Question 1

Let \( h(n) = \{1, 0, -1\} \) be the impulse response of an FIR filter, and let \( x(n) = \{1, 2, 1, 2\} \) be the input sequence. (The underline corresponds to time zero.)

(a) How many zeros need to be padded in \( x(n) \) and \( h(n) \), respectively, in order to avoid aliasing in the output, \( y(n) = x * h(n) \), if we want to use the DFT and IDFT to calculate \( y(n) \)?

**Solution:** The filter \( h(n) \) and input \( x(n) \) have duration \( N_h = 3 \) and \( N_x = 4 \), respectively. Therefore, the duration of \( y(n) \) is \( N_y = N_h + N_x - 1 = 3 + 4 - 1 = 6 \). Therefore, 3 zeros need to be padded in \( h(n) \) (\( N_y - N_h = 6 - 3 = 3 \)), and 2 zeros need to be padded in \( x(n) \) (\( N_y - N_x = 6 - 4 = 2 \)).

(b) Suppose that the zero padded versions of \( x \) and \( h \) have been defined in Matlab, and are called \( xpad \) and \( hpad \), respectively. Please provide Matlab code that calculates \( y(n) \) via DFT and IDFT.

**Solution:**

```matlab
x=[1 2 1 2];
h=[1 0 -1];
xpad=[x 0 0];
hpad=[h 0 0 0];
X = fft(xpad);
H = fft(hpad);
y = ifft(X.*H);
```
(c) Denote the length of $x(n)$ after zero padding by $L$. Suppose that you got $L = 4$ in part (a). Please compute the $L$ point DFT of $x(n)$.

Solution:

$$X(0) = 1 + 2 + 1 + 2 = 6;$$

$$X(1) = \sum_{k=0}^{3} x(k)e^{-j\frac{2\pi k}{4}} = 1 - 2j - 1 + 2j = 0;$$

$$X(2) = \sum_{k=0}^{3} x(k)e^{-j\frac{2\pi k \cdot 2}{4}} = 1 - 2 + 1 - 2 = -2;$$

$$X(3) = \sum_{k=0}^{3} x(k)e^{-j\frac{2\pi k \cdot 3}{4}} = 0.$$
Question 2
If \( x(n) \) is a periodic sequence with a period \( N \), i.e.,
\[
x(n) = x(n + N),
\]
then \( x(n) \) is also periodic with period \( 2N \). Let \( X_N(k) \) denote the discrete Fourier transform (DFT) coefficients when \( x(n) \) is considered to be periodic with period \( N \), and let \( X_{2N}(k) \) be the DFT coefficients when the period is assumed to be \( 2N \).

(a) Write the expression for \( X_{2N}(k) \) using \( x(n) \).

Solution: The DFT \( X_{2N}(k) \) can be written as
\[
X_{2N}(k) = \sum_{n=0}^{2N-1} x(n) e^{-j \frac{2\pi}{2N} nk}.
\]

(b) Express the DFT coefficients \( X_{2N}(k) \) in terms of \( X_N(k) \). (Hints: because \( x(n) = x(n + N) \), you can partition the sum into two parts. Note also that \( 1 + e^{-j\pi k} \) is either 0 or 2 depending on whether \( k \) is odd or even.)

Solution: Because \( x(n) = x(n + N) \), the sum in part (a) may be written as
\[
X_{2N}(k) = \sum_{n=0}^{N-1} x(n)[e^{-j \frac{2\pi}{N} nk} + e^{-j \frac{2\pi}{N} (n+N)k}]
\]
\[
= \sum_{n=0}^{N-1} x(n)e^{-j \frac{2\pi}{N} nk}[1 + e^{-j\pi k}].
\]
Note that the term in square brackets is equal to 2 when \( k \) is even, and it is zero when \( k \) is odd. When \( k \) is even,
\[
X_{2N}(k) = 2 \sum_{n=0}^{N-1} x(n)e^{-j \frac{2\pi}{N} n(k/2)} = 2X_N \left( \frac{k}{2} \right).
\]
Therefore, the DFT coefficients are
\[
X_{2N}(k) = \begin{cases} 
2X_N \left( \frac{k}{2} \right) & k = 0, 2, ..., 2N - 2 \\
0 & k = 1, 3, ..., 2N - 1 
\end{cases}
\]
Question 3
The first five points of the eight-point DFT of a real-valued sequence are \{1, 1 - j, 2, 2 + j, -1\}. Determine the remaining three points.

