Chapter 3 Fourier Analysis of Discrete-time Signals

3.2

DTFT:

$$X(\omega) = \text{DTFT}\{x[n]\} = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n},$$

$$-\pi \le \omega < \pi$$

$$x[n] = \text{IDFT}\{X(\omega)\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)\, e^{j\omega n}\, d\omega,$$

$$-\infty < n < \infty$$

where $x[n]$ is the discrete-time sequence (signal), $X(\omega)$ is the discrete-time Fourier Transform (DTFT).

The DTFT of any discrete-time signal is always periodic, such that

$$X(\omega + 2\pi) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}\, e^{-j2\pi n}$$

$$= X(\omega)$$

Therefore, the DTFT spectra, namely, the magnitude and angular spectra, can be visualized in the focused primary range $[-\pi, \pi)$

Example 3.1

Consider the discrete-time signal

$$x[n] = 0.5^n u[n] \; ; \quad u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Compute the DTFT and depict the figures for both spectra.

Answer :

$$X(\omega) = \sum_{n=-\infty}^{\infty} 0.5^n u[n] e^{-j\omega n}$$

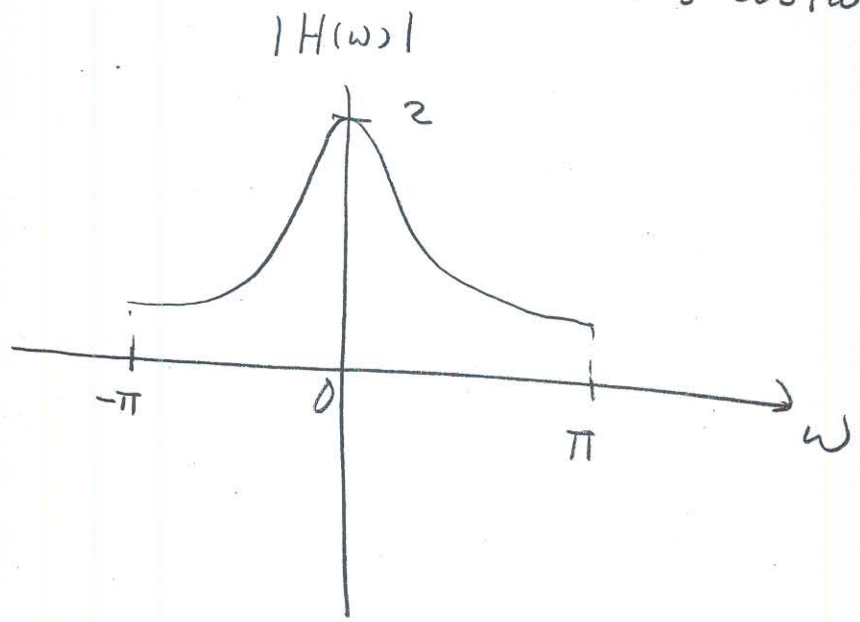
$$= \sum_{n=0}^{\infty} 0.5^n e^{-j\omega n} = \frac{1}{1-0.5e^{-j\omega}}$$

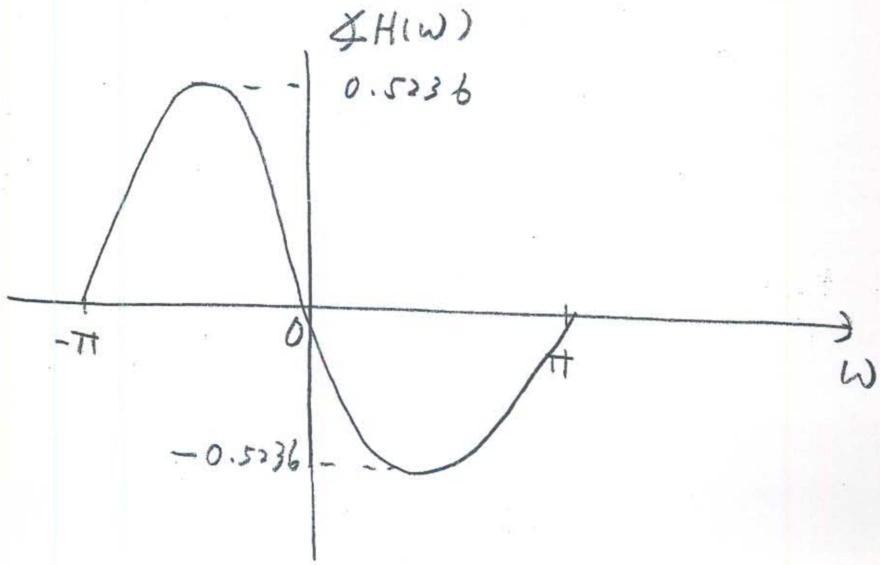
$$|X(\omega)| = \frac{1}{\left|[1-0.5\cos(\omega)]^2 + 0.5\sin^2(\omega)\right|}$$

