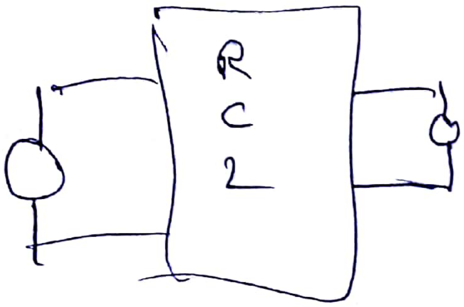
Lect. - 29 26-3-18

Where is Fourier Transform useful?

Akash: Electrical circuits, to analyze signals.

i/p sine wave



o/p. A complex signal.

↓

Analyze it.

Fourier transform tells you frequencies present & amplitudes associated with those frequencies.

Tangent: In a kayak.

signal.

A

same time period: Same single freq.

$2\frac{1}{2}$ sine waves T_1 noise, again

Sum of two sine waves with nearby frequencies. Useful in music.

$S(t) = A \sin 2\pi f_1 t + A \sin 2\pi f_2 t$

f_1 & f_2 are very close

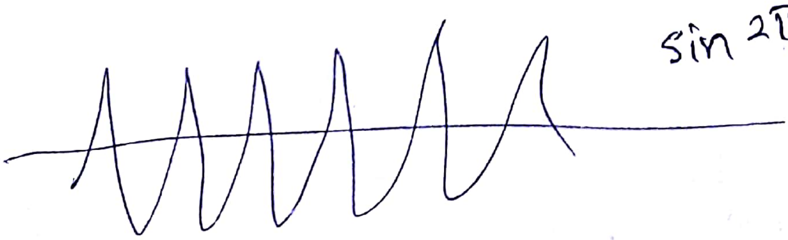
Beats:

$$s(t) = A \sin 2\pi \frac{(f_1+f_2)}{2} t \cos 2\pi \frac{(f_1-f_2)}{2} t$$

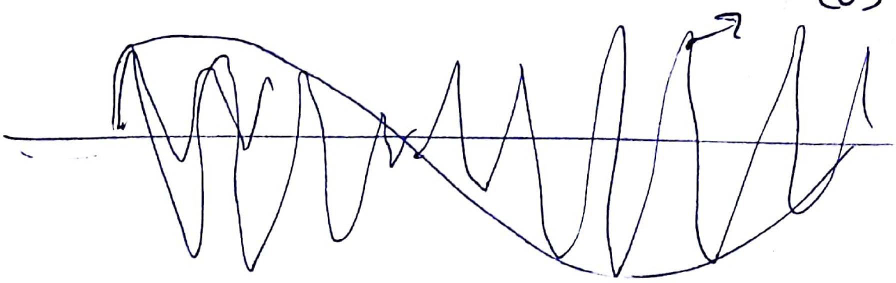
take $f_1 = 10$ $\quad f_2 = 10.1$

$\frac{f_1+f_2}{2} = 10.05 \qquad \frac{f_2-f_1}{2} = .05$

$\sin 2\pi (10.1) t$



$\cos 2\pi (.05) t$



Scanned by CamScanner

L-29

Revise

$$f(x) = \sum_{n=0}^{\infty} a_n e^{inx}$$

→ Can this always be done?

Try writing $\sin x$.

$$\sin x = \frac{1}{2i} e^{ix} + \frac{1}{(-2i)} e^{-ix}$$

→ No. Negative exponents are needed.

→ For real functions, is there a relationship between a_n & a_{-n}?

$$\boxed{a_n = a_{-n}^*} \Rightarrow \boxed{a_n^* = a_{-n}}$$

→ Why? True?

Real functions have the property that $f^* = f$.

Take $f(x) = a_{-n} e^{-inx} + a_n e^{inx}$

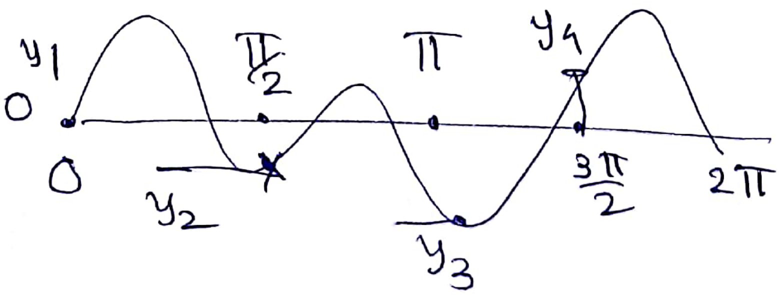
$f^*(x) = (a_{-n})^* e^{inx} + a_n^* e^{-inx}$

$a_n^* = a_{-n}$ for real functions.

