Toronto Metropolitan University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering
& Architectural Science

ELE 532
Signals & Systems I

Laboratory Assignment 4

The Fourier Transform: Properties and Applications

Objective

In this assignment, you will use the Fourier Transform to analyze, observe and study frequency-domain characteristics of time waveforms. You will explore properties of the Fourier Transform and use the Fourier Transform as a diagnostic tool to identify the characteristics of a bandpass transmission channel. You will design and implement a simple communications system that will allow you to transmit a voice signal over the bandpass transmission system and decode the received signal.

Introduction

The Fourier Transform provides a convenient frequency-domain representation of a signal. It extends the Fourier series representation from periodic signals to aperiodic signals. Let $x(t)$ be a finite-energy signal (alternatively assume that $x(t)$ satisfies *Dirichlet conditions*[1]. Let $X(\omega)$ be the Fourier Transform of $x(t)$. We determine $X(\omega)$ using the **Fourier integral**, the *analysis equation*:

$$X(\omega) = \mathcal{F}[\,x(t)\,] = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt, \tag{1}$$

where $\omega = 2\pi f$ is the angular frequency measured in [rad/s]. We can "recover" $x(t)$ from its Fourier Transform using the *synthesis equation*:

$$x(t) = \mathcal{F}^{-1}[\,x(t)\,] = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega. \tag{2}$$

Equivalently, we use the symbolic representation $x(t) \Longleftrightarrow X(\omega)$ to show that $x(t)$ and $X(\omega)$ form a Fourier Transform pair.

Preparation

- Read *Lathi, Chapter 7*, specifically sections 7.1, 7.2, 7.3 and 7.4, pp. 680–729.
- Work through Section 7.9-1 on page 757 of the course reference text *Lathi*.

Lab Assignment

A. The Fourier Transform and its Properties

You can generate the signal in Figure (1) using the MATLAB commands:

```
>> N = 100; PulseWidth = 10;
>> t = [0:1:(N-1)];
>> x = [ones(1,PulseWidth), zeros(1,N-PulseWidth)];
```

[1] Dirichlet conditions are sufficient (but not necessary) for point-wise convergence of the Fourier Transform. They are: (i) the function $x(t)$ must be absolutely integrable, i.e., $\int |x(t)|dt < \infty$ (ii) $x(t)$ may have only a finite number of maxima and minima within a finite interval, and (iii) $x(t)$ may have only a finite number of discontinuities within a finite interval.

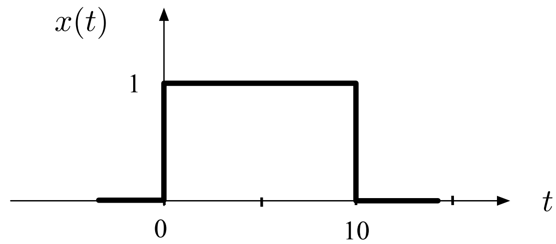

Figure 1: Single pulse signal $x(t)$.

and display[2] the corresponding waveform:

```
>> stairs(t,x); grid on; axis([-10,110,-0.1,1.1])
```

The single pulse described by `x` has a length of 100 points. We compute the the Fourier Transform of `x` using the MATLAB command `fft`[3]:

```
>> Xf = fft(x);
```

In general, the Fourier Transform $X(\omega)$ is a complex-valued function of the frequency variable ω. Therefore, to plot the frequency spectrum of the signal `x`, we first extract the magnitude- and phase-spectra from $X(\omega)$. We also define the frequency vector `f` as:

```
>> f = [-(N/2):1:(N/2)-1]*(1/N);
```

The choice of the frequency variable f measured in units of [Hz] rather that the radian frequency variable ω measured in units of [rad/s] is for convenience. If you prefer to work with the radian frequency variable, the use `w=2*pi*f` in generating the spectral plots. We also use the MATLAB command `fftshift` to centre the double sided spectrum on the frequency origin. Display the magnitude- and phase spectra of the pulse waveform `x` using the commands:

```
>> subplot(211); plot(f,fftshift(  abs(Xf))); grid on;
>> subplot(212); plot(f,fftshift(angle(Xf))); grid on;
```

You can convert the Fourier Transform `Xf` back to time-domain using the MATLAB command `ifft`:

```
>> xhat = ifft(Xf);
```

The data array `xhat` recovered with the inverse Fourier Transform will be identical to the original waveform `x` (within the limitations of numerical round-off errors).

Problem A.1 [**2 Marks**] For the signal $x(t)$ shown in Figure (1), compute and plot $z(t) = x(t) * x(t)$.

