Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.341: Discrete-Time Signal Processing

Fall 2005

Solutions for Problem Set 8

Issued: Tuesday, November 15, 2005.

Problem 8.1

OSB Problem 9.1

There are several possible answers to this problem. For one of them see the back of the book. Another is to use the program to compute $g[n]$, the DFT of $X[n]$, and use $x[n] = \frac{1}{N} g[((-n))_N]$ to determine $x[n]$.

Problem 8.2

OSB Problem 9.7

The figure corresponds to the flow graph of a second-order recursive system implementing Goertzel's algorithm. This system finds $X[k]$ for $k = 7$, which corresponds to a frequency of

$$\omega_k = \frac{14\pi}{32} = \frac{7\pi}{16}$$

Problem 8.3

OSB Problem 9.21

(a) Assume $x[n] = 0$, for $n < 0$ and $n > N - 1$. From the figure, we see that

$$y_k[n] = x[n] + W_N^k y_k[n-1]$$

Starting with $n = 0$, and iterating this recursive equation, we find

$$\begin{aligned}
y_k[0] &= x[0] \\
y_k[1] &= x[1] + W_N^k x[0] \\
y_k[2] &= x[2] + W_N^k x[1] + W_N^{2k} x[0] \\
&\vdots \\
y_k[N] &= x[N] + W_N^k x[N-1] + \cdots + W_N^{k(N-1)} x[1] + W_N^{kN} x[0] \\
&= 0 + \sum_{\ell=0}^{N-1} W_N^{k(N-\ell)} x[\ell] \\
&= \sum_{\ell=0}^{N-1} W_N^{-k\ell} x[\ell] \\
&= \sum_{\ell=0}^{N-1} x[\ell] W_N^{(N-k)\ell} \\
&= X[N-k]
\end{aligned}$$

(b) Using the figure, we find the system function $Y_k(z)$.

$$\begin{aligned}
Y_k(z) &= X(z) \frac{1 - W_N^{-k} z^{-1}}{1 - 2z^{-1}\cos(\frac{2\pi k}{N}) + z^{-2}} \\
&= X(z) \frac{1 - W_N^{-k} z^{-1}}{(1 - W_N^{-k} z^{-1})(1 - W_N^{k} z^{-1})} \\
&= \frac{X(z)}{1 - W_N^k z^{-1}}
\end{aligned}$$

Therefore, $y_k[n] = x[n] + W_N^k y_k[n-1]$. This is the same difference equation as was used in part (a).

Problem 8.4

OSB Problem 9.31

(a) Since $x[n]$ is real, $x[n] = x^*[n]$, and $X[k]$ is conjugate symmetric.

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x^*[n] e^{-j\frac{2\pi}{N}kn} \\ &= \left(\sum_{n=0}^{N-1} x[n] e^{j\frac{2\pi}{N}kn} e^{-j\frac{2\pi}{N}Nn} \right)^* \\ &= X^*[N-k] \end{aligned}$$

Hence, $X_R[k] = X_R[N-k]$ and $X_I[k] = -X_I[N-k]$.

(b) In Part (a) it was shown that the DFT of a real sequence $x[n]$ consists of a real part that has even symmetry, and an imaginary part that has odd symmetry. We use this fact in the DFT of the sequence $g[n]$ below.

$$\begin{aligned} G[k] &= X_1[k] + jX_2[k] \\ &= (X_{1ER}[k] + jX_{1OI}[k]) + j(X_{2ER}[k] + jX_{2OI}[k]) \\ &= \underbrace{X_{1ER}[k] - X_{2OI}[k]}_{\text{real part}} + j \underbrace{(X_{1OI}[k] + X_{2ER}[k])}_{\text{imaginary part}} \end{aligned}$$

In these expressions, the subscripts "E" and "O" denote even and odd symmetry, respectively, and the subscripts "R" and "I" denote real and imaginary parts, respectively.

