ECE 310 Fall 2023

Lecture 31

Downsampling and decimation

Corey Snyder

Learning Objectives

After this lecture, you should be able to:

- State how downsampling by a given integer factor changes a signal in the time-domain and frequency-domain.
- Explain the importance of an anti-aliasing filter in a downsampling system.
- Understand why downsampling and digital rate conversion in general is an important application of digital signal processing.

Recap from previous lecture

The previous lectures explored finite impulse response (FIR) digital filters with particular focus on linear phase filters. We introduced the four types of linear phase filters and the window method for FIR filter design. In this lecture, we turn our attention to the concept of rate conversion in discrete-time systems. Over these two lectures, we will motivate the need for increasing or decreasing the implicit sampling rate of a digital signal and derive the necessary math to work with the two fundamental operations: downsampling and upsampling.

1 Digital rate conversion

Many digital systems require for data to be reduced or increased in size. Audio processing, digital communications, image processing, machine learning, and many more application fields all rely on *digital rate conversion* operations. For example, consider a piece of music you may have stored in a file on your computer. Suppose you would like to perform spectral analysis on this audio, but there is too much data to efficiently run your algorithms. If you know the music was sampled well above the Nyquist rate (as most music is), you could theoretically convert the digital signal back to analog, then resample at a slower rate to reduce the size of the audio file. While effective, this procedure requires quite a few steps and conversion from digital-to-analog and back.

Instead, we can achieve the same operation by working directly with the digital signal using one of the key rate conversion operations: downsampling. The two fundamental operations for digital rate conversion are *downsampling* and *upsampling*. In this lecture, we will carefully define downsampling and discuss upsampling in the following lecture. We will also make sure to elaborate on the practical benefits of both operations.

2 Downsampling

2.1 Downsampling and the time-domain

Downsampling, or *decimation*, is the process reducing the number of samples in a discrete-time signal. We associate an integer downsampling factor, often denoted by D, to describe the fraction of samples we keep after decimation. Let $x[n]$ be a discrete-time signal and $y[n]$ denote the signal we obtain after downsampling $x[n]$ by a factor of D. We can express $y[n]$ as

$$y[n] = x[Dn]. \tag{1}$$

In words, we keep every D'th sample from $x[n]$ to form $y[n]$. For example, if $D = 5$, we keep every 5th sample at $x[0]$, $x[5]$, $x[10]$, and so on.

Downsampling has the effect of reducing the implicit sampling rate of our discrete-time signal. Consider if we sample a piece of audio given by $x(t)$ for 5 seconds at $f_s = 1$ kHz to obtain $x[n]$. The discrete-time signal $x[n]$ will have 5,000 samples that follow the relation

$$\begin{align} x[n] &= x(nT) \tag{2} \\ &= x\left(\frac{n}{1000}\right),\ 0 \leq n < 5{,}000. \tag{3} \end{align}$$

If we downsample $x[n]$ by $D = 5$ to obtain $y[n]$, we will have $y[n]$ is length-1000 and correspond to $x(t)$ as follows:

$$\begin{align} y[n] &= x[5n] \tag{4} \\ &= x(5nT) \tag{5} \\ &= x\left(\frac{n}{200}\right) \tag{6} \\ &= x(nT_D). \tag{7} \end{align}$$

We see then that downsampling $x[n]$ by a factor of 5 generates the same signal as if we had sampled $x(t)$ with sampling period $T_D = 5T = \frac{5}{f_s}$. Thus, downsampling by integer factor D reduces the associated sampling rate of a discrete-time signal by D. This means when performing digital-to-analog conversion, we would need to use a slower sampling rate to correctly recover the original analog signal $x(t)$.

