Lecture Notes

Control Systems Theory and Design

Herbert Werner

Version: November 18th, 2020

Contents

Chapter 1

Introduction

Classical techniques for analysis and design of control systems, which employ the concepts of frequency response and root locus, have been used successfully for more than five decades in a vast variety of industrial applications. These methods are based on transfer function models of the plant to be controlled, and allow an efficient design of controllers for single-input single-output systems. However, when it comes to more complex systems with several input and output variables, classical techniques based on transfer functions tend to become tedious and soon reach their limits. For such applications, the so-called modern control techniques developed in the 1960s and 70s turned out to be more suitable. They are based on state space models of the plant rather than on transfer function models. One major advantage of state space models is that multi-input multi-output systems can be treated in much the same way as single-input single-output systems. This course provides an introduction to the modern state space approach to control and its theoretical foundations.

State space models are introduced in Chapter 2; to facilitate the understanding of the basic concepts, the discussion is initially limited to single-input single-output systems. The main idea - introduced in Chapters 2 through 5 - is to break the controller design problem up in two subproblems: (i) the problem of designing a static controller that feeds back internal variables - the *state variables* of the plant, and (ii) the problem of obtaining a good estimate of these internal state variables. It is shown that these two problems are closely related, they are called *dual problems*. Associated with state feedback control and state estimation (the latter is usually referred to as *state observation*) are the concepts of controllability and observability. In Chapter 6 these concepts are extended to multivariable systems, and it will be seen that this extension is in most aspects rather straightforward, even though the details are more complex.

A further important topic, discussed in Chapter 7, is the digital implementation of controllers. It is shown that most of the methods and concepts developed for continuous-time systems have their direct counterparts in a discrete-time framework. This is true for trans-

fer function models as well as for state space models.

The third major issue taken up in this course is about obtaining a model of the plant to be controlled. In practice, a model is often obtained from experimental data; this approach is known as *system identification*. In Chapter 8 the basic ideas of system identification are introduced; identification of transfer function models as well as state space models are discussed. In Chapter 9 the issue of reducing the dynamic order of a model is addressed - this can be important in applications where complex plant models lead to a large number of state variables.

Most chapters are followed by a number of exercise problems. The exercises play an important role in this course. To encourage student participation and active learning, derivations of theoretical results are sometimes left as exercises. A second objective is to familiarize students with the state-of-the-art software tools for modern controller design. For this reason, a number of analysis and design problems are provided that are to be solved using MATLAB and Simulink. MATLAB code and Simulink models for these problems can be downloaded from the web page of this course. A complete design exercise that takes up all topics of this course - identification of a model, controller and observer design and digital implementation - is presented in Chapter 10.

This course assumes familiarity with elementary linear algebra, and with engineering applications of some basic concepts of probability theory and stochastic processes, such as white noise. A brief tutorial introduction to each of these fields is provided in the Appendix. The Appendix also provides worked solutions to all exercises. Students are encouraged to try to actively solve the exercise problems, before checking the solutions. Some exercises that are more demanding and point to more advanced concepts are marked with an asterisk.

The exercise problems and worked solutions for this course were prepared by Martyn Durrant MEng.

Chapter 2

State Space Models

In this chapter we will discuss linear state space models for single-input single-output systems. The relationship between state space models and transfer function models is explored, and basic concepts are introduced.
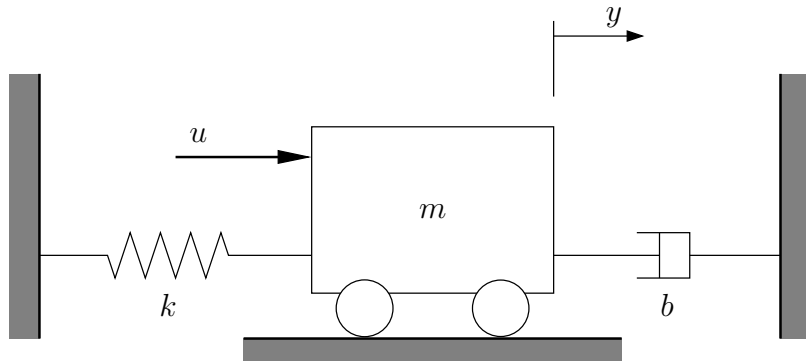
Figure 2.1: Spring-mass-damper system

We begin with an example. Consider the spring-mass-damper system shown in Fig.2.1 with mass m, spring constant k and damping coefficient b. The equation of motion is

$$\ddot{y}(t) + \frac{b}{m}\,\dot{y}(t) + \frac{k}{m}\,y(t) \;=\; \frac{1}{m}\,u(t). \tag{2.1}$$

where $u(t)$ is an external force acting on the mass and $y(t)$ is the displacement from equilibrium. From the equation of motion, the transfer function is easily found to be

$$G(s) = \frac{\frac{1}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} \tag{2.2}$$

Introducing the velocity $v = \dot{y}$ as a new variable, it is straightforward to check that the second order differential equation (2.1) can be rewritten as a first order vector differential

equation

$$\begin{bmatrix} \dot{y}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} \begin{bmatrix} y(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t) \tag{2.3}$$

together with an output equation

$$y(t) = [1 \ 0] \begin{bmatrix} y(t) \\ v(t) \end{bmatrix} \tag{2.4}$$

The model (2.3), (2.4) describes the same dynamic properties as the transfer function (2.2). It is a special case of a state space model

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.5}$$
$$y(t) = Cx(t) + Du(t) \tag{2.6}$$

In general, a system modelled in this form can have m inputs and l outputs, which are then collected into an input vector $u(t) \in \mathbb{R}^m$ and an output vector $y(t) \in \mathbb{R}^l$. The vector $x(t) \in \mathbb{R}^n$ is called *state vector*, $A \in \mathbb{R}^{n \times n}$ is the *system matrix*, $B \in \mathbb{R}^{n \times m}$ is the *input matrix*, $C \in \mathbb{R}^{l \times n}$ is the *output matrix* and $D \in \mathbb{R}^{l \times m}$ is the *feedthrough matrix*. Equation (2.5) is referred to as the *state equation* and (2.6) as the *output equation*.

The spring-mass-damper system considered above has only one input and one output. Such systems are called single-input single-output (SISO) systems, as opposed to multi-input multi-output (MIMO) systems. For a SISO system the input matrix B degenerates into a column vector b, the output matrix C into a row vector c and the feedthrough matrix D into a scalar d. Initially, we will limit the discussion of state space models to SISO systems of the form

$$\dot{x}(t) = Ax(t) + bu(t) \tag{2.7}$$
$$y(t) = cx(t) + du(t) \tag{2.8}$$

In the above example the direct feedthrough term d is zero - this is generally true for physically realizable systems, as will be discussed later.

2.1 From Transfer Function to State Space Model

The state space model (2.3), (2.4) for the second order system in the above example was constructed by introducing a single new variable. We will now show a general method for constructing a state space model from a given transfer function. Consider the nth order linear system governed by the differential equation

$$\frac{d^n}{dt^n}y(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}y(t) + \ldots + a_1\frac{d}{dt}y(t) + a_0y(t) =$$
$$b_m\frac{d^m}{dt^m}u(t) + b_{m-1}\frac{d^{m-1}}{dt^{m-1}}u(t) + \ldots + b_1\frac{d}{dt}u(t) + b_0u(t) \quad (2.9)$$

where we assume for simplicity that the system is strictly proper, i.e. $n > m$ (the case of bi-proper systems is discussed at the end of this section). A transfer function model of this system is

$$Y(s) = G(s)U(s) \tag{2.10}$$

where

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} = \frac{b(s)}{a(s)} \tag{2.11}$$

$$u(t) \longrightarrow \boxed{\frac{b(s)}{a(s)}} \longrightarrow y(t)$$
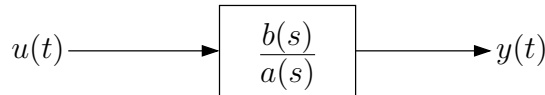
Figure 2.2: Transfer function model

The transfer function model of this system is shown in Fig. 2.2. In order to find a state space model for this system, we will first construct a simulation model by using integrator blocks. For this purpose, we split the model in Fig. 2.2 into two blocks as shown in Fig. 2.3, and let $v(t)$ denote the fictitious output of the filter $1/a(s)$.

$$u(t) \longrightarrow \boxed{\frac{1}{a(s)}} \xrightarrow{v(t)} \boxed{b(s)} \longrightarrow y(t)$$
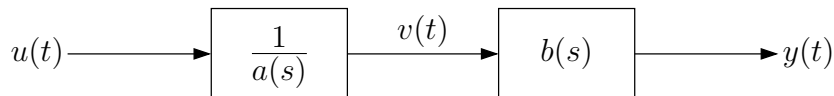
Figure 2.3: Transfer function model

From Fig. 2.3 the input and output signals can be expressed in terms of the new variable as $U(s) = a(s)V(s)$ and $Y(s) = b(s)V(s)$, or in time domain

$$u(t) = \frac{d^n}{dt^n} v(t) + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} v(t) + \ldots + a_1 \frac{d}{dt} v(t) + a_0 v(t) \tag{2.12}$$

and

$$y(t) = b_m \frac{d^m}{dt^m} v(t) + b_{m-1} \frac{d^{m-1}}{dt^{m-1}} v(t) + \ldots + b_1 \frac{d}{dt} v(t) + b_0 v(t) \tag{2.13}$$

From (2.12) the signal $v(t)$ can be generated by using a chain of integrators: assume first that $d^n v(t)/dt^n$ is somehow known, then integrating n times, introducing feedback loops as shown in the lower half of Fig. 2.4 (for $n = 3$ and $m = 2$) and adding the input signal $u(t)$ yields the required signal $d^n v(t)/dt^n$. The output signal $y(t)$ can then be constructed as a linear combination of $v(t)$ and its derivatives according to (2.13), as shown in the upper half of Fig. 2.4.

An implicit assumption was made in the above construction: namely that the initial values at $t = 0$ in (2.12) and (2.13) are zero. In the model in Fig. 2.4 this corresponds to the assumption that the integrator outputs are zero initially. When transfer function models
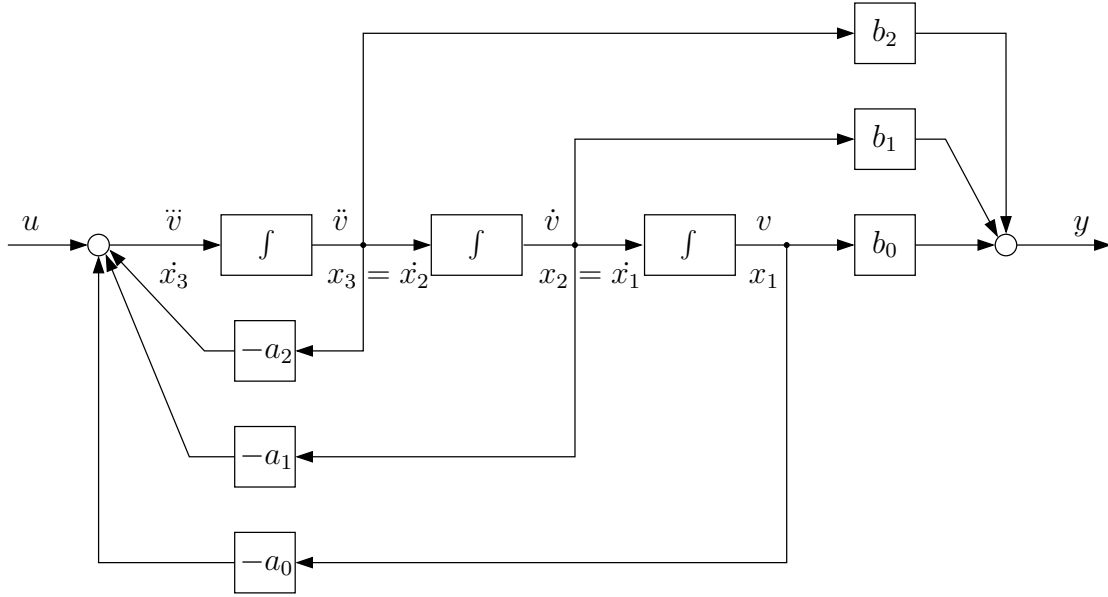
Figure 2.4: Simulation model for system (2.9) when $n = 3$ and $m = 2$

are used, this assumption is usually made, whereas state space models allow non-zero initial conditions to be taken into account.

State Variables

Before we derive a state space model from the simulation model in Fig. 2.4, we introduce the concept of *state variables*. For a given system, a collection of internal variables

$$x_1(t), \ x_2(t), \ \ldots, \ x_n(t)$$

is referred to as state variables if they completely determine the state of the system at time t. By this we mean that if the values of the state variables at some time, say t_0, are known, then for a given input signal $u(t)$ where $t \geq t_0$ all future values of the state variables can be uniquely determined. Obviously, for a given system the choice of state variables is not unique.

We now return to the system represented by the simulation model in Fig. 2.4. Here the dynamic elements are the integrators, and the state of the system is uniquely determined by the values of the integrator outputs at a given time. Therefore, we choose the integrator outputs as state variables of the system, i.e. we define

$$x_1(t) = v(t), \ x_2(t) = \frac{d}{dt} v(t), \ldots, \ x_n(t) = \frac{d^{n-1}}{dt^{n-1}} v(t)$$

The chain of integrators and the feedback loops determine the relationship between the state variables and their derivatives. The system dynamics can now be described by a set

of first order differential equations

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = x_3$$
$$\vdots$$
$$\dot{x}_{n-1} = x_n$$
$$\dot{x}_n = u - a_0 x_1 - a_1 x_2 - \ldots - a_{n-1} x_n$$

The last equation is obtained at the input summing junction in Fig. 2.4. The first order differential equations can be rewritten in vector form: introduce the state vector

$$x(t) = [x_1(t) \ \ x_2(t) \ \ \ldots x_n(t)]^T$$

then we have

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t) \quad (2.14)$$

This equation has the form of (2.7) and is a state equation of the system (2.9). The state equation describes the dynamic properties of the system. If the system is physically realizable, i.e. if $n > m$, then from (2.13) and Fig. 2.4, the output signal is a linear combination of the state variables and can be expressed in terms of the state vector as

$$y(t) = [b_0 \ \ b_1 \ \ \ldots \ \ b_{n-1}] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.15)$$

This equation has the form of (2.8) and is the output equation of the system (2.9) associated with the state equation (2.14). The system matrices (vectors) A, b and c of this state space model contain the same information about the dynamic properties of the system as the transfer function model (2.11). As mentioned above, the choice of state variables for a given system is not unique, and other choices lead to different state space models for the same system (2.9). For reasons that will be discussed later, the particular form (2.14) and (2.15) of a state space model is called *controller canonical form*. A second canonical form, referred to as *observer canonical form*, is considered in Exercise 2.7.

Note that for bi-proper systems, i.e. systems where $n = m$, the state equation will still be as shown in (2.14), but the output equation will be different from (2.15): There will be a feedthrough term $du(t)$ in (2.15); if the example in Figure 2.4 was bi-proper there would be a direct path with gain b_3 from d^3v/dt^3 to y. Moreover, the elements of the measurement vector c will not comprise the numerator coefficients of the bi-proper transfer function, but of the strictly proper remainder obtained after polynomial division. This is illustrated in Exercise 2.8.

2.2 From State Space Model to Transfer Function

We have seen how a particular state space model of a system can be constructed when its transfer function is given. We now consider the case where a state space model is given and we wish to find its transfer function. Thus, consider a system described by the state space model

$$\dot{x}(t) = Ax(t) + bu(t)$$
$$y(t) = cx(t) + du(t)$$

and assume again that the initial conditions are zero, i.e. $x(0) = 0$. Taking Laplace transforms, we have

$$sX(s) = AX(s) + bU(s)$$

and solving for $X(s)$ yields

$$X(s) = (sI - A)^{-1}bU(s) \tag{2.16}$$

where I denotes the identity matrix. Substituting the above in the Laplace transform of the output equation leads to

$$Y(s) = c(sI - A)^{-1}bU(s) + dU(s)$$

Comparing this with the transfer function model (2.10) yields

$$G(s) = c(sI - A)^{-1}b + d \tag{2.17}$$

Further insight into the relationship between transfer function and state space model can be gained by noting that

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

where adj (M) denotes the adjugate of a matrix M. The determinant of $sI - A$ is a polynomial of degree n in s. On the other hand, the adjugate of $sI - A$ is an $n \times n$ matrix whose entries are polynomials in s of degree less than n. Substituting in (2.17) and assuming $n > m$ (i.e. $d = 0$) gives

$$G(s) = \frac{c \, \text{adj}(sI - A)b}{\det(sI - A)}$$

The adjugate is multiplied by the row vector c from the left and by the column vector b from the right, resulting in a single polynomial of degree less than n. This polynomial is the numerator polynomial of the transfer function, whereas $\det(sI - A)$ is the denominator polynomial. The *characteristic equation* of the system is therefore

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

Note that the characteristic equation is the same when there is a direct feedthrough term $d \neq 0$. The values of s that satisfy this equation are the eigenvalues of the system matrix A. This leads to the important observation that the poles of the transfer function - which determine the dynamic properties of the system - can be found in a state space model as eigenvalues of the system matrix A.

2.3   Changing Eigenvalues Via State Feedback

A block diagram representing the state space model (2.7), (2.8) is shown in Fig. 2.5. Note that no graphical distinction is made between scalar signals and vector signals. The integrator block represents n integrators in parallel. When the relationship between state space models and transfer functions was discussed, we assumed $x(0) = 0$. From now on we will include the possibility of non-zero initial values of the state variables in the state space model. In Fig. 2.5 this is done by adding the initial values to the integrator outputs.
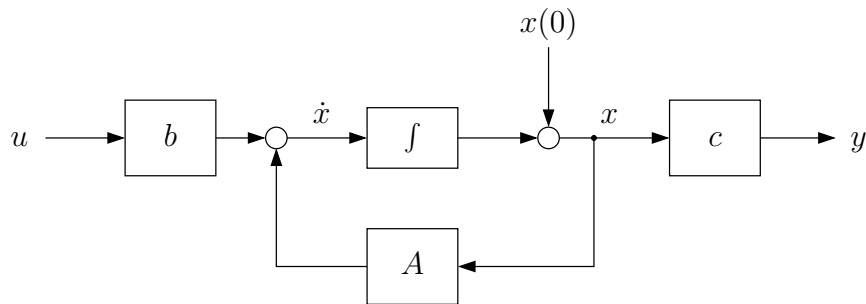
Figure 2.5: Block diagram of SISO state space model

The dynamic properties of this system are determined by the eigenvalues of the system matrix A. Suppose we wish to improve the dynamic behaviour - e.g. when the system is unstable or has poorly damped eigenvalues. If all state variables are measured, we can use them for feedback by taking the input as

$$u(t) = fx(t) + u_v(t) \tag{2.18}$$

where $f = [f_1 \ \ f_2 \ \ \ldots f_n]$ is a gain vector, and $u_v(t)$ is a new external input. This type of feedback is called *state feedback*, the resulting closed-loop system is shown in Fig. 2.6.

Substituting (2.18) in the state equation yields

$$\dot{x}(t) = Ax(t) + b(fx(t) + u_v(t))$$
$$= (A + bf)x(t) + bu_v(t)$$

Comparing this with the original state equation shows that as a result of state feedback the system matrix A is replaced by $A + bf$, and the control input $u(t)$ is replaced by $u_v(t)$. The eigenvalues of the closed-loop system are the eigenvalues of $A + bf$, and the freedom in choosing the state feedback gain vector f can be used to move the eigenvalues of the original system to desired locations; this will be discussed later (see also Exercise 2.6).
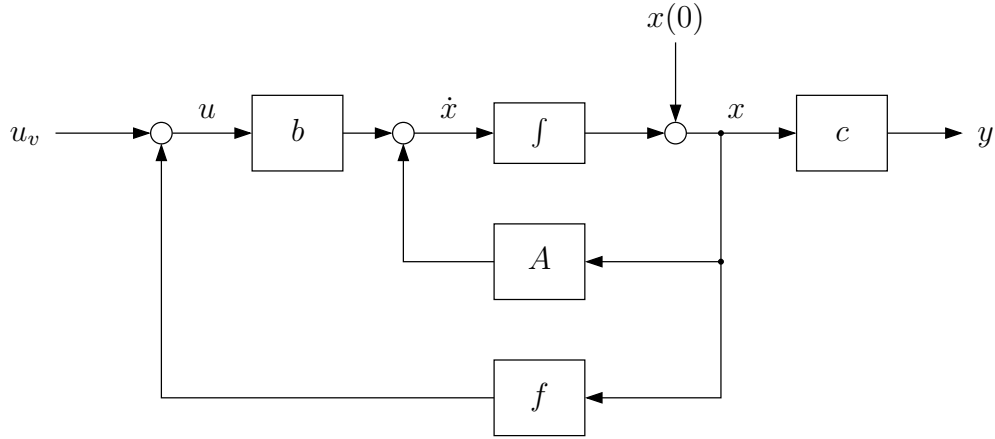
Figure 2.6: State feedback

2.4 Non-Uniqueness of State Space Models and Similarity Transformations

The state variables we chose for the spring-mass-damper system have a physical meaning (they represent position and velocity). This is often the case when a state space model is derived from a physical description of the plant. In general however state variables need not have any physical significance. The signals that represent the interaction of a system with its environment are the input signal $u(t)$ and the output signal $y(t)$. The elements of the state vector $x(t)$ on the other hand are internal variables which are chosen in a way that is convenient for modelling internal dynamics; they may or may not exist as physical quantities. The solution $x(t)$ of the forced vector differential equation $\dot{x}(t) = Ax(t) + bu(t)$ with initial condition $x(0) = x_0$ can be thought of as a trajectory in an n-dimensional *state space*, which starts at point x_0. The non-uniqueness of the choice of state variables reflects the freedom of choosing a coordinate basis for the state space. Given a state space model of a system, one can generate a different state space model of the same system by applying a coordinate transformation.

To illustrate this point, consider a system modelled as

$$\dot{x}(t) = Ax(t) + bu(t), \quad x(0) = x_0$$
$$y(t) = cx(t) + du(t)$$

A different state space model describing the same system can be generated as follows. Let T be any non-singular $n \times n$ matrix, and consider a new state vector $\tilde{x}(t)$ defined by

$$x(t) = T\tilde{x}(t)$$

Substituting in the above state space model gives

$$T\dot{\tilde{x}}(t) = AT\tilde{x}(t) + bu(t)$$

or

$$\dot{\tilde{x}}(t) = T^{-1}AT\tilde{x}(t) + T^{-1}bu(t)$$

and

$$y(t) = cT\tilde{x}(t) + du(t)$$

Comparing this with the original model shows that the model (A, b, c, d) has been replaced by a model $(\tilde{A}, \tilde{b}, \tilde{c}, d)$, where

$$\tilde{A} = T^{-1}AT, \quad \tilde{b} = T^{-1}b, \quad \tilde{c} = cT$$

Note that the feedthrough term d is not affected by the transformation, because it is not related to the state variables.

In matrix theory, matrices A and $\tilde{A}$ related by

$$\tilde{A} = T^{-1}AT$$

where T is nonsingular, are said to be *similar*, and the above state variable transformation is referred to as a *similarity transformation*. Similarity transformations do not change the eigenvalues, we have

$$\det(sI - \tilde{A}) = \det(sI - T^{-1}AT) = \det(T^{-1}(sI - A)T) = \det(sI - A)$$

In fact, it is straightforward to check that the models (A, b, c) and $(\tilde{A}, \tilde{b}, \tilde{c})$ have the same transfer function (see Exercise 2.9).

To see that T represents a change of coordinate basis, write $x = T\tilde{x}$ as

$$x = \begin{bmatrix} t_1 & t_2 & \dots t_n \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = t_1\tilde{x}_1 + t_2\tilde{x}_2 + \dots + t_n\tilde{x}_n$$

where t_i is the ith column of T. Thus, $\tilde{x}_i$ is the coordinate of x in the direction of the basis vector t_i. In the original coordinate basis, the vector x is expressed as

$$x = e_1x_1 + e_2x_2 + \dots + e_nx_n$$

where e_i is a vector with zeros everywhere except for the ith element which is 1.

We have thus seen that for a given transfer function model, by choosing a nonsingular transformation matrix T we can find infinitely many different but equivalent state space models. As a consequence, it is not meaningful to refer to *the state variables of a system*, but only of the state variables of a *particular state space model* of that system. Conversely, all state space models that are related by a similarity transformation represent the same system. For this reason, a state space model is also referred to as a particular *state space realization* of a transfer function model.

Diagonal Form

To illustrate the idea of a similarity transformation, assume that we are given a 3rd order state space model (A, b, c) and that the eigenvalues of A are distinct and real. Suppose we wish to bring this state space model into a form where the new system matrix is diagonal. We need a transformation such that

$$\tilde{A} = T^{-1}AT = \Lambda$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$. It is clear that the λ_i are the eigenvalues of A. To find the required transformation matrix T, note that $AT = T\tilde{A}$ or

$$A[t_1 \ t_2 \ t_3] = [t_1 \ t_2 \ t_3]\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Looking at this equation column by column, we find that

$$At_i = \lambda_i t_i, \quad i = 1, \ 2, \ 3$$

Therefore, the columns of T are the (right) eigenvectors of A. Since we assumed that the eigenvalues of A are distinct, the three eigenvectors are linearly independent and T is non-singular as required.

State space models with a diagonal system matrix are referred to as *modal canonical form*, because the poles of a system transfer function (the eigenvalues appearing along the diagonal) are sometimes called the *normal modes* or simply the *modes* of the system.

2.5 Solutions of State Equations and Matrix Exponentials

We now turn to the solution of the forced vector differential equation

$$\dot{x}(t) = Ax(t) + bu(t), \quad x(0) = x_0$$

First, we consider the scalar version of this problem

$$\dot{x}(t) = ax(t) + bu(t), \quad x(0) = x_0$$

where $x(t)$ is a single state variable and a is a real number. The solution to this problem can be found by separating variables: we have

$$\frac{d}{dt}\left(e^{-at}x\right) = e^{-at}(\dot{x} - ax) = e^{-at}bu$$

Integration from 0 to t and multiplication by e^{at} yields

$$x(t) = e^{at}x_0 + \int_0^t e^{a(t-\tau)}bu(\tau)d\tau$$

Crucial for finding the solution in this scalar case is the property

$$\frac{d}{dt}e^{at} = ae^{at}$$

of the exponential function. To solve the state equation when $x \in \mathbb{R}^n$, we would need something like

$$\frac{d}{dt}e^{At} = Ae^{At}$$

for a $n \times n$ matrix A. This leads to the definition of the *matrix exponential*

$$e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots \tag{2.19}$$

It can be shown that this power series converges. Differentiating shows that

$$\begin{aligned}\frac{d}{dt}e^{At} &= A + A^2t + \frac{1}{2!}A^3t^2 + \frac{1}{3!}A^4t^3 + \dots \\ &= A(I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots) \\ &= Ae^{At}\end{aligned}$$

as required. Incidentally, this also shows that $Ae^{At} = e^{At}A$ because A can be taken as a right factor in the second equation above.

The solution of the state equation can thus be written as

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}bu(\tau)d\tau$$

The *transition matrix* $\Phi(t)$ of a state space model is defined as

$$\Phi(t) = e^{At}$$

With this definition, the solution becomes

$$x(t) = \Phi(t)x_0 + \int_0^t \Phi(t-\tau)bu(\tau)d\tau \tag{2.20}$$

The first term on the right hand side is called the *zero-input response*, it represents the part of the system response that is due to the initial state x_0. The second term is called the *zero-initial-state response*, it is the part of the response that is due to the external input $u(t)$.

The frequency domain expression (2.16) of the solution was derived by assuming a zero initial state. When the initial state is non-zero, it is easy to verify that the Laplace transform of $x(t)$ is

$$X(s) = \Phi(s)x_0 + \Phi(s)bU(s) \tag{2.21}$$

where

$$\Phi(s) = \mathcal{L}[\Phi(t)\sigma(t)] = (sI - A)^{-1}$$

is the Laplace transform of the transition matrix. The second equation follows directly from comparing the expression obtained for $X(s)$ with the time-domain solution (2.20).

The definition of $\Phi(t)$ implies that this matrix function is invertible, we have

$$\Phi^{-1}(t) = e^{-At} = \Phi(-t)$$

Another useful property of $\Phi(t)$ was shown above, namely:

$$A\Phi(t) = \Phi(t)A$$

Stability

From the study of transfer function models we know that a system is stable if all its poles are in the left half plane. For a transfer function model, stability means that after a system has been excited by an external input, the transient response will die out and the output will settle to the equilibrium value once the external stimulus has been removed. Since we have seen that poles of a transfer function become eigenvalues of the system matrix A of a state space model, we might ask whether stability of a state space model is equivalent to all eigenvalues of A being in the left half plane. In contrast to transfer function models, the output of a state space model is determined not only by the input but also by the initial value of the state vector. The notion of stability used for transfer functions is meaningful for the part of the system dynamics represented by the zero-initial-state response, but we also need to define stability with respect to the zero-input response, i.e. when an initial state vector $x(0) \neq 0$ is driving the system state.

Definition 2.1 *An unforced system $\dot{x}(t) = Ax(t)$ is said to be stable if for all $x(0) = x_0$, $x_0 \in \mathbb{R}$ we have $x(t) \to 0$ as $t \to \infty$.*

Now consider the zero-input response

$$x(t) = e^{At}x_0$$

or in frequency domain

$$X(s) = (sI - A)^{-1}x_0$$

Assuming that the eigenvalues are distinct, a partial fraction expansion reveals that

$$x(t) = \phi_1 e^{\lambda_1 t} + \ldots + \phi_n e^{\lambda_n t}$$

where λ_i are the eigenvalues of A, and ϕ_i are column vectors that depend on the residual at λ_i. It is clear that $x(t) \to 0$ if and only if all eigenvalues have negative real parts. Thus, we

find that stability with respect to both zero-initial state response and zero-input response requires that all eigenvalues of A are in the left half plane.

The Cayley-Hamilton Theorem

We conclude this chapter with an important result on the representation of the transition matrix. The following is derived (for the case of distinct eigenvalues) in Exercise 2.2:

Theorem 2.1 (Cayley-Hamilton Theorem)

Consider a matrix $A \in \mathbb{R}^{n \times n}$. Let

$$\det(sI - A) = a(s) = s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0 = 0$$

be its characteristic equation. Then

$$A^n + a_{n-1}A^{n-1} + \ldots + a_1 A + a_0 I = 0$$

In the last equation the 0 on the right hand side stands for the $n \times n$ zero matrix. This theorem states that every square matrix satisfies its characteristic equation.

With the help of the Cayley-Hamilton Theorem we can express the matrix exponential e^{At} - defined as an infinite power series - as a polynomial of degree $n - 1$. Recall the definition

$$e^{At} = I + At + \frac{1}{2!}A^2 t^2 + \frac{1}{3!}A^3 t^3 + \ldots + \frac{1}{n!}A^n t^n + \ldots \tag{2.22}$$

From Theorem 2.1 we have

$$A^n = -a_{n-1}A^{n-1} - \ldots - a_1 A - a_0 I$$

and we can substitute the right hand side for A^n in (2.22). By repeating this, we can in fact reduce all terms with powers of A greater than n to $n - 1$. Alternatively, we can achieve the same result by polynomial division. Let $a(s) = \det(sI - A)$ be the characteristic polynomial of A with degree n, and let $p(s)$ be any polynomial of degree $k > n$. We can divide $p(s)$ by $a(s)$ and obtain

$$\frac{p(s)}{a(s)} = q(s) + \frac{r(s)}{a(s)}$$

where the remainder $r(s)$ has degree less than n. Thus, $p(s)$ can be written as

$$p(s) = a(s)q(s) + r(s)$$

Replacing s as variable by the matrix A, we obtain the matrix polynomial equation

$$p(A) = a(A)q(A) + r(A) = r(A)$$

where the second equation follows from $a(A) = 0$. Since the polynomial p was arbitrary, this shows that we can reduce any matrix polynomial in A to a degree less than n. This is also true for the infinite polynomial in (2.22), and we have the following result.

Theorem 2.2

The transition matrix of an nth-order state space model can be written in the form

$$\Phi(t) = \alpha_0(t)I + \alpha_1(t)A + \alpha_2(t)A^2 + \ldots + \alpha_{n-1}(t)A^{n-1} \tag{2.23}$$

Note that the polynomial coefficients are time-varying, because the coefficients of the infinite polynomial (2.22) are time-dependent. One way of computing the functions $\alpha_i(t)$ for a given state space model is suggested in Exercise 2.4.

Exercises

Problem 2.1

Consider the circuit in Figure 2.7.
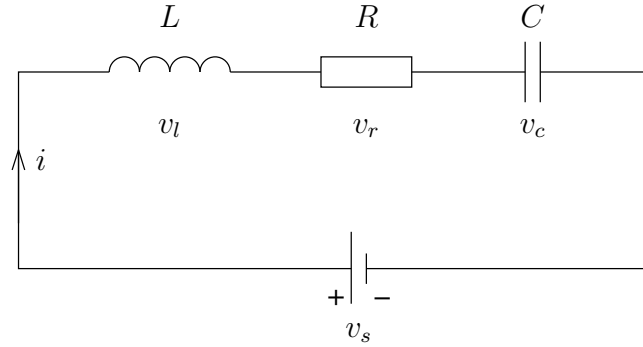
Figure 2.7: State feedback

a) Show that the voltage v_r over the resistance R can be described by the state space representation

$$\dot{x} = Ax + bu, \qquad y = cx$$

where

$$x = \begin{bmatrix} i \\ v_c \end{bmatrix}, \qquad u = v_s, \qquad y = v_r$$

$$A = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 1/L \\ 0 \end{bmatrix}, \qquad c = \begin{bmatrix} R & 0 \end{bmatrix}$$

Use the physical relationships $v_s = v_c + v_l + v_r$, $\qquad v_l = L\frac{di}{dt}$, $\qquad i = C\frac{dv_c}{dt}$.

b) Show that the circuit can also be described by the following differential equation:

$$LC\frac{\mathrm{d}^2 v_c}{\mathrm{d}t^2} + RC\frac{\mathrm{d}v_c}{\mathrm{d}t} + v_c = v_s$$

c) Write this equation in state space form with state variables $x_1 = v_c$ and $x_2 = \dot{v}_c$.

Problem 2.2

Consider a matrix $A \in \mathbb{R}^{n \times n}$ with the characteristic polynomial

$$\det(sI - A) = a(s) = s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0$$

a) Show that if A has distinct eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_n)$, the following relationship holds:

$$\Lambda^n + a_{n-1}\Lambda^{n-1} + \ldots + a_0 I = 0$$

$$\text{with} \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & \lambda_n \end{bmatrix}$$

b) Now show that

$$A^n + a_{n-1}A^{n-1} + \ldots + a_0 I = 0$$

(This proves the Cayley-Hamilton Theorem for distinct eigenvalues.)
Hint: Use the fact that a matrix A with distinct eigenvalues can be written as $A = T\Lambda T^{-1}$; where Λ is diagonal.

Problem 2.3

Consider the homogeneous 2×2 system

$$\dot{x} = Ax$$

where the distinct, real eigenvalues of A are λ_1 and λ_2 with corresponding eigenvectors t_1 and t_2.

a) Using the Laplace transform

$$sX(s) - x(0) = AX(s)$$

show that

$$X(s) = T \begin{bmatrix} \frac{1}{s-\lambda_1} & 0 \\ 0 & \frac{1}{s-\lambda_2} \end{bmatrix} T^{-1}x(0), \qquad T = [t_1 \; t_2]$$

where t_i are the columns of T.

b) Show that with the initial condition

$$x(0) = kt_1$$

we have

$$X(s) = \frac{k}{s - \lambda_1}t_1$$

c) For

$$A = \begin{bmatrix} -1 & 1 \\ -2 & -4 \end{bmatrix}$$

and with the initial condition

$$x(0) = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

use the result of part (b) to derive $x(t)$ analytically. Then, by using MATLAB, plot the behaviour of the system in a phase plane diagram (i.e. sketch $x_2(t)$ over $x_1(t)$ as t goes from zero to infinity).

Problem 2.4

a) Use the fact that the eigenvalues of a square matrix A, are solutions of the characteristic equation to show that $e^{\lambda_i t}$ for all eigenvalues $\lambda = \lambda_1, \lambda_2 \ldots \lambda_n$ can be expressed as

$$e^{\lambda t} = \alpha_0(t) + \alpha_1(t)\lambda + \ldots + \alpha_{n-1}(t)\lambda^{n-1}$$

b) For the case of distinct eigenvalues, show that the functions $\alpha_i(t)$, $i = 1, \ldots, n-1$ in (2.23), are the same as the ones in part a.

   Hint: Use the fact that if $A = T\Lambda T^{-1}$ then $e^{At} = Te^{\Lambda t}T^{-1}$.

c) Find a method to determine the functions $\alpha_i(t)$, $i = 1, \ldots, n-1$ in part b.

   Hint: Use the fact that for distinct eigenvalues the following matrix is invertible:

$$\begin{bmatrix} 1 & \lambda_1 & \ldots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \ldots & \lambda_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_n & \ldots & \lambda_n^{n-1} \end{bmatrix}$$

Problem 2.5

Consider the system $\dot{x} = Ax + bu$ with

$$A = \begin{bmatrix} -6 & 2 \\ -6 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad c = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

a) Use the results from Problem 2.4 to calculate the functions $\alpha_0(t)$, $\alpha_1(t)$ and e^{At}.

b) Calculate the state and output responses, $x(t)$ and $y(t)$, with initial values $x(0) = \begin{bmatrix} 2 & 1 \end{bmatrix}^T$

c) Calculate the output response with initial values from (b) and a step input $u(t) = 2\sigma(t)$

Problem 2.6

a) Convert the 2nd order model of the spring mass system (2.1) into the controller canonical form. Use the values

$$m = 1, \quad k = 1, \quad b = 0.1$$

b) For this model, calculate $(sI - A)^{-1}$ and $c(sI - A)^{-1}b$.

c) Calculate gain matrix $f = [f_1 \ f_2]$ for a state feedback controller $u = fx$, which will bring the system with an initial state of $x(0) = [1 \ 0]^T$ back to a steady state with a settling time (1%) $t_s \leq 5$ and with damping ratio $\zeta \geq 0.7$.

d) Calculate f, using MATLAB, for the settling time $t_s \leq 5$ and damping ratio $\zeta \geq 0.7$.

   Hint: Consider the characteristic polynomial of the closed loop system.

Problem 2.7

Consider the system in Figure 2.8.

a) Determine a state space model with states x_1, x_2, x_3 for this system.
   (This particular state space representation is called the *observer canonical form.*)

b) Show that the transfer function of this system is

$$Y(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s^1 + a_0} U(s)$$
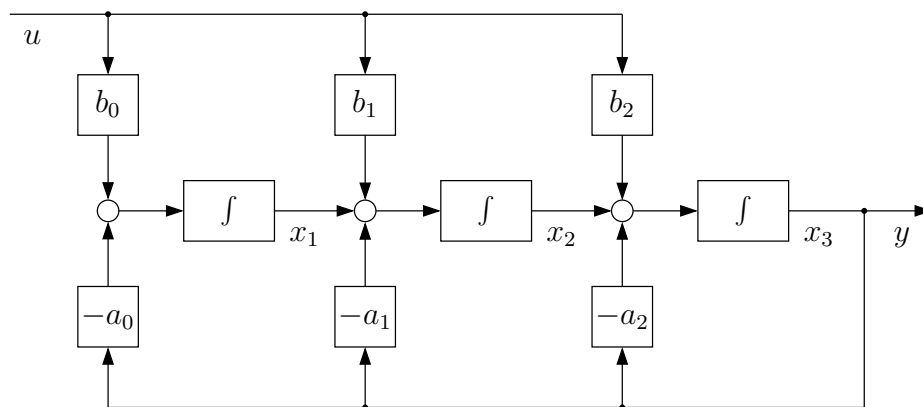
Figure 2.8: Block diagram

Problem 2.8

Determine the controller and observer canonical forms for the system with transfer function

$$H(s) = \frac{4s^3 + 25s^2 + 45s + 34}{s^3 + 6s^2 + 10s + 8}$$

Problem 2.9

Show that two state space models (A_1, b_1, c_1) and (A_2, b_2, c_2) represent the same transfer function *if* a matrix T exists (det $T \neq 0$) such that

$$T^{-1}A_1 T = A_2, \qquad T^{-1}b_1 = b_2, \qquad c_1 T = c_2$$

Problem 2.10

This exercise is a short revision of the concept of linearisation of physical model equations. Consider the water tank in Figure 2.9.
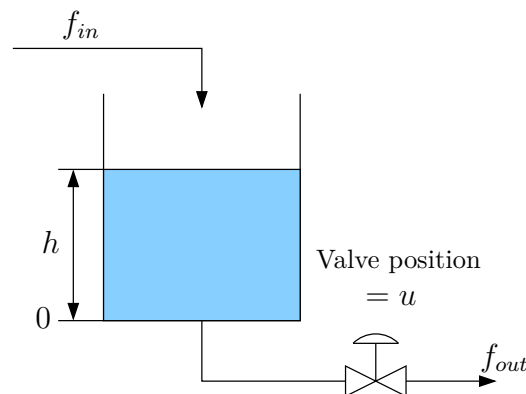
Figure 2.9: State feedback

With the notation

$$
\begin{array}{ll}
A_t & \text{tank cross sectional area} \\
h & \text{water height} \\
f_{in} & \text{inlet flow rate (vol/sec)} \\
f_{out} & \text{outlet flow rate (vol/sec)} \\
P & \text{hydrostatic pressure at the tank's bottom} \\
u & \text{valve position}
\end{array}
$$

the relationships between pressure and water depth, and pressure and water flow out of

the tank are given by

$$P = h\rho g$$
$$f_{out} = k_v \sqrt{P} u$$

a) Describe the relationship between water height h, inflow f_{in} and valve position u by a differential equation. Why is this not linear?

b) Determine the valve position u_0, that for a constant water height h_0 and inflow f_{in0} keeps the level in the tank constant.

c) Show that for small deviations δh, δu of h and u, respectively, from a steady state (h_0, u_0) the following linear approximation can be used to describe the water level dynamics, where $k_t = \sqrt{\rho g} k_v$:

$$\delta \dot{h} = -\frac{f_{in0}}{2 A_t h_0} \delta h - \frac{k_t}{A_t} \sqrt{h_0} \delta u$$

Hint: Use the Taylor-series expansion of $f(h + \delta)$ for small δ

d) Write down the transfer function of the system at the linearised operating point around the steady state (h_0, u_0). Identify the steady state process gain and the time constant of the linearised system.

e) For the linearised system in (c) with input u and output h, determine a state space model of the form

$$\dot{x} = Ax + bu$$
$$y = Cx + du$$

Problem 2.11   *Mini Segway exercise.*

Consider a self balancing robot in Figure 2.10. The relation between the input voltage $u(t)$ and the degrees of freedoms $s(t)$, $\alpha(t)$ are described by the set of *nonlinear differential equations*

$$(m_p + 2m_w + 2\frac{J_w}{r^2})\ddot{s} + m_p l cos(\alpha)\ddot{\alpha} + \frac{k_t k_b}{Rr^2}\dot{s} - \frac{k_t k_b}{Rr}\dot{\alpha} - m_p l sin(\alpha)\dot{\alpha}^2 = \frac{k_t}{Rr}u$$

$$(J_p + m_p l^2)\ddot{\alpha} + m_p l cos(\alpha)\ddot{s} - \frac{k_t k_b}{Rr}\dot{s} + \frac{k_t k_b}{R}\dot{\alpha} - m_p g l sin(\alpha) = -\frac{k_t}{R}u \qquad (2.24)$$

To derive these equations, the Lagrange Formulation and the DC Motor Equation are used. (2.24) can be rewritten as a *nonlinear state space model*

$$\dot{x} = f(x, u) \qquad (2.25)$$

Figure 2.10: The Mini Segway
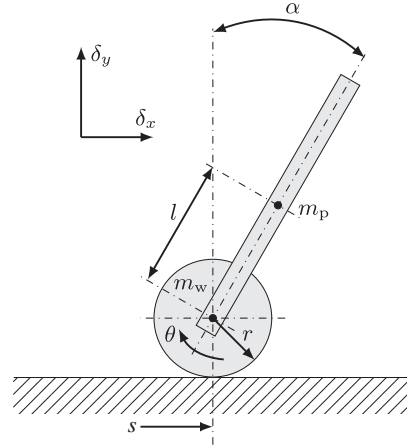
Figure 2.11: Model of the Mini Segway with its degrees of freedoms

where the states are defines as $x = \begin{bmatrix} s & \alpha & \dot{s} & \dot{\alpha} \end{bmatrix}^T$. Equation (2.25) can be linearized around the operating point $(\overline{x}, \overline{u})$, where $x(t) = \overline{x} + \delta x(t)$ and $u(t) = \overline{u} + \delta u(t)$. Finally, this will lead to a *linear state space model*

$$\delta \dot{x} = A\,\delta x + B\,\delta u$$
$$y = Cx + Du$$

(2.26)

Variable	Description	MATLAB	Value
g	gravity constant	g	$9.81\,m/s^2$
R	resistance	R	$4.5\,\Omega$
k_b	EMF constant	kb	$0.495\,Vs/rad$
k_t	torque constant	kt	$0.470\,Nm/A$
m_w	wheel mass	mw	$0.0183\,kg$
J_w	wheel moment of intertia	Jw	$7.462 \cdot 10^{-6}\,kg\,m^2$
r	torque constant	r	$0.0216\,m$
m_p	board mass	mp	$0.3723\,kg$
J_p	board moment of intertia	Jp	$4.67 \cdot 10^{-3}\,kg\,m^2$
l	length to center	l	$0.112\,m$

Table 2.1: List of variables

a) Derive the function $f(x, u)$ for the nonlinear state space model.

b) Find a stable equilibrium state $\overline{x}$ such that $f(\overline{x}, 0) = 0$.

c) Derive the matrices A and B for the linear state space model.

d) Determine the matrices C and D. Assume that all outputs can be measured, therefore $y = x$.

Hint: If you have problems to solve a), go through the following steps

- write the nonlinear differential equations in matrix notation

$$M(s, \alpha) \begin{bmatrix} \ddot{s} \\ \ddot{\alpha} \end{bmatrix} = f_q(x, u)$$

- bring it in the canonical form

$$\dot{x} = \begin{bmatrix} \dot{s} \\ \dot{\alpha} \\ M(s, \alpha)^{-1} f_q(x, u) \end{bmatrix}$$

Problem 2.12 *Mini Segway exercise.*

Test your model in a closed loop configuration with a state feedback controller to check the validity of the linear model. Assume the following initial conditions

$$x(0) = \begin{bmatrix} 0 \\ \alpha_0 \\ 0 \\ 0 \end{bmatrix}$$

initial angle: $\alpha_0 = 5°$

state feedback gain: $K = \begin{bmatrix} 26.4575 & 82.3959 & 56.2528 & 12.0057 \end{bmatrix}$

saturation voltage: $V_s = 7.2\,V$

limited input voltage: $|u(t)| \leq V_s$

a) Build two block diagrams in Simulink, one for the linear model and one for the nonlinear model. Each block diagram should look like Figure 2.12.

- open the Simulink file *Simulation_LQR.slx* and use the blocks on the left
- define all necessary variables in the workspace
- add a saturation block for the nonlinear model
- finally, run *simulation_LQR_script.m*

b) Compare the output from linear and nonlinear model. What do you observe? Is the input voltage inside the limits?

c) Now increase the initial angle first to 10° and then to 15°. Compare the results with the previous task. Is the linear model still a good approximation?
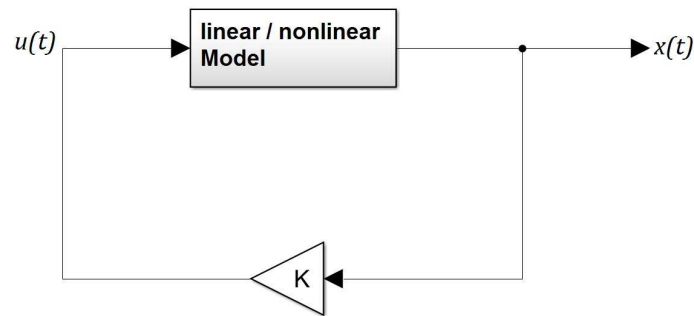
Figure 2.12: Block diagram of the model with state feedback controller

Problem 2.13 *Mini Segway exercise.*

Familiarize yourself with the Mini Segway; for this purpose read and follow the file *Tutorial.pdf.*

a) Build the disturbance signal $d(t)$ in *simulation_LQR_disturbance.slx* as a function of d_0, t_{start} and Δt. Next, run *simulation_LQR_disturbance_script.m*. The signal should look like 2.13. What do you observe?
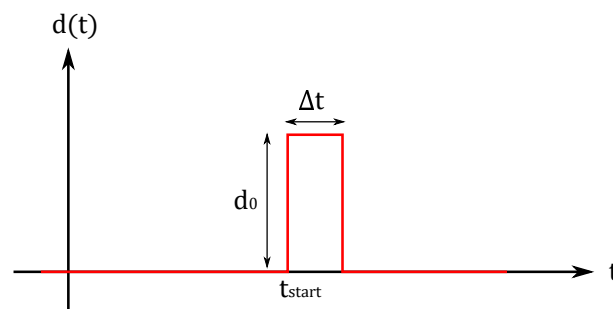
Figure 2.13: input disturbance signal $d(t)$

b) Build the same disturbance signal $d(t)$ again in *experiment_LQR_disturbance.slx*. Compare simulation with experiment by running *comparison_LQR_disturbance_script.m*. Does the experiment match the simulation?

Chapter 3

Controllability and Pole Placement

This chapter will introduce the first of two important properties of linear systems that determine whether or not given control objectives can be achieved. A system is said to be *controllable* if it is possible to find a control input that takes the system from any initial state to any final state in any given time interval. In this chapter, necessary and sufficient conditions for controllability are derived. It is also shown that the closed-loop poles can be placed at arbitrary locations by state feedback if and only if the system is controllable.

We start with a definition of controllability. Consider a system with state space realization

$$\dot{x}(t) = Ax(t) + bu(t) \tag{3.1}$$
$$y(t) = cx(t)$$

Definition 3.1

The system (3.1) is said to be controllable if for any initial state $x(0) = x_0$, time $t_f > 0$ and final state x_f there exists a control input $u(t)$, $0 \leq t \leq t_f$, such that the solution of (3.1) satisfies $x(t_f) = x_f$. Otherwise, the system is said to be uncontrollable.

Since this definition involves only the state equation, controllability is a property of the data pair (A, b).

Example 3.1

As an example of an uncontrollable system, consider

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \quad x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x_f = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

From the solution of the state equation

$$x_1(t) = x_2(t) = \int_0^t e^{\lambda(t-\tau)} u(\tau) d\tau$$

and it is clear that there exists no input $u(t)$ that will bring the system to the final state x_f.

3.1 The Controllability Gramian

Returning to the system (3.1), controllability requires that there exists an input $u(t)$ on the interval $0 \leq t \leq t_f$ such that

$$x_f = e^{At_f}x_0 + \int_0^{t_f} e^{A(t_f-\tau)}bu(\tau)d\tau \tag{3.2}$$

To establish conditions for the existence of such a control input, we will first present a particular choice of input that satisfies (3.2) under the assumption that a certain matrix is invertible. We will then show that if (3.1) is controllable, this assumption is always true.

Thus, consider the input

$$u(t) = -b^T e^{A^T(t_f-t)}W_c^{-1}(t_f)(e^{At_f}x_0 - x_f) \tag{3.3}$$

where the matrix $W_c(t) \in \mathbb{R}^{n \times n}$ is defined as

$$W_c(t) = \int_0^t e^{A\tau}bb^T e^{A^T\tau}d\tau \tag{3.4}$$

That this input takes the system from x_0 to x_f can be easily verified by substituting (3.3) into (3.2), this leads to

$$x_f = e^{At_f}x_0 - W_c(t_f)W_c^{-1}(t_f)(e^{At_f}x_0 - x_f)$$

To see that the left factor in the second term is $W_c(t_f)$, observe that (using a change of variables) $\int_0^t f(t-\tau)d\tau = \int_0^t f(\lambda)d\lambda$.

In the above derivation the assumption that $W_c(t_f)$ is invertible is necessary for the input (3.3) to exist. The matrix function $W_c(t)$ in (3.4) plays an important role in system theory, it is called the *controllability Gramian* of the system (3.1).

The matrix $W_c(t)$ is positive semidefinite for any $t > 0$, because for any column vector $q \in \mathbb{R}^n$ we have

$$q^T \left(\int_0^t e^{A\tau}bb^T e^{A^T\tau}d\tau \right) q = \int_0^t (q^T e^{A\tau}b)^2 d\tau \geq 0$$

This also shows that if $W_c(t)$ is positive definite *for some* $t > 0$, it is positive definite *for all* $t > 0$. We will use the notation $M > 0$ and $M \geq 0$ to indicate that a matrix M is positive definite or positive semi-definite, respectively. Since $W_c(t)$ has full rank and is invertible only if $W_c(t) > 0$, a necessary condition for $u(t)$ in (3.3) to exist is that the controllability Gramian is positive definite.

The following Theorem gives a necessary and sufficient condition for controllability.

Theorem 3.1

The system (3.1) is controllable if and only if the controllability Gramian $W_c(t)$ in (3.4) is positive definite for any $t > 0$.

Proof

That $W_c(t) > 0$ for any $t > 0$ implies controllability has already been shown, it follows from the existence of $u(t)$ in (3.3). To prove the Theorem, it remains to show that controllability also implies $W_c(t) > 0$ for any $t > 0$. Thus, assume that (A, b) is controllable but that there exists a time $t_f > 0$ such that the controllability Gramian is not invertible, i.e. rank $W_c(t_f) < n$. Then there exists a column vector $q \neq 0$ such that

$$q^T \left(\int_0^{t_f} e^{A\tau} bb^T e^{A^T \tau} d\tau \right) q = \int_0^{t_f} (q^T e^{A\tau} b)^2 d\tau = 0$$

which implies

$$q^T e^{A\tau} b = 0, \quad 0 \leq \tau \leq t_f$$

Now choose $x_f = 0$, then

$$0 = e^{At_f} x_0 + \int_0^{t_f} e^{A(t_f - \tau)} bu(\tau) d\tau$$

and therefore

$$q^T e^{At_f} x_0 = 0$$

If we choose $x_0 = e^{-At_f} q$ we have $q^T q = 0$, which contradicts the assumption $q \neq 0$. Thus $W_c(t)$ cannot be singular for any t. This completes the proof.

Note that in order to show that (3.1) is controllable, it is sufficient to show that $W_c(t) > 0$ for some $t > 0$.

3.2 The Controllability Matrix

We will now see that we can check whether a given system is controllable without computing the controllability Gramian (3.4). An equivalent condition for controllability is provided by the following theorem.

Theorem 3.2

The controllability Gramian $W_c(t)$ is positive definite for all $t > 0$ if and only if the
controllability matrix

$$\mathcal{C}(A, b) = [b \ \ Ab \ \ A^2 b \ldots A^{n-1} b] \tag{3.5}$$

has full rank.

Proof

First assume that $W_c(t) > 0$ for all $t > 0$ but rank $\mathcal{C}(A, b) < n$. Then there exists a vector
$q \neq 0$ such that

$$q^T A^i b = 0, \quad i = 0, 1, \ldots, n - 1$$

It follows from Theorem 2.2 that in this case

$$q^T e^{At} b = 0$$

for all $t > 0$, or equivalently $q^T W_c(t) = 0$ for all $t > 0$. This contradicts the assumption,
therefore rank $\mathcal{C}(A, b) = n$.

Conversely, assume that rank $\mathcal{C}(A, b) = n$ but $W_c(t_f)$ is singular for some t_f. Then there
exists a vector $q \neq 0$ such that

$$q^T e^{At} b = 0$$

for $0 \leq t \leq t_f$. Thus, we have for $t = 0$

$$q^T b = 0$$

and evaluating the ith derivative of $q^T e^{At} b$ at $t = 0$ yields

$$q^T A^i b = 0, \quad i = 1, 2, \ldots$$

This implies

$$q^T [b \ \ Ab \ \ A^2 b \ldots A^{n-1} b] = 0$$

which contradicts the assumption that $\mathcal{C}(A, b)$ has full rank. Therefore, $W_c(t)$ must be
non-singular for any $t > 0$. This completes the proof.

Combining Theorem 3.1 and 3.2, we have

Corollary 3.1

*The system (3.1) is controllable if and only if the controllability matrix $\mathcal{C}(A, b)$ has full
rank.*

3.3   Pole Placement

It was shown in Chapter 2 that state feedback can be used to change the poles of a system (which turned out to be eigenvalues of the system matrix), and that the location of the closed-loop eigenvalues depends on the choice of the feedback gain vector f. An important question is whether for any given choice of pole locations - constrained of course by the fact that complex eigenvalues of a real matrix are symmetric about the real axis - there exists a feedback gain vector f that achieves the desired closed-loop poles. We will now show that this is indeed the case if the system is controllable.

Consider again the system (3.1) with $x(0) = 0$. Assume the control input is taken to be

$$u(t) = fx(t) + u_v(t)$$

leading to the closed-loop system

$$\dot{x}(t) = (A + bf)x(t) + bu_v(t)$$

Let

$$a(s) = \det(sI - A) = s^n + a_{n-1}s^{n-1} + \ldots + a_0$$

denote the open-loop characteristic polynomial, and

$$\bar{a}(s) = \det(sI - A - bf) = s^n + \bar{a}_{n-1}s^{n-1} + \ldots + \bar{a}_0 \tag{3.6}$$

the closed-loop characteristic polynomial. The question is whether for any choice of closed-loop eigenvalues - equivalently for any polynomial $\bar{a}(s)$ - it is possible to find a gain vector f that satisfies (3.6).

To answer this question, we investigate the closed-loop transfer function from u_v to y. The open-loop transfer function from u to y is

$$G(s) = \frac{b(s)}{a(s)} = c(sI - A)^{-1}b$$

From Fig. 3.1 - where we introduced a new signal $z(t)$ - we find that

$$\frac{Z(s)}{U(s)} = f(sI - A)^{-1}b = \frac{m(s)}{a(s)} \tag{3.7}$$

where $m(s)$ denotes the numerator polynomial of the transfer function from u to z.

Now closing the loop, we have

$$\frac{Z(s)}{U_v(s)} = \frac{m(s)/a(s)}{1 - m(s)/a(s)} = \frac{m(s)}{a(s) - m(s)}$$

and the closed-loop transfer function is

$$\frac{Y(s)}{U_v(s)} = \frac{Y(s)}{U(s)} \frac{U(s)}{Z(s)} \frac{Z(s)}{U_v(s)} = \frac{b(s)}{a(s) - m(s)}$$

An important observation that follows from the last equation is
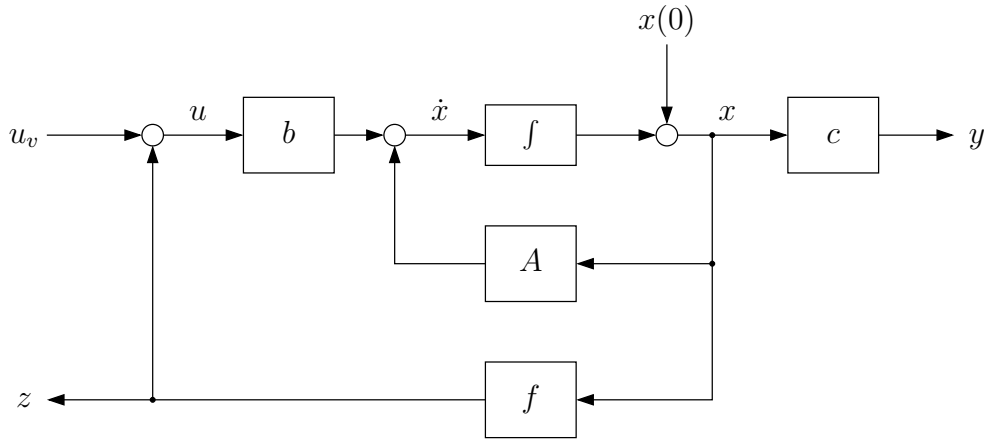
Figure 3.1: State feedback

Theorem 3.3

State feedback does not change the zeros of a system.

The closed-loop characteristic polynomial is related to the open-loop characteristic polynomial by

$$\bar{a}(s) = a(s) - m(s)$$

Note that from (3.7)

$$m(s) = a(s)f(sI - A)^{-1}b$$

thus

$$\bar{a}(s) - a(s) = -a(s)f(sI - A)^{-1}b \tag{3.8}$$

For a given choice of closed-loop eigenvalues, the left hand side is fixed, and the question is whether a vector f exists that satisfies this equation.

We will need the following resolvent identity, which is derived in Exercise 3.3.

$$a(s)(sI - A)^{-1} = s^{n-1}I + s^{n-2}(a_{n-1}I + A) + s^{n-3}(a_{n-2}I + a_{n-1}A + A^2) + \ldots$$

Substituting this in (3.8) yields

$$\bar{a}(s) - a(s) = -f(s^{n-1}I + s^{n-2}(a_{n-1}I + A) + s^{n-3}(a_{n-2}I + a_{n-1}A + A^2) + \ldots)b$$
$$= -fbs^{n-1} - f(a_{n-1}I + A)bs^{n-2} - f(a_{n-2}I + a_{n-1}A + A^2)bs^{n-3} - \ldots$$

Equating coefficients, we have

$$\bar{a}_{n-1} - a_{n-1} = -fb$$
$$\bar{a}_{n-2} - a_{n-2} = -f(a_{n-1}b + Ab)$$
$$\bar{a}_{n-3} - a_{n-3} = -f(a_{n-2}b + a_{n-1}Ab + A^2b)$$
$$\vdots$$

If we introduce the vector

$$p = [\bar{a}_{n-1} - a_{n-1} \quad \bar{a}_{n-2} - a_{n-2} \quad \ldots \quad \bar{a}_1 - a_1 \quad \bar{a}_0 - a_0]$$

then we can write the above in matrix form as

$$p = -f[b \ Ab \ A^2b \ \ldots \ A^{n-1}b] \begin{bmatrix} 1 & a_{n-1} & a_{n-2} & \ldots & a_1 \\ 0 & 1 & a_{n-1} & \ldots & a_2 \\ 0 & 0 & 1 & & a_3 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

Let T_a denote the Toeplitz matrix on the right hand side, and note that the second factor on the right hand side is the controllability matrix $\mathcal{C}(A, b)$. Since T_a is invertible, we can solve for the desired gain vector f if and only if $\mathcal{C}(A, b)$ has full rank. In this case

$$f = -p \, T_a^{-1} \, \mathcal{C}^{-1}(A, b) \tag{3.9}$$

This proves the following.

Theorem 3.4

The eigenvalues of the system (3.1) can be placed at arbitrary locations by state feedback if and only if (A, b) is controllable.

If the system is controllable, equation (3.9) - which is known as *Bass-Gura formula* - can be used to compute the state feedback gain required to assign the desired eigenvalues.

3.4 Uncontrollable Systems

We derived two equivalent tests for the controllability of a system - checking the rank either of the controllability Gramian or of the controllability matrix. Now we will address the question of what can be said about a system if it fails these rank tests. It is often helpful for gaining insight if a state space model is in diagonal form. Consider a model with state equation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix} u$$

In this diagonal form the state variables are decoupled, and because the last element in b is zero, there is no way to influence the solution $x_3(t)$ via the control input $u(t)$. The system is clearly uncontrollable in the sense of Definition 3.1, because we cannot take x_3 to any desired value at a given time. On the other hand, it may well be possible to take

x_1 and x_2 to any desired value - this is indeed the case if $\lambda_1 \neq \lambda_2$ and b_1, b_2 are non-zero. This example illustrates that when a system is uncontrollable, it is often of interest to identify a controllable subsystem. The following Theorem suggests a way of doing this.

Theorem 3.5

Consider the state space model (3.1), and assume that rank $\mathcal{C}(A, b) = r < n$. Then there exists a similarity transformation

$$x = T_c \bar{x}, \qquad \bar{x} = \begin{bmatrix} \bar{x}_c \\ \bar{x}_{\bar{c}} \end{bmatrix}$$

such that

$$\begin{bmatrix} \dot{\bar{x}}_c \\ \dot{\bar{x}}_{\bar{c}} \end{bmatrix} = \begin{bmatrix} \bar{A}_c & \bar{A}_{12} \\ 0 & \bar{A}_{\bar{c}} \end{bmatrix} \begin{bmatrix} \bar{x}_c \\ \bar{x}_{\bar{c}} \end{bmatrix} + \begin{bmatrix} \bar{b}_c \\ 0 \end{bmatrix} u, \qquad y = \begin{bmatrix} \bar{c}_c & \bar{c}_{\bar{c}} \end{bmatrix} \begin{bmatrix} \bar{x}_c \\ \bar{x}_{\bar{c}} \end{bmatrix} \qquad (3.10)$$

with $\bar{A}_c \in \mathbb{R}^{r \times r}$ and $(\bar{A}_c, \bar{b}_c)$ controllable. Moreover, the transfer function of the system is

$$G(s) = c(sI - A)^{-1}b = \bar{c}_c(sI - \bar{A}_c)^{-1}\bar{b}_c$$

Proof

Let $(\bar{A}, \bar{b}, \bar{c})$ denote the transformed model (3.10), then

$$\mathcal{C}(\bar{A}, \bar{b}) = \mathcal{C}(T_c^{-1}AT_c, T_c^{-1}b) = [T_c^{-1}b \ \ T_c^{-1}Ab \ \ \dots \ \ T_c^{-1}A^{n-1}b]$$

Thus $\mathcal{C}(\bar{A}, \bar{b}) = T_c^{-1}\mathcal{C}(A, b)$. This shows that rank $\mathcal{C}(\bar{A}, \bar{b}) = $ rank $\mathcal{C}(A, b) = r$. The controllability matrix $\mathcal{C}(\bar{A}, \bar{b})$ has the form

$$\mathcal{C}(\bar{A}, \bar{b}) = \begin{bmatrix} \bar{b}_c & \bar{A}_c\bar{b}_c & \dots & \bar{A}_c^{n-1}\bar{b}_c \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

The first r columns are linearly independent; to see this, note that for each $k \geq r$, $\bar{A}_c^k$ is a linear combination of $\bar{A}_c^i$, $0 \leq i < r$ by the Cayley-Hamilton Theorem. Therefore

$$\text{rank } [\bar{b}_c \ \ \bar{A}_c\bar{b}_c \ \ \dots \ \ \bar{A}_c^{r-1}\bar{b}_c] = r$$

i.e. $(\bar{A}_c, \bar{b}_c)$ is controllable.

Next we show that a matrix T_c that transforms an uncontrollable state space realization into the form of (3.10) always exists. In fact, such a transformation matrix is

$$T_c = [b \ \ Ab \ \ \dots \ \ A^{r-1}b \ \ q_{r+1} \ \ \dots q_n]$$

where the first r columns are the linearly independent columns of $\mathcal{C}(A, b)$, and $q_{r+1} \dots q_n$ are any $n - r$ linearly independent vectors such that T_c is nonsingular. To verify that this

choice of T_c indeed results in (3.10), substitute in $AT_c = T_c\bar{A}$ to get

$$[Ab \; A^2b \; \dots \; A^rb \; Aq_{r+1} \; \dots Aq_n] = [b \; Ab \; \dots \; A^{r-1}b \; q_{r+1} \; \dots q_n]\begin{bmatrix} * & * \\ 0 & * \end{bmatrix}$$

$$= T_c\begin{bmatrix} \bar{A}_c & \bar{A}_{12} \\ 0 & \bar{A}_{\bar{c}} \end{bmatrix}$$

Here $*$ denotes matrix blocks with possibly non-zero elements. That the lower left block in the matrix on the right is zero follows from the fact that the first r columns of the matrix on the left hand side of the equation are linear combinations of the first r columns of $\mathcal{C}(A, b)$. Actually this is the result of the fact that when the rank of controllability matrix is r, the first r columns are linearly independent. We can prove this in the following: Consider that the rank of $\mathcal{C}(A, b)$ is r and the first r columns are not linearly independent. Then there exist an s, which is less that r and we have

$$A^sb + a_{s-1}A^{s-1}b + a_{s-2}A^{s-2}b + \dots + a_0b = 0$$

This means that A^sb can be expressed as linear combination of components with less powers of A. Now if we multiply the above equation by A, we can conclude that $A^{s+1}b$ can be expressed as linear combination of components with powers of A less than and/or equal to s. However it was seen that A^sb can be expressed as linear combination of components with the powers of A less than s, thus, $A^{s+1}b$ also can be expressed as linear combination of components with the powers of A less than s. In the same way we can continue to show that columns from $A^{s+2}b$ to $A^{n-1}b$ can be expressed as linear combination of columns with the powers of A less than s as well. This means that the rank of $\mathcal{C}(A, b)$ is $s < r$ which contradicts our assumption. Similarly, we have

$$b = T_c\,\bar{b} = [b \; Ab \; \dots \;]\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The last statement of the Theorem is easily verified by computing the transfer function of the model $(\bar{A}, \bar{b}, \bar{c})$ in (3.10). This completes the proof.

The transformation matrix T_c used in this proof is not the best choice from a numerical point of view. A numerically reliable way of constructing a transformation matrix T_c is to use QR factorization: if $\mathcal{C} = QR$ is a QR factorization of $\mathcal{C}$, then $T_c = Q$.

The fact that $\mathcal{C}(\bar{A}, \bar{b}) = T_c^{-1}\mathcal{C}(A, b)$ was used above to show that the subsystem with r state variables is controllable. An important observation is that this equation holds for any transformation T. In particular, if the state space model (3.1) is controllable and T is a transformation to $(\tilde{A}, \tilde{b}, \tilde{c})$, then we have $\mathcal{C}(\tilde{A}, \tilde{b}) = T^{-1}\mathcal{C}(A, b)$ and therefore rank $\mathcal{C}(\tilde{A}, \tilde{b}) = \text{rank}\,\mathcal{C}(A, b) = n$. This proves the following:

Theorem 3.6

Controllability is invariant under similarity transformations.

This result shows that controllability is not a property of a particular state space realization, but a property of a system which is independent of the coordinate basis.

Example 3.2

Consider a system with state space realization

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

The system is not controllable, we have

$$\mathcal{C}(A, b) = \begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix}$$

and rank $\mathcal{C} = 1$. To bring this system in the form of (3.10), we construct the transformation matrix $T_c = [t_1 \ t_2]$ by taking $t_1 = b$ and choosing t_2 orthogonal to t_1. Thus

$$T_c = \begin{bmatrix} 1 & 1 \\ 2 & -0.5 \end{bmatrix} \quad \text{and} \quad T_c^{-1} = \begin{bmatrix} 0.2 & 0.4 \\ 0.8 & -0.4 \end{bmatrix}$$

Applying this transformation yields

$$\bar{A} = T_c^{-1} A T_c = -T_c^{-1} T_c = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \bar{b} = T_c^{-1} b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \bar{c} = c T_c = \begin{bmatrix} 3 & 0.5 \end{bmatrix}$$

This realization has the required zeros in $\bar{A}$ and $\bar{b}$, and

$$\mathcal{C}(\bar{A}, \bar{b}) = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$$

Controllable Subspace

If a system is not controllable, one might ask which parts of the state space can be reached by the state vector, starting from the origin. Assuming $x(0) = 0$, we have

$$x(t) = \int_0^t e^{A(t-\tau)} b u(\tau) d\tau$$

$$= \int_0^t \left(\alpha_0(t-\tau) I + \alpha_1(t-\tau) A + \ldots + \alpha_{n-1}(t-\tau) A^{n-1} \right) b u(\tau) d\tau$$

where Theorem 2.2 has been used in the last equation. The expression inside the integral can be rearranged as

$$x(t) = \int_0^t [b \ \ Ab \ \ \dots \ \ A^{n-1}b] \begin{bmatrix} \alpha_0(t - \tau) \\ \vdots \\ \alpha_{n-1}(t - \tau) \end{bmatrix} u(\tau) d\tau$$

which can also be written as

$$x(t) = [b \ \ Ab \ \ \dots \ \ A^{n-1}b] \begin{bmatrix} \beta_0(t) \\ \vdots \\ \beta_{n-1}(t) \end{bmatrix}$$

where

$$\beta_i(t) = \int_0^t \alpha_i(t - \tau) u(\tau) d\tau$$

Observing that the matrix on the right hand side is the controllability matrix $\mathcal{C}$, we conclude that the state vector $x(t)$ can only take values that are linear combinations of the columns of $\mathcal{C}$, i.e.

$$x(t) \in \mathcal{R}(\mathcal{C})$$

Here $\mathcal{R}()$ denotes the *column space* of a matrix. Thus, the part of the state space that is reachable from the origin is precisely the column space of the controllability matrix. This space is called the *controllable subspace.* For a controllable system, the controllable subspace is the entire state space.

Returning to Example 3.2, the controllable subspace is spanned by the vector $[1 \ \ 2]^T$, it is a line through the origin with slope 2.

Stabilizability

The state variables of the uncontrollable subsystem $(\bar{A}_{\bar{c}}, 0, \bar{c}_{\bar{c}})$ cannot be influenced through the system input, but they can have an effect on the system output through $\bar{c}_{\bar{c}}$. It is clear that if $\bar{A}_{\bar{c}}$ has eigenvalues in the right half plane, then there is no way to stabilize the system by state feedback. This motivates the following definition.

Definition 3.2

The system with state space realization (3.1) is said to be stabilizable *if there exists a state feedback law $u(t) = fx(t)$ such that the resulting system is stable.*

If (3.1) is uncontrollable and (3.10) is another realization of the same system, then the system is stabilizable if and only if $\bar{A}_{\bar{c}}$ has no eigenvalues in the right half plane.

The decomposition of a state space model into a controllable and an uncontrollable subsystem is shown in the form of a block diagram in Fig. 3.2.
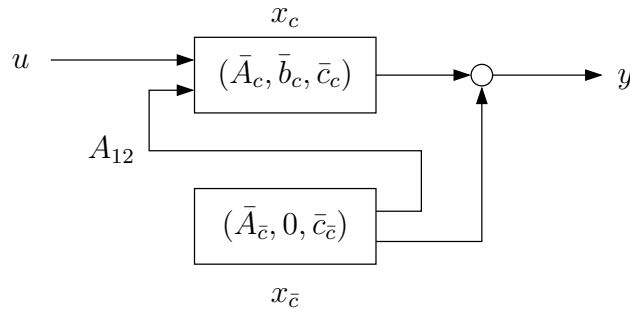
Figure 3.2: Decomposition of uncontrollable realization

The Popov-Belevitch-Hautus Test

We have already seen two necessary and sufficient conditions for a system to be controllable, in terms of the rank of the controllability Gramian and the controllability matrix, respectively. We finally mention an alternative condition for controllability that is sometimes useful, the Popov-Belevitch-Hautus (PBH) test.

Theorem 3.7   *The system (3.1) is controllable if and only if the matrix*

$$[sI - A \ \ b] \tag{3.11}$$

has full row rank for all $s \in \mathbb{C}$.

A proof is presented in Exercise 3.5.

Note that as a consequence of this result, a system is stabilizable if and only if the matrix (3.11) does not loose rank for all s in the right half plane.

Exercises

Problem 3.1

Consider the system

$$\dot{x} = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix} x + \begin{bmatrix} -2 \\ 1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 0 & 1 \end{bmatrix} x$$

with initial values

$$\begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$$

a) Identify whether the system is stable or unstable.

b) Determine whether the system is controllable or uncontrollable, and stabilizable or unstabilizible.

c) Show the controllable and uncontrollable subspaces in a phase plane diagram.

d) Calculate the Laplace transform of the response $x(t)$ to initial values $[x_{10}\ x_{20}]^T$ with the input $u(t) = 0$. Describe the relationship between initial values required to ensure that the system eventually reaches equilibrium at $[0\ 0]^T$. Compare it to the answer of part (c) and explain whether there can be any relationship between them or not.

e) Calculate the transfer function of the system.

f) Does the transfer function fully describe the dynamic behaviour of the system? If not, why not?

Problem 3.2

This problem shows how the limit of the controllability Gramian as $t \to \infty$ can be represented as the solution of an algebraic matrix equation.

For the *stable* system

$$\dot{x}(t) = Ax(t) + bu(t)$$

The limiting value of the controllability Gramian, W_c is

$$W_c = \lim_{t \to \infty} W_c(t)$$

a) Calculate the derivative

$$\frac{d}{dt}(e^{At}bb^T e^{A^T t})$$

b) Show that for all $t > 0$ the following is true:

$$AW_c(t) + W_c(t)A^T = \int_0^t \frac{d}{d\tau} e^{A\tau} bb^T e^{A^T\tau} d\tau$$

c) Show that as $t \to \infty$ the controllability Gramian $W_c(t)$ satisfies the algebraic equation

$$AW_c + W_c A^T + bb^T = 0$$

Problem 3.3

Derive the resolvent identity of Theorem 3.3:

$$a(s)(sI - A)^{-1} = s^{n-1}I + s^{n-2}(a_{n-1}I + A)$$
$$+ s^{n-3}(a_{n-2}I + a_{n-1}A + A^2) + \ldots + (a_1 I + \ldots + a_{n-1}A^{n-2} + A^{n-1})$$

Hint: Use the Cayley-Hamilton Theorem.

Problem 3.4

For the system

$$\dot{x} = Ax + bu$$

where

$$A = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix}, \qquad b = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

use the Bass-Gura formula (3.9) to calculate a state feedback gain vector f that achieves a damping ratio $\zeta = 0.7$ and a natural frequency $\omega_n = 2.0$.

Use Matlab to simulate the responses of the states to a unit step change in the closed loop system input u_v.

Problem 3.5

For the system of order n

$$\dot{x} = Ax + bu$$

consider a vector $q \neq 0$ such that

$$q^T A = \lambda q^T \qquad \text{and} \qquad q^T b = 0$$

a) Show that if such a q exists then
$$q^T \mathcal{C} = 0$$
where $\mathcal{C}$ is the controllability matrix of A, b.

b) Explain why the system is uncontrollable if such a q exists. Then show that the converse is also true.

c) Show that

$$\text{rank} \begin{bmatrix} sI - A & b \end{bmatrix} < n$$

for some $s \in \mathbb{C}$ if and only if such a q exists.

d) Use the above results to show that the system is controllable iff $\begin{bmatrix} sI - A & b \end{bmatrix}$ has full row rank for all $s \in \mathbb{C}$ (PBH test).

Problem 3.6

Consider the system in Figure 3.3. The equations of motion of this system are

$$M\dot{v} = -mg\theta_1 - mg\theta_2 + u$$
$$m(\dot{v} + l_i\ddot{\theta}_i) = mg\theta_i, \qquad i = 1, 2$$

where $v(t)$ is the speed of the cart and $u(t)$ is a force applied to the cart. Note that these equations represent the system dynamics only in a small neighborhood of the equilibrium.
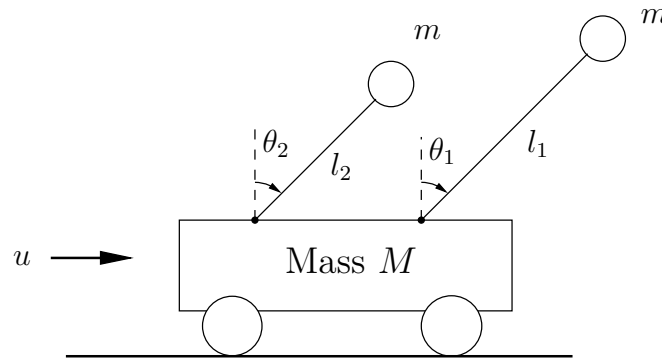
Figure 3.3: Pendulum on cart

a) Determine a state space model of this system with state variables θ_1, θ_2, $\dot{\theta}_1$ and $\dot{\theta}_2$.

b) What happens to the last 2 rows of A and b if the pendulums have the same length?

c) Assume now that the pendulums have the same length. Show that the controllability matrix of the system can be written in the form:

$$\mathcal{C} = \begin{bmatrix} 0 & b_1 & 0 & \tilde{b}_1 \\ 0 & b_1 & 0 & \tilde{b}_1 \\ b_1 & 0 & \tilde{b}_1 & 0 \\ b_1 & 0 & \tilde{b}_1 & 0 \end{bmatrix}$$

Show then that the controllable subspace is defined by $\theta_1 = \theta_2$ and $\dot{\theta}_1 = \dot{\theta}_2$.

