdot \frac{d\log(|G(j\omega)|)}{d\log(\omega)} \tag{1.57}$$

That is, if the Bode magnitude plot has a gradient of $-n$, the minimum negative phase we can expect is around $-90n^\circ$. Non-minimum phase systems have additional negative phase. Whereas this approximation is exact at all frequencies only for a series of integrators ($G(s) = s^{-n}$), it can be a reasonable approximation for most minimum phase systems except at frequencies where complex poles or zeros are close to the imaginary axis. From the Bode stability criterion in Section 1.5.4, it will become clear that stability is incompatible with a transfer function magnitude that has a steep negative gradient in the crossover region.

From the brief introduction to frequency analysis presented above, it should be clear that a minimum phase system has

- no poles or zeros in the RHP, and
- has no time delay.

Minimum phase systems are often relatively easy to control, as the system dynamics pose no special limitations or requirements for feedback control. In contrast, as we will see later in this book, RHP poles imply a minimum bandwidth requirement, whereas RHP zeros or time delays implies a bandwidth limitation.

1.5.3.3 Frequency Analysis for Discrete-Time Systems

In principle, one may perform frequency analysis for discrete-time systems, using the identity

$$z = e^{st}$$

and using $s = j\omega$. Here, t denotes the sampling interval of the discrete-time system – and one needs to keep in mind that the frequency analysis is only valid up to the Shannon sampling frequency, i.e. the frequency for which

$$\omega t = \pi$$

Alternatively, if one wants to use the Bode stability criterion (see below), one may use a bilinear transform mapping the unstable region for the discrete-time system (outside the unit circle) to the unstable region for continuous-time systems (the RHP).

Techniques for frequency analysis for discrete-time systems will not be further detailed here – as it is this author's distinct impression that it is by far more common to perform frequency analysis using continuous-time models. Nevertheless, it is worthwhile pointing out that not only discrete-time poles outside the unit circle are problematic (as they indicate instability), but so are also discrete-time zeros outside the unit circle (corresponding to non-minimum phase zeros in the continuous-time case).

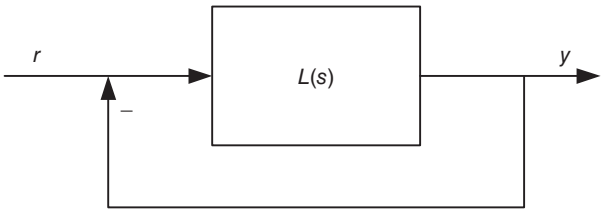

Figure 1.10 A simple feedback loop.

1.5.4 Assessing Closed-Loop Stability Using the Open-Loop Frequency Response

Let $L(s)$ be the open-loop transfer function matrix of a feedback system, as illustrated in Figure 1.10. The loop transfer function $L(s)$ may be monovariable or multivariable, and a feedback control setting typically results from connecting a controller $K(s)$ and a plant $G(s)$ in series, i.e. $L(s) = G(s)K(s)$. We will assume that there are no hidden (unobservable or uncontrollable) unstable modes in $L(s)$ and are interested in determining closed-loop stability based on open-loop properties of $L(s)$. The Nyquist stability theorem provides such a method for determining closed-loop stability, using the so-called *Principle of the Argument*.

1.5.4.1 The Principle of the Argument and the Nyquist D-Contour

The Principle of the Argument is a result from mathematical complex analysis. Let $t(s)$ be a transfer function and C be a closed contour in the complex plane. Assume that the transfer function $t(s)$ has n_Z zeros and n_P poles inside the closed contour C, and that there are no poles on C.

The Principle of the Argument Let s follow C once in the clockwise direction. Then, $t(s)$ will make $n_Z - n_P$ clockwise encirclements of the origin.

In this context, the term "Argument" refers to the phase of the transfer function.

We are interested in stability of the closed loop, which clearly means that we want to investigate whether the closed loop has any poles in the RHP. Thus, the contour C will in our case be the "border" of the entire RHP, i.e. the entire imaginary axis – turned into a closed loop by connecting the two ends with an "infinitely large" semicircle around the RHP.[17] To fulfill the requirement that there should be no poles on the closed contour, we must make infinitesimal "detours "into the RHP to go around any poles on the imaginary axis (most commonly due to pure integrators in the plant $G(s)$ or controller $K(s)$). The closed contour described above is commonly known as the *Nyquist D-contour*.

17 A brief look at the expression for $G(s)$ in (1.26) – while remembering that the transfer function $t(s)$ above can be expressed similarly – should suffice to convince the reader that the value of $t(s)$ will remain constant as s traverses the "infinitely large semicircle" around the RHP. For *very* large s, $C(sI - A)^{-1}B \approx 0$ regardless of the direction from the origin to s.

1.5.4.2 The Multivariable Nyquist Theorem

It can be shown (e.g. [8]) that the open and closed-loop characteristic polynomials are related through

$$\det(I + L(s)) = \frac{\phi_{cl}(s)}{\phi_{ol}(s)} \cdot c \tag{1.58}$$

where c is a constant. The number of open-loop poles in the RHP cannot be changed by feedback. However, for closed-loop stability, we must ensure that there are no closed-loop poles in the RHP. Using the Principle of the Argument, we thus arrive at the general or *multivariable Nyquist theorem*:

Theorem 1.3 *Let the number of open-loop unstable poles in $L(s)$ be n_{ol}. The closed-loop system with negative feedback will then be stable if the plot of* $\det(I + L(s))$ *does not pass through the origin but makes $-n_{ol}$ (clockwise) encirclements of the origin as s traverses the Nyquist D-contour.*

Note that in practice we only need to plot $\det(I + L(s))$ for positive frequencies only, since the plot for negative frequencies can be obtained by mirroring about the real axis.

1.5.4.3 The Monovariable Nyquist Theorem

Most readers are probably more familiar with the monovariable Nyquist theorem, which follows from the multivariable version by noting that for a scalar $L(s)$ it is equivalent to count encirclements of $\det(I + L(s))$ around the origin and encirclements of $L(s)$ around -1.

1.5.4.4 The Bode Stability Criterion

The Bode stability criterion follows from the monovariable Nyquist theorem and *thus applies only to monovariable systems*.

