1

Reconstruction of sampled signals → Lecture 5

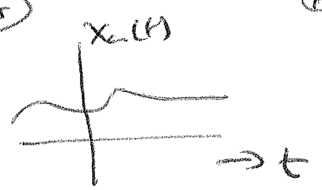
Outline

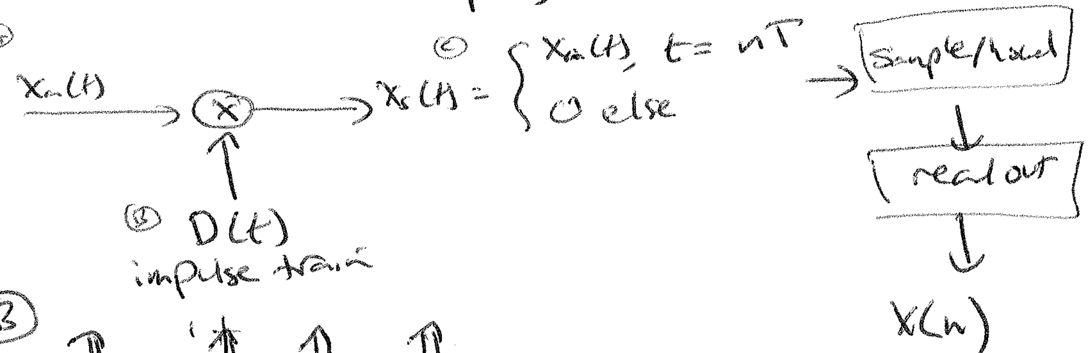
1) quick review of sampling ($x_a(t) \to x(n)$)
2) idealized reconstruction
3) practical reconstruction
4) Problems w/ aliasing (Matlab!)
   - sine waves
   - non-bandlimited samples ; white noise
   - need for prefiltering
5) bandlimited sampling (Matlab!)

---

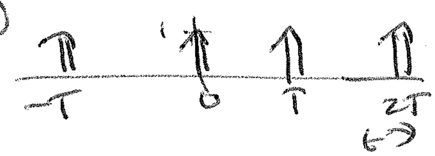
quick review : idealized sampling

(A)

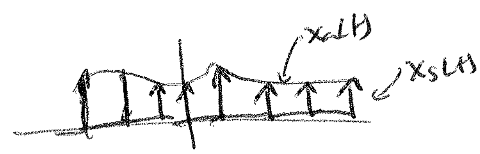




(B)



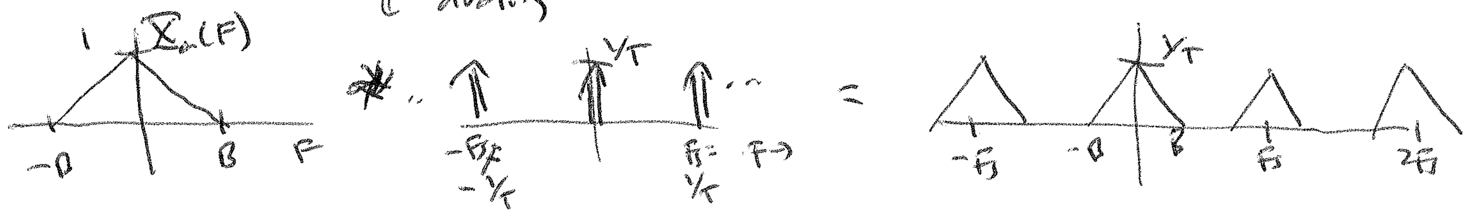
(C)