$$= \frac{1}{|1.25 - \cos(\omega)|}$$

$$\measuredangle X(\omega) = -\measuredangle\left(1 - 0.5e^{-j\omega}\right)$$

$$= -\tan^{-1}\left[\frac{0.5\sin(\omega)}{1-0.5\cos(\omega)}\right]$$





Example: Consider the discrete-time signal

$$x[n] = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

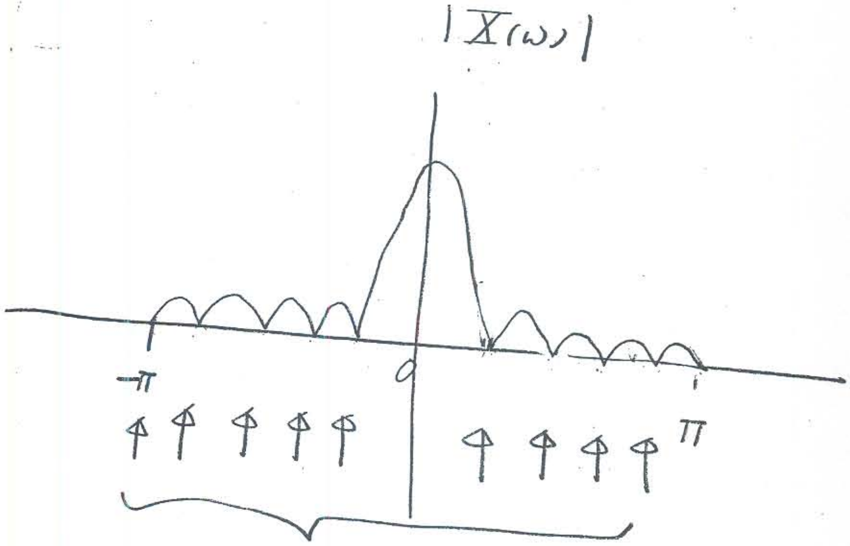
Compute the DTFT.

Answer:

$$X(\omega) = \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}$$

$$= \frac{e^{-j\frac{\omega N}{2}}\left[e^{j\frac{\omega N}{2}} - e^{-j\frac{\omega N}{2}}\right]}{e^{-j\omega/2}\left[e^{j\omega/2} - e^{-j\omega/2}\right]}$$

$$= e^{-j\omega(N-1)/2} \, \frac{\sin(\omega N/2)}{\sin(\omega/2)}$$



N-1 zeros in one period

The magnitude spectrum of such a "rectangular window" sequence has $(N-1)$ zeros in one period $[-\pi, \pi)$ as shown in the figure above.

Properties of DTFT:

a. Linearity:

$$\text{DTFT}\{a\,x[n] + b\,y[n]\} = a\,X(\omega) + b\,Y(\omega)$$

b. Time shift:

$$\text{DTFT}\{x[n-n_0]\} = e^{-j\omega n_0}\,X(\omega)$$

c. Symmety:

If $x[n]$ is real, $X(\omega) = X^*(-\omega) = X^*(2\pi-\omega)$

d. Convolution in time

$$\text{DTFT}\{h[n] \circledast x[n]\} = H(\omega)\,X(\omega)$$

e. Multiplication in time:

linear convolution

$$\text{DTFT}\{x[n]\,y[n]\} = \frac{1}{2\pi}\,X(\omega) \circledast Y(\omega)$$

f. Parseval's theorem:

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} X^*(\omega)\,Y(\omega)\,d\omega = \sum_{n=-\infty}^{\infty} x^*[n]\,y[n]$$

g. Differentiation in frequency:

$$j\,\frac{dX(\omega)}{d\omega} = \text{DTFT}\{n\,x[n]\}$$

3.3

When we sample the DTFT, we can have the discrete-frequency samples, that is DFT, for finite-support $x[n] \begin{cases} \neq 0, & 0 \le n \le N-1 \\ = 0, & \text{otherwise} \end{cases}$

$$X[k] = X(\omega)\Big|_{\omega = \frac{2\pi k}{N}}$$

$$= \text{DTFT}\{x[n]\}\Big|_{\omega = \frac{2\pi k}{N}}$$

$$= \sum_{n=0}^{N-1} x[n]\,e^{-j\frac{2\pi k}{N}n}, \quad k = 0, \dots, N-1$$

$$x[n] = \text{IDFT}\{X[k]\}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X[k]\,e^{j\frac{2\pi kn}{N}}, \quad n = 0, \dots, N-1$$

The DFT is also periodic, with period N.

$$X[k+N] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi k n}{N}}\, e^{-j 2\pi n}$$

$$= X[k]$$

Usually, the DFT, IDFT are both represented in a matrix form such that

$$\vec{X} = \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} \vec{e}_0^{\,H} \\ \vec{e}_1^{\,H} \\ \vdots \\ \vec{e}_{N-1}^{\,H} \end{bmatrix} \vec{x}$$

where $\vec{e}_k = \begin{bmatrix} 1 \\ W_N^{-k} \\ \vdots \\ W_N^{-k(N-1)} \end{bmatrix}$, $W_N = e^{-j\frac{2\pi}{N}}$

$\vec{x} = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$, $X[k] = \vec{e}_k^{\,H} \vec{x}$.

If we define Fourier Matrix $\widetilde{W}_N$ as

$$\widetilde{W}_N \equiv \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{N-1} \\ \vdots & & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

then $\quad \vec{X} = \tilde{W}_N \vec{x}$,

where $\tilde{W}_N[k,n] = \omega_N^{kn} = e^{-j\frac{2\pi kn}{N}}$,

for $0 \le n \le N-1$, $\quad 0 \le k \le N-1$

In order to formulate the IDFT also in the matrix form, we can apply the geometric series, such that

$$\vec{e}_\ell^{\,H} \vec{e}_m = \sum_{k=0}^{N-1} \omega_N^{(\ell-m)k}$$

$$= \begin{cases} \dfrac{1-\omega_N^{(\ell-m)N}}{1-\omega_N^{(\ell-m)}} = 0, & \text{if } \ell-m \neq 0 \\ N, & \text{if } \ell-m = 0 \end{cases}$$

Or $\quad \tilde{W}_N \tilde{W}_N^H = \underbrace{\begin{bmatrix} \vec{e}_0^{\,H} \\ \vec{e}_1^{\,H} \\ \vdots \\ \vec{e}_{N-1}^{\,H} \end{bmatrix}}_{\tilde{W}_N} \underbrace{[\vec{e}_0 \; \vec{e}_1 \; \cdots \; \vec{e}_{N-1}]}_{\tilde{W}_N^H} = N \tilde{I}_{N\times N}$

where $\tilde{I}_{N\times N}$ is the identity matrix of size $N\times N$.

