Lecture 13. *Tuesday 10th November, 2020.*

0.1 DFT of real sequences

DFT of real sequences

In most practical applications, sequences of interest are real. In such cases, the symmetry properties of the DFT can be exploited to make the DFT computations more efficient.
Let $g[n]$ and $h[n]$ be two length-N real sequences with $G[k]$ and $H[k]$ denoting their respective N-point DFTs. These two N-point DFTs can be computed efficiently using a single N-point DFT. Now, define a complex length-N sequence:

$$x[n] = g[n] + jh[n] \tag{1}$$

Hence, $g[n] = \mathrm{Re}\{x[n]\}$ and $h[n] = \mathrm{Im}\{x[n]\}$. Let $X[k]$ denote the N-point DFT of $x[n]$. Then, we arrive at:

$$G[k] = \frac{1}{2}\{X[k] + X^*[\langle -k\rangle_N]\} \tag{2}$$

$$H[k] = \frac{1}{2j}\{X[k] - X^*[\langle -k\rangle_N]\} \tag{3}$$

Note that for $0 \leq k \leq N-1$:

$$X^*[\langle -k\rangle_N] = X^*[\langle N-k\rangle_N] \tag{4}$$

Example 1: DFT of real sequences

We compute the 4-point DFTs of the two real sequences $g[n]$ and $h[n]$:

$$g[n] = \{1, 2, 0, 1\} \tag{5}$$

$$h[n] = \{2, 2, 1, 1\} \tag{6}$$

Then $x[n] = g[n] + jh[n]$ is given by:

$$x[n] = \{1 + j2, 2 + j2, 1 + j\} \tag{7}$$

Its DFT $X[k]$ is:

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -1 \end{bmatrix} \begin{bmatrix} 1+j2 \\ 2+j2 \\ j \\ 1+j \end{bmatrix} = \begin{bmatrix} 4+j6 \\ 2 \\ -2 \\ j2 \end{bmatrix} \tag{8}$$

From the above:

$$X^*[k] = \{4 - j6, 2, -2, -j2\} \tag{9}$$

Hence:

$$X^*[\langle 4-k\rangle_4] = \{4 - j6, -j2, -2, 2\} \tag{10}$$

Therefore:

$$G[k] = \{4, 1 - j, -2, 1 + j\} \tag{11}$$

$$H[k] = \{6, 1 - j, 0, 1 + j\} \tag{12}$$

verifying the results derived in Lecture 12.

Now, let $v[n]$ be a length-$2N$ real sequence with a $2N$-point DFT $V[k]$. Then, we define two length-N real sequences $g[n]$ and $h[n]$ as follows. Let $G[k]$ and $H[k]$ denote their respective N-point DFTs:

$$\begin{cases} g[n] = v[2n] \\ h[n] = v[2n+1] \end{cases} \qquad 0 \leq n \leq N \tag{13}$$

We define a length-N complex sequence $x[n] = g[n] + jh[n]$ with an N-point DFT $X[k]$. Then, as showed earlier:

$$G[k] = \frac{1}{2}\{X[k] + X^*[\langle -k \rangle_N]\} \tag{14}$$

$$H[k] = \frac{1}{2j}\{X[k] - X^*[\langle -k \rangle_N]\} \tag{15}$$

Now, for $0 \leq k \leq 2N-1$:

$$\begin{aligned} V[k] &= \sum_{n=0}^{2N-1} v[n] W_{2N}^{nk} \\ &= \sum_{n=0}^{N-1} v[2n] W_{2N}^{2nk} + \sum_{n=0}^{N-1} v[2n+1] W_{2N}^{(2n+1)k} \\ &= \sum_{n=0}^{N-1} g[n] W_{N}^{nk} + \sum_{n=0}^{N-1} W_{N}^{nk} W_{2N}^{k} \\ &= \sum_{n=0}^{N-1} g[n] W_{N}^{nk} + W_{2N}^{k} \sum_{n=0}^{N-1} h[n] W_{N}^{nk} \\ &= G[\langle k \rangle_N] + W_{2N}^{k} H[\langle k \rangle_N] \end{aligned} \tag{16}$$