Solution: Because the signal is real valued, \(X(5) = X(3)^* = 2 - j\), \(X(6) = X(2)^* = 2\), and \(X(7) = X(1)^* = 1 + j\).
Question 4
Consider the signal \( x_1(n) = \sin \left( \frac{2\pi n}{N} \right) \). Compute \( \sum_{n=0}^{N-1} (x_1(n))^2 \) using the DFT.

**Solution:** We have seen that

\[
X_1(k) = \begin{cases} 
\frac{N}{2j} & k = 1 \\
-\frac{N}{2j} & k = N - 1 \\
0 & \text{else}
\end{cases}
\]

Parseval gives us

\[
\sum_{n=0}^{N-1} (x_1(n))^2 = \sum_{n=0}^{N-1} x_1(n) (x_1(n))^* = \frac{1}{N} \sum_{k=0}^{N-1} X_1(k) (X_1(k))^* = \frac{1}{N} \sum_{k=0}^{N-1} |X_1(k)|^2 = \frac{1}{N} [2 \cdot (N/2)^2] = \frac{N}{2}.
\]
Question 5
A discrete time signal \( x(n) \) has the form
\[
    x(n) = A_1 \cos(\omega_1 n) + A_2 \cos(\omega_2 n) + z(n),
\]
where \( z(n) \) is noise, and the frequencies \( \omega_1 \) and \( \omega_2 \) are unknown. You are given \( N = 10^4 \) samples of \( x(n) \) for \( n = 0, \ldots, N - 1 \).

(a) Suppose that the amplitudes of the cosines are identical, i.e., \( A_1 = A_2 \), and that these amplitudes are larger than the amplitude of the noise \( z(n) \). What type of window seems well matched for this spectral estimation problem? Justify your answer by describing how you would estimate \( \omega_1 \) and \( \omega_2 \).

**Solution:** The samples will be multiplied by the window, i.e., \( y(n) = x(n) \cdot w(n) \), where \( w(n) \) are samples of the window. We will then compute the Fourier transform of \( y \) and select the two frequencies that correspond to peaks in the Fourier transform. Because the amplitudes are identical, the side lobe need not be small; a rectangular window will be fine in this case.

(b) How would your answer change if you are told that \( A_1 = A_2 = 30A \), and that \( |\omega_1 - \omega_2| > 0.2\pi \)? (As before, \( A \) is larger than the amplitude of the noise \( z(n) \).) Justify your answer.

**Solution:** In this case, we must use a window with very small side lobes. Because the frequencies are well-separated it is not crucial that the main lobe is narrow. Hamming and Blackman windows have very low side lobes. Among these, the Blackman might be better because its side lobes taper off. With a big gap between the frequencies, the amplitudes of the side lobes will have plenty of space to decay significantly, and the lobe of the smaller cosine should be noticeable.
Suppose that we have two integers \( x_a \) and \( x_b \) that can be expressed using \( N \) digits in base-\( D \),

\[
x_a = \sum_{n=0}^{N-1} \tilde{x}_a(n)D^n,
\]

\[
x_b = \sum_{n=0}^{N-1} \tilde{x}_b(n)D^n.
\]

That is, \( x_a \) and \( x_b \) are the actual numbers and \( \tilde{x}_a \) and \( \tilde{x}_b \) are corresponding signals in base \( D \). (For example, we could have \( x_a = 12, x_b = 13, \tilde{x}_a = \{2, 1\} \), and \( \tilde{x}_b = \{3, 1\} \).)