Theorem 1.4 *Let ω_c denote the "crossover frequency," i.e. $|L(j\omega_c)| = 1$, and assume that $|L(j\omega)| < 1$ for $\omega > \omega_c$. Then the closed-loop system is stable provided $\angle L(j\omega_c) > -180°$.*

The Bode stability criterion ensures that the Nyquist plot of $L(s)$ passes between the origin and the critical point -1 in the complex plane. For open-loop stable systems, it is then straightforward to see that there can be no encirclements of the critical point. However, the criterion may also be used for open-loop unstable systems provided the Bode phase plot starts from the correct phase of $-180°n_p$, where n_p is the number of RHP poles, and the crossover frequency ω_c is unique (i.e. that there is only one frequency ω_c for which $|L(j\omega_c)| = 1$).

If the assumption $|L(j\omega)| < 1$ *for* $\omega > \omega_c$ is violated, the Bode stability criterion is easily misinterpreted, and the use of the Nyquist criterion is recommended instead.

For open-loop stable systems, the Bode stability criterion may equivalently be stated in terms of ω_{180}, defined such that $\angle L(j\omega_{180}) = -180°$. The closed-loop

system is then stable if $|L(j\omega)| < 1$ for $\omega \geq \omega_{180}$. For most systems, the magnitude $|L(j\omega)|$ will decrease with increasing frequency, and it will thus suffice to check the criterion only at ω_{180}. However, this version of the criterion cannot be used for open-loop unstable systems, since ω_{180} need not be uniquely defined – and the criterion must indeed be violated for one or more of the ω_{180}'s.

Mini-tutorial 1.2 Bode diagram and feedback stabilization of an unstable system

Consider the unstable system $g(s) = \frac{1}{10s-1}$, that we want to stabilize with the proportional feedback controller k. The closed-loop pole can be found from the closed-loop characteristic polynomial, by solving the equation $1 + g(s)k = 0$. We thereby find that the closed-loop pole is located at $s = \frac{1-k}{10}$, and the closed loop will be stable for $k > 1$. We note that $\omega_{180} = 0$, and that $\angle L(j\omega) > -180^\circ \ \forall \omega > 0$. We can easily calculate $\omega_c = \frac{\sqrt{k^2-1}}{10}$. That is, for $k < 1$, $|L(j\omega)| = |g(j\omega)k| < 1 \ \forall \omega$, and there is thus no crossover frequency ω_c. Thus, we find also from the Bode stability criterion (in terms of ω_c) that we need $k > 1$ for stability. The Bode stability criterion in terms of ω_{180} would fail – but as noted above this is only valid for stable systems.

In Figure 1.11, the Bode diagram for the system in this example is shown for $k = 2$. We find that $\omega_c = \frac{\sqrt{3}}{10}$ and $\angle L(j\omega_c) = -120^\circ$, i.e. the system is stable and we have a phase margin of 60°.

Stability of the closed-loop system can also be verified from the monovariable Nyquist theorem. We find that the image of $L(s)$ under the Nyquist D-contour encircles the critical point $(-1,0)$ once in the anticlockwise direction, as shown in Figure 1.12.

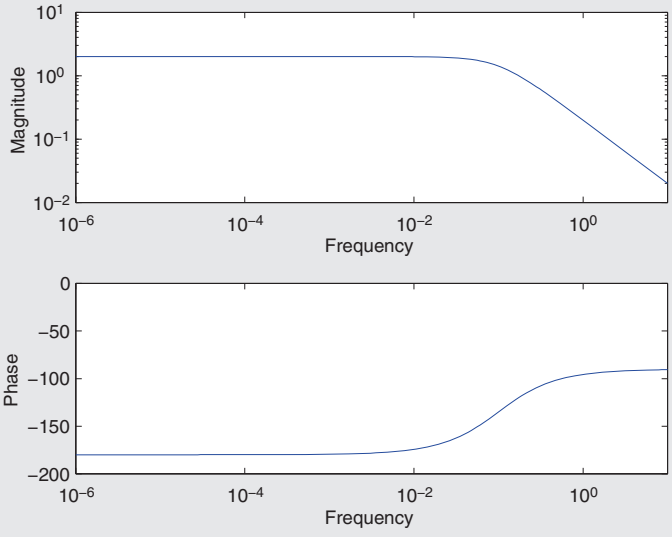

Figure 1.11 Bode diagram for the system $L(s) = \frac{2}{10s-1}$.

(Continued)

Mini-tutorial 1.2 (Continued)

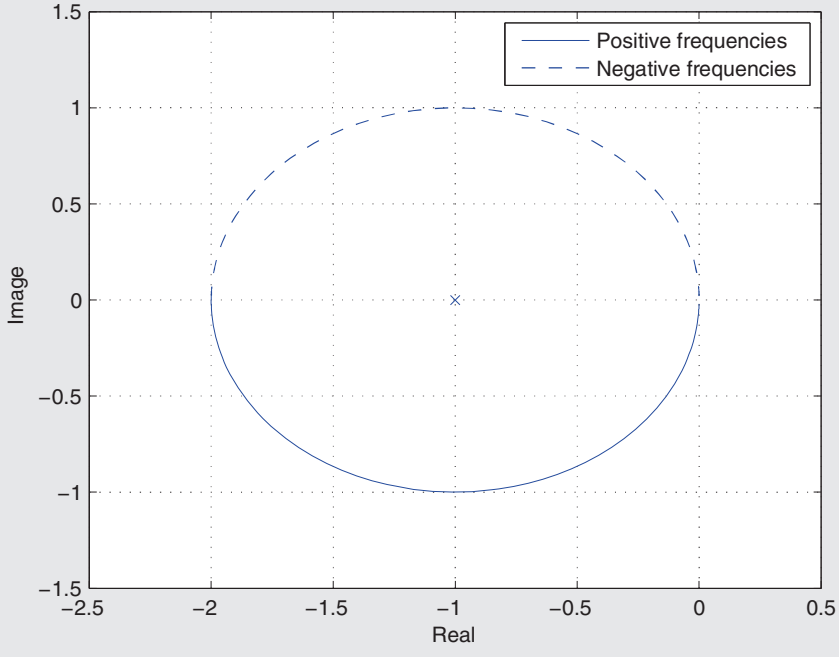

Figure 1.12 The monovariable Nyquist theorem applied to the system $L(s) = \frac{2}{10s-1}$. The curve encircles the critical point $(-1,0)$ once in the anticlockwise direction, and the system is hence stable.

