EE438 DIGITAL SIGNAL PROCESSING WITH APPLICATIONS

Assignment #4 - Spring 2001

due in class on Wednesday, 21 February 2001

1. Find expressions for the N point DFT's of the following signals.

   a. $x(n) = \begin{cases} \cos(2\pi k_o n/N), & n = 0,1,...,N-1 \\ 0, & \text{else} \end{cases}$

      Here k_o is some fixed integer; $0 \le k_o \le N-1$.

   b. $x(n) = \begin{cases} \cos[2\pi(k_o + 0.5)n/N], & n = 0,1,...,N-1 \\ 0, & \text{else} \end{cases}$

   c. $x(n) = \begin{cases} 1, & 0 \le n \le N/4 - 1 \\ & N/2 \le n \le 3N/4 - 1 \\ 0, & N/4 \le n \le N/2 - 1 \\ & 3N/4 \le n \le N-1 \end{cases}$

2. The way interpolation was presented in class involved inserting zeroes between samples and passing the resulting signal through an ideal lowpass filter. In this problem we consider a non ideal means of accomplishing interpolation by a factor of 2 amenable to practical implementation. To this end consider a real-valued N length sequence x(n), n = 0,1,...N-1, with DFT $X_N(k)$. Let $Y_{2N}(k)$ be a 2N length DFT sequence formed as

$$Y_{2N}(k) = \begin{cases} 2X_N(k), & k = 0,1,...,N/2-1 \\ 0, & k = N/2,\ldots,3N/2-1 \\ 2X_N(k-N), & k = 3N/2,\ldots,2N-1 \end{cases}$$

   Let y(n) denote the 2N pt. inverse DFT of $Y_{2N}(k)$.

   a. Show that y(2n) = x(n), n = 0,1,...,N-1.

   b. Find a *simple* expression for y(n) in terms of x(n) for arbitrary n.

3. Let x(n) and y(n) be two finite duration sequences of length L = 6, i.e., x(n) = y(n) = 0 for $n < 0$ and $n \ge 6$. Let $X_6(k)$ and $Y_6(k)$ denote 6-point DFT's of x(n) and y(n), respectively. The 6-point inverse DFT of the product $Z_6(k) = X_6(k)Y_6(k)$, denotes $z_6(n)$, produces the following values

$$z_6(n) = 21 \qquad n = 0,1,2,3,4,5$$

   Let $X_9(k)$ and $Y_9(k)$ denote the 9-point DFT's of the zero-padded sequences x(n) and y(n) defined above. The 9-point inverse DFT of the product $Z_9(k) = X_9(k)Y_9(k)$, denoted $z_9(n)$, produces the following values:

n	0	1	2	3	4	5	6	7	8
$z_9(n)$	9	12	15	18	20	21	15	10	6

   Given $z_6(n)$ and $z_9(n)$, find the *linear* convolution of x(n) and y(n).

4. Derive a decimation-in-time FFT algorithm for a 12 point DFT, and draw a complete flow diagram for the algorithm.

5. In class we examined an example where an analog signal was sampled and then truncated so that we could perform a finite-length DFT. Mathematically truncation is equivalent to multiplication by a rectangular window. Consider instead multiplication by an N length raised cosine window of the form

$$w(n) = a + b\cos\left[\frac{2\pi}{N}\left(n - \frac{N-1}{2}\right)\right] \quad n = 0,1,\ldots,N-1$$

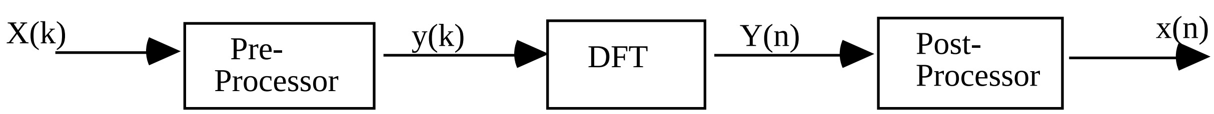
where a and b are just fixed constants. Determine the DTFT (NOT DFT) of w(n), $W(e^{j\omega})$, in terms of a and b. Simplify as much as possible. Sketch the magnitude of $W(e^{j\omega})$ over $-\pi < \omega < \pi$ for the following two cases:

i. Hanning window: a = .5 and b = .5

ii. Hamming window: a = 0.54 and b = 0.46

Specifically show and explain why the sidelobes achieved with either window is lower than that achieved with a rectangular window of the same length while the mainlobe achieved with either window is 50% wider than that achieved with a rectangular window.

6. Software libraries frequently do not contain routines for computing the inverse DFT. Instead, the inverse DFT is computed by preprocessing X(k) to produce a sequence y(k). The DFT of y(k) is then computed, resulting in Y(n). Finally, postprocessing of Y(n) results in the desired inverse DFT x(n) as shown below:



Find the required preprocessing and postprocessing operations. (They are both memory-less.)

7. Let v(n) be a length N complex-valued signal with DFT V(k).

a. Show that DFT[v*(n)] = V*(N-k). Let x(n) and y(n) be two real-valued signals with N point DFT's X(k) and Y(k). Form the complex-valued signal v(n) = x(n) + j y(n).

b. Find expressions for x(n) and y(n) in terms of v(n).

c. Combining your answers to Parts a and b, show how the N point DFT's of *two* real signals can be calculated by computing just *one* N point DFT of a complex-valued signal, *i.e.* show how X(k) and Y(k) may be recovered from V(k).

d. Combine your result from Part c with the first step in the derivation of the radix 2 decimation-in-time FFT algorithm to determine how the N point DFT of a real signal may be calculated via an N/2 point DFT.