Lec 9 F01 Mon 9/10/01
(used Lec 9 from F00)

We started by considering some examples of signal processing algorithms in action, and saw that all those examples neatly fit into one basic picture:

$$x \longrightarrow \boxed{S} \longrightarrow y$$

We then talked about a few very basic notions, which the whole field of signal processing is based on. We've answered questions like: what's a signal? what's a system? We've concentrated on linear time-invariant systems, and saw that, in order to analyze such systems, it was important to study representations of signals in orthogonal bases.

$$s(n) = \sum_k a_k g_k(n)$$

For example,

* when we represented our input signal as the sum of shifted and scaled impulses, we obtained the interpretation of an LTI system as the convolution of its impulse response with the input.
* When we represented our input signal as the sum of complex exponentials, we got an equally important interpretation of an LTI system--namely, that it modifies different frequency components independently of each other, by just multiplying them by complex numbers. We called these frequency-dependent numbers the frequency response, and saw that it was the discrete-time Fourier transform of the impulse response.

Because of this property of LTI systems, it is really important to be able to think of signals both in time domain and in frequency domain, which is why Lab 3 and Homeworks 3 and 4 emphasize spectral representations of signals and playing with frequency responses of various systems.

This background material that we've been covering for three weeks now allows us to start considering several important practical matters. One example is filter design, when you want to attenuate certain frequencies in a signal and enhance others. If you have audio equipment, you are modifying a digital filter every time you adjust bass or treble. If you are playing with Adobe Photoshop on your computer, and click on "enhance", you are applying a digital filter to your image.

Now, if you want to implement a digital filter as a convolution, how do you design a fast algorithm to do that? How can you quickly compute spectral representations of signals? Well, this is where the FFT algorithm comes in. You will see it in a few weeks in lab 6.

Another practical issue that we will start looking at very shortly is sampling. You may have noticed that most real-world signals are continuous-time (or continuous-space). When you go to a concert, the music you hear is a continuous-time signal. How can you reliably store it as discrete samples on a compact disc? Similarly, the world you see around you is continuous. Then how can we store digital images on a computer, and make them look realistic and distortion-free? Sampling theory deals with these issues.

RHS Blackboard

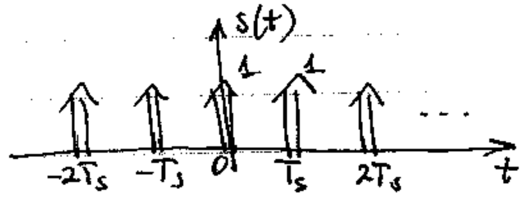
1.4 Sampling.

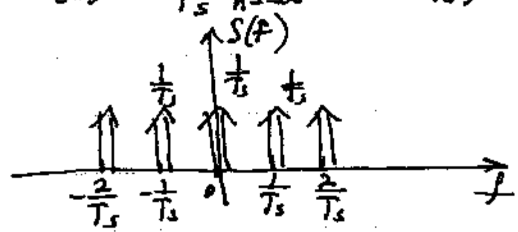
Recall some facts from CT:

1) CTFT: $X(f)=\int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$

$$x(t)=\int_{-\infty}^{\infty} X(f)\,e^{j2\pi ft}df$$

$$s(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT_s) \longrightarrow S(f)=\frac{1}{T_s}\sum_{n=-\infty}^{\infty}\delta\left(f-\frac{n}{T_s}\right)$$

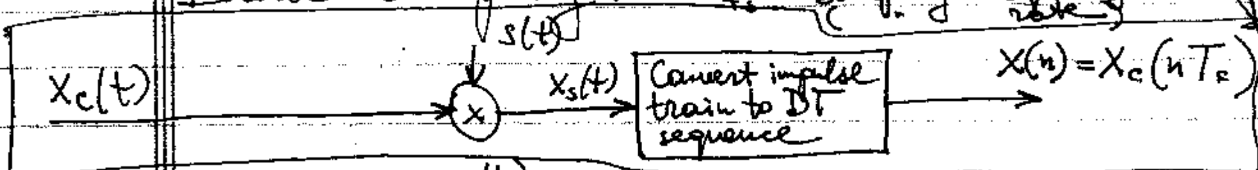


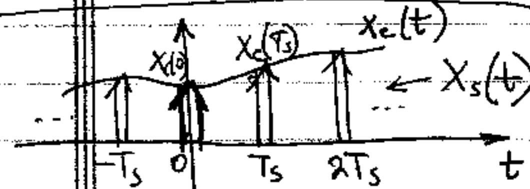


2) Convolution with CT δ:

$$x(t)*\delta(t-t_0)=\int_{-\infty}^{\infty}\delta(\tau-t_0)x(t-\tau)d\tau = x(t-t_0).$$

