24/9/25

DSP lecture 1

Signal :- any physical quantity that changes with time.

e.g. speech, images (2d signal), video, etc.

mathematical representation :- $A(t) = 5t$, many signals can't be represented like this.

~~Sign~~ * The stimulus that generates a signal is a ~~siogl~~ signal source.

~~Signal So~~ * Signal processing is any operation on a signal for any desired outcome. e.g filtering.

can be done using either → Software programs.
→ digital hardware.

Analog signal → any signal continuous in time & value.

Digital signal → next page in ☆

analog signals are converted to digital signals, after processing they're returned to analog again (using A/D converters) = DSP

Why DSP ?

#1 efficiency & size of devices in current times is thanks to DSP.

#2 allows hardware flexibility via software changes only.

Signal Classifications

①

* Multidimensional signal :-

multiple independant variables.

$s(x, y)$ ~~different variables~~

* Multichannel signal :-

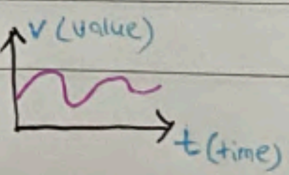
a single measured by many sources.

$$S(t) = \begin{cases} s_1(t) \\ s_2(t) \end{cases} \rightarrow \text{diff. sources}$$

②

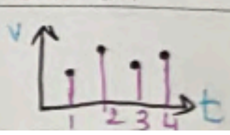
* Continuous signals :-

(analog) signals defined for every value of time from $(-\infty, \infty)$



* Discrete signals :-

only defined at certain intervals.



to convert from continuous to discrete:-

taking values every T interval, a process called $\boxed{\text{sampling process}}$ ①
time

to change the $\boxed{\text{values}}$ themselves, we choose the closest value in the discrete values range, a process called $\boxed{\text{quantization}}$ ②

when both the time & value are discrete → $\boxed{\text{digital signal}}$ ☆

Deterministic signal → can be described mathematically.

↳ else it's a random signal.

DSP lec. 2

1/10/25

trig rules:-

		reference angle:-
1st quadrant	= sin, cos +ve	~~~~ $\theta = \theta$
2nd "	= sin +ve, cos -ve	$180 - \theta$
3rd "	= sin, cos, -ve	$\theta - 180$
4th "	= cos +ve, sin -ve	$360 - \theta$ OR $-\theta$

$\boxed{F = 1/T}$ where F is frequency & T is period → analog signals

symbols:

* $x(t)$ → always analog F → analog frequency

θ → phase shift Ω → angular frequency $= 2\pi F$

A = amplitude.

$\boxed{\text{analog freq. range} = (-\infty, \infty)}$

$x(t) = A\cos(\Omega t + \theta)$ → analog signal

Discrete signals

$x(n) = A\cos(\omega n + \theta)$ → discrete signal

Symbols:

$\omega = 2\pi f$ → angular frequency f → discrete frequency.

n = sample number

k/N → freq. index, → period

discrete frequency = cycles per sample

Sampling: converting a continuous time signal to discrete time signal by taking periodic samples.

* unique range of frequency: $0 \leq \omega < 2\pi$ anything outside it causes

Aliasing: different frequencies producing same samples, ~~prod~~ distorting the signal. e.g → values repeat every 2π multiple.

$0 \leq \omega \leq 2\pi$ is the same as → $-\pi \leq \omega \leq \pi$
→ $-\frac{1}{2} \leq f \leq \frac{1}{2}$
fundamental range.

$F_s = 1/T$

f_s = sampling frequency

T = " interval

Relationship between analog & discrete

$t = nT$

t = analog time

n = discrete index

relative frequency (discrete) = $\frac{F_{analog}}{F_{sampling}}$

Sampling theorem: $F_s \geq 2F_{max\ analog}$

$F_s = 2F_{max}$ → minimum ~~largest~~ frequency fit for sampling without aliasing = Nyquist rate

8/10/25

DSP lec. 3
cont of lec. 2

ADC (Analog → digital converter) techniques :-

* PCM (pulse coding modulation) :

sampling ⟶ quantizing ⟶ encoding

Sampling methods :-

$$-\frac{F_s}{2} \le F_a \le \frac{F_s}{2}$$

folding f. ... nequisite f.

*1 Ideal sampling : sample every T_s (one point)

*2 natural sampling : short width sample, varying amplitude

*3 flat-top sampling : " " " fixed amplitude.

bandwidth : $\boxed{F_{max} - F_{min}}$

nequisite rate = choose maximum if there are multiple frequencies

Quantization.

varying → turning a sample from a range of amplitudes into one average value.

$$\Delta = \frac{max - min}{L}$$ → levels number

Δ = step size

midpoint = $\Delta/2$ error

$\boxed{2^{\text{number of bits}} = \text{levels number}}$

average quantized error power → $\frac{1}{L} \sum [x(n) - x_q(n)]^2$

continuous signal value ($x(n)$), quantized value ($x_q(n)$)

Encoding.