Problem 3.7

a) For the system in Problem 3.6, generate a state space model in Matlab with the following parameter values: $g = 10$, $M = 10$, $m = 1.0$, $l_1 = 1.0$, $l_2 = 1.5$.

b) Use Matlab to design a state feedback controller with the following properties

   – A damping ratio of $\zeta > 0.9$ for each pair of poles

   – A natural frequency of $\omega_n > 5$ for each pair of poles

   – In responding to the initial conditions $\theta_1 = 0.5$, $\theta_2 = 1.0$, $\dot{\theta}_1 = \dot{\theta}_2 = 0$, the magnitude of the input signal u should be less than 2000 and the maximum angle error should be less than 1.5

Problem 3.8

In this problem it is demonstrated for a 3rd order system, that a similarity transformation matrix T exists that takes a state space realization into controller canonical form if and only if (A, b) is controllable.

a) First assume that a transformation matrix

$$T = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}$$

exists, where t_i are the columns of T, and that the resulting controller form is (A_c, b_c, c_c). Use the structure of the controller form to show that $t_3 = b$.

b) Use the structure of the controller form to show that

$$t_1 = A^2 b + A b a_2 + b a_1$$

$$t_2 = A b + a_2 b$$

where a_0, a_1 and a_2 are the coefficients of the characteristic polynomial of A.

c) Show that
$$T = \mathcal{C} \begin{bmatrix} a_1 & a_2 & 1 \\ a_2 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$
where $\mathcal{C}$ is the controllability matrix of (A, b).

d)   Why is T a transformation matrix only if (A, b) is controllable?

Problem 3.9

Find a state space model for the system in Figure 3.4. Calculate its controllability matrix and its transfer function.
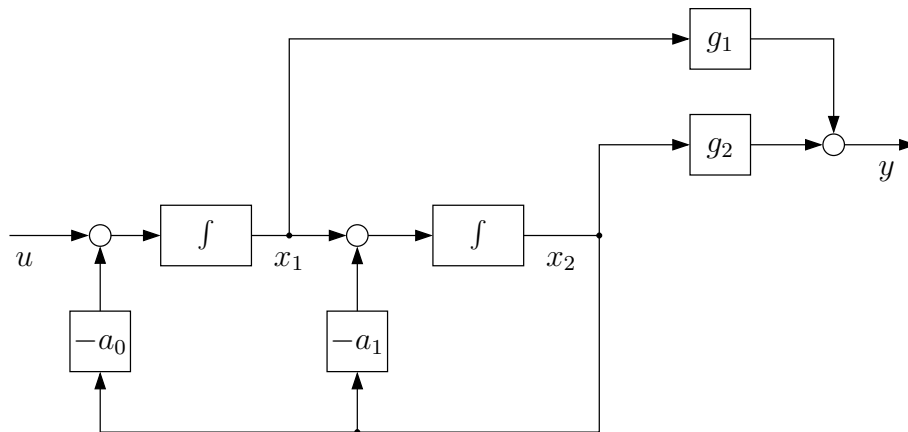
Figure 3.4: Controllability form

Problem 3.10

Consider a controllable system

$$\dot{x} = Ax + bu$$

Show that with the state feedback $u = fx + u_v$ the closed loop system

$$\dot{x} = (A + bf)x + bu_v$$

is still controllable.

Hint: Use the fact that only controllable systems can have state space models in controller canonical form.

Chapter 4

Observability and State Estimation

The discussion of state feedback in the previous chapter assumed that all state variables are available for feedback. This is an unrealistic assumption and in practice rarely the case. A large number of sensors would be required, while it is known from classical control theory that efficient control loops can be designed that make use of a much smaller number of measured feedback signals (often just one). However, the idea of state feedback can still be used even if not all state variables are measured: the measurements of state variables can be replaced by estimates of their values. In this chapter we discuss the concept of state estimation via observers. Moreover, after discussing controllability we introduce the second of two important properties of a linear system: a system is called *observable* if the values of its state variables can be uniquely determined from its input and output signals. It turns out that the problem of designing a state estimator has the same mathematical structure as the problem of designing a state feedback controller; for this reason they are called *dual problems*.

Since the plant models we encounter are usually strictly proper, we will limit the discussion to strictly proper systems.

4.1 State Estimation

Consider the system represented by the state space model

$$\dot{x}(t) = Ax(t) + bu(t), \qquad x(0) = x_0$$
$$y(t) = cx(t) \tag{4.1}$$

Assume it is desired to change the eigenvalues of the system to improve its dynamic properties, but only the input signal $u(t)$ and the output signal $y(t)$ are available for feedback. The idea of state feedback discussed in the previous chapter can still be used if we can obtain an estimate of the state vector $x(t)$. One possible approach is indicated

in Fig. 4.1. As part of the controller, we could simulate the system using the same state space model as in (4.1), and apply the same input $u(t)$ to the simulation model that is applied to the actual plant. Provided that the initial values of the state variables (the integrator outputs) are known, and that the simulation model is an exact replication of the actual system, the estimated state vector $\hat{x}(t)$ will track the true state vector exactly. This estimated state vector could then be used to implement a state feedback controller.
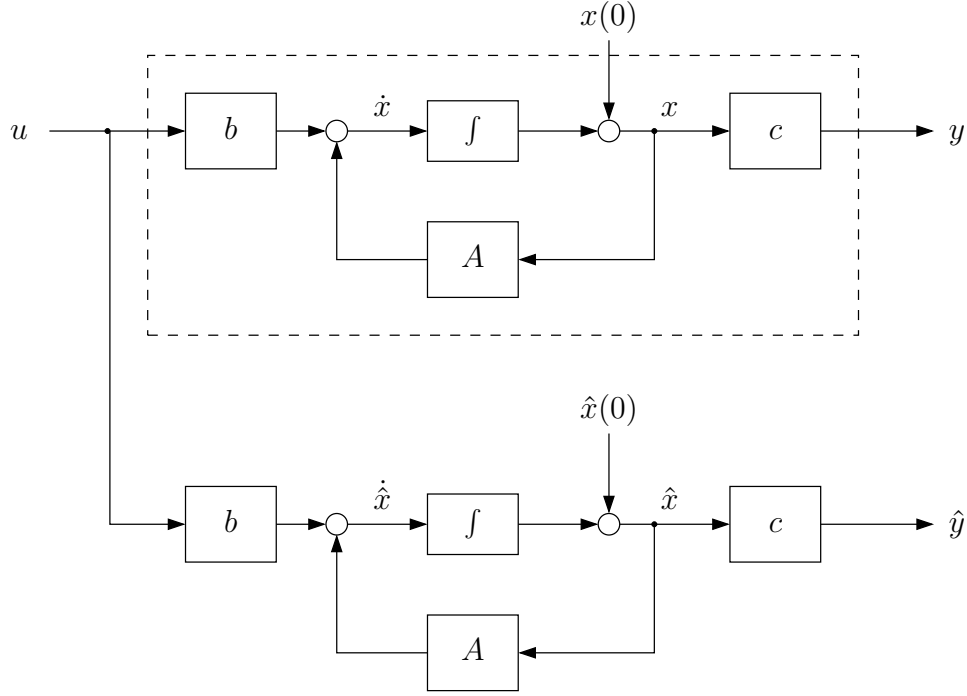
Figure 4.1: Open-loop state estimator

Unfortunately, the initial state values are in general not known, and for this reason the scheme in Fig. 4.1 is impractical. The estimated state vector $\hat{x}(t)$ is governed by

$$\dot{\hat{x}}(t) = A\hat{x}(t) + bu(t), \qquad \hat{x}(0) = \hat{x}_0$$

and subtracting this from (4.1) shows that the dynamics of the estimation error $\tilde{x}(t) = x(t) - \hat{x}(t)$ are determined by

$$\dot{\tilde{x}}(t) = A\tilde{x}(t), \qquad \tilde{x}(0) = x(0) - \hat{x}(0)$$

In general $\tilde{x}(0) \neq 0$, and it will depend on the eigenvalues $(\lambda_1, \ldots, \lambda_n)$ of A how fast (or if at all) the error will go to zero: a partial fraction expansion of $\tilde{X}(s) = (sI - A)^{-1}\tilde{x}(0)$ shows that

$$\tilde{x}(t) = \phi_1 e^{\lambda_1 t} + \ldots + \phi_n e^{\lambda_n t}$$

where ϕ_i is a column vector that depends on the residual at λ_i. If A has eigenvalues close to the imaginary axis, error decay will be slow, and if the plant is unstable, the error will become infinite.
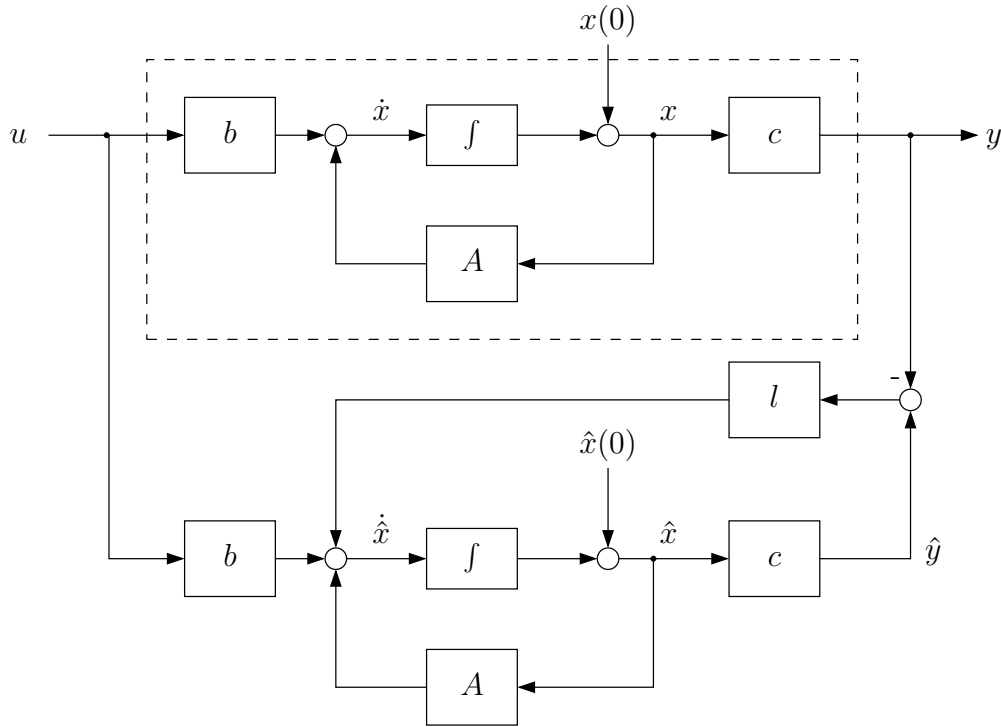
Figure 4.2: Closed-loop state estimator

Estimation Error Feedback

The problem of dealing with unsatisfactory error dynamics can be addressed in the same way as the problem of modifying unsatisfactory plant dynamics - by using feedback. To improve the error dynamics of the estimator in Fig. 4.1, we introduce error feedback as shown in Fig. 4.2. The measured output signal $y(t)$ is used to compute the output estimation error $\hat{y}(t) - y(t)$, which is multiplied by a gain (column) vector l and added to the integrator input of the estimator. This configuration is known as a *state observer*. The estimator dynamics are now

$$\dot{\hat{x}}(t) = A\hat{x}(t) + lc(\hat{x}(t) - x(t)) + bu(t) \tag{4.2}$$

and subtracting this from (4.1) leads to

$$\dot{x}(t) - \dot{\hat{x}}(t) = A(x(t) - \hat{x}(t)) + lc(x(t) - \hat{x}(t))$$

or

$$\dot{\tilde{x}}(t) = (A + lc)\,\tilde{x}(t), \qquad \tilde{x}(0) = x(0) - \hat{x}(0) \tag{4.3}$$

The error dynamics are now determined by the eigenvalues of $A + lc$. A and c are given plant data, but the gain vector l can be chosen freely to obtain desired eigenvalues. The situation closely resembles pole placement via state feedback, where a gain (row) vector f is chosen to place the eigenvalues of $A + bf$ in desired locations of the complex plane. The only difference is that f is a right factor in the product bf, whereas l is a left factor

in the product lc. However, observing that the eigenvalues do not change if a matrix is transposed, we can equivalently consider the problem of choosing a row vector l^T to obtain desired eigenvalues of $(A + lc)^T = A^T + c^T l^T$. This problem has exactly the same form as the problem of pole placement by state feedback.

Duality of State Feedback and State Estimation

The last observation suggests that we can use the formula (3.9) for pole placement to compute the estimator gain vector l. All we need to do is to make the replacements

$$A \to A^T, \quad b \to c^T, \quad f \to l^T$$

and then use (3.9) to get

$$l^T = -p\, T_a^{-1} \mathcal{C}^{-1}(A^T, c^T)$$

In the coefficient vector p the desired characteristic polynomial of the observer $\bar{a}_o(s) = \det(sI - A - lc)$ is then used instead of the desired characteristic plant polynomial.

However, the question remains whether it is always possible to find a gain vector l that achieves any set of desired observer eigenvalues. For the pole placement problem a necessary and sufficient condition was found to be that the controllability matrix $\mathcal{C}(A, b)$ has full rank. Using again the above replacements, it follows that a necessary and sufficient condition for arbitrary assignability of observer eigenvalues is that the matrix

$$\mathcal{C}(A^T, c^T) = [c^T \quad A^T c^T \quad \dots \quad (A^T)^{n-1} c^T]$$

has full rank. We will call the transpose of this matrix the *observability matrix* of the model (4.1)

$$\mathcal{O}(c, A) = \mathcal{C}^T(A^T, c^T) = \begin{bmatrix} c \\ cA \\ \vdots \\ cA^{n-1} \end{bmatrix} \tag{4.4}$$

The Theorem below follows now simply from the fact that the pole placement problem and the state estimation problem have the same mathematical form.

Theorem 4.1

The eigenvalues of the state estimator (4.2) for the system (4.1) can be placed at arbitrary locations by a suitable choice of the estimator gain vector l if and only if the observability matrix $\mathcal{O}(c, A)$ has full rank.

Pole placement and state estimation are said to be *dual problems*. For each result about state feedback in the last chapter, we will obtain an equivalent result about state estimation by invoking the duality between both problems, i.e. by making suitable replacements.

Comparing both problems, we recall that rank $\mathcal{C}(A, b) = n$ is a necessary and sufficient condition for a system to be controllable, and that controllability is an important property of a system that determines whether the poles can be shifted via state feedback to arbitrary locations. This raises the question of what the property rank $\mathcal{O}(c, A) = n$ tells us about a system. The following definition will help to clarify this.

Definition 4.1

The system with state space model (4.1) is said to be observable if for any $t_f > 0$ the initial state $x(0)$ can be uniquely determined from the time history of the input $u(t)$ and the output $y(t)$ in the time interval $0 \leq t \leq t_f$. Otherwise, the system is said to be unobservable.

Note that - given the time history of $u(t)$ - knowledge of $x(0)$ is all that is needed to uniquely determine the state vector $x(t)$ by solving the state equation.

Theorem 4.2

The system (4.1) is observable if and only if the observability matrix $\mathcal{O}(c, A)$ has full rank.

Proof

First we show that rank $\mathcal{O}(c, A) = n$ implies observability. For $0 \leq t \leq t_f$ we have

$$y(t) = ce^{At}x(0) + \int_0^t ce^{A(t-\tau)}bu(\tau)d\tau$$

Since $u(t)$ is known, the second term on the right hand side can be computed and subtracted on both sides of the equation. We can therefore replace the above by an equivalent system with a modified output and assume without loss of generality that $u(t) = 0$, $t > 0$. We thus consider the zero-input response

$$y(t) = ce^{At}x(0), \quad 0 \leq t \leq t_f$$

At $t = 0$ we have $y(0) = cx(0)$, and taking derivatives we obtain $\dot{y}(0) = cAx(0)$, $\ddot{y}(t) = cA^2x(0)$ etc, and thus

$$\begin{bmatrix} y(0) \\ \dot{y}(0) \\ \ddot{y}(0) \\ \vdots \\ y^{(n-1)}(0) \end{bmatrix} = \begin{bmatrix} c \\ cA \\ cA^2 \\ \vdots \\ cA^{n-1} \end{bmatrix} x(0) \tag{4.5}$$

This equation can be uniquely solved for $x(0)$ if the observability matrix has full rank. This proves the first claim.

To show that observability implies rank $\mathcal{O}(c, A) = n$, assume that the system is observable but rank $\mathcal{O}(c, A) < n$. Then there exists a column vector $x_0 \neq 0$ such that $\mathcal{O}(c, A)x_0 = 0$, this implies $cA^i x_0 = 0$ for $i = 0, 1, \ldots, n-1$. Now assume that $x(0) = x_0$, then by Theorem 2.2 we have $y(t) = ce^{At}x_0 = 0$. Therefore, the initial state cannot be determined from the output $y(t) = 0$ and the system is unobservable by Definition 4.1, which contradicts the assumption. This completes the proof.

Having established the non-singularity of the observability matrix as a necessary and sufficient condition for observability, we can now derive the following results from Theorems 3.1 and 3.2 simply by making the replacements $A \to A^T$ and $b \to c^T$.

Theorem 4.3

The following statements are equivalent:

(i)   *The system (c, A) is observable.*

(ii) *The system (A^T, c^T) is controllable.*

(iii) *The observability Gramian*

$$W_o(t) = \int_0^t e^{A^T \tau} c^T c e^{A\tau} d\tau$$

is positive definite for $t > 0$.

(iv) *The observability matrix $\mathcal{O}(c, A)$ has full rank.*

(v)   *The matrix*

$$\begin{bmatrix} sI - A \\ c \end{bmatrix} \tag{4.6}$$

has full column rank for all $s \in \mathbb{C}$.

Statement (v) is the dual of the PBH test for controllability.

4.2   Unobservable Systems

To explore the analogy between controllability and observability further, we will now establish a decomposition result for unobservable systems. Consider the example

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} u, \qquad y = [c_1 \quad c_2 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

This system is not observable because the output $y(t)$ is completely independent of the state variable $x_3(t)$. On the other hand, the initial values of the state variables $x_1(t)$ and $x_2(t)$ can be determined from the input and output if $\lambda_1 \neq \lambda_2$ and c_1, c_2 are non-zero. Again, the question is how to identify observable subsystems in general. The answer is given by the following dual version of Theorem 3.5.

Theorem 4.4

Consider the state space model (4.1) and assume that rank $\mathcal{O}(c, A) = r < n$. Then there exists a similarity transformation

$$x = T_o \bar{x}, \qquad \bar{x} = \begin{bmatrix} \bar{x}_o \\ \bar{x}_{\bar{o}} \end{bmatrix}$$

such that

$$\begin{bmatrix} \dot{\bar{x}}_o \\ \dot{\bar{x}}_{\bar{o}} \end{bmatrix} = \begin{bmatrix} \bar{A}_o & 0 \\ \bar{A}_{21} & \bar{A}_{\bar{o}} \end{bmatrix} \begin{bmatrix} \bar{x}_o \\ \bar{x}_{\bar{o}} \end{bmatrix} + \begin{bmatrix} \bar{b}_o \\ \bar{b}_{\bar{o}} \end{bmatrix} u, \qquad y = [\bar{c}_o \ \ 0] \begin{bmatrix} \bar{x}_o \\ \bar{x}_{\bar{o}} \end{bmatrix} \tag{4.7}$$

with $\bar{A}_o \in \mathbb{R}^{r \times r}$ and $(\bar{c}_o, \bar{A}_o)$ observable. Moreover, the transfer function of the system is

$$G(s) = c(sI - A)^{-1}b = \bar{c}_o(sI - \bar{A}_o)^{-1}\bar{b}_o.$$

The transformation T_o is a dual version of T_c. The inverse of a such a transformation T_o is

$$T_o^{-1} = \begin{bmatrix} c^T A^T c^T \dots A^{rT} c^T & q_{r+1}^T \dots q_n^T \end{bmatrix}^T$$

For any similarity transformation T such that $x = T\tilde{x}$, we have

$$\mathcal{O}(\tilde{c}, \tilde{A}) = \mathcal{O}(cT, T^{-1}AT) = \mathcal{O}(c, A)T$$

and the dual version of Theorem 3.6 is

Theorem 4.5

Observability is invariant under similarity transformations.

Example 3.2 continued

Consider again the system with state space realization

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(t), \qquad y(t) = [1 \ \ 1] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

The system is not observable, we have

$$\mathcal{O}(c, A) = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

and rank $\mathcal{O} = 1$. To bring this system in the form of (4.7), we construct the transformation matrix T_o^{-1} by taking c as first row and choosing the second row orthogonal to c. Thus

$$T_o^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad T_o = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$$

Applying this transformation yields

$$\bar{A} = T_o^{-1} A T_o = -T_o^{-1} T_o = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \bar{b} = T_o^{-1} b = \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \quad \bar{c} = c T_o = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

This realization has the required zeros in $\bar{A}$ and $\bar{c}$, and

$$\mathcal{O}(\bar{c}, \bar{A}) = \begin{bmatrix} 2 & 0 \\ -2 & 0 \end{bmatrix}$$

Observable Subspace

In the previous chapter we introduced the concept of a controllable subspace. As one would expect, there is a dual concept associated with observability. We will illustrate it with the above example. From (4.5) we have

$$\begin{bmatrix} y(0) \\ \dot{y}(0) \end{bmatrix} = \mathcal{O}(c, A) x_0$$

Assuming that

$$x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{and} \quad u(t) = 0$$

we have $x_1(t) = x_2(t) = e^{-t}$ and $y(t) = x_1(t) + x_2(t) = 2e^{-t}$. Pretending we do not know the value of x_0, we can try use the above to find an estimate $\hat{x}_0$ of the initial state vector from the observed output; for this purpose we write

$$\mathcal{Y} = \mathcal{O} \, \hat{x}_0 \tag{4.8}$$

Substituting $y(0) = 2$ and $\dot{y}(0) = -2$ yields

$$\begin{bmatrix} 2 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} \hat{x}_{10} \\ \hat{x}_{20} \end{bmatrix}$$

A solution $\hat{x}_0$ to this linear equation exists if and only if

$$\mathcal{Y} \in \mathcal{R}(\mathcal{O})$$

i.e. if $\mathcal{Y}$ is in the range space of the observability matrix. Of course, we know that in general a solution *must exist*, because there must have been some initial value x_0 of the state vector at time $t = 0$ that generated the output data in $\mathcal{Y}$ (and for this example we know the solution anyway). More insight is obtained by considering the *null space* of the observability matrix

$$\mathcal{N}(\mathcal{O}) = \{x : \mathcal{O}x = 0\}$$

By inspection, the null space in this example is

$$\mathcal{N}(\mathcal{O}) = \mathcal{R}\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right)$$

Therefore, all vectors

$$\hat{x}_0 = x_0 + \alpha \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

with arbitrary $\alpha \in \mathbb{R}$ will satisfy (4.8). This is a line through the point x_0 and the direction of the vector $[1 \quad -1]^T$. If we did not know the value of x_0, we could from the observed output only conclude that it must be on this line. In general, the null space of the observability matrix is called the *unobservable subspace*, and its complement, the row space of the observability matrix, is called the *observable subspace*. For an observable system, the observable subspace coincides with the entire state space.

Detectability

We now define the dual concept to stabilizability.

Definition 4.2

The system with state space realization (4.1) is said to be detectable *if there exists a gain vector l such that all eigenvalues of $A + lc$ are in the left half plane.*

If (4.1) is unobservable and (4.7) is another realization of the same system, then the system is detectable if and only if $\bar{A}_{\bar{o}}$ has no eigenvalues in the right half plane. The decomposition of a state space model into an observable and an unobservable subsystem is shown in Fig. 4.3.

4.3   Kalman Canonical Decomposition

If a system is both uncontrollable and unobservable, then a careful combination of Theorem 3.5 and Theorem 4.4 can be used to show that there exists a similarity transformation
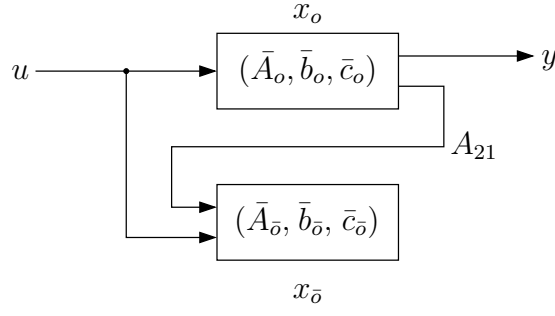
Figure 4.3: Decomposition of unobservable realization

that takes the system into the following form

$$\begin{bmatrix} \dot{\bar{x}}_{co} \\ \dot{\bar{x}}_{c\bar{o}} \\ \dot{\bar{x}}_{\bar{c}o} \\ \dot{\bar{x}}_{\bar{c}\bar{o}} \end{bmatrix} = \begin{bmatrix} \bar{A}_{co} & 0 & \bar{A}_{13} & 0 \\ \bar{A}_{21} & \bar{A}_{c\bar{o}} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & 0 & \bar{A}_{\bar{c}o} & 0 \\ 0 & 0 & \bar{A}_{43} & \bar{A}_{\bar{c}\bar{o}} \end{bmatrix} \begin{bmatrix} \bar{x}_{co} \\ \bar{x}_{c\bar{o}} \\ \bar{x}_{\bar{c}o} \\ \bar{x}_{\bar{c}\bar{o}} \end{bmatrix} + \begin{bmatrix} \bar{b}_{co} \\ \bar{b}_{c\bar{o}} \\ 0 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} \bar{c}_{co} & 0 & \bar{c}_{\bar{c}o} & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_{co} \\ \bar{x}_{c\bar{o}} \\ \bar{x}_{\bar{c}o} \\ \bar{x}_{\bar{c}\bar{o}} \end{bmatrix} \quad (4.9)$$

Of the four subsystems, it is only the controllable and observable subsystem $(\bar{A}_{co}, \bar{b}_{co}, \bar{c}_{co})$ that determines the input-output behaviour of the system. A block diagram interpretation of this decomposition is shown in Fig. 4.4; the dashed lines indicate interaction between subsystems due to the non-zero off-diagonal blocks in $\bar{A}$.
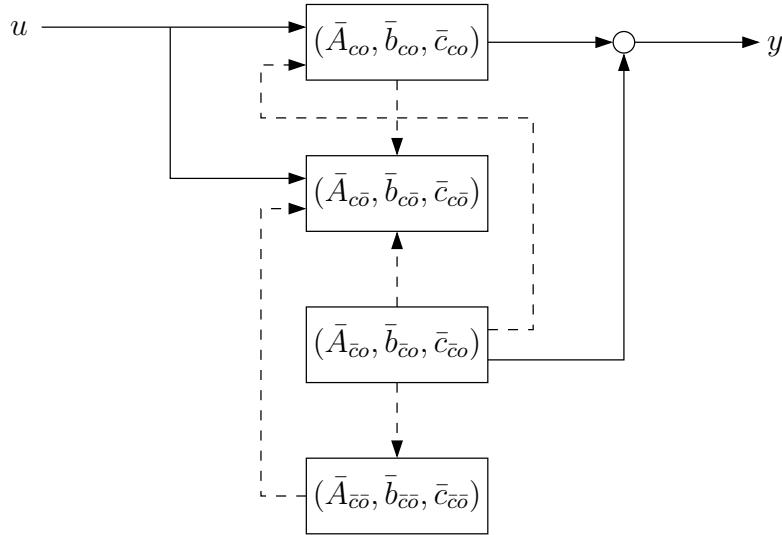
Figure 4.4: Kalman decomposition of uncontrollable and unobservable realization

The transfer function of the system is

$$G(s) = c(sI - A)^{-1}b = \bar{c}_{co}(sI - \bar{A}_{co})^{-1}\bar{b}_{co}$$

The transfer function of the realization (A, b, c) is of order n, whereas that of $(\bar{A}_{co}, \bar{b}_{co}, \bar{c}_{co})$ is of order $r < n$, where r is the dimension of $\bar{x}_{co}$. The fact that they are equal shows that

a pole-zero cancellation must occur in the transfer function of an uncontrollable and/or unobservable realization. This motivates the following definition.

Definition 4.3

A realization (A, b, c) is said to be a minimal realization *if it has the smallest order, i.e. the smallest number of state variables, among all realizations having the same transfer function $c(sI - A)^{-1}b$.*

From this definition and the above discussion of the Kalman decomposition, we conclude the following.

Theorem 4.6

A realization (A, b, c) is minimal if and only if (A, b) is controllable and (c, A) is observable.

Exercises

Problem 4.1

Determine whether the system below is stabilizable, and whether it is detectable.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} 2 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

Problem 4.2

Consider the system in Figure 4.5. Here m is the mass of the pendulum, M is the mass of the trolley, d the position of the trolley, θ the pendulum angle, F the coefficient of friction, and L the distance from the center of gravity of the pendulum to the point of connection to the trolley. With the state variables

$$x^T = \begin{bmatrix} d & \dot{d} & d + L\theta & \dot{d} + L\dot{\theta} \end{bmatrix}$$

the system dynamics - linearized about the upright pendulum position - can be described by the state space model

$$\dot{x} = Ax + bu, \quad y = cx$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -F/M & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -g/L & 0 & g/L & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 0 \end{bmatrix}$$

Note that this linearized model represents the system dynamics only in a small neighborhood of the equilibrium.

a) If the angle θ is available as a measured signal, what is the output vector c of the state space model?

b) What are the eigenvalues of the system?

   Hint: Use the fact that the top right 2×2 submatrix is zero

c) With the parameter values $M = 2$, $F = 0.1$, $L = 0.5$ use `Matlab` to program the PBH test from Problem 3.5, and show that the system is controllable but not observable. What is the eigenvalue of the unobservable mode?

d) Construct a new state space model with the state variables

$$x^T = \begin{bmatrix} \dot{d} & L\theta & L\dot{\theta} & d \end{bmatrix}$$

   Show that the state variable d is unobservable. Give a physical interpretation of this unobservability.
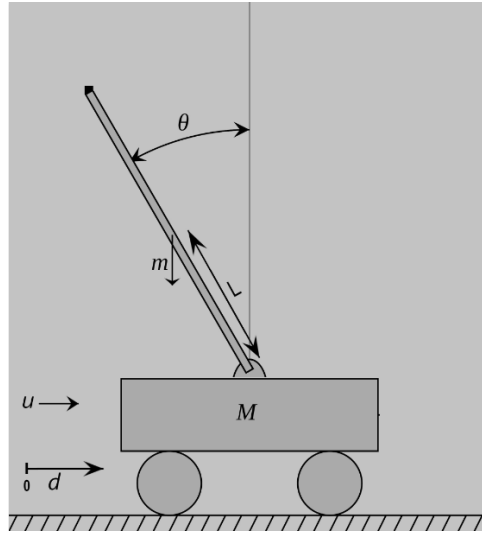
Figure 4.5: Pendulum on cart

e)   Show that the system is observable when the measured variable is $d + L\theta$ instead of θ.

⋆ Problem 4.3

Consider two *minimal* realisations of SISO systems $S_1 = (A_1, b_1, c_1)$ and $S_2 = (A_2, b_2, c_2)$ that represent the same transfer function, with similarity transforms T_{c1} and T_{c2} that transform S_1 and S_2 respectively to the controller canonical form S_c.

a)   What is the similarity transform that transforms S_1 to S_2?

b)   Prove that if two minimal realisations represent the same transfer function, then a matrix T exists which can transform S_1 to S_2.

Problem 4.4

Consider the transfer function

$$G(s) = \frac{s + 1}{(s + 1)(s + 2)} = \frac{s + 1}{s^2 + 3s + 2}$$

a)   What does this pole-zero cancellation tell us about a state space realization of this transfer function?

b)   Construct the controller and the observer canonical forms corresponding to this transfer function and discuss the controllability and observability of these two state space models.

Chapter 5

Observer-Based Control

In this chapter we will combine the ideas of state feedback and state estimation to construct a controller that uses only the measured plant output as feedback signal. We investigate the dynamic properties of the closed-loop system, and we will see that the dynamics of state feedback control and of state estimation can be designed independently. Tracking of reference inputs will also be discussed; this will lead us to a discussion of the zeros of a state space model and how they are determined by the observer configuration.

5.1 State Estimate Feedback

Fig. 5.1 shows a state feedback loop where feedback of the state vector (see Fig. 2.6) has been replaced by feedback of an estimated state vector. From the figure, it can be seen that the dynamic behaviour of the closed-loop system is governed by the plant equation

$$\dot{x} = Ax + bf\hat{x} + bu_v, \quad x(0) = x_0$$

and the observer equation

$$\dot{\hat{x}} = (A + bf + lc)\hat{x} - lcx + bu_v, \quad \hat{x}(0) = \hat{x}_0$$

The state variables of the closed-loop system are the state variables of the plant and the observer. Introducing the closed-loop state vector $[x^T \ \hat{x}^T]^T$, the closed-loop state and output equations can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & bf \\ -lc & A+bf+lc \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} b \\ b \end{bmatrix} u_v, \quad \begin{bmatrix} x(0) \\ \hat{x}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \hat{x}_0 \end{bmatrix}$$

$$y = \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$
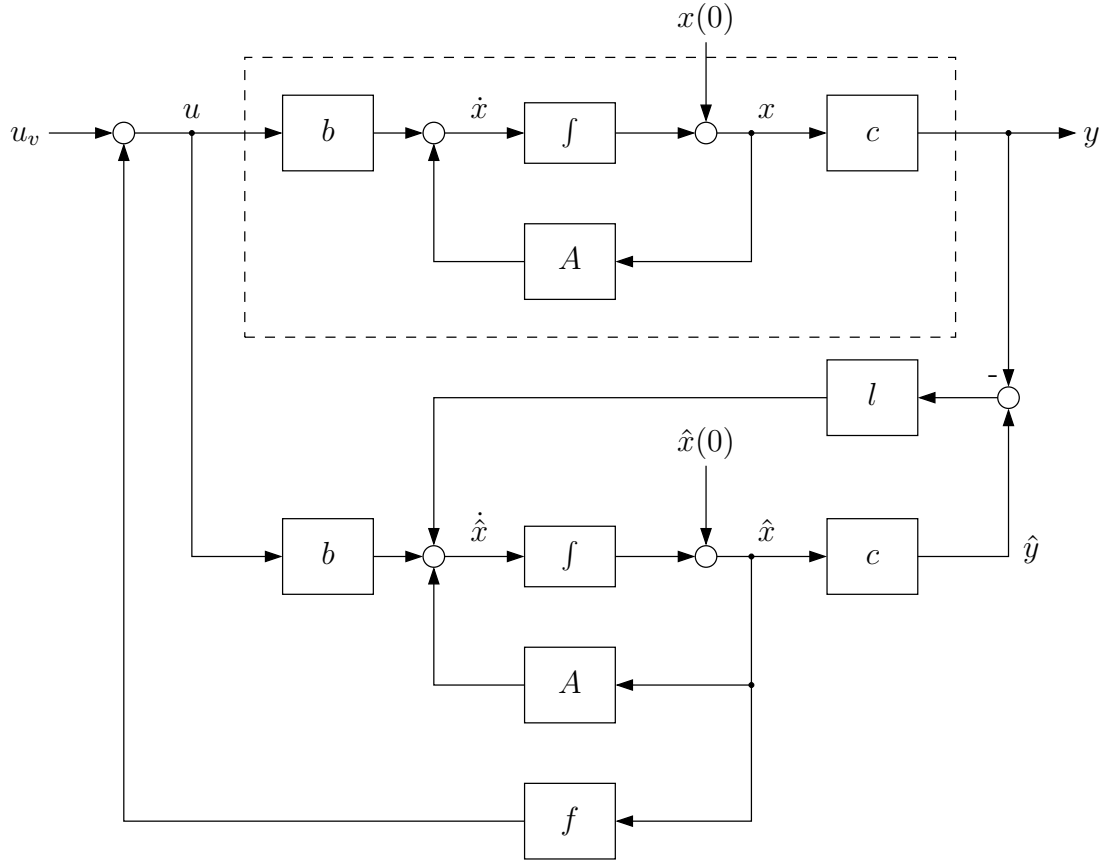
Figure 5.1: Observer-based state feedback

More insight into the structure of this system is gained by applying to this state space model of the closed-loop system the similarity transformation

$$\begin{bmatrix} x \\ \hat{x} \end{bmatrix} = T \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}, \qquad T = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$$

Under this transformation, the observer state vector $\hat{x}$ is replaced by the estimation error $\tilde{x}$, and the closed-loop state and output equations become

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A + bf & -bf \\ 0 & A + lc \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u_v, \qquad \begin{bmatrix} x(0) \\ \tilde{x}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \tilde{x}_0 \end{bmatrix} \tag{5.1}$$

$$y = \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$$

The block triangular form of the system matrix reveals that the eigenvalues of the closed-loop system are the eigenvalues of $(A + bf)$ together with the eigenvalues of $(A + lc)$. But these are precisely the eigenvalues assigned by state feedback and the observer eigenvalues, respectively. An important conclusion is that state feedback and state observer can be designed independently; this fact is referred to as the *separation principle*.

A second observation is that the state space model (5.1) has the same form as the controllable/uncontrollable decomposition of Theorem 3.5. The estimation error $\tilde{x}$ is uncontrollable from the input u_v, whereas the system state x is controllable if (A, b) is controllable, because - as shown in Exercise 3.10 - the system $(A + bf, b)$ is controllable if (A, b) is controllable. The estimation error is observable: a non-zero initial error will have an effect on the output. The closed-loop transfer function from u_v to y is

$$G_{cl}(s) = c(sI - A - bf)^{-1}b$$

which is the same transfer function as that achieved with direct state feedback. This is of course a consequence of the fact that the observer is not controllable from the input: if the initial estimation error is zero, the estimated state vector used for feedback is identical with the actual plant state vector.

Choice of Observer Eigenvalues

The observer eigenvalues determine the speed at which the estimate error decays; they should obviously be placed to the left of the plant eigenvalues, so that the estimates can track the evolution of the state variables. The question is how fast one can make the observer - the limiting factor is measurement noise. Placing observer eigenvalues far to the left will lead to a high gain at high frequencies and thus amplify noise. Choosing the location of the observer eigenvalues is therefore a trade-off between speed of estimation and sensitivity to sensor noise; this is illustrated in Exercise 5.1.

5.2 Reference Tracking

When the idea of state feedback was introduced in Chapter 2, the objective was to modify the dynamic properties of a system by moving its eigenvalues to desired locations. We will now discuss how observer-based state feedback can be used to design a control system for tracking a reference input. The question is how to introduce the reference input into the loop. One possibility is shown in Fig. 5.2, this loop has the structure of a standard control loop with unity feedback.
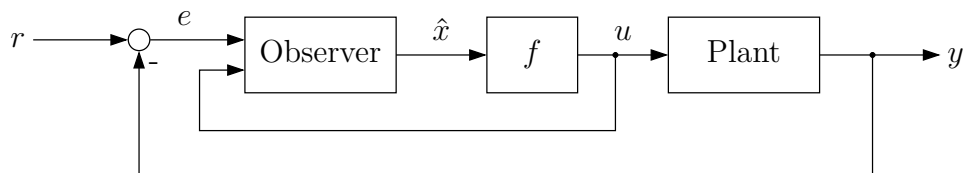
Figure 5.2: Introduction of reference input

An alternative way of introducing the reference input into the loop is shown in Fig. 5.3. The difference between both configurations can be seen by considering a step function

$r(t) = \sigma(t)$ as reference input. In both configurations we assume that the plant output will not change instantaneously in response to a changing plant input. In Fig. 5.2, the reference step will then result in a step change in the control error e, which in turn excites the observer dynamics but not the plant, and leads to an estimation error instantaneously after the reference step is applied. In contrast, in Fig. 5.3 plant and observer are excited in exactly the same way; no estimation error is introduced by the reference step. We would therefore expect that the configuration in Fig.5.3 is superior to that in Fig.5.2.
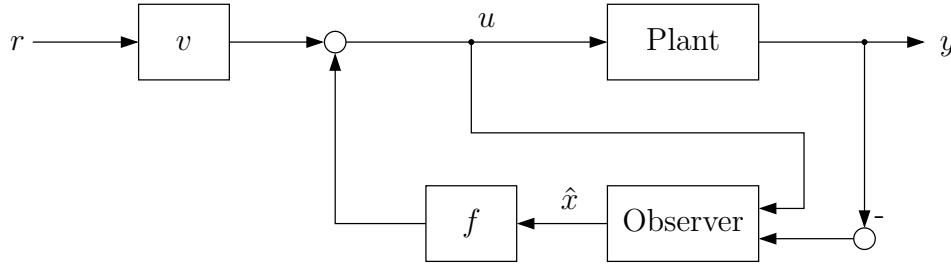
Figure 5.3: Alternative introduction of reference input

Zeros of State Space Models

Since the loop itself is the same in Fig. 5.2 and Fig. 5.3, the closed-loop eigenvalues are the same - they are the eigenvalues of $(A + bf)$ and $(A + lc)$. The difference between both configurations is due to the fact that they have different *zeros*. Before we discuss the zeros of a state space model, we briefly review the concept of a zero of a transfer function model. Consider a system with transfer function

$$G(s) = \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)} \tag{5.2}$$

Assume that an input signal $u(t) = u_0 e^{s_0 t} \sigma(t)$ is applied to this system. Here s_0 can be real or complex. If s_0 is complex there will also be an input component corresponding to its complex conjugate; here we consider for simplicity only a single real component. The Laplace transform of the input is

$$U(s) = \frac{u_0}{s - s_0}$$

resulting in the output

$$Y(s) = \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)} \cdot \frac{u_0}{s - s_0} \tag{5.3}$$

or in time domain

$$y(t) = (k_1 e^{p_1 t} + k_2 e^{p_2 t} + k_0 e^{s_0 t}) \sigma(t) = y_{\text{trans}}(t) + y_{\text{ext}}(t)$$

where $y_{\text{trans}}(t) = (k_1 e^{p_1 t} + k_2 e^{p_2 t}) \sigma(t)$ is the transient component of the response - determined by the system poles - and $y_{\text{ext}}(t) = k_0 e^{s_0 t} \sigma(t)$ is the component of the response that reflects the direct effect of the external input signal $u(t)$.

A zero of the system (5.2) is a value of s for which $G(s) = 0$; in the above case there are zeros at $s = z_1$ and $s = z_2$. In time domain, another way of looking at these zeros is to assume that s_0 in the exponent of the above input signal is equal to one of the zeros, i.e. $u(t) = u_0 e^{z_i t} \sigma(t)$ where $i = 1$ or $i = 2$. In (5.3) the factor $s - s_0 = s - z_i$ in the denominator will disappear, thus $y_{\text{ext}}(t) = 0$, $t \geq 0$ and $y(t) = y_{\text{trans}}(t)$ - only the transient component will appear in the output.

Now assume that

$$\dot{x} = Ax + bu, \quad x(0) = x_0$$
$$y = cx + du \tag{5.4}$$

is a state space realization of this system. Note that a non-zero feedthrough term du is included in the output equation, reflecting the assumption that $G(s)$ in (5.2) has an equal number of poles and zeros; to consider direct feedthrough will turn out to be useful when we discuss the closed-loop zeros of the control loops in Fig. 5.2 and Fig. 5.3. Applying again the input $u(t) = u_0 e^{s_0 t} \sigma(t)$ will give a response with Laplace transform

$$Y(s) = c\Phi(s)x_0 + (c\Phi(s)b + d)\frac{u_0}{s - s_0} \tag{5.5}$$

where $\Phi(s) = (sI - A)^{-1}$. The second term on the right hand side - the zero-initial-state response - is precisely the response in (5.3), which can be decomposed into $y_{\text{trans}}(t)$ and $y_{\text{ext}}(t)$. If $s_0 = z_i$, $i = 1, 2$, then with the above input we have $y_{\text{ext}}(t) = 0$, and the output signal $y(t)$ consists of the zero-input response and the transient component of the zero-initial-state response. Compared with the transfer function response (5.3), there is an additional term - the zero-input response $c\Phi(s)x_0$ - present in the output, and we could ask the question whether it is possible that y_{trans} and $c\Phi(s)x_0$ cancel each other. In other words: can the response to $u(t) = u_0 e^{z_i t} \sigma(t)$ be made zero for all $t \geq 0$ by a suitable choice of x_0?

The answer to this question is yes: if the initial state is

$$x_0 = \Phi(z_i)bu_0 \tag{5.6}$$

the output (5.5) will vanish completely. To see this, substitute the above in (5.5) to get

$$Y(s) = c\left(\Phi(s)\Phi(z_i) + \Phi(s)\frac{1}{s - z_i}\right)bu_0 + du_0\frac{1}{s - z_i}$$

It can then be shown that

$$\Phi(s)\Phi(z_i) + \Phi(s)\frac{1}{s - z_i} = \Phi(z_i)\frac{1}{s - z_i}$$

thus

$$Y(s) = (c\Phi(z_i)b + d)u_0\frac{1}{s - z_i} = 0 \tag{5.7}$$

because $c\Phi(z_i)b + d = G(z_i) = 0$, and we have $y(t) = 0$, $t \geq 0$.

The above discussion shows that it is possible to make the output vanish by choosing x_0 as in (5.6) if z_i is a zero of the transfer function model of the system and the input is $u(t) = u_0 e^{z_i t} \sigma(t)$. (It can in fact be shown that this is true *only if* z_i is a zero of the transfer function.) We can use this fact for a definition of the zeros of a state space model that is consistent with the concept of transfer function zeros.

Definition 5.1

The complex number z is a zero of the model (5.4) if and only if there exists an initial state x_0 such that the response to the input vector $u(t) = u_0 e^{zt} \sigma(t)$ is zero for all $t \geq 0$.

From (5.7) we see that if z is a zero, and the input $u(t) = u_0 e^{zt}$ is applied for $t \geq 0$, the state vector will be $x(t) = x_0 e^{zt}$, and for the output we have $y(t) = 0$. Substituting these in (5.4) yields

$$\dot{x} = z x_0 e^{zt} = A x_0 e^{zt} + b u_0 e^{zt}$$

and

$$y = c x_0 e^{zt} + d u_0 e^{zt} = 0$$

After dividing by e^{zt} this can be written as

$$\begin{bmatrix} zI - A & -b \\ c & d \end{bmatrix} \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Therefore, z is a zero if and only if the matrix on the left hand side is singular.

Theorem 5.1

A complex number s is a zero of the model (5.4) if and only if

$$\det \begin{bmatrix} sI - A & -b \\ c & d \end{bmatrix} = 0$$

A value of s that satisfies the condition of this theorem is called an *invariant zero* of the system, whereas a value of s such that $G(s) = 0$ is called a *transmission zero*. This distinction will be taken up again in Definitions 6.5 and 6.6. One can show that if a state space realization is minimal, invariant and transmission zeros are the same, but if a realization has uncontrollable or unobservable modes, then there will be invariant zeros that do not appear as transmission zeros.

5.3   Closed-Loop Transfer Function

Returning to the reference tracking problem, we can now use Theorem 5.1 to study the closed-loop zeros of the configurations in Fig. 5.2 and Fig. 5.3. We will do this however in a more general framework. The controller is determined by the state feedback gain vector f and the observer with gain vector l. Referring to Fig. 5.1, a state space model of the controller with input y and output u is

$$\dot{\hat{x}} = (A + bf + lc)\hat{x} - ly$$
$$u = f\hat{x}$$

The most general way of introducing a reference input signal r into the control loop is to add r to both the state equation and the output equation of the controller, this leads to

$$\dot{\hat{x}} = (A + bf + lc)\hat{x} - ly + wr$$
$$u = f\hat{x} + vr \qquad\qquad (5.8)$$

Fig. 5.4 shows the resulting control loop. Here w and v are a gain vector and a constant, respectively, that can be chosen by the designer. The configurations in Fig. 5.2 and Fig. 5.3 can now be obtained as special cases, see Exercise 5.3 and the discussion below.
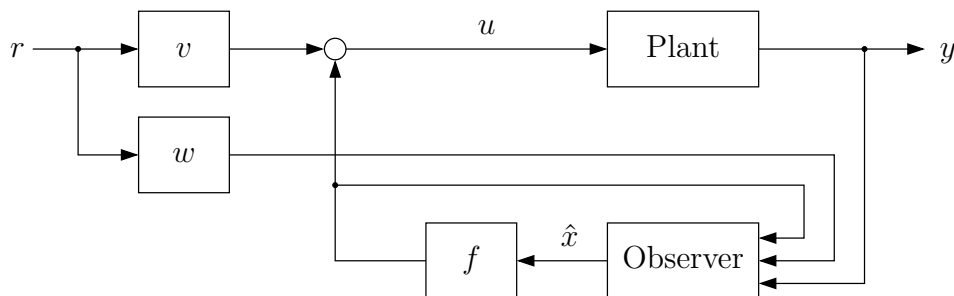
Figure 5.4: General structure for introducing reference input

It is clear that the choice of w and v has no influence on the closed-loop eigenvalues, but after the discussion above we expect it to have an effect on the closed-loop zeros. To study this effect, we first observe that the control loop can be redrawn as in Fig. 5.5, and that any zero of the controller, i.e. any zero from r to u will also be a zero from r to y unless it is cancelled by a pole of the plant. The closed-loop zeros are therefore - if no pole-zero cancellation occurs - the controller zeros together with the plant zeros.

We can now use Theorem 5.1 to find the zeros of the controller. With the state space model (5.8) of the controller, and assuming $y = 0$ (because we are interested in the dynamic behaviour from r to u), we obtain as condition for s to be a zero of the controller
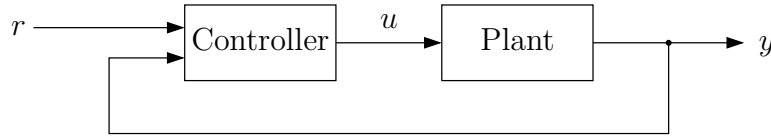
Figure 5.5: Control loop

- and therefore of the closed loop

$$\det \begin{bmatrix} sI - A - bf - lc & -w \\ f & v \end{bmatrix} = 0$$

The roots of the determinant polynomial are not changed if we perform row or column operations on the matrix, so we can divide the last column by v and - after multiplying from the right by f - subtract it from the first block column. This yields

$$\det \begin{bmatrix} sI - A - bf - lc + \frac{1}{v}wf & -\frac{1}{v}w \\ 0 & 1 \end{bmatrix} = 0$$

or

$$\gamma(s) = \det(sI - A - bf - lc + \frac{1}{v}wf) = 0 \tag{5.9}$$

Introducing $A_{fl} = A + bf + lc$ and $\tilde{w} = -w/v$, this last equation tells us that - after f and l have been fixed - we can choose the vector $\tilde{w}$ to arbitrarily assign the closed-loop zeros, i.e. the eigenvalues of $A_{fl} + \tilde{w}f$, in the same way that we can choose l to assign the observer eigenvalues, provided (f, A_{fl}) is observable.

To summarize the discussion of closed-loop zeros in this section, we have seen that the closed-loop transfer function from r to y in Fig. 5.4 is

$$G_{cl}(s) = \frac{Y(s)}{R(s)} = K\frac{\gamma(s)b(s)}{\bar{a}(s)\bar{a}_e(s)} \tag{5.10}$$

where $\gamma(s)$ is the polynomial in (5.9), K a constant gain, $\bar{a}(s) = \det(sI - A - bf)$ the characteristic polynomial of the system under direct state feedback and $\bar{a}_e(s) = \det(sI - A - lc)$ the observer characteristic polynomial. This explains why the control loop in Fig. 5.3 is superior to that in Fig. 5.2: in Fig. 5.3 we have $w = vb$, thus $\gamma(s) = \bar{a}_e(s)$ and the observer dynamics are cancelled in the closed-loop transfer function. In fact the closed-loop system in 5.3 is equivalent to that in Fig. 5.1 if we take $u_v = vr$, and the pole-zero cancellation in (5.10) is equivalent to the fact observed earlier that the estimation error in Fig. 5.1 is uncontrollable from the input u_v. Note that the static gain from r to y in Fig.5.3 is $-vc(A + bf)^{-1}b$; to achieve a zero steady-state tracking error one should therefore choose

$$v = -\frac{1}{c(A + bf)^{-1}b}.$$

5.4 Symmetric Root Locus Design

In this and the previous two chapters we established that when a system is controllable and observable, we can design an observer-based state feedback controller and place the plant and observer eigenvalues at will by choice of the gain vectors f and l. This is a powerful result that opens the door to efficient design techniques. However, when faced directly with the question of where to place the plant and observer eigenvalues for a given control problem, it often turns out to be surprisingly difficult to achieve a good design by choosing plant and observer poles in a trial and error fashion. Of course, we can approximate the closed-loop behaviour by a dominant pole pair and use guidelines from classical control to translate design specifications like damping ratio or rise time into required pole locations. However, for higher order systems that still leaves the question of what to do about the remaining poles. Moreover, one will find that in general it is not a good idea to choose closed-loop pole locations without considering the open-loop poles and zeros. We have seen from equation (3.9) for example that the feedback gains (and thus the required control effort) will increase if the closed-loop poles are moved far away from their open-loop locations. But physical actuator limitations will always impose a constraint on the controller design, and require a trade-off between performance and control effort.

In fact, the real power of state space design methods reveals itself when the design is based on a systematic procedure, which usually involves the search for a controller that is *optimal* - i.e. the best that can be achieved - in the sense of a given performance measure. While a rigorous treatment of optimal controller design methods is beyond the scope of this course, we will conclude this chapter with a brief presentation of a widely used method for choosing pole locations, which for single-input single-output systems can be used in a way similar to root locus design. The main result is presented here without proof - this problem and its solution is discussed in detail in the lecture course "Optimal and Robust Control".

Linear Quadratic Regulators for SISO Systems

Consider a system described by the state space model

$$
\begin{aligned}
\dot{x} &= Ax + bu, \quad x(0) = x_0 \\
y &= cx
\end{aligned}
\tag{5.11}
$$

where (A, b) is stabilizable and (c, A) is detectable. Assume that the desired setpoint for this system is $y = 0$, but that at time $t = 0$ some external disturbance has driven the system away from its equilibrium to a state $x(0) = x_0 \neq 0$ and thus $y \neq 0$. We wish to find a control input $u(t)$ that brings the system back to its equilibrium at $x = 0$ as quickly as possible, but with "reasonable" control effort. One way of assessing the capability of a

controller to achieve this objective is to evaluate the *cost function*

$$V = \int_0^\infty (y^2(t) + \rho u^2(t)) \, dt, \qquad \rho > 0 \tag{5.12}$$

The first term under the integral represents the control error, and the second term the control effort. The parameter ρ - a positive constant - is a tuning parameter that allows the designer to adjust the balance between the weights attached to control error and control effort in this cost function. The *optimal controller* in the sense of this cost function is the controller that minimizes V. It is clear that if ρ is chosen to be large, the optimal controller will use less control effort (control being "expensive") and thus achieve a slower response than an optimal controller with a smaller value of ρ in the cost function.

It turns out that the optimal control law takes the form of state feedback control

$$u_{\mathrm{opt}}(t) = fx(t)$$

where f is the optimal state feedback gain. The following Theorem - presented here without proof - provides a characterization of the optimal controller in terms of the optimal closed-loop eigenvalues.

Let $G(s) = b(s)/a(s)$ denote the transfer function of the system (5.11), i.e. $G(s) = c(sI - A)^{-1}b$. Then one can show the following

Theorem 5.2

The optimal state feedback controller $u_{\mathrm{opt}}(t) = fx(t)$ *that minimizes the cost function (5.12) for the system (5.11) places the eigenvalues of* $A + bf$ *at the stable roots (the roots in the left half plane) of the polynomial*

$$p(s) = a(-s)a(s) + \frac{1}{\rho}b(-s)b(s) \tag{5.13}$$

Before we discuss how to use this Theorem for optimal controller design, we will have a closer look at the properties of the polynomial $p(s)$. If $x(t) \in \mathbb{R}^n$, the open loop characteristic polynomial $a(s)$ has degree n, and $p(s)$ will have degree $2n$. Thus $p(s)$ has $2n$ roots. It is easy to see that if s_0 is a zero of $p(s)$, then $-s_0$ is also a zero of $p(s)$. This means that in the complex plane, the roots of $p(s)$ are symmetric with respect to the imaginary axis - half of the roots are stable and the other half unstable. The n stable roots are the optimal eigenvalues, and Theorem 5.2 provides a direct way of computing these from the open-loop model. Knowing the optimal eigenvalues, we can compute the optimal state feedback gain f by solving a pole placement problem.

We would face a difficulty, however, if it turns out that $p(s)$ has roots on the imaginary axis, because then the roots could not be divided into n stable and n unstable ones. That

this situation does not arise is guaranteed by the following Theorem, the proof of which is left to Exercise 5.9.

Theorem 5.3

The polynomial $p(s)$ has no roots on the imaginary axis if (A, b) is stabilizable and (c, A) is detectable.

When designing an optimal controller, the tuning parameter ρ in the cost function plays an important role. To display the effect of ρ on the closed-loop eigenvalues graphically, divide the equation $p(s) = 0$ by $a(-s)a(s)$ to obtain

$$1 + \frac{1}{\rho}G(-s)G(s) = 0 \tag{5.14}$$

The optimal eigenvalues are the values of s in the left half plane that satisfy this equation. Now recall that the effect of the controller gain K on the closed-loop eigenvalues of a control system with loop transfer function $L(s)$ is displayed as the root locus of the characteristic equation $1 + KL(s) = 0$. Comparing this with (5.14), we see that if we make the replacements

$$L(s) \to G(-s)G(s), \quad K \to \frac{1}{\rho}$$

we can use standard root locus techniques to display the optimal eigenvalue locations for values of ρ in the range $0 < \rho < \infty$. This optimal root locus is symmetric with respect to the imaginary axis, the root locus branches in the left half plane represent the optimal eigenvalues.

In summary, an optimal state feedback controller for the system (5.11) with cost function (5.12) can be designed as follows.

- Use standard root locus tools to plot the root locus of $1 + \frac{1}{\rho}G(-s)G(s) = 0$

- Choose a set of optimal closed-loop eigenvalues from the root locus plot (this is a choice of ρ and thus a decision on the trade-off between performance and control effort)

- Compute the optimal state feedback gain by solving a pole placement problem.

This design technique is known as *symmetric root locus* technique, and the resulting optimal state feedback controller is called a *linear quadratic regulator* (LQR).

Optimal State Estimation

Having designed an optimal state feedback controller is only half the solution of the controller design problem - we still need to choose observer eigenvalues if the controller is to be implemented using output measurement. We have seen that state estimation is the dual problem to state feedback; this raises the question whether a dual version of the symmetric root locus design technique is available for observer design. This is indeed the case. State feedback design involves a trade-off between control performance and control effort, for which the symmetric root locus provides the optimal choices. There is a dual trade-off in state estimation: recall the discussion in Section 5.1 about a compromise between speed of estimation and sensitivity to sensor noise. If observer eigenvalues are too slow (i.e. slower than the poles assigned by the controller), then the system response would be determined by the observer rather than the controller. On the other hand, if the observer poles are too fast, measurement noise will lead to excessive control action, which can lead to mechanical wear.

Since the observer design problem involves noise processes, it is useful to extend the plant model in the following way. Consider the extended model

$$\dot{x} = Ax + bu + b_n n_x$$
$$y = cx + n_y \tag{5.15}$$

where n_x and n_y are white noise processes representing process and measurement noise, respectively. Both noise processes are assumed to be wide-sense stationary, zero-mean, Gaussian distributed and uncorrelated, with spectral densities S_x and S_y, respectively. A review of stochastic processes is provided in the appendix.

The input vector b_n describes how the process noise affects the individual state variables. The observer equation for this stochastic plant model is

$$\dot{\hat{x}} = A\hat{x} + bu + lc(\hat{x} - x) - ln_y$$

Subtracting this from (5.15) yields

$$\dot{\tilde{x}} = (A + lc)\tilde{x} + b_n n_x + ln_y \tag{5.16}$$

This equation again illustrates the two conflicting objectives of state estimation. To reduce the estimation error, the observer should be fast enough to track the state movements induced by the process noise n_x; this requires large gains in the vector l. But because l multiplies n_y, large gains in turn lead to amplification of measurement noise. An optimal balance - given the spectral densities of process and measurement noise - can again be obtained by using a symmetric root locus technique. Let $G_n(s)$ denote the transfer function from process noise to measured output, i.e. $G_n(s) = c(sI - A)^{-1}b_n$, and let $q = S_x/S_y$ denote the ratio of noise spectral densities. Then one can show the following.

Theorem 5.4

The observer gain l that minimizes

$$\lim_{t\to\infty} E[\tilde{x}^T(t)\tilde{x}(t)]$$

places the eigenvalues of $A + lc$ at the stable solutions (the solutions in the left half plane) of

$$1 + qG_n(-s)G_n(s) = 0 \qquad (5.17)$$

The term $E[\tilde{x}^T\tilde{x}]$ is a measure of the "size" of the estimation error. An observer designed according to Theorem 5.4 to minimize the estimation error is also known as a *Kalman filter*. An output feedback controller obtained by combining an LQR state feedback gain with a Kalman filter is referred to as a *linear quadratic Gaussian* (LQG) controller. Some insight into the nature of optimal state feedback and estimation in the above sense can be gained by considering the limiting cases $\rho \to \infty$, $\rho \to 0$ and $q \to \infty$, $q \to 0$; this is explored in Exercises 5.7 and 5.8.

Exercises

Problem 5.1

Let the state space representation of a plant model be given as

$$\dot{x} = Ax + bu$$
$$y = cx$$

where,

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad c = \begin{bmatrix} -1 & 0 \end{bmatrix}$$

a)   Find the state feedback gain matrix f to place the closed loop poles at $[-0.5\pm0.5j]$.

b)   Find state observer gain matrix l to place the observer poles at

   i) $s = -10 \pm 10j$
   ii) $s = -1 \pm 1j$

c)   Find the bode plots of the closed loop system from

   i)   the input disturbance d to the output y
   ii)   the output noise n to the output y

d)   Use SIMULINK to simulate the response of the closed loop system to initial state $x(0)$ with and without white noise of power 0.001. Where,

$$x(0) = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

   Explain the results in terms of the frequency response.

e)   Find observer poles to achieve a response with $|y(t)| < 0.05$ for $t > 7.0$ s and $|u(t)| < 0.3$ for all $t \geq 0$.

Problem 5.2

For the system $\dot{x} = Ax + bu, \ y = cx$, where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -4 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad c = \begin{bmatrix} 5 & 4 & 1 \end{bmatrix}$$

a)   Calculate the poles and zeros of the system.

b)   Calculate the input and initial conditions that produce zero output: $y(t) = 0, t \geq 0$.

        Hint: Use complex conjugates to find real input signals associated with complex zeros.

c)   Simulate the system with these initial conditions in Matlab.

Problem 5.3

Consider the two degree of freedom (2DOF) system in Figure 5.6, where $\hat{x}$ is the state estimate and r is the setpoint. The details of the observer are shown in Figure 5.7.

A state space realization of the system $G(s)$ is

$$A = \begin{bmatrix} -2 & 1 \\ 0 & -3 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 & 3 \end{bmatrix}$$
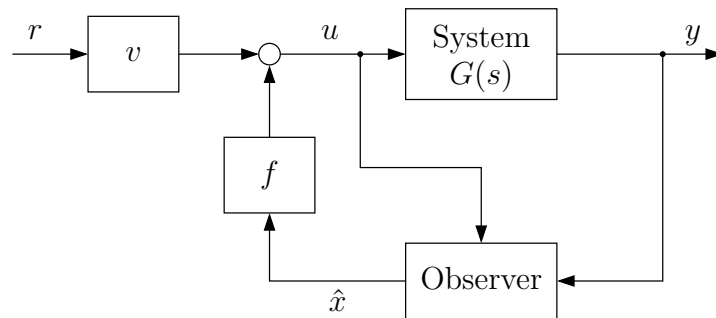
Figure 5.6: 2DOF arrangement 1

a)   Design the controller and the observer so that the closed-loop poles are at $-3 \pm 3j$ and the poles of the observer are at $-10 \pm 10j$.

b)   Construct a state space model of the closed-loop system, in Figure 5.6, from r to y. What can be said about the controllability of the state estimation error from the input r?

        Determine analytically the closed-loop transfer function from r to y

c)   Calculate the constant v such that $y(\infty) = r$.

d)   Consider the closed loop system as shown in Figure 5.8, where instead of $l(\hat{y} - y)$ we have $l(\hat{y} - y + r)$ but the controller and the observer gain matrices are same as found above.
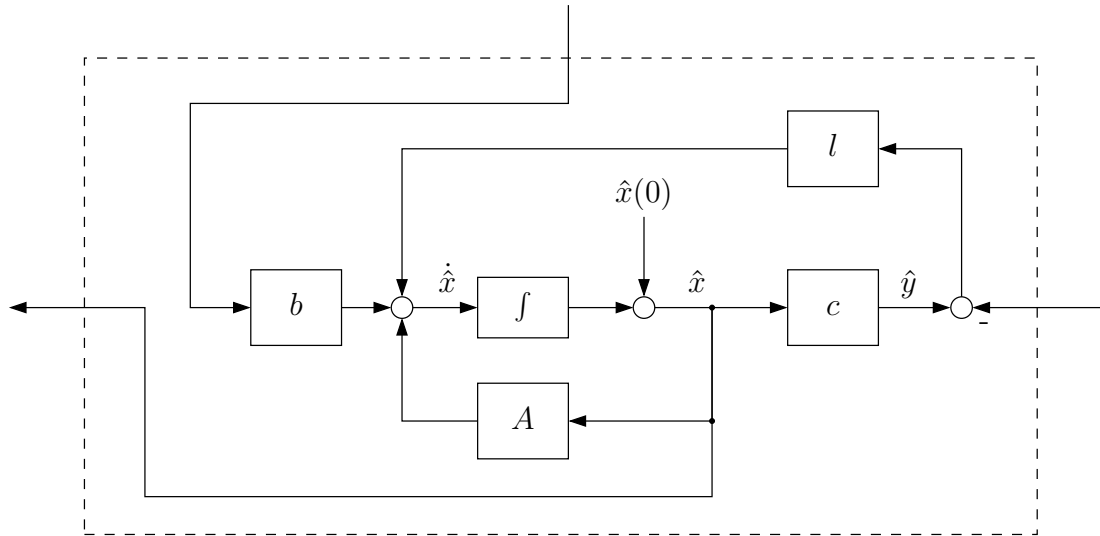
Figure 5.7: Observer detail

What is the relationship between the poles of the new closed-loop system from r to y and the designed closed-loop and observer pole positions?
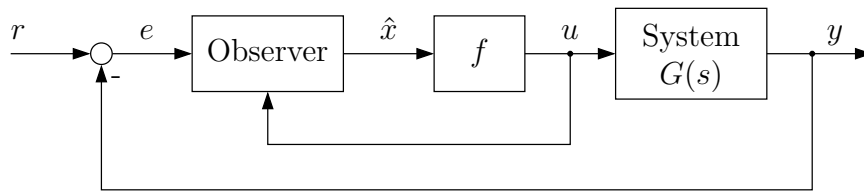
Figure 5.8: 2DOF arrangement 2

Hint: Use a similarity transformation to make the $[2,1]$ block of the closed loop system matrix zero.

What are the zeros of the closed-loop system? Why do the open-loop zeros appear in the closed-loop system?

e)   Use Matlab to compare the step response of the observer states for the observer configurations in Figures 5.6 and 5.8. Use f and l designed in part (a). What is the effect of the open-loop zeros on the response?

f)   Show that the behaviour of the system in Figure 5.9 is described by

$$\dot{\hat{x}} = (A + bf + lc)\hat{x} - ly + wr$$
$$u = f\hat{x} + vr$$

where, the observer structure is shown in Figure 5.10.
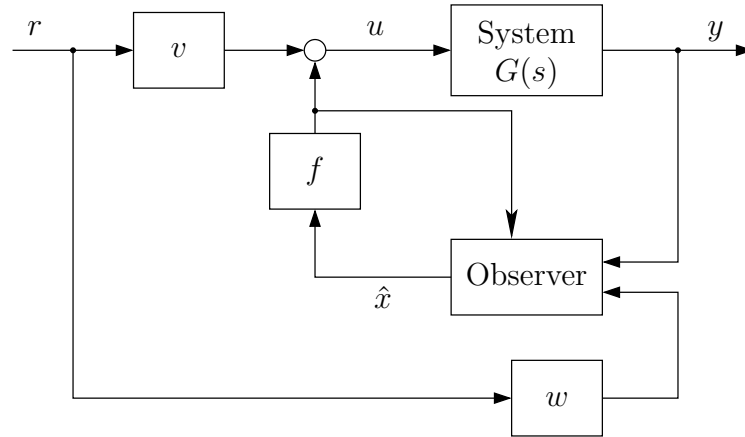
Figure 5.9: 2DOF arrangement 3

How should the values of v and w be chosen to achieve the closed-loop behaviour identical to that in Figure 5.6?

g) How should v and w be chosen if $e = r - y$ is used by the observer (as in Figure 5.8)?

Problem 5.4

Consider a controllable and observable system

$$\dot{x} = Ax + bu, \qquad y = cx$$

a) How are the poles and zeros of this system related to the numerator and denominator of its transfer function?

b) Under what conditions could such a system become unobservable under state feedback?

Hint: Consider the transfer function of a system with state feedback.

Problem 5.5

Assume that the controller, the observer and the gain v are the same as in Problem 5.3 parts (a) and (b).

a) Consider Figure 5.11 with the control loop broken at point 'BP'. Calculate the following open loop transfer functions analytically, and use the command `linmod` to confirm your answers.

   i) $G_{yd}(s)$ from d to y
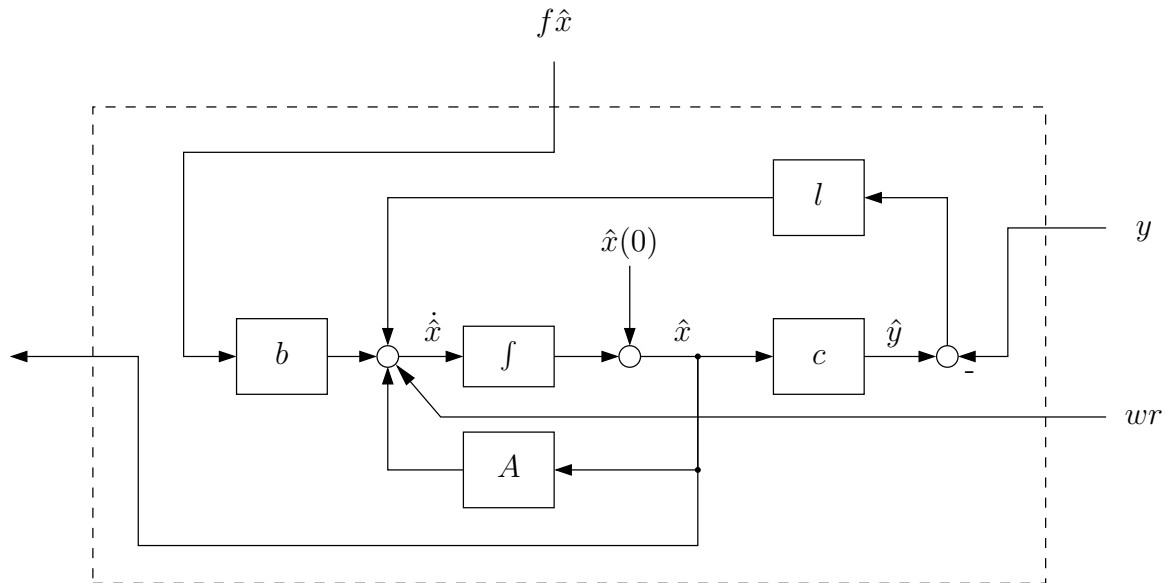
Figure 5.10: 2DOF arrangement 3 observer detail
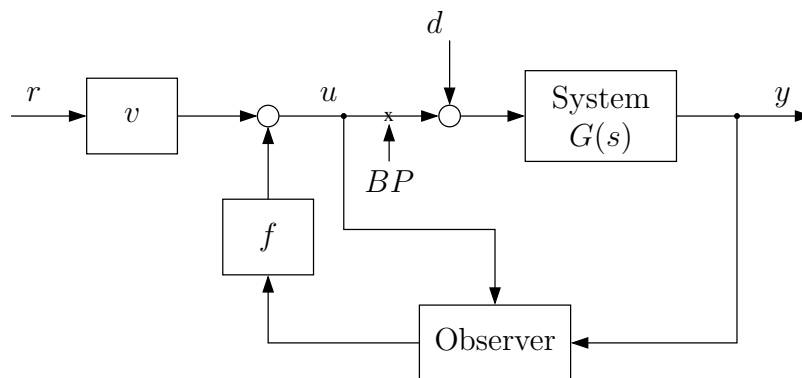
Figure 5.11: 2DOF controller

  ii) $G_{ur}(s)$ from r to u

  iii) $G_{uy}(s)$ from y to u

b) With the loop closed, write the transfer function from r to y as a function of $G_{yd}(s)$, $G_{ur}(s)$ and $G_{yu}(s)$. What happens to the zeros of $G_{ur}(s)$ in closed loop?

c) How large is the error

$$\lim_{t \to \infty} (y - r),$$

when the disturbance d in Figure 5.11 has the constant value d_0 and $r = 0$?

d) Assume that the system matrix of the plant is replaced by $(1 + \epsilon)A$ (representing model uncertainty). With $r(t) = \sigma(t)$ and $\epsilon = 0.1$ use `Matlab` to calculate $\lim_{t \to \infty}(y - r)$

e)   The system is augmented by an integrator as shown in Figure 5.12. Use Simulink
     to build a simulation model of this system. Use `Matlab` to design the controller
     components f and f_I to place the closed loop poles in the positions

$$s = -5, \quad s = -3 + 3j, \quad s = -3 - 3j$$

     *Hint: The integrator adds an extra state variable x_I to the plant, so it is possible to
     assign three pole locations with the composite controller $\bar{f} = \begin{bmatrix} f & f_I \end{bmatrix}$*

f)   With the controller from part (e), how large is the error

$$\lim_{t \to \infty} (y - r),$$

     with constant reference input r, constant disturbance d or constant $\epsilon = 0.05$?
     Compare the results with ones in parts (c) and (d).

     Hint: It is not necessary to run a simulation to answer this question
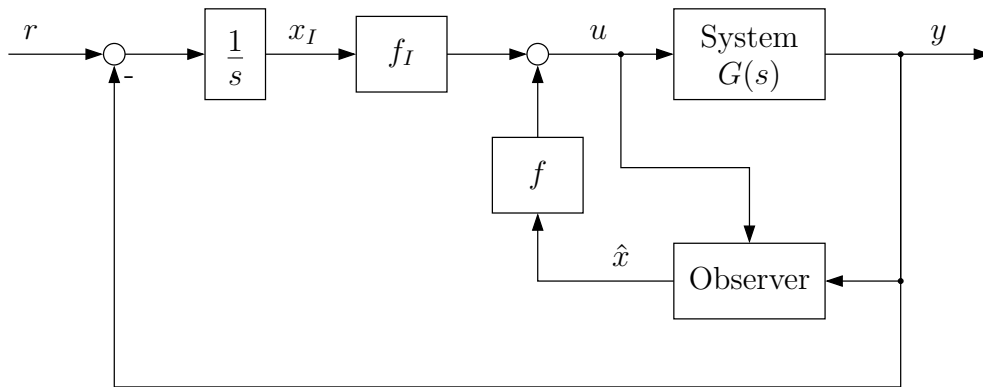
Figure 5.12: Addition of integrator

Problem 5.6

An inverted pendulum can be described by a state space model

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The first state variable is the position of the pendulum and the second state variable is
its speed. In the performance index

$$J = \int_0^\infty [z^2(t) + \rho u^2(t)] dt$$

z is two times the position of the pendulum added to its speed.

a)   Plot the symmetric root locus for this performance index using `Matlab`.

b)   Use `Matlab` to design an optimal state feedback controller and constant gain pre-filter v (similar to that in Figure 5.11) such that

   i)   the steady state tracking error is zero, i.e.

$$\lim_{t \to \infty} (y - r) = 0$$

   ii)   following a unit step input $r(t) = \sigma(t)$, the 95% rise time is less than 2.5 and the maximum input magnitude $|u|$ is less than 3.0.

c)   A stochastic model of this system is

$$\dot{x} = Ax + bu + b_n n_x$$
$$y = cx + n_y$$

where n_x and n_y are white noise processes and

$$b_n = \begin{bmatrix} 0.1 & 1.0 \end{bmatrix}^T, \qquad S_x = 0.2, \qquad S_y = 5 \cdot 10^{-6}$$

Design a Kalman filter (using `Matlab`) to minimize $E[\tilde{x}^T \tilde{x}]$, where $\tilde{x}$ is the estimation error.

d)   Simulate the response of the closed-loop system to a unit step input in $r(t)$. Use the block `band_limited_white_noise` in Simulink with `sample time=0.01` to simulate the white noise signals.

e)   Consider now the system with the deterministic disturbance d instead of the stochastic noises n_x and n_y

$$\dot{x} = Ax + bu + b_n d$$
$$y = cx$$

Although the system is now deterministic, it is still possible to design a Kalman filter for the system, but now the noise covariance matrices are fictitious and can be used as tuning parameters to adjust the overall control response.

With q defined as

$$q = \frac{S_x}{S_y}$$

in a Kalman filter design, tune q such that with a pulse disturbance $d(t) = \sigma(t) - \sigma(t-1)$ the maximum magnitude of y during the transient is 1.0 (with $n_y = 0$). Describe the connection between the trade-off required here and the trade-off in part (c).

Problem 5.7

The symmetric root locus displays the roots of the polynomial

$$p(s) = a_c(s)a_c(-s) = a(-s)a(s) + \frac{1}{\rho}b(s)b(-s)$$

where $b(s)/a(s)$ is the open-loop transfer function and $b_c(s)/a_c(s)$ is the closed-loop transfer function under optimal state feedback.

a)   Discuss the relationship between open-loop and closed-loop poles as $\rho \to \infty$. Assume that the plant has stable and unstable open-loop poles. Give a physical explanation for the behaviour as $\rho \to \infty$ for both optimal controller and Kalman filter design ($\rho \to \infty$ in a controller design is equivalent to $q \to 0$ in a Kalman filter design).

b)   Consider the system

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

    i)   Show that of the four roots of $p(s)$ only two remain finite as $\rho \to 0$. Determine the value of the finite root of the closed loop system characteristic under optimal state feedback control as $\rho \to 0$.

    ii)   Show that the 'expanding' stable root is at the position

$$\lim_{\rho \to 0} s = -\frac{1}{\sqrt{\rho}}$$

    iii)   Give an explanation for the behaviour as $\rho \to 0$ when the symmetric root locus design method is used for optimal controller and Kalman filter design (in a Kalman filter design this is equivalent to $q \to \infty$).

Problem 5.8

Consider again the polynomial $p(s)$ of Problem 5.7. Let

$$a(s) = s^n + a_{n-1}s^{n-1} + \ldots + a_0$$

$$b(s) = b_m s^m + b_{m-1}s^{m-1} + \ldots + b_0$$

where $m < n$, and assume $\rho \to 0$.

a)   Show that $P(s) = a_c(s)a_c(-s)$ has $2m$ finite roots. What are their locations?

b) Show that for large values of s the roots of the polynomial $p(s)$ tend towards the roots of

$$(-1)^n s^{2n} + \frac{1}{\rho}(-1)^m b_m^2 s^{2m} = 0$$

Hint: For large s, you can assume that $a(s) \approx s^n$ and $b(s) \approx b_m s^m$

d) Show that in general the roots are complex with magnitudes

$$|s| = r = \left(\frac{b_m^2}{\rho}\right)^{\frac{1}{2(n-m)}}$$

e) For $n - m = 1$ show that the infinite root of $a_c(s)$ is $-r$.

f) For $n - m = 2$ show that the infinite roots of $a_c(s)$ are $\frac{1}{\sqrt{2}}r(-1 \pm j)$

Problem 5.9

Consider a system with transfer function

$$G(s) = \frac{b(s)}{a(s)}$$

a) Assume that the symmetric root locus polynomial $p(s)$ has a root on the imaginary axis. Show that this implies that

$$|a(jw)|^2 + \frac{1}{\rho}|b(jw)|^2 = 0$$

b) Show that this can only be true if $a(s)$ and $b(s)$ can be factored as

$$a(s) = \tilde{a}(s)(s - j\omega_0)$$
$$b(s) = \tilde{b}(s)(s - j\omega_0)$$

c) Use (b) to show that $p(s)$ has no roots on the imaginary axis if the system is stabilizable and detectable.

