ECE4522

Exam No.3

Meysam Golmohammadi

Problem No. 1

(1.a)

Direct multiplication of two complex number is:

$(a + jb)(c + jd) = ac + jbc + jad - bd = (ac - bd) + j(bc + ad)$

As we can see it contains four real multiplications and two real additions. Now we define three new variables:

$k1 = c \cdot (a + b)$

$k2 = a \cdot (d - c)$

$k3 = b \cdot (c + d)$

Hence, we can conclude that

$ac - bd = k1 - k3$, $bc + ad = k1 + k2$

Using these three new variables we have:

$(a + jb)(c + jd) =(k1- k3) + j(k1+k2)$

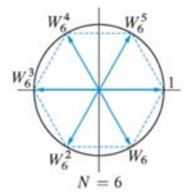
So by properly arranging the terms the multiplication needs just three real multiplication and five real addition.

(1.b)

A Fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. The DFT definition is:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn} => X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn} \qquad where \qquad W_N^{kn} = e^{-j\frac{2\pi}{N}kn}$$

Computing the DFT of N points in the direct way, takes $O(N^2)$ arithmetical operations. This means the computations is increasing with power of 2 of N and when we want to calculate DFT of a large data we need a huge amount of calculations. But a FFT can compute the same DFT in only O(N log2 N) operations. We know that W_N^{kn} are samples on the unit circle. For example:



<u>FFT has been developed on two important symmetry properties</u> which are:

Complex conjugate symmetry (Symmetry about the imaginary axis) : $W_N^{k+N/2} = -W_N^k$

Periodicity: $W_N^{k+N} = W_N^k$

This symmetry allows the number of computations for a DFT to be reduced significantly. Using periodicity and complex conjugate symmetry properties of W_N^k, a family of O(Nlog2N) algorithms, known collectively as Fast Fourier Transform (FFT) algorithms have been developed. These algorithms use a “divide-and-conquer” approach, that is, they decompose the DFT of a sequence of length N into smaller-length DFTs that are “merged” to form the N-point DFT. This procedure may be applied again to the smaller DFTs. Algorithms that decompose the sequence x[n] into smaller sequences are known as decimation-in-time FFTs; algorithms that decompose the DFT X[k] into smaller sequences are known as decimation-in-frequency FFTs. Under the constraint of $N=2^m$ the Radix-2 Fast Fourier Transform has been developed. <u>The fact that FFT is faster can be explained based on the heart of the algorithm: Divide and Conquer.</u> So rather than working with big size Signals, we divide our signal into smaller ones, and perform DFT of these smaller signals. At the end we add all the smaller DFT to get actual DFT of the big signal.

Problem No. 2:

(2.a)

$$w[n] = \left[0.5 - 0.5cos\left(\frac{2\pi n}{M}\right)\right] w_R[n] \qquad =>$$

$$w[n] = [0.5 - 0.5\frac{e^{j\frac{2\pi n}{M}} + e^{-j\frac{2\pi n}{M}}}{2}]w_R[n] \qquad =>$$

$$w[n] = \frac{1}{2}w_R[n] - \frac{1}{4}e^{j\frac{2\pi n}{M}}w_R[n] - \frac{1}{4}e^{-j\frac{2\pi n}{M}}w_R[n] \qquad (1.1)$$

According to the property of frequency shifting we have:

$$x[n]e^{j\omega_0 n} \overset{DFT}{\Longleftrightarrow} X\left[e^{j(\omega-\omega_0)}\right] \qquad (1.2)$$

Using (1.2) and getting DFT of (1.1) we have:

$$W\left[e^{j\omega}\right] = \frac{1}{2}W_R\left[e^{j\omega}\right] - \frac{1}{4}W_R\left[e^{j(\omega-\frac{2\pi}{M})}\right] - \frac{1}{4}W_R\left[e^{e^{j(\omega+\frac{2\pi}{M})}}\right] \qquad (1.1)$$

(2.b)

The second and third terms widen the mainlobe of Hann window and the sidelobes are lowed by the scaling factor.

(2.c)

In ideal lowpass filter the width of the main lobe is inversely proportional to the bandwidth. In this case, the predominant effect of the mainlobe is to smear or spread the original spectrum. The result is loss of resolution. A “good” window should have a narrow mainlobe (to minimize spectral spreading) and “low” sidelobes (to minimize spectral leakage). Unfortunately, it is impossible to satisfy both of these requirements simultaneously. In general, the smoothness of a signal is measured by the number of continuous derivatives it possesses. The smoother the signal, the faster the decay of its spectrum. Thus, we can improve leakage behavior by choosing a smooth (tapered) window. For a given duration, smoothing the window by tapering to reduce the level of sidelobes decreases the effective time-duration, and therefore increases the width of the mainlobe. Thus, we cannot simultaneously increase spectral resolution and decrease leakage. For example using Hann window the main laobe will be wider (disadvantage) and at the same time the sidelobes will be lower (advantage).

Problem No. 3:

(3.a)

$y_1[n] = x[n] * h[n]$

$y_2[n] = x[-n] * h[n]$ => $y_2[n] = x[n] * h[-n]$

$y[n] = y_1[n] + y_2[n]$ => $y[n] = x[n] * (h[n] + h[-n])$

$y[n] = h_s[n] * x[n]$ =>

$h_s[n] = h[n] + h[-n]$

(3.b)

The frequency response is

$h_s[n] = h[n] + h[-n] \xRightarrow{Fourier\ Transform}$

$H_s(e^{j\omega}) = H(e^{j\omega}) + H(e^{-j\omega})$ =>

$H_s(e^{j\omega}) = H_R(e^{j\omega}) + jH_I(e^{j\omega}) + H_R(e^{-j\omega}) + jH_I(e^{-j\omega})$ (3.1)

According to odd symmetry we have: $H_I(e^{-j\omega}) = -H_I(e^{j\omega})$ (3.2)

Additionally according to even symmetry we have: $H_R(e^{-j\omega}) = H_R(e^{j\omega})$ (3.3)

From (3.1), (3.2) and (3.3) we have:

$H_s(e^{j\omega}) = H_R(e^{j\omega}) + jH_I(e^{j\omega}) + H_R(e^{j\omega}) - jH_I(e^{j\omega})$ =>

$H_s(e^{j\omega}) = 2H_R(e^{j\omega})$

As we can see the phase response is zero. Also the filter is a real filter.

(3.c)

According to the assumption of (3.a) h[n] is IIR filter. So, practical computation of the autocorrelation function for an IIR can thus be done by first computing filter's impulse response, then flipping it left-to-right (the time inversion), and then computing the convolution of the original impulse response with the flipped one.