ECE 310 Fall 2023

Lecture 33

Practical sampling and reconstruction of analog signals

Corey Snyder

Learning Objectives

After this lecture, you should be able to:

- Identify the practical considerations for ideal analog-to-digital conversion.
- Explain the practical issues with ideal digital-to-analog conversion.
- Demonstrate how a zero-order hold analog-to-digital conversion scheme provides a practical solution for reconstructing analog signals.

Recap from previous lecture

We concluded our discussion of digital rate conversion in the previous lecture and sampling as a whole for this course by covering upsampling. We saw how upsampling provides the complementary operation to downsampling since upsampling raises the implicit sampling rate while downsampling lowers the implicit sampling rate of a digital signal. In this lecture, we consider another key topic from this course through a more real-world perspective. We will discuss practical implementations of both analog-to-digital (ADC) and digital-to-analog conversion (DAC), with greater focus on practical DAC.

1 Practical analog-to-digital conversion

Recall from lecture 18 that we achieve ideal analog-to-digital conversion (ADC) by first multiplying, or sampling, a continuous-time signal $x(t)$ with an impulse train using sampling period T:

$$x_{\text{sampled}}(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT). \tag{1}$$

The second step is then to convert these sampled measurements into a discrete-time sequence $x[n]$. In this section, we want to briefly consider how we may actually achieve each of these two steps for analog-to-digital conversion.

Sample-and-hold. Multiplying $x(t)$ with an impulse train is the ideal mathematical realization of sampling. In practice, we may achieve a similar effect by rapidly charging a capacitor in the "sampling" stage, then holding this voltage across the capacitor while the sampled value is measured and encoded in the second step. We refer to such a system as a *sample-and-hold* circuit. Figure 1 depicts the sample-and-hold circuit. When the switch is instantaneously in the "sample" position, the capacitor charges to the instantaneous voltage of the signal $x(t)$. The switch then moves to the "hold" position where the capacitor now has no closed loop to discharge and we may store this measured value in computer memory. The design choices of capacitance, resistance, and other factors are best left to an analog or digital integrated circuits course.

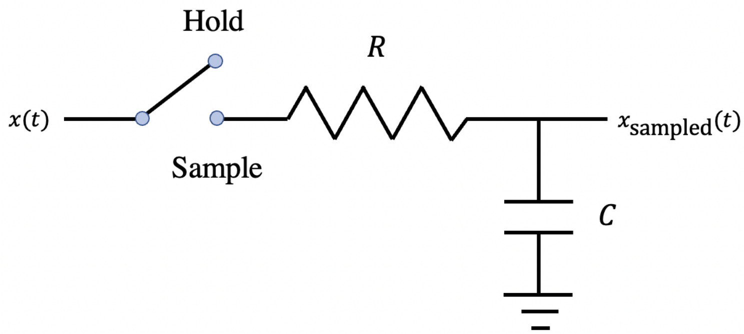


Figure 1: Sample-and-hold circuit.

Quantization. The second step stores the voltage measurements of the sample-and-hold circuit in computer memory as a discrete-time sequence. We will not discuss the physical circuitry/digital logic that performs the storing operation; however, we do note that this step requires the *quantization* of the measurements from $x_{\text{sampled}}(t)$. Our computing systems have finite precision and thus cannot represent all of the infinite measurable voltages. Thus, we must round measurements to a pre-defined set of values or *quantization levels*. This rounding incurs *quantization error* between the true and quantized values. For most modern-day computing systems, quantization errors are small enough that we often do not need to worry about accounting for them. In this course, we do not consider the finite-precision of digital signals and instead treat stored values as being effectively continuous as this approximation creates no significant errors in our analysis.

2 Practical digital-to-analog conversion

We derived the ideal digital-to-analog conversion (DAC) formulas in lecture 19. In the time-domain, we had the sinc interpolation formula for reconstructing bandlimited signal $x(t)$ from its discrete-time samples $x[n]$:

$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\text{sinc}\left(\frac{\pi(t-nT)}{T}\right). \tag{2}$$

However, there are two issues with sinc interpolation. First, we see that the sinc interpolation formula expressed by Eqn. 2 is a non-causal system since each sinc function extends infinitely far into the past and future around a given sample. This is clearly a problem if we want to recover analog signals in real-time, e.g. playing digitally-stored music. Second, this system is also not stable since the sinc function is not absolutely integrable. Note that this is similar to our statement that an ideal low-pass filter is not BIBO stable. Next, we will introduce a practical interpolation scheme given by the zero-order hold.