Scanned by CamScanner

An example of a Fourier series.

$$f(x) = c_0 + c_1 e^{ix} + c_2 e^{2ix} + c_3 e^{3ix}$$



$$c_0 + c_1 + c_2 + c_3 = f(0) = y_1$$

$$c_0 + c_1 e^{\frac{i\pi}{2}} + c_2 e^{i\pi} + c_3 e^{\frac{3i\pi}{2}} = f(\tfrac{\pi}{2}) = y_2$$

$$c_0 + c_1 e^{i\pi} + c_2 e^{2i\pi} + c_3 e^{3i\pi} = f(\pi) = y_3.$$

$$c_0 + c_1 e^{\frac{3\pi}{2}i} + c_2 e^{3\pi i} + c_3 e^{\frac{9\pi}{2}i} = f(\tfrac{3\pi}{2}) = y_4$$

The question is given the data y_1, y_2, y_3, y_4, find the amplitudes c_0, c_1, c_2, c_3 corresspronding to the frequencies $n = 0, 1, 2, 3$.

The corresponding matrix eqn. is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{\frac{i\pi}{2}} & e^{i\pi} & e^{\frac{3\pi}{2}i} \\ 1 & e^{i\pi} & e^{2\pi i} & e^{3\pi i} \\ 1 & e^{\frac{3\pi}{2}i} & e^{3\pi i} & e^{\frac{9\pi}{2}i} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

known : Assumed frequencies — Unknowns. — known.

Find inverse of the matrix & multiply on both sides to get the amplitudes,

→ For inverse Fourier transform, given the amplitudes one can find the data.

In general for an n point data:

$$F \vec{c} = \vec{y}$$

$$\Rightarrow \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{(n-1)} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & & & & \\ 1 & \omega^{n-1} & \cdots & & \omega^{(n-1)^2} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$\omega = e^{\frac{2\pi i}{n}}$. The Fourier matrix F

$$\vec{c} = F^{-1} \vec{y}$$

$$F^{-1} = \frac{1}{n} \begin{bmatrix} 1 & 1 & \cdots & & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \vdots & & & & \\ 1 & \omega^{-(n-1)} & \cdots & & \omega^{-(n-1)^2} \end{bmatrix}$$

Mathematically correct but rarely used computationally why?

A more efficient technique is to use FFT (the fast Fourier Transform)

Mayank : Fast.

Insight From Drashti : Accuracy.

Less resources : memory :

Shubham.

In FFT n is always taken as an integer power of 2.

$$y_j = \sum_{k=0}^{n-1} \omega_n^{jk} c_k.$$

A single eqn. from the matrix multiplication.

Rewriting the sum with even and odd parts :

$m = n/2.$

$$y_j = \sum_{k=0}^{m-1} \omega_n^{2jk} \underline{\underline{c_{2k}}} + \cancel{\sum_{k=0}^{m-1} \omega_n^{2jk+1}}$$

$$+ \sum_{k=0}^{m-1} \omega_n^{(2k+1)j} c_{2k+1}$$

Use $w_n^2 = w_m$.

$$\Rightarrow y_j = \boxed{\sum_{k=0}^{m-1} w_m^{jk} c_k'} \rightarrow \text{A new name for } k^{th} \text{ coefficient}$$

$$+ w_m^j \boxed{\sum_{k=0}^{m-1} w_m^{kj} c_k''}$$

Note that

if $N = 64$ $\qquad w_N = e^{\frac{2\pi i}{64}}$

$m = 32$ $\qquad w_m = e^{\frac{2\pi i}{32}}$

$$w_m = w_N^2$$

* Note that an $n \times n$ matrix equation now "reduces" (in some sense) to two $\frac{n}{2} \times \frac{n}{2}$ $(m \times m)$ matrix eqn.s.

* Since m is also a power of two. The same process can be repeated.

FFT requires. $\frac{k}{2} N$ operations against N^2 operations of a fourier transform. $\quad N = 2^k \Rightarrow k = \log_2 N$

If N is 1024

$N^2 \cong 10^6$ Fourier transform

For FFT

$\frac{N}{2} k = 5 \cdot 1024.$

FFT is approx 200 times faster than the Fourier transform.