---

in frequency: (conv/mult duality

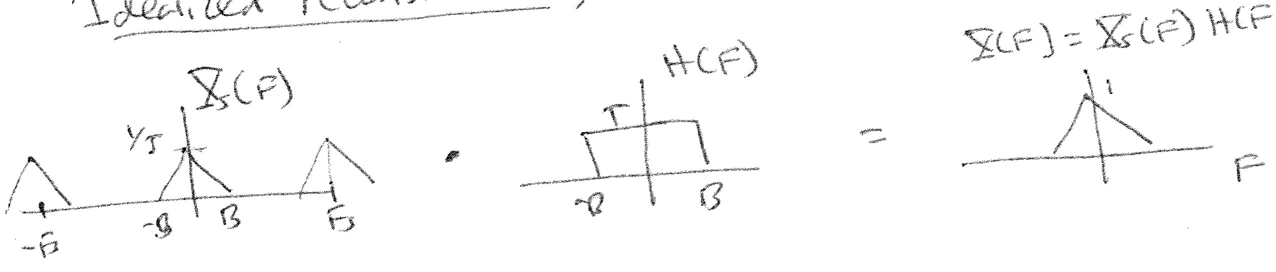
case $B < F_s/2$



→ overlap if $B > F_s/2$

Recovery is 2 steps:
1. apply filter + get $X(F)$ back
2. apply inverse FT

<u>Idealized reconstruction</u>, Graphically:



Mathematically:

Say $B = F_s/2$ (i.e., $H(F)$ will recover all unaliased F's)

$$x_a(t) = \int X_s(F)\,H(F)\,e^{+j2\pi F t}\,dF$$

$$= T\int_{-F_s/2}^{F_s/2} X_s(F)\,e^{+j2\pi F t}\,dF \quad \leftarrow \text{use fact that } H(F) = T \text{ in } [-F_s/2, F_s/2],\ = 0 \text{ else}$$

$$= T\int_{-F_s/2}^{F_s/2} \Big[\sum_{n=-\infty}^{\infty} x(n)\,e^{-j2\pi (F/F_s) n}\Big]\,e^{+j2\pi F t}\,dF$$

↑ from ☆ on p. 2

$$= \sum_{n=-\infty}^{\infty} x(n)\,T\int_{-F_s/2}^{F_s/2} e^{+j2\pi F\left(t-\frac{n}{F_s}\right)}\,dF$$

↑ note $t - \frac{n}{F_s} = t - nT$

after the usual algebra for getting a sinc,

$$x_a(t) = \sum_{n=-\infty}^{\infty} x(n)\,\frac{\sin\left(\frac{\pi}{T}(t-nT)\right)}{\frac{\pi}{T}(t-nT)}$$

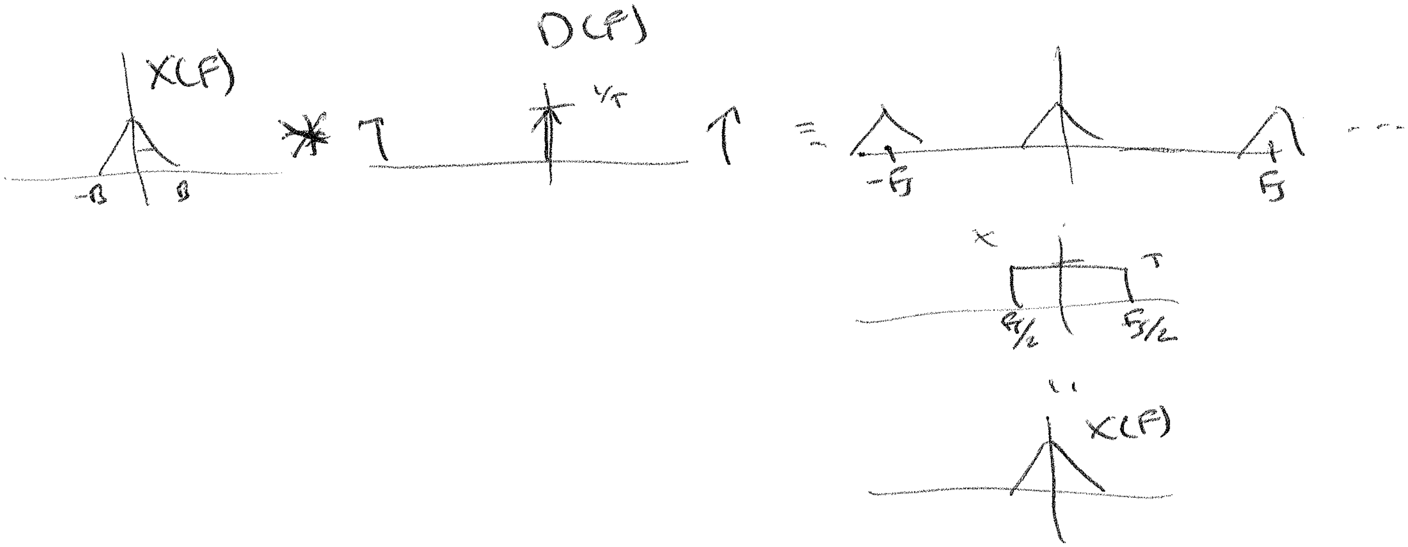
this gives $x_a(t) = x(n)$ at $t = nT$
and a smoothed (interpolated) version in between