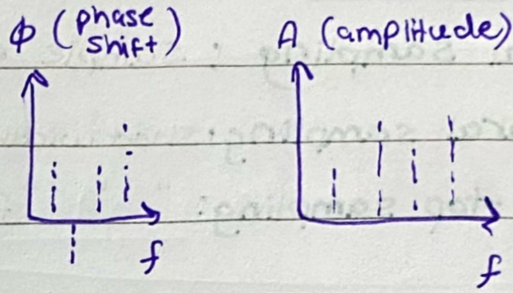
assigning a binary value to each level.

Digital to Analog.

Higher level interpolation converts digital to analog accurately but is very computationally expensive. Hence we convert using $\boxed{\text{linear interpolation + low pass filter}}$

fourier Series. lec 3.

any periodic wave can be split into its constituent sinusoidal waves.



x axis = frequency domain

- non-periodic signals can also be represented by fourier series if we let T (period) tend to ∞,

imaginary num.

$= aj + b$
- → amplitude : $\sqrt{a^2+b^2}$
- → phase shift : $\tan^{-1}\left(\frac{b}{a}\right)$

Discrete fourier transform (DFT)

$$\boxed{\Omega = \frac{2\pi}{NT}}$$

NT → time period

N → no. of samples.

Ω → stepsize.

#samples = # of frequencies

$$X(k) = \sum_{n=0}^{N-1} x(nT)\, e^{-jk\frac{2\pi n}{N}}$$

15|10|25

DSP lec. 4

DFT

* number of samples = number of components (frequency) ← k = index of component

n = index of sample

$\boxed{e^{\pm xj} = \cos(x) \pm j\sin(x)}$

* properties:-

* ① DFT is periodic every $\boxed{T}$ and its amplitude ② is symmetric every $\boxed{N/2}$ & its phase shift ③ is antisymmetric every $\boxed{N/2}$.

imaginary plane

quadrants → r = real, j = imaginary

* 1st) +r +j

2nd) −r +j

3rd) −r −j

4rth) +r −j

Inverse DFT

$$\boxed{\underbrace{x(nT)}_{\text{single sample}} = \frac{1}{N} \underbrace{X(k)}_{\text{component}} \cdot e^{jk2\pi n/N}}$$

$e^{jk2\pi n/N}$ → positive power

* low frequencies :- smooth peaks , in images leads to blurring.

* high " :- sharp peaks , " " " " high contrast.

→ caused by : $\boxed{\text{smoothing}}$:- moving average or low pass filters

→ " " : $\boxed{\text{sharpening}}$:- derivative or high pass filters.

for each point add to it the previous & next point and ÷ by 3.

DC component removal = leads to signal oscillating about x axis.

a constant component, it's frequency is 0 (aka no change).

in time domain : subtract the mean from the signal

in freq domain : remove X(0) aka the 1st harmonic.

22/10/25
DSP lec 5.

(DIT)
$\boxed{\text{Decimation in time}}$: splitting the sequence into smaller frequencies for FFT

* ① split input signal into even & odd samples.
② recursively compute DFT then combine

Radix 2 DIT : DIT algorithm for when N = power of 2.

DIT FFT steps

get binary form of indices $\{0, 1, 2, 3 \dots\}$, reverse them and then reorder samples according to new index order

properties:

1) $W_N^0 = W_1^0 = W_8^0 \dots = 1$

symmetry $N/2$

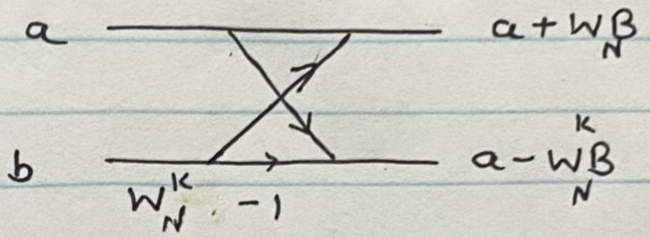
2) $W_N^{k+N/2} = -W_N^k$

periodicity

3) $W_N^{(k+N)} = W_N^k$

4) $W_{N}^{2} = W_{N/2}^{1}$

single butterfly -

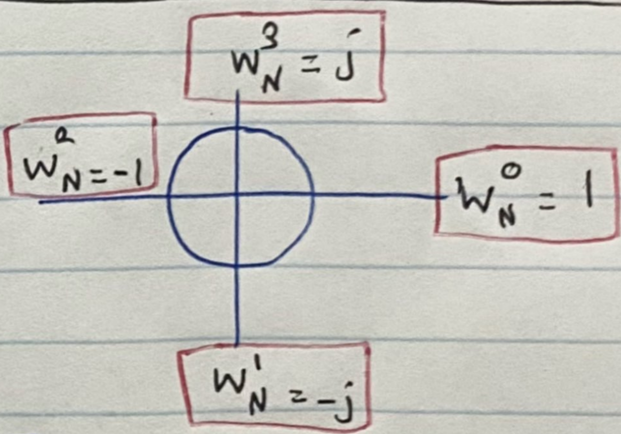


index of butterfly per stage

$$\boxed{W_N^k = e^{-j2\pi k/N}}$$

number of samples

to remember W_N^k values easily →



* Numbers of stages → $\boxed{\log_2 N}$
* " " block/stage → $\boxed{N/2^{stage}}$
* " " butterfly/block → $\boxed{2^{stage-1}}$

