5

Sampling and Sampled-Data Systems

5.1 Introduction

Today, virtually all control systems are implemented digitally, on platforms that range from large computer systems (e.g. the mainframe of an industrial SCADA[1] system) to small embedded processors. Since computers work with digital signals and the systems they control often live in the analog world, part of the engineered system must be responsible for converting the signals from one domain to another. In particular, the control engineer should understand the principles of sampling and quantization, and the basics of Analog to Digital and Digital to Analog Converters (ADC and DAC).

Control systems that combine an analog part with some digital components are traditionally referred to as *sampled-data systems*. Alternative names such as hybrid systems or cyber-physical systems (CPS) have also been used more recently. In this case, the implied convention seems to be that the digital part of the system is more complex than in traditional sampled-data systems, involving for example logic statements so that the system can switch between different behaviors and/or many distributed networked components. Another important concern for the development of CPS is system integration, since we must often assemble complex systems from heterogeneous components that might be designed independently using perhaps different modeling abstractions. In this chapter however we are concerned with the classical theory of sampled-data system and with digital systems that are assumed to be essentially dedicated to the control task and as powerful for this purpose as needed. Even in this case, interesting questions are raised by the capabilities of modern digital systems, such as the possibility of very high sampling rates. Moreover, we can build on this theory to relax its unrealistic assumptions for modern embedded implementation platforms and to consider more complex hybrid system and CPS issues, as discussed in later chapters.

5.1.1 Architecture of Sampled-Data Systems

The input signals of a digital controller consist of discrete sequences of finite precision numbers. We call such a sequence a digital signal. Often we ignore quantization (i.e., finite precision) issues and still call the discrete sequence a digital signal. In sampled-data systems, the plant to be controlled is an analog system (continuous-time and usually continuous-state) and measurements about the state of this plant that are initially in the analog domain need to be converted to digital signals. This conversion process from analog to digital signals is generally called sampling, although sampling can also refer to a particular part of this process, as we discuss below. Similarly, the digital controller produces digital signals, which need to be transformed to analog signals to actuate the plant. In control systems, this transformation is typically done by a form of signal holding device, most commonly a zero-order hold (ZOH) producing piecewise constant signals, as discussed in section 5.3. Fig. 5.1 shows a sampled-data model, i.e., the continuous plant together with the DAC and ADC devices, which takes digital input signals and produces digital output signals and can be connected directly to a digital controller. The convention used throughout these notes is that continuous-time signals are represented with full lines and sampled or digital signals are represented with dashed lines. Note that the DAC and ADC can be integrated for example on the microcontroller where the digital controller is implemented, and so the diagram

[1] SCADA = Supervisory Control and Data Acquisition

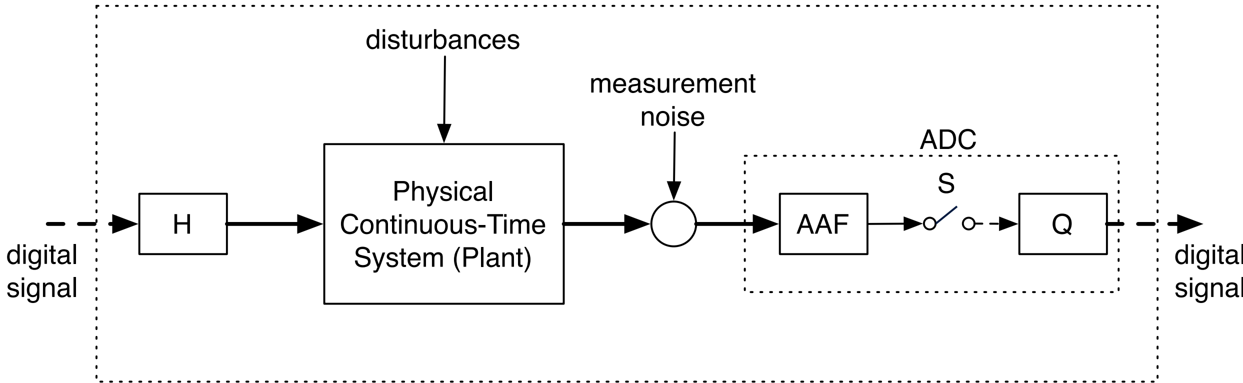


Figure 5.1: Sampled-data model. H = hold, S = sampler, Q = quantizer (+ decoder), AAF = anti-aliasing (low-pass) filter. Continuous-time signals are represented with full lines, discrete-time signals with dashed lines.

does not necessarily represent the physical configuration of the system. We will revisit this point later as we discuss more complex models including latency and communication networks. The various parts of the system represented on Fig. 5.1 are discussed in more detail in this chapter.

5.2 Sampling

5.2.1 Preliminaries

We first introduce our notation for basic transforms, without discussing issues of convergence. Distributions (e.g., Dirac delta) are also used informally (Fourier transforms can be defined for tempered distributions).

Continuous-Time Fourier Transform (CTFT): For a continuous-time function $f(t)$, its Fourier transform is

$$\hat{f}(\omega) = \mathcal{F}f(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt.$$

Inverting the Fourier transform, we then have

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega.$$

Laplace Transform: the Laplace transform generalizes the Fourier transform to a fonction defined on the whole complex plane rather than just the imaginary axis. In control however, we generally use the one-sided Laplace transform

$$\hat{f}(s) = \mathcal{L}f(s) = \int_{0^-}^{\infty} f(t) e^{-st} dt, \quad s \in \mathbb{C},$$

which is typically not an issue since we also assume that signals are zero for negative time. We can invert it using

$$f(t) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \hat{f}(s) e^{st} ds,$$

where c is a real constant that is greater than the real parts of all the singularities of $\hat{f}(s)$.

For G a CT LTI system (continuous-time linear time-invariant system) with matrices A, B, C, D, as usual and (in general matrix valued) impulse response $g(t) = D\delta(t) + Ce^{At}B$, its transfer function (a matrix in general) is

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

and is also denoted

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

Discrete-Time Fourier Transform (DTFT): for a discrete-time sequence $\{x[k]\}_k$, its Fourier transform is

$$\hat{x}(e^{i\omega}) = \mathcal{F}x(e^{i\omega}) = \sum_{k=-\infty}^{\infty} x[k]e^{-i\omega k}.$$

The inversion is

$$x[k] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{x}(e^{i\omega})e^{i\omega k}d\omega,$$

where the integration could have been performed over any interval of length 2π. Since we use both continuous-time and discrete-time signals, we use the term discrete-time Fourier transform (DTFT) for the Fourier transform of a discrete-time sequence. The notation should remind the reader that $\hat{x}(e^{i\omega})$ is periodic of period 2π (similarly in the following, we use $\hat{x}(e^{i\omega h})$, which is periodic of period $\omega_s = 2\pi/h$). It is sufficient to consider the DTFT of a sequence over the interval $(-\pi, \pi]$. The frequencies close to $0 + 2k\pi$ correspond to the low frequencies, and the frequencies close to $\pi + 2k\pi$ to the high frequencies. The theory of the DTFT is related to that of the Fourier *series* for the periodic function $\hat{x}(e^{i\omega})$.

z-transform and λ-transform: generalizing the DTFT for a discrete sequence $\{x[k]\}_k$, we have the two-sided z-transform

$$\hat{x}(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k}, \quad z \in \mathbb{C}.$$

In control however, we often use the one-sided z-transform

$$\hat{x}(z) = \sum_{k=0}^{\infty} x[k]z^{-k}, \quad z \in \mathbb{C},$$

but this not an issue, because the sequences are typically assumed to be zero for negative values of k. The z-transform is analogous to the Laplace transform, now for discrete-time sequences. It is often convenient to use the variable $\lambda = 1/z$ instead, and we call the resulting transform the λ-transform

$$\hat{x}(\lambda) = \sum_{k=-\infty}^{\infty} x[k]\lambda^{k}, \quad \lambda \in \mathbb{C}.$$

For G a DT LTI system (discrete-time linear time-invariant system) with matrices A, B, C, D, as usual and impulse response $\{g(k)\}$, its transfer function (a matrix in general) is

$$G(\lambda) = D + \lambda C(I - \lambda A)^{-1}B,$$

or

$$G(z) = D + C(zI - A)^{-1}B,$$

and also denoted

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

5.2.2 Periodic Sampling of Continuous-Time Signals

In sampled-data systems the plant lives in the analog world and data conversion devices must be used to convert its analog signals in a digital form that can be processed by a computer. We first consider the output signals of the plant, which are transformed into digital signals by an Analog-to-Digital Converter (ADC). For our purposes, an ADC consists of four blocks[2].

