$\cos 0° \Rightarrow 1$
$\sin 0° \Rightarrow 0$

$$C_k[i] = \cos\left(\frac{2\pi k i}{N}\right) \qquad C_k = \text{Cos wave}$$

(1) Emre ÖZTOKLU

$$S_k[i] = \sin\left(\frac{2\pi k i}{N}\right) \qquad S_k = \text{Sin wave}$$

32 N ⇒ sample points ⇒ 16 + 16

Sin + Cos ⇒ 32 N Sample points

$$\underbrace{Re\bar{X}_{[k]}, Im\bar{X}_{[k]}}_{\text{Amplitudes needed synthesis}} \qquad \underbrace{ReX_{[k]}, ImX_{[k]}}_{\text{frequency domain signals}}$$

$ReX_{[k]}$
$ImX_{[k]}$

$$Z_{[k]} = ReX_{[k]} + ImX_{[k]}$$

$32 = 16 + 16$

$$\boxed{Re\bar{X}_{[k]} = \frac{ReX_{[k]}}{N/2}} \qquad \text{Except} = Re\bar{X}_{[0]} = \frac{ReX_{[k]}}{N}$$

$$\boxed{Im\bar{X}_{[k]} = \frac{ImX_{[k]}}{N/2}} \qquad \text{Except} = Im\bar{X}_{[0]} = \frac{ImX_{[k]}}{N}$$

IDFT Inverse Discreat Fourier Transform.

$$\boxed{x_{[i]} = \sum_{k=0}^{N/2} Re\bar{X}_{[k]} \cdot \cos\left(\frac{2\pi k i}{N}\right) + \sum_{k=0}^{N/2} Im\bar{X}_{[k]} \cdot \sin\left(\frac{2\pi k i}{N}\right)}$$

$i \Rightarrow 0$ to $N-1$

IDFT
Inverse Discreate Fourier Transform

* Frekans domain'deki sinyal'leri Time domain'deki durumunu bulmak için ⇒

$$\boxed{\frac{ReX_{[k]}}{N/2}} \Rightarrow \sum_{k=0}^{N/2} Re\bar{X}_{[k]} \cdot \cos\left(\frac{2\pi k i}{N}\right)$$

(1) ReX Frequency Domain

(2) $Re\bar{X}[\,] =$ Cos. wave amplitude
Freq. sample number

(3) Time domain of Cos

DFT equations

$$\mathrm{Re}X_{[k]} = \sum_{i=0}^{N-1} x_{[i]} \cdot \cos\left(\frac{2\pi k i}{N}\right)$$

$$\mathrm{Im}X_{[k]} = -\sum_{i=0}^{N-1} x_{[i]} \cdot \sin\left(\frac{2\pi k i}{N}\right)$$

$+$

$x_{[0]} \cdot \cos \frac{2\pi k 0}{} =$

$$\boxed{w = \frac{2\pi}{T}}$$

$$\boxed{w = 2\pi f}$$

$$\cos t = \frac{e^{it} + e^{-it}}{2}$$

$$\sin t = \frac{e^{it} - e^{-it}}{2i}$$

$$e^{it} = \cos t + i \sin t$$

$$\boxed{i = \sqrt{-1}}$$

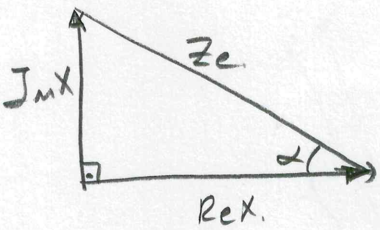
$$\Rightarrow g(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{i \frac{2\pi n t}{T}}$$

$w.$

$$\Rightarrow g(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{i w n t}$$

Magnitude (Büyüklük).

$$Ze_{[k]} = \sqrt{\left(\mathrm{Re}X_{[k]}\right)^2 + \left(\mathrm{Im}X_{[k]}\right)^2}$$



$$\tan\alpha = \frac{\mathrm{Im}X_{[k]}}{\mathrm{Re}X_{[k]}}$$

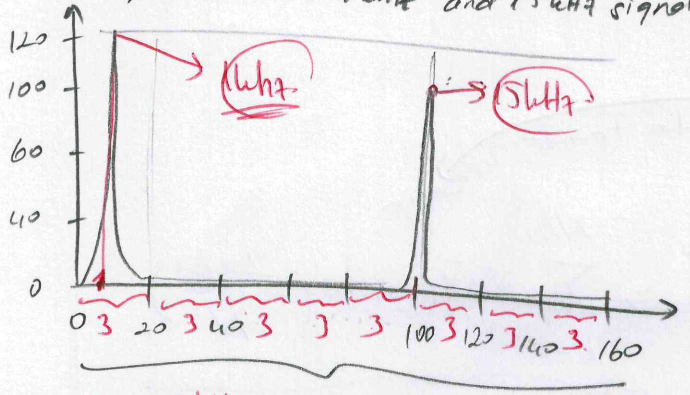
$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha}$$

Nyquist Theory

$Ze = \mathrm{Re}X \cdot \cos\alpha + i \cdot \mathrm{Im}X \cdot \sin\alpha$

$\sin\alpha \cdot \mathrm{Re}X = \cos\alpha \, \mathrm{Im}X$

Magnitude of 1kHz and 15 kHz signal.



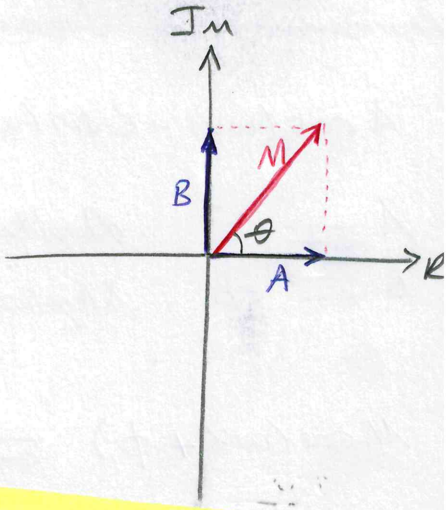
320 samples 48kHz
160 samples 24kHz

8 birim
Aralıklı

24/8 = 3kHz each gap represent
it 3kHz.

Frequency Domain		Time Domain
Single Point	→	Sinusoid
Sinusoid	→	Single Point
Convolution	→	Multiplication
Multiplication	→	Convolution

$\underbrace{ReX_{[k]}, ImX_{[k]}}_{\text{Rectangular Notation}}$ $\underbrace{MagX_{[k]}, Phase_{[k]}}_{\text{Polar Notation}}$



$$A\cos(x) + B\sin(x) = M\cos(x+\theta)$$

$$\boxed{\begin{aligned} Mag[k] &= \sqrt{(ReX_{[k]})^2 + (ImX_{[k]})^2} \\ PhaseX_{[k]} &= \arctan\left(\frac{ImX_{[k]}}{ReX_{[k]}}\right) \end{aligned}} \qquad \boxed{\begin{aligned} ReX_{[k]} &= MagX_{[k]} \cdot \cos(PhaseX_{[k]}) \\ ImX_{[k]} &= MagX_{[k]} \cdot \sin(PhaseX_{[k]}) \end{aligned}}$$

Polar to Rectangular Conversion

*Frequency Response → Amplitude and phase of output cosine wave.

$$x[n] * h[n] = y[n]$$

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

IDFT ↑ ↓ DFT IDFT ↑ ↓ DFT IDFT ↑ ↓ DFT

$$X[n] \times H[n] = Y[n]$$

$$X[n] \longrightarrow \boxed{H[n]} \longrightarrow Y[n]$$

Impulse Response
↓
Frequency Response

Polar Notation

$M = \sqrt{Re^2 + Im^2}$

$\theta = \arctan\left(\frac{Im}{Re}\right)$

$Re = M.\cos(\theta)$

$Im = M.\sin(\theta)$

$a+jb = M.(\cos\theta + j\sin\theta)$

$e^{jx} = \cos x + j\sin x$.

