# Sample rate changing

Motivation / applications

1) audio - record at high rate (196 kHz), apply effects, downsample for distribution

2) same ADC used for different signals
→ 10 kHz ~~ADC~~ ADC used to get signals w/ both high + low content

look zero: we could do $\xrightarrow[T_1]{x(n)} \boxed{D/A} \xrightarrow{x(t)} \underset{T_2}{\boxed{ADC}} \rightarrow \tilde{x}(n)$ but that is silly...

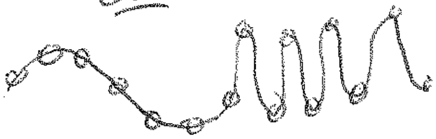
~~Sampling rate reduction (downsampling)~~

First look: upsampling. We can always "add" more samples between the old.


Question: what's the best way?

downsampling: we can just discard samples in between old samples.

old



This is a little dangerous... need to be careful

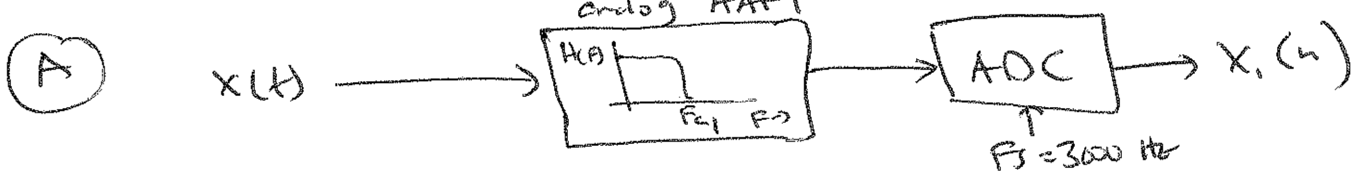
every 2nd

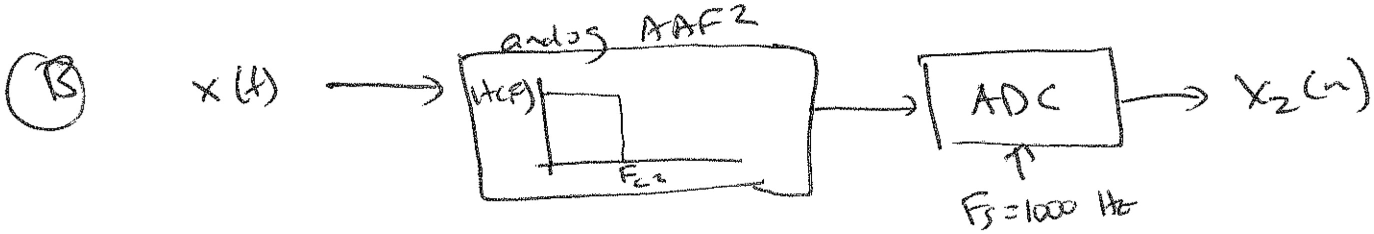

Question: how to avoid aliasing?

## Downsampling

First, think about regular sampling. This shows what the answer should be.