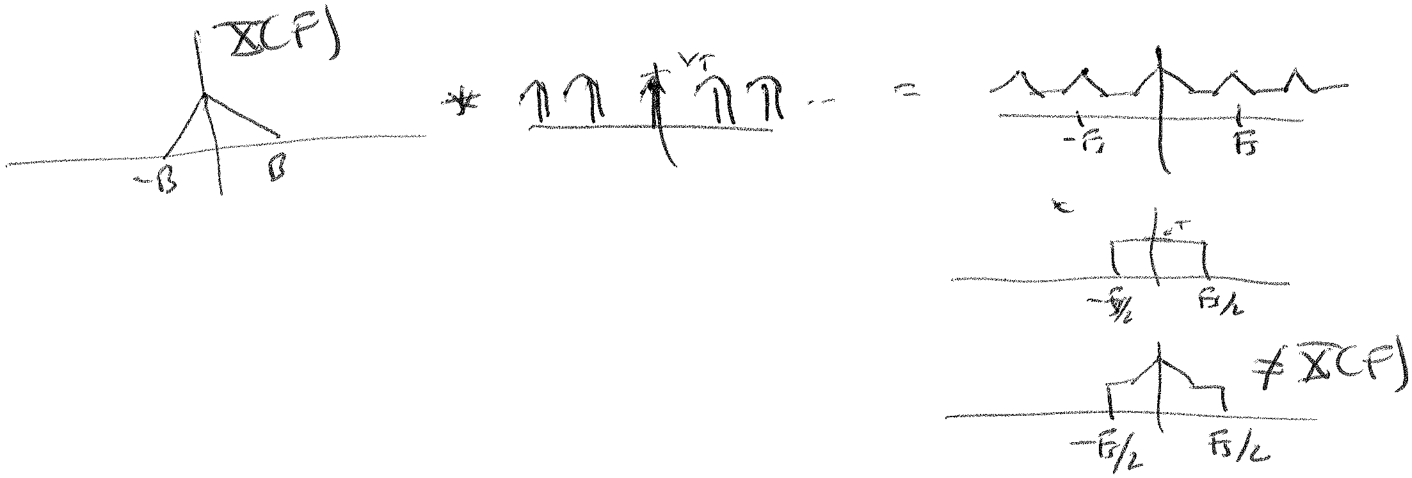
note we can write this as

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

2.5

Now we can see why the overlap in frequency when don't satisfy $B < F_s/2$ is bad; we lose info & cannot reconstruct