2.1 Zero-order hold DAC

Recall that digital-to-analog conversion can be expressed via the general interpolation formula:

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]g(t-nT), \tag{3}$$

where $g(t)$ is a continuous-time interpolation function. We will use $x_r(t)$ to denote the recovered signal from interpolation since we will no longer guarantee it is the same as the originally sampled $x(t)$. In ideal DAC, $g(t)$ is a sinc function and $x_r(t) = x(t)$. However, we would now like a $g(t)$ that is (1) causal and (2) absolutely integrable, thus yielding a BIBO stable system. One popular choice is the *zero-order hold* (ZOH). Note that the *Manolakis and Ingle* text refers to ZOH as sample-and-hold as well; however, we will use the term ZOH since it is popular elsewhere and helps us distinguish the two concepts. The term "zero-order" comes from the fact we are interpolating a constant, or zero-order, polynomial.

Let $g_{\text{zoh}}(t)$ be the ZOH interpolation function. We define $g_{\text{zoh}}(t)$ as

$$g_{\text{zoh}}(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{else} \end{cases}. \tag{4}$$

The corresponding CTFT of $g_{\text{zoh}}(t)$, $G_{\text{zoh}}(\Omega)$, is then

$$G_{\text{zoh}}(\Omega) = Te^{-j\frac{\Omega T}{2}} \operatorname{sinc}\left(\frac{\Omega T}{2}\right). \tag{5}$$

Figure 2 shows the result of ZOH interpolation in the time-domain. The interpolated signal is a piecewise-constant approximation of $x(t)$. Clearly, we have introduced some distortions in the recovered signal. We can also see this in the frequency domain. Recall from lecture 19 that the resulting CTFT spectrum from Eqn. 3 is given by

$$X_r(\Omega) = X_s(\Omega)G_a(\Omega) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X_a\left(\Omega - n\frac{2\pi}{T}\right)G_a(\Omega), \tag{6}$$

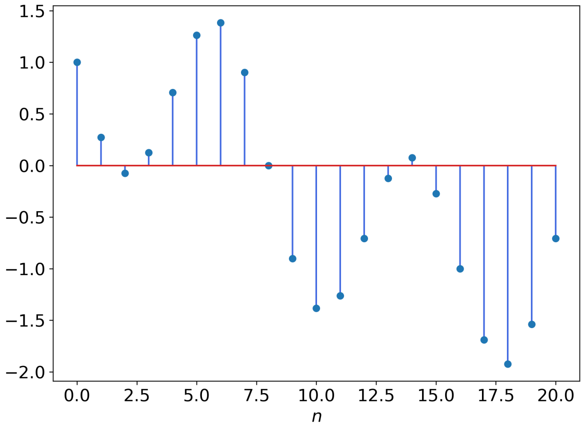
where $X_r(\Omega)$ is the CTFT of the recovered signal, $X_a(\Omega)$ is the CTFT of the original sampled signal $x(t)$, $X_s(\Omega)$ is the CTFT of $x_{\text{sampled}}(t)$, and $G_a(\Omega)$ is the CTFT of the interpolation function. With ideal sinc interpolation, the CTFT of $G_a(\Omega)$ was an ideal low-pass filter that extracted only the center spectral copy in $X_s(\Omega)$ which perfectly replicated $X_a(\Omega)$. Figure 3 shows how $G_{\text{zoh}}(\Omega)$ does not perfectly extract the central copy from $X_s(\Omega)$. Moreover, we see that the outer regions of the central copy are pushed down by $G_{\text{zoh}}(\Omega)$. This phenomenon is sometimes referred to as *droop*.