Hint: Show that there is a contradiction between the symmetric root locus having roots on the imaginary axis and the plant being stabilizable and detectable.

Problem 5.10   *Mini Segway exercise.*

Read and understand the MATLAB script `Task_2_Simulation_LQR_Design.m` and Simulink file `Task_2_Simulation_LQR.slx`. The MATLAB files simulate and plot the closed-loop response of the linear and nonlinear models of the Mini Segway.

a) Design an LQR state feedback controller to stabilize the system given the vectors of initial conditions $x_{0,1} = [0; 5\pi/180; 0; 0]^T$ and $x_{0,2} = [0; 9\pi/180; 0; 0]^T$.

   i) Compare the response of the linear and nonlinear-model.

   ii) Tune your controller to have a settling time for which $|s(t)| < 0.01$ of $t_s < 1.5$s.

   iii) Why are the differences between the linear and non-linear model more obvious with initial conditions $x_{0,2}$?

b) Set the parameter *sinewave* in the m-file to 1 (this will enable a position reference sinusoidal input). Re-tune the controller to minimize the mean square error to be $e_{RMS} < 0.004$ between the reference input and the position state

$$\frac{1}{n} \sum_{i=1}^{n} (r_i - x_i)^2$$

Compare the value of the linear and nonlinear-models.

Problem 5.11 *Mini Segway exercise.*

Run the Matlab script `Experiment_parameters.m` and open the Experiment simulink file `Experiment_LQR.slx`.

a) Implement the controller designed in 5.10b. Run the experiment and extract the states from the simulink experimental model. Compare between the simulation and the experiment. *(Hint: export the data as a structure with time as a vector* [states;Control input;reference] *and name the data* "expOut" *so that you can use implemented code).*

Chapter 6

Multivariable Systems

The previous chapters introduced the basic concepts of state space methods for the analysis and design of control systems; however, the discussion was limited to single-input single-output systems. A major advantage of state space methods is the fact that multi-input multi-output systems can be handled within the same framework. In this chapter we will extend the concepts and results introduced so far to MIMO systems.

6.1 Transfer Function Models

As an example of a multivariable system, a simplified model of a turbogenerator is shown in Fig. 6.1. A gas turbine is driving a synchronous generator; the gas inflow is controlled by a servo valve (control input u_1). A second control input (u_2) is the adjustable excitation current. Measured outputs are the turbine speed (y_1) and the rectified voltage across a symmetric load (y_2).

Each control input has an effect on both measured outputs. Assuming that the plant behaviour has been linearized about a given operating point, a dynamic model of this system includes four linear sub-models, describing the dynamics from each input to each output, i.e. gas inflow $\rightarrow$ turbine speed, gas inflow $\rightarrow$ output voltage, excitation current $\rightarrow$ turbine speed and excitation current $\rightarrow$ output voltage. If both turbine speed and output voltage are to be controlled independently, then an efficient control strategy should take each of these four sub-models into account. A compact way of representing the overall dynamics of the plant is to collect input and output signals into an input and an output signal vector, respectively. A transfer function model can then be written as

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}$$

where each linear sub-model is represented by a single-input single-output transfer function model $G_{ij}(s)$.
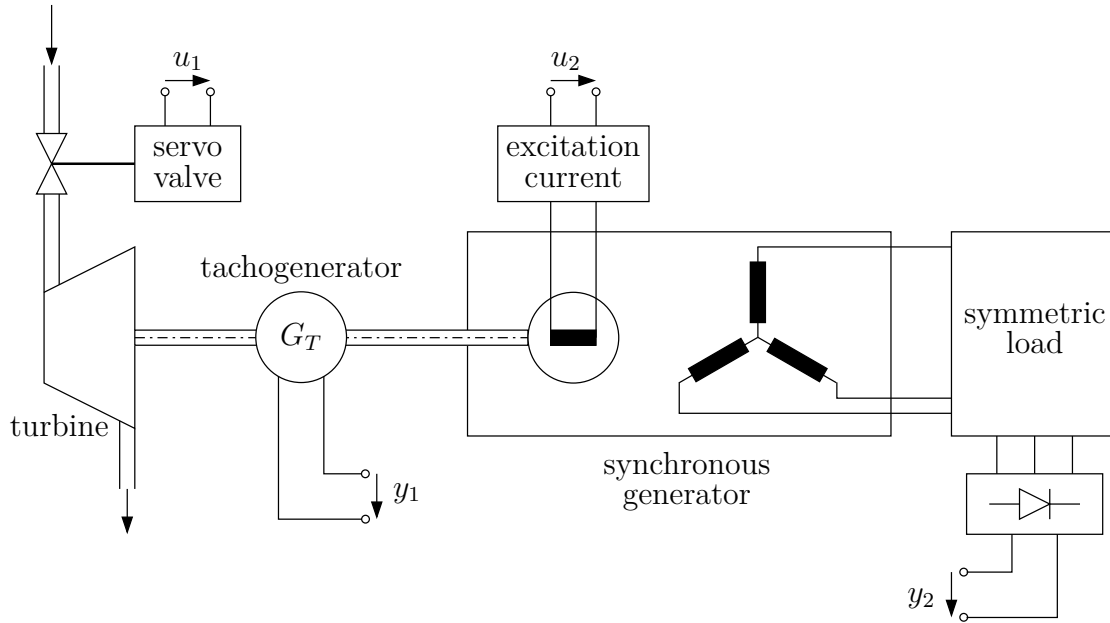
Figure 6.1: Turbogenerator

Multivariable systems can be inerconnected in the same way as single-input single-output systems. Care must however be taken because matrix multiplication is not commutative. For two given systems with transfer functions $G_1(s)$ and $G_2(s)$, some basic connections are shown in Figures 6.2 - 6.4.

A series connection is shown in Figure 6.2. This connection is possible if the dimensions of y_1 and u_2 are equal, an equivalent system is then $G_2(s)G_1(s)$. Note that in general this is different from $G_1(s)G_2(s)$, the latter product is not even defined if the dimensions of y_2 and u_1 are not the same.
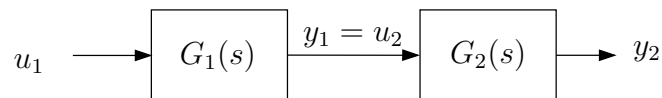
Figure 6.2: Blocks in series

Figure 6.3 shows a parallel connection. If both systems have the same number of inputs and outputs, this connection is possible and an equivalent system is $G_1(s) + G_2(s)$.

The feedback loop shown in Figure 6.4 requires that the dimensions of y_1 and u_2 as well as the dimensions of y_2 and u_1, respectively, are the same. Solving

$$y_1 = G_1(r + G_2 y_1)$$

for y_1 yields

$$y_1 = (I - G_1 G_2)^{-1} G_1 r \tag{6.1}$$
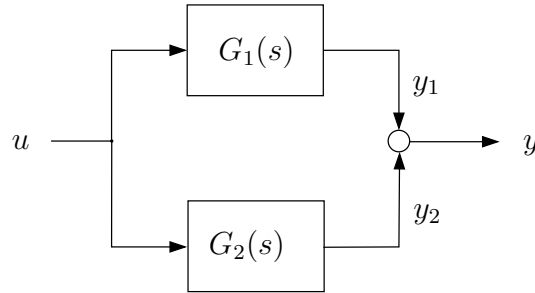
Figure 6.3: Blocks in parallel

On the other hand, if we solve

$$u_1 = r + G_2 G_1 u_1$$

for u_1 we obtain

$$y_1 = G_1 (I - G_2 G_1)^{-1} r \tag{6.2}$$

Note that (6.1) and (6.2) are equivalent expressions for the closed-loop system in Figure 6.4.
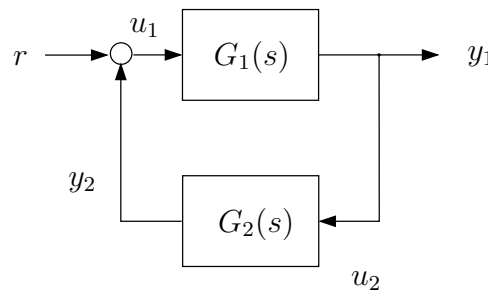
Figure 6.4: Blocks in feedback

6.2 State Space Models

Recall that a multivariable state space model

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

was introduced in Chapter 2, where B and D have as many columns as there are plant inputs, and the number of rows of C and D equals the number of plant outputs. For strictly proper systems we have $D = 0$.

An experimentally identified state space model of a turbogenerator plant as described above is introduced and explored in Exercise 6.3. We will now discuss the conversion

between transfer function models and state space models of multivariable systems. Given a state space model, it is straightforward to check - just as in the SISO case discussed in Chapter 2 - that the associated transfer function is

$$G(s) = C(sI - A)^{-1}B + D \tag{6.3}$$

Obtaining a state space model from a given transfer function matrix is less straightforward, this will be illustrated by an example.

Example 6.1

Consider a plant with two inputs and two outputs, that can be described by the transfer function matrix

$$G(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{2}{s+1} \\ \frac{-1}{(s+1)(s+2)} & \frac{1}{s+2} \end{bmatrix} \tag{6.4}$$

As a first attempt to construct a state space realization of this model, we combine the state space models (A_{ij}, b_{ij}, c_{ij}) of the four SISO transfer functions $G_{ij}(s)$ to obtain the MIMO model. Controller forms of the four sub-models can be written down by inspection as

$$\begin{aligned} G_{11}(s) &\rightarrow (-1, 1, 1) \\ G_{12}(s) &\rightarrow (-1, 1, 2) \\ G_{21}(s) &\rightarrow \left(\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, [-1 \ 0] \right) \\ G_{22}(s) &\rightarrow (-2, 1, 1) \end{aligned}$$

These four models can be arranged as

$$\dot{x} = \begin{bmatrix} A_{11} & & & 0 \\ & A_{12} & & \\ & & A_{21} & \\ 0 & & & A_{22} \end{bmatrix} x + \begin{bmatrix} b_{11} & 0 \\ 0 & b_{12} \\ b_{21} & 0 \\ 0 & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & 0 & 0 \\ 0 & 0 & c_{21} & c_{22} \end{bmatrix} x$$

yielding the MIMO state space model

$$\dot{x} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & -3 & 0 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} x \tag{6.5}$$

The state vector of the MIMO model contains five state variables - the state variables of each sub-model; the order of the MIMO model equals the sum of the orders of the sub-models. One can check that substituting the matrices (A, B, C) of this model in (6.3) yields indeed the transfer function matrix (6.4).

6.3 The Gilbert Realization

We will now present a different way of finding a state space realization of a transfer function model, known as *Gilbert realization*. This realization is only possible if all eigenvalues of the model are distinct, i.e. if no sub-model has repeated eigenvalues. The idea is as follows. First, a transfer function matrix $G(s)$ can be rewritten as

$$G(s) = \frac{1}{d(s)} N(s)$$

where the least common multiple $d(s)$ of all denominator polynomials has been pulled out as a common factor, leaving a "numerator" polynomial matrix $N(s)$. Assuming deg $d(s) = r$, a partial fraction expansion is

$$G(s) = \frac{N_1}{s - \lambda_1} + \frac{N_2}{s - \lambda_2} + \ldots + \frac{N_r}{s - \lambda_r}$$

where the residuals N_i can be computed elementwise. Defining

$$\rho_i = \text{rank } N_i$$

each residual matrix N_i can be factored as

$$N_i = C_i B_i, \quad C_i \in \mathbb{R}^{l \times \rho_i}, \ B_i \in \mathbb{R}^{\rho_i \times m}$$

Now each term in the partial fraction expansion can be written as

$$\frac{N_i}{s - \lambda_i} = C_i \frac{1}{s - \lambda_i} B_i = C_i (s I_{\rho_i} - \lambda_i I_{\rho_i})^{-1} B_i$$

where I_{ρ_i} denotes the $\rho_i \times \rho_i$ identity matrix. Thus we obtain for each term a state space realization

$$\frac{N_i}{s - \lambda_i} \quad \rightarrow \quad (A_i, B_i, C_i)$$

where $A_i = \lambda_i I_{\rho_i}$. A combination of these terms by forming a parallel connection yields a MIMO state space model

$$\dot{x} = \begin{bmatrix} \lambda_1 I_{\rho_1} & & 0 \\ & \ddots & \\ 0 & & \lambda_r I_{\rho_r} \end{bmatrix} x + \begin{bmatrix} B_1 \\ \vdots \\ B_r \end{bmatrix} u$$

$$y = [C_1 \ \ldots \ C_r] x$$

This realization of $G(s)$ has $\rho = \sum_{i=1}^{r} \rho_i$ state variables.

Example 6.1 (continued)

To obtain the Gilbert realization of the model (6.4), we rewrite $G(s)$ as

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

and compute the partial fraction expansion as

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

We have $\rho_1 = 2$, $\rho_2 = 1$. Factorizing N_1 and N_2 as

$$N_1 = C_1 B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}, \qquad N_2 = C_2 B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix}$$

we obtain the MIMO model

$$\dot{x} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} x \tag{6.6}$$

This realization of $G(s)$ in Example 6.1 has three state variables, two less than the realization in (6.5). However, it is straightforward to check that substituting the matrices of this model in (6.3) yields the same transfer function. From the discussion in Chapter 4, we would therefore expect that at least two state variables in (6.5) are either uncontrollable or unobservable. This is indeed the case, but before we can establish this, we need to extend the concepts of controllability and observability to MIMO systems.

6.4   Controllability and Observability

We will now extend the results of Chapters 3 and 4 to MIMO systems. Consider a system with state space realization

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t) \tag{6.7}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^l$, and the matrices (A, B, C, D) are of compatible dimensions.

Definition 6.1

The system (6.7) is said to be controllable if for any initial state $x(0) = x_0$, time $t_f > 0$ and final state x_f there exists a control input $u(t)$, $0 \leq t \leq t_f$, such that the solution of (6.7) satisfies $x(t_f) = x_f$. Otherwise, the system is said to be uncontrollable.

The following can be shown in a way similar to the proofs of the corresponding SISO results in Chapter 3.

Theorem 6.1

The following statements are equivalent:

(i)   *The system (6.7) is controllable.*

(ii)   *The controllability Gramian*

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

   is positive definite for any $t > 0$.

(iii) *The controllability matrix*

$$\mathcal{C}(A, B) = [B \quad AB \quad \ldots \quad A^{n-1}B]$$

   has full row rank.

(iv)   *The matrix*

$$[sI - A \ \ B]$$

   has full row rank for all $s \in \mathbb{C}$.

Notice that in contrast to SISO systems where the controllability matrix $\mathcal{C}(A, b)$ is a $n \times n$ square matrix, the controllability matrix $\mathcal{C}(A, B)$ of a MIMO system is $n \times mn$, i.e. it has more columns than rows. But just as for SISO systems, rank $\mathcal{C}(A, B) = n$ (full row rank) ensures that the controllable subspace - the column space of $\mathcal{C}(A, B)$ - is the whole state space. It is possible that the *partial controllability matrix*

$$\mathcal{C}_r(A, B) = [B \quad AB \quad \ldots \quad A^{r-1}B]$$

where $r < n$, has rank n. In this case the smallest integer ν_c for which rank $\mathcal{C}_{\nu_c}(A, B) = n$ is called the *controllability index* of the system (A, B).

Definition 6.2

The system (6.7) is said to be observable if for any $t_f > 0$ the initial state $x(0)$ can be uniquely determined from the time history of the input $u(t)$ and the output $y(t)$ in the time interval $0 \le t \le t_f$. Otherwise, the system is said to be unobservable.

Theorem 6.2

The following statements are equivalent:

(i)   *The system (6.7) is observable.*

(ii)   *The observability Gramian*

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

is positive definite for any $t > 0$.

(iii)   *The observability matrix*

$$\mathcal{O}(C, A) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has full column rank.

(iv)   *The matrix*

$$\begin{bmatrix} sI - A \\ C \end{bmatrix}$$

has full column rank for all $s \in \mathbb{C}$.

The proof of Theorem 4.2 (the SISO version of the statement (iii) $\Leftrightarrow$ (i) in Theorem 6.2) was based on the fact that the equation

$$\mathcal{Y} = \mathcal{O}(c, A)x(0)$$

where $\mathcal{Y}$ is a column vector containing $y(0)$, $\dot{y}(0)$ etc., can be solved uniquely for $x(0)$ if the observability matrix $\mathcal{O}(c, A)$ has full rank. For MIMO systems the observability matrix is $ln \times n$, i.e. it has more rows than columns. However, multiplying the above from the right by $\mathcal{O}^T$ leads to

$$x(0) = (\mathcal{O}^T \mathcal{O})^{-1} \mathcal{O}^T \mathcal{Y}$$

which shows that rank $\mathcal{O}(C, A) = n$ is also in the MIMO case a sufficient condition for the existence of a unique solution $x(0)$, because rank $\mathcal{O} = n$ implies rank $\mathcal{O}^T \mathcal{O} = n$.

The *partial observability matrix* $\mathcal{O}_r(C, A)$ and the *observability index* ν_o are defined in the same way as the corresponding controllability concepts.

The following can be shown in a way similar to the proofs of Theorems 3.5 and 4.4.

Theorem 6.3

Consider the system (6.7) and assume that rank $\mathcal{C}(A, B) < n$ and $\mathcal{O}(C, A) < n$. There exists a similarity transformation that takes the system into the form

$$
\begin{bmatrix} \dot{\bar{x}}_{co} \\ \dot{\bar{x}}_{c\bar{o}} \\ \dot{\bar{x}}_{\bar{c}o} \\ \dot{\bar{x}}_{\bar{c}\bar{o}} \end{bmatrix} = \begin{bmatrix} \bar{A}_{co} & 0 & \bar{A}_{13} & 0 \\ \bar{A}_{21} & \bar{A}_{c\bar{o}} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & 0 & \bar{A}_{\bar{c}o} & 0 \\ 0 & 0 & \bar{A}_{43} & \bar{A}_{\bar{c}\bar{o}} \end{bmatrix} \begin{bmatrix} \bar{x}_{co} \\ \bar{x}_{c\bar{o}} \\ \bar{x}_{\bar{c}o} \\ \bar{x}_{\bar{c}\bar{o}} \end{bmatrix} + \begin{bmatrix} \bar{B}_{co} \\ \bar{B}_{c\bar{o}} \\ 0 \\ 0 \end{bmatrix} u
$$

$$
y = [\bar{C}_{co} \quad 0 \quad \bar{C}_{\bar{c}o} \quad 0] \begin{bmatrix} \bar{x}_{co} \\ \bar{x}_{c\bar{o}} \\ \bar{x}_{\bar{c}o} \\ \bar{x}_{\bar{c}\bar{o}} \end{bmatrix} + Du \tag{6.8}
$$

where the subsystem $(\bar{A}_{co}, \bar{B}_{co}, \bar{C}_{co}, D)$ is controllable and observable. The transfer function from u to y is

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

The concepts of stabilizability and detectability are defined for MIMO systems in the same way as for SISO systems. Likewise, the following are straightforward extensions of Definition 4.3 and Theorem 4.6

Definition 6.3

A realization (A, B, C, D) is said to be a minimal realization *if it has the smallest order, i.e. the smallest number of state variables, among all realizations having the same transfer function $C(sI - A)^{-1}B + D$.*

Theorem 6.4

A realization (A, B, C, D) is minimal if and only if (A, B) is controllable and (C, A) is observable.

Returning to Example 6.1 and its Gilbert realization, it turns out that (6.3) is a minimal realization, because - as shown in Exercise 6.9 - a Gilbert realization is always controllable and observable.

Multivariable Poles and Zeros

A linear system is stable if all its poles are strictly inside the left half of the complex plane. So far we have not yet extended the concept of poles and zeros to multivariable systems. As Example 6.1 indicated, it may not be trivial to determine what the poles of a multivariable system are. Clearly, in that example we can say by inspection of the transfer function matrix that the poles are located at $s = -1$ and $s = -2$. What is however not immediately clear is the *multiplicity* of the poles at these locations. Exactly how many poles the system has at each of these locations plays for example a role when a multivariable version of the Nyquist stability criterion is used to assess the closed-loop stability of a control system. At first glance, it is not even clear how many poles the system has at all - to answer that question we need to know the degree of its minimal realization.

We will define poles of a multivariable system in terms of a minimal state space realization.

Let (A, B, C, D) be a realization of $G(s)$, and recall that

$$G(s) = \frac{C \operatorname{adj}(sI - A)B}{\det(sI - A)} + D$$

The numerator $C \operatorname{adj}(sI - A)B$ is a polynomial matrix, and it is clear that every pole of $G(s)$ must be a zero of $\det(sI - A)$ (i.e. an eigenvalue of A). However, not every zero of $\det(sI - A)$ needs to be a pole (because there may be cancellations by zeros in the numerator that occur in all subsystems). If this happens, the realization is either uncontrollable or unobservable, which motivates the following definition.

Definition 6.4

Let (A, B, C, D) be a minimal realization of a system with transfer function matrix $G(s)$. The eigenvalues of A are called the poles of $G(s)$.

In Section 5.2 we introduced the distinction between invariant zeros and transmission zeros.

Definition 6.5

Let (A, B, C, D) be a realization of $G(s)$. $G(s)$ has an invariant zero at z_i if z_i satisfies

$$\det \begin{bmatrix} z_i I - A & -B \\ C & D \end{bmatrix} = 0.$$

Definition 6.6

$G(s)$ has a transmission zero at z_i if there exists a vector $u_0 \neq 0$ such that $G(z_i)u_0 = 0$.

Recall from Section 5.2 that if a state space realization is minimal, invariant and transmission zeros are the same, but if a realization has uncontrollable or unobservable modes, then there will be invariant zeros that do not appear as transmission zeros.

Multivariable poles and zeros are usually defined in terms of the *Smith-McMillan form* of a system which is discussed in the next section. One can prove that the poles according to Definition 6.4 and the transmission zeros according to Definition 6.6 are the same as those defined in terms of the Smith-McMillan form.

6.5 The Smith-McMillan Form

We will now introduce an alternative definition of multivariable poles and zeros. Consider again the system of Example 6.1. The transfer function is

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

where $d(s)$ is the least common multiple of all denominator polynomials. Obviously, the poles of this system should be defined such that the poles of the individual elements of $G(s)$ are included; what is less obvious is with what multiplicity these poles - at $s = -1$ and $s = -2$ in the above example - should be counted as poles of the MIMO system. Even less obvious are the zeros of this system. The multivariable poles and zeros are usually defined in terms of a diagonal canonical form to which every transfer function matrix can be reduced, known as the *Smith-McMillan form*.

The Smith-McMillan Form

To introduce the Smith-McMillan form, we start with the factorization $G(s) = N(s)/d(s)$, where $N(s)$ is a polynomial matrix and $d(s)$ the least common multiple of the denominators. The following Theorem states that every polynomial matrix can be transformed into a special diagonal form by elementary row or column operations. (Elementary row or column operations are the interchange of two rows or columns, and the addition of a polynomial multiple of a row or column to another.)

Theorem 6.5

A polynomial matrix $N(s)$ of dimension $l \times m$ can always be transformed by a sequence of elementary column and row operations into the form

$$\Lambda(s) = \begin{bmatrix} \beta_1(s) & & & 0 \\ & \ddots & & \\ & & \beta_r(s) & \\ 0 & & & 0 \end{bmatrix} \tag{6.9}$$

where the $\beta_i(s)$ are unique monic *polynomials (i.e. the highest power of s has coefficient 1) such that $\beta_i(s) \,|\, \beta_{i+1}(s)$ ($\beta_i(s)$ divides $\beta_{i+1}(s)$ without remainder) for $i = 1, \ldots, r$, and r is the rank of $N(s)$ for almost all s.*

The matrix $\Lambda(s)$ is known as the *Smith form* of $N(s)$. To prove the Theorem, we outline how the Smith form can be constructed. By interchange of columns and rows, bring the element of $N(s)$ with least degree to the (1,1) position. Use elementary row operations to make all entries in the first column below the (1,1) entry zero. Use elementary column operations to make all entries in the first row except the (1,1) entry zero. These column operations may bring back non-zero elements to the first column. In this case repeat the above steps until all elements of row 1 and column 1 are zero except the (1,1) entry. This yields a matrix of the form

$$\begin{bmatrix} \beta_1(s) & 0 \\ 0 & N_1(s) \end{bmatrix}$$

where $\beta_1(s)$ divides every element of $N_1(s)$. Repeat the whole procedure on $N_1(s)$. Proceeding in this way leads to the Smith form $\Lambda(s)$ of $N(s)$.

We will say that $N(s)$ is *similar* to $\Lambda(s)$ and use the notation $N(s) \sim \Lambda(s)$. Now returning to the factorization

$$G(s) = \frac{1}{d(s)} N(s)$$

of a transfer function matrix, it follows that

$$G(s) \sim \frac{1}{d(s)} \Lambda(s) = \begin{bmatrix} \frac{\beta_1(s)}{\alpha_1(s)} & & & 0 \\ & \ddots & & \\ & & \frac{\beta_r(s)}{\alpha_r(s)} & \\ 0 & & & 0 \end{bmatrix} \tag{6.10}$$

where $\beta_i(s) \,|\, \beta_{i+1}(s)$ and $\alpha_{i+1}(s) \,|\, \alpha_i(s)$. This form of a transfer function matrix is called the Smith-McMillan form.

Definition 6.7

Consider a transfer function matrix $G(s)$ and its Smith-McMillan form as in (6.10). Introduce the polynomials

$$\beta(s) = \beta_1(s)\beta_2(s)\ldots\beta_r(s), \quad \alpha(s) = \alpha_1(s)\alpha_2(s)\ldots\alpha_r(s) \tag{6.11}$$

The roots of the polynomials $\beta(s)$ and $\alpha(s)$ are called the zeros and poles of $G(s)$, respectively. The degree of $\alpha(s)$ is called the McMillan degree *of $G(s)$.*

It is clear from the above definition that the poles of $G(s)$ include all poles of its individual entries. Moreover, the Smith-McMillan form determines the multiplicity of each pole. The

McMillan degree is equal to the number of poles. It can be shown that the poles of $G(s)$ in the sense of Definition 6.7 are the eigenvalues of the state matrix of a minimal realization. This also implies that the McMillan degree of $G(s)$ is the order of its minimal realization.

Returning to the system of Example 6.1, we find that (see Exercise 6.6)

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

Thus, the poles are -1, -1 and -2. These are exactly the eigenvalues of the Gilbert realization, and the order of the Gilbert realization is equal to the McMillan degree. The system also has a zero at $s = -3$. This zero is not obvious from the transfer function matrix $G(s)$, but we have

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

so -3 is a value of s where $G(s)$ loses its full rank. The significance of such multivariable zeros is illustrated in Exercises 6.11 and 6.12, where it is shown that a multivariable right-half-plane zero has an effect similar to a right-half-plane zero of a SISO system.

6.6 Multivariable Feedback Systems and Closed-Loop Stability

In this section we introduce a multivariable feedback structure and discuss its stability.

Well-Posedness of a Feedback Loop

Consider the feedback loop in Figure 6.5. For given proper transfer functions $G_1(s)$ and $G_2(s)$, we might first want to check whether the closed-loop transfer function is proper. That this may not always be the case is illustrated by the following example. Let

$$G_1(s) = 1, \quad G_2(s) = \frac{s-2}{s+3}$$

Even though both transfer functions are proper, we have

$$e_2 = \frac{s+3}{5}(u_1 + u_2)$$

Thus, the transfer function from u_1 to e_2 is not proper.

Definition 6.8 *A feedback loop is* well-posed *if all closed-loop transfer functions are proper.*
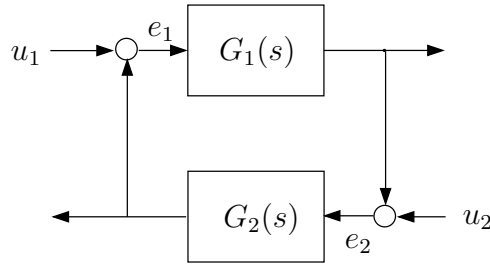
Figure 6.5: Feedback loop

One can show that if either $G_1(s)$ or $G_2(s)$ is strictly proper, the feedback loop in Figure 6.5 is well-posed. From now on we will assume that the feedback loops we consider are well-posed.

Internal Stability

Assume we want to check the stability of the closed-loop transfer function from u_1 to the output of G_1 in Figure 6.5. In the case of single-input single-output systems, the closed-loop transfer function from u_1 to the output of G_1 is

$$\frac{G_1(s)}{1 - G_1(s)G_2(s)}$$

To check stability we could investigate the zeros of the denominator $1 - G_1 G_2$ - this is in fact what we do when we apply the Nyquist stability test or root locus techniques. Again, a simple example illustrates that we need to be careful.

Example 6.2

Let

$$G_1(s) = \frac{1}{s-1}, \qquad G_2(s) = \frac{s-1}{s+2}$$

We can check that

$$1 - G_1(s)G_2(s) = \frac{s+1}{s+2}$$

which has no zeros in the right half plane. However, the closed-loop transfer function is

$$\frac{G_1(s)}{1 - G_1(s)G_2(s)} = \frac{s+2}{(s+1)(s-1)}$$

which is unstable. The problem here is that an unstable pole-zero cancellation prevents the unstable closed-loop pole from being detected. One might ask why this issue has not been raised when the Nyquist stability test for SISO systems was introduced. The answer is that when using classical design techniques for SISO systems, one can asssume that the designer has explicit control over pole and zero locations and will usually avoid unstable pole-zero cancellations. For MIMO systems, automated design techniques are often used to find a controller, and in some cases internal stability must be checked.

We now turn to MIMO systems and allow u_1, u_2, e_1 and e_2 to be vector signals. Again we want to check whether the feedback loop in Figure 6.5 is internally stable. Define the transfer functions $H_{ij}(s)$, $i = 1, 2$ by

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Definition 6.9

The feedback system in Figure 6.5 is called internally stable *if and only if the transfer function matrix*

$$\begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix}$$

is stable.

This definition makes sure that no unstable pole-zero cancellation prevents an unstable closed-loop pole from being detected. Note that internal stability requires that each of the four transfer matrices $H_{ij}(s)$ is stable.

One can show that the feedback loop is internally stable if and only if

(i)   there is no unstable pole-zero cancellation in $G_1(s)G_2(s)$, and

(ii)   the transfer function $(I - G_1(s)G_2(s))^{-1}$ is stable.

Now introduce

$$\phi(s) = \det(I - G_1(s)G_2(s))$$

and note that (ii) is equivalent to $\phi(s)$ having all its zeros in the open left half plane. With these results we can now formulate a MIMO version of the Nyquist stability criterion. Let n_1 be the number of unstable poles of $G_1(s)$ and n_2 the number of unstable poles of $G_2(s)$. Observing that $\phi(s)$ has all zeros in the open left half plane if and only if the Nyquist plot of $\phi(j\omega)$ encircles the origin $n_1 + n_2$ times counter-clockwise, we conclude the following.

Theorem 6.6   *The feedback loop in Figure 6.5 is internally stable if and only if*

(i)   there is no pole-zero cancellation in $G_1(s)G_2(s)$

(ii)   the Nyquist plot of $\phi(j\omega)$ encircles the origin $n_1 + n_2$ times counter-clockwise.

6.7   A Multivariable Controller Form

We will now extend the idea used in Section 2.1 for the construction of the controller form state space realization to multivariable systems. The resulting multivariable controller

form will be the basis for a discussion of multivariable pole placement in the following section.

In the SISO case, the starting point for the construction of a state space model from a transfer function was the factorization of the transfer function model shown in Fig. 2.3. For a $l \times m$ MIMO transfer function model $G(s)$, we now introduce the factorization

$$G(s) = N(s)D^{-1}(s) \tag{6.12}$$

where $N(s)$ is a $l \times m$ and $D(s)$ a $m \times m$ polynomial matrix. Such a representation is called a *(right) matrix fraction description* (MFD) of $G(s)$ (a *left matrix fraction description* is $G(s) = D^{-1}(s)N(s)$). An MFD of $G(s)$ is not unique, one choice is to start with the factorization $G(s) = N(s)/d(s)$ that was used for the Gilbert realization, and to define $D(s) = d(s)I$. Other MFDs can be generated by multiplying $G(s)$ from the right by a polynomial matrix $D(s)$ which is chosen such that all denominator polynomials in $G(s)$ are cancelled, leading to $G(s)D(s) = N(s)$ where $N(s)$ is a polynomial matrix.

We will now illustrate the construction of a multivariable controller form with an example.

Example 6.3

Consider a plant with transfer function model

$$G(s) = \begin{bmatrix} \frac{s}{(s+1)^2(s+2)^2} & \frac{s}{(s+2)^2} \\ \frac{-s}{(s+2)^2} & \frac{-s}{(s+2)^2} \end{bmatrix} \tag{6.13}$$

An MFD of this plant is

$$G(s) = \begin{bmatrix} s & 0 \\ -s & s^2 \end{bmatrix} \begin{bmatrix} 0 & -(s+1)^2(s+2) \\ (s+2)^2 & s+2 \end{bmatrix}^{-1} \tag{6.14}$$

To follow the development in Section 2.1, we introduce the representation shown in Fig. 6.6 (the MIMO equivalent of Fig. 2.3), where the m-dimensional signal vector $v(t)$ is introduced as the output of the multivariable filter $D^{-1}(s)$. In the SISO case, the idea was to express input and output signal in terms of $v(t)$, to assume that the highest derivative of this signal (determined by the highest power of s in $a(s)$) is somehow available and then to generate it by using a chain of integrators. To do something similar for a MIMO system, we first need the equivalent of the highest derivative of $v(t)$. Now $v(t)$ is a signal vector, and we should consider the highest powers of s in $D(s)$. We will do this by writing $D(s)$ as

$$D(s) = D_h S(s) + D_l \Psi(s) \tag{6.15}$$

where

$$S(s) = \begin{bmatrix} s^2 & 0 \\ 0 & s^3 \end{bmatrix}$$
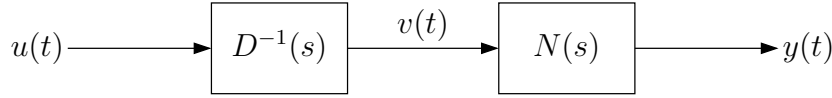
Figure 6.6: Matrix fraction description of plant model

is a diagonal matrix that has the highest power of s in each column of $D(s)$ as element in the corresponding position, $\Psi(s)$ contains the lower powers of s, and D_h and D_l are the coefficient matrices for the highest and the lower powers of s, respectively. For the MFD in (6.14) this decomposition is

$$D(s) = \begin{bmatrix} 0 & -(s+1)^2(s+2) \\ (s+2)^2 & s+2 \end{bmatrix} = \begin{bmatrix} 0 & -s^3-4s^2-5s-2 \\ s^2+4s+4 & s+2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} s^2 & 0 \\ 0 & s^3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -4 & -5 & -2 \\ 4 & 4 & 0 & 1 & 2 \end{bmatrix}\begin{bmatrix} s & 0 \\ 1 & 0 \\ 0 & s^2 \\ 0 & s \\ 0 & 1 \end{bmatrix} = D_h S(s) + D_l \Psi(s)$$

The next step is to generate $v(t)$ by repeatedly integrating the highest derivatives of each element of this signal vector. We have

$$U(s) = D(s)V(s) = D_h S(s)V(s) + D_l \Psi(s)V(s) \tag{6.16}$$

thus

$$S(s)V(s) = -D_h^{-1}D_l\Psi(s)V(s) + D_h^{-1}U(s) \tag{6.17}$$

Equation (6.16) is the MIMO equivalent of (2.12) in the SISO case, where the plant model was normalized such that the characteristic polynomial $a(s)$ is monic. The assumption that D_h is invertible corresponds in the scalar case to the assumption that $a(s)$ is indeed a polynomial of degree n (i.e. $a_n \neq 0$). Because

$$S(s)V(s) = \begin{bmatrix} s^2 & 0 \\ 0 & s^3 \end{bmatrix}\begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \ddot{v}_1(t) \\ \dddot{v}_2(t) \end{bmatrix}$$

and

$$\Psi(s)V(s) = \begin{bmatrix} s & 0 \\ 1 & 0 \\ 0 & s^2 \\ 0 & s \\ 0 & 1 \end{bmatrix}\begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \dot{v}_1(t) \\ v_1(t) \\ \ddot{v}_2(t) \\ \dot{v}_2(t) \\ v_2(t) \end{bmatrix}$$

the vector $S(s)V(s)$ represents the highest derivatives of each element of $v(t)$, whereas $\Psi(s)V(s)$ collects all lower derivatives. We can now proceed in the same way as in the SISO case: we assume temporarily that the highest derivative of each element of $v(t)$

is available, and integrate repeatedly to obtain all lower derivatives. Instead of a single integrator chain, we need however m integrator chains. The outputs of the integrators are the entries of $\Psi(s)V(s)$, which can be used for feedback through the gain matrix $D_h^{-1}D_l$ and added to $D_h^{-1}U(s)$ to generate the vector $S(s)V(s)$, according to (6.17). Like in (2.13) for SISO systems, the output vector $Y(s)$ can be obtained as a weighted sum of the elements of $\Psi(s)V(s)$: from Fig. 6.6 we have

$$Y(s) = N(s)V(s) = N_l\Psi(s)V(s)$$

where N_l is the coefficient matrix of $N(s)$. This leads to the model in Fig. 6.7, which is a multivariable version of Fig. 2.4. A controllable state space realization of this model is developed in Exercise 6.13.
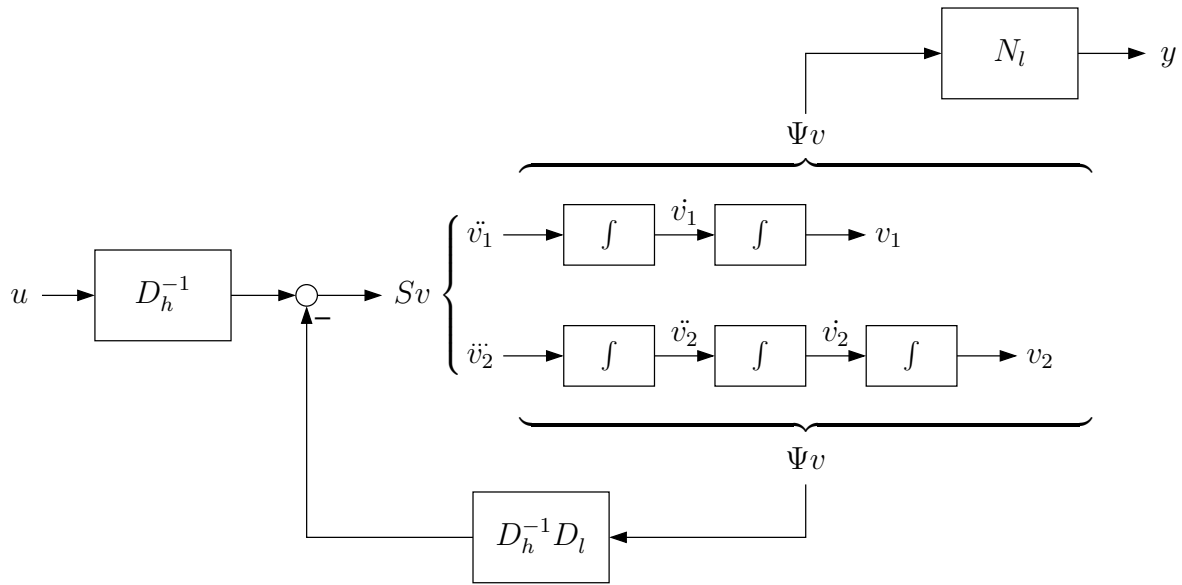
Figure 6.7: Time-domain model of $G(s)$

For reasons that will become clear when pole placement by state feedback around this realization is discussed in the next section, we will call this realization a *multivariable controller form*. In the scalar case, we called the equivalent realization a *canonical* controller form, reflecting the fact that for a SISO system, the procedure described in Section 2.1 leads to a *unique* realization. This is not the case for a MIMO system: the model in Fig. 6.7 depends on the choice of MFD (in the given example on (6.14)), which is not unique. Because different MFDs will lead to different controller forms of the same system, this multivariable controller form is not called a canonical realization.

6.8   Pole Placement

We will now develop a multivariable version of the pole placement technique that was used in Exercise 2.6. In the previous section we saw that a multivariable controller form can be constructed from an MFD

$$G(s) = N(s)D^{-1}(s)$$

where $D(s)$ is chosen such that its highest coefficient matrix D_h in (6.15) has full rank. Referring to the model in Fig. 6.7, it is clear that the state variables of this system are completely determined by the signal vector $\Psi(s)V(s)$. We can therefore represent state feedback through a gain matrix F as shown in Fig. 6.8. From Fig. 6.8, the closed-loop transfer function is

$$G_F(s) = \frac{Y(s)}{U_v(s)} = N(s)D_F^{-1}(s)$$

We have

$$U(s) = D(s)V(s) = U_v(s) + F\Psi(s)V(s)$$

thus

$$U_v(s) = (D(s) - F\Psi(s))V(s) = D_F(s)V(s)$$

which shows that

$$D_F(s) = D(s) - F\Psi(s) \tag{6.18}$$

and after substituting from (6.15)

$$D_F(s) = D_h S(s) + (D_l - F)\Psi(s) \tag{6.19}$$

$$D_F^{-1}(s)$$
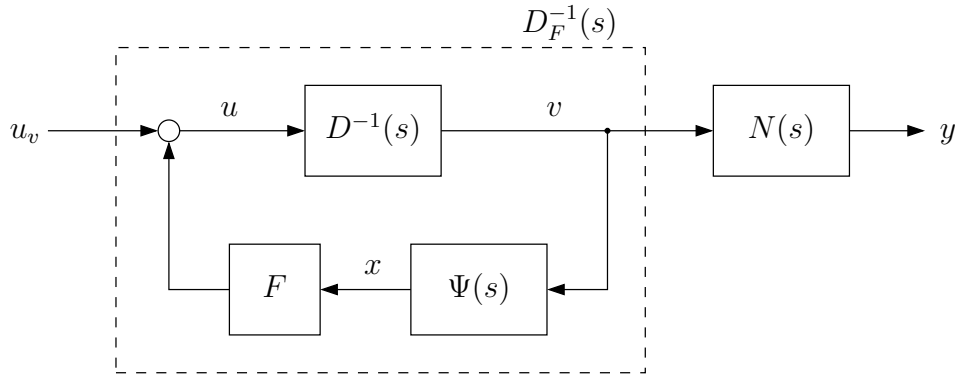
Figure 6.8: State feedback for multivariable controller form

This should be compared with the observation made in Exercise 2.6, that for a SISO model under state feedback around a controller form realization (A_c, b_c, c_c) we have

$$\det(sI - A_c - b_c f) = s^n + (a_{n-1} - f_n)s^{n-1} + (a_{n-2} - f_{n-1})s^{n-2} + \ldots + (a_0 - f_1)$$

where

$$\det(sI - A) = s^n + a_{n-1}s^{n-1} + \ldots + a_0$$

is the open-loop characteristic polynomial. All coefficients (except for the highest power of s) of the closed-loop polynomial can be assigned arbitrary values by the appropriate choice of the state feedback gains f_i, therefore the closed-loop eigenvalues can be arbitrarily chosen. From (6.19) we see that the same is true for MIMO systems: by choosing the gain matrix F we can arbitrarily change all coefficients in $D(s)$ except for the highest powers in each column. We would therefore expect that by choice of F we can obtain any desired closed-loop polynomial. This is indeed the case, and we will derive a procedure for computing a gain matrix F that achieves this. We also observe that - just as we found for SISO systems - state feedback does not alter the numerator polynomial matrix $N(s)$.

An Eigenvector Method for Pole Placement

We will now assume that the plant is represented by a state space model in the controller form (A_c, B_c, C_c) developed in Exercise 6.13 from Fig. 6.7. Thus assume that the plant dynamics are governed by the state equation

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

and state feedback $u(t) = Fx(t)$ is to be used to assign a set of distinct closed-loop eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. These closed-loop eigenvalues are the solutions of

$$\det D_F(s) = 0$$

We will also need the *closed-loop eigenvectors*, i.e. the vectors h_i, $i = 1, \ldots, n$ that satisfy

$$(A_c + B_c F)h_i = \lambda_i h_i$$

It can be shown that the eigenvectors h_i can be expressed as

$$h_i = \Psi(\lambda_i)p_i$$

where p_i is any vector in the nullspace of $D_F(\lambda_i)$, i.e. a vector such that

$$D_F(\lambda_i)p_i = 0$$

(this is illustrated in Exercise 6.14.) Now from (6.18) it follows that

$$D(\lambda_i)p_i - F\Psi(\lambda_i)p_i = 0$$

Defining the vectors $g_i = D(\lambda_i)p_i$, $i = 1, \ldots, n$, we thus have

$$Fh_i = g_i, \quad i = 1, \ldots, n$$

or

$$F[h_1 \quad \ldots \quad h_n] = [g_1 \quad \ldots \quad g_n]$$

Because the eigenvalues are assumed distinct, the eigenvectors are linearly independent and we can solve for the state feedback gain

$$F = [g_1 \quad \ldots \quad g_n][h_1 \quad \ldots \quad h_n]^{-1} \tag{6.20}$$

If the desired eigenvalues are not distinct, one can use generalized eigenvectors to obtain a unique solution F to the pole placement problem.

The discussion leading to (6.20) shows that not only the closed-loop eigenvalues, but - within limits - also the closed-loop eigenvectors can be assigned by state feedback. The procedure can be summarized as follows.

1.   Choose distinct closed-loop eigenvalues $\lambda_1, \ldots, \lambda_n$.

2.   Choose closed-loop eigenvectors $h_1, \ldots, h_n$ such that

$$h_i \in \mathcal{R}(\Psi(\lambda_i)), \quad i = 1, \ldots, n \tag{6.21}$$

   and such that eigenvectors associated with a complex conjugate eigenvalue pair are also a complex conjugate pair. With this choice, there exist vectors p_i such that $h_i = \Psi(\lambda_i)p_i, \ i = 1, \ldots, n$.

3.   Define $g_i = D(\lambda_i)p_i, \ i = 1, \ldots, n$.

4.   Compute the state feedback gain matrix F from (6.20). This gain matrix satisfies

$$(A_c + B_c F)h_i = \lambda_i h_i, \quad i = 1, \ldots, n$$

Step 2 of this procedure indicates that a given set of closed-loop eigenvalues does not uniquely determine the state feedback gain, but that we can choose the closed-loop eigenvectors as well - subject to the constraint (6.21). Recall that for SISO systems equation (3.9) shows that the solution to the pole placement problem is unique. For MIMO systems, the additional degree of freedom can be exploited to find a solution that improves certain performance measures. One way of choosing closed-loop eigenvalues and eigenvectors is briefly outlined next.

6.9   Optimal Control of MIMO Systems

We will now extend the symmetric root locus design introduced in Section 5.4 to MIMO systems. Consider a system with state space representation

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0$$
$$y(t) = Cx(t)$$

where (A, B) is stabilizable and (C, A) is detectable. As in the SISO case, assume that $x_0 \neq 0$ and that we wish to find a control input $u(t)$ that brings the system quickly back to $x = 0$. A MIMO version of the cost function (5.12) is

$$V = \int_0^\infty (y^T(t)y(t) + u^T(t)Ru(t))dt, \quad R > 0$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite matrix that places a cost on the control effort. A more general form of a cost function is

$$V = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t))dt, \quad R > 0, \ Q \geq 0 \tag{6.22}$$

where $Q \in \mathbb{R}^{n \times n}$ is a positive semidefinite weighting matrix on the control error. A common choice of weighting matrices is

$$Q = C^T C, \quad R = \begin{bmatrix} \rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_m \end{bmatrix}$$

where Q is fixed and the ρ_i can be used as tuning parameters as discussed below.

Next, define the matrix

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \tag{6.23}$$

This $2n \times 2n$ matrix plays an important role in optimal control. By applying the similarity transformation

$$T = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

one can show that H is similar to $-H^T$, which implies that if λ is an eigenvalue of H then so is $-\lambda$. A matrix with this property is called a *Hamiltonian matrix*, its eigenvalues are symmetric about the imaginary axis. For a SISO system and the choices $Q = c^T c$ and $R = \rho$, one can verify that the characteristic polynomial of H is exactly the polynomial $p(s)$ of the symmetric root locus design, that was introduced in Section 5.4 and provides the optimal closed-loop eigenvalues. For MIMO systems, it turns out that the Hamiltonian matrix (6.23) provides not only the optimal eigenvalues, but also the optimal eigenvectors. Partition the eigenvectors of H as

$$H \begin{bmatrix} h_i \\ g_i \end{bmatrix} = \lambda_i \begin{bmatrix} h_i \\ g_i \end{bmatrix}, \quad i = 1, \ldots, 2n \tag{6.24}$$

where h_i contains the top n entries of the $2n$-dimensional eigenvectors. Assume that H has no eigenvalues on the imaginary axis (as in the SISO case, it can be shown that eigenvalues on the imaginary axis represent unobservable or uncontrollable modes, which are excluded by the assumption of stabilizability and detectability). Then the optimal control law that minimizes the cost V in (6.22) takes the form of state feedback

$$u_{\text{opt}}(t) = Fx(t)$$

and we have the following result, which is given here without proof.

Theorem 6.7

The optimal state feedback control law $u_{\text{opt}}(t) = Fx(t)$ that minimizes the cost function (6.22) places the eigenvalues of $(A + BF)$ at the stable eigenvalues of H. Moreover, the eigenvectors of $(A + BF)$ are the partitions h_i of the eigenvectors of H that are associated with the stable eigenvalues.

Theorem 6.7 can be combined with the formula (6.20) to solve an optimal pole placement problem: compute the stable eigenvalues and eigenvectors of H, and use (6.20) to compute the optimal state feedback gain F. With the choice of weighting functions $Q = C^T C$ and $R = \text{diag}\,(\rho_1, \ldots, \rho_m)$, the tuning parameters ρ_i can be used to trade control performance against control effort in each input channel independently. This is illustrated in Exercise 6.10 for the turbogenerator design problem.

Optimal State Estimation

For a multivariable version of the method for optimal state estimation in Section 5.4, we extend the deterministic state space model to

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$$
$$y(t) = Cx(t) + v(t)$$

where $w(t)$ and $v(t)$ are white noise processes: the n-dimensional vector $w(t)$ represents process noise, and the l-dimensional vector $v(t)$ measurement noise. As before, both noise processes are assumed to be wide-sense stationary, zero-mean, Gaussian distributed and uncorrelated, and to satisfy

$$E[w(t)w^T(t + \tau)] = Q_e\delta(\tau), \quad E[v(t)v^T(t + \tau)] = R_e\delta(\tau), \quad E[w(t)v^T(t + \tau)] = 0$$

One can show that the observer gain L that solves

$$\min_L \lim_{t \to \infty} E[\tilde{x}^T(t)\tilde{x}(t)]$$

i.e. that minimizes the noise power in the state estimation error, is obtained by solving the dual version of the optimal pole placement problem, with the replacements

$$A, \ B \quad \to \quad A^T, \ C^T$$

and

$$Q, \ R \quad \to \quad Q_e, \ R_e$$

In practice, the noise covariance matrices are often not known explicitly, but are used as tuning parameters. Common choices are $Q_e = BB^T$ and $R_e = \text{diag}(r_1, \ldots, r_l)$, where the values of r_i can be used to tune the speed of estimation for each output channel. This is illustrated in Exercise 6.15 for the turbogenerator design problem.

Exercises

Problem 6.1

Consider the plant

$$G(s) = \frac{1}{(s+1)(s+5)(s+0.1)} \begin{bmatrix} 1 & 0.2 \\ -0.1 & 2 \end{bmatrix}$$

a) How many characteristic loci does this plant have?

b) Write a Matlab script to determine the eigenvalues of $G(s)$ as s takes values along the positive and negative imaginary axis, and to draw the characteristic loci.

c) Use the plot to determine the maximum value of the constant k such that with the controller kI in negative feedback the system remains stable.

Hint: The commands `frd` and `frdata` are useful Matlab commands for working with multivariable frequency responses.

Problem 6.2

Consider the multivariable closed-loop system in Figure 6.9.

a) Show that the transfer function from r to e is

$$S = (I + GK)^{-1}$$

This function is called the *sensitivity function.* Show that S is also the transfer function from d_o to y.

b) Show that the transfer function from r to y is

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

This function is called the *complementary sensitivity function.*

c) What is the transfer function from d_i to u_g? (This function is sometimes called the *input sensitivity function S_I.*) Show that $GS_I = SG$. Note that $S_I = S$ for SISO systems.
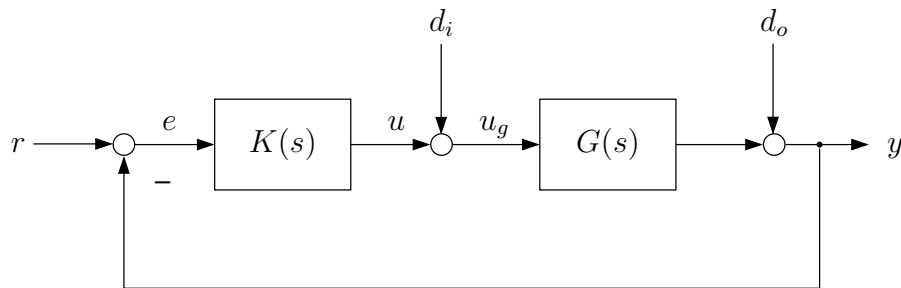
Figure 6.9: Multi-variable closed loop system

Problem 6.3

Consider the state space model of an experimental turbogenerator with the following inputs and outputs:

Input 1 (u_1): Gas control valve of the turbine

Input 2 (u_2): Excitation current of the generator

Output 1 (y_1): Turbine speed

Output 2 (y_2): Generator terminal voltage

Figure 6.10 indicates the inputs (u_1, u_2), the outputs (y_1, y_2) and the disturbances (d_1, d_2) associated with the plant, $G(s)$.
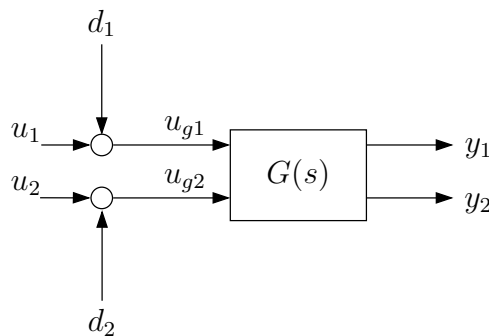
Figure 6.10: Turbogenerator inputs, outputs and disturbances

An observer-based state feedback controller is to be designed, for this system, such that the following operating requirements are satisfied.

i)   There should be zero steady state error for:

   −   step changes in set points.

     — step disturbances.

     — small variations in model parameters (A, B, C, D).

ii)   There should be no state estimation errors following a change in setpoint.

Matrices A, B, C and D of the plant model can be obtained from Matlab file `cs5_tgen.m`. *Same model will be used in problems 6.10 and 6.15 also*

a)   Draw the block diagram of the closed-loop system. Explain the structure of the closed-loop system. Show the dimensions of the matrices and the signals in the system.

b)   Create the closed loop system in Simulink. Tune the pole locations of the controller using pole placement to achieve the following setpoint tracking specifications.

    i)   The 5% settling time for output y_1 should be less than 3.0 and the maximum change in y_2 should be less than 0.15 for step changes to setpoint r_1.

    ii)   The 5% settling time for output y_1 should be less than 2.0 and the maximum change in y_1 should be less than 0.15 for step changes to setpoint r_2.

   Hints:

     — *Consider the effect of the observer poles on the responses.*

     — *To restrict the search for appropriate pole positions, use damping ratios greater than 0.5 for pairs of poles.*

     — *Matlab function* `place()` *can be used for MIMO systems as well as for SISO systems.*

c)   Modify the observer pole locations with pole placement so that the maximum changes in y_1 and y_2 are both less than 1.0 in response to a unit step disturbance $d_1(t) = \sigma(t)$.

Problem 6.4

The turbogenerator from Problem 6.3 is controlled by the controller

$$K(s) = \begin{pmatrix} k_{11}\left(1 + \frac{1}{\tau_I s}\right) & 0 \\ 0 & k_{22} \end{pmatrix}$$

where $k_{11} = 4.5$, $k_{22}=3.2$ and $\tau_I = 0.1$ in the configuration shown in Figure 6.9.

a)   Use Matlab to show that the resulting closed loop system is stable.

b)   Use the Matlab command `sigma` to plot the singular values of the sensitivity function S and the complementary sensitivity function T. Explain the low and high frequency behaviour in terms of the controller.

c)   Calculate the input and output direction corresponding to each singular value of S at zero frequency. Explain the significance of these directions in terms of the controller.

Problem 6.5

The plant

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

is subject to an input disturbance d, and a controller is to be designed for tracking a reference r, see Figure 6.11.
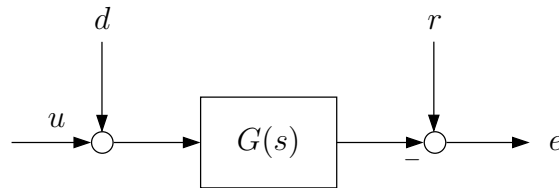
Figure 6.11: System to be controlled

The maximum allowable magnitude of error and input in each channel are given by the elements of the vectors

$$e_{max} = \begin{bmatrix} 0.2 \\ 0.5 \end{bmatrix}, \quad u_{max} = \begin{bmatrix} 1.0 \\ 2.0 \end{bmatrix}$$

respectively, i.e.

$$|e_1(t)| < 0.2, \quad |e_2(t)| < 0.5, \quad |u_1(t)| < 1.0, \quad |u_2(t)| < 2.0 \ \forall t > 0$$

The input disturbance vector d_i has its maximum allowable magnitudes at d_{max} and the setpoint vector r has its maximum at r_{max}

$$d_{max} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}, \quad r_{max} = \begin{bmatrix} 4 \\ 0.4 \end{bmatrix}$$

The plant model is to be scaled so that with scaled variables $\bar{u}$, $\bar{y}$ and $\bar{d}$ the plant dynamics can be described by

$$\bar{y} = \bar{G}\bar{u} + \bar{G}_d\bar{d}$$

and for a scaled setpoint $\tilde{r}$ the error dynamics can be described by

$$\bar{e} = R\tilde{r} - \bar{y}$$

where R is a diagonal scaling matrix.

a)   Calculate a diagonal scaling matrix for the error e so that the scaled error vector $\bar{e}$ at its maximum is

$$\bar{e}_{max} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

b)   Use diagonal scalings of the true inputs u and outputs y to determine $\bar{G}$ so that the scaled output and input vectors $\bar{y}$ and $\bar{u}$ at their maximum magnitudes are

$$\bar{y}_{max} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \qquad \bar{u}_{max} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

Interpret these scalings in terms of the importance assigned to the scaled input and output variables.

Hint: y should have the same scaling as e in part (a).

c)   Determine a diagonal scaling of the true disturbance d_i to determine $\bar{G}_d$ so that the scaled disturbance $\bar{d}$ at its maximum magnitude is

$$\bar{d}_{max} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

d)   Calculate a diagonal scaling matrix R so that the scaled setpoint vector $\tilde{r}$ at its maximum magnitude is

$$\tilde{r}_{max} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

Hint: Note that $R\tilde{r}$ should be in the same scaled units as $\bar{e}$ and $\bar{y}$.

e)   Consider the scaled plant in closed loop with scaled setpoint changes $\tilde{r}$. What constraint on the corresponding sensitivity function S ensures that the error does not violate the constraints during the expected variation of r?

Problem 6.6

Consider a system with transfer function

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

a)   Write the transfer function as

$$G(s) = \frac{1}{d(s)}\tilde{G}(s)$$

where $d(s)$ is the lowest common multiple of the denominator.

b)   Use row and column exchange to bring the non-zero elements of $\tilde{G}(s)$ with lowest order into the position $(1,1)$.

c)   Use elementary row and column operations to bring the positions $(1,2)$ and $(2,1)$ to zero.

d)   Divide the elements of the matrix obtained in (c) by $d(s)$. (The resulting matrix is the Smith-McMillan form of the system.) What are the system poles and zeros?

Problem 6.7

Consider a system described by the state space model

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & -3 & 0 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Use the Matlab command *minreal* to compute the Kalman canonical form of this system. Identify the uncontrollable and unobservable states from this form.

Hint: Read the online help of the command `minreal`.

Problem 6.8

Construct by hand the Gilbert realization of the system with transfer function

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

Problem 6.9

The controllability matrix $\mathcal{C}$ of the Gilbert realization of a system with m inputs and outputs, n state variables and a denominator with r distinct roots can be factorised as

$$\mathcal{C} = \mathcal{B}\mathcal{V} \quad \text{where} \quad \mathcal{V} = \begin{bmatrix} I_m & \lambda_1 I_m & \cdots & \lambda_1^{n-1} I_m \\ \vdots & \vdots & \ddots & \vdots \\ I_m & \lambda_r I_m & \cdots & \lambda_r^{n-1} I_m \end{bmatrix}$$

For the special case $n = 2$, $m = 2$ and $r = 2$ (that is, no root is repeated) show this and find the matrix $\mathcal{B}$. Hint: Show first that the state space model B matrix can be written as

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} I \\ I \end{bmatrix}$$

Use the fact that $\mathcal{V}$ has full rank to show that the Gilbert realization of a system is controllable and, by duality, also observable.

Problem 6.10

For the turbogenerator plant from Problem 6.3 and with the controller structure in Figure 6.12, the cost function

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$

is to be minimized, where

$$\begin{aligned} u_1, \; u_2 \quad & \text{are the plant inputs} \\ x_1 \dots x_4 \quad & \text{are the plant state variables} \\ x_5, \; x_6 \quad & \text{are the state variables of the integrator block} \\ y_1, \; y_2 \quad & \text{are the plant outputs} \\ r_1, \; r_2 \quad & \text{are the setpoints for } y_1 \text{ and } y_2 \end{aligned}$$

a)   Determine Q and R such that

$$J = \int_0^\infty (k_1(y_1^2 + y_2^2) + k_2(x_5^2 + x_6^2) + k_3(u_1^2 + u_2^2)) dt$$

b)   Explain qualitatively what change to performance is expected when

   i) k_1 is increased while k_2 and k_3 are held constant

   ii) k_2 is increased while k_3 and k_1 are held constant

   iii) k_3 is increased while k_1 and k_2 are held constant
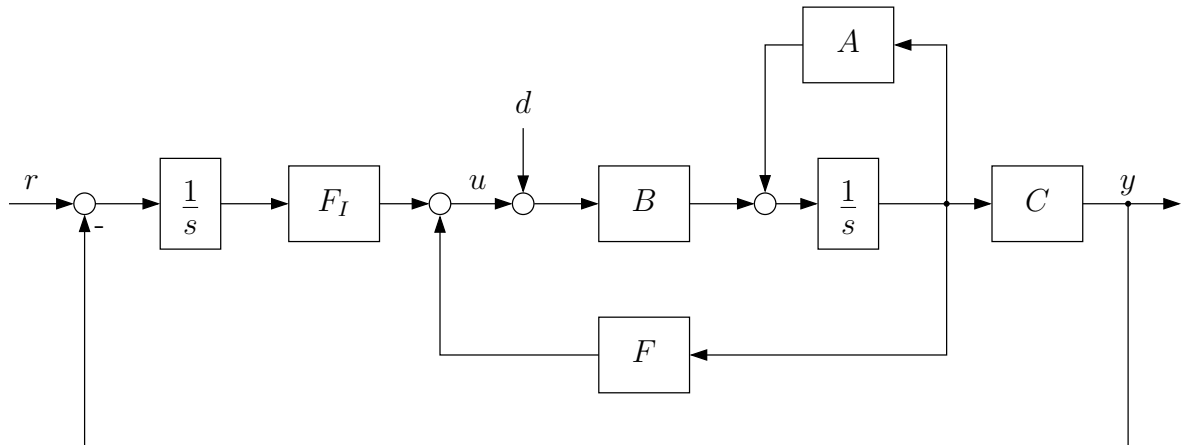
Figure 6.12: State feedback control with integral action

Why can k_1 be set to 1 without loss of generality?

c) Use the Matlab command `lqr` and the Simulink model of the turbogenerator from Problem 6.3 to find values of k_2 and k_3 with $k_1 = 1$ such that in closed loop the following conditions are satisfied:

   i) Following a step change in setpoint $r_1(t) = \sigma(t)$ or $r_2(t) = \sigma(t)$,

      *   5% settling times for y_1 or y_2 remain less than 1.2s.
      *   The overshoots in y_1 and y_2 are less than 5%.
      *   Cross coupling remains less than 1%.

   ii) The overshoots on inputs (u_1, u_2) following setpoint change $r_1(t) = \sigma(t)$ or $r_2(t) = \sigma(t)$ are less than 50%.

   iii) Following a disturbance $d_1(t) = \sigma(t)$ or $d_2(t) = \sigma(t)$ the maximum change to y_1 or y_2 is less than 0.05.

   Hints: To make a significant change to performance k_2 and k_3 have to be changed by a factor of greater than 5; consider k_2 in the range 10^{-1} to 10^3 and k_3 in the range 10^{-6} to 10^2.

d) Compare the LQR approach used here to the pole placement approach used in Problem 6.3 in terms of achieved performance and time used in tuning the controller.

⋆ **Problem 6.11**

Consider the system

$$G(s) = \frac{1}{(0.5s+1)(s+1)} \begin{bmatrix} 2 & 1 \\ 1+4s & 2 \end{bmatrix}$$

with the controller $K(s)$ in the configuration shown in Figure 6.13.
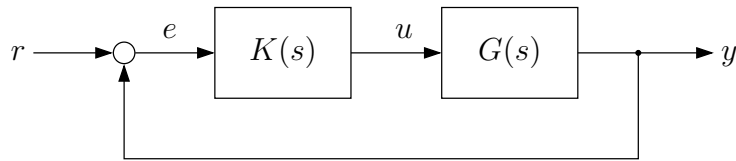
Figure 6.13: Closed-loop system

a) Use Matlab to determine the poles p_i and zeros z_i of the plant $G(s)$.

b) Show that for each zero z_i of $G(s)$ there is a vector $y_{zi} \neq 0$ such that

$$y_{zi}^T G(z_i) = 0$$

Calculate the vector y_{zi} for each of the zeros z_i of $G(s)$. (The vectors y_{zi} are called the output directions of the zeros.)

c) Use the result from (b) to explain why internal stability requires that for all *right half plane* zeros z_i of $G(s)$,

$$y_{zi}^T G(z_i) K(z_i) = 0$$

d) Show that $(1 + K(s)G(s))^{-1}$ must not have any poles in the right half plane for the closed loop to be internally stable. Use this fact and the result from (b) to show that for all zeros z_i in the right half plane we must have

$$y_{zi}^T G(z_i) K(z_i)(1 + G(z_i)K(z_i))^{-1} = 0$$

that is

$$y_{zi}^T T(z_i) = 0$$

where

$$T(s) = G(s)K(s)(1 + G(s)K(s))^{-1}$$

is the transfer function from r to y.

e) Consider the elements of the transfer matrix $T(s)$

$$T(s) = \begin{bmatrix} T_{11}(s) & T_{12}(s) \\ T_{21}(s) & T_{22}(s) \end{bmatrix}$$

(Recall from Problem 6.2 that $T(s)$ is the transfer function from r to y.) Show that internal stability imposes the following constraints on the relationship between the elements of T at $s = z_i$ for any right half plane zero z_i

$$\alpha_1 T_{11}(z_i) + \alpha_2 T_{21}(z_i) = 0$$
$$\alpha_1 T_{12}(z_i) + \alpha_2 T_{22}(z_i) = 0$$

where α_1 and α_2 are constants depending on z_i. What are their values for $G(s)$?

⋆ Problem 6.12

Here the constraints from Problem 6.11.e imposed by internal stability in the presence of right half plane zeros and their effect on achievable closed-loop performance are further investigated.

a) What constraints on the elements of $T(s)$ for any 2 input 2 output system are imposed by internal stability if in steady state the outputs must be equal to the setpoints, that is as $t \to \infty$, $r \to y$?

   Hint: Consider the conditions on $T(s)$ as $t \to \infty$, that is as $s \to 0$

b) What constraints are imposed on the elements of $T(s)$ for any 2 input 2 output plant if it is desired that the two input-output channels of the system are completely decoupled in closed loop, i.e. there is no movement in y_2 following a change in r_1 and no movement in y_1 following a change in r_2?

   Hint: This is a constraint only on the off-diagonal elements.

c) From (b), the constraints

$$\alpha_1 T_{11}(z_i) + \alpha_2 T_{21}(z_i) = 0$$
$$\alpha_1 T_{12}(z_i) + \alpha_2 T_{22}(z_i) = 0$$

where α_1 and α_2 are constants, must hold on the elements of $T(s)$ if $G(s)$ is a 2×2 transfer function with a right half plane zero at z_i.

Show that the complementary sensitivity function

$$T_1(s) = \begin{bmatrix} \frac{1}{1+0.01s} & 0 \\ 0 & \frac{1}{1+0.01s} \end{bmatrix}$$

violates this constraint and the design constraints from parts (a) and (b), whereas

$$T_2(s) = \begin{bmatrix} \frac{z_i - s}{z_i + s} & 0 \\ 0 & \frac{z_i - s}{z_i + s} \end{bmatrix}$$

does not violate these constraints. Sketch the step responses. While fulfilling the above constraints, is it possible to find a $T(s)$ with a better step response? Explain how these constraints lead to a fundamental limitation imposed by right half plane zeros.

⋆ Problem 6.13

Consider the system

$$G(s) = \begin{bmatrix} \frac{s}{(s+1)^2(s+2)^2} & \frac{s}{(s+2)^2} \\ \frac{-s}{(s+2)^2} & \frac{-s}{(s+2)^2} \end{bmatrix}$$
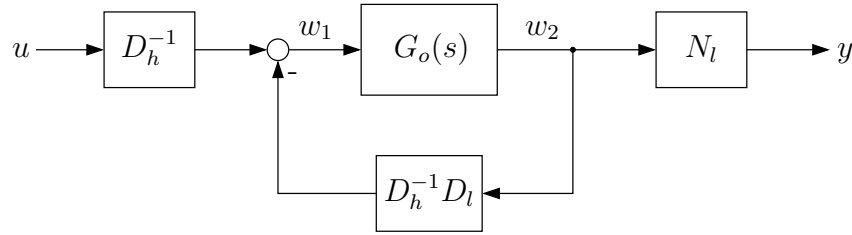
Figure 6.14: Multivariable controller form

$G(s)$ can be written as

$$G(s) = N(s)D^{-1}(s), \qquad N(s) = \begin{bmatrix} s & 0 \\ -s & s^2 \end{bmatrix}, \qquad D(s) = \begin{bmatrix} 0 & -(s+1)^2(s+2) \\ (s+2)^2 & s+2 \end{bmatrix}$$

where

$$D(s) = D_h S(s) + D_l \psi(s), \quad S(s) = \begin{bmatrix} s^2 & 0 \\ 0 & s^3 \end{bmatrix}, \quad \psi(s) = \begin{bmatrix} s & 0 \\ 1 & 0 \\ 0 & s^2 \\ 0 & s \\ 0 & 1 \end{bmatrix}$$

The signal vector v is defined as

$$u = D(s)v$$

The system $G(s)$ can then be represented as in Figure 6.14, where $w_1 = S(s)v$ and $w_2 = \psi(s)v$.

a) What is the transfer matrix $G_0(s)$ in Figure 6.14 as a function of the matrices introduced here?

b) Determine $G_0(s)$ for the given system $G(s)$

c) What is the matrix N_l in Figure 6.14 for the given System $G(s)$?

   Hint: What is y as a function of v and of $\psi(s)v$?

d) Give a state space realization (A_0, B_0, C_0, D_0) of $G_0(s)$ with

$$\dot{x}_1 = u_1, \quad \dot{x}_4 = u_2, \quad C = I$$

e) Determine a state space realization (A, B, C, D) of the complete system $G(s)$ as a matrix function of $G_0(s)$.

   Hint: The whole system can be seen as the system (A_0, B_0, C_0, D_0) under state feedback.

f) Calculate a state space realization A, B, C, D of the given system $G(s)$

⋆ Problem 6.14

For the following, use the state space realization of $G(s)$ from Problem 6.13.

a) Use Matlab to calculate a state feedback gain matrix F that places the closed-loop poles at
$$-2, \quad -3 + 2j, \quad -3 - 2j, \quad -5 + 3j, \quad -5 - 3j$$

b) Calculate the eigenvectors h_i $(i = 1 \ldots 5)$ for each closed-loop eigenvalue λ_i

c) With the Matlab command '\' find vectors p_i that satisfy

$$h_i = \psi(\lambda_i)p_i$$

   where
$$D_F(s) = D(s) - F\psi(s)$$

d) Check that
$$D_F(\lambda_i)p_i = 0$$

   holds for all closed-loop eigenvalues.

Problem 6.15

Download the files `cs5_tgen_LQG_FIXME.m`, `cs5_tgen_LQG_plot.m` and `cs5_tgensfmod2.mdl`.

`cs5_tgen_LQG_FIXME.m` contains a program designing state-feedback gain matrices F and F_I for the turbogenerator model as discussed in Problem 6.10 and a Kalman filter for the closed-loop configuration shown in Figure 6.15. This Figure includes input disturbance and measurement noise

$$d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \qquad \text{and} \qquad n = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$
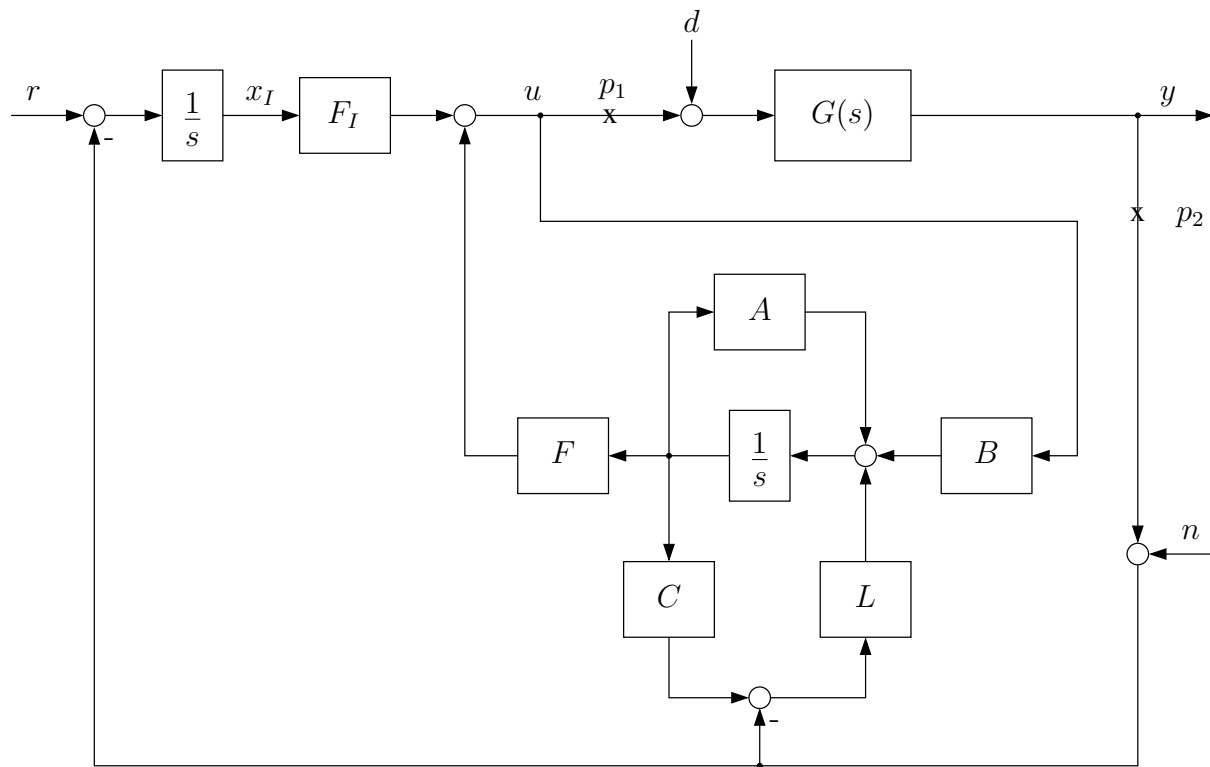
Figure 6.15: LQG control with integral action

a)   With covariance matrices

$$Q_e = BB^T, \qquad R_e = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$

how would you expect the observer poles and Kalman filter gains to change as R_e is varied? How would you expect responses to setpoint changes, disturbances at input d and sensitivity to output noise to vary as R_e is varied? Explain your answers.

b)   Tune the filter by changing the gains r_1 and r_2, so that the following conditions are fulfilled

i)   Conditions (i) to (iii) from Problem 6.10 part (c) are satisfied.

ii) Following a step disturbance $d_1(t) = \sigma(t)$ or $d_2(t) = \sigma(t)$ the maximum changes in y_1 and y_2 should both be less than 0.6.

iii) The spectral density of the noise in u_1 and u_2 is less than 0.05 when measurement noise is present at each of the outputs of the plant (parameters as defined for the Simulink block **band-limited white noise**) with noise power $= 0.0001$ and sample time $= 0.01$

Hint: You should modify the Simulink model of the controller to include the Kalman filter. The closed-loop performance is not sensitive to small changes in R_e - consider values of r_1 between 10^{-2} and 10^4 and variations in r_1/r_2 of between 10^{-4} and 10^4

c) Use Matlab to plot the singular value plots of the closed-loop system from

$$\text{(i) } r \text{ to } y, \qquad \text{(ii) } d \text{ to } y, \qquad \text{(iii) } n \text{ to } u$$

and for the open-loop system when the loop is broken at points p_1 and p_2.

Describe the relationship between the open-loop and closed-loop frequency responses, and between the closed-loop frequency response and the time responses.

Problem 6.16

A system with transfer function $G(s)$ and state space realization

$$A = \begin{bmatrix} 0 & 10 \\ -10 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 10 \\ -10 & 1 \end{bmatrix}, \qquad D = 0$$

has a constant feedback gain matrix
$$K = I$$
connected in negative feedback between input and output.

a) What is the transfer function matrix $G(s)$?

b) Determine the transfer function from w_1 to z_1 in Figure 6.16, and from w_2 to z_2 in Figure 6.17

Hint: Determine $K(s)G(s)$, then put a feedback connection into the loop.

c) What are the gain and phase margins for the closed-loop system from w_1 to z_1, and what are the margins from w_2 to z_2?
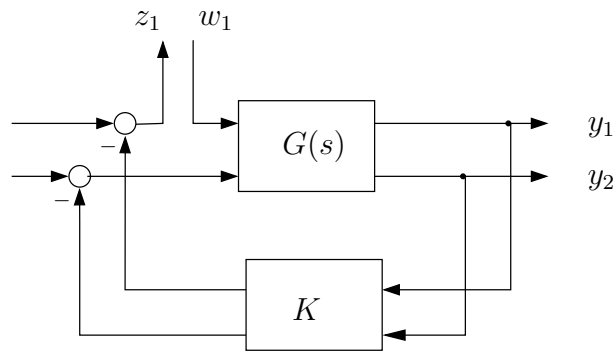
Figure 6.16: Open loop in channel 1
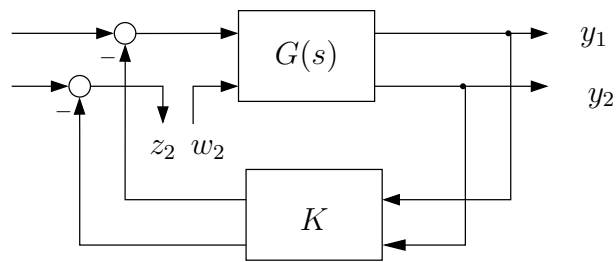
Figure 6.17: Open loop in channel 2

d) The plant inputs are now perturbed by a constant matrix $\tilde{G}$ such that

$$\tilde{u} = \tilde{G}u, \ y = G(s)\tilde{u}$$

where

$$\tilde{u}_1 = (1 + \epsilon_1)u_1, \ \tilde{u}_2 = (1 + \epsilon_2)u_2$$

and $\epsilon_1, \ \epsilon_2$ are constants.

Determine an open-loop state space model $\tilde{A}, \tilde{B}, \tilde{C}$ with these errors, and the corresponding closed-loop model $\tilde{A}_{cl}$ and $\tilde{B}_{cl}$.

e) Write down the characteristic polynomial for $\tilde{A}(\epsilon)_{cl}$. With $\epsilon_2 = 0$ how large can ϵ_1 become before the system becomes unstable. With $\epsilon_1 = 0$, how large can ϵ_2 become before the system becomes unstable?

f) With $\epsilon_2 = -\epsilon_1 = -\epsilon$ how large can ϵ become, before the system becomes unstable?

g) What does this suggest about the value of SISO gain margins in determining the robustness of MIMO systems?