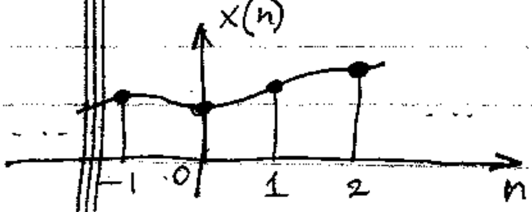
Ideal sampling: T_s = sampling period, f_s = sampling frequency (sampling rate)





$\longleftarrow x_s(t)=x_c(t)s(t) \longrightarrow$ $\boxed{X_s(f)=X_c(f)*S(f) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty}X_c\left(f-\frac{n}{T_s}\right)}$ DNE (*)

(for every impulse in $S(f)$, we get a copy of $X_c(f)$ at the location of that impulse)



How is DTFT of $x(n)$ related to CTFTs of $x_s(t)$ and $x_c(t)$?

Try a different method of calculating CTFT of $x_s(t)$:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x_c(nT_s)\,\delta(t-nT_s)$$

$$X_s(f) = \sum_{n=-\infty}^{\infty} x_c(nT_s)\,\mathrm{CTFT}\{\delta(t-nT_s)\}$$

$$= \sum_{n=-\infty}^{\infty} x_c(nT_s)\left\{\int_{-\infty}^{\infty} \delta(t-nT_s)\,e^{-j2\pi f t}\,dt\right\}$$

$$= \sum_{n=-\infty}^{\infty} x(n)\,e^{-j\overbrace{2\pi f T_s}^{=\omega} n} = X(e^{j\omega})\Big|_{\omega=2\pi f T_s}$$

$$\underbrace{X_s(f)}_{\text{CTFT of } x_s(t)} = \underbrace{X\left(e^{j2\pi f T_s}\right)}_{\text{DTFT of } x(n)} \quad ; \quad X(e^{j\omega}) = X_s\left(\frac{\omega}{2\pi T_s}\right)$$

I.e., to get $X(e^{j\omega})$ from $X_s(f)$, just re-scale the frequency axis by replacing $\boxed{f_s \to 2\pi;\ \frac{f_s}{2} \to \pi,}$ etc.

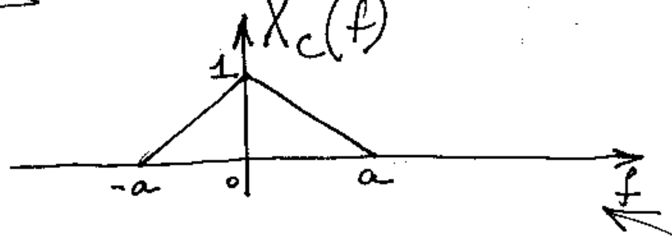
Finally, express the spectrum of the DT sampled signal in terms of the spectrum of the original CT signal $x_c(t)$:

$$X(e^{j\omega}) = X_s\left(\frac{\omega}{2\pi T_s}\right) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} X_c\left(\frac{\omega}{2\pi T_s} - \frac{n}{T_s}\right)$$

— Eq. (1) from Lab 4.

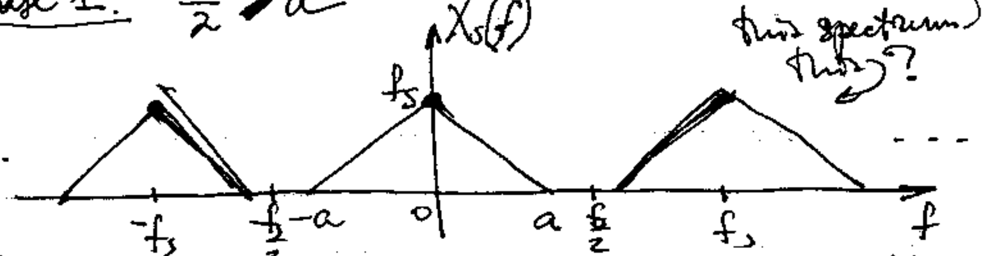
The major point of concern here is how accurately these discrete samples represent the original CT signal. We will try to answer this question by looking at the spectra, and determining whether the original spectrum $X_c(f)$ can be recovered by filtering this spectrum.

