Experiment
1

Fourier Series & Fourier Transforms

MATLAB Simulation

Objectives

Fourier analysis plays an important role in communication theory. The main objectives of this experiment are:

1) To gain a good understanding and practice with Fourier series and Fourier Transform techniques, and their applications in communication theory.

2) Learn how to implement Fourier analysis techniques using MATLAB.

Pre-Lab Work

You are expected to do the following tasks in preparation for this lab:

- MATLAB is a user-friendly, widely used software for numerical computations (as you learned in EE207). You should have a quick review of the basic commands and syntax for this software. The following exercises will also help in this regard.

  Note: it is important to remember that Matlab is vector-oriented. That is, you are mainly dealing with vectors (or matrices).

1) Consider the following:

Y=3+5j

a. How do you get MATLAB to compute the magnitude of the complex number Y?
b. How do you get MATLAB to compute the phase of the complex number Y?

2) Vector manipulations are very easy to do In MATLAB. Consider the following:

```
xx=[ones(1,4), [2:2:11], zeros(1,3)]
xx(3:7)
length(xx)
xx(2:2:length(xx))
```

Explain the result obtained from the last three lines of this code. Now, the vector **xx** contains 12 elements. Observe the result of the following assignment:

```
xx(3,7)=pi*(1:5)
```

Now, write a statement that will replace the odd-indexed elements of **xx** with the constant –77 (i.e., **xx(1)**, **xx(3)**, etc). Use vector indexing and vector replacement.

3) Consider the following file, named example.m:

```
f=200;
tt=[0:1/(20*f):1];
z=exp(j*2*pi*f*tt);
subplot(211)
plot(real(z))
title('REAL PART OF z')
subplot(212)
plot(imag(z))
title('IMAGINARY OF z')
```

a. How do you execute the file from the MATLAB prompt?
b. Suppose the file name was "example.cat". Would it run? How should you change it to make it work in MATLAB?
c. Assuming that the M-file runs, what do you expect the plots to look like? If you're not sure, type in the code and run it.

Introduction

Recall from what you learned in EE207 that the input-output relationship of a linear time-invariant (LTI) system is given by the convolution of the input signal with the impulse response of the LTI system. Recall also that computing the impulse response of LTI systems when the input is an exponential function is particularly easy. Therefore, it is natural in linear system analysis to look for methods of expanding signals as the sum of complex exponentials. Fourier series and Fourier transforms are mathematical techniques that do exactly that!, i.e., they are used for expanding signals in terms of complex exponentials.

Fourier Series:

A Fourier series is the orthogonal expansion of periodic signals with period T_o when the signal set $\{e^{j2\pi nt/T_o}\}_{n=-\infty}^{\infty}$ is employed as the basis for the expansion. With this basis, any given periodic signal $x(t)$ with period T_o can be expressed as:

$$x(t) = \sum_{n-\infty}^{\infty} x_n e^{j2\pi nt/T_o}$$

where the x_n's are called the Fourier series coefficients of the signal $x(t)$. These coefficients are given by:

$$x_n = \frac{1}{T_o}\int_0^{T_o} x(t)e^{-j2\pi nt/T_o}\,dt$$

This type of Fourier series is called the exponential Fourier series. The frequency $f_o = 1/T_o$ is called the fundamental frequency of the periodic signal. The n[th] harmonic is given by the frequency $f_n = nf_o$.

If $x(t)$ is a real-valued periodic signal, then the conjugate symmetry property is satisfied. This basically states that $x_{-n} = x_n^*$, where * denotes the complex conjugate. That is, one can compute the negative coefficients by only taking the complex conjugate of the positive coefficients. Based on this result, it is obvious to see that:

$$| x_n | = | x_{-n} |$$
$$\angle x_n = -\angle x_{-n}$$

Fourier Transform:

The Fourier transform is an extension of the Fourier series to arbitrary signals. As you have seen in class, the Fourier Transform of a signal $x(t)$, denoted by $X(f)$, is defined by:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi f t} dt$$

On the other hand, the inverse Fourier Transform is given by:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{2\pi f t} df$$

If $x(t)$ is a real signal, then $X(f)$ satisfies the following conjugate symmetry property:

$$X(-f) = X^*(f)$$

In other words, the magnitude spectrum is even while the phase spectrum is odd. There are many properties satisfied by the Fourier Transform. These include Linearity, Duality, Scaling, Time Shift, Modulation, Differentiation, Integration, Convolution, and Parserval's relation.

Lab Work

Part A: Fourier Series

1) In MATLAB, go to the command window and type **Fourier_series_demo.m.** This will bring up a Graphical User Interface (GUI) that can be used to test and demonstrate many concepts and properties of the Fourier series expansion.

- Try different types of functions, starting with the square wave, fully rectified sine, sawtooth, etc. Run different examples while changing the fundamental frequency and number of harmonics in the FS expansion. Report your observations. In particular, explain why the Fourier series for the square and sawtooth waves require many more harmonics than the rectified sine waves in order to get a close match between the FS and the original function?

- Consider the plots for amplitude and phase spectra. State what kind of symmetry is present in each type of spectrum, and why? The plots also indicate the presence of FS terms with "negative" frequencies! What's the interpretation of that? Are there really negative frequencies? Explain.

