university of groningen

faculty of science and engineering

Control theory for discrete-time systems

Internship MSc Applied Mathematics

January - March 2019

Student: J.J. Koning, S2406829

Company/Institute: Schut Geometrical Metrology, Groningen

Internal UoG supervisor/first assessor: dr. ir. B. Besselink

External supervisor: G. Schut, MSc

Internal UoG second assessor: prof. dr. M.K. Camlibel

Abstract

In this report the results of my internship at the company Schut Geometrical Metrology are documented. We investigate how a discrete-time representation of a process can be obtained and how we can estimate future process outputs from such a representation. For this we use a Kalman filter. We then use the obtained estimations to control the process using a PID controller.

Contents

1 Introduction

1.1 The company

Schut Geometrical Metrology (SGM) is an international organisation that is specialized in the development and production of precision measuring apparatus. SGM has five offices throughout Europe. My internship took place at the office in Groningen, consisting of approximately 45 employees. The office consists of the following departments:

- Sales Department,
- Accounting Department,
- Human Resources Department (HR),
- Storage, Packaging and Distribution Department,
- Electronics and Hardware Department,
- Testing and Product Support team,
- Software Development Department.

In the past 11 weeks I have been working at the software development department.

1.2 Control theory for discrete-time systems

The aim of my internship was to improve the control inputs for directing the measuring systems. For this aspect of the research we mainly focused on finding a precise estimation of the current and future states of the machine. Furthermore, we aimed to realize useful simulations of a measuring instrument (hypothetically), to get more insight in the precise effect of control inputs that we intend to apply to the machine. For this, it is of interest to consider discretizations that take place when the computed input is sent to the processor, and when the measurements are obtained (for each, say, 0.1 µm that the machine has traveled, the processor receives a pulse). The latter purpose is especially useful if one wants to try a new method without applying it directly to an actual machine, which is of interest for research that is done at SGM.

In general, a system can be used to give a description of the dynamics of a process, allowing one to gain insight in the expected behavior of the process. The processes that take place in the measuring systems at SGM operate in discrete time, which is why we are interested in a discrete-time description of the dynamics. In this report we elaborate on how a process can be represented by a discrete-time system. We give an overview of the theory that is involved in defining such a system and how a discrete-time system can be transformed into a continuous-time representation and vice versa. To learn more about system identification for discrete-time systems, one can consult e.g. [6]. Once we have established this theory, we investigate how a process can be controlled using feedback. For this we consider a PID controller.

In order for the machine to run more smoothly, we are interested in predicting the output of the processes as accurately as possible. For example, if there are delays in the measurements, we are interested in estimating the unknown current process output.

Furthermore, it is sometimes beneficial to find a prediction of future states to base our control input on. To obtain a good prediction, we use a so-called Kalman filter. A Kalman filter is a filter that uses a model of the process to estimate the current or future output of the process. Generally, a Kalman filter is used to overcome noise in the measurement data as well, see [7]. However, in this report this aspect of the Kalman filter is left out, and can be seen as a possible extension for further research.

Throughout the internship, simulations of the results for several example processes have been performed using Mathematica. For one such process, the plots that were found are presented in the report.

2 Discrete-time systems

A discrete-time system can be viewed as a representation of a certain process, where each time step in the system represents a specified amount of time in the process, called the *sampling period* T. Throughout this work we will refer to the actual physical behavior of a machine or apparatus as the *process*, and we refer to a mathematical description of this behavior as a discrete- or continuous-time *system*.

From an engineering perspective, a discrete-time system represents a process by simply mapping a discrete-time signal called the input, to a discrete-time signal called the output. As they are discrete-time signals, both the input and the output are represented by sequences $x[n] = x(nT)$ and $y[n] \approx y(nT)$ respectively, where $x(t)$ and $y(t)$ are the real-time input and output signals of the process. Given the first few entries of the output, i.e. the *initial conditions*, the system maps a given input sequence to its corresponding output sequence, which gives an approximation of the process output.

A system is said to be *linear* if for any two constants $\alpha_1, \alpha_2 \in \mathbb{R}$ and any two input sequences $x_1[n], x_2[n]$, the output corresponding to the input $x[n] = \alpha_1 x_1[n] + \alpha_2 x_2[n]$ with initial conditions zero is given by $y[n] = \alpha_1 y_1[n] + \alpha_2 y_2[n]$, where $y_1[n]$ and $y_2[n]$ are the outputs corresponding to the inputs $x_1[n]$ and $x_2[n]$ respectively. The system is said to be *shift invariant* if for any input sequence $x[n]$ with corresponding output $y[n]$ and any constant $n_0 \in \mathbb{Z}$, the output corresponding to the input sequence $x[n-n_0]$ is given by the sequence $y[n-n_0]$. We say that the system is *causal* if for any constant $n_0 \in \mathbb{Z}$, the output at time n_0 depends only on the input for $n \leq n_0$.

In the continuous-time case, linear systems are often described using linear differential equations. The discrete-time equivalent of a differential equation is the so-called *difference equation*. Generally, a discrete-time system described by a difference equation can be written as

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{r=0}^{M} b_r x[n-r], \tag{1}$$

where $a_0, a_1, \ldots, a_N$ and $b_0, b_1, \ldots, b_M$ are constant coefficients with $a_0 \neq 0$. Notice that $a_0 \neq 0$ ensures that the system is causal. In the following we mainly focus on systems described by difference equations.

In the next section we will see how a discrete-time representation for a given process can be obtained. For this we use an important characteristic of the process, called the *impulse response*.

2.1 Discrete time and continuous time

An important characteristic of a process is its response to an impulse. For continuous-time systems, this is interpreted as the response to the so-called *impulse function* $\delta(t)$, which is defined by

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

constrained to $\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 0$. The response of the process to this input function is called the *impulse response* of the process, and it is denoted by $h(t)$. For processes described by linear systems, the impulse response allows one to compute the output of the process corresponding to any input function $x(t)$ via the following formula (see Section 3.1 in [5]):

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\mathrm{d}\tau. \tag{3}$$

A discrete-time representation of the process can be defined by requiring that the response of the discrete-time system to the so-called *unit impulse sequence* approximates the impulse response of the process. The unit impulse sequence is defined by

$$\delta[n] = \begin{cases} 1 & \text{if } n = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The output $y[n]$ that results from applying the unit impulse sequence is referred to as the *impulse response of the system*, and we denote it by $h[n]$. In Chapter 5 we will discuss a means to find $h[n]$ for a given impulse response $h(t)$, such that $h[n]$ represents the impulse response of a discrete-time system.

Just as in the continuous-time case, the impulse response of a linear discrete-time system can be used to compute the response to any input sequence $x[n]$. To see this, notice that we can write $x[n]$ as the sum of impulses $x[n] = \sum_{n=-\infty}^{\infty} x[k]\delta[n-k]$. Since the response to $\delta[n-k]$ is given by $h[n-k]$, it follows from linearity of the system that the response to $x[n]$ can be written as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]. \tag{5}$$

In the next section we try to find a relation between the process output and the output of the system.

2.2 Relation between $h(t)$ and $h[n]$

As mentioned, for a given process with impulse response $h(t)$, one is interested in finding a discrete-time system for which the impulse response $h[n]$ gives a representation of $h(t)$. The actual requirement is that $h[n] \approx Th(nT)$, where T is the sampling period. The scaling by T has a valid reason. To understand this, notice that $h[n]$ can be interpreted in two ways. On the one hand, we want it to be a representation of the impulse response of the process $h(t)$. On the other hand, $h[n]$ is simply the output of the discrete-time system to the sequence $\delta[n]$, which can be interpreted as a discrete representation of the input function

$$u_T(t) = \begin{cases} 1 & \text{if } 0 \leq t < T, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

According to (3), the output corresponding to this input function, which should be approximated by $h[n]$, is given by $y_T(t) = \int_0^T h(t-\tau)\mathrm{d}\tau$. Notice that the integral can be approximated by $Th(t)$ if T is small enough. Hence $h[n] \approx y_T(nT)$ is achieved by requiring $h[n] \approx Th(nT)$. Figure 1 shows the impulse response of a certain process together with the impulse response of a discretization with sampling period $T = \frac{1}{2}$ that is based on the requirement $h[n] \approx Th(nT)$ (the precise derivation of $h[n]$ from $h(t)$ will be dealt with in Chapter 7). The process output $y_T(t)$ is also presented. From this plot it becomes clear that $h[n]$ is indeed close to $y_T(t)$, but that it was intended to approximate $Th(t)$.

