DIGITAL ~~SYSTEMS~~ SIGNALS ANALYSIS & APPLICATIONS

Textbooks –

- The Scientist and Engineer's Guide to Digital Signal Processing by Steven W Smith
- Foundations of Signal Processing – Vetterli
- DSP by Proakis and Manolakis.

Grading Policy –

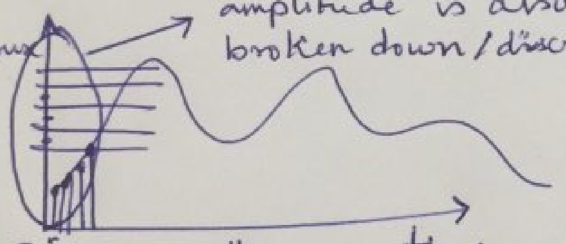
* 2 Mids (30%) + End Sems (30%)
* Assignments (40%) (3 Day Extension)

SIGNAL

Analog signal

pressure

amplitude is also broken down / discretized ? → To store values like $\underbrace{42\times10^{-108}}$ there must be a limit to the values being stored



time

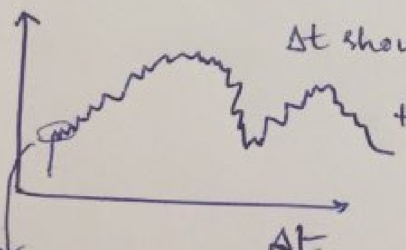
due to infinite values.

$\boxed{\text{sample the signal}}$ ≡ breakdown the time space.

↓

digital.

Can be retrieved back if * F signal should be sampled with sign $\geq 2F$

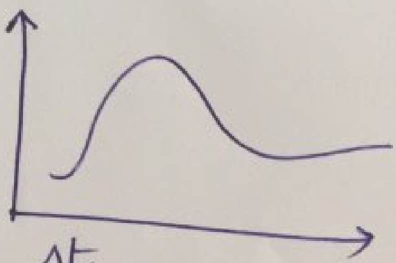


higher frequencies of the signal

Δt should be very small in this case

Δt_1 < Δt_2

Should be small as there are many variations.



SIGNALS AND SYSTEMS

Dennis Freeman [SNS]

SIGNALS →

signal in → SYSTEM → signal out

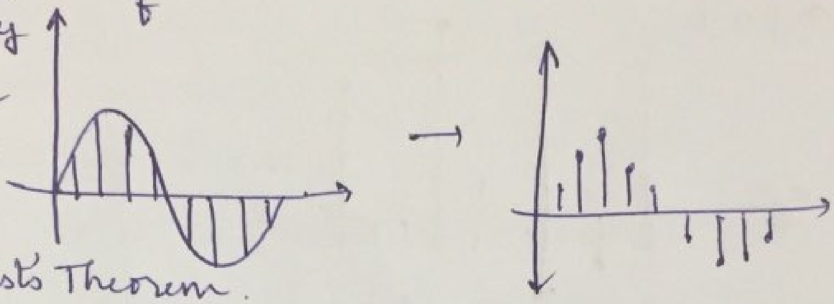
Aliasing
↳ a duplicate one

Sampling – $x(t) \equiv$ continuous signal

$x(n) = x(n\,\Delta T)$ for every ΔT (switch operates)

For a signal of frequency F, sample it with $\geq 2F$.

— Nyquist's Theorem.



Digital Images – 2D Matrix of intensities or numbers.

Sampling and **Quantization** – Pixels ≡ Sampling

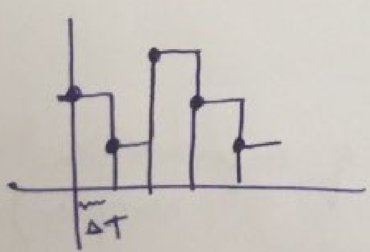
$\underbrace{\text{Sampling}}_{\uparrow \text{ clarity}}$ and $\underbrace{\text{Quantization}}_{\text{quality}}$ ↑ with ↑ bits per pixel

Colour image → 3 planes $(R + G + B) \equiv$ 24 bits per pixel

8 bits per pixel, 8, 8

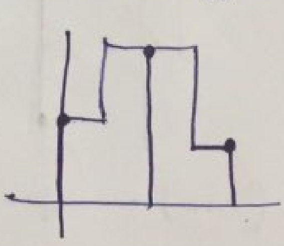
Digital to Analog

→ Zero Order hold:



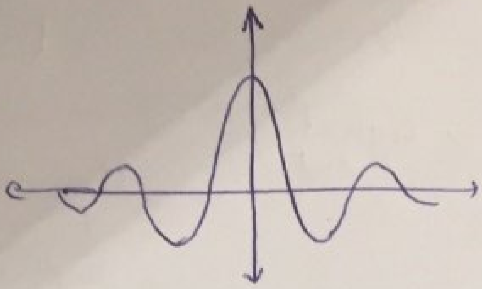
→ Nearest Neighbor: (offline processing)

cannot be used for real time systems [as the data comes]



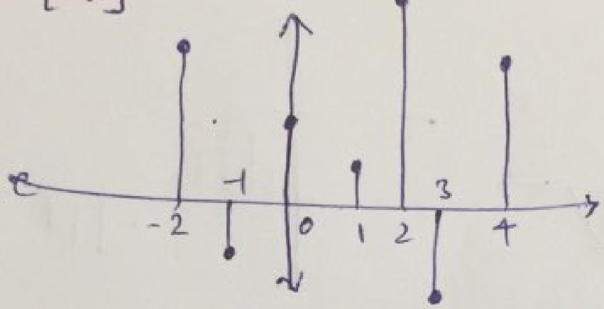
→ Linear Interpolation:

④ Sync function

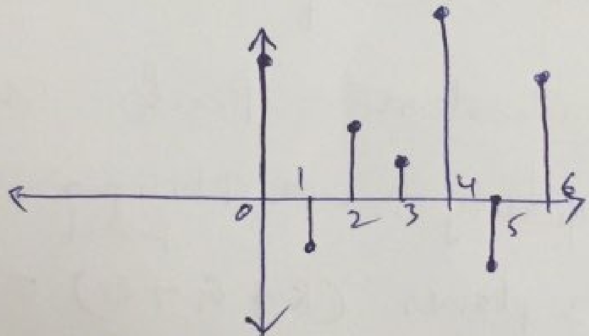


OPERATIONS ON SIGNALS –

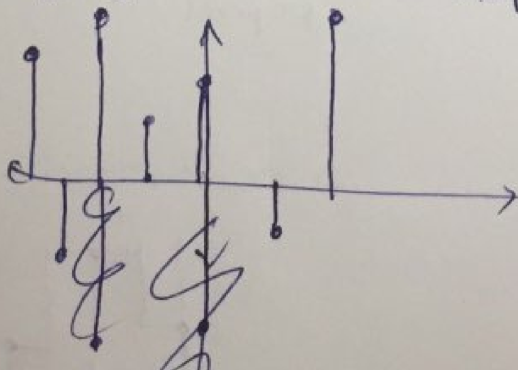
x[n]



① Shifting : x[n−2] ≡ delay of 2

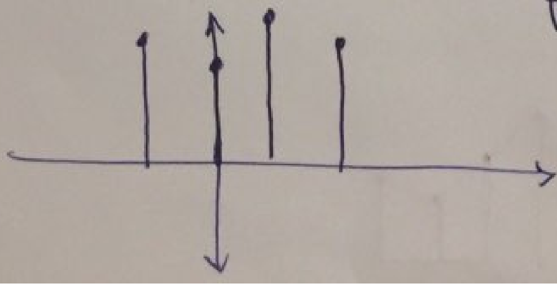


② Flipping : x[−n] ← Reflection



③ Scaling : x[2n] → Loss of information (samples)

cannot recover back



→ How to prevent loss of information → Sample → (10F) & then scaling helps.

ORDER: Shifting + Flipping + Scaling

→ $x[-2n+2]$

$y(n) = x[n+2]$ "Shift" by 2

!Scaling at start doesn't work!

$z(n) = y(-n) = x(-n+2)$ "Flip"

$w[n] = z(2n) = x(-2n+2)$ "scale".

Characteristics of Signals:—

① Odd or Even Signals.—

$$x(n) = x(-n) \quad [\because \text{Even signal}]$$

$$x(n) = -x(-n) \quad [\because \text{Odd Signal}]$$

$$x(n) = \frac{x(n) + x(-n)}{2} + \frac{x(n) - (x(-n))}{2}$$

② Periodic Signal —

$$x(n) = x(n+N)$$

③ Energy or Power signal —

$$E = \sum_{k=-\infty}^{\infty} |x(k)|^2$$

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} |x(k)|^2$$

Unit Signal —

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$E(u(n)) = \infty$

$P(u(n)) = 1/2$

Special Signals →

Continuous "Dirac"?

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases}$$

Any signal $x[n]$ in terms of $\delta(n)$

$$x[n] = \sum_{K=-\infty}^{\infty} x[K] \cdot \delta[n-K]$$

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

$$r(n) = \begin{cases} n, & n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

$$e[n] = \{ |a|^n u(n)$$

$$u(n) = \sum_{K=0}^{\infty} \delta(n-K)$$ ↷ Shift and add.

SYSTEMS ←

$$x(n) \longrightarrow \boxed{H} \longrightarrow y(n)$$

for Ex:

→ $y(n) = \frac{1}{2p+1} \sum_{K=-p}^{p} x(n-K)$ [Averaging system] Filter

moving window

Properties of Systems —

→ Causal System (real time systems) —

$$y(n) = x(n) + 2x(n-1)$$

Non-Causal System —

$$y(n) = x(n) + x(n-1) + 3x(n+2) \Rightarrow$$ "History & future values are being used"

→ Static or Dynamic ≡

$$y(n) = (x(n))^3 + 3(x(n)) \longrightarrow$$ [only at that point of time "n"]

$$y(n) = x(n) + 2x(n-1) \longrightarrow$$ which involves other time values.

→ Linear Systems ≡

$$\left.\begin{array}{l} x_1(n) \xrightarrow{H} y_1(n) \\ x_2(n) \xrightarrow{H} y_2(n) \end{array}\right\} \quad x_1(n) + x_2(n) \xrightarrow{H} y_1(n) + y_2(n).$$

— Additivity

$$a x_1(n) \xrightarrow{H} a\, y_1(n)$$

— Homogeniety

$$y(n) = x(n) + 2x(n-1) \longrightarrow \quad x(n) \rightarrow \boxed{H} \longrightarrow x(n) + 2(x(n-1))$$

$$z(n) = x_1(n) + x_2(n)$$

$$y(n) = z(n) + 2z(n-1)$$

— it is linear

$$z_1(n) = a(x(n))$$

$$y(n) = [z_1(n) + 2z_1(n-1)]$$

$$= a[x(n) + 2x(n-1)]$$

$$= a\, y(n)$$

$$y(n) = x(n) + 5.$$

$$= z(n) + 5$$

$$= x_1(n) + x_2(n) + 5$$ — Not Linear

Time Invariance –

$$x(n) \xrightarrow{H} y(n)$$
$$x(n-n_0) \xrightarrow{H} y(n-n_0)$$

doesnt depend on the time

Ex: $y(n) = x(n) + x(n-1)$

$z(n) = x(n-n_0) \equiv z(n) + z(n-1) \equiv x(n-n_0) + x(n-1-n_0)$

$= \underbrace{x(n-n_0) + x(n-n_0-1)}_{y(n-n_0)}$ — Time Invariant

→ $y(n) = x(n^2)$

$z(n) = x(n-n_0)$ $\qquad$ $y(n-n_0) = x((n-n_0)^2)$

$y(n) = z(n^2)$ $\qquad$ — Not Time Invariant

$= x(n^2 - n_0) \longrightarrow$ donot tally

Linear Time Invariant Systems: [LTI system]

$a x_1(n) + b x_2(n-1) + c.x_3(n-2) .. \longrightarrow \boxed{H} \longrightarrow a y_1(n) + b y_2(n-1) + c y_3(n-2) ..$

systems like – Telephony, Skype.

⇒ Impulse Response –

$$\delta(n) \xrightarrow{H} h(n)$$

→ Impulse response of a system completly characterizes the system.

$$x(n) = \sum_{K=-\infty}^{\infty} x(k) \cdot \delta(n-k).$$

$$T(x(n)) = T\left(\sum_{K=-\infty}^{\infty} x(k) \cdot \delta(n-k)\right)$$ = Additivity ↓

$$= \sum_{K=-\infty}^{\infty} T(x(K) \cdot \delta(n-K)).$$

= Homogeniety ↓

$$= \sum_{K=-\infty}^{\infty} x(K) \cdot T(\delta(n-K))$$

= Time Invariance

$$= \sum_{K=-\infty}^{\infty} x(k) \cdot h(n-K).$$

Impulse Response.

$$T(x(n)) = \sum_{K=-\infty}^{\infty} x(k) \cdot h(n-k).$$

we just need to know "h" if it is LTI system.

$$y(n) = \underbrace{\sum_{K=-\infty}^{\infty} x(k) \cdot h(n-k)}_{\text{convolution}}$$

Impulse Response.

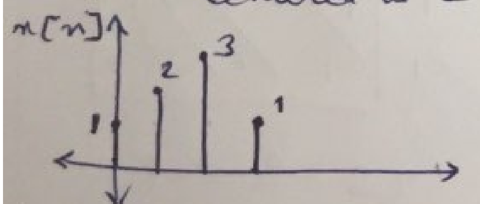
CONVOLUTION –

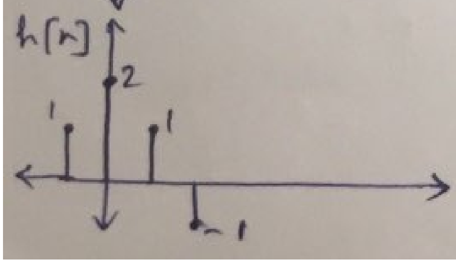
$$y(n) = \sum_{K=-\infty}^{\infty} x(K) \cdot h(n-k)$$

Impulse Response.

