4 Time-Discrete Systems

Abstract

With more and more powerful digital processing devices (computers and microcontrollers), digital feedback controls have become ubiquitous. However, a fundamental difference to time-continuous systems exists: data processing systems take samples of continuous signals at constant time intervals T. The signal is *discretized*, and the true value of the signal is not known until the next sample is taken. The resulting time lag T introduces an irrational term e^{-sT} in the Laplace domain and thus makes it impossible to accurately compute a system response with Laplace-domain methods. For time-discrete systems, an equivalent transform exists, the z-transform, which allows the mathematical treatment of discrete systems in the z-domain in a similar fashion to the treatment of continuous systems in the Laplace domain. In this chapter, the z-transform and its justification though a discrete sampler model are introduced. The relationship between Laplace- and z-domains is explored, and methods are provided to convert z-transformed signals into time-domain sequences of signals.

Digital control systems are time-discrete systems. The fundamental difference between continuous and time-discrete systems comes from the need to convert analog signals into digital numbers, and from the time a computer system needs to compute the corrective action and apply it to the output. A typical digital controller is sketched in Figure 4.1. An analog signal, either a sensor signal or the control deviation, is fed into an analog-to-digital converter (ADC). The ADC samples the analog signal periodically with a sampling period T. The sampling process can be interpreted as a sample-and-hold circuit followed by digital read-out of the converted digital value Z. The microcontroller reads Z and computes the corrective action A, which is applied as a digital value at its output. Since the computation takes a certain amount of time, A becomes available with a delay T_D after Z has been read. The digital value A is converted into an analog signal through a digital-to-analog converter (DAC). The analog signal still has the stair-step characteristic of digital systems, and a lowpass filter smoothes the steps. Both the sampling interval T and the processing delay T_D introduce phase shift terms in the Laplace domain that fundamentally change the behavior of time-discrete systems when compared to continuous systems. Time-delay systems with a delay τ have a Laplace transfer function $H(s) = \exp(-s\tau)$ (see Section 3.6.9). We can see that $H(s) \rightarrow 1$ when $\tau \rightarrow 0$. In other words, when the sampling frequency is very high compared to the dynamic response of the system, treatment as a continuous system is possible. As a rule of thumb, the approximated Laplace-domain pole created at $-2/T$ should be at least ten to twenty times further to the left than the fastest system pole. If this is not the case, the methods introduced in this chapter need to be applied.

Linear Feedback Controls. http://dx.doi.org/10.1016/B978-0-12-405875-0.00004-8

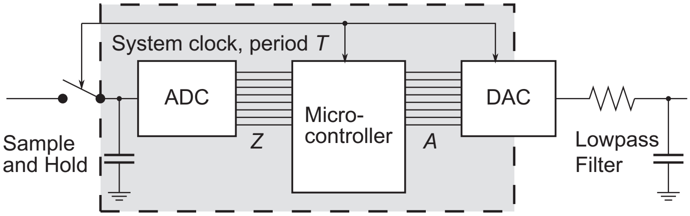

Figure 4.1 Basic schematic of a digital control system. The analog-to-digital converter (ADC) samples the analog input signal in discrete time intervals T. The sampling process can be interpreted as a sample-and-hold circuit with subsequent digital read-out. The microcontroller receives a digital value Z as its input, from which it computes the corrective action A. A, itself a digital value, is applied to a digital-to-analog converter (DAC), and its analog output signal passes through a lowpass filter to remove frequency components related to the sampling frequency. The gray-shaded part of the system represents the time-discrete domain.

For the considerations in this chapter, we need to make two assumptions. First, the sampling interval is regular, that is, the time from one discrete sample to the next is the constant T. Second, we either assume that the processing time T_D is small compared to T, and the corrective action is available immediately after a sample has been taken, or we assume that the processing time T_D is constant and can be included in the definition of T.

4.1 Analog-to-Digital Conversion and the Zero-Order Hold

Many integrated ADC have a digital input signal that initiates a conversion. At the exact instant the conversion is started, the ADC takes a sample of the analog signal $f(t)$ and, after completion of the conversion, provides the proportional digital value Z. Strictly, the sampled values of $f(t)$ are known on the digital side only for integer multiples of the sampling interval T. When a sample is taken at $t = k \cdot T$, the corresponding digital value is $Z_k(t) = f(kT) \cdot \delta(t - kT)$. The discretized (sampled) signal $f_s(t)$ can therefore be interpreted as the set of all Z_k, superimposed:

$$f_s(t) = f(t) \cdot \sum_{k=-\infty}^{\infty} \delta(t - kT) = \sum_{k=-\infty}^{\infty} f(kT) \cdot \delta(t - kT) \tag{4.1}$$

Equation (4.1) describes a discrete convolution of the original signal $f(t)$ with a sequence of equidistant delta-pulses. The interpretation of the sampled signal as a sequence of scaled delta-pulses in shown in Figure 4.2b. An alternative interpretation is possible where any discretely sampled value Z_k remains valid until the next sampling takes place. This interpretation leads to a stair-step-like function as depicted in Figure 4.2c.

We can argue that the stair-step function in Figure 4.2c is a sequence of scaled step functions followed by the delayed step function with a negative sign. For example, at $t = 0$ (corresponding to $k = 0$) we obtain the value Z_0. For $t = T$ (corresponding

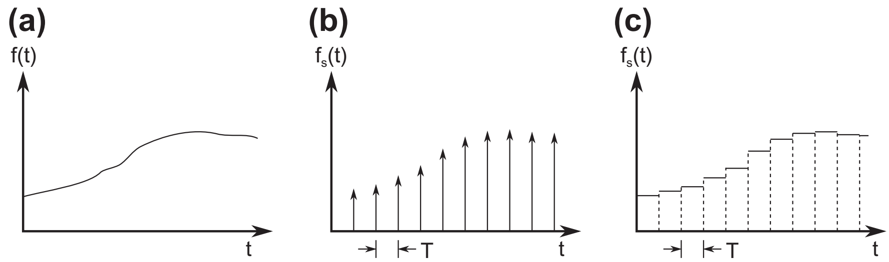

Figure 4.2 Continuous signal $f(t)$ and two interpretations of the discretely-sampled signal $f_s(t)$. The center diagram (b) shows the individual samples as delta pulses with a rate of T (Eq. (4.1)), and the right diagram (c) shows the interpretation of the signal where the discrete sample remains valid and unchanged until the next sampling occurs.

