Additional Lecture Notes for the course SC4090

Discrete-time systems analysis

Ton van den Boom

December 3, 2014

Contents

Chapter 1

Introduction

1.1 Discrete-time versus continuous-time

Engineers and Physical scientists have for many years utilized the concept of a *system* to facilitate the study of the interaction between forces and matter. A system is a mathematical abstraction that is devised to serve as a model for a dynamic phenomenon. It represents the dynamic phenomenon in terms of mathematical relations among three sets of variables known as the *input*, the *output*, and the *state*.

The input represents, in the form of a set of time functions or sequences, the external forces that are acting upon the dynamic phenomenon. In similar form, the output represents the measures of the directly observable behavior of the phenomenon. Input and output bear a cause-effect relation to each other; however, depending on the nature of the phenomenon, this relation may be strong or weak.

A basic characteristic of any dynamic phenomenon is that the behavior at any time is traceable not only to the presently applied forces but also to those applied in the past. We may say that a dynamic phenomenon possesses a "memory" in which the effect of past applied forces is stored. In formulating a system model, the state of the system represents, as a vector function of time, the instantaneous content of the "cells" of this memory. Knowledge of the state at any time t, plus knowledge of the forces subsequently applied is sufficient to determine the output (and state) at any time $t \geq t_0$.

As an example, a set of moving particles can be represented by a system in which the state describes the instantaneous position and momentum of each particle. Knowledge of position and momentum, together with knowledge of the external forces acting on the particles (i.e., the system input) is sufficient to determine the position and momentum at any future time.

A system is, of course, not limited to modeling only physical dynamic phenomena; the concept is equally applicable to abstract dynamic phenomena such as those encountered in economics or other social sciences.

If the time space is continuous, the system is known as a *continuous-time* system. However, if the input and state vectors are defined only for discrete instants of time k, where k ranges over the integers, the time space is discrete and the system is referred to as a *discrete-time system.* We shall denote a continuous-time function at time t by $f(t)$. Similarly, a discrete-time function at time k shall be denoted by $f(k)$. We shall make no distinction between scalar and vector functions. This will usually become clear from the context. Where no ambiguity can arise, a function may be represented simply by f.

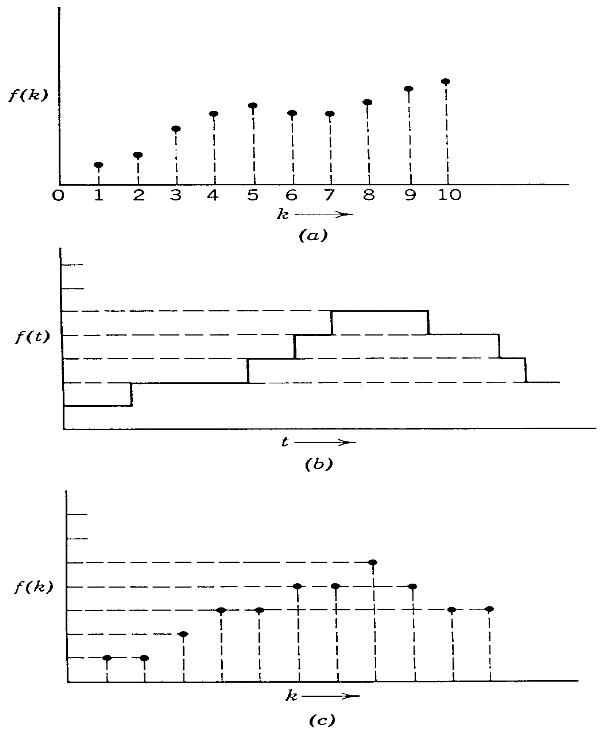

Figure 1.1: Illustration of discrete-time functions and quantized functions. (a) discrete-time function, (b) quantized function, (c) quantized, discrete-time function

It is important to distinguish between functions whose argument is discrete (i.e., functions of a discrete variable) and those that in themselves vary over a discrete set. Functions of the latter type will be referred to as *quantized functions* and the systems in which they appear will be called *quantized-data systems.* Thus the function illustrated in Fig. 1.1.a is a discrete-time function and the one shown in Fig.1.1.b is a quantized function. Note that in Fig. 1.1 $f(k)$ can range only over discrete values (which need not necessarily be uniformly spaced). A discrete-time function may, of course, be quantized as well; this is shown in Fig. 1.1.c.
In certain continuous-time systems, some state variables are allowed to change only at discrete instants of time t_k, where k ranges over the integers and where the spacing between successive instants may be arbitrary or uniform. Such systems are in effect a kind of hybrid between a continuous- time and a discrete-time system. They are encountered whenever

a continuous-time function is sampled at discrete instants of time. Although they are frequently analyzed most easily by treating them as discrete-time systems, they differ from discrete-time systems in that special consideration may have to be given to the instants of time at which the sampling occurs. They have been given a special name and are known as *sampled-data systems*.
We shall be concerned here primarily with the analysis of discrete-time systems. Our interest in these systems is motivated by a desire to predict the performance of the physical devices for which this kind of system is an appropriate model. This is, however, not the only motive for studying discrete-time systems. A second, nearly as important reason is that there are many continuous-time systems that are more easily analyzed when a discrete-time model is fitted to them. A common example of this is the simulation of a continuous-time system by a digital computer. Further, a considerable body of mathematical theory has been developed for the analysis of discrete-time systems. Much of this is valuable for gaining insight into the theory of continuous-time systems as well as discrete-time systems. We note that a continuous-time function can always be viewed as the limit of a time sequence whose spacing between successive terms is approaching zero.

1.2 Sampling and Interpolation

Many physical signals, such as electrical voltages produced by a sound or image recording instrument or a measuring device, are essentially continuous-time signals. Computers and related devices operate on a discrete-time axis. Continuous-time signals that are to be processed by such devices therefore first need be converted to discrete-time signals. One way of doing this is sampling.

Definition 1 *Sampling.*
Let $T_{con} = \mathbb{R}$ be the continuous-time axis, and let $T_{dis} = \mathbb{Z}$ be the discrete-time axis. Let $x(t)$ be a continuous-time signal with $t \in T_{con}$. Then, sampling the continuous-time signal $x(t)$, $t \in T_{con}$ on the discrete-time axis T_{dis} with sampling period T, results in the discrete-time signal $x^*(k)$ defined by

$$x^*(k) = x(k\,T) \qquad \textit{for all} \qquad k \in T_{dis} \tag{1.1}$$

A device that performs the sampling operation is called a sampler.

Example 2 (Sampled real harmonic) *Let the continuous-time signal x, given by*

$$x(t) = 1/2[1 - \cos(2\pi t)], \quad t \in \mathbb{R}$$

be sampled on the uniformly spaced discrete-time axis $\mathbb{Z}$. This results in the sampled signal x^* given by

$$x^*(k) = 1/2[1 - \cos(2\pi kT)], \quad k \in \mathbb{Z}$$

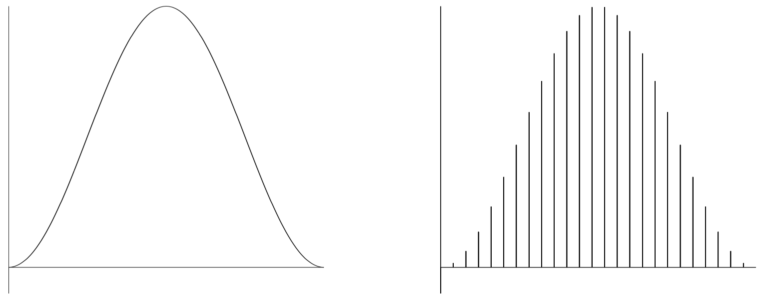

Figure 1.2: Sampling. Left: a continuous-time signal. Right: its sampled version.

The converse problem of sampling presents itself when a discrete-time device, such as a computer, produces signals that need drive a physical instrument requiring a continuous-time signal as input. Suppose that a discrete-time signal x^* is defined on the discrete-time axis T_{dis} and that we wish to construct from x^* a continuous-time signal x defined on the continuous-time axis $T_{con} \supset T_{dis}$. There obviously are many ways to do this. We introduce a particular class of conversions from discrete-time to continuous-time signals, for which we reserve the term interpolation. This type of conversion has the property that the continuous-time signal x agrees with the discrete-time signal x^* at the sampling times.

Definition 3 *Interpolation.*
Let $T_{con} = \mathbb{R}$ be the continuous-time axis, and let $T_{dis} = \mathbb{Z}$ be the discrete-time axis. Let $x^*(k)$ be a discrete-time signal with $k \in T_{dis}$ with sampling period T. Then, any continuous-time signal $x(t)$, $t \in \mathbb{R}$ is called an interpolation of x^* on T_{con}, if

$$x(kT) = x^*(k) \qquad \text{for all} \qquad k \in T_{dis} \tag{1.2}$$

Another way of saying that x is an interpolation of x^* is the statement that sampling the continuous-time signal x generated by interpolating the discrete-time signal x^* on T_{dis}, reproduces the discrete-time signal x^*. Clearly, there is no unique interpolation for a given discrete-time signal x^*. Suppose that x^* is defined on the uniformly sampled discrete-time axis $\mathbb{Z}$. Then, one possible interpolation method is step interpolation as illustrated in Fig. 1.3(a).
Interpolation is done using an interpolating function. An interpolating function is any function i: $\mathbb{R} \to \mathbb{R}$ such that

$$i(t) = \begin{cases} 1 & \text{for } t = 0 \\ 0 & \text{for } t = k, \quad \text{where} \quad k \neq 0, \ k \in \mathbb{Z} \qquad t \in \mathbb{R} \\ \text{arbitrary} & \text{elsewehere} \end{cases}$$

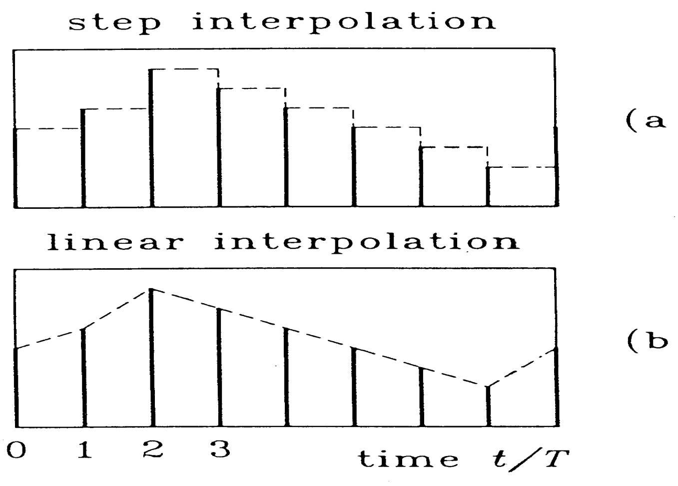

Figure 1.3: Interpolation. Top: step interpolation. Bottom: linear interpolation.