Thus,

$$\vec{x} = \tilde{W}_N^{-1} \vec{X}, \quad \text{where } \tilde{W}_N^{-1} = \frac{1}{N} \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^{*} & \cdots & W_N^{*(N-1)} \\ \vdots & & & \vdots \\ 1 & W_N^{*(N-1)} & \cdots & W_N^{*(N-1)(N-1)} \end{bmatrix}}_{\tilde{W}_N^{H}}$$

Example:

Consider a discrete time signal

$x[0]=1, \; x[1]=2, \; x[2]=-1,$

$x[3]=-1, \; x[n]=0, \text{ for } n<0, \; n \geq 4$

Compute $X[k]$ by use of the matrix operation.

Answer:

$$\vec{x} = \begin{bmatrix} 1 \\ 2 \\ -1 \\ -1 \end{bmatrix}, \quad N=4, \quad W_4 = e^{-j\frac{2\pi}{4}} = -j$$

$$\tilde{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$\vec{X} = \tilde{W}_4 \vec{x} = \begin{bmatrix} 1 \\ 2-3j \\ -1 \\ 2+3j \end{bmatrix} \begin{matrix} \to X[0] \\ \to X[1] \\ \to X[2] \\ \to X[3] \end{matrix}$$

On the other hand,

$$\vec{x} = \widetilde{W}_4^{-1} \vec{X}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 2-3j \\ -1 \\ 2+3j \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \\ -1 \end{bmatrix}$$

Properties of Fourier Matrix $\widetilde{W}_N$

1. The Fourier matrix $\widetilde{W}_N$ is a N-dimensional Vandermonde matrix.

   ※ A Vandermonde matrix $\widetilde{A}_{N\times N}$ is a matrix of the form

$$\widetilde{A}_{N\times N} \equiv \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \cdots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & x_2^3 & \cdot & x_2^{N-1} \\ \vdots & & \cdot & \cdot & & \vdots \\ 1 & x_N & x_N^2 & x_N^3 & \cdots & x_N^{N-1} \end{bmatrix},$$

   where $\widetilde{A} = [a_{kn}]$ and $a_{kn} = x_k^{n-1}$.

2. Since $\widetilde{W}_N$ is a Vandermonde matrix,

$$\det(\widetilde{A}) = \prod_{\substack{k,n=1 \\ k>n}}^{N} (x_k - x_n),$$

$$\det(\tilde{W}_N) = \prod_{\substack{k,n=0 \\ k>n}}^{N-1} (W_N^k - W_N^n)$$

$$= \prod_{\substack{k,n=0 \\ k>n}}^{N-1} \left(e^{-j\frac{2\pi k}{N}} - e^{-j\frac{2\pi n}{N}} \right)$$

3. $\tilde{W}_N = \tilde{W}_N^T$

4. $\tilde{W}_N^{-1} = \frac{1}{N} \tilde{W}_N^*$

5. $\tilde{W}_N^2 = N\tilde{P}$, $\tilde{P}$ is the permutation matrix such that $[\vec{\varepsilon}_1, \vec{\varepsilon}_N, \vec{\varepsilon}_{N-1}, \dots \vec{\varepsilon}_2]$, where $\vec{\varepsilon}_n$ is the n^{th} column of the $N \times N$ identity matrix

For example, $\tilde{W}_4^2 = 4\tilde{P} = 4\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

6. $\tilde{W}_N^4 = N^2 \tilde{I}_N$

7. $\tilde{W}_N = \tilde{C}_N + j\tilde{S}_N$, where $\tilde{C}_N$ and $\tilde{S}_N$ are both real matrices such that

$$\tilde{C}_N = [C_{kn}]_{N\times N}, \quad \tilde{S}_N = [S_{kn}]_{N\times N},$$

$$C_{kn} \equiv \cos\left[\frac{2\pi}{N}(k-1)(n-1)\right]$$

$$S_{kn} \equiv \sin\left[\frac{2\pi}{N}(k-1)(n-1)\right]$$

8. $\tilde{C}_N \tilde{S}_N = \tilde{S}_N \tilde{C}_N, \quad \tilde{C}_N^2 + \tilde{S}_N^2 = \tilde{I}_N$

9. $\tilde{C}_N = \tilde{C}_N^T, \quad \tilde{S}_N = \tilde{S}_N^T$

Example: Prove that $\tilde{W}_N \tilde{W}_N^H = \tilde{W}_N^H \tilde{W}_N = N\tilde{I}_N$

Solution: $\tilde{W}_N^H = (\tilde{W}_N^T)^* = \tilde{W}_N^* = N\,\tilde{W}_N^{-1}$

$$\therefore \quad \tilde{W}_N \tilde{W}_N^H = \tilde{W}_N^H \tilde{W}_N = N\tilde{I}_N$$

Example: Prove that

$$\tilde{W}_N^{4a} = N^{2a}\tilde{I}_N$$

$$\tilde{W}_N^{4a+2} = N^{2a+1}\tilde{P},$$ for a is any positive integer.