to $k = 1$) we obtain the value Z_1. To make this work, we compose the first stair-step from $Z_0u(t) - Z_0u(t - T)$. The second stair-step follows immediately as $Z_1u(t - T) - Z_1u(t - 2T)$, and so on. The Laplace transform of the sample-and-hold unit (also referred to as *zero-order hold*) emerges as:

$$u(t) - u(t - T) \quad \circ\!\!-\!\!\bullet \quad \frac{1}{s} - \frac{1}{s}e^{-sT} = \frac{1 - e^{-sT}}{s} \tag{4.2}$$

Once again, we can see that any zero-order hold element (that is, any digital, time-discrete processing element) introduces an irrational term, namely, e^{-sT} into the Laplace transform. It is therefore no longer possible to determine the system response by computing the roots of the characteristic polynomial. However, we can examine the Laplace transform of the discretely sampled signal by transforming the convolution in Eq. (4.1):

$$\mathscr{L}\{f_s(t)\} = \int_0^{\infty} \left[\sum_{k=-\infty}^{\infty} f(kT) \cdot \delta(t - kT) \right] e^{-st}\, \mathrm{d}t = \sum_{k=0}^{\infty} f(kT) \cdot e^{-skT} \tag{4.3}$$

Once again, the time-delay term e^{-sT} is part of the transform. The Laplace transform in Eq. (4.3) becomes particularly interesting when we define a new complex variable, z, as

$$z = e^{sT} \tag{4.4}$$

and rewrite Eq. (4.3) as

$$\mathscr{L}\{f_s(t)\} = \sum_{k=0}^{\infty} f(kT) \cdot z^{-k} \tag{4.5}$$

The Laplace transform of a discretely sampled signal has now become an infinite sum, and a new variable z has been introduced that depends on the sampling period T. This infinite sum leads directly to the definition of the z-transform (Section 4.2).

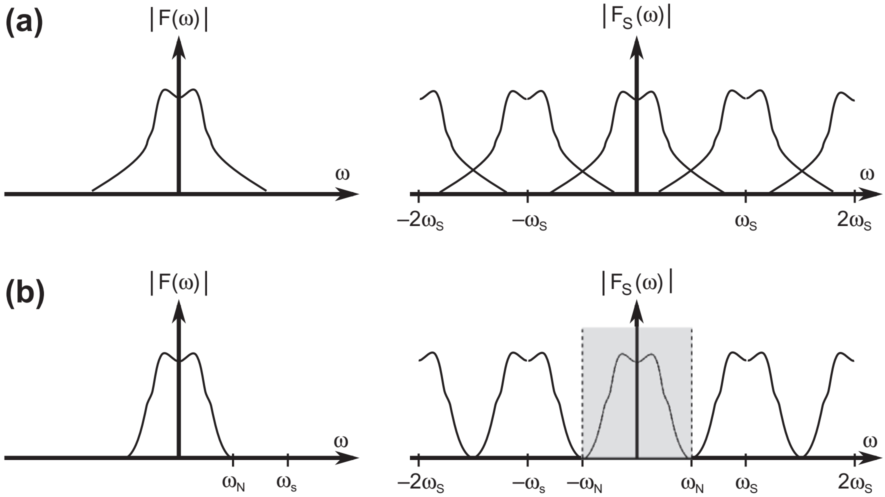

Figure 4.3 (a) Sketch of the frequency spectrum $|F(\omega)|$ of $f(t)$ and the corresponding spectrum of the sampled signal $f_s(t)$. The spectrum of the sampled signal contains replicated image spectra of the baseband spectrum at intervals of ω_s. If the signal $f(t)$ is not bandlimited, spectral components extend beyond $\omega_s/2$ and overlap with the image spectra. Original and overlapping spectral components cannot be separated. (b) Spectrum of $f(t)$ after limitation of the bandwidth (lowpass filtering). Since no spectral components extend beyond the Nyquist frequency ω_N, an ideal lowpass filter (shaded area, dashed lines) can be used to separate the baseband spectrum from the image spectra and thus restore $f(t)$.

At this point, the sampling theorem needs to be briefly introduced. The *comb*-function, that is, the sequence of shifted delta-pulses that is used to sample the signal, has its own frequency spectrum. The multiplication of a signal with a comb function (shown in Eq. (4.1)) therefore corresponds to a convolution of the frequency spectra of $f(t)$ and the comb function. The Fourier spectrum of the comb function was derived in Section 3.2. In simple terms, frequency spectra are replicated along multiples of the sampling frequency $\omega_s = 2\pi/T$ as shown in Figure 4.3. Any frequency components that extend above ω_s overlap with frequency components of the replicates. This ambiguity of the spectra in the frequency domain is known as *aliasing*. Reconstruction of the original signal $f(t)$ is not possible, because the original spectrum and the aliased components cannot be separated.

In practice, any signal to be sampled needs to be band-limited to half of the sampling frequency. We define this frequency $\omega_N = 2\pi f_N = \omega_s/2$ as *Nyquist frequency.* Band-limiting can be achieved by using an *analog* lowpass filter before the ADC. Often, a first-order lowpass is sufficient when only high-frequency noise needs to be attenuated and when the sampling frequency is sufficiently high. On the other hand, higher-order filters are needed when fast transients of the system play a major role in the controller design. These filters need to be considered in the overall transfer function of the system.

Here, the importance of the lowpass filter at the output of the digital system in Figure 4.1 becomes evident. The discrete, sampled signal at the output of the DAC consists of the *baseband spectrum*, that is, the original spectrum centered at $\omega = 0$

and the *image spectra* (i.e., the replicated spectra at $\pm k\omega_s$). A lowpass filter can isolate the baseband spectrum and thus restore the original signal. The boxcar filter indicated in Figure 4.3b does not exist in practice. Rather, a compromise needs to be found between the effort for designing a steep filter—both at the input and the output—and for simply raising the sampling rate: the more separation exists between the bandlimit of the baseband spectrum and the Nyquist frequency, the less effort is needed to filter the unwanted frequency components.

An example for this challenge can be found in the early designs of the compact disc (CD) player. Music on a CD is digitized at 44.1 kHz, and a bandlimit of $f_N = 22.05$ kHz is necessary. The maximum frequency was chosen for physiological reasons. When the CD was first introduced, digital signal processors were just barely able to perform reconstruction and error correction in real time, and high-order analog Tchebyshev filters were used at the output to remove frequency components from the sampler. Many users complained about a "cold" or "technical" aspect of the sound, which may be attributed to the poor step response and the passband ripple of the filters. As digital signal processors became more powerful, the problem was solved by digitally interpolating new samples between existing ones and thus raising the apparent sampling frequency to 88.2 kHz or even higher. The use of less steep filters with better step response now became feasible. Not unusual for high-end audio, however, other listeners now complained about perceived deficiencies in the sound quality …

4.2 The *z*-Transform

Given a continuous signal $f(t)$ and a sequence of samples taken at time points $t_k = kT$, we can interpret the sequence $f_k = f(kT)$ as the discretely sampled version $f_s(t)$ of the signal $f(t)$. The z-transform of a sequence is defined as

$$F(z) = \mathscr{Z}\{f_s(t)\} = \mathscr{Z}\{f(t)\} = \sum_{k=0}^{\infty} f(kT) \cdot z^{-k} \tag{4.6}$$

We note that the z-transform of the continuous signal $f(t)$ and the sampled signal $f_s(t)$ are identical, and that the sampled signal f_k has identical z- and Laplace-transforms. Moreover, the z-transform, like the Laplace transform, is a linear operation, that is, the scaling and superposition principles hold (see Table B.6).