Mini-tutorial 1.3 Controller adjustment based on Bode diagram asymptotes

When a control loop is oscillating, operators will often "detune" the loop, (i.e. reduce the gain in the controller), as high gain control usually leads to instability. We shall see that this approach will not always be successful in removing oscillations.

Consider a liquid level control problem, with the outlet flowrate being used to control the level. In practice, a valve is used to manipulate the flowrate, and a local flow controller is used in cascade with the level controller. The flow controller receives a flow measurement, and the setpoint (reference value) for the flow controller is the output of the level controller (see subsequent section on controllers in cascade). The flow control loop should be much faster than the outer level control loop, and an approximate model of the system as seen by the level controller is then

$$y(s) = g(s)u(s) = \frac{h}{s}u(s)$$

This model is good in the frequency range for which the flow control is good, i.e. inside the bandwidth of the flow controller. At higher frequencies, one must expect

the flow control to contribute additional negative phase. The level controller is a PI controller

$$u(s) = k(s)e(s) = k_p \frac{T_I s + 1}{T_I s} e(s)$$

where $e(s) = r(s) - y(s)$ is the control offset, $r(s)$ is the setpoint, and $y(s)$ is the measurement. The level control loop is observed to be oscillating – should the controller gain k_p be decreased?

To answer this question, one should first consider the frequency of the oscillation. This can be estimated from $\omega_c = t_p/2\pi$, with t_p being the time between subsequent peaks in the oscillating response. The oscillations indicate that the loop transfer function $L(s) = g(s)k(s)$ has a phase of approximately -180° at ω_c. Observe that the phase asymptote for the controller $k(s)$ is -90° for frequencies $\omega < 1/T_I$, and 0° for frequencies $\omega > 1/T_I$, while the phase of the plant transfer function $g(s)$ is -90°.

We can now distinguish two cases:

1. If $\omega_c < 1/T_I$, the crossover frequency ω_c is in the region where the loop transfer function phase asymptote is -180°, and the oscillations are to be expected. Furthermore, decreasing the controller gain k_p will *not* increase the phase at the crossover frequency – so the oscillations would persist, but at a lower frequency. Instead, the controller gain k_p should be increased to move the crossover frequency beyond $1/T_I$. This will result in a positive gain margin at ω_c, and the oscillations will be removed.
2. If $\omega_c > 1/T_I$, the loop transfer function phase asymptote should be -90° at ω_c, while the observed oscillations indicate that the actual phase of the loop transfer function is close to -180°. The additional negative phase probably comes from unmodeled dynamics in the flow control loop. The phase contribution of this neglected dynamics must generally be expected to increase with increasing frequency. Thus, decreasing the controller gain k_p will improve the phase margin and reduce the oscillations.

Simple considerations involving the asymptotes of the Bode plot, thus, suffice to understand how to modify the controller tuning in this case.

1.5.4.5 Some Remarks on Stability Analysis Using the Frequency Response

Frequency analysis can indeed be very useful. However, some remarks seem to be needed to warn against misuse of frequency analysis for analyzing stability:

- The Nyquist stability theorems and the Bode stability criterion are tools to assess *closed-loop stability based on open-loop frequency response data*.
- Knowledge of the number of open-loop unstable poles is crucial when using Nyquist or Bode.
- Nyquist or Bode should **never** be used to assess open-loop stability!

- It is *utterly absurd* to apply the Bode stability criterion to the individual elements of a multivariable system, and the Bode stability criterion applies to monovariable systems only. The multivariable Nyquist theorem is used to assess closed-loop stability of multivariable systems based on the open-loop frequency response.

1.5.4.6 The Small Gain Theorem

In the multivariable Nyquist theorem, we count the number of encirclements of $\det(I + L(s))$ around the origin as s traverses the Nyquist D-contour. It is therefore intuitively obvious that if the loop gain $L(s)$ is "smaller than 1" (in some sense), we cannot have any encirclements of the origin, and any open-loop stable system will remain stable in closed loop. For a scalar $L(s)$, we may of course use the ordinary transfer function magnitude to measure the size of $L(s)$.

For multivariable systems, we will require a system *norm* to measure magnitude. This is denoted $\|L\|_x$. There are several different system norms, and the subscript x will identify the specific norm in question. While we will not use the norm concept much in this book, it is widely used in robustness analysis. Interested readers may find an accessible introduction to (vector, signal, and system) norms in [4], and their use in *robustness analysis*[18] in [9]. However, it is pertinent to point out that eigenvalues are not system norms (when evaluating the transfer function matrix at some given value of s). The most frequently used norm in robustness analysis is $\|L\|_\infty$, which corresponds to the peak value along the imaginary axis of the maximum singular value of $L(s)$.[19]

In its basic form, the small gain theorem may not appear very useful. From single loop control, we know that we need high gain for good control performance. It may therefore appear that we can tolerate only very small uncertainty at frequencies where good performance is required – and hence the loop gain is large. However, sometimes one can factorize the loop gain in ways which makes the small gain theorem very useful, especially in robustness analysis. Consider Figure 1.13. The left part of the figure depicts an ordinary control loop, with uncertainty in the effect

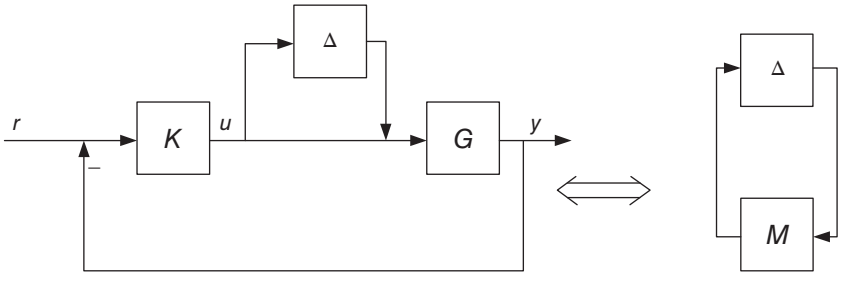

Figure 1.13 Feedback loop with uncertainty converted to $M - \Delta$ structure for small gain analysis.

