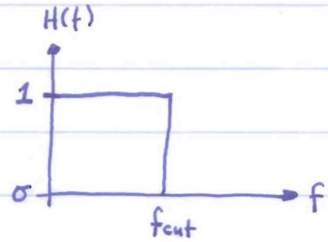
class #7 start

LINEAR FILTERS

Ideal filter



frequency response function
FFT of filter weights

$$Y(f) = H(f) \cdot X(f)$$

filtered output — filter — raw input

↓

$$y(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau = h(t) * x(t)$$

note that for $x(t-\tau) = \delta(t-\tau)$.

$$y(t) = \int_{-\infty}^{\infty} h(\tau)\, \delta(t-\tau)\, d\tau = h(t)$$

↓ $H(f)$ is the FFT of the filtered output due to a unit impulse function.

Filter Design is to come up with "good" filter weights $h(t)$

1. sharp cut-off	causes ringing
2. good transient response	minimize ringing
3. minimize "phase shift"	$H(f)$ real
4. leave low frequency content undisturbed	flat pass-band response
5. short filter span	$h(t)$ drops to zero fast
6. efficient computationally	→ ~~little~~ minimize filterloss

Some of these criteria are contradictory, i.e., a sharp cut off requires a long filterspan (Gibbs phenomena)

Ideal Filter continued:

(4) What does the ideal filter do in the time domain?

$$\mathcal{F}^{-1}(\Pi(f, f_{cut})) = f_{cut}\ \mathrm{sinc}(f_{cut}\, t)$$

what was leakage before is now called ringing

as the data $x(t)$ is convolved with the sinc fctn.

→ smooth the edges of the filter, i.e., taper the FFT coefficients, i.e., apply windowing / tapering techniques.

A Time Domain Non-recursive filters

1. Running Mean, Box Car Filter

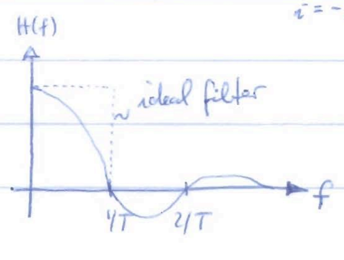
$$h_i = 1/(2N+1)$$ $2N+1$ is # of weights

i.e.,

$$y_t = \sum_{i=-N}^{N} h_i\, x_{t-i} \qquad \int_{-\infty}^{\infty} h(t)\, x(t-\tau)\, d\tau$$

$$\sum_{i=-N}^{N} h_i = 1$$

T is the filter length



kills of variance at $1/T$, $2/T$, $3/T$ etc. but lets through at $1.5/T$, $2.5/T$, $3.5/T$ etc.

(2) Godin Filter

one of the first digital filters
used to remove tides

3 box cars applied successively with

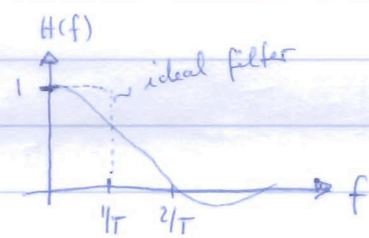
1. $T = 25$ hrs $h_1(t)$
2. $T = 25$ hrs $h_2(t)$
3. $T = 24$ hrs $h_3(t)$

length

$$\mathcal{F}\left(h_1(t) * h_2(t) * h_3(t) \right) = H_1(f) \cdot H_2(f) \cdot H_3(f)$$

real smooth response, too smooth as it also removes frequencies ~~beyon~~ below the cut-off

(3) Full Raised Cosine

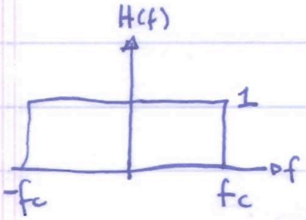


$$h_n = \frac{1}{2N} \left[\frac{1}{2} \left(1 + \cos\left(\frac{2\pi n}{2N+1} \right) \right) \right]$$

$n = -N, \ldots, +N$

Lanczos filter

- attempt to mimic the "ideal" filter with a few ~~bars~~ sine waves



$$H(f, f_c) = \begin{cases} 1 & f \le f_c \\ 0 & f > f_c \end{cases}$$

inverse Fourier transform gives the filter weights

$$h(t = n \cdot \Delta t) = \int_{-\infty}^{\infty} H(f)\, e^{j 2\pi f n \Delta t}\, df =$$

$\mathcal{F}(\Pi(t,T)) = T \,\mathrm{sinc}(fT)$
$\mathcal{F}^{-1}(\Pi(f, f_c)) = f_c \,\mathrm{sinc}(f_c t)$

$$= \underbrace{2 f_c}_{\text{length of window in frequency}} \frac{\sin(2\pi f_c n \cdot \Delta t)}{2\pi f_c n \Delta t}$$

~~$2 f_c$ sinc~~

infinite # of filter weights here → truncation

this is from the Lanczos fortran program

(1) $h_0 = 2 f_c / \Delta t$

(2) $h_n = h_{-n}$

(3) need to normalize filterweights

(4) finite number of ~~filter~~ weights: $2s+1$

given $h_{-s}, h_{-s+1}, \ldots h_0, h_1 \ldots h_s$

what is $\mathcal{F}(h) = \hat{H}$

symmetric, H(f) is real (no imaginary part, no phase shift by design)

$$\hat{H}(f) = \sum_{n=-s}^{s} h_n e^{-j 2\pi f n \Delta t} = h_0 + 2\sum_{n=1}^{S} h_n \cos(2\pi f n \Delta t)$$

$\hat{H}(f)$ is the approximation of the step fct by a limited # of cosine fct.

→ Gibb's phenomena