Similar to the Laplace transform, the z-transform of discretized signals can be computed and interpreted as two-dimensional functions in the complex z-plane. Correspondence tables are available to provide the z-transform for many common signals (Tables B.1 through B.5). Let us, for example, determine the z-transform of the unit step $u(t)$. By using Eq. (4.6), we obtain

$$\mathscr{Z}\{u(t)\} = \sum_{k=0}^{\infty} z^{-k} = \frac{1}{1 - z^{-1}} \tag{4.7}$$

Convergence to the closed-term solution on the right-hand side of Eq. (4.7) is only possible when $|z| > 1$, that is, only for z that lie outside the unit circle in the z-plane. This observation will later lead to criteria of stability that differ from the Laplace transform.

As a second example, let us determine the z-transform of an exponential decay, $f(t) = e^{-at}$.

$$\mathscr{Z}\left\{e^{-at}\right\} = \sum_{k=0}^{\infty} e^{-akT} z^{-k} = \sum_{k=0}^{\infty} \left(e^{aT} \cdot z\right)^{-k} = \frac{1}{1-(ze^{aT})^{-1}} = \frac{z}{z-e^{-aT}} \tag{4.8}$$

Note that e^{-aT} is a constant that depends on the exponential decay constant a and the sampling period T. The resulting z-domain function has a zero in the origin and a pole at $z = +e^{-aT}$.

The z-transform is of importance, because the transfer functions of both continuous and time-discrete systems can be expressed in the z-domain. Furthermore, the convolution theorem holds for the z-transform. If a system with the z-domain transfer function $H(z)$ receives an input signal $X(z)$, we can compute its z-domain response through

$$Y(z) = X(z) \cdot H(z) \tag{4.9}$$

The most notable difference between the Laplace- and z-transforms is the discrete nature of the digital output signal. Consistent with time-discrete systems, the output signal is known only at integer multiples of the sampling period, that is, at $t = kT$. This behavior emerges from the digital-to-analog output stage in Figure 4.1, which receives a new value only once per sampling interval. This value is kept valid until it is replaced by the next computed output value (i.e., the behavior indicated in Figure 4.2c). **This behavior also complicates the translation of Laplace-domain problems into z-domain problems, because it is generally not possible to translate a transfer function $H(s)$ into $H(z)$ through direct correspondence.** In the next section, we will discuss how a time-continuous system with a time-discrete control element can be converted into a completely time-discrete model.

If the z-transform of a signal is known, the time-domain samples can be determined through the inverse z-transform. Formally, the inverse z-transform involves evaluating a contour integral for all integer multiples of T:

$$f_k = f(kT) = \mathscr{Z}^{-1}\left\{F(z)\right\} = \frac{1}{2\pi j} \oint_c F(z) z^{k-1}\, \mathrm{d}z \tag{4.10}$$

The contour c encircles the origin of the z-plane and lies inside the region of convergence of $F(z)$. For the contour $|z| = 1$, the contour integral turns into a form of the inverse time-discrete Fourier transform of $F(z)$,

$$f_k = \frac{1}{2\pi j} \int_0^{2\pi} F(e^{j\varphi}) e^{j\varphi(k-1)}\, \mathrm{d}\varphi \tag{4.11}$$

The strict definition of the inverse z-transform does not invite its practical application. Any of the following methods provides a more practical approach to obtain the discretely sampled signal values f_k from a z-domain signal $F(z)$, and each item in the list is followed by an example below:

- Correspondence tables: If $F(z)$ can be expressed as a linear combination of expressions found in the correspondence tables (Appendix B), obtaining $f_k = f(kT)$ is as straightforward as in the Laplace domain.

- Partial fraction expansion: With the exact same method as described in Section 3.5, the fraction of two polynomials $p(z)/q(z)$ can be transformed into a sum of first- and second-order polynomials, whose correspondence can usually be found in Appendix B.
- Polynomial long division: The formal polynomial long division of a fraction $p(z)/q(z)$ leads to an expression in the form $f_0 + f_1 z^{-1} + f_2 z^{-2} + \cdots$ When we consider Eq. (4.6), we can see that the coefficients of the polynomial long division are identical to the sampled values, that is, $f_k = \{f_0, f_1, f_2, \ldots\}$.
- Digital filter response: When the transfer function $p(z)/q(z)$ is known, the present discrete output value can be computed from past output values, and present and past input values. This method advertises itself for a computer-based solution.

Example: The z-transform of a sequence f_k, sampled at $T = 0.1$ s, is known to be

$$F(z) = \frac{z}{z^2 - 0.8296z + 0.1353} \tag{4.12}$$

For partial fraction expansion, it is convenient to substitute $F(z)$ for a shifted series $G(z)/z$. The reason will become clear further below. The roots of the denominator polynomial are $z_1 = 0.6065$ and $z_2 = 0.2231$. We therefore write

$$\begin{aligned} \frac{G(z)}{z} &= \frac{z}{z^2 - 0.8296z + 0.1353} \\ &= \frac{P}{z - 0.6065} + \frac{Q}{z - 0.2331} = \frac{1.582}{z - 0.6065} - \frac{0.582}{z - 0.2331} \end{aligned} \tag{4.13}$$

where the residuals $P = 0.1582$ and $Q = -0.582$ were found with Eq. (3.79). The two additive terms in Eq. (4.13) do not exist in the correspondence tables. However, if we multiply both sides with z, we obtain correspondences for the exponential decay (Table B.2):

$$G(z) = 1.582 \frac{z}{z - 0.6065} - 0.582 \frac{z}{z - 0.2331} \tag{4.14}$$

We can determine the decay constants a and b with $e^{-aT} = 0.6065$ and $e^{-bT} = 0.2331$ and obtain $a = 5\ \mathrm{s}^{-1}$ and $b = 15\ \mathrm{s}^{-1}$. The sequence $g_k = g(kT)$ is therefore

$$g(kT) = 1.582 \cdot e^{-5kT} - 0.582 \cdot e^{-15kT}; \quad k = 0, 1, 2, \ldots \tag{4.15}$$

Evaluated for $k = 0, 1, 2, \ldots$, we obtain $g_k = \{1, 0.8297, 0.553, 0.3465, 0.2127, \ldots\}$. However, we initially substituted $F(z) = G(z) \cdot z^{-1}$. We need to apply the time shift property (Table B.6), since z^{-1} corresponds to a delay by T:

$$f((k+1)T) \quad \circ\!\!-\!\!\bullet \quad zF(z) - zf(0) \tag{4.16}$$

Clearly, $zF(z) = G(z)$ with the correspondence $f((k+1)T) = g(kT)$. Because of the causality of $g(kT)$, we know $f(0) = g(-T) = 0$. The sequence f_k is therefore obtained directly from g_k by shifting the values one place to the right: $f_k = \{0, 1, 0.8297, 0.553, 0.3465, 0.2127, \ldots\}$.