$a+jb = M.e^{j\theta}$

Multiplication

$M_1.e^{j\theta_1} . M_2.e^{j\theta_2} = M_1.M_2.e^{j(\theta_1+\theta_2)}$

Division

$\frac{M_1.e^{j\theta_1}}{M_2.e^{j\theta_2}} = \left[\frac{M_1}{M_2}\right].e^{j(\theta_1-\theta_2)}$

$f = \frac{1}{T}$ freq f Period T

Representation of Sinusoids

$\omega = 2\pi f = \frac{2\pi}{T}$

$$A.\cos(\omega x) + B.\sin(\omega x) \leftrightarrows a + jb.$$

$A \rightleftarrows a$: Amplitude of cosine wave
$B \rightleftarrows -b$: Negative amplitude of sine wave

$$M.\cos(\omega t + \phi) \rightleftarrows M.e^{j\theta}$$

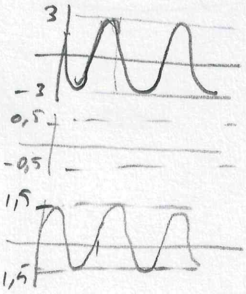
$M \rightleftarrows M$: Amplitude of cosine wave
$\phi \rightleftarrows -\phi$: Negative amplitude of sine wave

Ex: $3.\cos(\omega t + \pi/4) \rightarrow 3.e^{-j\pi/4}$

$\quad 0{,}5.\cos(\omega t - 3\pi/8) \rightarrow 0{,}5.e^{+j3\pi/8}$

$\times$

$\quad 1{,}5.\cos(\omega t + \pi/8) \rightarrow 1{,}5.e^{-j\pi/8}$



Complex Fourier Transform

$$Re\,X_{[k]} = \frac{2}{N}\sum_{n=0}^{N-1} x_{[n]}.\cos\left(\frac{2\pi kn}{N}\right)$$

$$Im\,X_{[k]} = -\frac{2}{N}\sum_{n=0}^{N-1} x_{[n]}.\sin\left(\frac{2\pi kn}{N}\right)$$

DFT equations

$$Re\,X_{[k]} = \sum_{i=0}^{N-1} x_{[i]}.\cos\left(\frac{2\pi k.i}{N}\right)$$

$$Im\,X_{[k]} = -\sum_{i=0}^{N-1} x_{[i]}.\sin\left(\frac{2\pi ki}{N}\right)$$

Matematical Equivalence

$$e^{jx} = \cos(x) + j\sin(x)$$

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2}$$

$$\sin(x) = \frac{e^{jx} - e^{-jx}}{2}$$

$$\cos(\omega t) = \frac{1}{2}.e^{j(-\omega)t} + \frac{1}{2}e^{j\omega t}$$

$$\sin(\omega t) = \frac{1}{2}e^{j(-\omega)t} - \frac{1}{2}e^{j\omega t}$$

Complex DFT

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n].e^{-j2\pi kn/N}$$

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n].\left(\underbrace{\cos\left(\frac{2\pi kn}{N}\right)}_{SR} - j\underbrace{\sin\left(\frac{2\pi kn}{N}\right)}_{SI}\right)$$

k → for 1
n → for 1 } N-1

Real DFT

$$Re\,X_{[k]} = \frac{2}{N}\sum_{i=0}^{N-1} x_{[i]}.\cos\left(\frac{2\pi ki}{N}\right)$$

$$Im\,X_{[k]} = -\frac{2}{N}\sum_{i=0}^{N-1} x_{[i]}.\sin\left(\frac{2\pi ki}{N}\right)$$

Complex DFT

$$X_{[k]} = \frac{1}{N}\sum_{i=0}^{N-1} x_{[n]}.\left(\overbrace{\cos\left(\frac{2\pi kn}{N}\right)}^{SR} - j\overbrace{\sin\left(\frac{2\pi kn}{N}\right)}^{SI}\right)$$

Inverse Complex DFT

$$X_{[k]} = \frac{1}{N}\sum_{i=0}^{N-1} x[n].e^{j2\pi kn/N}$$

How FFT works? (fast Fourier Transform)

N points : X[0] to X[N-1]

$\swarrow$ ReX[0] $\qquad \searrow$ ImX[0]

Answer

① N points time domain signal $\xrightarrow{\text{Decomposition}}$ N time domain signals each made of signal points.

② Calculate N frequency spectra corresponding to these N time domain signals

③ Synthesize N frequency spectra into a single frequency spectrum.

Interlaced Decomposition → Even Samples
→ Odd Samples

N points : $\log_2 N$ stages of decomposition

Ex: 16 points signal (2^4) : 4 stages.
512 points signal (2^7) : 7 stages.
4096 points signal (2^{12}) : 12 stages.

Bit reversal

Normal Sample Order		⇒	Bit reversed Order	
0 →	0000		0	0000
1 →	0001		8	1000
2 →	0010		4	0100
$\boxed{3 \rightarrow 0011}$		—	$\boxed{12 \quad 1100}$	
4 →	0100		2	0010
⋮	⋮		10	1010
			6	0110
			14	1110
			⋮	⋮
13 →	1101			
$\boxed{14 \rightarrow 1110}$		—	$\boxed{7 \quad 0111}$	
15 →	1111		15 ·	1111

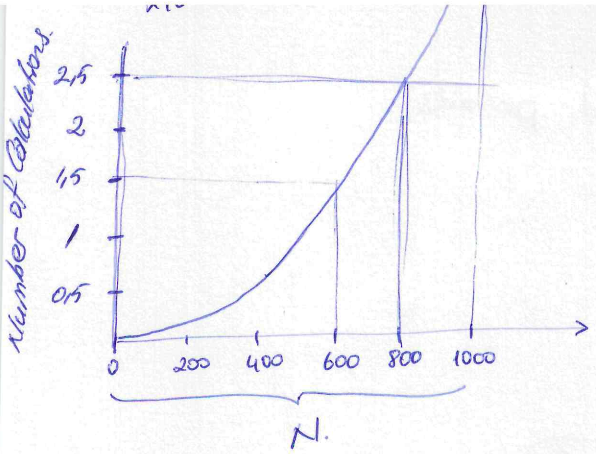
Complex DFT equation

$$X[k] = \frac{1}{N}\sum_{i=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N} \quad \rightarrow \text{equation 1}$$

$$x(\phi) \cdot e^{-j\frac{2\pi}{N}k\times\phi},\ x(1) \cdot e^{-j\frac{2\pi}{N}k\times 1},\ x(2) \cdot e^{-j\frac{2\pi}{N}\cdot k\times 2},\ \ldots,\ x(N-1) \cdot e^{-j\frac{2\pi}{N}k\times(N-1)}$$

each DFT coefficent : $2N+2(N-1) = 4N-2$ real addittions.

