From "Digital Signal Processing: A Computer-Based Approach" by Sanjit Mitra, McGraw Hill, © 2001

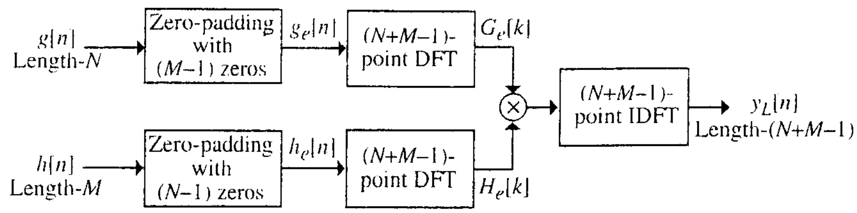

Figure 3.14: DFT-based implementation of the linear convolution of two finite-length sequences.

3.6 Linear Convolution Using the DFT

Linear convolution is a key operation in most signal processing applications. Since an N-point DFT can be implemented very efficiently using approximately $N(\log_2 N)$ arithmetic operations, it is of interest to investigate methods for the implementation of the linear convolution using the DFT. Earlier in Example 3.17, we have already illustrated for a very specific case how to implement a linear convolution using a circular convolution. We first generalize this example for the linear convolution of two finite-length sequences of unequal lengths. Later we consider the implementation of the linear convolution of a finite-length sequence with an infinite-length sequence.

3.6.1 Linear Convolution of Two Finite-Length Sequences

Let $g[n]$ and $h[n]$ be finite-length sequences of lengths N and M, respectively. Denote $L = M + N - 1$. Define two length-L sequences,

$$g_e[n] = \begin{cases} g[n], & 0 \le n \le N-1, \\ 0, & N \le n \le L-1, \end{cases} \tag{3.91}$$

$$h_e[n] = \begin{cases} h[n], & 0 \le n \le M-1, \\ 0, & M \le n \le L-1, \end{cases} \tag{3.92}$$

obtained by appending $g[n]$ and $h[n]$ with zero-valued samples. Then

$$y_L[n] = g[n] \circledast h[n] = y_C[n] = g_e[n] \textcircled{\scriptsize L} h_e[n]. \tag{3.93}$$

To implement Eq. (3.93) using the DFT, we first zero-pad $g[n]$ with $(M-1)$ zeros to obtain $g_e[n]$, and zero-pad $h[n]$ with $(N-1)$ zeros to obtain $h_e[n]$. Then we compute the $(N+M-1)$-point DFTs of $g_e[n]$ and $h_e[n]$, respectively, resulting in $G_e[k]$ and $H_e[k]$. An $(N+M-1)$-point IDFT of the product $G_e[k]H_e[k]$ results in $y_L[n]$. The process involved is sketched in Figure 3.14.

The following example uses MATLAB to illustrate the above approach.

EXAMPLE 3.20 Program 3_5 can be used to determine the linear convolution of two finite-length sequences via the DFT-based approach and to compare the result using a direct linear convolution. The input data to be entered in vector format inside square brackets are the two sequences to be convolved. The program plots the result of the linear convolution obtained using the DFT-based approach, and the difference between this result and the sequence obtained using a direct linear convolution using the M-file `conv`. Using this program we verify the result of Example 3.17 as demonstrated in Figure 3.15.

```
% Program 3_5
% Linear Convolution Via the DFT
```

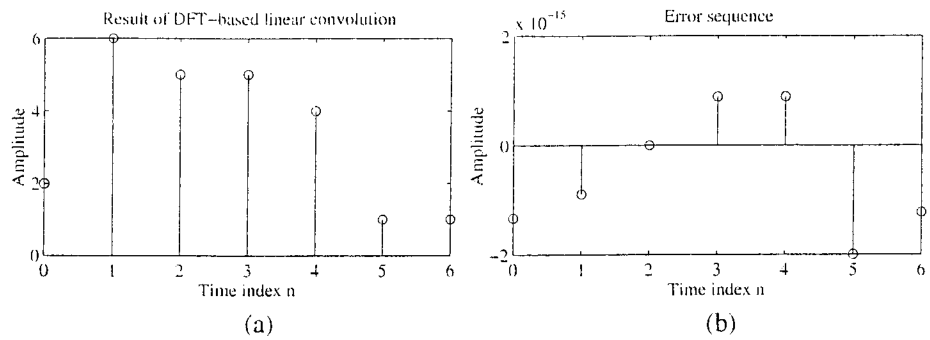

Figure 3.15: Plots of the output and the error sequences.

```
%
% Read in the two sequences
x = input('Type in the first sequence = ');
h = input('Type in the second sequence = ');
% Determine the length of the result of convolution
L = length(x)+length(h)-1;
% Compute the DFTs by zero-padding
XE = fft(x,L);
HE = fft(h,L);
% Determine the IDFT of the product
y1 = ifft(XE.*HE);
% Plot the sequence generated by DFT-based
% convolution and the error from direct linear
% convolution
k = 0:1:L-1;
subplot(2,1,1)
stem(k,y1)
xlabel('Time index n');ylabel('Amplitude');
title('Result of DFT-based linear convolution')
y2 = conv(x,h);
error = y1-y2;
subplot(2,1,2)
stem(k,abs(error))
xlabel('Time index n');ylabel('Amplitude')
title('Magnitude of error sequence')
```