Table 4.1 Polynomial long division of $z^2 - 0.8296z + 0.1353$ into z (Eq. (4.12)).

	z^{-1}	$+0.8296z^{-2}$	$+0.5529z^{-3}$	$+0.3465z^{-4}$	$+0.2127z^{-5}$	$(\ldots)$
$z^2 - 0.8296z + 0.1353$]	z					
	z	-0.8296	$+0.1353z^{-1}$			
		$+0.8296$	$-0.1353z^{-1}$			
		$+0.8296$	$-0.6882z^{-1}$	$+0.1122z^{-2}$		
			$+0.5529z^{-1}$	$-0.1122z^{-2}$		
			$+0.5529z^{-1}$	$-0.4587z^{-2}$	$+0.0748z^{-3}$	
				$+0.3465z^{-2}$	$-0.0748z^{-3}$	
				$+0.3465z^{-2}$	$-0.2875z^{-3}$	$+0.4688z^{-4}$
					$+0.2127z^{-3}$	$-0.4688z^{-4}$
						$(\ldots)$

The same result can be obtained by polynomial long division, that is, by dividing $z^2-0.8296z + 0.1353$ into z as demonstrated in Table 4.1. With the forward z-transform

$$F(z) = \sum_{k=0}^{\infty} f_k \cdot z^{-k} \tag{4.17}$$

we can read the coefficients of z^{-k} from the result of the polynomial division and obtain $f_k = \{0, 1, 0.8296, 0.5529, 0.3465, 0.2127, \ldots\}$ in accordance with the sequence found by partial fraction expansion. Note that the coefficient for z^0 is zero and therefore $f_0 = 0$. Note also that knowledge of the value for T was necessary to determine the decay constants in the z-transform correspondences, but was not needed for the polynomial long division.

To explain the inverse z-transform through implementation of a discrete filter, we re-interpret the output signal in Eq. (4.12) in two ways. First, we divide by z^2 to obtain z^0 as the highest power of z. All z^{-1} are now time delays by T. Second, we interpret $F(z)$ as a transfer function $H(z)$ with a z-domain input of 1 (i.e., a delta pulse). Since the z-transform of the delta pulse is 1, we have $F(z) = H(z)$ when the input is a delta pulse, i.e., $\delta_k = \{1, 0, 0, \ldots\}$. By cross-multiplying with the denominator polynomial, Eq. (4.12) becomes

$$F(z)\left[1 - 0.8296z^{-1} + 0.1353z^{-2}\right] = z^{-1} \cdot 1 \tag{4.18}$$

where the right-hand 1 is the z-transform of the delta pulse, $F(z)$ is the output of the digital filter, and the remaining polynomials of z are the actual filter. A filter that realizes Eq. (4.18) is sketched in Figure 4.4. Each block labeled z^{-1} represents a unit delay, that is, it holds its input value from the previous clock cycle for the duration of T. For example, if the input sequence is $\delta_k = \{1, 0, 0, 0, \ldots\}$, then the output of the first unit delay is $z^{-1}\delta_k = \{0, 1, 0, 0, \ldots\}$. Similarly, when the current output value is f_k,

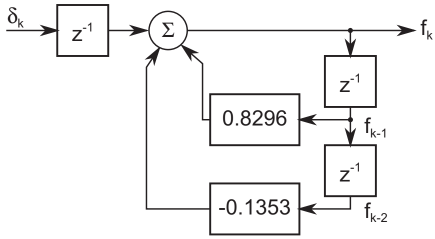

Figure 4.4 Schematic of a discrete filter that realizes the finite-difference equation, Eq. (4.18). Each block with z^{-1} represents a unit delay and can be seen as a memory unit that stores the value from the previous clock cycle. The summation point combines not only input values, but also past output values.

Table 4.2 Finite-difference calculation of the output sequence of Eq. (4.19), which is represented by the digital filter in Figure 4.4. Each line is one finite time step. Note how the output values shift to the right into the past-value columns.

f_k	f_{k-1}	f_{k-2}	x_{k-1}	x_k
0	–	–	0	1
1	0	–	1	0
0.8296	1	0	0	0
0.5529	0.8296	1	0	0
0.3465	0.5529	0.8296	0	0
0.2126	0.3465	0.5529	0	0
0.1295	0.2126	0.3465	0	0
...	...	...	...	...

the term $z^{-1}F(z)$ represents the shifted sequence f_{k-1}, and $z^{-2}F(z)$ corresponds to f_{k-2}. The summation point therefore combines input values with past output values.[1] By using these relationships, Eq. (4.18) turns into a finite-difference equation:

$$f_k = 0.8296 \cdot f_{k-1} - 0.1353 \cdot f_{k-2} + \delta_{k-1} \tag{4.19}$$

From this point, we can compute the output values as shown in Table 4.2.

