ECE113, Winter 2023 **Homework #5 Solutions**
Digital Signal Processing Prof. A. Kadambi
University of California, Los Angeles; Department of ECE TA: S. Zhou, A. Vilesov

Due Friday, 10 March 2023, by 11:59pm to Gradescope.
50 points total.

Note: Unless specified, you are free to use any of the properties of DFT that were taught in class. Also, all repeated derivations can be referenced with appropriate equation/result numbers.

1. (10 points) **Compression**
   For this problem, assume we are sampling a real-valued signal at 10,000 Hz for 5 seconds. Each sample is a float datatype that is 4 bytes. After sampling the signal, we apply the DFT to the discrete signal. Our goal is to send our friend the DFT representation of our signal using as few bytes as possible.

   (a) How many bytes are needed to naively represent the time-domain signal? And how many are needed for the DFT representation of the signal? (Remember, in the frequency domain, a discrete frequency is complex, so it would take 8 bytes to represent the real and imaginary part).

   (b) Since this is a real-valued signal, is there some kind of symmetry we can take advantage of in the frequency domain so that we do not have to send all the complex frequency components? How many bytes would we need to send our friend and are there any special instructions to reconstruct the original DFT?

   (c) After analyzing the signal, you found that most of the frequency content of the signal is between 2,000 and 4,000 Hz. Can you use this information to send even less data? How many bytes would we need to send our friend and are there any special instructions to reconstruct the original DFT?

   For this problem, we are sampling a real-valued signal for 5 seconds, but we have not determined a sampling frequency. Each sample is a float datatype that is 4 bytes. Our goal is to figure out a sampling rate such that once we get the DFT representation of the signal we use as few bytes as possible to send it to our friend.

   (d) Assume that the max frequency we would ever see in the signal is 8,000 Hz. What sampling frequency should we use and how many bytes of data would we require to naively store the time-domain and frequency-domain representation of the signal.

   (e) It turns out after some analysis of the signals we are recording, that the minimum frequency we would ever see is 4,000 Hz. Can we lower the sampling frequency while still retaining all the information present in the original signal?

   Solutions:

   a.) Bytes for Time-domain Signal = (4 bytes)(10 kHz)(5 s) = 200 kB
   Bytes for Frequency-domain Signal = (8 bytes)(10 kHz)(5 s) = 400 kB

b.) We notice that since the signal is real, there is symmetry in the spectrum. Then we only have to send half the amount of information, 400 kB/2 = 200 kB. We can send one side of the spectrum for $k = 0, 1, ...N/2 - 1$. Then to reconstruct the rest of the spectrum for $k = N/2, N/2+1, ...N-1$ apply the following formula, $X[k] = X^*[N-k]$, where the asterisk is the complex conjugate.

c.) If most of the frequency content is between 2,000 and 4,000 Hz, we can send just the frequency indices corresponding to that band of frequencies. $k_1 = 50,000\frac{2{,}000Hz}{10{,}000Hz} = 10,000$ and $k_2 = 50,000\frac{4{,}000Hz}{10{,}000Hz} = 20,000$. Then we can send the the values of the DFT at frequency indices between $[k_1, k_2]$ as well as $[N-k_2, N-k_1]$ where N is 50,000. The instructions we would then send are to initialize a complex array of size 50,000 with all zeros. Then fill in the values of the indices at $[k_1, k_2]$ and $[N-k_2, N-k_1]$ with the values sent. The total number of bytes sent would be $(2*10,000)(8bytes) = 160kB$, and if we apply the trick from part b.) then it would be $80kB$.

d.) From the Nyquist Sampling Theorem, we know that the max sampling frequency should be 16 kHz = (2)(8 kHz). This would then require, 5*16 kS = 80,000 samples or 320 kB.

e.) Yes, we can take advantage of aliasing to sample at a lower frequency, given the knowledge that there are no frequencies in our signal that are lower than 4 kHz. Then, we could sample at 8 kHz, and, a true analog frequency of 5 kHz would alias with a digital frequency at 3 kHz.

2. (10 points) **FFT and Polynomial Multiplication**
In this question we will explore how we can apply the FFT algorithm to polynomial multiplication.

   (a) Express in Big-O notation what the computational complexity of polynomial multiplication is.

   (b) See if you can rewrite polynomial multiplication as convolution. Is this circular or linear convolution?
   (Hint: rewrite the polynomial equation into a signal whose values are the coefficients of the polynomials. Also, zero-padding will help)

   (c) Show the steps needed to accelerate polynomial multiplication with FFT algorithm. Express in Big-O notation what the computational complexity of polynomial multiplication would be using this method.

Solutions:
Assume we have two polynomials of the same order, d, such that $A = a_0 + a_1x + a_2x^2 + ...a_dx^d$ and $B = b_0 + b_1x + b_2x^2 + ...b_dx^d$.

(a) A polynomial of order d has $d+1$ coefficients. Then the complexity is:

$$(d+1) + \sum_{n=1}^{d} n + \sum_{n=1}^{d} n = d + 1 + d\frac{1+d}{2} + d\frac{1+d}{2} = d^2 + 2d + 1 = (d+1)^2 \quad (1)$$

Then, the complexity is $\mathbf{O}(d^2)$.

