Signal Processing First

Lab 7: The Fast Fourier Transform

Pre-Lab and Warm-Up: You should read at least the Pre-Lab and Warm-up sections of this lab assignment and go over all exercises in the Pre-Lab section before going to your assigned lab session.

Verification: The Warm-up section of each lab must be completed **during your assigned Lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. One of the laboratory instructors must verify the appropriate steps by signing on the **Instructor Verification** line. When you have completed a step that requires verification, simply demonstrate the step to the TA or instructor. Turn in the completed verification sheet to your TA when you leave the lab.

1 Introduction & Objective

The goal of the laboratory project is to introduce the Fast Fourier Transform (FFT) algorithm for efficient computer calculation of the Fourier transform and to investigate some of the Fourier Transform's properties.

2 Background

2.1 The Fast Fourier Transform

Suppose `g` is an array of N values representing the time signal $g[n] = g(nT_s)$. The MATLAB command

```
>>  G  =  fft(g);
```

causes Matlab to compute the discrete Fouirer transform of the signal *g(nTs)* and place the result in array G. The array G represents a spectrum *G(nΔf),* also of N values. Remember that Matlab numbers its array elements starting with one. This means that *g[0]* is stored in array element *g(1)* and *g((N-1)Ts)* is stored in *g(N).* Similarly *G(0)* is stored in array element *G(1)* and *G((N-1) Δf)* is stored in *G(N).* for greatest computational efficiency, N should be a power of two. If g(nTs), n=0, … , N-1 is a time signal, the Fourier transform that Matlab calculates is given by

$$G(k\Delta f) = \sum_{k=0}^{N-1} g(nT_s)e^{-j2\pi kn/N}, \quad n = 0, ..., \qquad (1)$$

Note that T_s represents the time between values of kT_s). It turns out that the frequency interval f between values of G(n f) is given by

$f= NT_s$

Given the array G, the MATLAB command

```
>> g = ifft(G);
```

calculates the inverse Fourier transform given by

$$g(nT_s) = \frac{1}{N}\sum_{k=0}^{N-1} G(k f)e^{+j2\pi nk/N}, \quad k = 0, \ldots, N-1. \qquad (2)$$

The Fourier transform given by equation 1 is called a discrete Fourier transform or DFT. MATLAB uses the discrete transform because MATLAB cannot store continuous-time signals. MATLAB uses an efficient algorithm called the Fast Fourier Transform (FFT) to calculate the discrete Fourier transform. The dis-crete Fourier transform has properties that are similar to those of the familiar continuous Fourier transform.

There is one important difference. The spectrum $G(k\ f)$ defined in equation 1 above is periodic in fre-quency with period $f_s = N\ f$. This periodicity is a consequence of the discrete-time nature of the time signal $g(nT_s)$. One period of the spectrum extends from frequency 0 to frequency $(N-1)\Delta f$. The posi-tive frequency components lie between frequency 0 and frequency $(\frac{N}{2}-1)\Delta f$. The spectral components from frequency N $f/2$ to $(N-1)\Delta f$ are repeats of the negative frequency components that lie between frequencies $-(\frac{N}{2})\Delta f$ and $-\ f$ respectively. Because the spectrum $G(k\ f)$ is defined only at discrete values of frequency, the FFT algorithm considers the time function $g(nT_s)$ to be periodic with period NT_s. Consequently, the N-value array g you define will be interpreted as one period of an infinite-duration peri-odic signal. The spectrum $G(k\ f)$ defined in equation (1) is actually the Fourier transform of the periodic signal.

2.2 Plotting the Spectrum

If $G(k\ f)$ is plotted against frequency, zero hertz will appear on the left of the graph. The positive fre-quency components will appear to the right of zero, followed farther to the right by the negative frequency components. Because this is contrary to convention, MATLAB provides a function to rearrange the compo-nents of the array `G` to place the negative frequency components to the left of zero. The command

```
>> H = fftshift(G);
```

will create an array `H` that represents a spectrum $H(n\ f)$ whose DC component is in the center as expected. Before plotting `H` (or `G`), recall that these arrays may contain complex numbers. The command `plot(H)` will cause MATLAB to plot the imaginary part against the real part. This usually gives an interesting graph, but probably not the one you had in mind! You may obtain the magnitude spectrum by the command `M = abs(H)`, and the angle spectrum by `a = angle(H)`. You may also want to use the commands `real()` and `imag()` to find the real and imaginary parts of the signals you are examining. Tip: if `H` is real, plot it. If `H` is complex, plot `abs(H)` and `angle(H)`.

3 Pre-Lab

A discrete time “unit impulse” is defined by the time signal

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$$

The spectrum of unit impulse (N = 16.) is computed and plotted by following commands.

```
>> xx=[1 zeros(1,15)];
>> stem(xx)
>>yy=fft(xx);
>>figure; plot(abs(yy))

>> stem(ifft(yy));
```

4 Warm Up

1. Let the time signal be a unit step (N = 16.)
   - Compute and plot the spectrum.
   - Compute and plot the inverse spectrum.
2. Let the time signal be a cosine of amplitude one whose frequency is 100 Hz.
   - Compute and plot the spectrum.
   - Compute and plot the inverse spectrum.