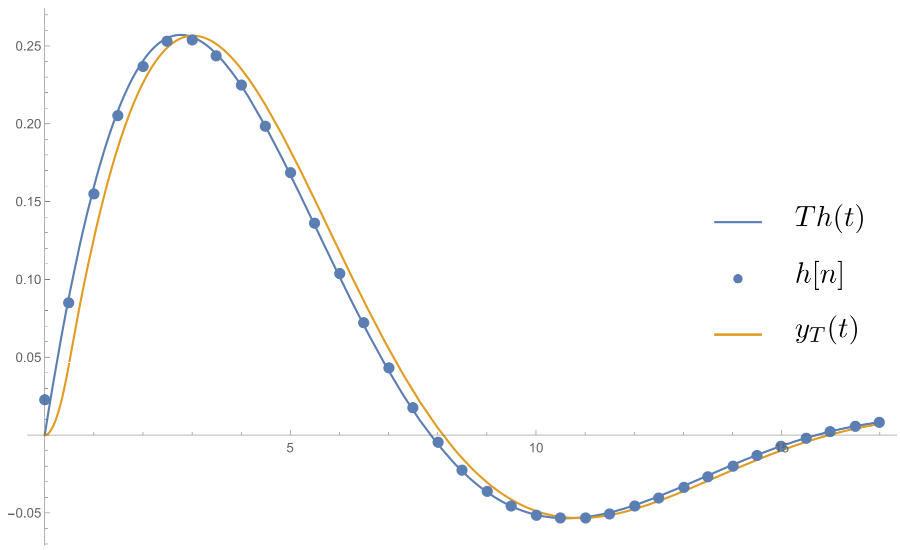


Figure 1: Impulse Response

2.3 Relation between $y(t)$ and $y[n]$

Now consider the situation where we apply an input $x(t)$ to a given process, for which a discrete-time representation is given. The output of the discrete-time system is then given by (5). Substituting $x[k] = x(kT)$ and $h[n] \approx Th(nT)$ into this expression yields

$$y[n] \approx \sum_{k=-\infty}^{\infty} Tx(kT)h((n-k)T). \tag{7}$$

Hence, $y[n]$ basically gives an approximation of the response of the process to an impulse at each time step k with weight $Tx(kT)$. In other words, $y[n]$ gives an approximation of the response of the process to the *sampled representation* of the input function $x(t)$, given by

$$x_q(t) := \sum_{k=-\infty}^{\infty} Tx(kT)\delta(t-kT). \tag{8}$$

To gain more insight into this fact, we compare the response of a process to the *step function* with its discrete-time representation in the following example.

Example 2.1 (Step response)**.** In this example we investigate how the *step response* of a process is related to the step response of a discrete-time representation. The step response of a process and of a discrete-time system, denoted by $g(t)$ and $g[n]$, are defined by the responses to the inputs

$$u(t) := \begin{cases} 1 & \text{if } t \geq 0, \\ 0 & \text{otherwise,} \end{cases} \qquad u[n] := \begin{cases} 1 & \text{if } n \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

respectively. Note that the sampled representation of $u(t)$ is given by $u_q(t) = \sum_{k=0}^{\infty} T\delta(t-kT)$. According to the preceding discussion, $g[n]$ gives an approximation to the corresponding process output $g_q(t) := \sum_{k=0}^{\infty} Th(t-kT)$. This is illustrated in Figure 2, which shows a plot of the response $g_q(t)$ of the process for $T = \frac{1}{2}$, together with the step response $g[n]$ and the actual step response of the process. Here we see that $g[n]$ indeed gives a discrete-time representation of $g_q(t)$, which in turn is an approximation of the step response $g(t)$.

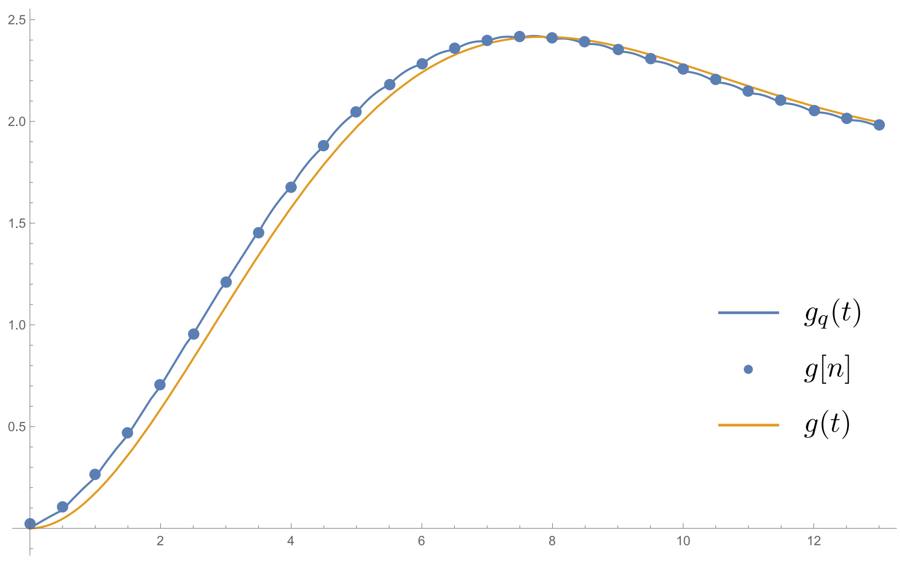


Figure 2: Step Response

3 BIBO Stability

In this chapter we discuss a form of stability for discrete-time systems, namely *bounded input bounded output* (BIBO) stability. BIBO stability basically means that the output of the process will be bounded whenever the applied input sequence is bounded.

Recall that a sequence $x[n]$ is said to be *bounded* if there exists a constant $M \in \mathbb{R}, M \geq 0$ such that $|x[n]| \leq M$ for all $n \in \mathbb{Z}$. Then stability of discrete-time systems can be defined as follows.

Definition 3.1 (BIBO stability)**.** A discrete-time system is said to be *bounded-input bounded-output stable* (BIBO stable) if its output $y[n]$ is bounded for any bounded input $x[n]$.

The following theorem gives a necessary and sufficient condition for BIBO stability on the impulse response of the system.

Theorem 3.1. *A discrete-time system is BIBO stable if and only if its impulse response h is absolutely summable, i.e.,* $S_h := \sum_{n=-\infty}^{\infty} |h[n]| < \infty$.

Proof. Suppose that the system is BIBO stable. Consider the input sequence

$$x[n] = \begin{cases} -1 & \text{if } h[-n] < 0, \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

Notice that this implies $x[n]h[-n] = |h[-n]|$ for all $n \in \mathbb{Z}$. Since $x[n]$ is a bounded sequence, the corresponding output sequence $y[n]$ is bounded as well, i.e. there exists a constant $M \in \mathbb{R}$ such that $|y[n]| \leq M$ for all $n \in \mathbb{Z}$. Hence

$$M \geq |y[0]| = \Big| \sum_{k=-\infty}^{\infty} x[k]h[-k] \Big| = \Big| \sum_{k=-\infty}^{\infty} |h[-k]| \Big| = \sum_{k=-\infty}^{\infty} |h[k]|, \tag{11}$$

which shows that h is absolutely summable.

Conversely, suppose that h is absolutely summable and let $x[n]$ be any bounded sequence with bound M. Then, by the triangle inequality,

$$|y[n]| = \Big| \sum_{k=-\infty}^{\infty} x[k]h[n-k] \Big| \leq \sum_{k=-\infty}^{\infty} |x[k]||h[n-k]| \leq M \sum_{k=-\infty}^{\infty} |h[n-k]| \leq MS_h, \tag{12}$$

which shows that $y[n]$ is bounded. This completes the proof. □

In practice, the condition $S_h < \infty$ from the above theorem is not easily verified and hence does not seem very useful. However, we will use it to find more insightful conditions in the last section of the next chapter.

4 The z-domain

It is often useful to convert systems from a time-domain representation into a frequency-domain representation. In the continuous-time case this can be done using the *Laplace transform*, transforming a system in the time domain into the frequency domain, also referred to as the s domain. For discrete-time systems, one can apply the so-called *$\mathcal{Z}$-transform*, which can be viewed as the discrete-time equivalent of the Laplace transform. The $\mathcal{Z}$-transform transforms a sequence from the step domain into the corresponding frequency domain, which is referred to as the z domain.

4.1 The $\mathcal{Z}$-transform

In this section we provide a definition for the $\mathcal{Z}$-transform, we discuss its domain (also called the *region of convergence* (ROC)), give some properties such as linearity, and give an expression for the inverse $\mathcal{Z}$-transform. In the next section we apply the $\mathcal{Z}$-transform to discrete-time systems.

