Convolution Properties.

Mathematical Properties.

- Convolution is commutative →
  the order in which two signals are convolved makes no difference; the result is identical.

IF

$$a[n] \rightarrow \boxed{b[n]} \rightarrow y[n]$$

THEN

$$b[n] \rightarrow \boxed{a[n]} \rightarrow y[n]$$

IF $y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$

let, $l = n-k \Rightarrow k = n-l$

THEN

$$y[n] = \sum_{n-l=-\infty}^{\infty} x[n-l] \cdot h[l]$$

$$= \sum_{-l=-\infty-n}^{\infty} h[l] \cdot x[n-l]$$

$$= \sum_{-l=-\infty}^{\infty} h[l] \cdot x[n-l]$$

$$= \sum_{l=-\infty}^{\infty} h[l] \cdot x[n-l]$$

⇓

for a sum the order doesn't matter

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

$$\boxed{x[n] * h[n] = h[n] * x[n]}$$

Commutative

It is possible to convolve 3 or more signals?

→ yes.

$\boxed{x[n] \text{ want to convolve with both } h_1[n] \;\; h_2[n]}$

Convolve two of the signals to produce an intermediate signa

$\boxed{\text{intermediate signal } x[n] * h_1[n]}$

then we convole the intermediate signal to the third signal

$$\boxed{(x[n] * h_1[n]) * h_2[n]}$$

What' should be the order of $h_1[n]$ $h_2[n]$

Does not matter.

• convolution is associative → used in system theory to describe how cascaded systems behave

IF

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

THEN

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

ALSO

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

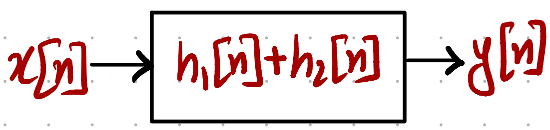
$$(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$$

Example → allows us to make bandpass filters as a sum of multiple high, low pass filters.

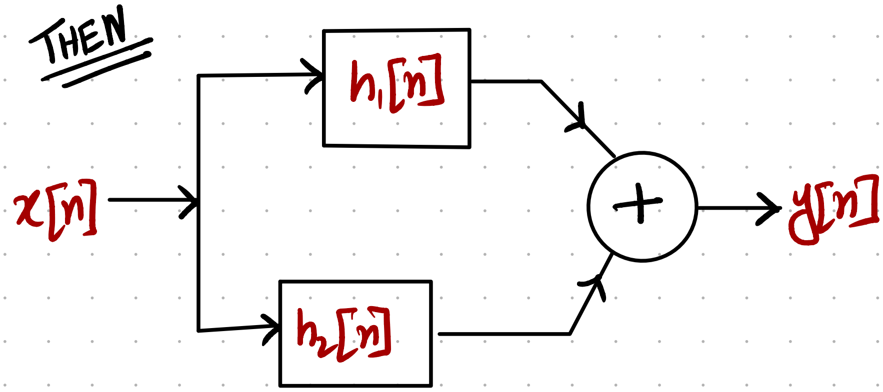
- Convolution is Distributive → describes the operation in parallel system with added output.

$$x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n])$$

IF



THEN



Test if a linear Time Invariant (LTI) is causal or not

causal → Do not get an output prior to the input that caused it (past & present)

LTI system is causal iff $h[n<0]=0$

To See

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\cdot x[n-k]$$

$$= \underbrace{\sum_{k=-\infty}^{-1} h[k]\cdot x[n-k]}_{\text{for } k<0} + \underbrace{\sum_{k=0}^{\infty} h[k]\cdot x[n-k]}_{\text{for } k\geq 0}$$

for $k<0$:

$h[-1]x[n-(-1)] + h[-2]x[n-(-2)] + \dots$

$h[-1]x[n+1] + h[-2]x[n+2] + \dots$

↓

future input

non-causal

for $k\geq 0$:

↓

$h[0]x[n] + h[1]x[n+1] + \dots$

↓

present/past inputs

↓

causal

Thus $h[n<0]=0$.