2.2 Downsampling and the frequency-domain

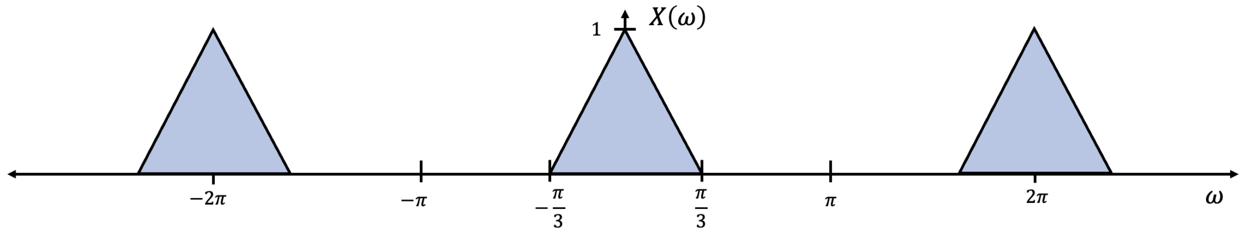
We must also consider how downsampling will change the frequency spectrum of a discrete-time signal. Let $X(\omega)$ denote the DTFT of the original discrete-time signal $x[n]$. Downsampling $x[n]$ by factor D to obtain $y[n]$ will yield the DTFT $Y(\omega)$ given by

$$Y(\omega) = \frac{1}{D}\sum_{k=0}^{D-1} X\left(\frac{\omega - 2\pi k}{D}\right). \tag{8}$$

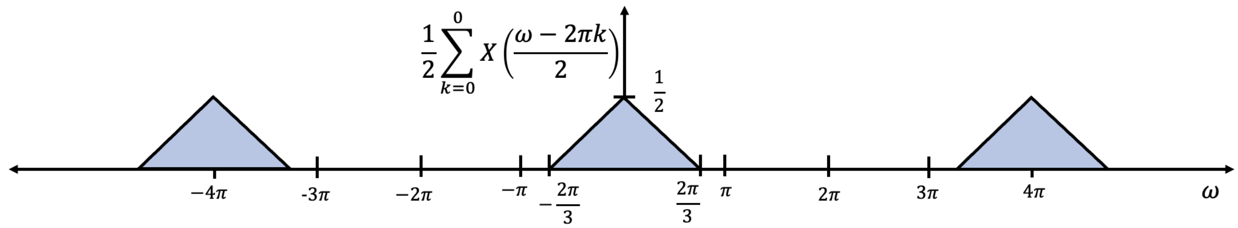
The derivation for $Y(\omega)$ is fairly involved, so we will focus instead on how to interpret and apply Eqn. 8. First, note that $Y(\omega)$ has height that is D times lower than $X(\omega)$. Looking inside the parentheses, we see that downsampling rescales the frequency axis by a factor of D. More specifically, downsampling *stretches* the frequency axis by a factor of D. For example, the first spectral copy of $X(\omega)$ centered at $\omega = 2\pi$ now lies at $2\pi D$ for $Y(\omega)$. The summation in Eqn. 8 is understandably confusing. Visualizing an example can be quite helpful as we can iterate through the values of k in the summation.

Figure 1 provides an example of downsampling by D=2. The middle plot shows the first step of rescaling the DTFT of $X(\omega)$ by $1/D$ and stretching the frequency axis by a factor of D as well. This first step is also equivalent to plotting the summation in Eqn. 8 at only $k = 0$. In the second step, we incorporate the shifted copies of the spectrum for when $k = 1$. Note how these added copies are simply at shifted by multiples of 2π. We perform the shifting and copying between Figs. 1b and 1c $D - 1$ times, which would be one time in this example.

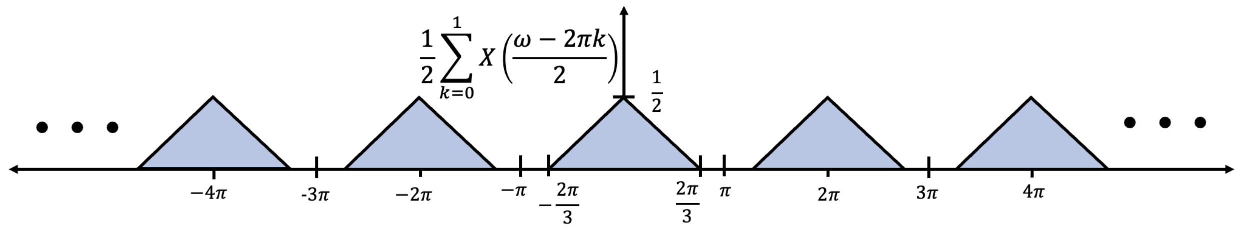
In general, we may simplify this procedure instead by stretching each spectral copy *in-place* and reducing the height by $1/D$. In other words, stretch each copy centered about $2\pi k$ without shifting its center like how we do with the central copy. This is equivalent to the procedure that Eqn. 8 describes.