Definition 4.1 ($\mathcal{Z}$-transform)**.** The *$\mathcal{Z}$-transform* of a sequence $x[n]$ is defined by

$$X(z) = \mathcal{Z}\{x[n]\} := \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}. \tag{13}$$

The *unilateral $\mathcal{Z}$-transform* of a sequence is defined as

$$\mathcal{UZ}\{x[n]\} := \mathcal{Z}\{x[n]u[n]\} = \sum_{n=-\infty}^{\infty} x[n]u[n]z^{-n} = \sum_{n=0}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}, \tag{14}$$

with $u[n]$ defined in (9).

The $\mathcal{Z}$-transform is defined at points $z \in \mathbb{C}$ where the infinite series (13) converges. The set of these points is called the *region of convergence* (ROC). The well known *root test* states that a series $\sum_{n=1}^{\infty} a_n$ converges if $\limsup_{n\to\infty} \sqrt[n]{|a_n|} < 1$. To apply the root test, notice that the series in (13) can be written as

$$\mathcal{Z}\{x[n]\} = x[0] + \sum_{n=1}^{\infty} x[n]z^{-n} + \sum_{m=1}^{\infty} x[-m]z^{m}. \tag{15}$$

By the root test, a sufficient condition for convergence of the above expression is given by

$$\limsup_{n\to\infty} \sqrt[n]{|x[n]z^{-n}|} < 1, \ \limsup_{n\to\infty} \sqrt[n]{|x[-n]z^{n}|} < 1 \quad \Longleftrightarrow \quad r < |z| < R, \tag{16}$$

where $r = \limsup_{n\to\infty} \sqrt[n]{|x[n]|}$ and $R = 1/\left(\limsup_{n\to\infty} \sqrt[n]{|x[-n]|}\right)$. If $\limsup_{n\to\infty} \sqrt[n]{|x[-n]|} = 0$, this condition simplifies to $|z| > r$, in which case the region of convergence contains the set $\{z \in \mathbb{C} \mid |z| > r\}$. Otherwise, the region of convergence contains the annulus $\mathcal{A} = \{z \in \mathbb{C} \mid r < |z| < R\}$.

The following theorem provides some properties of the $\mathcal{Z}$-transform. These properties can easily be verified by directly applying the definition (13).

Theorem 4.1 (Properties of the $\mathcal{Z}$-transform)**.** *Given arbitrary sequences $x[n], x_1[n]$ and $x_2[n]$ with $\mathcal{Z}$-transforms $X(z), X_1(z)$ and $X_2(z)$ and ROC's given by sets $\mathcal{R}, \mathcal{R}_1$ and $\mathcal{R}_2$ respectively, the $\mathcal{Z}$-transform satisfies the equalities stated below. For each equality a sufficient condition on z for convergence of the corresponding $\mathcal{Z}$-transform is given.*

1. $\mathcal{Z}\{\alpha_1 x_1[n] + \alpha_2 x_2[n]\} = \alpha_1 X_1(z) + \alpha_2 X_2(z)$, *for any* $\alpha_1, \alpha_2 \in \mathbb{R}$, $z \in \mathcal{R}_1 \cap \mathcal{R}_2$,
2. $\mathcal{Z}\{x[n-n_0]\} = z^{-n_0} X(z)$, *for any* $n_0 \in \mathbb{Z}$, $z \in \mathcal{R}$,
3. $\mathcal{Z}\{a^n x[n]\} = X(z/a)$, *for any* $a \in \mathbb{R}$, $z \in a\mathcal{R}$,
4. $\mathcal{Z}\{n x[n]\} = -z \dfrac{\mathrm{d}}{\mathrm{d}z} X(z)$, $z \in \mathcal{R}$,

where $a\mathcal{R} = \{z \in \mathbb{C} \mid z/a \in \mathcal{R}\}$.

To transform a function from the frequency domain to a sequence, the following theorem provides an expression for the *inverse $\mathcal{Z}$-transform.* This transformation can be used to retrieve a sequence in case only its $\mathcal{Z}$-transform is known.

Theorem 4.2 (Inverse $\mathcal{Z}$-transform)**.** *Let $X(z)$ be the $\mathcal{Z}$-transform of the sequence $x[n]$ defined in the region $|z| > R$. Then $x[n]$ can be retrieved from $X(z)$ by the inverse $\mathcal{Z}$-transform*

$$x[n] = \mathcal{Z}^{-1}\{X(z)\} = \frac{1}{2\,\pi\,i} \oint_C X(z)\, z^{n-1}\, d\,z, \tag{17}$$

where C is any positively (counterclockwise) oriented curve in the complex number plane that lies in the region $|z| > R$ and winds around the origin.

The most common way to find the inverse $\mathcal{Z}$-transform of a given function is to use a table of $\mathcal{Z}$-transform pairs. An example of such a table is given in Appendix A. Note that the $\mathcal{Z}$-transform of each of the sequences in this table is given by a rational function of z. To use such a table, one needs to write the function in a suitable form, for example as a linear combination of functions that appear in the table, and use the properties given by Theorem 4.1. An important class of functions for which the inverse $\mathcal{Z}$-transform can be computed in this way is the class of *rational functions.* In Mathematica, the inverse $\mathcal{Z}$-transform of certain functions can be computed using the command `InverseZTransform`. The scope of this command includes rational functions.

4.2 Discrete-time systems in the z-domain

In the z-domain, the behaviour of a given discrete-time system can be characterized using its so-called *system function* $H(z)$, which is defined as the $\mathcal{Z}$-transform of the impulse response. The system function can be viewed as the discrete-time equivalent of the transfer function. Similar to the transfer function, the system function relates any possible input to the output of the system in the frequency domain. For linear systems, this relation can be written as $Y(z) = H(z)X(z)$, where $X(z)$ and $Y(z)$ denote the

$\mathcal{Z}$-transforms of the input and output sequences respectively. Hence, if an expression for the output is given in terms of the input, the system function can be computed as

$$H(z) = Y(z)/X(z). \tag{18}$$

To find a representation of a system of the form (1) in the z-domain, we simply take the $\mathcal{Z}$-transform on both sides of the equation. Using linearity and the translation property stated in Theorem 4.1, we obtain

$$\sum_{k=0}^{N} a_k z^{-k} Y(z) = \sum_{r=0}^{M} b_r z^{-r} X(z). \tag{19}$$

This equality can be solved for $Y(z)$ as

$$Y(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{\sum_{k=0}^{N} a_k z^{-k}} X(z). \tag{20}$$

From this expression it becomes clear that the system function is given by

$$H(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{\sum_{k=0}^{N} a_k z^{-k}}. \tag{21}$$

Recall that for causal systems we have $N \geq M$, and hence $H(z)$ is a proper rational function if and only if the system is causal.

In case the system function $H(z)$ is a rational function, as in the above case, the poles and roots of $H(z)$ are referred to as the *system poles* and *system roots*. Their continuous-time equivalent are the poles and roots of the transfer function of the system. In the next section we will see that the poles of a system play an important role in determining stability of the system.

Example 4.1. To illustrate the usefulness of a representation of the system in the z-domain, consider the following situation. Suppose that we want the output of a given system to track a certain reference signal $\hat{y}[n]$, that is, we want to find an input sequence $\hat{x}[n]$ such that the output satisfies $y[n] = \hat{y}[n]$ for all n. By definition of the system function, this is the case if and only if $\hat{Y}(z) = H(z)\hat{X}(z)$, where $\hat{X}(z)$ and $\hat{Y}(z)$ denote the $\mathcal{Z}$-transforms of $\hat{x}[n]$ and $\hat{y}[n]$ respectively. This yields $\hat{X}(z) = H^{-1}(z)\hat{Y}(z)$. Hence, from the representation of the system in the z-domain the input sequence that does the job is seen to be

$$\hat{x}[n] = \mathcal{Z}^{-1}\{H^{-1}(z)\hat{Y}(z)\}. \tag{22}$$

In fact, because of the relation $Y(z) = H(z)X(z)$, one can easily "invert" a system by simply taking the inverse of the system function.

4.3 Stability in the z-domain

In this section we try to find conditions for stability in the z-domain. We do this for causal systems that are described by a difference equation of the form (1). In the previous chapter, we saw that the system function of such systems is a proper rational function. In this section we derive conditions for BIBO stability in terms of the *poles* of the system function. We will see that a system is BIBO stable if and only if all its poles have magnitude smaller than 1. In general, poles that satisfy this property are referred to as *stable poles*.