N-point DFT : $4N^2$ real multiplications and $N(4N-2)$ realAdditions.



$\Rightarrow$ proportional

Decimation-in-Time FFT Algorithm

Ex:

$$X[k] = \frac{1}{N}\sum_{i=0}^{N-1} x[n]\cdot e^{-j2\pi kn/N} \rightarrow \text{1 eq.}$$

$N=8$

$$X[k] = x(0).e^{-j\frac{2\pi}{8}k.0} + x(1).e^{-j\frac{2\pi}{8}.k.1} + x(2).e^{-j\frac{2\pi}{8}.k.2} + \dots + x(7).e^{-j\frac{2\pi}{8}.k.7} \rightarrow \text{2 eq}$$

Terms with even indices: $x(0), x(2), x(4), x(6)$

$$G(k) = x(0).e^{-j\frac{2\pi}{8}.k.0} + x(2).e^{-j\frac{2\pi}{8}k.2} + x(4).e^{-j\frac{2\pi}{8}.k.4} + x(6).e^{-j\frac{2\pi}{8}.k.6} \rightarrow \text{3 eq}$$

$$G(k) = x(0).e^{0} + x(2).e^{-j\frac{2\pi}{4}k.1} + x(4).e^{-j\frac{2\pi}{4}k.2} + x(6).e^{-j\frac{2\pi}{4}.k.3} \rightarrow \text{4 eq}$$

Terms with odd indices: $x(1), x(3), x(5), x(7)$.

$$H_1(k) = x(1).e^{-j\frac{2\pi}{8}k.1} + x(3).e^{-j\frac{2\pi}{8}.k.3} + x(5).e^{-j\frac{2\pi}{8}k.5} + x(7).e^{-j\frac{2\pi}{8}.k.7} \rightarrow \text{5 eq}$$

$$H_1(k) = e^{-j\frac{2\pi}{8}.k.1}\left(x(1).\underbrace{e^{-j\frac{2\pi}{8}.k.0}}_{e^0=1} + x(3).e^{-j\frac{2\pi}{8}k.2} + x(5).e^{-j\frac{2\pi}{8}k.4} + x(7).e^{-j\frac{2\pi}{8}.k.6}\right) \rightarrow \text{eq 6}$$

$$H_1(k) = e^{-j\frac{2\pi}{8}.k.1}\underbrace{\left(x(1).\underbrace{e^{-j\frac{2\pi}{4}k.0}}_{e^0=1} + x(3).e^{-j\frac{2\pi}{4}.k.1} + x(5).e^{-j\frac{2\pi}{4}.k.2} + x(7).e^{-j\frac{2\pi}{4}.k.3}\right)}_{H(k)} \rightarrow \text{eq 7}$$

$H(k). \rightarrow \boxed{\text{eq 8}}$

$$\boxed{X(k) = G(k) + e^{-j\frac{2\pi}{8}.k\times 1}\cdot H(k)} \rightarrow \boxed{\text{eq}=9}$$

$N=8$ (4) (4)

$$\boxed{e^{-j\frac{2\pi}{8}k.n}}$$: Periodic function of k with N periods.

: k is from 0 to 7

$\boxed{eg\,2} \rightarrow X(k) = X(0).e^{-j\frac{2\pi}{8}k.0} + \dots + X(7).e^{-j\frac{2\pi}{8}.k.7}$

$$\boxed{X(k) = \underbrace{G(k)}_{\boxed{eg\,4}} + e^{-j\frac{2\pi}{8}k.1} . \underbrace{H(k)}_{\boxed{eg\,8}}} \rightarrow \boxed{eg\,9}$$

$$G(k) = X(0).e^{-j\frac{2\pi}{4}k.0} + X(2).e^{-j\frac{2\pi}{4}.k.1} + X(4).e^{-j\frac{2\pi}{4}.k\,2} + X(6).e^{-j\frac{2\pi}{4}.k.3}$$

$$H(k) = X(1).e^{-j\frac{2\pi}{4}.k.0} + X(3)..e^{-j\frac{2\pi}{4}.k\,1} + X(5).e^{-j\frac{2\pi}{4}.k\,2} + X(7).e^{-j\frac{2\pi}{4}.k.3}$$

Ex Calculating $X(k)$ and $X(k+4)$

$$X(k) = G(k) + e^{-\frac{2\pi}{8}.k.1} . H(k) \rightarrow \boxed{eg.10}$$

$$X(k+4) = G(k+4) + e^{-\frac{2\pi}{8}(k+4).1} . H(k+4) \rightarrow \boxed{eg\,11}$$

$\boxed{\text{Note That}} \Rightarrow G(k)$ and $H(k)$ is periodic.

So; $\boxed{X(k+4) = G(k) + e^{-j\frac{2\pi}{8}(k+4).1} . H(k)} \rightarrow$ eg.12

8 point DFT: $4N^2$, $N=8$ $\Rightarrow 4.(8)^2 \Rightarrow 256$ point $\longrightarrow$ this is loop slow.

Same result

Two 4point DFT: $\underbrace{4(4)^2}_{64} + \underbrace{4.(4)^2}_{64} \rightarrow 128$

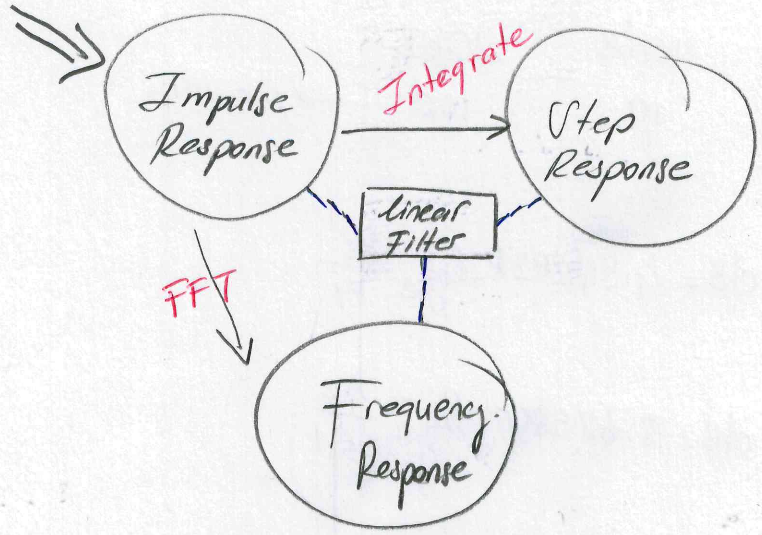
4x1 extra: $(4\times4) + (4\times4) = 32$

} 160 point. $\downarrow$ this is faster calc.