4.3 The Relationship between Laplace- and z-domains

Laplace- and z-domain are strictly related when we consider the definition of the complex variable $z = e^{sT}$. Since $s = \sigma + j\omega$, we can relate any point in the s-plane

[1] Digital filters that feed back past output values are known as *infinite impulse response filters*, as opposed to *finite impulse response filters*, which only output a weighted sum of past and present input values. Digital filter theory goes beyond this book. The interested reader is referred to Richard G. Lyons' book *Understanding Digital Signal Processing.*

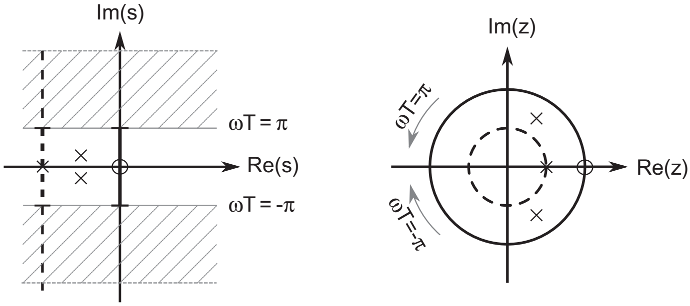

Figure 4.5 Relationship between the s- and z-planes. The imaginary axis of the s-plane is mapped to the unit circle of the z-plane, and the origin of the s-plane maps to $z = 1$ (small circles). A vertical line in the left half-plane near $\sigma = -0.69/T$ (dashed line) maps to a circle inside the unit circle (dashed). Also shown are three stable poles and their approximate corresponding location in the z-plane. The thick section of the imaginary axis maps to the complete unit circle. Increasing the frequency repeats the same circle, and the regions subject to aliasing are hatched in gray. For time-discrete systems, all frequencies need to be kept in the white band between $-\pi < \omega T \leq \pi$.

to its corresponding point in the z-plane through

$$z = e^{(\sigma + j\omega)T} = e^{\sigma T} \cdot e^{j\omega T} \tag{4.20}$$

which describes z in polar coordinates. Vertical lines in the s-plane map to circles in the z-plane as shown in Figure 4.5.

The origin of the s-plane maps to $z = 1$, and the negative branch of the real axis of the s-plane maps to the section of the real axis of the z-plane with $0 < z \leq 1$. Correspondingly, the positive (unstable) branch of the real axis of the s-plane maps to the real axis of the z-plane with $z > 1$. Negative real values of z can only be achieved with complex values of s when $\omega T = \pm\pi$, and the gray horizontal lines in Figure 4.5 map ambiguously to the negative real axis of the z-plane. A hypothetical horizontal line that begins in one of the complex poles in the s-plane and extends to $-\infty$ parallel to the real axis maps to a line in the z-plane connecting the corresponding pole to the origin. The imaginary axis of the s-plane, which separates the stable left half-plane from the unstable right half-plane maps to the unit circle $|z| = 1$. Stable poles of a z-domain transfer function lie inside the unit circle. Moreover, the frequency ambiguity known as aliasing becomes evident as increasing frequencies ω in the s-plane merely map to the same circle over and over. In fact, the harmonic component of z leads to the requirement that $-\pi < \omega T \leq \pi$. With $\omega = 2\pi f$, we arrive again at the Nyquist frequency $f_N = \pm 1/(2T)$.

The unit circle in the z-plane has special significance. From Eq. (4.20) follows $\sigma = 0$ for all $|z| = 1$, which makes intuitive sense from the fact that the imaginary axis of the s-plane maps to the unit circle of the z-plane. For $|z| = 1$ (equivalent to $z = e^{j\omega}$) we

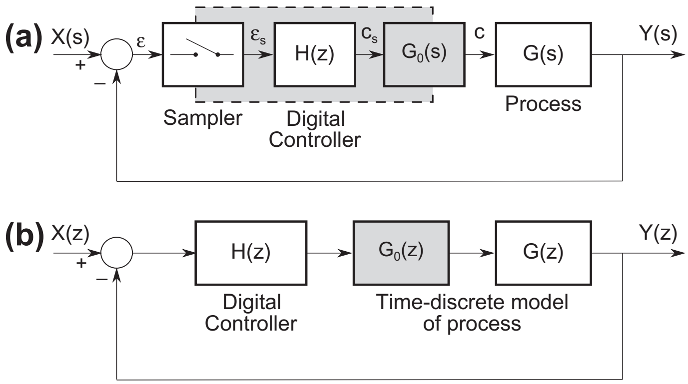

Figure 4.6 Combining time-continuous and time-discrete elements in one feedback loop. (a) The digital controller uses a sampler to sample the input signal ϵ at intervals T. The sampled, time-discrete sequence ϵ_s acts as the input of the controller, and its output is the equally time-discrete sequence c_s. The time-continuous process $G(s)$ interprets the input signal as continuous, and a virtual zero-order hold element $G_0(s)$ (gray shaded box) needs to be inserted in the model between controller and process to correctly consider the transition between discrete and continuous signals. The time-discrete elements are highlighted by the dashed box. (b) With the virtual zero-order hold, the time-discrete correspondences $G(z)$ and $G_0(z)$ of the continuous transfer functions $G(s)$ and $G_0(s)$ can be used, and the entire feedback loop turns into a time-discrete system.

can rewrite the z-transform as

$$F(\omega) = \sum_{k=0}^{\infty} f(kT) \cdot e^{-j\omega k} \tag{4.21}$$

which is actually a form of the discrete Fourier transform of the sequence $f_k = f(kT)$. The Fourier-transform nature of the unit circle becomes important when we discuss the frequency response of time-discrete systems (Chapter 11).

It is important to see that the mapping from the s- to the z-plane always depends on T. A long T (slow controller) reduces the width of the alias-free band in the s-plane, and a long T moves the mapped (stable) poles closer to the origin of the z-plane. Lastly, it is worth noting that s has units of inverse seconds, whereas z carries no units.

Frequently, a process with a time-continuous transfer function $G(s)$ is combined with a time-discrete digital controller. When the controller output is used as input for the process, a *virtual* zero-order hold element $G_0(s)$ needs to be inserted in the mathematical model. The zero-order hold is not a separate physical component, but its model is needed to account for the time-discrete nature of the controller. We can envision the zero-order hold as some form of type-adjusting element that converts a time-discrete input signal mathematically correct into a time-continuous output signal. Only then is it possible to use direct correspondences to convert the Laplace-domain transfer functions into z-domain transfer functions. The model is explained in Figure 4.6.

Let us consider an example system to illustrate these steps. To keep the example simple, we will examine an open-loop system (i.e., the feedback path in Figure 4.6 does not exist). The digital controller—again for reasons of simplification—is a P controller with $H(z) = k_p$. Therefore, $c_s(kT) = k_p \cdot \epsilon_s(kT)$. The process has the transfer function $G(s) = 5/(s(s+5))$. The transfer function of the forward path $L(s)$ in the Laplace domain is therefore

$$L(s) = \frac{Y(s)}{\epsilon(s)} = k_p \cdot G_0(s) \cdot G(s) = k_p \frac{1 - e^{-sT}}{s} \frac{5}{s(s+5)} \tag{4.22}$$

Through partial fraction expansion (note the double root at $s = 0$), we obtain a sum of first- and second-order polynomials:

$$L(s) = k_p \cdot \left(1 - e^{-sT}\right) \left(\frac{1}{s^2} - \frac{1}{5s} + \frac{1}{5(s+5)}\right) \tag{4.23}$$

The three fractions in large parentheses represent a ramp function, a step function, and an exponential decay, respectively. The equivalent z-domain terms can now be substituted:

$$L(z) = k_p \cdot \left(1 - z^{-1}\right) \left(\frac{Tz}{(z-1)^2} - \frac{z}{5(z-1)} + \frac{z}{5(z - e^{-5T})}\right) \tag{4.24}$$

Further arithmetic can be simplified by rearranging the terms and by using numerical values where applicable. If we assume $T = 0.1$ s, we can rewrite $L(z)$ as

$$L(z) = k_p \cdot \left(\frac{0.1065z + 0.0902}{5(z-1)(z-0.6065)}\right) \tag{4.25}$$

where it now could be used to compute the closed-loop impulse or step response, or to optimize the k_p parameter. It can be seen, however, that z-transform arithmetic usually suffers from considerably greater complexity than time-continuous Laplace-domain arithmetic.