First, let us write $H(z)$ in terms of its poles for the special case where all poles have multiplicity one. Since $H(z)$ is a proper rational function, its partial fraction expansion is given by

$$H(z) = A_0 + \sum_{k=1}^{N} \frac{A_k}{z - p_k}, \tag{23}$$

where $A_0, A_1, \ldots, A_N \in \mathbb{C}$ are nonzero constant coefficients and $\{p_1, \ldots, p_N\}$ are the poles of $H(z)$. In the general case, $H(z)$ has $r \leq N$ distinct poles $\{p_1, \ldots, p_r\}$ with multiplicities $\{n_1, \ldots, n_r\}$ respectively. In that case, the summation in (23) runs up to $k = r$, and for each $k \in \{1, \ldots, r\}$, we replace the term $\frac{A_k}{z-p_k}$ by $\sum_{m=1}^{n_k} \frac{A_{k,m}}{(z-p_k)^m}$ for some $A_{k,1}, A_{k,2}, \ldots, A_{k,n_k} \in \mathbb{C}$. Note that $A_{k,n_k} \neq 0$, since p_k has multiplicity n_k. Thus, the partial fraction expansion of $H(z)$ is given by

$$H(z) = A_0 + \sum_{k=1}^{r} \sum_{m=1}^{n_k} \frac{A_{k,m}}{(z - p_k)^m}. \tag{24}$$

Before we move on to the main result, we deduce an expression for the impulse response in terms of the poles of the system function as well, using the above equation. Recall that the system function is the $\mathcal{Z}$-transform of the impulse response. Hence, taking the inverse $\mathcal{Z}$-transform of (24) and using the fact that the $\mathcal{Z}$-transform is linear, we obtain

$$h[n] = \mathcal{Z}^{-1}\{A_0 + \sum_{k=1}^{r} \sum_{m=1}^{n_k} \frac{A_{k,m}}{(z - p_k)^m}\} = A_0\delta[n] + \sum_{k=1}^{r} \sum_{m=1}^{n_k} \mathcal{Z}^{-1}\{\frac{A_{k,m}}{(z - p_k)^m}\}. \tag{25}$$

The inverse $\mathcal{Z}$-transform of $\frac{1}{(z-p_k)^m}$ is given by

$$\mathcal{Z}^{-1}\Big\{\frac{1}{(z - p_k)^m}\Big\} = \frac{\Pi_{r=1}^{m-1}(n - r)}{m!} p_k^{n-m} u[n-1]. \tag{26}$$

Substituting this into (25) yields

$$h[n] = A_0\delta[n] + \sum_{k=1}^{r} \sum_{m=1}^{n_k} A_{k,m} \frac{\Pi_{r=1}^{m-1}(n - r)}{m!} p_k^{n-m} u[n-1]. \tag{27}$$

From this expression it can already be seen that if one of the poles of the system function has magnitude *greater* than 1, then $\lim_{n\to\infty} h[n] = \infty$, which implies that the system is *not* BIBO stable. This observation brings us to the following theorem, which states that BIBO stability is equivalent to stability of all system poles.

Theorem 4.3. *Consider a system with rational system function $H(z)$. The system is BIBO stable if and only if all poles of $H(z)$ have magnitude smaller than 1.*

Proof. First, suppose that the system is BIBO stable. Suppose, for contradiction, that $H(z)$ has a pole p with magnitude greater than or equal to 1. By Theorem 3.1 we know that S_h is finite. Since p is a pole of $H(z)$, we have $\lim_{z\to p}|H(z)|=\infty$. Hence, there exists a constant $\delta>0$ such that for all $z\in\mathbb{C}$ with $|z-p|<\delta$ we have $|H(z)|>S_h$. This implies that we can find $\bar{z}$ with $|\bar{z}|>|p|$ such that $H(\bar{z})>S_h$. Since $|p|\geq 1$ and $|\bar{z}|>|p|$, we have $|\bar{z}|^{-n}\leq 1$ for all $n\geq 0$. Hence we can write

$$S_h=\sum_{n=-\infty}^{\infty}|h[n]|=\sum_{n=0}^{\infty}|h[n]|\geq\sum_{n=0}^{\infty}|h[n]\bar{z}^{-n}|\geq\left|\sum_{n=0}^{\infty}h[n]\bar{z}^{-n}\right|=|H(\bar{z})|>S_h, \tag{28}$$

which is a contradiction. Hence all poles of $H(z)$ must have magnitude smaller than 1.

To prove the converse, we again use the result from Theorem 3.1. To use this theorem, we first compute the absolute value of $h[n]$ from (27). By the triangle inequality we can write

$$|h[n]|\leq|A_0|\delta[n]+\sum_{k=1}^{r}\sum_{m=1}^{n_k}|A_{k,m}|\left|\frac{\Pi_{r=1}^{m-1}(n-r)}{m!}p_k^{n-m}\right|u[n-1]. \tag{29}$$

Then it follows that

$$\begin{aligned}\sum_{n=-\infty}^{\infty}|h[n]| &\leq |A_0|+\sum_{n=-\infty}^{\infty}\sum_{k=1}^{r}\sum_{m=1}^{n_k}|A_{k,m}|\left|\frac{\Pi_{r=1}^{m-1}(n-r)}{m!}p_k^{n-m}\right|u[n-1]\\ &=\sum_{k=1}^{r}\sum_{m=1}^{n_k}|A_{k,m}|\sum_{n=-\infty}^{\infty}\left|\frac{\Pi_{r=1}^{m-1}(n-r)}{m!}p_k^{n-m}\right|u[n-1]\\ &=\sum_{k=1}^{r}\sum_{m=1}^{n_k}|A_{k,m}|\sum_{n=1}^{\infty}\left|\frac{\Pi_{r=1}^{m-1}(n-r)}{m!}p_k^{n-m}\right|.\end{aligned} \tag{30}$$

From this expression it becomes clear that $h[n]$ is absolutely summable if for all $k\in\{1,\dots,r\}$ and $m\in\{1,\dots,n_k\}$ we have

$$\sum_{n=1}^{\infty}\left|\frac{\Pi_{r=1}^{m-1}(n-r)}{m!}p_k^{n-m}\right|<\infty. \tag{31}$$

By the *ratio test*, this holds if $\lim_{n\to\infty}|\frac{a_{n+1}}{a_n}|<1$, where $a_n=\frac{\Pi_{r=1}^{m-1}(n-r)}{m!}p_k^{n-m}$. We have

$$\lim_{n\to\infty}|\frac{a_{n+1}}{a_n}|=\lim_{n\to\infty}\left|\frac{\Pi_{r=1}^{m-1}(n+1-r)p_k^{n+1-m}}{\Pi_{r=1}^{m-1}(n-r)p_k^{n-m}}\right|=\lim_{n\to\infty}\left|\frac{\Pi_{r=1}^{m-1}(n+1-r)}{\Pi_{r=1}^{m-1}(n-r)}\right||p_k|. \tag{32}$$

As $\Pi_{r=1}^{m-1}(n+1-r)$ and $\Pi_{r=1}^{m-1}(n-r)$ are polynomials of the same degree, both with leading coefficient 1, we have

$$\lim_{n\to\infty}\frac{\Pi_{r=1}^{m-1}(n+1-r)}{\Pi_{r=1}^{m-1}(n-r)}=1. \tag{33}$$

It is well known that the limit of the absolute value of a sequence converges to the absolute value of the limit. In other words, the above equality implies

$$\lim_{n\to\infty} \Big| \frac{\Pi_{r=1}^{m-1}(n+1-r)}{\Pi_{r=1}^{m-1}(n-r)} \Big| = |1| = 1. \tag{34}$$

Substituting into (32) yields $\lim_{n\to\infty} |\frac{a_{n+1}}{a_n}| = |p_k|$, which is smaller than 1 by assumption. Hence, the ratio test tells us that (31) holds for all $k \in \{1, \dots, r\}$ and $m \in \{1, \dots, n_k\}$, which implies that $h[n]$ is absolutely summable. By Theorem 3.1 we conclude that the system is BIBO stable. □

Similar results can be obtained for the continuous-time case. In fact, it is well-known that a continuous-time system is BIBO stable if and only if all its poles (i.e. the poles of the transfer function) have negative real part, see Theorem 3.21 in [5]. Therefore, a pole of a continuous-time system is said to be stable if it has negative real part.

5 Transforming between discrete and continuous time

In this chapter we try to find a means to obtain a discrete-time system of the form (1) to represent a given process. We will also see how a continuous-time representation of the process, e.g. an expression for $h(t)$, can be found from a discrete one. To learn more about system identification, one can consult e.g. [6]. This book also contains theory on Kalman filtering. Furthermore, an interesting survey on transforming between discrete-time and continuous-time representations is given by [1].

In Chapter 4.2 we saw how the system function can be used as a representation of the system. In the following section we investigate the relation between the system function and the transfer function of the process. For this we use the requirement on $h[n]$ and $h(t)$ that we discussed in Section 2.2.