Example 2: DFT of real sequences

Let us determine the 8-point DFT $V[k]$ of the length-8 real sequence:

$$v[n] = \{1, 2, 2, 2, 0, 1, 1, 1\} \tag{17}$$

We form two length-4 real sequences as follows:

$$g[n] = v[2n] = \{1, 2, 0, 1\} \tag{18}$$

$$h[n] = v[2n+1] = \{2, 2, 1, 1\} \tag{19}$$

Now:

$$V[k] = G[\langle k \rangle_4] + W_8^k H[\langle k \rangle_4] \qquad 0 \leq k \leq 7 \tag{20}$$

Substituting the values of of the 4-point DFTs $G[k]$ and $H[k]$ computed earlier, we get:

$$V[0] = G[0] + H[0] = 4 + 6 = 10 \tag{21}$$

$$V[1] = G[1] + W_1^0 H[1] = (1 - j) + e^{-j\frac{\pi}{4}}(1 - j) = 1 - j2.4142 \tag{22}$$

$$V[2] = G[2] + W_2^0 H[2] = -2 + e^{-j\frac{3\pi}{4}} \cdot 0 = -2 \tag{23}$$

$$V[3] = G[3] + W_3^0 H[3] = (1 + j) + e^{-j\frac{3\pi}{4}}(1 + j) = 1 - j0.4142 \tag{24}$$

$$V[4] = G[0] + W_4^0 H[0] = 4 + e^{-j\pi} \cdot 6 = -2 \tag{25}$$

$$V[5] = G[1] + W_5^0 H[1] = (1 - j) + e^{-j\frac{5\pi}{4}}(1 - j) = 1 + j0.4142 \tag{26}$$

$$V[6] = G[2] + W_6^0 H[2] = -2 + e^{-j\frac{3\pi}{2}} \cdot 0 = -2 \tag{27}$$

$$V[7] = G[3] + W_7^0 H[3] = (1 + j) + e^{-j\frac{7\pi}{4}}(1 + j) = 1 + j2.4142 \tag{28}$$

0.2 Linear convolution using the DFT

Linear convolution using the DFT

Linear convolution is a key operation in many signal processing applications. Since a DFT can be efficiently implemented using FFT algorithms, it is of particular interest to develop methods for the implementation of linear convolution using the DFT.
For the purpose, let $g[n]$ and $h[n]$ be two finite-length sequences of length N and M, respectively. Moreover, we denote with $L = N + M + 1$. Then, we define two length-L sequences:

$$g_e[n] = \begin{cases} g[n] & 0 \le n \le N - 1 \\ 0 & N \le n \le L - 1 \end{cases} \tag{29}$$

$$h_e[n] = \begin{cases} h[n] & 0 \le n \le M - 1 \\ 0 & M \le n \le L - 1 \end{cases} \tag{30}$$

Then:

$$y_L[n] = g[n] \circledast h[n] = y_C[n] = g_e[n] \textcircled{\scriptsize L} h_e[n] \tag{31}$$

The corresponding implementation scheme is illustrated in Figure 1.

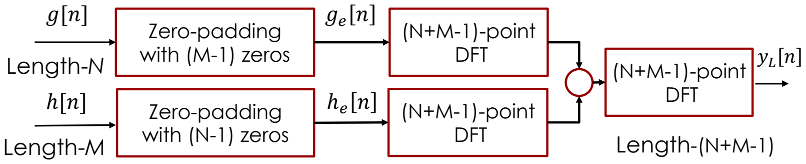

Figure 1: Scheme of linear convolution of two finite-length sequences.

We next consider the **DFT-based implementation** of:

$$y[n] = \sum_{\ell=0}^{M-1} h[\ell]x[n - \ell] = h[n] \circledast x[n] \tag{32}$$

where $h[n]$ is a finite-length sequence of length M and $x[n]$ is an infinite length (or a finite length sequence of length much greater that M). We first segment $x[n]$, assumed to be a causal sequence here without any loss of generality, into a set of contiguous finite-length subsequences $x_m[n]$ of length N each:

$$x[n] = \sum_{m=0}^{\infty} x_m[n - mN] \tag{33}$$

where:

$$x_m[n] = \begin{cases} x[n+mN] & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

Thus, we can write:

$$y[n] = h[n] \circledast x[n] = \sum_{m=0}^{\infty} y_m[n - mN] \tag{35}$$

where:

$$y_m = h[n] \circledast x_m[n] \tag{36}$$

Since $h[n]$ is of length M and $x_m[n]$ is of length N, the linear convolution $h[n] \circledast x_m[n]$ is of length $N + M - 1$.

