Bandlimited Signals and Sampling

Objectives:

- Find the bandlimits of bandlimited signals
- Use low/high/bandpass filters to make signals bandlimited.
- Find the CTFT of an impulse-train sampled signal
- Find the DTFT of a sampled CT signal
- Compute the Nyquist rate for bandlimited signals
- Explain the meaning and implications of the Sampling Theorem
- Explain how aliasing arises from undersampled signals

Main

We call signals (CT or DT) <u>bandlimited</u> if their spectrum is only nonzero for $|\omega| < B$ for some B:

$$x(t) = e^{-at} \xleftrightarrow{\mathcal{F}} X(j\omega) = \frac{1}{a + j\omega} \quad \text{not bandlimited}$$

$$x(t) = \frac{\sin(Bt)}{\pi t} \xleftrightarrow{\mathcal{F}} X(j\omega) = \begin{cases} 1 & |\omega| < B \\ 0 & \text{otherwise} \end{cases}$$

bandlimited!

$$x[n] = a^n u[n] \xleftrightarrow{\mathcal{F}} \frac{1}{1 - e^{-j\omega}} \quad \text{not bandlimited}$$

$$x[n] = \frac{\sin(Bn)}{\pi n} \xleftrightarrow{\mathcal{F}} X(e^{j\omega}) = \begin{cases} 1 & |\omega| < B \\ 0 & \text{otherwise} \end{cases}$$

bandlimited!

We can make a signal bandlimited by using an ideal LPF to cut off the spectrum.

<u>Fact</u>: If $x(t)$ (or $x[n]$) is finite in time (that is, there is a time T (or N) such that $x(t) = 0$ for $|T| > 0$ (or $x[n] = 0$ for $|N| > 0$) then $x(t)$ (or $x[n]$) is not bandlimited — that is, $X(j\omega)$ (or $X(e^{j\omega})$) is infinite in time.

Phil

This fact follows from a more general uncertainty principle (related to Heisenberg's famous uncertainty principle from physics):

a signal cannot be well-localized in both time and frequency.

Here, "localized" means concentrated in a small area. So if all of the signal is in a finite time interval, it has to have infinite frequency content. The reverse is true too: bandlimited signals are infinite in time.

This is also related to why we cannot really build ideal LPFs: they are not causal and infinite in time.

Main

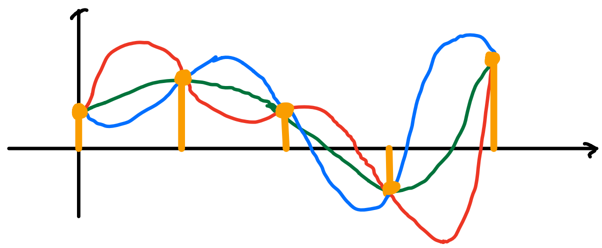
It turns out that bandlimited CT signals are very special — we can sample them to get a DT signal and then interpolate to get back the CT signal. That is, sampling doesn't lose any information about the signal.

Suppose $x(t)$ is bandlimited to $[-\omega_0, \omega_0]$ What happens when we sample it?

$$x_T[n] = x(nT)$$

here T is the sampling interval — smaller T means we are sampling $x(t)$ more finely.

Doesn't seem reasonable to be able to reconstruct an arbitrary signal:



All three signals have the same samples

There are several ways to create a sampled signal:

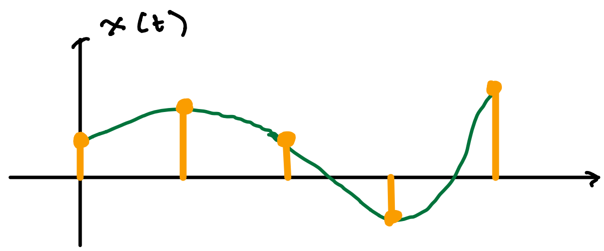
Impulse-train sampling:

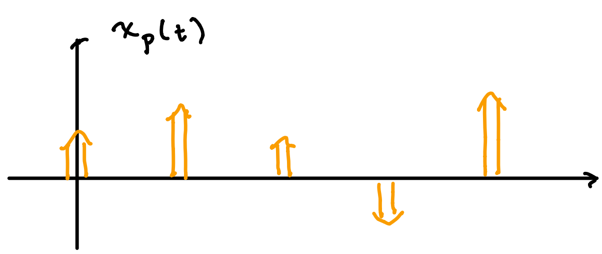
$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

train of impulses at multiples of t

Then

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



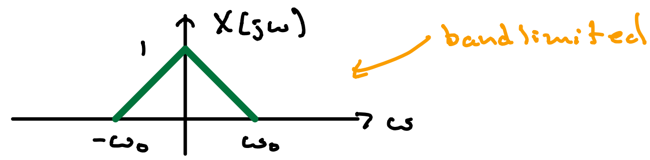


If we set $\omega_s = \frac{2\pi}{T}$ we can use the multiplication property of the CTFT to write the CTFT of $x_p(t)$:

$$X_p(j\omega) = \frac{1}{2\pi}(X * P)(j\omega)$$

We need the CTFTs $X(j\omega)$ and $P(j\omega)$:

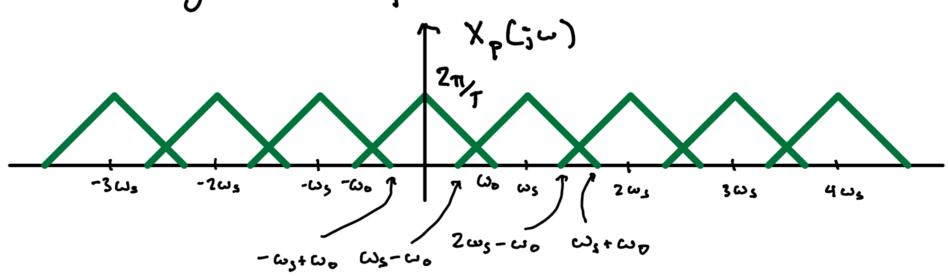
$X(j\omega) =$



impulse train

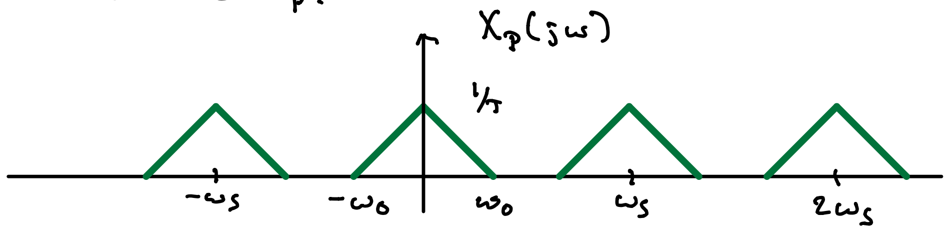
$$P(j\omega) = \frac{2\pi}{T}\sum_{l=-\infty}^{\infty} \delta(\omega - l\,\omega_s)$$

Convolving with an impulse just makes a copy of $X(j\omega)$ so $\frac{1}{2\pi}(X * P)(j\omega)$ is a train of copies of $X(j\omega)$ centered at integer multiples of ω_s:

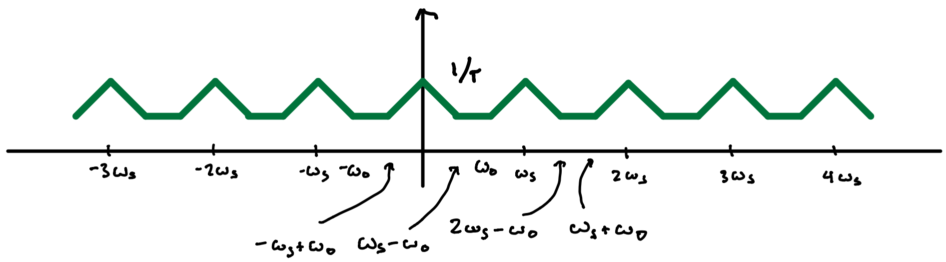


So we have two cases:

If $\omega_s > 2\omega_0$ then the copies do not overlap:

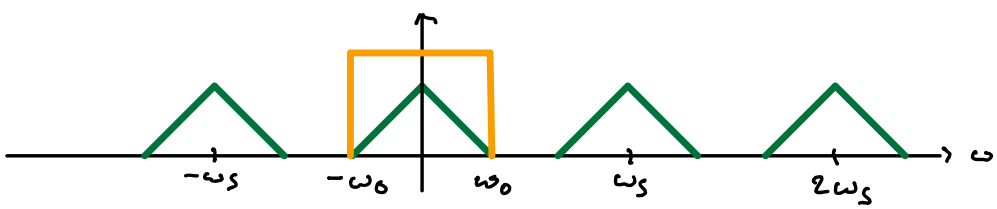


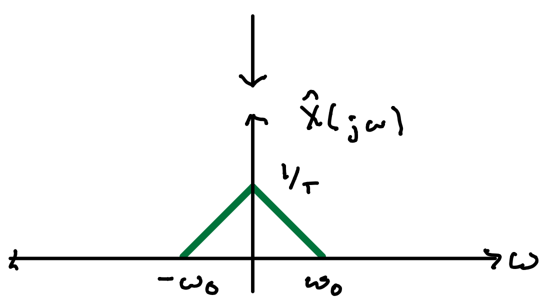
If $\omega_s < 2\omega_0$ the copies will overlap, causing aliasing:



Note that in the first case, we could get back $X(j\omega)$ (and $x(t)$) by applying a LPF with cutoff ω_0:

$$x_p(t) \longrightarrow \boxed{\text{LPF }(\omega_0)} \longrightarrow \hat{x}(t)$$





This shows that if $\omega_s > 2\omega_0$ we can recover the original $x(t)$!

What is happening in the time domain? The LPF has impulse response

$$h(t) = \frac{\sin(\omega_o t)}{\pi t}.$$

Convolving an impulse train with this sinc function is called <u>sinc interpolation</u>.

Ex

Suppose you have a DT signal $x[n]$ that is formed by sampling a bandlimited CT signal $x(t)$ with bandlimit 20,000 Hz (the limits of human hearing) at a sampling frequency $\omega_s =$
For example, this is how digital audio (like an MP3) encodes an acoustic signal (your favorite song).

Theoretically, you could recover $x(t)$ by converting $x[n]$ into a CT signal of δ-functions multiplied by $x[n]$ at $t = nT$:

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

and then passing $\tilde{x}(t)$ through an ideal LPF.

Main

To summarize, if you have a signal $x(t)$ which is bandlimited to $[-\omega_o, \omega_o]$ then you can recover $x(t)$ from a sampled version

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

as long as $\omega_s = \frac{2\pi}{T} > 2\omega_o$.

We can think of $2\omega_0$ as the total bandwidth of the signal.

How should we understand $x_p(t)$? It is equivalent to the DT signal

$$x[n] = x(nT)$$

so how does the DTFT of $x[n]$ relate to the CTFT of $x_p(t)$?

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \qquad X_p(j\varphi) = \int_{-\infty}^{\infty} x_p(t) e^{-j\varphi t}\, dt$$

$$= \sum_{n=-\infty}^{\infty} x(nT)\, e^{-j\omega n} \qquad = \sum_{n=-\infty}^{\infty} x(nT)\, e^{-j\varphi n T}$$

So this means $\varphi = \omega / T$

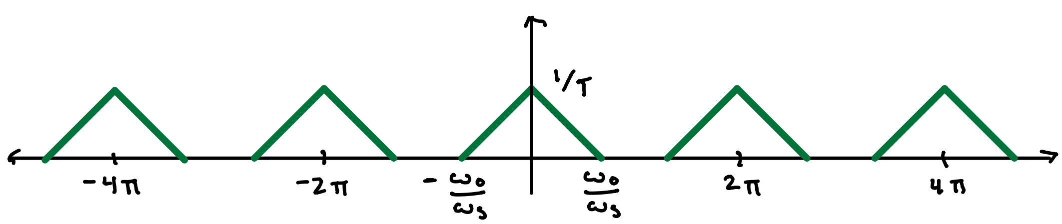
$$\text{DTFT} \quad X(e^{j\omega}) = X_p(j\, \omega/T) \quad \text{CTFT}$$

