follows Oppenheim + Shafer, 10.3, 3rd edition

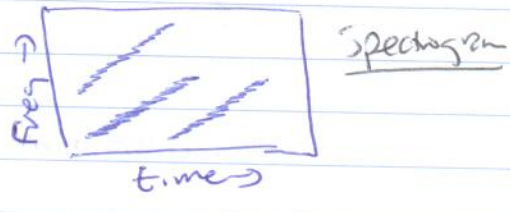
STFT lecture

1. Motivation and uses of short-time Fourier transform (STFT)

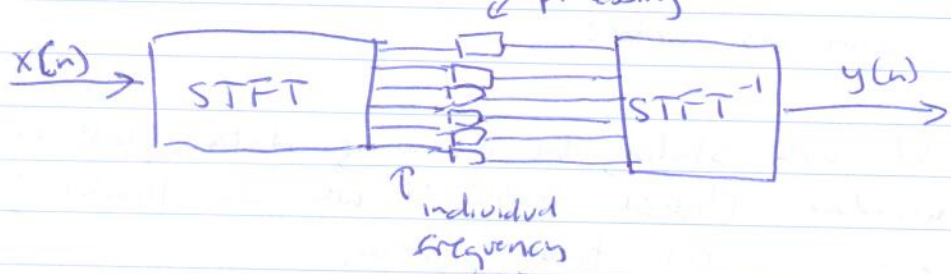
a) analysis of nonstationary signals (speech, etc.)

$$x(n) \rightarrow \boxed{STFT} \xrightarrow{X[n,k]}$$

(may also process $X[n,k]$)



b) audio coding (MPEG) or other frequency-by-frequency processing, followed by <u>reconstruction</u>



2. Basic definition + examples.

Note: we will start by defining things in continuous frequency λ, then move to DFT

def: $$X[n,\lambda) = \sum_{m=-\infty}^{\infty} x(n+m)\, w(m)\, e^{-j\lambda m}$$

<u>in words</u>: at each time, we apply a window so we are just looking at a finite time window around n. We take the DTFT of this window, then move to a new time.

in pictures: chirp example

$\swarrow x(100+m)$

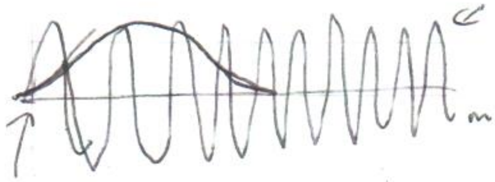
$\nwarrow$ made-up #

$w(m)$

$w(m)$ selects out a portion of the data which is then transformed

later time - chirp frequency is higher

$\swarrow x(400+m)$

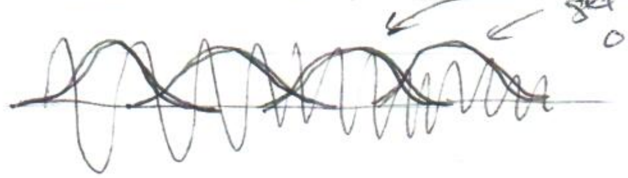


$w(m)$ - same as before.

We are sliding the incoming data past a fixed window. Makes sense if we are thinking about, say, a real-time system.

Equivalently, we could re-define things so window shifts

get a Fourier transformed output for each window



but we'll stick with Oppenheim + Shafer definition

③
STFT

<u>Examples</u> chirp_stft_example.m

<u>note</u>: - time-varying frequency is clear
- short windows good for abrupt transitions
- long windows/tapered windows good for resolution in frequency.

③ Filter bank interpretation + windowing

If we substitute $m' = n+m$, we can get

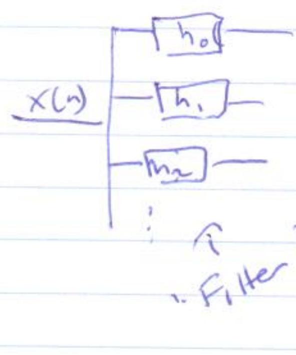
$$\underline{X}[n, \lambda) = \sum_{m'=-\infty}^{\infty} x(m')\, w(-(n-m'))\, e^{j\lambda(n-m')}$$

$$= x(n) * h_\lambda(n)$$

where $h_\lambda(n) \equiv w(-n)\, e^{j\lambda n}$

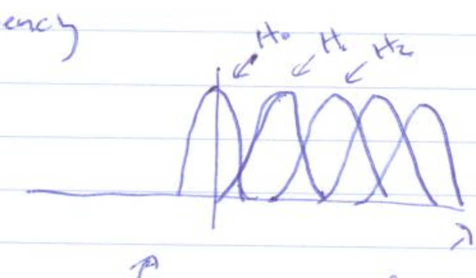
and $H_\lambda(\omega) = W(\lambda-\omega)$

In other words, each DTFT calculation for each λ can be thought of as a separate filter.



Filter bank

in frequency →



We have a bunch of shifted responses of the window response.

(4) - STFT

if we keep n fixed, we can think about effect of multiplying x by w. Multiplication → convolution, so

$$\tilde{X}(\lambda) = X(\omega) * W(\lambda - \omega)$$

so we get smearing, just as before.