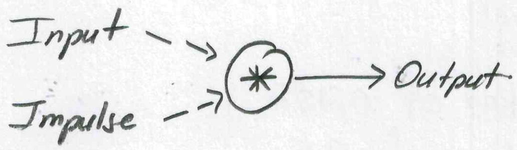
Digital Filter Design

Uses for → Signal Separation
→ Signal Restoration

Every filter has



Filter Kernel (recursion)



Impulse, Step, Frequency Response

Impulse Response : Response of System's output
Step Response: response is the output when the input a step (Edge Respons
freq. Response: Can find by taking DFT using FFT algorithm

logaritmic Scale & Decibel

bel: Power Change by factor of 10

Eg: 4 bels = 10x10x10x10 = 10000

decibel: (dB): one-tenth of a bel

-20dB	-10dB	0dB	10dB	20dB	
0,01	0,1	1	10	100	→ power.

$$dB = 10 \cdot \log_{10} \frac{P_2}{P_1} \qquad dB = 4.342925 \log_e \frac{P_2}{P_1}$$

$$dB = 20 \cdot \log_{10} \frac{X_2}{A_1} \qquad dB = 8.685890 \log_e \frac{A_2}{A_1}$$

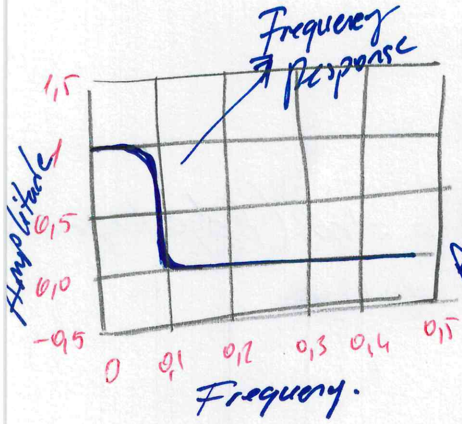
$\log_e x = \ln x.$

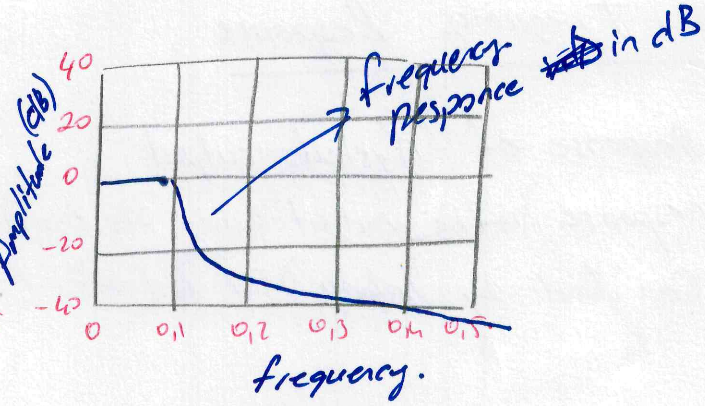
A = Amplitude

P = Power

dBV = 1 volt rms signal

-3dB = amplitude reduction of 0,707
power reduction of 0,5.



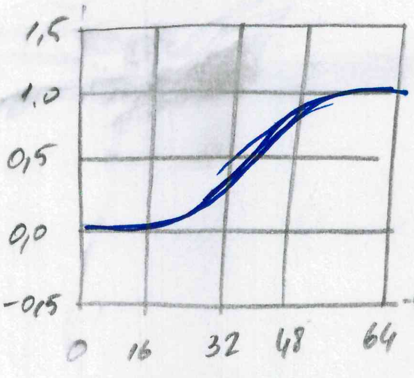


Signal Information Representation

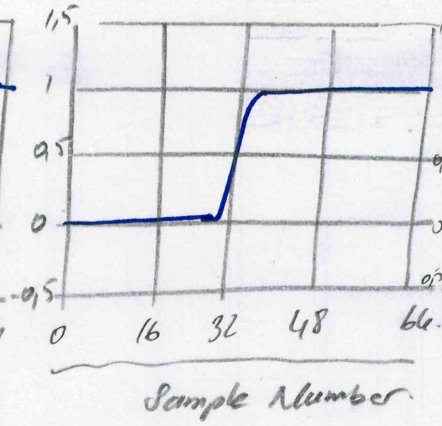
Time domain Information
Frequency domain Information

Time Domain Parameters

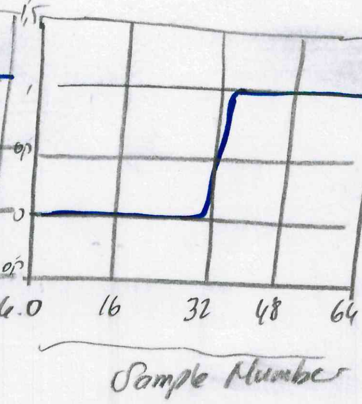
① Step Response → it should be fast Response



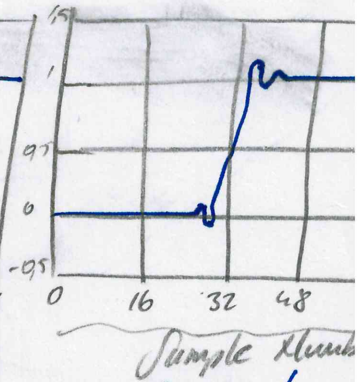
Slow Step Response



Fast Step Response



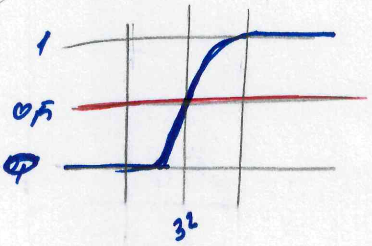
No OverShoot



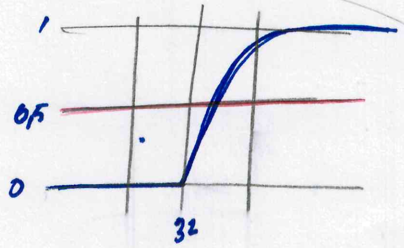
Over Shoot

② OverShoot

(*) Over Shoot has to eliminate it.



linear Phase



Non-linear Phase

③ → Phase linearity.