Consider case A + case B




we need $F_{c_1} = 1500$ Hz, and $F_{c_2} = 500$ Hz

so, $x_1(n) \neq x_2(n)$ even if look at every 3rd sample

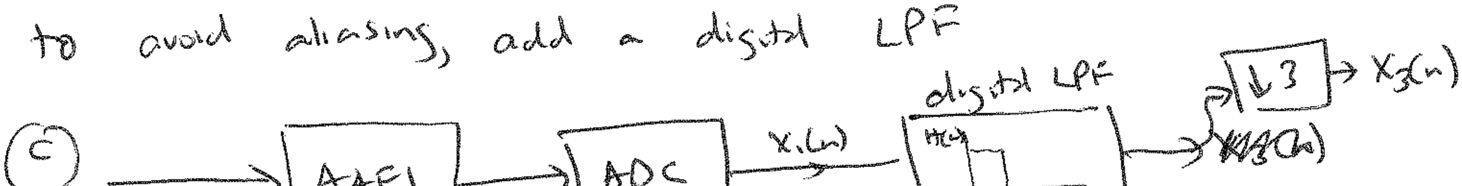
matlab 'downsample' → keep every kth

$\boxed{\downarrow L}$ is symbol

$x_1(n) \to \boxed{\downarrow 3} \to x_{d_1}(n) \neq x_2(n)$

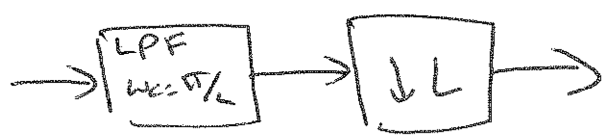
↑ this would be aliased

to avoid aliasing, add a digital LPF




$x_3(n) \approx x_2(n)$

So, ~~downsample~~ decimation looks like



## Sampling rate reduction / downsampling

Pick every $M^{th}$ sample

$$x_d(n) = x(nM) = x_c(nMT)$$

$$\underset{\substack{\text{sample rate}\\ T}}{x(n)} \rightarrow \underset{\text{prefilter}}{\boxed{\begin{matrix}\text{lowpass}\\ \text{cutoff } \pi/M\end{matrix}}} \xrightarrow{\tilde{x}(n)} \underset{\text{downsample}}{\boxed{\downarrow M}} \rightarrow \tilde{x}_d(n) = \tilde{x}(nM)$$

<u>new</u> sample rate $T' = MT$

we get $x_d(n)$, which is the same as if we'd sampled the original $x(t)$ with period $T' = MT$

$\boxed{\text{P+M } 6.5.1}$ analyze $\boxed{\downarrow M}$

ex) $x = [1, 2, 3, 4, ..] \rightarrow \boxed{\downarrow 2} \rightarrow [1, 3, 5, ...]$

Recall from <u>before</u>: after we sample a signal we get

$$X(F) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a(F - kF_s) \qquad F_s = 1/T$$

if we change to new rate $T' = MT$, we get

$$X_d(F) = \frac{1}{MT} \sum_{k=-\infty}^{\infty} X_a(F - kF_s/M)$$

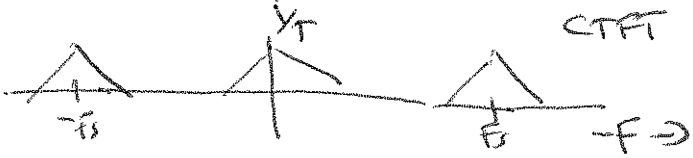
by inspection the difference is a scale factor plus a "stretching" in frequency

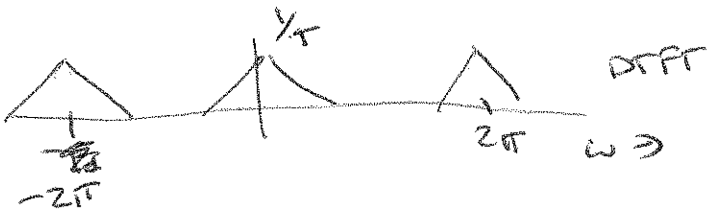
$$X_d(F) = \frac{1}{M} \sum_{k=0}^{M-1} X(F - kF_s/M)$$

(note at $k = M$, we recover original copies)

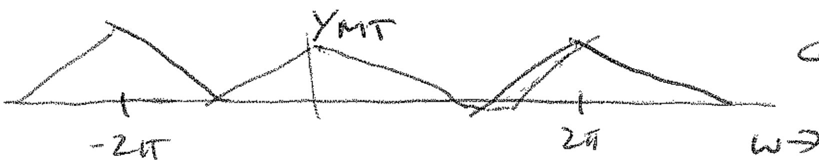
So – looks like old sampled spectrum, but "closer" ~~[illegible]~~

example: see Fig 6.5.1

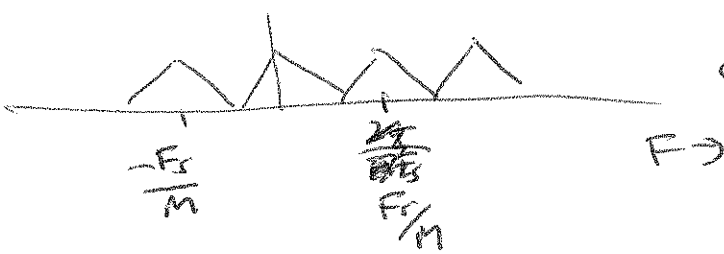




now, downsample by M




← spread out by M in $\omega$




← lower sample rate = more frequent replicas in F

clearly, there could be aliasing.