As a result, the desired linear convolution $y[n] = h[n] \circledast x[n]$ has been broken up into a **sum of infinite number of short-length linear convolutions** of length $N + M - 1$ each:

$$y_m[n] = h[n] \circledast x_m[n] \tag{37}$$

Each of these short convolutions can be implemented using the DFT-based method discussed earlier, where now the DFTs (and the inverse DFT) are computed on the basis of $N + M - 1$ points.

Overlapping between short convolutions using the DFT-based approach

There is one more subtlety to take care of before we can implement: the calculation in Eq. 35 using the DFT-based approach. Let us explain it proceding in order:

- the first convolution in Eq. 35, namely $y_0 = h[n] \circledast x_0[n]$, is of length $N+M-1$ and it is defined for $0 \leq n \leq N + M - 2$;
- the second short convolution, namely $y_1[n] = h[n] \circledast x_1[n]$, is also of length $N + M - 1$ but it is defined for $N \leq n \leq 2N + M - 2$;
- likewise, the third short convolution, namely $y_2[n] = h[n] \circledast x_2[n]$, is also of length $N + M - 1$ but it is defined for $2N \leq b \leq 3N + M - 2$, and so on and so forth;
- in these particular cases, we can observe that there is an overlap of $M - 1$ samples between $h[n] \circledast x_1[n]$ and $h[n] \circledast x_2[n]$.

In general, there will be an overlap of $M - 1$ samples between the samples of the short convolutions $h[n] \circledast x_{r-1}[n]$ and $h[n] \circledast x_r[n]$ for $(r-1)N \leq n \leq rN + M - 1$. This process is illustrated in Figures 2 and 3 for $M = 5$ and $N = 7$. In these cases, $y[n]$ obtained by a linear convolution of $x[n]$ and $h[n]$ is given by:

$$y[n] = y_0[n] \qquad 0 \leq n \leq 6 \tag{38}$$

$$y[n] = y_0[n] + y_1[n-7] \qquad 7 \leq n \leq 10 \tag{39}$$

$$y[n] = y_1[n-7] \qquad 11 \leq n \leq 13 \tag{40}$$

$$y[n] = y_1[n-7] + y_2[n-14] \qquad 14 \leq n \leq 17 \tag{41}$$

$$y[n] = y_2[n-14] \qquad 18 \leq n \leq 20 \tag{42}$$

$$\vdots \qquad\qquad \vdots$$

Overlap-add method

The above procedure is called the **overlap-add method** since the results of the short linear convolutions overlap and the overlapped portions are added to get the correct final result.

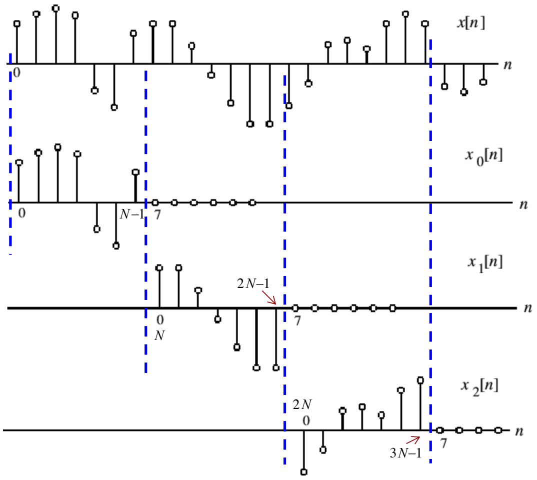

Figure 2: Overlap-add method for $M = 5$ and $N = 7$.

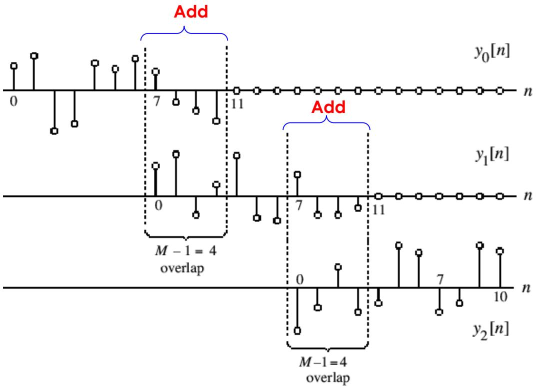

Figure 3: Overlap-add method for $M = 5$ and $N = 7$.