(a) Original DTFT $X(\omega)$



(b) Partial DTFT of $Y(\omega)$ with only $k = 0$ term.



(c) Final DTFT of $Y(\omega)$ including $k = 0$ and $k = 1$ terms.

Figure 1: Example of how downsampling $X(\omega)$ by D=2 affects the DTFT of the downsampled signal $y[n]$. Plot (a) shows the original spectrum. Plot (b) shows the result of first taking the $k = 0$ term in the summation for the DTFT of $Y(\omega)$. Finally, plot (c) shows the end result of adding the terms at $k \in \{0, 1\}$.

2.3 Anti-aliasing filter

The example in Fig. 1 shows how downsampling stretches the DTFT and its spectral copies. We see in Fig. 1a the maximum frequency of the central copy lies at $\frac{\pi}{3}$. After downsampling by D=2, the maximum frequency expands to $2 \cdot \frac{\pi}{3} = \frac{2\pi}{3}$. Consider if we choose D=4. The maximum frequency of $\frac{\pi}{3}$ would now stretch to $\frac{4\pi}{3}$ outside the $[-\pi, \pi]$ limits of the central copy! This is an example of how downsampling may cause aliasing. This makes sense given how we demonstrated that downsampling lowers the implicit sampling rate of a discrete-time signal. Thus, we may downsample such that the sampling rate lowers below the Nyquist rate.

This motivates the use of an *anti-alising filter* when downsampling a digital signal. We want to ensure that no frequencies in our signal alias after downsampling. We know that downsampling stretches the frequency spectrum by a factor of D; thus, we want to ensure that the signal we downsample contains no frequencies past $\frac{\pi}{D}$. Let $h_D[n]$ denote the impulse response of the anti-aliasing filter and $H_D(\omega)$ be its DTFT. The DTFT of $H_D(\omega)$ is then

$$H_D(\omega) = \begin{cases} 1, & |\omega| \leq \frac{\pi}{D} \\ 0, & |\omega| > \frac{\pi}{D} \end{cases}. \tag{9}$$

The anti-aliasing filter is an ideal low-pass filter with cutoff frequency $\frac{\pi}{D}$. This cutoff frequency will guarantee all frequencies lie within the $[-\pi, \pi]$ range after downsampling by factor D. *It is important to remember that we apply an anti-aliasing filter before downsampling.* Figure 2 illustrates the decimation system to downsample $x[n]$ into output signal $y[n]$. The input signal is first processed by the anti-aliasing filter before downsampling by D to obtain $y[n]$. Finally, Fig. 3 shows an example application of a decimation system with and without an anti-aliasing filter for a signal with maximum frequency $\frac{\pi}{2}$ and downsampling factor