the prefilter handles this

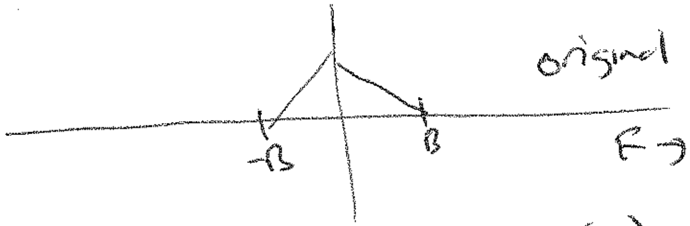
(4)

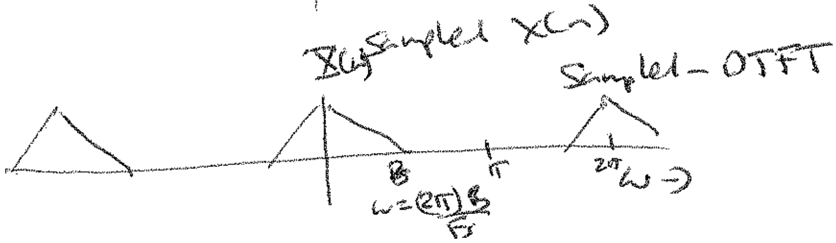
using prefilter

$F_s \rightarrow 2\pi$

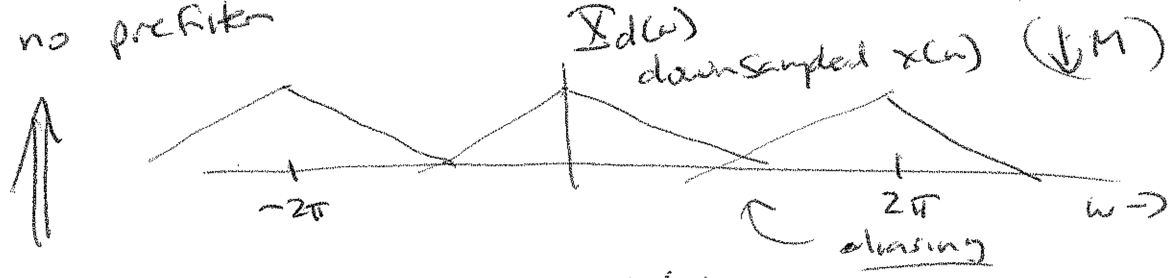
$\frac{\omega}{F} = \frac{2\pi}{F_s}$

$\omega = F\left(\frac{2\pi}{F_s}\right)$

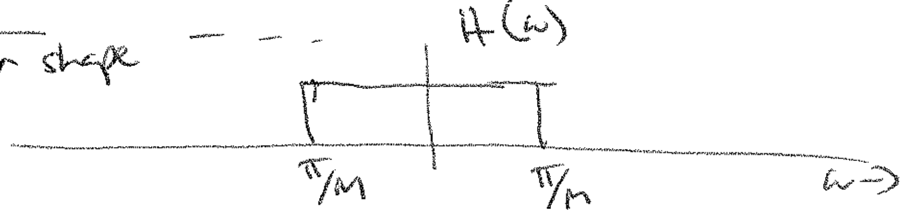


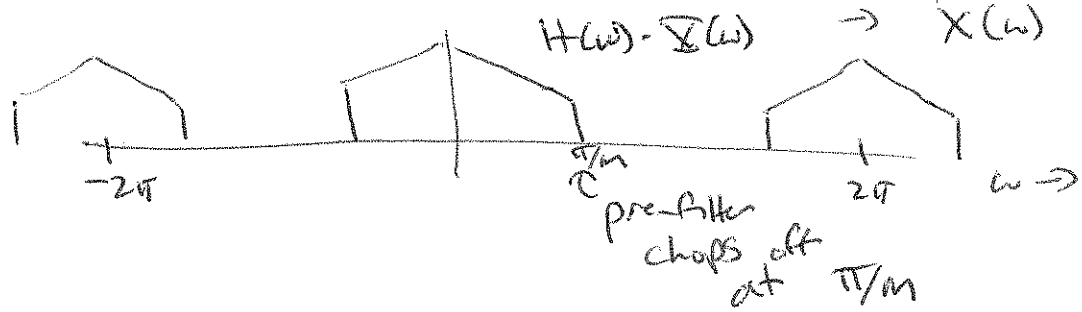

no prefilter

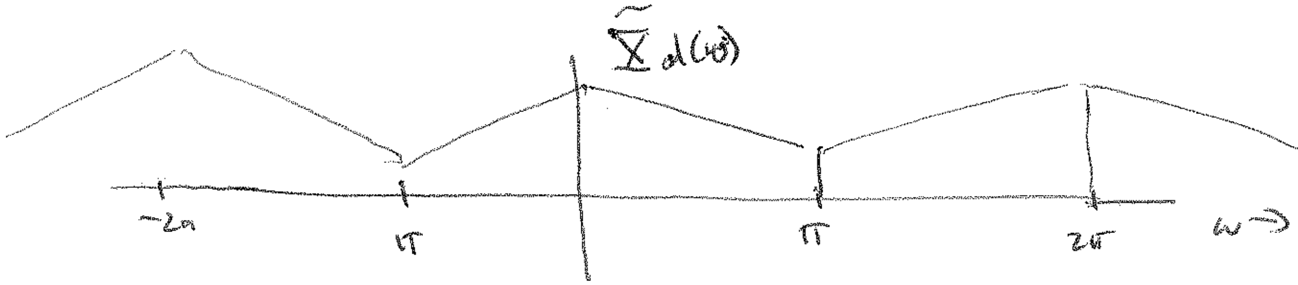



prefilter shape




$\tilde{X} = h * x$




the prefilter helps by chopping to $\pi/M$

so we can expand by M

and have signal $\leq \pi$

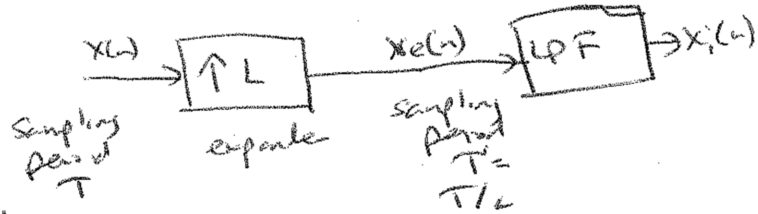
(5)

Increasing the sample rate: upsampling.

This is a problem in interpolation: how do we fill in the missing samples?

example: upsample by 3