18 The analysis of whether system stability and/or performance is sensitive to (inevitable) model errors.

19 Any norm used in robustness analysis must fulfill the multiplicative property $\|AB\|_x \leq \|A\|_x \cdot \|B\|_x$, which the ∞-norm fulfills, see [9].

of the inputs modeled by the Δ block. Assume that the loop is *nominally stable*, i.e. it is stable when $\Delta = 0$. Inputs and outputs can be ignored with respect to stability analysis, and hence for stability analysis the left part of the figure can be converted to the $M - \Delta$ feedback structure in the right part, where $M = -KG(I + KG)^{-1} = -T_I$. At frequencies within the closed-loop bandwidth, i.e. where the loop gain is large, we will have $\|M(j\omega)\| \approx 1$ (despite the loop gain $\|L_I(j\omega)\| = \|K(j\omega)G(j\omega\|$ being large).[20] Thus, substantial error may be tolerated at low frequencies without jeopardizing stability – since what the small gain theorem tells us is that we require $\|M\Delta\| < 1\ \forall\omega$. Even larger model errors may be tolerated at frequencies well beyond the closed-loop bandwidth, where $\|M(j\omega)\| \ll 1$. The system will be most sensitive to uncertainty in the bandwidth region, where we may have a peak in $\|M(j\omega)\|$.

1.5.5 Controllability

Definition 1.1 The continuous-time dynamical system $\dot{x} = Ax + Bu$ (or, the matrix pair (A, B)) is *controllable*; if for any initial state $x(0)$ there exists a (piecewise continuous) input $u(t)$ that brings the state to any $x(t_1)$ for any $t_1 > 0$.

There exists a number of different criteria for testing controllability. Zhou et al. [11] prove that the following are equivalent:

- (A, B) is controllable
- The Gramian matrix

$$W_c(t) := \int_0^t e^{A\tau} B B^T e^{A^T \tau} d\tau \tag{1.59}$$

  is positive definite for any $t > 0$.
- The controllability matrix

$$\mathcal{C} := \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{1.60}$$

  has full row rank, where n is the number of states (i.e. A is of dimension $n \times n$).
- The matrix $[A - \lambda I \quad B]$ has full row rank for all values of the complex-valued scalar λ.
- For any eigenvalue λ and corresponding left eigenvector m of A (i.e. $m^*A = m^*\lambda$), $m^*B \neq 0$.
- The eigenvalues of $A + BF$ can be freely assigned – with the only restriction that complex eigenvalues must appear in conjugate pairs – by a suitable choice of F.

Using the Gramian $W_c(t)$ in (1.59), an explicit expression can be found for the input that brings the system from $x(0)$ to $x(t_1)$[21] :

$$u(t) = -B^T e^{A^T(t_1 - t)} W_c(t_1)^{-1} (e^{At_1} x_0 - x_1) \tag{1.61}$$

---

20 Here, the notation $\|M(j\omega)\|$ indicates that we are evaluating the norm of M on a frequency-by-frequency basis, and hence we are applying a matrix norm instead of a system norm.

21 This input is not unique, and there are in general infinitely many input trajectories that bring the system from $x(0)$ to $x(t_1)$; see [3]. The particular input trajectory in (1.61) minimizes the cost $< u, u >= \int_{t_0}^{t_1} u^T(t)u(t)dt$.

For discrete-time dynamical systems $x_{k+1} = Ax_k + Bu_k$, criteria for controllability are very similar to those for continuous time. However, one will in general not be able to bring the system to an arbitrary new state over an arbitrary short time period – one must allow for n timesteps to pass before an arbitrary $x(t_1)$ can be achieved. Similarly, the discrete-time version of the Gramian matrix is calculated using summing rather than the integration in (1.59).

1.5.6 Observability

Definition 1.2 The continuous-time dynamical system $\dot{x} = Ax + Bu, y = Cx + Du$ (or the matrix pair (C, A)) is termed *observable* if, for any $t_1 > 0$, the initial state $x(0)$ can be determined from the time history of the input $u(t)$ and the output $y(t)$ over the time interval $t \in [0, t_1]$.

Zhou et al. [11] prove that the following are equivalent:

- (C, A) is observable
- The Gramian matrix

$$W_o(t) := \int_0^t e^{A^T\tau} C^T C e^{A\tau} d\tau \tag{1.62}$$

  is positive definite for any $t > 0$.
- The observability matrix

$$\mathcal{O} := \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{1.63}$$

  has full column rank, where n is the number of states.
- The matrix $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ has full column rank for all values of the complex-valued scalar λ.
- For any eigenvalue λ and corresponding left eigenvector q of A (i.e. $Aq = \lambda q$), $Cq \neq 0$.
- The eigenvalues of $A + LC$ can be freely assigned – with the only restriction that complex eigenvalues must appear in conjugate pairs – by a suitable choice of L.

For discrete-time dynamical systems $x_{k+1} = Ax_k + Bu_k$, $y_k = Cx_k + Du_k$, criteria for observability are very similar to those for continuous time. However, one will in general not be able to determine the state at $t = 0$ by observing inputs and outputs over an arbitrary short time – one must in general allow for n timesteps to pass before $x(0)$ can be determined. Similarly, the discrete-time version of the Gramian matrix is calculated using summing rather than the integration in (1.62).

1.5.7 Some Comments on Controllability and Observability

Although controllability and observability in general are desirable properties, their relationship with achievable control performance is easily exaggerated. For instance,

- An uncontrollable state may cause no problem in achieving acceptable control, if that state is unrelated or only weakly related to the control objective.
- If an uncontrollable state is asymptotically stable, its effect on the measured variables will die out over time – since it is not excited by the manipulated variables. Note that this observation does not hold if the state is "controllable" from (and hence can be excited by) the disturbances.
- If a state is unobservable, it does not affect the measured variables. Hence, if the measurements reflect the control objective – and the state is stable – it is of little relevance for control quality.

