Computation of the DFT of Real Sequences

- In most practical applications, sequences of interest are real.
- In such cases, the symmetry properties of the DFT given in Table 3.7 can be exploited to make the DFT computations more efficient.

\[ \text{N-Point DFTs of Two Length-}\ N \text{ Real Sequences} \]

- 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.
- Define a complex length-\( N \) sequence

\[
\begin{align*}
x[n] &= g[n] + jh[n] \\
g[n] &= \text{Re}\{x[n]\} \\
h[n] &= \text{Im}\{x[n]\}
\end{align*}
\]

- Then is given by

\[
\begin{align*}
x[n] &= g[n] + jh[n] \\
g[n] &= \text{Re}\{x[n]\} \\
h[n] &= \text{Im}\{x[n]\}
\end{align*}
\]

\[
\begin{align*}
x[n] &= g[n] + jh[n] \\
g[n] &= \text{Re}\{x[n]\} \\
h[n] &= \text{Im}\{x[n]\}
\end{align*}
\]

- Its DFT \( X[k] \) is

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

- From the above

\[
X[k] = \begin{bmatrix}
4-j6 \\
2 \\
-2 \\
-j2
\end{bmatrix}
\]

- Therefore

\[
\begin{align*}
G[k] &= \begin{bmatrix}
4 & 1-j & -2 & 1+j
\end{bmatrix} \\
H[k] &= \begin{bmatrix}
6 & 1-j & 0 & 1+j
\end{bmatrix}
\end{align*}
\]

verifying the results derived earlier.
2N-Point DFT of a Real Sequence Using an N-point DFT

- Let \( v[n] \) be a length-2N real sequence with an 2N-point DFT \( V[k] \)
- Define two length-N real sequences \( g[n] \) and \( h[n] \) as follows:
  \[ g[n] = v[2n], \quad h[n] = v[2n+1], \quad 0 \leq n \leq N \]
- Let \( G[k] \) and \( H[k] \) denote their respective N-point DFTs

\[
\begin{align*}
\text{Let} \quad & v[n] = \{v[2n]\}, \quad 0 \leq n \leq 2N - 1 \\
\text{Define two length-N real sequences} \quad & g[n] = v[2n], \quad h[n] = v[2n+1], \quad 0 \leq n \leq N \\
\text{Let} \quad & G[k] \text{ and } H[k] \text{ denote their respective N-point DFTs}
\end{align*}
\]

2N-Point DFT of a Real Sequence Using an N-point DFT

- Define a length-N complex sequence \( x[n] = \{g[n] + jh[n]\} \) with an N-point DFT \( X[k] \)
- Then as shown earlier
  \[
  G[k] = \frac{1}{2} \{X[k] + X^*[(-k)_N]\}
  \]
  \[
  H[k] = \frac{1}{2} \{X[k] - X^*[(-k)_N]\}
  \]

- Now
  \[
  V[k] = \sum_{n=0}^{2N-1} v[n] W_{2N}^{nk} = \sum_{n=0}^{N-1} v[2n] W_{2N}^{2nk} + \sum_{n=0}^{N-1} v[2n+1] W_{2N}^{(2n+1)k}
  \]
  \[
  V[k] = \sum_{n=0}^{N-1} g[n] W_{N}^{nk} + \sum_{n=0}^{N-1} h[n] W_{N}^{nk} W_{2N}^{k} \quad 0 \leq k \leq 2N - 1
  \]

- We form two length-4 real sequences as follows

2N-Point DFT of a Real Sequence Using an N-point DFT

- Example - Let us determine the 8-point DFT \( V[k] \) of the length-8 real sequence
  \( \{v[n]\} = \{1 \quad 2 \quad 2 \quad 0 \quad 1 \quad 1 \quad 1\} \)

- Now
  \[
  V[k] = G[k] + W_{2N}^{k} H[k] \quad 0 \leq k \leq 2N - 1
  \]