2) Now, consider a periodic signal *x*(*t*). Compute and plot the discrete magnitude and phase spectra of this signal given by $x(t) = e^{-t/2}$ where $t \in [0, \pi]$. For this, you need to use the Fast Fourier Transform (FFT) function in MATLAB (refer to the notes below for more details). For the expansion of the signal x(t), the number of harmonics N_o to be used should be 32, the period T_o is π, and the step size is $t_s = T_o / N_o$. The output should be in two figure windows. The first window should contain $x(t)$ while the second window should contain both the magnitude and phase spectra versus a vector of harmonics indices (for example, *n*). You also need to include labels and titles in all plots. What can you observe from these plots?

Notes: In MATLAB, Fourier series computations are performed numerically using the Discrete Fourier Transform (DFT), which in turn is implemented numerically using an efficient algorithm known as the Fast Fourier Transform (FFT). Refer to the textbook (Sect.2.10 & 3.9) for more theoretical details. You should also type: **help fft** at the MATLAB prompt and browse through the online description of the **fft** function.

Because of the peculiar way MATLAB implements the FFT algorithm, the **fft** MATLAB function will provide you with the positive Fourier coefficients including the coefficient located at 0 Hz. You need to use the even amplitude symmetry and odd phase symmetry properties of the Fourier series for real signals (see the introduction to Fourier series of this experiment) in order to find the coefficients for negative harmonics.

As an illustration, the following code shows how to use **fft** to obtain Fourier expansion coefficients. You can study this code, and further enhance it to complete your work.

```
Xn = fft(x,No)/No;
Xn = [conj(Xn(No:-1:2)), Xn];
Xnmag = abs(Xn);
Xnangle = angle(Xn);
k=-N0/2+1:N0/2-1
stem(k, Xnmag(No/2+1:length(Xn)-No/2))
stem(k,Xnangle(No/2+1:length(Xn)-No/2))
```

Useful MATLAB Functions: exp, fft(x,No), length( ), conj, abs, angle, stem, figure, xlabel, ylabel, title.

Part B: Fourier Transform

3) In the MATLAB command window, type **Fourier_trans_demo.m** to launch a GUI that will demonstrate and review the basic properties of the Fourier transform. The basic function used is a rectangular unit pulse.

- First, introduce a certain time delay in the function, and notice what happens to the amplitude spectra. Explain why?
- Next, introduce different scaling factors and comment on what you are observing.
- Now, introduce a frequency shift, which means that the unit pulse is multiplied by a given sine or cosine signal with some frequency (later, we will see this is known as Amplitude Modulation). Referring to the basic properties of the FT, explain what you are observing in the plots.

4) Now, consider the signals $x_1(t)$ and $x_2(t)$ described as follows:

$$x_1(t) = \begin{cases} t+1, & -1 \le t \le 0 \\ 1, & 0 < t \le 1 \\ 0, & elsewhere \end{cases}$$

$$x_2(t) = \begin{cases} t, & 0 \le t \le 1 \\ 1, & 1 < t \le 2 \\ 0, & elsewhere \end{cases}$$

Plot these signals and their relative spectra in MATLAB. What do you conclude from the results you obtained? Are there any differences?
You need to plot both time signals in one figure window. Similarly, you need to plot the magnitude and phase spectra for both signals in one figure window, i.e, overlapping each other. For the phase, display small values by using the **axis** command. You also need to normalize the magnitude and phase values, and you should include the labels, titles, grid, etc. Assume the x-axis to work as a ruler of units. Each unit contains 100 points and let the starting point to be at –5 and the last point to be at 5.

Notes: Similar to Fourier series, Fourier transform computations in MATLAB are easily implemented using the **fft** function. The following code illustrates that. Notice in particular the function **fftshift** is very useful for presenting the Fourier spectrum in an understandable format. The internal algorithm used in MATLAB to find the FFT points spreads the signal points in the frequency domain at the edges of the plotting area, and the function **fftshift** centers the frequency plots back around the origin.

```
X = fft(x);
X = fftshift(X);
Xmag = abs(X);
Xmag = Xmag/max(X1mag);%Normalization
Xangle = angle(X);
Xangle = Xangle/max(Xangle);
F = [-length(X)/2:(length(X)/2)-1]*fs/length(Xmag);
plot(F, Xmag), plot(F, Xangle);
```

5) Repeat the above for the following signals, and report your observations & conclusions

$$x_1(t) = \begin{cases} 1, & |t| \le 3 \\ 0, & elsewhere \end{cases}$$

$$x_2(t) = \begin{cases} 1, & |t| \le 1 \\ 0, & elsewhere \end{cases}$$

6) In the MATLAB directory you are working in, you will find a MAT-file named **Exp1Part4.mat**. You need to load that file as follows:

```
load Exp1Part4.mat
```

After you successfully loaded the file, go to the command window and type **whos** and press **Enter**. You will notice three stored variables fs (sampling frequency or 1/ts), t (time axis vector) and m (speech signal). These correspond to a portion of speech recording.

The next step is to plot the speech signal versus the time vector t. In the same figure window and a second window panel, display the magnitude spectrum of m (call it M). What is the bandwidth of the signal? What can you notice in terms of the speech signal? In order to play the signal properly, make sure that the speakers are turned on and write the following MATLAB statement:

```
sound(m,fs)
```