① Step Response
② OverShoot
③ Phase linearity

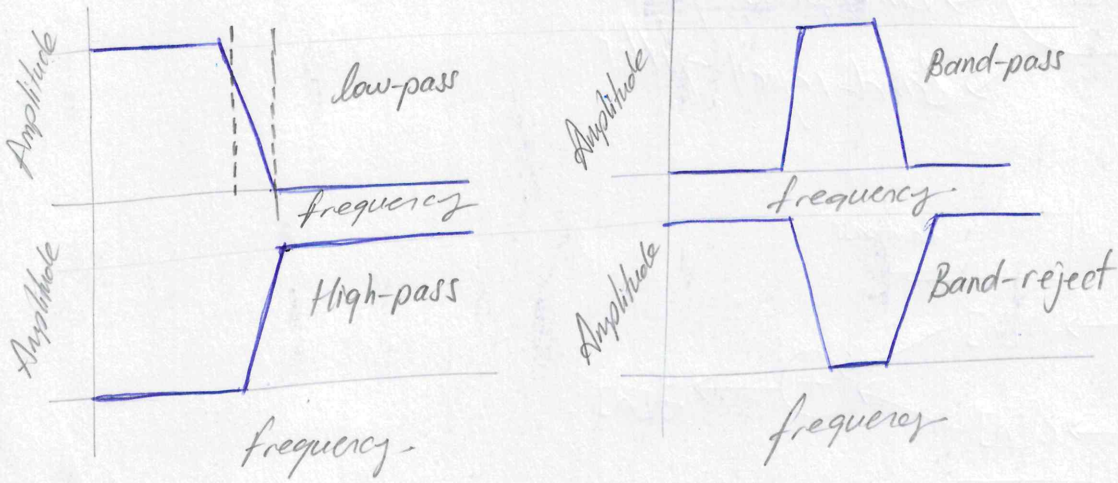
Frequency Domain Parameters

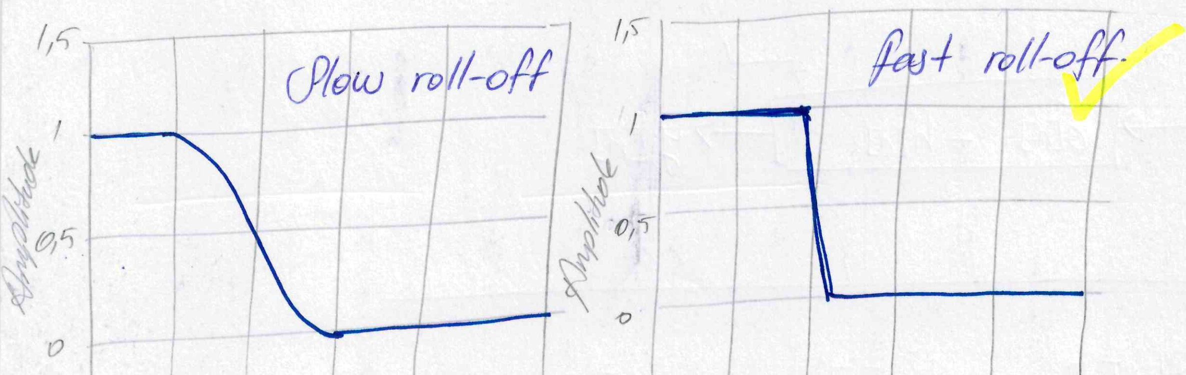
Passband: Frequencies allowed to pass

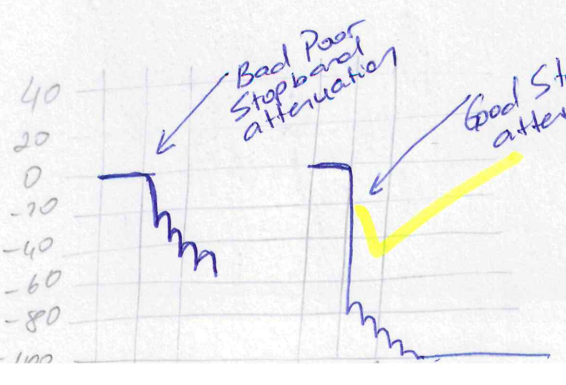
Transition band: Frequencies between passband and stopband.

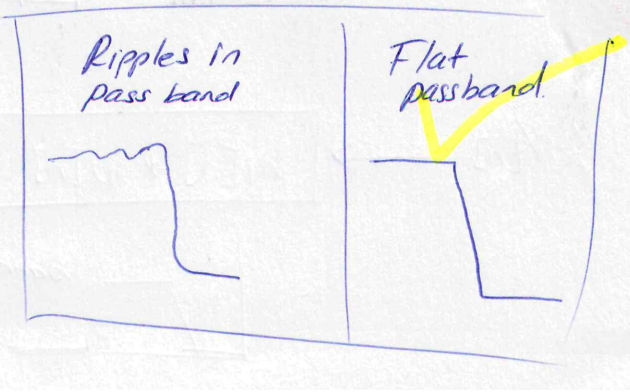
Stop band: Frequencies blocked

Fast Roll-off: Narrow transmission band.







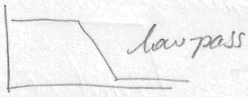


Filter Design Using Spectral Inversion

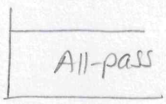
$$\text{low-pass filter} \longrightarrow \text{Spectral Inversion} \longrightarrow \begin{cases} \text{High-pass filter} \\ \text{Band-pass filter} \\ \text{Band-reject filter} \end{cases}$$

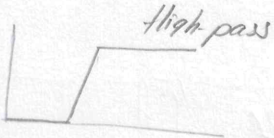
$$\text{low-pass filter} \longrightarrow \text{Spectral Reversal} \longrightarrow \begin{cases} \text{High-pass filter} \\ \text{Band-pass filter} \\ \text{Band-reject filter} \end{cases}$$

(1)



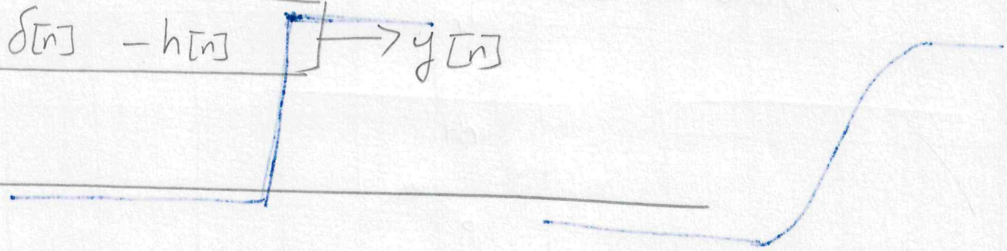
$$\begin{tikzcd} & h[n] \arrow[dr] & & \\ x[n] \arrow[ur] \arrow[dr] & & + \arrow[r] & y[n] \\ & \delta[n] \arrow[ur] & & \end{tikzcd}$$





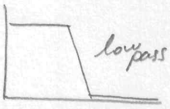
(2)

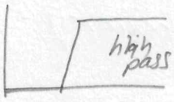
$$x[n] \longrightarrow \boxed{\delta[n] \;-\; h[n]} \longrightarrow y[n]$$



(1)

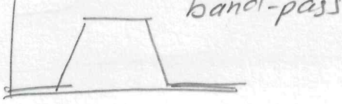
$$x[n] \longrightarrow h_1[n] \longrightarrow h_2[n] \longrightarrow y[n]$$





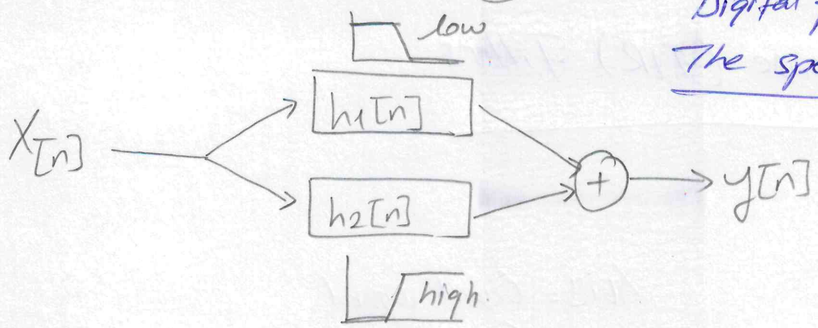
(2)

$$x[n] \longrightarrow \boxed{h_1[n] * h_2[n]} \longrightarrow y[n]$$

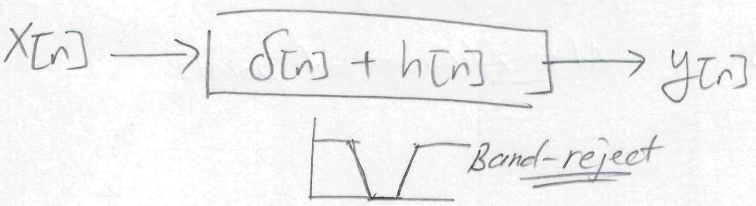


Digital filters using The spectral Reversal Metod.