If $x^*(k)$, $k \in \mathbb{Z}$ is a discrete-time signal defined on the time axis T_{dis}, and i an interpolating function, the continuous-time signal x given by

$$x(t) = \sum_{n \in \mathbb{Z}} x^*(n)\, i(t/T - n), \qquad t \in \mathbb{R}$$

is an interpolation of x^*. The reason is that by setting $t = k\,T$, with $k \in \mathbb{Z}$, it follows that

$$x(k\,T) = \sum_{n \in \mathbb{Z}} x^*(n)\, i(kT/T - n) = x^*(k), \qquad k \in \mathbb{Z}$$

Step interpolation is achieved with the step interpolating function

$$i_{step}(t) = \begin{cases} 1 & \text{for } 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \qquad t \in \mathbb{R}$$

while linear interpolation is obtained with the linear interpolating function

$$i_{lin}(t) = \begin{cases} 1 - |t| & \text{for } |t| < 1 \\ 0 & \text{otherwise} \end{cases} \qquad t \in \mathbb{R}$$

Another interpolation function is the sinc interpolating function

$$i_{sinc}(t) = \mathrm{sinc}(\pi\, t)$$

Here, sinc: $\mathbb{R} \to \mathbb{R}$ is the function defined by

$$\mathrm{sinc}(t) = \begin{cases} \dfrac{sin(t)}{t} & \text{for } t \neq 0 \\ 1 & \text{for } t = 0 \end{cases} \qquad t \in \mathbb{R}$$

Graphs of these interpolating functions are given in Fig. 1.4.

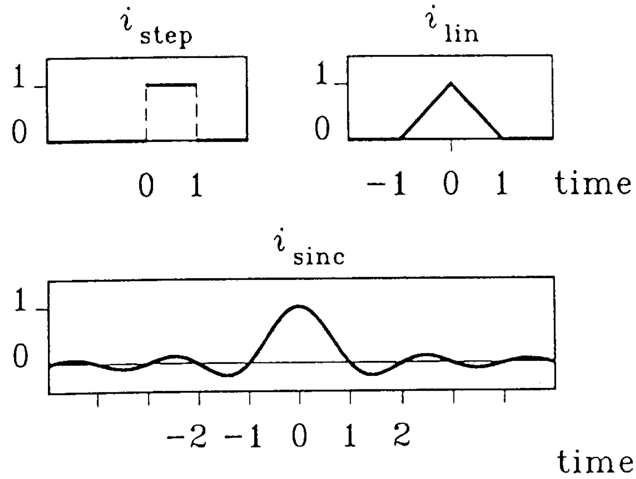

Figure 1.4: Interpolating functions. Top left: step interpolating function. Top right: linear interpolating function. Bottom: sinc interpolating function.

Example 4 (Interpolation) *Let a sampled signal x^* with sampling period T be given by*

$$x^*(k) = \begin{cases} \cos\left(\frac{\pi\, k}{8}\right) & \textit{for } -4 \leq k < 4 \\ 0 & \textit{otherwise} \end{cases} \qquad k \in \mathbb{Z}$$

A plot of x^* is given in Fig. 1.5(a). Interpolation with the step interpolating function results in the staircase-like continuous-time signal of Fig. 1.5(b). Interpolation with the linear interpolating function leads to the signal of Fig. 1.5(c), which is obtained by connecting the sampled values of the original discrete-time signal by straight lines. Interpolation with the sinc interpolating function, finally, yields the signal of Fig. 1.5(d), which is smooth within the interval $[-4T, 4T]$ but shows , "ripple" outside it.

Remark: Are sampling and interpolation inverses of each other?
Interpolation of a discrete-time signal followed by sampling on the original discrete-time axis results in exact reconstruction of the original discrete-time signal. On the other hand, sampling a continuous-time signal followed by interpolation generally does not reproduce the original continuous-time signal. Evidently sampling is an inverse operation to interpolation, but interpolation is not inverse to sampling. Figure 1.6 illustrates this.

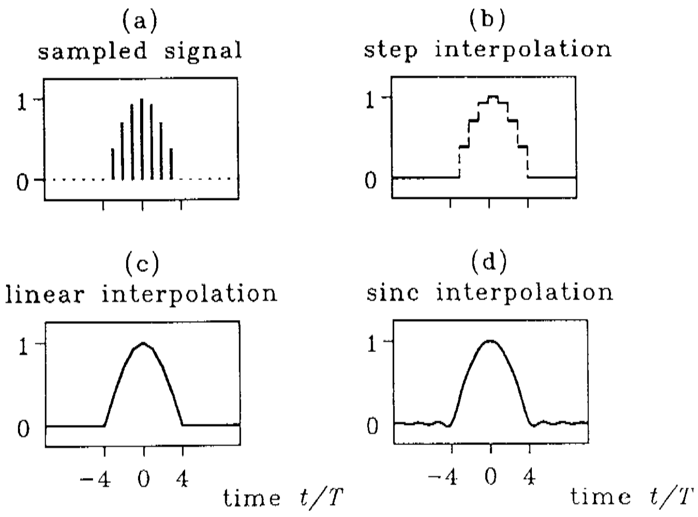

Figure 1.5: Interpolation. (a) A discrete-time signal. (b) Step interpolation (c) Linear interpolation. (d) Sinc interpolation.

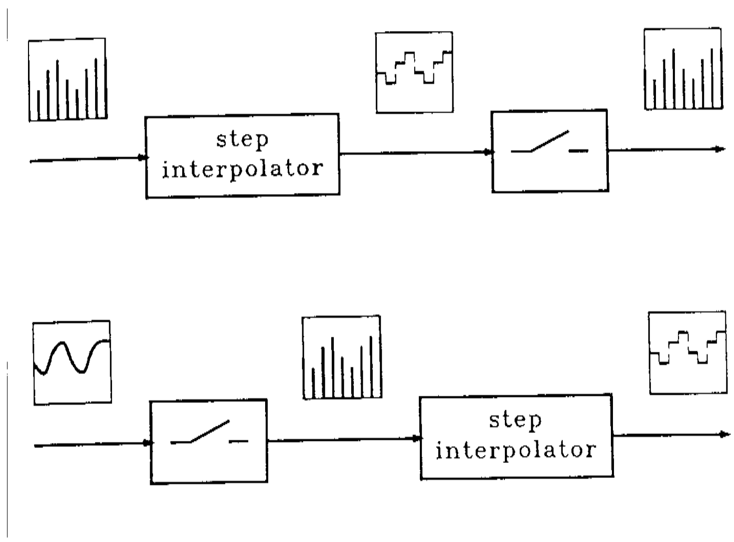

Figure 1.6: Sampling and interpolation. Top: sampling is the inverse of interpolation. Bottom: interpolation is not the inverse of sampling

Remark: Analog-to-Digital and digital-to-analog conversion
(a) *A/D conversion.* Computers and other digital equipment do not only operate on a discrete-time axis but are also limited to finite signal ranges. Thus, apart from sampling, conversion of real-valued continuous-time signals to input for digital equipment also involves quantization. The combined process of sampling and quantization is called analog-to-digital (A/D) conversion.
(b) *D/A conversion.* The inverse process of converting a quantized discrete-time signal to a real-valued continuous-time signal is called digital-to -analog (D/A) conversion. Devices that perform D/A conversion by step interpolation are known as zero-order hold circuits. They can function in "real time" meaning that given the sampled signal up to, and including, time t, the continuous-time signal up to and including that same time may be generated. Devices that perform D/A conversion by linear interpolation are known as first-order hold circuits. They cannot precisely function in real time because two successive sampled signal values have to be received before it is known how the continuous-time signal goes. First-order hold circuits therefore introduce a delay, equal to the sampling period T. Sinc interpolation, finally, cannot be implemented in real time at all because all sampled signal values need be received before any point of the continuous-time signal (except at the sampling times) may be computed. Sinc interpolation, however, has great theoretical importance.

1.3 Discrete-time systems

1.3.1 Difference Equations

In continuous-time the relationships between different model variables are described with the help of differential equations (see Rowell & Wormley [4]). In discrete-time this can be done using difference equations. There are two different ways of describing these difference equations. One is to directly relate inputs u to outputs y in one difference equation. In principle it looks like this:

$$g(y(k+n), y(k+n-1), \dots, y(k), u(k+m), u(k+m-1), \dots, u(k)) = 0 \tag{1.3}$$

where $g(\cdot, \cdot, \dots, \cdot)$ is an arbitrary, vector-valued, nonlinear function. The other way is to write the difference equation as a system of first-order difference equations by introducing a number of internal variables. If we denote these internal variables by

$$x_1(k), \dots, x_n(k)$$

and introduce the vector notation

$$x(t) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} \tag{1.4}$$

we can, in principle, write a system of first-order difference equations as

$$x(k+1) = f(x(k), u(k)) \tag{1.5}$$

In (1.5), $f(x, u)$ is a vector function with n components:

$$f(x,u) = \begin{bmatrix} f_1(x,u) \\ \vdots \\ f_n(x,u) \end{bmatrix} \tag{1.6}$$

The functions $f_i(x, u)$ are in turn functions of $n + m$ variables, the components of the x and u vectors. Without vector notation, (1.5) becomes

$$\begin{aligned} x_1(k+1) &= f_1(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)) \\ x_2(k+1) &= f_2(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)) \\ &\vdots \\ x_n(k+1) &= f_n(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)) \end{aligned} \tag{1.7}$$

The outputs of the model can then be calculated from the internal variables $x_i(k)$ and the inputs $u_i(k)$:

$$y(k) = h(x(k), u(k)) \tag{1.8}$$

which written in longhand means

$$\begin{aligned} y_1(k) &= h_1(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)) \\ y_2(k) &= h_2(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)) \\ &\vdots \\ y_p(k) &= h_p(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k)) \end{aligned} \tag{1.9}$$

