ECE 4624: Meeting 4
Sampling and Reconstruction

Chris Wyatt

2025-09-08

Today we review how to convert CT signals to DT Signals and vice-versa. This is an important step in DSP as in many applications the system is in practice acting on a CT signal. Note again, today's lecture is (mostly) a review of prerequisite material.

READING:

- ▶ PM 4.1.4 and Chapter 6
- ▶ 2714 Supplementary Notes 23 and 24

Sampling converts from a CT signal $x(t)$ to a digital (quantized DT) signal $x[n]$ using a sample time T.

Reconstruction converts from a DT signal $x[n]$ to a CT signal $x(t)$, interpolating between samples.

Ideally a conversion from $x(t)$ to $x[n]$ and back again would result in identical signal (an all-pass system).

Sampling Theory

The process of sampling is to produce a DT signal $x[n]$ from a CT signal $x(t)$ by sampling time at regular intervals $T \in \mathbb{R}^+$ called the *sample-time*, or equivalently sampling at a frequency of $\frac{1}{T}$ Hz or $\frac{2\pi}{T}$ rad/s.

Mathematically this is simple to express in the time domain as $x[n] = x(nT)$, however we seek a system that can perform this task.

Recall the impulse train is the periodic signal

$$x_1(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_0)$$

with period T_0 and frequency $\omega_0 = \frac{2\pi}{T_0}$.

The exponential CT Fourier series of the impulse train is given by

$$x_1(t) = \sum_{n=-\infty}^{\infty} a_n e^{j\frac{2\pi}{T_0}nt}$$

where the Fourier series coefficients are

$$a_n = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} \delta(t)e^{-jn\omega_0 t} \, dt = \frac{1}{T_0}$$

Now, lets take the Fourier Transform of the Fourier series representation

$$X_1(j\omega) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \frac{1}{T_0} e^{j\frac{2\pi}{T_0}nt} e^{j\omega t} \, dt$$

$$= \frac{2\pi}{T_0} \sum_{n=-\infty}^{\infty} \delta(\omega - \omega_0 n)$$

also an impulse train in the frequency domain.

Now suppose we have another signal $x_2(t)$ and we multiply $x_1(t)$ and $x_2(t)$ to get a signal $y(t)$.

$$y(t) = x_1(t) \cdot x_2(t) = \sum_{n=-\infty}^{\infty} x_2(t)\delta(t - nT_0) = \sum_{n=-\infty}^{\infty} x_2(nT_0)\delta(t - nT_0)$$

Since $y(t)$ is non-zero only at the locations of the delta functions, we can treat $y(nT_0) = x_2(nT_0)$ as the DT signal $x_2[n]$.

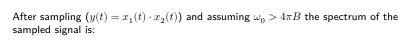Equivalently in the frequency domain the modulation theorem gives

$$y(t) = x_1(t) \cdot x_2(t) \overset{\mathcal{F}}{\longleftrightarrow} \frac{1}{2\pi} X_1(j\omega) * X_2(j\omega) = Y(j\omega)$$
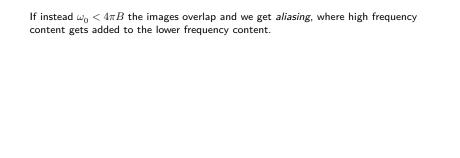
The convolution gives

$$
\begin{aligned}
Y(j\omega) &= \frac{1}{2\pi} X_1(j\omega) * X_2(j\omega) \\
&= \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} \frac{2\pi}{T_0} \sum_{n=-\infty}^{\infty} \delta(\omega - \omega' - \omega_0 n) X_2(j\omega') \, d\omega' \\
&= \frac{1}{T_0} \sum_{n=-\infty}^{\infty} X_2(j(\omega - n\omega_0))
\end{aligned}
$$

Thus the sampling process in the frequency domain causes periodic replication of the Fourier transform of the signal being sampled, $x_2(t)$, which are sometimes called *images*. This signal $Y(j\omega)$ is periodic in $\omega_0 = \frac{2\pi}{T_0}$ *radians per second* and corresponds to the DT Fourier Transform of $x_2[n] \overset{\mathcal{F}}{\longleftrightarrow} X_2\left(e^{j\omega}\right)$, which is periodic in 2π *radians per sample time*.

To help us visualize this, suppose that the signal $x_2(t) \overset{\mathcal{F}}{\longleftrightarrow} X_2(j\omega)$ is *band-limited* to B Hz, that is $X_2(j\omega) = 0$ for all frequencies outside the band $-2\pi B < \omega < 2\pi B$.