[2] this discussion assumes a "Nyquist-rate" ADC rather than an "oversampling" Delta-Sigma converter, see [Raz94].

1. First, an analog low-pass filter limits the signal bandwidth so that subsequent sampling does not alias unwanted noise or signal components into the signal band of interest. In a control system, the role of this *Anti-Aliasing Filter* (AAF) is also to remove undesirable high-frequency disturbances that can perturb the behavior of the closed-loop system. Aliasing is discussed in more details in the next paragraph. For the purpose of analysis, the AAF can be considered as part of the plant (the dynamics of the AAF can in some rare instances be neglected, see [ÅW97, p. 255]). Note that most analog sensors include some kind of filter, but the filter is generally not chosen for a particular control application and therefore might not be adequate for our purposes.

2. Next, the filter output is sampled to produce a discrete-time signal, still real-valued.

3. The amplitude of this signal is then quantized, i.e., approximated by one of a fixed set of reference levels, producing a discrete-time discrete-valued signal.

4. Finally, a digital representation is of this signal is produced by a decoder and constitutes the input of the processor. From the mathematical point of view, we can ignore this last step which is simply a choice of digital signal representation, and work with the signal produced by the quantizer.

Let $x(t)$ be a continuous-time signal. A sampler (block S on Fig. 5.1) operating at times t_k with $k = 0, 1, \ldots$ or $k = \ldots, -1, 0, 1, \ldots$, takes x as input-signal and produces the discrete sequence $\{x_k\}_k$, with $x_k = x(t_k)$. Traditionally, sampling is performed at regular intervals, as determined by the *sampling period* denoted h, so that we have $t_k = kh$. This is the situation considered in this chapter. We then let $\omega_s = \frac{2\pi}{h}$ denote the sampling frequency (in rad/s), and $\omega_N = \frac{\omega_s}{2} = \frac{\pi}{h}$ is called the Nyquist frequency. Note that we will revisit the periodic sampling assumption in later chapters, because it is hard to satisfy in networked embedded systems.

Aliasing

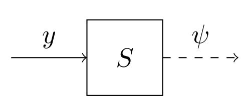
Sampling is a linear operation, but sampled systems in general are *not* time-invariant. Perhaps more precisely, consider a simple system HS, consisting of a sampler followed by a perfectly synchronized ZOH device. This system maps a continuous-time signal into another one, as follows. If u is the input signal, then the output signal y is

$$y(t) = u(t_k), \quad t_k \le t < t_{k+1}, \forall k,$$

where $\{t_k\}_k$ is the sequence of sampling (and hold) times. It is easy to see that it is linear and causal, but not time-invariant. For example, consider the output produced when the input is a simple ramp, and shift the input ramp in time. This system is in fact *periodic* of period h if the sampling is periodic with this period, in the sense that shifting the input signal u by h results in shifting the output y by h. Indeed, periodically sampled systems are often periodic systems.

Exercise 2. Assume that two systems H_1S_1 and H_2S_2 with sampling periods h_1 and h_2 are connected in parallel. For what values of h_1, h_2 is the connected system periodic?

In particular, sampled systems do not have a transfer function, and new frequencies are created in the signal by the process of sampling, leading to the distortion phenomenon known as *aliasing*. Consider a periodic sampling block S, with sampling period h, analog input signal y and discrete output sequence ψ, i.e., $\psi[k] = y(kh)$ for all k.



The following result relates the Fourier transforms of the continuous-time input signal and the discrete-time output signal. Define the periodic-extension of $\hat{y}(\omega)$ by

$$\hat{y}_e(\omega) := \sum_{k=-\infty}^{\infty} \hat{y}(\omega + k\omega_s),$$

and note that $\hat{y}_e$ is periodic of period ω_s, and is characterized by its values in the band $(-\omega_N, \omega_N]$.

Lemma 5.2.1. *The DTFT of $\psi = \{y(kh)\}_k$ and the CTFT of y are related by the relation*

$$\hat{\psi}(e^{i\omega h}) = \frac{1}{h}\hat{y}_e(\omega),$$

i.e.,

$$\hat{\psi}(e^{i\omega}) = \frac{1}{h}\hat{y}_e\left(\frac{\omega}{h}\right).$$

Proof. Consider the impulse-train $\sum_k \delta(t-kh)$ (or Dirac comb), which is a continuous-time signal of period h. The Poisson summation formula gives the identity

$$\sum_k \delta(t-kh) = \frac{1}{h}\sum_k e^{ik\omega_s t}.$$

Then define the impulse-train modulation

$$v(t) = y(t)\sum_k \delta(t-kh) = \frac{1}{h}\sum_k y(t)e^{ik\omega_s t}.$$

Taking Fourier transforms, we get

$$\hat{v}(\omega) = \frac{1}{h}\sum_k y(\omega + k\omega_s) = \frac{1}{h}\hat{y}_e(\omega).$$

On the other hand, we can also write

$$v(t) = \sum_k \psi(k)\delta(t-kh).$$

And taking again Fourier transforms

$$\hat{v}(\omega) = \int \left[\sum_k \psi(k)\delta(t-kh)\right] e^{i\omega t}dt = \sum_k \psi(k)e^{-i\omega kh} = \hat{\psi}(e^{i\omega h}).$$

□

In other words, periodic sampling results essentially in the periodization of the Fourier transform of the original signal y (the rescaling by h in frequency is not important here: it is just due to the fact that the discrete-time sequence is always normalized, with intersample distance equal to 1 instead of h for the impulse train). If the frequency content of y extends beyond the Nyquist frequency ω_N, i.e. $\hat{y}$ is not zero outside of the band $(-\omega_N, \omega_N)$, then the sum defining $\hat{y}_e$ involves more than one term in general at a particular frequency, and the signal y is distorted by the sampling process. On the other hand, if $\hat{y}$ is limited to $(-\omega_N, \omega_N)$, then we have $\hat{\psi}(e^{i\omega h}) = y(\omega)$ for the defining interval $(-\omega_N, \omega_N)$ and the sampling block does not distort the signal, but acts as a simple multiplicative block with gain $1/h$. The presence of an anti-aliasing low-pass filter, with *cut-off frequency* at ω_N, is then a means to avoid the folding of high signal frequencies of the continuous-time signals into the frequency band of interest due to the sampling process.