Therefore, the even and real part of $G[k]$ is

$$G_{ER}[k] = X_{1ER}[k]$$

the odd and real part of $G[k]$ is

$$G_{OR}[k] = -X_{2OI}[k]$$

the even and imaginary part of $G[k]$ is

$$G_{EI}[k] = X_{2ER}[k]$$

and the odd and imaginary part of $G[k]$ is

$$G_{OI}[k] = X_{1OI}[k]$$

Having established these relationships, it is easy to come up with expressions for $\mathrm{X}_1[k]$ and $X_2[k]$.

$$\begin{aligned} X_1[k] &= X_{1ER}[k] + jX_{1OI}[k] \\ &= G_{ER}[k] + jG_{OI}[k] \\ X_2[k] &= X_{2ER}[k] + jX_{2OI}[k] \\ &= G_{EI}[k] - jG_{OR}[k] \end{aligned}$$

(c) An $N = 2^\nu$ point FFT requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions. This is equivalent to $2N\log_2 N$ real multiplications and $3N\log_2 N$ real additions.

(i) The two N-point FFTs, $X_1[k]$ and $X_2[k]$, require a total of $4N\log_2 N$ real multiplications and $6N\log_2 N$ real additions.

(ii) Computing the N-point FFT, $G[k]$, requires $2N\log_2 N$ real multiplications and $3N\log_2 N$ real additions. Then, the computation of $G_{ER}[k]$, $G_{EI}[k]$, $G_{OI}[k]$, and $G_{OR}[k]$ from $G[k]$ requires approximately $4N$ real multiplications and $4N$ real additions. Then, the formation of $X_1[k]$ and $X_2[k]$ from $G_{ER}[k]$, $G_{EI}[k]$, $G_{OI}[k]$, and $G_{OR}[k]$ requires no real additions or multiplications. So this technique requires a total of approximately $2N\log_2 N + 4N$ real multiplications and $3N\log_2 N + 4N$ real additions.

(d) Starting with

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

and separating $x[n]$ into its even and odd numbered parts, we get

$$X[k] = \sum_{n \text{ even}} x[n]e^{-j2\pi kn/N} + \sum_{n \text{ odd}} x[n]e^{-j2\pi kn/N}$$

Substituting $n = 2\ell$ for n even, and $n = 2\ell + 1$ for n odd, gives

$$\begin{aligned} X[k] &= \sum_{\ell=0}^{(N/2)-1} x[2\ell]e^{-j2\pi k\ell/(N/2)} + \sum_{\ell=0}^{(N/2)-1} x[2\ell+1]e^{-j2\pi k(2\ell+1)/N} \\ &= \sum_{\ell=0}^{(N/2)-1} x[2\ell]e^{-j2\pi k\ell/(N/2)} + e^{-j2\pi k/N} \sum_{\ell=0}^{(N/2)-1} x[2\ell+1]e^{-j2\pi k\ell/(N/2)} \\ &= \begin{cases} X_1[k] + e^{-j2\pi k/N}X_2[k], & 0 \le k < \frac{N}{2} \\ X_1[k-(N/2)] - e^{-j2\pi k/N}X_2[k-(N/2)], & \frac{N}{2} \le k < N \end{cases} \end{aligned}$$

(e) The algorithm is then

step 1: Form the sequence $g[n] = x[2n] + jx[2n+1]$, which has length $N/2$.

step 2: Compute $G[k]$, the $N/2$ point DFT of $g[n]$.

step 3: Separate $G[k]$ into the four parts, for $k = 1, \ldots, (N/2) - 1$

$$\begin{aligned} G_{OR}[k] &= \frac{1}{2}(G_R[k] - G_R[(N/2) - k]) \\ G_{ER}[k] &= \frac{1}{2}(G_R[k] + G_R[(N/2) - k]) \\ G_{OI}[k] &= \frac{1}{2}(G_I[k] - G_I[(N/2) - k]) \\ G_{EI}[k] &= \frac{1}{2}(G_I[k] + G_I[(N/2) - k]) \end{aligned}$$

which each have length $N/2$.

step 4: Form

$$\begin{aligned} X_1[k] &= G_{ER}[k] + jG_{OI}[k] \\ X_2{}'[k] &= e^{-j2\pi k/N}(G_{EI}[k] - jG_{OR}[k]) \end{aligned}$$

which each have length $N/2$.

step 5: Then, form

$$X[k] = X_1[k] + X_2{}'[k], \qquad 0 \le k < \frac{N}{2}$$

step 6: Finally, form

$$X[k] = X^*[N-k], \qquad \frac{N}{2} \le k < N$$

Adding up the computational requirements for each step of the algorithm gives (approximately)

step 1: 0 real multiplications and 0 real additions.

step 2: $2\frac{N}{2}\log_2\frac{N}{2}$ real multiplications and $3\frac{N}{2}\log_2\frac{N}{2}$ real additions.

step 3: $2N$ real multiplications and $2N$ real additions.

step 4: $2N$ real multiplications and N real additions.

step 5: 0 real multiplications and N real additions.

step 6: 0 real multiplications and 0 real additions.