Ideal upsampling: goal is to get the signal we'd obtain by reconstructing the signal ~~and~~ into $x(t)$ and resampling with a new $T' = T/L$




2 steps:

1) expanding

$$x_e = \begin{cases} x(n/L) & n = 0, \pm L, \text{etc} \\ 0 & \text{else} \end{cases}$$

2) interpolate

First look at the output of the expander $\boxed{\uparrow L}$; this will help us understand how to interpolate

$$X_e(\omega) = \sum_{n=-\infty}^{\infty} x_e(n) e^{-j\omega n} \quad \text{(DTFT)}$$

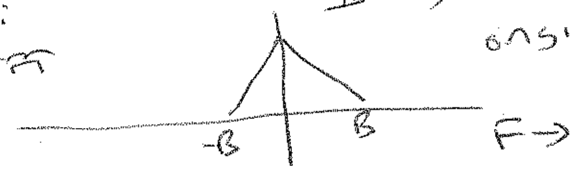
$$= \sum_{n=-\infty}^{\infty} \left( \sum_k x(k)\, \delta(n-kL) \right) e^{-j\omega n} \quad \text{(plug in)}$$

$$= \sum_{k=-\infty}^{\infty} x(k)\, e^{-j\omega L k} \quad \delta \text{ pulls out } n = kL$$

$$= X(\omega L) \quad \Leftarrow \text{so answer is original DT spectrum, scaled by } L$$