Since $T = \frac{2\pi}{\omega_s}$, $\quad X(e^{j\omega}) = X_p\left(j \frac{\omega\, \omega_s}{2\pi}\right)$

So in the DTFT, when $\omega = 2\pi$ we get

$$X(e^{j0}) = X(e^{j2\pi}) = X_p(j\omega_s) = X_p(0)$$

This makes sense: $X(e^{j\omega})$ is periodic with period 2π and $X_p(j\varphi)$ is periodic with period ω_s — the DTFT $X(e^{j\omega})$ is just the squished or stretched version of the CTFT $X(j\varphi)$:



This sampling/reconstruction result is known as the <u>Sampling Theorem</u>.

<u>Theorem</u>: Let $x(t)$ be a signal whose CTFT $X(j\omega) = 0$ for $|\omega| > \omega_0$. Then $x(t)$ is uniquely determined by its samples

$$\{x(nT)\} \qquad n = 0, \pm 1, \pm 2, \ldots$$

as long as

$$\omega_s = \frac{2\pi}{T} > 2\omega_0 .$$

The signal $x(t)$ can be recovered from $\{x(nT)\}$ by using sinc interpolation.

!!! This is probably the second-most important fact from this class.

The $>$ is necessary: choosing $\omega_s = 2\omega_0$ is not quite good enough because we really need a little separation to prevent aliasing. See the book, Example 7.1, for a reason.

Def

<u>Def</u>. If $x(t)$ is bandlimited to $[-\omega_0, \omega_0]$ we call the threshold $2\omega_0$ the Nyquist rate of the signal

Phil

Harry Nyquist (1889–1976) was born in Sweden and moved to the US in 1907. He spent his entire career at AT&T / Bell Labs and did a ton of foundational work on signals, systems/control, and communications.

The sampling theorem is sometimes called the Shannon-Nyquist theorem because Shannon wrote a paper in 1949 about it. It was also proved independently by Whittaker, Gabor, Kotelnikov, and others. The Wikipedia page has a nice rundown. Nyquist showed that $\omega_s > 2\omega_0$ was enough to represent $x(t)$ — the other part is showing that sinc interpolation recovers $x(t)$.

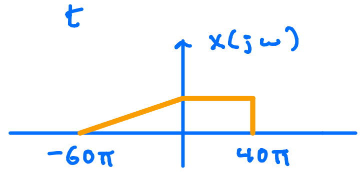
Ex

Example: Find the Nyquist rate for the following CT signals:

a) $\cos(200\pi t) + 2\sin(40\pi t)$

b) $\dfrac{\sin(30t)}{t}$

c)



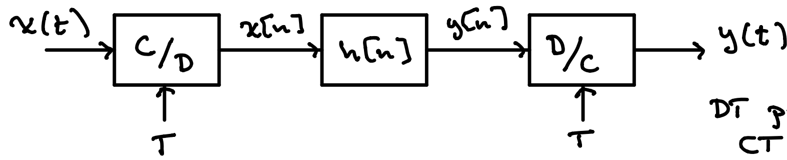
a) The CTFT has δ-functions at $\pm 40\pi$ and $\pm 200\pi$ so $\omega_s > 400\pi$

b) This is $\pi \cdot \mathrm{LPF}(30)$ so $\omega_s > 60$

c) Here $\omega_0 = 60\pi$ (to get the whole signal): $\omega_s > 120\pi$.

Main

Why sample signals? Doing computation on analog (CT) signals can be challenging: building good analog ICs is hard. If we can sample to DT we can apply more general-purpose computing to do our signal processing and then convert the output back to CT:



DT processing of CT signals

A couple of things to note:

- putting in the C/D and D/C blocks means there is some delay in processing — doing stuff all in analog is <u>faster</u>

- To get back to CT we need to keep track of T. This is why you need $f_s = \frac{\omega_s}{2\pi}$ when playing sound in MATLAB

- Since ideal LPFs are impossible to build, a "real" D/C block will introduce some additional distortion.