Implementation using circular convolution

In implementing the overlap-add method using the DFT, we need to compute two $(N + M - 1)$-point DFTs and one $(N + M - 1)$-point IDFT since the overall linear convolution was expressed as a sum of short-length linear convolutions of length $N + M - 1$ each. It is possible to implement the overall linear convolution by performing instead circular convolution of length shorter than $N + M - 1$. To this end, it is necessary to segment $x[n]$ into overlapping blocks $x_m[n]$, keep the terms of the circular convolution of $h[n]$ with $x_m[n]$ that corresponds to the terms obtained by a linear convolution of $h[n]$ and $x_m[n]$, and throw away the other parts of the circular convolution.

To understand the correspondence between the linear and circular convolutions, consider a length-4 sequence $x[n]$ and a length-3 sequence $h[n]$. Let $y_L[n]$ denote the result of a linear convolution of $x[n]$ with $h[n]$. The six samples of $y_L[n]$ are given by:

$$y_L[0] = h[0]x[0] \tag{43}$$

$$y_L[1] = h[0]x[1] + h[1]x[0] \tag{44}$$

$$y_L[2] = h[0]x[2] + h[1]x[1] + h[2]x[0] \tag{45}$$

$$y_L[3] = h[0]x[3] + h[1]x[2] + h[2]x[1] \tag{46}$$

$$y_L[4] = h[1]x[3] + h[2]x[2] \tag{47}$$

$$y_L[5] = h[2]x[3] \tag{48}$$

If we append $h[n]$ with a single zero-valued sample and convert it into a length-4 sequence $h_e[n]$, the 4-point circular convolution $y_C[n]$ of $h_e[n]$ and $x[n]$ is given by:

$$\begin{aligned} y_C[0] &= h[0]x[0] + h[1]x[3] + h[2]x[2] && (49)\\ y_C[1] &= h[0]x[1] + h[1]x[0] + h[2]x[3] && (50)\\ y_C[2] &= h[0]x[2] + h[1]x[1] + h[2]x[0] && (51)\\ y_C[3] &= h[0]x[3] + h[1]x[2] + h[2]x[1] && (52) \end{aligned}$$

If we compare the expressions for the samples of $y_L[n]$ with the samples of $y_C[n]$, we observe that the first 2 terms of $y_C[n]$ do not correspond to the first 2 terms of $y_L[n]$, whereas the last 2 terms of $y_C[n]$ are precisely the same as the third and the forth terms of $y_L[n]$.

In general, if we consider the N-point circular convolution of a length-M sequence $h[n]$ with a length-N sequence $x[n]$ with $N > M$, the first $M - 1$ samples of the circular convolution are incorrect and are rejected. The remaining $N-M+1$ samples correspond to the correct samples of the linear convolution of $h[n]$ with $x[n]$.

Now we consider an infinitely long or very long sequence $x[n]$. We break it up as a collection of smaller length (length-4) overlapping sequences $x_m[n]$ as $x_m[n] = x[n+2m]$, with $0 \leq n \leq 3$, $0 \leq m \leq \infty$. Next, we form:

$$w_m[n] = h[n] \textcircled{4}\, x_m[n] \qquad (53)$$

or, equivalently

$$\begin{aligned} w_m[0] &= h[0]x_m[0] + h[1]x_m[3] + h[2]x_m[2] && (54)\\ w_m[1] &= h[0]x_m[1] + h[1]x_m[0] + h[2]x_m[3] && (55)\\ w_m[2] &= h[0]x_m[2] + h[1]x_m[1] + h[2]x_m[0] && (56)\\ w_m[3] &= h[0]x_m[3] + h[1]x_m[2] + h[2]x_m[1] && (57) \end{aligned}$$