$x[n] = [1\ 2\ 3\ 1]$ — Centered at 1

$h[n] = [1\ 2\ 1\ -1]$ — Centered at 2



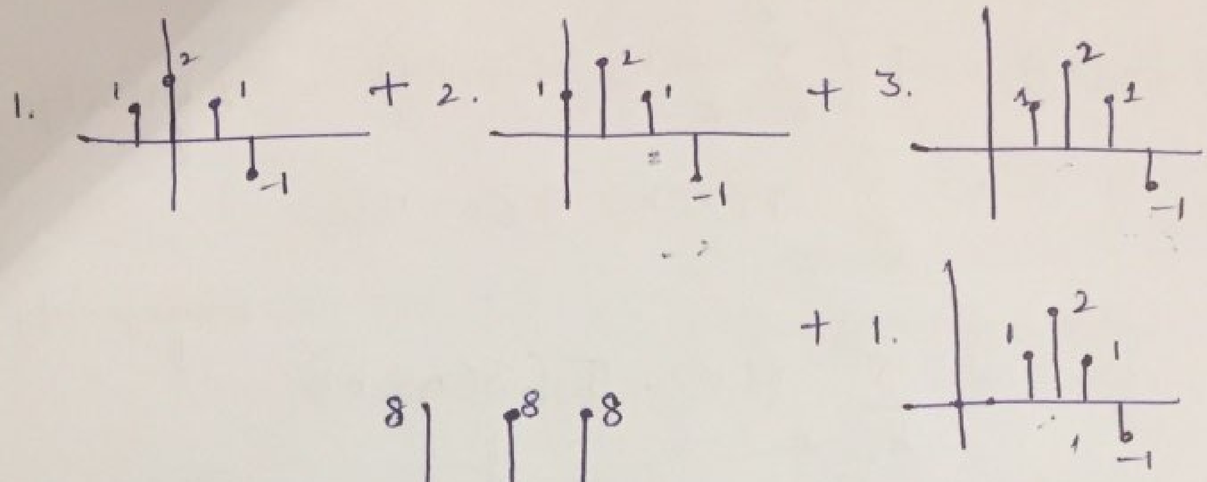


x can take only 0 to 4 values; so
$K = (0-4)$

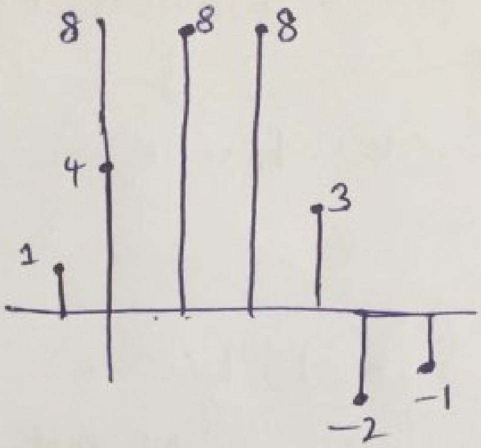
$$y(n) = x(K) * h(n) = x(0)h(n) + x(1)h(n-1) + x(2)h(n-2) + x(3)h(n-3)$$

Scanned by CamScanner

$= 1 \cdot h(n) + 2\, h(n-1) + 3 \cdot h(n-2) + 1 \cdot h(n-3)$



Resultant → ≡



---

other Method – Algorithm

0 0 0 0 1 2 3 1

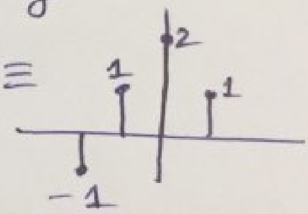
−1 1 2 1

slide it over

arrows meet (zeroth value) of the resultant

Flipping

$h(-n) \equiv$



slide 1   −1 1 2 $\binom{1}{1}$ 2 3 1   = 1

slide 2   −1 1 $\binom{1}{2}$ $\binom{2}{1}$ 3 1   = 4

slide 3   −1 $\binom{1}{1}$ $\binom{2}{2}$ $\binom{3}{1}$ 1   = 8

slide 4 $\binom{1}{-1}$ $\binom{2}{1}$ $\binom{3}{2}$ $\binom{1}{1}$ = 8

slide 5   1 $\binom{2}{-1}$ $\binom{3}{1}$ $\binom{1}{2}$ $\binom{0}{1}$ = 3

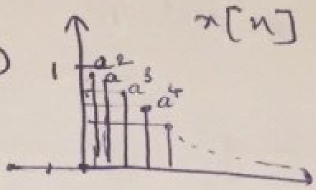
0 0 0 1 4 8 8 3 −2 −1 0 0

zero padding

	1	2	3	1
1	1	2	3	1
2	2	4	6	2
1	1	2	3	1
−1	−1	−2	−3	−1

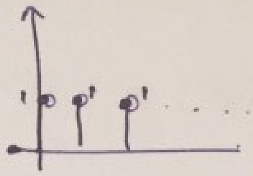
1 4 8 8 3 −2 −1

$x[n] = a^n u[n]$ $\quad a \in (0,1)$ $\quad x[n]$

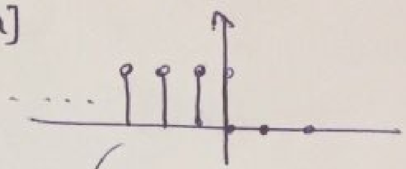


$h[n] = u[n]$

$h[n]$



$h[-n]$



$y[0] = 1$

$y[1] = 1 + a$

$y[2] = 1 + a + a^2$

$\vdots$

$y[n] = 1 + a + a^2 \dots a^n$

$$y[n] = \frac{(1 - a^{n+1})}{1-a} u[n]$$

if this is a finite series, then it decreases later

→ Properties of Convolution:

① $y[n] = x[n] * h[n] = h[n] * x[n]$

let $m = n - K$
$K = n - m$

$$= \sum_{m=-\infty}^{\infty} x(n-m) h(m) = \sum_{m=-\infty}^{\infty} h(m) x(n-m)$$

— Commutative Property

② $x(n)$ —— $\boxed{\diagdown}$ $\quad x(n) * h_1(n) * h_2(n) =$

$x(n) * h_2(n) * h_1(n)$

— Associative Property

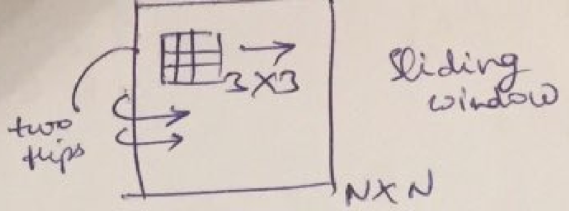
③ $x(n) * (h_1(n) + h_2(n)) = x(n) * h_1(n) +$
$x(n) * h_2(n)$.

Correlation — No flipping, only sliding

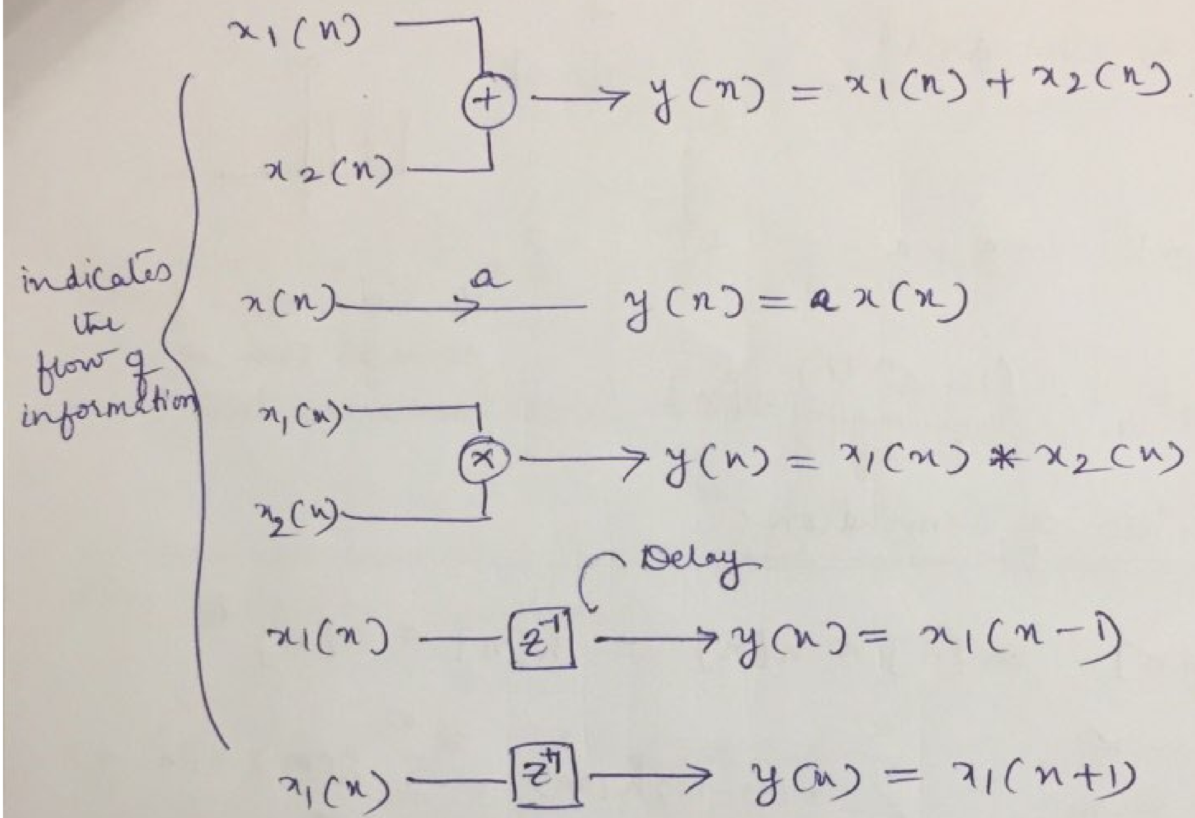
$\sum x(k) h(n+k)$

2D Convolution –

padding – efficient



$y[n] = x[n] + x[n-1]$ ≡ Difference Equations



2D Convolution → & Statistical Signal Processing

↓

Convoluted Signal = depends on the size of the signal and the size of the filter.

$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

averaging filter = blurs the image when there's a transition

↓

tells you whether it is blurred or not

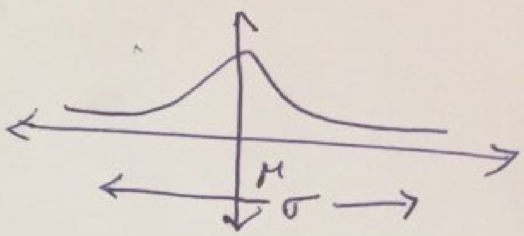
35x35 I → averaging filter ~ not the correct filter as you are giving equal priority to the 35^{th} value and the centroid one, so its better using Gaussian

$$y(n) = \frac{1}{N} \sum_{m=0}^{N-1} x(n-m)$$

Ex: 1D case

$\frac{1}{6}$ | 1 | 1 | 1 | 1 | 1 | 1 |

→ center, as it stores the history (values before it)



Gaussian Formula:- $\frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$; Central Limits Theorem

in case of 1D - center

→ | e^{-1} | $e^{-1/2}$ | $\boxed{1}$ | $e^{-1/2}$ | e^{-1} |

$\mu=0$

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$\mu = 0$

$\sqrt{2\pi}\sigma = 1/A = 1$

$A \cdot e^{-x^2/2}$

2D version of gaussian formula →

$$\frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\left[\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right]}$$

$$= A \cdot e^{-\left[\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right]}$$

different deviations in x and y axes.

$$= Ae^{-\left[\frac{x^2}{2} + \frac{y^2}{2}\right]}$$

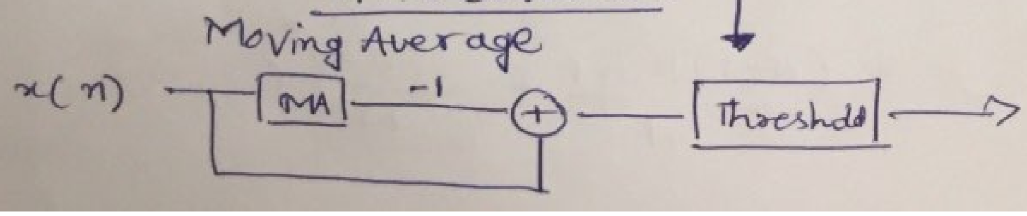
e^{-1}	$e^{-1/2}$	e^{-1}
$e^{-1/2}$	$\boxed{1}$	$e^{-1/2}$
e^{-1}	$e^{-1/2}$	e^{-1}

→ (0,0)

→ if σ is increasing, blurring increases as it takes distant points contribution

To avoid reducing in size - generally padding is done.

<u>spike detection</u>



If window size ↑, you get smooth curves.
window size ↓, noise gets added.

Recursive Moving Average →

$$y(n) = \frac{1}{N}\sum_{m=0}^{N-1} x(n-m)$$

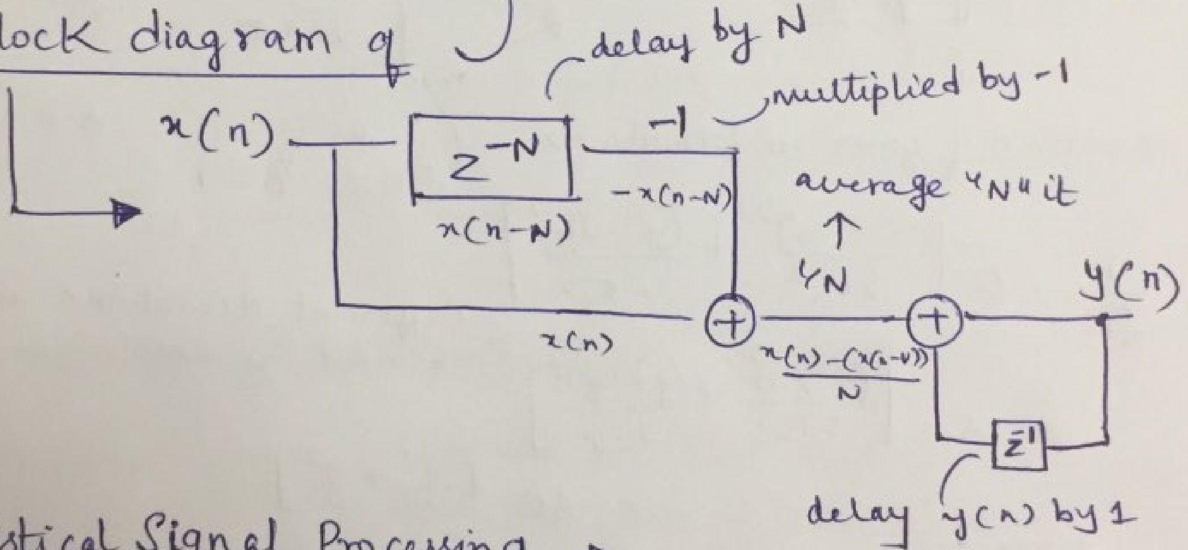
$$= \frac{x(n)}{N} + \underbrace{\frac{1}{N}\sum_{m=1}^{N-1} x(n-m) + \frac{1}{N}x(n-N)}$$

$$\Downarrow \qquad -\frac{1}{N}x(n-N)$$

$$y(n-1)$$

$$= y(n-1) + \frac{1}{N}\left(x(n) - x(n-N)\right)$$

Block diagram of ↻



Statistical Signal Processing ⇒

$$\text{Mean, } \mu = \frac{1}{N}\sum_{i=0}^{N-1} x(i)$$

Recursive version of mean

$$?^{*} \quad \mu = \frac{(N-1)\mu_{N-1}}{N} + \frac{x(N-1)}{N} \; ?$$

$$\sigma = \frac{1}{N-1} \sum_{i=0}^{N-1} (x(i) - \mu)^2$$

$$= \frac{1}{N-1} \sum_{i=0}^{N-1} [x(i)]^2 + \sum_{i=0}^{N-1} \mu^2 - 2\mu \sum_{i=0}^{N-1} x(i)$$

$$= \frac{1}{N-1} \left[\sum_{i=0}^{N-1} (x(i))^2 + N\mu^2 - 2N\mu^2 \right]$$

$$= \frac{1}{N-1} \sum_{i=0}^{N-1} (x(i))^2 - \frac{N\mu^2}{N-1}$$

$$= \frac{N}{N-1} \left[\frac{\sum_{i=0}^{N-1} (x(i))^2}{N} - \mu^2 \right]$$

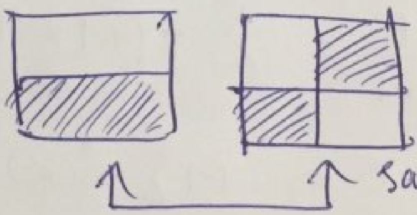
⇒ If there's any disturbance, you just need to update 2 values – sum of squares & the new computed mean instead of recomputing the data.