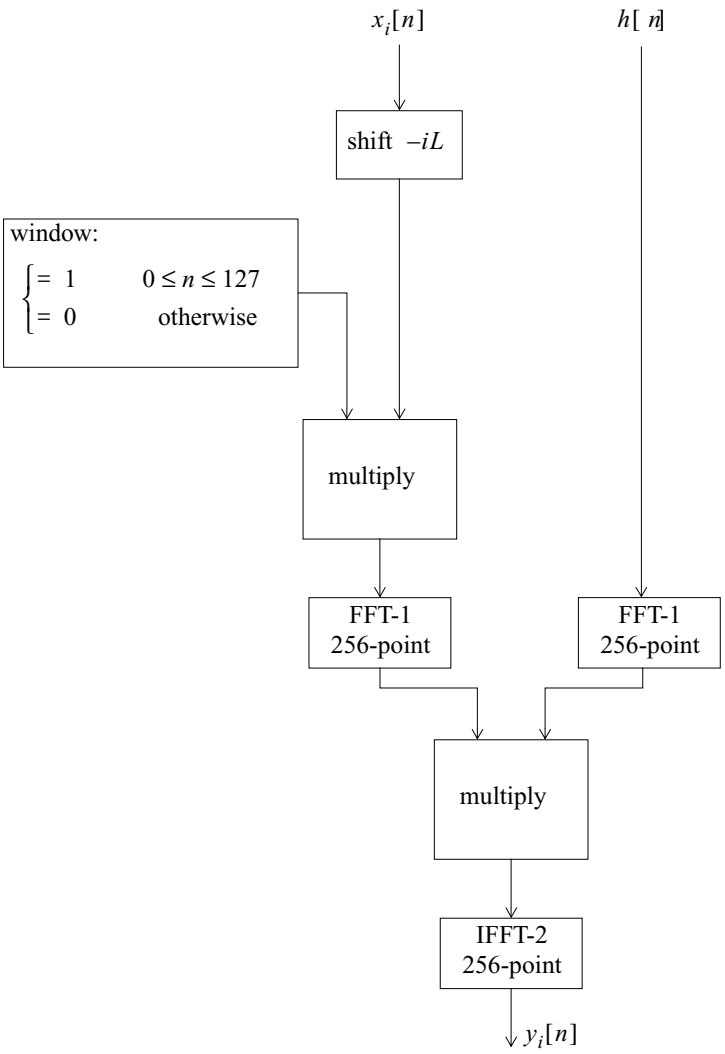
In total, approximately $N\log_2\frac{N}{2} + 4N$ real multiplications and $\frac{3}{2}N\log_2\frac{N}{2} + 4N$ real additions are required by this technique.

The number of real multiplications and real additions required if $X[k]$ is computed using one N-point FFT computation with the imaginary part set to zero is $2N\log_2 N$ real multiplications and $3N\log_2 N$ real additions.

Problem 8.5

OSB Problem 9.35

The blocks $x_i[n]$ are length 128, and the filter $h[n]$ is length 64. Therefore their linear convolution has length 191. Since the FFT and IFFT blocks are radix-2, all values N must be powers of 2, so we need to use 256-point FFTs and IFFTs. The block diagram is shown below.



We could also use FFT-2 blocks with IFFT-1 blocks. Whichever FFT blocks are used, the signals $x_i[n]$ and $h[n]$ need to be zero padded to length 256 before the FFTs.

Problem 8.6

OSB Problem 9.10

The answer is in the back of the book.

$$r[n] = e^{-j(2\pi/19)} W^{n^2/2}, \quad \text{where} \quad W = e^{-j(2\pi/10)}.$$