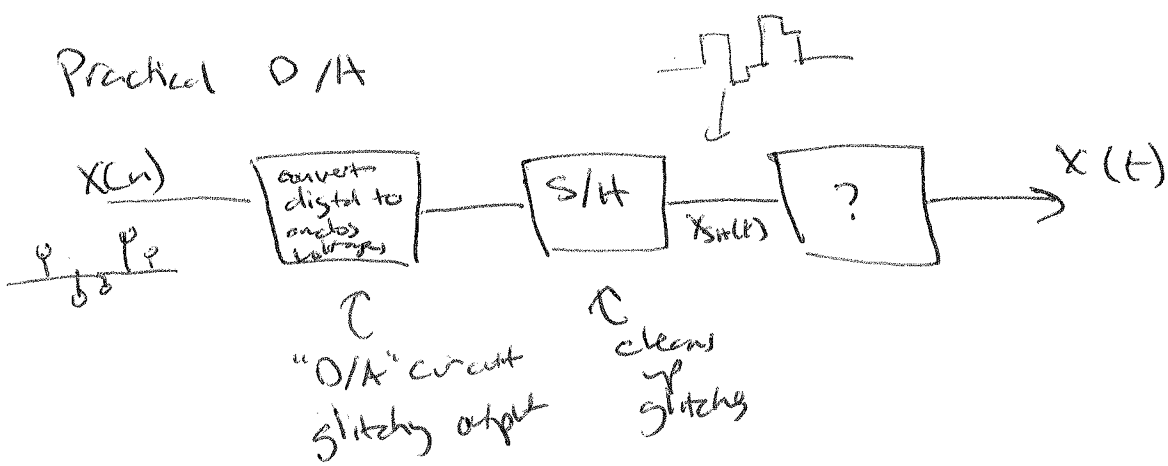
3

The reconstruction above is "idealized" — can't really be done in practice.

why not?

→ Brick-wall filter gives ∞-long impulse response, so theoretically we'd need to add contributions from points ∞'ly far away.

Practical D/A



see Fig 6.3.8 for S/H output

what to do next? → some kind of low-pass filter

→ linear interp is fine

analyze S/H output

$$x_{SH}(t) = \sum_n x(n)\, g_{SH}(t-nT)$$

S/H grabs & holds, so is a square pulse:

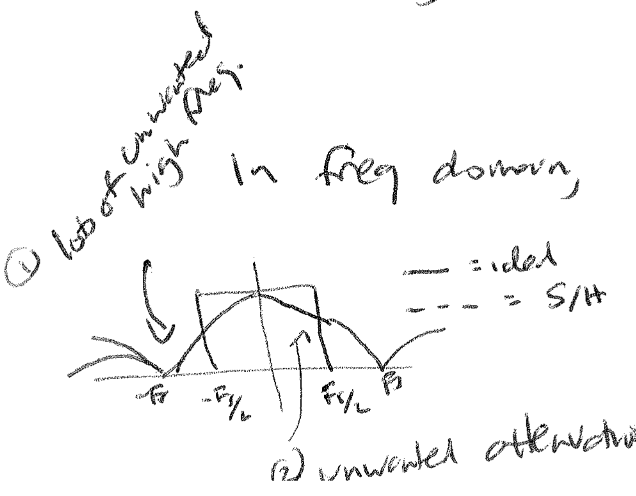
$$g_{SH} = \begin{cases} 1 & 0 \le t \le T \\ 0 & \text{else} \end{cases}$$

In freq domain,

$$G_{SH}(F) = \int_{-\infty}^{\infty} g_{SH}(t)\, e^{-j2\pi Ft}\, dt = \int_0^T e^{-j2\pi Ft}\, dt$$

$$= T\,\frac{\sin \pi FT}{\pi FT}\, e^{-2\pi F(T/2)}$$

↑ like HW problem



The high-frequency info in previous figure (Fig 6.3.9) comes from sharp edges in time.