The aforementioned points illustrate that good control can be achievable even though some states are not controllable and/or observable. In addition, there is no guarantee that good control can be achieved even if all states are controllable and observable:

- Controllability guarantees that any state x_1 can be reached at time $t_1 > t_0$. However, what happens before or after time t_1 is not specified.
  1. Large excursions in the state values may happen before or after time t_1.
  2. It may not be possible to maintain the state at x_1 at steady state.
  3. Bringing the state to x_1 at t_1 may require excessively large inputs.
- The ability to freely assign the eigenvalues of $(A + BK)$ does not necessarily mean fast control is achievable
  1. The state may not be known with high precision, in which case the appropriate feedback will be uncertain.
  2. Fast control generally implies use of large manipulated variables, which may not be possible if the manipulated variables are constrained (which, in practice, they always are).
- If state estimation becomes very fast, the estimator essentially approaches high-order differentiation of the measured variables. This will amplify measurement noise. Only if all states are directly measurable does it make much sense with very fast state estimation. This might be the case for some motion control problems, but essentially never happens in process control. In practice, very fast state estimation is therefore often not desirable.
- There may be bandwidth limitations that cannot be found by studying the matrix pairs (C, A) and (A, B) in isolation. For instance, in order to find RHP zeros, the entire state-space model (or transfer function matrix) is required, and analyzing the whether constraints are likely to cause problems requires information about expected or allowable range of variation of different variables.

In [9], a simple example with four water tanks in series is presented, where the control objective is to control the temperature in all tanks by changing the temperature

of the water flowing into the first tank. The system is controllable but displays many of the problems indicated above. Indeed, the systems theory concept *controllability* should be used with some care when discussing with operators and control practitioners in industry – due to the weak link between the controllability property and the achievable quality of control. In industrial parlance, a statement like "this plant is not controllable" will typically mean that it is not possible to achieve acceptable control performance for the plant – or at least that the staff at the plant has been unable to achieve this. Skogestad and Postlethwaite therefore use the terms *state controllability* and *state observability* when referring to the system theoretic concepts and use the term *controllability* (alone) when referring to the ability to achieve acceptable control performance. This use of the term *controllability* actually has a long history; see, e.g. Ziegler and Nichols [12]. In this book, the term *controllability* may be used in both meanings, but it is hopefully clear from context what is meant.

Assuming we have a minimal model of the plant, the properties of *stabilizability* and *detectability* are (in contrast to controllability and observability) necessary criteria for stabilizing an unstable plant – and hence necessary for acceptable control performance (however lax the performance criteria applied). These two properties will be addressed next.

1.5.8 Stabilizability

Definition 1.3 The continuous-time dynamical system $\dot{x} = Ax + Bu$ (or, the matrix pair (A, B)) is *stabilizable* if there exists a state feedback $u = Fx$ such that the resulting closed-loop system is stable.

The following are equivalent criteria for stabilizability:

- There exists a feedback $u = Fx$ such that $A + BF$ is stable.
- The matrix $[A - \lambda I \qquad B]$ has full row rank for all values of the complex-valued scalar λ such that $Re(\lambda) > 0$.
- For any eigenvalue λ such that $Re(\lambda) > 0$ and corresponding left eigenvector m of A, $m^*B \neq 0$.

For discrete-time systems, the only difference is that we have to consider $abs(\lambda) > 1$ instead of $Re(\lambda) > 0$.

Zhou et al. [11] argue that a more appropriate name for this property would be *state feedback stabilizability*, since (state feedback) stabilizability is not sufficient to guarantee that it is possible to stabilize the system using feedback from the *outputs*. However, if the system is both (state feedback) stabilizable and detectable (see below), the system can indeed be stabilized by output feedback.

1.5.9 Detectability

Definition 1.4 The continuous-time dynamical system $\dot{x} = Ax + Bu, y = Cx + Du$ (or the matrix pair (C, A)) is termed *detectable* if there exists a matrix L such that $A + LC$ is stable.

The following are equivalent criteria for detectability:

- There exists a matrix L such that $A + LC$ is stable.
- The matrix $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ has full column rank for all values of the complex-valued scalar λ such that $Re(\lambda) > 0$.
- For any eigenvalue λ such that $Re(\lambda) > 0$ and corresponding left eigenvector q of A (i.e. $Aq = \lambda q$), $Cq \neq 0$.

For discrete-time systems, the only difference is that we have to consider $abs(\lambda) > 1$ instead of $Re(\lambda) > 0$.

1.5.10 Hidden Modes

When calculating the transfer function from a state-space model, any unobservable or uncontrollable modes will cancel and will not be reflected in the transfer function. The canceled modes are called *hidden modes*, as these modes do not affect the dynamic relationship between inputs and outputs. It follows that in order to be able to stabilize a system with feedback, any hidden modes must be *stable*, which corresponds to the requirement that *all unstable states must be both stabilizable and detectable*.

1.5.11 Internal Stability

A system is internally stable if the injection of bounded signals anywhere in the system leads to bounded responses everywhere. For analyzing internal stability of a simple feedback loop such as the one in Figure 1.14, it suffices to consider injection (addition) of a signal d_1 to the signal going from K to G, and a signal d_2 to the signal going from G to K. The transfer function from d_2 to y is $S = (I + GK)^{-1}$, whereas the transfer function from r (not shown in the figure) to y is $T = I - S$, and verifying stability from d_2 to y also verifies stability from r to y.

When verifying internal stability, it is necessary to assume that none of the individual blocks in the system (in this case K and G) contain any hidden unstable modes – and this must be separately verified. We are here concerned with verifying that the feedback interconnection does not result in any hidden unstable modes.

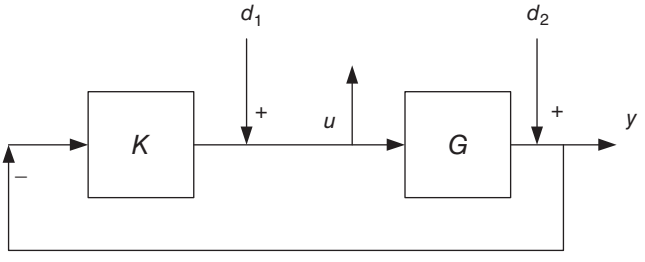

Figure 1.14 A simple feedback loop with input and output disturbances.

Example 1.1 Consider a case with $G(s) = \frac{10(10s+1)}{(5s-1)}$ and $K(s) = \frac{k(5s-1)}{s(10s+1)}$. The loop gain $G(s)K(s) = \frac{10k}{s}$, and it would appear that we have 90° phase margin irrespective of the value of k, and k can thus be adjusted to give any desired bandwidth. The transfer function from d_2 to y is $S(s) = \frac{s}{s+10k}$, which is clearly stable. However, we observe that whereas G and K together has three modes, S can be described with only one mode – two modes have been canceled. The transfer function from d_1 to y is $GS_I = SG = \frac{10s(10s+1)}{(s+10k)(5s-1)}$, which is unstable (for any k)! In practice, we must allow for disturbances entering anywhere in the system, and this closed-loop system is unacceptable since it is not *internally* stable even though it is stable from d_2 (or r) to y.