5.2.3 Sampling Noisy Signals

The previous discussion concerns purely deterministic signals and justifies the AAF to avoid undesired folding of high-frequency components in the frequency band of interest. Now, for a stochastic input signal y to the sampler, it is perhaps more intuitively clear that direct sampling is not a robust scheme, as the values of the samples become overly sensitive to high-frequency noise. Instead, a popular way of sampling a continuous-time stochastic signal is to integrate (or average) the signal over a sampling period before sampling

$$\psi[k] = \int_{t=(k-1)h}^{kh} y(\tau)d\tau,$$

which is in fact also form of (low-pass) prefiltering since we can rewrite

$$\psi[k] = (f * y)(t)|_{t=kh},$$

where

$$f(t) = \begin{cases} 1, & 0 \le t < h \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

This filter can be called an averaging filter or "integrate and reset" filter. In other words, the continuous-time signal $y(t)$ first passes through the filter that produces a signal $\bar{y}(t)$, in this case

$$\bar{y}(t) = \int_{(k-1)h}^{t} y(s)ds, \text{ for } (k-1)h \le t < kh.$$

Note that $\bar{y}((kh)^+) = 0, \forall k$, i.e., the filter resets the signal just after the sampling time. Also, even though the signal is reset, if we have access to the samples $\{\psi[k]\}_k$, we can immediately reconstruct the integrated signal $z(t) = \int_0^t y(s)ds$ at the sampling instants, since $z(kh) = \sum_{i=1}^{k} \psi[k]$, and hence the signal $y(t)$ at the sampling instants as well (since $y = \dot{z}$). In Section 5.5 we provide additional mathematical justification for the integrate and reset filter to sample stochastic differential equations.

Exercise 3. Compute the Fourier transform of f in (5.1) and explain why this is a form of low pass filtering.

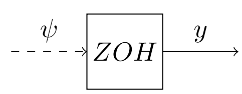
5.3 Reconstruction of Analog Signals: Digital-to-Analog Conversion

5.3.1 Zero-Order Hold

The simplest and most popular way of reconstructing a continuous-time signal from a discrete-time signal in control systems is to simply hold the signal constant until a new sample becomes available. This transforms the discrete-time sequence into a piecewise constant continuous-time signal[3], and the device performing this transformation is called a zero-order hold (ZOH). Let us consider the effect of the ZOH in the frequency domain, assuming an inter-sample interval of h seconds. Introduce

$$r(t) = \begin{cases} 1/h, & 0 \le t < h \\ 0, & \text{elsewhere.} \end{cases}$$

Then the relationship between the input and output of the zero-order hold device



can be written

$$y(t) = h \sum_k \psi[k]\, r(t - kh).$$

Taking Fourier transforms, we get

$$\begin{aligned} \hat{y}(\omega) &= h \sum_k \psi[k]\, \hat{r}(\omega) e^{-i\omega kh} \\ &= h\hat{r}(\omega)\hat{\psi}(e^{i\omega h}). \end{aligned}$$

Note that we can write

$$r(t) = \frac{1}{h} 1(t) - \frac{1}{h} 1(t-h),$$

and so we have the Laplace transform

$$\hat{r}(s) = \frac{1 - e^{-sh}}{sh}.$$

[3]This is again a mathematical idealization. In practice, the physical output of the hold device is continous.

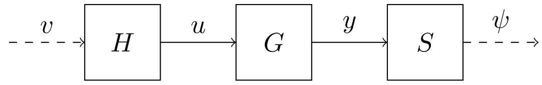


Figure 5.2: Step-invariant transformation for discretizing G. H here is a ZOH.

We then get the Fourier transform

$$\begin{aligned}\hat{r}(\omega) &= \frac{1-e^{-i\omega h}}{i\omega h} \\ &= e^{-i\omega h/2}\frac{\sin \omega \frac{h}{2}}{\omega \frac{h}{2}}.\end{aligned}$$

Note in particular that $\hat{r}(s) \approx e^{-sh/2}$ at low frequency, and so in this regime $\hat{r}$ acts like a time delay of $h/2$.

Lemma 5.3.1. *The Fourier transforms of the input and output signals of the zero-order hold device are related by the equation*

$$\hat{y}(\omega) = h\hat{r}(\omega)\hat{\psi}(e^{i\omega h}).$$

5.3.2 First-Order Hold

Other types of hold devices can be considered. In particular, we can try to obtain smaller reconstruction errors by extrapolation with high-order polynomials. For example, a first-order hold is given by

$$y(t) = \psi[k] + \frac{t-t_k}{t_k - t_{k+1}}(\psi[k]-\psi[k-1]), \;\; t_k \leq t < t_{k+1},$$

where the times t_k are the times at which new samples $\psi[k]$ become available. This produces a piecewise linear CT signal, by continuing a line passing through the two most recent samples. The reconstruction is still discontinuous however. Postsampling filters at the DAC can be used if these discontinuities are a problem.

One can also try a predictive first-order hold

$$y(t) = \psi[k] + \frac{t-t_k}{t_{k+1} - t_k}(\psi[k+1]-\psi[k]), \;\; t_k \leq t < t_{k+1},$$

but this scheme is non-causal since it requires $\psi[k+1]$ to be available at t_k. For implementing a predictive FOH, one can thus either introduce a delay or better use a prediction of $\psi[k+1]$. See [ÅW97, chapter 7] for more details on the implementation of predictive FOH. In any case, an important issue with any type of higher-order hold is that it is usually not available in hardware, largely dominated by ZOH DAC devices[4].

5.4 Discretization of Continuous-Time Plants

5.4.1 Step-Invariant discretization of linear systems

We can now consider the discrete-time system obtained by putting a zero-order hold device H and sampling block S at the input and output respectively of a continuous-time plant G, see Fig. 5.2. Let us assume that S is a periodic sampling block and H is a perfectly synchronized with S. The resulting discrete-time system is denoted $G_d = SGH$, and is called the *step-invariant transformation* of G. The name can be explained by the fact that unit steps are left invariant by the transformation, in the sense that

$$G_d S1 = G_d 1_d = SGH1_d = SG1,$$

where 1 and 1_d are the CT and DT unit steps respectively. Using Lemmas 5.2.1 and 5.3.1, we obtain the following result.

[4]There is also the possibility of adding an inverse-sinc filter and a low-pass filter at the back-end of the ZOH DAC, see [Raz94].

Theorem 5.4.1. *The CTFT $\hat{g}(\omega)$ of G and the DTFT of G_d obtained by a step-invariant transformation are related by the equation*

$$\hat{g}_d(e^{i\omega h}) = \sum_{k=-\infty}^{\infty} \hat{g}(\omega + k\omega_s)\hat{r}(\omega + k\omega_s),$$

or

$$\hat{g}_d(e^{i\omega}) = \sum_{k=-\infty}^{\infty} \hat{g}\left(\frac{\omega}{h} + k\omega_s\right)\hat{r}\left(\frac{\omega}{h} + k\omega_s\right).$$

Proof. To compute $\hat{g}_d$, we take v on Fig. 5.2 to be the discrete-time unit impulse, whose DTFT is the constant unit function. Then we have $\hat{u}(\omega) = h\hat{r}(\omega)$, thus $y = \hat{g}(\omega)\hat{u}(\omega)$, and the result follows from Lemma 5.2.1. □

Note that if the transfer function $\hat{g}(\omega)$ of the continuous-time plant G is band-limited to the interval $(-\omega_N, \omega_N)$, we have then

$$\hat{g}_d(e^{i\omega h}) = \hat{g}(\omega)\hat{r}(i\omega), \;\; -\omega_N < \omega < \omega_N,$$

and at low frequencies

$$\hat{g}_d(e^{i\omega h}) \approx \hat{g}(i\omega),$$

or somewhat more precisely, the ZOH essentially introduces a delay of $h/2$. Otherwise distortion by aliasing occurs, which can significantly change the frequency content of the system. This in particular intuitively justifies the presence of an anti-aliasing filter (AAF) with cutoff frequency $\omega_N < \omega_s/2$ at the output of the plant, if a digital control implementation with sampling frequency ω_s is expected. Indeed, aliasing can potentially introduce undesired oscillations impacting the performance of a closed-loop control system without proper AAF. The following example, taken from [ÅW97, chapter 1], illustrates this issue.