Solution:

$$\tilde{W}_N^{4a} = (\tilde{W}_N^4)^a = (N^2 \tilde{I}_N)^a = N^{2a} \tilde{I}_N$$

$$\tilde{W}_N^{4a+2} = (\tilde{W}_N^4)^a (\tilde{W}_N)^2$$
$$= N^{2a} \tilde{I}_N \, N \tilde{P}$$
$$= N^{2a+1} \tilde{P}, \quad a = 1, 2, 3, \ldots$$

How to generate a Fourier matrix A of size N in Matlab?

Answer: `A = fft(eye(N));`

Cirular shift and Circular convolution

The finite-support sequence

$$x[n] \begin{cases} \neq 0, & 0 \leq n \leq N-1 \\ = 0, & \text{otherwise} \end{cases}$$

can be regarded as one period of a periodic sequence.

Since $X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}}$

$$x_p[n] = x[n+pN] = x[n]$$

$$= IDFT\{X[k]\}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{-j\frac{2\pi kn}{N}}, \quad p \in \mathbb{Z}$$

$\therefore$ $x_p[n] = x[(n)_N]$,

where $(n)_N = n$ modulo N

$\times$ Circular shift L

$$x_p[n-L] = x[(n-L)_N]$$

$$= \begin{cases} x[n-L], & \text{if } 0 \le n-L \le N-1 \\ x[n+\ell N-L], & \text{if } L-\ell N \le n \le \ell N+N+L-1 \end{cases}$$

Example:

$$x[n] = \{\underset{\substack{\uparrow \\ n=0}}{1}, -2, 5, 3, 2\}$$

a. $x[(n-1)_5] = [\underset{\substack{\uparrow \\ n=0}}{2}, 1, -2, 5, 3]$

b. $x[(n+1)_5] = [\underset{\substack{\uparrow \\ n=0}}{-2}, 5, 3, 2, 1]$

* Circular Convolution

$$y[n] = \underbrace{h[n] \circledast_C x[n]}_{\text{circular convolution}} = \sum_{m=0}^{N-1} h[m]\, x[(n-m)_N]$$

where $y[n]$, $x[n]$, $h[n]$ have the same length N

Example: $x[n] = \{ \underset{n=0}{\underset{\uparrow}{-1}}, 2, -5 \}$

$h[n] = \{ \underset{n=0}{\underset{\uparrow}{2}}, -1, -2 \}$, $N = 3$

$$y[0] = h[n] \circledast_C x[n] = x[n] \circledast_C h[n]$$

$$= \sum_{m=0}^{2} h[m]\, x[(n-m)_3]$$

$$= h[0]\,x[0] + h[1]\,x[(-1)_3] + h[2]\,x[(-2)_3]$$

$$= h[0]\,x[0] + h[1]\,x[2] + h[2]\,x[1]$$

$$= -1$$

$$y[1] = h[0]\,x[1] + h[1]\,x[0] + h[2]\,x[2] = 15$$

$$y[2] = h[0]\,x[2] + h[1]\,x[1] + h[2]\,x[0] = -10$$

Properties of DFT :

a. Linearity :

$$DFT\{a x[n] + b y[n]\} = a X[k] + b Y[k],$$

$$k = 0, 1, \cdots, N-1$$

b. Symmetry :

$$X[k] = DFT\{x[n]\} = X^*[N-k] \quad , k = 0, 1, \ldots, N-1$$

c. Cirular time-shift :

$$X[k] = DFT\{x[(n-L)_N]\} = W_N^{kL} X[k],$$

$$k = 0, 1, 2, \ldots, N-1$$

d. Circular convolution :

$$DFT\{h[n] \circledast_c x[n]\} = H[k]\, X[k] \quad , k = 0, 1, 2, \ldots, N-1$$

$$= DFT\{h[n]\}\; DFT\{x[n]\}$$

e. Multiplication by exponential :

$$DFT\{x[n]\, e^{j\frac{2\pi n M}{N}}\} = X[(k-M)_N]$$

f. Multiplication in time :

$$DFT\{x[n]\, y[n]\} = \frac{1}{N} X[k] \circledast_c Y[k]$$

g. Inner Product or Parseval's theorem:

$$\sum_{n=0}^{N-1} x^*[n]\, y[n] = \frac{1}{N} \sum_{k=0}^{N-1} X^*[k]\, Y[k]$$

Proof of a.-g. are referred to PP. 116-117 in the text.

3.4 Relationship between DFT and DTFT

The DFT can be modified as

$$X[k] = \sum_{n=n_0}^{n_0+N-1} x[n]\, e^{-j \underbrace{(\frac{2\pi}{N}) k}_{\omega} n}$$

We assume the initial time $n_0 = -\frac{N}{2}$; then

$$\hat{X}_N(\omega) \equiv \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} x[n]\, e^{-j\omega n}$$

and $X[k] = \hat{X}_N\left(\frac{2\pi k}{N}\right)$

Recall the DTFT as an infinite sum,

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

Therefore, $\hat{X}_N(\omega) \to X(\omega)$ as $N \to \infty$.

3.5 DFT for Spectrum estimation

Read for yourselves!

3.6. DFT for Convolution

The DFT can be efficiently implemented by the fast fourier transform (FFT), which will be covered later. FFT can help us efficiently compute a convolution.

A typical case is to compute a convolution such that

$$y[n] = \sum_{l=0}^{M-1} h[l]\, x[n-l],$$

where
$$x[n] \begin{cases} \neq 0, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

$$h[n] \begin{cases} \neq 0, & 0 \le n \le M-1 \\ 0, & \text{otherwise} \end{cases}$$

and $N \gg M$.