• LTI systems is stable or bounded input results in bounded output

iff $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

implies → $h[n] \to 0$ as $n \to \infty$

also

if $x[n]$ is finite duration $y[n] \to 0$ as $n \to \infty$

Finite Impulse Response (FIR)

If $h[n] \neq 0$ over finite range

AND $h[n] < \infty \quad \forall n$

Then Stable system (BIBO)

$\downarrow$

system is a Finite Impulse Response (FIR)

$\downarrow$

when you apply an impulse to the system you get an impulse response/ an output that is non-zero only for a finite amount of time

so the response to the impulse is only for a finite duration

- System has finite memory
- easy to achieve BIBO stability

$\downarrow$

just need the coefficients of the system to be not infinity

Infinite Impulse Response (IIR)

if $h[n]$ has infinite range

- System has infinite memory
  → if you have an input to the system, the effect of that system is there forever and it never goes away.

- harder to achieve BIBO stability

Why do we use reccursion/ feedbacks in digital systems?

Recurrsion gives us infinite impulse response system where we have to think more about stability

↓

It gives us a compact & better description of the system

Example cummulative sum of $x[n]$ starting at $n=0$

$$y[n] = \sum_{k=0}^{n} x[k]$$ → system with an impulse response that is a step function

Issues → computation of sum grows with n.

→ All input for $n \geq 0$ must be stored

⇒ large memory.

Efficient Recursive Formulation

$$y[n] = \begin{cases} 0 & , n<0 \\ y[n-1] + x[n] & , n \geq 0 \end{cases}$$

- computation is easy → every step need one addition.
- only one stored value
- utilizes feedback

Linear Costant Coefficient Difference Equation

$$y[n] = -a_1 y[n-1] - a_2 y[n-2] - \cdots + b_0 x[n] + b_1 x[n-1]$$ Causal formulation

— For non-causal, use future $x[\cdot]$ values

a's are → feed back co-efficients

b's are → feed forward co-efficients.

$$y[n] = \underbrace{\sum_{k=1}^{R} a_k\, y[n-k]}_{\text{past output}} + \underbrace{\sum_{k=0}^{N} b_k\, x[n-k]}_{\text{Present, Past inputs.}}$$

(non causal → future inputs)

If a's and b's are constants. ⇒ System LTI

linear, constant-coefficient difference equation.

Determining Impulse Response of a Linear Constant Co-Efficient Difference Equation

To determine $h[n]$:

1. Apply implkse $\delta[n]$ to resting system

   value 1 at time $= 0$
   value 0 at other time.

2. Assume $y[n<0] = 0$ (Causal system)

Example $\boxed{\text{Cumulative Sum}}$

$$y[n] = \begin{cases} 0 & , n<0 \\ y[n-1] + x[n] & , n \geq 0 \end{cases}$$

so,

$$h[n] = \begin{cases} 0 & , n<0 \\ h[n-1] + \delta[n] & , n \geq 0 \end{cases}$$

Thus, $h[0] = h[-1] + \delta[0] = 0 + 1 = 1$

$h[1] = h[0] + \delta[1] = 1 + 0 = 1$

$h[2] = h[1] + \delta[2] = 1 + 0 = 1$ ...

or $h[n] = u[n]$

FIR vs IIR via Difference Equation

Find $h[n]$ when $a's = 0$ $\longrightarrow$ Finite Impulse Response.

Then $y[n] = \sum_{k=0}^{N} b_k \, x[n-k]$

Find impulse Response $h[n]$

- Apply Impulse to it $x[n] = \delta[n]$

$$h[n] = \sum_{k=0}^{N} b_k \, \delta[n-k]$$

- For $n<0$ $\delta[n-m] = 0 \Rightarrow h[n] = 0$
- $n > N$, $\delta[n-k] = 0 \Rightarrow h[n] = 0$

$0 \leq n \leq N$, $h[n] = b_n$

So $h[n] = \begin{cases} b_n & , 0 \leq n \leq N \\ 0 & , \text{otherwise} \end{cases}$

- If $a's = 0$ & no feedback $\Rightarrow$ finite output for impulse input
  - FIR system.
- else $\rightarrow$ IIR system.

Solving Difference Equation

↳ we are not going to solve in the time domain but look into the frequency domain

Why do the frequency transforms?