To correct these issues with ZOH interpolation, we introduce one additional system to the practical DAC process: the *reconstruction* or *compensation filter*. Let $h_r(t)$ denote the impulse response of the reconstruction filter. The CTFT of $h_r(t)$, $H_r(\Omega)$ is

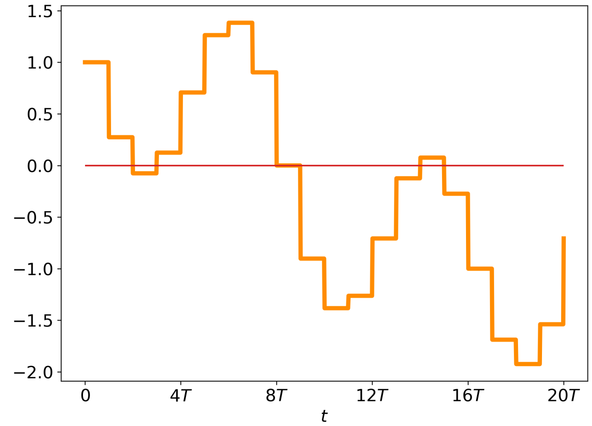
$$H_r(\Omega) = \begin{cases} \frac{1}{\operatorname{sinc}\left(\frac{\Omega T}{2}\right)}, & |\Omega| < \frac{\pi}{T} \\ 0, & |\Omega| \geq \frac{\pi}{T} \end{cases}. \tag{7}$$

The design of $H_r(\Omega)$ is meant to accomplish two things. First, remove any remaining contributions of spectral copies to the left or right of the central spectrum. Second, undo the droop caused by $G_{\text{zoh}}(\Omega)$. The support of $H_r(\Omega)$ achieves the first goal as it is an ideal low-pass filter. Of course, in practice, we can only hope to have a sufficiently good approximation of this ideal low-pass filter structure in our analog filter. Thus, while we would like zero transition bandwidth, we allow any non-zero transition bandwidth that prevents the additional spectral copies from appearing in the recovered signal.

The reconstruction filter addresses the second point by rescaling the frequencies of the central spectral copy to correct the droop phenomenon. We do not attempt to offset the $e^{-j\Omega T/2}$ factor from $G_{\text{zoh}}(\Omega)$ since this would make $H_r(\Omega)$ non-causal. Figure 3 depicts $H_r(\Omega)$ and the entire signal path from $X_s(\Omega)$ to $X_r(\Omega)$ to the final result $X_a(\Omega)$. Figure 4 summarizes the practical DAC system we have described using ZOH.