(b) We can rewrite polynomial multiplication as convolution by rewriting the coefficients of the polynomials as signals and adding some zero-padding. Then $AB = c = a \circ b$, where $a = [a_0, a_1, a_2, ...a_d, 0, 0, ..., 0]$. The number of zeros we add is d. One can than observe that the signal, $c[n]$, is the result of the polynomial multiplication where the value at a certain index, k, is the coefficient of the kth order term in the polynomial AB.

(c) Since we have shown that polynomial multiplication can be rewritten as convolution, we can apply the FFT algorithm. Then the steps are as follows.

   i. Convert the 2 polynomials to a signal as stated in (b).
   ii. Take the FFT of both signals.
   iii. Multiply the frequency domain signals.
   iv. Take the IFFT of the resulting of the multiplication in the frequency domain.
   v. Convert the signal back to a polynomial.

The resulting complexity from the 2 FFTs, mulitplication in the frequency domain, and IFFT is:

$$\mathbf{O}(2(2d{+}1)\log_2(2d{+}1))) + (2d{+}1) + \mathbf{O}((2d{+}1)\log_2(2d{+}1)) = \mathbf{O}(2d{+}1)\log_2(2d{+}1)) \quad (2)$$

$$\implies \text{Time Complexity} = \mathbf{O}(d\log_2(d)) \quad (3)$$

3. (10 points) **DFT Properties**
Let x[n] be a finite length sequence. Its DFT is defined as, $DFT(x[n]) = [0, 1-j, 1, 1+j]$. Using the DFT properties, solve for the DFTs of the following sequences:

   (a) $y[n] = x[n]cos(\frac{\pi}{2}n)$

   (b) $y[n] = [0, 0, 1, 0] \circ x[n]$

   (c) $y[n] = g[n]x[n]$, where $g[n] = [0, 0, 1, 0]$

   (d) $y[n] = x[n-1]$

Solutions:

   (a) Here we can use the modulation property where $x[n]cos(2\pi k_o n)$ has a DFT $\frac{1}{2}(X[k-k_o) + X[k+k_o]$. Additionally, $cos(\frac{\pi}{2}n) = cos(2\pi\frac{1}{4})$.
   Then $Y[k] = \frac{1}{2}([1+j, 0, 1-j, 1] + [1-j, 1, 1+j, 0]) = \frac{1}{2}[2, 1, 2, 1]$.

   (b) Here we recognize that the DFT of $[0, 0, 1, 0]$ is $e^{-j2\pi\frac{2}{4}k}$. Then, using the convolution theorem, $Y[k] = e^{-j2\pi\frac{2}{4}k}X[k] = [0, -1+j, 1, -1-j]$.

   (c) We can use the convolution theorem again, but this time it will be convolution in the frequency domain. $Y[k] = \sum_{l=0}^{3} X[l]G[k-l]$. From the previous problem we know the DFT of $g[n]$ is $G[k] = e^{-j2\pi\frac{2}{4}k} = (-1)^k$. Then $Y[k] = \frac{1}{4}[-1, 1, -1, 1]$

   (d) This is just the time shift theorem, $x[n-n_d]$ has a DFT of $e^{-j2\pi\frac{k}{N}n_d}X[k]$. Since, $e^{-j2\pi\frac{k}{4}} = [1, -j, -1, j]$ then $Y[k] = [0, -1-j, -1, -1+j]$.

4. (10 points) **DFT Properties II** You are given a N length signal (where N is even) such that:

$$x[n] = u[n] - u[n - N/2], \quad n = 0, 1, 2, ... \; N - 1 \tag{4}$$

where $u[n]$ is a unit step function defined as:

$$u[n] = \begin{cases} 1, \; n \geq 0 \\ 0, \; n < 0 \end{cases} \tag{5}$$

Find the sequence $y[n]$ such that its DFT is $Y[k] = X[k]^2$.

Solutions: If $Y[k]$ is the result of multiplication of $X[k]$ with $X[k]$ to create $X^2[k]$. Then $y[n]$ is the convolution of $x[n]$ with itself. Therefore:

$$y[n] = \begin{cases} n + 1, & n = 0, 1, 2, ..., N/2 - 1 \\ N - n - 1, & n = N/2, N/2 + 1, ...N - 1 \end{cases} \tag{6}$$

5. (10 points) **Coding**
Coding Assignment Instructions:

   (a) Find the hw5_coding.zip file on BruinLearn. Download and unzip it.

   (b) The zip contains one file that you will be editing "hw5.ipynb".

   (c) Make sure you only make changes in the code where specifically asked of you by Python comments.

   Solutions: Solutions are be posted on BruinLearn in the file 'hw5_coding_sol.zip'.

Practice Problems

1. **CYU: DFT**

   (a) You are given a 64 point DFT of the real sequence $x[n]$. Unfortunately, the value at k=40 was accidentally changed. Can we get the original value back? If so, what was the value originally?

   Solutions: Since for a real sequence the DFT has symmetry, we can get the original value from the index at $k = 24$ and taking the complex conjugate. $X[40] = X^*[24]$.

   (b) You sample an analog signal at a sampling frequency of 20 kHz and acquire 800 samples. You then zero-pad the signal up to N=2,000 samples. The resulting zero-padded signal is $x[n]$ and its DFT is $X[k]$.
   Solutions:

   i. What is the maximum analog frequency recoverable from these samples?
   10 kHz

   ii. What is the frequency spacing between neighboring frequency bins? (What is the difference in the analog frequencies corresponding to the indices k and k+1)
   10 Hz

   iii. What analog frequency does the index at $k = 250$ in the DFT of $x[n]$ correspond to?
   2.5 kHz