underexposed & overexposed images (more light) } – utilize histograms of images

→ probabilistic analysis on a histogram can be done

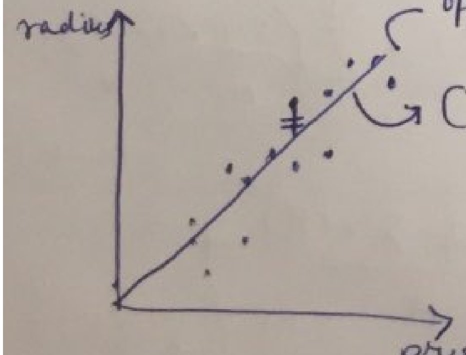
do not preserve the spatial information of the image.

For Ex:



same histograms.

Linear Regression:



$\{x(K), y(K)\}$ → Data given

$$\hat{y}(K) = M\,x(K) + C$$

$$e(K) = \hat{y}(K) - y(K)$$

$$E = \sum_{\forall K} (e(K))^2$$

$$E = \sum_{\forall K} \left(\hat{y}(K) - y(K) \right)^2$$

$$= \sum_{\forall K} \left(M x(K) + c - y(K) \right)^2$$

For Minimum error, E ;

$$\frac{\partial E}{\partial c} = 0.$$

$$\frac{\partial E}{\partial c} = \sum 2 \left(M x(K) + c - y(K) \right) = 0.$$

$$= M \sum x(K) + Nc - \sum y(K) = 0.$$

$$c = \frac{-M \sum x(K) + \sum y(K)}{N}$$

$$= -\frac{M}{N} \sum x(K) + \frac{\sum y(K)}{N}$$

$$c = \frac{\sum y(K)}{N} - \frac{M}{N} \sum x(K)$$

$$\frac{\partial E}{\partial M} = 0.$$

$$\frac{\partial E}{\partial M} = \sum 2 \left(M x(K) + c - y(K) \right) . x(K) = 0$$

$$0 = M \sum x^2(K) + c \sum x(K) - \sum x(K) . y(K)$$

$$M \sum x^2(K) = \sum x(K) . y(K) - c \sum x(K)$$

$$M \sum x^2(K) = \sum x(K) . y(K) - \sum \left(\frac{\sum y(K)}{N} - \frac{M}{N} \sum x(K) \right) x(K)$$

$$M \sum x^2(K) = \sum x(K) . y(K) - \frac{\sum y(K) . \sum x(K)}{N} + \frac{M}{N} \left(\sum x(K) \right)^2$$

$$M.\sum x^2(K) = \sum x(K).y(K) - \frac{\sum y(K).\sum x(K)}{N} + \frac{M}{N}\left(\sum x(K)\right)^2$$

$$M.\sum x^2(K) - \frac{M}{N}\left(\sum x(K)\right)^2 = \sum x(K).y(K) - \frac{\sum y(K).\sum x(K)}{N}$$

$$M\left[\frac{N\sum x^2(K) - \left(\sum x(K)\right)^2}{N}\right] = \frac{N\sum x(K).y(K) - \sum y(K).\sum x(K)}{N}$$

$$M = \frac{N\sum x(K).y(K) - \sum x(K).\sum y(K)}{N\sum x^2(K) - \left(\sum x(K)\right)^2}$$

→ Does it pass through (μ_x, μ_y)? (Mean of x, Mean of y)

$\mu_x = \frac{\sum x(K)}{N}$

$\mu_y = \frac{\sum y(K)}{N}$

yes; $c = \frac{\sum y(K)}{N} - \frac{M.\sum x(K)}{N}$

$c = \mu_y - M\mu_x$ ← passes.

→ What is $\sum_{\forall K} (\hat{y}(K) - y(K))$?

$$\sum_{\forall K}\left[\underbrace{M.x(K) + c - y(K)}_{0}\right] = 0$$ — from ©

Polynomial Regression —

larger curve — degree causes overfitting

instead of a curve we can do "local" regression (segments of lines are fit)

2-d curve —

$\hat{y}(K) = A\,x(K)^2 + B\,x(K) + C$

diff wrt A, B, C = 0 and solve for A, B, C

Sinusoids and Complex Exponentials →

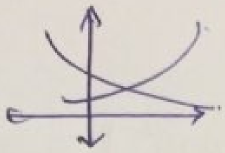
Quaternions

SINUSOIDS → "form": $A\sin(\omega_0 t + \phi)$

shifted and scaled
(ϕ) (ω_0)

EXPONENTIALS →

$ce^{at} \equiv$



SINUSOIDS AND EXPONENTIALS

$$e^{j\theta} = \cos\theta + j\sin\theta$$

$$\cos(\omega_0 t + \phi) = \text{Real}\{e^{j(\omega_0 t + \phi)}\} = Re\{e^{j\omega_0 t} \cdot e^{j\phi}\}$$

PERIODICITY → In discrete sense of time → integer

$$x(n) = e^{j\omega_0(n+N)} = e^{j\omega_0 n}$$

$$e^{j\omega_0 N} = 1 \Rightarrow \omega_0 N = 2\pi K$$

$$N = \frac{2\pi K}{\omega_0}$$

N should be an integer.

Ex: $\cos(7n) \Rightarrow N = \frac{2\pi K}{7}$ ✗
Not Periodic

Visualization concepts:

$$x(t) = c \cdot e^{at} = r \cdot e^{j\theta} \cdot e^{(p+jq)t}$$

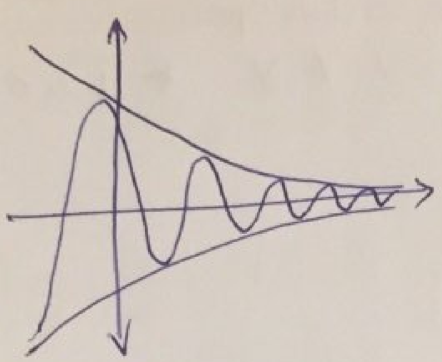
$$c = re^{j\theta} \qquad = (r \cdot e^{pt})(e^{j(\theta + qt)})$$

$$a = p + jq$$

$(r \cdot e^{pt})$ determines the amplitude (envelope)

$\underbrace{e^{j(\theta+qt)}}_{\text{sinusoid}}$

Complex exponential Generating Machine →

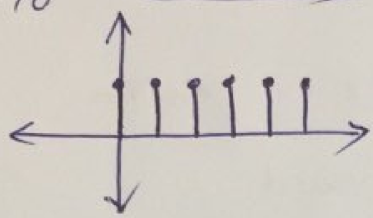


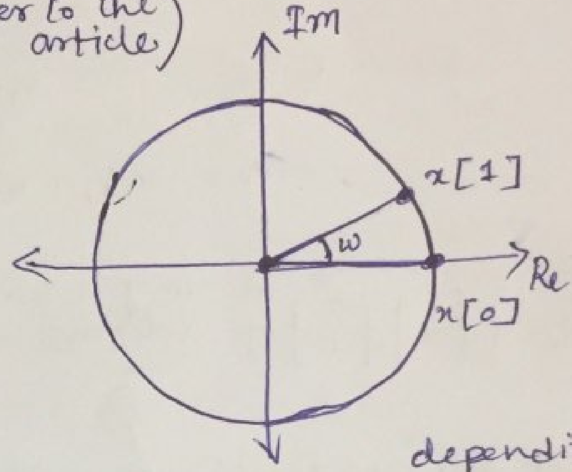
$$x(n) = e^{j\omega n}$$

$$x(n+1) = x(n) \cdot e^{j\omega}$$

* $\boxed{\text{wagon wheel Effect}}$ → (Refer to the article)

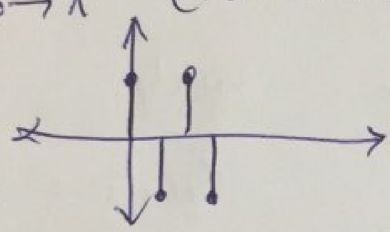
$\omega \to 0$





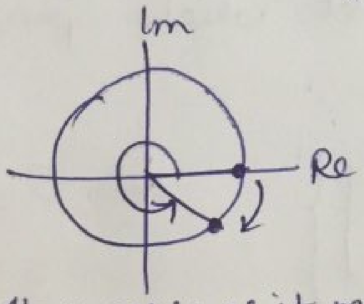
depending on ω] it may be periodic or not

$\omega \to \pi$ (maximum case)



Sample every $\omega = \pi/2$ (it appears "not moving") → "Aliasing"

$\omega > \pi$



"It appears as it moved backwards"

Vector Space →

$$V = \{ v_1, v_2, \ldots \; v_N \}, \; v \in \mathbb{R}^N$$

V is a vector space, iff

$$v_i + v_k \in V \quad , \forall i, k \in N$$

$$a v_i \in V \quad , a \in \mathbb{R}$$

$\mathbb{R}^N$, $\mathbb{C}^N$ → vector spaces

$V = \{(x, y) \; ; \; x \geq 0, y \geq 0\}$ — Not a vector space

(ii) $aV_i \notin V$ if $a \in \mathbb{R}^-$

$V = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\}$ — it is a vector space

$= \phi$ (Null set)

<u>Spanning Set</u> →

$$S = \{v_1, v_2, \dots v_n\} \; ; \; v \in V$$

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 \dots \alpha_n v_n$$

$$\mathbb{R}^2 \quad S = \left\{ \overset{v_1}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} \overset{v_2}{\begin{bmatrix} 0 \\ 1 \end{bmatrix}} \right\}$$

any v can be expressed interms of v_1, v_2

<u>Linear Independence</u> → $V = \{v_0, v_1, \dots v_{N-1}\}$

→ $\alpha_0 v_0 + \alpha_1 v_1 + \dots \alpha_{N-1} v_{N-1} = 0$

only when $\alpha_0 = \alpha_1 = \dots \alpha_{N-1} = 0$

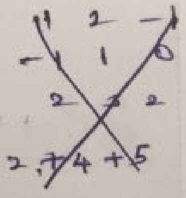
<u>Basis</u>: Linearly independent set which spans V is called basis of V.

<u>ex:</u>

$$S = \left\{ \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} \right\}$$

does S span $\mathbb{R}^3$?

$$\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



$$\alpha_1 + 2\alpha_2 - \alpha_3 = a$$
$$-\alpha_1 + \alpha_2 = b$$
$$2\alpha_1 + 3\alpha_2 + 2\alpha_3 = c$$

$\Delta \neq 0$

Hence, there exists a solution for $\alpha_1, \alpha_2, \alpha_3$

$\{(1,0,0) \; (1,1,0) \; (0,0,1)\}$ $\mathbb{C}^3$

Linearly Independent $\equiv$ Basis of $\mathbb{C}^3$

Inner Product →

$$\langle u, v \rangle = \sum_{i=0}^{N-1} u(i)\,\overline{v[i]}$$

Orthogonality → if innerproduct of 2 vectors = 0, then they are orthogonal

$\langle u, v \rangle = 0$. , $[u, v = \text{orthogonal}]$

Norm → $\langle u, u \rangle$

, if $\langle u, u \rangle = 1$, & $\langle v, v \rangle = 1$ } $\langle u, v \rangle = 0$

↓

u, v — orthonormal vectors

If a set $S \subset V$ of non zero vectors is orthogonal, then S is linearly independent ; $S = \{v_1, v_2 \ldots v_n\}$

$$\langle \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 \ldots + \ldots \alpha_n v_n \rangle = \langle 0, v_1 \rangle$$

$$\alpha_1 \langle v_1, v_1 \rangle + \alpha_2 \langle v_2, v_1 \rangle + \alpha_3 \langle v_3, v_1 \rangle \ldots = 0.$$

$$\alpha_1 \langle v_1, v_1 \rangle = 0 \Rightarrow \alpha_1 = 0$$

Non zero vector ||ly $\alpha_1 = \alpha_2 \ldots \alpha_n = 0$

$\Rightarrow$ linearly Independent

Theorem of orthogonal decomposition —

$$V = \sum_{k=0}^{N-1} \alpha_k v_k \Rightarrow \alpha_k = \frac{\langle v, v_K \rangle}{\langle v_K, v_K \rangle}$$

$\{v_0, v_1, \ldots v_{N-1}\} \equiv$ orthogonal set

form a basis

$$v = \alpha_0 v_0 + \alpha_1 v_1 \ldots - \alpha_K v_K + \ldots \alpha_{N-1} v_{N-1}$$

$$\langle v, v_K \rangle = \langle \alpha_0 v_0 + \alpha_1 v_1 \ldots \alpha_K v_K + \ldots \alpha_{N-1} v_{N-1}, v_K \rangle$$