Example 1.

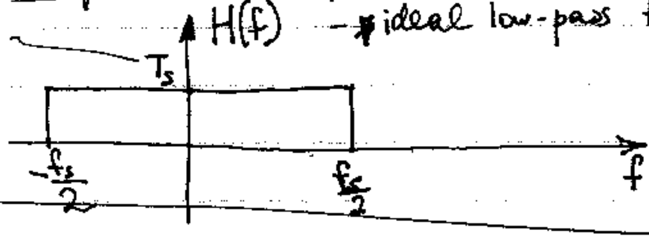


Case 1. $\frac{f_s}{2} > a$

Can we reconstruct this spectrum from this?

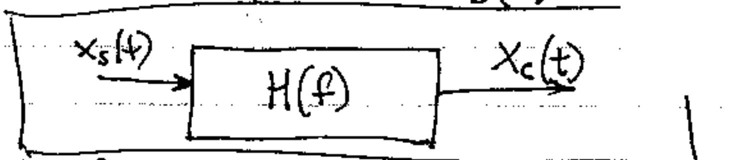


Yes: perfect reconstruction with a low-pass filter is possible if the sampling frequency is at least twice the highest frequency of the signal.

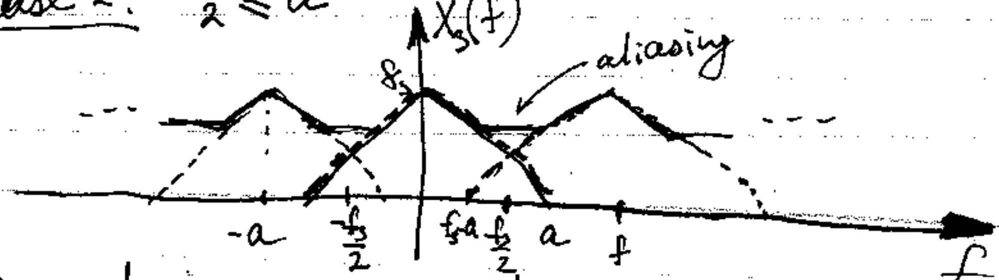


– ideal low-pass filter

Reconstruct $X_c(t)$ from $X_s(t)$:

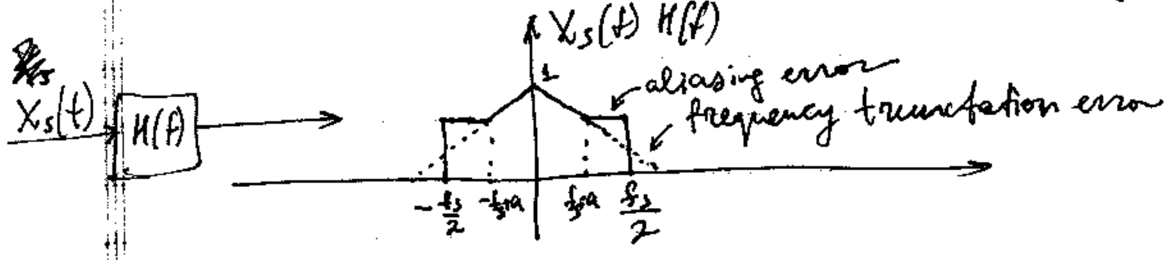


Case 2. $\frac{f_s}{2} \leqslant a$

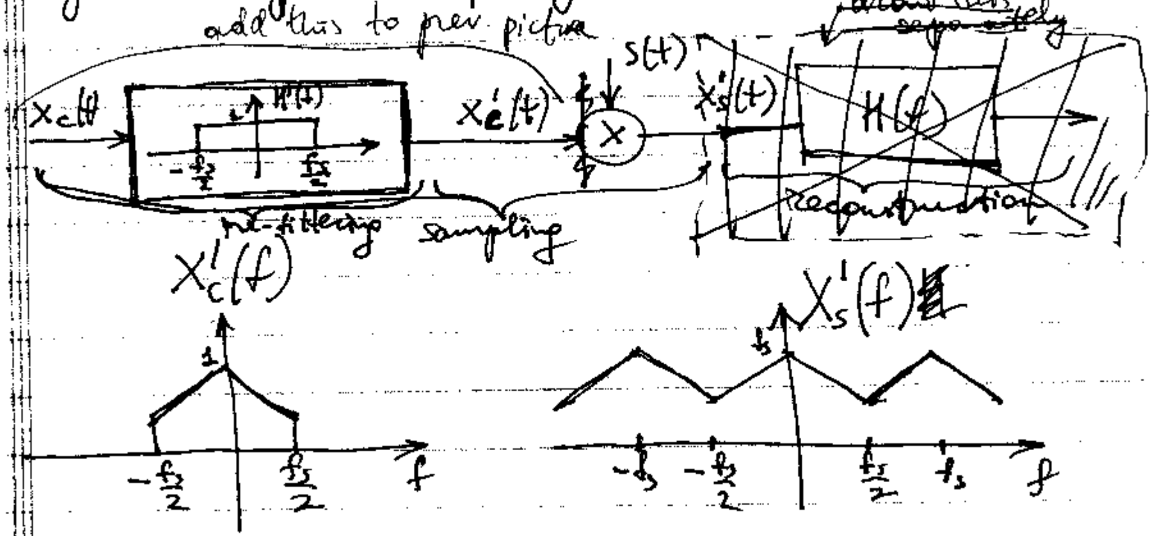


Cannot recover X_c with a low-pass filter.

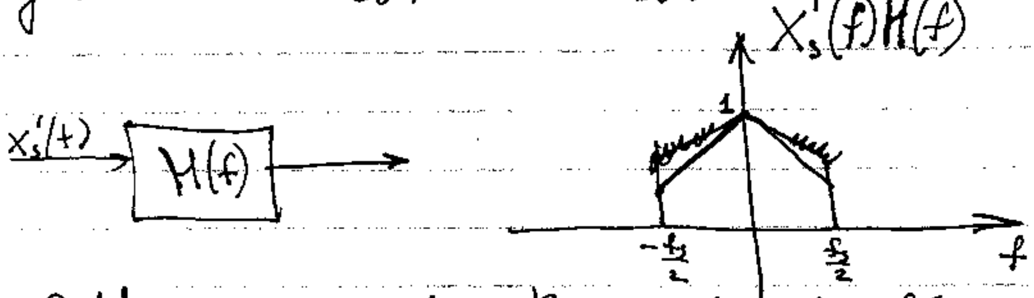
Filter $X_s(f)$ with $H(f)$ $\Rightarrow$ recover the following:



Better to pre-filter, to make ~~at~~ sure that the signal's highest frequency is no larger than $\frac{f_s}{2}$:



Try to recover $x_c(t)$ from $X_s'(t)$:



- better reconstruction than $X_s(f)H(f)$: recovered frequencies in the range $[f_s-a, \frac{f_s}{2}]$.
(Got rid of the aliasing error.)

<u>Nyquist Sampling Theorem.</u> Let $x_c(t)$ be a bandlimited signal with

$$X_c(f) = 0 \quad \text{for } |f| > a. \leftarrow \text{Nyquist frequency}$$

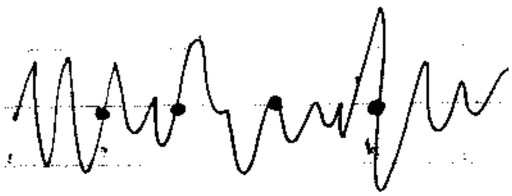
If $x_c(t)$ is sampled with frequency f_s samples/second, with

$$f_s > 2a, \leftarrow \text{Nyquist rate}$$

then $x_c(t)$ ~~can be recovered from~~ is uniquely determined by its samples

$$x(n) = x_c(nT), \quad n = 0, \pm 1, \pm 2, \ldots$$

The idea is very simple and intuitive, and we have seen it before:



if there is a lot of things happening in between samples, there is no way to recover ~~these~~ things from these samples. In order to accurately represent such a signal, we need to sample more densely.

~~Let us now consider the last stage of this diagram. How do we relate the spectrum of this discrete-time signal $x(n)$ to the spectrum of this continuous-time impulse train?~~