Tradeoffs, old + new: <u>table form?</u>

→ longer windows improve frequency resolution

→ smoother windows give lower sidelobes, at cost of resolution

→ (NEW) shorter windows give better <u>time</u> resolution

Obvious conflict between 1st and 3rd - best choice depends on application

(4) Use of DFT/FFT + practical calculation

Now, let's move to DFT + talk about how we actually calculate STFT

→ finite length window: $w[m] = 0$ outside $0 \le m \le L-1$
then, we can define:

$$X[n,k] = X\left(n, \frac{2\pi k}{N}\right) = \sum_{m=0}^{L-1} x(n+m)\, w(m)\, e^{-j(2\pi/N)km} \qquad 0 \le k \le N-1$$

but, we usually don't want/need an output at every sample n!

instead, "skip" by R samples

$$X_r[k] \equiv X[rR, k] = \sum_{m=0}^{L-1} x(rR+m)\, w(m)\, e^{-j(2\pi/N)km}$$

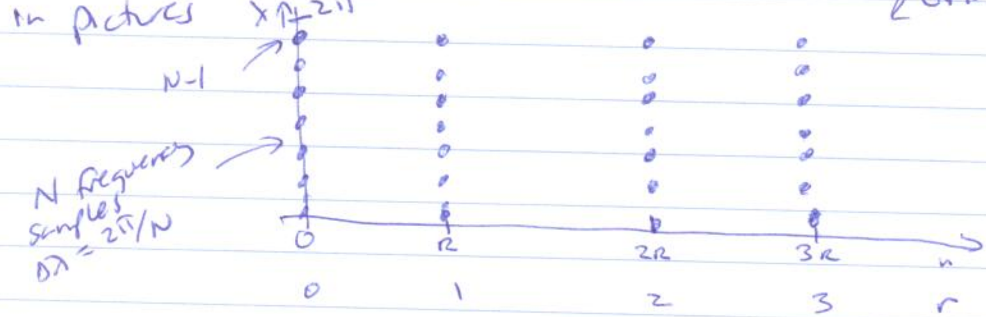
5 STFT

this shows that really we're taking a bunch of N-point DFT's of the windowed segments

$$x_r[m] = x(rR+m)\,w(m)$$

in pictures

← Grid sampling



for analysis: → the window usually overlapped, so

$$R < M$$

no overlap, $R = M$

4 Reconstruction : Just take inverse DFT

Single window: If $X[n,k] = \sum_{m=0}^{L-1} x(n+m)\,w(m)\,e^{-j\left(\frac{2\pi}{N}\right)km}$

then (by IDFT) $x(n+m)\,w(m) = \frac{1}{N}\sum_{k=0}^{N-1} X[n,k]\,e^{j\left(\frac{2\pi}{N}\right)km}$ $\quad 0 \le m \le L-1$

then we can reconstruct using

$$x(n+m) = \frac{1}{N\,w(m)}\sum_{k=0}^{N-1} X[n,k]\,e^{j\left(\frac{2\pi}{N}\right)km} \qquad 0 \le m < L-1$$

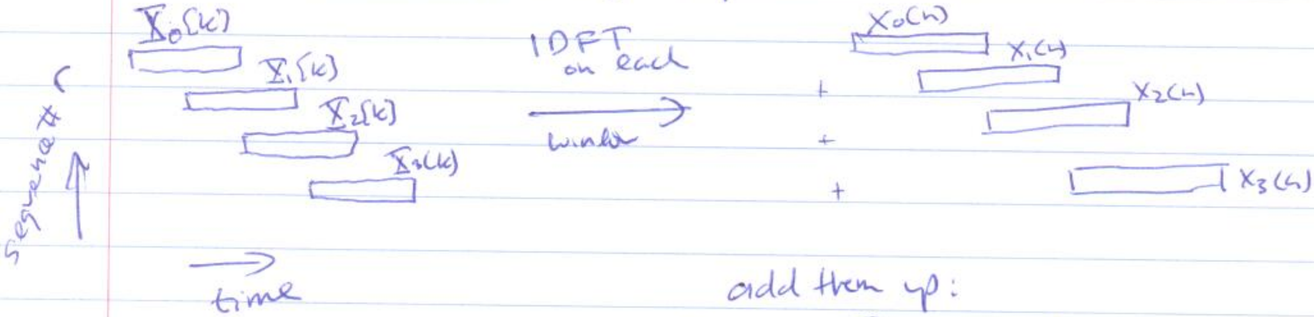
so we can get values from $x(n)$ $(m=0)$

up to $x(n+L-1)$ $(m = L-1)$

Ⓒ STFT

reconstructing multiple windows

we can use same overlap-add idea that we used for filtering long sequences:



add them up:

$$\hat{x}(n) = \sum_{r=-\infty}^{\infty} x_r(n-rR)$$

We have to check: when does $\hat{x}(n) = x(n)$? ⟸ original signal

$$\hat{x}(n) = \sum_{r=-\infty}^{\infty} x(rR+n-rR)\,w(n-rR)$$

$$= x(n) \sum_{r=-\infty}^{\infty} w(n-rR)$$

so we can reconstruct if

$$\tilde{w}(n) = \sum_{r=-\infty}^{\infty} w(n-rR) = C \quad \text{for all } n$$

⟸ constant

what satisfies this?

- rectangular window, no overlap ($C=1$)
- " " , 50% overlap ($C=2$)

- triangle windows, any ~~overlap~~ factor-of-two overlap

- Hann, similar,
- approximately true for other windows

Next steps....

there are many kinds of time-frequency analysis; STFT is just the beginning (though *very* useful)

Wavelets: the concept is that transient signals may better be represented by some shape other than the sinusoid assumed in STFT.

We can pick a "mother ~~wate~~ wavelet" — shape defined by some parameter — and then *scale* + *shift* it in time.

We basically set up a bank of filters, ~~whe~~ where each is a scaled wavelet — get a *scalogram* instead of *spectrogram*

Haar | scaled versions:

Mexican hat