Chapter 7

Digital Control

All controllers discussed so far were modelled by differential equations, transfer function (matrices) or by linear state space models involving a first order vector differential equation. These representations have in common that the input and output signals of such controllers are defined on the whole time axis - we will call such signals *continuous-time signals*. These assumptions imply that controllers are implemented using analog electronic devices. Today, most controllers are implemented digitally - the control law is realized as code on a microprocessor. Microprocessors cannot handle continuous-time signals directly, they can only process sequences of numbers. A signal consisting of a sequence of numbers is called a *discrete-time signal*. In this chapter we will develop tools for analyzing and designing digital controllers, and we will see that many results on continuous-time controller design can be translated into a discrete-time framework.
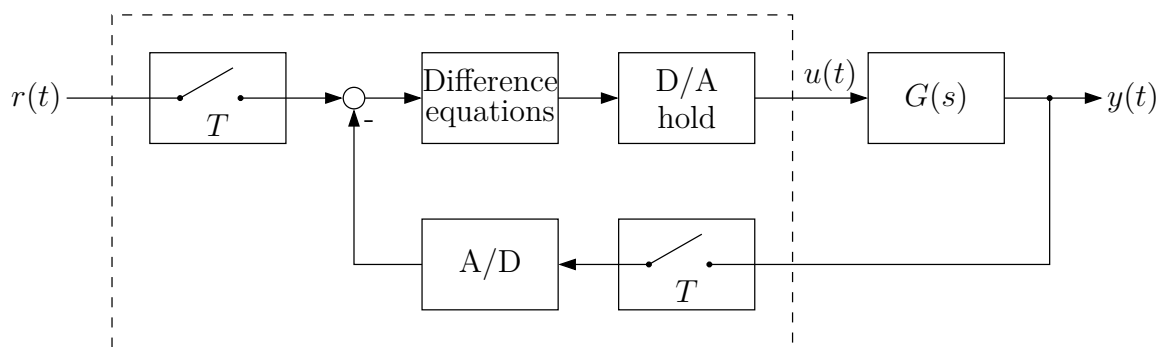
Figure 7.1: Control loop with digital controller

The block diagram in Fig. 7.1 shows the structure of a control loop with a digital controller. The plant - represented by the transfer function $G(s)$ - has continuous-time input and output signals. An analog-to-digital (A/D) converter takes samples of the measured feedback signal $y(t)$ repeatedly every T seconds, where T is the *sampling period*, and converts the sampled values into binary numbers. These are subtracted from samples

of the reference input (here assumed also binary) to generate a sequence of sampled control errors. A control algorithm in the form of difference equations takes the sampled control error as input and generates a sequence of values, which is then converted into a continuous-time control signal $u(t)$ by a digital-to-analog (D/A) converter and a hold element. We assume that the operation of the A/D and D/A converters and of the sample and hold devices are synchronized. The components inside the dashed box in Fig. 7.1 together form a digital controller.

The operation of a digital controller involves two types of discretization: a discretization of time, and a discretization of signal values due to the conversion into binary numbers with finite word length. In this chapter we will study the consequences of the discretization of time. We will assume that the resolution of the AD converter is sufficiently high so that quantization effects can be neglected. The effect of finite word length at fast sampling rates is however explored in Exercise 7.12.
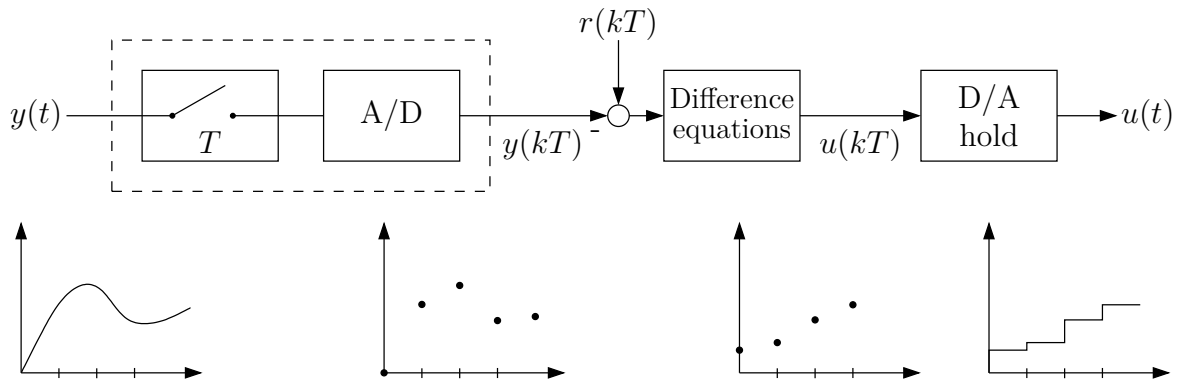
Figure 7.2: Continuous-time and discrete-time signals in the loop

The conversion between continuous-time and discrete-time signals inside the digital controller is shown again in Fig. 7.2. The signals entering and leaving the controller are continuous-time signals, they represent plant input $u(t)$ and output $y(t)$. The control algorithm itself operates on discrete-time signals, i.e. sequences of numbers $u(kT)$ and $y(kT)$, which are the values of the continuous-time plant input and output sampled at time instants $t = kT$, where $k = 0, 1, \ldots$. The block labelled "D/A hold" converts the sequence of control inputs $u(kT)$ into a continuous-time signal $u(t)$ that can be applied to the plant input. Usually a *zero-order hold* is used for this purpose; a zero-order hold generates a continuous-time signal according to

$$u(t) = u(kT), \quad kT \le t < (k+1)T. \tag{7.1}$$

There are two aspects of time-discretization that need to be understood: we need to study the process of converting continuous-time into discrete-time signals and vice versa, and we need to know how to represent systems that operate only on discrete-time signals. Systems that operate only on discrete-time signals are called *discrete-time systems*,

whereas systems that involve both continuous-time and discrete-time signals are called *sampled-data systems.*

7.1 Discrete-Time Systems - z-Transform and State Space Models

The block labelled "difference equations" in Fig. 7.2 represents the dynamic behaviour of the controller. The input to this block is the sequence of numbers $y(kT)$, and the output is the sequence of numbers $u(kT)$, where $k = 0, 1, \ldots$ The factor T in the time argument reflects the fact that these sequences are generated by sampling continuous-time signals every T seconds. The processor that generates the control action is only sequentially processing incoming numbers - the sampling period itself has no effect on the sequence of numbers that is produced as output. (We assume however that the sampling period is longer than the computation time required for generating the next output value). When studying discrete-time systems, we will use a simplified notation and write $y(k)$ and $u(k)$ instead of $y(kT)$ and $u(kT)$, but we will return to the latter notation when we are interested in the interaction between discrete-time signals and continuous-time systems. Formally we let the integer variable k run from $-\infty$ to $+\infty$, but - similar to our treatment of continuous-time systems - we will assume that

$$x(k) = 0, \quad k < 0$$

i.e. all signals that we consider are zero for $k < 0$. Important discrete-time signals which we will use frequently in this chapter are the *discrete-time unit impulse*

$$\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & \text{else} \end{cases} \tag{7.2}$$

and the *discrete-time unit step function*

$$\sigma(k) = \begin{cases} 1, & k \geq 0 \\ 0, & k < 0 \end{cases} \tag{7.3}$$

Another important signal type is described by the discrete-time exponential function

$$x(k) = \begin{cases} e^{-ak}, & k \geq 0 \\ 0, & k < 0 \end{cases} \tag{7.4}$$

Note that using the step function we can also write $x(k) = e^{-ak}\sigma(k)$.

Just as the dynamic behaviour of a continuous-time system can be represented by differential equations, the dynamics of a discrete-time system can be described - as the block label in Fig. 7.2 suggests - by *difference equations.* A linear difference equation has the form

$$y(k) + a_1 y(k-1) + \ldots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \ldots + b_n u(k-n) \tag{7.5}$$

If $b_0 \neq 0$ the output $y(k)$ at time k depends on the input $u(k)$ at the same time - the system responds instantaneously. Physically realizable systems cannot do this, and when a discrete-time model is used to represent a plant to be controlled we will often assume $b_0 = 0$. However, models with $b_0 \neq 0$ are sometimes used as an approximation of a controller when its response is fast compared with the sampling period.

A comment is also in order about the number of samples involved in the above difference equation. The integer n is used to denote the number of past input and output samples that have an effect on the present value of the output - equation (7.5) is called an nth order difference equation. In general, the number of past input samples and output samples that determine the present output need not be the same, in that case we let n denote the larger one of these values and set the coefficients that are not needed to zero.

Example 7.1

The use of difference equations for describing discrete-time dynamic behaviour is now illustrated with an example. Consider the system

$$y(k) + 0.5y(k-1) = u(k-1) \tag{7.6}$$

and the input signal $u(k) = \sigma(k)$. Because input and output are zero for $k < 0$ we have $y(0) = 0$. The solution for $k > 0$ is obtained by computing

$$y(k) = -0.5y(k-1) + u(k-1)$$

successively, this yields

k	0	1	2	3	4	5	...
$u(k)$	1	1	1	1	1	1	...
$y(k)$	0	1	0.5	0.75	0.625	0.6875	...

The output is oscillating and appears to converge towards a value between 0.625 and 0.6875. If the difference equation (7.6) is changed to

$$y(k) - 0.5y(k-1) = u(k-1) \tag{7.7}$$

we have to solve

$$y(k) = 0.5y(k-1) + u(k-1)$$

yielding

k	0	1	2	3	4	5	...
$u(k)$	1	1	1	1	1	1	...
$y(k)$	0	1	1.5	1.75	1.875	1.9375	...

Now the solution is monotonically increasing and appears to converge to a value around 2. We will return to this example after introducing the z-transform.

The z-Transform

When dealing with continuous-time systems, we found it convenient to transform linear differential equations into algebraic equations in frequency domain; this allows us to represent linear systems by transfer function models. The tool for doing this is the Laplace transform, and a similar tool - the z-transform - is available for transforming difference equations into algebraic frequency domain equations. Recall that the Laplace transform of a continuous-time signal $x(t)$ which is zero for negative time is defined as

$$\mathcal{L}[x(t)] = X(s) = \int_0^\infty x(t)e^{-st}dt$$

This transformation maps functions of time into functions of the complex variable s. Its usefulness comes from the fact that (assuming $x(0) = 0$)

$$\mathcal{L}[\dot{x}(t)] = sX(s)$$

which means that taking derivatives in time domain reduces to multiplication by the complex variable s in frequency domain. For a discrete-time signal $x(k)$ which is zero for $k < 0$, we now define its z-transform as

$$\mathcal{Z}[x(k)] = X(z) = \sum_{k=0}^\infty x(k)z^{-k} \tag{7.8}$$

This transformation maps functions of the discrete time variable k into functions of the complex variable z. Equation (7.8) defines the z-transform as an infinite power series in z^{-1}. Just as the Laplace integral converges for all s such that $\operatorname{Re}(s) > c$ if the limit

$$\lim_{t\to\infty} e^{-ct}|x(t)|$$

exists, the power series in (7.8) converges for all z such that $|z| > r$ if the limit

$$\lim_{k\to\infty} r^k|x(k)|$$

exists. Because the signals we usually encounter in control systems do not grow faster than exponentially, both Laplace transform and z-transform converge, so that the region of convergence is usually not an issue of concern.

That the z-transform is as useful as the Laplace transform is due to the fact that

$$\mathcal{Z}[x(k-1)] = \sum_{k=0}^\infty x(k-1)z^{-k} = \sum_{l=-1}^\infty x(l)z^{-l-1} = z^{-1}\sum_{l=0}^\infty x(l)z^{-l}$$

where a change of variables has been used together with the assumption that $x(k) = 0$ for $k \leq 0$. Comparing the last expression with the definition of the z-transform, we find

$$\mathcal{Z}[x(k-1)] = z^{-1}X(z) \tag{7.9}$$

Repeating this n times, one obtains

$$\mathcal{Z}[x(k-n)] = z^{-n}X(z) \tag{7.10}$$

Similarly one can show that

$$\mathcal{Z}[x(k+1)] = z(X(z) - x(0))$$

Pulse Transfer Function

We can use this result to take the difference equation (7.5) into frequency domain. Applying (7.9) term by term (and assuming $b_0 = 0$) we obtain

$$Y(z) + a_1 z^{-1} Y(z) + \ldots + a_n z^{-n} Y(z) = b_1 z^{-1} U(z) + \ldots + b_n z^{-n} U(z)$$

or

$$(1 + a_1 z^{-1} + \ldots + a_n z^{-n}) Y(z) = (b_1 z^{-1} + \ldots + b_n z^{-n}) U(z)$$

Dividing by the polynomial on the left hand side, we get

$$Y(z) = G(z) U(z) \tag{7.11}$$

where $G(z)$ is the *pulse transfer function*

$$G(z) = \frac{b_1 z^{-1} + \ldots + b_n z^{-n}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} = \frac{b_1 z^{n-1} + \ldots + b_n}{z^n + a_1 z^{n-1} + \ldots + a_n} \tag{7.12}$$

This shows that the z-transform can be used in a way similar to the Laplace transform to obtain a transfer function model of a discrete-time system. Before we study the dynamic behaviour of discrete-time systems, we first compute the z-transform of the three discrete-time signals introduced before: unit impulse, unit step and exponential function.

Unit Impulse

The discrete-time unit impulse was defined in (7.2), and from the definition we have immediately

$$\mathcal{Z}[\delta(k)] = \sum_{k=0}^{\infty} \delta(k) z^{-k} = 1 \tag{7.13}$$

Unit Step

From (7.3) we obtain

$$\mathcal{Z}[\sigma(k)] = \sum_{k=0}^{\infty} z^{-k} = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1} \tag{7.14}$$

Note that the above geometric series converges if $|z| > 1$.

Exponential Function

The z-transform of signals described by (7.4) can be computed as

$$\mathcal{Z}[x(k)] = \sum_{k=0}^{\infty} e^{-ak} z^{-k} = \sum_{k=0}^{\infty} (e^{-a} z^{-1})^k = \frac{1}{1 - e^{-a} z^{-1}} = \frac{z}{z - e^{-a}} \tag{7.15}$$

where the region of convergence is $|z| > e^{-a}$.

Table 7.1 lists some commonly encountered discrete-time signals together with their z-transforms. Except for the first and the last entry, the discrete-time signals are assumed to have been generated by sampling the continuous-time signals in the second column with sampling time T. The corresponding Laplace transforms are also shown.

$X(s)$	$x(t)$	$x(k)$	$X(z)$
		$\delta(k)$	1
$\dfrac{1}{s}$	$\sigma(t)$	$\sigma(k)$	$\dfrac{1}{1-z^{-1}}$
$\dfrac{1}{s+a}$	$e^{-at}\sigma(t)$	$e^{-akT}\sigma(k)$	$\dfrac{1}{1-e^{-aT}z^{-1}}$
$\dfrac{1}{s^2}$	$t\sigma(t)$	$kT\sigma(k)$	$\dfrac{Tz^{-1}}{(1-z^{-1})^2}$
$\dfrac{a}{s(s+a)}$	$(1-e^{-at})\sigma(t)$	$(1-e^{-akT})\sigma(k)$	$\dfrac{(1-e^{-aT})z^{-1}}{(1-z^{-1})(1-e^{-aT}z^{-1})}$
$\dfrac{b-a}{(s+a)(s+b)}$	$(e^{-at}-e^{-bt})\sigma(t)$	$(e^{-akT}-e^{-bkT})\sigma(k)$	$\dfrac{(e^{-aT}-e^{-bT})z^{-1}}{(1-e^{-aT}z^{-1})(1-e^{-bT}z^{-1})}$
$\dfrac{1}{(s+a)^2}$	$te^{-at}\sigma(t)$	$kTe^{-akT}\sigma(k)$	$\dfrac{Te^{-aT}z^{-1}}{(1-e^{-aT}z^{-1})^2}$
$\dfrac{s}{(s+a)^2}$	$(1-at)e^{-at}\sigma(t)$	$(1-akT)e^{-akT}\sigma(k)$	$\dfrac{1-(1+aT)e^{-aT}z^{-1}}{(1-e^{-aT}z^{-1})^2}$
$\dfrac{b}{s^2+b^2}$	$\sin bt\,\sigma(t)$	$\sin bkT\,\sigma(k)$	$\dfrac{z^{-1}\sin bT}{1-2z^{-1}\cos bT+z^{-2}}$
$\dfrac{s}{s^2+b^2}$	$\cos bt\,\sigma(t)$	$\cos bkT\,\sigma(k)$	$\dfrac{1-z^{-1}\cos bT}{1-2z^{-1}\cos bT+z^{-2}}$
		$a^k\sigma(k)$	$\dfrac{1}{1-az^{-1}}$

Table 7.1: Table of z-transforms

Final Value Theorem and Steady State Gain

In Exercise 7.2 it is shown that the final value of a discrete-time signal $x(k)$ - if it exists - can be obtained from

$$\lim_{k \to \infty} x(k) = \lim_{z \to 1} (z-1)X(z) \tag{7.16}$$

This result can be used to compute the static gain of a system with transfer function $G(z)$. The step response is

$$Y(z) = G(z)\frac{z}{z-1}$$

and using (7.16) yields

$$y(\infty) = \lim_{z \to 1} (z-1)G(z)\frac{z}{z-1} = G(1) \tag{7.17}$$

provided the limit exists. Thus, the static gain of a discrete-time system is obtained by evaluating its transfer function at $z = 1$.

Example 7.2

Returning to the example (7.6), we have

$$G(z) = \frac{z^{-1}}{1 + 0.5z^{-1}} = \frac{1}{z + 0.5}$$

and thus $G(1) = 0.67$. For the system in (7.7) we obtain

$$G(z) = \frac{1}{z - 0.5}$$

and a static gain $G(1) = 2$, confirming the time domain solution. The use of pulse transfer functions is further explored in Exercise 7.1.

Initial Value Theorem

The initial value $x(0)$ of a discrete-time signal $x(k)$ is given by

$$x(0) = \lim_{z \to \infty} X(z)$$

This follows directly from

$$X(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + \dots$$

Impulse Response and Discrete Convolution

We know that the transfer function of a continuous-time system is equal to the Laplace transform of its impulse response. In frequency domain, the output is given by the product of input and transfer function. In time domain, this frequency domain product corresponds to the convolution of input signal and impulse response.

A similar time domain interpretation of the frequency domain relationship (7.11) between input and output can be given for discrete-time systems. Consider a discrete-time system with transfer function $G(z)$, and assume that a unit impulse input $u(k) = \delta(k)$ is applied. In frequency domain we have

$$Y(z) = G(z)U(z) = G(z)$$

Thus $G(z)$ is the z-transform of the impulse response $y(k) = g(k)$, and we have

$$G(z) = \sum_{k=0}^{\infty} g(k)z^{-k}$$

In Exercise 7.9 it is shown that for an arbitrary input signal $u(k)$ the output is given by

$$y(k) = \sum_{l=0}^{k} g(l)u(k-l) \tag{7.18}$$

which is a discrete-time version of the convolution integral.

Discrete-Time State Space Models

It is also possible to represent a discrete-time model in state space form. Consider the system shown in Fig. 7.3.
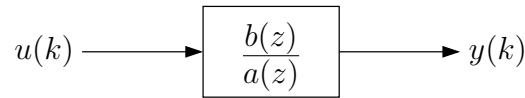
Figure 7.3: Pulse transfer function model

Just as for continuous-time systems, a simulation model can be constructed by splitting the transfer function up as shown in Fig. 7.4.
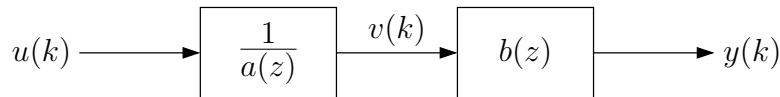
Figure 7.4: Pulse transfer function model

Using the fictitious signal $v(k)$, it is possible to construct the discrete-time version of the controller canonical form. In contrast to continuous-time systems, here it is not a chain of integrators but a chain of time delay blocks that forms the core of the simulation model. This reflects the fact that we are dealing with difference equations rather than differential equations. Accordingly, the state equation is a first order vector difference equation instead of a vector differential equation. A state space model of a discrete-time

MIMO system has the form

$$x(k + 1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = C x(k) + D u(k) \tag{7.19}$$

where Φ and Γ denote the discrete-time system matrix and input matrix, and C and D the measurement and feedthrough matrix, respectively. Note that for a SISO system $D = 0$ if $b_0 = 0$ in (7.5) or (7.12); for a MIMO system, the b_0s of all entries of the transfer function matrix must be zero. A way of constructing the discrete-time controller canonical form of a SISO model from the representation in Fig. 7.4 is shown in Exercise 7.4.

Since the state equation is a difference equation, its solution can be constructed recursively. Assume that $x(0)$ and an input sequence $u(k)$, $k = 0, 1, 2, \ldots$ for the model (7.19) are given. Starting at time $k = 1$ we have

$$x(1) = \Phi x(0) + \Gamma u(0)$$
$$x(2) = \Phi x(1) + \Gamma u(1)$$
$$= \Phi^2 x(0) + \Phi \Gamma u(0) + \Gamma u(1)$$
$$x(3) = \Phi x(2) + \Gamma u(2)$$
$$= \Phi^3 x(0) + \Phi^2 \Gamma u(0) + \Phi \Gamma u(1) + \Gamma u(2)$$
$$\vdots$$
$$x(k) = \Phi^k x(0) + \sum_{l=0}^{k-1} \Phi^{k-l-1} \Gamma u(l) \tag{7.20}$$

The first term on the right hand side of the last equation represents the unforced (zero-input) response, whereas the second term describes the forced (zero-initial-state) response.

Pulse Transfer Function from State Space Model

Taking the z-transform of (7.19) and assuming $x(0) = 0$ we have

$$z X(z) = \Phi X(z) + \Gamma U(z)$$

which can be solved for $X(z)$ to obtain

$$Y(z) = \left[C(zI - \Phi)^{-1} \Gamma + D \right] U(z) = G(z) U(z)$$

Thus, the relationship between state space model and transfer function of a discrete-time system is

$$G(z) = C(zI - \Phi)^{-1} \Gamma + D \tag{7.21}$$

Note that this relationship is the same as that for continuous-time systems. In particular, the poles of the pulse transfer function are eigenvalues of the system matrix Φ.

Stability of Discrete-Time Systems

Following Definition 2.1 for continuous-time systems, we define the stability of an unforced discrete-time system in terms of its state space model as follows.

Definition 7.1   *An unforced system $x(k+1) = \Phi x(k)$ is said to be stable if for all $x(0) = x_0$, $x_0 \in \mathbb{R}$ we have $x(k) \to 0$ as $k \to \infty$.*

From (7.20), the response to a non-zero initial state $x(0)$ is

$$x(k) = \Phi^k x(0)$$

Assuming that it is possible to diagonalize Φ, the elements of the response $x(k)$ are combinations of terms λ_i^k, where λ_i are the eigenvalues of Φ. Since stability requires that all solutions go to zero as k goes to infinity, we need

$$|\lambda_i| < 1, \quad i = 1, \ldots, n$$

We thus have

Theorem 7.1   *A discrete-time system $x(k+1) = \Phi x(k)$ is stable if and only if all eigenvalues of Φ are strictly inside the unit disc.*

From 7.21 and the relationship between pulse transfer function poles and eigenvalues of Φ, we expect that a pulse transfer function is stable if its poles are strictly inside the unit disc. This is indeed confirmed in the next section where stability of sampled-data systems is discussed.

Controllability and Observability

It is straightforward to extend the definitions of controllability and observability to a discrete-time system with state space model (7.19).

Definition 7.2

The discrete-time system (7.19) is said to be controllable if for any initial state $x(0) = x_0$, $k_f > 0$ and final state x_f there exists a control sequence $u(k)$, $0 \le k \le k_f$, such that the solution of (7.19) satisfies $x(k_f) = x_f$. Otherwise, the system is said to be uncontrollable.

Definition 7.3

The discrete-time system with state space model (7.19) is said to be observable if there exists a $k_f > 0$ such that the initial state $x(0)$ can be uniquely determined from the input sequence $u(k)$ and the output sequence $y(k)$ in the interval $0 \le k \le k_f$. Otherwise, the system is said to be unobservable.

Whereas the controllability and observability Gramians take different forms for discrete-time systems, the controllability and observability matrices have the same form as for continuous-time systems.

To show this for the controllability matrix, consider the solution of (7.19) given by (7.20) at $k = n$

$$
\begin{aligned}
x(n) &= \Phi^n x(0) + \sum_{l=0}^{n-1} \Phi^{n-l-1} \Gamma u(l) \\
&= \Phi^n x(0) + \Phi^{n-1} \Gamma u(0) + \ldots + \Gamma u(n-1) \\
&= \Phi^n x(0) + \mathcal{C}_d \, U
\end{aligned}
$$

where

$$
\mathcal{C}_d = [\Gamma \ \ \Phi\Gamma \ \ \ldots \ \ \Phi^{n-1}\Gamma]
$$

is the discrete-time controllability matrix and

$$
U = [u^T(n-1) \ \ u^T(n-2) \ \ \ldots \ \ u^T(0)]
$$

From

$$
x(n) - \Phi^n x(0) = \mathcal{C}_d \, U
$$

we have

Theorem 7.2 *There exists a sequence $\{u(0), u(1), \ldots, u(n-1)\}$ that takes the system (7.19) from any initial state to any desired state in no more than n steps if and only if $\mathcal{C}_d$ has rank n.*

Note that the part of the state space that can be reached from the origin (the controllable subspace) is spanned by the columns of the controllability matrix.

To derive an equivalent result for observability of discrete-time systems, we consider without loss of generality instead of (7.19) the unforced system

$$
\begin{aligned}
x(k+1) &= \Phi x(k) \\
y(k) &= C x(k)
\end{aligned}
$$

because - as in the continuous-time case - the effect of a known input sequence can be eliminated from the model. Assume that $y(0)$, $y(1), \ldots, y(n-1)$ are known. We then have

$$
\begin{aligned}
y(0) &= C x(0) \\
y(1) &= C x(1) = C \Phi x(0) \\
&\vdots \\
y(n-1) &= C \Phi^{n-1} x(0)
\end{aligned}
$$

or in vector notation

$$Y = \mathcal{O}_d \, x(0)$$

where

$$\mathcal{O}_d = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix}$$

is the discrete-time observability matrix and

$$Y = [y^T(0) \ \ y^T(1) \ \ \dots \ \ y^T(n-1)]$$

We thus obtain the following.

Theorem 7.3 *The discrete-time system (7.19) is observable if and only if the observability matrix $\mathcal{O}_d$ has rank n.*

Stabilizability and detectability of discrete-time systems can be defined in the same way as for continuous-time systems, also the Kalman decomposition and the concept of a minimal realization.

7.2 Sampled Data Systems

In (7.8) we defined the z-transform of a discrete-time signal $x(k)$ without making any reference to the possibility that this sequence may have been obtained by sampling a continuous-time signal $x(t)$ at time instants $t = kT$, $k = 0, 1, 2, \dots$ Therefore the notion of a sampling period T played no role when we introduced the pulse transfer function. We now return to the continuous-time origin of the discretized signals, and we will explore the relationship between the Laplace transform of a continuous-time signal $x(t)$ and its sampled version $x(kT)$. For this purpose, we will now have a closer look at the sampling process. A useful mathematical model for the process of sampling a signal is given by the *impulse sampler* shown in Fig. 7.5. In this model the sampling process is viewed as modulating a train of delta impulses $\delta(t)$, $\delta(t-T)$, $\delta(t-2T)$ etc., i.e. the sampled signal is represented by a train of delta impulses where the impulse weight at time kT is equal to the value of the sampled signal $x(kT)$ at this sampling instant.

The output $x^*(t)$ of the impulse sampler is then given by

$$x^*(t) = \sum_{k=0}^{\infty} x(kT)\delta(t - kT) \tag{7.22}$$

and its Laplace transform is

$$\mathcal{L}[x^*(t)] = \sum_{k=0}^{\infty} x(kT)e^{-kTs} \tag{7.23}$$
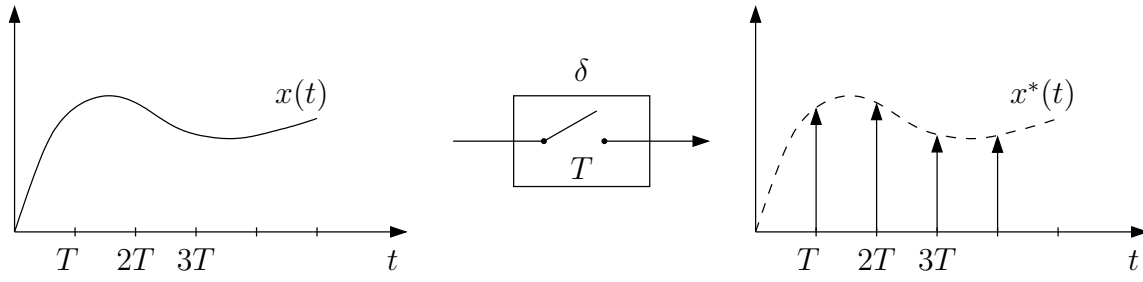
Figure 7.5: Impulse sampler

Comparing this with the z-transform of $x(kT)$

$$\mathcal{Z}[x(kT)] = \sum_{k=0}^{\infty} x(kT)z^{-k}$$

we find that these two transforms are equivalent if we define the complex variable z as

$$z = e^{Ts} \tag{7.24}$$

Example 7.3

To illustrate the above relationship between the complex variables of Laplace and z-transform, respectively, consider the exponential function

$$x(t) = Ke^{-at}\sigma(t)$$

Its Laplace transform is

$$X(s) = \frac{K}{s+a}$$

Such a signal would arise as a component of the step response of a system with a pole at $s = -a$. Now assume $x(t)$ is sampled with sampling period T so that

$$x(kT) = Ke^{-akT}\sigma(kT)$$

From (7.15) the z-transform of the sampled signal is

$$\mathcal{Z}[x(kT)] = \frac{K}{1 - e^{-aT}z^{-1}}$$

Note that we now include the sampling period T. Again, this signal would be a component of the step response of a discrete-time system with a pole at $z = e^{-aT}$. Comparing the pole locations of the continuous-time and the discrete-time system shows that the continuous-time pole in the s-plane is mapped into the z-plane according to (7.24).
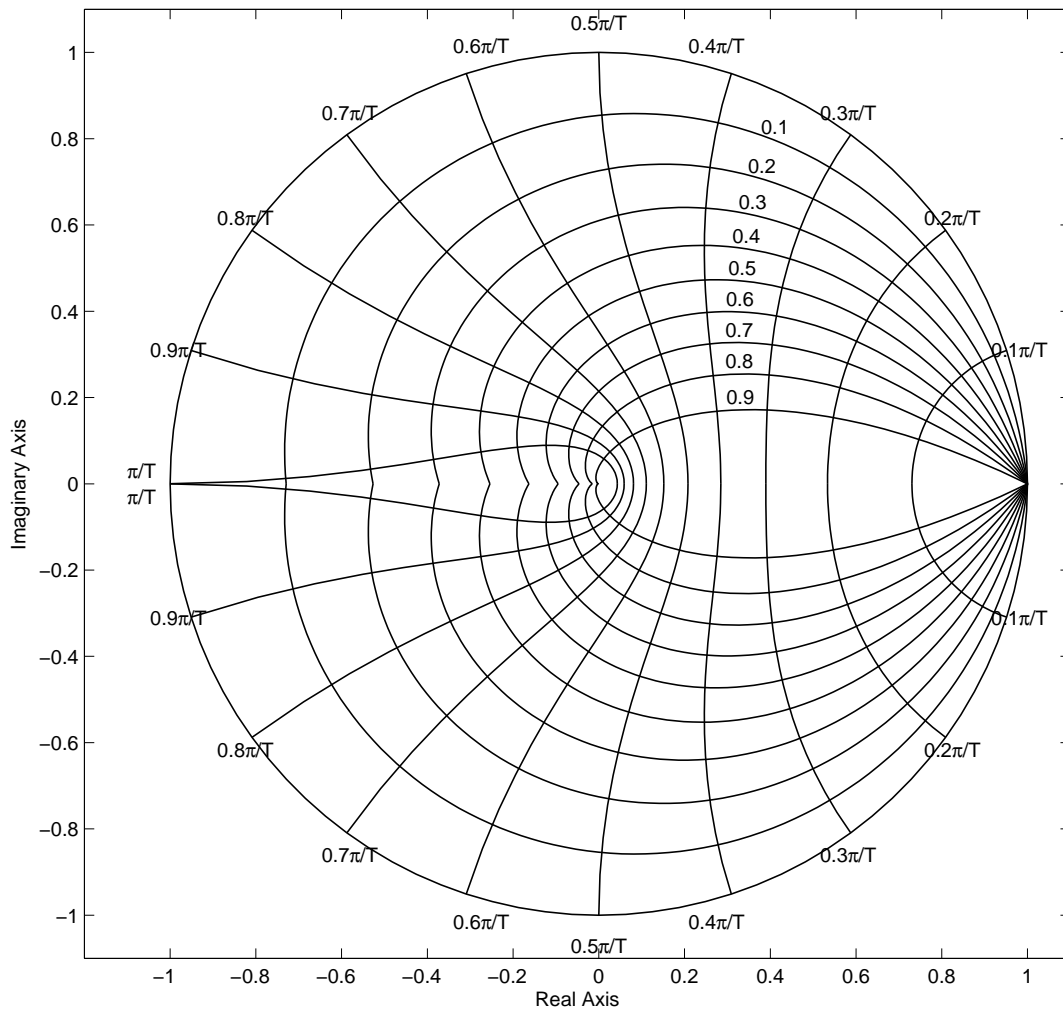
Figure 7.6: Curves of constant damping and constant natural frequency

Pole Location and Stability

A continuous-time system is stable if all poles satisfy Re $s < 0$, i.e. if all poles are in the left half plane. Moreover, a pole or a pole pair with a real part satisfying Re $s < -\sigma$ will correspond to an exponential decay with time constant less than $1/\sigma$. Using the definition (7.24), an equivalent interpretation of pole locations can be given for a discrete-time system that is obtained by sampling a continuous-time system. Points in the s-plane satisfying Re $s < 0$ are mapped into points in the z-plane that satisfy $|z| < 1$, i.e. the left half plane is mapped into a disc centered at the origin with radius 1 - the *unit disc*. Discrete-time systems are stable if all poles are located inside the unit disc. Note that this result - obtained for sampled continuous-time systems - is consistent with Theorem

7.1 that was derived for purely discrete-time systems.

Poles corresponding to a response with time constant faster than $1/\sigma$ satisfy $|z| < e^{-\sigma T}$ - they are located inside a disc with radius $e^{-\sigma T} < 1$. We can take the analogy between continuous-time and discrete-time systems further: consider the second order continuous-time system

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The poles of this system are mapped by (7.24) into the roots of

$$z^2 + a_1 z + a_0$$

where

$$a_1 = -2e^{-\zeta\omega_n T} \cos\left(\sqrt{1 - \zeta^2}\omega_n T\right)$$

and

$$a_0 = e^{-2\zeta\omega_n T}$$

Figure 7.6 shows the curves of constant ζ and $\omega_n T$ for the sampled system. This figure relates the damping ratio and the natural frequency of a discrete-time second order system to its pole locations. See also Exercise 7.3.

7.3 Sampled-Data Controller Design

We are now in a position to address the problem of designing a digital controller for the control loop shown in Fig. 7.1. The controller is a discrete-time system, while the plant inputs and outputs are continuous-time signals. There are two ways of dealing with this hybrid nature of the control loop:

1. One can first design a continuous-time controller $C(s)$, using standard design techniques for continuous-time systems, and then try to find a pulse transfer function $D(z)$ that approximates the dynamic behaviour of the controller $C(s)$.

2. Alternatively, the continuous-time plant can be discretized first, and a discrete-time controller $D(z)$ can be designed directly for the discretized plant. Both approaches are used in practice and will now be discussed.

Continuous-Time Design - Tustin Approximation

Fig. 7.7 shows a slightly modified version of the control loop of Fig. 7.1 with a continuous-time plant transfer function $G(s)$ and the discrete-time controller $D(z)$. The controller input is connected to a sampler, and the controller output to a zero order hold (zoh).
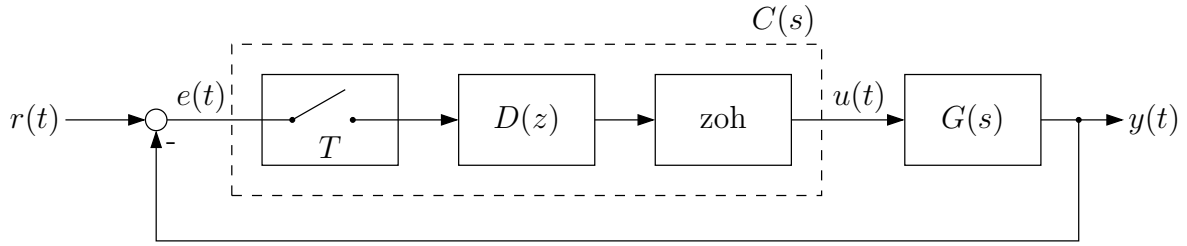
Figure 7.7: Continuous-time controller design

Note that the input $e(t)$ and the output $u(t)$ of the controller are continuous-time signals. Assume that a continuous-time controller $C(s)$ has been designed that meets the design specifications. The task is then to find a pulse transfer function $D(z)$ such that the dashed block - which is a continuous-time system due to the presence of the sampler and the hold - approximates $C(s)$ sufficiently well so that the design specifications are also met with a digital controller. There are several ways of constructing discrete-time approximations of continuous-time transfer functions; one of them - known as *Tustin approximation* - will be discussed next.

We begin with the discrete-time approximation of the basic building block of continuous-time systems, an integrator. Thus, consider a pure integral controller

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

At time $t = kT$ we have

$$u(kT) = \int_0^{kT} e(\tau)d\tau = \int_0^{kT-T} e(\tau)d\tau + \int_{kT-T}^{kT} e(\tau)d\tau$$

or

$$u(kT) = u(kT - T) + \int_{kT-T}^{kT} e(\tau)d\tau$$
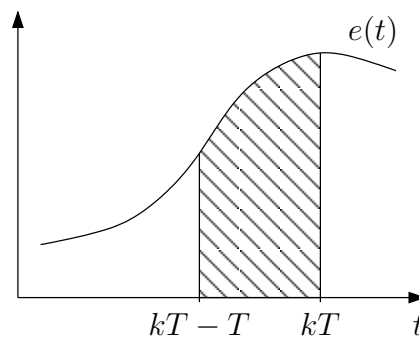
Figure 7.8: Sampling an integrator

The second term on the right hand side represents the area under $e(t)$ marked in Fig. 7.8. For the Tustin method, $e(t)$ is approximated between the last two samples by a straight

line. This yields

$$u(kT) = u(kT - T) + \frac{T}{2} \left(e(kT - T) + e(kT) \right)$$

or in z-domain

$$U(z) = z^{-1}U(z) + \frac{T}{2}(z^{-1}E(z) + E(z))$$

Thus we find the pulse transfer function from e to u as

$$\frac{U(z)}{E(z)} = \frac{T}{2} \cdot \frac{1 + z^{-1}}{1 - z^{-1}} = \frac{1}{\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

Similarly one can check that for a controller with continuous-time transfer function

$$C(s) = \frac{K}{s + a}$$

the same approximation as above leads to a discrete-time controller

$$D(z) = \frac{K}{\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} + a}$$

More generally, making the substitution

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{2}{T} \cdot \frac{z - 1}{z + 1} \tag{7.25}$$

in every term in the controller transfer function that contains s yields a pulse transfer function $D(z)$ that is based on the above trapezoidal approximation of an integral.
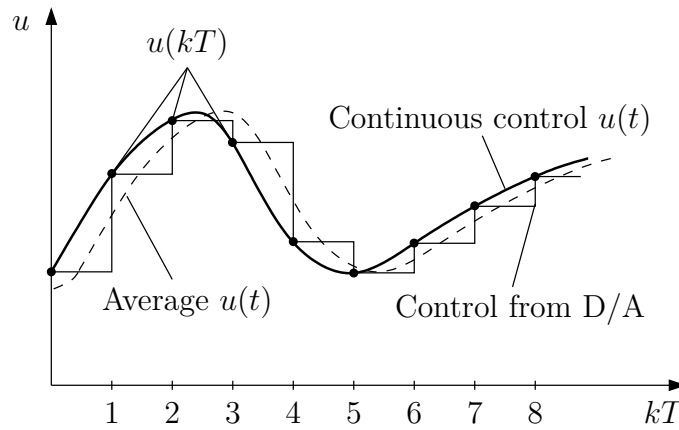
Figure 7.9: Introduction of a time delay through the hold operation

If a discrete-time controller $D(z)$ is implemented in the control system in Fig. 7.7, the value of the control signal $u(t)$ is held constant by the zero order hold block connected to the controller output until the next value is available, so that the continuous-time control input consists of steps as shown in Fig. 7.2. Due to the fact that physical systems have a

limited bandwidth, the effect of this control input on the plant will have the same effect as a low-pass filtered version of this step-shaped signal. This is illustrated in Fig. 7.9, where it can be seen that the hold operation introduces a time-delay of approximately $T/2$. Thus, while the idea was to approximate the behaviour of the continuous-time controller $C(s)$ by a discrete-time controller, the discretized controller $D(z)$ is actually emulating the effect of a controller

$$\tilde{C}(s) \approx C(s)e^{-\frac{T}{2}s} \approx C(s)\frac{2/T}{s + 2/T}$$

Taking this time delay into account when designing the continuous-time controller gives a reasonable prediction of the effect of the zero-order hold when the sampling rate is slower than $20\omega_b$. To keep this effect small, the sampling frequency $\omega_s = 2\pi/T$ should be much higher than the system bandwidth ω_b - experience suggests that the sampling frequency should be at least $20 \sim 30\omega_b$.

Discrete-Time Design - Exact Discretization

A modified version of the control loop in Fig. 7.7 is shown in Fig. 7.10. Now the whole control loop is viewed as a discrete-time system - the idea is to model the continuous-time plant only in the sampling instants and to carry out the design in discrete-time.
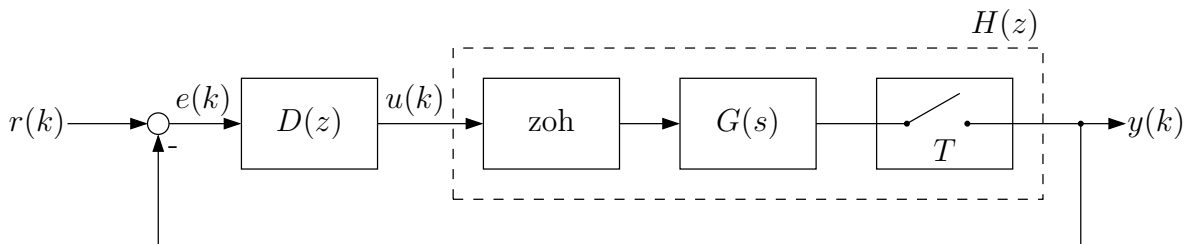
Figure 7.10: Discrete-time controller design

For this purpose, the sampler has been moved "around the loop" to the plant output. The task is now to find a discrete-time transfer function $H(z)$ that describes the behaviour of the system inside the dashed box, i.e. the plant with hold and sampler. We will see that - in contrast to the Tustin approximation - an *exact discretization* is possible in this case, in the sense that the discretized model describes the behaviour of the continuous-time plant exactly in the sampling instants.

For the derivation of the discretized model we will use a state space realization of the plant transfer function. Thus, assume that

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

is a realization of $G(s)$. To describe the evolution of the state vector $x(t)$ from time kT to $kT + T$, we recall that the solution of the state equation at time t, starting at $t_0 < t$, is given by

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau$$

Substituting $kT + T$ and kT for t and t_0, respectively, yields

$$x(kT + T) = e^{AT}x(kT) + \int_{kT}^{kT+T} e^{A(kT+T-\tau)}Bu(\tau)d\tau$$

Now we can exploit the fact that the shape of the input signal between two sampling instants is known exactly: due to the presence of the zero-order hold we have

$$u(kT + \tau) = u(kT), \quad 0 \le \tau < T$$

With the change of variables $t = kT + T - \tau$ we obtain

$$x(kT + T) = e^{AT}x(kT) + \int_0^T e^{At}Bdt \, u(kT) \tag{7.26}$$

Defining

$$\Phi = e^{AT}, \quad \Gamma = \int_0^T e^{At}Bdt \tag{7.27}$$

this becomes

$$x(kT + T) = \Phi x(kT) + \Gamma u(kT) \tag{7.28}$$

This is the discrete-time state equation (7.19) that was introduced earlier, now obtained by sampling a continuous-time system at sampling instants kT. Given a continuous-time model, the discrete-time matrices Φ and Γ can be computed from (7.27). Note that the discrete-time model describes the continuous-time system *exactly* in the sampling instants kT. The reason for this is that due to the presence of the zero-order hold the shape of the input signal between sampling instants is known exactly - which is not the case for the Tustin approximation. No time delay is produced in this approach - the discrete-time model describes the continuous-time system exactly in the sampling instants even when the sampling frequency is low. There are however other considerations that demand sampling to be sufficiently fast, this will be discussed in the next section.

Having obtained a discrete-time plant model, one can design a controller directly by using discrete-time versions of continuous-time techniques such as discrete-time root locus design, pole placement or discrete-time optimal state feedback and state estimation.

Optimal State Feedback and Optimal State Estimation

Given the discrete-time state equation (7.28) that describes the states exactly in the sampling instants, one can use the idea of state feedback just as in the continuous-time case: the control law $u(k) = Fx(k) + u_v(k)$ yields the closed-loop system

$$x(k + 1) = (\Phi + \Gamma F)x(k) + \Gamma u_v(k) \tag{7.29}$$

If the system is controllable, the state feedback gain matrix F can be chosen to obtain any desired closed-loop poles. Moreover, one can compute the state feedback gain that minimizes the quadratic cost function

$$V_d = \sum_{k=0}^{\infty} \left(x^T(k)Qx(k) + u^T(k)Ru(k) \right) \tag{7.30}$$

by using a discrete-time version of the method discussed in Section 6.9. The dual problem of finding an optimal state estimator can be solved with the same tools.

Examples of discrete-time controller design are provided in Exercises 7.5, 7.6 and 7.8.

Deadbeat Control

While discrete-time versions exist for all continuous-time design methods, there is a discrete-time control scheme for which no continuous-time equivalent exists. This scheme is called *deadbeat control* because it brings a system from any initial state to the origin in at most n sampling periods. Given a controllable system (7.28), the idea is to use state feedback to assign all closed-loop poles to the origin. Assuming that the model (7.28) is in controller canonical form, the closed loop matrix then has the form

$$\Phi + \Gamma F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

Note that this closed-loop matrix is nilpotent, i.e. we have

$$(\Phi + \Gamma F)^n = 0$$

Since the solution of the closed-loop state equation at $k = n$ is (assuming $u_v(k) = 0$)

$$x(n) = (\Phi + \Gamma F)^n x(0)$$

it is clear that any initial state is taken to the origin in no more than n steps.

It should however be mentioned that deadbeat control is mainly of theoretical interest: the only design parameter is the sampling time T, and when a short sampling time is chosen the control signal that would theoretically result in a settling time of n sampling periods is typically very large. In practice, the actuators would be driven into saturation, resulting in a longer settling time.

Deadbeat control is explored in Exercise 7.7.

Zeros of Sampled-Data Systems We have seen earlier in this section that when input and output signals of a continuous-time system are sampled at $t = kT, \; k = 0, 1, 2, \dots$

using the Tustin approximation or exact (zero-order hold) discretization, a pole at $s = p_i$ is mapped into a pole at $z = e^{p_i T}$ of the resulting discrete-time system. Unfortunately, it is not possible to give an equally simple formula for the mapping of zeros. However, results are available for the limiting case where the sampling interval goes to zero; these can also be used to approximate the mapping of zeros when sampling is sufficiently fast.

Consider a continuous-time system with transfer function

$$G(s) = K_0 \frac{(s - z_1)(s - z_2) \ldots (s - z_m)}{(s - p_1)(s - p_2) \ldots (s - p_n)}, \quad m < n \tag{7.31}$$

This system has n poles and m zeros and therefore a pole excess of $n - m$. Assume we want to find a pulse transfer function

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_n z^{-n}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} = \frac{b_0 z^n + b_1 z^{n-1} + \ldots + b_n}{z^n + a_1 z^{n-1} + \ldots + a_n} \tag{7.32}$$

that describes the behaviour of $G(s)$ sampled with sampling time T. It is straightforward to check that when using the Tustin approximation, i.e. making the substitution (7.25), in the resulting pulse transfer function we will always have $b_0 \neq 0$. Thus the sampled system has n zeros, $n - m$ more than the underlying continuous-time system. One can also verify that when we use the model

$$H(z) = C(zI - \Phi)^{-1}\Gamma + D \tag{7.33}$$

where Φ and Γ are computed from (7.27), the discrete-time system will have n zeros if $D \neq 0$ and $n - 1$ zeros if $D = 0$.

This means that the sampling process can introduce additional zeros; these are referred to as *sampling zeros*. A useful fact, stated here without proof, is the following. Assume that the continuous-time system (7.31) is sampled with sampling time T, and that exact (zero-order hold) discretization results in the discrete-time model (7.32). Then one can show that when the sampling time approaches zero, m zeros of the discrete-time model will approach $e^{z_i T}$, where $z_i, \ i = 1, 2, \ldots, m$ are the continuous-time zeros. The remaining $n - m - 1$ sampling zeros will approach the zeros of the system shown in Fig. 7.11. This issue is explored further in Exercise 7.11.
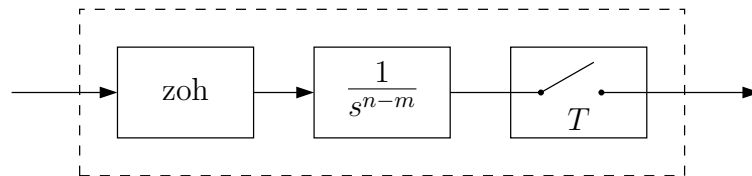
Figure 7.11: Model for the approximation of sampling zeros

Time Delay Systems

It is straightforward to include a time delay in a discrete-time model - in fact much easier than including a delay in a continuous-time model. To see this, consider a continuous-time system with transfer function $G(s)$. If the response of this system is delayed by the time T_d, the transfer function of the delayed system is $G'(s) = G(s)e^{-sT_d}$. Now suppose the system is sampled with a sampling time T, and assume for simplicity that the time delay is an integer multiple of the sampling time, i.e.

$$d = \frac{T_d}{T}$$

is an integer. If $H(z)$ is a pulse transfer function describing the system $G(s)$ in the sampling instants kT, then it follows from the properties of the z-transform that the transfer function of the delayed system is

$$H'(z) = H(z)z^{-d} \tag{7.34}$$

In contrast to continuous-time systems, the pulse transfer function of a time delay system is still a rational function in the complex variable z - adding a time delay of d sampling intervals to a discrete-time system is equivalent to adding d poles at the origin. For this reason it is also much simpler to implement a time delay in a digital controller than in an analog, continuous-time controller; this is important for example when using a Smith predictor.

Example 7.4

Consider a plant with transfer function

$$G(s) = \frac{K_0}{\tau s + 1}$$

Assume that the control input to this plant is formed by a zero-order hold with sampling time T. An exact discretization is then possible and yields the pulse transfer function

$$H(z) = \frac{K_1 z^{-1}}{1 - az^{-1}}$$

The pole at $s = -1/\tau$ is mapped into a pole at $z = a = e^{-T/\tau}$, and from $H(1) = G(0)$ we have $K_1 = K_0(1 - a)$. If a time delay of $T_d = 2T$ is added to the model, the continuous-time model becomes

$$G'(s) = \frac{K_0}{\tau s + 1}e^{-2Ts}$$

and for the discrete-time model we obtain

$$H'(z) = \frac{K_1 z^{-3}}{1 - az^{-1}} = \frac{K_1}{z^2(z - a)} = z^{-2}H(z)$$

7.4 Frequency Response of Sampled-Data Systems

The frequency response of a system is determined by its response to sinusoidal inputs over a range of frequencies. To study the frequency response of a sampled-data system, consider a continuous-time system with input signal $u(t) = \sin \omega t$. Let input and output be sampled at time instants $t = kT$, $k = 0, 1, 2, \ldots$, and assume that the discrete-time behaviour of the system is represented by $H(z)$. The sampled input is then

$$u(kT) = \sin \omega kT = \operatorname{Im} e^{j\omega kT}$$

and using (7.18) with the discrete-time impulse response $h(kT)$ we obtain

$$y(kT) = \sum_{l=0}^{\infty} h(lT) \operatorname{Im} e^{j\omega(k-l)T}$$

$$= \operatorname{Im} e^{j\omega kT} \sum_{l=0}^{\infty} h(lT) e^{-j\omega lT}$$

$$= \operatorname{Im} H(e^{j\omega T}) e^{j\omega kT}$$

$$= \operatorname{Im} |H(e^{j\omega T})| e^{j\phi} e^{j\omega kT}$$

where

$$\phi(e^{j\omega T}) = \arg H(e^{j\omega T})$$

is the phase angle of $H(e^{j\omega T})$. Taking the imaginary part in the last equation yields

$$y(kT) = |H(e^{j\omega T})| \sin(\omega kT + \phi(e^{j\omega T})) \tag{7.35}$$

The response to a sampled sinusoidal input is therefore again a sampled sinusoidal signal with the same frequency. Amplitude and phase shift of the response are determined by the magnitude and phase, respectively, of the pulse transfer function evaluated at the given frequency. The difference between a continuous-time and a discrete-time frequency response is that the continuous-time frequency response is obtained by evaluating $G(s)$ at $s = j\omega$, whereas the pulse transfer function $H(z)$ is evaluated at $z = e^{j\omega T}$. Moreover, the fact that

$$e^{j\omega T} = e^{j(\omega T + 2\pi)}$$

implies that the discrete-time frequency response is periodic. Travelling from $-j\pi/T$ to $+j\pi/T$ along the $j\omega$ axis in the s plane corresponds to travelling from $-j\pi$ to $+j\pi$ anticlockwise along the unit circle. Further excursions along the $j\omega$ axis merely retrace this same path. For a discrete-time frequency response it is therefore sufficient to consider the frequency range

$$-\frac{\pi}{T} \leq \omega \leq \frac{\pi}{T}$$

Aliasing

To explore the periodicity of the discrete-time frequency response further, we consider again the Laplace transform of a sampled signal. Recall the representation of the sampling process in (7.22) as modulation of a train of delta impulses

$$x^*(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t - kT) = \sum_{k=-\infty}^{\infty} x(kT)\delta(t - kT)$$

Note that because $x(t) = 0$ for $t < 0$, starting the summation at $-\infty$ does not change (7.22). The pulse train is a periodic signal and can be developed into a Fourier series

$$\sum_{k=-\infty}^{\infty} \delta(t - kT) = \sum_{l=-\infty}^{\infty} \alpha_l \, e^{jl\omega_s t}$$

where $\omega_s = 2\pi/T$ and

$$\alpha_l = \frac{1}{T} \int_{-T/2}^{+T/2} \delta(t)e^{-jl\omega_s t} \, dt = \frac{1}{T}$$

Substituting this in the above expression we obtain

$$x^*(t) = x(t) \sum_{l=-\infty}^{\infty} \frac{1}{T}e^{jl\omega_s t} = \frac{1}{T} \sum_{l=-\infty}^{\infty} x(t)e^{jl\omega_s t}$$

Taking the Laplace transform yields

$$\mathcal{L}[x^*(t)] = \frac{1}{T} \sum_{l=-\infty}^{\infty} \mathcal{L}\left[x(t)e^{jl\omega_s t}\right]$$

From

$$\mathcal{L}\left[x(t)e^{jl\omega_s t}\right] = \int_0^{\infty} x(t)e^{-(s-jl\omega_s)t} dt = X(s - jl\omega_s)$$

we obtain the Laplace transform of the sampled signal as

$$\mathcal{L}[x^*(t)] = X^*(s) = \frac{1}{T} \sum_{l=-\infty}^{\infty} X(s - jl\omega_s) \tag{7.36}$$

Assuming that $\int_0^{\infty} |x(t)|dt < \infty$, we have therefore

$$X^*(j\omega) = \frac{1}{T} \sum_{l=-\infty}^{\infty} X(j(\omega - l\omega_s))$$

This is the superposition on the $j\omega$-axis of the transform of the original continuous-time signal and its replications shifted by $l\omega_s$, $l = \pm 1, \pm 2, \ldots$ If the signal $x(t)$ contains no frequency components at $\omega > \omega_b$ and if $\omega_b < \frac{\omega_s}{2}$, then the replicated spectra do not overlap, and the original signal could theoretically be recovered by ideal low pass filtering. Fig. 7.12 illustrates this important result, which is in fact a statement of *Shannon's sampling theorem*: a signal $x(t)$ can only be exactly reconstructed from its sampled version $x^*(t)$ if $\omega_s > 2\omega_b$, i.e. the sampling frequency is more than twice the signal bandwidth. The maximum signal bandwidth

$$\omega_N = \frac{\omega_s}{2} = \frac{\pi}{T}$$

for which a reconstruction is possible, is called the *Nyquist frequency.* If the signal contains frequency components above the Nyquist frequency, the sampling process will bring high frequency components back into the range below the Nyquist frequency and thus distort the original signal. This effect is known as *aliasing.* A common measure called *anti-aliasing* is to low-pass filter a signal before sampling with a filter bandwidth below the Nyquist frequency but well above the signal bandwidth.

The frequency response of a sampled-data system including a zero-order hold is discussed in Exercises 7.14 and 7.15.

Choice of Sampling Frequency
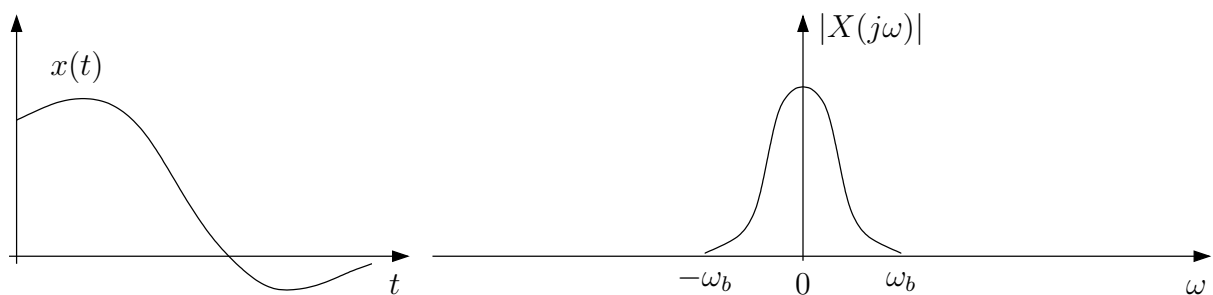
(a) Amplitude spectrum of continuous-time signal
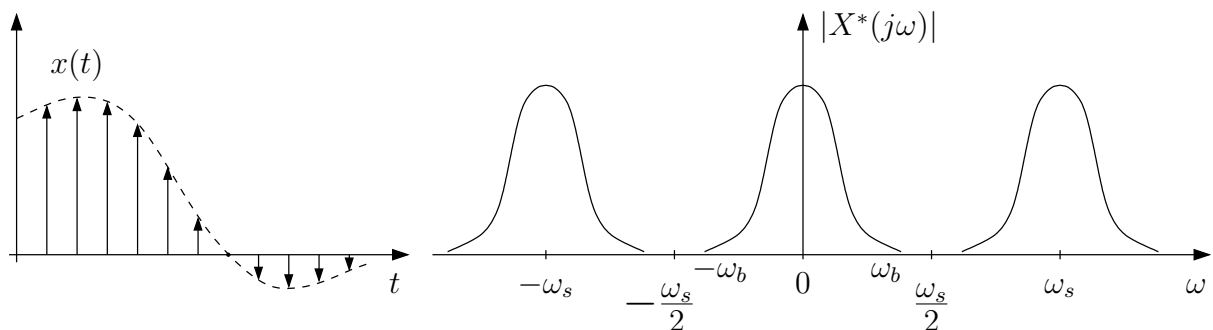
(b) Amplitude spectrum of sampled signal, $\omega_s > 2\omega_b$
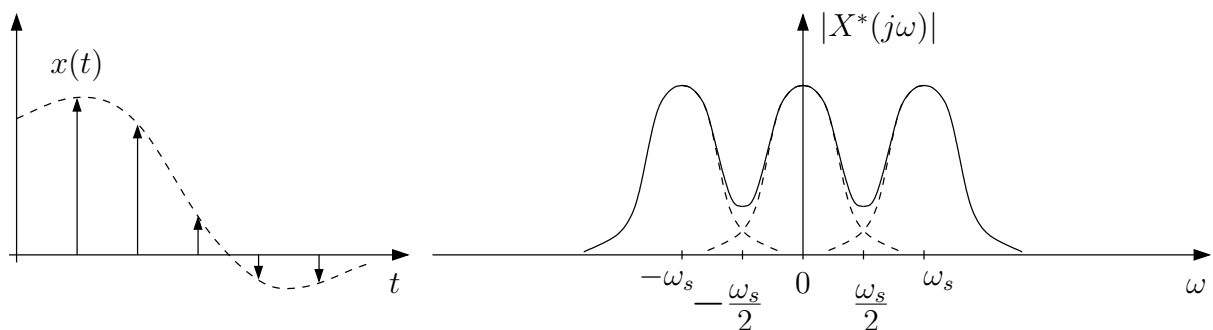
(c) Amplitude spectrum of sampled signal, $\omega_s < 2\omega_b$

Figure 7.12: Aliasing effect of sampling

The sampling theorem suggests a lower bound on the sampling frequency for a digital control system: it must be higher than twice the bandwidth of the system, for otherwise the information about the underlying continuous-time signals carried by the sampled signals will be distorted. In practice, the sampling frequency is chosen higher than this. When a digital controller is obtained via continuous-time design and Tustin approximation, the time delay of $T/2$ resulting from the approximation demands a sampling frequency of at least $20 \sim 30\omega_b$, as discussed earlier.

On the other hand, in the case of a direct discrete-time design with plant model (7.27), the discretization is exact if a zero-order hold is present at plant input, even if sampling is slow. However, it is important to realize that between sampling instants *the control system is effectively operating in open loop* - the feedback loop is closed only in the sampling instants. The effect of a disturbance occurring between two sampling instants will not be detected and acted against before the following sampling instant. If the sampling time is long, the effect of such a disturbance may be significant. A rule of thumb is to choose the sampling rate at least five to ten times larger than the system bandwidth.

Exercises

Problem 7.1

a) For the discrete-time system with transfer function

$$G(z) = \frac{2z^2 - 6z}{2z^2 - 6z + 4}$$

use polynomial division to determine the first 3 values of the impulse response of the system.

b) By partial fraction expansion, determine the response of the system

$$G(z) = \frac{1}{1 - 2z^{-1}}$$

to the input

$$u(k) = 2e^{-k}\sigma(k)$$

Problem 7.2

a) Show that

$$\sum_{k=0}^{\infty} x(k)z^{-k} - \sum_{k=0}^{\infty} x(k-1)z^{-k} = X(z) - z^{-1}X(z)$$

and, with $x(k) = 0$ when $k < 0$, that:

$$\lim_{z \to 1} \sum_{k=0}^{\infty} [x(k)z^{-k} - x(k-1)z^{-k}] = \lim_{k \to \infty} x(k)$$

b) Use the result from (a) to prove the Final Value Theorem for discrete-time signals

$$\lim_{k \to \infty} x(k) = \lim_{z \to 1}(z-1)X(z)$$

Problem 7.3

a) For a given sampling time T, sketch in the z-plane the regions corresponding to the following regions in the s-plane: $\operatorname{Re}(s)$ and $\operatorname{Im}(s)$ refer to the real and imaginary parts of a complex point s respectively.

i) $\operatorname{Re}(s) < -\frac{0.1}{T}$, $\operatorname{Re}(s) < -\frac{0.5}{T}$ and $\operatorname{Re}(s) < -\frac{1.0}{T}$

ii) $|\operatorname{Im}(s)| < 0.5\frac{\pi}{T}$, $|\operatorname{Im}(s)| < 1.0\frac{\pi}{T}$ and $|\operatorname{Im}(s)| < 1.5\frac{\pi}{T}$

b) For a second order system, sketch lines in the z-plane corresponding to constant damping ratios $\zeta = 0$, 0.5 and 1.

c) For a second order system, sketch lines in the z-plane corresponding to constant natural frequencies $\omega_n = \frac{\pi}{2T}$ and $\frac{\pi}{T}$.

Problem 7.4

For the discrete-time system governed by the difference equation

$$y(k) + a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) = b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3)$$

construct a state space model in controller canonical form, similar to the controller form for continuous system. Sketch the corresponding block diagram.

Problem 7.5

Consider a plant with continuous transfer function

$$G(s) = \frac{b}{s+a}, \qquad a > 0$$

This plant is to be controlled by a controller K as shown in figure 7.13.
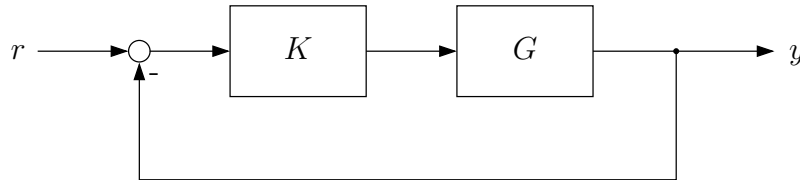
Figure 7.13: Closed-loop system

a) At first, $G(s)$ is to be controlled using a continuous-time proportional feedback controller $K(s) = K_p$. Sketch the continuous-time root locus for $a = 2$ and $b = 1$.

b) Calculate an exact (zero-order-hold) discretization $G_d(z)$ of $G(s)$ as a function of a and b.

c) The plant is now to be controlled with the discrete-time proportional feedback controller $K_d(z) = K_{pd}$ and sampling time $T = 0.2$. Using the discrete-time plant transfer function $G_d(z)$ from (b), sketch the discrete-time root locus with $a = 2$ and $b = 1$. At what value of K_{pd} does the system become unstable?

d) Using the continuous-time and discrete-time root loci, compare the closed-loop behaviour as $K_p \to \infty$ and $K_{pd} \to \infty$ for the system with continuous-time and discrete-time controllers, respectively. Describe qualitatively why this behaviour occurs.

e) Create a Simulink model to compare the following closed-loop responses ($a = 2$, $b = 1$)

   i) with continuous-time plant and continuous-time controller where $K_p = 5$

   ii) with the continuous-time plant and discrete-time controller where $K_{pd} = 5$

   iii) with discretized plant from (b) and the discrete-time controller where $K_{pd} = 5$.

   Hint: Generate the discrete-time controller using the command `c2d` with the option 'tustin'.

Problem 7.6

Consider a continuous-time system with transfer function

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

This system is to be controlled as shown in Figure 7.13.

a) Calculate the exact (zero-order-hold) discretization $G_d(z)$ of $G(s)$.

b) Sketch the root loci for the continuous-time plant and for the discrete-time plant, each with proportional controllers. Can the plant be stabilized using either a continuous-time or discrete-time proportional controller?

c) $G_d(z)$ is now to be controlled with a discrete-time PD-controller

$$K_d(z) = K_{pd}(1 + T_d(1 - z^{-1}))$$

   Show that this can be written as

$$K_d(z) = K_{pd2}\frac{z - \alpha}{z}$$

   where α is a function of T_d and K_{pd2} is $K_{pd}/(1 - \alpha)$.

d) Use *rltool* to find K_{pd2} and α so that a settling time < 2.0 and a damping ratio $\zeta \approx 0.7$ are achieved. Use a sampling time $T = 0.2$ s.

   Hint: You can use the design constraints settling time and damping ratio in rltool.

e)   Use *rltool* again to calculate K_p and α, so that the settling time < 4.5 and $\zeta \approx 0.707$. This time use a sampling period of $T = 0.5$ s. Is it possible to achieve a settling time < 2.0 with a discrete-time PD controller with $T = 0.5$ s?

Problem 7.7

a)   Use Matlab to design a state feedback dead-beat controller K_{db} for the discretized double integrator $G_d(z)$ from Problem 7.6. You can either use the function `acker` if available in your Matlab distribution, or `place` where you set the desired pole locations to values close to 0 (`place` requires distinct pole locations, e.g. `[0 1e-5]`).

b)   Use Matlab and Simulink to construct the closed loop system consisting of plant $G(s)$, the discrete state feedback controller K_{db}, a sampler and a zero order hold unit, and a disturbance at input of $G(s)$. Plot the step response of the output of $G(s)$ to this disturbance.

Problem 7.8

This problem refers to the state space model of the turbogenerator and the continuous-time controller and observer designed in Problem 6.15. You can use the script file `cs5_tgen_LQG.m` to generate the controller..

a)   Use Matlab to generate the Tustin-approximation Ki of the controller with sampling times $T_s = 0.02$ and $T_s = 0.5$.

Hint: You need to determine the Tustin approximation of the complete controller, including the state feedback controller, Kalman filter and integrators. One (of many) ways to do this is to use the Matlab function `linmod` *with a Simulink model of the complete controller.*

b)   Simulate the response of the closed-loop system with both discretized controllers (for $T_s = 0.02$, 0.5) to a disturbance $d_1(t) = \sigma(t)$. Compare your response with the one obtained in Problem 6.15.

c)   Discretise the augmented plant (that is, including the integrators) for $T_s = 0.02$ and $T_s = 0.5$. For each of these discretized systems design an LQG controller with the same performance weights as in 6.15.

Hint: Use the command `dlqr` *to design a discrete LQR controller and Kalman filter.*

d)   Compare the performance achieved with discrete time controllers of part a) and c) for disturbance $d_1(t) = \sigma(t)$.

e)   Simulate the response of the controller from exercise 6.15 and the disturbance $d_1 =$ $\sigma(z)$, but with a delay of 0.25 s between the controller output and the plant input. Compare your answer with the response obtained in part b).

Hint: Use either Pad'e approximation, if you are performing the simulation in Matlab, or transport delay block if you are using Simulink.

Problem 7.9

Consider a system with impulse response $g(k)$ and transfer function

$$G(z) = \sum_{k=0}^{\infty} g(k)z^{-k}$$

Show that the response $y(k)$ to an input signal $u(k)$ with z-transform

$$U(z) = \sum_{k=0}^{\infty} u(k)z^{-k}$$

is given by

$$y_k = \sum_{l=0}^{k} g(l)u(k-l)$$

Problem 7.10

The discrete-time system $G(z)$ is described by the following set of matrices:

$$\Phi = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0.2 & 2 \\ 0 & 0 & 0.1 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}, \qquad D = 0$$

a)   Compute the impulse response of the system for the first 4 sampling instants.

b)   Compute the system response to an input signal $u = \{5,\ 0,\ -1,\ 2\}$. Consider zero initial conditions.

Problem 7.11

The locations of zeros of a discrete-time system produced by exact (zero-order-hold) discretization of a continuous-time system can be approximated by the model shown in Figure 7.11.

a)   What zeros would you therefore expect, when the continuous-time systems

(i)

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

(ii)

$$G(s) = \frac{1}{s^3 + s^2 + s}$$

are sampled at high sampling rates with a zero order hold?

Hint: Calculate a state space model of the exact discretization of $1/s^{n-m}$

b)   Check the results from (a) with the Matlab function `c2d`. For which sampling times is the sampled system (ii) minimum phase? For which sampling times is it non-minimum phase?

Problem 7.12

Practical representations of digital signals have finite word-length. Explain why this can lead to problems if high sampling rates are used.

Hint: What happens to the locations of the poles and to the system matrix of a discrete-time state space model if the sampling time approaches zero?

Problem 7.13

Consider the system

$$G(s) = \frac{0.2}{s + 0.2}$$

a)   What is the time constant (τ), bandwidth and static gain of $G(s)$.

b)   Compute the frequency ω_3 of a sinusoidal signal $u(t) = \sin(\omega_3 t)$, which when applied as an to $G(s)$ it will generate an output $y(t)$ with amplitude 3 dB smaller than the steady-state gain.

c)   The system is to be controlled in closed-loop as in Fig. 7.10. Assume that the closed-loop bandwidth is the same as the bandwidth of the plant. What sampling times T_s are acceptable for the task?

Problem 7.14

Figure 7.14 shows a delta sampler with a zero-order hold (*zoh*) unit.

a)   Find an expression for $y(t)$ as a function of $u(t)$ at the sampling instants.

Hint: Use positive and negative step functions to represent the signal $y(t)$.
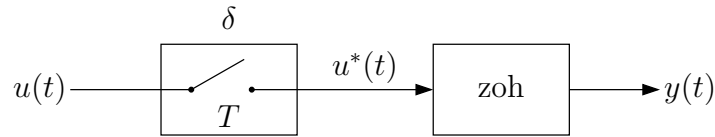
Figure 7.14: Zero order hold with delta sampler

b)   Calculate the Laplace transform $Y(s)$ of the output signal as a function of $U(s)$.

c)   Show that the transfer function of the *zoh* unit is

$$G_{\mathrm{zoh}}(s) = \frac{Y(s)}{U^*(s)} = \frac{1}{s}(1 - e^{-Ts})$$

d)   Show using (c) that the frequency response of the *zoh* unit is

$$G_{\mathrm{zoh}}(j\omega) = T\frac{\sin\left(\omega T/2\right)}{\omega T/2}e^{-\frac{1}{2}Tj\omega}$$

Sketch the bode diagram of the sample and hold unit for $T = 1\mathrm{s}$.

Problem 7.15

The frequency response of the system in Figure 7.15 is to be investigated, where

$$G(s) = \frac{1}{s+1}$$
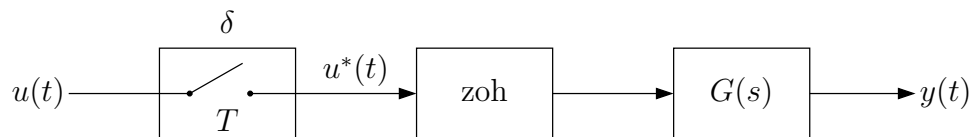
Figure 7.15: Sampler and system $G(s)$

a)   Calculate $u^*(t)$ when $u(t) = \sin \omega t$.

   Hint: Represent the sequence of delta impulses associated with the delta-sampler by a real Fourier series

$$\sum_{k=-\infty}^{\infty} \delta(t - kT) = \frac{1}{T}\left(1 + 2\sum_{l=1}^{\infty} \cos l\omega_s t\right)$$

b)   Simulate the response to the input signal in (a) for the frequencies $\omega = 2$ rad/s, $\omega = 20$ rad/s and $\omega = 60$ rad/s when the sampling period is $T = 0.05$ s. Interpret the simulation with the help of the results in (a) and Problem 7.14.d. For the simulation, use the Simulink model `cs6_simzoh.mdl`

Problem 7.16   *Mini Segway exercise.*

The gyroscope sensor, which is used to measure the angular rate $\dot{\alpha}$, has a low pass filter with cut-off frequency of 20Hz (DLPF mode=4).

a)   Open the Simulink file `Experiment_LQR.slx`, go into the gyroscope mask and set DLPF mode=0. This sets the low pass filter's cut-off frequency to 250Hz. What should we expect when experimenting using this setting?

b)   Experiment **with caution** using this setting to track the same reference trajectory presented in the LQR tracking problem (Problem 5.11). Compare the results of the experiments with the different gyroscope filter settings (0 and 4), is the tracking performance comparable? How could we solve this problem without changing the gyroscope filter setting?

Problem 7.17   *Mini Segway exercise.*

Read and understand the MATLAB script `lqrd_comparison.m` and Simulink files `Task_5_Simulation_LQR_Contin.slx` and `Task_5_Simulation_LQR_Discrete.slx`. The MATLAB files simulate and plot the closed-loop response of the continuous time and discrete time linear model.

a)   Sample the continuous time controller implemented in Problem 5.10 and compare between the discrete time and continuous time response. *(Hint: use the MATLAB command `lqrd`.)*

   i)   Check the documentation of the MATLAB commands `dlqr` and `lqrd`. What is the difference between these commands?

   ii)   Is it possible to design a discrete time observer by dualism using `lqrd`?

Problem 7.18   *Mini Segway exercise.*

A discrete time observer is to be used to output the filtered states by combining the state estimates from the model and the measured states from the sensors. The gyroscope's

low pass filter setting should be set to zero(DLPF mode=0) since now the states are filtered using an observer. Read and understand the MATLAB script `Task_6_Simulation _LQG_Design.m` and Simulink file `Task_6_Simulation_LQG_Discrete.slx`. The MATLAB files simulate and plot the closed-loop response of the linear model with observer state feedback.

a)   Discretise the continuous model in the MATLAB file using zero order hold.

b)   Implement the discrete time observer based state feedback in the Simulink file, tune the observer and controller gains using `dlqr` command. Compare between the simulation with and without the observer for tracking a sinusoidal position reference. *(Hint: make sure that the observer poles are at least 3 times faster than the controller poles, except for the fast controller pole).*

Problem 7.19   *Mini Segway exercise.*

Run the MATLAB script `Experiment_parameters.m` and open the experiment simulink file `Experiment_LQG.slx`.

a)   Run the experiment and extract the states from the simulink experimental model *(Hint: export the data as a structure with time as a vector* [states;Control input;reference] *and name the data* "expOut" *so that you can use implemented code)* and compare between the simulation and the experiment for sine wave tracking with:

  – DLPF mode=0, with observer: "expOut1".
  – DLPF mode=4, without observer: "expOut2".

Chapter 8

System Identification

All methods for designing and analyzing control systems that have been introduced in this course are *model based*, i.e. it is assumed that a dynamic plant model in the form of a transfer function or state space realization is available. In practice, obtaining such a model can take up a significant part of the time required for solving a given control problem. In Exercises 2.10, 2.1 or 3.6, state space models of physical systems were derived from the knowledge of the underlying physical laws. However, such physical modelling can become difficult if the plant dynamics are complex and not well understood. An alternative is to obtain a plant model experimentally by measuring the response of the plant to suitable test signals; this approach is known as *system identification*. Because no physical insight into the plant behaviour is utilized, this method is also referred to as *black box modelling*. This chapter gives an introduction to the basic concepts of system identification.

Transfer functions and state space models are called *parametric models*, because the complete information about the dynamic behaviour of the system is contained in a fixed number of parameters - e.g. the coefficients of the transfer function polynomials or the entries of the matrices of a state space model. *Nonparametric models* on the other hand are representations of plant dynamics that cannot be expressed by a finite number of parameters, such as the shape of the frequency response or the impulse response of the system. In this chapter we will consider the experimental identification of parametric plant models. Because the input-output data used for system identification are usually sampled-data sequences, the identified models are discrete-time models. After introducing the concept of a linear regression model in Section 8.1, we will discuss the identification of transfer function models for SISO systems in Section 8.2. Sections 8.3 and 8.4 then present two recently developed techniques for identifying SISO and MIMO state space models.

8.1 Least Squares Estimation

Assume that we are observing a process - characterized by a quantity $y(t)$ - at time instants $t = 0, T, 2T, \ldots$ where T is a given sampling time. Using the shorthand notation introduced earlier and suppressing the sampling time, this observation yields a data sequence $y(k)$, $k = 0, 1, 2, \ldots$ Assume further that the process variable $y(k)$ at time instant kT depends linearly on the values of other variables $m_1(k)$, $m_2(k), \ldots,$ $m_n(k)$ which are known and available at the same time. A linear process model is then

$$y(k) = m_1(k)p_1 + m_2(k)p_2 + \ldots + m_n(k)p_n + e(k) \tag{8.1}$$

where the dependence on the measured variables is determined by the parameters p_1, $p_2, \ldots,$ p_n. The term

$$e(k) = y(k) - m_1(k)p_1 - m_2(k)p_2 - \ldots - m_n(k)p_n$$

is added to allow for modelling errors, e.g. measurement errors or inaccurate knowledge of values of the parameters p_i. Further modelling errors could arise if the true process depends on the measured variables in a nonlinear way, or if it depends on additional variables that are not included in the above model.

The model (8.1) can be written in a more compact form as a linear regression model

$$y(k) = [m_1(k) \ldots m_n(k)] \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} + e(k) = m^T(k)p + e(k) \tag{8.2}$$

where two column vectors - the vector of regression variables $m(k)$ and the parameter vector p - have been introduced. Given a set of measured data $y(l)$ and $m(l)$, $l = 0, 1, \ldots, k$, we can now pose the *least squares estimation problem*: find the parameter vector p that best fits the data, in the sense that the sum of the squared errors

$$V(p) = \sum_{l=0}^{k} e^2(l) \tag{8.3}$$

is minimized.