- Direct (time domain) solution to difference equation is
  - Tedius
  - complicated when more than a few non-zero coefficients.

Discrete Time		Continuous Time
z-transform	⟷	Laplace Transform
⟵	Fourier Transform	⟶

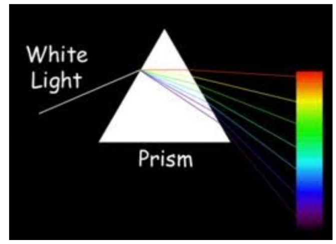
Discrete Fourier Transform

↳ variant of fourier transform

↳ evaluated at finite number of frequencies.

↳ for digital system

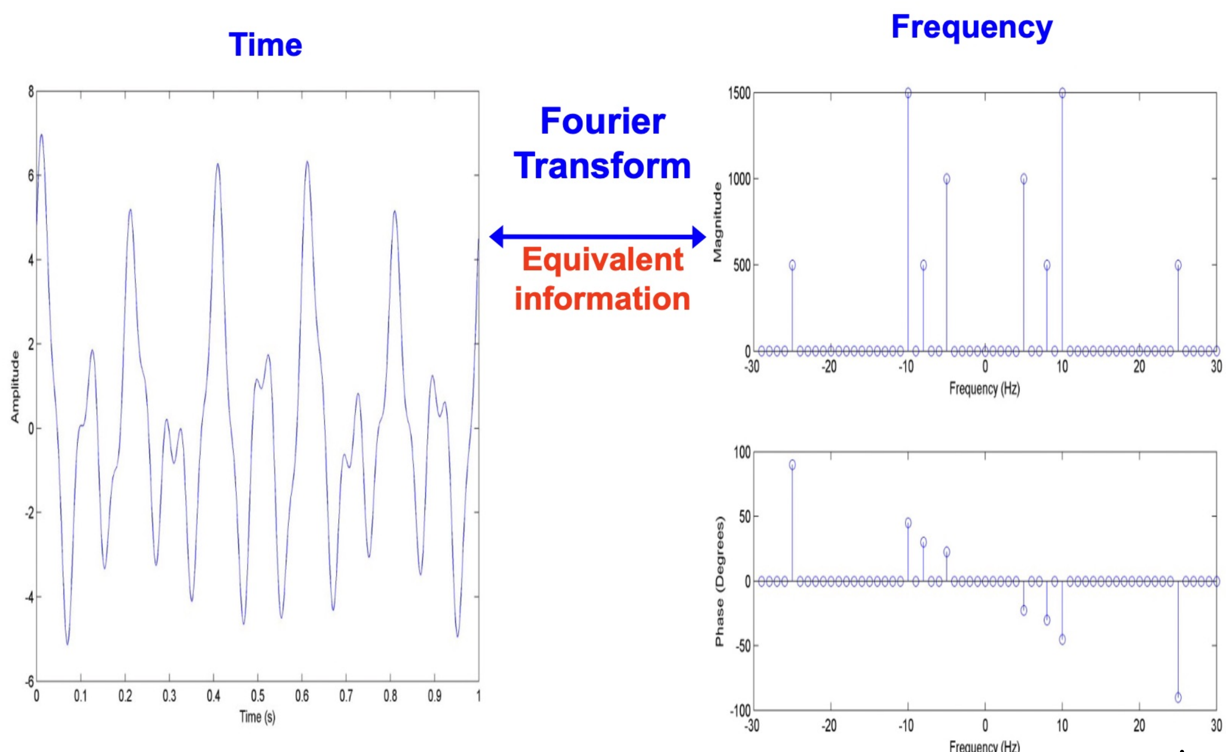
Analogy of light through Prism



chemtrails.cc

- Prism separates light into its component colors
  - Each color is an electromagnetic wave of distinct frequency
- Combinations of component colors give other colors
  - (100% Red) + (100% Green) = Yellow
  - (100% Red) + (50% Green) = Orange

Time Domain vs Frequency Domain



Time Domain

- good for localization information occurring at specific times.

Frequency domain

- good for localizing information occurring at specific frequencies
- No direct time localization information
- Fourier Transform is complex valued → magnitude & phase.