ECE 417 Lab Exercise # 2
Discrete Fourier Transform and Convolution

In this lab, you will study the computation of the Discrete Fourier Transform (DFT) using the Fast Fourier Transform (FFT) algorithm, and examine some of its properties and applications. You will also use FFT to perform linear convolution. For computing the DFT, you will use the *fft* command in Matlab.

1 DFT Computation and Properties

In this part you will compute the DFT of given sequences and examine some of the properties of DFT.

1.1 DFT of a real-valued sequence

Sometimes DFT is computed with size twice or 4 times the duration of the signal. This allows the frequency content to be viewed on a denser grid (better visualization) though it does not provide any additional information.

Consider $x[n] = (n+3)(u[n] - u[n-8])$ (x = 3:10 in Matlab). Compute its 8-point DFT, $X_8[k]$, and its 16-point DFT, $X_{16}[k]$, using the *fft* command. Note that in Matlab the indexes start from 1 (not 0) by default.

Q2.1 Examine any symmetries in the real and imaginary parts of $X_8[k]$ beyond the first (DC) sample. Show that $X_8[0] = X_8^*[0]$ and $X_8^*[8-k] = X_8[k]$ for $k = 1, ..., 7$ (in Matlab with an offset of 1).

Q2.2 Examine $X_8[k]$ and $X_{16}[k]$ for and report the values of index k for $X_8[k]$ and $X_{16}[k]$ are equal (except possibly for some small numerical error in computation). Explain why this is so. Provide a stem plot of the absolute values of $X_{16}[k]$ for $k = 0, 1, 2, \ldots, 15$. For example "stem([0:15], abs(fft(x,16)))" in Matlab). Note any symmetries about $k = 8$.

1.2 DFT of single-tone exponential and sinusoidal sequences

- Case 1: Let $N = 64$ and $\alpha = 10$. Compute the N-point DFT of the following complex-valued sequence:

$$x[n] = \begin{cases} e^{\frac{j2\pi\alpha n}{N}} & 0 \leq n \leq N-1 \\ 0, & \text{otherwise.} \end{cases}$$

Sample code is given below

```
N=64;
n = [1:N];
alpha=10;
x = exp(2j*pi*(n-1)*alpha/N); % We use (n-1) to start the index at 0
X = fft(x,N);
stem(n-1,real(X));
figure
stem(n-1,imag(X));
```

Q2.3 Show a plot of real part of the DFT. Note that the imaginary part should be negligible in this case – small values due to arithmetic quantization errors. Now consider $\alpha = -10$. Explain the relation between DFTs in the cases for $\alpha = -10$ and $\alpha = 54$.

- Case 2: Let $N = 64$ and $\alpha = 10.3$. Compute the N-point DFT of the following complex-valued sequence:

$$x[n] = \begin{cases} e^{\frac{j2\pi\alpha n}{N}} & 0 \leq n \leq N-1 \\ 0, & \text{otherwise.} \end{cases}$$

Q2.4 Compare the results in Cases 1 and 2 and explain the difference in the nature of DFTs.

- Case 3: Let $N = 64$ and $\alpha = 10$. Compute the N-point DFT of the following real-valued sequence:

$$x[n] = \begin{cases} 2\cos\frac{2\pi\alpha n}{N} & 0 \leq n \leq N-1 \\ 0, & \text{otherwise.} \end{cases}$$

Q2.5 Explain any relationship with the results obtained in case 1.

2 DFT for estimating signal frequency content (spectra)

DFT helps to estimate signal frequency content (spectrum estimation) and used in a variety of applications. Here we will use it for removing undesired content.

Listen to the signal 'clean_signal.wav' and examine the DFT of the signal. Next listen to a contaminated version 'signal_plus_noise.wav' of the signal and examine the DFT of the signal.

Design a suitable FIR filter of order 32 using the routine *firpm* and process 'signal_plus_noise.wav' to remove the contamination.

Q2.6 Report on the comparison between the sounds of the clean signal and the de-contaminated signal. Show the plot of the filter magnitude response.

Q2.7 Find MSE_1, the mean squared error between the clean signal and the noise-contaminated signal. Also find MSE_2, the mean squared error between the clean signal and the filtered noise-contaminated signal. Compare MSE_1 and MSE_2.

Q2.8 Now alter the FIR filter a little and check if you get a lower mean squared error.

3 DFT and convolution

Consider sequences:

$$x[n] = (n+3)(u[n] - u[n-15]), \quad h[n] = (u[n] - u[n-5])$$

Compute the 16-point DFTs $X_{16}[k]$ and $H_{16}[k]$ and then compute $y_{16}[n]$, the IDFT of $Y_{16}[k] = X_{16}[k]H_{16}[k]$. Similarly compute $y_{32}[n]$, the IDFT of $Y_{32}[k] = X_{32}[k]H_{32}[k]$ using 32-point DFTs $X_{32}[k]$ and $H_{32}[k]$

Q2.9 Plot the sequences $y_{16}[n]$ and $y_{32}[n]$. How are they related to the linear convolution $y[n] = x[n] * h[n]$.

Now suppose you are restricted to using an 8-point DFT processor and you wish to perform the linear convolution $y[n] = x[n] * h[n]$. This can be done using overlap-add and overlap-save methods.

Q2.10 Create suitable segments of $x[n]$ and use the *overlap-add* method to perform the linear convolution. Plot the output for each input segment and show the sum of the suitably shifted-and-overlapped versions of the output as the final convolution.

Submit a copy of your code along with the report

(Written by R. Ansari. Updated Fall 2023.)