Scanned by CamScanner

$$\langle v, v_K \rangle = \alpha_K \langle v_K, v_K \rangle$$

$$\alpha_K = \frac{\langle v, v_K \rangle}{\langle v_K, v_K \rangle}$$

Ex:

$$v_1 = \begin{bmatrix} 1/3\sqrt{2} \\ 1/3\sqrt{2} \\ -4/3\sqrt{2} \end{bmatrix} \quad v_2 = \begin{bmatrix} 2/3 \\ 2/3 \\ 1/3 \end{bmatrix} \quad v_3 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{bmatrix} \quad \langle v_1 v_2 v_3 \rangle$$

$\Downarrow$

orthogonal

$$v = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\alpha_1 = \frac{\left\langle \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1/3\sqrt{2} \\ 1/3\sqrt{2} \\ -4/3\sqrt{2} \end{bmatrix} \right\rangle}{1} = -2/3\sqrt{2}, \quad \alpha_2 = 5/3, \quad \alpha_3 = 0$$

$$\boxed{v = -\frac{2}{3\sqrt{2}} V_1 + \frac{5}{3} V_2 + 0 \cdot V_3}$$

$$\langle v, v_K \rangle = \alpha_K \langle v_K, v_K \rangle$$

$$\alpha_K = \frac{\langle v, v_K \rangle}{\langle v_K, v_K \rangle}$$

Ex:

$$v_1 = \begin{bmatrix} 1/3\sqrt{2} \\ 1/3\sqrt{2} \\ -4/3\sqrt{2} \end{bmatrix} \quad v_2 = \begin{bmatrix} 2/3 \\ 2/3 \\ 1/3 \end{bmatrix} \quad v_3 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{bmatrix} \qquad \langle v_1 v_2 v_3 \rangle \Downarrow \text{orthogonal}$$

$$v = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\alpha_1 = \frac{\left\langle \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1/3\sqrt{2} \\ 1/3\sqrt{2} \\ -4/3\sqrt{2} \end{bmatrix} \right\rangle}{1} = -2/3\sqrt{2}, \quad \alpha_2 = 5/3, \quad \alpha_3 = 0$$

$$\boxed{v = -\frac{2}{3\sqrt{2}} v_1 + \frac{5}{3} v_2 + 0 \cdot v_3}$$

DISCRETE FOURIER TRANSFORM →

$$V_K[n] = e^{j\frac{2\pi K n}{N}}, \quad n \in [0, \dots N-1]$$

$$V_0 = [1 \ 1 \ 1 \ \dots \ 1]$$

$$V_1 = \left[1 \ e^{\frac{2\pi j}{N}} \ e^{\frac{4\pi j}{N}} \ \dots \ e^{\frac{2\pi(N-1)}{N} j}\right]$$

$$\langle v_k, v_l \rangle = \sum_{n=0}^{N-1} v_K(n) \cdot \bar{v}_l(n)$$

$$= \sum_{n=0}^{N-1} e^{j\frac{2\pi K n}{N}} \cdot e^{-j\frac{2\pi l n}{N}}$$

$$= \sum_{n=0}^{N-1} e^{j\frac{2\pi n}{N}(K-l)}$$

$$\Rightarrow \langle v_k, v_l \rangle = \sum_{n=0}^{N-1} e^{j\frac{2\pi n}{N}(k-l)}$$

(1) $K = l$;

$$\langle v_K, v_l \rangle = \sum_{u=0}^{N-1} 1 = \underline{\underline{N}}$$

(2) $K \neq l$;

$$= e^{j2\pi 0 \frac{(K-l)}{N}} + e^{j2\pi 1 \frac{(K-l)}{N}} + \cdots + e^{j2\pi \frac{(N-1)(K-l)}{N}}$$

$$= \frac{1 - \left(e^{j2\pi \frac{K-l}{N}}\right)^{N}}{1 - e^{j\frac{2\pi(K-l)}{N}}} = 0$$

$\Rightarrow$ (*) $x = \sum_{K=0}^{N-1} X_K v_K$ $\qquad X_K = \frac{\langle x, v_K \rangle}{\langle v_K, v_K \rangle}$

$$x = X_0 v_0 + X_1 v_1 + X_2 v_2 \cdots + X_{N-1} v_{N-1}$$

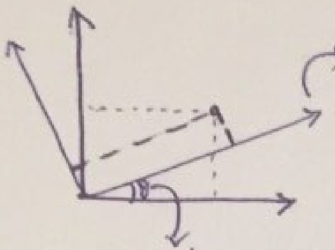
$$x[n] = \sum_{K=0}^{N-1} X_K \cdot e^{j\frac{2\pi Kn}{N}}$$

$$x_K = \frac{1}{N} \sum_{K=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi Kn}{N}}$$

$$\begin{bmatrix} x_0 \\ x_1 \\ \\ x_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} z^0 & z^0 & z^0 & z^0 & \cdots & \\ z^0 & z^1 & z^2 & & & z^{N-1} \\ \vdots & & & & & \\ z^0 & z^{N-1} & z^{2(N-1)} & \cdots & & z^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \\ x[N-1] \end{bmatrix}$$

$$e^{-j\frac{2\pi Kn}{N}} = z^{Kn} \Rightarrow z = e^{-j\frac{2\pi}{N}}$$

$$\Rightarrow \boxed{z = e^{-j\frac{2\pi}{N}}}.$$



transforms of a basis

≡ Euclidean space

forms a basis

$$X_K = \frac{1}{N}\sum_{n=0}^{N-1} x[n].\, e^{-j\frac{2\pi Kn}{N}} \qquad \left(\text{Discrete Fourier Transform}\right)$$

$$x(n) = \sum_{K=0}^{N-1} X_K \cdot e^{j\frac{2\pi Kn}{N}}$$

Ex: Let the signal be ≡ $(N=4)$

$$x(t) = 5 + 2\cos\left(2\pi t - \frac{\pi}{2}\right) + 3\cos \underbrace{4\pi t}_{2\pi(2)t}$$

$$f_{max} = 2 \text{ Hz}$$

Sampling frequency = 4 Hz , $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$

$$x[0] = 5 + 2\cos\left(\frac{-\pi}{2}\right) + 3\cos 0 = 8$$

$$x[1] = 5 + 2\cos\left(\frac{\pi}{2} - \frac{\pi}{2}\right) + 3\cos\pi = 4.$$

$$x[2] = 5 + 2\cos \pi/2 + 3\cos 2\pi t = 8$$

$$x[3] = 5 + 2\cos\pi + 3\cos 3\pi = 0.$$

$$x[n] = \{8, 4, 8, 0\}$$

$$\frac{1}{4}\begin{bmatrix} 20 \\ -4j \\ 12 \\ 4j \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}\begin{bmatrix} 8 \\ 4 \\ 8 \\ 0 \end{bmatrix} \begin{array}{l} k=0. \\ K=1 \\ K=2 \\ \end{array}$$

roots of unity $= z^{1/4} = 1.$

0 1Hz 2Hz → conjugate of 1 ⟹ $|j| + |-j| = 2$

Ans ≡ $[5 \; -j \; 3 \; j]$ $\boxed{X[K] = X^*[N-K]}$

Sinusoid of 1 Hz, amplitude ≡ 2

DC Value

sinusoid of 2Hz where amplitude is 3.

since it is sampled at 4HZ, Max freq } = 2Hz

Since it is complex; there may be a phase shift in the 1Hz signal. $[-\pi/2]$

First Value ≡ Mean of the signal. [No frequency term] DC Value.

magnitude of 1Hz frequency ≡ distributed between the two values $[-j, j]$

> length of the signal, {8,4,8,0} ≡ 4.

> 2Hz value ⟹ It is real; No phase shift, Amplitude = 3, Frequency = 2Hz. } By convention it is "cos"

frequency vs time (just like Heisenberg's Uncertainity Principle).

If we take more samples ≡ frequency shifts accordingly

CONTINUOUS FOURIER TRANSFORM [FT]

$$X(f) = \int_{-\infty}^{\infty} x(t).\, e^{-j2\pi f t}\, dt$$

$$X(\omega) = \int_{-\infty}^{\infty} x(t).\, e^{-j\omega t}\, d\omega.$$

① $\delta(t)$.

$$X(f) = \int_{-\infty}^{\infty} \delta(t).\, e^{-j2\pi f t}\, dt \quad ; \text{at } t=0;\ 1$$

$$X(f) = 1$$

② $\delta(t-t_0)$

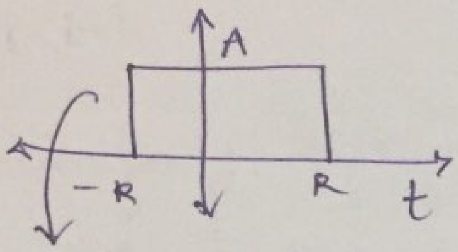
$$X(f) = \int_{-\infty}^{\infty} \delta(t-t_0).\, e^{-j2\pi f t}\, dt \quad , \text{at } t = t_0,\ \overset{\delta(t-t_0)}{\curvearrowright} 1$$

$$= e^{-j2\pi f t_0} = e^{-j\omega t_0}.$$

③ $\delta(t-t_0) + \delta(t+t_0)$ (2 symmetric impulse responses)

$\hookrightarrow e^{-j\omega t_0} + e^{j\omega t_0} = 2\cos\omega t_0$

④



box function

"can be reduced from $(-\infty, \infty)$ to $(-R, R)$".

$$\equiv \int_{-\infty}^{\infty} x(t).\, e^{-j\omega t}.\, dt$$

$$\equiv \int_{-R}^{R} A.\, e^{-j\omega t}\, dt$$

$$\equiv A \int_{-R}^{R} e^{-j\omega t}\, dt$$

$$\equiv \frac{A}{-j\omega} \left. e^{-j\omega t} \right]_{-R}^{R}$$

$$\equiv \frac{A}{-j\omega}\left[e^{-j\omega R} - e^{j\omega R}\right]$$

$$\equiv \frac{A}{j\omega}\left[e^{j\omega R} - e^{-j\omega R}\right]$$

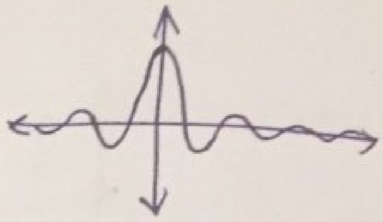
$$\equiv \frac{A}{j\omega}\left[2j\sin\omega R\right] \equiv \frac{2A}{\omega}\sin\omega R$$

$$\equiv 2AR.\left(\frac{\sin\omega R}{\omega R}\right)$$

$$\equiv 2AR \underbrace{\left(\frac{\sin \omega R}{\omega R}\right)}.$$

$$\mathrm{sinc}(a) = \frac{\sin a}{a}$$

$$\equiv 2AR \, \mathrm{sinc}(\omega R)$$



light - periodic waves.

FAST FOURIER TRANSFORM →

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \,.\, e^{-j \cdot \frac{2\pi K n}{N}}$$

dft → takes a lot of time to do the large matrix multiplication

vs

fft ↓ very fast

(1.) $K = \frac{N}{2} K_1 + K_0$; $K_1 = 0$ (or) 1

$0 \leq K_0 \leq \frac{N}{2} - 1$

$K \curvearrowright$

$K_1 = 0$ first half of the signal

$K_1 = 1$ next half of the signal

$K \curvearrowleft$

$$X_K = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \,.\, e^{-2\pi j \left(\frac{N}{2} K_1 + K_0\right)\frac{n}{N}}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} x[n] \,.\, e^{-2\pi j \frac{N K_1 n}{2N}} \,.\, e^{-2\pi j \frac{K_0 n}{N}}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} x[n] \,.\, e^{-\pi j K_1 n} \,.\, e^{-2\pi j \frac{K_0 n}{N}}$$

$$\therefore e^{-\pi j} = -1$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} x[n] \,.\, (-1)^{K_1 n} \,.\, e^{-2\pi j \frac{K_0 n}{N}}$$

↓

divide into even & odd indices

$$X_K = \frac{1}{N} \sum_{n=0}^{\frac{N}{2}-1} x[2n] \,.\, e^{-2\pi j \frac{K_0 (2n)}{N}} \,.\, \underbrace{(-1)^{K_1(2n)}}_{1} + \sum_{n=0}^{N/2-1} x[2n+1] \,.\, e^{-2\pi j \frac{K_0 (2n+1)}{N}} \,.\, \underbrace{(-1)^{K_1(2n+1)}}_{(-1)^{K_1}}$$

$$X_K = \frac{1}{N} \sum_{n=0}^{\frac{N}{2}-1} x[2n] \,.\, e^{-\frac{2\pi j K_0 n}{N/2}} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] \,.\, e^{-2\pi j \frac{K_0 (2n+1)}{N}} \, (-1)^{K_1}$$

$$X_K = \frac{1}{N}\sum_{u=0}^{\frac{N}{2}-1} x(2n)\cdot e^{-\frac{2\pi j K_0 n}{N/2}} + (-1)^{K_1} \cdot e^{-\frac{2\pi j K_0}{N}} \rightarrow Q \; \sum_{u=0}^{\frac{N}{2}-1} x(2n+1)\cdot e^{-\frac{2\pi j K_0 n}{N/2}}$$

computing just the fourier transforms of half signals.

$$X_K = F_0 + (-1)^{K_1} Q F_1$$

$$X_{K_1=0} = F_0 + QF_1$$

$$X_{K_1=1} = F_0 - QF_1$$

$f(x) =$ $x[0]$ $x[1]$ $x[2]$ . . . . . $x[N-2]$ $x[N-1]$

F_0 F_1 $2T_{M/2}$

$F(u) =$ $F_0 + MF_1$ $F_0 - MF_1$ $2M$

Complexity $\equiv$ $T(N) = 2\,T(N/2) + 2N$

$O[N \log N]$

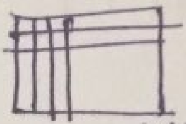
FFT Computation time

2D FOURIER TRANSFORM →

physical Intution?

$$X = W_N x^T \quad (1\text{D case})$$

A



MXN matrix

row wise ← fourier transform

& then the output is applied fourier transform column wise.

$$\hat{A} = W_{M\times M}\left(A_{M\times N}\, W_{N\times N}\right)$$

if it rotates, it changes
but on transleting, it doesnt change.