- Substituting the values of the 4-point DFTs \( G[k] \) and \( H[k] \) computed earlier we get

\[
\begin{align*}
V[0] &= G[0] + H[0] = 4 + 6 = 10 \\
V[3] &= G[3] + W_{2N}^{k} H[3] = (1 + j) + e^{-j3\pi/4}(1 + j) = 1 - j0.4142 \\
V[4] &= G[0] + W_{2N}^{k} H[0] = 4 + e^{-j\pi} = -2
\end{align*}
\]
2N-Point DFT of a Real Sequence Using an N-point DFT

\[ \begin{align*}
&= (1 - j) + e^{-j \frac{\pi}{2}} (1 - j) = 1 + j0.4142 \\
&= (1 + j) + e^{-j \frac{3\pi}{2}} (1 + j) = 1 + j2.4142
\end{align*} \]

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 interest to develop methods for the implementation of linear convolution using the DFT.

Linear Convolution of Two Finite-Length Sequences

- Let \( g[n] \) and \( h[n] \) be two finite-length sequences of length \( N \) and \( M \), respectively.
- Denote \( L = N + M - 1 \).
- Define two length-\( L \) sequences
  \[ g_L[n] = \begin{cases} 
  g[n], & 0 \leq n \leq N - 1 \\
  0, & N \leq n \leq L - 1 
  \end{cases} \]
  \[ h_L[n] = \begin{cases} 
  h[n], & 0 \leq n \leq M - 1 \\
  0, & M \leq n \leq L - 1 
  \end{cases} \]

Linear Convolution of a Finite-Length Sequence with an Infinite-Length Sequence

- We next consider the DFT-based implementation of
  \[ y[n] = \sum_{\ell=0}^{M-1} h[\ell] x[n-\ell] = h[n] \oplus x[n] \]
  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 than \( M \)).

Overlap-Add Method

- 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:
  \[ x[n] = \sum_{m=0}^{N-1} x_m[n - mN] \]
  where
  \[ x_m[n] = \begin{cases} 
  x[n + mN], & 0 \leq n \leq N - 1 \\
  0, & \text{otherwise}
  \end{cases} \]
Overlap-Add Method

• Thus we can write

\[ y[n] = h[n] \ast x[n] = \sum_{m=0}^{\infty} y_m[n - mN] \]

where

\[ y_m[n] = h[n] \ast x_m[n] \]

• Since \( h[n] \) is of length \( M \) and \( x_m[n] \) is of length \( N \), the linear convolution \( h[n] \ast x_m[n] \) is of length \( N + M - 1 \)

• The desired linear convolution \( y[n] = h[n] \ast x[n] \) has been broken up into a sum of infinite number of short-length linear convolutions of length \( N + M - 1 \) each:

\[ y_m[n] = x_m[n] \ast h[n] \]

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

Overlap-Add Method

• There is one more subtlety to take care of before we can implement

\[ y[n] = \sum_{m=0}^{\infty} y_m[n - mN] \]

using the DFT-based approach

• Now the first convolution in the above sum, \( y_0[n] = h[n] \ast x_0[n] \), is of length \( N + M - 1 \) and is defined for \( 0 \leq n \leq N + M - 2 \)

• The second short convolution \( y_1[n] = h[n] \ast x_1[n] \), is also of length \( N + M - 1 \) but is defined for \( N \leq n \leq 2N + M - 2 \)

• There is an overlap of \( M - 1 \) samples between these two short linear convolutions

• Likewise, the third short convolution \( y_2[n] = h[n] \ast x_2[n] \), is also of length \( N + M - 1 \) but is defined for \( 0 \leq n \leq N + M - 2 \)

• Thus there is an overlap of \( M - 1 \) samples between \( h[n] \ast x[n] \) and \( h[n] \ast x_2[n] \)

• In general, there will be an overlap of \( M - 1 \) samples between the samples of the short convolutions \( h[n] \ast x_{p-1}[n] \) and \( h[n] \ast x_p[n] \) for