We observe such a convolution and find that

$$y[n] = 0, \quad \text{for} \quad \begin{array}{l} n-l < 0 \text{ or} \\ n-l > N-1 \end{array}$$

because $x[n-l] = 0$ in both conditions.

Therefore $$y[n] = \sum_{l=0}^{M-1} h[l]\, x[n-l],$$

$$\text{for} \quad \underbrace{0 \le n \le N+M-2}_{N+M-1 \text{ points}}$$

The goal is to use the convolution property of the DFT to comput the output sequence $y[n]$. However, there are two problems:

(1) the convolution defined with the DFT is circular (not linear).

(2) the sequences must have the same length.

Thus, we need to do the "zero-padding" at first, that is, padding zeros at the end of each sequence $x[n]$, $h[n]$ to make them of the same length, such that

$$x_e[n] \equiv \{x[0], x[1], \ldots, x[N-1], \underbrace{0, 0 \cdots, 0}_{L-N-1 \text{ zeros}}\}$$

$$h_e[n] \equiv \{h[0], h[1], \ldots, h[M-1], \underbrace{0, 0, \cdot\ 0}_{L-M-1 \text{ zeros}}\}.$$

for $L \geq N+M-1$

Then, $y[n] = \text{IDFT}\{H_e[k]\, X_e[k],\ k=0,1,\ldots,L-1\}$

where $H_e[k] = \text{DFT}\{h_e[n],\ n=0,1,2,\ldots,L-1\}$

$X_e[k] = \text{DFT}\{x_e[n],\ n=0,1,2,\ldots,L-1\}$

Convolution of Long data Sequences

It is not uncommon to have a data length N of several tens of thousands of samples and a filter length M of just a few hundred samples. Also, we would like to see the result of the convolution in almost real time, in the sense that we do not want to wait for the entire convolution to be computed.

A solution to this problem is obtained by sectioning the input data $x[n]$ into blocks of length $L > M$, where M is the filter length, $h = \{h[0], h[1], \cdots, h[M-1]\}$; then compute the convolution for each block, and recombine the results.

There are two ways to implement this sectioning convolution techniques, namely, overlap and add, overlap and save. In both cases, the data x is sectioned into the blocks:

$$x_k = \underbrace{\{x_k[0], x_k[1], \cdots x_k[L-1]\}}_{\text{the } k^{th} \text{ section of data } x}$$

where $x_k[n] = x[kL + n]$.

Overlap and Add:

With this approach, the convolution in the k^{th} section can be achieved by

$$\underbrace{y_k[n]}_{\substack{L+M-1 \\ \text{points}}} = \underbrace{h[n]}_{M \text{ points}} \circledast \underbrace{x_k[n]}_{L \text{ points}}, \quad n = 0, \ldots, L+M-2$$

$$\cdots\ y_k[0],\ y_k[1], \cdots\ y_k[L-1],\ \boxed{\begin{matrix} y_k[L] & \cdots & y_k[L+M-2] \\ y_{k+1}[0] & \cdots & y_{k+1}[M-2] \end{matrix}}$$

$$\cdots$$

overlap and add

As above, the last $M-1$ points of the sequence y_k are then added to the first M points of the sequence y_{k+1} such that

$$y[(k+1)L+n] = \begin{cases} y_k[L+n] + y_{k+1}[n], & \text{if } 0 \le n \le M-1 \\ y_{k+1}[n], & \text{if } M \le n \le L-1 \end{cases}$$

Example:

$$x[n] = \{\underset{\underset{n=0}{\uparrow}}{1}, 2, 1, 0, 3, 5, 7, 8, 1\}$$

$$h[n] = \{\underset{\underset{n=0}{\uparrow}}{1}, 1\}$$

Use the overlap and add to compute

$y[n] = x[n] \circledast h[n]$

Answer:

$N = 9, \quad M = 2, \quad$ choose $L = 3 > M$

$x[n] = \{ \underbrace{1, 2, 1}_{x_0[n]}, \underbrace{0, 3, 5}_{x_1[n]}, \underbrace{7, 8, 1}_{x_2[n]} \}$

$y_0[n] = h[n] \circledast x_0[n]$

$= \{ \underset{\uparrow \atop n=0}{1}, 3, 3, 1 \}$

$y_1[n] = h[n] \circledast x_1[n]$

$= \{ \underset{\uparrow \atop n=0}{0}, 3, 8, 5 \}$

$y_2[n] = h[n] \circledast x_2[n]$

$= \{ \underset{\uparrow \atop n=0}{7}, 15, 9, 1 \}$

$$
\begin{array}{l}
1, 3, 3, 1 \\
\qquad\quad\;\; 0, 3, 8, 5 \\
+ \qquad\qquad\qquad\;\; 7, 15, 9, 1 \\
\hline
\end{array}
$$

$y[n] = \{ \underset{\uparrow \atop n=0}{1}, 3, 3, 1, 3, 8, 12, 15, 9, 1 \}$

Example: $x[n] = \begin{cases} (\frac{1}{2})^n, & 0 \le n \le 1000000 \\ 0, & \text{otherwise} \end{cases}$

$$h[n] = \{\underset{\underset{n=0}{\uparrow}}{1}, 8, 7, 2, 4, 6, 10\}$$

Try to apply overlap and add to compute $y[n] = x[n] \circledast h[n]$ by computer.

Solution: Practice by yourself.