CONVOLUTION THEOREM →

$$y(n) = \sum_{k=0}^{N-1} x[k] \cdot h[n-k]$$

$$x[n] = [\underset{\uparrow}{1}\ 2\ 0\ 1] \qquad h[n] = [\underset{\uparrow}{2}\ 2\ 1\ 1]$$

$$\begin{array}{l} \quad\quad 1\ 2\ 0\ 1 \\ 1\ 1\ 2\ \underset{\uparrow}{2} \end{array} = 0\ 2\ 6\ 5\ 5\ 4\ 1\ 1\ 0 \quad \underset{\text{Convolution}}{\overset{*}{\uparrow}}$$

circuler Convolution : $y[n] = \sum_{K=0}^{N-1} x[k] \cdot h[\underbrace{(n-K) \bmod N}_{?\ \text{what is happening}}]$

$$y[0] = x[0] \cdot h[0] + x[1]\,h[3] + x[2]\,h[2] + x[3]\,h[1] = 6$$

$$y[1] = x[0] \cdot h[1] + x[1] \cdot h[0] + x[2]\,h[3] + x[3]\,h[2] = 7$$

1 2 0 1
2 1 1 2
2 2 1 1
1 2 2 1
1 1 2 2

$$\circledast = 6\ 7\ 6\ 5$$

Circular convolution

If $x_K = x_{K \bmod N}$ (a repeating signal)

2 2 1 1

Normal convolution

... 1 2 0 1 1 2 0 1 1 2 0 1 1 2 0 1 ...

1 1 2 2

5 $\boxed{6\ 7\ 6\ 5}$ 6 7 6 5 .

For Ex: Padding the signals with zeros and doing a circular convolution — to result in the size of "output"

	1	2	0	1	0	0	0
2 ⇐	2	0	0	0	0	1	2
6 ⇐	2	2	0	0	0	1	1
5 ⇐	1	2	2	0	0	0	1
	1	1	2	2	0	0	0

h[n]

[2 2 1 1 0 0 0]

0 6 5 4 3 2 1

circular convolution

results in → linear convolution output.

→ $y[n] = x[n] \circledast h[n]$ [∴ Circular Convolution] (Discrete Domain)

$y[n] = F^{-1}(F(x[n]) . F(h[n]))$ ⇒ DFT is used.

→ $y[n] = x[n] * h[n]$ [∴ Convolution] (Continuous Domain)

$y[n] = F^{-1}(F(x(n)) . F(h(n)))$

FT is used

CFT : $Y(n) = \int_{-\infty}^{\infty} y(n) . e^{-j\omega n} \, dn$

$= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(k) \, h(n-k) dk \right) e^{-j\omega n} \, dn$

$n - k = m$

$$= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(k).\, h(m)\, dk \right) e^{-j\omega(k+m)}\, dm$$

$n - k = m$

$dn = dm$

$$= \int_{-\infty}^{\infty} h(m).\, e^{-j\omega m}\, dm \,.\, \int_{-\infty}^{\infty} x(k).\, e^{-j\omega k}\, dk.$$

$$= FT(x).\, FT(h).$$

Discrete Domain:

(circular convolution)

$$y(n) = \sum_{r=0}^{N-1} x(r).\, h(n-r) \quad \text{(convolution formula)}.$$

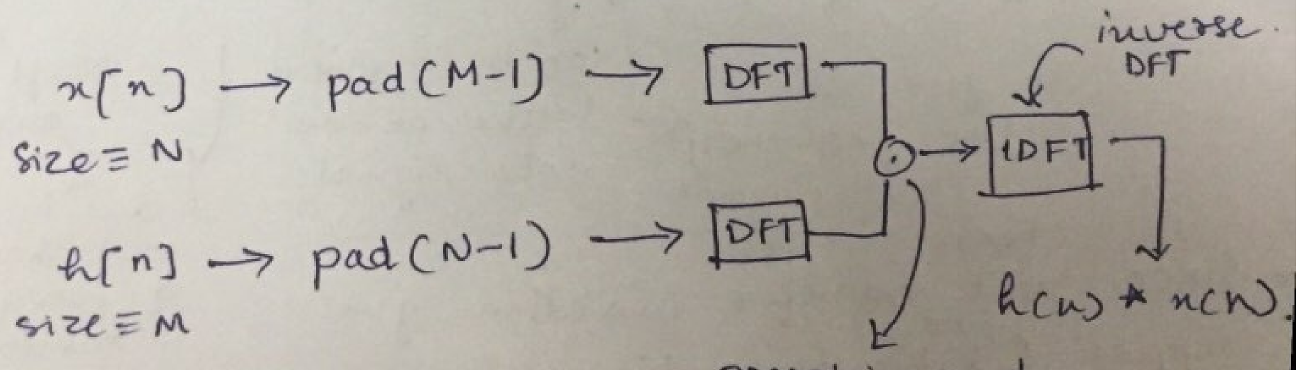
$$Y_K = \frac{1}{N} \sum_{n=0}^{N-1} y(n).\, e^{-j\frac{2\pi K n}{N}}$$

$n - r = m.$

$$= \sum_{n=0}^{N-1} \left(\sum_{r=0}^{N-1} x(r).\, h(n-r) \right).\, e^{-j\frac{2\pi K(n)}{N}}$$

$$= \sum_{n=0}^{N-1} \left(\sum_{r=0}^{N-1} x(r).\, h(m) \right) e^{-j\frac{2\pi K(m+r)}{N}}$$

$$= \sum_{n=0}^{N-1} h(m).\, e^{-j\frac{2\pi K m}{N}} \sum_{r=0}^{N-1} x(r).\, e^{-j\frac{2\pi K r}{N}}.$$

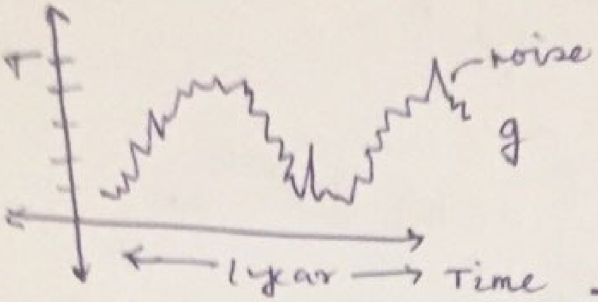


operation must have same size output signals.

[Circulant matrix]

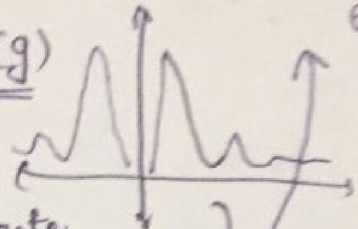
Transfer Function?

Real life Examples: — Turbidity of lake (data)



FFT

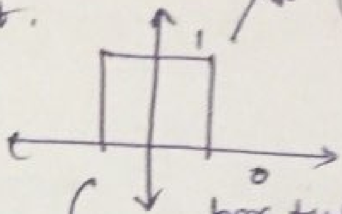
$\underline{F(g)}$



makes the high freq zero

→ Conjugate should be taken care of.

* (point to point multiplication)



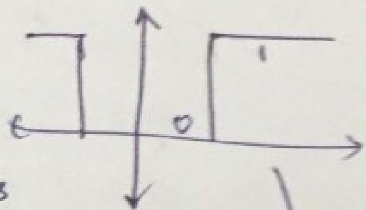
*

Fourier transform } of ~~those~~ is similar to convolution <u>with sinc</u> — low pass filtering

box function

transfer function.

- Band pass filtering → fixed band width size.



— high pass filtering (removing the lower frequencies).

inverse fourier → you'll get the ~~noise~~ ~~less~~ filtered signal.

<u>Moving Average Filter</u>

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$$

Ex: $\left[\frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\right]$

can be applied to the above data signal → to give the low frequency signal

— shift occurs due to the contribution of previous values

→ Since (box) rect transfer function gives ripples (just like convoluting with sinc function) ⇒ we can do multiply with an "exponentially decaying function"

⇓

<u>gaussian function</u>

pepper noise = max filter
salt noise = min filter
salt and pepper noise = median filter

Leaky Integrator -

M previous values

moving average = $y_M[n] = \frac{1}{M}\sum_{k=0}^{M-1} x(n-k)$.

$\rightarrow y_M[n-1] = \frac{1}{M}\sum_{k=0}^{M-1} x(n-1-k)$

$k+1 = k$

$* \; y_M[n-1] = \frac{1}{M}\sum_{k=1}^{M} x(n-k)$.

$\rightarrow y_{M-1}[n] = \frac{1}{M-1}\sum_{k=0}^{M-2} x(n-k)$

$\rightarrow y_{M-1}[n-1] = \frac{1}{M-1}\sum_{k=0}^{M-2} x(n-1-k)$.

$= \frac{1}{M-1}\sum_{k=1}^{M-1} x(n-k)$.

$* \; y_{M-1}[n-1] = \frac{1}{M-1}\sum_{k=1}^{M-1} x[n-k]$.

$$\sum_{k=0}^{M-1} x(n-k) = x(n) + \sum_{k=1}^{M-1} x(n-k)$$

from above;

$$M y_M[n] = x[n] + (M-1) y_{M-1}[n-1]$$

$$y_M[n] = \frac{x[n]}{M} + \frac{M-1}{M} y_{M-1}[n-1]$$

let $\frac{M-1}{M} = \lambda$ $\quad 1 - \frac{1}{M} = \lambda \Rightarrow \frac{1}{M} = 1 - \lambda$

$$y_M[n] = (1-\lambda)\, x[n] + \lambda\, y_{M-1}[n-1]$$

Suppose M is very Large, $\lambda \approx 1$ & that results in
(window is large)

$y_M[n-1] = y_{M-1}[n-1]$

mean of m values is equal to mean of m-1 values.

[just the position].

Leaky Integrator

$$y[n] = (1-\lambda)\, x[n] + \lambda\, y_M[n-1].$$

* current value * weight + Previous (Mean value) = mean.

Though you're integrating the values, it gives slight weightage to the current value (there's a leak).

used for Analog signals.

Ex: $y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x(n-k).$

Impulse Response $\longrightarrow$ $h[n] = \frac{1}{M} \sum_{k=0}^{M-1} \delta(n-k).$

for $n=0$; $k=0 \Rightarrow \frac{1}{M}$

$n=1$; $k=1 \Rightarrow \frac{1}{M}.$

$$h[n] = \begin{cases} \frac{1}{M} & ; 0 \le n < M \\ 0 & ; \text{otherwise} \end{cases}$$

FIR filter (finite values).

{"Finite Impulse Response".}

leaky integrator — impulse response ?

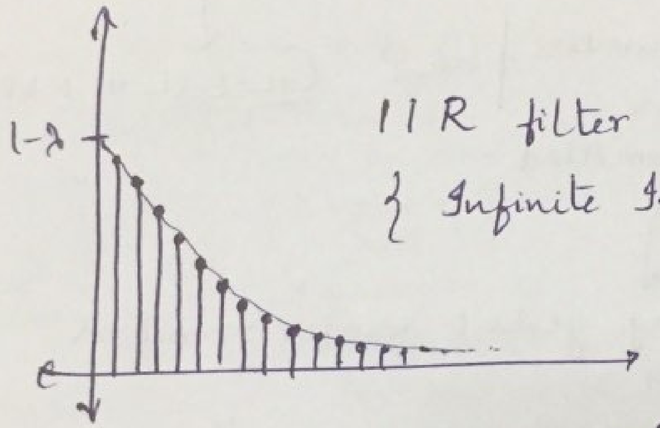
$$h[n] = (1-\lambda)\,\delta(n) + \lambda\, h[n-1]$$

for Ex: $n=0;\quad h[0] = (1-\lambda)\,\delta(0) + \lambda \underbrace{h[-1]}_{0} = 1-\lambda$

$$h[1] = (1-\lambda)\,\delta(1) + \lambda\, h[0] = \lambda(1-\lambda).$$

$$\vdots$$

$$h[n] = \begin{cases} 1-\lambda\ ; & n=0. \\ \lambda(1-\lambda)\ ; & n=1 \\ \vdots & \\ \lambda^K(1-\lambda)\ ; & n=K \end{cases}$$



IIR filter

{ Infinite Impulse Response }.

Leaky Integrator ≡ IIR filter.

CDE.

$$\sum_{k=0}^{P-1} a_k\, x(n-k) = \sum_{r=0}^{q-1} b_r\, y(n-r)$$

leaky integrator can be expressed like this.

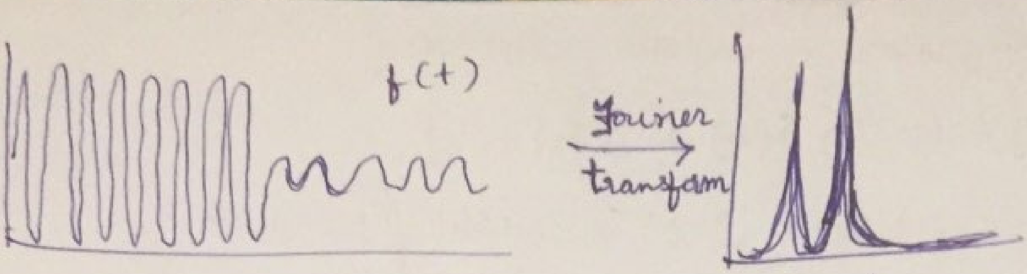
$$y[n] - \lambda\, y[n-1] = (1-\lambda)\, x[n].$$

{λ shouldn't be too high, as the current value contribution decreases.}

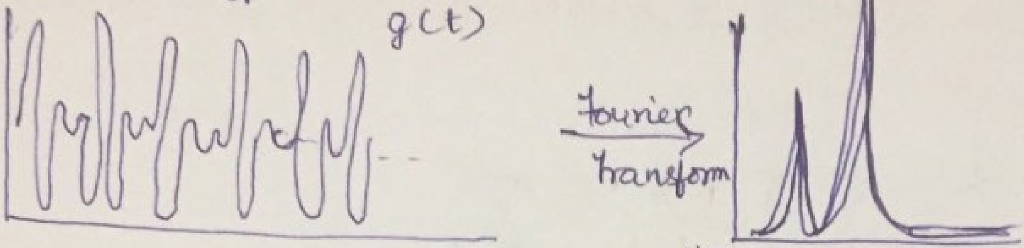
Short Time Fourier Transform:

For Ex:

$$g(t) = \begin{cases} 2\times \sin(2\pi \times 39t), & 0<t<1/2 \\ \sin(2\pi\times 15t), & 1/2<t<1. \end{cases}$$



or



f(t) & g(t) = though are different in time domain
↓
give same Fourier
↓
GLOBAL IN NATURE

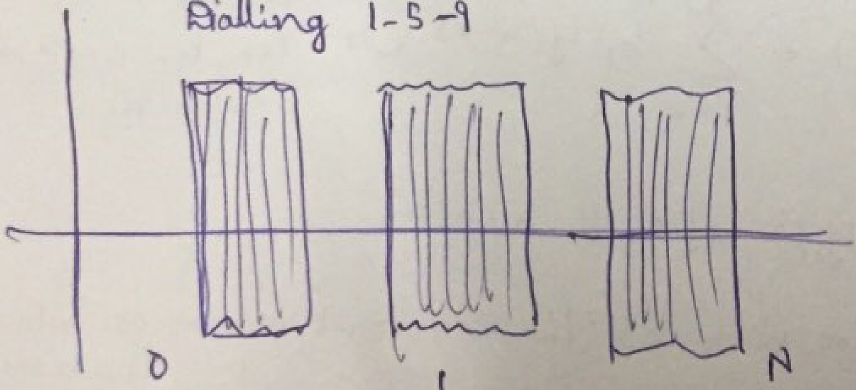
Telephones, on dialling numbers, different frequencies are (very carefully chosen) transmitted
coprime numbers

| Why?