Example 8.1

Suppose a transfer function model of an unknown system is to be identified which is thought to be governed by a first order linear difference equation. The task is then to find parameters a and b of the transfer function

$$G(z) = \frac{bz^{-1}}{1 + az^{-1}}$$

that lead to a good fit between measured data and output data predicted by the transfer function model. The situation is illustrated in Fig. 8.1: the same input sequence $u(k)$

is applied to the unknown system and to the model $G(z)$, yielding the actual system response $y(k)$ and the response $\hat{y}(k)$ predicted by the model, respectively. The modelling error is

$$e(k) = y(k) - \hat{y}(k)$$

This problem can be expressed in the form of a linear regression model by writing the difference equation of the system as

$$
\begin{aligned}
y(k) &= -ay(k-1) + bu(k-1) + e(k) \\
&= [-y(k-1) \ \ u(k-1)] \begin{bmatrix} a \\ b \end{bmatrix} + e(k) \\
&= m^T(k)p + e(k) = \hat{y}(k) + e(k)
\end{aligned}
$$
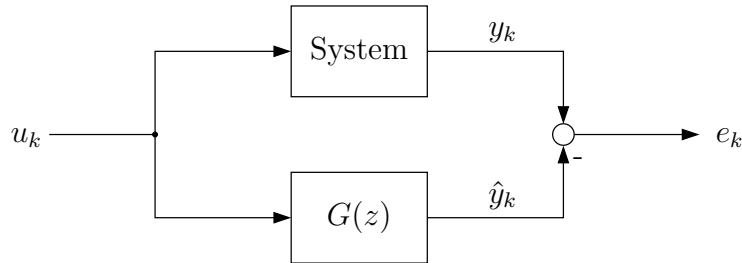
Figure 8.1: Identification of a transfer function model

Solving the Least Squares Estimation Problem

If a data set $\{y(l), u(l)\}$, $l = 0, 1, 2, \ldots, k$ is available, we can arrange it in the form

$$Y = Mp + E \tag{8.4}$$

where we define

$$
Y = \begin{bmatrix} y(0) \\ \vdots \\ y(k) \end{bmatrix}, \quad
M = \begin{bmatrix} m^T(0) \\ \vdots \\ m^T(k) \end{bmatrix}, \quad
E = \begin{bmatrix} e(0) \\ \vdots \\ e(k) \end{bmatrix}
$$

Similarly, we introduce the vector

$$\hat{Y} = Mp$$

of predicted outputs. Assuming initially that the unknown system can indeed be accurately described by a linear model of the assumed order, and that there are no measurement errors, the modelling error will be zero if the parameter vector p takes its "true" value: in that case we have $\hat{Y} = Y$, or

$$Mp = Y \tag{8.5}$$

If we have more measurements available than model parameters, i.e. if $k > n$, this is an overdetermined system of equations. Multiplying from left by M^T yields

$$M^T Mp = M^T Y \tag{8.6}$$

This equation is called the *normal equation* associated with the given estimation problem. If M has full column rank, the matrix $M^T M \in \mathbb{R}^{n \times n}$ is non-singular, and we can compute

$$p = (M^T M)^{-1} M^T Y \tag{8.7}$$

However, the parameter vector p obtained from (8.7) will satisfy (8.5) only if the system is indeed exactly governed by a linear difference equation of the assumed order, and if there are no measurement errors. In real life problems, neither condition will be met, so that p will not satisfy (8.5) but only (8.4) with $E \neq 0$. The best we can then achieve is to find the parameter vector p that is associated with the "smallest modelling error" - in other words the closest approximation we can get with this model in the presence of measurement errors. The following result is derived in Exercise 8.1.

Theorem 8.1

The sum of square errors $V(p)$ (8.3) is minimized if the parameter vector satisfies the normal equation (8.6).

If the matrix $M^T M$ is nonsingular, the minimizing parameter vector is given by (8.7).

Geometric Interpretation

A geometric interpretation of the normal equation goes as follows. Rewrite (8.4) as

$$E = Y - Mp$$

or

$$\begin{bmatrix} e(0) \\ \vdots \\ e(k) \end{bmatrix} = \begin{bmatrix} y(0) \\ \vdots \\ y(k) \end{bmatrix} - \begin{bmatrix} m_1(0) \\ \vdots \\ m_1(k) \end{bmatrix} p_1 - \ldots - \begin{bmatrix} m_n(0) \\ \vdots \\ m_n(k) \end{bmatrix} p_n$$

Introducing the column vectors

$$\varphi_i = \begin{bmatrix} m_i(0) \\ \vdots \\ m_i(k) \end{bmatrix}, \quad i = 1, \ldots, n$$

we thus have

$$E = Y - \varphi_1 p_1 - \ldots - \varphi_n p_n$$

If the true system can be accurately described by the assumed linear model and if there are no measurement errors, then Y would be in the space spanned by the vectors $\varphi_1 \ldots \varphi_n$. In real life, unmodelled features of the system and measurement errors will in general lead to a vector Y that is outside the data space. The estimation problem can then be interpreted as searching for the linear combination of the vectors $\varphi_1 \ldots \varphi_n$ that comes
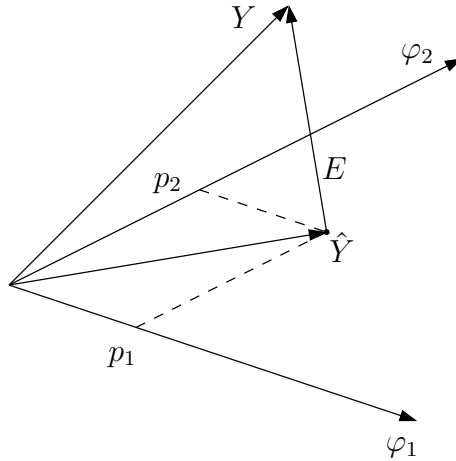
Figure 8.2: Geometric interpretation of the least squares estimation problem

closest to the vector Y, i.e. that minimizes the squared error $E^T E$. This is illustrated in Fig. 8.2 for the special case $n = 2$: what we are looking for is the *projection* $\hat{Y}$ of the vector Y onto the space (a plane if $n = 2$) spanned by the measurement vectors φ_i, and $\hat{Y}$ is the vector closest to Y if the error E is orthogonal to this space (plane). But E is orthogonal to this space if it is orthogonal to each of the measurement vectors, i.e. if it satisfies

$$\varphi_i^T E = 0, \qquad i = 1, \dots, n$$

This can be written in a more compact form as $M^T E = 0$ or

$$M^T(Y - Mp) = 0$$

which is just the normal equation (8.6).

8.2   Estimation of Transfer Function Models

We will now apply the idea of least squares estimation to identify a transfer function model of a system from measured input and output data. Thus, assume that data sequences $\{u(0), \dots, u(k)\}$ and $\{y(0), \dots, y(k)\}$ are available and that we want to find the pulse transfer function that gives the best fit between input and output data. In order to apply the above estimation technique, we need to fix the number of parameters, in this case the order of numerator and denominator polynomial of the estimated transfer function. We will initially assume for simplicity that the system to be identified can be modelled by the difference equation

$$\hat{y}(k) = b_1 u(k-1) + b_2 u(k-2) + \dots + b_n u(k-n) \tag{8.8}$$

which means there is no autoregressive component in the output (the a_i's are assumed to be zero) - we will later remove this assumption. The difference equation can be written

in regressor form as

$$\hat{y}(k) = [u(k-1)\ u(k-2)\ \ldots\ u(k-n)] \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = m^T(k)p$$

The measurement matrix M takes the form

$$M = \begin{bmatrix} u(n-1) & u(n-2) & \ldots & u(0) \\ u(n) & u(n-1) & \ldots & u(1) \\ \vdots & & & \vdots \\ u(k) & u(k-1) & \ldots & u(k-n+1) \end{bmatrix}$$

where we take $m^T(n)$ as the first row, and we have

$$M^T M = \begin{bmatrix} \sum_{i=n-1}^{k} u_i^2 & \sum_{i=n-1}^{k} u_i u_{i-1} & \ldots & \sum_{i=n-1}^{k} u_i u_{i-n+1} \\ \sum_{i=n-1}^{k} u_{i-1} u_i & \sum_{i=n-1}^{k} u_{i-1}^2 & \ldots & \sum_{i=n-1}^{k} u_{i-1} u_{i-n+1} \\ \vdots & & \ddots & \vdots \\ \sum_{i=n-1}^{k} u_{i-n+1} u_i & \sum_{i=n-1}^{k} u_{i-n+1} u_{i-1} & \ldots & \sum_{i=n-1}^{k} u_{i-n+1}^2 \end{bmatrix}$$

where we used the shorthand notation u_i for $u(i)$. For a solution (8.7) to the estimation problem to exist, this $n \times n$ matrix needs to be invertible. This requirement places a condition on the input sequence $\{u(0), \ldots, u(k)\}$. For example, it is obvious that with a constant input sequence $\{1, \ldots, 1\}$ the rank of $M^T M$ will be one and a solution for a model with more than one estimated parameter will in general not exist. To explore this further, we will use the *empirical autocorrelation* of the data sequence $\{u(k)\}$, defined as

$$c_{uu}(l) = \lim_{k \to \infty} \frac{1}{k} \sum_{i=1}^{k} u(i)u(i-l)$$

Introducing the matrix

$$C_{uu}(n) = \begin{bmatrix} c_{uu}(0) & c_{uu}(1) & \ldots & c_{uu}(n-1) \\ c_{uu}(1) & c_{uu}(0) & \ldots & c_{uu}(n-2) \\ \vdots & & \ddots & \vdots \\ c_{uu}(n-1) & c_{uu}(n-2) & \ldots & c_{uu}(0) \end{bmatrix}$$

we find that

$$\lim_{k \to \infty} \frac{1}{k} M^T M = C_{uu}(n)$$

Thus, for sufficiently long data sequences (when the end effects can be neglected and we can consider all sums as taken from 1 to k) we may interpret the matrix $M^T M$ as a scaled version of the empirical covariance $C_{uu}(n)$ of the input signal.

Persistent Excitation

The condition that the matrix $M^T M$ must have full rank is called an *excitation condition* - the input signal must be *sufficiently rich* to excite all dynamic modes of the system. We have seen that for long data sequences we can consider the matrix $C_{uu}(n)$ instead of $M^T M$. The following definition provides a measure for the richness of an input signal.

Definition 8.1

A signal $u(k)$ is said to be persistently exciting of order n if the matrix $C_{uu}(n)$ is positive definite.

The next result is useful for checking whether a signal is persistently exciting of a given order.

Theorem 8.2

A signal $u(k)$ is persistently exciting of order n if and only if

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} \Big(a(z)u(l) \Big)^2 > 0 \quad \forall a(z) : \ \deg a(z) \le n - 1 \tag{8.9}$$

Here $a(z)$ is a polynomial in the *forward shift operator z*, i.e.

$$a(z) = a_0 + a_1 z + a_2 z^2 + \ldots + a_{n-1} z^{n-1}$$

Recall that the forward shift operator is defined by

$$zu(l) = u(l+1)$$

With these definitions, multiplying a signal $u(k)$ by $a(z)$ yields

$$a(z)u(l) = a_0 u(l) + a_1 u(l+1) + a_2 u(l+2) + \ldots + a_{n-1} u(l+n-1)$$

It is straightforward to prove the above Theorem by observing that the sum on the left hand side of the inequality can be rewritten as

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} (a(z)u(l))^2 = a^T C_{uu}(n) a$$

where $a^T = [a_{n-1} \ a_{n-2} \ \ldots \ a_0]$

Theorem 8.2 can be used to determine an upper bound on the order of the persistent excitation of a given signal: if one can find a polynomial $a(z)$ of order n that does not satisfy (8.9), then the signal is *not* persistently exciting (PE) of order n. This idea can be used to show that

- an impulse $\delta(k)$ is PE of order 0

- a step function $\sigma(k)$ is PE of order 1

- a sine wave is PE of order 2

- white noise is PE of any order

For step functions, sine waves and white noise, this is discussed in Exercises 8.2, 8.3 and 8.4. White noise is therefore commonly used as test input when a linear model is to be identified experimentally.

ARX Models

The model (8.8) was based on the assumption that the present output does not depend on past outputs, i.e. there is no autoregressive component in the output. We now remove this assumption and consider the model

$$\hat{y}(k) = -a_1 y(k-1) - \ldots - a_n y(k-n) + b_1 u(k-1) + \ldots b_n u(k-n)$$

$$= [-y(k-1) \ \ldots \ -y(k-n) \ u(k-1) \ \ldots \ u(k-n)] \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$

which corresponds to the difference equation model introduced for discrete-time systems in the previous chapter. Such a model is called an *ARX model*, where ARX stands for AutoRegressive with eXogenous input. The results discussed in this section can be extended to ARX models by using the empirical cross-covariance function

$$c_{uy}(l) = \lim_{k \to \infty} \frac{1}{k} \sum_{i=1}^{k} u(i) y(i-l)$$

It follows then that

$$\lim_{k \to \infty} \frac{1}{k} M^T M = \begin{bmatrix} C_{yy} & -C_{uy} \\ -C_{uy} & C_{uu} \end{bmatrix}$$

where the matrices C_{yy} and C_{uy} are defined in the same way as C_{uu}.

The least-squares estimation of an ARX model from measured data is illustrated in Exercise 8.5.

8.3   Subspace Identification of State Space Models

The discussion in the previous section was limited to SISO transfer function models. The approach presented there can be extended to cover MIMO models in ARX form, but working with multivariable systems is usually more convenient in a state space framework. In this and the following section we present a recently developed approach to estimating SISO and MIMO state space models.

To introduce the idea, we begin with a SISO state space model

$$x(k + 1) = \Phi x(k) + \Gamma u(k), \quad x(0) = 0$$
$$y(k) = cx(k) + du(k)$$

Note that we use the same symbol Γ for discrete-time SISO and MIMO models. In a SISO model it represents a column vector. Now assume that $x(0) = 0$, and consider the impulse response of the above model, i.e. the response to the input $u(k) = \delta(k)$. Observing that for $k > 0$ we have $x(k) = \Phi^{k-1}\Gamma$, we find that the impulse response $g(k)$ is given by

$$g(k) = \begin{cases} 0, & k < 0 \\ d, & k = 0 \\ c\Phi^{k-1}\Gamma, & k > 0 \end{cases}$$

The values $\{d, \ c\Gamma, \ c\Phi\Gamma, \ldots\}$ of the impulse response sequence are called the *Markov parameters* of the system.

Turning now to multivariable systems, we first need to clarify what we mean by the impulse response of a MIMO model

$$x(k + 1) = \Phi x(k) + \Gamma u(k), \quad x(0) = 0$$
$$y(k) = Cx(k) + Du(k) \tag{8.10}$$

We can apply a unit impulse to one input channel at a time and observe the resulting response at each output channel

$$u_{\delta i}(k) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \delta(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \rightarrow \quad y_{\delta i}(k) = \begin{bmatrix} g_{1i}(k) \\ \vdots \\ g_{li}(k) \end{bmatrix}$$

Here $\delta(k)$ is placed in the ith entry of the input vector, while all other inputs are zero. An entry $g_{ji}(k)$ in the output vector represents the response at output channel j to a unit

impulse applied at input channel i. The complete information about the impulse responses from each input to each output can then be represented by the impulse response matrix

$$g(k) = \begin{bmatrix} g_{11}(k) & \dots & g_{1m}(k) \\ \vdots & & \vdots \\ g_{l1}(k) & \dots & g_{lm}(k) \end{bmatrix}$$

Introducing the notation

$$\Gamma = [\Gamma_1 \ \Gamma_2 \ \dots \ \Gamma_m], \qquad D = [d_1 \ d_2 \ \dots \ d_m]$$

where Γ_i and d_i denote the ith column of the matrices Γ and D, respectively, we find that with input $u_{\delta i}(k)$ we have $x_{\delta i}(k) = \Phi^{k-1}\Gamma_i$ for $k > 0$, and at $k = 0$ we have $y_{\delta i}(k) = d_i$. Combining the responses to impulses at all input channels, we obtain

$$g(k) = [y_{\delta 1}(k) \ \dots \ y_{\delta m}(k)] = \begin{cases} 0, & k < 0 \\ D, & k = 0 \\ C\Phi^{k-1}\Gamma, & k > 0 \end{cases} \qquad (8.11)$$

The impulse response describes input-output properties of a system, and we would expect it to be independent of a particular coordinate basis that has been chosen for a given state space model. This seems to contradict the fact that the impulse response in (8.11) is given in terms of the matrices (Φ, Γ, C) of a state space model, which clearly depend on the choice of coordinate basis. However, it is easily checked that applying a similarity transformation T - which yields a realization $(T^{-1}\Phi T, T^{-1}\Gamma, CT)$ - will not change the impulse response.

Constructing a Model from the Impulse Response

Assume that measured impulse response data of a system are available and have been arranged in the form of a matrix

$$H_k = \begin{bmatrix} g(1) & g(2) & g(3) & \dots & g(k) \\ g(2) & g(3) & \dots & & g(k+1) \\ g(3) & \vdots & & & \\ \vdots & & & & \vdots \\ g(k) & g(k+1) & & \dots & g(2k-1) \end{bmatrix}$$

A matrix with this structure is called a *Hankel matrix* if the $g(l)$ are scalar, and a *block-Hankel matrix* if the $g(l)$ are matrices. Using (8.11) in the above we obtain

$$H_k = \begin{bmatrix} C\Gamma & C\Phi\Gamma & C\Phi^2\Gamma & \dots & C\Phi^{k-1}\Gamma \\ C\Phi\Gamma & C\Phi^2\Gamma & \dots & & C\Phi^k\Gamma \\ C\Phi^2\Gamma & \vdots & & & \\ \vdots & & & & \vdots \\ C\Phi^{k-1}\Gamma & C\Phi^k\Gamma & & \dots & C\Phi^{2k-2}\Gamma \end{bmatrix} \qquad (8.12)$$

Assume the number of samples is sufficiently large so that $k > n$, where n is the order of the state space model. Note that at this point we know nothing about the system apart from its impulse response. In particular, we do not know the order n of the system. Important in this context is the rank of the matrix H_k. To investigate this, we first observe that we can factor H_k as

$$H_k = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{k-1} \end{bmatrix} [\Gamma \ \ \Phi\Gamma \ \ldots \Phi^{k-1}\Gamma] = \mathcal{O}_k \mathcal{C}_k$$

Here $\mathcal{O}_k$ and $\mathcal{C}_k$ are the *extended observability* and *controllability matrices*, respectively, of the model (8.10), where the term "extended" is added because the number of samples k is greater than the expected order n of the system. Assuming that we are interested in estimating a model (8.10) that represents a minimal realization of a system, i.e. if (Φ, Γ) is controllable and (C, Φ) is observable, then we have

$$\text{rank } \mathcal{O}_k = \text{rank } \mathcal{C}_k = n$$

which implies

$$\text{rank } H_k = n \tag{8.13}$$

Thus, we can obtain the order from the measured data by computing the rank of H_k.

The Ideal Case

Assume that for a given system with m inputs and l outputs the measured discrete-time impulse response $g(k)$ is available, and that we want to identify a discrete-time state space model (Φ, Γ, C). For systems where $D \neq 0$, the feedthrough matrix is given by $g(0)$. Initially we do not know the dynamic order of this model, but we assume that we have a sufficient number of samples of the impulse response, so that we can form the $mk \times lk$ matrix H_k for a value of k that is larger than the expected order of the model. If the impulse response data were indeed generated by a linear state space model of the form (8.10) with n state variables, and if no measurement errors are present in the data, then the order n of the model can be easily determined by checking the rank of H_k. Knowing n we can then factor H_k as

$$H_k = ML, \quad M \in \mathbb{R}^{lk \times n}, \ L \in \mathbb{R}^{n \times mk}$$

such that

$$\text{rank } M = \text{rank } L = n$$

This can be done using singular value decomposition as explained below. Note that this factorization is not unique. Finally, we define the matrices M and L to be the *extended observability and controllability matrices*

$$\mathcal{O}_k = M, \quad \mathcal{C}_k = L$$

The first l rows of M therefore represent the measurement matrix C, and the first m columns of L form the input matrix Γ. To find the state matrix Φ, define

$$\bar{\mathcal{O}}_k = \begin{bmatrix} C\Phi \\ \vdots \\ C\Phi^k \end{bmatrix} = \mathcal{O}_k \Phi$$

Note that we can generate $\bar{\mathcal{O}}_k$ from measured data by factorizing the larger Hankel matrix H_{k+1} and removing the first l rows from $\mathcal{O}_{k+1}$. Multiplying the above from the left by $\mathcal{O}_k^T$ we obtain

$$\mathcal{O}_k^T \mathcal{O}_k \Phi = \mathcal{O}_k^T \bar{\mathcal{O}}_k$$

Since $\mathcal{O}_k$ has full row rank, we can compute Φ from

$$\Phi = (\mathcal{O}_k^T \mathcal{O}_k)^{-1} \mathcal{O}_k^T \bar{\mathcal{O}}_k$$

The above describes a procedure for constructing the matrices Φ, Γ and C from measured impulse response data. At this point a question arises: we know that a state space model of a given system is not unique but depends on the coordinate basis chosen for the state space. One could therefore ask where this choice was made in the above construction. The answer is that the factorization $H_k = ML$ is not unique, in fact if M and L are factors of rank n and if T is an arbitrary nonsingular $n \times n$ matrix, then it is easy to see that MT and $T^{-1}L$ are also rank n factors of H_k. With this latter choice we obtain

$$\tilde{\mathcal{O}}_k = \mathcal{O}_k T, \qquad \tilde{\mathcal{C}}_k = T^{-1} \mathcal{C}_k$$

From Chapters 3 and 4 we know however that these are the observability and controllability matrices, respectively, of the model obtained by applying the similarity transformation T to (Φ, Γ, C). This shows that a choice of coordinate basis is made implicitly when H_k is factored into ML.

Modelling Errors and Measurement Noise

The above procedure for identifying a state space model relies on the assumption that the measured data were indeed generated by a linear system and are not corrupted by measurement noise. In practice, neither assumption will be true. One consequence of this is that no matter how large k is chosen, the matrix H_k will usually have full rank. To extract information about the model order in spite of data being corrupted, one can use the technique of *singular value decomposition*, a brief review of which is given in the Appendix.

Singular Value Decomposition

Consider a singular value decomposition of the data matrix $H_k \in \mathbb{R}^{kl \times km}$

$$H_k = Q\Sigma V^T \tag{8.14}$$

where Σ is a diagonal matrix with nonnegative diagonal entries, and Q and V are orthogonal, i.e. they satisfy

$$QQ^T = I_{kl}, \qquad VV^T = I_{km}$$

Assume that the singular values are arranged in decreasing order such that

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots \sigma_p = 0$$

where $p = \min(kl, km)$. In this case we have rank $H_k = r$, because an important fact about singular value decomposition is that the rank of a matrix is equal to the number of its nonzero singular values. Here σ_{r+1} and all the following singular values are zero. On the other hand, if the singular values $\sigma_{r+1}, \ldots, \sigma_p$ are very small - much smaller than σ_r - but nonzero, the matrix H_k has full rank but is "close to being singular". This is precisely the situation we encounter when a block-Hankel matrix is constructed from impulse response data that are corrupted by measurement noise. One of the powerful features of the singular value decomposition is that it allows us to distinguish significant information from noise effects by inspection of the singular values. An example is shown in Fig. 8.3, where the singular values of a matrix are shown in decreasing order. If these were the singular values of a Hankel matrix constructed from a measured impulse response, we would conclude that the system dynamics can be described reasonably well by a 4th order model and that the remaining nonzero but small singular values represent noise effects.
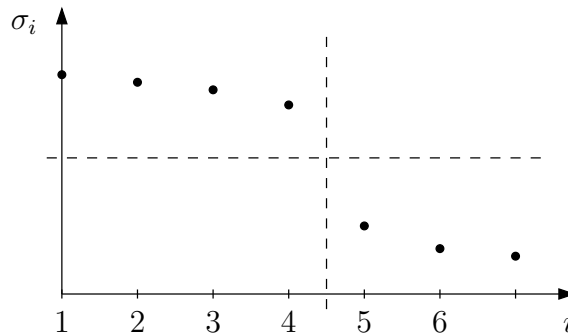
Figure 8.3: Determining the numerical rank

If σ_{r+1} is much smaller than σ_r, we say that the *numerical rank* of H_k is r. Another way of looking at this is to write (8.14) as

$$H_k = \begin{bmatrix} q_1 & q_2 & \ldots & q_{kl} \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 & 0 & \ldots & 0 \\ & \ddots & & 0 & \ldots & 0 \\ 0 & & \sigma_p & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_{km}^T \end{bmatrix}$$

where q_i and v_i represent the ith column of Q and V, respectively. Expanding the right hand side column by column, we obtain

$$H_k = \sum_{i=1}^{p} \sigma_i q_i v_i^T = \sum_{i=1}^{r} \sigma_i q_i v_i^T + \sum_{i=r+1}^{p} \sigma_i q_i v_i^T = Q_s \Sigma_s V_s^T + Q_n \Sigma_n V_n^T$$

where $Q_s \in \mathbb{R}^{kl \times r}$ and $V_s \in \mathbb{R}^{km \times r}$ are the matrices formed by the first r columns of Q and V, respectively. The matrices $Q_n \in \mathbb{R}^{kl \times (kl-r)}$ and $V_n \in \mathbb{R}^{km \times (km-r)}$ are similarly formed by the remaining columns. If the singular values $\sigma_{r+1}, \ldots, \sigma_p$ are much smaller than σ_r, the last term on the right hand side can be neglected and we have

$$H_k \approx Q_s \Sigma_s V_s^T$$

or

$$H_k \approx \begin{bmatrix} q_1 & q_2 & \ldots & q_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 & 0 & \ldots & 0 \\ & \ddots & & 0 & \ldots & 0 \\ 0 & & \sigma_r & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_r^T \end{bmatrix} = Q_s \Sigma_s^{1/2} \Sigma_s^{1/2} V_s^T$$

where r is the numerical rank of H_k. Now taking r as the estimated model order $\hat{n}$, we can define the extended observability and controllability matrices $\mathcal{O}_r \in \mathbb{R}^{kl \times r}$ and $\mathcal{C}_r \in \mathbb{R}^{km \times r}$ as

$$\mathcal{O}_r = Q_s \Sigma_s^{1/2}, \qquad \mathcal{C}_r = \Sigma_s^{1/2} V_s^T$$

A state space model (Φ, Γ, C) of order $\hat{n}$ can then be obtained as in the case of ideal measurements.

The identification of a state space model from the impulse response is illustrated in Exercise 8.6.

8.4 Direct Subspace Identification

The method outlined in the previous section assumes that the measured impulse response is available. In practice it is usually better to use more general data, obtained for example by applying a white noise input signal. We will now present a technique for identifying state space models without using the measured impulse response, referred to as *direct subspace identification.*

Consider again the model (8.10). Beginning at time k, the output at successive time instants is given by

$$\begin{aligned} y(k) &= Cx(k) + Du(k) \\ y(k+1) &= C\Phi x(k) + C\Gamma u(k) + Du(k+1) \\ y(k+2) &= C\Phi^2 x(k) + C\Phi\Gamma u(k) + C\Gamma u(k+1) + Du(k+2) \\ &\vdots \end{aligned}$$

Introducing the vectors

$$Y_k = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+\alpha-1) \end{bmatrix}, \quad U_k = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+\alpha-1) \end{bmatrix}$$

of input and output data, we can write

$$Y_k = \mathcal{O}_\alpha x(k) + \Psi_\alpha U_k \tag{8.15}$$

where

$$\mathcal{O}_\alpha = \begin{bmatrix} C \\ C\Phi \\ C\Phi^2 \\ \vdots \\ C\Phi^{\alpha-1} \end{bmatrix}, \quad \Psi_\alpha = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ C\Gamma & D & 0 & \cdots & 0 \\ C\Phi\Gamma & C\Gamma & D & & 0 \\ \vdots & & & \ddots & \ddots \\ C\Phi^{\alpha-2}\Gamma & C\Phi^{\alpha-1}\Gamma & \cdots & C\Gamma & D \end{bmatrix}$$

Assume that a sufficient number of measurements has been collected so that we can form the input and output data matrices

$$\mathcal{Y} = [Y_1 \ Y_2 \ \ldots \ Y_N], \quad \mathcal{U} = [U_1 \ U_2 \ \ldots \ U_N],$$

Define also the matrix of state variables

$$X = [x(1) \ x(2) \ \ldots \ x(N)]$$

From (8.15), these data matrices satisfy

$$\mathcal{Y} = \mathcal{O}_\alpha X + \Psi_\alpha \mathcal{U} \tag{8.16}$$

In this equation only $\mathcal{U}$ and $\mathcal{Y}$ are known; note that $\mathcal{U} \in \mathbb{R}^{m\alpha \times N}$. We assume that the number $(N + \alpha - 1)$ of measurements - which is required to fill the above matrices - is large enough such that α can be chosen greater than the expected model order, and N such that $N > m\alpha$. To identify a state space model, we need to estimate the matrices $\mathcal{O}_\alpha$ (from which C and Φ can be extracted) and Ψ_α (from which we get D and Γ).

Estimating the Term $\mathcal{O}_\alpha X$

As a first step, we will eliminate the effect of the input data on the output data in (8.16) and estimate the product $\mathcal{O}_\alpha X$. This can be achieved by projecting the output data onto the *nullspace* of the input data matrix $\mathcal{U}$. The nullspace $\mathcal{N}(\mathcal{U})$ is defined as the space of all vectors q that are made zero when multiplied from the left by $\mathcal{U}$:

$$\mathcal{N}(\mathcal{U}) = \{q : \mathcal{U}q = 0\}$$

Now define the matrix Π as

$$\Pi = I - \mathcal{U}^T(\mathcal{U}\mathcal{U}^T)^{-1}\mathcal{U}$$

All columns of Π are orthogonal to $\mathcal{U}$, this can be seen from

$$\mathcal{U}\Pi = \mathcal{U} - \mathcal{U}\mathcal{U}^T(\mathcal{U}\mathcal{U}^T)^{-1}\mathcal{U} = 0$$

Note that Π is constructed from measured data only. Here we assumed that $(\mathcal{U}\mathcal{U}^T)$ is invertible, a condition for this is that the input is persistently exciting of order $m\alpha$. Multiplying equation (8.16) from the right by Π then yields

$$\mathcal{Y}\Pi = (\mathcal{O}_\alpha X + \Psi_\alpha \mathcal{U})\Pi = \mathcal{O}_\alpha X\Pi$$

The left hand side is known (because it is constructed from measured data), thus the product $\mathcal{O}_\alpha X\Pi$ is known. Observing that $\mathcal{O}_\alpha \in \mathbb{R}^{l\alpha \times n}$ and $X\Pi \in \mathbb{R}^{n \times N}$, we can obtain an estimate of the extended observability matrix by determining the numerical rank $\hat{n}$ of the matrix $\mathcal{Y}\Pi$ and by factoring it into a left factor with $\hat{n}$ columns and full column rank, and a right factor with $\hat{n}$ rows. This can be done by computing the singular value decomposition

$$\mathcal{Y}\Pi = Q_s\Sigma_s V_s^T + Q_n\Sigma_n V_n^T \approx Q_s\Sigma_s^{1/2}\Sigma_s^{1/2}V_s^T$$

and by taking

$$\mathcal{O}_\alpha = Q_s\Sigma_s^{1/2}$$

Here again the order of the system is estimated by inspection of the singular values - this time of the data matrix $\mathcal{Y}\Pi$. From $\mathcal{O}_\alpha$ the matrices C and Φ can be obtained as described in the previous section.

Estimating the Term $\Psi_\alpha \mathcal{U}$

We can now use the estimate of $\mathcal{O}_\alpha$ to eliminate the first term on the right hand side of (8.16). For this purpose, observe that from

$$QQ^T = \begin{bmatrix} Q_s^T \\ Q_n^T \end{bmatrix} \begin{bmatrix} Q_s & Q_n \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

we have $Q_n^T Q_s = 0$ and therefore $Q_n^T \mathcal{O}_\alpha \approx 0$. Thus, from (8.16) we obtain

$$Q_n^T \mathcal{Y}\mathcal{U}^{-R} = Q_n^T \Psi_\alpha$$

where $\mathcal{U}^{-R} = \mathcal{U}^T(\mathcal{U}\mathcal{U}^T)^{-1}$ denotes the right inverse of $\mathcal{U}$. The left hand side of this equation and Q_n are known, so that Ψ_α is the only unknown term. The matrices Γ and D can then be obtained by solving a linear system of equations, details are omitted.

Exercises

Problem 8.1

Consider the sum of squared errors

$$V(p) = \sum_{l=0}^{k} e^2(l) = E^T E = (Y - Mp)^T(Y - Mp)$$

introduced in (8.3). Show that $V(p)$ is minimized by the parameter vector $p = \hat{p}$ where

$$\hat{p} = (M^T M)^{-1} M^T Y$$

Problem 8.2

a) Show that for the step function $\sigma(k)$

$$(z - 1)\sigma(k) = 1 \text{ at } k = -1$$

and

$$(z - 1)\sigma(k) = 0 \text{ at } k \geq 0$$

Hint: Theorem 8.2 can be used to solve this problem.

b) If for a given signal $u(k)$ there is at least one polynomial $a(z)$ of order n such that

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} (a(z)u(l))^2 = 0$$

what does this indicate about the order of persistent excitation $u(k)$?

c) Use the polynomial

$$a(z) = z - 1$$

and the results from (a) and (b) to find the greatest possible order of persistent excitation of a step function.

d) Calculate the empirical covariance matrix $C_{uu}(1)$ for the step function. Use this matrix to show that the order of persistent excitation of a step function is 1.

Problem 8.3

a)   For the input signal

$$u(k) = \sin \omega kT$$

show that

$$(z^2 - 2z \cos \omega T + 1)u(k) = 0$$

where T is the sampling time.

Hint: Determine $(z^2 - 2z \cos \omega T + 1)u(k)$, simplify using trigonometric identities and apply Theorem 8.2.

b)   Find the greatest possible order of persistent excitation of the signal $u(k)$?

c)   The autocorrelation function for any signal $x(t)$ is defined as,

$$R_x(\tau) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N} x(kT)x(kT \pm \tau)$$

It can be shown that for $u(t) = \sin \omega t$,

$$R_u(\tau) = \frac{1}{2} \cos \omega \tau$$

Using these facts show that for the signal $u(k)$

$$C_{uu}(2) = \frac{1}{2} \begin{bmatrix} 1 & \cos \omega T \\ \cos \omega T & 1 \end{bmatrix}$$

Hint: Write the elements of $C_{uu}(2)$ in terms of the autocorrelation function.

d)   What is the order of persistent excitation of the signal $u(k)$ when

i) $T = \frac{2\pi}{\omega}$ ii) $T \neq \frac{2\pi}{\omega}$

Explain these results.

Problem 8.4

Use the empirical covariance matrices $C_{uu}(1)$, $C_{uu}(2) \ldots C_{uu}(n)$ to show that the order of persistent excitation of sampled white noise is arbitrarily high.

Hint:Use the fact that the correlation between white noise at times T and $kT + T$ is 0 and $\lim_{k \to \infty} \frac{1}{k} \sum_{i=0}^{k} u_i^2 = S_0$, where S_0 is the spectral density of the white noise.

Problem 8.5

Download `cs7_LSsysdat.mat`. The MAT file contains sampled input and output signals of a SISO system, where the input signal is a step (input `u1`, output `y1`), a sinusoid (input `u2`, output `y2`) or white noise (input `u3`, output `y3`). A pulse transfer function is to be determined that approximates the behaviour of the system from which the measurements were taken.

a)   From N samples of the inputs and N samples of the output create the measurement matrix M for a system of order n. What is the dimension of the matrix M produced from these data?

b)   Determine the rank of the matrix $M^T M$ for

   i)   the sinusoid

   ii)   the white noise signal

   Explain the results.

c)   From output data generated from white noise, calculate a least squares estimate of the model parameters.

d)   i)   Estimate the models of order 2, 3 and 4 using the white noise input signal.

   ii)   Validate the model for the step and sinusoidal input signals.

   iii)   What is the order of the system?

e)   Explain the results from (d) with the help of Problems 8.2 and 8.3.b.

Problem 8.6

Download the Matlab script `cs7_mkdata.m`. This script generates the impulse responses $g(k)$ and $gn(k)$ of a system with 2 inputs and 2 outputs. The sequence $g(k)$ is noise free, whereas $gn(k)$ is corrupted by measurement noise.

a)   Generate from the sequence $g(k)$ the block Hankel matrix of the impulse response. Estimate upper and lower limits for the order n of the system and determine by factorization of the block Hankel matrix linear state space models of the system for different values of the order n. Compare the impulse responses of the estimated models with the output sequence $g(k)$.

   Hint: You can use the function `mkhankel.m` *to generate the Hankel matrix.*

b)   Repeat the estimation for the noisy impulse response $gn(k)$.

Problem 8.7

This exercise uses the Matlab Identification toolbox GUI `ident` to identify a state space model from sets of data with two inputs and two outputs. The data is in the file `cs9_identGUI.mat`.

Two sets data are contained in the file,`iodata1` and `iodata2`. They are in the `Matlab` format `iddata` that can be directly imported into `ident`.

a)   Import the data set `iodata1` and generate direct subspace identified models of different orders using the command `n4sid`.

b)   validate the models generated against the data set `iodata2`. What is the model order that most effectively describes the plant behaviour?

Chapter 9

Model Order Reduction

Modern state space design techniques - like those discussed in this course - produce controllers of the same dynamic order as the generalized plant. Thus, when the system to be controlled has a high dynamic order, the controller may be too complex to be acceptable for implementation. In such cases, the plant model should be approximated by a simplified model. Alternatively, a high-order controller can be approximated by a low-order controller. This chapter gives a brief introduction to the topic of model order reduction.

Consider a stable system with transfer function $G(s)$ and state space realization

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du$$

with n dimensional state vector x. If the number of states n is very large, one could try to find a model of lower order that behaves "similar" to the original system. For example, if some of the state variables do not have much effect on the system behaviour, one might consider removing these states from the model. Thus, we need to know which states are "important" for the model and which ones are not. The controllability Gramian and the observability Gramian turn out to be helpful for answering this question.

Controllability Gramian

Recall the definition of the controllability Gramian

$$W_c = \int_0^\infty e^{At} B B^T e^{A^T t} \, dt$$

A useful geometrical interpretation of W_c - given here without proof - is the following. Define the set S_c as the set of all points in state space to which the state vector $x(t)$ of the system can be driven from the origin with an input signal $u(\tau)$, $0 \le \tau \le t$, that satisfies

$$\int_0^t u^T(\tau) u(\tau) d\tau \le 1$$

In other words: we consider all input signals with energy less than or equal to 1 (assume that $u(\tau) = 0$ for $\tau > t$). Compute the singular value decomposition of the controllability Gramian

$$W_c = V\Sigma V^T$$

where

$$\Sigma = \text{diag}\,(\sigma_1,\ \sigma_2, \ldots \sigma_n) \quad \text{and} \quad V = [v_1\ v_2\ \ldots v_n]$$

The singular values are ordered such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ as usual. The set S_c of points in the state space reachable with input energy 1 is then a hyper-ellipsoid centered at the origin, with semi-axes in the directions of the columns v_i of V, and length given by the square root of the corresponding singular value. This is illustrated in Figure 9.1 for a second order system. Note that v_1 is the direction that is most easily controlled, whereas small singular values indicate directions which are difficult to control.
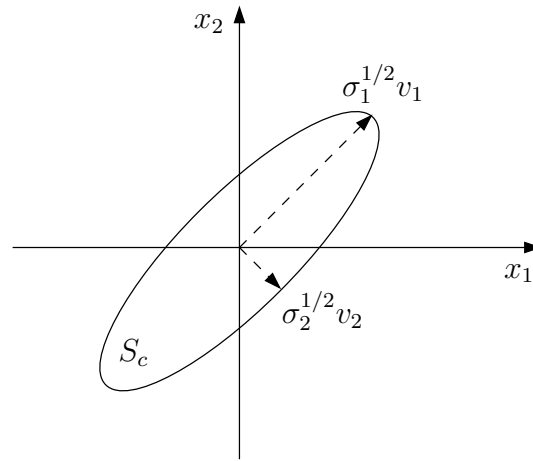
Figure 9.1: Interpretation of the controllability Gramian

Observability Gramian

A similar interpretation can be given for the observability Gramian

$$W_o = \int_0^\infty e^{A^T t} C^T C e^{At}\, dt$$

Define the set S_o as the set of all points in the state space which - when taken as initial conditions $x(0)$ - lead to a zero-input response $y(t)$ that satisfies

$$\int_0^\infty y^T(t)y(t)dt \leq 1$$

Again the singular value decomposition

$$W_o = V\Sigma V^T$$

where

$$\Sigma = \text{diag}\left(\sigma_1,\ \sigma_2, \ldots \sigma_n\right) \quad \text{and} \quad V = [v_1\ v_2\ \ldots v_n]$$

determines the set S_o - it is a hyper-ellipsoid centered at the origin with semi-axes given by $v_i/\sqrt{\sigma_i}$. This set is illustrated for a second order system in Figure 9.2. Note that the axes are long in directions with small singular values, indicating directions that have little effect on the output. These directions are difficult to observe.
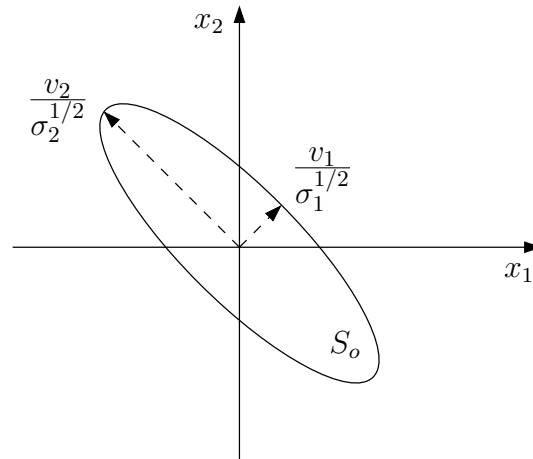
Figure 9.2: Interpretation of the observability Gramian

Balanced Realization

The question posed at the beginning of the chapter was: which state variables are important for the system and which ones are not? The singular value decomposition of the Gramians tells us which states show only a weak response to a control input (the ones associated with small singular values of W_c), and which ones have only weak influence on the observed output (the ones where the singular values of W_o are small). Now it would be unwise to remove a state variable from the model only because it shows little response to control inputs - the same state variable may have a strong effect on the output. The reverse may be true for states with small singular values of W_o. To find out which states have little influence both in terms of controllability *and* observability, we will use a special state space realization of the plant model which is known as *balanced realization*.

We should keep in mind that controllability and observability are state space concepts. The problem considered here can of course also be expressed in terms of input-output behaviour, i.e. transfer function models: the notion of near-uncontrollable or near-unobservable then takes the form of near pole-zero cancellations. The state space framework lends itself however better to a numerical treatment.

Initially we assumed a state space realization of $G(s)$ with system matrices A, B, C and

D. Applying a similarity transformation T leads to a different state space model

$$(A, B, C, D) \quad \rightarrow \quad (T^{-1}AT, T^{-1}B, CT, D)$$

for the same plant. The eigenvalues and the input/output behaviour of both state space models are the same, because they are realizations of the same transfer function. The controllability and observability Gramians however are different: it is straightforward to check that if W_c and W_o are the Gramians of the original model, then $T^{-1}W_cT^{-T}$ and T^TW_oT are the Gramians of the transformed model, respectively.

A balanced realization of $G(s)$ has the property that its controllability and observability Gramians are equal and diagonal

$$W_c = W_o = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix}$$

Using Cholesky factorization and singular value decomposition, it is always possible to find a similarity transformation T that brings a given state space model into this form - in MATLAB one can use the function *balreal()* for this task. The diagonal entries of the Gramian are called the *Hankel singular values* of the system. In the coordinate basis of the state space associated with the balanced realization, a small Hankel singular value indicates that a state has little influence both in terms of controllability and observability. Therefore, this realization is well suited for model reduction by removing "unimportant" state variables. We will discuss two different ways of doing this.

Let (A, B, C, D) be a balanced realization of $G(s)$ with n state variables. Assume the inspection of the Hankel singular values indicates that only r states are significant and that the last $n - r$ Hankel singular values are small enough to be neglected. Partition the state space model as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad C = [C_1 \ C_2] \tag{9.1}$$

where $A_{11} \in \mathbb{R}^{r \times r}$, $A_{22} \in \mathbb{R}^{(n-r) \times (n-r)}$ etc.

Balanced Truncation

The subsystem (A_{11}, B_1, C_1, D) of the partitioned model (9.1) contains the states with significantly large Hankel singular values. One approach to model order reduction is to use this system with r state variables as an approximation of the full order model $G(s)$. This approach is known as *balanced truncation*, because the parts of the model associated with insignificant states are simply ignored.

An important property of the resulting reduced order model

$$G_{\text{tr}}(s) = C_1(sI - A_{11})^{-1}B_1 + D$$

is that it satisfies

$$G_{\text{tr}}(j\infty) = G(j\infty) = D$$

The direct feedthrough terms of full order and truncated model are the same. This indicates that both models will exhibit similar high-frequency behaviour.

Balanced Residualization

An alternative approach is not to ignore the insignificant states, but to assume that they are constant and take them into account in the reduced order model. This method is known as *balanced residualization.* In the partitioned model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

$$y = [C_1 \;\; C_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du$$

we make the assumption

$$\dot{x}_2 = 0$$

and eliminate x_2 from the model. It is straightforward to check that the resulting reduced order model is

$$G_{\text{res}}(s) = C_{\text{res}}(sI - A_{\text{res}})^{-1} B_{\text{res}} + D_{\text{res}}$$

where

$$A_{\text{res}} = A_{11} - A_{12}A_{22}^{-1}A_{21}, \qquad B_{\text{res}} = B_1 - A_{12}A_{22}^{-1}B_2$$

$$C_{\text{res}} = C_1 - C_2A_{22}^{-1}A_{21}, \qquad D_{\text{res}} = D - C_2A_{22}^{-1}B_2$$

The feedthrough term is different from that of the full order model, indicating that the high-frequency behaviour will not be the same. In fact the reduced model was arrived at by assuming that derivatives of some state variables are zero. This is true in steady state, so we would expect similar behaviour of full order and reduced model at low frequencies. One can indeed verify that the steady state gains of both models are the same, i.e.

$$G(0) = G_{\text{res}}(0) = D - CA^{-1}B$$

Example 9.1

A linearized model of a High Voltage DC transmission system with 31 state variables is approximated by a second order model. Figure 9.3.a shows the magnitude frequency response when balanced truncation is used. Figure 9.3.b shows the result of balanced residualization. The results illustrate that balanced truncation attempts to capture the high frequency behaviour of the full order model, whereas balanced residualization captures its low frequency behaviour.

Order Reduction for Unstable Models

So far we assumed that the full order model is stable - a balanced realization is defined only for stable systems. If the reduced order model is to be used for controller design, it would be unwise to ignore any unstable pole of the plant. One way of dealing with an
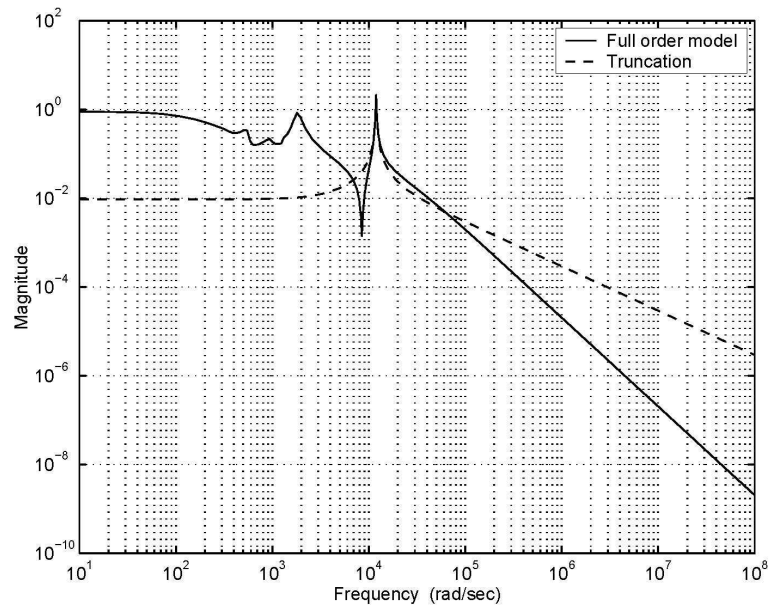
(a) Balanced truncation
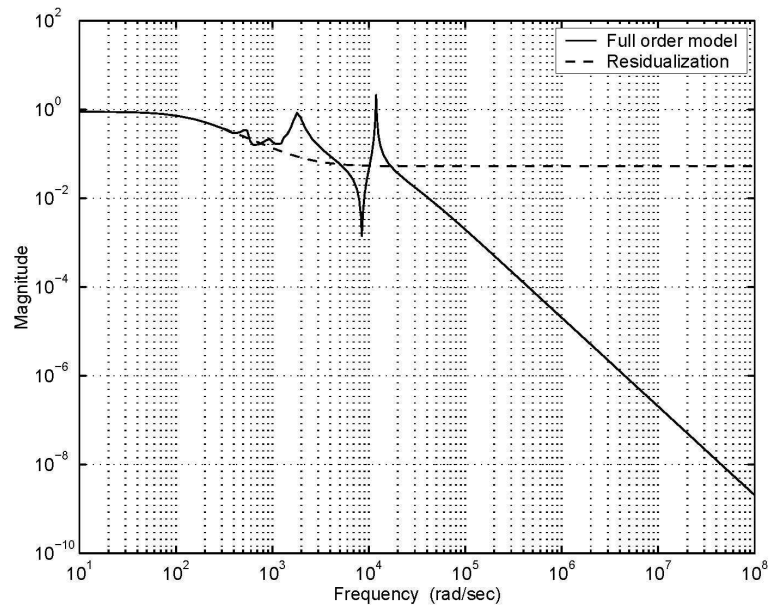
(b) Balanced residualization

Figure 9.3: Comparison of balanced truncation (a) and balanced residualization (b)

unstable high order system is to split it up into a stable and an antistable part, i.e.

$$G(s) = G_s(s) + G_u(s)$$

where $G_s(s)$ is stable and $G_u(s)$ has all poles in the right half plane. The stable part of the system can then be reduced by balanced truncation or residualization to $\hat{G}_s(s)$, and a reduced model of the plant that retains all unstable modes is

$$\hat{G}(s) = \hat{G}_s(s) + G_u(s)$$

This approach is implemented in the MATLAB function *balmr()*.

Exercises

Problem 9.1

a) Write `Matlab` functions based on the theory in this Chapter to perform model order reduction by both truncation and residualisation. You may use the matlab command `balreal` to get a balanced realisation.

b) The file `cs8_HVDC_31.mat` contains a linearised model of a high voltage DC transmission scheme described in Example 9.1. Use your functions to reproduce the magnitude plots for the second order, reduced, models created by both truncation and residualisation.

Chapter 10

Case Study: Modelling and Multivariable Control of a Process Evaporator

This chapter uses a realistic example of a the design of a controller for a chemical process to demonstrate how the various tools described in this course should be applied in practice. This is an important aspect of the study of control systems - without such study it can be difficult to see how the different identification, control design and discretization tools fit together and can be applied to practically meaningful problems.

Figure 10.1 shows a schematic diagram of a process evaporator. In the evaporation vessel certain conditions on temperature and pressure must be fulfilled by adjustment of liquid, gas and heating fluid flows so that the required rate of evaporation is maintained.

In this chapter a linear model of the process will be identified, where the plant is represented by a non-linear Simulink model. Based on the linear model a controller is then designed and applied to the non-linear plant model.

Download the file `cs9_evaporator.zip`. It contains several files concerning the model and the design of the controller for it. These files are explained in the next section. The control objectives and the controller design procedure to be followed are then described.

10.1   Evaporator Model

`evapmod.mdl` is a nonlinear Simulink model of the evaporator. The model has three inputs:

$u1 =$ Liquid outflow
$u2 =$ Gas outflow
$u3 =$ Heating fluid flow
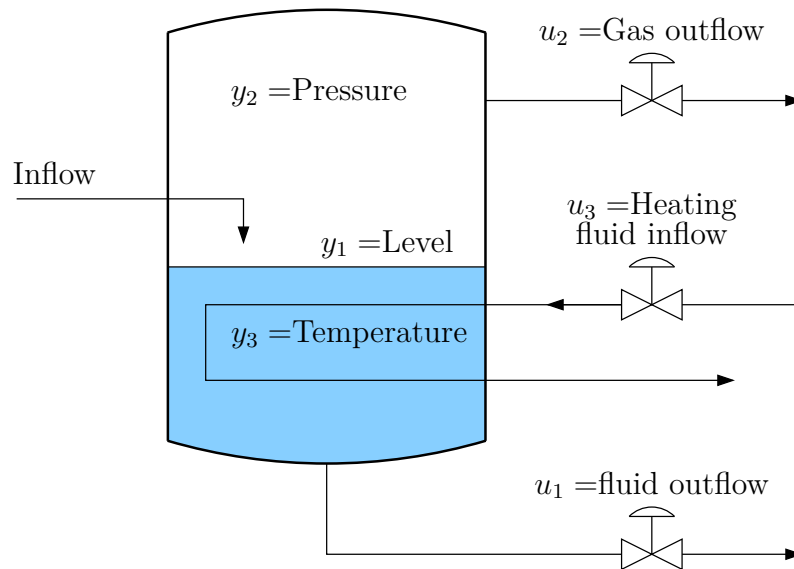
Figure 10.1: Process diagram of evaporator

There are three outputs that are measured and are to be controlled:

$y1 =$ Reactor liquid level
$y2 =$ Reactor pressure
$y3 =$ Reactor temperature

as well as an unmeasured disturbance – Reactor inflow.

The time unit of the model is the minute.

The file `plantdata.mat` contains the steady state values for input `u0`, state vector `x0evap`, output `y0` and a vector of process parameters `pvec` used in obtaining the test data. The file also contains two sets of measured input and output signals for the plant in open loop with a sampling time of 0.05 minutes which should be used for system identification: `utest1`, `ytest1` and `utest2`, `ytest2`.

10.2   Control Objectives

A controller is to be designed to achieve the following objectives:

1.   After step changes in setpoints or reactor inflow the steady state error must be zero.

2.   Following a step disturbance of size 0.4 in the evaporator inflow the following conditions must be satisfied:

   (a)   The outputs must always be within the following limits,
        level: 0.1, pressure : 2.0, temperature: 2.0

(b) The 5% settling times must not be greater than the following
level: 50 min, pressure: 10 min, temperature: 10 min

(c) The inputs should always remain with in ± 1.0

3. The standard deviation of the noise on the individual inputs should be less than 0.01.

10.3 Modelling and Design

Use the system identification toolbox (command `ident`) to estimate a linear model from the data sequences `utest1` and `ytest1` with a sampling time of 0.05 minutes. Validate this model against the data in the matrices `utest2` and `ytest2`.

Hints for using the Identification Toolbox:

- Open the toolbox with the command `ident`.

- First you must import the signals: *Import data → Time domain data*:

Input:	`utest1`
Output:	`ytest1`
Starting time:	1
Samp. interv.:	0.05

  Repeat for the second data set.

- The signals then appear in the left panel *Data Views*.

- Remove the means from all signals using the *Preprocess* drop-down list → *Remove means*. The new set of signals should be used as *Working data* and *Validation data*.

- Estimate models of 3rd, 4th and 5th order using N4SID (subspace identification). For this purpose choose *Linear parametric models* from the *Estimate* drop-down box. Select *State-space* as Structure and repeat the identification for the different orders.

- Validate the identified models using the second data set. Use the *Model Views* check-boxes in the lower-right corner of the GUI.

When you have found a good model, call this model `mod1` (by right-clicking it) and export this model into the workspace.

10.4 Controller Design Files

The file `contdesign.m` is a Matlab script that performs the following tasks:

- Controller and observer design for a linearized model

- Display of setpoint step response, comparison of linear and non-linear models for a particular controller.

- Response of the non-linear model to a disturbance in the evaporator inflow

- Calculation of the following control performance measures

    (a) `stdvec`: Standard deviations of outputs and inputs
    (b) `rangevec`: The 'amplitude range ' for input and output signals
        (for a signal $x(k), k = 0 \dots N$, the amplitude range is $\max x(k) - \min x(k)$)
    (c) `stvec`: 5% settling time for inputs and outputs

Re-running the script, without closing the figures allows comparing different designs. A set of weighting matrices for testing is offered in the file `tuningparset.m`.

10.5 Controller Design

1. Read and understand the Matlab script `contdesign.m`.

2. Write a Matlab script `controllerdesign.m` to design a state-feedback controller and an observer. The script should return the state-feedback gain and the observer's A and B matrices.

3. Scale the inputs and outputs to do the controller design. Remember that the scaling needs to be accounted for when the designed controllers are applied to the plant.

4. To simulate the non-linear model, the initial state of the observer and integrator blocks need to be calculated so that the simulation can begin in steady state.

5. The script `simandplot.m` simulates and plots the closed-loop response of the non-linear system and the designed controller to different step signals.

6. Run the script `contdesign.m` to design a controller and see the closed-loop simulation. Repeat for different controller settings. Tune the controller and the observer to achieve the control objectives.

7. Discretize the continuous-time controller and simulate the closed-loop response with the discrete-time controller and a sampling time of 3 seconds (0.05 minutes) in Simulink. Compare your results with the solution provided in `discretise.m`

Bibliography

[1] G. Strang, *Linear Algebra and Its Applications.* Stanford: Elsevier Science, 2014, ISBN: 978-1-483-26511-7.

[2] G. R. Cooper, C. D. McGillem, B. P. o. E. McGillem, and C. E. T. L. C. D, *Probabilistic Methods of Signal and System Analysis -.* New York: Oxford University Press, 1999, ISBN: 978-0-195-12354-8.

[3] L. Ljung, *System Identification - Theory for the User.* Amsterdam: Pearson Education, 1998, ISBN: 978-0-132-44053-0.

[4] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems - Theory and Design, Third Edition.* New York: Courier Corporation, 2013, ISBN: 978-0-486-28404-0.

[5] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control -.* London: Prentice Hall, 1996, ISBN: 978-0-134-56567-5.

[6] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems -.* München: Pearson, 2015, ISBN: 978-0-133-49659-8.

[7] T. Kailath, *Linear Systems.* Prentice-Hall Englewood Cliffs, NJ, 1980, vol. 156.

Part I

APPENDICES

Appendix A

Vector Norms, Matrix Norms and Singular Value Decomposition

A.1   Vector Norms

Consider a vector space $\mathcal{X}$. A norm $\|x\|$ is function mapping a vector x into a real number, that satisfies the following four properties for any $x, y \in \mathcal{X}$

1) $\|x\| \geq 0$

2) $\|x\| = 0 \quad \Leftrightarrow \quad x = 0$

3) $\|\alpha x\| = \alpha \|x\|$ for any scalar α

4) $\|x + y\| \leq \|x\| + \|y\|$

We consider finite dimensional real or complex vector spaces, $\mathcal{X} = \mathbb{R}^n$ or $\mathcal{X} = \mathbb{C}^n$. Let $x = [x_1 \quad x_2 \quad \ldots \quad x_n]^T$ be a vector with $x_i \in \mathbb{R}, \ i = 1, \ldots, n$ or $x_i \in \mathbb{C}, \ i = 1, \ldots, n$. A frequently used norm on x is the vector-p-norm

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

where p is a positive integer. Of practical importance are the three cases

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$$

$$\|x\|_\infty = \max_i |x_i|$$

The vector-2-norm can also be written as

$$\|x\|_2 = \sqrt{x^T x} \quad \text{if} \quad x \in \mathbb{R}^n \quad \text{or} \quad \|x\|_2 = \sqrt{x^H x} \quad \text{if} \quad x \in \mathbb{C}^n$$

where A^H denotes the *Hermitian* of a matrix A: $A^H = \bar{A}^T$ where $\bar{A}$ is the complex conjugate of A. Here we will use the 2-norm for vectors, and we will drop the subscript and write $\|x\|$ for the vector-2-norm of x.

A.2 The Matrix-2-Norm

Consider two complex vectors $x \in \mathbb{C}^n$ and $y \in \mathbb{C}^m$, and a linear mapping $y = Ax$. The complex matrix $A \in \mathbb{C}^{m \times n}$ can be interpreted as an operator that maps x into y, or more generally $\mathbb{C}^n$ into $\mathbb{C}^m$. One can compare the vector norms of x and y, and associate a "gain" with A as the ratio of these vector norms. This ratio depends on x, and an important property of the matrix A is the maximum value of $\|y\|/\|x\|$ over all $x \in \mathbb{C}^n$ (the "maximum gain" of A). This positive real number is defined to be the norm of the matrix A; since it depends also on the choice of vector norm, it is called an *induced norm*. The matrix-2-norm induced by the vector-2-norm is defined as

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} \tag{A.1}$$

Again, we will drop the subscript and write $\|A\|$ for the matrix-2-norm. It is straightforward to verify that the matrix-2-norm - and indeed all induced matrix-p-norms - satisfy the four properties of a norm.

To find the value of $\|A\|$, we take squares on both sides of (A.1) to get

$$\|A\|^2 = \max_{x \neq 0} \frac{\|Ax\|^2}{\|x\|^2} = \max_{x \neq 0} \frac{x^H A^H A x}{x^H x} = \max_{x \neq 0} \frac{x^H M x}{x^H x}$$

where we introduced the Hermitian matrix $M = A^H A$. In order to find the maximum value of the last term, we diagonalise M, i.e. we compute $M = V \Lambda V^{-1}$, where Λ is a diagonal matrix containing the eigenvalues of M, and V is a matrix with right eigenvectors of M as columns.

We will first establish the following useful facts about the Hermitian matrix $M = A^H A$

1) M is positive semi-definite ($x^H M x \geq 0 \ \ \forall x \in \mathbb{C}^n$)

2) the eigenvalues of M are real

3) the eigenvectors of M are orthogonal (two vectors x and y are orthogonal if $x^H y = 0$)

With $y = Ax$, property (1) follows immediately from

$$x^H M x = x^H A^H A x = y^H y \geq 0$$

Note that this implies that $x^H M x$ is real even if x is complex. That the eigenvalues of M are real can be shown as follows. Let λ be an eigenvalue and v be an eigenvector of M, and consider

$$M v = \lambda v$$

Multiplying with v^H from the left yields $v^H M v = \lambda v^H v$. We established already that the left hand side of this equation is real, and on the right hand side $v^H v$ is also real. Thus, λ must be real.

To show that two eigenvectors of M belonging to different eigenvalues are orthogonal, consider

$$M v_1 = \lambda_1 v_1, \qquad M v_2 = \lambda_2 v_2, , \qquad \lambda_1 \neq \lambda_2$$

We have

$$(\lambda_1 v_1)^H v_2 = (M v_1)^H v_2 = v_1^H M v_2 = v_1^H \lambda_2 v_2$$

thus $\lambda_1 v_1^H v_2 = \lambda_2 v_1^H v_2$, and from the assumption $\lambda_1 \neq \lambda_2$ it then follows that $v_1^H v_2 = 0$.

A consequence of property (3) is that if all eigenvectors v_i of M are normalized such that $\|v_i\| = 1$, $i = 1, \ldots, n$, the eigenvector matrix V is *unitary*, i.e. $V^H V = I$, or $V^{-1} = V^H$. (Strictly speaking, we have shown this only for matrices with distinct eigenvalues. It can be shown however that even a matrix with repeated eigenvalues has a full set of orthogonal eigenvectors.)

Note that properties (2) and (3) are true for any Hermitian matrix even when it is not positive semidefinite.

We now return to finding the value of $\|A\|$ by solving

$$\max_{x \neq 0} \frac{x^H A^H A x}{x^H x}$$

With the diagonalisation $A^H A = V \Lambda V^H$ this becomes

$$\max_{x \neq 0} \frac{x^H V \Lambda V^H x}{x^H x}$$

and introducing $y = V^H x$ and thus $x = V y$ (using orthonormality of V), we obtain

$$\max_{y \neq 0} \frac{y^H \Lambda y}{y^H V^H V y} = \max_{y \neq 0} \frac{y^H \Lambda y}{y^H y} = \max_{y \neq 0} \frac{\lambda_1 |y_1|^2 + \lambda_2 |y_2|^2 + \ldots + \lambda_n |y_n|^2}{|y_1|^2 + |y_2|^2 + \ldots + |y_n|^2}$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $A^H A$. Assume that the eigenvalues are ordered such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Then it is easy to see that the maximum value of the above expression is λ_1, which is achieved if we choose $y = [1 \ 0 \ \ldots \ 0]^T$, and the minimum value is λ_n, achieved by choosing $y = [0 \ \ldots \ 0 \ 1]^T$.

Because the above expression is the square of the matrix-2-norm of A, we have thus established that

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sqrt{\lambda_{\max}(A^H A)}$$

and we also found that

$$\min_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sqrt{\lambda_{\min}(A^H A)}$$

The eigenvalues of $A^H A$ are called the *singular values* of A. This leads us to the most important and useful of matrix factorizations, the *singular value decomposition* (SVD), which is discussed next.

A.3   The Singular Value Decomposition

In the last section we used the fact that any Hermitian matrix M can be factored into

$$M = V \Lambda V^H$$

where V is the eigenvector matrix of M and unitary, and Λ is the diagonal eigenvalue matrix of M. The same factorization is obviously not possible for non-Hermitian or even non-square matrices. A similar factorization is however possible in these cases, if we do not insist on the same matrix V on both sides, but allow different unitary matrices U and V as left and right factors.

Theorem A.1 (Singular Value Decomposition)

For every matrix $A \in \mathbb{C}^{m \times n}$ there exist unitary matrices $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ such that

$$A = U \Sigma V^H \tag{A.2}$$

and Σ is real and diagonal with non-negative entries.

The matrix Σ has the same size as A. For example, if A is a 3×2 or 2×3 matrix, then

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix} \quad \text{or} \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{bmatrix}$$

respectively, where $\sigma_{1,2} \geq 0$. The diagonal entries σ_i are called the *singular values* of A.

From (A.2) we obtain $AV = U\Sigma$ and thus

$$Av_i = \sigma_i u_i, \quad i = 1, \ldots, n$$

where v_i and u_i are the columns of V and U, respectively. Compare this with $Mv_i = \lambda_i v_i$ - an eigenvector v_i is transformed into $\lambda_i v_i$, whereas A transforms v_i into $\sigma_i u_i$. From (A.2) we also have

$$AA^H = U\Sigma V^H V \Sigma^T U^H = U\Sigma\Sigma^T U^H \tag{A.3}$$

and

$$A^H A = V\Sigma^T U^H U \Sigma V^H = V\Sigma^T\Sigma V^H \tag{A.4}$$

Equation (A.3) shows that U is the eigenvector matrix of AA^H, and (A.4) shows that V is the eigenvector matrix of $A^H A$. The eigenvalue matrices are $\Sigma\Sigma^T$ and $\Sigma^T\Sigma$, respectively. Again, if A is 3×2 then

$$\Sigma\Sigma^T = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Sigma^T\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

This shows that the singular values of A are the square roots of the eigenvalues of AA^H and $A^H A$.

Proof

To prove Theorem A.1, we show how to construct U, V and Σ that satisfy (A.2) for a given matrix A. We start with the diagonalisation of $A^H A$: we established already that there exists a unitary matrix V such that

$$A^H A = V^H \Lambda V$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal eigenvalue matrix of $A^H A$, and the columns v_i of V are the corresponding eigenvectors. Thus

$$A^H A v_i = \lambda_i v_i \quad \text{and} \quad v_i^H A^H A v_i = \lambda_i v_i^H v_i = \lambda_i$$

because V is unitary, and therefore

$$\|Av_i\|^2 = \lambda_i \tag{A.5}$$

This implies that $\lambda_i \geq 0$. Assume that the eigenvalues $\lambda_1, \ldots, \lambda_r$ are positive and the remaining $n - r$ eigenvalues λ_i and vectors Av_i are zero. Note that $r \leq \min(n, m)$. Define

$$\sigma_i = \sqrt{\lambda_i}, \quad u_i = \frac{1}{\sigma_i} A v_i, \quad i = 1, \ldots, r$$

It follows from (A.5) that $\|u_i\| = 1$. Moreover, we have

$$u_i^H u_j = \frac{v_i^H A^H A v_j}{\sigma_i \sigma_j} = \frac{\lambda_i v_i^H v_j}{\sigma_i \sigma_j} = 0, \quad i \neq j$$

This shows that the vectors $u_1, \ldots, u_r$ defined above have the properties required of column vectors for U to be unitary. If $r < m$, one can fill up the matrix U with $m - r$ further

orthogonal vectors (by using Gram-Schmidt orthogonalization) to make it into a $m \times m$ unitary matrix.

Now it remains to show that the matrices U, V as defined above satisfy

$$U^H A V = \Sigma$$

where Σ is diagonal with σ_i as diagonal entries. The (i, j) entry of $U^H A V$ is

$$(U^H A V)_{i,j} = u_i^H A v_j = \begin{cases} \sigma_j u_i^H u_j, & j \leq r \\ 0, & j > r \end{cases}$$

Because $\sigma_j u_i^H u_j$ is zero if $i \neq j$ and σ_j if $i = j$, the above shows that the entries of $U^H A V$ are all zero except for the first r entries on the main diagonal, which are the singular values of A. This completes the proof.

Exercises

Problem A.1

Show that $\|AB\| \leq \|A\|\|B\|$

Problem A.2

Consider the matrix

$$A = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & 1 \end{bmatrix}$$

The *four fundamental subspaces* of a matrix A are defined as follows:

The *column space* of A is the space spanned by the columns of A.

The *nullspace* of A is the space of all vectors x such that $Ax = 0$.

The *row space* of A is the column space of A^T. It is spanned by the rows of A.

The *left nullspace* is the nullspace of A^T. It contains all vectors y such that $A^T y = 0$.

a)   What is the rank of A?

b)   Find a basis for the nullspace of A, the column space of A, the row space of A and the left nullspace of A.

c)   Use the matrix command `svd` in `Matlab` to calculate bases for these subspaces. Verify that the results given by `Matlab` are equivalent to your results from (b).

Appendix B

Probability and Stochastic Processes

This chapter briefly reviews in a tutorial fashion some fundamental concepts of probability theory, stochastic processes, and of systems with random inputs.

B.1 Probability and Random Variables

An important concept in probability theory is that of a *random experiment*. An experiment is an action that results in a certain *outcome*. A random experiment is characterized by the fact that the outcome is uncertain before the experiment takes place. Examples of random experiments are

- flipping a coin, possible outcomes are head (H) or tail (T),

- measuring an unknown voltage, possible outcomes are voltage between v_1 and v_2, or not between v_1 and v_2.

A single performance of a well-defined experiment is referred to as a *trial*. A related concept is that of an *event*, which is the occurrence of a possible outcome, such as getting a head when flipping a coin. The notion of an event will be made more precise below.

There are several ways of defining the *probability* of an event. Two widely used approaches to probability are the *relative-frequency* approach, and the *axiomatic* approach. The former tries to attach some physical significance to the concept of probability; its usefulness is however limited in the sense that it is difficult to deduce sophisticated mathematical tools from it. The axiomatic approach, on the other hand, is more abstract: it defines probability of an event simply as a number, associated with that event, that satisfies certain conditions. From an engineering point of view, a combination of both ideas is useful: the relative-frequency approach relates the term probability to physical reality, whereas the axiomatic approach can be used to develop the mathematical machinery for analyzing complex situations.

Relative-Frequency Approach to Probability

Consider the experiment of flipping two coins. Assuming that we can tell the coins apart, there are four possible outcomes: head-head (HH), head-tail (HT), tail-head (TH) and tail-tail (TT). Suppose that this experiment is repeated N times, and that the event HH occurs N_{HH} times. An intuitive way of explaining the concept of probability P associated with this particular event is to consider a large number of trials ($N \to \infty$) and view the probability as the relative frequency of the given event, i.e.

$$P(\text{HH}) = \lim_{N \to \infty} \frac{N_{\text{HH}}}{N}$$

Assuming that all events are *equally likely* (i.e. that the coins are unbiased) and *mutually exclusive*, we conclude that

$$P(\text{HH}) = P(\text{HT}) = P(\text{TH}) = P(\text{TT}) = \frac{1}{4}$$

More generally, if there are N possible, equally likely and mutually exclusive outcomes of a random experiment, and if N_A denotes the number of outcomes that correspond to a given event A, then the probability of that event is

$$P(\text{A}) = \frac{N_A}{N}$$

There is however a fundamental flaw in the above reasoning: using the words "equally likely" amounts to saying that outcomes are equally probable - in other words, we are using probability to define probability.

Axiomatic Approach to Probability

Because of the above difficulty, a more systematic approach on an axiomatic basis is preferable. To illustrate the idea, we will use again the experiment of flipping two coins. This time, consider the following two events:

- event A: at least one head

- event B: both coins equal

We define the *probability space* $\mathcal{S}$ associated with this experiment as the set of all possible outcomes, i.e.

$$\mathcal{S} = \{\text{HH, HT, TH, TT}\}$$

An event can then be viewed as a subset of $\mathcal{S}$. Here we have

$$\text{event A} = \{\text{HH, HT, TH}\}, \quad \text{event B} = \{\text{HH, TT}\},$$

See Figure B.1. More events can be associated with different subsets of $\mathcal{S}$. There are two events that are of particular significance. Since at least one outcome must be obtained

on each trial, the space $\mathcal{S}$ corresponds to the *certain event*. Similarly, the empty set $\emptyset$ corresponds to the *impossible event*. An event consisting of only one element of $\mathcal{S}$ is called an *elementary event*, whereas events consisting of more than one elements are called *composite events*.
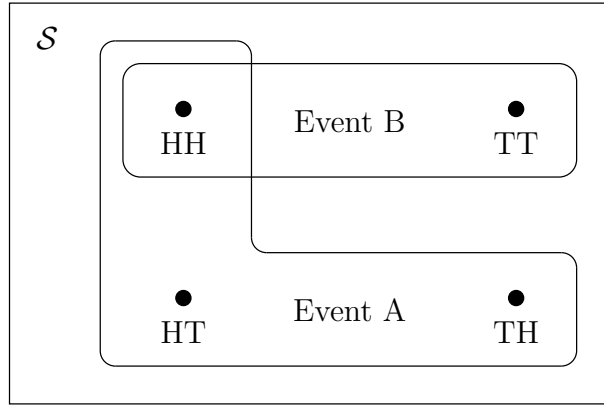

Figure B.1: Probability space

The axiomatic approach defines the probability of an event as a number that satisfies certain conditions (axioms). Let A and B denote two possible events. Also, let $(A \cup B)$ denote the event "A or B or both", and $(A \cap B)$ the event "both A and B". The probability P of an event, say A, is a number associated with that event that satisfies the following three conditions.

$$P(A) \geq 0 \quad \forall A \in \mathcal{S} \tag{B.1}$$

$$P(\mathcal{S}) = 1 \tag{B.2}$$

$$A \cap B = \emptyset \quad \Rightarrow \quad P(A \cup B) = P(A) + P(B) \quad \forall A, B \in \mathcal{S} \tag{B.3}$$

From these axioms, the whole body of probability theory can be derived. For example, the probability $P(A \cup B)$ that A or B or both occur, is given by (B.3) for the case that A and B are mutually exclusive. For events A and B that are not mutually exclusive, we can use the above axioms to show that

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{B.4}$$

The proof is left as an exercise. The probability $P(A \cap B)$ that A *and* B occur simultaneously, is called *joint probability* of A and B, for this we also use the notation $P(A, B)$.

Note that the axiomatic approach does not give us the numerical value of a probability $P(A)$, this must be obtained by other means.

Conditional Probability

The *conditional probability* is the probability of one event A, given that another event B has occurred, it is denoted $P(A|B)$. This concept can be introduced intuitively using

the relative-frequency approach. Consider for example an experiment where resistors are picked at random from a bin that contains resistors with different resistance values and power ratings, as shown in Table B.1.

	1Ω	10Ω	Totals
2W	50	100	150
5W	10	200	210
Totals	60	300	360

Table B.1: Random experiment: picking resistors from a bin

We could ask: what is the probability of picking a 1Ω resistor, when it is already known that the chosen resistor is 5W? Since there are 210 5W resistors, and 10 of these are 1Ω, from a relative-frequency point of view the conditional probability is

$$P(1\Omega|5\text{W}) = \frac{10}{210} = 0.048$$

In the axiomatic approach, the conditional probability of an event A, given B, is defined as

$$P(\text{A}|\text{B}) = \frac{P(\text{A} \cap \text{B})}{P(\text{B})} \tag{B.5}$$

where it is assumed that $P(\text{B}) > 0$. If we define event A as selecting a 1Ω resistor and event B as selecting a 5W resistor, it is straightforward to check that with $P(5\text{W}) = 210/360$ and $P(1\Omega \cap 5\text{W}) = 10/360$ this definition leads to the same result.

When two events A and B are considered, then the probability of event A without regard to the outcome of event B is called the *marginal probability* of A. For example, the marginal probability of selecting a 5W resistor without regard to its resistance value is $P(5\text{W}) = 210/360 = 0.583$.

Independence

An important concept in probability theory is that of statistical independence. Two events A and B are said to be independent if and only if

$$P(\text{A} \cap \text{B}) = P(\text{A})P(\text{B}) \tag{B.6}$$

As a consequence, we have from (B.5) that $P(\text{A}|\text{B}) = P(\text{A})$ if A and B are independent.

Random Variables

Returning to the resistor bin example, we could ask whether a resistor labelled 1 Ω actually has a resistance value of exactly 1 Ω. In reality, the value of resistance can be expected to be close to 1 Ω, but will probably differ from that value by a small amount. We may then

consider the resistance of a 1-Ω resistor as a quantity whose exact value is uncertain, but about which some statistical information is available. Such a quantity is called a *random variable*. Depending on the set of values that a random variable can take, we distinguish between *continuous* and *discrete* random variables. In the resistor example, even if we know that is has a resistance value between 0.9 and 1.1 Ω, there is still an infinite number of possible values in this range. On the other hand, if we consider the experiment of throwing a die, there are only six possible values as outcomes. If the number of possible values is finite, the random variable is discrete (as is the value showing on a die), otherwise it is continuous (like the value of resistance).