• This process is illustrated in the figure on the next slide for \( M = 5 \) and \( N = 7 \)
Overlap-Add Method

- Therefore, $y[n]$ obtained by a linear convolution of $x[n]$ and $h[n]$ is given by

\[
y[n] = y_0[n], \quad 0 \leq n \leq 6
\]
\[
y[n] = y_0[n] + y_1[n - 7], \quad 7 \leq n \leq 10
\]
\[
y[n] = y_1[n - 7], \quad 11 \leq n \leq 13
\]
\[
y[n] = y_1[n - 7] + y_2[n - 14], \quad 14 \leq n \leq 17
\]
\[
y[n] = y_2[n - 14], \quad 18 \leq n \leq 20
\]

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.
- The function `fftfilt` can be used to implement the above method.

Overlap-Save Method

- 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 $(N + M - 1)$.
Overlap-Save Method

- 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:

  \[
  \begin{align*}
  y_L[0] &= h[0]x[0] \\
  \end{align*}
  \]

Overlap-Save Method

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

  \[
  \begin{align*}
  \end{align*}
  \]

Overlap-Save Method

- General case: \( N \)-point circular convolution of a length-\( M \) sequence \( h[n] \) with a length-\( N \) sequence \( x[n] \) with \( N > M \).
- First \( M - 1 \) samples of the circular convolution are incorrect and are rejected.
- Remaining \( N - M + 1 \) samples correspond to the correct samples of the linear convolution of \( h[n] \) with \( x[n] \).

Overlap-Save Method

- 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 3rd and 4th terms of \( y_L[n] \), i.e.,

  \[
  \begin{align*}
  y_L[0] &= y_C[0], & y_L[1] &= y_C[1] \\
  \end{align*}
  \]

Overlap-Save Method

- Now, consider an infinitely long or very long sequence \( x[n] \).
- Break it up as a collection of smaller length (length-4) overlapping sequences \( x_m[n] \) as 

  \[
  x_m[n] = x[n + 2m], \quad 0 \leq n \leq 3, \quad 0 \leq m \leq \infty
  \]
- Next, form

  \[
  w_m[n] = h[n] \bigcirc x_m[n]
  \]
Overlap-Save Method

- Or, equivalently,

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

Overlap-Save Method

- General Case: 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
    \[ x_m[n] = x[n + m(N - m + 1)] \quad 0 \leq n \leq N - 1 \]
    with \( M < N \)

Overlap-Save Method

- It should be noted that to determine \( y[0] \) and \( y[1] \), we need to form \( x_{-1}[n] \):
  \[ x_{-1}[0] = 0 \]
  \[ x_{-1}[1] = 0 \]
  \[ x_{-1}[2] = x[0] \]
  \[ x_{-1}[3] = x[1] \]
  and compute \( w_{-1}[n] = h[n] \odot x_{-1}[n] \) for \( 0 \leq n \leq 3 \) reject \( w_{-1}[0] \) and \( w_{-1}[1] \), and save \( w_{-1}[2] = y[0] \) and \( w_{-1}[3] = y[1] \)

Overlap-Save Method

- Let \( w_m[n] = h[n] \odot x_m[n] \)
  - Then, we reject the first \( M - 1 \) samples of \( w_m[n] \) and “abut” the remaining \( N - M + 1 \) samples of \( w_m[n] \) to form \( y_L[n] \) the linear convolution of \( h[n] \) and \( x[n] \)
  - If \( y_m[n] \) denotes the saved portion of \( w_m[n] \), i.e.
    \[ y_m[n] = \begin{cases} 
      0, & 0 \leq n \leq M - 2 \\
      w_m[n], & M - 1 \leq n \leq N - 2 
    \end{cases} \]
Overlap-Save Method

- Then
  \[ y_{L}[n + m(N - M + 1)] = y_{m}[n], \quad M - 1 \leq n \leq N - 1 \]
- 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.

Process is illustrated next.