A different equivalence between s- and z-domains can be based on the finite-difference approximation of simple s-domain functions. For example, the integral of a continuous function $x(t)$ from $t = 0$ to some arbitrary time point $t = N \cdot T$ can be approximated by the sum of the discrete signal values:

$$\int_0^t x(\tau)\mathrm{d}\tau \approx \sum_{k=0}^{N} x_k \cdot T \tag{4.26}$$

The multiplication with the sampling interval in Eq. (4.26) corresponds to the multiplication with $\mathrm{d}t$ in the integral. Sine the integral value is likely stored in memory anyway, Eq. (4.26) can be rewritten as a summation where each new value depends on the current sample and the previous sum,

$$y_k = \sum_{k=0}^{N} x_k \cdot T = y_{k-1} + x_k \cdot T \tag{4.27}$$

allowing for an elegant recursive formulation. Eq. (4.27) belongs in the category of *digital filters*. The z-domain transfer function of the filter can be determined by considering that a one-step delay (i.e., y_{k-1}) corresponds to a multiplication with z^{-1}. The z-transform of the integrating filter therefore becomes

$$Y(z) = z^{-1}Y(z) + X(z) \cdot T \tag{4.28}$$

Since this filter has an input (namely, $X(z)$) and an output (namely, $Y(z)$), we can determine its transfer function by factoring out $Y(z)$ and applying Eq. (4.9),

$$H_{intg}(z) = \frac{Y(z)}{X(z)} = \frac{T}{1 - z^{-1}} = \frac{Tz}{z - 1} \tag{4.29}$$

Along the same lines, we could formulate the trapezoidal approximation of an integral as

$$y_k = y_{k-1} + \frac{(x_k + x_{k-1})}{2} \cdot T \quad \circ\!\!-\!\!\bullet \quad H_{intg}(z) = \frac{T}{2} \cdot \frac{z+1}{z-1} \tag{4.30}$$

The ideal integrator in the Laplace domain is $1/s$. Intuitively, we could claim that $1/s$ and $H_{intg}(z)$ describe the same functionality. Their equality provides an approximate relationship between s- and z-plane,

$$s = \frac{1}{H_{intg}(z)} = \frac{2}{T} \cdot \frac{z-1}{z+1} \tag{4.31}$$

Solving Eq. (4.31) for z leads to the first-order Padé approximation of a time delay (*cf.* Eq. (3.138) with $e^{-s\tau} = z^{-1}$). Converting transfer functions between the s- and the z-plane with Eq. (4.31) is known as the *bilinear transformation*. At the start of this chapter, we mentioned that a digital controller can be approximated by its time-continuous transfer function provided that the time lag T is very short compared to any dynamic response of the other system elements. As a rule of thumb, we can distinguish these three approaches:

- The sampling period of the digital system is two orders of magnitude shorter than the fastest time constant of the continuous system. In this case, the digital controller can be modeled as a continuous system, and Laplace-domain treatment is possible.
- When the sampling period of the digital system is faster than five to ten times of the fastest time constant of the continuous system, the bilinear transformation can be used to approximate Laplace-domain behavior. In this case, the z-domain transfer function of the controller is transformed into an s-domain approximation with Eq. (4.31), and the entire system is modeled in the Laplace domain. Care needs to be taken that all signals are band-limited, because the phase response of the dead-time delay $z = e^{sT}$ deviates strongly from its approximation at higher frequencies, and stability criteria such as the gain margin could become inaccurate in the approximation.
- When the sampling period of the digital system is less than five to ten times of the fastest time constant of the continuous system, the continuous components of the system are converted into their z-domain counterparts, and the entire system is modeled as a time-discrete system.

4.4 The *w*-Transform

Sometimes, it is desirable to know whether a pole in the z-plane is inside or outside of the unit circle, but the exact location of the pole is not important. s-plane methods can be applied when we define the w-transform as a mapping function that maps the unit circle of the z-plane onto the imaginary axis of a w-plane, and that maps any location of the z-plane inside the unit circle onto the left half-plane of the w-plane. Such a mapping function can be defined as

$$z = \frac{1+w}{1-w}; \quad w = \frac{z-1}{z+1} \tag{4.32}$$

Like s, w is a complex variable. However, the w-transform (Eq. (4.32)) is not an inversion of $z = e^{sT}$, and the s- and w-planes are not identical. However, any location z inside the unit circle (that is, $|z| < 1$) is mapped to a location in the w-plane with $\Re(w) < 0$. In some cases, the w-transform can provide a quick stability assessment of a z-domain transfer function. To examine a z-domain transfer function, for example,

$$H(z) = \frac{z-1}{z^2 + 2z + 0.5} \tag{4.33}$$

we take the denominator polynomial $q(z) = z^2 + 2z + 0.5$ and substitute z to obtain

$$\begin{aligned} q(w) &= \left(\frac{1+w}{1-w}\right)^2 + 2\frac{1+w}{1-w} + 0.5 \\ &= \frac{1}{2}\frac{-w^2 + w + 7}{w^2 - 2w + 1} \end{aligned} \tag{4.34}$$

If the roots of $q(z)$ lie inside the unit circle, the zeros of $q(s)$ lie in the left half-plane. In Eq. (4.34), we find zeros at -2.193 and $+3.193$, and we know that one pole of $H(z)$ lies outside of the unit circle and $H(z)$ is unstable. We make use of the w-transform in stability analysis (Chapter 10) and Bode analysis (Chapter 11).