Example 5.4.1. Consider the continuous-time control system shown on Fig. 5.3. The plant in this case is a disk-drive arm, which can be modeled approximately by the transfer function

$$P(s) = \frac{k}{Js^2},$$

where k is a constant and J is the moment of inertia of the arm assembly. The arm should to be positioned with great precision at a given position, in the fastest possible way in order to reduce access time. The controller is implemented for this purpose and we focus here to the response to step inputs u_c. Classical continuous-time design techniques suggest a controller of the form

$$K(s) = M\left(\frac{b}{a}U_c(s) - \frac{s+b}{s+a}Y(s)\right) = M\left(\frac{b}{a}U_c(s) - Y(s) + \frac{a-b}{s+a}Y(s)\right), \tag{5.2}$$

where

$$a = 2\omega_0, \;\; b = \omega_0/2, \;\; \text{and } M = 2\frac{J\omega_0^2}{k},$$

and ω_0 is a design coefficient. The transfer function of the closed loop system is

$$\frac{Y(s)}{U(s)} = \frac{\frac{\omega_0^2}{2}(s + 2\omega_0)}{s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3}. \tag{5.3}$$

This system has a settling time to 5% equal to $5.52/\omega_0$. The step response of the closed-loop system is shown on Fig. 5.3 as well.

Let us now assume that a sinusoidal signal $n(t) = 0.1\sin(12\,t)$ of amplitude 0.1 and frequency 12 rad/s perturbs the measurements of the plant output. It turns out that this disturbance has little impact on the performance of the continuous-time closed-loop system, see the left column of Fig. 5.4. Then consider the following simple discretization of the controller (5.2). First, a continuous-time state-space realization of the transfer function $K(s)$ is realized by

$$u(t) = M\left(\frac{b}{a}u_c(t) - y(t) + x(t)\right)$$
$$\dot{x}(t) = -ax + (a-b)y.$$

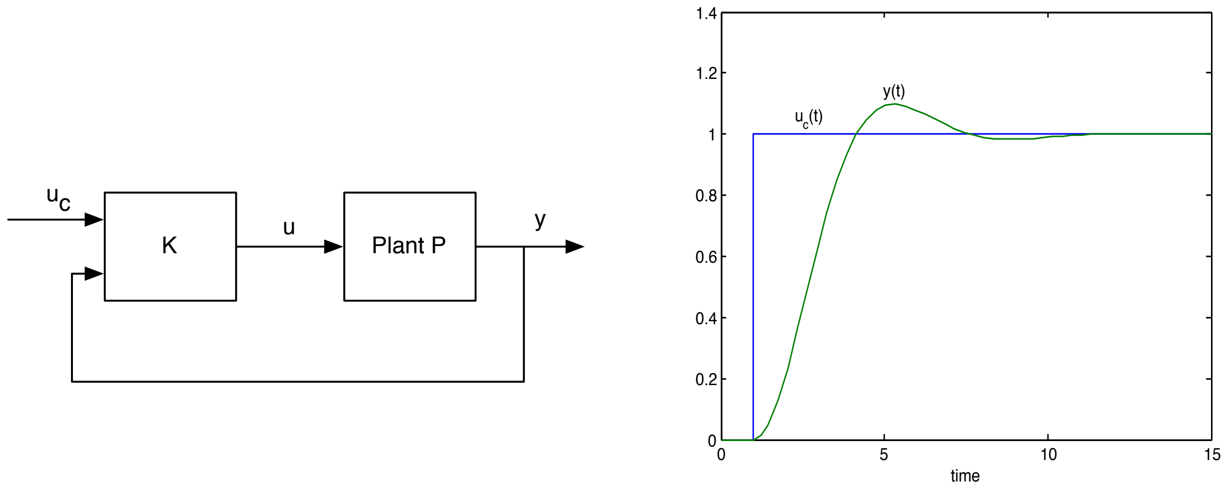


Figure 5.3: Feedback control system and step response for example 5.4.1 with $J = k = \omega_0 = 1$.

Next, we approximate the derivative of the controller state with a simple forward difference scheme (we will see soon that we could be more precise here, but this will do for now)

$$\frac{x(t+h) - x(t)}{h} = -ax(t) + (a-b)y(t),$$

where h is the sampling period. We then obtain the following approximation of the continuous-time controller

$$x[k+1] = (1-ah)x[k] + h(a-b)y[k],$$
$$u[k] = M\left(\frac{b}{a}u_c[k] - y[k] + x[k]\right),$$

where $u_c[k] = u_c(kh)$ and $y[k] = y(kh)$ are the sampled values of the input reference and (noise perturbed) plant output. A Simulink model of this implementation is shown on Fig. 5.5. As you can see on the right of Fig. 5.4, for a choice of sampling period $h = 0.5$ that is too large, there is a significant deterioration of the output signal due to the presence of a clearly visible low frequency component.

This phenomenon is explained by the aliasing phenomenon, as discussed above. We have $\omega_s = 2\pi/h = 2\pi/0.5 \approx 12.57$ rad/s, and the measured signal has a frequency 12 rad/s, well above the Nyquist frequency. After sampling, we then have the creation of a low frequency component with the frequency $12.57 - \omega_s = 0.57$ rad/s, in other word with a period of approximately 11 s, which is the signal observed on the right of Fig. 5.4.

Exercise 4. Derive (5.3).

Step-Invariant discretization of linear state-space systems

In this section, we consider a state-space realization of an LTI plant instead of working in the frequency domain as in the previous paragraph. It turns out that the step-invariant discretization of such a plant can be described *exactly* (i.e., without approximation error) by a discrete-time LTI system. This discrete-time system gives exactly the state of the continuous-time plant at the sampling instants. Although this process might not be able to detect certain hidden oscillations in the continuous-time system, it forms the basis of a popular design approach that consists in working only with the discretized version of the plant and using discrete-time control methods.