Problem A.2 [**2 Marks**] Using MATLAB, calculate $Z(\omega) = X(\omega)X(\omega)$.

[2] The MATLAB command `plot` draws the graph of the data array by "connecting the dots"; this approach results in a sloped transitions at pulse edges. The MATLAB command `stairs`, on the other hand, draws a staircase graph from the data array, and is therefore more suitable to plot the graphs of rectangular-shaped pulse waveforms. To compare the graphs generated by `plot` and `stairs`, at the MATLAB prompt enter the commands:

```
>> subplot(211); stairs(t,x); grid on; axis([-10,110,-0.1,1.1])
>> subplot(212);   plot(t,x); grid on; axis([-10,110,-0.1,1.1])
```

[3] If `x` is of length `N`, then the result generated by `fft(x)` corresponds to the Fourier Transform of `x` computed at `N` points.

Problem A.3 [3 Marks] Plot the magnitude- and phase-spectra of $z(t)$.

Problem A.4 [3 Marks] Compute $z(t)$ using **time-domain and frequency-domain operations** implemented in MATLAB. Plot both results and compare with the analytic result you determined in Problem A.1. Determine the appropriate time indices for proper labelling of the time-domain plots of $z(t)$. How do the results you generated in MATLAB using time- and frequency-domain operations compare with the analytic result you computed in Problem A.1? Explain which property of the Fourier Transform you have demonstrated.

Problem A.5 [3 Marks] Change the width of the pulse $x(t)$ to 5 while keeping the total length at $N = 100$. Compute the Fourier Transform of the narrower pulse and plot the corresponding magnitude- and phase-spectra. Repeat for a pulse width of 25. Explain the observed differences from the comparison of the frequency spectra generated by the three pulses with different pulse-widths. Explain which property of the Fourier Transform you have demonstrated.

Problem A.6 [3 Marks] Let $w_+(t) = x(t)e^{j(\pi/3)t}$ where $x(t)$ is the original pulse of pulse-width 10 shown in Figure (1). Using MATLAB compute and plot the magnitude- and phase-spectra of $w_+(t)$. Compare the frequency spectra result with those you generated in Problem A.3 and comment on the observed differences. Repeat for $w_-(t) = x(t)e^{-j(\pi/3)t}$ and $w_c(t) = x(t)\cos(\pi/3)t$. Explain which property of the Fourier Transform you have demonstrated.

B. Application of the Fourier Transform

Problem B.1 [4 Marks]

The data file `Lab4_Data.mat` allows you access to the following data arrays.

`xspeech`: This data array is a 80,000 sample-long row vector representing 2.5 sec of a speech signal from a radio broadcast sampled at 32 kHz. The signal has a bandwidth of 3.5 kHz. You can observe the magnitude spectrum of the speech signal using the utility function `MagSpect.m`.

`hLPF2000`: The unit impulse response function of a lowpass filter with passband [0, 2.0] kHz. You can plot this function in time domain or alternatively you can display its magnitude spectrum using the utility function `MagSpect.m`.

`hLPF2500`: The unit impulse response function of a lowpass filter with passband [0, 2.5] kHz. You can plot this function in time domain or alternatively you can display its magnitude spectrum using the utility function `MagSpect.m`.

`hChannel`: The unit impulse response function of a bandpass communications channel. You should observe the channel characteristics by displaying its magnitude spectrum using the utility function `MagSpect.m`.

You may find it convenient to use the following utility functions to complete this part of the lab assignment:

`osc.m` An oscillator function which generates user-specified number of samples from $\cos 2\pi F_0 t$ at user defined frequency F_0. Type `help osc` at the MATLAB prompt for further information.

`MagSpect.m` A utility function that simplifies the process of generating and displaying the magnitude spectra of signals and systems used in Part B. Type `help MagSpect` at the

MATLAB prompt for further information. (**Note:** the utility function `MagSpect.m` is based the MATLAB function `fft` you used in Part A; it does not provide any new functionality, but rather it simplifies the process of displaying the magnitude spectra for signals and systems used in this part of the lab assignment.)

Study the frequency-domain (magnitude only) characteristics of the signal `xspeech`, the lowpass filters defined by `hLPF2000` and `hLPF2500`, and the transmission channel defined by `hChannel`. Design a coding system that will allow the transmission of the signal `xspeech` over the channel. Also design the corresponding decoder to recover `xspeech` from the channel output. Implement both the coder and the decoder in MATLAB. Provide block diagrams of the coder and the decoder you designed and explain the rationale of your design.

You can listen to the speech signals generated at any given point of your decoder or decoder design using the command `sound(xaudio,32000)` where `xaudio` is the data array you want to listen to.