We note that the problems arise from canceling a pole in the RHP, and cancelation of the pole in the LHP does not lead to any particular problem.

Assigning the following state-space representations to $G(s)$ and $K(s)$:

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]; \quad K(s) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right]$$

and with negative feedback as in Figure 1.14, tedious but straightforward manipulations lead to

$$\begin{bmatrix} \dot{x}_G \\ \dot{x}_K \end{bmatrix} = \overbrace{\begin{bmatrix} A - B(I + D_K D)^{-1} D_K C & B(I + D_K D)^{-1} D_K \\ -B_K(I + DD_K)^{-1} & A_K - B_K(I + DD_K)^{-1} D C_K \end{bmatrix}}^{\tilde{A}} \begin{bmatrix} x_G \\ x_K \end{bmatrix} + \begin{bmatrix} B(I + D_K D)^{-1} & -B(I + D_K D)^{-1} D_K \\ -B_K(I + DD_K)^{-1} D & -B_K(I + DD_K)^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{1.64}$$

where x_G are the states in $G(s)$ and x_K are the states in $K(s)$. The stability of the overall system depends on the matrix $\tilde{A}$, which may be expressed as:

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & A_K \end{bmatrix} + \begin{bmatrix} B \\ B_K \end{bmatrix} \begin{bmatrix} -D & -I \\ I & -D_K \end{bmatrix}^{-1} \begin{bmatrix} C & C_K \end{bmatrix} \tag{1.65}$$

The internal stability of the closed-loop system may thus be determined from the eigenvalues of $\tilde{A}$. We note that a prerequisite for internal stability is that the matrix $\tilde{A}$ is well defined, i.e. that the matrix

$$\begin{bmatrix} -D & -I \\ I & -D_K \end{bmatrix}$$

is invertible (full rank). This is often stated as the requirement that the closed-loop feedback system should be *well posed*. Note that the closed loop is always well posed if $G(s)$ is strictly proper, i.e. if $D = 0$.

Alternatively, internal stability may be checked by checking all four closed-loop transfer functions in Figure 1.14:

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} (I + KG)^{-1} & -K(I + GK)^{-1} \\ G(I + KG)^{-1} & (I + GK)^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{1.66}$$

Only if there is no pole-zero cancelation between G and K in the RHP does it suffice to check the stability of only one of these transfer functions.

1.5.12 Coprime Factorizations

Coprime factorizations may at first seem a little daunting. However, the formulas for coprime factorizations are straightforward, and an important use is in the parametrization of all stabilizing controllers that are presented the next section.

A right coprime factorization of $G(s)$ is given by $G = NM^{-1}$, if there exist stable X_r and Y_r such that M and N are both stable and fulfill

$$\begin{bmatrix} X_r & Y_r \end{bmatrix} \begin{bmatrix} M \\ N \end{bmatrix} = I$$

Similarly, a left coprime factorization of $G(s)$ is given by $G = \tilde{M}^{-1}\tilde{N}$, if there exist stable X_l and Y_l such that $\tilde{M}$ and $\tilde{N}$ are both stable and fulfill

$$\begin{bmatrix} \tilde{M} & \tilde{N} \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \end{bmatrix} = I$$

A coprime factorization may be found from any stabilizing state feedback gain F and stabilizing observer gain L (such that both $A + BF$ and $A + LC$ are stable), using the formulas [11]:

$$\begin{bmatrix} M & -Y_l \\ N & X_l \end{bmatrix} = \left[\begin{array}{c|cc} A+BF & B & -L \\ \hline F & I & 0 \\ C+DF & D & I \end{array}\right] \tag{1.67}$$

$$\begin{bmatrix} X_r & Y_r \\ -\tilde{N} & \tilde{M} \end{bmatrix} = \left[\begin{array}{c|cc} A+LC & -(B+LD) & L \\ \hline F & I & 0 \\ C & -D & I \end{array}\right] \tag{1.68}$$

We observe that $M/\tilde{M}$ must have as RHP zeros all RHP poles of G, whereas $N/\tilde{N}$ contain all RHP zeros of G.

Clearly, the coprime factorizations are nonunique, since the stabilizing gains F and L are not unique. Any coprime factorization (with corresponding X_l, Y_l, X_r, Y_l) can be used for the parametrization of all stabilizing controllers. However, there are particular choices of coprime factorizations that have special uses. Before these particular coprime factorizations are presented, we will need the definition of a *conjugate system*.

Definition 1.5 ***Conjugate system*** The conjugate system of $G(s)$ is defined as:

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

The conjugate system of $G(s)$ is sometimes also termed the *para-hermitian conjugate* of $G(s)$.

1.5.12.1 Inner–Outer Factorization

Definition 1.6 ***Inner function*** A transfer function matrix $W_I(s)$ is called inner if $W_I(s)$ is stable and $W_I^* W_I = I$, and co-inner if $W_I W_I^* = I$.

Note that W_I does not need to be square, and that W_I is inner if W_I^T is co-inner and vice versa.

Definition 1.7 ***Outer function*** A transfer function matrix $W_O(s)$ is called outer if $W_O(s)$ is stable and has full row rank in the open RHP.

Clearly, a transfer function matrix cannot be outer if it has more rows than columns, and in order to be an outer function it cannot have any zeros in the open RHP.

Inner–outer factorizations of stable transfer function matrices may be found by factoring out RHP zeros using Blaschke products, as explained in Appendix D.

We will use the inner–outer factorization when assessing possible reduction in input usage obtainable by using feedforward from disturbances.

1.5.12.2 Normalized Coprime Factorization

A right coprime factorization $G = NM^{-1}$ is *normalized* if

$$M^*M + N^*N = I$$

i.e. if

$$\begin{bmatrix} M \\ N \end{bmatrix}$$

is an *inner* function. Similarly, a left coprime factorization is normalized if

$$\begin{bmatrix} \tilde{M} & \tilde{N} \end{bmatrix}$$

is co-inner. Note that $X_y \neq M^*$, $Y_r \neq N^*$, etc. Normalized coprime factorizations are unique up to the multiplication by a (constant) unitary matrix U. Normalized coprime factorizations are found from particular choices of the stabilizing gains F and L, see [11].