<u>Overlap and save:</u>

In the k^{th} section, the output y can be given by

$$y[kL+n] = \sum_{l=0}^{M-1} h[l]\, x[kL+n-l]$$

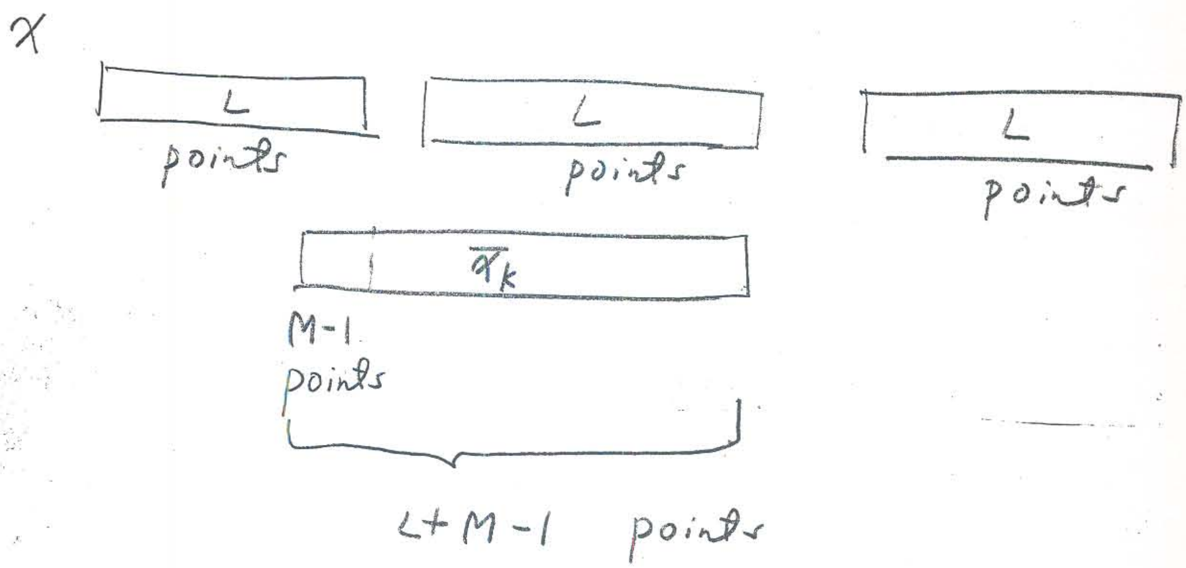
$$\text{for } n = 0, 1, \cdots, L-1$$

Therefore, we can save the last $M-1$ values in the previous block to complete this convolution, such that

$$\bar{x}_k = \{ x[kL-M+1], x[kL-M+2], \cdots x[kL-1],$$
$$x[kL], x[kL+1], \cdots x[kL+L-1] \}$$

$$\bar{y}_k[n] = \sum_{l=0}^{M-1} h[l]\, x[n+kL-l], \quad \text{for } 0 \le n \le L-1$$

$$y[kL+n] = \bar{y}_k[n], \quad \begin{cases} n = 0, 1, \cdots, L-1 \\ k = 0, 1, \cdots \end{cases}$$



Example: $x[n] = \{ \underset{n=0}{\underset{\uparrow}{1}}, 2, 1, 0, 3, 5, 7, 8, 1 \}$

$h[n] = \{ \underset{n=0}{\underset{\uparrow}{1}}, 1 \}$, $L = 3$, $M = 2$

$\bar{x}_0 = \{0, 1, 2, 1\}$ $\quad \bar{x}_3 = \{1, 0, 0, 0\}$

$\bar{x}_1 = \{1, 0, 3, 5\}$

$\bar{x}_2 = \{5, 7, 8, 1\}$

$\bar{y}_0[n] = \{1, 3, 3\}$

$\bar{y}_1[n] = \{1, 3, 8\}$

$\bar{y}_2[n] = \{12, 15, 9\}$

$\bar{y}_3[n] = \{1, 0, 0\}$

$$\therefore\ y[n] = \{\underset{\substack{\uparrow \\ n=0}}{1}, 3, 3, 1, 3, 8, 12, 15, 9, 1\}$$

3.7 DFT and DCT

For a finite-support continuous-time signal $x(t)$, $0 \le t \le T_0$, it can be represented as the Fourier Series,

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j2\pi F_0 t}$$

where $F_0 = \frac{1}{T_0}$ and $a_k = \frac{1}{T_0}\int_0^{T_0} x(t) e^{-j2\pi F_0 t}\, dt$

If $\frac{1}{T_0}\int_0^{T_0} |x(t)|^2 dt = \sum_{k=-\infty}^{\infty} |a_k|^2 < \infty$,

then $\lim_{k \to \pm\infty} a_k = 0$

Therefore, we can have

1. A continuous time signal $x(t)$ can be written in terms of a discrete sequence a_k.
2. A continuous-time signal with finite energy can be approximated arbitrarily closely by a discrete, finite sequence, that is

$$x(t) \cong \hat{x}_N(t) = \sum_{k=-N}^{N} a_k \, e^{j2\pi F_0 t} \; ; \quad F_0 \equiv \frac{1}{T_0}$$

However, if $x(t)$ has several discontinuities the error $\lim_{N \to \infty} x(t) - \hat{x}_N(t)$ at the discontinuities will never decay to zero. (Please refer to Figure 3.25) This means that no matter how many coefficients we retain, the approximation $\hat{x}_N(t)$ has always a large error at the two extremes of the window interval for $x(t)$.

To resolve this problem caused by window discontinuities, we can fold the window of signal $x(t)$ first to make an even signal $x_E(t)$ such that

$$x_E(t) \equiv \begin{cases} x(-t), & \text{if } -T_0 \le t \le 0 \\ x(t), & \text{if } 0 \le t \le T_0 , \end{cases}$$

where $x_E(t) = x_E(-t)$.