(1)



(2)



Classification of digital filters

By Metod of Implementation

BY USE

Convolution	Recursion
Finite Impulse Reponse (FIR)	Infinite Impulse Response (IIR)
Moving Average	Single Pole
Windowed-Sinc	Chebyshev
Custom FIR	Custom IIR

Finite Impulse Response (FIR) Filters

① Moving Average filters

$$y[i] = \frac{1}{M}\sum_{j=0}^{M-1} x[i+j]$$

$x[i]$ = input signal.
$y[i]$ = output signal
M = Number of points

Ex Point 50 in a 5-point moving average

$$y[50] = \frac{x[50]+x[51]+x[52]+x[53]+x[54]}{5}$$

Using symmetrically chosen points.

$$y[50] = \frac{x[48]+x[49]+x[50]+x[51]+x[52]}{5}$$

* Amount of noise reduction is equal to the square-root of number of point averaged
eg: 100-point moving avarage → Noise reduction by factor of 10

frequency Response of Moving-Average

$$H[f] = \frac{\sin(\pi f M)}{M.\sin(\pi f)}$$

M = Numbers of points
f = Runs between 0 and 0,5
When $f=0 \Rightarrow H[f]=1$.

Note.. • average filter is role is very slow
• moving av. filter cannot separate one band of freq. from another

time domain performance is poor

if u design low-pass filter dont use moving average filter

The multipable-Pass Moving Average filter

The Recursiver Moving Average filter

$$y[50] = X[47] + X[48] + X[49] + X[50] + X[51] + X[52] + X[53]$$

$$y[51] = X[48] + X[49] + \ldots X[53] + X[54]$$

$$y[51] = y[50] + X[54] - X[47]$$

$\boxed{\text{The Recursive Moving-Average Algorithm}}$

(*)

$$\underset{51}{y[i]} = \underset{51-1}{y[i-1]} + \underset{51+3}{X[i+p]} - \underset{51-4}{X[i-q]}$$

where ⇒ $p = \frac{M-1}{2}$ M: Member of points in moving-average

$q = p+1$

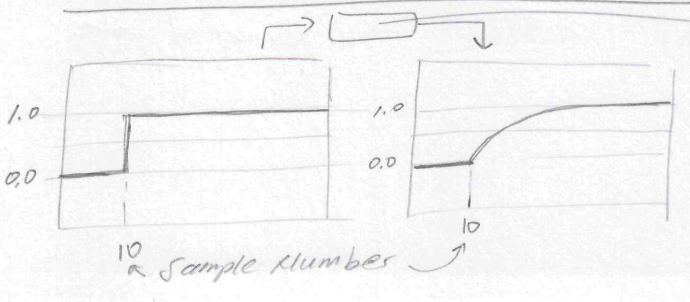
Infinite Impulse Response (IIR) Filters

* Also known as recursive filters
* long impulse response short convolution
* Rapid execution, less peformanee, less flexibility.

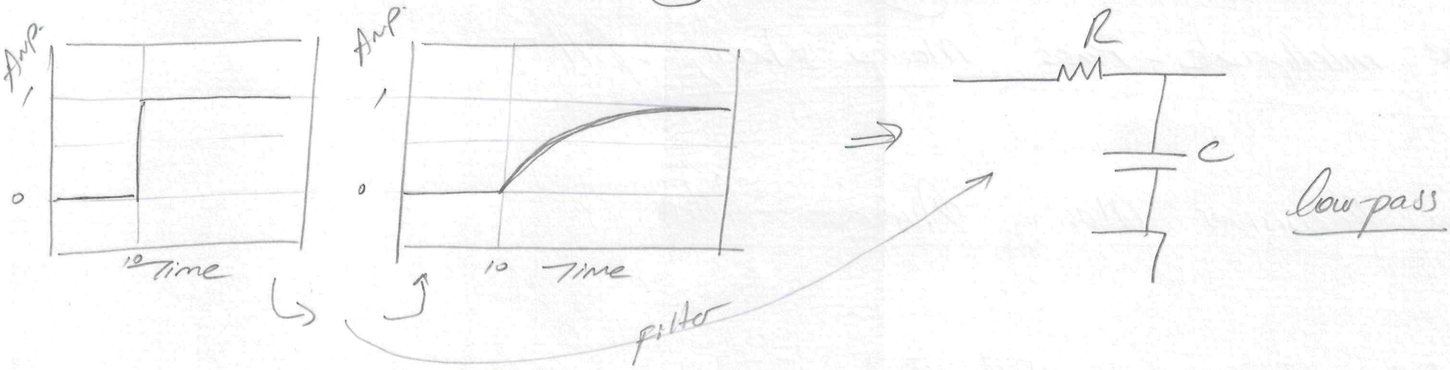
The recursion equation

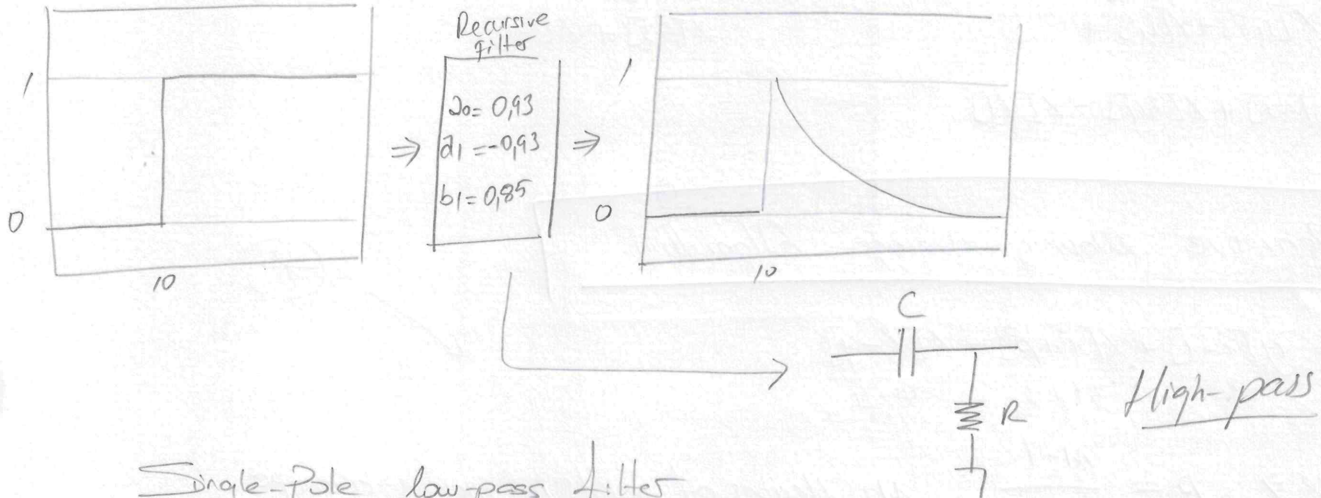
$$y[n] = a_0 X[n] + a_1 X[n-1] + a_2 . X[n-2] + a_3 X[n-3] + \ldots$$
$$+ b_1 Y[n-1] + b_2 . Y[n-2] + b_3 . Y[n-3] + \ldots$$