5.1 Relation between the transfer function and the system function

Recall that the transfer function of a process is defined as the Laplace transform of the impulse response, i.e.,

$$\hat{H}(s) = \int_{-\infty}^{\infty} h(t)e^{-st}\mathrm{d}t. \tag{35}$$

To compare the transfer function to the system function, recall that we require the impulse responses of the system and the process to satisfy $h[n] \approx Th(nT)$. Hence, discretizing the above integral we find that $h[n] \approx Th(nT)$ is equivalent to

$$\hat{H}(s) = \sum_{n=-\infty}^{\infty} \int_{nT}^{(n+1)T} h(t)e^{-st}\mathrm{d}t \approx \sum_{n=-\infty}^{\infty} Th(nT)e^{-snT} \approx \sum_{n=-\infty}^{\infty} h[n](e^{sT})^{-n}. \tag{36}$$

Notice that the latter term is equal to the $\mathcal{Z}$-transform of the impulse response $h[n]$ with e^{sT} substituted for z. Since the $\mathcal{Z}$-transform of $h[n]$ is given by $H(z)$, it follows that

$$\hat{H}(s) \approx H(e^{sT}). \tag{37}$$

Hence we have found a relation between the system function and the transfer function of the process that is equivalent to $h[n] \approx Th(nT)$. For a given process, a discrete-time system that represents the process can thus be obtained by substituting $z = e^{sT}$ into the transfer function of the process. This substitution can be seen as a *transform* between the z-domain and the s-domain with inverse $s = \frac{1}{T}\ln(z)$. In the next section we discuss other transforms that can be used to find a system function, which might be desirable over $z = e^{sT}$.

5.2 Transforms between z- and s-domain

Using a transform between the z- and s-domain, one can transform a continuous-time representation of a process into a discrete-time one and vice versa. We have already seen that, if T is small, the transform $z = e^{sT}$ leads to an accurate relation between the system function and the transfer function. A major drawback of this transform however, is that substituting the exponential function $z = e^{sT}$ into the rational function $H(z)$ in general does not yield a function that represents the transfer function of a process. To obtain a useful relation between the system function and the transfer function, we aim to find a *rational* transform $z = f(s)$ that approximates $z = e^{sT}$, which we can substitute into $H(z)$.

A simple approximation to the exponential that comes to mind is $e^{sT} \approx 1 + sT$. The transform $z = 1 + sT$ is often referred to as the *Forward Euler Transform*. Using this transform, a system function can be obtained from $\hat{H}(s)$ as $H(z) := \hat{H}(\frac{1}{T}(z-1)$. Notice that each pole p of $\hat{H}(s)$ is mapped to a pole $1 + pT$ of $H(z)$. Now, recall from Chapter 3 that in the s-domain, poles are stable if they have real part less than 0, and in the z-domain if they have magnitude less than 1. Clearly, the Forward Euler Transform could map a stable pole of $\hat{H}(s)$ to an unstable pole of $H(z)$. Hence, a stable continuous-time representation might be transformed to an unstable discrete-time one, or vice versa.

To find a transform that does preserve stability, consider the following approximation to the exponential function:

$$e^{sT} = \frac{e^{sT/2}}{e^{-sT/2}} \approx \frac{1 + sT/2}{1 - sT/2}. \tag{38}$$

The transform that is based on this approximation is known as the *Bilinear Transform*. It is defined by

$$z = \frac{1 + sT/2}{1 - sT/2}, \qquad s = \frac{2}{T}\frac{z-1}{z+1}. \tag{39}$$

A consequence of the following lemma is that the bilinear transform preserves stability. Notice that the factor $2/T$ can be omitted without loss of generality.

Lemma 5.1. *A complex number z has magnitude less than 1 if and only if the number $s = (z-1)/(z+1)$ has real part less than 0.*

Proof. We can write the relation between s and z as

$$s = \frac{z-1}{z+1} = \frac{\bar{z}+1}{\bar{z}+1} \cdot \frac{z-1}{z+1} = \frac{|z|^2 - 1 + z - \bar{z}}{|z+1|^2}. \tag{40}$$

Since $z - \bar{z}$ is purely imaginary for any $z \in \mathbb{C}$, it follows that $\mathrm{Re}(s) = \frac{|z|^2-1}{|z+1|^2}$, which is less than zero if and only if $|z|< 1$. $\square$

The bilinear transform can be used to transform a continuous-time representation of a process into a discrete-time one while preserving stability. Mind that the transform can also be used to convert a discrete-time system into another discrete-time system that represents the same process with a different sampling period T_2. Given a discrete-time system with system function $H(z)$, one first obtains the transfer function $\hat{H}(s)$ of a continuous-time representation by plugging $z = \frac{1+sT/2}{1-sT/2}$ into $H(z)$. After that, one transforms this representation to a discrete-time one with sampling period T_2 by substituting $s = \frac{2}{T_2}\frac{z-1}{z+1}$ into $\hat{H}(s)$. Put shortly, the system function of a discrete-time system that represents the process with sampling period T_2 can be obtained by substituting the following expression for z into the system function $H(z)$:

$$z = \frac{1+sT/2}{1-sT/2} = \frac{1+\frac{T}{T_2}\frac{z-1}{z+1}}{1-\frac{T}{T_2}\frac{z-1}{z+1}} = \frac{z+1+\frac{T}{T_2}(z-1)}{z+1-\frac{T}{T_2}(z-1)}. \tag{41}$$

5.3 Construction of discrete-time systems from measurement data

In this section we investigate how one can obtain a discrete-time system of the form (1) for a given process, if measurements of the impulse response $h(t)$ are available. Here we are interested in finding second- and third-order systems, i.e. systems of the form (1) with $N = M = 2$ and $N = M = 3$.

First, consider the case $N = M = 2$. Since we want to find a discrete-time system whose impulse response approximates $h(t)$, we first try to find a suitable impulse response of a continuous-time second-order system and use the theory from the previous section to transform this to a discrete-time representation.

In this work we are interested in systems for which the impulse response is of the form

$$\tilde{h}(t) = e^{-at}(b\cos(ct) + d\sin(ct)), \qquad a, b, c, d \in \mathbb{R}. \tag{42}$$

The same analysis can be done for other second-order systems as well. The ideas are the same for those cases. Furthermore, the book [6] also describes various methods to obtain discrete-time models from sampled data, which might be useful for further research.

For a system with an impulse response of the form (42), note that in order to ensure $\lim_{t\to\infty}\tilde{h}(t) = 0$ it is sufficient to require $a > 0$. Using the Mathematica function `NonlinearModelFit`, one can then compute appropriate parameters a, b, c, d such that $\tilde{h}(t) \approx h(t)$. This can be done by creating a list of data points `data` from $h(t)$, and then running the command

```
NonlinearModelFit[data,Exp[-a t](b Cos[c t]+d Sin[c t]),{a,b,c,d},t].
```

Once a fit $\tilde{h}(t)$ has been found, a corresponding discrete-time system with sampling period T can be found as follows.

1. Compute the transfer function (assuming $a > 0$ and $\tilde{h}(t) = 0$ for $t \leq 0$):

$$\begin{aligned}\tilde{H}(s) &= \mathcal{L}\{\tilde{h}(t)\} \\ &= \int_0^\infty e^{-st} e^{-at}(b\cos(ct) + d\sin(ct))\mathrm{d}s \\ &= e^{-(a+s)t} \frac{(bc - da - ds)\sin(ct) - (ab + bs + cd)\cos(ct)}{a^2 + 2as + c^2 + s^2}\Bigg|_{t=0}^{\infty} \\ &= \frac{ab + cd + bs}{a^2 + c^2 + 2as + s^2}.\end{aligned} \tag{43}$$

2. Transform $\tilde{H}(s)$ to the z-domain using the bilinear transform to obtain the system function:

$$H(z) = \tilde{H}(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}. \tag{44}$$

3. Since $\tilde{H}(s)$ is a rational function, $H(z)$ is rational as well. Hence we can easily retrieve the parameters a_0, a_1, a_2 and b_0, b_1, b_2 in (21) that define the discrete-time system.

In case measurements of the *step response* of the process are available, one can also fit a function $\tilde{g}(t)$ that represents the step response of a system to the data. For systems that have an impulse response of the form (42) whose step response approaches 1 as $t \to \infty$, $\tilde{g}(t)$ is of the form

$$\tilde{g}(t) = 1 + e^{-at}(b\cos(ct) + d\sin(ct)), \qquad a, b, c, d \in \mathbb{R}. \tag{45}$$

Once a fit has been found, we want to find the transfer function of the system $\tilde{H}(s)$. For this, we use the fact that it relates the input and output of the system via $\mathcal{L}\{y(t)\} = \tilde{H}(s)\mathcal{L}\{x(t)\}$. Hence

$$\tilde{H}(s) = \frac{\mathcal{L}\{\tilde{g}(t)\}}{\mathcal{L}\{u(t)\}} = s\mathcal{L}\{\tilde{g}(t)\} = 1 + \frac{(ab + cd)s + bs^2}{a^2 + c^2 + 2as + s^2}. \tag{46}$$

Now that the transfer function has been found, steps 2 and 3 as listed above can be applied to find the corresponding discrete-time system.