The Fourier series of $x_E(t)$ can be given by

$$x_E(t) = \sum_{k=-\infty}^{\infty} a_k \, e^{j\left(\frac{2\pi k t}{2T_0}\right)} dt$$

where

$$a_k = \frac{1}{2T_0} \int_{-T_0}^{T_0} x_E(t) \, e^{-j\frac{\pi k}{T_0} t} dt$$

$$= \frac{1}{2T_0} \int_{-T_0}^{T_0} x_E(-t) \, e^{-j\frac{\pi k}{T_0} t} dt$$

$$t' = -t$$

$$= \frac{1}{2T_0} \int_{T_0}^{-T_0} x_E(t') \, e^{j\frac{\pi k}{T_0} t'} (-dt')$$

$$= \frac{1}{2T_0} \int_{-T_0}^{T_0} x_E(t') \, e^{-j\frac{\pi(-k)}{T_0} t'} dt'$$

$$= a_{-k}$$

$$\therefore \quad x_E(t) = \sum_{k=-\infty}^{-1} a_k e^{j\frac{\pi k}{T_0}t} + \sum_{k=0}^{\infty} a_k e^{j\frac{\pi k}{T_0}t}$$

$$= \sum_{k=1}^{\infty} a_k e^{-j\frac{\pi k}{T_0}t} + \sum_{k=0}^{\infty} a_k e^{j\frac{\pi k}{T_0}t}$$

$$= a_0 + \sum_{k=1}^{\infty} 2a_k \cos\left(\frac{\pi k}{T_0}t\right),$$

where $a_0 = \frac{1}{2T_0}\int_{-T_0}^{T_0} x_E(t)\,dt$

$$= \frac{1}{T_0}\int_0^{T_0} x(t)\,dt$$

$$2a_k = a_k + a_{-k}$$

$$= \frac{1}{2T_0}\int_{-T_0}^{T_0} x_E(t)\left[e^{-j\frac{\pi k}{T_0}t} + e^{j\frac{\pi k}{T_0}t}\right]dt$$

$$= \frac{1}{2T_0}\int_{-T_0}^{T_0} \underbrace{x_E(t)}_{\text{even}}\, 2\underbrace{\cos\left(\frac{\pi k}{T_0}t\right)}_{\text{even}}\,dt$$

$$= \frac{2}{T_0}\int_0^{T_0} x(t)\cos\left(\frac{\pi k}{T_0}t\right)dt,$$

$$\forall k \geq 1$$

Here, we can define the discrete cosine transform (DCT) as follows: (DCT-II)

$$X^{II}[k] = \text{DCT II}\{x[n]\}$$
$$= \sqrt{\frac{2}{N}}\, C[k] \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi k(2n+1)}{2N}\right],$$
$$k = 0, \cdots, N-1$$

$$x[n] = \text{IDCT II}\{X^{II}[k]\}$$
$$= \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C[k]\, X^{II}[k] \cos\left[\frac{\pi k(2n+1)}{2N}\right],$$
$$n = 0, 1, \cdots, N-1$$

where $C[k] = \begin{cases} 1, & \text{if } k \neq 0 \\ 1/\sqrt{2}, & \text{if } k = 0 \end{cases}$

The DCT basis signals can be defined as

$$C_k[n] \equiv \cos\left[\frac{\pi k(2n+1)}{2N}\right], \quad n = 0, \cdots, N-1$$

It can easily verified that, the basis signals satisfy the orthogonality conditions

$$\sum_{n=0}^{N-1} C_l[n]\, C_m[n] = 0 \quad \text{if } l \neq m$$

The DCT will have a more "energy-compacted" spectrum than that of the DFT. Hence the DCT is widely applied for multi-media coding techniques such as MPEG-series CODEC.

3.8 Fast Fourier Transform (FFT)

We can see in this section that for a data set of length N, the complexity of the DFT (the number of additions and multiplications required) grows as N^2 while the FFT as $N \log_2 N$.

Recall the expression for the N-point DFT of a sequence $x[n]$, $n = 0, 1, \cdots, N-1$ as

$$X_N[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k = 0, 1, \cdots, N-1$$

where $W_N = e^{-j\frac{2\pi}{N}}$

$$X_N[k] = \sum_{\substack{n \text{ is} \\ \text{even}}} x[n]\, W_N^{kn} + \sum_{\substack{n \text{ is} \\ \text{odd}}} x[n]\, W_N^{kn}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} x[2m]\, W_N^{2mk} + W_N^{k} \cdot \sum_{m=0}^{\frac{N}{2}-1} x[2m+1]\, W_N^{2mk}$$

We notice two facts:

① $W_N^2 = e^{-j 2\pi \frac{2}{N}} = W_{N/2}$

② $X_N[k] = \text{DFT}\{x[0], x[2], \ldots, x[N-2]\}$

$\quad + W_N^{k}\, \text{DFT}\{x[1], x[3], \ldots, x[N-1]\}$

Therefore, $X_N[k] = X_{\frac{N}{2}}^{\text{even}}[k] + W_N^{k}\, X_{\frac{N}{2}}^{\text{odd}}[k]$

where

$$X_{\frac{N}{2}}^{\text{even}}[k] = \text{DFT}\{x[0], x[2], \ldots, x[N-2]\}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} x[2m]\, W_{\frac{N}{2}}^{km}$$

$$X_{\frac{N}{2}}^{\text{odd}}[k] = \text{DFT}\{x[1], x[3], \ldots, x[N-1]\}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} x[2m+1]\, W_{\frac{N}{2}}^{km}$$