Computing the above for $m = 0, 1, 2, 3, \ldots$, and substituting the values of $x_m[n]$, we arrive at:

$$\begin{aligned} w_0[0] &= h[0]x[0] + h[1]x[3] + h[2]x[2] && \leftarrow \text{Reject}\\ w_0[1] &= h[0]x[1] + h[1]x[0] + h[2]x[3] && \leftarrow \text{Reject}\\ w_0[2] &= h[0]x[2] + h[1]x[1] + h[2]x[0] = y[2] && \leftarrow \text{Save}\\ w_0[3] &= h[0]x[3] + h[1]x[2] + h[2]x[1] = y[3] && \leftarrow \text{Save}\\ w_1[0] &= h[0]x[2] + h[1]x[5] + h[2]x[4] && \leftarrow \text{Reject}\\ w_1[1] &= h[0]x[3] + h[1]x[2] + h[2]x[5] && \leftarrow \text{Reject}\\ w_1[2] &= h[0]x[4] + h[1]x[3] + h[2]x[2] = y[4] && \leftarrow \text{Save}\\ w_1[3] &= h[0]x[5] + h[1]x[4] + h[2]x[3] = y[5] && \leftarrow \text{Save}\\ w_2[0] &= h[0]x[4] + h[1]x[5] + h[2]x[6] && \leftarrow \text{Reject}\\ w_2[1] &= h[0]x[5] + h[1]x[4] + h[2]x[7] && \leftarrow \text{Reject}\\ w_2[2] &= h[0]x[6] + h[1]x[5] + h[2]x[4] = y[6] && \leftarrow \text{Save}\\ w_2[3] &= h[0]x[7] + h[1]x[6] + h[2]x[5] = y[7] && \leftarrow \text{Save} \end{aligned}$$

It should be noted that tho determine $y[0]$ and $y[1]$ we need to form $x_{-1}[n]$, setting $x_{-1}[0] = x_{-1}[1] = 0$, $x_{-1}[2] = x[0]$, $x_{-1}[3] = x[1]$, and compute:

$$w_{-1}[n] = h[n] \textcircled{4}\, x_{-1}[n] \qquad 0 \leq n \leq 3 \qquad (58)$$

then reject $w_{-1}[0]$ and $w_{-1}[1]$, and save $w_{-1}[2] = y[0]$ and $w_{-1}[3] = y[1]$.

General case using circular convolution

In general, let $h[n]$ be a length-N sequence. Let $x_m[n]$ denote the m^{th} section of an infinitely long sequence $x[n]$ of length N and defined by:

$$x_m[n] = x[n + m(N - m + 1)] \qquad 0 \leq n \leq N - 1 \tag{59}$$

with $M < N$. Let $w_m[n] = h[n] \textcircled{\scriptsize N} x_m[n]$. Then, we reject the first $M - 1$ samples of $w_m[n]$ and "about" the remaining $M - M + 1$ samples of $w_m[n]$ to form $y_L[n]$, namely the linear convolution of $h[n]$ and $x[n]$. If $y_m[n]$ denotes the saved portion of $w_m[n]$, i.e.:

$$y_m[n] = \begin{cases} 0 & 0 \leq n \leq M - 2 \\ w_m[n] & M - 1 \leq n \leq N - 2 \end{cases} \tag{60}$$

then:

$$y_L[n + m(N - M + 1)] = y_m[n] \qquad M - 1 \leq n \leq N - 1 \tag{61}$$

The approach is called **overlap-save method** since the input is segmented into overlapping sections and parts of the results of the circular convolutions are saved and abutted to determine the linear convolution result. The process is illustrated in Figure 4.

Overlap-save method

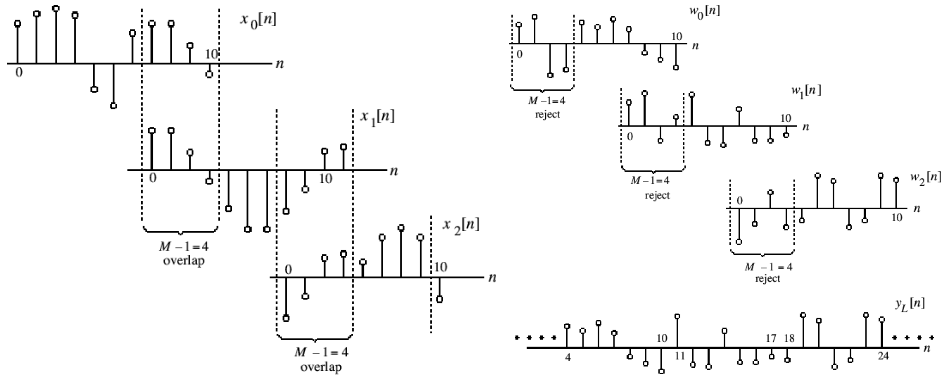

Figure 4: Illustration of the overlap-save method.