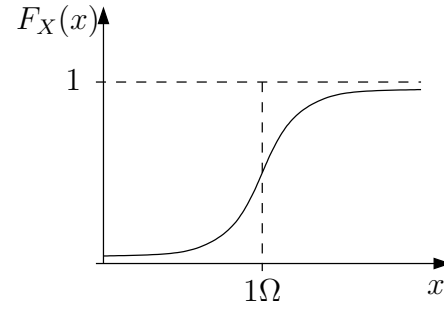**Probability Distribution Function**

In order to be able to use the ideas introduced so far when working with continuous random variables, we need to relate their values to the concept of an event. Let X denote a random variable, and let x denote a possible value the variable X might take when observed in an experiment. We define the *probability distribution function* of the random variable X as the probability of the event that the random variable X takes a value less than or equal to x, i.e.

$$F_X(x) = P(X \leq x) \tag{B.7}$$

Note that $F_X(x)$ is a function of x, not of the random variable X. Since for a given x the value $F_X(x)$ is a probability, the axioms (B.1) - (B.3) impose certain constraints on the probability distribution function:

- $0 \leq F_X(x) \leq 1 \quad \forall -\infty < x < \infty$

- $F_X(-\infty) = 0 \quad \text{and} \quad F_X(\infty) = 1$

- $x_2 > x_1 \quad \Rightarrow \quad F_X(x_2) \geq F_X(x_1)$

Figures B.2.a and B.2.b show the probability distribution functions of a discrete (throwing a die) and a continuous (resistance value) random variable, respectively. Note that the probability distribution of a discrete random variable is discontinuous, and that the magnitude of a jump of $F_X(x)$ at say x_0 is equal to the probability that $X = x_0$.
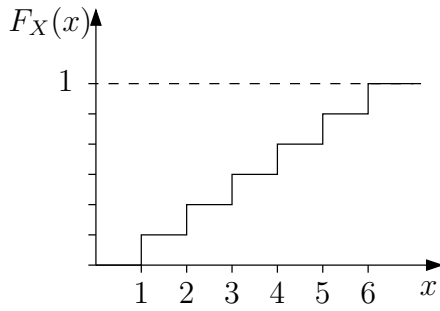
a) Discrete probability distribution   b) Continuous probability distribution
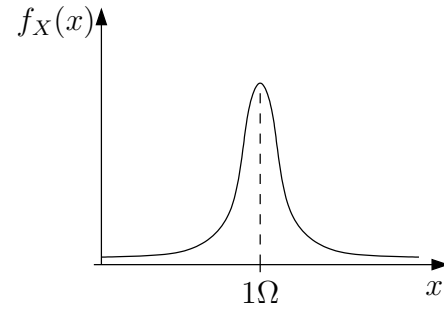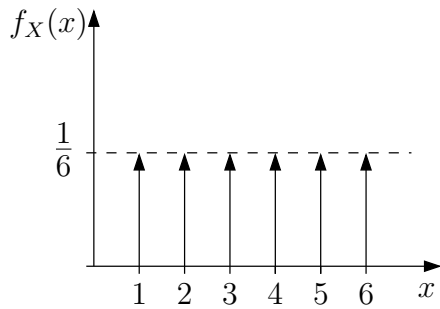
c) Discrete probability density   d) Continuous probability density

Figure B.2: Probability distribution and density, respectively, of throwing a die (a,c) and measuring a resistance (b,d)

Probability Density Function

The relative-frequency approach tells us that - assuming all outcomes are equally likely when a die is thrown - the probability of the die showing the value 3 is 1/6. On the other hand, the probability of selecting a resistor from the bin whose resistance has the exact value of 1 Ω is zero (because this is just one of infinitely many possible values), whereas the probability of a value between 0.99 and 1.01 Ω is nonzero. These observations are expressed more clearly by the *probability density function $f_X(x)$*, which is defined as the derivative of the probability distribution function

$$f_X(x) = \frac{dF_X(x)}{dx} \tag{B.8}$$

The probability density function has the properties

- $f_X(x) \geq 0 \quad \forall -\infty < x < \infty$

- $\int_{-\infty}^{\infty} f_X(x)dx = 1$

The probability of the event that X takes a value in the interval $[x_1, x_2]$ can be expressed both in terms of the probability distribution function and the probability density function, we have

$$P(x_1 < X \leq x_2) = F_X(x_2) - F_X(x_1) = \int_{x_1}^{x_2} f_X(x)dx$$

Taking the limit $x_2 \to x_1$ we obtain

$$P(x - dx < X \leq x) = f_X(x)dx$$

Figures B.2.c and B.2.d show the probability density functions of the random variables associated with throwing a die and selecting a resistor. Since the probability density function of a discrete random variable is discontinuous, the derivative does strictly speaking not exist. However, a reasonable way of handling this difficulty is to represent the derivative at a point of discontinuity by a delta function of area equal to the magnitude of the jump.

Joint Probability

Random experiments may involve two or more random variables. We can define the joint probability distribution function of two random variables X and Y as

$$F_{XY}(x, y) = P(X \leq x, Y \leq z) \tag{B.9}$$

and the joint probability density function as

$$f_{XY}(x, y) = \frac{\partial^2 F_{XY}(x, y)}{\partial x \partial y} \tag{B.10}$$

With these definitions, we have

$$P(x_1 < X \leq x_2, y_1 < Y \leq y_2) = \int_{y_1}^{y_2} \int_{x_1}^{x_2} f_{XY}(x, y)dxdy \tag{B.11}$$

and

$$P(x - dx < X \leq x, y - dy < Y \leq y) = f_{XY}(x, y)dxdy$$

If we integrate over the entire sample space, we obtain

$$F_{XY}(\infty, \infty) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{XY}(x, y)dxdy = 1$$

In the same way the concept of marginal probability was introduced, we define the probability distribution function of X irrespective of the value Y takes as the *marginal probability distribution function*

$$F_X(x) = F_{XY}(x, \infty)$$

From (B.9) and (B.11), it follows that

$$F_X(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{x} f_{XY}(x', y)dx'dy$$

and since

$$f_X(x) = \frac{dF_X(x)}{dx}$$

we obtain

$$f_X(x) = \int_{-\infty}^{\infty} f(x, y) dy$$

A similar result can be shown for $f_Y(y)$. We say that the random variables X and Y are independent if and only if

$$F_{XY}(x, y) = F_X(x)F_Y(y)$$

which also implies

$$f_{XY}(x, y) = f_X(x)f_Y(y)$$

Expectation, Mean Values and Moments

The probability distribution function and the probability density function each contain a complete characterization of a random variable. When solving practical problems, however, it is often sufficient and more convenient to work only with a partial description of a random variable in terms of statistical averages, or mean values.

To introduce the idea, consider first a discrete random variable X with possible values $\{x_1, x_2, \ldots, x_\mu\}$, and associated probabilities $\{P_1, P_2, \ldots, P_\mu\}$. The statistical average, or *expectation* of X is defined as

$$\mathrm{E}[X] = \sum_{j=1}^{\mu} x_j P_j \tag{B.12}$$

where E denotes the expectation operator. Thus, the expectation of X is the average of the possible values weighted with their associated probability. This quantity is also referred to as *mean value* or *first moment* of the discrete random variable X. Similarly, for a continuous random variable X the expectation is defined as

$$\mathrm{E}[X] = \int_{-\infty}^{\infty} x f_X(x) dx \tag{B.13}$$

We will use the notation $\bar{X} = \mathrm{E}[X]$ for the expectation of X. The expectation operator can be applied to functions of random variables as well. Let n be a positive integer, then

$$\mathrm{E}[X^n]$$

is called the nth moment of X. Of particular importance are the first and second moment. The first moment was already introduced as the expectation. The second moment

$$E[X^2]$$

is called the *mean-square value* of X. Subtracting $\bar{X}$ before taking powers yields the nth *central moment*

$$\mathrm{E}[(X - \bar{X})^n]$$

Whereas the first central moment is zero, the second central moment

$$\sigma_X^2 = \text{E}[(X - \bar{X})^2]$$

is called the *variance* of X. σ_X is called the *standard deviation* of X. It is left as an exercise to show that

$$\sigma_X^2 = \text{E}\left[X^2\right] - \bar{X}^2$$

Covariance and Correlation

Two joint probability measures of a pair of random variables X and Y, are the *covariance*

$$C_{XY} = \text{E}\left[(X - \bar{X})(Y - \bar{Y})\right] \tag{B.14}$$

and the *correlation coefficient*

$$\rho_{XY} = \frac{C_{XY}}{\sigma_X \sigma_Y} \tag{B.15}$$

Both are measures of the independence of X and Y. Note that the correlation coefficient is normalized such that $-1 \leq \rho_{XY} \leq 1$.

It is straightforward to show that $\rho_{XY} = 0$ if X and Y are statistically independent. Because in this case $f_{XY}(x, y) = f_X(x) f_Y(y)$, we have

$$
\begin{aligned}
C_{XY} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{X})(y - \bar{Y}) f_X(x) f_Y(y) dx dy \\
&= \int_{-\infty}^{\infty} (x - \bar{X}) f_X(x) dx \int_{-\infty}^{\infty} (y - \bar{Y}) f_Y(y) dy \\
&= (\bar{X} - \bar{X})(\bar{Y} - \bar{Y}) = 0
\end{aligned}
$$

Two random variables X and Y are said to be *uncorrelated* if $\rho_{XY} = 0$. Note that while statistically independent random variables are always uncorrelated, the converse is not necessarily true, unless their probability distribution is Gaussian.

Gaussian Random Variables and Central Limit Theorem

The most important probability distribution function we encounter in practice is the *Gaussian distribution*. This is due to a remarkable phenomenon, expressed by the famous *central-limit theorem*. The central limit theorem states that given N independent random variables, the sum of these random variables will have a probability distribution that converges towards a Gaussian distribution as $N \to \infty$, even if the distributions of the individual variables are not Gaussian. Many random phenomena result from a combination of a great number of individual processes that can be represented each by a random variable. Regardless of the probability distributions of these individual variables (which typically are not even known), the observed phenomena display a Gaussian distribution. A random variable with Gaussian distribution is called a Gaussian random variable.

a) Probability density function   b) Probability distribution function
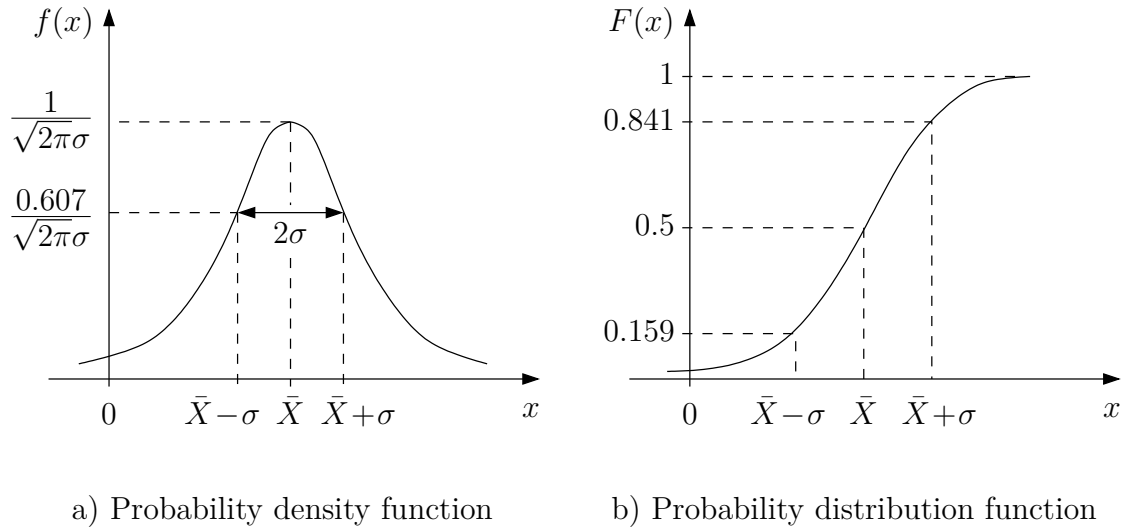
Figure B.3: Gaussian probability density and distribution

The probability density function of a Gaussian random variable X is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left[\frac{-(x - \bar{X})^2}{2\sigma_X^2}\right], \quad -\infty < x < \infty \tag{B.16}$$

where $\bar{X}$ and σ_X^2 are the mean value and the variance, respectively, of X. The Gaussian distribution function cannot be expressed in closed form; both Gaussian density and distribution function are shown in Figure B.3.

Sum of Independent Random Variables

A frequently encountered problem is: given two statistically independent random variables X and Y with known probability density functions $f_X(x)$ and $f_Y(y)$, respectively, what is the probability density function of $Z = X + Y$? It can be shown that the resulting density function is the *convolution* of the individual density functions

$$f_Z(z) = f_X(x) * f_Y(y) = \int_{-\infty}^{\infty} f_X(z - y)f_Y(y)dy \tag{B.17}$$

In Exercise B.4, this result is used to illustrate the central limit theorem.

An important fact concerning Gaussian random variables is that any linear combination of Gaussian random variables, independent or not, is also Gaussian.

B.2 Stochastic Processes

The intuitive interpretation of probability via the relative-frequency approach is based on the idea of repeating random experiments many times, the implication being that the replication process is carried out sequentially in time. In many cases, including control applications, we are interested in *signals* of random shape. In terms of the relative-frequency approach, the outcome of a random experiment then depends on time as a parameter (sometimes, space may also be a parameter). In this section we discuss ways of characterizing such random experiments. We will distinguish between a *probabilistic description* of a process, which means a description in terms of probability and joint probability density functions, and a *statistical* description in terms of mean values and moments.

To visualize the concept of a stochastic process, consider a binary waveform generator whose output switches randomly between +1 and -1 in given time intervals as shown in Figure B.4.
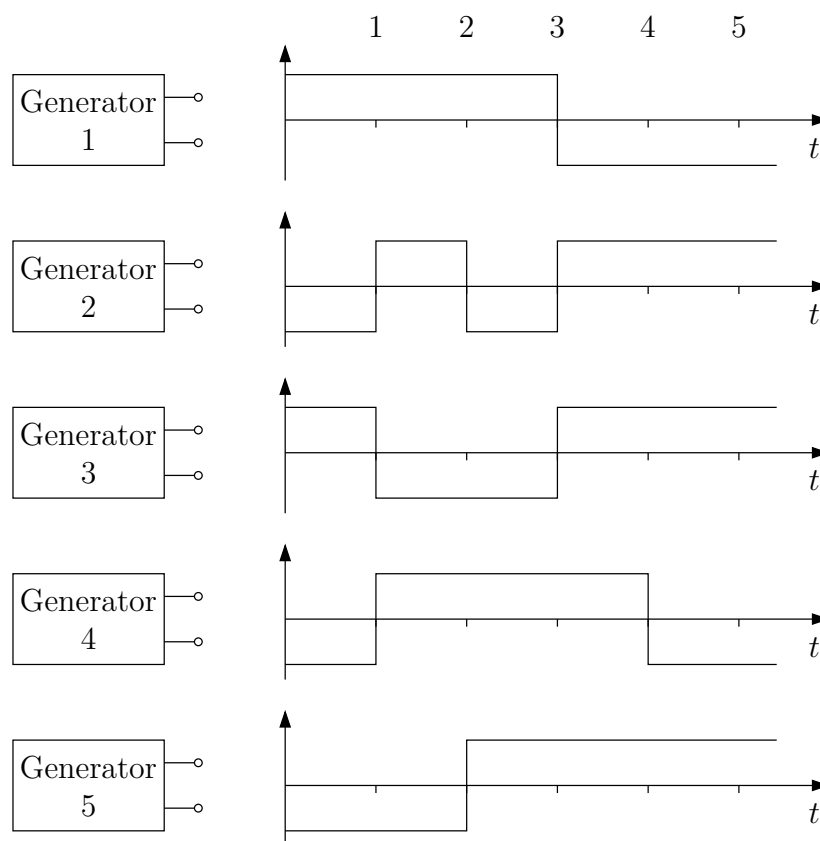

Figure B.4: Binary wave generators

Let X be a random variable that takes on the value of the generator output during a given time interval. Using the relative-frequency approach, we might then estimate the

probability $P(X = +1)$ as the number of times where $X = 1$, divided by the number N of observed time intervals, where N is a large number. In this case, the output values are observed sequentially in time. An alternative - and more useful - way of interpreting the probability $P(X = +1)$ is to assume that we have N identical random generators, and observe their outputs simultaneously in time. The relative frequency is then the number of generators with output $+1$ at a given time, divided by the number N of generators. This is illustrated in Figure B.4. An important advantage of the latter approach is that it is able to account for changes of the statistical properties of the generators over time (e.g. aging of the random generators).

In the same way as in Figure B.4, we can imagine performing any random experiment many times simultaneously. Another example is shown in Figure B.5: a random variable is used to represent the voltage at the terminal of a noise generator. We can in fact define two random variables $X(t_1)$ and $X(t_2)$ to represent the voltage at time t_1 and t_2, respectively. The outcome of a particular experiment is then a waveform $x(t)$. The entire collection of possible waveforms $\{x(t)\}$ is called an *ensemble*, and a particular outcome $x(t)$ is called a *sample function* of the ensemble. The underlying random experiment is called a *stochastic process*. The difference between a random variable and a stochastic process is that for a random variable an outcome in the probability space is mapped into a number, whereas for a stochastic process it is mapped into a function of time.
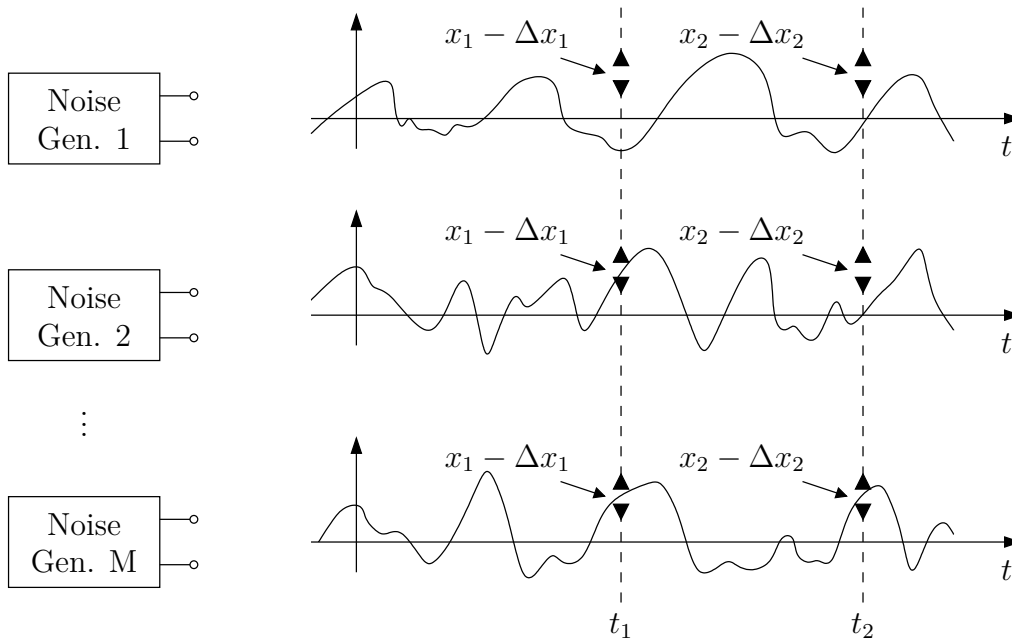

Figure B.5: Noise generators

Stationary Stochastic Processes

Following the above approach to consider the probability in terms of an ensemble of possible outcomes, we can now define marginal and joint probability density functions of the random variables $X(t_1)$ and $X(t_2)$ in Figure B.5. We can for example consider the joint probability density of the event that $x_1 - \Delta < X(t_1) \le x_1$ and $x_2 - \Delta < X(t_2) \le x_2$, as indicated in the Figure.

As mentioned before, the statistical properties and therefore the probability density functions of the underlying stochastic process may be changing with time. If all marginal and joint density functions of the process are independent of the choice of time origin, the process is said to be *stationary*. If any of the probability density functions do change with time, the process is *non-stationary*. A slightly relaxed assumption is that the mean value of any random variable $X(t_1)$ is independent of the choice of t_1, and that the correlation of two random variables $E[X(t_1)X(t_2)]$ depends only on the time difference $t_2 - t_1$. A process that satisfies these two conditions is said to be *stationary in the wide sense*. For a wide-sense-stationary process, mean value, variance and correlation coefficients between any pair of random variables are constant, independent of the choice of time origin. If the process has a Gaussian distribution, then wide-sense stationarity is equivalent to stationarity in the strict sense.

The analysis of stationary and wide-sense stationary processes is considerably simpler that that of non-stationary processes. Therefore, the assumption of wide-sense-stationarity is usually made in practical applications.

Ergodic Processes

It is possible that almost every member $x(t)$ of the ensemble of outcomes of a given stationary stochastic process $X(t)$ has the same statistical properties as the whole ensemble. In this case, it is possible to determine the statistical properties by examining only one sample function. A process having this property is said to be *ergodic*. For an ergodic process, time and ensemble averages are interchangeable: we can determine mean values and moments by taking time averages as well as ensemble averages. For example, we have for the nth moment

$$E[X^n(t)] = \int_{-\infty}^{\infty} x^n(t) f_X(x(t)) dx(t) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x^n(t) dt$$

The first integral represents the average over the ensemble of outcomes at a given time t, whereas the second integral represents the average over time. We will use the notation

$$\langle x(t) \rangle = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t) dt$$

to denote the time average of a sample function $x(t)$. The above can therefore be written as

$$E[X^n(t)] = \langle x^n(t) \rangle$$

We also have

$$\sigma_X^2 = E[(X(t) - \bar{X})^2] = \langle [X(t) - \langle X(t) \rangle]^2 \rangle$$

Since a time average cannot be a function of time, it is clear from the above that an ergodic process must be stationary - all non-stationary processes are non-ergodic. On the other hand, stationarity does not necessarily imply ergodicity - it is possible for a stationary process to be non-ergodic. Even though it is generally difficult to give physical reasons for this, it is customary to assume ergodicity in practical applications unless there are compelling physical reasons for not doing so.

Covariance and Autocorrelation Function

The concepts of covariance and correlation introduced in the previous section can be extended to provide a statistical description of a stochastic process. For this purpose, for a given process we consider the values $X(t_1)$ and $X(t_2)$ taken at time t_1 and t_2, respectively, as two random variables, and consider their covariance

$$\begin{aligned} C(t_1, t_2) &= E[X(t_1) - \overline{X(t_1)}][X(t_2) - \overline{X(t_2)}] \\ &= E[X(t_1)X(t_2)] - \overline{X(t_1)}\,\overline{X(t_2)} \end{aligned}$$

The first term on the right hand side is called the *autocorrelation function*

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] \tag{B.18}$$

of the process $X(t)$.

For ergodic processes, covariance and autocorrelation are independent of the choice of time origin, and we can write

$$R_X(\tau) = E[X(t)X(t + \tau)] = \langle x(t)x(t + \tau) \rangle$$

where $x(t)$ is any sample function of $X(t)$. Note that the time average on the right hand side is identical with the definition of the *time autocorrelation function* of a deterministic power signal.

It is straightforward to show that the autocorrelation function $R_X(\tau)$ of an ergodic process has the following properties, see Exercise B.5.

1. $|R_X(\tau)| \leq R_X(0) \quad \forall \tau.$

2. $R_X(-\tau) = R_X(\tau) \quad \forall \tau.$

3. $\lim_{|\tau| \to \infty} R_X(\tau) = \bar{X}^2$ if $X(t)$ does not contain a periodic component.

4. $R_X(\tau)$ has a periodic component if $X(t)$ has a periodic component.

Crosscorrelation Function

Consider two random processes $X(t)$ and $Y(t)$ which are jointly stationary in the wide sense. For fixed t and τ, $X(t)$ and $Y(t+\tau)$ are two random variables, and we can define the *crosscorrelation functions*

$$R_{XY}(\tau) = E[X(t)Y(t+\tau)]$$

and

$$R_{YX}(\tau) = E[Y(t)X(t+\tau)]$$

A crosscorrelation function is not an even function of τ, however $R_{XY}(\tau) = R_{YX}(-\tau) \; \forall \tau$. There is not necessarily a maximum at $\tau = 0$, however one can show that

$$|R_{XY}(\tau)| \leq (R_X(0)R_Y(0))^{1/2}$$

If $X(t)$ and $Y(t)$ are jointly ergodic, we have

$$R_{XY}(\tau) = \langle x(t)y(t+\tau)\rangle \quad \text{and} \quad R_{YX}(\tau) = \langle y(t)x(t+\tau)\rangle$$

The time average on the right hand side in the above equations is again identical with the definition of the *time crosscorrelation function* of a deterministic power signal.

Interpretation of Statistical Averages of Ergodic Processes

Various statistical averages of ergodic processes and their physical significance are summarized below. For an intuitive interpretation of these averages, it is helpful to view a random process as a signal, e.g. represented by a voltage across the terminal of a random generator.

- The mean value $\bar{X} = \langle x(t)\rangle$ is the dc component of the signal.

- $\bar{X}^2 = \langle x(t)\rangle^2$ is the power in the dc component of the signal.

- The variance $\sigma_X^2 = E[X^2(t)] - \bar{X}^2 = \langle x^2(t)\rangle - \langle x(t)\rangle^2$ is the power in the ac component of the signal.

- The total power (the mean-square value) $E[X^2(t)] = \sigma_x^2 + \bar{X}^2$ is the sum of ac power and dc power.

Figure B.6 shows a typical example of an autocorrelation function. We can infer the following from the plot:

- The dc power is $B = R_X(\infty)$.

- The total power is $A = R_X(0)$.

- The ac power is $A - B$.

- The process does not have a periodic component.
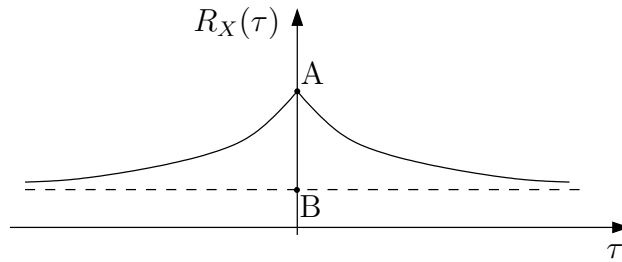
Figure B.6: Autocorrelation function

Covariance and Correlation Matrices

In many applications it will be necessary to deal with a number of random variables or stochastic processes; in this case the use of vector notation turns out to be convenient.

Consider a stochastic process sampled at periodic time instants. Each sample is a then random variable. If N such samples are considered, they can be collected in a column vector

$$\mathcal{X}^T = [X(t) \ X(t + \Delta) \ \ldots \ X(t + (N - 1)\Delta)]$$

We can now define a $N \times N$ *correlation matrix* that describes the correlation between each pair of samples as

$$\mathcal{R}_\mathcal{X} = E[\mathcal{X}\mathcal{X}^T] \tag{B.19}$$

If the process is wide-sense stationary, the correlation matrix is

$$\mathcal{R}_\mathcal{X} = \begin{bmatrix} R_X(0) & R_X(\Delta) & \ldots & R_X((N-1)\Delta) \\ R_X(\Delta) & R_X(0) & & \\ \vdots & & \ddots & \vdots \\ R_X((N-1)\Delta) & & \ldots & R_X(0) \end{bmatrix} \tag{B.20}$$

A more widely used way of representing statistical properties of the process samples is the *covariance matrix*, which - following the definition in (B.14) - is defined as

$$\Lambda_\mathcal{X} = E[(\mathcal{X} - \bar{\mathcal{X}})(\mathcal{X} - \bar{\mathcal{X}})^T] = \mathcal{R}_\mathcal{X} - \bar{X}\bar{X}^T \tag{B.21}$$

If the process is wide-sense stationary, the covariance matrix becomes

$$\Lambda_{\mathcal{X}} = \sigma_X^2 \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{N-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{N-2} \\ \rho_2 & \rho_1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \rho_1 \\ \rho_{N-1} & & & \rho_1 & 1 \end{bmatrix} \tag{B.22}$$

where ρ_i denotes the correlation coefficient defined in (B.15) between samples taken at time instants separated by $i\Delta$.

Another situation encountered in practical applications arises when the relationship between a number of different stochastic processes - say $X_1(t)$, $X_2(t)$, ..., $X_N(t)$ - is considered. The collection of processes can be represented by a single vector process

$$X^T(t) = [X_1(t) \ \ X_2(t) \ \ldots \ X_N(t)]$$

Assuming that the vector process is wide-sense stationary, its correlation matrix is

$$\mathcal{R}_X(\tau) = E[X(t)X^T(t+\tau)] = \begin{bmatrix} R_1(\tau) & R_{12}(\tau) & \cdots & R_{1N}(\tau) \\ R_{21}(\tau) & R_2(\tau) & & \\ \vdots & & \ddots & \vdots \\ R_{N1}(\tau) & & \cdots & R_N(\tau) \end{bmatrix} \tag{B.23}$$

where

$$R_i(\tau) = E[X_i(t)X_i(t+\tau)] \quad \text{and} \quad R_{ij}(\tau) = E[X_i(t)X_j(t+\tau)]$$

The covariance matrix is in this case defined by

$$\Lambda_X(\tau) = E[(X(t) - \bar{X})(X(t+\tau) - \bar{X})^T] = \mathcal{R}_X(\tau) - \bar{X}\bar{X}^T \tag{B.24}$$

Spectral Density

When dealing with deterministic signals, the transformation from time to frequency domain via Fourier and Laplace transform is known to greatly simplify the analysis of linear systems. Reasons for this simplification are that differential equations and convolution in time domain are replaced by algebraic equations and multiplication, respectively, in frequency domain. We will see that similar simplifications are possible when dealing with stochastic processes. Earlier it was shown that the autocorrelation function of an ergodic process - defined as a statistical average - is equal to the definition of the time autocorrelation function of a deterministic signal - computed as a time average. An important frequency domain concept for *deterministic* signals is the spectral density, defined as the Fourier transform of the autocorrelation function. We will see that this concept can be extended to stochastic processes.

We first recall some concepts and definitions related to a deterministic signal $x(t)$. Let

$$E = \int_{-\infty}^{\infty} x^2(t) dt$$

denote the signal energy, and

$$P = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x^2(t) dt$$

the signal power of $x(t)$. We call a given a deterministic signal $x(t)$ an *energy signal* if

$$0 < E < \infty$$

and a *power signal* if

$$0 < P < \infty$$

For energy signals, the Fourier transform

$$X(\omega) = \mathcal{F}[x(t)] = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

has the physical significance that its magnitude represents the *amplitude density* as a function of frequency. Moreover, Parseval's theorem states that the signal energy can be expressed in terms of the Fourier transform as

$$\int_{-\infty}^{\infty} x^2(t) dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega$$

This relationship motivates the definition of the *energy spectral density*

$$\Psi_X(\omega) = |X(\omega)|^2$$

which indicates how the energy of the signal $x(t)$ is distributed over frequency.

When dealing with power signals, we recall that - strictly speaking - the Fourier transform of a power signal does not exist, but that for certain power signals a *generalized Fourier transform* may be defined. For example, it is possible to define the generalized Fourier transform of a periodic signal by using delta functions to represent the amplitude densities at discrete frequencies. Similarly, if a power signal can be expressed as the sum of an energy signal and a dc component, the dc component can be represented by a delta function in the generalized Fourier transform.

For a general deterministic power signal $x(t)$, the *power spectral density* $S_X(\omega)$ is defined as the Fourier transform of its autocorrelation function $R_X(\tau)$, i.e.

$$S_X(\omega) = \mathcal{F}[R_X(\tau)] = \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega\tau} d\tau$$

The discussion earlier in this section showed that the autocorrelation itself may not be square integrable, so that its Fourier transform may not exist. However, if that is the

case, the autocorrelation function has either a periodic component or can be decomposed into a square integrable component and a dc component (see e.g. Figure B.6), so that a generalized Fourier transform containing delta functions can be used instead.

Power Spectral Density of a Stochastic Process

We now return to stochastic processes, and we assume that the processes considered here are ergodic. It is clear that the sample functions of a stationary process are power signals. An intuitive approach to defining the power spectral density of a stationary process $X(t)$ is to take a sample function $x(t)$, and consider a truncated version $x_T(t)$ defined as

$$x_T(t) = \begin{cases} x(t), & |t| < T \\ 0, & \text{otherwise} \end{cases}$$

The reason for truncating the sample function is that the Fourier transform $\mathcal{F}[x_T(t)]$ of $x_T(t)$ does exist, even if that of $x(t)$ does not. The energy spectral density of the truncated sample function is $|\mathcal{F}[x_T(t)]|^2$. The power density in the interval $[-T, \ T]$ - taken as time average - is $|\mathcal{F}[x_T(t)]|^2/2T$. Noting that $\mathcal{F}[x_T(t)]$ is the Fourier transform only of a particular sample function, we then obtain the distribution of power density over frequency of the stochastic process $X(t)$ by

i)   taking the expectation, i.e. the average over the ensemble of sample functions, and

ii)   letting $T \to \infty$.

In this way, we can define the power spectral density of $X(t)$ as

$$S_X(\omega) = \lim_{T \to \infty} \frac{1}{2T} E[|\mathcal{F}[x_T(t)]|^2] \tag{B.25}$$

Wiener-Khinchine Theorem

Note that (B.25) is one way of *defining* the spectral density of a stochastic process in terms of the Fourier transform of truncated sample functions. For *deterministic* power signals, the spectral density is defined as the Fourier transform of the autocorrelation function. It turns out that for an ergodic process, the same relationship can be derived from the definition (B.25). Thus, if $S_X(\omega)$ is the power spectral density of the ergodic process $X(t)$ as defined in (B.25), we can show that

$$S_X(\omega) = \mathcal{F}[R_X(\tau)] = \int_{-\infty}^{\infty} R_X(\tau)e^{-j\omega\tau}d\tau \tag{B.26}$$

This result is known as the *Wiener-Khinchine Theorem.* The Wiener-Khinchine Theorem plays an important role in the analysis of stochastic processes, because it provides the link between time domain and frequency domain. To prove (B.26), note that

$$|\mathcal{F}[x_T(t)]|^2 = \left| \int_{-T}^{T} x(t)e^{j\omega t}dt \right|^2 = \int_{-T}^{T}\int_{-T}^{T} x(t)x(\tau)e^{j\omega(t-\tau)}dtd\tau$$

Taking the ensemble average, and changing the order of averaging and integration, we obtain

$$E\left[|\mathcal{F}[x_T(t)]|^2\right] = \int_{-T}^{T}\int_{-T}^{T} E[x(t)x(\tau)e^{j\omega(t-\tau)}]dtd\tau$$

$$= \int_{-T}^{T}\int_{-T}^{T} R_X(t-\tau)e^{j\omega(t-\tau)}dtd\tau$$

Now a change of variables

$$u \to t - \tau \quad \text{and} \quad v \to t$$

is used to obtain

$$E\left[|\mathcal{F}[x_T(t)]|^2\right] = 2T\int_{-2T}^{2T}\left(1 - \frac{|u|}{2T}\right)R_X(u)e^{j\omega u}du$$

Substituting this in (B.25) and taking the limit as $T \to \infty$ yields (B.26).

Properties of the Power Spectral Density

Since the Fourier transform of a real, even function is itself real and even, it follows from (B.26) that the power spectral density is a real and even function of frequency. Moreover, from (B.25) it is clear that the power spectral density is positive for all ω.

Using Parseval's Theorem and (B.25) one can show (see Exercise B.6) that

$$E[x^2(t)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(\omega)d\omega$$

Thus, the mean-square value (the total power) of a process is proportional to the area of the power spectral density.

A frequently encountered class of power spectral densities is characterized by being rational functions of ω. Since the functions are even in ω, only even powers are involved; thus

$$S_X(\omega) = S_0 \frac{\omega^{2m} + b_{2m-2}\omega^{2m-2} + \ldots + b_2\omega^2 + b_0}{\omega^{2n} + a_{2n-2}\omega^{2n-2} + \ldots + a_2\omega^2 + a_0}$$

Note that a finite mean-square value (total power) of the process requires $m < n$. If a process has a dc or periodic component, there will be delta impulses $\delta(\omega \pm \omega_0)$ present in the power spectral density, where ω_0 is the frequency of the periodic component, or zero for a dc component.

Spectral densities have been expressed so far as functions of the angular frequency ω. When analyzing linear systems, it is often convenient to use the complex variable s instead. This can be done by replacing ω by $-s$ or ω^2 by $-s^2$, yielding $S_X(-s)$ instead of $S_X(\omega)$. Since this notation is somewhat clumsy, we will simply write $S_X(s)$, keeping in mind that this and $S_X(\omega)$ are not the same function of their respective arguments. Note however that for rational spectral densities, where only even powers are involved, the two are equivalent.

Cross-Power Spectral Density

Given two stationary random processes $X(t)$ and $Y(t)$, we define their *cross-power spectral densities* as

$$S_{XY}(\omega) = \mathcal{F}[R_{XY}(\tau)] \quad \text{and} \quad S_{YX}(\omega) = \mathcal{F}[R_{YX}(\tau)]$$

where $R_{XY}(\tau)$ and $R_{YX}(\tau)$ are the cross-correlation functions of the two processes.

In contrast to the power spectral density, a cross spectral density needs not be real, positive or an even function of ω. One can however show that

- $S_{XY}(\omega) = S_{YX}^*(\omega)$, where * denotes the complex conjugate.

- The real parts of cross spectral densities are even in ω.

- The imaginary parts of cross spectral densities are odd in ω.

B.3   Systems with Stochastic Inputs

The mathematical representations of stochastic processes introduced so far can be used to study the response of linear systems to input signals that are modelled as stochastic processes rather than deterministic functions of time. In control applications, we are often interested in the effect of unknown, random disturbances on a control loop. Examples for disturbances that can be modelled in a probabilistic way are noise effects, wind gusts, the effect of waves, or the *a priori* unknown command inputs generated by a human operator.

White Noise

When studying linear systems with deterministic inputs, certain input signals play a prominent role in analysis, the most important one being the unit delta impulse $\delta(t)$. When dealing with stochastic inputs, a role similar to that of the delta impulse is played by a particular stochastic process referred to as *white noise*. A stochastic process is called white noise if its spectral density is constant over all frequencies, i.e.

$$S_X(\omega) = S_0$$

Since the power spectral density is the Fourier transform of the autocorrelation function, it is clear that the autocorrelation function of a white noise process is a delta function

$$R_X(\tau) = S_0 \delta(\tau) \tag{B.27}$$

and therefore

$$S_X(\omega) = \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega\tau} d\tau = \int_{-\infty}^{\infty} S_0 \delta(\tau) e^{-j\omega\tau} d\tau = S_0$$

An intuitive interpretation of this autocorrelation function is that no matter how close together we sample a white noise process, the samples will be uncorrelated. If the process is also Gaussian, the samples are independent.

Since a white noise process has infinite total power, it is only a fictitious concept and not physically realizable. White noise is nevertheless an extremely useful concept. In practice, white noise is often replaced by *bandlimited white noise*, defined as

$$S_X(\omega) = \begin{cases} S_0, & |\omega| < 2\pi W \\ 0, & \text{otherwise} \end{cases}$$

where W is called the bandwidth of the process. Power spectral density and autocorrelation function of this process are shown in Figure B.7. Even though the total power is finite (it is $2WS_0$), this process is also not realizable because physical processes cannot have flat spectral density functions. It can however be approached arbitrarily close, and its usefulness comes from the fact that when used as input to a system with a bandwidth much smaller than W, the response will be very close to that obtained with unlimited white noise as input.

a) Power spectral density   b) autocorrelation function
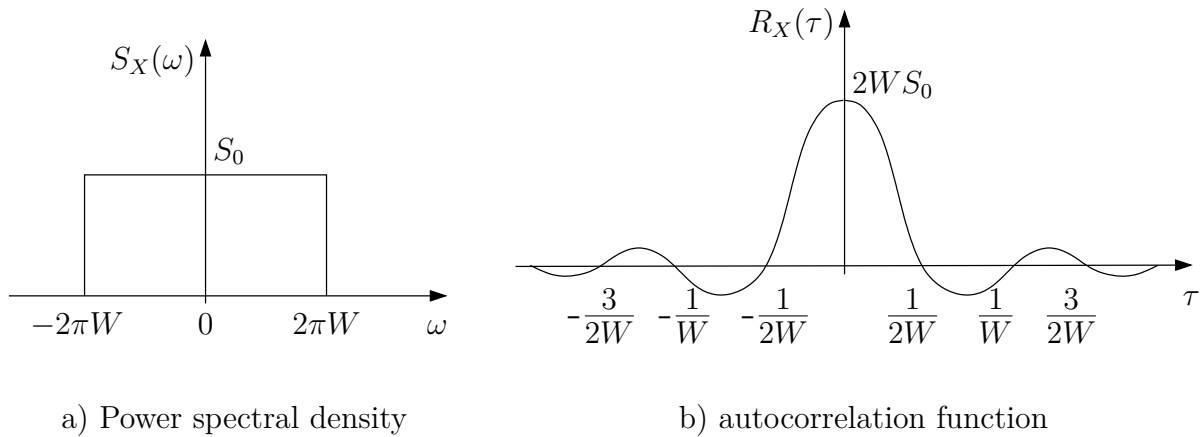
Figure B.7: Power spectral density and autocorrelation function of bandlimited white noise

In control applications, we frequently encounter white noise vector processes

$$X(t) = [X_1(t) \ X_2(t) \ \dots \ X_N(t)]^T$$

If these processes are wide-sense stationary, mutually uncorrelated and have zero mean,
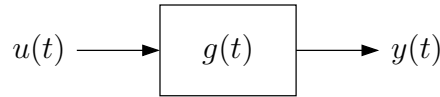
Figure B.8: Linear system

then the covariance matrix defined in (B.24) takes the form

$$\Lambda_X(\tau) = \begin{bmatrix} S_1 & 0 & \dots & 0 \\ 0 & S_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & & & S_N \end{bmatrix} \delta(\tau) \tag{B.28}$$

where S_i denotes the spectral density of the ith process.

Gaussian Processes

The importance of the Gaussian distribution was already discussed earlier. It turns out that Gaussian processes are not only realistic models of many real-life processes, they are also very useful when analyzing linear systems. Whereas the former is due to the central limit theorem, the latter is due to the fact that if the input signal to a linear system is a Gaussian process, the output will also be Gaussian. Thus, when all inputs are Gaussian, all signals in the system under consideration will have that property. Gaussian processes therefore play a role similar to that of sinusoidal signals for steady state analysis.

Time Domain Analysis

Consider a linear time-invariant system with impulse response $g(t)$ as shown in Figure B.8. We assume that the system is stable and has no poles on the imaginary axis. For a given input signal $u(t)$, the output is

$$y(t) = \int_0^\infty u(t - \tau)g(\tau)d\tau \tag{B.29}$$

If the input is a random signal that can be modelled as a stochastic process $U(t)$, we write

$$Y(t) = \int_0^\infty U(t - \tau)g(\tau)d\tau$$

This equation defines the output as a stochastic process $Y(t)$, whose sample functions $y(t)$ are generated by the sample functions $u(t)$ of the input process according to (B.29). In the rest of this section we will assume that the stochastic processes are stationary or wide-sense stationary.

Mean Value of Output

The mean value of the output is given by

$$\bar{Y} = E[Y(t)] = E\left[\int_0^\infty U(t-\tau)g(\tau)d\tau\right]$$

Changing the order of integration and ensemble average, we obtain

$$\bar{Y} = \int_0^\infty E[U(t-\tau)]g(\tau)d\tau = \bar{U}\int_0^\infty g(\tau)d\tau$$

Observing that the integral of the impulse response is the static gain, we conclude that

$$\bar{Y} = G(0)\bar{U}$$

where $G(s)$ denotes the transfer function of the system. Thus, the mean value of the output is equal to the mean value of the input times the static gain. In particular, if the input has zero mean the output will also have zero mean.

Mean-Square Value of Output

The mean-square value of the output can be obtained from

$$E[Y^2(t)] = E\left[\int_0^\infty U(t-\tau_1)g(\tau_1)d\tau_1\int_0^\infty U(t-\tau_2)g(\tau_2)d\tau_2\right]$$

$$= \int_0^\infty\int_0^\infty E[U(t-\tau_1)U(t-\tau_2)]g(\tau_1)g(\tau_2)d\tau_1d\tau_2$$

Since the expectation on the right hand side is the autocorrelation function of the input, we have

$$E[Y^2(t)] = \int_0^\infty\int_0^\infty R_U(\tau_2-\tau_1)g(\tau_1)g(\tau_2)d\tau_1d\tau_2 \tag{B.30}$$

For the special case of a white noise input process with spectral density S_0, substituting (B.27) in the above yields

$$E[Y^2(t)] = S_0\int_0^\infty g^2(\tau)d\tau$$

provided the integral on the right hand side exists.

Autocorrelation Function of Output

To find the autocorrelation of the system output, we consider

$$R_Y(\tau) = \int_0^\infty\int_0^\infty E[Y(t-\lambda_1)Y(t-\lambda_2+\tau)]g(\lambda_1)g(\lambda_2)d\lambda_1d\lambda_2$$

$$= \int_0^\infty\int_0^\infty R_U(\lambda_2-\lambda_1-\tau)g(\lambda_1)g(\lambda_2)d\lambda_1d\lambda_2 \tag{B.31}$$

For $\tau = 0$, this reduces to (B.30).

If the input is a white noise process, then

$$R_U(\tau) = S_0 \delta(\tau)$$

and we obtain

$$R_Y(\tau) = S_0 \int_0^\infty g(\lambda)g(\lambda + \tau)d\lambda$$

Crosscorrelation Between Input and Output

To find the crosscorrelation between input and output, we consider

$$\begin{aligned}
R_{UY}(\tau) &= E[U(t)Y(t + \tau)] \\
&= E\left[U(t) \int_0^\infty U(t + \tau - \lambda)g(\lambda)d\lambda\right] \\
&= \int_0^\infty E[U(t)U(t + \tau - \lambda)]g(\lambda)d\lambda
\end{aligned}$$

and thus

$$R_{UY}(\tau) = \int_0^\infty R_U(\tau - \lambda)g(\lambda)d\lambda \tag{B.32}$$

Similarly we obtain

$$R_{YU}(\tau) = \int_0^\infty R_U(\tau + \lambda)g(\lambda)d\lambda$$

If the input is a white noise process with $R_U(\tau) = S_U\delta(\tau)$, then (B.32) simplifies to

$$R_{UY}(\tau) = \begin{cases} S_U\, g(\tau), & \tau \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

This result provides a way of determining the impulse response of a linear system experimentally by applying white noise at the input and computing the crosscorrelation between input and output.

Frequency Domain Analysis

The power spectral density indicates how the signal power of a stochastic process is distributed over frequency. We assume again that the processes under consideration are stationary or wide-sense stationary. To find the power spectral density of the output, we use (B.31) and (B.26) to obtain

$$\begin{aligned}
S_Y(\omega) &= \mathcal{F}[R_Y(\tau)] \\
&= \int_{-\infty}^\infty \left(\int_0^\infty \int_0^\infty R_U(\lambda_2 - \lambda_1 - \tau)g(\lambda_1)g(\lambda_2)d\lambda_1 d\lambda_2\right)e^{j\omega\tau}d\tau
\end{aligned}$$

Changing the order of integration, we find

$$
\begin{aligned}
S_Y(\omega) &= \int_0^\infty \int_0^\infty g(\lambda_1)g(\lambda_2) \int_{-\infty}^\infty R_U(\lambda_2 - \lambda_1 - \tau)e^{j\omega\tau}d\tau d\lambda_1 d\lambda_2 \\
&= \int_0^\infty \int_0^\infty g(\lambda_1)g(\lambda_2)S_U(\omega)e^{j\omega(\lambda_2-\lambda_1)}d\lambda_1 d\lambda_2 \\
&= S_U(\omega) \int_0^\infty g(\lambda_1)e^{j\omega\lambda_1}d\lambda_1 \int_0^\infty g(\lambda_2)e^{j\omega\lambda_2}d\lambda_2 \\
&= S_U(\omega)G(\omega)G(-\omega) = S_U(\omega)|G(\omega)|^2
\end{aligned}
$$

where $G(\omega)$ is the Fourier transform of $g(\lambda)$. Since we assumed that the system under consideration is stable and has no poles on the imaginary axis, we can replace the Fourier transform $G(\omega)$ by the Laplace transform $G(s)$ to obtain

$$
S_Y(s) = G(s)G(-s)S_U(s) \tag{B.33}
$$

Note however that $S_Y(s)$ and $S_U(s)$ are not the same functions of their arguments as $S_Y(\omega)$ and $S_U(\omega)$ unless the spectral densities are rational.

The relationship (B.33) shows that the term $G(s)G(-s)$ plays the same role in relating stochastic input and output processes, as does the transfer function $G(s)$ in relating deterministic input and output signals. Note however that we assumed that the input process is stationary (or wide-sense stationary). For non-stationary processes, the above results do not apply in general.

Cross-Spectral Density Between Input and Output

For completeness, we briefly state the following results about the cross-spectral density between input and output. One can show that

$$
S_{UY}(s) = G(s)S_U(s)
$$

and

$$
S_{YU}(s) = G(-s)S_U(s)
$$

Exercises

Problem B.1

Using the axioms (B.1) - (B.3), prove that

a)   the probability of the impossible event is zero, i.e. $P(\emptyset) = 0$

b)   the joint probability of any two events A and B is given by

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Problem B.2

Show that

$$\sigma_X^2 = \mathrm{E}\left[X^2\right] - \bar{X}^2$$

Problem B.3

Let the random variable X have the uniform probability density function

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}$$

a)   What is the mean value $\mathrm{E}[X]$ in terms of a and b?

b)   What is the variance $\mathrm{E}^2[X]$ in terms of a and b?

c)   Compute the variance for the special cases

   i) $a = 1$ and $b = 2$

   ii) $a = 0$ and $b = 2$.

Problem B.4

Consider the sum of four independent random variables

$$Z = X_1 + X_2 + X_3 + X_4$$

with the identical probability distribution functions

$$f_{X_i}(x) = \begin{cases} 1, & |x| \leq \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$$

for $i = 1,\ 2,\ 3,\ 4$. Use (B.17) and Matlab to plot $f_Z(z)$. Plot also the Gaussian probability density function with zero mean and $\sigma^2 = 1/3$, and compare both functions.

Problem B.5

Show that the autocorrelation function (B.18) has the four properties listed in Section B.2.

Problem B.6

Use Parseval's Theorem and (B.25) to show that

$$E[x^2(t)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(\omega)d\omega$$

Appendix C

Solutions to Exercises

C.1 Chapter 2

Solution to Problem q:RLC   State space representations

a) From the differential equations

$$i = C\frac{\mathrm{d}v_c}{\mathrm{d}t} \qquad v_l = L\frac{\mathrm{d}i}{\mathrm{d}t}$$

with $x_1 = i$; $x_2 = v_c$; $u = v_s$

$$\dot{x}_1 = \frac{1}{L}(u - Rx_1 - x_2)$$
$$\dot{x}_2 = \frac{1}{C}x_1$$
$$v_r = Rx_1$$

$$A = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1/L \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} R & 0 \end{bmatrix}$$

b) From the physical relationships

$$i = C\frac{\mathrm{d}v_c}{\mathrm{d}t} \qquad \frac{\mathrm{d}i}{\mathrm{d}t} = C\frac{\mathrm{d}^2 v_c}{\mathrm{d}t^2}$$

$$v_l = L\frac{\mathrm{d}i}{\mathrm{d}t} = LC\frac{\mathrm{d}^2 v_c}{\mathrm{d}t^2}$$
$$v_r = iR = RC\frac{\mathrm{d}v_c}{\mathrm{d}t}$$
$$v_s = v_c + v_r + v_l$$
$$v_s = v_c + RC\frac{\mathrm{d}v_c}{\mathrm{d}t} + LC\frac{\mathrm{d}^2 v_c}{\mathrm{d}t^2}$$

c) According to the differential equation which is derived in part (b), by defining state variables $x_1 = v_c$, $x_2 = \mathrm{d}v_c/\mathrm{d}t$ and also v_s as the input, we have

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \frac{1}{LC}(-RCx_2 - x_1 + u)$$

Then a state space model can be formed as

$$A = \begin{bmatrix} 0 & 1 \\ -1/LC & -R/L \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \frac{1}{LC} \end{bmatrix}$$

$$v_r = RC\dot{v}_c = RCx_2$$
$$C = \begin{bmatrix} 0 & RC \end{bmatrix}$$

Solution to Problem q:cal-ham-thrm   Cayley-Hamilton Theorem

a) The characteristic equation is true for all eigenvalues of A, $\lambda_1 \ldots \lambda_n$

$$\lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_0 = 0$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & \lambda_n \end{bmatrix}$$

so

$$\Lambda^n + a_{n-1}\Lambda^{n-1} + \ldots + a_0 I = 0$$

This is the matrix characteristic equation.

b) With distinct eigenvalues and diagonal Λ we have

$$A = T\Lambda T^{-1}$$
$$A^2 = T\Lambda T^{-1}T\Lambda T^{-1} = T\Lambda^2 T^{-1}$$
$$\vdots$$
$$A^m = T\Lambda^m T^{-1}$$

Multiply the matrix characteristic equation by T (left) and T^{-1} (right) to obtain

$$T\Lambda^n T^{-1} + a_{n-1}T\Lambda^{n-1}T^{-1} + \ldots + a_0 TT^{-1} = 0$$

$$A^n + a_{n-1}A^{n-1} + \ldots + a_0 I = 0$$

Solution to Problem q:phaseplane   Phase plane diagrams

a) By eigenvalue decomposition

$$A = T\Lambda T^{-1}, \qquad T = [t_1 \ t_2], \qquad \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

so

$$(sI - A)X(s) = x(0), \qquad T(sI - \Lambda)T^{-1}X(s) = x(0)$$

$$X(s) = T(sI - \Lambda)^{-1}T^{-1}x(0) = T \begin{bmatrix} \frac{1}{s-\lambda_1} & 0 \\ 0 & \frac{1}{s-\lambda_2} \end{bmatrix} T^{-1}x(0)$$

It is worthy to observe that we could not investigate the dynamic of a system with its transfer function when it is affected only by some none-zero initial condition, while this possibility is provided by state space model.

b) With initial conditions $x(0) = kt_1$ we have

$$T^{-1}x(0) = \begin{bmatrix} k \\ 0 \end{bmatrix} \qquad (T^{-1}[t_1 \ t_2] = I \Rightarrow \begin{bmatrix} T^{-1}t_1 & T^{-1}t_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$$

$$X(s) = k[t_1 \ t_2] \begin{bmatrix} \frac{1}{s-\lambda_1} \\ 0 \end{bmatrix} = k \frac{1}{s - \lambda_1} t_1$$

The solution is in the direction t_1 and only depends on λ_1.

c) From the result of part (b) for any initial condition of the form $x(0) = k_1 t_1 + k_2 t_2$, the solution x in frequency domain is

$$X(s) = \frac{k_1}{s - \lambda_1} t_1 + \frac{k_2}{s - \lambda_2} t_2$$

and in time domain

$$x(t) = k_1 e^{\lambda_1 t} t_1 + k_2 e^{\lambda_2 t} t_2$$

By eigenvalue decomposition of A with the MATLAB command `eig`, we obtain

$$\begin{bmatrix} t_1 & t_2 \end{bmatrix} = \begin{bmatrix} 0.7071 & -0.4472 \\ -0.7071 & 0.8944 \end{bmatrix}, \qquad \lambda_1 = -2, \quad \lambda_2 = -3$$

We choose eigenvectors in a way to have $\begin{bmatrix} t_1 & t_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$; note that the eigenvectors are not unique and only their directions matter.

As t_1 and t_2 are linearly independent, any arbitrary vector like $x(0) = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$, can be written as linear combination of t_1 and t_2. It is straightforward to calculate $k_1 = 1$ and $k_2 = 2$ so that we have $x(0) = t_1 + 2t_2$ and the solution is

$$x(t) = e^{-2t} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + 2e^{-3t} \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

MATLAB: see `cs_phaseplane.m`

Solution to Problem q:matrixexponential   Characteristic Polynomial

a) If the matrix A has eigenvalues $\lambda = \lambda_1, \lambda_2, \ldots, \lambda_n$, with the characteristic polynomial

$$\lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_0 = 0$$

the eigenvalues of At are λt and are solutions of

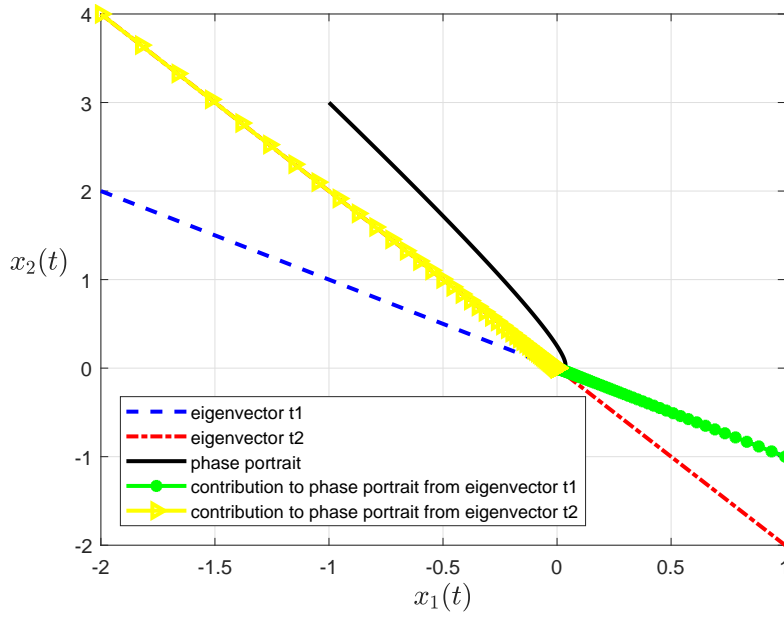$$(\lambda t)^n + (a_{n-1}t)(\lambda t)^{n-1} + \ldots + a_0 t^n = 0$$

Figure C.1: behaviour of the system in a phase plane diagram

The expansion of $e^{\lambda t}$ is

$$e^{\lambda t} = 1 + \lambda t + \frac{1}{2}(\lambda t)^2 + \ldots + \frac{1}{k!}(\lambda t)^k \qquad k \to \infty$$

Using the characteristic polynomial, the term $(\lambda t)^n$ can be replaced by an expression where the highest power of (λt) is $n - 1$

$$(\lambda t)^n = -\Big((a_{n-1}t)(\lambda t)^{n-1} + \ldots + a_0 t^n\Big) = E_{n-1}\Big((\lambda t)^{n-1}\Big)$$

Similarly $(\lambda t)^{n+1}$ is

$$(\lambda t)^{n+1} = E_n((\lambda t)^n) = E_n(E_{n-1}((\lambda t)^{n-1}))$$

All further terms can be replaced by expressions where the highest power of λt is $n - 1$, so

$$e^{\lambda t} = \alpha_0(t) + \alpha_1(t)\lambda + \ldots + \alpha_{n-1}(t)\lambda^{n-1}$$

The $\alpha's$ are functions of t because each coefficient of $(\lambda t)^k$ in the expressions E involves t.

b) From the definition of the matrix exponential we have

$$e^{At} = I + At + \frac{1}{2!}A^2 t^2 + \ldots$$

As we know that if $A = T\Lambda T^{-1}$ then $A^n = T\Lambda^n T^{-1}$(why?) which Λ is *similar* to A and diagonal, we can write

$$
\begin{aligned}
e^{At} &= TIT^{-1} + T\Lambda t T^{-1} + T\frac{1}{2!}\Lambda^2 t^2 T^{-1} + \dots \\
&= T(I + \Lambda t + \frac{1}{2!}\Lambda^2 t^2 + \dots)T^{-1} \\
&= Te^{\Lambda t}T^{-1}
\end{aligned}
$$

This shows that if $A = T\Lambda T^{-1}$ then $e^{At} = Te^{\Lambda t}T^{-1}$. Now considering equation (2.23), which is a result of the Cayley-Hamilton theorem, we have

$$
\begin{aligned}
e^{At} &= \alpha_0(t)I + \alpha_1(t)A + \dots + \alpha_{n-1}(t)A^{n-1} \\
&= T\alpha_0(t)IT^{-1} + T\alpha_1(t)\Lambda T^{-1} + \dots + T\alpha_{n-1}(t)\Lambda^{n-1}T^{-1} \\
&= T(\alpha_0(t)I + \alpha_1(t)\Lambda + \dots + \alpha_{n-1}(t)\Lambda^{n-1})T^{-1}
\end{aligned}
$$

By comparing the above expression with what we have already proved about $e^{At} = Te^{\Lambda t}T^{-1}$, we can conclude

$$
e^{\Lambda t} = \alpha_0(t)I + \alpha_1(t)\Lambda + \dots + \alpha_{n-1}(t)\Lambda^{n-1} \quad (*)
$$

As the matrix Λ is diagonal

$$
\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \lambda_n \end{bmatrix}
$$

we can easily substitute it in equation $(*)$ and obtain

$$
e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \dots & 0 \\ 0 & e^{\lambda_2 t} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & e^{\lambda_n t} \end{bmatrix} = \begin{bmatrix} \alpha_0(t) & 0 & \dots & 0 \\ 0 & \alpha_0(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \alpha_0(t) \end{bmatrix} + \begin{bmatrix} \alpha_1(t)\lambda_1 & 0 & \dots & 0 \\ 0 & \alpha_1(t)\lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \alpha_1(t)\lambda_n \end{bmatrix} +
$$

$$
\dots + \begin{bmatrix} \alpha_{n-1}(t)\lambda_1^{n-1} & 0 & \dots & 0 \\ 0 & \alpha_{n-1}(t)\lambda_2^{n-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \alpha_{n-1}(t)\lambda_n^{n-1} \end{bmatrix}
$$

The above equation is a matrix equation and can be expressed by n separate equations

$$e^{\lambda_i t} = \alpha_0(t) + \alpha_1(t)\lambda_i + \ldots + \alpha_{n-1}\lambda_i^{n-1}, i = 1, \ldots, n$$

As for the above equation, we started from the equation (2.23), it is now clear that the functions $\alpha_i(t), i = 1, \ldots, n-1$ are identical in both equations.

c) Combining the n equations for the n eigenvalues into a matrix equation

$$\begin{bmatrix} e^{\lambda_1 t} \\ e^{\lambda_2 t} \\ \vdots \\ e^{\lambda_n t} \end{bmatrix} = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{n-1}(t) \end{bmatrix}$$

When the eigenvalues are distinct, the rows of the square matrix in the above equation are linearly independent and this matrix would be invertible and then it would be possible to solve for $\alpha_i(t), i = 1, \ldots, n-1$:

$$\begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{n-1}(t) \end{bmatrix} = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} e^{\lambda_1 t} \\ e^{\lambda_2 t} \\ \vdots \\ e^{\lambda_n t} \end{bmatrix}$$

Solution to Problem q:timeresp   Time response

a) For the state space system $\dot{x} = Ax + bu$ with

$$A = \begin{bmatrix} -6 & 2 \\ -6 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad c = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

The corresponding equations for $e^{\lambda_1 t}$ and $e^{\lambda_2 t}$ are

$$e^{\lambda_1 t} = \alpha_0(t) + \alpha_1(t)\lambda_1$$
$$e^{\lambda_2 t} = \alpha_0(t) + \alpha_1(t)\lambda_2$$

The eigenvalues of A for this system are $\lambda_1 = -3$ and $\lambda_2 = -2$, so α_0 and α_1 can be found as solutions to this equation

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 1 & -2 \end{bmatrix}^{-1} \begin{bmatrix} e^{-3t} \\ e^{-2t} \end{bmatrix} = \begin{bmatrix} -2 & 3 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} e^{-3t} \\ e^{-2t} \end{bmatrix}$$

$$\alpha_0(t) = -2e^{-3t} + 3e^{-2t}$$
$$\alpha_1(t) = -e^{-3t} + e^{-2t}$$

So, by Theorem 2.2:

$$e^{At} = \alpha_0(t)I + \alpha_1(t)A = \begin{bmatrix} \alpha_0(t) - 6\alpha_1(t) & 2\alpha_1(t) \\ -6\alpha_1(t) & \alpha_0(t) + \alpha_1(t) \end{bmatrix}$$

b)   The homogeneous time response of the state is

$$x_0(t) = e^{At}x(0) = e^{At}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2\alpha_0 - 10\alpha_1 \\ -11\alpha_1 + \alpha_0 \end{bmatrix} = \begin{bmatrix} 6e^{-3t} - 4e^{-2t} \\ 9e^{-3t} - 8e^{-2t} \end{bmatrix}$$

While that of the output is

$$y_0(t) = 15e^{-3t} - 12e^{-2t}$$

c)   The state response with the given input is

$$x(t) = x_0(t) + \int_0^t e^{A(t-\tau)}bu(\tau)d\tau$$

The output response with the given input is, with $y_0(t)$ from above as shown below. The matrix $e^{A(t-\tau)}$ is partitioned as

$$e^{A(t-\tau)} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}$$

$$\begin{aligned}
y(t) &= y_0(t) + c\int_0^t e^{A(t-\tau)}bu(\tau)d\tau \\
&= y_0(t) + 2\int_0^t \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}d\tau \\
&= y_0(t) + 2\int_0^t (\Phi_{11} + \Phi_{21})\,d\tau \\
&= y_0(t) + 2\int_0^t (\alpha_0(t-\tau) - 6\alpha_1(t-\tau) - 6\alpha_1(t-\tau))\,d\tau \\
&= y_0(t) + 2\int_0^t \left(10e^{-3(t-\tau)} - 9e^{-2(t-\tau)}\right)d\tau
\end{aligned}$$

$$y(t) = y_0(t) - \frac{20}{3}(e^{-3t} - 1) + \frac{18}{2}(e^{-2t} - 1)$$

$$\left(\int_0^t e^{g(t-\tau)}d\tau = \frac{1}{g}(e^{gt} - 1)\right)$$

Solution to Problem q:springmassccf   Mass-Spring System

a)   The 2nd order differential equation of the system is

$$\ddot{y} + \frac{b}{m}\dot{y} + \frac{k}{m}y = \frac{1}{m}u$$

We present two ways for constructing its controller canonical form.
First way: By taking Laplace Transforms of both sides of the differential equation, we find the transfer function of the system

$$\frac{Y}{U} = \frac{\frac{1}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} = \frac{b(s)}{a(s)}$$

As has been shown in Section 1.1, we can construct a state space model of a system in controller canonical form from its transfer function by comparing (2.11) with (2.14) and (2.15). In this way there are two first order differential equations and an output equation

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \ddot{x}_1 = -\frac{k}{m}x_1 - \frac{b}{m}x_2 + u$$

$$y = \frac{1}{m}x_1$$

which are equivalent to the following state space model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} \frac{1}{m} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Second way: We can choose the states as $x_1 = y$ and $x_2 = \dot{y}$ which have physical significance and are, respectively, the displacement and velocity of the mass. In this case it is straightforward to write a state space model of the system as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

By comparing this state space model with (2.14) and (2.15), we realise that this model is not in the controller canonical form. However if we apply similarity transformation $T = \frac{1}{m}I$, we can transform this model to the controller canonical form.

b) By substituting the given values in the corresponding controller canonical representation of the system, we get system matrices

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -0.1 \end{bmatrix} \qquad\qquad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$c = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad\qquad d = 0$$

Thus,

$$(sI - A)^{-1} = \begin{bmatrix} s & -1 \\ 1 & s + .1 \end{bmatrix}^{-1} = \frac{1}{s^2 + 0.1s + 1} \begin{bmatrix} s + 0.1 & 1 \\ -1 & s \end{bmatrix}$$

and

$$c(sI - A)^{-1}b = \frac{1}{s^2 + 0.1s + 1}$$

c) In open loop

$$\dot{x} = Ax + bu \qquad y = cx + du$$

In closed loop

$$u = fx$$
$$\dot{x} = (A + bf)x$$

For the given system

$$A_{fb} = A + bf = \begin{bmatrix} 0 & 1 \\ -1 & -0.1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ f_1 & f_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 + f_1 & -0.1 + f_2 \end{bmatrix}$$

The closed loop system (A_{fb}, b, c) is again in controller canonical form (why?), therefore we can directly observe that the coefficients of the closed-loop characteristic polynomial are

$$\bar{a}_0 = 1.0 - f_1$$
$$\bar{a}_1 = 0.1 - f_2$$

These can be compared with those of a standard 2nd order system

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

$$\bar{a}_1 = 2\zeta\omega_n \qquad \bar{a}_0 = \omega_n^2$$

Substituting in the design conditions gives

$$\zeta = 0.7 \qquad t_s = 5.0 = \frac{4.6}{\zeta\omega_n}$$

$$\omega_n = 1.314$$

So the solution is

$$\bar{a}_1 = 2 \times 0.7 \times 1.314$$
$$\bar{a}_0 = 1.314^2$$

and finally

$$f_1 = \frac{1.0}{1.0} - (1.314)^2 = -0.73$$
$$f_2 = \frac{0.1}{1.0} - 2 \times 0.7 \times 1.314 = -1.74$$

Note that the fact that the closed loop system was again in controller canonical form, simplified the calculation of state feedback coefficients; this is the main reason for constructing the model in this canonical form.

Matlab: see `cs1_springmass.m`

Solution to Problem q:observercf   Observer canonical form

a)   From the diagram

$$\dot{x}_1 = -a_0 x_3 + b_0 u$$
$$\dot{x}_2 = -a_1 x_3 + b_1 u + x_1$$
$$\dot{x}_3 = -a_2 x_3 + b_2 u + x_2$$
$$y = x_3$$

Therefore the state space matrices are

$$A = \begin{bmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{bmatrix}, \qquad b = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}, \qquad c = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

b)   The transfer function can be directly calculated in the following way. We take Laplace Transforms of both sides of the first order differential equations and the output equation to obtain

$$sX_1 = -a_0 X_3 + b_0 U$$
$$sX_2 = -a_1 X_3 + b_1 U + X_1$$
$$sX_3 = -a_2 X_3 + b_2 U + X_2$$
$$Y = X_3$$

Now the standard procedure is to solve the first three equations to find X_1, X_2 and X_3 based on the given model parameters and then substituting them in the output equation, which is equivalent to use the formula $G(s) = c \begin{bmatrix} sI - A \end{bmatrix}^{-1} b$. Here as the output equals X_3 we can simply solve only the first two equations. Thus, we rearrange the first two equations in matrix form

$$\begin{bmatrix} s & 0 \\ -1 & s \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} -a_0 X_3 + b_0 U \\ -a_1 X_3 + b_1 U \end{bmatrix}$$

Then we have

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \frac{1}{s^2} \begin{bmatrix} s & 0 \\ 1 & s \end{bmatrix} \begin{bmatrix} -a_0 X_3 + b_0 U \\ -a_1 X_3 + b_1 U \end{bmatrix}$$

which results in

$$X_2 = \frac{1}{s^2} (-(a_1 s + a_0) X_3 + (b_1 s + b_0) U)$$

By substituting this in the third equation to find X_3 based on U and then substituting X_3 in the output equation, we finally obtain the transfer function as

$$\frac{Y(s)}{U(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

Another way: it is also possible to first construct the governing differential equation of the system and then calculating the transfer function by taking Laplace Transforms of both sides of the equation.

We construct the governing differential equation

$$\dot{y} = \dot{x}_3 \qquad\qquad = -a_2 y + b_2 u + x_2$$
$$\ddot{y} = -a_2 \dot{y} + b_2 \dot{u} + \dot{x}_2 \qquad\qquad = -a_2 \dot{y} + b_2 \dot{u} + (-a_1)y + b_1 u + x_1$$
$$\dddot{y} = -a_2 \ddot{y} + b_2 \ddot{u} + (-a_1)\dot{y} + b_1 \dot{u} + \dot{x}_1 \quad = -a_2 \ddot{y} + b_2 \ddot{u} + (-a_1)\dot{y} + b_1 \dot{u} + (-a_0)y + b_0 u$$

By taking the Laplace Transform and reorganising we obtain

$$Y(s^3 + a_2 s^2 + a_1 s + a_0) = U(b_2 s^2 + b_1 s + b_0)$$

The transfer function is then

$$\frac{Y(s)}{U(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

Solution to Problem q:contobsD   Bi-proper system

By polynomial division, we obtain

$$H(s) = 4 + \frac{s^2 + 5s + 2}{s^3 + 6s^2 + 10s + 8} = 4 + \frac{\tilde{b}(s)}{a(s)}$$

Then by comparing this with the equation $G(s) = c(sI - A)^{-1}b + d$ (2.17), we realize that the first term is a 'feedthrough' term, corresponding to $d = 4$.

The controller and observer forms are calculated from the coefficients of $a(s)$ and $\tilde{b}(s)$; note that $a(s)$ is the denominator of H(s), whereas $\tilde{b}(s)$ is not the numerator of H(s) but of the strictly proper remainder after polynomial division.

Controller canonical form

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -10 & -6 \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$c = \begin{bmatrix} 2 & 5 & 1 \end{bmatrix} \qquad\qquad\qquad d = 4$$

Observer canonical form

$$A = \begin{bmatrix} 0 & 0 & -8 \\ 1 & 0 & -10 \\ 0 & 1 & -6 \end{bmatrix} \qquad\qquad b = \begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix}$$

$$c = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \qquad\qquad\qquad d = 4$$

Alternatively, the controller and observer canonical forms can be obtained without polynomial division by using equation (2.13) and following the derivation in time domain discussed in Chapter 1. This is illustrated here for the case of the controller canonical form:

For a bi-proper system, equation (2.13) changes to

$$y(t) = b_n \frac{d^n}{dt^n} v(t) + b_{n-1} \frac{d^{n-1}}{dt^{n-1}} v(t) + \ldots + b_1 \frac{d}{dt} v(t) + b_0 v(t)$$

and consequently, we have an additional term in equation 2.15

$$y(t) = [b_0 \ b_1 \ \ldots \ b_{n-1}] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + b_n \dot{x}_n$$

By applying the coefficients from the transfer function, we have

$$y(t) = [34 \ 45 \ 25] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 4\dot{x}_3$$

For $\dot{x}_3$, from equation 2.14 we derive

$$\dot{x}_3 = [-8 \ -10 \ -6] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + u(t)$$

Thus, for $y(t)$ we obtain

$$y(t) = [34 \ 45 \ 25] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 4[-8 \ -10 \ -6] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 4u(t) = [2 \ 5 \ 1] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 4u(t)$$

From the above result, we see again that $c = \begin{bmatrix} 2 & 5 & 1 \end{bmatrix}$ and $d = 4$.

Solution to Problem q:simtrans   Similarity transformation

The 'if' part of the proof follows by algebraic substitution:

$$G_1(s) = c_1[sI - A_1]^{-1} b_1$$
$$G_2(s) = c_2[sI - A_2]^{-1} b_2$$
$$G_2(s) = c_1 T[sI - T^{-1} A_1 T]^{-1} T^{-1} b_1$$

Then, because $I = T^{-1} T$

$$G_2(s) = c_1 T[T^{-1} sT - T^{-1} A_1 T]^{-1} T^{-1} b_1$$

The factors T^{-1} (left) and T (right) can be divided

$$G_2(s) = c_1 T [T^{-1}(sI - A_1)T]^{-1} T^{-1} b_1$$

because $[X_1 Y X_2]^{-1} = X_2^{-1} Y^{-1} X_1^{-1}$

$$G_2(s) = c_1 T T^{-1} [sI - A_1]^{-1} T T^{-1} b_1 = c_1 [sI - A_1]^{-1} b_1 = G_1(s)$$

Solution to Problem q:watertank   Linearisation tank with valve

a)   The behaviour of the valve is described by $f_{out} = k_v \sqrt{P} u = k_v \sqrt{\rho g h} u$, with $k_t = \sqrt{\rho g} k_v$.

$$\dot{h} = \frac{1}{A_t}(f_{in} - f_{out}) = \frac{1}{A_t}\left(f_{in} - k_t \sqrt{h} u\right) = f_{nl}(h, u)$$

b)   With variables at steady state indicated by u_0, h_0, f_{in0}, the steady state is defined by $\dot{h}_0 = 0$, so

$$u_0 = \frac{f_{in0}}{k_t \sqrt{h_0}}$$

c)   For small changes in h, u

$$\dot{h}_0 + \delta \dot{h} = f_{nl}(h_0 + \delta h, u_0 + \delta u)$$

$$\simeq f_{nl}(h_0, u_0) + \frac{\partial}{\partial h} f_{nl}(h_0, u_0) \delta h + \frac{\partial}{\partial u} f_{nl}(h_0, u_0) \delta u$$

$$\frac{\partial}{\partial h} f_{nl}(h, u) = -\frac{k_t}{A_t} \frac{u}{2\sqrt{h}}$$

$$\frac{\partial}{\partial u} f_{nl}(h, u) = -\frac{k_t}{A_t} \sqrt{h}$$

so at h_0, u_0

$$\delta \dot{h} = -\frac{k_t u_0}{2 A_t \sqrt{h_0}} \delta h - \frac{k_t}{A_t} \sqrt{h_0} \delta u = -\frac{f_{in0}}{2 A_t h_0} \delta h - \frac{k_t}{A_t} \sqrt{h_0} \delta u$$

d)   The transfer function is

$$\frac{H(s)}{U(s)} = \frac{K_l(u_0, h_0)}{\tau_l(u_0, h_0) s + 1}$$

where

$$K_l(u_0, h_0) = -2 \frac{h_0}{u_0}$$

$$\tau_l(u_0, h_0) = \frac{2 A_t \sqrt{h_0}}{k_t u_0}$$

e)   The state space matrices are

$$A = -\frac{f_{in0}}{2A_t h_0} \qquad B = -\frac{k_t}{A_t}\sqrt{h_0} \qquad C = 1 \qquad D = 0$$

Solution to Problem q:MinSegLinearize   Mini Segway: Modeling

a)   The function $f(x, u)$ has the form

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{s} \\ \dot{\alpha} \\ f_1 \\ f_2 \end{bmatrix}.$$

We can see that

$$\ddot{s} = f_1(x, u)$$
$$\ddot{\alpha} = f_2(x, u)$$

From the hint, the unknown functions can be derived as

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = M(s, \alpha)^{-1} f_q(x, u)$$

for the matrix $M(s, \alpha)$

$$M(s, \alpha) = \begin{bmatrix} m_p + 2m_w + 2\frac{J_w}{r^2} & m_p l\, cos(\alpha) \\ m_p l\, cos(\alpha) & J_p + m_p l^2 \end{bmatrix}$$

for the matrix inverse $M(s, \alpha)^{-1}$

$$M(s, \alpha)^{-1} = \frac{1}{\Delta} \begin{bmatrix} J_p + m_p l^2 & -m_p l\, cos(\alpha) \\ -m_p l\, cos(\alpha) & m_p + 2m_w + 2\frac{J_w}{r^2} \end{bmatrix}.$$

In addition, the following variables are defined

$$m_{eq} := m_p + 2m_w + 2\frac{J_w}{r^2}$$
$$J_{eq} := J_p + m_p l^2$$
$$\Delta := m_{eq} J_{eq} - (m_p l\, cos(\alpha))^2 = det(M(s, \alpha))$$

for the vector $f_q(x, u)$

$$f_q(s, \alpha) = \left[\begin{array}{c} \frac{k_t}{Rr}u - \frac{k_t k_b}{Rr^2}\dot{s} + \frac{k_t k_b}{Rr}\dot{\alpha} + m_p l sin(\alpha)\dot{\alpha}^2 \\ -\frac{k_t}{R}u + \frac{k_t k_b}{Rr}\dot{s} - \frac{k_t k_b}{R}\dot{\alpha} + m_p g l sin(\alpha) \end{array} \right]$$

$f_q(x, u)$ can be rewritten as

$$f_q(x, u) = (\frac{k_t}{Rr}u - \frac{k_t k_b}{Rr^2}\dot{s} + \frac{k_t k_b}{Rr}\dot{\alpha})\underbrace{\left[\begin{array}{c} 1 \\ -r \end{array} \right]}_{linear\ part} + \underbrace{m_p l sin(\alpha)\left[\begin{array}{c} \dot{\alpha}^2 \\ g \end{array} \right]}_{nonlinear\ part}.$$

Finally, it follows that

$$f_1(x, u) = \frac{J_{eq} + m_p l r cos(\alpha)}{\Delta}(\frac{k_t}{Rr}u - \frac{k_t k_b}{Rr^2}\dot{s} + \frac{k_t k_b}{Rr}\dot{\alpha}) + \frac{J_{eq}\dot{\alpha}^2 - m_p g l cos(\alpha)}{\Delta}m_p l sin(\alpha)$$

$$f_2(x, u) = \frac{-m_{eq}r - m_p l cos(\alpha)}{\Delta}(\frac{k_t}{Rr}u - \frac{k_t k_b}{Rr^2}\dot{s} + \frac{k_t k_b}{Rr}\dot{\alpha}) + \frac{m_{eq}g - m_p l cos(\alpha)\dot{\alpha}^2}{\Delta}m_p l sin(\alpha).$$

b)   From the substitution of $\overline{x}$ and 0 into $f(x, u)$

$$f(\overline{x}, 0) = \left[\begin{array}{c} \dot{\overline{s}} \\ \dot{\overline{\alpha}} \\ f_1(\overline{x}, 0) \\ f_2(\overline{x}, 0) \end{array} \right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right]$$

it follows that $\dot{\overline{s}} = \dot{\overline{\alpha}} = 0$ and $\ddot{\overline{s}} = \ddot{\overline{\alpha}} = 0$.