Normalized coprime factorizations allow a relatively simple and yet general uncertainty description in terms of uncertainty in the coprime factors. This uncertainty description is the starting point for H_∞ *robust loopshaping* design, a relatively simple robust controller design method. Readers are referred to [5, 6, 9] for details.

1.5.13 Parametrization of All Stabilizing Controllers

This section will present a parametrization of all stabilizing controller for a system $G(s)$. For open-loop stable systems this parametrization is commonly known as the Youla parametrization [10]. Naturally, we require not only input–output stability, but also internal stability of the closed-loop system.

1.5.13.1 Stable Plants

Consider an open-loop asymptotically stable plant, with plant model $G_m(s)$, which is assumed to be a perfect model of the true plant. Then, all feedback controllers K resulting in a stable closed-loop system can be parameterized as:

$$K = Q(I - G_m Q)^{-1} \tag{1.69}$$

where Q is any asymptotically stable system. This result holds also for nonlinear plants G_m. We see from Figure 2.21 that the model G_m in the nominal case (no model error) perfectly cancels the feedback signal, leading to the series interconnection of the two stable systems Q and G. This technique is used directly in the so-called *internal model control* (IMC), as addressed in Section 2.4.3.6.

1.5.13.2 Unstable Plants

For a stabilizable and detectable plant G with state-space realization

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

all stabilizing controllers can be represented as in Figure 1.15 [11] where Q is stable and $I + DQ(j\infty)$ is invertible. The dynamic interconnection J is given by:

$$J = \left[\begin{array}{c|cc} A + BF + LC + LDF & -L & B + LD \\ \hline F & 0 & I \\ -(C + DF) & I & -D \end{array}\right] \tag{1.70}$$

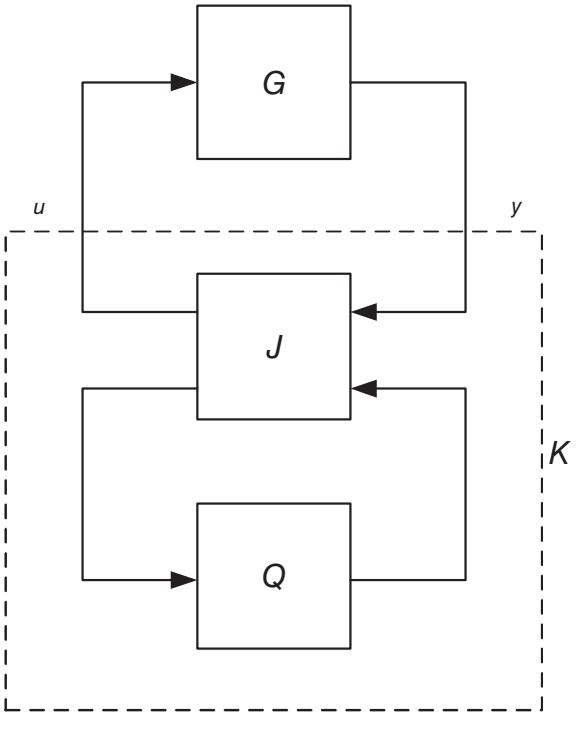

Figure 1.15 Representation of all stabilizing controllers.

where F is a stabilizing state feedback ($A + BF$ stable) and L is a stabilizing observer ($A + LC$ stable). Such stabilizing gains F and L can always be found for stabilizable and detectable systems, as explained above. Noting that stabilizing F and L also can be used to define a coprime factorization for G, the parametrization of all stabilizing controllers may equivalently be presented using coprime factors, such as in [9].

It can be verified that (1.70) results in the controller (1.69) if one chooses $F = 0$ and $L = 0$ – which is obviously possible for open-loop stable plants. Thus, as one should expect, the parametrization of all stabilizing controllers for stable plants is a special case of the parametrization for unstable plants.

Zhou et al. [11] show that the closed-loop transfer function from an external input w to an output z, for any internally stabilizing, proper controller, is an affine function of the free parameter Q, *i.e.* that $T_{wz} = T_{11} + T_{12}QT_{21}$ (where T_{ij} can be found by straightforward but tedious algebra). Controller design methods have been proposed that instead of searching directly for the controller, one searches only over stabilizing controllers, due to the simple affine relationship. The main drawback with such an approach is the difficulty of specifying a sufficiently flexible parametrization for Q – often an FIR description is used. One should also bear in mind that although nominal stability is guaranteed by choosing a stable Q, there is no inherent robustness guarantee.

1.5.14 Hankel Norm and Hankel Singular Values

For open-loop stable systems, the infinite time controllability Gramian (or just "controllability Gramian," for short) in (1.59) can be obtained by setting the upper limit of the integration to infinity. A simpler way of finding it is to solve the Lyapunov equation:

$$AW_c + W_cA^T + BB^T = 0 \tag{1.71}$$

Similarly, the infinite time observability Gramian is found from

$$A^TW_o + W_oA + C^TC = 0 \tag{1.72}$$

For discrete-time models, the corresponding equations are

$$AW_cA^T - W_c + BB^T = 0 \tag{1.73}$$

and

$$A^TW_oA - W_o + C^TC = 0 \tag{1.74}$$

It is hopefully clear that the continuous-time state-space model is used in (1.71) and (1.72), while the discrete-time state-space model is used in (1.73) and (1.74). Note that since the controllability and observability Gramians correspond to solutions to infinite-horizon integrals[22] (from (1.59) and (1.62), respectively), they are only defined for asymptotically stable systems.

22 Or infinite sums in the case of discrete-time Gramians.

In somewhat imprecise terms, it may be stated that the controllability Gramian measures how strongly the inputs affect the states, whereas the observability Gramian measures how strongly the outputs are affected by the states. The Gramians are affected by similarity transformations, but their product $H = W_c W_o$ is not affected by similarity transforms. For minimal models, there is a particular state representation for which $W_c = W_o$, and the corresponding state-space model is termed a *balanced realization* of the model. For many numerical calculations it may be an advantage to use the balanced realization of the model, as often the results will be less sensitive to numerical error when this realization is used.