1.3.2 Discrete-time state-space models

For a dynamic system the output depends on all earlier input values. This leads to the fact that it is not enough to know $u(k)$ for $k \geq k_0$ in order to be able to calculate the output $y(k)$ for $k \geq k_0$. We need information about the system. By the state of the system at time k_0 we mean an amount of information such that with this state and the knowledge of $u(k)$, $k \geq k_0$, we can calculate $y(k)$, $k \geq k_0$. This definition is well in line with the everyday meaning of the word "state." It is also obvious from the definition of state that this concept will play a major role in the simulation of the model. The state is exactly the information that has to be stored and updated at the simulation in order to be able to calculate the output. Consider a general system of first-order difference equations (1.5) with the output given by (1.8):

$$x(k+1) = f(x(k), u(k)) \tag{1.10}$$

$$y(k) = h(x(k), u(k)) \tag{1.11}$$

For this system the vector $x(k_0)$ is a state at time k_0. The difference equation (1.10)-(1.11) with $x(k_0) = x_0$ has a unique solution for $k \geq k_0$. The solution can easily be found using successive substitution:
Assume that we know $x(k)$ at time k_0 and $u(k)$ for $k \geq k_0$. We can then according to (1.10) calculate $x(k+1)$.
From this value we can use $u(k+1)$ to calculate $x(k+2)$, again using (1.10). We can continue and calculate $x(k)$ for all $k > k_0$. The output $y(k)$, $k \geq k_0$, can then also be computed according to (1.11).
We have thus established that the variables $x_1(k), \ldots, x_n(k)$ or, in other words, the vector

$$x(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

in the internal model description (1.10)-(1.11) is a state for the model. Herein lies the importance of this model description for simulation. The model (1.10)-(1.11) is therefore called a state-space model, the vector $x(k)$ the state vector, and its components $x_i(k)$ state variables. The dimension of $x(k)$, that is, n, is called the model order.
State-space models will be our standard model for dynamic discrete-time systems. In conclusion, we have the following model:
Discrete-time state space models

$$x(k+1) = f(x(k), u(k)) \qquad k = 0, 1, 2, \ldots \tag{1.12}$$
$$y(k) = h(x(k), u(k)) \tag{1.13}$$

$u(k)$: input at time k , an m-dimensional column vector.
$y(k)$: output at time k , a p-dimensional column vector.
$x(k)$: state at time k , an n-dimensional column vector.
The model is said to be n-th order. For a given initial value $x(k_0) = x_0$, (1.12)-(1.13) always has a unique solution.

1.3.3 Linear discrete-time models

The model (1.12)-(1.13) is said to be linear if $f(x, u)$ and $h(x, u)$ are linear functions of x and u:

$$f(x, u) = Ax + Bu \tag{1.14}$$
$$h(x, u) = Cx + Du \tag{1.15}$$

Here the matrices have the following dimensions

$$A : n \times n \qquad B : n \times m$$

$$C : p \times n \qquad D : p \times m$$

If these matrices are independent of time the model (1.14)-(1.15) is said to be linear and time invariant.
The model (1.3) is said to be linear if $g(\cdot,\cdot,\ldots,\cdot)$ is a linear function in y and u:

$$\begin{aligned} &g(y(k+n), y(k+n-1), \ldots, y(k), u(k+m), u(k+m-1), \ldots, u(k)) \\ &\quad = a_0\, y(k+n) + a_{n-1}\, y(k+n-1) + \ldots + a_0\, y(k) \\ &\qquad - b_m\, u(k+m) - b_{m-1}\, u(k+m-1) - \ldots - b_0\, u(k) = 0 \end{aligned}$$

or

$$a_0\, y(k+n) + a_{n-1}\, y(k+n-1) + \ldots + a_0\, y(k) = b_m\, u(k+m) + b_{m-1}\, u(k+m-1) + \ldots + b_0\, u(k) \quad (1.16)$$

Example 5 (Savings account)
Suppose that $y(k)$ represents the balance of a savings account at the beginning of day k , and $u(k)$ the amount deposited during day k. If interest is computed and added daily at a rate of $\alpha \cdot 100\%$, the balance at the beginning of day $k+1$ is given by

$$y(k+1) = (1+\alpha)y(k) + u(k), \quad k = 0, 1, 2, \cdots \quad (1.17)$$

This describes the savings account as a discrete-time system on the time axis $\mathbb{Z}_+$. If the interest rate α does not change with time, the system is time-invariant; otherwise, it is time-varying.

Example 6 (First-order exponential smoother)
In signal processing sometimes a procedure called exponential smoothing is used to remove unwanted fluctuations from observed time series, such as the Dow Jones index, the daily temperature, or measurements of a human's blood pressure. Let $u(k)$, $k \in \mathbb{Z}$ be a discrete-time signal, then, as we observe the successive values of the signal u we may form from these values another signal y, $k \in \mathbb{Z}$, which is a smoothed version of the signal u, according to

$$y(k+1) = a\, y(k) + (1-a)\, u(k+1) \quad (1.18)$$

Here, a is a constant such that $0 < a < 1$. At each time $k+1$, the output $y(k+1)$ is formed as a weighted average of the new input $u(k+1)$ at time $k+1$ and the output $y(k)$ at preceding time instant k. The closer the constant a is to 1, the more the preceding output value is weighted and the "smoother" the output is. Figure 1.7 illustrates the smoothing effect.

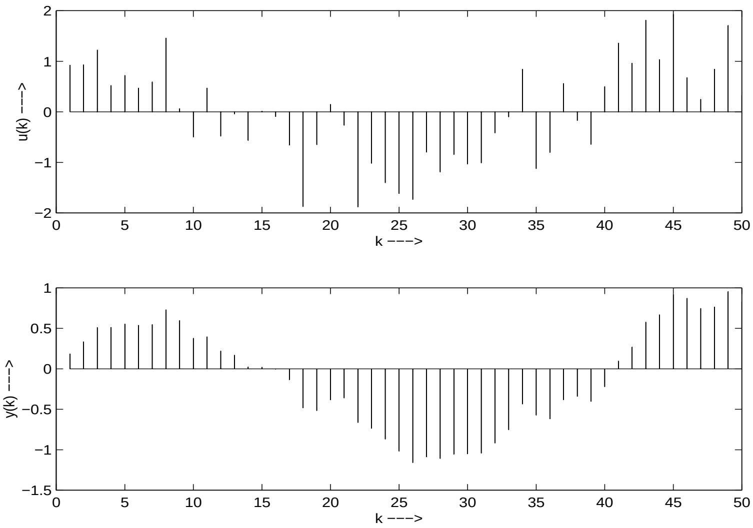

Figure 1.7: Exponential smoothing ($a = 0.8$)

Chapter 2

Operational methods for discrete-time linear systems

In this chapter we will consider operational methods for discrete-time linear systems. For the continuous-time case see Rowell & Wormley [4].

2.1 Discrete-time linear system operators

Consider a discrete-time signal $f(k)$, for $k \in \mathbb{Z}$. All linear time-invariant systems may be represented by an interconnection of the following primitive operators:

1. **The constant scaling operator:** The scaling operator multiplies the input function by a constant factor. It is denoted by the value of the constant, either numerically or symbolically, for example $2.0\{\}$, or $\alpha\{\}$.

2. **The forward shift operator:** The shift operator, designated $Z\{\}$, shifts the input signal in one step ahead in time:

$$y(k) = Z\{f(k)\} = f(k+1)$$

3. **The backward shift operator:** The shift operator, designated $Z^{-1}\{\}$, shifts the input signal in one step back in time:

$$y(k) = Z^{-1}\{f(k)\} = f(k-1)$$

4. **The difference operator:** The difference operator, written $\Delta\{\}$, generates the increment of the input $f(k)$:

$$y(k) = \Delta\{f(k)\} = f(k) - f(k-1)$$

5. **The summation operator:** The summation operator, written $\Delta^{-1}\{\}$, generates the summation of the input $f(k)$:

$$y(k) = \Delta^{-1}\{f(k)\} = \sum_{m=0}^{k} f(m)$$

where it is assumed that at time $k < 0$, the output $y(k) = 0$. If an initial condition $y(-1)$ is specified,

$$y(k) = \Delta^{-1}\{f(k)\} + y(-1)$$

and a separate summing block is included at the output of the summation block to account for the initial condition.

6. **The identity operator:** The identity operator leaves the value $f(k)$ unchanged, that is:

$$y(k) = I\{f(k)\} = f(k)$$

7. **The null operator:** The null operator $N\{\}$ identically produces an output of zero for any input, that is,

$$y(k) = N\{f(k)\} = 0$$

Remark 1: Note that by substituting $f(k-1) = Z^{-1}\{f(k)\}$ we find that

$$\Delta\{\} = I\{\} - Z^{-1}\{\}$$

Remark 2: The discrete-time operators Z, Z^{-1}, Δ, Δ^{-1}, I and N are linear operators. Therefore all properties for linear operators will also hold for these discrete-time operators.

Block diagram of a discrete-time system

The block diagram of the state-space description of a linear discrete-time system is equivalent to the one for continuous-time systems if we substitute the backward shift block Z^{-1} for the integral block. (In the book of Rowell & Wormley [4] it is the block S^{-1} as in Fig. 2.1.) We obtain the block diagram of figure 2.1.
The block diagram of general second-order discrete-time state system

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + d\, u(k)
\end{aligned}$$

is shown in figure 2.2.

