\[ x(t) = x(0) + \int_{0}^{t} x'(s) \, ds \]

\[ y(t) = y(0) + \int_{0}^{t} y'(s) \, ds \]

Case 1: use DFT to do circular convolve.

\[ x(n) = x_1(n) \times x_2(n) \]

\[ y(n) = y_1(n) \times y_2(n) \]

\[ x(n) = x(0) + \int_{0}^{t} x'(s) \, ds \]

\[ y(n) = y(0) + \int_{0}^{t} y'(s) \, ds \]
Take \( N \) point of \( L \) point. Define \( x_1 \)

\[ x_{L}(K) = \sum_{h=0}^{L-1} x_1(h) e^{-j \frac{2\pi h K}{L}} \]

\[ = \sum_{K=0}^{L-1} 0 \quad \text{if} \ k \neq 0 \]

\[ x_3(K) = x_2(K) x_1(K) \]

\[ = \sum_{L=0}^{L-1} 0 \quad \text{otherwise} \]
$$L \cdot \text{DTFT} \sum_{k} X_2(k) \delta_k = \sum_{0 \leq n \leq L-1} x_2(n) \delta_{L} \quad \text{otherwise}$$

$X_1(n)$

$X_2(n)$

$X_3(n)$

$X_4(n)$

$L$ point
Case 2: \( N = 2L \)

Point along central \( x_i \) and \( y_i \).

Compute 2L \( x_1, y_1 \) for \( x_i \) and \( y_i \).

\[
x(x) = \frac{1}{2L} \sum_{n=0}^{L-1} x_n(n)
\]

\[
y(x) = \frac{1}{2L} \sum_{n=0}^{L-1} y_n(n)
\]
\[ \text{IDFT}_{2L} \left\{ X_{2L}(k) \right\} = x_1 \oplus x_2 \text{ 2L pt.} \]

Pictorial circular convolution. 8.16 028
location 0 of output: \( 1x_1 \)
location 1 of output: \( 1x_1 + 1(\bar{x}_1) \)
How To Use DFT To Do Linear Convolution

Consider

\[ x_1(n) \rightarrow L \text{ pt} \]
\[ x_2(n) \rightarrow P \text{ pt} \]
\[ N > L \quad N > P \]

Goal

\[ x_3(n) = x_1 \ast x_2 = \text{linear convolution} \]
\[ x_3(n) = \sum_{m} x_1(m) x_2(n - m) \leftarrow L + P - 1 \text{ pt} \]

\[
\text{DTFT of } \{x_3(n)\} = X_3(w) = \sum_{n} x_3(n) e^{-j\omega n}
\]

Properties of DFT:

\[ X_3(w) = X_1(w) \ast X_2(w) \]

DTFT of \[ x_1(n) \]

DTFT of \[ x_2(n) \]
Suppose sample $X_3(w)$ at $N$ equally spaced points. To get $Y(k)$

$$Y(k) = \left[ X_3(w) \right]_{w=\frac{2\pi k}{N}}$$

$$\text{IDFT}\{Y(k)\} = y(n) = \left\{ \begin{array}{ll} \sum_{r=-N}^{N} X_3(n+rN) & 0 \leq n < N \\ 0 & \text{otherwise} \end{array} \right. $$

Thought Exp from 2 lectures ago.
\[ Y(k) = \left[ X_1(\omega) \right]_{\omega = \frac{2\pi k}{N}} \left[ X_2(\omega) \right]_{\omega = \frac{2\pi k}{N}} \]

- \( X_1 \rightarrow \text{LP}^+ \)
- \( X_2 \rightarrow \text{P Point} \)

\[ N > L \]
\[ N > P \]

Apply Circular Convolution:

\[ y(n) = X_1 \bigotimes X_2 \]

\( N \text{ PT Circular PT.} \)
\[ y(n) = x_1 * x_2 = \begin{cases} \sum_{r=-\infty}^{+\infty} x_3(n + rN) & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases} \]

Conclude \( N \) pt circular convolution of \( x_1 \) and \( x_2 \) is the same as their linear convolution of \( x_1 \) and \( x_2 \), periodicized with period \( N \), and take one period.

\( x_3 \rightarrow L + P - 1 \) pt sequence.

Case 0

\( N < L + P - 1 \) aliasing in time domain.
Conclude: If $N \geq L + P - 1$ \implies X_3 = X_1 \ast X_2$

\implies no aliasing

N pt circular conv of $x_1$ and $x_2$

results in linear convolution of $x_1$ and $x_2$

\implies Can use product of DFTs to do

Linear Convolution \implies DFT can be used

for LTI processing.
What is the process for computing linear Conv. using DFT

\[
\begin{align*}
    x_1 & \rightarrow L \text{ pt.} \\
    x_2 & \rightarrow P \text{ pt.} \\
    \Rightarrow x_3 &= x_1 \times x_2 \\
    L + P - 1 \\
    N > L + P - 1
\end{align*}
\]

(1) Pad \( x_1 \) with \( P - 1 \) or more zeros to get \( N \) pt. seq.

(2) Pad \( x_2 \) with \( L - 1 \) or more zeros to get \( N \) pt. seq.

(3) Take \( N \) pt. DFT of padded \( x_1 \) and padded \( x_2 \)

(4) Multiply the \( N \) pt. DFTs in step 3

(5) Take IDFT \( N \) pt. of the DFT is step 4

\[ \Rightarrow x_3 \text{ is linear conv. of } x_1 \text{ and } x_2 \]
Using DFT for LT filters of long sequence.

 FIR filter: p taps, p coeff. P is small compared to x.

Very long y(n) function.
2 Methods:

1. Overlap add
2. Overlap save

Overlap Add:

Take advantage of linearity prop of convolut

\[ [x_1(n) + x_2(n)] * h = x_1(n) * h + x_2(n) * h \]

\[ e(n) \rightarrow P \text{ pt.} \]
Steps to do overlap add:

1. Segment only long sequence into L non-overlapping chunks.
2. Convolve each L-point chunk with h to get \((L+P-1)\) new points.
3. Add up (while lining up properly) The resulting convolution of \(x_i * h\) to get \(x * h\).

See Fig 8.23 in O&S.