For third-order systems we follow the same steps, where we use the following form for the impulse and step response respectively:

$$\begin{aligned}\tilde{h}(t) &= e^{-at}(b\cos(ct) + d\sin(ct)) + fe^{-gt}, \quad & a, b, c, d, f, g \in \mathbb{R}, \\ \tilde{g}(t) &= 1 + e^{-at}(b\cos(ct) + d\sin(ct)) + fe^{-gt}, \quad & a, b, c, d, f, g \in \mathbb{R}.\end{aligned} \tag{47}$$

Note that $a, g > 0$ is a sufficient condition for $\lim_{t\to\infty} \tilde{h}(t) = 0$.

Example 5.1. In this example we consider a process described by a 4th order system of the form (1) with the following parameters

$$\begin{array}{lllll} a_1 = 3.343 & a_2 = -4.462 & a_3 = 2.839 & a_4 = -0.7249 & \\ b_0 = 0.0384 & b_1 = -0.01186 & b_2 = -0.05623 & b_3 = 0.01489 & b_4 = 0.02083. \end{array} \tag{48}$$

Suppose that this system gives an exact representation of the process, and that we only have knowledge of a list of samples from the step response of the process. In this example we want to find a *third-order* discrete-time system to describe the process.

First we find an appropriate function $\tilde{g}(t)$ using the command `NonlinearModelFit` in Mathematica, as explained above. We find the function

$$\tilde{g}(t) = 1 - 0.84\, e^{-0.075t} + e^{-0.033t}(0.08\sin(0.64t) - 0.16\cos(0.64t)). \tag{49}$$

Figure 3 shows the step response of the process together with $\tilde{g}(t)$. The figure also shows the data points that we used to find the fit.

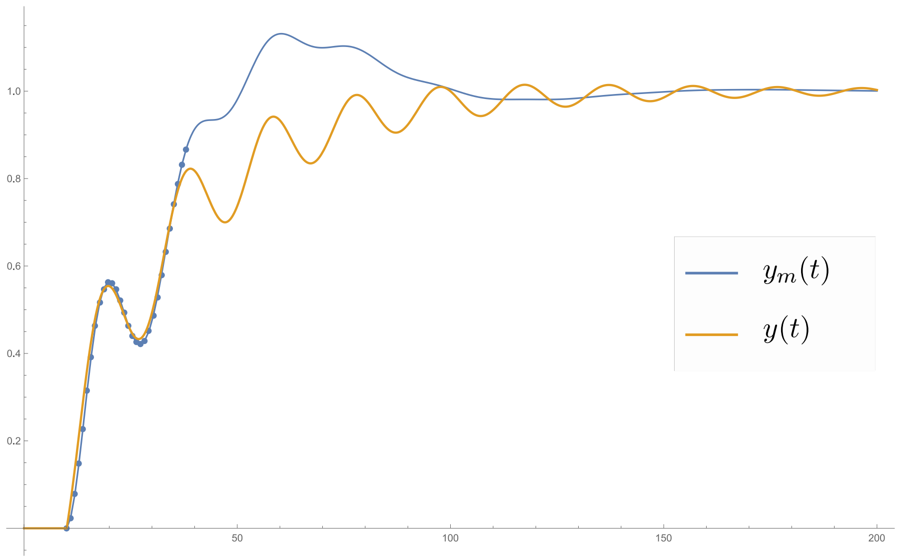


Figure 3: Third order fit

To find the discrete-time system corresponding to $\tilde{g}(t)$, we apply the procedure that was described above. First, we compute the transfer function $\tilde{H}(s) = s\mathcal{L}\{\tilde{g}(t)\}$ using the command `LaplaceTransform` which yields

$$\tilde{H}(s) \approx \frac{0.12s^2 + 0.073s + 0.030}{s^3 + 0.14s^2 + 0.41s + 0.030}. \tag{50}$$

Then we find the system function by substituting $s = 2/T(z-1)/(z+1)$. We choose the sampling period $T = 1$. This yields

$$H(z) = \frac{0.069 - 0.026z^{-1} - 0.056z^{-2} + 0.038z^{-3}}{1 - 2.5z^{-1} + 2.4z^{-2} - 0.87z^{-3}}, \tag{51}$$

with the corresponding difference equation given by

$$\Sigma_{\mathrm{DT}}: \quad \begin{aligned} y[n] =& 2.5\, y[n-1] - 2.4\, y[n-2] + 0.87\, y[n-3] \\ &+ 0.069\, x[n] - 0.026\, x[n-1] - 0.056\, x[n-2] + 0.038\, x[n-3]. \end{aligned} \tag{52}$$

Remark 5.1. In finding the discrete-time system corresponding to the step response $g(t)$, we lose some precision in the transformation from the s-domain to the z-domain. This can be overcome by using a different discretization method. For this, define the sampled version $\hat{g}[n]$ of $g(t)$ as $\hat{g}[n] := g(nT)$. The system function $\hat{H}(z)$ corresponding to the step response $\hat{g}[n]$ can be computed directly as

$$\hat{H}(z) = \frac{Y(z)}{X(z)} = \frac{\mathcal{Z}\{\hat{g}[n]\}}{\mathcal{Z}\{u[n]\}} = \mathcal{Z}\{\hat{g}[n]\}\frac{z-1}{z}, \tag{53}$$

yielding a discretization $\hat{\Sigma}_{\mathrm{DT}}$ which does not involve an approximation. The only requirement is that the $\mathcal{Z}$-transform of $\hat{g}[n]$ can be computed, which is generally the case if $\hat{g}[n]$ is obtained from (45) or (47). Figure 4 shows a plot of the step response $g(t)$ from

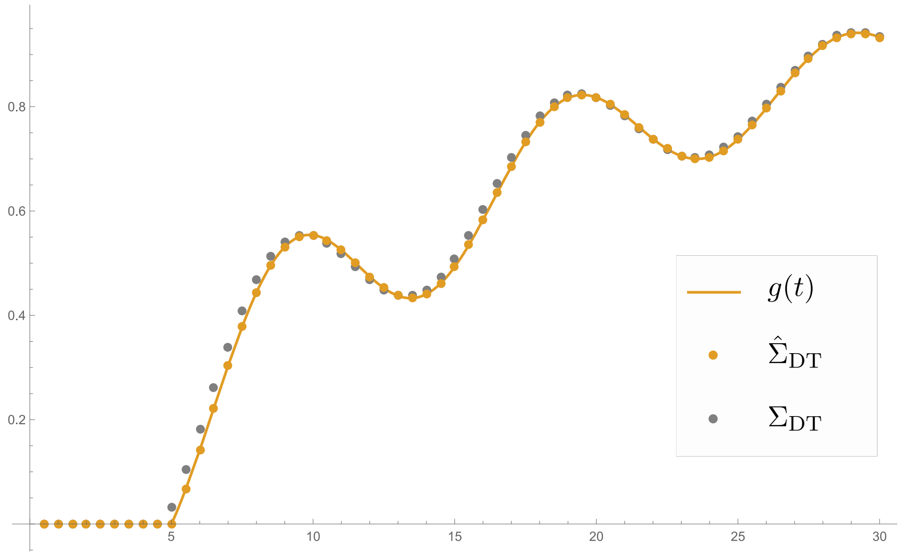


Figure 4: Two discretizations. $\hat{\Sigma}_{\mathrm{DT}}$ is obtained by sampling the step response $g(t)$, and Σ_{DT} is obtained via the bilinear transform.

Example 5.1 together with the step response of the discretization $\hat{\Sigma}_{\mathrm{DT}}$ that is found from (53), and the step response of Σ_{DT} in (52). From this plot we see that $\hat{\Sigma}_{\mathrm{DT}}$ indeed gives an exact discretization, whereas Σ_{DT} is not exact due to the transformation from the s domain to the z domain.

6 Feedback control of discrete-time systems

In this chapter we investigate how one can control the output of discrete-time systems using output measurements. This type of control is referred to as *feedback control*. The most commonly used feedback controller in engineering practices is the so-called *PID controller*. In the next section we give a definition of the PID controller together with an example. After that, we will see how a so-called *Kalman filter* can be used to deal with delays in the measurement data. Let us start by introducing the PID controller.