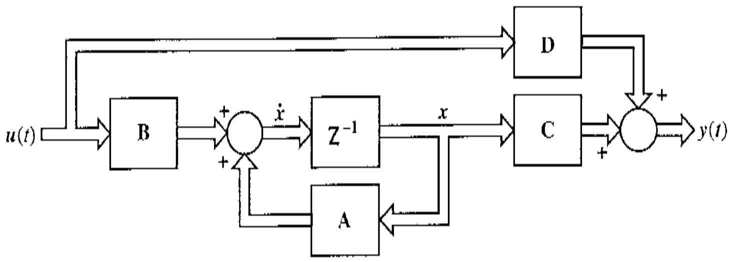

Figure 2.1: Vector block diagram for a linear discrete-time system described by state space system dynamics

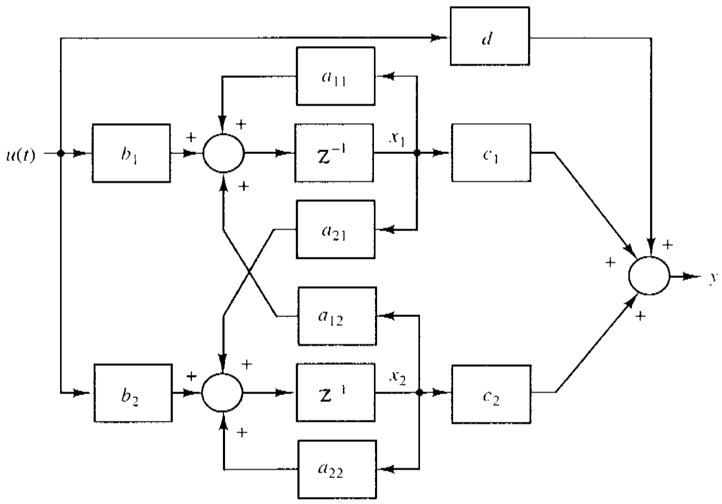

Figure 2.2: Block diagram for a state equation-based second-order discrete-time system

Input-output linear discrete-time system models

For systems that have only one input and one output, it is frequently convenient to work with the classical system input-output description in Eq. (1.16) consisting of a single n-th order difference equation relating the output to the system input:

$$\begin{aligned} & a_n\, y(k+n) + a_{n-1}\, y(k+n-1) + \ldots + a_0\, y(k) \\ = \; & b_m\, u(k+m) + b_{m-1}\, u(k+m-1) + \ldots + b_0\, u(k) \end{aligned} \tag{2.1}$$

The constant coefficients a_i and b_i are defined by the system parameters and for causal systems there holds: $m \leq n$.
Equation (2.1) may be written in operational form using polynomial operators as:

$$\left[a_n\, Z^n + a_{n-1}\, Z^{n-1} + \ldots + a_0\right] \{y\} = \left[b_m\, Z^m + b_{m-1}\, Z^{m-1} + \ldots + b_0\right] \{u\} \tag{2.2}$$

If the inverse operator $\left[a_n\, Z^n + a_{n-1}\, Z^{n-1} + \ldots + a_0\right]^{-1}$ exists, it may be applied to each side to produce an explicit operational expression for the output variable:

$$y(k) = \frac{b_m\, Z^m + b_{m-1}\, Z^{m-1} + \ldots + b_0}{a_n\, Z^n + a_{n-1}\, Z^{n-1} + \ldots + a_0} \{u(k)\} \tag{2.3}$$

The operational description of the system dynamics has been reduced to the form of a single linear operator, the dynamic transfer operator $H\{\}$,

$$y(k) = H\{u(k)\} \tag{2.4}$$

where

$$H\{\} = \frac{b_m\, Z^m + b_{m-1}\, Z^{m-1} + \ldots + b_0}{a_n\, Z^n + a_{n-1}\, Z^{n-1} + \ldots + a_0} \{\} \tag{2.5}$$

In (1.14)-(1.15), the state space description was given for a linear discrete-time system:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.6}$$
$$y(k) = Cx(k) + Du(k) \tag{2.7}$$

The matrix transfer operator for this system can be computed by:

$$H\{\} = \frac{C\, \mathrm{adj}(Z\, I - A)\, B + \det[Z\, I - A]\, D}{\det[Z\, I - A]} \tag{2.8}$$

2.2 Transformation of discrete-time systems

State realization of a polynomial difference system

The polynomial difference system (2.3), described by

$$\left[a_n\, Z^n + a_{n-1}\, Z^{n-1} + \ldots + a_0\right] y(k) = \left[b_m\, Z^m + b_{m-1}\, Z^{m-1} + \ldots + b_0\right] u(k)$$

has a state realization

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

such that

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -a_0/a_n & -a_1/a_n & \cdots & -a_{n-2}/a_n & -a_{n-1}/a_n \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1/a_n \end{bmatrix}$$

$$C = \begin{bmatrix} b_0 - a_0 b_n/a_n & b_1 - a_1 b_n/a_n & \cdots & b_{n-2} - a_{n-2} b_n/a_n & b_{n-1} - a_{n-1} b_n/a_n \end{bmatrix}$$

$$D = b_n/a_n$$

Polynomial realization of a state system

The state difference system (2.6)-(2.7), described by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

has a polynomial realization

$$\left[\det[Z\,I - A]\right] y(k) = \left[C\,\mathrm{adj}(ZI - A)B + \det[ZI - A]D\right] u(k) \tag{2.9}$$

and a transfer operator

$$\begin{aligned} H &= C\,(ZI - A)^{-1}\,B \,+\, D && (2.10) \\ &= \frac{C\,\mathrm{adj}(ZI - A)B + \det[ZI - A]D}{\det[Z\,I - A]} && (2.11) \end{aligned}$$

Example 7 (Polynomial to state-space)
Consider the difference system

$$16\,y(k+3) - 20\,y(k+2) + 8\,y(k+1) - \,y(k) = 5\,u(k+2) - 7\,u(k+1) + 2\,u(k)$$

the state space realization is given by

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/16 & -8/16 & 20/16 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/16 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} 2 & -7 & 5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}
\end{aligned}$$

Example 8 (State-space to polynomial)
Consider the state system

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} -0.5 & 1.5 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + 2\,u(k)
\end{aligned}$$

The determinant of $[Z\,I - A]$ *is*

$$det[Z\,I - A] = Z^2 - 1.5\,Z + 0.5$$

and the adjoint of $[Z\,I - A]$ *is*

$$adj\begin{bmatrix} Z+0.5 & -1.5 \\ 1 & Z-2 \end{bmatrix} = \begin{bmatrix} Z-2 & 1.5 \\ -1 & Z+0.5 \end{bmatrix}$$

Further we compute

$$\begin{aligned}
&C\;adj(ZI - A)B + det[ZI - A]D = \\
&\qquad = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} Z-2 & 1.5 \\ -1 & Z+0.5 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} + (Z^2 - 1.5\,Z + 0.5)\,2 \\
&\qquad = (2\,Z - 6) + (2Z^2 - 3\,Z + 1) \\
&\qquad = 2\,Z^2 - Z - 5
\end{aligned}$$

So the related difference equation is given by equation (2.9):

$$(Z^2 - 1.5\,Z + 0.5)\,y(k) = (2\,Z^2 - Z - 5)\,u(k)$$

or

$$y(k+2) - 1.5\,y(k+1) + 0.5\,y(k) = 2\,u(k+2) - u(k+1) - 5\,u(k)$$

Chapter 3

System Properties and Solution techniques

In this chapter we discuss system properties and solution techniques for discrete-time systems. (For the continuous case, see Rowell & Wormley [4], chapter 8).

3.1 Singularity input functions

In Rowell & Wormley [4], chapter 8, some singularity input functions for continuous-time systems were defined. In this section we redefine some of the functions for discrete-time:

The Unit Step Function: The discrete-time unit step function $u_s(k)$, $k \in \mathbb{Z}$ is defined as:

$$u_s(k) = \begin{cases} 0 & \text{for } k < 0 \\ 1 & \text{for } k \geq 0 \end{cases} \tag{3.1}$$

The Unit Impulse Function: The discrete-time unit Impulse function $\delta(k)$, $k \in \mathbb{Z}$ is defined as:

$$\delta(k) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases} \tag{3.2}$$

Note that for discrete-time the definition of a function $\delta_T(k)$ is not appropriate. The discrete-time impulse function (or discrete-time Dirac delta function) has the property that

$$\sum_{m=-\infty}^{\infty} \delta(m) = 1$$

The unit ramp function: The discrete-time unit ramp function $u_r(k)$, $k \in \mathbb{Z}$ is defined to be a linearly increasing function of time with a unity increment:

$$u_r(k) = \begin{cases} 0 & \text{for } k \leq 0 \\ k & \text{for } k > 0 \end{cases} \tag{3.3}$$

Note that

$$\Delta\{u_r(k+1)\} = u_r(k+1) - u_r(k) = u_s(k) \tag{3.4}$$

and

$$\Delta\{u_s(k)\} = u_s(k) - u_s(k-1) = \delta(k) \tag{3.5}$$

and also in the reverse direction:

$$\Delta^{-1}\{\delta(k)\} = \sum_{m=0}^{k} \delta(k) = u_s(k) \tag{3.6}$$

and

$$\Delta^{-1}\{u_s(k)\} = \sum_{m=0}^{k} u_s(k) = u_r(k+1) \tag{3.7}$$

3.2 Classical solution of linear difference equations

In this section we briefly review the classical method for solving a linear nth-order ordinary difference equation with constant coefficients, given by

$$\begin{aligned} & y(k+n) + a_{n-1}\, y(k+n-1) + \ldots + a_0\, y(k) \\ = \; & b_m\, u(k+m) + b_{m-1}\, u(k+m-1) + \ldots + b_0\, u(k) \end{aligned} \tag{3.8}$$

where in general $m \leq n$. We define the forcing function

$$f(k) = b_m\, u(k+m) + b_{m-1}\, u(k+m-1) + \ldots + b_0\, u(k) \tag{3.9}$$

which is known, because $u(k)$ is known for all $k \in \mathbb{Z}$. Eq. (3.8) now becomes

$$y(k+n) + a_{n-1}\, y(k+n-1) + \ldots + a_0\, y(k) = f(k) \tag{3.10}$$

The task is to find a unique function $y(k)$ for $k \geq k_0$ that satisfies the difference equation (3.8) given the forcing function $f(k)$ and a set of initial conditions $y(k_0), y(k_0+1), \ldots, y(k_0+n-1)$.

The general solution to Eq. (3.8) may be derived as the sum of two solution components

$$y(k) = y_h(k) + y_p(k) \tag{3.11}$$

where y_h is the solution of the homogeneous equation (so for $f(k) = 0$) and $y_p(k)$ is a particular solution that satisfies (3.10) for the specific $f(k)$ (but arbitrary initial conditions).