For example, when $N=4$, we need to compute $X_4[0]$, $X_4[1]$, $X_4[2]$, $X_4[3]$ on the basis of $X_2^{even}[0]$, $X_2^{even}[1]$ and $X_2^{odd}[0]$, $X_2^{odd}[1]$, such that

$$X_4[k] = X_2^{even}[k] + W_4^k X_2^{odd}[k], \quad k=0,1$$

$$X_4[k+2] = X_2^{even}[k] + W_4^{k+2} X_2^{odd}[k]$$

$$= X_2^{even}[k] - W_4^k X_2^{odd}[k], \quad k=0,1$$

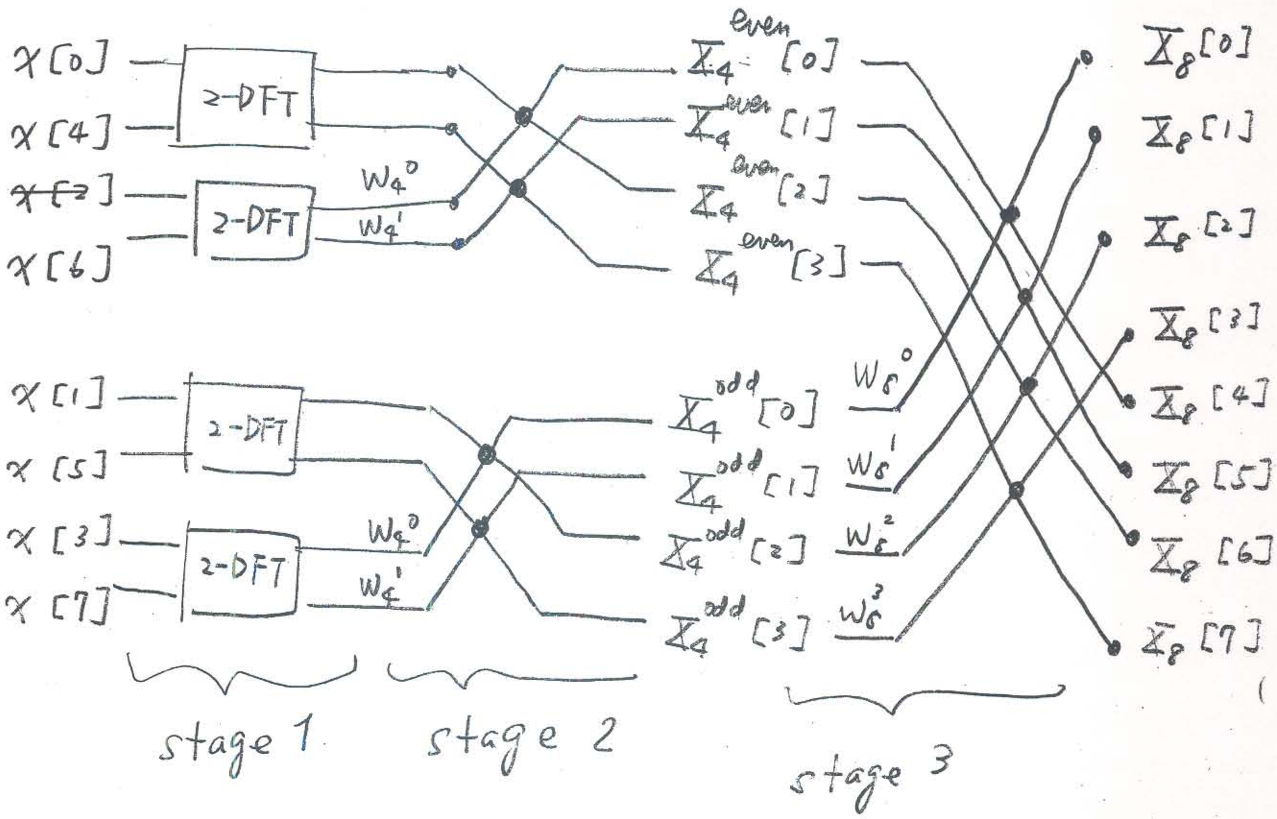
Thus, in general, the DFT of any sequence of length N can be given by

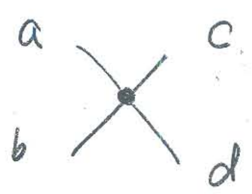
$$X_N[k] = X_{\frac{N}{2}}^{even}[k] + W_N^k X_{\frac{N}{2}}^{odd}[k],$$

$$X_N[k+\tfrac{N}{2}] = X_{\frac{N}{2}}^{even}[k] - W_N^k X_{\frac{N}{2}}^{odd}[k],$$

for $k = 0, 1, \ldots \frac{N}{2}-1$

Example : Decomposition of an 8-point DFT for FFT





$c = a + b$

$d = a - b$

Total 3 stages of FFT decomposition (butterfly)

Look at the previous example. The number of operations required by the FFT when the data length $N = 2^L$ is a power of 2. We can see that there are $L = \log_2 N$ stateges of butterfly decomposition and each stage has the same number of additions and multiplications: N additions and $\frac{N}{2}$ complex multiplications per stage. Therefore, the radix-2 FFT of a sequence of length $N = 2^L$ has a total of

$$N \log_2 N \text{ additions and } \frac{N}{2} \log_2 N \text{ multiplications}$$

We say that this FFT algorithm has a complexity of order $O(N \log_2 N)$ while the DFT algorithm has a complexity of order $O(N^2)$, which can be easily verified from its definition.