After sampling $(y(t) = x_1(t) \cdot x_2(t))$ and assuming $\omega_0 > 4\pi B$ the spectrum of the sampled signal is:

If instead $\omega_0 < 4\pi B$ the images overlap and we get *aliasing*, where high frequency content gets added to the lower frequency content.

Nyquist Sampling

To reconstruct the signal $x_2[n]$ back to $x_2(t)$ we need to ensure that $\omega_0 > 4\pi B$ rad/s or equivalently $f_0 > 2B$ Hz, which requires the sample time $T_0 < \frac{1}{2B}$ seconds.

This is called the *Nyquist* sample rate/frequency.

Practical Sampling

Sampling in practice requires addressing three issues.

1. We cannot generate the impulse train, but can only approximate it.
2. Digital signals must have a fixed bit width so we have to convert the real signal value to a *quantized* one.
3. The quantized value then needs to be coded into a bit representation (e.g. signed twos-complement)
4. Since in general we have no control over the input signal, we need to ensure the signal is approximately band-limited before sampling.

Sample and Hold

Sampling is typically accomplished using a circuit called a *sample-and-hold* or *track-and-hold*

Quantization

During the hold phase we need to *quantize* the signal into N bits.

▶ The quantizer is a nonlinear system

$$\hat{x}[n] = Q\{x[n]\}$$

▶ The input range (e.g. from 0 to V_{ref}) is divided into 2^N discrete levels.

▶ The LSB *weight* q is the smallest possible change in analog input that results in a change in the digital output – i.e., the resolution of the ADC.

$$q = \frac{V_{\text{ref}}}{2^N}$$

▶ For example, if the ADC has a reference voltage of 1 V and 12 bits of resolution, then:

$$q = \frac{1}{2^{12}} = \frac{1}{4096} \approx 244\,\mu\text{V}$$

Quantization Noise

In the simplest model, quantization adds *noise* uniformly distributed over the passband

▶ define the error as

$$e[n] = \hat{x}[n] - x[n]$$

▶ then

$$\hat{x}[n] = x[n] + e[n]$$

▶ e is assumed to be sampled from a stationary random process
▶ e is assumed to be uncorrelated with $x[n]$
▶ $e[n]$ is assumed to be uncorrelated to $e[n + m]$ for any m, thus a white noise process
▶ the probability distribution of e is assumed to be uniform over the sampling range

There are cases where this model breaks down but it often good enough. In this model the RMS value of the noise is $q/\sqrt{12}$, where q is weight of the LSB.

Quantization using successive approximation

Quantization can be thought of as a binary search algorithm:

Initialize all bits of $\hat{x}$ to zero.

Let bit b_i of $\hat{x}$ be the MSB. Given a real value x.

1. Set b_i to one
2. If $x < \hat{x}$ toggle the bit to zero
3. If no more bits, stop, else let $i \to i + 1$

This is implemented using an FSM.

The speed of conversion is limited by

▶ the settling time of the DAC
▶ the settling time of the Comparator
▶ the logic overhead, $O(N)$ state transitions

Anti-aliasing

Before the sample and hold we need to include a filter to limit the bandwidth. This can be accomplished by a CT low-pass filter called an *anti-aliasing* filter whose cutoff frequency in the ideal case is $\omega_c = 2\pi B$.

As we discussed last time ideal filters cannot be implemented, thus we specify the anti-aliasing filter as a pass-band gain/frequency and a stop-band gain/frequency.

Since the transition band is non-zero for a practical filter, this means we have to either lower the pass-band relative to the ideal or increase the sample rate.

In the best case, the filter should have a stop-band frequency at half the sampling frequency with the order of the filter and pass-band frequency adjusted as needed.

Alternatively the gain that defines the stop-band can be relaxed.

Anti-aliasing Design Example

Sigma-Delta ADC

A different approach to ADC that is often used in high-precision integrated VLSI devices is to:

1. oversample the input signal by a factor K
2. shape the quantization noise using a sigma-delta modulator
3. digital lowpass filter the result
4. decimate the digital signal by K

Oversampling with analog and digital filtering

Recall the quantization noise is added to the passband uniformly. If we increase the sampling rate by a factor K (oversample) then this noise is spread out over a larger bandwidth.

We can then digitally low-pass filter to remove some of this noise. This improves the signal-to-noise ratio by $10 \log_{10}(K)$ dB.

For example if K=4, then we get a modest 6 dB improvement.

Sigma-Delta Modulation

We can "shape" the noise power spectral density using a modulator to push more of it into the digital filter stop band.

Reconstruction Theory

Given a DT signal $x[n]$ and a sample spacing T, we can define a corresponding CT signal as

$$x_p(t) = \sum_{n=-\infty}^{\infty} x[n]\, \delta(t - nT)$$

the impulse train with each impulse weighted by the DT signal.