3.2.1 Homogeneous solution of the difference equation

The homogeneous difference equation ($a_n \neq 0$) is given by

$$a_n\, y(k+n) + a_{n-1}\, y(k+n-1) + \ldots + a_0\, y(k) = 0 \tag{3.12}$$

The standard method of solving difference equations assumes there exists a solution of the form $y(k) = C\,\lambda^k$, where λ and C are both constants. Substitution in the homogeneous equation gives:

$$C\Big(a_n\lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_0\Big)\lambda^k = 0 \tag{3.13}$$

For any nontrivial solution, C is nonzero and λ^k is never zero, we require that

$$a_n\lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_0 = 0 \tag{3.14}$$

For an nth-order system with $a_n \neq 0$, there are n (complex) roots of the characteristic polynomial and n possible solution terms $C_i\,\lambda_i^k$ $(i = 1, \ldots, n)$, each with its associated constant. The homogeneous solution will be the sum of all such terms:

$$y_h(k) = C_1\,\lambda_1^k + C_2\,\lambda_2^k + \ldots + C_n\,\lambda_n^k \tag{3.15}$$

$$= \sum_{i=1}^{n} C_i\,\lambda_i^k \tag{3.16}$$

The n coefficients C_i are arbitrary constants and must be found from the n initial conditions.
If the characteristic polynomial has repeated roots, that is $\lambda_i = \lambda_j$ for $i \neq j$, there are not n linearly independent terms in the general homogeneous response (3.16). In general if a root λ of multiplicity m occurs, the m components in the general solution are

$$C_1\,\lambda^k \;\;,\;\; C_2\,k\,\lambda^k \;\;, \ldots ,\;\; C_m\,k^{m-1}\,\lambda^m$$

If one or more roots are equal to zero ($\lambda_i = 0$), the solution $C\,\lambda^k = 0$ has no meaning. In that case we add a term $C\,\delta(k-k_0)$ to the general solution. If the root $\lambda = 0$ has multiplicity m, the m components in the general solution are

$$C_1\,\delta(k-k_0) \;\;,\;\; C_2\,\delta(k-k_0-1) \;\;, \ldots ,\;\; C_m\,\delta(k-k_0-m+1)$$

3.2.2 Particular solution of the difference equation

Also for discrete-time systems, the method of undetermined coefficients can be used (see [4], page 255-257), where we use table 3.1 to find the particular solutions for some specific forcing functions.

Terms in $u(k)$	Assumed form for $y_p(k)$	Test value
α	β_1	0
$\alpha\, k^n$, $(n = 1, 2, 3, \ldots)$	$\beta_n\, k^n + \beta_{n-1}\, k^{n-1} + \ldots + \beta_1\, k + \beta_0$	0
$\alpha\, \lambda^k$	$\beta\, \lambda^k$	λ
$\alpha\, \cos(\omega k)$	$\beta_1\, \cos(\omega k) + \beta_2\, \sin(\omega k)$	$j\omega$
$\alpha\, \sin(\omega k)$	$\beta_1\, \cos(\omega k) + \beta_2\, \sin(\omega k)$	$j\omega$

Figure 3.1: Definition of Particular $y_p(k)$ using the method of undetermined coefficients

Example 9 (Solution of a difference equation)
Consider the system described by the difference equation

$$2\, y(k+3) + y(k+2) = 7\, u(k+1) - u(k)$$

Find the system response to a ramp input $u(k) = k$ *and initial conditions given by* $y(0) = 2$, $y(1) = -1$ *and* $y(2) = 2$.

Solution:
homogeneous solution: The characteristic equation is:

$$2\, \lambda^3 + \lambda^2 = 0$$

which has a root $\lambda_1 = -0.5$ *and a double root* $\lambda_{2,3} = 0$. *The general solution of the homogeneous equation is therefore*

$$y_h(k) = C_1\, (-0.5)^k + C_2\, \delta(k) + C_3\, \delta(k-1)$$

particular solution: From table 3.1 we find that for $u(k) = k$, *the particular solution is selected:*

$$y_p = \beta_1\, k + \beta_0$$

Testing gives:

$$2\, (\beta_1\, (k+3) + \beta_0) + (\beta_1\, (k+2) + \beta_0) = 7\, (k+1) - k = 6\, k + 7$$

We find $\beta_1 = 2$ *and* $\beta_0 = -3$ *and so the particular solution is*

$$y_p(k) = 2\, k - 3$$

complete solution: The complete solution will have the form:

$$y(k) = y_h(k) + y_p(k) = C_1\, (-0.5)^k + C_2\, \delta(k) + C_3\, \delta(k-1) + 2\, k - 3$$

Now the initial conditions are evaluated in $k = 0$, $k = 1$ and $k = 2$.

$$\begin{aligned} y(0) &= C_1\,(-0.5)^0 + C_2\,\delta(0) + C_3\,\delta(0-1) + 2\,0 - 3 \\ &= C_1 + C_2 - 3 = 2 \\ y(1) &= C_1\,(-0.5)^1 + C_2\,\delta(1) + C_3\,\delta(1-1) + 2\,1 - 3 \\ &= -C_1\,0.5 + C_3 - 1 = -1 \\ y(2) &= C_1\,(-0.5)^2 + C_2\,\delta(2) + C_3\,\delta(2-1) + 2\,2 - 3 \\ &= C_1\,0.25 + 1 = 2 \end{aligned}$$

We find a solution for $C_1 = 4$, $C_2 = 1$, $C_3 = 2$ and so the final solution becomes:

$$y(k) = 4\,(-0.5)^k + \delta(k) + 2\,\delta(k-1) + 2\,k - 3$$

3.3 Discrete-time Convolution

In this section we derive the computational form of the discrete-time system $H\{u(k)\}$, defined in [4], chapter 7, and section 2.1 of these lecture notes, that is based on a system's response to an impulse input. We assume that the system is initially at rest, that is, all initial conditions are zero at time $t = 0$, and examine the discrete-time domain forced response $y(k)$, $k \in \mathbb{Z}$ to a discrete-time waveform $u(k)$.
To start with, we assume that the system response to $\delta(k)$ is a known function and is designated $h(k)$ as shown in figure 3.2.

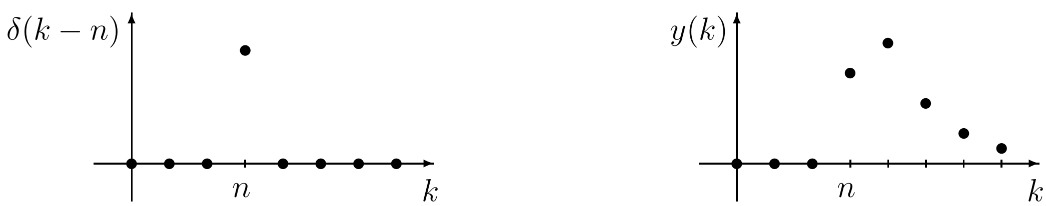

Figure 3.2: System response to a delayed unit pulse

Then if the system is linear and time-invariant, the response to a delayed unit pulse, occurring at time n is simply a delayed version of the pulse response:

$$u(k) = \delta(k-n) \quad \text{gives} \quad y(k) = h(k-n) \tag{3.17}$$

Multiplication with a constant α gives:

$$u(k) = \alpha\,\delta(k-n) \quad \text{gives} \quad y(k) = \alpha\,h(k-n) \tag{3.18}$$

The input signal $u(k)$ may be considered to be the sum of non-overlapping delayed pulses $p_n(k)$:

$$u(k) \quad = \quad \sum_{n=-\infty}^{\infty} p_n(k) \tag{3.19}$$

where

$$p_n(k) \quad = \quad \begin{cases} u(k) & \text{for} \quad k = n \\ 0 & \text{for} \quad k \neq n \end{cases} \tag{3.20}$$

Each component $p_n(k)$ may be written in terms of a delayed unit pulse $\delta(k)$, that is

$$p_n(k) = u(n)\delta(k-n)$$

From equation (3.18) it follows that for a delayed version of the pulse response, multiplied by a constant $u(n)$ we obtain the output $y(k) = u(n)\, h(k-n)$.
For an input

$$u(k) = \sum_{n=-\infty}^{\infty} p_n(k)$$

we can use the principle of superposition and the output can be written as a sum of all responses to $p_n(k)$, so:

$$y(k) = \sum_{n=-\infty}^{\infty} u(n)\, h(k-n) \tag{3.21}$$

This sum is denoted as the convolution sum for discrete-time systems.
For physical systems, the pulse response $h(k)$ is zero for time $k < 0$, and future components of the input do not contribute to the sum. So the convolution sum becomes:

$$y(k) = \sum_{n=-\infty}^{k} u(n)\, h(k-n) \tag{3.22}$$

The convolution operation is often denoted by the symbol $*$:

$$y(k) = u(k) * h(k) = \sum_{n=-\infty}^{k} u(n)\, h(k-n) \tag{3.23}$$

Example 10 (Discrete-time convolution)
The first-order exponential smoother of example 6 is described by the difference equation

$$y(k+1) = a\, y(k) + (1-a)\, u(k+1)$$

By repeated substitution it easily follows that if $y(k) = 0$ for $k < n_0$, the output of the system is given by

$$y(k) = (1-a) \sum_{n=n_0}^{k} a^{k-n}\, u(n) \qquad k \geq n_0$$

assuming that the input $u(k)$ *is such that the sum converges, and let* n_0 *approach to* $-\infty$. *Then the response of the system takes the form*

$$y(k) = (1-a)\sum_{n=-\infty}^{k} a^{k-n}\, u(n) \qquad k \in \mathbb{Z}$$

Define the function h *such that*

$$\begin{aligned} h(k) &= \begin{cases} 0 & \textit{for} \quad k < 0 \\ (1-a)a^k & \textit{for} \quad k \geq 0 \end{cases} \\ &= (1-a)a^k\, u_s(k) \end{aligned}$$

we see that on the infinite time axis the system may be represented as the convolution sum

$$y(k) = \sum_{n=-\infty}^{\infty} h(k-n)\, u(k)$$

Chapter 4

Solution of discrete-time linear state equations

In this chapter we will discuss the general solution of discrete-time linear state equations. (For the continuous-time case, see Rowell & Wormley [4], chapter 10).

4.1 State variable responses of discrete-time linear systems

In this chapter we examine the responses of linear time-invariant discrete-time models in the standard state equation form

$$x(k+1) = Ax(k) + Bu(k) \tag{4.1}$$

$$y(k) = Cx(k) + Du(k) \tag{4.2}$$

The solution proceeds in two steps: First the state-variable response $x(k)$ is determined by solving the set of first-order state equations Eq. (4.1), and then the state response is substituted into the algebraic output equations, Eq. (4.2) in order to compute $y(k)$.