Hence suppose that we have a CT LTI system G with state-space realization

$$\dot{x} = Ax + Bu \tag{5.4}$$
$$y = Cx + Du, \tag{5.5}$$

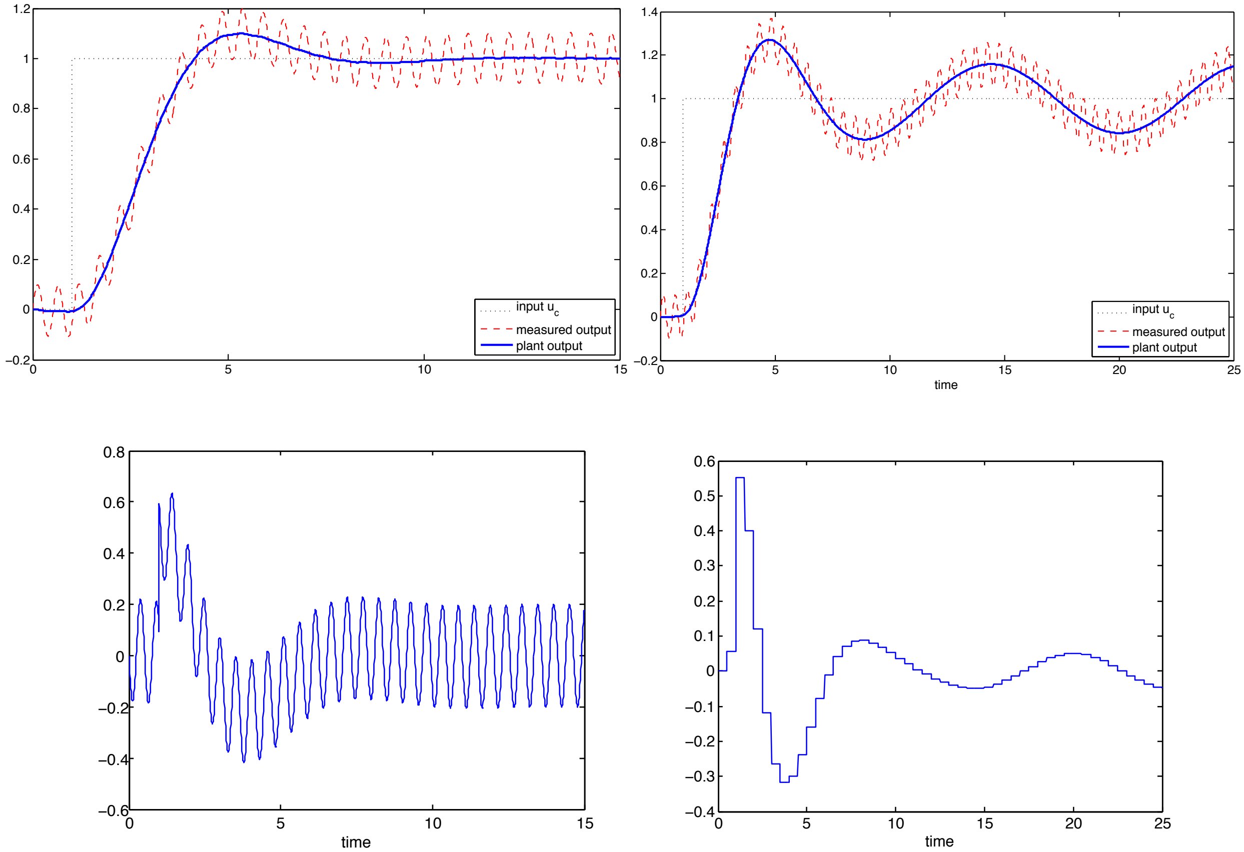


Figure 5.4: Effect of a periodic perturbation on the continuous-time design (left) and discretized design with sampling period 0.5 (right), for example 5.4.1. The bottom row shows the analog input u to the plant. The analog controller provides a significant action, whereas the digital controller does not seem to be able to detect the high-frequency measurement noise.

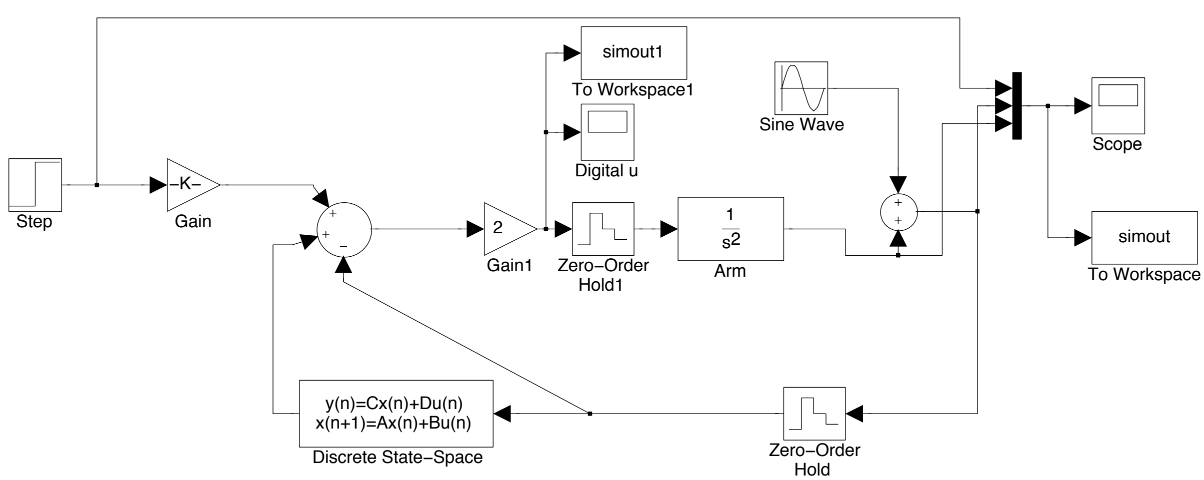


Figure 5.5: Simulink model for the digital implementation of example 5.4.1.

and consider the step-invariant transformation SGH, with sampling times $\{t_k\}_k$. The control input u to the CT plant in (5.4), (5.5) is then piecewise constant equal to $u[k]$ on $t_k \leq t < t_{k+1}$. We are also interested in the sampled values of the output $y[k] = y(t_k)$. Directly integrating the differential equation (5.4), we have

$$x(t) = e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-\tau)}Bd\tau \; u[k], \text{ for } t_k \leq t < t_{k+1}.$$

In particular, for a periodic sampling scheme with $t_{k+1} - t_k = h$,

$$x(t_{k+1}) = e^{Ah}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bd\tau \; u[k]$$
$$x(t_{k+1}) = e^{Ah}x(t_k) + \int_{0}^{h} e^{A\tau}Bd\tau \; u[k], \text{ for } t_k \leq t < t_{k+1}.$$

Writing $x[k] = x(t_k)$ for all k, the state of the plant at the sampling times can then be described by the following LTI difference equation

$$x[k+1] = A_d x[k] + B_d u[k],$$

with $A_d = e^{Ah}$ and $B_d = \int_0^h e^{A\tau}Bd\tau$. Note that this exact discretization could have been used in example 5.4.1 instead of the approximate Euler scheme. In summary, with *periodic sampling* the plant is represented at the sampling instants by the DT LTI system

$$x[k+1] = A_d x[k] + B_d u[k]$$
$$y[k] = Cx[k] + Du[k].$$

Remark 5.4.1. The matrix exponential, necessary to compute A_d, can be computed using the MATLAB `expm` funtion. There are various ways of computing B_d. The most straightforward is to simply use the MATLAB function `c2d` which performs the step-invariant discretization with a sampling period value provided by the user. In fact, this function also provides other types of discretization, including plant discretization using FOH, and other types of discretization for continuous-time controllers (rather than controlled plants), discussed later in this chapter.

A few identities are sometimes useful for computations or in proofs. First define

$$\Psi = \int_0^h e^{A\tau}d\tau = Ih + \frac{Ah^2}{2!} + \dots$$

Then we have $B_d = \Psi B$, and $A_d = I + A\Psi$. Moreover, we have the following result.

Lemma 5.4.2. *Let A_{11} and A_{22} be square matrices, and define for $t \geq 0$*

$$\exp\left\{t\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22}\end{bmatrix}\right\} = \begin{bmatrix} F_{11}(t) & F_{12}(t) \\ 0 & F_{22}(t)\end{bmatrix}. \tag{5.6}$$

Then $F_{11}(t) = e^{tA_{11}}$, $F_{22} = e^{tA_{22}}$, and

$$F_{12}(t) = \int_0^t e^{(t-\tau)A_{11}}A_{12}e^{\tau A_{22}}d\tau.$$

Using this lemma, one can compute A_d and B_d using only the matrix exponential function for example, by taking $t = h, A_{11} = A, A_{22} = 0, A_{12} = B$, so that $F_{11}(h) = A_d$ and $F_{12}(h) = B_d$.

Proof. The expressions for F_{11} and F_{22} are immediate since the matrices are block triangular. To obtain F_{12}, differentiate (5.6)

$$\frac{d}{dt}\begin{bmatrix} F_{11}(t) & F_{12}(t) \\ 0 & F_{22}(t)\end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22}\end{bmatrix}\begin{bmatrix} F_{11}(t) & F_{12}(t) \\ 0 & F_{22}(t)\end{bmatrix},$$

hence

$$\frac{d}{dt}F_{12}(t) = A_{11}F_{12}(t) + A_{12}F_{22}(t).$$

We then solve this differential equation, using the facts $F_{22}(t) = e^{tA_{22}}$ and $F_{12}(0) = 0$. □

5.4.2 Nonlinear Systems

5.4.3 Poles and Zeros of Linear Sampled-Data Systems

5.4.4 Incremental Models

5.4.5 Choice of Sampling Frequency

5.4.6 Generalized Sample and Hold Functions

5.5 Discretization of Linear Stochastic Differential Equations

Many models of dynamical systems involve differential equations containing white noise. The mathematically rigorous way to treat white noise is by using Brownian motion. However, an engineering approach will be sufficient for our purposes, with the caveat that certain formulas cannot really be justified and one needs to be careful with differential calculus since the rules familiar from the deterministic case do not necessarily apply.