CT signal reconstruction can be viewed from two perspectives.

In the time domain perspective, the CT signal $x(t)$ corresponding to a DT signal $x[n]$ can be viewed as *interpolation*, where the values of the CT signal are equal to the DT signal at intervals of the sample time, i.e. $x(nT) = x[n]$, and in between the value of $x(t)$ is interpolated.

If the interpolation is of zero-order, the value at $x(nT)$ is held constant until $x(nT + T)$.

This is called a *zero-order hold*, and can mathematically modeled as convolution of the weighted impulse train with a pulse $p(t) = u(t) - u(t - T)$ whose width is the sample time, called the interpolation function.

$$y(t) = p(t) * x_p(t)$$

The zero-order hold is not a very accurate representation of a band-limited signal. So, what interpolation function is optimal?

To answer this question we can turn to the alternative perspective on reconstruction, that of the frequency domain.

Recall the sampled signal $x(nT)$ in the frequency domain can be viewed as the summation of the Fourier transform of $x(t)$, $X(j\omega)$, and periodic replicas or images centered at multiples of the sampling frequency.

If we assume the original signal was band-limited and sampled appropriately (using the Nyquist criteria), then if we ideal low-pass filter the sampled signal we will preserve the central portion of the Fourier spectrum that corresponds to the original signal, and chop off the images.

For this reason the reconstruction filter is also called an anti-imaging filter.

Recall filtering is multiplication the frequency domain and convolution in the time domain, so the optimal interpolation function corresponds to the impulse response of the ideal low-pass filter with cutoff frequency $\omega_c = 2\pi B$, a sinc function.

$$h(t) = \mathcal{F}^{-1}\left\{H(j\omega)\right\} = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega t}\, d\omega = \frac{1}{\pi t}\sin(\omega_c t)$$

Thus the ideal ideal interpolation function is the sinc function, and reconstruction is low-pass filtering of the weighted impulse train $x_p(t)$.

Practical Reconstruction

As we have seen before we cannot physically represent the impulse train nor the ideal low-pass filter.

Thus practical reconstruction uses an approximation of the ideal reconstruction filter by a digital-to-analog converter (DAC), followed by a causal (and thus physically possible) low-pass filter.

Zero-order hold using an R-2R ladder

A zero-order hold DAC can be implemented by a circuit called a resistor ladder. Consider a digital output with N bits and a reference voltage V_{ref} (for example an 8-bit output port on a micro-controller using CMOS 3.3v logic).

If this port is connected to a resistor network consisting of resistor values R and $2R$ then depending on the bit pattern at the output port V, the output of the buffer op-amp will be

$$V_o = V_{ref} \frac{V}{2^N}$$

If the port value is changed every sample time T, then the resister ladder and buffer op-amp combine to implement a zero-order hold circuit.

Capacitive DACs

Resister ladders have a number of drawbacks that limit them to around 8-bit resolution

▶ requires very accurate and closely matched resistor values that are hard to manufacture

▶ a current flows through the resistor network, regardless of the output value = static power dissipation that increases with resolution

▶ integrated (on-chip) matching and trimming of resistors is a complex and expensive process

Switched-capacitor DACs can use arrays of identical, unit-sized capacitors.

▶ all capacitors are identical so their relative matching is more precise and has less variation.

▶ current flows only when charging/discharging = less power at idle

▶ precise capacitors are easier and more cost-effective to manufacture in integrated devices than resistors

Most SAR ADC devices use a capactive DAC internally to reach 12-14 bits of precision.

Glitches

- The charging and discharging in capacitive DACs can lead to "glitches"
- These can be pronounced when there are large changes (high order bits)

Segmented DACS

One approach to reducing glitches is to use a segmented DAC

- ▶ Higher order bits use a themometer coded DAC
- ▶ lower order bits use a binary weighted capacitive DAC
- ▶ the two outputs are summed

Anti-imaging (reconstruction) filter

The zero-order hold is followed by the reconstruction (anti-imaging) filter which low-pass filters the output and smooths-out the jumps from value to value and glitches during transitions.

In general the reconstruction filter is of a similar, or identical form to an anti-aliasing filter.

Sigma-Delta DAC

The Sigma-Delta ADC can effectively be reversed to get a Sigma-Delta DAC for more than 12 bits of accuracy.

Additional References

- Understanding SAR ADCs: Their Architecture and Comparison with Other ADCs Analog Devices Technical Article
- Sigma-Delta ADCs and DACs, Analog Devices Application Note AN283