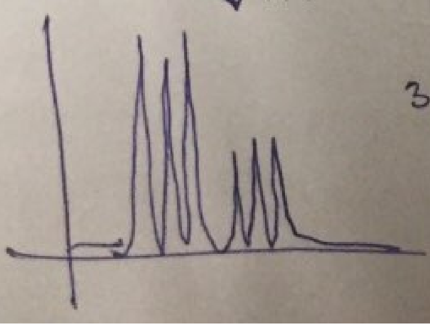
↓

On taking global fourier transform
— Limitations —
— order of dialling
— Number of dials
} can't be figured out.

Dialling 1-5-9



↓ FFT



3 pairs for 3 ranges.

To resolve this issue ; → sliding window fourier transform

but we lose on "resolution in fourier transform"

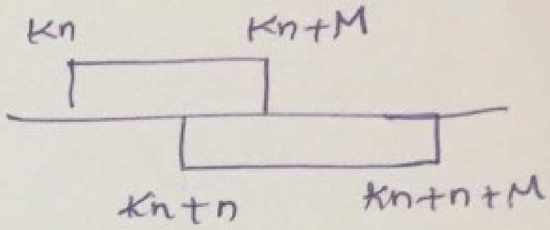
intervals – less, frequency is roughly calculated.
(samples are less)

$\boxed{\text{uncertainity} \quad \Delta t \cdot \Delta f = 2\pi}$

resolution in time & frequencies

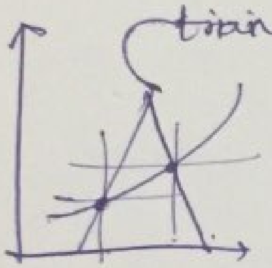
Windowing → break the signal into blocks "windows" in time domain.

Apply DFT to each window independently



$n = M$
(No overlap).

Adjacent windows may overlap.

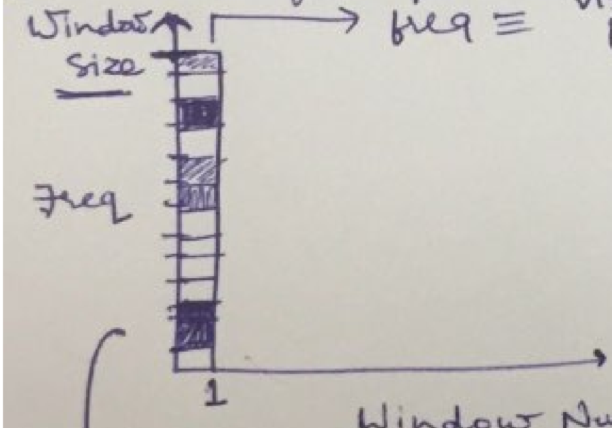


triangular window can be used [instead of rect]

* gaussiam is not advisable for ~~cont~~ analog

resolution of freq.

freq ≡ visualised in terms of colour | as per each window.



→ graph ≡ Spectrogram.

Can be used in Music also

Lighter ≡ ↑ DFT Magnitude
Darker ≡ ↓ DFT Magnitude

Compression →

1 movie (2hrs) = $2 \times 60 \times 60 \times 30 \times 1920 \times 1080 \times 24$ = 1.074×10^{13} = 10TB

What parameters can be reduced to reduce the size?

How come you are able to store it on the computer?.

- temporal redundancy
- spatial redundancy
- $\boxed{\text{perceptual redundancy}}$ ✓ frames that are not at all perceived.

Run length Encoding → RLE

1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0.

(1, 5) (0, 8) (1, 5) (0, 2).

Encoding

13 × 8

W W W W A A A S S S B S S

W W 4 A A 3 S S 3 B S S 2

[W A S B S] 5 × 8
[4 3 3 1 2] 5 × 8

Signal with high entropy - hard to compress.

Huffman Coding -

[Idea: smaller encoding - more frequent characters]

ex: AABRAA_KA_DAABRAA!!!

20 × 8 bit encoding

frequency:

A	!	_	B	R	D	K	F
9	3	3	2	2	1	1	

min: add them up

Variable Length Encoding

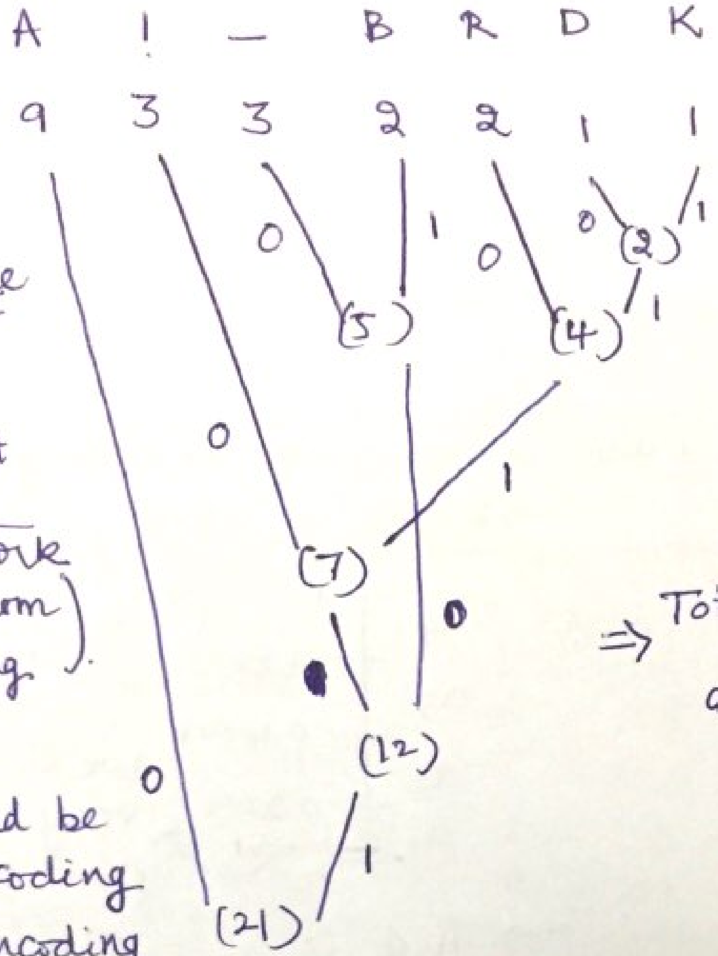
↓

structure to ~~decrypt~~ decode back is important

(doesnt work for random encoding).

↓

it should be prefix encoding as one encoding should be within the other encoding.



* prefix encoding
* dictionary coding

↓ to decode it back.

A – 0

! – 110

— – 100

B – 101

R – 1110

K – 11110

D – 11111

⇒ Total Encoding =

$9\times1 + 3\times3 + 3\times3 + 2\times3 + 2\times4 + 1\times5 + 1\times5$

$= 51$ bits.

⇒ To encode, we need the dictionary

Initial $= 21\times8$, since there are only 7 characters we can use 3 bits only so, possible bits $= 21\times3$.

Compression factor : $\dfrac{21\times3}{51}$.

"frequentist view coding"

↓

only looking at the frequency of the characters

Self Energy → $\log_2 \frac{1}{p}$ → probability of the data.

Entropy → $\sum p_i \log p_i$

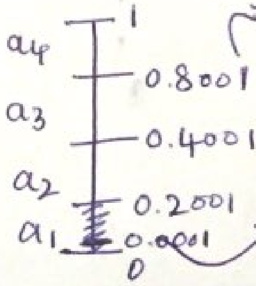
Other methods of Compression →

⇒ Arithmatic Coding (also a dictionary encoding)

Ex:

Symbol	Prob
a_1	0.2
a_2	0.2
a_3	0.4
a_4	0.2

→ Make a scale

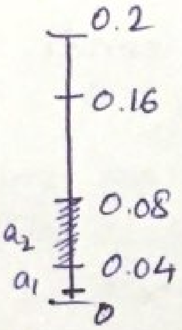


(end of message) we'll have a very small partition

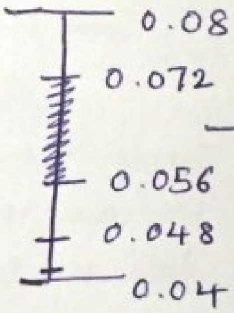
→ First is a_1 ≡ Then a_2 ≡ Then a_3

Data –
$a_1 a_2 a_3$.

So, in this encoding we store 0.06 instead of $a_1 a_2 a_3$



Partition of a_1



Partition of a_2

At the END

→ Now, store a value in between the a_3 partition

Let suppose, we store ≡ 0.06

While Decoding; first we check for 0.06 in Scale 1, we enter a_1 partition, and then again a_2 in a_1 partition and again a_3 in a_2 partition (very limited)

↓

in this process how do we know when to stop?

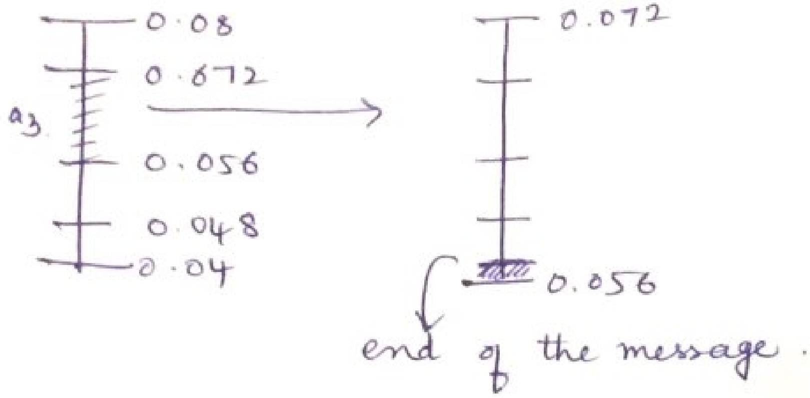
To avoid this, we ~~give~~ enter the "end of message" partition in a_3 partition.

~~Dis~~ Limitation –?

≡ When do we stop

So we store the midpoint the limits

or we'll have an "end of the message" partition of small probability.



⇒ Thus, we have encoded "$a_1 a_2 a_3$" to a number.

⇒ LZW coding → (used in Linux "compress")

No need of any dictionary (fixed) for the decoding

string: thisisthe

→ ASCII values.

t – 116
h – 104
i – 105
s – 115
e – 101

[0 – 255] – ASCII fixed values

(Large data) augment the dictionary to 10 bits.

Encoding Algorithm:

Current	Next	output	Dictionary
t	h	t (116)	th (256)
h	i	h (104)	hi (257)
i	s	i (105)	is (258)
s	i	s (115)	si (259)
is	t	is (258)	ist (260)
th	e	th (256)	the (261)
e	–	e (101)	–

9 bits

this is the ⟶ ASCII encoding = 72 bits

{116, 104, 105, 115, 258, 256, 101} → 63 bits

Decoding Algorithm:

Reconstructing the dictionary

Current	Next	Output	Dictionary
116	104	116 (t)	116 104 (256)
104	105	104 (h)	104 105 (257)
105	115	105 (i)	105 115 (258)
115	258	115 (s)	115 105 (259)

added the first char of 258 = 105

258 256 105 115 (is) 105 115 116 (260)

256 101 116 104 (th) instead of 256, add 116

101 (e) 116 104 101 (261)

→ reconstructing the dictionary for decoding similar to the encoding dictionary.

---

* FFT Algorithm ⇒ [nlogn] ✓
* Prefix coding is a property of "Huffman encoding" but you cannot leave the nodes empty.

→ jpeg Compression: → both RLE & Huffman encoding

1 frame = 1920 × 1080 × 24 bits = 6.22MB

Ways to compress → temporal redundancy
spatial redundancy
perceptual redundancy (Irrelevant information)
coding redundancy

difference between the matrices of frames — why storing every frame if there's no difference between any two frames.

Huffman coding

Not all visual information is perceived by human eye.

when added.

(Noise in frames) in few channels doesnt make diff.

changes in such channels if are not perceptible

* — can be quentized
* — reduce the size of such matrices

while reconstructing, you linearly interpolate

* reduce the size ≡ throw away the values.
⇓
Chroma sampling * look up.

Two Kinds →
* Lossless compression
* Lossy compression

Which compression algorithm is better? - always compressed "image at the user end"

↳ if all the algorithms give a compression factor, $\gamma = 10$.
↳ correlation of the result and original ≡ Quantitative measure. is tough because of " perceptual redundancy "

Quality measurement — judged by human viewers
→ five scale system on the degree of impairment.

$$e(x,y) = f(x,y) - g(x,y)$$

Signal to Noise Ratio

Overview of Image Compression →

$$SNR_{ms} = 10\log_{10}\left(\frac{\sum^{M-1}\sum^{N-1} g(x,y)^2}{MN \cdot E_{ms}}\right)$$

Input image → Forward Transform → Quantizer → Symbol Encoder → Compressed image

if there are 3 channels, (Y, Cb, Cr)
then do it for 3 channels (this flow for every channel)
(Computationally feasible)

jpeg → 8X8 patches

for each patch/block

Input image → break into nxn (8x8) patches → Forward transform → Quantizer → Symbol Encoder → Compressed image

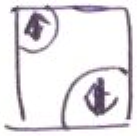
Which transform to use? → reduces the spatial redundancy
→ any of Fourier, cosine → error = less (↑ correlation)
Walsh — Hadamard transform.

Quantization matrices → different coefficients quantized with diff step-size (divide by Q (point to point))

} more than 75% entries are zero and their placement

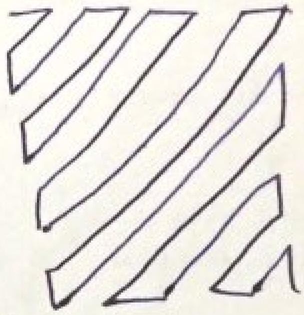
~~to~~ check again.

high mostly

image → fft () just look it up, why?

values less.

Symbol Encoding ≡ ZigZag ordering → matrix is stored using RLE / huffmen encoding → image is stored using RLE.