Mathematically, it is also impossible to sample directly a signal containing white noise. Formally, the autocovariance function of a continuous-time zero-mean, vector valued white noise process $w(t) \in \mathbb{R}^d$ is a Dirac delta

$$E[w(t)w(t')^T] = r(t-t') = W\delta(t-t'), \tag{5.7}$$

where $W \in \mathbb{R}^{d \times d}$. In other words, the values of the signal at different times are uncorrelated, as for discrete-time white noise. The power spectral density (PSD) of w is defined as the Fourier transform of the autocovariance function

$$\phi_w(\omega) = \int_{-\infty}^{\infty} r(\tau) e^{-i\omega\tau} d\tau = \int_{-\infty}^{\infty} W\delta(\tau) e^{-i\omega\tau} d\tau = W,$$

hence W is called the *power spectral density matrix* of w. The frequency content of w is flat with infinite bandwidth. Hence, intuitively, according to the frequency folding phenomenon due to sampling, the resulting sampled signal would have infinite power in a finite band.

Mathematically rigorous theories for manipulating models involving white noise are usually developed by working instead with an integral version of white noise

$$B(t) = \int_0^t w(s)ds, \quad B(0) = 0.$$

One can then use these results to justify a posteriori the engineering formulas often formulated in terms of Dirac deltas such as (5.7)[5]. The stochastic process B has zero mean value and its increments $I(s,t) = B(t) - B(s)$ over disjoint intervals are uncorrelated with covariance

$$\mathbb{E}[(B(t)-B(s))(B(t)-B(s))^T] = |t-s|W.$$

For this reason, $W dt$ is sometimes called the incremental covariance matrix of the process B. The stochastic process B is called Brownian motion or Wiener process if in addition the increments have a Gaussian distribution. Note that in this case the increments over disjoint intervals are in fact independent, as a consequence of a well-known property of Gaussian random variables. We then called the corresponding white noise a (zero-mean) *Gaussian white noise.*

Gaussian white noise (or equivalently the Wiener process) is the most important stochastic process for signal processing and control system applications, in particular because one can derive from it other noise processes with a desired power spectral density by using it as an input to an appropriate filter (this is a consequence of spectral factorization, which you might want to review). The usefulness of this approach is that we can then only work with white noise and take advantage of the uncorrelated samples property to simplify computations.

Continuous-time processes with stochastic disturbances are thus often described by a stochastic differential equation, e.g., of the linear form

$$\frac{dx}{dt} = Ax + Bu + Gw(t), \tag{5.8}$$

[5] Another rigorous approach would be to work with white noise directly using the theory of distributions, but in general this is unnecessarily complicated and it is just simpler to use integrated signals.

where w is a zero-mean white noise process with power spectral density matrix W, which moreover we will assume to be Gaussian. Mathematically, it is more rigorous to write this equation in the incremental form

$$dx = (Ax + Bu)dt + GdB_1(t), \tag{5.9}$$

where B_1 is a Wiener process with incremental covariance Wdt. This equation is just a formal expression for the integral form

$$x(t) = x(0) + \int_0^t (Ax + Bu)dt + \int_0^t GdB_1(t),$$

where the last term is called a stochastic integral and can be rigorously defined. Similarly, white noise can be used to model measurement noise. In this case, instead of the form

$$y = (Cx + Du) + Hv, \tag{5.10}$$

where y is the measured signal and v is white Gaussian noise with power spectral density matrix V, we work with the integrated signal $z(t) = \int_0^t y(s)ds$, so that we can write again more rigorously

$$dz = (Cx + Du)dt + HdB_2(t), \tag{5.11}$$

where B_2 is a Wiener process with incremental covariance Vdt. It is assumed that the processes B_1 and B_2 are independent.

Now assume that the process (5.8), (5.10), or equivalently (5.9), (5.11), is sampled at discrete times $\{t_k\}_k$, and that we want as in the deterministic case to evaluate the values of x and z at the sampling times. Integrating (5.9) and denoting $h_k = t_{k+1} - t_k$, we get[6]

$$x(t_{k+1}) = e^{Ah_k}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bud\tau + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}GdB_1(\tau).$$

Assuming a fixed value $h_k = h$ for the intersample intervals and a ZOH, we get

$$x[k+1] = A_d x[k] + B_d u[k] + w[k],$$

where A_d and B_d are given by (**??**) as in the deterministic case, and the sequence $\{w[k]\}_k$ is a random sequence. This random sequence has the following properties, which come from the construction of stochastic integrals and are admitted here. First, the random variables $w[k]$ have zero mean

$$\mathbb{E}(w[k]) = \mathbb{E}\left(\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}GdB_1(\tau)\right) = 0,$$

and a jointly Gaussian distribution. Their variance is equal to

$$\begin{aligned}
\mathbb{E}(w[k]w[k]^T) &= \mathbb{E}\left(\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}GdB_1(\tau) \int_{t_k}^{t_{k+1}} dB_1^T(\tau')G^T e^{A^T(t_{k+1}-\tau')}\right) \\
&= \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}GWG^T\delta(\tau-\tau')e^{A^T(t_{k+1}-\tau')}d\tau d\tau' \\
&= \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}GWG^T e^{A^T(t_{k+1}-\tau)}d\tau.
\end{aligned} \tag{5.12}$$

Here we have used formal manipulations of Dirac deltas, but formula (5.12) is in fact a consequence of the so-called *Ito isometry*. Finally, essentially by the independent increment property of the Brownian motion, we have that variables $w[k]$ and $w[k']$ for $k \neq k'$ are uncorrelated (hence independent since they are Gaussian)

$$\mathbb{E}(w[k]w[k']) = 0, \quad k \neq k'.$$

[6] This can be admitted formally, by analogy with the deterministic case.

In other work, $\{w[k]\}_k$ is a discrete-time white Gaussian noise sequence with covariance matrix W_d given by (5.12), which we can rewrite

$$W_d = \int_0^h e^{At} G W G^T e^{A^T t} dt, \tag{5.13}$$

by the change of variables $u = t_{k+1} - \tau$. Note in particular that

$$W_d \approx G W G^T h \tag{5.14}$$

for h sufficiently small, which is a sometimes used approximation (but can be quite inaccurate [Far08, p. 142]). Higher order Taylor expansions can provide approximate analytical formulas that are useful for online computation in the time-dependent case [Far08, p. 144]. We can also compute the matrix (5.13) much more precisely using Lemma 5.4.2. Indeed, from (5.6), we have

$$\left[\exp\left\{h \begin{bmatrix} -A & GWG^T \\ 0 & A^T \end{bmatrix}\right\}\right]_{12} := M_{12} = \int_0^h e^{A(\tau - h)} G W G^T e^{A^T \tau} d\tau = A_d^{-1} W_d.$$

Hence we can compute W_d by first computing M, extracting its $(1,2)$ block M_{12}, and letting

$$W_d = A_d M_{12} = M_{22}^T M_{12}. \tag{5.15}$$

Let us now consider the sampling of the stochastic measurement process (5.11). Note first that we have

$$\bar{y}[k+1] = z(t_{k+1}) - z(t_k) = \int_{t_k}^{t_{k+1}} y(\tau) d\tau, \tag{5.16}$$

which corresponds physically to the fact mentioned earlier that the continuous-time random signal y containing high-frequency noise is not sampled directly but first integrated by the digital sensor[7]. Thus we have

$$\begin{aligned} &\bar{y}[k+1] = z(t_{k+1}) - z(t_k) \\ &= \left(\int_{t_k}^{t_{k+1}} C e^{A(t-t_k)} dt\right) x[k] + \left(\int_{t_k}^{t_{k+1}} \int_{t_k}^{t} C e^{A(t-\tau)} d\tau dt \, B + D h_k\right) u[k] + v[k], \\ &=: C_d x[k] + D_d u[k] + v[k], \end{aligned}$$

where

$$v[k] = \int_{t_k}^{t_{k+1}} \int_{t_k}^{t} C e^{A(t-\tau)} G dB_1(\tau) dt + H B_2(t_{k+1}) - H B_2(t_k).$$