4.5 Building Blocks for Digital Controllers

In analogy to Section 3.6, several low-order digital filters are introduced here. Without covering digital filter theory in detail, any digital filter can be seen as a weighted discrete sum of present and past input and output values as described by Eq. (4.35):

$$y_k = \sum_{n=1}^{N} a_n y_{k-n} + \sum_{m=0}^{M} b_m x_{k-m} \tag{4.35}$$

Two examples are the integrator formulas given in Eqs. (4.27) and (4.30). The first sum is referred to as the *recursive* part of the filter, and nonzero a_n give rise to infinite impulse response filters. The second sum represents the nonrecursive part. Filters with

$a_n = 0$ for all n are finite impulse response filters, because the impulse response returns to zero after no more than M discrete time steps T. The z-domain transfer function can be found by using the delay property of the z-transform, that is, a delay by one sampling interval T is represented by a multiplication with z^{-1} in the z-domain: when the z-transform of the sequence y_k is $Y(z)$, the delayed sequence y_{k-1} corresponds to $z^{-1}Y(z)$. The z-transform of Eq. (4.35) can therefore be written as

$$Y(z) - \sum_{n=1}^{N} a_n z^{-n} Y(z) = \sum_{m=0}^{M} b_m z^{-m} X(z) \tag{4.36}$$

By factoring out $X(z)$ and $Y(z)$, the transfer function of the digital filter becomes

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{-\sum_{n=0}^{N} a_n z^{-n}} \tag{4.37}$$

The negative sign in Eq. (4.37) is a result of moving the first summation term in Eq. (4.35) to the left-hand side. As a consequence we also get $a_0 = -1$.

Any recursive filter can lead to instability. As a general rule, **all poles of the z-domain transfer function must lie inside the unit circle in the z-plane**. We can demonstrate the rule with a simple first-order system,

$$y_k = a \cdot y_{k-1} + (1-a) \cdot x_{k-1} \tag{4.38}$$

Clearly, when $a > 1$, each subsequent value of y_k is larger than the previous one. The corresponding z-domain transfer function is

$$H(z) = \frac{1-a}{z-a} \tag{4.39}$$

and has one pole at $z = a$. Only for $a < 1$ is the stability requirement met.

4.5.1 Gain Block

The time-discrete analog of the gain block with gain g is realized by multiplying the current input value with g and applying it to the output. The only nonzero coefficient in Eq. (4.35) is $b_0 = g$ (and implicitly $a_0 = -1$).

- Time-domain function:

$$y_k = g \cdot x_k \tag{4.40}$$

- Z-domain transfer function:

$$H(z) = \frac{Y(z)}{X(z)} = g \tag{4.41}$$

- Impulse response:

$$y_{imp,k} = g \cdot \delta_k \tag{4.42}$$

In digital systems, the bit-discrete representation of numbers can play an important role, particularly when fast integer arithmetic is used. To provide an extreme example, assume an 8-bit value subjected to a gain of 1/64. The result only uses 2 bits. Rescaling this result back to 8 bits amplifies the digitization noise, and the output can only carry the discrete values 192, 128, 64, and 0. For integer implementations of digital filters, a careful implementation of scaling operations is needed that considers and avoids possible rounding errors.

4.5.2 Differentiator

The output signal of a differentiator approximates the first derivative of the input signal by applying a finite-difference formula. The finite difference can take three forms, backward, forward, and central difference. For real-time processing, only the backward difference can be realized as the other formulations depend on future input signals. Off-line processing allows using forward and central differences, and the central difference is attractive, because it does not introduce a phase shift.

- Time-domain function:

$$\begin{aligned} y_k &= \frac{x_k - x_{k-1}}{T} && \text{Backward difference} \\ y_k &= \frac{x_{k+1} - x_k}{T} && \text{Forward difference} \\ y_k &= \frac{x_{k+1} - x_{k-1}}{2T} && \text{Central difference} \end{aligned} \tag{4.43}$$

- Z-domain transfer function (backward difference):

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z-1}{Tz} \tag{4.44}$$

- Impulse response:

$$y_{imp,k} = \{1/T, -1/T, 0, 0, \ldots\} \tag{4.45}$$

4.5.3 Integrator

The integrator was briefly introduced in the previous section. Time-discrete integration involves summation of the input values. A recursive formulation, where the previous output value represents the previous sum of input values, yields the filter function. Similar to the differentiator, a forward, backward, and central (i.e., trapezoidal) approximation can be used. However, all three integrators are causal and do not depend on future values.

- Time-domain function:

$$\begin{aligned} y_k &= y_{k-1} + T \cdot x_{k-1} && \text{Forward rectangular approximation} \\ y_k &= y_{k-1} + T \cdot x_k && \text{Backward rectangular approximation} \\ y_k &= y_{k-1} + \frac{T}{2}(x_k + x_{k-1}) && \text{Trapezoidal approximation} \end{aligned} \tag{4.46}$$

- Z-domain transfer function (trapezoidal approximation):

$$H(z) = \frac{T}{2} \cdot \frac{z+1}{z-1} \tag{4.47}$$

- Impulse response (trapezoidal approximation):

$$y_{imp,k} = \{T/2, T, T, T, \ldots\} \tag{4.48}$$

4.5.4 PID Controller

It is straightforward to combine gain (k_p) with an integrator and a differentiator to obtain a simple *PID* formulation. By setting $k_D = 0$, the *PID* controller is reduced to a *PI* controller. In its simplest form, the combination of backward difference and backward rectangular integration yields.

- Time-domain function:

$$y_k = k_I y_{k-1} + \left(k_p + k_I T + \frac{k_D}{T}\right) x_k - \frac{k_D}{T} x_{k-1} \tag{4.49}$$

- Z-domain transfer function:

$$H(z) = \frac{z(k_p + k_I T + k_D/T) - k_D/T}{z - k_I} \tag{4.50}$$

To accelerate the computation, some coefficients would typically be calculated beforehand, such as $k_1 = k_p + k_I T + k_D/T$ and $k_2 = k_D/T$, which leads to a simpler form of Eq. (4.50):

$$H(z) = \frac{k_1 z - k_2}{z - k_I} \tag{4.51}$$

- Impulse response:

$$y_{imp,k} = \begin{cases} k_1 & \text{for} \quad k = 0 \\ k_1 \cdot k_I^k - k_2 \cdot k_I^{k-1} & \text{for} \quad k > 0 \end{cases} \tag{4.52}$$

An alternative formulation uses first-order finite differences in which the *change* of the output, $y_k - y_{k-1}$ depends on the change of the input $x_k - x_{k-1}$. By using the more complex (but more accurate) central difference and the trapezoidal rule, the *PID* controller can be described as follows:

- Time-domain function:

$$y_k - y_{k-1} = \left(k_p + \frac{k_I T}{2} + \frac{k_D}{T}\right) x_k - \left(k_p + \frac{2k_D}{T} - \frac{k_I T}{2}\right) x_{k-1} + \frac{k_D}{T} x_{k-2} \tag{4.53}$$

Once again it is convenient to precalculate the coefficients for x_k, x_{k-1}, and x_{k-2} as k_0, k_1, and k_2, respectively. With this definition, the time-domain function is simplified to

$$y_k - y_{k-1} = k_0 x_k - k_1 x_{k-1} + k_2 x_{k-2} \tag{4.54}$$

- Z-domain transfer function:

$$H(z) = \frac{k_0 z^2 - k_1 z + k_2}{z(z-1)} \tag{4.55}$$

- Impulse response:

$$y_{imp,k} = \{k_0, k_0 - k_1, k_0 - k_1 + k_2, k_0 - k_1 + k_2, \ldots\} \tag{4.56}$$

The superiority of the second formulation can also be seen in the fact that the impulse response converges to a finite value, which is the expected behavior of the *PID* system. The first formulation does not show convergent behavior.