4.1.1 Homogeneous State responses

The state variable response of a system described by Eq. (4.1) with zero input and an arbitrary set of initial conditions $x(0)$ is the solution of the set of n homogeneous first-order difference equations

$$x(k+1) = Ax(k) \tag{4.3}$$

We can find the solution by successive substitution. Note that $x(1) = A\,x(0)$ and $x(2) = A\,x(1)$, so by substitution we obtain $x(2) = A^2\,x(0)$. Proceeding in this way we derive:

$$x(k) = A^k\,x(0) \tag{4.4}$$

The solution is often written as:

$$x(k) = \Phi(k)\, x(0) \tag{4.5}$$

where $\Phi(k) = A^k$ is defined to be the state transition matrix.

Example 11 (First-order exponential smoother)
In example 6, the first-order exponential smoother was presented. The state difference representation and output equations are

$$\begin{aligned} x(k+1) &= a\, x(k) + a(1-a)\, u(k) \\ y(k) &= x(k) + (1-a)\, u(k), \qquad k \in \mathbb{Z} \end{aligned}$$

The homogeneous state difference equation is given by

$$x(k+1) = ax(k)$$

It follows that the 1×1 *state transition matrix is given by*

$$\Phi(k) = A\, A \cdots A = a \cdot a \cdots a = a^k, \;\; k > 0, \;\; k \in \mathbb{Z}.$$

Example 12 (Second-order exponential smoother)
Consider now the second-order smoother described by the difference equation

$$y(k+2) - a_1 y(k+1) - a_0 y(k) = b_2 u(k+2) + b_1 u(k+1), \qquad k \in \mathbb{Z}$$

Since the order of the system is $N = 2$*, the dimension of the state is also 2. According to the transformation formulae in section 2.2, the system may be represented in state form as*

$$\begin{aligned} x(k+1) &= \begin{bmatrix} a_1 & 1 \\ a_0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} b_1 + a_1 b_2 \\ a_0 b_2 \end{bmatrix} u(k), \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} + b_2 u(k), \qquad k \in \mathbb{Z} \end{aligned}$$

The homogeneous state difference equation is

$$x(k+1) = \begin{bmatrix} a_1 & 1 \\ a_0 & 0 \end{bmatrix} x(k) = Ax(k) \tag{4.6}$$

Then the 2×2 *state transition matrix is given by*

$$\Phi(k) = A \cdot A \cdots A = A^k$$

For instance, let us take numerical values $a_0 = 0$*, and* $a_1 = \frac{1}{2}$*, then*

$$A = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & 0 \end{bmatrix}; \tag{4.7}$$

and it can be seen that

$$\Phi(k) = \begin{bmatrix} (\frac{1}{2})^k & 2(\frac{1}{2})^k \\ 0 & 0 \end{bmatrix}; \qquad k \geq 0,\ k \in \mathbb{Z} \tag{4.8}$$

Consider homogeneous system (4.6) with system matrix (4.7) and initial condition

$$x(0) = \begin{bmatrix} 16 \\ 4 \end{bmatrix}$$

Then for $k = 3$ we find:

$$x(3) = \Phi(3)\,x(0) = \Phi(k) = \begin{bmatrix} (\frac{1}{2})^3 & 2(\frac{1}{2})^3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 16 \\ 4 \end{bmatrix} = \begin{bmatrix} 1/8 & 1/4 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 16 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \tag{4.9}$$

4.1.2 The forced response of discrete-time linear systems

The solution of the inhomogeneous discrete-time state equation

$$x(k+1) = Ax(k) + Bu(k) \tag{4.10}$$

$$\tag{4.11}$$

follows easily by induction

$$\begin{aligned} x(k+2) &= Ax(k+1) + Bu(k+1) \\ &= A\Big(Ax(k) + Bu(k)\Big) + Bu(k+1) \\ &= A^2x(k) + ABu(k) + Bu(k+1) \end{aligned}$$

Proceeding in this way we derive

$$x(k) = A^k\,x(0) + \sum_{m=0}^{k-1} A^{k-m-1}B\,u(m) \tag{4.12}$$

for $k \geq 1$. For $k \leq 0$, the summation does not contribute to the solution, and we therefore multiply this term by $u_s(k-1)$, which is zero for $k \leq 0$. We obtain:

$$x(k) = A^k\,x(0) + \sum_{m=0}^{k-1} A^{k-m-1}B\,u(m)\,u_s(k-1) \tag{4.13}$$

Example 13 (Second-order exponential smoother)
Consider again the second order exponential smoother of example 12. Since

$$A(k) = \begin{bmatrix} a_1 & 1 \\ a_0 & 0 \end{bmatrix}; \qquad B(k) = \begin{bmatrix} b_1 + a_1b_2 \\ a_0b_2 \end{bmatrix}$$

yields

$$x(k) = A^k\,x(0) + \sum_{m=0}^{k-1} A^{k-m-1}B\,u(m)\,u_s(k-1)$$

Let us adopt the numerical values $a_0 = 0$, $a_1 = \frac{1}{2}$, $b_1 = 1$ *and* $b_2 = 0$ *resulting in* A *as in (4.7) and* B, C, *and* D *as follows*

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}; \qquad D = 0 \tag{4.14}$$

With (4.8) we may express the forced response of this system as

$$x(k) = \begin{bmatrix} (\frac{1}{2})^k & 2(\frac{1}{2})^k \\ 0 & 0 \end{bmatrix} x(0) + \sum_{m=0}^{k-1} \begin{bmatrix} (\frac{1}{2})^{k-m-1} \\ 0 \end{bmatrix} u(m)\, u_s(k-1)$$

4.1.3 The system output response of discrete-time linear systems

The output response of a discrete-time linear system is easily derived by substitution of (4.13) in (4.2):

$$\begin{aligned} y(k) &= Cx(k) + Du(k) \\ &= C\left(A^k\, x(0) + \sum_{m=0}^{k-1} A^{k-m-1} B\, u(m)\, u_s(k-1)\right) + Du(k) \\ &= CA^k\, x(0) + \sum_{m=0}^{k-1} CA^{k-m-1} B\, u(m)\, u_s(k-1) + Du(k) \end{aligned}$$

So the forced output response is given by

$$y(k) = CA^k\, x(0) + \sum_{m=0}^{k-1} CA^{k-m-1} B\, u(m)\, u_s(k-1) + Du(k)$$

Example 14 (Second-order exponential smoother)
Based on example 13, we may obtain for the second-order exponential smoother the following output response

$$\begin{aligned} y(k) &= (\frac{1}{2})^k x_1(0) + \sum_{m=0}^{k-1} (\frac{1}{2})^{k-m-1}\, u(m)\, u_s(k-1) \\ &= (\frac{1}{2})^k x_1(0) + \sum_{m=0}^{k-1} (\frac{1}{2})^{m}\, u(k-m-1)\, u_s(k-1) \end{aligned}$$

4.1.4 The discrete-time transition matrix

The discrete-time transition matrix $\Phi(k)$ has the following properties:

1. $\Phi(0) = I$.
2. $\Phi(-k) = \Phi^{-1}(k)$.

3. $\Phi(k_1)\Phi(k_2) = \Phi(k_1 + k_2)$, and so

$$x(k_2) = \Phi(k_2)x(0) = \Phi(k_2)\Phi(-k_1)x(k_1) = \Phi(k_2 - k_1)x(k_1)$$

or

$$x(k_2) = \Phi(k_2 - k_1)x(k_1)$$

4. If A is a diagonal matrix, then A^k is also a diagonal matrix and each element is the k-th power of the corresponding diagonal element of the A matrix, that is, a_{ii}^k.

4.1.5 System Eigenvalues and Eigenvectors

Let A be the system matrix of system (4.1)-(4.2). The values λ_i satisfying the equation

$$\lambda_i \, m_i = A \, m_i \qquad \text{for} \;\; m_i \neq 0 \tag{4.15}$$

are known as the eigenvalues, or characteristic values, of A. The corresponding column vectors m are defined as eigenvectors, or characteristic vectors. Equation (4.15) can be rewritten as:

$$(\lambda_i I - A) \, m_i = 0 \tag{4.16}$$

The condition for a non-trivial solution of such a set of linear equations is that

$$\det(\lambda_i I - A) = 0 \tag{4.17}$$

which is defined as the characteristic polynomial of the A matrix. Eq. (4.17) may be written as

$$\lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_1\lambda + a_0 = 0 \tag{4.18}$$

or, in factored form in terms of its roots $\lambda_1, \ldots, \lambda_n$,

$$(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) = 0 \tag{4.19}$$

Define

$$M = \begin{bmatrix} m_1 & m_2 & \cdots & m_n \end{bmatrix}$$

and

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & \lambda_n \end{bmatrix}$$

then

$$\Lambda^k = \begin{bmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \lambda_2^k & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & \lambda_n^k \end{bmatrix}$$

and

$$\Phi(k) = A^k = (M\,\Lambda\,M^{-1})^k = M\,\Lambda^k\,M^{-1}$$

4.1.6 Stability of discrete-time linear systems

For asymptotic stability, the homogeneous response of the state vector $x(k)$ returns to the origin for arbitrary initial conditions $x(0)$ at time $k \to \infty$, or

$$\lim_{k\to\infty} x(k) = \lim_{k\to\infty} \Phi(k)\,x(0) = \lim_{k\to\infty} M\,\Lambda^k\,M^{-1}\,x(0) = 0$$

for any $x(0)$. All the elements are a linear combination of the modal components λ_i^k, therefore, the stability of a system response depends on all components decaying to zero with time. If $|\lambda| > 1$, the component will grow exponentially with time and the sum is by definition unstable. The requirements for system stability may therefore be summarized:

A linear discrete-time system described by the state equation $x(k+1) = A\,x(k) + b\,u(k)$ is asymptotically stable if and only if all eigenvalues have magnitude smaller than one.

Three other separate conditions should be considered:

1. If one or more eigenvalues, or pair of conjugate eigenvalues, has a magnitude larger than one, there is at least one corresponding modal component that increases exponentially without bound from any initial condition, violating the definition of stability.

2. Any pair of conjugate eigenvalues that have magnitude equal to one, $\lambda_{i,i+1} = e^{\pm j\omega}$, generates an undamped oscillatory component in the state response. The magnitude of the homogeneous system response neither decays nor grows but continues to oscillate for all time at a frequency ω. Such a system is defined to be marginally stable.