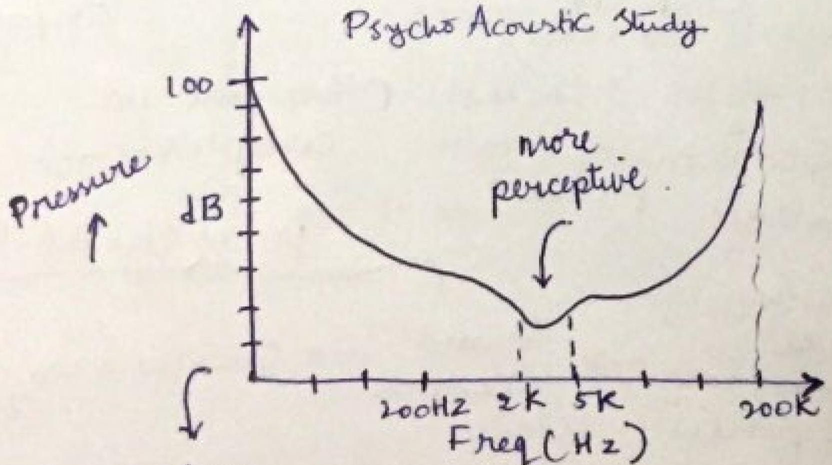


* For colour images, channel matrices ≡ different quantisation matrices

MP3 Compression

Human Ear ↓ Perception 20Hz - 20KHz

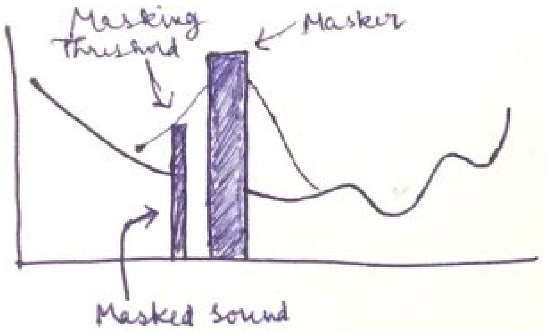
Psycho Acoustic Study



more perceptive ↓ need more bits to store } Quantization variation

2 channels ≡ left, right different perceptions.
↓
localize sounds.

<u>Masking</u> – In the presence of one sound, sensitivity to the other sounds changes (perception of other frequencies changes).



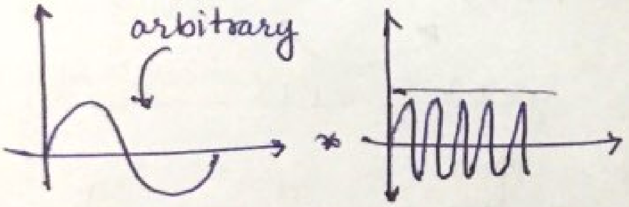
Modulation

(AM, FM ?)

AM – Amplitude Modulation

FM – Frequency Modulation.

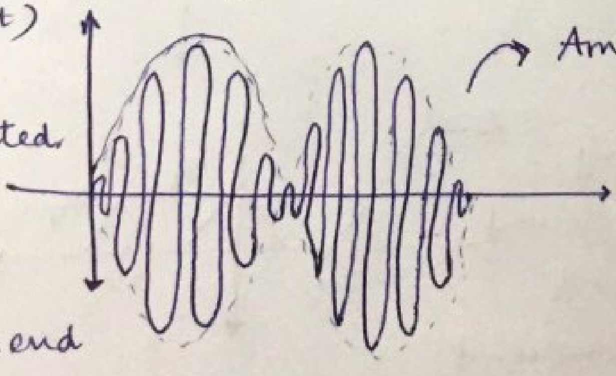
Amplitude Modulation:



$x(t)$ $\quad$ $c(t) = \cos w_c t$

$y(t) = x(t) \cdot c(t)$.

↓



$y(t)$ this is transmitted → received at the receiver end

Amplitude is changing

At the receiver end, multiply it with $\cos w_c t$

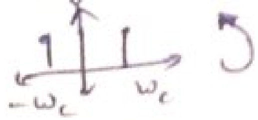
↓

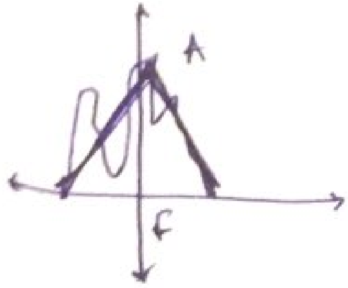
$$x(t) \cdot \cos^2 w_c t$$

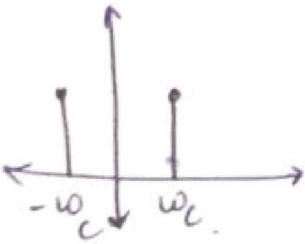
$$= x(t) \left[\frac{1 + \cos 2 w_c t}{2} \right] = \frac{x(t)}{2} + x(t) \frac{\cos 2 w_c t}{2}$$

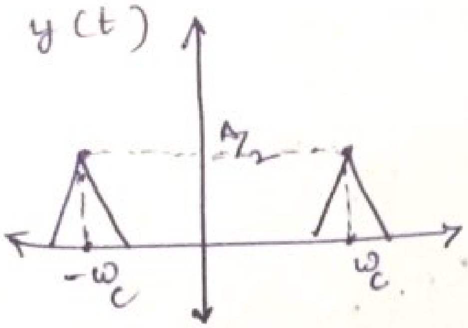
Using Fourier transform →

FFT of $\cos w_c t$









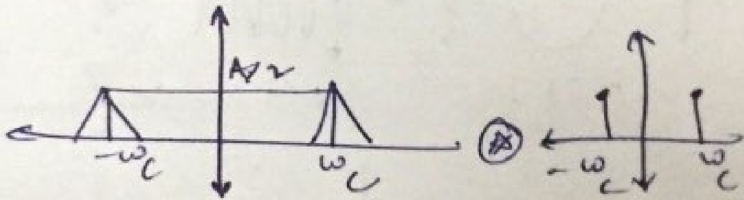
$\sin w_c t \; e^{-jwt}$

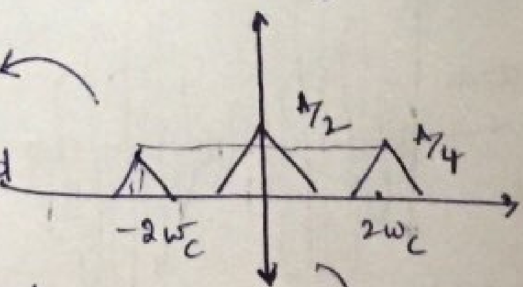
Multiplication – in fourier domain ≡ Convolution in Spatial domain

$\|^{ly}$ convolution in fourier domain ≡ Multiplication in Spatial domain

$$(F(x) * F(c)) = (x(t) \cdot c(t))$$

At the receiver end, $y(t)$ & $c(t)$





Limitations →

- bandwidth is Changing
- 'Not' the only sound that is received (Noise gets introduced in the process that affects the amplitude specifically)

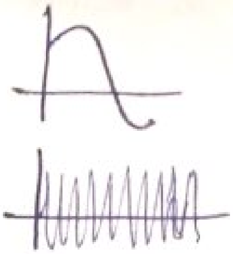
too noisy.

we hid the original signal in the amplitude

Frequency Modulation →

$$x(t) = A\cos(\omega_x t + \phi)$$

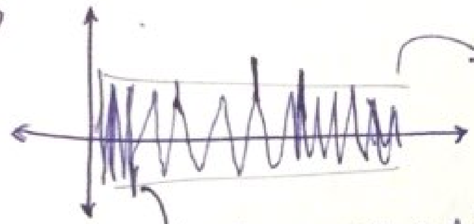
$$c(t) = A_c \sin(2\pi f_c t).$$



$$y(t) = A\sin(2\pi(f_c + x(t))t)$$

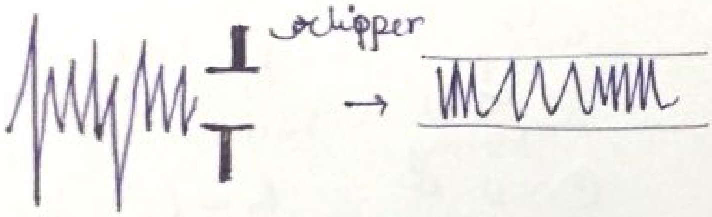
$$= A\sin(2\pi(f_c + A\cos(\omega_x t + \phi))t)$$

frequency varies



Amp. is accidently changing.

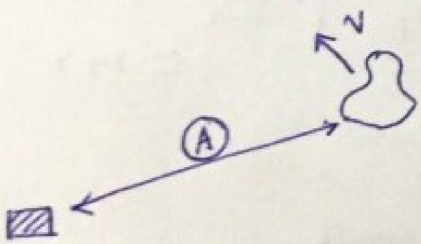
extra noise added



Hetrodyn principle –

$$2\cos\omega_a t \cdot \cos\omega_b t = \cos(\omega_a + \omega_b)t + \cos(\omega_a - \omega_b)t.$$

RADAR — Radio detection and ranging



How to detect A?

Naive method – Transmit a radiowave & captured at the receiver. (speed: c) 3×10^8

$$C = \frac{2A}{\Delta T}$$

distance = 2A

$$\Delta T = \frac{1}{44100\,\text{Hz}}$$ (time taken to record next sample)

$$A = \frac{\Delta T \times c}{2}$$

$$= \frac{1}{44100} \times \frac{3 \times 10^8}{2} = 3.4\,\text{km}.$$

* Resolution in distance [object may have moved 4 km till I take next sample, in Δt]

↳ (very bad)

Resolution in sampler affects the resolution in distance

Can cascading the radio waves change? the resolution.

Cannot resolve distances less than 4 km.

DOPPLER RADAR :-

[redacted] 4120198

Frequency perceived depends on the relative motion of the source & receiver.

Formula; $f_r = \left(\frac{c+v_r}{c+v_s}\right) f_t$

source ≡ police

Stationary

① fired with frequency f_t

Moving

② reflects what it receives

$f_r' = \left(\frac{c+v_r}{c}\right) f_t$

source ≡ car

$f_r = \left(\frac{c}{c-v}\right) f_r'$

$f_r = \left(\frac{c}{c-v}\right)\left(\frac{c+v}{c}\right) f_t$

what is received back.

$= \left(\frac{c+v}{c-v}\right) f_t = \left(\frac{1+v/c}{1-v/c}\right) f_t$

$= (1+v/c)(1+v/c) f_t \quad \left[\because \frac{1}{1-r} = 1+r+r^2 \ldots \right] \approx 1+r$

$r \ll 1$

$= \left(1+\frac{2v}{c}+\frac{v^2}{c^2}\right) f_t$

$\frac{v^2}{c^2} \simeq$ too small

$f_r = \left(1+\frac{2v}{c}\right) f_t$.

$f_r - f_t = \frac{2v}{c} f_t \rightarrow f_r - f_t = \frac{2v}{c} f_t$

$\Delta f = \frac{2v}{c} f_t$

(Speed gun)

$f_t = 2.4\,GHz$

$\Delta f = 24\,Hz$ $\qquad v = \dfrac{24 \times 3 \times 10^8}{2.4 \times 10^9 \times 2} = 3/2 = 1.5\,m/s$

$c = 3 \times 10^8\,m/s$

$f_t = 2400000000$ } this diff → require very high decimal computation.

$f_r = 2400000024$ } → taking fft at what <u>resolution</u>?

↓ use Retrodyn principle;

$2.\cos f_r . \cos f_t = \cos(f_r + f_t) + \underbrace{\cos(f_r - f_t)}_{24\,HZ}$

$\quad 4.8\,GHz$

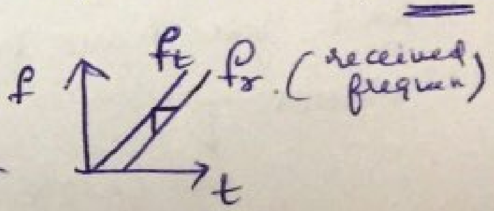
↓ low pass filter

↓ fft → Δf

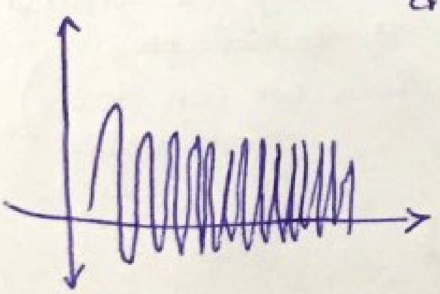
↑ Now, low pass filter ↓ and then fft ↓ of 200 terms, determine $\equiv$ 24Hz

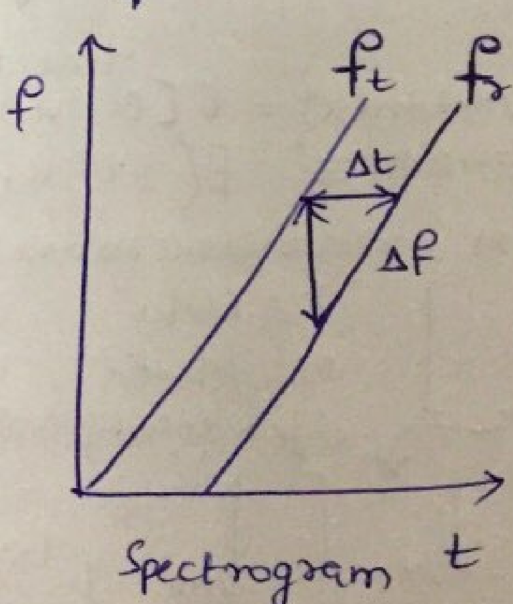
<u>Chirp</u> – frequency varies over time.

$\dfrac{df}{dt}$ → linear.



