Convolution Properties.

Mathematical Properties:

• Convolution is **commutative** →

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

\[
\text{IF } a[n] \rightarrow [b[n]] \rightarrow y[n] \quad \text{THEN } \quad b[n] \rightarrow [a[n]] \rightarrow y[n]
\]

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

let \( l = n-k \Rightarrow k = n-l \)

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

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

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

\[
x[n] \cdot h[n] = h[n] \cdot x[n]
\]

**Commutative**
It is possible to convolve 3 or more signals?

→ yes.

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

Convolve two of the signals to produce an intermediate signal

intermediate signal $x[n] * h_1[n]$  

then we convolve the intermediate signal to the third signal

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

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

Does not matter.

• Convolution is associative how cascaded systems behave.

IF

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


THEN

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

Also

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

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

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:

\[ x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n]) \]
Test if a linear Time Invariant (LTI) is causal or not:

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:

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

\[
= \sum_{k=-\infty}^{0} h[k] \cdot x[n-k] + \sum_{k=0}^{\infty} h[k] \cdot x[n-k]
\]

\[
= \sum_{k=-\infty}^{0} h[k] \cdot x[n-k] + \sum_{k=0}^{\infty} h[k] \cdot x[n-k]
\]

\[
\text{for } k < 0
\]

\[
\text{for } k > 0
\]

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

\[
h[-1]x[n+1] + h[-2]x[n+2] + \ldots
\]

\[
\downarrow \text{future input}
\]

\[
\downarrow \text{non-causal}
\]

Thus \( h[n < 0] = 0 \).
LTI systems is stable, or bounded input results in bounded output

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

implies \( h[n] \to 0 \) as \( n \to \infty \)

also

if \( x[n] \) is finite duration \( y[n] \to 0 \)

\( \text{as } n \to \infty \)
Finite Impulse Response (FIR)

If \( h[n] \neq 0 \) over finite range AND \( h[n] < \infty \) for \( n \geq 0 \) then

Stable System (BIBO)

↓

system is a Finite Impulse Response (FIR)

↓

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

↓

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 recursion/feedbacks in digital systems?

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

\[ \downarrow \]

It gives us a compact & better description of the system.

**Example**

Cumulative sum of \( x[n] \) starting at \( n=0 \)

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

**Issues**

Computation of sum grows with \( n \).

\[ \rightarrow \text{All input for } n \geq 0 \text{ must be stored} \]

\[ \Rightarrow \text{large memory.} \]

**Efficient Recursive Formulation**

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

- Computation is easy → every step need one addition.
- Only one stored value
- Utilizes feedback
Linear Constant Coefficient Difference Equation

\[ y[n] = -a_1 y[n-1] - a_2 y[n-2] - \ldots \]
\[ + b_0 x[n] + b_1 x[n-1] \]

\[ \{ \text{Causal formulation} \] 

- For non-causal, use future \( x[n] \) values

\( \alpha 's \) are \( \rightarrow \) feed back co-efficients

\( \beta 's \) are \( \rightarrow \) feed forward co-efficients.

\[ y[n] = \sum_{k=1}^{R} a_k y[n-k] + \sum_{k=0}^{N} b_k x[n-k] \]

- Past output

<table>
<thead>
<tr>
<th>Past output</th>
<th>Present, past inputs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(non causal)</td>
<td>(future inputs)</td>
</tr>
</tbody>
</table>

If \( \alpha 's \) and \( \beta 's \) are constants. \( \Rightarrow \) System LTI

\[ \text{linear, constant-coefficient difference equation.} \]
To determine \( h[n] \):

1. Apply impulse \( S[n] \) to existing system

\[ \text{value 1 at time } t = 0 \]
\[ \text{value 0 at other time} \]

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

**Example**

**Cumulative Sum**

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

So,

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

Thus,

\[
h[0] = h[-1] + S[0] = 0 + 1 = 1
\]
\[
h[1] = h[0] + S[1] = 1 + 0 = 1
\]
\[
\]
\[\cdots\]

\[
h[n] = \mu[n]
\]
FIR vs IIR via Difference Equation

Find \( h[n] \) when \( a's = 0 \) \( \rightarrow \) Finite Impulse Response.

Then \( h[n] = \sum_{k=0}^{n} b_k \delta[n-k] \)

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 \( n < 0 \) \( \delta[n-m] = 0 \) \( \Rightarrow h[n] = 0 \)
- For \( n > 0 \) \( \delta[n-k] = 0 \) \( \Rightarrow h[n] = 0 \)

So \( 0 \leq n \leq N \) \( h[n] = b_n \)

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

- 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
  - Tediws
  - Complicated when more than a few non-zero coefficients.

Discrete Time

\begin{align*}
\text{Z-transform} & \quad \longleftrightarrow & \quad \text{Laplace Transform} \\
\text{Fourier Transform} & \quad \longleftrightarrow & \quad \text{Continuous Time}
\end{align*}

Discrete Fourier Transform

\begin{itemize}
\item Variant of Fourier transform
\item Evaluated at finite number of frequencies
\item For digital system
\end{itemize}
Analogy of light through Prism

Time Domain vs Frequency Domain

- **Time Domain**
  - good for localizing 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.

- 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