Simplifying the *nonlinear differential equations*, if follows that

$$sin(\overline{\alpha}) = 0.$$

Since $\overline{x}$ should be *stable*, we set $\overline{\alpha} = 0$. This leads to

$$\overline{x} = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right]$$

c)   There are two ways to derive the matrices A and B

   solution I: linearizing (2.24) first using *small angle approximation*, then deriving a linear state space model

   solution II: directly calculating the jacobians from the Taylor series

solution I

simplifying the following *nonlinear terms* with *small angle approximation*

$$cos(\alpha) \approx 1$$
$$sin(\alpha) \approx \alpha$$
$$\dot{\alpha}^2 \approx 0$$

replacing the *nonlinear terms* in (2.24)

$$(m_p + 2m_w + 2\frac{J_w}{r^2})\ddot{s} + m_p l\ddot{\alpha} + \frac{k_t k_b}{Rr^2}\dot{s} - \frac{k_t k_b}{Rr}\dot{\alpha} = \frac{k_t}{Rr}u$$

$$(J_p + m_p l^2)\ddot{\alpha} + m_p l\ddot{s} - \frac{k_t k_b}{Rr}\dot{s} + \frac{k_t k_b}{R}\dot{\alpha} - m_p gl\alpha = -\frac{k_t}{R}u$$

this *linearized equations* can be written in matrix notation

$$M_0 \begin{bmatrix} \ddot{s} \\ \ddot{\alpha} \end{bmatrix} + D_0 \begin{bmatrix} \dot{s} \\ \dot{\alpha} \end{bmatrix} + K_0 \begin{bmatrix} s \\ \alpha \end{bmatrix} = F_0 u$$

and finally in canonical form

$$\dot{x} = \begin{bmatrix} 0 & I \\ -M_0^{-1}K_0 & -M_0^{-1}D_0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -M_0^{-1}F_0 \end{bmatrix} u$$

with the matrices and vectors

$$M_0 = \begin{bmatrix} m_p + 2m_w + 2\frac{J_w}{r^2} & m_p l \\ m_p l & J_p + m_p l^2 \end{bmatrix}$$

$$D_0 = \begin{bmatrix} \frac{k_t k_b}{Rr^2} & -\frac{k_t k_b}{Rr} \\ -\frac{k_t k_b}{Rr} & \frac{k_t k_b}{R} \end{bmatrix}$$

$$K_0 = \begin{bmatrix} 0 & 0 \\ 0 & -m_p gl \end{bmatrix}$$

$$F_0 = \begin{bmatrix} \frac{k_t}{Rr} \\ -\frac{k_t}{R} \end{bmatrix}$$

and $\Delta_0 := m_{eq}J_{eq} - (m_p l)^2 = det(M_0)$.

Finally, the matrices A and B are

$$
A = \begin{bmatrix} 0 & I \\ -M_0^{-1}K_0 & -M_0^{-1}D_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{31} & a_{11} & a_{12} \\ 0 & a_{32} & a_{21} & a_{22} \end{bmatrix}
$$

$$
B = \begin{bmatrix} 0 \\ -M_0^{-1}F_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b_1 \\ b_2 \end{bmatrix}
$$

with the coefficients

$$
a_{11} = -\frac{k_t k_b}{Rr} \frac{1}{\Delta_0} \left(\frac{J_{eq}}{r} + m_p l \right)
$$

$$
a_{12} = \frac{k_t k_b}{Rr} \frac{1}{\Delta_0} (J_{eq} + m_p l r)
$$

$$
a_{21} = \frac{k_t k_b}{Rr} \frac{1}{\Delta_0} \left(m_{eq} + m_p \frac{l}{r} \right)
$$

$$
a_{22} = -\frac{k_t k_b}{Rr} \frac{1}{\Delta_0} (m_{eq} r + m_p l)
$$

$$
a_{31} = -\frac{m_p g l}{\Delta_0} m_p l
$$

$$
a_{32} = \frac{m_p g l}{\Delta_0} m_{eq}
$$

$$
b_1 = \frac{k_t}{Rr} \frac{1}{\Delta_0} (J_{eq} + m_p l r)
$$

$$
b_2 = -\frac{k_t}{Rr} \frac{1}{\Delta_0} (m_{eq} r + m_p l).
$$

solution II

directly calculating the jacobians from the Taylor series

$$
A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_1}{\partial s} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \dot{s}} & \frac{\partial f_1}{\partial \dot{\alpha}} \\ \frac{\partial f_2}{\partial s} & \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \dot{s}} & \frac{\partial f_2}{\partial \dot{\alpha}} \end{bmatrix} \Big|_{\bar{x},0}
$$

$$
B = \begin{bmatrix} 0 \\ 0 \\ \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix} \Big|_{\bar{x},0}
$$

derivative w.r.t. s

$$\frac{\partial f_1}{\partial s} = \frac{\partial f_2}{\partial s} = 0$$

since f_1 and f_2 are independent of s.

derivative w.r.t. α

For $\frac{\partial f_1}{\partial \alpha}$ and $\frac{\partial f_2}{\partial \alpha}$ the partial derivation gets simplified because $\Delta'|_{\overline{x},0} = 0$

$$\frac{\partial f_1}{\partial \alpha}\Big|_{\overline{x},0} = \frac{\partial}{\partial \alpha}\frac{\tilde{f}_1}{\Delta}\Big|_{\overline{x},0} = \frac{1}{\Delta}\frac{\partial \tilde{f}_1}{\partial \alpha} + \tilde{f}_1\frac{\partial}{\partial \alpha}\frac{1}{\Delta}\Big|_{\overline{x},0} = \frac{1}{\Delta}\frac{\partial \tilde{f}_1}{\partial \alpha} - \tilde{f}_1\frac{\Delta'}{\Delta^2}\Big|_{\overline{x},0} = \frac{1}{\Delta}\frac{\partial \tilde{f}_1}{\partial \alpha}\Big|_{\overline{x},0}$$

(this makes it easier)

$$\frac{\partial f_1}{\partial \alpha}\Big|_{\overline{x},0} = \frac{-m_p lr sin(\overline{\alpha})}{\Delta_0}(\frac{k_t}{Rr}0 - \frac{k_t k_b}{Rr^2}\dot{\overline{s}} + \frac{k_t k_b}{Rr}\dot{\overline{\alpha}}) + \frac{0 + m_p gl sin(\overline{\alpha})}{\Delta_0}m_p l sin(\overline{\alpha}) +$$

$$\frac{J_{eq}\dot{\overline{\alpha}}^2 - m_p gl cos(\overline{\alpha})}{\Delta_0}m_p l cos(\overline{\alpha}) = -\frac{m_p gl}{\Delta_0}m_p l = a_{31}$$

$$\frac{\partial f_2}{\partial \alpha}\Big|_{\overline{x},0} = \frac{m_p l sin(\overline{\alpha})}{\Delta_0}(\frac{k_t}{Rr}0 - \frac{k_t k_b}{Rr^2}\dot{\overline{s}} + \frac{k_t k_b}{Rr}\dot{\overline{\alpha}}) + \frac{0 + m_p l sin(\overline{\alpha})\dot{\overline{\alpha}}^2}{\Delta_0}m_p l sin(\overline{\alpha}) +$$

$$\frac{m_{eq}g - m_p l cos(\overline{\alpha})\dot{\overline{\alpha}}^2}{\Delta_0}m_p l cos(\overline{\alpha}) = \frac{m_{eq}g}{\Delta_0}m_p l = a_{32}$$

derivative w.r.t. $\dot{s}$

$$\frac{\partial f_1}{\partial \dot{s}}\Big|_{\overline{x},0} = \frac{J_{eq} + m_p lr cos(\overline{\alpha})}{\Delta_0}(... - \frac{k_t k_b}{Rr^2}...) = -\frac{J_{eq} + m_p lr}{\Delta_0}\frac{k_t k_b}{Rr^2} = a_{11}$$

$$\frac{\partial f_2}{\partial \dot{s}}\Big|_{\overline{x},0} = \frac{-m_{eq}r - m_p l cos(\overline{\alpha})}{\Delta_0}(... - \frac{k_t k_b}{Rr^2}...) = \frac{m_{eq}r + m_p l}{\Delta_0}\frac{k_t k_b}{Rr^2} = a_{21}$$

derivative w.r.t. $\dot{\alpha}$

$$\frac{\partial f_1}{\partial \dot{\alpha}}\Big|_{\overline{x},0} = \frac{J_{eq} + m_p lr cos(\overline{\alpha})}{\Delta_0}(... + \frac{k_t k_b}{Rr}) = \frac{J_{eq} + m_p lr}{\Delta_0}\frac{k_t k_b}{Rr} = a_{12}$$

$$\frac{\partial f_2}{\partial \dot{\alpha}}\Big|_{\overline{x},0} = \frac{-m_{eq}r - m_p l cos(\overline{\alpha})}{\Delta_0}(... + \frac{k_t k_b}{Rr}) = -\frac{m_{eq}r + m_p l}{\Delta_0}\frac{k_t k_b}{Rr} = a_{22}$$

derivative w.r.t. u

$$\frac{\partial f_1}{\partial u}\Big|_{\overline{x},0} = \frac{J_{eq} + m_p lr cos(\overline{\alpha})}{\Delta_0}(\frac{k_t}{Rr} + ...) = \frac{J_{eq} + m_p lr}{\Delta_0}\frac{k_t}{Rr} = b_1$$

$$\frac{\partial f_2}{\partial u}\Big|_{\overline{x},0} = \frac{-m_{eq}r - m_p l cos(\overline{\alpha})}{\Delta_0}(\frac{k_t}{Rr} + ...) = -\frac{m_{eq}r + m_p l}{\Delta_0}\frac{k_t}{Rr} = b_2$$

d)   The matrices C and D are

$$
C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

Solution to Problem q:MinSegVerifySim   Mini Segway: Initial Condition Simulation
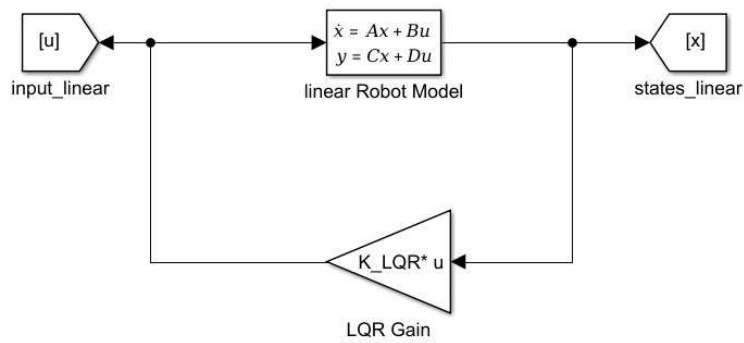
a)   See *simulation_LQR.slx*
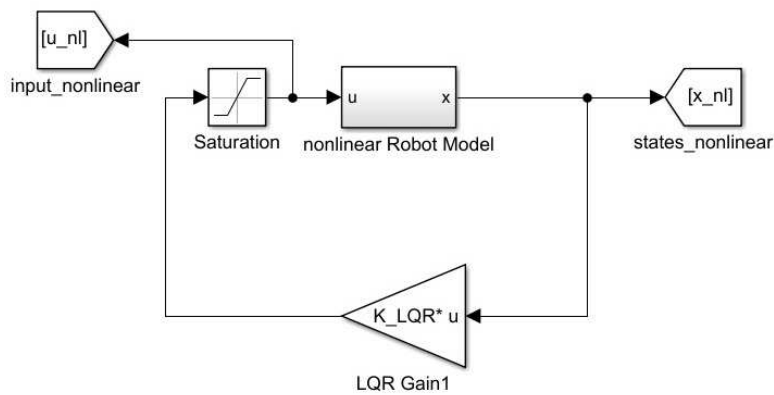

Figure C.2: Block diagram of linear model

Figure C.3: Block diagram of nonlinear model

b)   See *simulation_LQR_script.m*

5°:   *stable* and good match between linear and nonlinear simulation. The maximum voltage is inside the limitation $(u_{max} < V_s)$.
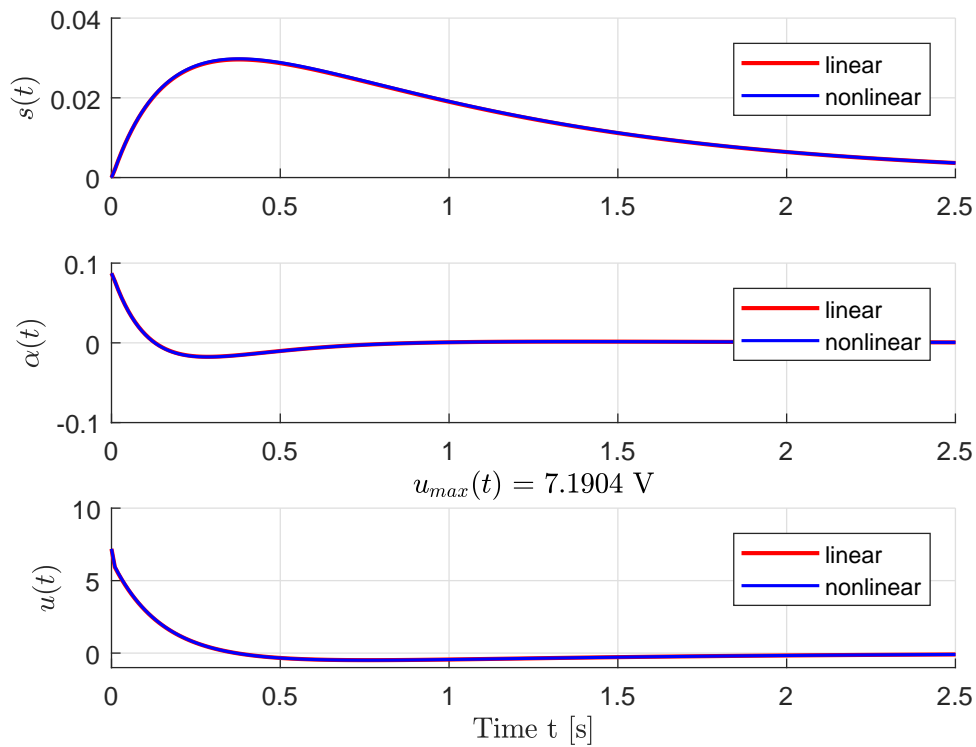

Figure C.4: simulation for $\alpha_0 = 5°$

c)   10°:   still stable, but *difference* between linear and nonlinear simulation. Since the maximum voltage is not inside the limitation anymore, the saturation block has an influence on the input voltage.

15°:   difference becomes more noticeable. The nonlinear states becomes *unstable*, the nonlinear input is completely in saturation.
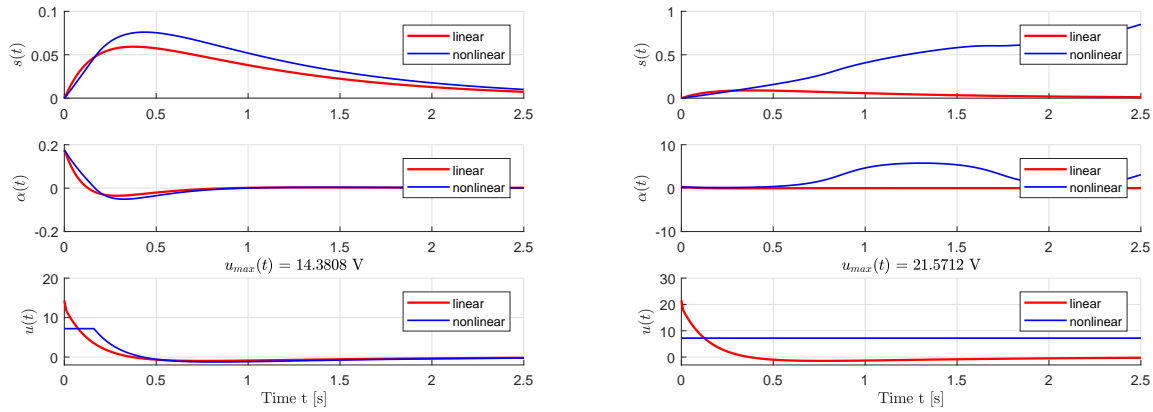
Figure C.5: simulation for $\alpha_0 = 10°$ (left) and $\alpha_0 = 15°$ (right)

Solution to Problem q:MinSegVerifyExp   Mini Segway: Experimental Validation

a)   See *simulation_LQR_disturbance.slx*

states get *stabilized*, where $s(t)$ reaches equilibrium after a 'long' time and $\alpha(t)$ reaches equilibrium after 1s. The input voltage vanishes after 0.5s.
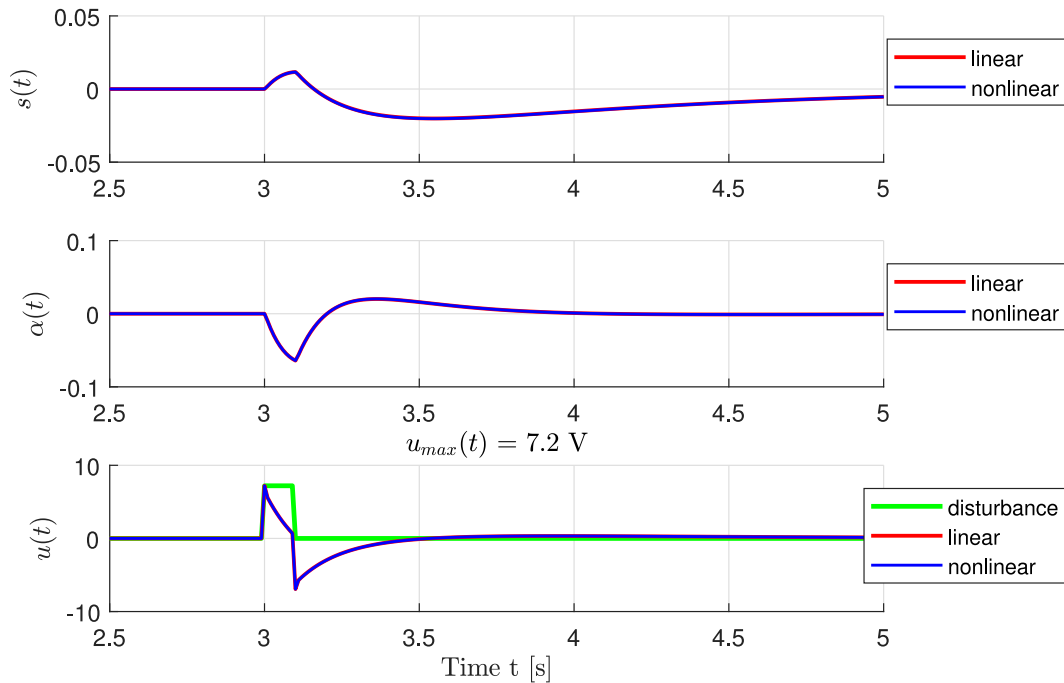

Figure C.6: simulation of disturbance rejection ($d_0 = 7.2\,V$, $t_{start} = 3\,s$, $\Delta t = 0.1\,s$)

b) See *experiment_LQR_disturbance.slx*

good match between experiment and simulation until $t = 3.5\,s$, but then the difference in the position gets higher over time. In the experiment, the input voltage is corrupted by noise.
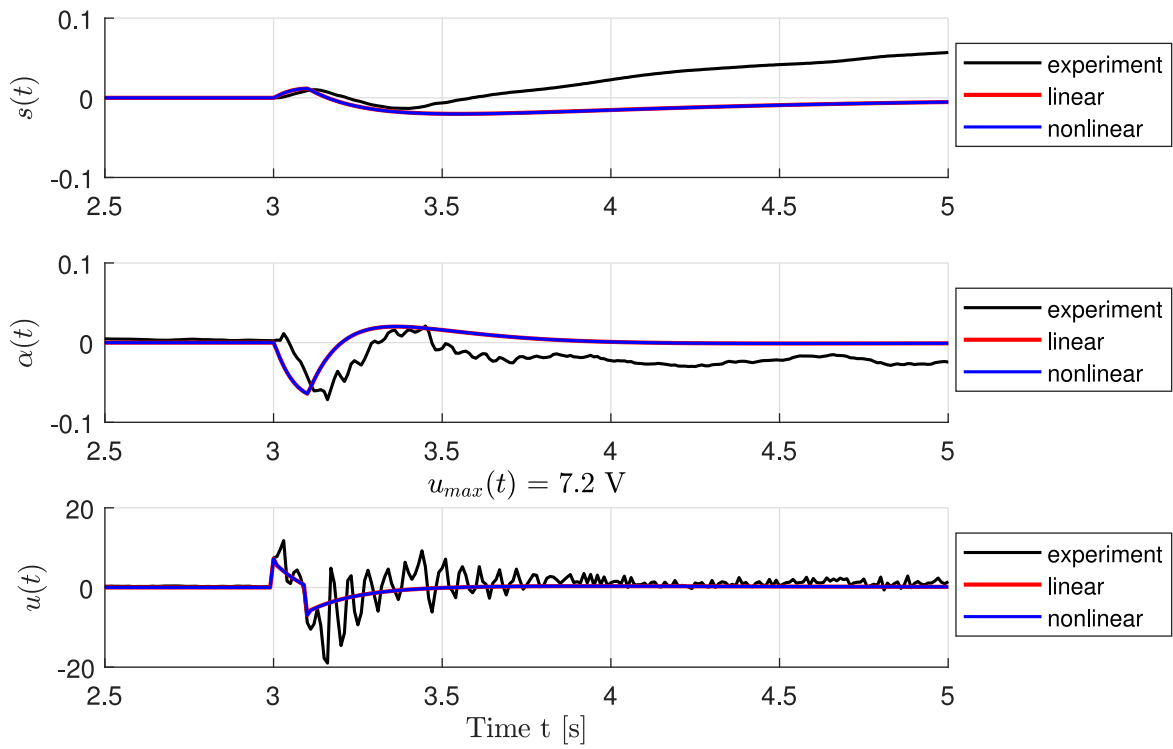

Figure C.7: comparison of simulation and experiment ($d_0 = 7.2\,V$, $t_{start} = 3\,s$, $\Delta t = 0.1\,s$)

C.2 Chapter 3

Solution to Problem q:pzcancellation   Analysis of controllability

a)   The system is unstable (eigenvalues are at 1,-1)

b)   The controllability matrix is

$$\begin{bmatrix} b & Ab \end{bmatrix} = \begin{bmatrix} -2 & 2 \\ 1 & -1 \end{bmatrix}$$

This has rank 1, so the system is not controllable. The system is transformed to the controller form using

$$T = \begin{bmatrix} -2 & 0.5 \\ 1 & 1 \end{bmatrix}$$

which gives the transformed system A and b matrices

$$A = \begin{bmatrix} -1 & 0.5 \\ 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

So the uncontrollable pole is unstable and the system is unstabilizable.

c)   The controllable subspace is the line $x_2 = -0.5x_1$ which is also the range of the controllability matrix. The uncontrollable subspace is perpendicular to this.

d)   The response to an initial state $x(0)$ is

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

The Laplace-Transform of this is

$$X(s) = (sI - A)^{-1} x(0) = \frac{1}{(s-1)(s+1)} \begin{bmatrix} s-1 & 0 \\ 1 & s+1 \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$x_{1i}(s) = \frac{1}{s+1} x_{10}$$

$$x_{2i}(s) = \frac{1}{(s-1)(s+1)} x_{10} + \frac{1}{s-1} x_{20}$$

$$Y_i(s) = \frac{1}{(s-1)(s+1)} x_{10} + \frac{1}{s-1} x_{20} = -\frac{x_{10}}{2(s+1)} + \frac{0.5x_{10} + x_{20}}{s-1}$$

The solution has a stable and an unstable part (eigenvalues -1,1). There is an unbounded, unstable, solution when $0.5x_{10} + x_{20} \neq 0$. Otherwise the solution is bounded and indeed eventually reaches the origin. Although comparing this to the answer of part (c), $0.5x_{10} + x_{20} \neq 0$ also means that the initial state is not in the

controllable subspace. But this is of course just a coincidence and in general there is no relationship between the concepts of controllability of a system and its free response!

In the phase portrait of a two-dimensional linear system, there would be some straight-line trajectories which are the lines spanned by eigenvectors of A. If we start on one of these lines, we would stay on it forever and the solution is a simple exponential growth or decay along it. If the line is spanned by an eigenvector which corresponds to a stable eigenvalue, the solution would be an exponential decay and if it is spanned by an eigenvector which corresponds to an unstable eigenvalue, the solution would be an exponential growth. Here A has two eigenvalues -1 and 1 and their corresponding eigenvectors are $[-2\ 1]^T$ and $[0\ 1]^T$. So if the initial values satisfy the equation $0.5x_{10} + x_{20} = 0$, it means that the solution stays on the line spanned by $[-2\ 1]^T$ which corresponds to the stable eigenvalue -1 and therefore the solution would be an exponential decay towards the origin. Such a solution has nothing to do with the vector b.

If the system is not controllable, the controllability matrix does not have full rank. In order for the 2×2 controllability matrix not to have full rank, its columns should be linearly dependent or $Ab = \lambda b$ which means that the vector b should be an eigenvector of A. Thus, if we have either $[-2\ \ 1]^T$ or $[0\ 1]^T$ as the vector b, the system would be uncontrollable. If we have a system with $b = [0\ 1]^T$ then the line spanned by this vector is the controllable subspace and the line spanned by $[-2\ \ 1]^T$ is still the trajectory which exhibits exponential decay and unlike the previous system, they are not the same lines.

e) The transfer function is

$$G(s) = c\left[sI - A\right]^{-1} b = \frac{1}{(s-1)(s+1)} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} s - 1 & 0 \\ 1 & s + 1 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix} = \frac{1}{s+1}$$

f) The transfer function is not a complete description of the behaviour: it assumes the initial conditions are zero,and has a pole zero cancellation.

Simulation: `cs2_unstab.m`

Solution to Problem q:gramian   Gramian calculation

a) Using the rule for the differentiation of matrix products,

$$\frac{d}{dt}(uv) = \frac{du}{dt}v + u\frac{dv}{dt}$$

and the relations

$$\frac{d}{dt}e^{At} = Ae^{At}, \qquad \frac{d}{dt}e^{A^T t} = (\frac{d}{dt}e^{At})^T = e^{A^T t}A^T$$

so

$$\frac{d}{dt}\left(e^{At}bb^T e^{A^T t}\right) = Ae^{At}bb^T e^{A^T t} + e^{At}bb^T e^{A^T t}A^T$$

b) From the definition,

$$AW_c + W_cA^T = A\int_0^t e^{A\tau}bb^T e^{A^T \tau}d\tau + \int_0^t e^{A\tau}bb^T e^{A^T \tau}d\tau A^T$$

$$= \int_0^t \{Ae^{A\tau}bb^T e^{A^T \tau} + e^{A\tau}bb^T e^{A^T \tau}A^T\}d\tau$$

$$= \int_0^t \frac{d}{d\tau}e^{A\tau}bb^T e^{A^T \tau}d\tau$$

from part (a).

c) Solving the integral,

$$AW_c + W_cA^T = \int_0^t \frac{d}{d\tau}e^{At}bb^T e^{A^T \tau}d\tau$$

$$= [e^{A\tau}bb^T e^{A^T \tau}]_{\tau=0}^{\tau=t}$$

$$e^{A\tau} \to 0 \text{ as } \tau \to \infty$$

for stable A:

$$AW_c + W_cA^T \to -bb^T \text{ as } t \to \infty$$

Solution to Problem q:resolventidentity   Resolvent Identity

$$\text{adj}(sI - A) = \left[Is^{n-1} + (A + a_{n-1}I)s^{n-2} + \ldots + (A^{n-1} + a_{n-1}A^{n-2} + \ldots + a_1 I)\right]$$

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

$$\text{adj}(sI - A) = \text{``Resolvent''} = (sI - A)^{-1}\det(sI - A)$$

Let

$$RHS = \left[Is^{n-1} + (A + a_{n-1}I)s^{n-2} + \ldots + (A^{n-1} + a_{n-1}A^{n-2} + \ldots + a_1 I)\right]$$

Then

$$RHS \cdot (sI - A) = \left[Is^{n-1} + (A + a_{n-1}I)s^{n-2} + \ldots \right.$$

$$\left. \ldots + (A^{n-1} + a_{n-1}A^{n-2} + \ldots + a_1 I)\right](sI - A)$$

and

$$RHS \cdot (sI - A) = Is^n + (A + a_{n-1}I - A)s^{n-1} + (a_{n-2}I + a_{n-1}A + A^2 - a_{n-1}A - A^2)s^{n-2} + \ldots$$

$$- (a_1 A + \ldots + a_{n-1}A^{n-1} + A^n)$$

$$RHS \cdot (sI - A) = I(s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_1 s) - (a_1 A + \ldots + a_{n-1}A^{n-1} + A^n)$$

From the Cayley-Hamilton Theorem

$$-(A^n + a_{n-1}A^{n-1} + \ldots a_1 A) = a_0 I$$

Thus

$$RHS \cdot (sI - A) = I(s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_1 s + a_0)$$

and hence

$$RHS = (sI - A)^{-1}\det(sI - A)$$
$$= \mathrm{adj}(sI - A)$$

Solution to Problem q:bassgura   Bass-Gura Formula

For the given system

$$A = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix}, \qquad b = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

the characteristic equation is

$$(s - 1)(s + 4) + 6 = s^2 + 3s + 2$$

The desired closed loop characteristic equation is

$$s^2 + 2\omega_n \zeta s + \omega_n^2 = s^2 + 2 \times 2 \times 0.7 \times s + 4$$

so

$$p = \begin{bmatrix} \bar{a}_1 - a_1 & \bar{a}_0 - a_0 \end{bmatrix}$$
$$p = \begin{bmatrix} (2.8 - 3) & (4 - 2) \end{bmatrix} = \begin{bmatrix} -0.2 & 2 \end{bmatrix}$$

$$T_a = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}, \qquad T_a^{-1} = \begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix}$$

The controllability matrix $\mathcal{C}$ is

$$\begin{bmatrix} b & Ab \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix}$$

$$\mathcal{C}^{-1} = \frac{1}{14} \begin{bmatrix} 5 & -1 \\ -1 & 3 \end{bmatrix}$$

So the solution is

$$f = -pT_a^{-1}\mathcal{C}^{-1}$$

$$f = -\begin{bmatrix} -0.2 & 2 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix} \frac{1}{14} \begin{bmatrix} 5 & -1 \\ -1 & 3 \end{bmatrix} = \begin{bmatrix} 0.26 & -0.57 \end{bmatrix}$$

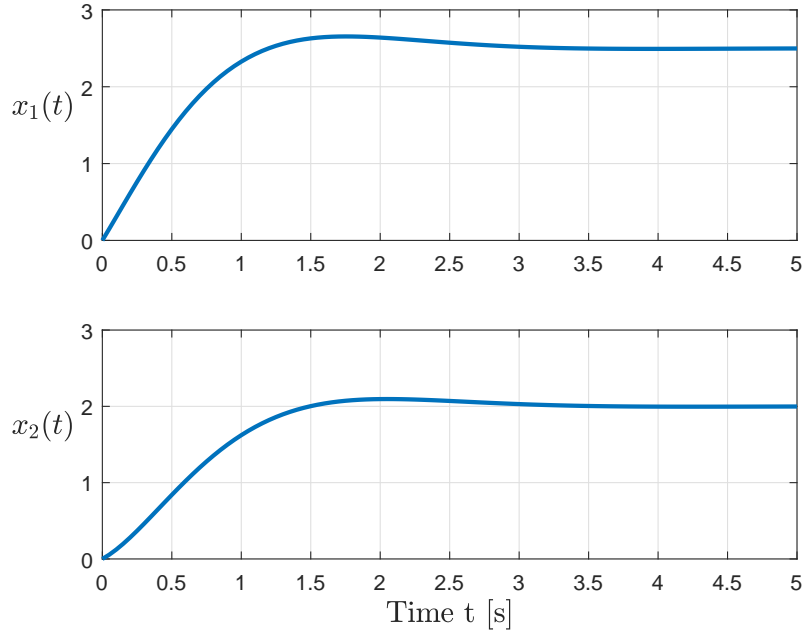For the script for the step response see `cs2_BassGura.m`.

Figure C.8: responses of the states to a unit step change in the closed loop system input u_v

Solution to Problem q:PBHtest **PBH test**

a)   If $q^T A = \lambda q^T$ and $q^T b = 0$ then

$$q^T \mathcal{C} = q^T \begin{bmatrix} b & Ab & A^2b & \dots A^{n-1}b \end{bmatrix}$$
$$q^T b = 0$$
$$q^T Ab = \lambda q^T b = 0$$

so

$$q^T A^m b = \lambda^m q^T b = 0$$
$$q^T \mathcal{C} = 0$$

b)   If such a $q \neq 0$ exists, $\mathcal{C}$ is singular and the system is not controllable. In fact the converse is also true. In order to show this, consider that for an uncontrollable system according to theorem 3.5 there exists a similarity transformation T_c such that

$$T_c^{-1} A T_c = \begin{bmatrix} \bar{A}_c & \bar{A}_{12} \\ 0 & \bar{A}_{\bar{c}} \end{bmatrix}, \qquad T_c^{-1} b = \begin{bmatrix} \bar{b}_c \\ 0 \end{bmatrix}$$

Now consider that for matrix $\bar{A}_{\bar{c}}$ there exists a left eigenvector $x_{\bar{c}}$ such that

$$x_{\bar{c}}^T \bar{A}_{\bar{c}} = \lambda x_{\bar{c}}^T$$

By choosing $x = \begin{bmatrix} 0 \\ x_{\bar{c}} \end{bmatrix}$ we obtain $x^T T_c^{-1} b = 0$. We also have

$$x^T T_c^{-1} A T_c = \lambda x^T$$

which implies $x^T T_c^{-1} A = \lambda x^T T_c^{-1}$.

By defining $q^T = x^T T_c^{-1}$, we have $q^T A = \lambda q^T$ and $q^T b = 0$.

c) For $\begin{bmatrix} sI - A & b \end{bmatrix}$ to lose rank, there must be a vector $q \neq 0$ such that

$$q^T \begin{bmatrix} sI - A & b \end{bmatrix} = 0$$

$$q^T (sI - A) = 0 \quad \text{and} \quad q^T b = 0$$
$$q^T A = s q^T \quad \text{and} \quad q^T b = 0$$

That is, such a vector q can only exist if and only if it is a left eigenvector of A and $q^T b = 0$.

d) From parts (a) and (b), it follows that a system is uncontrollable iff any left eigenvector of A belongs to the left null space of b, i.e. a q exists such that $q^T A = \lambda q^T$ and $q^T b = 0$. By having such a q, we know from part (c) that there exists some $s \in \mathbb{C}$ such that $\begin{bmatrix} sI - A & b \end{bmatrix}$ does not have full rank. This implies the following: There exists an $s \in \mathbb{C}$, such that $\begin{bmatrix} sI - A & b \end{bmatrix}$ does not have full rank, if and only if the system is uncontrollable. Note that such an s must be an eigenvalue of A, because if s is not an eigenvalue of A, $(sI - A)$ always has full rank, by definition of eigenvalues (and therefore $\begin{bmatrix} sI - A & b \end{bmatrix}$ always has full (row) rank if s is not an eigenvalue of A.).

From the above, we conclude that the system is controllable iff for all $s \in \mathbb{C}$, $\begin{bmatrix} sI - A & b \end{bmatrix}$ has full rank.

Solution to Problem q:invpendulum1   Inverted Pendulum

a) First it is necessary to eliminate $\dot{v}$:

$$m\left(-mg\theta_1/M - mg\theta_2/M + l_1\ddot{\theta}_1 + u/M \right) = mg\theta_1$$
$$m\left(-mg\theta_1/M - mg\theta_2/M + l_2\ddot{\theta}_2 + u/M \right) = mg\theta_2$$

then with state variables

$$x_1 = \theta_1, \qquad x_2 = \theta_2, \qquad x_3 = \dot{\theta}_1, \qquad x_4 = \dot{\theta}_2$$

we have

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_1 & a_2 & 0 & 0 \\ a_3 & a_4 & 0 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ -1/(Ml_1) \\ -1/(Ml_2) \end{bmatrix}$$

$$a_1 = \frac{(M+m)g}{Ml_1} \qquad\qquad a_2 = \frac{mg}{Ml_1}$$

$$a_3 = \frac{mg}{Ml_2} \qquad\qquad a_4 = \frac{(M+m)g}{Ml_2}$$

b) When the two lengths are identical, $l_1 = l_2$, $a_1 = a_4$ and $a_2 = a_3$. These conditions cause A and b to be

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_1 & a_2 & 0 & 0 \\ a_2 & a_1 & 0 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ b_1 \\ b_1 \end{bmatrix}$$

$$a_1 = \frac{(M+m)g}{Ml_1} \qquad a_2 = \frac{mg}{Ml_1}$$

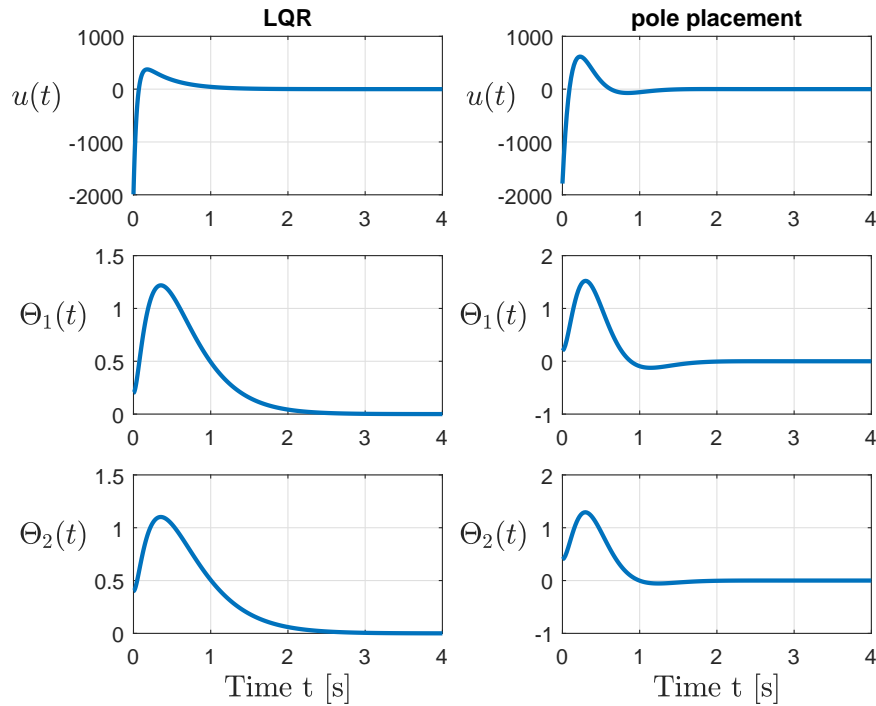c) This then causes the last two rows of the controllability matrix to be identical:

$$\mathcal{C} = \begin{bmatrix} b & Ab & A^2b & A^3b \end{bmatrix}$$

$$Ab = \begin{bmatrix} b_1 \\ b_1 \\ 0 \\ 0 \end{bmatrix}, A^2b = \begin{bmatrix} 0 \\ 0 \\ \begin{bmatrix} a_1 & a_2 \\ a_2 & a_1 \end{bmatrix}\begin{bmatrix} b_1 \\ b_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tilde{b}_1 \\ \tilde{b}_1 \end{bmatrix}, A^3b = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} 0 & b_1 & 0 & \tilde{b}_1 \\ 0 & b_1 & 0 & \tilde{b}_1 \\ b_1 & 0 & \tilde{b}_1 & 0 \\ b_1 & 0 & \tilde{b}_1 & 0 \end{bmatrix}$$

$\mathcal{C}$ is singular: the controllable subspace is defined by $x_1 = x_2, x_3 = x_4$, that is with $\theta_1 = \theta_2$ and $\dot{\theta}_1 = \dot{\theta}_2$

Solution to Problem q:invpendulumdesign   Inverted Pendulum - feedback controller

Figure C.9: response to initial conditions using LQR with Q and R

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \qquad R = \begin{bmatrix} 0.003 \end{bmatrix}$$

MATLAB: see `cs2_pendulums.m`

Solution to Problem q:simtransCO   Transformation into controller form

a)   We are looking for a transformation matrix T, such that

$$A_c = T^{-1}AT \qquad b_c = T^{-1}b \qquad c_c = cT$$

where $\{A_c, b_c, c_c\}$ are in the controller form.

with $T = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}$

$$b = Tb_c = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = t_3$$

b)   To calculate t_1, t_2

$$AT = TA_c$$

$$\begin{bmatrix} At_1 & At_2 & At_3 \end{bmatrix} = TA_c = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}$$

$$At_3 = t_2 - a_2 t_3$$

$$At_2 = t_1 - a_1 t_3$$

c)   Using the results from a) and b),

$$t_3 = b$$

$$t_2 = Ab + a_2 b$$

$$t_1 = At_2 + a_1 b = A^2 b + Aba_2 + ba_1$$

$$t_1 = \begin{bmatrix} b & Ab & A^2 b \end{bmatrix} \begin{bmatrix} a_1 & a_2 & 1 \end{bmatrix}^T$$

$$t_2 = \begin{bmatrix} b & Ab & A^2 b \end{bmatrix} \begin{bmatrix} a_2 & 1 & 0 \end{bmatrix}^T$$

$$t_3 = \begin{bmatrix} b & Ab & A^2 b \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$T = \mathcal{C} \begin{bmatrix} a_1 & a_2 & 1 \\ a_2 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

d)   The second matrix on the right hand side is always invertible. So T is invertible, and therefore is an allowable transformation matrix, if and only if $\mathcal{C}$ is invertible.

Solution to Problem q:controllability2   **Controllability form**

The differential equations are

$$\dot{x}_1 = u - a_0 x_2$$

$$\dot{x}_2 = x_1 - a_1 x_2$$

$$y = g_1 x_1 + g_2 x_2$$

$$A = \begin{bmatrix} 0 & -a_0 \\ 1 & -a_1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad c = \begin{bmatrix} g_1 & g_2 \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} b & Ab \end{bmatrix} = I$$

The transfer function is $c(sI - A)^{-1}b$

$$(sI - A) = \begin{bmatrix} s & a_0 \\ -1 & s + a_1 \end{bmatrix}, \qquad (sI - A)^{-1} = \frac{1}{s^2 + a_1 s + a_0} \begin{bmatrix} s + a_1 & -a_0 \\ 1 & s \end{bmatrix}$$

so

$$c(sI - A)^{-1}b = \frac{1}{s^2 + a_1 s + a_0} \begin{bmatrix} g_1 & g_2 \end{bmatrix} \begin{bmatrix} s + a_1 & -a_0 \\ 1 & s \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{g_1(s + a_1) + g_2}{s^2 + a_1 s + a_0}$$

Solution to Problem q:statefeedbackC   Controllability with state feedback

Consider the system with state feedback

$$\dot{x} = Ax + bu, \qquad y = cx \qquad u = fx + u_v$$

If the system is controllable, there exists a similarity transformation $x = T_{co}x_{co}$ such that the matrices A_{co}, b_{co} are in the controllability form

$$\dot{x}_{co} = A_{co}x_{co} + b_{co}u, \qquad y = c_{co}x_{co} \qquad u = fT_{co}x_{co} + u_v$$

because $b_{co} = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix}^T$.

Only the final row of A is changed by closing the loop

$$\dot{x}_{clco} = (A_{co} + b_{co}fT_{co})x_{clco} + b_{co}u_v$$

So the closed loop A matrix

$$A_{clco} = A_{co} + b_{co}fT_{co}$$

is in the controllability form. The closed loop b matrix b_{co} is also in the controllability form (in fact it has not changed). So the system with feedback is still controllable.

C.3   Chapter 4

Solution to Problem q:stabdetect   Stabilisability and detectability

The controllability matrix is

$$\begin{bmatrix} 1 & -2 \\ 1 & 2 \end{bmatrix}$$

Which has rank 2, so the system is controllable, and hence stabilizable.
The observability matrix is

$$\begin{bmatrix} 2 & 0 \\ -4 & 0 \end{bmatrix}$$

which has rank 1; from the standard unobservability form, the mode associated with the eigenvalue $+1$ is unobservable. It is also unstable, so the system is undetectable.

Solution to Problem q:pendulum2   Single inverted pendulum with friction

a)   The states and the Matrices A and b of the state space model are

$$x^T = \begin{bmatrix} d & \dot{d} & d + L\theta & \dot{d} + L\dot{\theta} \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -F/M & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -g/L & 0 & g/L & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 0 \end{bmatrix}$$

The angle θ is the output of the system. So

$$c = \begin{bmatrix} -1/L & 0 & 1/L & 0 \end{bmatrix}$$

b) A is in a *Block-triangular-Form* so the eigenvalues of A are the eigenvalues of the diagonal blocks. The eigenvalues are then the solutions of

$$s(s + F/M) = 0$$
$$s^2 - gL = 0$$

The eigenvalues are $\{0, -F/M, \pm\sqrt{g/L}\}$

c)   With the values $M = 2$, $F = 0.1$ $L = 0.5$ The controllability and observability matrices may be calculated using the **Matlab** commands `ctrb` and `obsv`.

It is also possible to check these properties using the PBH test: The advantage of this approach is that it can be used more immediately to see which modes are controllable/observable.

A condition for controllability is that there does not exist a vector $q \neq 0$ such that

$$q^T A = \lambda q^T$$
$$A^T q = \lambda q$$

i.e. q is the left eigenvector of A^T and $q^T b = 0$. In `Matlab`:

```
>> [Q,D]=eig(A');
>> Q=[q_1 q_2 q_3 q_4]
```

To check controllability calculate $q^T b$: In `Matlab`

```
>> Q'*b
```

For all 4 eigenvectors $q^T b \neq 0$ so the system is controllable.

The PBH test for observability is the dual of the test for controllability:

A system is unobservable if and only if there exists a vector p such that

$$Ap = \lambda p, \qquad cp = 0$$

p is a right eigenvector of A. In `Matlab`:

```
>> [P,D]=eig(A);
>> test=c*P
```

$cp = 0$ for the eigenvector with eigenvalue=0, so the unobservable mode has eigenvalue=0.

The tests are used in the `Matlab` file `cs3_pendel.m`

d) With the new state variables, from the original state space model the following relations follow:

$$\ddot{d} = -\frac{F}{M}\dot{d} + \frac{1}{M}u$$

$$\ddot{d} + L\ddot{\theta} = -\frac{g}{L}d + \frac{g}{L}d + \frac{g}{L}L\theta$$
$$L\ddot{\theta} = g\theta - \ddot{d}$$

So with the new state vector $\begin{bmatrix} \dot{d} & L\theta & L\dot{\theta} & d \end{bmatrix}^T$:

$$\dot{x}_1 = \ddot{d} = -\frac{F}{M}x_1 + \frac{1}{M}u$$
$$\dot{x}_2 = L\dot{\theta} = x_3$$
$$\dot{x}_3 = L\ddot{\theta} = g\theta + \frac{F}{M}\dot{d} - \frac{1}{M}u = \frac{F}{M}x_1 + \frac{g}{L}x_2 - \frac{1}{M}u$$
$$\dot{x}_4 = \dot{d} = x_1$$

The state space matrices are

$$
\bar{A} = \begin{bmatrix}
-\frac{F}{M} & 0 & 0 & \vdots & 0 \\
0 & 0 & 1 & \vdots & 0 \\
\frac{F}{M} & \frac{g}{L} & 0 & \vdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
1 & 0 & 0 & \vdots & 0
\end{bmatrix} = \begin{bmatrix} \bar{A}_o & 0 \\ \bar{A}_{21} & \bar{A}_{\bar{o}} \end{bmatrix}
$$

$$
\bar{b} = \begin{bmatrix} \frac{1}{M} & 0 & -\frac{1}{M} & 0 \end{bmatrix}^T
$$

$$
\bar{c} = \begin{bmatrix} 0 & \frac{1}{L} & 0 & \vdots & 0 \end{bmatrix} = \begin{bmatrix} c_o & 0 \end{bmatrix}
$$

The matrices are in the form of the Kalman decomposition and $\bar{A}_{\bar{o}}$ - which is zero - represents the unobservable state d. Physically, this means that it is not possible to establish how far the cart has moved by measuring the angle alone.

e)   With the measurements $d + L\theta$ the system becomes observable

$$
c = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}
$$

$$
\text{rank } \mathcal{O} = 4
$$

This can also be shown, for example, using the PBH observability test. The system is simulated in `cs3_pendel.m`

Solution to Problem q:simtransformonlyif   Necessity of existence of T for two state space realizations with identical transfer functions

a)   The transform from S_1 to the controller form S_c is T_{1c} and the transform from S_c to S_2 is T_{2c}^{-1}. Hence the transform from S_1 to S_2 is $T_{2c}^{-1} T_{1c}$.

b)   There is a one-to-one correspondence between a transfer function and the controller canonical form, so if two systems represent different transfer functions it is not possible to construct a T that 'connects' them. Hence two minimal systems represent the same transfer function if and only if there is a similarity transform that connects them.

Solution to Problem q:zpCancelationandssmodel   Effect of p-z cancellation

a) the term $(s + 1)$ in the numerator and denominator, cancel each other and this cancellation corresponds to an uncontrollable mode, an unobservable mode or an uncontrollable-unobservable mode.

b) The controller and observer canonical forms are calculated from the coefficients of numerator and denominator of the transfer function:
Controller canonical form

$$A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \qquad\qquad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$c = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Observer canonical form

$$A = \begin{bmatrix} 0 & -2 \\ 1 & -3 \end{bmatrix} \qquad\qquad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$c = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Two state space models corresponding to the given transfer function are constructed; the controller canonical form which is always controllable and the observer canonical form which is always observable. We also know that controllability and observability are properties of a linear system and are invariant under similarity transformation. On the other hand, the p-z cancellation represents an uncontrollable or/and unobservable mode. Hence we predict that the above controller canonical model is not observable and the above observer canonical model is not controllable. We also predict there does not exist a similarity transformation to convert one of these models into the other. These two state space realizations belong to different systems, although they have an identical transfer function.

C.4 Chapter 5

Solution to Problem q:pendobs   Observer design using Matlab

a)   The model is in the controllable canonical form form. Thus, the gain is $f = \begin{bmatrix} -1.5 & -1 \end{bmatrix}$.

b)   The state observer gain matrices are:

  i) $\begin{bmatrix} 20 & 201 \end{bmatrix}^T$

  ii) $\begin{bmatrix} 2 & 3 \end{bmatrix}^T$

c)   In order to obtain the closed loop bode plot we should first construct the closed loop system, which is given as

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & bf \\ -lc & A+bf+lc \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} b & 0 \\ 0 & -l \end{bmatrix} \begin{bmatrix} d \\ n \end{bmatrix}$$

$$y = \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

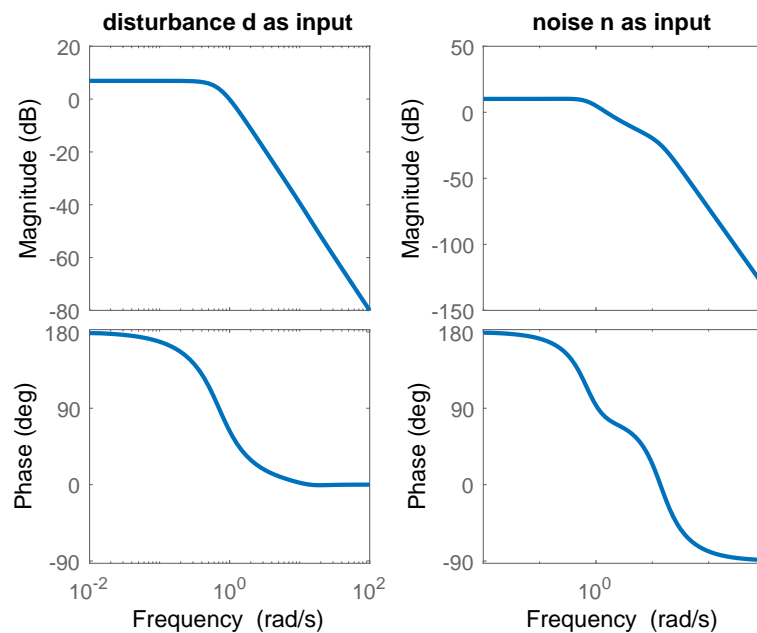the Bode plot can then be generated from the above closed loop model.

Figure C.10: bode plot for observer poles $s = -10 \pm 10j$

d)   The faster observer poles lead to faster response of the plant, but at the expense of increased sensitivity to high frequency noise: this is evident in the time and frequency domain responses.
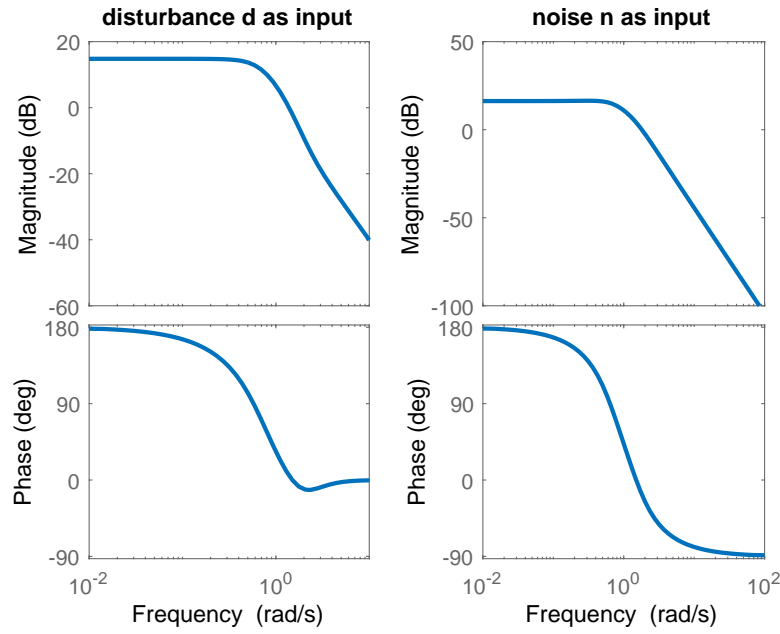
Figure C.11: bode plot for observer poles $s = -1 \pm 1j$

e)   One possible pair of poles is $-3 \pm 3j$

Solution to Problem q:mvzeros   **Zeros of state space model**

a)   The system is in the controllable canonical form, so the solutions of the equation

$$s^3 + 4s^2 + 5s + 2 = 0$$

are the poles $s = -1, s = -1, s = -2$.

The solutions of the equation

$$c_3 s^2 + c_2 s + c_1 = 0$$
$$s^2 + 4s + 5 = 0$$

are the (complex) zeros $-2 \pm 1j$.
We can also find the zeros by use of the theorem 5.1.

b)   By existence of a zero, u_0 can be selected arbitrarily. We set $u_0 = 1$, which gives

$$x_0 = (zI - A)^{-1}b$$

The corresponding initial values are complex conjugates:

With $z_1 = -2 + i$

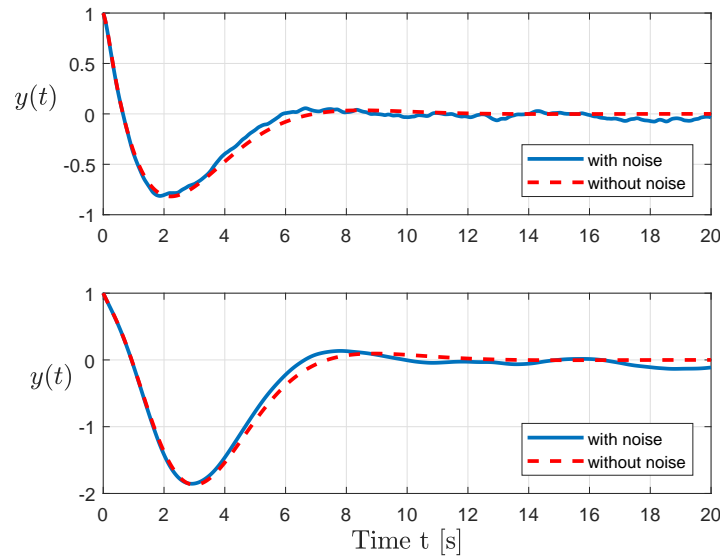$$x_0 = \begin{bmatrix} 0.5 \\ -1 + 0.5i \\ 1.5 - 2i \end{bmatrix}$$

Figure C.12: closed loop system response with and without white noise with observer poles $s = -10 \pm 10j$ (top) and $s = -1 \pm 1j$ (bottom)

With $z = \bar{z}_1 = -2 - i$

$$\bar{x}_0 = \begin{bmatrix} 0.5 \\ -1 - 0.5i \\ 1.5 + 2i \end{bmatrix}$$

To cope with the complex signals, we make use of the principle of superposition to find real signals for the initial condition and control input. As the system is linear if we have a linear combination for the initial conditions x_0 and $\bar{x}_0$, and at the same time, a similar combination for their corresponding control inputs, the output of the system and the solution for x, are expected to be again a combination of the results of each individual case, in the same fashion. Thus, such combinations again produce zero output.

We form two sets of such combinations. In the first set, the *real* initial condition and the corresponding *real* input are

$$x_{01} = (x_0 + \bar{x}_0)/2 = \begin{bmatrix} 0.5 & -1 & 1.5 \end{bmatrix}^T, \quad u_1 = u_0(e^{z_1 t} + e^{\bar{z}_1 t})/2 = e^{-2t}\cos(t)$$

and in the second set, the *real* initial condition and the corresponding *real* input are

$$x_{02} = (x_0 - \bar{x}_0)/2i = \begin{bmatrix} 0 & 0.5 & -2 \end{bmatrix}^T, \quad u_2 = u_0(e^{z_1 t} - e^{\bar{z}_1 t})/2i = e^{-2t}\sin(t)$$

c)   Simulation: see `cs4_zero.m`

Solution to Problem q:spfollow   Setpoint tracking design with static prefilter

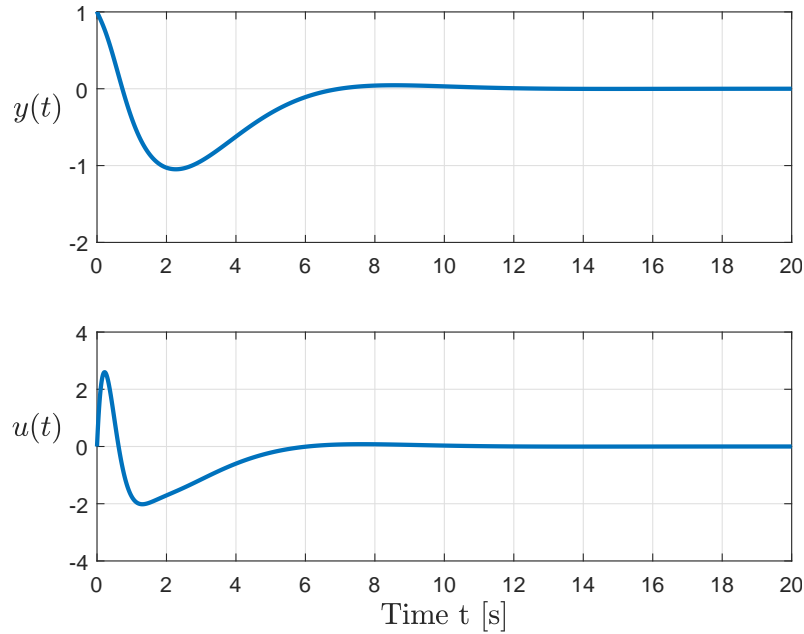Matlab and Simulink files: `cs4_ssprefilt.m`, `cs4_ssprefiltmod.mdl`.

Figure C.13: response with $|y(t)| < 0.05$ for $t > 7.0$ sec where however $u(t)$ violates its constraint

a)   This is similar to previous exercises: see Matlab file `cs4_ssprefilt.m`

b)   We have

$$\dot{\hat{x}} = A\hat{x} + bf\hat{x} + lc(\hat{x} - x) + bvr$$
$$\dot{x} = Ax + b(vr + f\hat{x})$$

Then with $\tilde{x} = x - \hat{x}$ the closed-loop system is

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A + bf & -bf \\ 0 & A + lc \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} bv \\ 0 \end{bmatrix} r$$
$$y = \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$$

The error states are not controllable, so the system $(A + bf, b, c)$ defines the closed loop transfer function $G_{cl}(s) = c(sI - A - bf)^{-1}bv$

$$G_{cl}(s) = v \cdot 4 \frac{(s + 2.5)}{(s^2 + 6s + 18)}$$

c) $G_{cl}(0) = 1$ is the condition used to calculate v. From (b)

$$G_{cl}(s) = c(sI - A - bf)^{-1}bv$$

$$v = -\frac{1}{c(A + bf)^{-1}b} = 1.8$$

d) The closed-loop system from r to y is now

$$\bar{A}_{cl} = \begin{bmatrix} A & bf \\ -lc & A + bf + lc \end{bmatrix}, \quad \bar{b}_{cl} = \begin{bmatrix} 0 \\ l \end{bmatrix}, \quad \bar{c}_{cl} = \begin{bmatrix} c & 0 \end{bmatrix}$$

Applying the similarity transformation

$$T = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$$

we get

$$A_{cl} = \begin{bmatrix} A + bf & -bf \\ 0 & A + lc \end{bmatrix}, \quad b_{cl} = \begin{bmatrix} 0 \\ -l \end{bmatrix}, \quad c_{cl} = \begin{bmatrix} c & 0 \end{bmatrix}$$

which, unlike the first position, does not split into controllable and uncontrollable parts.

The poles are the poles of the whole closed loop system, minus any cancellations that may happen: The 'possible' poles are the eigenvalues of $A + bf$ and $A + lc$ which have already been designed: $s = -3 \pm 3i, -10 \pm 10i$.

One way of calculating the zeros is directly from the transfer function

$$G(s) = c_{cl}(sI - A_{cl})^{-1}b_{cl}$$

Another way of calculating the zeros is shown in Fig.C.14, which highlights the interaction between the components of the system.
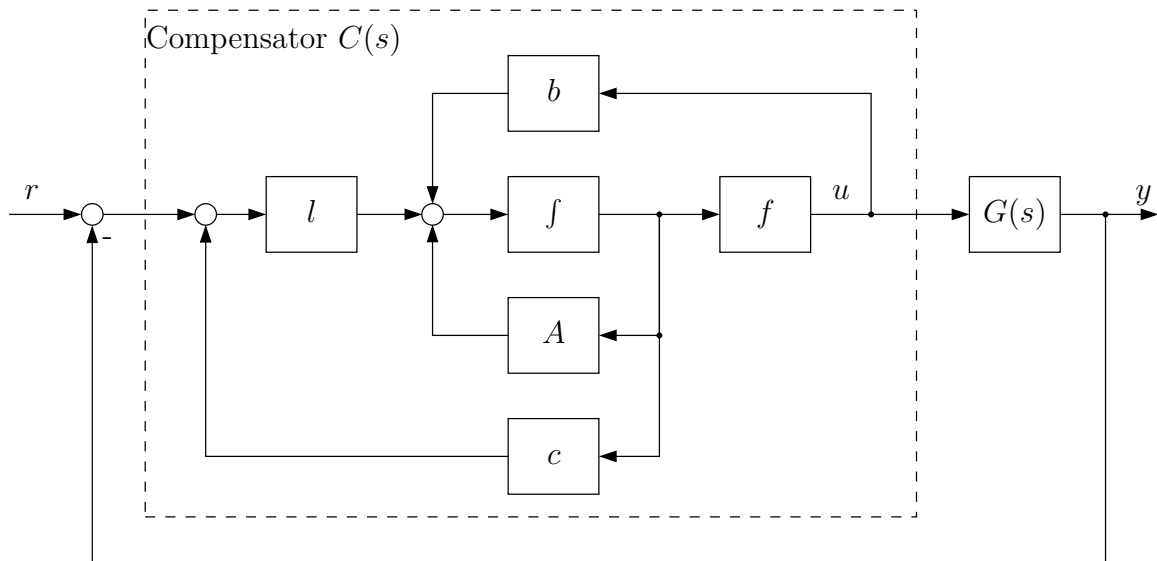

Figure C.14: Observer arrangement

The equations of the compensator $C(s)$ and the plant $\mathrm{G}(s)$ are

$$\dot{\hat{x}} = (bf + lc + A)\hat{x} + l(r - y)$$
$$u = f\hat{x}$$
$$C(s) = f(sI - A - lc - bf)^{-1}l$$
$$G(s) = c(sI - A)^{-1}b$$

It therefore follows that the closed loop system zeros are the zeros of $G(s)$ plus the zeros of $C(s)$, minus any cancellations.

$$C(s) = \frac{1490.5(s + 2.55)}{(s + 86.95)(s - 65.95)}$$
$$G(s) = 4\frac{(s + 2.5)}{(s + 2)(s + 3)}$$

In this case there are no cancellations, so the zeros are $s = -2.55, s = 2.5$

Note that although the close loop system is stable, the compensator itself is unstable, which in practice is not preferable.
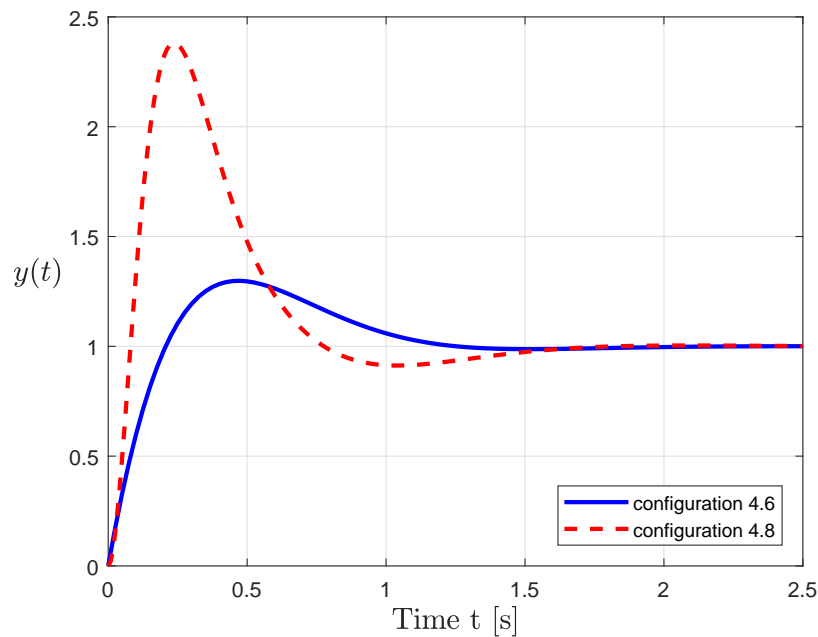
e)   Simulation: see `cs4_ssprefilt.m`

Figure C.15: step response of $r(t) = \sigma(t)$ for the observer configurations in Figures 4.6 and 4.8

The overshoot in (b) is larger due to the zeros in $C(s)$ (two zeros in the closed loop System). In part (a), zeros cancel out the poles of the observer.

f)

$$\dot{x} = Ax + bu = Ax + bf\hat{x} + bvr$$
$$\dot{\hat{x}} = (A + bf + lc)\hat{x} - ly + wr$$

$$\tilde{x} = x - \hat{x}$$
$$\dot{\tilde{x}} = (A + lc)\tilde{x} + bvr - wr$$

To make the observer errors $\tilde{x}$ independent of r, we take $w = bv$. The scalar v should be chosen so that as $t \to \infty$, $(y - r) \to 0$:

$$v = -\frac{1}{c(A + bf)^{-1}b}$$

g)   For this condition, $v = 0$, $w = l$ (w is the observer gain matrix)

Solution to Problem q:unobsstatefb   Observability under state feedback

a)   The numerator contains the zeros of the system; as the system is minimal none of these have been cancelled with poles of the system.

b)   The closed loop has the transfer function

$$c(sI - A_{cl})^{-1}b = \frac{b(s)}{a_{cl}(s)}$$

The roots of $b(s)$ are the same as the zeros of the open loop system (state feedback does not change the zeros).
The roots of $a_{cl}(s)$ are the poles of the closed loop system. These roots can be chosen arbitrarily because the system is controllable, so some of them could cancel out zeros.
If this happened the system (A, b, c) would no longer be minimal and hence not observable. We know that such a system is controllable as controllability does not change with state feedback (see problem 3.10).

Solution to Problem q:includeintegrator   Setpoint following using integrators

Matlab and Simulink files: `cs4_spfollow.m`, `cs4_spfollow_linmod.mdl`, `cs4_spfollow_int.mdl`

a)   From problem 5.3, a state space realization of the system $G(s)$ is

$$A = \begin{bmatrix} -2 & 1 \\ 0 & -3 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 & 3 \end{bmatrix}$$

and from part (a) of this problem, by designing the controller and the observer, we have

$$f = \begin{bmatrix} -5 & 4 \end{bmatrix}, \quad l = \begin{bmatrix} -238.5 \\ 74.5 \end{bmatrix}$$

The transfer function $G_{yd}(s)$ is clearly just that of the system $G(s)$ and it can be calculated directly from its state space model. Therefore we calculate

$$i) \qquad G_{yd}(s) = G(s) = 4\frac{(s+2.5)}{(s+2)(s+3)}$$

We use equation (5.8)- compare with problem 5.3 part (f)- together with $w = bv$ to obtain

$$\dot{\hat{x}} = (A + bf + lc)\hat{x} - ly + bvr$$
$$u = f\hat{x} + vr$$

By considering u as output and r as input, we have

$$A_{ur} = A + bf + lc$$
$$b_{ur} = bv$$
$$c_{ur} = f$$
$$d_{ur} = v$$

to calculate

$$ii) \qquad G_{ur}(s) = 1.8\frac{(s^2 + 20s + 200)}{(s+86.95)(s-65.95)}$$

With $r = 0$, the state space equations are

$$\dot{\hat{x}} = (A + bf + lc)\hat{x} - ly$$
$$u = f\hat{x}$$

and by considering u as output and y as input, we have

$$A_{uy} = A + bf + lc$$
$$b_{uy} = -l$$
$$c_{uy} = f$$

to obtain

$$iii) \qquad G_{uy}(s) = -1490.5\frac{(s+2.55)}{(s+86.95)(s-65.95)}$$

b)   The closed loop transfer function from r to y is

$$G_{cl}(s) = \frac{G(s)G_{yu}(s)}{1 - G(s)G_{uy}(s)}$$

(Note that the negative sign in the denominator arises because the feedback is positive!)

About the zeros of G_{ur}: In problem 5.3 part (a) we placed the eigenvalues of $A+lc$ to $-10 \pm 10j$ and we know, in this combination of the controller and the observer, the eigenvalues of $A+lc$ are uncontrollable from the control input. Thus, we expect them to be canceled out with some zeros in the closed loop transfer function. Considering the equation

$$G_{cl}(s) = \frac{G(s)G_{yu}(s)}{1 - G(s)G_{uy}(s)}$$

we see that such zeros cannot be zeros of $G(s)$ as the denominator would be $1 \neq 0$ and therefore no cancellation occurs. Thus, they must be zeros of G_{ur} i.e. zeros of $s^2 + 20s + 200$, as they are.

c)   From these transfer functions

$$\frac{Y(s)}{D(s)} = \frac{G(s)}{1 - G(s)G_{uy}(s)}$$

as $t \to \infty$, $s \to 0$

$$\frac{Y(0)}{D(0)} = \frac{G(0)}{1 - G(0)G_{uy}(0)} = -15.93$$

d)   From the simulation in `cs4_spfollow.m`, $\lim_{t\to\infty}(y - r) = 0.032$.

e)   see `cs4_spfollow_int.mdl`

f)   Before adding an integrator in parts (c) and (d), we had a non-zero steady state error. We investigate the steady state tracking error after augmentation by forming the closed-loop transfer functions. The closed-loop transfer function of the augmented system from r to y is

$$G_{cl-aug}(s) = \frac{G_{cl}(s)\frac{f_I}{s}}{1 + G_{cl}(s)\frac{f_I}{s}} = \frac{f_I G_{cl}(s)}{s + f_I G_{cl}(s)}$$

which as s goes to zero, goes to 1. This simple calculation shows that even if we have any modeling uncertainty, the steady state error would always be zero after adding an integrator.

We check disturbance rejection in steady state by forming the closed-loop transfer function of the augmented system from d to y

$$Gd_{cl-aug}(s) = \frac{G_{cl}(s)}{1 + G_{cl}(s)frac f_I s} = \frac{s G_{cl}(s)}{s + f_I G_{cl}(s)}$$

which goes to 0 when s goes to 0, indicating zero steady state error.

Solution to Problem q:symrootlocus   Optimal controller and observer design with symmetric root locus

a)   First it should be clear that $c_z = \begin{bmatrix} 2 & 1 \end{bmatrix}$.

The transfer function equivalent of state space representation A, b, c_z is calculable by hand or using Matlab commands `tf` on the system form of the plant

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

The transfer function to be used in the symmetric root locus is

$$G_{ss}(s) = \frac{(s-2)(-s-2)}{(s^2-1)(s^2-1)}$$

The simulation of the root locus for G_{ss} is in `cs4_6symrl.m`
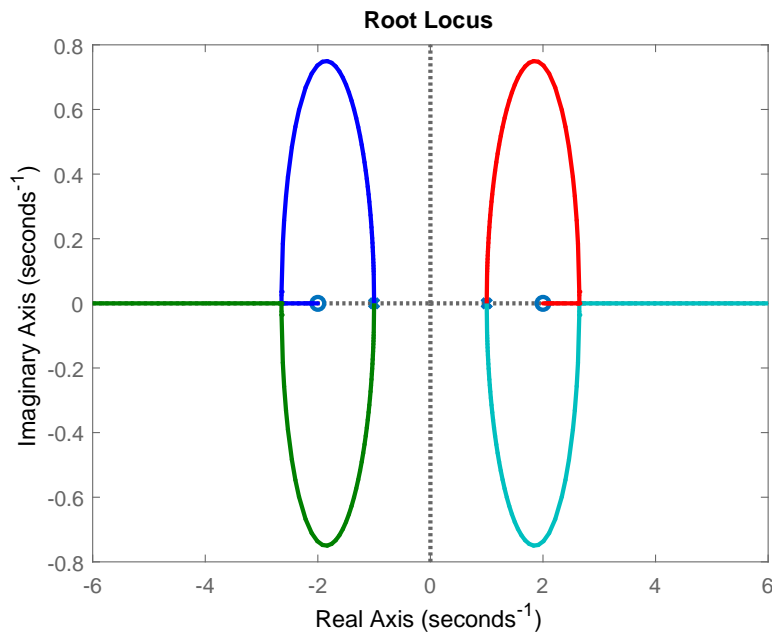

Figure C.16: root locus for G_{ss}

b)   Simulation with `cs4_6symrlsysmod.mdl` $\rho \approx 0.6$

c)   The transfer function for use in getting the symmetric root locus for the observer is
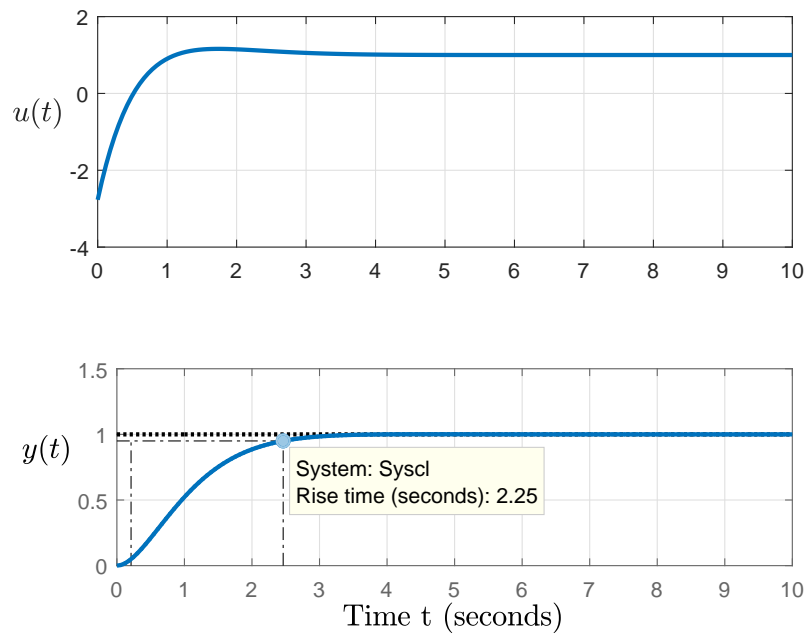
$$G_n(s) = \frac{0.1s+1}{(s^2-1)}$$

Figure C.17: step response $y(t)$ with zero steady state tracking error, 95% rise time less than 2.5 s and $|u(t)| < 3.0$

$$G_{nn}(s) = \frac{(0.1s + 1)(-0.1s + 1)}{(s^2 - 1)(s^2 - 1)}$$

The root locus for G_{nn} is in `cs4_6symrl.m`

d) Simulation with `cs4_6symrlKFnoisemod.mdl`. Note that to simulate a white noise process with spectral density S_0 and a sampling time τ the value of `noise power` in the Simulink block `band limited white noise` is τS_0.

e) Simulation with `cs4_6symrlKFdistmod.mdl`. The required value of q is approx. 1.7. Increasing q leads to better disturbance rejection. This is to be expected as greater q corresponds to greater noise power, which in the optimal formulation puts a greater emphasis on rejecting state noise rejection at the expense of output noise rejection.
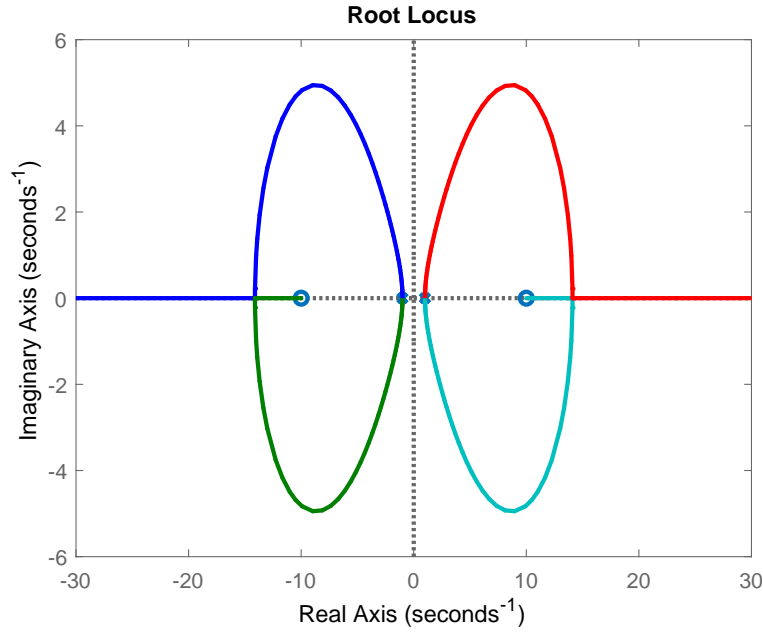
Solution to Problem q:symrllimits   Roots of SRL at limits of ρ

a) In the limit
$$a_c(s)a_c(-s) = a(s)a(-s)$$

for stable poles the feedback is small as little 'effort' is used, there is little movement of the poles.
Unstable poles are moved to their mirror image position in the left half plane.

Figure C.18: root locus for G_{nn}

Interpretation for controller: Stable poles need not be moved if energy is to be minimised; controller K is small. For minimal energy unstable poles are moved to their mirror image location in the LHP.

Interpretation for Kalman filter: With a lot of noise in the output and relatively little state noise ($q \to 0$), The states are best estimated without using the output at all ($l \to 0$).

b) i)

$$a(s) = s^2 + 1, \quad a(-s) = s^2 + 1$$
$$b(s) = s + 1, \quad b(-s) = -s + 1$$

so the roots of the symmetric root locus lie at roots of

$$s^4 + 2s^2 + 1 + \frac{1}{\rho}(-s^2 + 1)$$

so total no. of roots $= 4$, as $\rho \to 0$ two roots tend towards the roots of $(-s^2+1)$, so closed loop root of system is at -1 (i.e. the stable root).

ii) For the 'large' values of s and corresponding small values of ρ

$$s^4 + (2s^2 + 1) + \frac{1}{\rho}(-s^2 + 1)$$
$$\approx s^4 + 2s^2 - \frac{1}{\rho}s^2 = s^4 + (2 - \frac{1}{\rho})s^2$$
$$\approx s^4 - \frac{1}{\rho}s^2$$
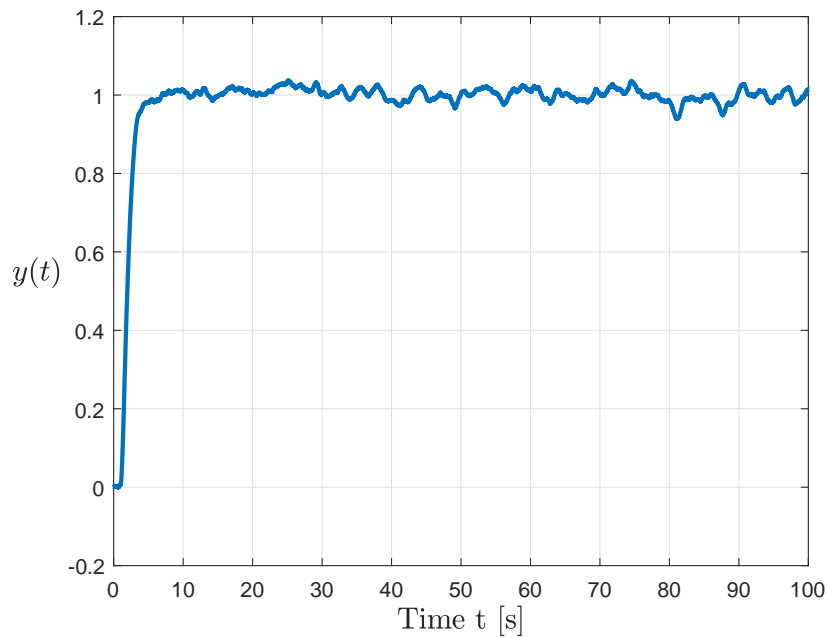
Figure C.19: response of the closed-loop system to a unit step input

Now the four solutions are the roots of

$$s^4 - \frac{1}{\rho}s^2 \approx 0$$

The two 'small' roots are the zero solutions of this equation found in part (i). The other 2 roots are the roots of

$$s^2 = \frac{1}{\rho}, \qquad s = \pm\frac{1}{\sqrt{\rho}}$$

iii) Physical explanations

Controller, $\rho \to 0$: With no limitations on the input, K can be large, one pole moves as far as possible into the left half plane. meanwhile the other pole tends to cancel the system zero.