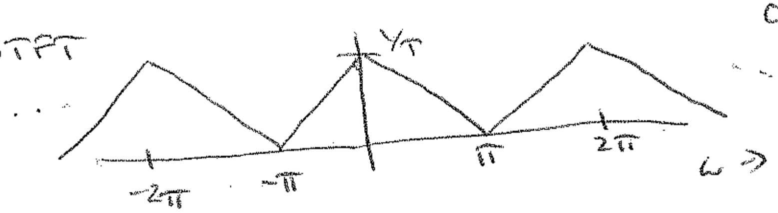
6

picture:
CTFT

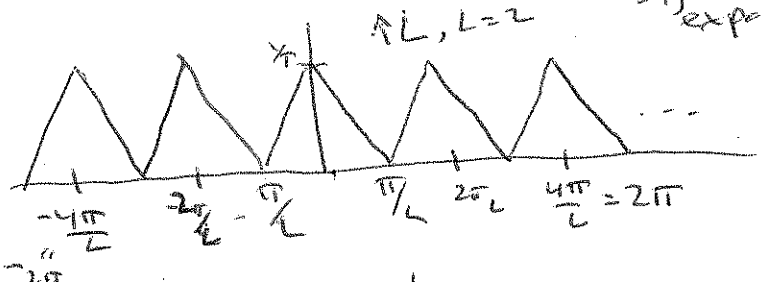



original signal

DTFT

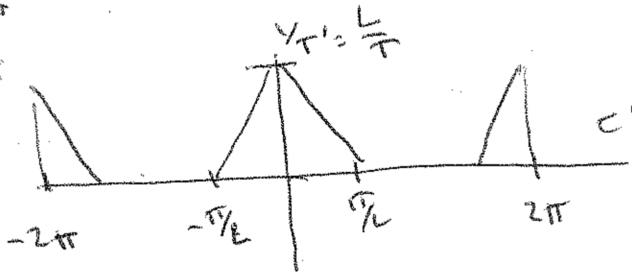



original signal, critically sampled



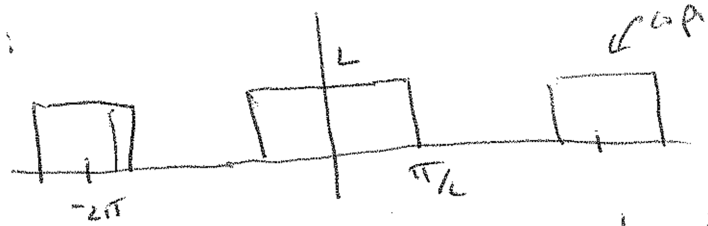

signal above expanded by L

what we ~~want~~:




← this would be result of sampling the original x(t) at T'

so, use filter:




a) multiplying the filter × the signal in frequency is like convolving in time.

b) rectangular function ⟺ sinc function

so $x_i(n) = x_e(n) * \frac{\sin(\pi n/L)}{\pi n L}$

$$= \sum_{k=-\infty}^{\infty} x(k) \frac{\sin(\pi(n-kL)/L)}{\pi(n-kL)/L}$$

//

Q) what's the problem with this?

A) sum over $\infty$ # samples

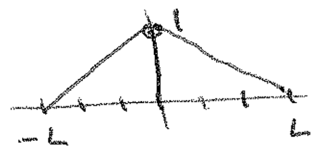
we can <u>truncate</u> it to make it practical