Note first that the expressions for C_d, D_d can be rewritten

$$\begin{aligned} C_d &= \int_0^h C e^{At} dt \\ D_d &= \int_{t_k}^{t_{k+1}} \int_{\tau}^{t_{k+1}} C e^{A(t-\tau)} dt d\tau B + D h \\ &= \int_{t_k}^{t_{k+1}} \left(\int_0^{t_{k+1}-\tau} C e^{As} ds\right) d\tau B + D h \\ &=: \int_{t_k}^{t_{k+1}} \theta(t_{k+1} - \tau) d\tau B + D h \\ &= \int_0^h \theta(u) du \, B + D h. \end{aligned}$$

[7] Other forms of analog pre-filtering are possible and must be accounted for explicitly, by including a state-space model of the anti-aliasing filter. The simple "integrate and reset" filter discussed here (or "average and reset") is the most commonly discussed in the literature on stochastic systems however.

In the expressions above, we defined $\theta(t) := C\int_0^t e^{As}ds$. Similarly for $v[k]$ we have

$$v[k] = \int_{t_k}^{t_{k+1}} \theta(t_{k+1}-\tau)GdB_1(\tau) + HB_2(t_{k+1}) - HB_2(t_k).$$

Again $\{v[k]\}$ is a sequence of jointly Gaussian, zero-mean and independent random variables, i.e., a discrete-time Gaussian white noise. We can also immediately compute [Åst06, p.83]

$$\begin{aligned}\mathbb{E}(v[k]v[k]^T) &= \int_{t_k}^{t_{k+1}} \theta(t_{k+1}-\tau)GWG^T\theta(t_{k+1}-\tau)d\tau + HVH^Th \\ &= \int_0^h \theta(s)GWG^T\theta^T(s)ds + HVH^Th =: V_d \qquad (5.17)\end{aligned}$$

$$\mathbb{E}(w[k]v[k']^T) = \delta[k-k']\int_0^h e^{As}GWG^T\theta^T(s)ds =: \delta[k-k']S_d. \qquad (5.18)$$

Note in particular that the discrete samples $w[k]$ and $v[k]$ are not independent even if B_1 and B_2 are independent Wiener processes (or in engineering terms, w and v are independent continuous time white noise signals)!

Summary of the Formulas for the Discretization of Linear Stochastic Differential Equation

In summary, we obtain from the continuous time model (5.7), (5.10), after integration of the output and sampling, a stochastic difference equation of the form

$$x[k+1] = A_dx[k] + B_du[k] + w[k] \qquad (5.19)$$

$$\bar{y}[k+1] = C_dx[k] + D_du[k] + v[k] \qquad (5.20)$$

where

$$A_d = e^{Ah},\;\; B_d = \int_0^h e^{A\tau}Bd\tau,\;\; C_d = \int_0^h Ce^{A\tau}d\tau =: \theta(h),\;\; D_d = \int_0^h \theta(\tau)d\tau\; B + Dh.$$

One can verify that the covariance matrix of the discrete-time noise can be compactly expressed as

$$\mathbb{E}\left\{\begin{bmatrix} w[k] \\ v[k]\end{bmatrix}\begin{bmatrix} w[k] \\ v[k]\end{bmatrix}^T\right\} = \begin{bmatrix} W_d & S_d \\ S_d^T & V_d\end{bmatrix} = \int_0^h e^{\bar{A}t}\begin{bmatrix} GWG^T & 0 \\ 0 & HVH^T\end{bmatrix}e^{\bar{A}^Tt}dt, \qquad (5.21)$$

with

$$\bar{A} = \begin{bmatrix} A & 0 \\ C & 0\end{bmatrix} \Rightarrow e^{\bar{A}t} = \begin{bmatrix} e^{At} & 0 \\ C\int_0^t e^{A\tau}d\tau & I\end{bmatrix} = \begin{bmatrix} e^{At} & 0 \\ \theta(t) & I\end{bmatrix},$$

using (**??**). Lemma 5.4.2 also allows us to compute the matrices C_d, D_d and (5.21) using the matrix exponential only, following the method used to compute W_d in (5.15). The details are left as en exercise.

Time-delay in the Measurement Equation: Note that in (5.20), there is a delay in the measurement, in contrast to the standard form for DT systems

$$y[k] = C_dx[k] + D_du[k] + v[k].$$

Such a discrete-time delay is theoretically not problematic, since we can redefine the state as $\tilde{x}[k] = [x[k]^T, x[k-1]^T]^T$ and the measurement as $\tilde{y}[k] = \begin{bmatrix} 0 & C_d\end{bmatrix}\tilde{x}[k]$. Doubling the dimension of the state space has computational consequences however. In practice, this issue is handled in a less rigorous way, making a slightly non-causal approximation consisting in replacing $\bar{y}[k+1]$ by $\bar{y}[k]$ in (5.20).

Integrate and Reset vs. Average and Reset Filter: The formulas above are valid for the interpretation of the discrete-time measurement as an integral of the continuous-time measurement over the sampling period, see (5.16). This expression corresponds physically to an integrate and reset filter in the sensor. An equally popular assumption is that of an "average and reset" filter, namely

$$\hat{y}[k] = \frac{z(t_{k+1}) - z(t_k)}{h} = \frac{1}{h}\int_{t_k}^{t_{k+1}} y(t)dt, \qquad (5.22)$$

see, e.g., [GYAC10]. Note in (5.22) that we have again made the convenient non-causal approximation assuming that the discrete-time sample k contains the information of the continuous-time signal until t_{k+1}. If convention (5.22) is used, this results in a covariance matrix where (5.21) should be multiplied on the left and right by $\texttt{blkdiag}(I, I/h)$. In other words, we get $\hat{W}_d = W_d, \hat{V}_d = V_d/h^2, \hat{S}_d = S_d/h$. Moreover, $\theta(h) \approx C$ for h close to zero, instead of $\theta(h) \approx Ch$ as $h \to 0$ above. As a consequence of this choice however, the variance of the discrete-time measurement noise $v[k]$ diverges as $h \to 0$, and one should work with power spectral densities [GYAC10].

Units: If w has units denoted by κ, then its PSD has units κ^2/Hz (an energy per Herz) or equivalently $\kappa^2 \cdot s$. For example, if w is a scalar white noise perturbing the derivative of an acceleration signal, its PSD σ_w^2 has units $(m/s^3)^2 \cdot s = m^2/s^5$. After discretization, a noise w entering the right-hand side of the dynamics produces a discrete-time noise sequence $w[k]$, with covariance expressed in $\kappa^2 \cdot s^2$, see (5.14). In other words, $w[k]$ has the same units as $x[k]$, and the covariance of a discrete-time noise perturbing an acceleration is expressed in $(m/s^2)^2$. In the measurement equations, $v(t)$ has the same units as $x(t)$, say μ. Its PSD V has units μ^2/Hz or $\mu^2 \cdot s$. The covariance V_d has units $\mu^2 \cdot s^2$, see (5.17). Note however that $\hat{V}_d$ for the average and reset filter has units μ^2.

Covariance Approximations for small sampling period: As $h \to 0$, we have $W_d \approx GWG^T h$, $V_d \approx HVH^T h$, $S_d = O(h^2)$. For the averaging filter, $\hat{V}_d \approx HVH^T/h, \hat{S}_d = O(h)$. There are various numerical issues arising as we take $h \to 0$, not to mention the difficulty of interpreting the z-transform of models of the standard form (5.19), (5.20). For example, we always have $A_d \to I$ as $h \to 0$, no matter what A is. Numerically and intuitively more satisfying discrete-time models can be derived in a different form, the so-called delta-form, which is particularly useful for fast sampling rates, see [GYAC10].