Observer, $q \to \infty$: here there is relatively little noise in the output but a relatively large amount of state noise, the states are best reconstructed with a fast observer, i.e. one with a lot of feedback (large l).

Solution to Problem q:srlgeneral   Generalisation of SRL results

a) $m < n$ for realizable systems.
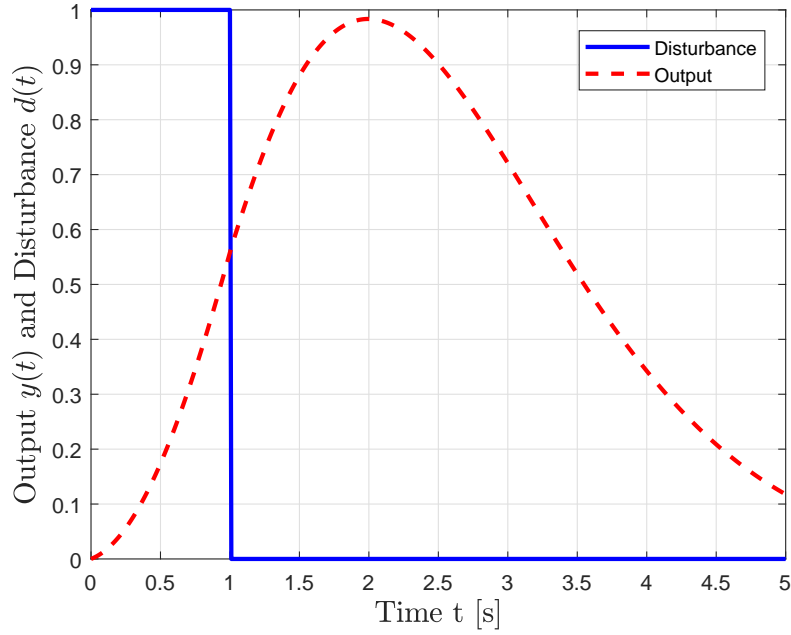
Figure C.20: Simulation for part e

$$a_c(s)a_c(-s) = a(-s)a(s) + \frac{1}{\rho}b(s)b(-s)$$

$$= a^{2n} + \ldots a_0 + b_0$$

Total no. of roots $= 2n$

$$a_c(s)a_c(-s) \to \frac{1}{\rho}b(s)b(-s)$$

so $2m$ roots are roots of $b(s)b(-s)$.

The remaining $2(n-m)$ must become very large to fulfil the equations as $\rho \to 0$.

b)

$$a(-s)a(s) \approx (-1)^n s^{2n}, \qquad b(-s)b(s) \approx (-1)^m b_m^2 s^{2m}$$

so

$$(-1)^n s^{2n} + \frac{1}{\rho}(-1)^m b_m^2 s^{2m} \approx 0$$

c) From (b)

$$s^{2(n-m)} = (-1)^{m-1-n}\frac{1}{\rho}b_m^2$$

d) Also from (c)

$$s^2 = \frac{1}{\rho}b_m^2$$

The left half plane solution is at $-r$.

e) Also from (c)

$$s^4 = -\frac{1}{\rho}b_m^2$$

The left half plane solutions lie on the circle $\frac{1}{\sqrt{2}}r(-1 \pm j)$.

Solution to Problem q:srlimroots   Symmetric roots on imaginary axis

a) On the imaginary axis $s = jw$

$$1 + \frac{1}{\rho}G(j\omega_0)G(-j\omega_0) = 0$$

$$a(jw)a(-jw) + \frac{1}{\rho}b(jw)b(-jw) = 0$$

$$|a(jw)|^2 + \frac{1}{\rho}|b(jw)|^2 = 0$$

b) Having

$$|a(jw)|^2 + \frac{1}{\rho}|b(jw)|^2 = 0$$

from part (a), implies:
$\Rightarrow$ $a(jw) = 0$ and $b(jw) = 0$
Thus, $a(jw)$ and $b(jw)$ must have roots at jw:

$$a(s) = \tilde{a}(s)(s - jw) \quad \text{and} \quad b(s) = \tilde{b}(s)(s - jw)$$

c) From parts (a) and (b) we have

$$\frac{b(s)}{a(s)} = \frac{\tilde{b}(s)(s - jw)}{\tilde{a}(s)(s - jw)}$$

We see that there is a cancellation of between numerator and denominator on the imaginary axis for there to be solutions for the symmetric root locus on the imaginary axis. Such an *unstable* cancellation can never exist for a system that is stabilizable and detectable.

Solution to Problem q:minsegLqrsim   Mini Segway simulation

a) i&ii)   Shown in figure C.21 is the response to the initial condition $[0; 9\pi/180; 0; 0]^T$. By setting the Q=diag(1000,1000,10,1) and R=0.1 the desired performance can be achieved.
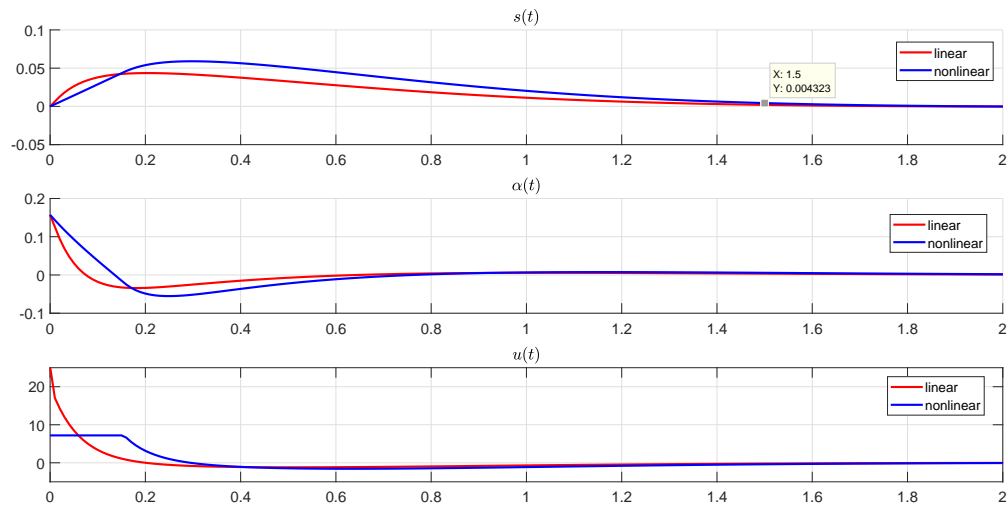
Figure C.21: Initial Perturbation

iii)   On increasing the initial perturbation of the inverted pendulum angle, it is
found that the error between the linear and the nonlinear response increases.
This is due to the fact that the system was linearised about the vertical position
and so moving away from this position reduces the accuracy of our linearised
model.

Solution to Problem q:minsegLqrexp   Mini Segway experiment

a)   Shown in figure C.22 is the tracking of a sine reference input with frequency $0.4\,\mathrm{rad/s}$
and amplitude $0.2\,\mathrm{m}$. It is shown that after fine tuning of the controller a good ex-
perimental performance could be achieved.
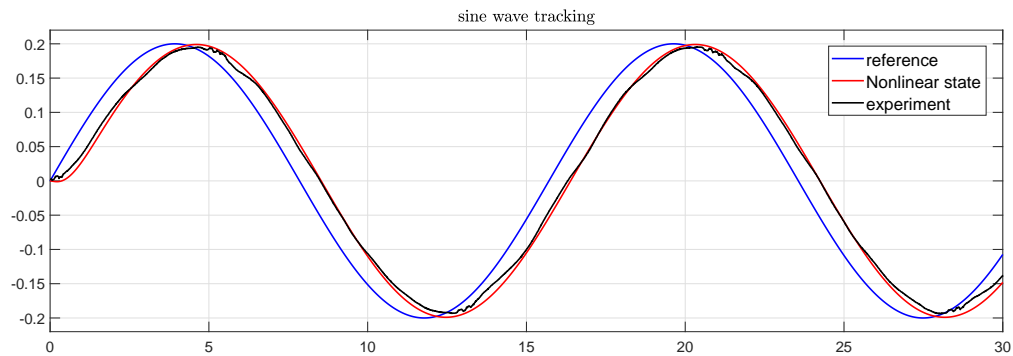
Figure C.22: Experiment LQR

C.5 Chapter 6

Solution to Problem q:Charloci **characteristic loci** There are two characteristic loci as the dimension of the system is 2×2.

The larger eigenvalue locus crosses the negative real axis at about -0.058, so the maximum value of k is approximately 17.

Solution to Problem q:mvST **Sensitivity S and complementary sensitivity T**

a) $G_{er}(s)$:
$$e = r - GKe$$
$$G_{er}(s) = S = (I + GK)^{-1}$$

$G_{ydo}(s)$:
$$y = d_o - GKy$$
$$(I + GK)y = d_o$$
$$G_{ydo} = (I + GK)^{-1}$$

b) $G_{yr}(s)$:
$$y = GKr - GKy$$
$$y = (I + GK)^{-1}GKr = \left\{ G^{-1}(I + GK) \right\}^{-1} Kr = \{G^{-1} + K\}^{-1} Kr$$
$$= \left\{ (I + KG)G^{-1} \right\}^{-1} Kr$$
$$= G(I + KG)^{-1} Kr$$

$$y = (I + GK)^{-1}GKr = \left\{ (GK)^{-1}(I + GK) \right\}^{-1} r = \left\{ (GK)^{-1} + I \right\}^{-1} r$$
$$= \left\{ (I + GK)(GK)^{-1} \right\}^{-1} r$$
$$= GK(I + GK)^{-1} r$$

c)
$$u_g = d_i - KGu_g$$
$$S_I = (I + KG)^{-1}$$

$$SG = (I + GK)^{-1}G$$
$$= \left\{ G^{-1}(I + GK) \right\}^{-1} = \left\{ (I + KG)G^{-1} \right\}^{-1}$$
$$= G(I + KG)^{-1} = GS_I$$

Solution to Problem q:tgen   Pole placement design for turbogenerator

The Matlab solutions is given in files `cs5_tgen_Place.m` and `cs5_tgensfmod.mdl`

a)   Block diagram (Figure C.23):

Figure C.23: Turbogenerator closed loop block diagram

$r \in R^2$, $y \in R^2$, $d \in R^2$, $x_O \in R^4$, $x_l \in R^2$ and $A \in R^{4 \times 4}$, $B \in R^{4 \times 2}$, $C \in R^{2 \times 4}$, $F_I \in R^{2 \times 2}$, $F \in R^{2 \times 4}$, $L \in R^{4 \times 2}$.

The reason why the obsesrver-based state feedback controller can handle small variations in model parameters (A, B, C, D) without zero steady state error is the is the same why this kind of controller structure can handle step changes in set points and step disturbances as well. Small changes in model parameters can also be modelled as input disturbances d. And a change in the disturbance d will excite the integral dynamics. The integrator output changes until its input, which is composed of the system error $r - y$, is zero.

b)   For this design it is only necessary to consider the poles of the controller, as the response to setpoint changes is not affected by the observer, with the 6 poles positions $-1 \pm 0.5j$, $-2 \pm 1.0j$, -3, -3.

The settling time for $r_1 = 2.45$, for $r_2 = 1.75$
maximum (y_2) with change in $r_1 = 0.11$
maximum (y_1) with change in $r_2 = 0.06$

See simulation `cs5_tgen_Place.m` and Simulink model `cs5_tgensfmod.mdl`, when the switch is in "state feedback" position

c) For this part the observer dynamics must be considered because the observer dynamics are controllable from d.

4 observer poles that fulfill this are: $-10 \pm 10j$, $-15 \pm 15j$

Solution to Problem q:STplots   S and T

a) Check closed loop pole positions: `cs5_tgen_SV_plots.m`

b) Singular value plots: also in `cs5_tgen_SV_plots.m` which uses `cs5_tgen_loop.mdl`

Explanation S: At low frequency one of the singular values has a value tending to zero, the other has a finite value.
This is due to the presence of an integrator in only one channel: at steady state the error in channel 1 is always zero. For a particular input direction the steady state channel 2 is also zero.

The steady state gain sensitivity matrix of the system is

$$S_0 = \begin{bmatrix} 0 & 0 \\ -0.1051 & 0.2832 \end{bmatrix}$$

The integral action enforces the first row to be zero. The rank of the matrix is clearly one, so one of the singular values is zero. This singular value corresponds to a direction defined by the null space of this matrix.

The maximum singular value at low frequencies corresponds to a direction where the steady state error of the second output is at its maximum.

At high frequencies both singular values tend to 1 because at high frequencies the feedback path has very low gain.

T at low frequencies T tends to $I - S_0$. Both singular values are close to 1 as there is reasonable setpoint following of both setpoints at low frequencies.

T at high frequencies tends to zero as the closed loop plant is strictly proper.

c) At low frequencies the input directions are the columns of the V matrix of the singular value decomposition of S_0. They correspond to the directions of the input setpoint vector corresponding to the error in channel 2 to being maximised and the error in channel 2 being zero.

The output directions are the vectors of the U matrix of the SVD of S_0. The direction corresponding to the maximum singular value is $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ i.e. it is completely in the direction of setpoint or output 2. The direction corresponding to the minimum singular value is the direction of setpoint or output 1.

Solution to Problem q:scaling   Scaling

a) Scaling of error: the scale in each channel is the reciprocal of the maximum error to make the scaled output have magnitude 1

$$\bar{e}_{max} = S_e e_{max}$$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = S_e \begin{bmatrix} 0.2 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$S_e = \frac{1}{\sqrt{2}} \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}$$

b) First, as scaling of output is the same as scaling of error,

$$y = Gu$$

$$\bar{y} = S_e Gu = S_e G S_u \bar{u}$$

$$= \bar{G}\bar{u}$$

Scaling of inputs: the scale in each channel is equal to the maximum input in that channel multiplied by $\sqrt{2}$

$$S_u \bar{u}_{max} = u_{max}$$

$$S_u \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$S_u = \sqrt{2} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

In the scaled model, 1.0 represents the maximum value of the magnitude of the signal vector: in a 2×2 system if any signal has the value $1/\sqrt{(2)}$, it means that the system is at the limit of acceptable performance.

c) Scaling of disturbance:

$$S_d \bar{d}_{max} = d_{max}$$

$$S_d \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$S_d = \sqrt{2} \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

d) Scaling of setpoint: $\tilde{r}$ is normalised setpoint. First define the scaling between the normalised setpoint $\tilde{r}$ and the true setpoint r

$$r_{max} = S_r \tilde{r}_{max}$$

$$S_r = \sqrt{2} \begin{bmatrix} 4 & 0 \\ 0 & 0.4 \end{bmatrix}$$

The scaling between r and $\bar{r}$ is the same as that between e and $\bar{e}$:

$$\bar{r} = S_e r = S_e S_r \tilde{r}$$
$$\bar{r} = R\tilde{r}$$
$$R = \begin{bmatrix} 20 & 0 \\ 0 & 0.8 \end{bmatrix}$$

e)   For a scaled system in closed loop

$$\bar{e} = S\bar{r}$$

therefore with a scaled setpoint
$$\bar{e} = SR\tilde{r}$$

As $\tilde{r}$ has a maximum magnitude of 1, $|\bar{e}_{max}|$ is guaranteed to be less than 1 if

$$\bar{\sigma}(SR) < 1$$

which is guaranteed if
$$\bar{\sigma}(S)\bar{\sigma}(R) < 1$$

that is
$$\bar{\sigma}(S) < \underline{\sigma}(R^{-1})$$

Solution to Problem q:SMform   Smith-McMillan form

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

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

b)   Exchange row 1 and row 2

$$G_1(s) = \frac{1}{(s+1)(s+2)} \begin{bmatrix} -1 & s+1 \\ (s+2) & 2(s+2) \end{bmatrix}$$

Use elementary operations to bring a zero into position $(1,2)$ into position $(2,1)$

$row2 \rightarrow row2 + (s+2) * row1$
then
$column2 \rightarrow column2 + (s+1) * column1$

$$G_2(s) = \begin{bmatrix} \frac{-1}{(s+1)(s+2)} & 0 \\ 0 & \frac{(s+3)}{(s+1)} \end{bmatrix}$$

Solution to Problem q:KalmanCF   Kalman decomposition

See simulation `cs5_kalman_decomp.m`

Matlab commands:
```
>> [Sysm,U]=minreal(Sys)
>> Ak=U*A*U'
>> Bk=U*B
>> Ck=C*U'
```

$$
A_k = \begin{bmatrix}
-1.1667 & 0.2041 & -0.3727 & \textbf{-0.0000} & -0.4564 \\
-0.0000 & -1.0000 & -0.0000 & \textbf{0.0000} & -0.0000 \\
-0.3727 & 0.4564 & -1.8333 & \textbf{-0.0000} & -1.0206 \\
0.0000 & -1.1180 & -0.0000 & -2.0000 & 2.5000 \\
\textbf{-0.0000} & \textbf{0.0000} & \textbf{-0.0000} & \textbf{0.0000} & -1.0000
\end{bmatrix}
$$

$$
B_k = \begin{bmatrix}
0.1491 & -0.5402 \\
-1.0954 & -0.0000 \\
0.3333 & 1.2416 \\
-0.8165 & 0.4082 \\
\textbf{0.0000} & \textbf{-0.0000}
\end{bmatrix}
$$

$$
C_k = \begin{bmatrix}
-1.8257 & -0.9129 & 0.8165 & \textbf{0.0000} & -0.4082 \\
0.4472 & 0.3651 & 1.0000 & \textbf{-0.0000} & -0.8165
\end{bmatrix}
$$

The states can be $(c,o),(nc,o),(c,no)$ or (nc,no):

$$
A_k = \begin{bmatrix}
A_{c,o} & 0 & M_1 & 0 \\
M_2 & A_{c,no} & M_3 & M_4 \\
0 & 0 & A_{nc,o} & 0 \\
0 & 0 & M_5 & A_{nc,no}
\end{bmatrix}
\qquad
B_k = \begin{bmatrix}
B_{c,o} \\
B_{c,no} \\
0 \\
0
\end{bmatrix}
$$

$$
C_k = \begin{bmatrix} C_{c,o} & 0 & C_{nc,o}, & 0 \end{bmatrix}
$$

For this system there are no (nc,no) states:

$$
A_k = \begin{bmatrix}
A_{c,o} & 0 & M_1 \\
M_2 & A_{c,no} & M_3 \\
0 & 0 & A_{nc,o}
\end{bmatrix}
\qquad
B_k = \begin{bmatrix}
B_{c,o} \\
B_{c,no} \\
0
\end{bmatrix}
$$

$$
C_k = \begin{bmatrix} C_{c,o} & 0 & C_{nc,o} \end{bmatrix}
$$

There are: 3 (c,o) states,1 (c,no) state and 1 (nc,o) state.

Solution to Problem q:Gilbert   Gilbert realisation

Gilbert realisation of

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

$$G(s) = \frac{1}{d(s)} \tilde{G}(s)$$

$$d(s) = s^2 + 3s + 2$$

Solutions of $d(\lambda) = 0$ are $\lambda_1 = -1$, $\lambda_2 = -2$

$$G(s) = \frac{R_1}{s - \lambda_1} + \frac{R_2}{s - \lambda_2}$$

$$R_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \qquad\qquad R_2 = \begin{bmatrix} -2 & -1 \\ -2 & -1 \end{bmatrix}$$

$$R_1 = C_1 B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \qquad R_2 = C_2 B_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} -2 & -1 \end{bmatrix}$$

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} \qquad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ -2 & -1 \end{bmatrix}$$

$$C = \begin{bmatrix} C_1 & C_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Solution to Problem q:GilbertCM   Gilbert Realisation Controllability

It is required to show that $\mathcal{C} = \mathcal{B}\mathcal{V}$ where

$$\mathcal{V} = \begin{bmatrix} I_m & \lambda_1 I_m & \cdots & \lambda_1^2 I_m \\ I_m & \lambda_2 I_m & \cdots & \lambda_2^2 I_m \end{bmatrix}$$

with $n = 2, r = 2, m = 2$

$$\mathcal{V} = \begin{bmatrix} I_2 & \lambda_1 I_2 \\ I_2 & \lambda_2 I_2 \end{bmatrix}$$

With $r = 2$:

$$\mathcal{C} = \begin{bmatrix} B & AB \end{bmatrix}$$

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}\begin{bmatrix} I \\ I \end{bmatrix}$$

$$A = \begin{bmatrix} \lambda_1 I & 0 \\ 0 & \lambda_2 I \end{bmatrix}$$

$$AB = \begin{bmatrix} \lambda_1 B_1 \\ \lambda_2 B_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} \lambda_1 I \\ \lambda_2 I \end{bmatrix}$$

$$\mathcal{B} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}$$

$\mathcal{V}$ has full rank so the system is controllable if and only if $\mathcal{B}$ has full rank.

B_1 has full rank, B_2 has full rank as these are conditions from the Gilbert realisation. Because of the block diagonal structure $\mathcal{B}$ also has full rank. Hence $\mathcal{C}$ has full rank. A similar argument can be used to show observability.

Solution to Problem q:tgenLQR   Turbogenerator LQR

a)   Let $x = \begin{bmatrix} x_p & x_I \end{bmatrix}$. Then $y = C x_p$

$$k_1(y_1^2 + y_2^2) = k_1 \begin{bmatrix} y_1 & y_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = k_1 x_p^T C^T C x_p$$

$$k_2(x_5^2 + x_6^2) = k_2 x_I^T x_I$$

so

$$k_1(y_1^2 + y_2^2) + k_2(x_5^2 + x_6^2) = k_1 x_p^T C^T C x_p + k_2 x_I^T x_I$$

$$= \begin{bmatrix} x_p^T & x_I^T \end{bmatrix} \begin{bmatrix} k_1 C^T C & 0 \\ 0 & k_2 I_2 \end{bmatrix} \begin{bmatrix} x_p \\ x_I \end{bmatrix}$$

$$Q = \begin{bmatrix} k_1 C^T C & 0 \\ 0 & k_2 I_2 \end{bmatrix}$$

$$k_3(u_1^2 + u_2^2) = k_3 \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$R = k_3 I_2$$

b)   i)   Greater weight on state errors: faster response

ii)   Greater weight on integrated errors: quicker return to setpoint

iii)   Greater weight on inputs: slower response

These are rules of thumb, which have to be considered carefully. Increasing the weight on one cost means decreasing the weight on the other costs. Thus increasing k1 with fixed k2 and k3 can also lead to a slower response. In this case, the weight

on the state errors would be increased, which means, the weight on the integrated errors and the inputs would be decreased. If the decreasing effect on the weight on the integrated error is stronger, then the response would be slower. To be sure, that the response will become faster, the weight on the input should be decreased, in which case all effects would lead to a faster response.

Only ratios between weights affect the resulting controller. So any one of the three constants could be fixed to one without loss of generality.

c,d)   See `cs5_tgen_LQR`, `cs5_tgensfmod2.mdl`, `cs5_tgen_LQR_plot.m`. Suitable values of the tuning parameters are $k_2 = 10$, $k_3 = 0.001$.

d)   Easier to set weights than to set pole placement positions.

Solution to Problem q:RHPpolconst   Right half plane zero constraints

a)   From the Smith-McMillan form (or the Matlab commands `zero` and `pole`: see `cs5_rhand_zeros.m`)

   Pole: $-2, -1, -2, -1$

   Zeros: $+0.75$

b)   Definition of zero: $G(s)$ loses rank at $s = z_i$

$$G(z_i)u_{zi} = 0$$

$G(z_i)$ does not have full rank, so nor does $G^T(z_i)$. So there must exist a y_{zi}, such that

$$G^T(z_i)y_{zi} = 0$$
$$y_{zi}^T G(z_i) = 0$$

There is only one zero at $s = 0.75$

$$G(0.75) = \frac{1}{(0.5 \cdot \mathbf{0.75} + 1)(\mathbf{0.75} + 1)} \begin{bmatrix} 2 & 1 \\ 1 + 4 \cdot \mathbf{0.75} & 2 \end{bmatrix} = \begin{bmatrix} 0.8312 & 0.4156 \\ 1.6623 & 0.8312 \end{bmatrix}$$

$$y_{zi} = k \cdot \begin{bmatrix} -0.8944 \\ 0.4472 \end{bmatrix}$$

The singular value decomposition can be used to calculate y_{zi}. See appendix A and the MATLAB file `cs5_rhand_zeros.m`.

c) The opposite of the condition is

$$y_{zi}^T G(z_i) K(z_i) \neq 0$$

and this can only be true if $K(z_i) = \infty$, which is exactly when z_i is a pole of $K(s)$. But $K(s)$ cannot have unstable poles which the corresponding zeros of $G(s)$ could cancel.

d) Pole condition follows directly from definition of internal stability.

The opposite of the condition is

$$y_{zi}^T G(z_i) K(z_i) \big(1 + G(z_i) K(z_i)\big)^{-1} \neq 0$$

which could only be true if

$$\big(1 + G(z_i) K(z_i)\big)^{-1} = \infty$$

that is, z_i is a **Pole** of $\big(1 + G(z_i) K(z_i)\big)^{-1}$.

But $(1 + K(s)G(s))^{-1}$ cannot have poles in the right half plane due to internal stability requirement. Therefore

$$y_{zi}^T G(z_i) K(z_i) \big(1 + G(z_i) K(z_i)\big)^{-1} = 0$$

e) Let $y_{zi} = \begin{bmatrix} y_{zi1} & y_{zi2} \end{bmatrix}^T$

$$\begin{bmatrix} y_{zi,1} & y_{z2,2} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = 0$$

$$y_{zi1} T_{11}(z_i) + y_{zi2} T_{21}(z_i) = 0$$

$$y_{zi1} T_{12}(z_i) + y_{zi2} T_{22}(z_i) = 0$$

$$\alpha_1 = -0.08944, \quad \alpha_2 = 0.4472$$

Solution to Problem q:RHPpolconst2 **Second question on RHP zero constraints**

a) For setpoint tracking: $T(0) = I$

$$\begin{aligned} T_{11}(0) &= 1, & T_{12}(0) &= 0 \\ T_{21}(0) &= 0, & T_{22}(0) &= 1 \end{aligned}$$

b) For no interaction:

$$T_{12}(s) = 0, \qquad T_{21}(s) = 0$$

c)   The conditions to be fulfilled are

$$y_{z1} T_{11}(z_i) = 0, \qquad\qquad T_{12}(0) = 0$$
$$T_{11}(0) = 1$$
$$T_{21}(0) = 0, \qquad\qquad y_{z1} T_{22}(z_i) = 0$$
$$T_{22}(0) = 1$$

By inspection $T_1(s)$ does not fulfill this constraint but $T_2(s)$ does.

Valid transfer functions always have an inverse response: the open loop unstable poles are also apparent in the closed loop if we want to fulfill the decoupling and steady state conditions. Similar limitations will always apply for any complementary sensitivity function.

Solution to Problem q:mvcontform   Multivariable controllability form

a) $w_1 = S(s)v, \qquad w_2 = \psi(s)v$

$$G_0(s) = \psi(s)S^{-1}(s)$$

b)

$$\psi(s) = \begin{bmatrix} s & 0 \\ 1 & 0 \\ 0 & s^2 \\ 0 & s \\ 0 & 1 \end{bmatrix}$$

$$S(s) = \begin{bmatrix} s^2 & 0 \\ 0 & s^3 \end{bmatrix}, \qquad S^{-1}(s) = \begin{bmatrix} \frac{1}{s^2} & 0 \\ 0 & \frac{1}{s^3} \end{bmatrix}$$

$$G_0(s) = \begin{bmatrix} \frac{1}{s} & 0 \\ \frac{1}{s^2} & 0 \\ 0 & \frac{1}{s} \\ 0 & \frac{1}{s^2} \\ 0 & \frac{1}{s^3} \end{bmatrix}$$

c)   One solution:

$$y = N(s)v$$
$$y = N_l \psi(s)v$$
$$N(s) = N_l \psi(s)$$

$$N(s) = \begin{bmatrix} s & 0 \\ -s & s^2 \end{bmatrix}, \qquad \psi(s) = \begin{bmatrix} s & 0 \\ 1 & 0 \\ 0 & s^2 \\ 0 & s \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} s & 0 \\ -s & s^2 \end{bmatrix} = \begin{bmatrix} n_{11} & n_{12} & n_{13} & n_{14} & n_{15} \\ n_{21} & n_{22} & n_{23} & n_{24} & n_{25} \end{bmatrix} \begin{bmatrix} s & 0 \\ 1 & 0 \\ 0 & s^2 \\ 0 & s \\ 0 & 1 \end{bmatrix}$$

with

$$N_l = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$N_l \psi(s) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s & 0 \\ 1 & 0 \\ 0 & s^2 \\ 0 & s \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} s & 0 \\ -s & s^2 \end{bmatrix}$$

d)   from $G_0(s)$:

$$\dot{y}_1 = u_1, \qquad \ddot{y}_2 = u_1$$
$$\dot{y}_3 = u_2, \qquad \ddot{y}_4 = u_2, \qquad \dddot{y}_5 = u_2$$

so

$$\dot{x}_2 = x_1$$
$$\dot{x}_4 = x_3, \qquad \dot{x}_5 = x_4$$

and

$$y_1 = x_1, \quad y_2 = x_2, \quad y_3 = x_3, \quad y_4 = x_4, \quad y_5 = x_5$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = I_{5 \times 5}$$

e)

$$A = A_0 - B_0 D_h^{-1} D_l$$
$$B = B_0 D_h^{-1}$$
$$C = N_l$$

Solution to Problem q:mvcontPP   Closed loop with MV controllability form

Matlab solution: `cs5_MCctrb.m`

c)
$$h_i = \psi(\lambda_i)p_i$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \\ m_{41} & m_{42} \\ m_{51} & m_{52} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is $A^L y$ (A^L is the left inverse of A).

Matlab: `p(:,ind)=pinv(psi)*h(:,ind);`

$$\begin{bmatrix} p_1^T \\ \vdots \\ p_5^T \end{bmatrix} = \begin{bmatrix} -0.0537 + 0.0985i & 0.0102 + 0.0191i \\ -0.0537 - 0.0985i & 0.0102 - 0.0191i \\ 0.0592 - 0.1399i & 0.0210 + 0.0280i \\ 0.0592 + 0.1399i & 0.0210 - 0.0280i \\ 0.0400 & 0.2173 \end{bmatrix}$$

Solution to Problem q:tgenKF   Turbogenerator LQG

The Matlab solution is given in `cs5_tgen_LQG.m` (`cs5_tgen_LQG_plot.m`)

a) R_e is the covariance matrix of the measurement noise. By increasing r_1 and r_2, one indicates that there is stronger measurement noise, and the result is an observer with a lower bandwidth. The filter gains will be smaller and observer poles and state estimation will be slower. This will have a negative effect on the performance of a state feedback controller. On the other hand, the effect of measurement noise on control action is suppressed more efficiently.

b) Suitable values are for example $r_1 = 1000$, $r_2 = 1000$.

c) Construct the required transfer functions (e.g., with `linmod`) and use the Matlab function `sigma` to obtain the singular value plots.

Note in particular:

The closed loop frequency response from r to y has a gain of 0dB at low frequencies, corresponding to high open loop gain.

The closed loop frequency response to noise is greater than 1 at low frequencies, but this is probably not a problem as noise is usually high frequency.

In general: At high frequencies the open loop with a break at p_1 , $L(s)$(equivalent to $G(s)K(s)$) is close to the complementary sensitivity function as $(I+L(s)) \approx I$; while the sensitivity function at low frequencies is close to $L^{-1}(s)$ as $(I + L(s)) \approx L(s)$.

Solution to Problem q:MVrobustness   Multivariable vs SISO robustness

a)   Calculation of $G(s)$:

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

$$= \begin{bmatrix} 1 & 10 \\ -10 & 1 \end{bmatrix} \begin{bmatrix} s & -10 \\ 10 & s \end{bmatrix}^{-1}$$

$$= \frac{1}{s^2 + 100} \begin{bmatrix} s - 100 & 10(s + 1) \\ -10(s + 1) & s - 100 \end{bmatrix}$$

(may be done by hand or with Matlab)

b)   with the inputs to $G(s)$ being w_1 and w_2 (in either closed loop or open loop),

$$y_1 = g_{11}w_1 + g_{12}w_2, y_2 = g_{21}w_1 + g_{22}w_2$$

With Channel 2 in closed loop

$$w_2 = -y_2$$
$$y_2 = \frac{g_{21}}{1 + g_{22}}w_1$$
$$z_1 = -y_1 = -g_{11}w_1 + g_{12}y_2$$
$$= -g_{11}w_1 + \frac{g_{12}g_{21}}{1 + g_{22}}w_1$$

after some algebra

$$z_1 = -\frac{1}{s}w_1$$

Similarly, with Channel 1 in closed loop $z_2 = -\frac{1}{s}w_2$

c)   The transfer functions with one loop closed are just integrators: Gain margin: ∞
Phase margin: 90°

Would therefore expect good robustness in the individual channels.

d)   The plant inputs are disturbed by errors ϵ_1, ϵ_2; with $10 = a$,

$$\tilde{u}_1 = (1 + \epsilon_1)u_1, \qquad \tilde{u}_2 = (1 + \epsilon_2)u_2$$

$$\tilde{B} = \begin{bmatrix} 1 + \epsilon_1 & 0 \\ 0 & 1 + \epsilon_2 \end{bmatrix}$$

$$\tilde{A}_{cl} = A - \tilde{B}C$$

$$= \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix} - \begin{bmatrix} 1 + \epsilon_1 & 0 \\ 0 & 1 + \epsilon_2 \end{bmatrix} \begin{bmatrix} 1 & a \\ -a & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -(1 + \epsilon_1) & -a\epsilon_1 \\ a\epsilon_2 & -(1 + \epsilon_2) \end{bmatrix}$$

e) Characteristic polynomial of $\tilde{A}(\epsilon)_{cl}$:

$$s^2 + (2 + \epsilon_1 + \epsilon_2)s + 1 + \epsilon_1 + \epsilon_2 + (a^2 + 1)\epsilon_1\epsilon_2$$

For stability both coefficients must be positive

With $\epsilon_2 = 0$: Stable with $\epsilon_1 > -1$

With $\epsilon_1 = 0$: Stable with $\epsilon_2 > -1$

f) With $\epsilon_2 = -\epsilon_1 = -\epsilon$, for stability need

$$1 - (a^2 + 1)\epsilon^2 > 0$$

$$\epsilon < \frac{1}{\sqrt{101}}$$

g) The interpretation of this is that robustness is very good with errors with 'direction' $\epsilon_1 = \epsilon_2$ or $\epsilon_1, \epsilon_2 = 0$ but is not good for the direction $\epsilon_1 = -\epsilon_2$.

The conclusion of this is: simple SISO open loop measures of robustness are not a good guide to MIMO robustness.

C.6   Chapter 7

Solution to Problem q:ztransform   Simple discrete time responses from z transforms

a)

$$G(z) = \frac{2z^2 - 6z}{2z^2 - 6z + 4}$$

$$G(z) = \frac{2 - 6z^{-1}}{2 - 6z^{-1} + 4z^{-2}}$$

$$G(z) = 1z^0 + 0z^{-1} - 2z^{-2} - 6z^{-3} - 14z^{-4} \dots$$

The response is $\{1,0,-2,-6,\dots\}$

b)

$$G(z) = \frac{1}{1 - 2z^{-1}}$$

with input

$$u(k) = 2e^{-k}\sigma(k)$$

$$U(z) = \frac{2z}{z - e^{-1}}$$

$$G(z)U(z) = \frac{2z}{(1 - 2z^{-1})(z - e^{-1})}$$

$$= \frac{2z^2}{(z-2)(z-e^{-1})} = 2 + \frac{4.736z - 1.472}{(z-2)(z-0.368)}$$

$$= 2 + \frac{a_1}{z-2} + \frac{a_2}{z - 0.368}$$

$$a_1 = 4.902, \qquad a_2 = -0.166$$

From $Z[\sigma(k)\alpha^k] = \frac{1}{1-\alpha z^{-1}}$,

$$Z[\sigma(k-1)\alpha^{k-1}] = \frac{1}{z - \alpha}$$

so

$$y(0) = 2$$

$$y(k) = 4.9 \cdot 2^{k-1} - 0.166 \cdot 0.368^{k-1}, \quad k \geq 1$$

Solution to Problem q:ztransformproofs   Discrete time final value theorem

$$x(\infty) = \lim_{z \to 1}(z-1)X(z)$$

Proof:

$$Z\big[x(k)\big] = \quad X(z) = \sum_{k=0}^{\infty} x(k)z^{-k}$$

$$Z\big[x(k-1)\big] = z^{-1}X(z) = \sum_{k=0}^{\infty} x(k-1)z^{-k}$$

$$\sum_{k=0}^{\infty} x(k)z^{-k} - \sum_{k=0}^{\infty} x(k-1)z^{-k} = X(z) - z^{-1}X(z)$$

$$\lim_{z \to 1}\left[\sum_{k=0}^{\infty} x(k)z^{-k} - \sum_{k=0}^{\infty} x(k-1)z^{-k}\right] = \lim_{z \to 1}\left[X(z) - z^{-1}X(z)\right]$$

with $x(k) = 0$ when $k < 0$, and when the system is stable the left hand side becomes

$$\lim_{z \to 1}\sum_{k=0}^{\infty}\big[x(k)z^{-k} - x(k-1)z^{-k}\big] = [x(0) - x(-1)] + [x(1) - x(0)] + [x(2) - x(1)] + \dots$$

$$= \lim_{k \to \infty} x(k)$$

so

$$\lim_{k \to \infty} x(k) = \lim_{z \to 1}\left[X(z) - z^{-1}X(z)\right] = \lim_{z \to 1}\left[(z-1)X(z)\right]$$

Solution to Problem q:ztransformlocus   Root loci for discrete systems

a)   i)   Lines of constant real part of continuous poles σ: $\sigma = -\frac{0.1}{T}$, $-\frac{0.5}{T}$ and $-\frac{1.0}{T}$.
The line in the z-plane is defined by $z = e^{sT} = e^{(\sigma+j\omega)T}$;
with $\sigma = -\frac{0.1}{T}$:

$$z = e^{\sigma T + j\omega T} = e^{\sigma T}e^{j\omega T}$$

$$= e^{-0.1}(\cos \omega T + j\sin \omega T)$$

as $\omega : 0 \to \frac{2\pi}{T}$ a circle is described, with radius $e^{-0.1}$. Larger magnitude σ correspond to smaller radiuses.

ii)   Lines of constant imaginary parts ω of continuous poles $\omega = 0.5\frac{\pi}{T}$, $1.0\frac{\pi}{T}$ and $1.5\frac{\pi}{T}$.
The line in the z-plane is $z = e^{sT}$. With $\omega = 0.5\frac{\pi}{T}$:

$$z = e^{\sigma T + j\omega T} = e^{\sigma T}e^{j\omega T}$$
$$= e^{\sigma T}(\cos 0.5\pi + j\sin 0.5\pi)$$

b) Second order system with damping factor ζ and natural frequency ω_n. Lines for $\zeta = 0, 0.5, 1$ as $\omega_n : 0 \to \frac{\pi}{T}$.

$$s = \zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

with $\zeta = 0$, circle radius is 1.0:

$$z = e^{\sigma T + j\omega T}$$
$$= (\cos\omega_n + j\sin\omega_n)$$

In general with $k = \sqrt{1 - \zeta^2}$

$$z = e^{\sigma T + j\omega T}$$
$$= e^{-\zeta\omega_n T}(\cos k\omega_n T + j\sin k\omega_n T)$$

when $\omega_n : 0 \to \frac{\pi}{T}$ starting radius is 1.0, final radius is $e^{-\pi\zeta}$.

c) For the second order system, lines for $\omega_n = \frac{\pi}{2T}, \frac{\pi}{T}$ as $\zeta : 0 \to 1$.

With $\omega_n = \frac{\pi}{2T}$

$$z = e^{\sigma T + j\omega T}$$
$$= e^{-\zeta\frac{\pi}{2}}\left(\cos k\frac{\pi}{2} + j\sin k\frac{\pi}{2}\right)$$

with $\zeta = 0$, angle $= 90^o$ and Radius$=1.0$
with $\zeta = 1$, angle $= 0^o$, radius $= e^{-\frac{\pi}{2}}$

With $\omega_n = \frac{\pi}{T}$:
with $\zeta = 0$, angle $= 180^o$ and Radius$=1.0$
with $\zeta = 1$, angle $= 0^o$, Radius $= e^{-\pi}$

These can easily be drawn using the Matlab command `rltool` with `grid`.

Solution to Problem q:DTcontrollerform   Discrete time controller form

z transform:

$$Y = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}U$$
$$= \frac{b_1 z^2 + b_2 z^1 + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}U$$

Use new variable $X_1(z)$, so that

$$(z^3 + a_1 z^2 + a_2 z + a_3)X_1 = U$$
$$Y = b_1 z^2 X_1 + b_2 z X_1 + b_3 X_1$$

with

$$X_2 = zX_1$$
$$X_3 = zX_2$$
$$zX_3 = z^3 X_1 = -a_1 X_3 - a_2 X_2 - a_3 X_1 + U$$
$$Y = b_1 X_3 + b_2 X_2 + b_3 X_1$$

The state space Z transform model is

$$zX = \Phi X + \Gamma U$$
$$Y = CX + DU$$

$$\Phi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}, \qquad\qquad \Gamma = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} b_3 & b_2 & b_1 \end{bmatrix} \qquad\qquad D = 0$$

or, in the discrete time domain

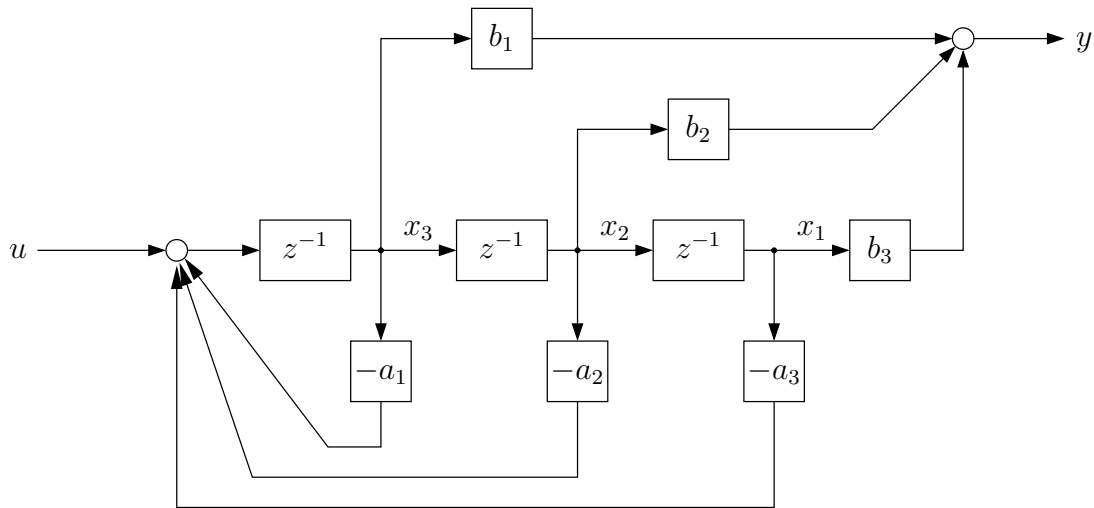$$x(k+1) = \Phi x(k) + \Gamma u(k), \qquad y(k) = Cx(k) + Du(k)$$

See Figure C.24

Figure C.24: discrete time controller form

Solution to Problem q:discreteP   Discrete time proportional controller

a) Root locus has a single pole at $-a$.

b) Continuous time state space model of the system is given as

$$\dot{x} = -ax + bu,$$
$$y = x$$

Let it's discrete time state space representation be given as

$$zx = \Phi x + \Gamma u, \qquad\qquad y = x$$

where,

$$\Phi = e^{-aT}, \qquad\qquad \Gamma = \int_0^T e^{-at}b\,dt$$
$$= [\frac{e^{-at}}{-a}]_0^T b$$
$$= \frac{(1 - e^{-aT})}{a}b$$

Then,

$$Y(z) = C(zI - \Phi)^{-1}\Gamma = \frac{\Gamma}{z - \Phi} = \frac{b}{a}\frac{1 - e^{-aT}}{z - e^{-aT}}$$

c) The discrete root locus has a pole at $z = e^{-aT} = 0.67$

$$G(z) = \frac{b}{a}\frac{1 - e^{-aT}}{z - e^{-aT}} = \frac{1}{2}\frac{1 - 0.67}{z - 0.67} = \frac{0.16}{z - 0.67}$$

so the characteristic equation in closed loop $1 + K(z)G(z) = 0$ is

$$1 + K(z)G(z) = 0$$
$$1 + K_{pd}\frac{0.16}{z - 0.67} = 0$$
$$z - 0.67 + 0.16K_{pd} = 0$$

The root locus is on the real axis, so the system becomes unstable when it hits the unit circle at $z = -1$: $K_{pd} = 10.44$.

d) The continuous proportional controller is always stable: the closed loop pole tends to $s = -\infty$. As shown in (c), this is not true for the discrete controller.

e) See Matlab/Simulink `cs6_discrete_P.m` and `cs6_discrete_P_mod.mdl`

Solution to Problem q:discretePD   Discrete time PD controller

a)   The continuous time state space model is given as

$$\dot{x} = Ax + bu,$$
$$y = x$$

where,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Let the discrete time state space representation be given as

$$zx = \Phi x + \Gamma u, \qquad\qquad\qquad y = x$$

where

$$\Phi = e^{AT}, \qquad\qquad\qquad \Gamma = \int_0^T e^{At} B dt$$

Since $A^2 = 0$, the matrices Φ and Γ can be calculated by using the definition of matrix exponential as following:

$$\Phi = I + AT + \frac{A^2 T^2}{2} + \ldots = I + AT = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

$$\Gamma = \int_0^T e^{At} B dt = \int_0^T (I + At) B dt = (T + A\frac{T^2}{2}) B$$

$$\Gamma = \left(\begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \frac{T^2}{2} \right) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

Then we obtain

$$G_2(z) = C(zI - \Phi)^{-1}\Gamma$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \left(z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

$$= \frac{T^2}{2} \frac{z+1}{(z-1)^2}$$

b)   Continuous RL (continuous controller):
     Two Poles at $(0, 0)$; RL is always on the imaginary axis.

     Discrete RL (discrete controller):
     Two poles at $z = 1$, zero at $z = -1$; RL is a circle with radius > 1 (see `cs6_discrete_PD.m`), thus, the system is always unstable.

     Note that RL rules are the same for continuous and discrete systems, but there would be a different interpretation for the poles.

c)

$$C_2(z) = K_{pd}\Big(1 + T_d(1 - z^{-1})\Big) = K_{pd}\frac{1}{z}\Big((1 + T_d)z - T_d\Big)$$

$$= K_{pd}\frac{1 + T_d}{z}\left(z - \frac{T_d}{1 + T_d}\right)$$

$$\alpha = \frac{T_d}{1 + T_d}, \qquad \left(\frac{1}{1 - \alpha} = 1 + T_d\right)$$

d)   See `cs6_discrete_PD.m`. To assist the design in `rltool` right-click over the root-locus, chose *Properties*, *Options* and select *Show grid*. One solution is `Kpd=12.4`, `alpha=0.83`.

e)   `Kpd=2.08`, `alpha=0.83` is one solution.
A solution as fast as that in (d) is not possible; design in the discrete time domain tells us explicitly what limitations arise as we increase the sampling time.

Solution to Problem q:Deadbeat   Deadbeat controller design

The command `acker` can be used to place the poles on the origin. If `acker` is not provided with the Matlab version, one can use also `place`, by positioning the poles close to the origin, e.g. `[0, 1e-5]` (`place` does not allow several pole on the same location"). The solution is in `cs6_deadbeat.m` and `cs6_deadbeat_sim.mdl`.

As expected, the steady state is reached after precisely two sampling periods.

Solution to Problem q:tgenD   Discrete turbogenerator

a)   This can be done in two steps:

step 1)   generate state space representation of complete controller. This can be done using `linmod()`

step 2)   use `c2d()` to get discrete time approximation.

b)   The simulation shows that:

–   for controller discretised with $T = 0.02$ the controlled response is almost identical.

–   for controller discretised with $T = 0.5$ the controlled response is notably different because of large sampling time.

c)   See `cs6_TG_discrete_K.m`.

d) The response of closed loop systems are notably different then the one achieved by approximating the continuous time controllers by Tustin approximation. This is because same weighting matrices Q, R, Q_e and R_e are used in both the cases. This is not the correct approach if one wants to design discrete time controllers to achieve same performance as achieved by continuous time controllers. It can be shown that to achieve same performance, one needs to modify the weighting matrices. This even requires modification in cost function J by including cross terms of the form $x(kT)Su(kT)$ in its discrete counterpart.

e) The response of continuous time model with time delay of 0.25 s resembles that of the discretised controller with sampling time 0.5 s. This is particularly clear in the first several samples after the step disturbance is applied. This shows that the effect of Tustin approximation with sampling time T is similar to the effect of adding a time delay $T/2$ to the original system.

Solution to Problem q:Dconvolution   Discrete convolution

Since,

$$G(z) = \frac{Y(z)}{U(z)}$$

Then,

$$Y(z) = G(z)U(z)$$

from definition of z-transform

$$\sum_{k=0}^{\infty} y(k)z^{-k} = \left(\sum_{k=0}^{\infty} g(k)z^{-k}\right)\left(\sum_{k=0}^{\infty} u(k)z^{-k}\right)$$

$$y(0) + y(1)z^{-1} + y(2)z^{-2} + \ldots = (g(0) + g(1)z^{-1} + g(2)z^{-2} + \ldots)(u(0)$$
$$+ u(1)z^{-1} + u(2)z^{-2} + \ldots)$$
$$= g(0)u(0) + (g(0)u(1) + g(1)u(0))z^{-1} +$$
$$(g(0)u(2) + g(1)u(1) + g(2)u(0))z^{-2} + \ldots$$

Comparing coefficients of $(z^0, z^{-1}, z^{-2}, \ldots)$ we get,

$$y(0) = g(0)u(0)$$
$$y(1) = g(0)u(1) + g(1)u(0)$$
$$y(2) = g(0)u(2) + g(1)u(1) + g(2)u(0)$$
$$\vdots$$
$$y(k) = \sum_{l=0}^{k} g(l)u(k-l)$$

Solution to Problem q:discrsystresponse   Impulse response

a) The impulse response can be easily computed by the so-called Markov parameters:

$$g(0) = D = 0$$
$$g(1) = C\Gamma = 1$$
$$g(2) = C\Phi\Gamma = 4.2$$
$$g(3) = C\Phi^2\Gamma = 0.84$$

b) The response to a particular input can be computed using the discrete convolution formula: $y(k) = \sum_{l=0}^{k} g(l)u(k-l)$

$$y(0) = g(0)u(0) = 0$$
$$y(1) = g(0)u(1) + g(1)u(0) = 0 + 5 = 5$$
$$y(2) = g(0)u(2) + g(1)u(1) + g(2)u(0) = 0 + 0 + 21 = 21$$
$$y(3) = g(0)u(3) + g(1)u(2) + g(2)u(1) + g(3)u(0) = 0 - 1 + 0 + 4.2 = 3.2$$

Solution to Problem q:DTzeros   Zeros of discrete time system

a) i)
$$G(s) = \frac{s+1}{s^2 + 4s + 1}$$
$$n = 2, \quad m = 1, \quad n - m - 1 = 0$$

For this system there is one continuous zero $s = -1$.

The discretisation results in only one zero, and it approaches e^{-T}.

  ii)
$$G(s) = \frac{1}{s^3 + s^2 + s}$$
$$n = 3, \quad m = 0, \quad n - m - 1 = 2$$

The discretisation of this system results in two zeros: they approach the zeros of the exact discretisation $1/s^3$, so we need the exact discretisation of $1/s^3$.

A state space representation of $1/s^3$ is(using that $\dddot{x} = u$)

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= u \\
y &= x_1
\end{aligned}
\qquad
A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad
B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$
$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Discretisation

$$A^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad A^3 = 0_{3\times 3}$$

$$\Phi = e^{AT} = I + AT + \frac{A^2 T^2}{2} + \ldots = I + AT + \frac{A^2 T^2}{2}$$

$$\Phi = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Psi = 1 + \frac{AT}{2} + \frac{A^2 T^2}{3!} = \begin{bmatrix} 1 & \frac{T}{2} & \frac{T^2}{6} \\ 0 & 1 & \frac{T}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Gamma = \Psi T B = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}$$

Let z be a zero of

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = Cx(k)$$

then

$$\begin{bmatrix} (zI - \Phi) & -\Gamma \\ Complex Numbers & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = 0$$

Since, $D = 0$, so we have $Cx = 0$

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0, \qquad \Rightarrow \qquad x_1 = 0, \qquad x = \begin{bmatrix} 0 \\ x_2 \\ x_3 \end{bmatrix}$$

First row: $u = 1$, $(zI - \Phi)x - \Gamma u = 0$

$$\begin{bmatrix} z-1 & -T & -T^2/2 \\ 0 & z-1 & -T \\ 0 & 0 & z-1 \end{bmatrix} \begin{bmatrix} 0 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} = 0$$

It then follows that

$$x_3 = \frac{T}{z-1} \qquad\qquad \text{(third row)}$$

$$x_2 = T^2 \left(-\frac{1}{6} - \frac{1}{2(z-1)} \right) \qquad\qquad \text{(first row)}$$

and finally, after substituting in the second row

$$z^2 + 4z + 1 = 0, \qquad z = -2 \pm \sqrt{3}$$

Therefore the zeros of the discretisation of $G(s)$ approach the values -3.73 and -0.268.

b) The requirement for a minimum phase system is $|z_i| < 1$. If T is increased, the zeros become closer and closer to the unit disk. When $T \geq 3.3$ the system becomes minimum phase. See `cs6_DTzeros.m`.

Solution to Problem q:DTwordlength   Finite word length

The poles of a system (controller or plant) approach $+1$, as $T \to 0$ (from $z = e^{sT}$).

As an example, the poles $z = 0.9996$, $z = 0.9998$ are not equal to the pole $z = 0.9999$, and the three have very different behaviours with $T = 0.0001$. With this sample time of $T = 0.0001$ and a word-length that allows precision up to the third decimal place the poles would have an identical representation. Finite word-length can be a problem at any sampling time, but the quicker the sampling time the more sensitive the implemented pole positions become to their finite word length representation. With a finite word-length the real implemented poles can lie a long distance from the required poles, making the performance of the closed loop different from that designed.

Finite word-length can lead to an additional problem in the digital systems due to quantization effects - during sampling the continuous time signals they are converted to a digital signal, that can take only a finite number of values, for example for N=8 bit word-length, the signal can take only values between 0 and 255. If the original signal is Y between, e.g., -5 and 5 V, then the quantized signal can take only values with a step of $q = (Y_{max} - Y_{min})/(2^N - 1) = 10/255 = 0.0392$ V/bit. Big quantization steps can lead to loss of performance and even instability. The effect of quantization can be simulated in Simulink using the `Quantizer` block.

Solution to Problem q:samplingtime   Sampling times

a) The time constant $\tau = 5$ sec, the bandwidth is $\omega_b = 0.2$ rad/s and the gain is 1.

b) Attenuation of 3 dB occurs at the system's bandwidth, therefore $\omega_3 = \omega_b = 0.2$ rad/s.

c) To avoid aliasing effect the sampling frequency should be $\omega_s > 2\omega_b$. If we chose $\omega_s = 10\omega_b = 2$ rad/s, then the sampling time is $T_s = 2\pi/\omega_s = 3.14$ sec. Therefore sampling times smaller than 3.14 sec. are suitable.

Solution to Problem q:SandHold   **Frequency response of** *zoh* **unit**

a) $y(t)$ as a function of $u(t)$ at the sample times $u(kT)$:

Use the pulse function $\sigma(t - nT) - \sigma\left(t - (n + 1)T\right)$.

The functions

$$\sigma(t) - \sigma(t - T)$$
$$\sigma(t - T) - \sigma(t - 2T)$$
$$\sigma(t - 2T) - \sigma(t - 3T)$$

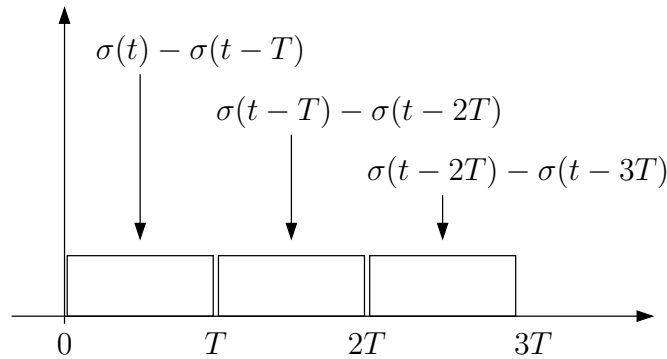are shown below on Figure C.25. The $y(t)$ is shown on Figure C.26.

Figure C.25: Series of pulses of width T superimposed

Mathematically $y(t)$ is

$$
\begin{aligned}
y(t) = \ & u(0)\Big[\sigma(t) - \sigma(t - T)\Big] \\
& + u(T)\Big[\sigma(t - T) - \sigma(t - 2T)\Big] \\
& + u(2T)\Big[\sigma(t - 2T) - \sigma(t - 3T)\Big] + \dots \\
= \ & \sum_{k=0}^{\infty} u(kT)\Big[\sigma(t - kT) - \sigma\left(t - (k + 1)T\right)\Big]
\end{aligned}
$$

b)   Laplace Transformation of $y(t)$: $u(kT)$ are constant multipliers.

The Laplace transform of $\sigma(t - kT) - \sigma\left(t - (k + 1)T\right)$ is

$$L\Big(\sigma(t - kT)\Big) = \frac{e^{-kTs}}{s}$$

$$L\Big(\sigma\left(t - (k + 1)T\right)\Big) = \frac{e^{-(k+1)Ts}}{s}$$

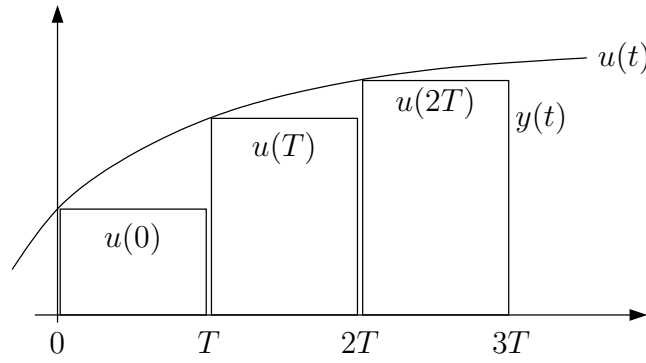$$L\Big(\sigma(t - kT) - \sigma\left(t - (k + 1)T\right)\Big) = \frac{e^{-kTs} - e^{-(k+1)Ts}}{s}$$

Figure C.26: Input and output of sampler

As $u(kT)$ are just constant multipliers the following holds:

$$Y(s) = L\big(y(t)\big)$$
$$= \sum_{k=0}^{\infty} u(kT)\frac{e^{-kTs} - e^{-(k+1)Ts}}{s}$$
$$= \frac{1 - e^{-Ts}}{s} \sum_{k=0}^{\infty} u(kT)e^{-kTs}$$

c) $u^*(t)$ is a sequence of δ pulses

$$L\big(u^*(T)\big) = u(T)e^{-Ts} \qquad\qquad \text{(1 Pulse)}$$
$$U^*(s) = L\big(u^*(t)\big)$$
$$= \sum_{k=0}^{\infty} u(kT)e^{-kTs} \qquad\qquad \text{(all Pulses)}$$

So it follows that

$$Y(s) = \frac{1 - e^{-Ts}}{s}U^*(s)$$

That is, the transfer function of the sampler G_{zoh} is

$$G_{zoh} = \frac{Y(s)}{U^*(s)} = \frac{1 - e^{-Ts}}{s}$$

d)   The frequency response of G_{zoh} is

$$G_{zoh}(j\omega) = \frac{1 - e^{-Tj\omega}}{j\omega}$$
$$= \frac{T}{\omega T/2}\left(\frac{e^{\frac{T}{2}j\omega} - e^{-\frac{T}{2}j\omega}}{2j}\right)e^{-0.5Tj\omega}$$
$$= T\frac{\sin(\omega T/2)}{\omega T/2}e^{-0.5Tj\omega}$$

Bode Diagram:

$$G_{zoh}(j\omega) = E(j\omega)F(j\omega), \qquad E(j\omega) = T\frac{\sin(\omega T/2)}{\omega T/2}, \quad F(j\omega) = e^{-0.5Tj\omega}$$

$$|E(0)| = T, \qquad \left|E\left(\frac{2n\pi}{T}\right)\right| = 0, \qquad |F(j\omega)| = 1 \;\; \forall \omega$$

$$\angle E(j\omega) = 0, \qquad\qquad 0 < \omega < 2\pi/T, \qquad 4\pi/T < \omega < 6\pi/T, \;\dots$$
$$\angle E(j\omega) = 180°(= -180°), \qquad 2\pi/T < \omega < 4\pi/T, \qquad 6\pi/T < \omega < 8\pi/T, \;\dots$$
$$\angle F(j\omega) = -0.5T\omega$$

Every $\frac{2n\pi}{T}$ the sign of $E(j\omega)$ changes. $G(j\omega)$ experiences at these values a further phase shift of $\pm 180°$.

Matlab: `cs6_S_and_H.m`

Solution to Problem q:DTfresp   Frequency response of discrete system with sinusoidal input

a)   Calculation of $u^*(t)$ when $u(t) = \sin \omega t$:

$$u^*(t) = \sin(\omega t) \sum_{k=-\infty}^{\infty} \delta(t - kT)$$

using the hint

$$u^*(t) = \sin(\omega t)\frac{1}{T}(1 + 2\cos \omega_s t + 2\cos 2\omega_s t + \dots)$$

b)   The Delta sampler is **not** a linear system! It is actually a modulator.

We can calculate the output of the modulator using a Fourier transform (not a transfer function).

The zero order hold is linear, so it has a transfer function, and its frequency response can be used to describe its behaviour.

Interpretation

Using the trigonometric formulae

$$2\sin \alpha \cos \beta = \sin(\alpha + \beta) - \sin(\beta - \alpha)$$

$$\sin \bar\alpha + \sin \bar\beta = 2\sin\frac{\bar\alpha + \bar\beta}{2}\cos\frac{\bar\beta - \bar\alpha}{2}$$

With the sample time $T = 0.05s$

$$\omega_s = 2\pi/T = 125.6 \text{ rad/s}$$

With an input frequency $\omega = 2$ rad/s

$$u^*(t) = 20\sin(2t)(1 + 2\cos 125.6t + 2\cos 251.2t + \ldots), \qquad (\alpha = 2, \ \beta = 125.6)$$
$$u^*(t) = 20\sin(2t) + 20\sin(125.6 + 2)t - 20\sin(125.6 - 2)t + \ldots$$

$$y(t) = G(j2)G_{zoh}(j2)u^*(t)$$
$$y(t) \approx 20G(j2)G_{zoh}(j2)\sin(2t)$$

After the zero order hold zoh and $G(s)$ (which act as filters) only the first elements are large.

With the input $\omega = 60 rad/s$

$$u^*(t) = 20\sin(60t)(1 + 2\cos 125.6t + 2\cos 251.2t + \ldots)$$
$$= 20\sin(60t) - 20\sin(125.6 - 60)t + 20\sin(125.6 + 60)t + \ldots$$

$$\bar{\alpha} = 60, \qquad \bar{\beta} = 65.6, \qquad \frac{\bar{\alpha} + \bar{\beta}}{2} = 62.8, \qquad \frac{\bar{\beta} - \bar{\alpha}}{2} = 2.8$$

$$y(t) \approx 20G(j60)G_{zoh}(j60)\sin 62.8t \cos 2.8t$$

The first two frequencies are similar: similar amplification by $G_{zoh}(s)G(s)$. Because of this we get the phenomena of 'beating'- a low frequency variation of amplitude.

Solution to Problem q:gyrofilter   Filter for gyroscope sensor.

a) Setting the low pass filter's cut-off frequency to a larger value will result in noisy measurement which in turn will introduce noise in the states. Depending the controller the system might be also unstable as the controller is a gain that amplifies the noise.

b) The noise is the difference and an observer is a possible solution to output the filtered states by combining the state estimates from the model and the measured states from the sensors.

Solution to Problem q:minsegDisc   Discrete time and continuous time controllers

a) Check the MATLAB file `lqrd_comparison.m`.

  i) When using lqrd command no re-tuning is needed since the command takes the continuous time 'A','B','Q' and 'R' matrices which are already tuned (cf. Problem)5.10) in continuous time design and generates the discretised controller.

Thus the discrete time response to initial conditions or position tracking will be identical to the continuous time one. On the other hand the input arguments of the dlqr command are the discretised 'A', 'B', 'Q' and 'R' where tuning is required to achieve the desired performance.

ii)   lqrd discretises the 'A','B','Q' and 'R' matrices in the controller case, however in the observer case the input arguments are the A^T and C^T. Thus in this case the lqrd will dicretise the C^T matrix which has no meaning and therefore it can be concluded that dualism fails in this specific problem. As a result observer gains could not be designed using lqrd.

Solution to Problem q:minsegDiscsim   Mini Segway: Simulation discrete time observed-based state feedback controller

a)   Discretising the system using zero order hold:

```
sys = ss(A, B, C, D);
sys_d = c2d(sys, Ts);
[Ad, Bd, Cd, Dd] = ssdata(sys_d );
```
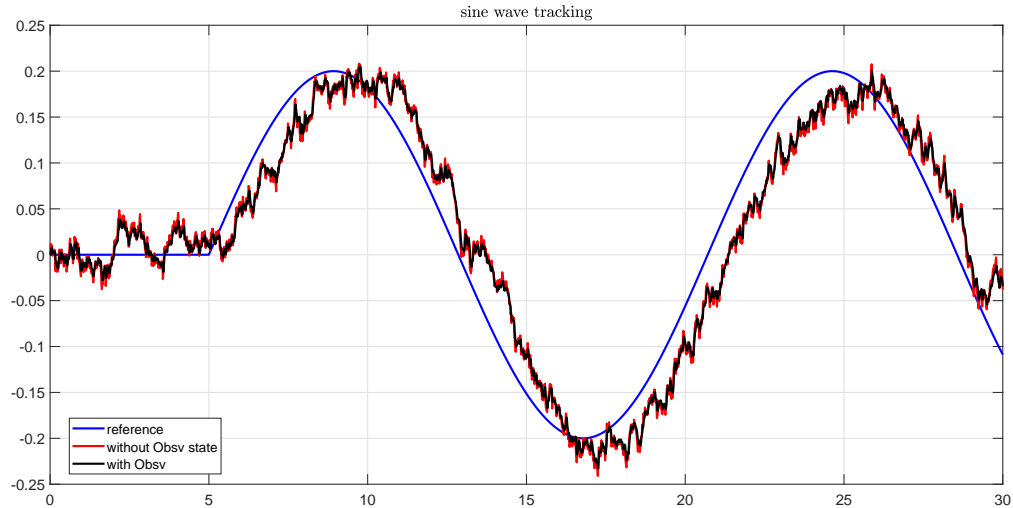

Figure C.27: Sine tracking with and without Observer

As shown in figure C.27 the observer filters the position and angular states such that the noise content in the position tracking is reduced.

Solution to Problem q:minsegDiscexp   Mini Segway: Experiment discrete time observed-based state feedback controller

a) As shown in figure C.28 is the tracking of a sine reference input with frequency
    0.4 rad/s and amplitude 0.2 m. It is shown that after fine tuning of the controller
    and the observer a good experimental performance could be achieved. From the red
    and black curve it can be noticed that the observer provides better performance at
    the peak of the sinewave where the change in direction of the Minseg Robot is much
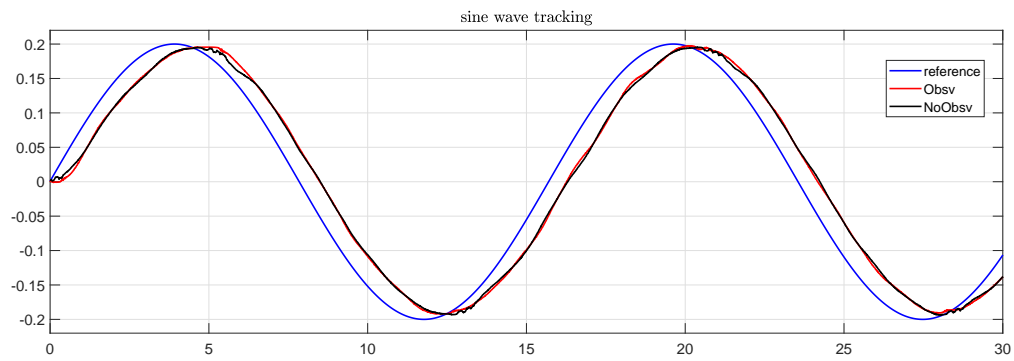   smoother.

Figure C.28: Experiment LQG

C.7   Chapter 8

Solution to Problem q:LSident   Proof of least squares optimality

We calculate the p which makes the first derivative of $V(p)$ zero, to find the p which minimizes $V(p)$. We have

$$
\begin{aligned}
V(p) &= (Y - Mp)^T (Y - Mp) \\
&= Y^T Y - p^T M^T Y - Y^T Mp + p^T M^T Mp
\end{aligned}
$$

So

$$
\begin{aligned}
\frac{dV(p)}{dp} &= -M^T Y - M^T Y + 2M^T Mp = 0 \\
0 &= -M^T Y + M^T Mp \\
p &= (M^T M)^{-1} M^T Y
\end{aligned}
$$

Another way of finding the p which minimizes $V(p)$ is by using the approach of completion of squares. Let

$$
\begin{aligned}
V(p) &= (Y - Mp)^T (Y - Mp) \\
&= Y^T Y - p^T M^T Y - Y^T Mp + p^T M^T Mp
\end{aligned}
$$

or

$$
V(p) - Y^T Y = p^T M^T Mp - p^T M^T Y - Y^T Mp
$$

Adding a constant term $Y^T M (M^T M)^{-1} M^T Y$ to both side will yield

$$
\begin{aligned}
V(p) - Y^T Y + Y^T M (M^T M)^{-1} M^T Y &= p^T M^T Mp - p^T M^T Y - Y^T Mp \\
&\quad + Y^T M (M^T M)^{-1} M^T Y \\
&= (p - (M^T M)^{-1} M^T Y)^T M^T M (p - (M^T M)^{-1} M^T Y)
\end{aligned}
$$

Thus

$$
V(p) = (p - (M^T M)^{-1} M^T Y)^T M^T M (p - (M^T M)^{-1} M^T Y) + Y^T Y - Y^T M (M^T M)^{-1} M^T Y
$$

Which shows that $V(p)$ is minimum if the first term on right hand side is minimum or equal to zero. Then

$$
p - (M^T M)^{-1} M^T Y = 0
$$

or

$$
p = (M^T M)^{-1} M^T Y
$$

Solution to Problem q:PEcalc   Persistent excitation of step functions

a)

$$(z - 1)u(kT) = u(kT + T) - u(kT)$$

For the step function $u(t + T) - u(t)$ is only 1 at $k = -1$.

b) If, for **all** polynomials $a(z)$ of order n $(a_n z^n + \ldots)$:

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} (a(z)u(l))^2 > 0$$

the PE order is $n + 1$.

This means that if **any** polynomial $a(z)$ can be found such that

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} (a(z)u(l))^2 = 0$$

then the PE order must be $\leq n$.

c) With the polynomial $a(z) = z - 1$ (order $n = 1$) and $u(l)$ a step function, then from a).

$$a(z)u(l) = 0, \qquad l = 0, 1, \ldots \infty$$

so

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} (a(z)u(l))^2 = 0$$

so the PE order is either 1 or 0.

d) Next, we will find the exact order by analyzing the auto correlation $C_{uu}(1) = c_{uu}(0)$. Since, it is a scaler hence it has rank of 1 or PE order is 1.

Solution to Problem q:PEsin   Persistent excitation of sinusoid

a)

$$(z^2 - 2z \cos \omega T + 1)u(kT) = \sin(\omega kT + 2\omega T) - 2 \cos \omega T \sin(\omega kT + \omega T) + \sin(\omega kT)$$

Since,

$$\sin A + \sin B = 2 \sin \frac{A + B}{2} \cos \frac{A - B}{2}$$

$$\sin(\omega kT + 2\omega T) + \sin(\omega kT) = 2 \sin(\omega kT + \omega T) \cos \omega T$$
$$(z^2 - 2z \cos \omega T + 1)u(kT) = 0$$

b)   If we can find any polynomial $a(z)$ of order n, such that

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} (a(z)u(l))^2 = 0$$

Then, the PE order must be $\leq n$

With $a(z) = (z^2 - 2z \cos \omega T + 1)$, i.e. $n = 2$. As shown in a),

$$a(z)u(l) = 0, \qquad \forall l = 0, 1, \ldots \infty$$

Then,

$$\lim_{k \to \infty} \frac{1}{k} \sum_{l=0}^{k} (a(z)u(l))^2 = 0$$

So the PE order must be ≤ 2.

c)   For the input signal $u(kT)$

$$R_u(\tau) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N} u(kT)u(kT \pm \tau)$$

At times 0 and T

$$R_u(0) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N} u(kT)u(kT)$$

$$R_u(T) = \lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N} u(kT)u\big((k+1)T\big)$$

These are the elements $(1,1)$ and $(1,2)$ of $C_{uu}(2)$, so

$$C_{uu}(2) = \begin{bmatrix} R_u(0) & R_u(T) \\ R_u(T) & R_u(0) \end{bmatrix}$$

$$C_{uu}(2) = \begin{bmatrix} \frac{1}{2}\cos 0 & \frac{1}{2}\cos \omega T \\ \frac{1}{2}\cos \omega T & \frac{1}{2}\cos 0 \end{bmatrix}$$

$$C_{uu}(2) = \frac{1}{2} \begin{bmatrix} 1 & \cos \omega T \\ \cos \omega T & 1 \end{bmatrix}$$

d)   At $T = \frac{2\pi}{\omega}$

$$C_{uu}(2) = \begin{bmatrix} \frac{1}{2}\cos 0 & \frac{1}{2}\cos \omega T \\ \frac{1}{2}\cos \omega T & \frac{1}{2}\cos 0 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\text{rank}C_{uu}(2) = 1$$

So PE order is 1. All the samples are at the same position in the sine wave so it looks like a step.

At $\omega \neq \frac{2\pi}{T}$, the samples are at different positions in the sine wave, so it has more information: PE order $= 2$.

Solution to Problem q:PEwn   Persistent excitation of white noise

For white noise,

$$C_{uu}(1) = c_{uu}(0) = \lim_{k \to \infty} \frac{1}{k} \sum_{i=0}^{k} u_i^2 = S_0$$

$$C_{uu}(2) = \begin{bmatrix} c_{uu}(0) & c_{uu}(1) \\ c_{uu}(1) & c_{uu}(0) \end{bmatrix}$$

Using, the property of white noise that $c_{uu}(i) = 0, \forall i = 1, 2 \dots,$

$$C_{uu}(2) = \begin{bmatrix} c_{uu}(0) & 0 \\ 0 & c_{uu}(0) \end{bmatrix}$$

$$= S_0 I_2$$

$$\vdots$$

$$C_{uu}(n) = S_0 I_n$$

so $C_{uu}(n)$ has rank n or PE condition is satisfied for all n.

Solution to Problem q:identex   Least Squares identification

a)   Let us have N samples and let $n = 2$ then,

$$y_0 = -a_1 y_{-1} - a_2 y_{-2} + b_1 u_{-1} + b_2 u_{-2} + e_0$$
$$y_1 = -a_1 y_0 - a_2 y_{-1} + b_1 u_0 + b_2 u_{-1} + e_1$$
$$y_2 = -a_1 y_1 - a_2 y_0 + b_1 u_1 + b_2 u_0 + e_2$$
$$y_3 = -a_1 y_2 - a_2 y_1 + b_1 u_2 + b_2 u_1 + e_3$$
$$\vdots$$
$$y_{N-1} = -a_1 y_{N-2} - a_2 y_{N-3} + b_1 u_{N-2} + b_2 u_{N-3} + e_{N-1}$$

or

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} -y_{-1} & -y_{-2} & u_{-1} & u_{-2} \\ -y_0 & -y_{-1} & u_0 & u_{-1} \\ \vdots & \vdots & \vdots & \vdots \\ -y_{N-2} & -y_{N-3} & u_{N-2} & u_{N-3} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} e_2 \\ e_3 \\ \vdots \\ e_{N-1} \end{bmatrix}$$

However, the values of input sequence $u(-1)$, $u(-2)$, $\dots u(-n)$, and $y(-1)$, $y(-2)$, $\dots y(-n)$ is not available in measurement data. This means that first n-rows will be

0, hence to make $M^T M$ full rank these rows should be eliminated. This results in,

$$
M = \begin{bmatrix} -y_1 & -y_0 & u_1 & u_0 \\ -y_2 & -y_1 & u_2 & u_1 \\ \vdots & \vdots & \vdots & \vdots \\ -y_{N-2} & -y_{N-3} & u_{N-2} & u_{N-3} \end{bmatrix}
$$

which has the dimensions of $N - n \times 2n$:

b) See `cs7_LSrankM.m`.

For the sinusoid: rank $= 4$ (singular values confirm this).

For the white noise: rank $=$ arbitrary

As the sequence becomes longer, the matrix $M^T M$ approaches a scaled version of the empirical covariance matrix; thus the rank of $M^T M$ for a long sequence can be expected to have the same rank as the PE order.

c) See Matlab solution in `cs7_LSparest.m`. 3rd and 4th order models generated are identical. A pole and zero cancel in the 4th order model.

d) Exact validation achieved with these models: the model order is clearly 3.

e) Inconsistent results when attempt to generate models from sinusoidal or step input. A true inverse is only possible when rank $M^T M = 2n$: with a PE order of 2 it is only possible to accurately estimate a system of order 1 (which has 2 parameters).

Solution to Problem q:subspaceID   Subspace identification

a) Clear cut-off, 9 non-zero singular values in Hankel matrix. Reasonable model achieved with model order 4. The model can be estimated using `cs7_parest.m`

b) The cut-off is not so clear for the noisy signal. Since after the 4th singular value of `Hn` the others are relative small one can chose 4th or 5th order model. Because the difference between the 3rd and the 4th singular values is also large one can try also identifying a 3rd order model of the system.

Solution to Problem q:IDtoolbox   Subspace identification using ident GUI

1. Open the toolbox with the command `ident`.

2. Import the data: *Import data → Data object → Object*: `iodata1`. Repeat for the second data set.

3. The signals then appear in the left panel *Data Views*.

4. Drag and drop the first signal set to *Working model*. Remove the means from all signals using the *Preprocess* drop-down list → *Remove means*. Repeat for the second signal set. One of the new set of signals should be used as *Working data* and the other one as *Validation data*.

5. Estimate models of 2nd, 3rd, 4th and 5th order using N4SID (subspace identification). For the purpose choose *Linear parametric models* from the *Estimate* drop-down box. Select *State-space* as *Structure* and repeat the identification for the different orders.

6. Validate the identified models using the second data set. Use the *Model Views* check-boxes in the lower-right corner of the GUI.

The model used to create the data was 4th order with noise. The identified models of order 2-4 all very accurately reproduce the original data.

C.8   Chapter 9

Solution to Problem q:reduction   Reduction by truncation and residualization

a) The functions are in the files `balresid.m`, `baltruncation.m`

b) The graphs are generated in the script `cs8_HVDC_reduce.m`. You can compare your results to the results obtained using the Matlab function `balred`.