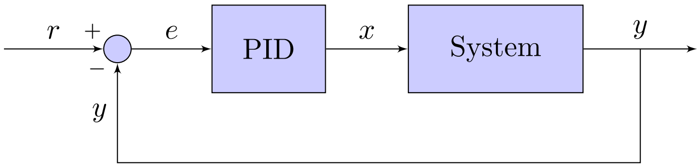
6.1 The PID controller

Given a discrete-time system and a *target*, i.e. a sequence $r[n]$ that we want the output to track, a PID controller takes the output of the system and computes a new input for the system such that $y[n]$ approaches $r[n]$. Specifically, a PID controller computes the error $e[n] := r[n] - y[n-1]$ from the available measurement $y[n-1]$ and computes the control input $x[n]$ as

$$x[n] = K_p e[n] + K_i \sum_{k=1}^{n} e[k] + K_d(e[n] - e[n-1]), \tag{54}$$

where K_p, K_i and K_d are the controller parameters. The first term in the above expression is referred to as the *proportional* action, the second is referred to as the *integral* action, and the third is called the *derivative* action, hence the name of the controller. The following block diagram gives a schematic representation of the *control loop* consisting of the system with a PID controller. For a more extensive discussion on PID controllers and parameter tuning, see [3].

Figure 5 shows a plot of the system from Example 5.1 interconnected with a PID controller. The parameters $K_p = 1.5$, $K_i = 0.048$ and $K_d = 25$ are found by tuning them using the Mathematica function `Manipulate`.



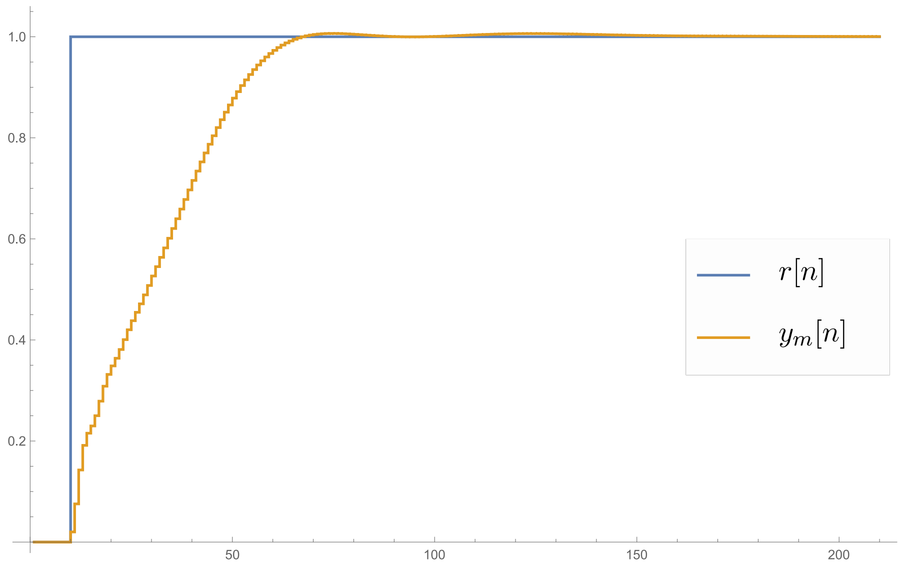


Figure 5: PID controller

6.2 Feedback control using measurement prediction

In many situations, one is interested in predicting future outputs of a process. For example, if the measurements that are available for a PID controller are delayed by a number of steps, it can be advantageous to compute an estimation of the process output at the current time step. In this section we investigate how this can be done.

A delayed measurement $y[n-\tau]$ can be seen as an *estimation* of the current output $y[n]$ of the process. An unsophisticated estimation, that is. It basically assumes that the output remains constant on the interval $[n-\tau, n]$. However, if some knowledge of the process dynamics is available, one can try to predict the course of the output based on previous measurements and hence obtain a better estimation. This is precisely the underlying idea of a so-called *Kalman filter*. A Kalman filter is a filter that predicts the future output of a process based on an available model of the process. To learn more about the Kalman filter, see e.g. [7], [2]. In Section 5.3 we saw a means to obtain a model of the process from measurement data. For now, let us assume that a system Σ of the form (1) is available for our Kalman filter, where we assume that $a_0 = 1$ without loss of generality. The system can be written as

$$\Sigma: \quad y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{r=0}^{M} b_r x[n-r]. \tag{55}$$

Furthermore, we assume that at each time step n, the computed input $x[n]$ is available.

To give a definition of the Kalman filter, let us start with the case $\tau = 1$, so the measurements $y_m[k]$ are available for all $k \leq n-1$. We aim to find an expression for $y_m[n]$ in terms of Σ. First, we try to find an expression for the entire output sequence y_m in terms of the approximate model Σ. For this, we need the inverse Σ^{-1} of the model, which can be found by simply inverting the rational function $H(z)$ and deducing the corresponding difference equation, as explained in Chapter 4.2. This yields

$$\Sigma^{-1}: \quad x[n] = \frac{1}{b_0} y[n] + \sum_{k=1}^{N} \frac{a_k}{b_0} y[n-k] - \sum_{r=1}^{M} \frac{b_r}{b_0} x[n-r]. \tag{56}$$

Notice that the inverse model is not necessarily stable, even if Σ is stable. The idea behind the inverse model is illustrated in Figure 6, where the measurements y_m are applied to it to obtain a sequence x_m. Applying x_m to the model we obtain the sequence y_m again. The clearest way to see this relation mathematically is from the z-domain: let $Y_m(z)$ and $X_m(z)$ denote the $\mathcal{Z}$-transforms of y_m and x_m respectively. Then by

$$y_m \longrightarrow \boxed{\Sigma^{-1}} \xrightarrow{x_m} \boxed{\Sigma} \xrightarrow{y_m}$$

Figure 6: The inverse model

definition, $X_m(z) = H^{-1}(z)Y_m(z)$, so applying x_m to Σ indeed yields the output $Y(z) = H(z)X_m(z) = Y_m(z)$. Hence, for all n we have the following expression for $y_m[n]$:

$$y_m[n] = -\sum_{k=1}^{N} a_k y_m[n-k] + \sum_{r=0}^{M} b_r x_m[n-r]. \tag{57}$$

Notice that the values $x_m[n-1], \ldots, x_m[n-M]$ can be obtained from the available measurements $y_m[l]$, $l \leq n-1$, using the inverse model (56). Since the measurement $y_m[n]$ itself is needed to compute the remaining value $x_m[n]$, we use the actual input $x[n]$ as an approximation. In this way, $y_m[n]$ can be estimated as

$$y_m[n] \approx \hat{y}[n] = -\sum_{k=1}^{N} a_k y_m[n-k] + b_0 x[n] + \sum_{r=1}^{M} b_r x_m[n-r]. \tag{58}$$

This procedure can be applied at each time step n to obtain an estimation for y_m. This is how we define a *Kalman filter*. To summarize, the Kalman filter computes an estimation $\hat{y}[n]$ for $y_m[n]$ using previous measurements, the last input value $x[n]$ and the values x_m that can be obtained from the previous measurements. Figure 7 gives a schematic overview of a control loop with a Kalman filter.

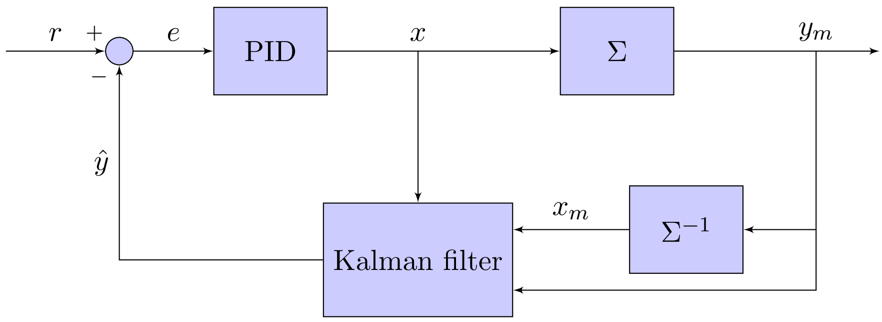


Figure 7: Kalman filter with the inverse model in a control loop

Now let us generalize this idea to include the case where at step n the measurement $y_m[l]$ is only known for $l \leq n - \tau$ where $\tau \geq 1$. We initialize the estimated output sequence $\hat{y}$ for $l \leq n - \tau$ as $\hat{y}[l] = y_m[l]$. Furthermore, we define the corresponding sequence $\hat{x}$ as

$$\hat{x}[l] = \begin{cases} x_m[l] & \text{for } l \leq n - \tau, \\ x[l] & \text{for } l > n - \tau. \end{cases} \tag{59}$$

Then for $l = n - \tau + 1, n - \tau + 2, \ldots, n$ we define $\hat{y}[l]$ recursively as

$$\hat{y}[l] = -\sum_{k=1}^{N} a_k \hat{y}[l-k] + \sum_{r=0}^{M} b_r \hat{x}[l-r]. \tag{60}$$

Then $\hat{y}[n]$ is an estimation for the measurement $y_m[n]$.