However, although H is independent of similarity transforms, it is affected by scaling of inputs and outputs, and for the uses we will have for H it is therefore advisable to scale the model as described in Section 1.2.9. The square roots of the eigenvalues of H are known as the *Hankel singular values*, and the largest Hankel singular value is the same as the *Hankel norm*. The Hankel norm can be seen as a measure of how strongly past inputs affect future outputs [9]. The most common use of the Hankel norm is in model reduction. However, we will be using it for selection and pairing of inputs and outputs.

Problems

1.1 Consider the system

$$\dot{x} = \begin{bmatrix} -1.4 & -3.5 \\ -3.5 & -1.5 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u$$
$$y = \begin{bmatrix} -1 & 3 \end{bmatrix} x$$

(a) Find the poles and (transmission) zeros of the system?
(b) Is the system controllable and observable?
(c) Find a minimal realization for the system.
(d) What considerations to we have to take before we decide to remove hidden modes from a system model?

1.2 Figure 1.16 shows the Bode plot for the open-loop transfer function of some control loop (as given by Matlab[23]). It is known that the open-loop system is asymptotically stable and has no zeros or poles at the origin.
(a) What is the correct phase in the Bode diagram at low frequencies for this system?
(b) Will this system be stable in closed loop?

23 One of the consequences of using the standard Matlab function is that the decibel scale is used for magnitude, not the $\log_{10}$ scale used throughout the book. A gain (magnitude) of 1 corresponds to 0 dB.

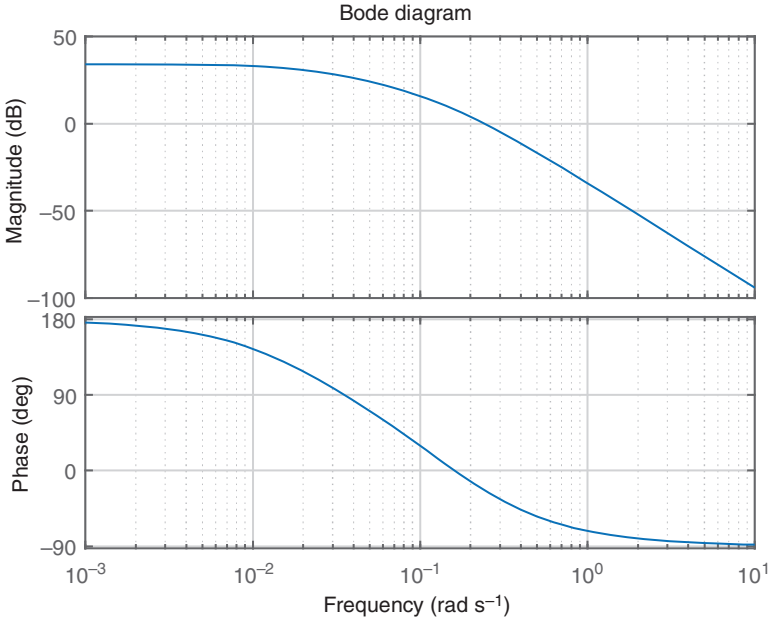

Figure 1.16 Bode diagram for a system – as given by Matlab.

1.3 Consider the system

$$y(s) = 1/(s+0.1)(s-2)\begin{bmatrix} s+2 & s \\ s+4 & s+1 \end{bmatrix} u(s)$$

(a) What is meant by the multiplicity of a pole of a dynamical system?
(b) What are the poles of this system?
(c) Calculate the transmission zero(s) of the system.
(d) Calculate the corresponding zero direction(s) at the output of the plant.

1.4 A system has transfer function $g(s) = \frac{10(s+1)}{(s+20)(s-0.1)}$. It is proposed to control this system with a controller given by $k(s) = \frac{0.1(s-0.1)}{s}$.
(a) Find the closed-loop transfer function from reference to measurement with the proposed controller. Is the system stable from r to y?
(b) Is the system *internally* stable?

1.5 Consider again a level control problem such as that shown in Figure 1.3. The flow controller is fast and accurate, but the liquid level is found to be oscillating – also when the inlet flowrate is steady. The level controller can be described by:

$$\Delta u = \frac{k}{p}\Delta y$$

where $\Delta u = u_k - u_{k-1}$ (i.e. the change in controller output between two sampling intervals), and $\Delta y = y_{ref} - y$, where y_{ref} denotes the reference (desired level). Explain why the level control is oscillating.
Hint: What is the continuous-time equivalent of the controller dynamics?

References

1 Åström, K.J. and Wittenmark, B. (1984). *Computer Controlled Systems: Theory and Design*. Englewood Cliffs, NJ: Prentice-Hall.
2 Balchen, J.G. and Mummé, K.I. (1988). *Process Control: Structures and Applications*. Glasgow: Blackie Academic & Professional.
3 Callier, F.M. and Desoer, C.A. (1991). *Linear Systems Theory*. New York: Springer-Verlag.
4 Doyle, J., Francis, B., and Tannenbaum, A. (1992). *Feedback Control Theory*. New York: Macmillan Publishing Co.
5 Glover, K. and McFarlane, D. (1986). Robust stabilization of normalized coprime factor plant descriptions with h_∞ bounded uncertainty. *IEEE Transactions on Automatic Control* 34: 821–830.
6 McFarlane, D. and Glover, K. (1990). *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*, *Lecture Notes in Control and Information Sciences*, vol. 138. Springer-Verlag.
7 MacFarlane, A.G.J. and Karcanias, N. (1976). Poles and zeros of linear multivariable systems: a survey of algebraic, geometric and complex variable theory. *International Journal of Control* 24: 33–74.
8 Morari, M. and Zafiriou, E. (1989). *Robust Process Control*. Englewood Cliffs, NJ: Prentice-Hall.
9 Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design*. Chichester: Wiley.
10 Youla, D.C., Jabr, H.A., and Bongiorno, J.J. (1976). Modern Wiener-HOPF design of optimal controllers, Part II: the multivariable case. *IEEE Transactions on Automatic Control* 21 (3): 319–338.
11 Zhou, K., Doyle, J.C., and Glover, K. (1996). *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice-Hall.
12 Ziegler, J.G. and Nichols, N.B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME* 64: 759–768.