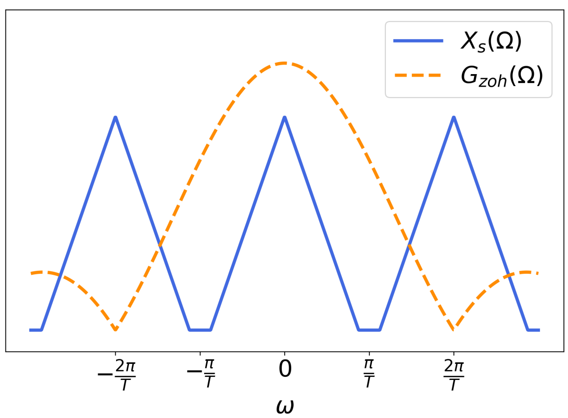


(a) $x[n]$

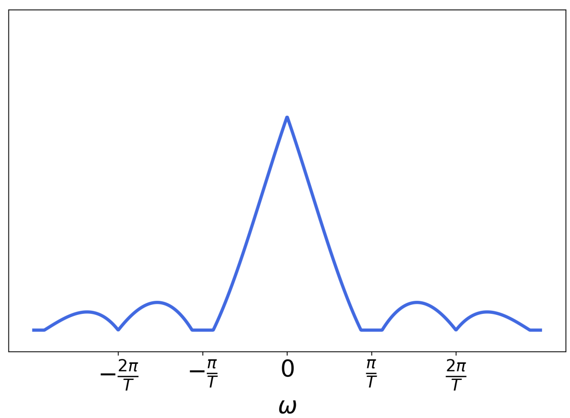


(b) $x_r(t) = \sum_{n=-\infty}^{\infty} x[n] g_{\mathrm{zoh}}(t - nT)$

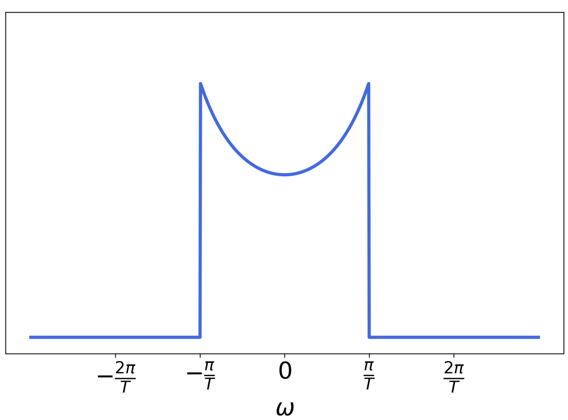
Figure 2: Zero-order hold in the time-domain.



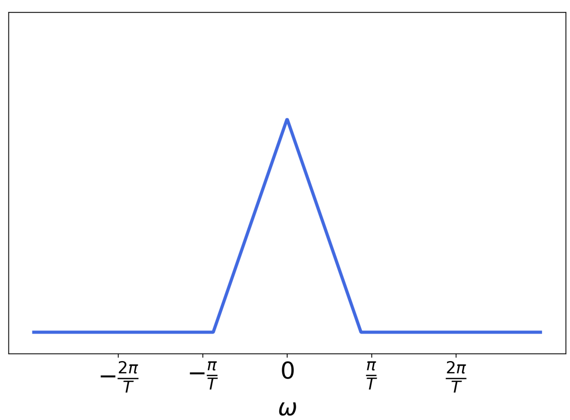
(a) $|X_s(\Omega)|$ and $|G_{\mathrm{zoh}}(\Omega)|$



(b) $|X_r(\Omega)|$



(c) $|H_r(\Omega)|$



(d) $|X_a(\Omega)|$

Figure 3: Effects of zero-order hold interpolation in the frequency domain.

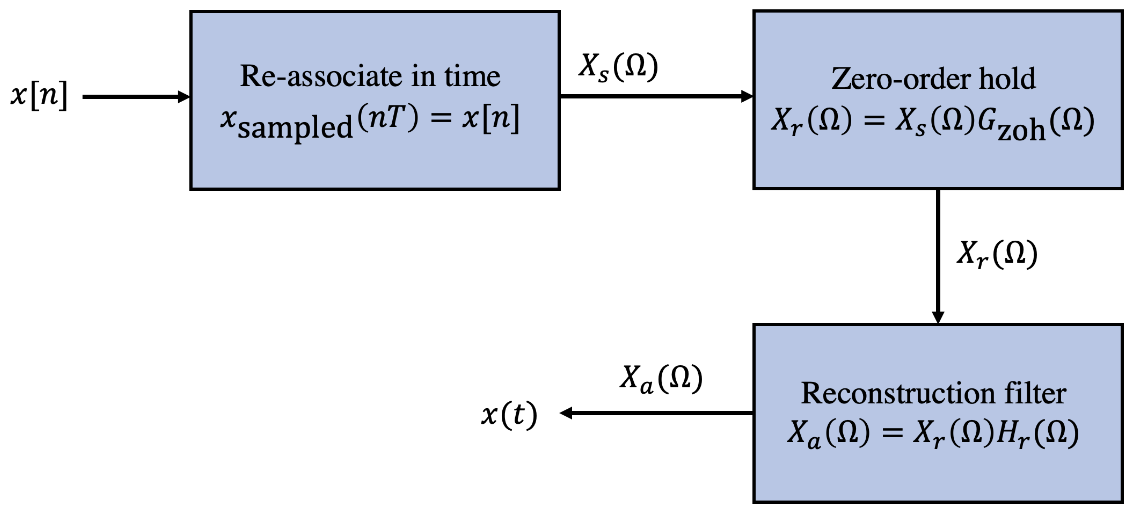


Figure 4: Summary of the signal path for practical digital-to-analog conversion via the zero-order hold.