$$x_i(n) = \sum_{k=-K}^{K} x(k) \frac{\sin(\pi(n-kL))}{\pi(n-kL)/L}$$

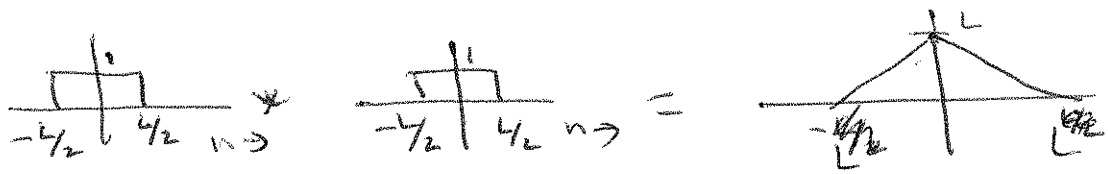
Another <u>Practical</u> interpolation is linear

$$x_i(n) = \sum_{k=-\infty}^{\infty} x(k)\, g(n-kL)$$

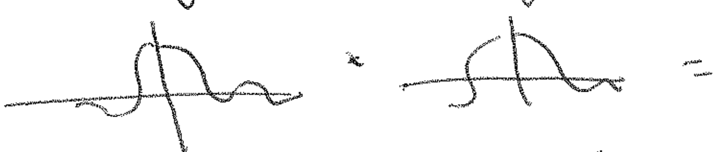
$$g_{lin}(n) = \begin{cases} 1 - \frac{|n|}{L} & |n| < L \\ 0 & \text{else} \end{cases}$$




<u>in class question</u> what is this in freq domain?



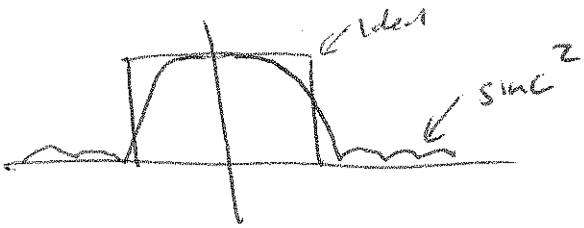

so




$\mathrm{sinc}^2$

$$G_{lin}(\omega) = \frac{1}{L}\left(\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right)^2$$

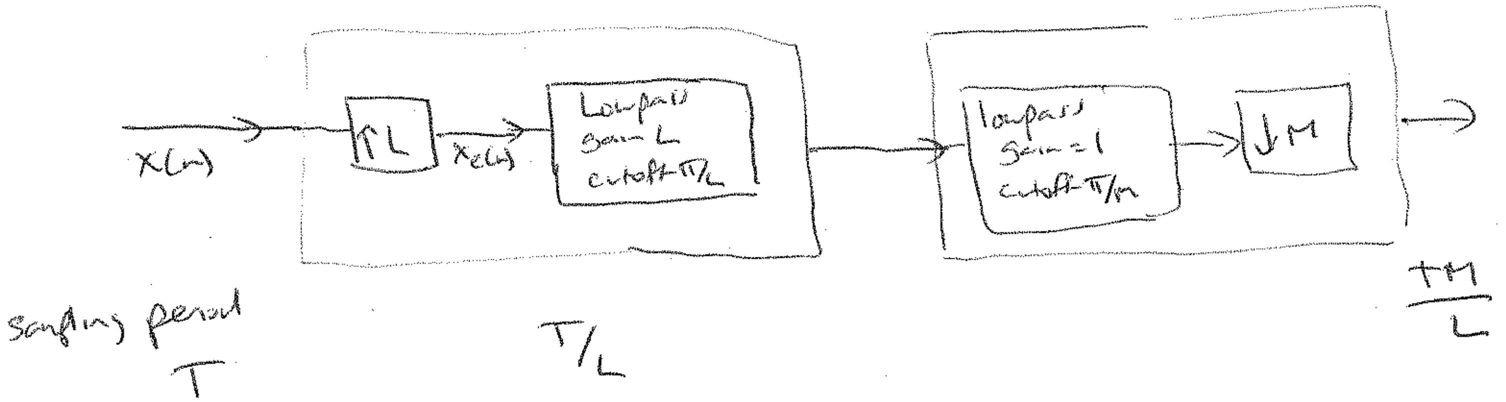
compare to "ideal"



9

Change sample rate by non-integer factor

run together and upsampler & downsampler




combine the two lowpass filters

cutoff $\omega_c = \min(\pi/L, \pi/M)$