3. An eigenvalue $\lambda = 1$ generates a model exponent $\lambda^k = 1^k = 1$ that is a constant. The system response neither decays nor grows, and again the system is defined to be marginally stable.

Example 15 (Second-order exponential smoother)
Consider the second-order exponential smoother of example 12. With the numerical values $a_0 = 0$, $a_1 = \frac{1}{2}$, $b_1 = 1$ *and* $b_2 = 0$, *the system matrices are given by 4.7 and 4.14. The characteristic polynomial of the matrix A is*

$$det(\lambda I - A) = det\begin{bmatrix} \lambda - \frac{1}{2} & -1 \\ 0 & \lambda \end{bmatrix} = \lambda(\lambda - \frac{1}{2}) \tag{4.20}$$

As a result, the eigenvalues of A are $\lambda_1 = 0$ *and* $\lambda_2 = \frac{1}{2}$. *Thus since all its eigenvalues are less than one, this discrete-time system is stable.*

4.1.7 Transformation of state variables

Consider the discrete-time system

$$x(k+1) = Ax(k) + Bu(k) \tag{4.21}$$
$$y(k) = Cx(k) + Du(k) \tag{4.22}$$

and let consider the eigenvalue decomposition of the A matrix

$$A = M \Lambda M^{-1}$$

Then this system can be transformed to the diagonal form

$$x'(k+1) = A'x'(k) + B'u(k) \tag{4.23}$$
$$y(k) = C'x'(k) + D'u(k) \tag{4.24}$$

by choosing

$$\begin{aligned} x'(k) &= M^{-1}x(k) \\ A' &= M^{-1}AM = \Lambda \\ B' &= M^{-1}B \\ C' &= C\,M \\ D' &= D \end{aligned}$$

Example 16 (Second-order exponential smoother)
Consider the second-order exponential smoother of example 12 and example 15. It can be found that the corresponding eigenvectors are

$$v_1 = \begin{bmatrix} 1 \\ -\frac{1}{2} \end{bmatrix}; \qquad v_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \tag{4.25}$$

It follows that the modal transformation matrix V, its inverse V^{-1} *and the diagonal matrix* Λ *are*

$$V = \begin{bmatrix} 1 & 1 \\ -\frac{1}{2} & 0 \end{bmatrix}; \quad V^{-1} = \begin{bmatrix} 0 & -2 \\ 1 & 2 \end{bmatrix}; \quad \Lambda = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}; \tag{4.26}$$

Thus, after modal transformation the system is represented as in Eqs. (4.23) and (4.24) where

$$A' = V^{-1}AV = \Lambda = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix};$$

$$B' = V^{-1}B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \qquad C' = CV = \begin{bmatrix} 1 & 1 \end{bmatrix};$$

4.2 The response of linear DT systems to the impulse response

Let

$$u(k) = \delta(k)$$

where $\delta(k)$ is the impulse response, defined in Eq. (3.2).
Following Eq. (4.15) we find

$$y(k) = CA^k x(0) + \sum_{m=0}^{k-1} CA^{k-m-1}B\,\delta(m)\,u_s(k-1) + D\delta(k) \tag{4.27}$$

Note that the term

$$\sum_{m=0}^{k-1} CA^{k-m-1}B\,\delta(m)\,u_s(k-1)$$

only gives a contribution for $m = 0$, so:

$$\sum_{m=0}^{k-1} CA^{k-m-1}B\,\delta(m)\,u_s(k-1) = CA^{k-1}B\,u_s(k-1)$$

Eq. (4.27) now becomes

$$y(k) = CA^k x(0) + CA^{k-1}B\,u_s(k-1) + D\delta(k) \tag{4.28}$$

which is the impulse response of a linear DT system.

Remark:
Note that a state transformation does not influence the input-output behaviour of the system. We therefore may use the diagonal form of the state space description of the system. The impulse response becomes:

$$\begin{aligned} y(k) &= C'A'^k x(0) + C'A'^{k-1}B'\,u_s(k-1) + D'\delta(k) \\ &= CM\Lambda^k x(0) + CM\Lambda^{k-1}M^{-1}B\,u_s(k-1) + D\delta(k) \end{aligned}$$

which can easily be computed.

Example 17 (Second-order exponential smoother)
Consider again the second order exponential smoother of example 12 and example 15. The state transition matrix of the transformed system is

$$\Phi'(k) = \Lambda^k = \begin{bmatrix} 0 & 0 \\ 0 & (\frac{1}{2})^k \end{bmatrix}; k \geq k_0 \tag{4.29}$$

It follows for the impulse response of the transformed system, which is also the impulse response of the untransformed system,

$$\begin{aligned} h(k) &= C'\Lambda^{k-1}B'u_s(k-1) + D\delta(k) \\ &= \begin{bmatrix} 1 \; 1 \end{bmatrix} \begin{bmatrix} \Delta(k-1) & 0 \\ 0 & (\frac{1}{2})^{k-1} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= (\frac{1}{2})^{k-1} u_s(k-1). \end{aligned}$$

Chapter 5

The Discrete-time transfer function

In this chapter we will discuss the transfer function of discrete-time linear state equations. (For the continuous-time case, see Rowell & Wormley [4], chapter 12).

5.1 Introduction

The concept of the transfer function is developed here in terms of the *particular solution* component of the total system response when the system is excited by a given exponential input waveform of the form

$$u(k) = U(z)z^k \tag{5.1}$$

where $z = \rho\, e^{j\,\psi}$ is a complex variable with magnitude ρ and phase ψ and the amplitude $U(z)$ is in general complex. Then the input

$$u(k) = U(z)\,\rho^k\, e^{j\,\psi\,k} = U(z)\,\rho^k\,(\,\cos\psi k + j\sin\psi k\,) \tag{5.2}$$

is itself complex and represent a broad class of input functions of engineering interest including growing and decaying real exponential waveforms as well as growing and decaying sinusoidal waveforms.

5.2 single-input single-output systems

Consider a system, described by a single-input single-output differential equation

$$\begin{aligned} &a_n\, y(k+n) + a_{n-1}\, y(k+n-1) + \ldots + a_0\, y(k) \\ &\qquad = b_m\, u(k+m) + b_{m-1}\, u(k+m-1) + \ldots + b_0\, u(k) \end{aligned} \tag{5.3}$$

where $m \leq n$ and all coefficients are real constants. For an exponential input signal of the form

$$u(k) = U(z)z^k \tag{5.4}$$

we may assume that the particular solution $y_p(k)$ is also exponential in form, that is,

$$y_p(k) = Y(z)z^k \tag{5.5}$$

where $y(z)$ is a complex amplitude to be determined. Substitution of (5.4), (5.5) in (5.3) gives

$$\begin{aligned} &a_n\, Y(z)z^{k+n} + a_{n-1}\, Y(z)z^{k+n-1} + \ldots + a_0\, Y(z)z^k \\ &\quad = b_m\, U(z)z^{k+m} + b_{m-1}\, U(z)z^{k+m-1} + \ldots + b_0\, U(z)z^k \end{aligned} \tag{5.6}$$

or

$$\begin{aligned} &\left(a_n\, z^n + a_{n-1}\, z^{n-1} + \ldots + a_0\right) Y(z)z^k \\ &\quad = (b_m\, z^m + b_{m-1}\, z^{m-1} + \ldots + b_0)\, U(z)z^k \end{aligned} \tag{5.7}$$

The transfer function $H(z)$ is defined to be the ratio of the response amplitude $Y(z)$ to the input amplitude $U(z)$ and is

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_m\, z^m + b_{m-1}\, z^{m-1} + \ldots + b_0}{a_n\, z^n + a_{n-1}\, z^{n-1} + \ldots + a_0} \tag{5.8}$$

The transfer function Eq. (5.8) is an algebraic rational function of the variable z.

Example 18 (Transfer function of discrete-time system)
Consider a discrete-time system described by the third order difference equation

$$\begin{aligned} &100\, y(k+3) - 180\, y(k+2) + 121\, y(k+1) - 41\, y(k) \\ &\quad = 100\, u(k+3) - 10\, u(k+2) + 48\, u(k+1) - 34\, u(k) \end{aligned}$$

The transfer function is now given by

$$H(z) = \frac{100\, z^3 - 10\, z^2 + 48\, z - 34}{100\, z^3 - 180\, z^2 + 121\, z - 41} \tag{5.9}$$

5.3 relationship to the transfer function

The use of $Z\{\}$ as the difference operator in chapter 2 and z as the exponent in the exponential input function creates a similarity in appearance. When consideration is considered to linear time-invariant systems, these two system representations are frequently used interchangeably. The similarity results directly from the difference operator relationship for an exponential waveform

$$Z^n\{U(z)z^k\} \equiv U(z)\, z^{k+n} \tag{5.10}$$

5.4 System poles and zeros

It is often convenient to factor the polynomials in the numerator and denominator of Eq. (5.8) and to write the transfer function in terms of those factors:

$$H(z) = \frac{N(z)}{D(z)} = K\,\frac{(z-w_1)(z-w_2)\ldots(z-w_m)}{(z-p_1)(z-p_2)\ldots(z-p_n)} \tag{5.11}$$

where the numerator and denominator polynomials, $N(z)$ and $D(z)$, have real coefficients defined by the systems's difference equation and $K = b_m/a_n$. As written in Eq. (5.11), the w_i's are the roots of the equation

$$N(z) = 0$$

and are defined to be the system *zeros*, and the p_i's are the roots of the equation

$$D(z) = 0$$

and are defined to be the system *poles*. Note that

$$\lim_{z\to w_i} H(z) = 0 \qquad \text{and} \qquad \lim_{z\to p_i} H(z) = \infty$$

Example 19 (Second-order exponential smoother) *Consider difference equation of the second-order exponential smoother as given in example 12.*

$$y(k+2) - a_1 y(k+1) - a_0 y(k) = b_2 u(k+2) + b_1(k+1), \qquad k \in \mathbb{Z}$$

The corresponding transfer function is found using Eq. (5.8):

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_2\,z^2 + b_1\,z}{z^2 + a_1\,z + a_0}$$

The zeros are given by the roots of

$$N(z) = b_2\,z^2 + b_1\,z = 0$$

so for $b_2 \neq 0$ *we find*

$$w_1 = 0 \qquad w_2 = -b_1/b_2$$

and for $b_2 = 0$ *we only have one zero* $w_1 = 0$*. The poles are given by the roots of*

$$D(z) = z^2 + a_1\,z + a_0 = 0$$

so we find

$$p_1 = -a_1/2 + \sqrt{a_1^2/4 - a_0} \qquad p_2 = -a_1/2 - \sqrt{a_1^2/4 - a_0}$$