The Single-Pole Recursive Filter



→ Recursive filter
$a_0 = 0,15$
$b_1 = 0,85$





Single-Pole low-pass filter

$a_0 = 1 - X$

$b_1 = X$

Single-Pole high-pass filter

$a_0 = (1+X)/2$

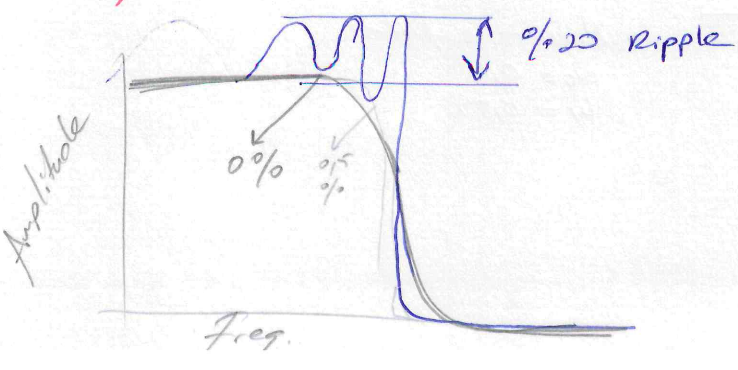
$a_1 = -(1+X)/2$

$b_1 = X$

Digital Chebyshev Filter

to seperate one band of freq from another.

- for frequency bands separation
- less performance compared to windowed-sinc
- faster than windowed-sinc



Designing a Chebyshev

1. High-Pass or low-pass respon
2. Cut-off frequency
3. Percentage of passband ripp
4. Number of poles

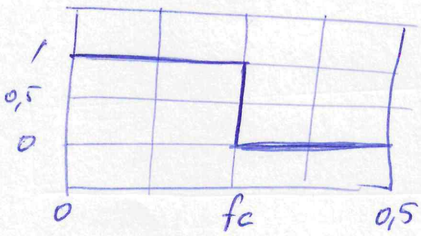
Windowed-Sinc Filters

to separate one band of freq. from another

- Frequency band separation
- Bad time domain performance

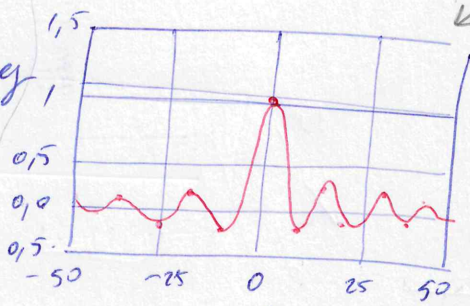
Truncated
↳ kesilmiş
kırpılmış

The Sinc Function and Truncated Sinc filter



⟶ ideal frequency Response

↳ cutoff freq.



↙ Sinc function

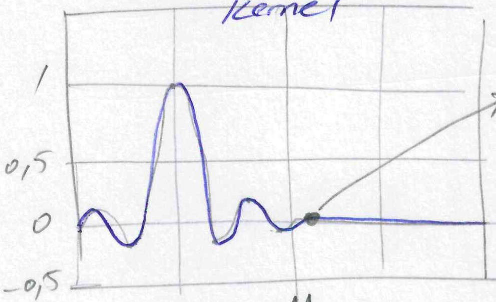
$$\frac{\sin(x)}{x}$$

↳ ideal filter Kernel

$$h[i] = \frac{\sin(2\pi f_c i)}{i\pi}$$

Implementation Software

Truncated-sinc filter Kernel



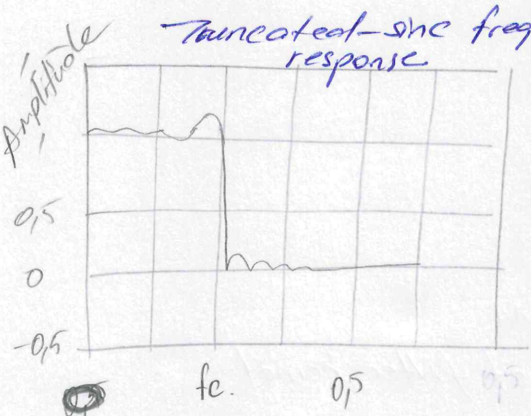
↗ abrupt end.

Sample number

Truncated Sinc Filters

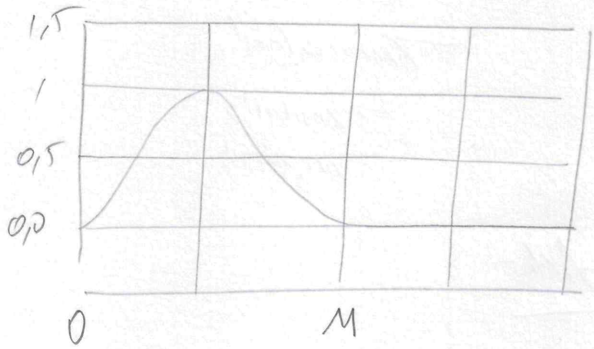
1. Points truncated to M-1 points, symetrically chosen around the lobe
2. All points outside M+1 points are set to zero i.e Ignored
3. Entire sequence is shifted to right so that it runs from 0 to M