Spectrogram



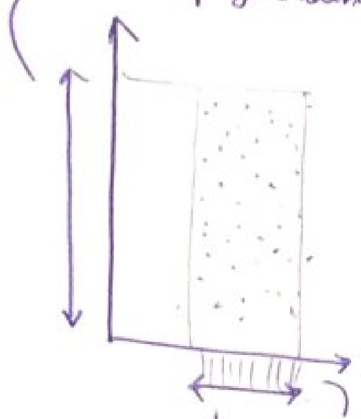


$$\frac{df}{dt} = \frac{\Delta f}{\Delta t}$$

Spectrogram

PCA - Principal Component Analysis

when projection on y-axis



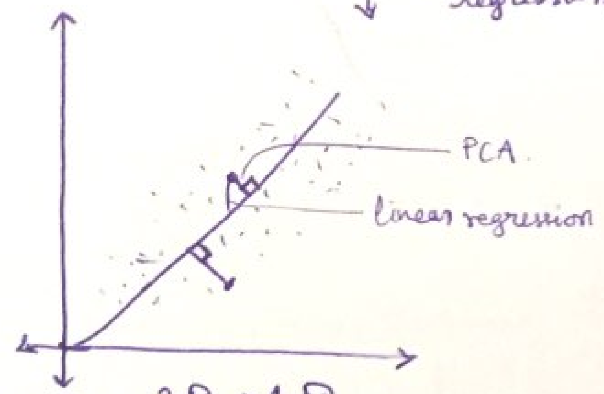
for conversion from 2-dimension to 1-dimensional data

— can be projected onto axes [but not accurate]

when projected on x-axis

variation is low

very different from linear regression



2D → 1D

looking for one direction

maximize the variance

$= \max(\mathrm{var}(v^T X))$

⇒ If there are 100 datavalues, and you have to store only 5 values

we check for correlation → (if they are similar ~~it is high~~, then we can skip that value).

done by PCA.

$\text{Variance}(X) = E[(X-u_x)(X-u_x)^T]$

$\text{Covariance} \equiv E((x-u_x)(y-u_y))$

$$\begin{array}{c|c} & X \quad M \\ \hline X & \left[\begin{matrix} \cdot & \\ & \cdot \end{matrix}\right] \\ M & \end{array}$$

→ Covariance Matrix.

X ≡ Data

M ≡ Marks

eigen decomposition

$[\ ]_{2\times1}$ $[\ ]_{2\times1}$ (large eigen values)

eigen vectors.

$A v = \lambda v$ (λ: eigen value)

$(A - \lambda I).v = 0$

suppose there are 4 data values

$$C = \begin{array}{c|c} & x \quad y \quad z \quad v \\ \hline \begin{matrix} x \\ y \\ z \\ v \end{matrix} & \left[\begin{matrix} \quad & \quad & \quad & \quad \\ & & & \\ & & & \\ & & & \end{matrix}\right] \end{array}$$

(Covariance matrix)

Covariance of x & y

do eigen decomposition

4-vectors ⇒ 4 eigen vectors

$[\ ]_{N\times2}$ ← multiplied with eigen vector. → single dimension values

N such (data, mark) values.

$[\ ]_{N\times1}$

Scanned by CamScanner

large dimensionality → smaller dimensionality with accurate results → done by → PCA

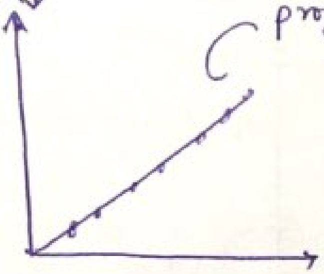
you get back the original matrix when multiplied by eigen matrix → we lose some values

PCA → new less dimension matrix

Variance preserved →

~~Variance preserved~~

$$= \frac{\sum_{1}^{k} \lambda_k}{\sum_{i}^{n} \lambda_i}$$ large k's.



$$= \max(\mathrm{var}(U^T x))$$

$$\mathrm{Var}(x) = E\left[(x - u_x)(x - u_x)^T\right]$$

$$= \max\left(E\left(U^T x - E(U^T x)\right)\left(U^T x - E(U^T x)\right)^T\right]$$

$$= \max\left(E\left[(U^T x - U^T u_x)(U^T x - U^T u_x)^T\right]\right)$$

$$= \max\left(E\left[(U^T x - U^T u_x)(x^T U - u_x^T U)\right)\right)$$

$$= \max\left(E\left[U^T (x - u_x)(x - u_x)^T U\right]\right)$$

$$= \max\left(E\left[U^T \underbrace{(x - u_x)(x - u_x)^T} U\right]\right) = \max(U^T C U)$$

C → covariance matrix

$\max(\mathrm{var}(U^T x))$, subject to $U^T U = 1$

→ $\max(U^T C U)$, subject to $U^T U = 1$

→ $\max(U^T C U - \lambda U^T U)$ → differentiate wrt U

$$\frac{\partial}{\partial U}\left[U^T C U - \lambda U^T U\right] = 0.$$

$$CU = \lambda U.$$

$\max(U^T \lambda U)$

↓

$\max(\lambda U^T U)$

↓

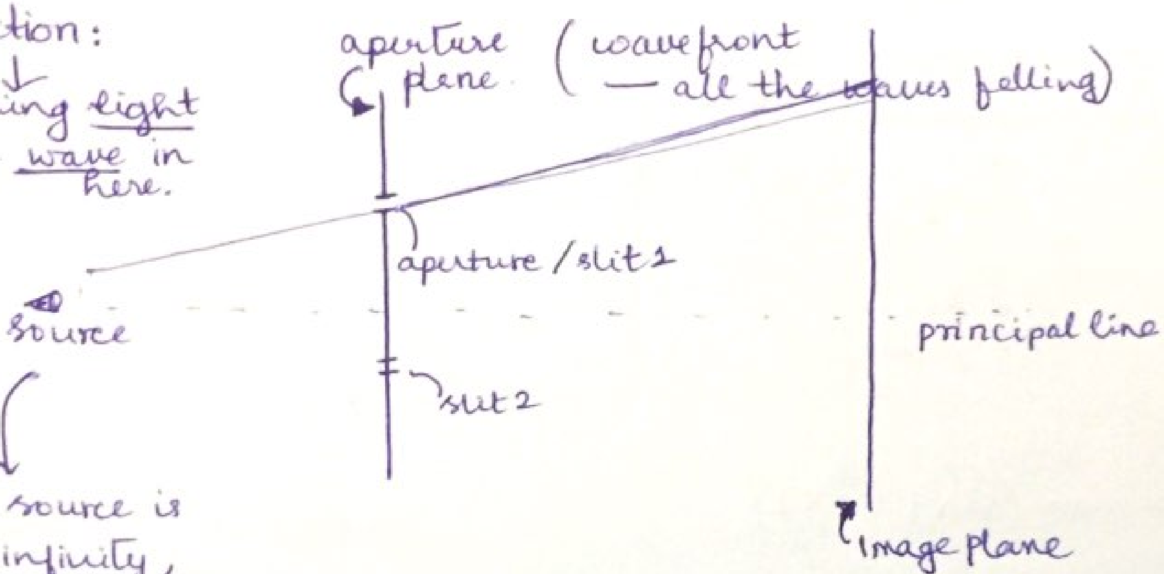
max → high values of λ.

Fourier Optics

used in medical imaging (CT scans, imaging etc)

Diffraction:

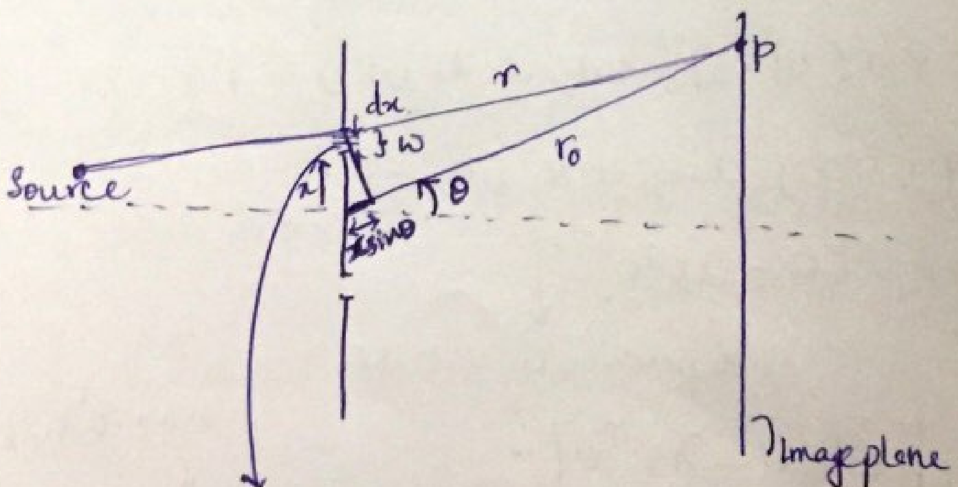
Assuming light to be wave in here.



if source is at infinity, "phase diff = 0" → all the waves falling on the wavefront have the same phase.

Diffraction $\xrightarrow{\text{based on the distance between image plane \& aperture}}$
- ↗ Fresnel diffraction (less dist)
- ↘ Fraunhofer diffraction ✓ (relatively farther).

Light falling (EM wave) ⟹ $E = E_0 \cdot e^{2\pi i \nu t}$



if light passes through the slit, each point becomes a virtual source by itself.

Huygen's Principle

Huygen's Principle → Each point of the aperture becomes a virtual source.

$$dE = E\,dx$$

$$= \underbrace{E_0 e^{2\pi i \nu t} \cdot e^{-\frac{2\pi i r}{\lambda}}}\,dx$$

actually the delay due to the extra distance being covered.

energy falling on P, coming from the source "dx" out of the whole slit.

For the total E coming through w, integrate.

$$E_p = \int_{\text{aperture}} E_0 \cdot e^{2\pi i \nu t} \cdot e^{-2\pi i r/\lambda}\,dx.$$

at point P

independent of x.

$$= \underbrace{E_0 e^{2\pi i \nu t}}_{C} \int e^{-2\pi i r/\lambda}\,dx$$

r is the only term dependent on "x"

Fraunhofer Approximation ≡ $\boxed{r \simeq r_0 - x\sin\theta}$

$$E_p = C \int_{\text{aperture}} e^{-2\pi i (r_0 - x\sin\theta)/\lambda} \cdot dx$$

$$= C \cdot \int_{\text{aperture}} e^{-2\pi i r_0/\lambda} \cdot e^{2\pi i x \sin\theta/\lambda}\,dx.$$

constant

$$= C \int_{\text{aperture}} e^{2\pi i x \sin\theta/\lambda}\,dx.$$

let $\frac{\sin\theta}{\lambda} = q$

$$E_p = C \cdot \int_{\text{aperture}} e^{2\pi i x q}\,dx$$

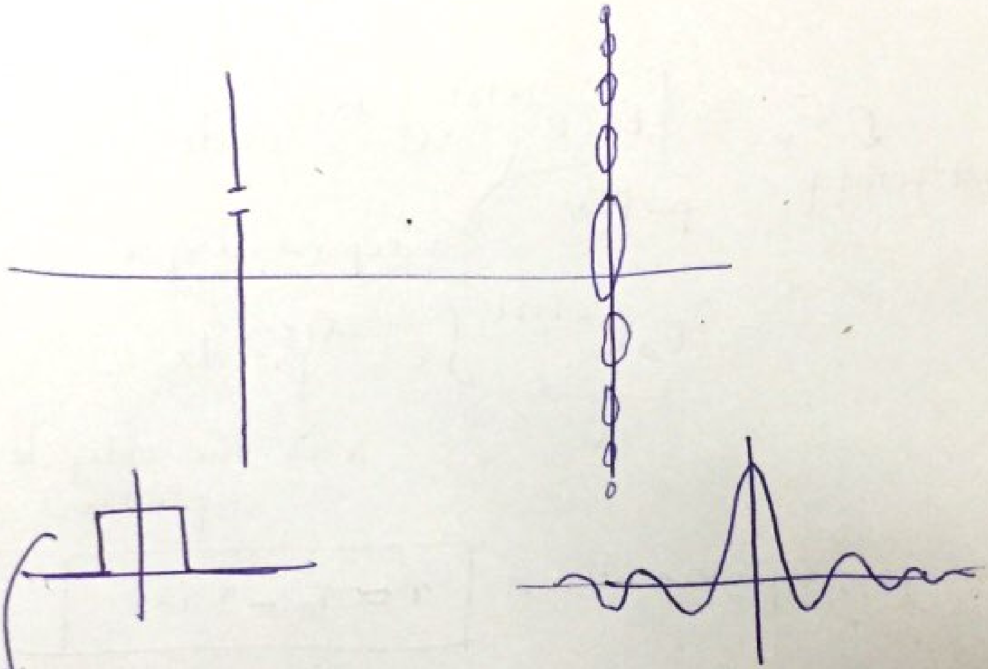
let $A(x) = \begin{cases} 1 & ; \text{ if light passes through} \\ 0 & ; \text{ otherwise} \end{cases}$

$$E_p = c \cdot \int_{-\infty}^{\infty} A(x) \cdot e^{2\pi i x q} \, dx.$$

→ (actually inverse fourier transform)

" Fourier transform in diffraction" & ≡ Pattern is fourier transform of apertures (For Ex square, circle ...).

can simulate for different apertures.



① $A(x) = \Pi_a(x)$

$E_{\text{at aperture}} \propto a \, \text{sinc}(ap)$ [where a is the width of the opening]

$= a \, \text{sinc}\left(\frac{a \sin\theta}{\lambda}\right).$

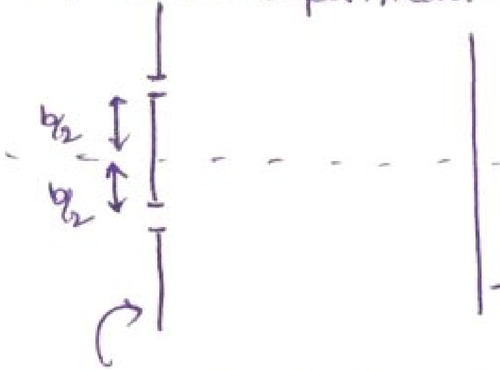
② $A(x) = \Pi_a(x - b/2) + \Pi_a(x + b/2).$

(double slit).

$$E_p \propto 2\cos(\pi b p) \, a \, \text{sinc}(ap)$$

$$= 2a\cos\left(\frac{\pi b \sin\theta}{\lambda}\right) \text{sinc}\left(\frac{a \sin\theta}{\lambda}\right)$$

double slit experiment.



"uniform pattern" everywhere

· delta function ⟶ fourier transform of delta function. $=1$

Can noise be helpful?

Noise can be helpful at times.

If we want to binarise the image ⟹ take the most significant bit of the 8 bits and truncate the other 7 bits.

info is usually lost

Quantisation

to restore ~~the~~ (preserve) some information

we can add noise to the image and then truncate (quantize).

adding noise before quantisation

usually done in mp3, ...

Dither → Noise helps preserving information

Spectral Power density: stochastic signal processing.

random signals.

used for light sources, tides, ground vibration

$$E = \lim_{N\to\infty} \sum_{n=-N}^{N} |x(n)|^2 = \infty$$

$$\text{"}P = \lim_{N\to\infty} \sum_{n=-N}^{N} |x(n)|^2 = 1\text{"}.$$

measure to talk about random signals (in frequency domain)