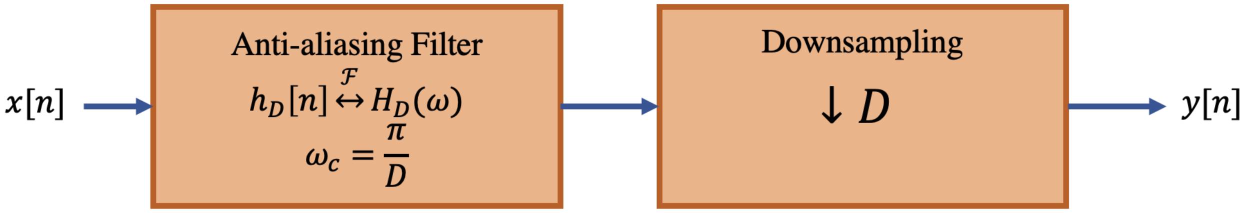
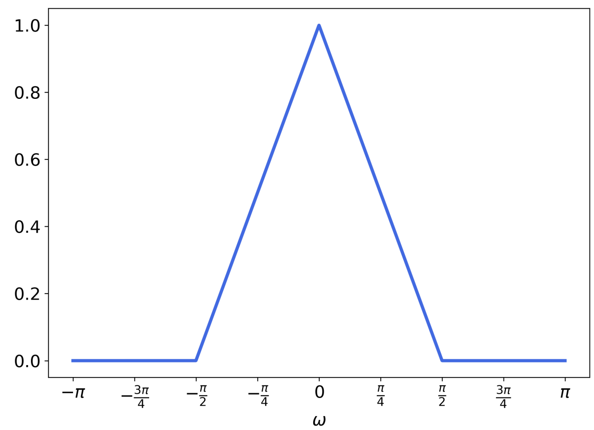
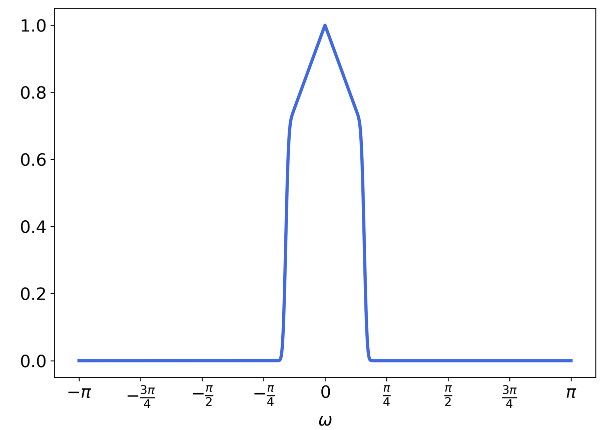


Figure 2: Decimation system including anti-aliasing filter followed by downsampling operation.

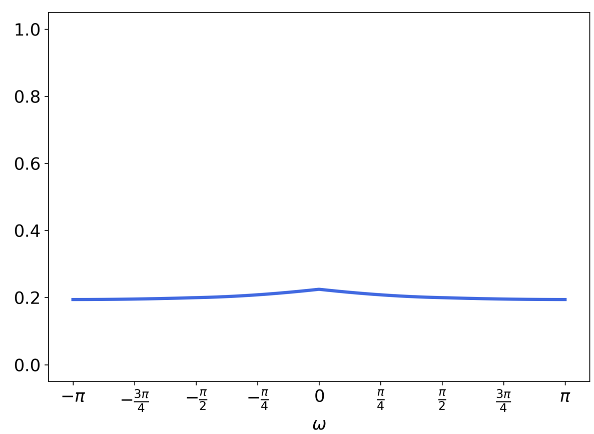
D=5.



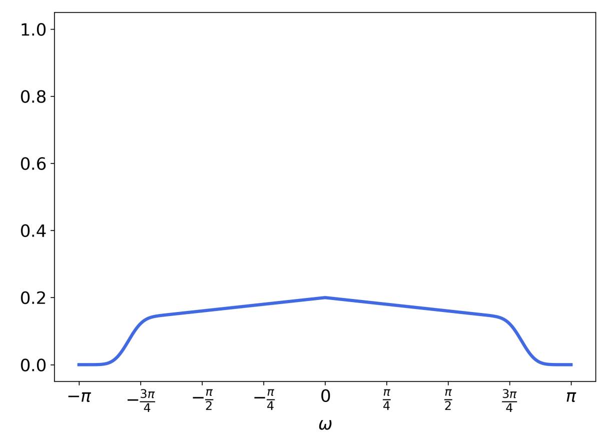
(a) Original $X(\omega)$



(b) DTFT after anti-aliasing filter.



(c) DTFT after downsampling with no anti-aliasing filter.



(d) Decimation result $Y(\omega)$

Figure 3: Example of full decimation system with and without anti-aliasing filter for D=5. Plot (a) shows the original DTFT $X(\omega)$ with maximum frequency $\frac{\pi}{2}$. Plot (b) shows the result after applying the necessary anti-aliasing filter with cutoff frequency $\frac{\pi}{5}$. Plots (c) and (d) show the final results after downsampling without and with the anti-aliasing filter, respectively. Note how (c) flattens out due to aliasing past $|\omega| > \frac{\pi}{2}$. Conversely, (d) retains the same shape as the result from (b) and prevents aliasing.