Truncated-sinc frequency response

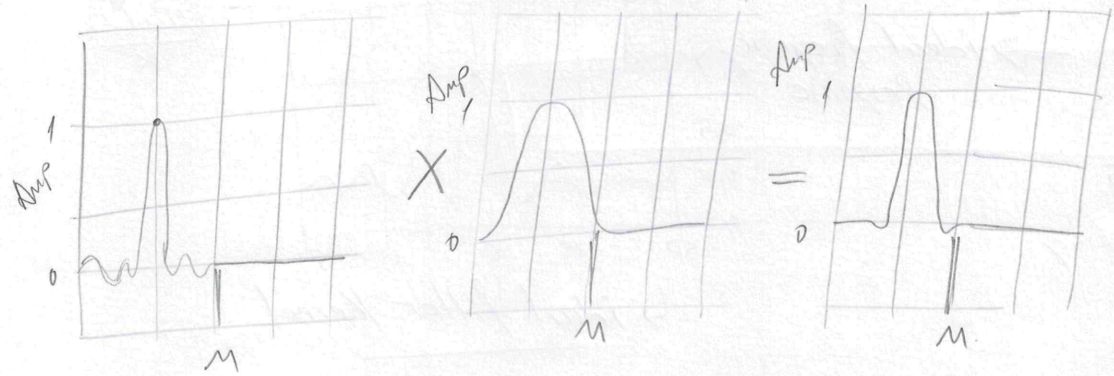


frequency

Blackman or Hamming Window



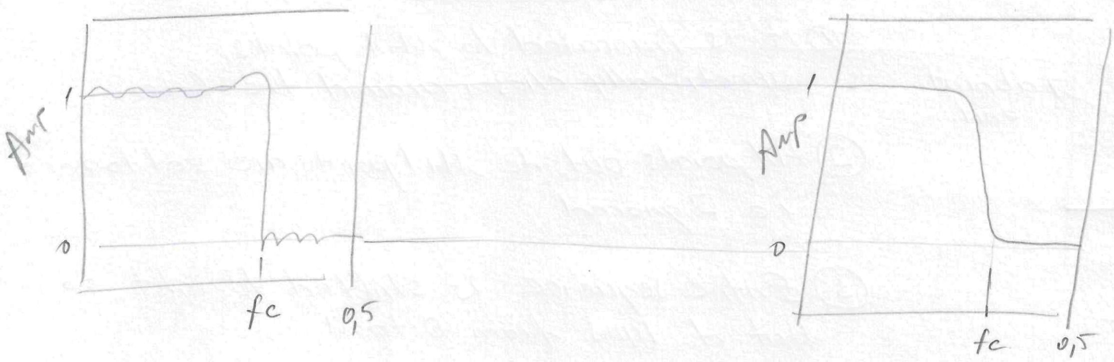
Multiple Truncated-sinc with blackman Window result is windowed-sinc filter.



Truncated-sinc filter kernel.

Blackman or Hamming Window

Windowed-sinc filter kernel.



Hamming and Blackman Window

- most known filter Window

⇒ Hamming Window

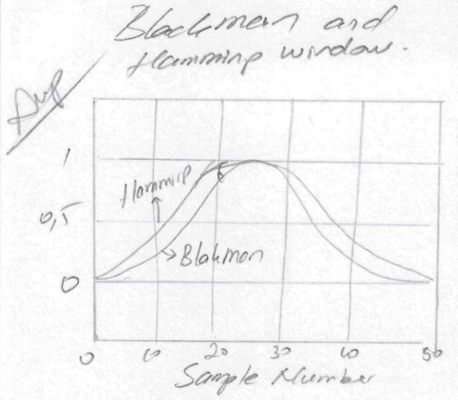
$$W[i] = 0{,}54 - 0{,}46 \cdot \cos\left(\frac{2\pi i}{M}\right)$$

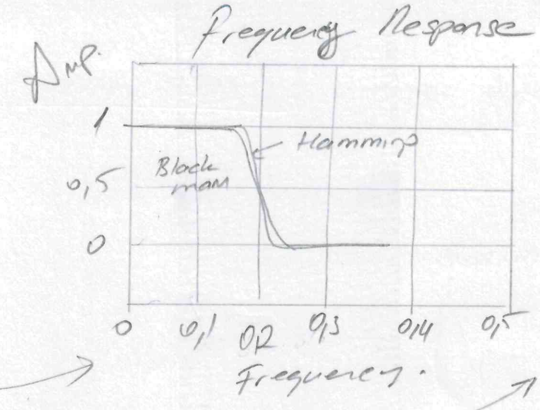
$i = 0$ to M

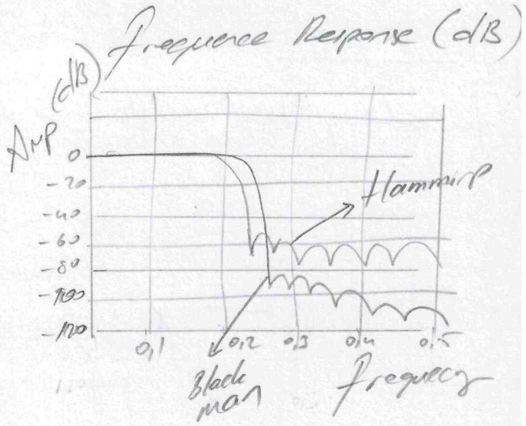
M = lenght of filter kernel

⇒ Blackman Window.

$$W[i] = 0{,}42 - 0{,}5 \cdot \cos\left(\frac{2\pi i}{M}\right) + 0{,}08 \cdot \cos\left(\frac{4\pi i}{M}\right)$$







Hamming Window

$$W[i] = 0{,}54 - 0{,}46 \cdot \cos\left(\frac{2\pi i}{M}\right)$$

Blackman is winner

Blackman Window

$$W[i] = 0{,}42 - 0{,}5 \cdot \cos\left(\frac{2\pi i}{M}\right) + 0{,}08 \cdot \cos\left(\frac{4\pi i}{M}\right)$$

Note

$i \neq 0$ durumlarda geçerli

$i == 0$ ise $(2\pi fc)$

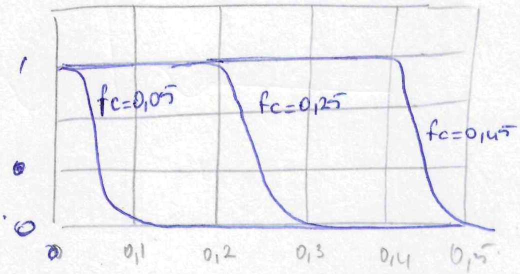
Designing the Windowed-Sinc filter

Required parameters

Cutoff frequency: fc | Between 0 and 0,5.

Filter Kernel length = M. | $M \cong \frac{4}{BW}$

BW: Width of transition Band.



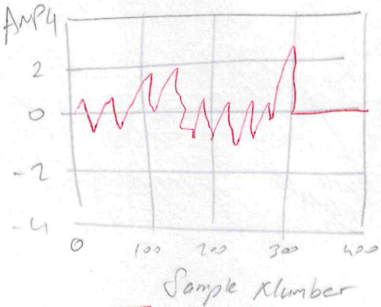
roll-off vs. Cutoff freq.

After selection required param

$$h[i] = K\left(\frac{\sin(2\pi fc\,(i - M/2))}{i - M/2}\right) \cdot \left[0{,}42 - 0{,}5 \cdot \cos\left(\frac{2\pi i}{M}\right) + 0{,}08 \cdot \cos\left(\frac{4\pi i}{M}\right)\right]$$

Overlap - Add

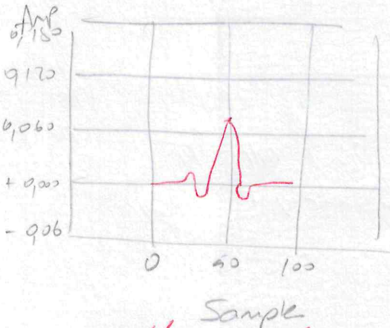
for breaking long signals into smaller segments for easier processing.



Input Signal

N: 300 samples

*



Filter Kernel

M: 101 samples.

= ?

length of Output

$N + M - 1$

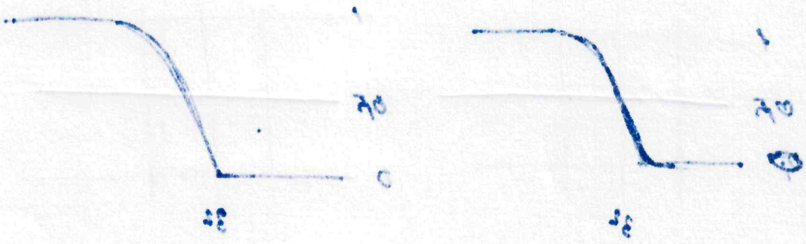
400 samples

Slow Step Response

Fast Step Response

No OverShoot

Over Shoot



Linear Phase

Non-Linear Phase