Our goal in this question is to derive a fast algorithm to multiply \( x_a \) and \( x_b \).

(a) The classical approach to multiplication, which most of us learned in elementary school, proceeds as follows,

\[
x_c = x_a \cdot x_b
\]

\[
= \left( \sum_{n=0}^{N-1} \tilde{x}_a(n)D^n \right) \left( \sum_{n=0}^{N-1} \tilde{x}_b(n)D^n \right)
\]

\[
= \sum_{n_a,n_b=0}^{N-1} \tilde{x}_a(n_a)\tilde{x}_b(n_b)D^{n_a+n_b}.
\]

Please show that \( x_c = \sum_{n=0}^{2N-2} D^n \cdot \{\tilde{x}_a * \tilde{x}_b\}(n) \). That is, convolution between \( \tilde{x}_a \) and \( \tilde{x}_b \) forms the entries of a signal \( \tilde{x}_c \) that corresponds to the product \( x_c \).

Solution:

\[
x_a \cdot x_b = \sum_{n=0}^{2N-2} D^n \left[ \sum_{\{n_a,n_b\}:n_a+n_b=n,0 \leq n_a,n_b \leq N-1} \tilde{x}_a(n_a)\tilde{x}_b(n_b) \right]
\]

\[
= \sum_{n=0}^{2N-2} D^n \{\tilde{x}_a * \tilde{x}_b\}(n).
\]

That is, \( D^n \) is multiplied by the \( n \)'th term in the convolution between the two signals \( \tilde{x}_a \) and \( \tilde{x}_b \).
We can see that \( x_c = x_a \cdot x_b \) can be computed using the following algorithm:

1. Form the sequences \( \tilde{x}_a \) and \( \tilde{x}_b \).

2. Compute the convolution \( \tilde{x}_c = \tilde{x}_a * \tilde{x}_b \).

3. Multiply each \( x_c(n) \) by \( D^n \).

4. Compute the sum of the product terms.

Show using this algorithm that the product of \( x_a = 26 \) and \( x_b = 23 \) is \( x_c = 598 \) (we use the decimal base \( D = 10 \)). (In our example from the previous page, \( \hat{x}_a * \hat{x}_b = \{6, 5, 1\} \), and \( 6 \cdot 10^0 + 5 \cdot 10^1 + 1 \cdot 10^2 = 156 \), which is the product of \( x_a = 12 \) and \( x_b = 13 \).)

**Solution:**

(i) Because \( x_a = 26 \) and \( x_b = 23 \), they can be expressed as sequences, \( \tilde{x}_a = \{6, 2\} \) and \( \tilde{x}_b = \{3, 2\} \). (ii) We convolve the two sequences, \( \{6, 2\} * \{3, 2\} = \{18, 18, 4\} = \tilde{x}_c \). (iii) This new sequence \( \tilde{x}_c \) is multiplied by powers of 10, \( \{18 \cdot 10^0, 18 \cdot 10^1, 4 \cdot 10^2\} \). (iv) The sum is \( 18 + 180 + 400 = 598 \), which matches the correct answer.
(c) Suppose that $x_a$ and $x_b$ that can be expressed using $N$ digits, and suppose further that the FFT of an input of length $N$ can be computed using approximately $3N \log(N)$ floating point operations. Approximately how many operations would you need to compute $x_c$ using the FFT to perform the convolution in the algorithm above?

**Solution:** It is easily seen that part (i) of the algorithm dominates the runtime, because the other parts are simple. Recall from the previous question that convolution can be computed by zero padding the two sequences, each of length $N$, to length $2N - 1$. We then compute two DFT’s and one IDFT. Each of these can be implemented with approximately $3 \cdot 2N \log(2N)$ operations, and in total this is $18N \log(2N)$.