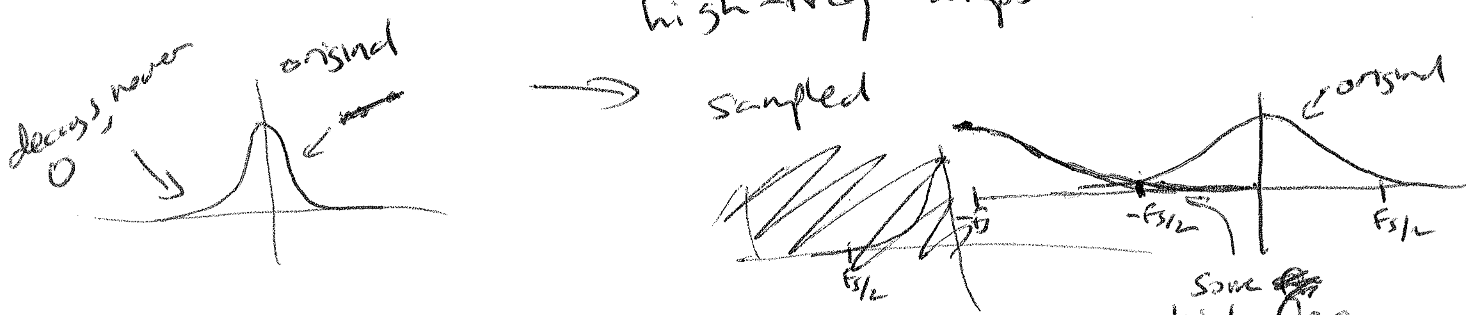
A low-pass filter <u>smoothes</u> the edges in time and therefore reduces the high-freq energy.

<u>Aliasing</u>

~~now we~~ Two cases to consider carefully

1. aliasing of sinusoids – ex. 6.1.1 in book
   MATLAB 1 explores this
2. Non-bandlimited signals – Ex 6.1.2
   see figures in PPT – repeat Figs. 6.1.7 & 6.1.8

If signal doesn't <u>exactly</u> → 0, there will always be some energy
high-freq wraps into low bands



Some high freq energy is mapped into our band of [-Fs/2, Fs/2]

(5)

<u>Therefore</u>, it's important to add a pre-filter, or AAF (antialias filter) before the A/D converter

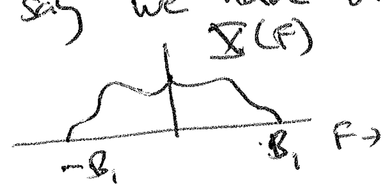
$\to$ AAF $\to$ A/D $\to$ system $\to$ D/A

(1) AAF will never be perfect — always have some leakage

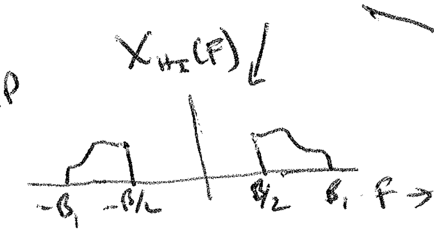
(2) usually try to design so aliased energy has amplitude < 1 bit of A/D (ie less than discretization level)

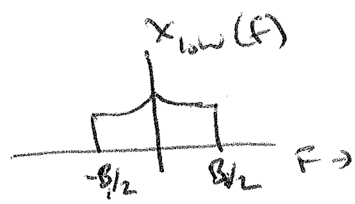
<u>Bandlimited Sampling</u>

<u>Motivation</u>: let's say we have a signal w/ $X(F)$ like:



next, we split it in 2 using <u>perfect</u> HP, LP filters





to sample $X(t)$, we need $F_s \geq 2B_1$

to sample $X_{low}(t)$, need $F_{s_2} \geq 2\ B/2 = B_1$

how about sampling $X_{hi}(t)$?

→ "2 × highest freq" says sample at $\geq 2B_1$

but X_{HI} has less "information" than $x(t)$; is this really needed?

Really, the sampling theorem says:

$$F_s \geq 2B$$

signal bandwidth!

both $X_{low}(F)$ + $X_{HS}(F)$ have <u>same</u> bandwidth

$$B = B_0/2$$

so both can be sampled at lower rate.

Simplest case - (only one we'll consider) is <u>integer band sampling</u>

$$F_H = mB$$

highest freq

integer

signal bandwidth

easiest seen graphically - examples on Trunk