Remark 5.5.1. The conceptual difficulties linked to the discretization of the measurement equation can be bypassed by providing a measurement model directly in discrete time. Such a model, involving continuous-time dynamics and discrete-time measurements, is sometimes called hybrid. Moreover, the issue of the correlation between the process and measurement noise in discrete-time is often conveniently ignored (one could design discrete-time filters taking this correlation into account, but the formulas are more complicated and not provided in the notes. The reader can find them for example in [AM05, Sim06]). This essentially means that the small sampling period approximation $S_d \approx 0$ is made.

5.6 Discretization of Analog Controllers

So far, we have mainly discussed the step-invariant discretization, which, given a sampler and hold device, produces from a continuous-time plant G and discrete-time system $G_d = SGH$. For linear systems, we have discussed the effect of this transformation in the frequency domain by establishing the relationship between the transfer functions of the G and G_d, including the potential creation of undesired frequency components by aliasing and the introduction of a perturbation term due to the hold device. We have also derived exact state-space recurrence equations for G_d.

The discrete-time G_d captures the behavior of G at the sampling times. One of the most common ways of designing digital controllers is then to perform the design in discrete-time directly using G_d, provided we somehow verify that nothing bad happens for the resulting closed-loop system between the sampling times. For example, we can design a discrete-time controller K_d that optimizes a discrete-time performance criterion for the discrete-time closed-loop system composed of G_d and K_d, see Chapter **??**. Note that optimizing a discrete-time performance criterion does not guarantee that the continuous-time behavior is satisfying in general however, see Example 5.10.1.

Continuing with our discussion of system discretization however, there are many situations where we have an analog controller design available, and we simply would like to obtain a digital implementation of it. This situation can result for example from the following facts

- for various reasons we prefer designing controller in continuous-time, e.g., because frequency domain reasoning is easier than for DT systems (PID controller design is usually discussed only in continuous-time), we do not have to worry about the choice of the sampling period at this stage (changing the sampling period changes the transfer function of the DT system) or about neglecting intersample behavior, and some calculations can be easier (.e.g. the CT Riccati equations are easier to handle).

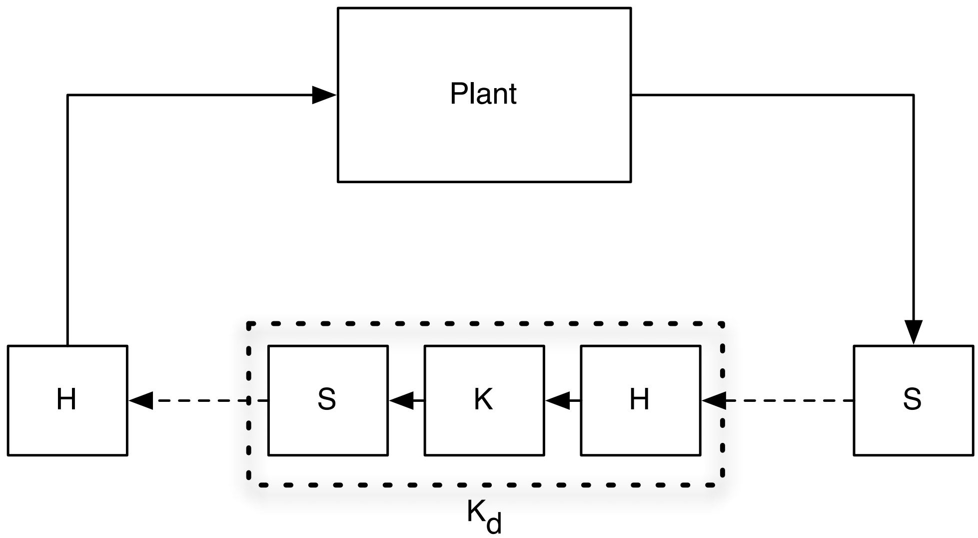


Figure 5.6: Step-Invariant Transformation for the Discretization of a Continuous-Time Controller Design. The blocks S and H that are directly connected operate at the same sampling times. The blocks S and H external to K_d correspond to physical devices, where the blocks S and H within K_d are mathematical operations producing $K_d = SKH$.

- an analog controller was inherited from a previous system implementation, and we do not wish to redesign it (for example, due to the cost of retesting and recertifying).

Assume therefore that a CT controller K is available, and we would like to derive from it a DT controller K_d, operating with the sampling period h. One way of doing this is to use again the step-invariant transformation and let $K_d = SKH$. Note however that here the operators S and H *do not correspond to actual physical devices*, they are just a mathematical representation of the discretization process. There is still however the physical barrier to the synchronicity assumption of these operators, coming from the computation time requirements of the processor. We continue to neglect this issue for now, but in general the mathematical H device at the input of K_d operates at the same instants as the physical S device it is connected to, and similarly at the output of K. Starting then from a state-space representation of K, we obtain then an implementation of K_d in terms of difference equations, as discussed in Section 5.4.1 (assuming K is linear).

For the plant, the discretization is dictated by the choice of sampling and hold device, and the use of the step-invariant transformation is essentially a consequence of these technological choices. On the controller side however, there is no such restriction and the step-invariant transformation is only one possible way of obtaining K_d from the continuous-time system K.

5.6.1 Bilinear Transformation

Another particularly common way of discretizing an analog controller is the bilinear transformation (also called Tustin's method). It is based on the trapezoidal approximation of integrals. Namely consider an integrator, i.e., a block with transfer function $1/s$, with input u and output y, over a sampling period

$$y((k+1)h) = y(kh) + \int_{kh}^{(k+1)h} u(\tau)d\tau.$$

We approximate this formula by

$$y((k+1)h) = y(kh) + \frac{h}{2}[u((k+1)h) + u(kh)].$$

The transfer function of this recurrence is

$$\begin{aligned}
\lambda^{-1}Y(\lambda) &= Y(\lambda) + \frac{h}{2}[\lambda^{-1}U(\lambda) + U(\lambda)] \\
\frac{Y(\lambda)}{U(\lambda)} &= \frac{h}{2}\frac{1+\lambda}{1-\lambda}.
\end{aligned}$$

This motivates the bilinear transformation

$$\begin{aligned}
\frac{1}{s} &\longleftrightarrow \frac{h}{2}\frac{1+\lambda}{1-\lambda} \\
s &\longleftrightarrow \frac{2}{h}\frac{1-\lambda}{1+\lambda}.
\end{aligned}$$

Using this change of variable, we can map a continuous-time transfer matrix $G(s)$ to a discrete-time transfer matrix $G_{bt}(\lambda)$, i.e.,

$$G_{bt}(\lambda) = G\left(\frac{2}{h}\frac{1-\lambda}{1+\lambda}\right).$$

In terms of state-space models, we can derive that starting with $G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$, we get $G_{bt}(\lambda) = \left[\begin{array}{c|c} A_{bt} & B_{bt} \\ \hline C_{bt} & D_{bt} \end{array}\right]$, with

$$\begin{aligned}
A_{bt} &= \left(I - \frac{h}{2}A\right)^{-1}\left(I + \frac{h}{2}A\right) \\
B_{bt} &= \frac{h}{2}\left(I - \frac{h}{2}A\right)^{-1} B \\
C_{ct} &= C(I + A_{bt}) \\
D_{dt} &= D + CB_{bt},
\end{aligned}$$

provided $2/h$ is not an eigenvalue of A. The mapping from s to λ is

$$\lambda = \frac{1 - hs/2}{1 + hs/2},$$

which maps the right half-plane into the unit disk.

Bilinear Transformation with Prewarping

5.6.2 Classical Software Implementation of a Digital Controller