4.5.5 Time-Lag System

Emulating a first-order time-lag system (first-order lowpass with a continuous-domain pole at $s = -|\sigma|$) in a time-discrete system is possible by directly using the correspondence tables for the step response function $f(t) = 1-e^{-\sigma t}$. We find the correspondence

$$f(t) = 1 - e^{-\sigma t} \quad \circ\!\!-\!\!\bullet \quad \frac{z}{z-1} \cdot \frac{1 - e^{-\sigma T}}{z - e^{-\sigma T}} \tag{4.57}$$

for the sampling rate T. The first term, $z/(z-1)$ represents the step input, while the second term is the actual filter transfer function. With the definition $a = e^{-\sigma T}$, we can now describe the filter:

- Time-domain function:

$$y_k = a y_{k-1} + (1-a) x_{k-1} \tag{4.58}$$

- Z-domain transfer function:

$$H(z) = \frac{1-a}{z-a} \tag{4.59}$$

- Impulse response:

$$y_{imp,k} = \{0, (1-a), (1-a)a, \ldots, (1-a)a^{k-1}\} \tag{4.60}$$

Note that $-1 < a < 1$ is required for stability.

4.5.6 Time-Lead System

In off-line digital signal processing and digital filters, a true phase-lead is achievable by accessing future input values x_{k+n}. For real-time processing, the idea of the first-order phase-lead system can be applied where the change (i.e., the first derivative) is added to the input signal. The idea behind this approach is to extrapolate the recent change to the present. The continuous equation $y(t) = x(t) + \tau_D \dot{x}(t)$ therefore leads the following digital filter (a is the normalized delay $a = \tau_D/T$):

- Time-domain function:

$$y_k = x_k + a(x_k - x_{k-1}) \tag{4.61}$$

- Z-domain transfer function:

$$H(z) = \frac{z(1+a) - a}{z} \tag{4.62}$$

- Impulse response:

$$y_{imp,k} = \{1 + a, -a, 0, 0, \ldots\} \tag{4.63}$$

4.5.7 Lead-Lag Compensator

We can combine the time-lead and time-lag compensators into a flexible lead-lag compensator that approximates the differential equation

$$\dot{y}(t) + ay(t) = bx(t) + \dot{x}(t) \tag{4.64}$$

where a and b determine the location of the pole and the zero, respectively, in the Laplace domain. A finite-difference approach is straightforward, with $(y_k - y_{k-1})/T$ on the left-hand side and $(x_k - x_{k-1})/T$ on the right-hand side. More elegantly, the differential Eq. (4.64) can be rearranged to require a discrete integration:

$$y(t) - x(t) = \int_0^t \big(bx(\tau) - ay(\tau)\big)\, \mathrm{d}\tau \tag{4.65}$$

The advantage of the form in Eq. (4.65) is that the more accurate trapezoidal integration can be applied instead of the backward difference. The discrete approximation of Eq. (4.65) can be written as

$$y_k - x_k = [y_{k-1} - x_{k-1}] + B(x_k + x_{k-1}) - A(y_k + y_{k-1}) \tag{4.66}$$

where $A = a \cdot T/2$ and $B = b \cdot T/2$. The two terms in square parentheses on the right hand side represent the accumulator for the integration; the following two terms are the trapezoidal finite areas added to the integral. Rearranging the terms yields the time-domain function and its z-transform:

- Time-domain function:

$$y_k = \frac{1}{1+A} \big[y_{k-1}(1-A) + x_k(1+B) + x_{k-1}(B-1)\big] \tag{4.67}$$

- Z-domain transfer function:

$$H(z) = \frac{1}{1+A} \cdot \frac{z(1+B) + B - 1}{z(1+A) + A - 1} \tag{4.68}$$

Because of the wide variety of configurations, the impulse response is not provided for this filter.

The above examples provide some digital filter functions that can be used as controllers or compensators in time-discrete systems. For the software implementation of a filter function, the time-domain equation would typically be used. To provide one example, the software implementation of Eq. (4.67) requires setting aside three memory locations to store the digital values for y_k, y_{k-1}, and x_{k-1}. Upon reset, these locations are initialized to zero. Furthermore, the coefficients A and B are computed from the given a, b, and T. In fact, it is even more convenient to compute $(1-A)/(1+A)$, $(B-1)/(1+A)$, and $(1+B)/(1+A)$ beforehand as can be seen in the listing below. Finally, a timer is started that causes the computation of Eq. (4.67) at each period T.

For this example, we assume that the conversion time of the ADC and the computation time are very short compared to T, meaning, the time between the start of the computation and the moment y_k is applied to the output is negligible. This assumption is usually not valid, and an additional time lag for conversion and processing time needs to be taken into account. We will examine this effect in more detail in Chapter 14.

The following steps (Algorithm 4.1) represent the pseudo-code to compute the filter output of the lead-lag compensator. This function is called in intervals of T. Since only immediate past values are stored, we use the shorthand `X` for x_k, `Y` for y_k, `Y1` for y_{k-1}, and `X1` for x_{k-1}.

Note that in line 2, `Y` is initialized and overwritten, whereas lines 3 and 6 add values to the existing `Y`. In lines 5 and 7, the old values for `X1` and `Y1` are updated and made available for the next sampling period. The ADC step in line 4 is not necessary if `X` is made available from outside this function.

Algorithm 4.1. Pseudocode algorithm to compute the output Y of a lead-lag compensator. This function needs to be called in regular intervals T, likely driven by a timer interrupt.

1 *Start of computation*;
2 $Y \longleftarrow Y1 \cdot (1-A)/(1+A)$;
3 $Y \longleftarrow Y + X1 \cdot (B-1)/(1+A)$;
4 Perform ADC and read X;
5 $X1 \longleftarrow X$;
6 $Y \longleftarrow Y + X \cdot (1+B)/(1+A)$;
7 $Y1 \longleftarrow Y$;
8 Apply Y to output;
9 *End of computation*;