Example 6.1. Consider again the process from Example 5.1. Let us assume that the fourth order model is unknown, and that we can measure the output of the process with a delay of 10 time steps. Then, the delayed measurements can be used to control the process with a PID controller. Choosing the controller parameters optimally, the process can be controlled as shown in Figure 8.

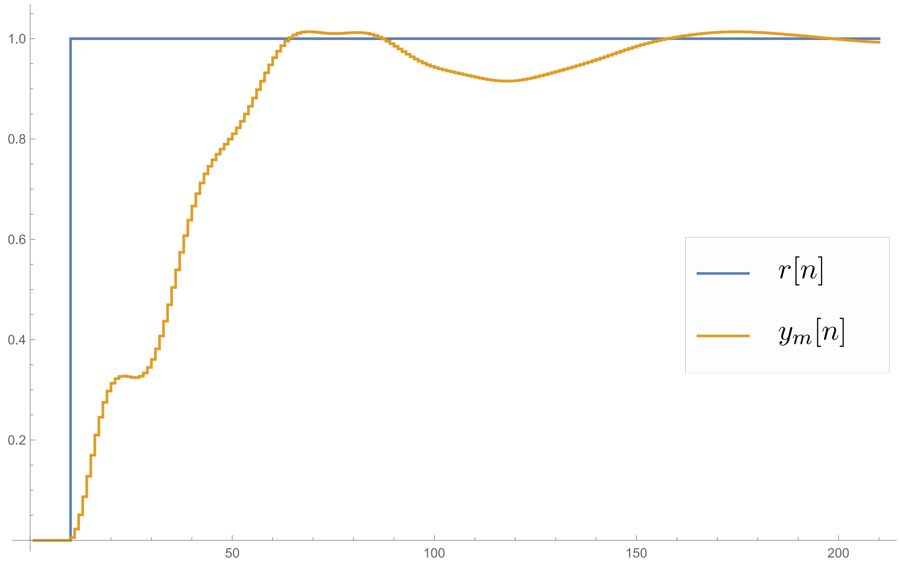


Figure 8: PID controller with delayed measurements

Alternatively, we can compute an estimation of the current output at each time step using a Kalman filter, and use the estimation to control the process. For this let us take the third-order discrete-time system that we found in Example 5.1. The inverse of the model can be computed by inverting the system function and deducing the difference equation. Equivalently we can solve the difference equation (52) for $x[n]$. We find

$$\begin{aligned} x[n] = 0.37x[n-1] + 0.81x[n-2] - 0.55x[n-3] \\ + 14.4y[n] - 36.3y[n-1] + 34.8y[n-2] - 12.6y[n-3]. \end{aligned} \tag{61}$$

At each time step n, we use the above model to compute the values $x_m[n-10], x_m[n-11]$, and $x_m[n-12]$ from the available measurements y_m. Then we compute the estimation $\hat{y}[l]$ for $l = n-\tau+1, n-\tau+2, \ldots, n$ defined by (60) and feed $e[n] := r[n] - \hat{y}[n]$ to the PID controller. In this way, the controller parameters can be tuned to obtain the result that is shown in Figure 9.

7 Concluding remarks and discussion

In this report we investigated how a discrete-time representation of a process can be obtained and how it relates to a continuous-time one. We gave a basic description of the theory that is involved in obtaining the representations and transforming between

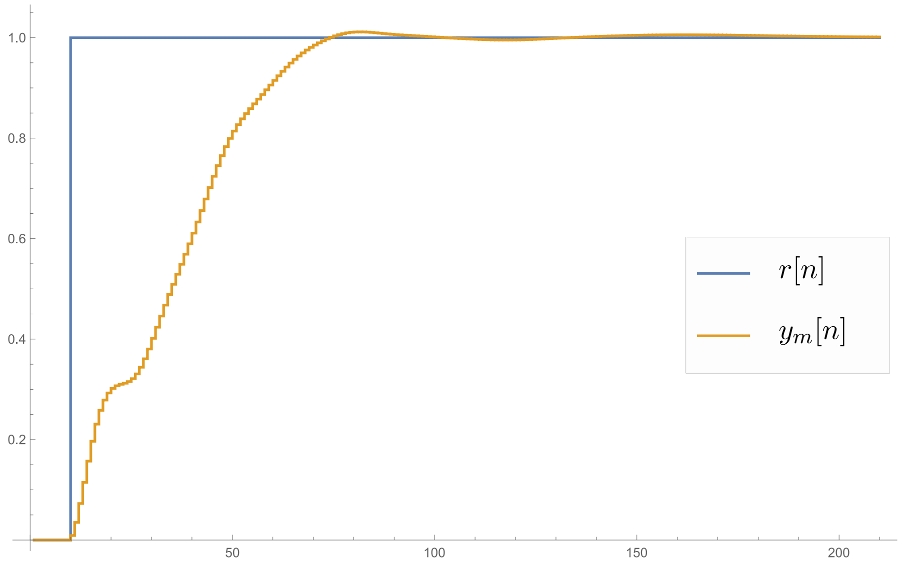


Figure 9: PID controller using Kalman filter

continuous and discrete time. After that we discussed control theory of discrete-time systems. In particular, we investigated how discrete-time systems can be controlled by a PID controller, and how performance of the PID controller can be improved by predicting process outputs.

In order to obtain a good prediction of the process outputs, we used a so-called inverse model. However, the inverse model can be highly unstable, even if the model itself is stable, yielding situations in which the PID parameters can only be chosen small to prevent oscillatory behavior. This was not elaborated on in the report, but during the project we also discussed other ways of determining a good prediction. One idea was to simply let the inverse model out and use the actual input $x[n]$ as an estimation for the sequence $x_m[n]$. For further research, these ideas can be made more concrete, to provide a good alternative to the inverse model.

We also saw that the discretization that was obtained via the bilinear transform involved an error, whereas a discretization can also be found by simply sampling the step response. This discretization does not yield an error, as was explained in Remark 5.1. This could already yield an improvement to the results from Chapter 6, but we did not have the time to compute new simulations to include in the report. Even though the discretization via the bilinear transform is inferior to the sampling method when it comes to accuracy, the bilinear transform might prove useful in case the $\mathcal{Z}$-transform of the sampled step response is difficult to find, or in case only the transfer function of a continuous-time representation is available.

Finally, as mentioned in the introduction and in [7], we did not consider noise in the measurement data for the construction of the Kalman filter in this project. Since a lot of literature on Kalman filtering to overcome such noise is available, this could be a topic of further research.

Word of thanks

I would like to thank George Schut for giving me the opportunity to do an internship at Schut Geometrical Metrology. The problem I have been working on was challenging for me, since some of the ideas and theories we treated were completely new. It was also very interesting to discover how the ideas are relevant in practice. Also, the communication was very pleasant. I have always felt the space to share my thoughts.

I would also like to thank Bart Besselink for showing genuine interest in the internship. It was really encouraging to see your interest and hear your ideas on the topic.

References

[1] R. Agarwal, M. Bohner, D. O'Regan, and A. Peterson. Dynamic equations on time scales: A survey. *Journal of Computational and Applied Mathematics*, 141:1–26, 07 2004.

[2] B.D.O Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

[3] K.J. Aström and R.M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[4] L.C. Ludeman. *Fundamentals of digital signal processing*. Harper and Row, 1986.

[5] H.L. Trentelman, A.A. Stoorvogel, and M. Hautus. *Control Theory for Linear Systems*. Communications and Control Engineering. Springer, London, UK, 2001.

[6] M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2007.

[7] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

Appendix A: Table of $\mathcal{Z}$-transform pairs

A more extensive version of this table can be found in [4].

Discrete time sequence	$\mathcal{Z}$-transform	Region of convergence
$\delta[n]$	1	all z
$u[n]$	$\frac{z}{z-1}$	$\lvert z\rvert > 1$
$nu[n]$	$\frac{z}{(z-1)^2}$	$\lvert z\rvert > 1$
$n^2u[n]$	$\frac{z(z+1)}{(z-1)^3}$	$\lvert z\rvert > 1$
$a^nu[n]$	$\frac{z}{z-a}$	$\lvert z\rvert > a$
$na^nu[n]$	$\frac{az}{(z-a)^2}$	$\lvert z\rvert > a$
$\sin(\omega n)u[n]$	$\frac{z\sin\omega}{z^2-2z\cos\omega+1}$	$\lvert z\rvert > 1$
$\cos(\omega n)u[n]$	$\frac{z^2-z\cos\omega}{z^2-2z\cos\omega+1}$	$\lvert z\rvert > 1$