Decimation in frequency (DIF)

same as DIT except in reverse → starts with full group then divides into smaller components.

Signal Representation

DSP lec 6

Date No 29/10/25

#1 As a function:- $x(n) = \begin{cases} 1 & \text{for } n = \dots \\ 0 & \text{elsewhere} \end{cases}$

#2 As a table:-

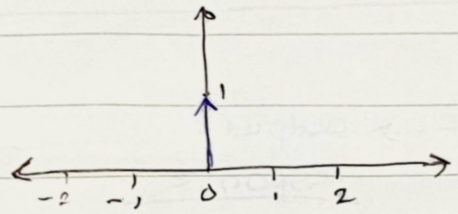
n	0	1	2	3
x(n)	1	0	1	1

#3 As a sequence:- $x(n) = \{\dots, 0, \underset{\uparrow}{1}, 2 \dots\}$

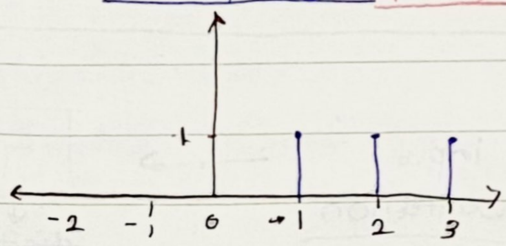
→ this arrow indicates $\boxed{\text{origin}}$, before it is negative n, after it is +ve n.

Elementry discrete signals

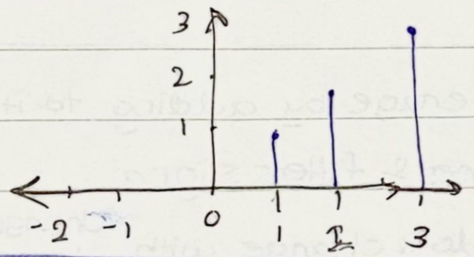
#1 Unit Sample(Impulse) signal $\boxed{\delta(n)}$ delta



#2 Unit step signal $\boxed{u(n)}$ u = unit



#3 Unit ramp signals $\boxed{u_r(n)}$



Classification of Signals

#1 Periodic & aperiodic:- $x(n+N) = x(n) \rightarrow$ periodic

#2 Symmetric & antisymmetric:- $x(-n) = x(n) \rightarrow$ symmetric

#3 Static & dynamic:
- Static: memoryless, doesn't need previous inputs.
- dynamic: needs to save previous values.

#4 Time variant & time invariant: next page.

#5 Linear vs non-linear

Signal Manipulation

1 * Delay signal by n → $\boxed{x(n-3)}$ → negative

shift right

origin +k ← ! (origin is also shifted in advance/delay) → origin −k

2 * Advance signal by n → $\boxed{x(n+3)}$ → positive

Shift left

3 * Fold signal about y axis → $\boxed{x(-n)}$

↳ delay and advance are switched in a folded signal.

4 Sampling down → $\boxed{y(n) = x(kn)}$ → $k > 1$

decreasing number of samples.

---

input / excitation → $\boxed{T}$ → output / response

T → ~~digital~~ discrete time system (tao)

Time invariant :- input & output characteristics don't change with change in time.

Time variant :- system is not consistent over time.

* To check which type a signal is :-

#1) Check if $y(n,k) = y(n-k)$, if true, system is time invariant.

$y(n,k)$ → shift every $x(n)$; $(n-k)$ → shift every n.

Linear system :-

x = each component is added as a combination, then passed as an input.

y = " " is passed as an input, then added as a ~~component~~ linear combination.

if $\boxed{x = y}$, the system is linear.

Special case :- Linear non-relaxed system → $\boxed{x(0) = \text{value}}$
of linearity check " relaxed " → $\boxed{x(0) = 0}$

Causal system :- All realtime signals (online) are causal. Depends only on current & past values. doesn't have $x(n+k)$

Non-causal systems :- Only offline systems. Depends on future values

online : real time, can compute output instantaneously.

offline : needs all input signals to be available prior to computing.

Stable system :- Bounded input always gives bounded output.

Unstable System :- " " may give unbounded " (aka. infinity).

$\boxed{\text{e.g.}}$ $1/x$.