For the setting $a_0 = 0.64$*,* $a_1 = -1.6$*,* $b_1 = -0.5$ *and* $b_2 = 1$ *we find:*

$$w_1 = 0 \quad , \quad w_2 = 0.5 \quad , \quad p_1 = p2 = 0.8$$

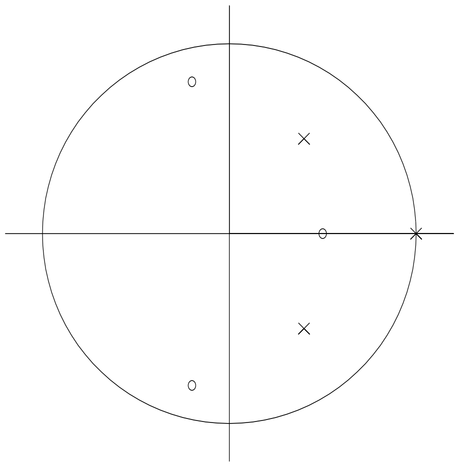

Figure 5.1: Pole-zero plot of a third order discrete-time system

Example 20 (Poles and zeros of third-order system)
Consider the third-order system in example 18, with transfer function

$$H(z) = \frac{100\,z^3 - 10\,z^2 + 48\,z - 34}{100\,z^3 - 180\,z^2 + 121\,z - 41} \tag{5.12}$$

The poles of the system are finding the roots of the equation

$$D(z) = 100\,z^3 - 180\,z^2 + 121\,z - 41 = 100(z-1)(z-0.4-j\,0.5)(z-0.4+j\,0.5) = 0$$

We find that the poles are equal to $p_1 = 1$*,* $p_2 = 0.4 + j\,0.5$ *and* $p_3 = 0.4 - j\,0.5$*. The zeros of the system are finding the roots of the equation*

$$N(z) = 100\,z^3 - 10\,z^2 + 48\,z - 34 = 100(z-0.5)(z+0.2-j\,0.8)(z+0.2+j\,0.8) = 0$$

We find that the zeros are equal to $p_1 = 0.5$*,* $p_2 = -0.2 + j\,0.8$ *and* $p_3 = -0.2 - j\,0.8$*.*
The pole-zero plot of this system is given in Fig. 5.1

5.4.1 System poles and the homogeneous response

Also in the discrete-time case there holds

> The transfer function poles are the roots of the characteristic equation and also the eigenvalues of the system A matrix (as discussed in the previous chapter).

So let the transfer function denominator polynomial be given by

$$D(z) = a_n\, z^n + a_{n-1}\, z^{n-1} + \ldots + a_0$$

and let the roots of $D(z) = 0$ be given by p_i, $i = 1, \ldots, n$. Then the homogeneous response of the system is given by

$$y_h(k) = \sum_{i=1}^{n} C_i\, p_i^k$$

The location of the poles in the z-plane therefore determines the n components in the homogeneous response as follows:

1. A real pole inside the unit circle ($|p| < 1$) defines an exponentially decaying component $C\, p^k$ in the homogeneous response. The rate is determined by the pole location; poles further away from the origin correspond to components that decay rapidly; poles close to the origin correspond to slowly decaying components. Negative real poles alternate in sign.

2. A pole in $p_i = 1$ defines a component that is constant in amplitude and defined by the initial conditions.

3. A real pole outside the unit circle ($|p| > 1$) corresponds to an exponentially increasing component $C\, p^k$ in the homogeneous response, thus defining the system to be unstable.

4. A complex conjugate pole pair $p = \rho\, e^{\pm j\,\psi}$ inside the unit circle combine to generate a response component that is decaying sinusoid of the form $A\, \rho^k\, \sin(\psi\, k + \phi)$ where A and ϕ are determined by the initial conditions. The rate of decay is specified by ρ; The frequency of oscillation is determined by ψ.

5. A complex conjugate pole pair on the unit circle $p = e^{\pm j\,\psi}$ generates an oscillatory component of the form $A\, \sin(\psi\, k + \phi)$ where A and ϕ are determined by the initial conditions.

6. A complex pole pair outside the unit circle generates a exponentially increasing oscillatory component.

5.4.2 System stability

A n-th order linear discrete-time system is asymptotically stable only if all components in the homogeneous response from a finite set of initial conditions decay to zero as time increases, or

$$\lim_{k\to\infty} \sum_{i=1}^{n} C_i p_i^k = 0$$

where p_i are the system poles. If any pole has a magnitude larger than one, there is a component in the output without bound, causing the system to be unstable.

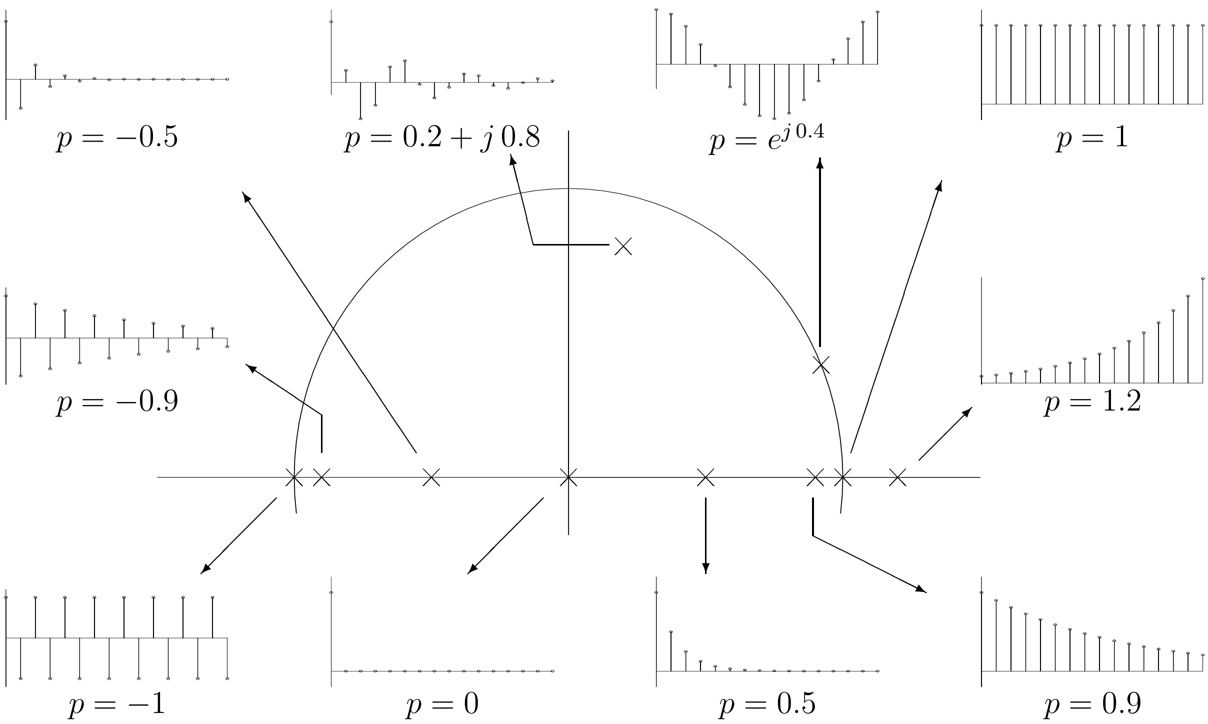

Figure 5.2: The specification of the form of components of the homogeneous response from the system pole locations on the pole-zero plot

In order for a linear discrete-time system to be stable, all its poles must have a magnitude strictly smaller than one, that is they must all lie inside the unit circle. An "unstable" pole, lying outside the unit circle, generates a component in the system homogeneous response that increases without bound from any finite initial conditions. A system having one or more poles lying on the unit circle has nondecaying oscillatory components in the homogeneous response and is defined to be *marginally* stable.

5.4.3 State space formulated systems

Consider the n-th order linear system describe by the set of n state equations and one output equation:

$$x(k+1) = Ax(k) + Bu(k) \tag{5.13}$$

$$y(k) = Cx(k) + Du(k) \tag{5.14}$$

The transfer function is given by

$$\begin{aligned} H(z) &= \frac{Y(z)}{U(z)} \\ &= [C\,(z\,I - A)^{-1}\,B + D] \\ &= \frac{C\,\text{adj}(z\,I - A)^{-1}\,B + D\,\det(z\,I - A)}{\det(z\,I - A)} \end{aligned}$$

Example 21 (Transfer function of a third-order system)
Consider the third-order system in example 18. The state-space representation is found using the results of section 2.2:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.41 & -1.21 & 1.8 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0.01 \end{bmatrix}$$

$$C = \begin{bmatrix} 7 & -73 & 170 \end{bmatrix} \qquad D = 1$$

We compute the adjoint of $(z\,I - A)$:

$$adj(z\,I - A) = \begin{bmatrix} z^2 - 1.8\,z + 1.21 & 0.41 & 0.41\,z \\ z - 1.8 & z^2 - 1.8\,z & -1.21\,z + 0.41 \\ 1 & z & z^2 \end{bmatrix}$$

and the determinant of $(z\,I - A)$:

$$det(z\,I - A) = z^3 - 1.8\,z^2 + 1.21\,z - 0.41$$

and so:

$$\begin{aligned} H(z) &= \frac{C\,adj(z\,I - A)^{-1}\,B + D\,det(z\,I - A)}{det(z\,I - A)} \\ &= \frac{z^3 - 0.1\,z^2 + 0.48\,z - 0.34}{z^3 - 1.80\,z^2 + 1.21\,z - 0.41} \end{aligned}$$

Bibliography

[1] H. Freeman. *Discrete-time systems.* John Wiley & Sons, New York, USA, 1965.

[2] H. Kwakernaak and R. Sivan. *Modern Signals and Systems.* Prentice Hall, New Jersey, USA, 1991.

[3] C. Phillips and J.M. Parr. *Signals, Systems, and Tranforms.* Prentice Hall, New Jersey, USA, 1999.

[4] D. Rowell and D.N. Wormley. *System Dynamics, an introduction.* Prentice Hall, New Jersey, USA, 1997.