Laboratory 02
ECE 478/ECE 570 – Principles of Communication Systems

Date: 02/10/2025
Section: Representation of signals and systems
Policy: No make-up labs will be given.
Submit a written report with all MatLab codes, plots, and answers

1 Numerical computation of the Fourier transform

In the first section of classes, we learned about the mathematical representation of signals and systems by using the Fourier transform theory and linear time-invariant systems. For efficient numerical computation of Fourier transforms, there exists a class of algorithms called fast Fourier transforms (FFT).
The FFT is derived from the discrete Fourier transform, where both time and frequency are represented in discrete form. In fact, the discrete Fourier transform provides an approximation to the Fourier transform. In order to properly represent the information content of the original signal, a special care needs to be taken in performing the sampling operation involved with discrete Fourier transforms. We are going to learn the theory behind sampling in detail in later sections in our classes. However, for the purpose of this lab, it is sufficient to understand that for a given band-limited signal, the sampling rate should be greater than twice the highest frequency component of the input signal. Furthermore, whenever the samples are spaced uniformly by T_s second, the spectrum of the signal becomes periodic repeating every $f_s = 1/T_s$ Hz.
Let $g(nT_s)$ denote the sampled version of the time-domain signal $g(t)$. Therefore, sample co-coefficients can be written as

$$g_n = g(nT_s) \tag{1}$$

Then, the discrete Fourier transform (DFT) can be defined as

$$G_k = \sum_{n=0}^{N-1} g_n \exp\left(-\frac{j2\pi}{N}kn\right) \quad \text{for} \quad k = 0, 1, \cdots, N-1. \tag{2}$$

The sequence $\{G_0, G_1, \cdots, G_N - 1\}$ is called the transform sequence.
The inverse discrete Fourier transform (IDFT) of g_n can be defined as

$$g_n = \frac{1}{N}\sum_{k=0}^{N-1} G_k \exp\left(\frac{j2\pi}{N}kn\right) \quad \text{for} \quad k = 0, 1, \cdots, N-1. \tag{3}$$

The DFT and IDFT are transform pairs. and given the data sequence g_n, the transform sequence G_k can be computed by using DFT; and given the transform sequence G_k, the original data sequence g_n can be recovered.

1.1 Fast Fourier Transform algorithms

In DFT, both the input and output consist of sequence of numbers defined uniformly spaced in time and frequency. Therefore, DFT can be implemented in a computationally efficient manner by using a class of algorithms called the Fast Fourier Transforms (FFT). Matlab's inbuilt FFT algorithms can be used to sketch and analyze the spectra of signals.

1.1.1 Example 01: Cosine wave

```
Fs = 200; % Fs is Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f = 10; x = cos(2*pi*t*f); % Generate a cosine wave of f=10 Hz
nfft = 1024; % Length of FFT
X = fft(x,nfft); % evaluate fft, padding with zeros to make length(X) ... is equal to fft
X = X(1:nfft/2); % FFT is symmetric, hence taking the first half
mx = abs(X/nfft); % magnitude of fft of x
f = (0:nfft/2-1)*Fs/nfft;  % Frequency vector
% Generate the plot
figure(1);
plot(t,x);
title('Cosine Wave');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a cosine Wave');
xlabel('Frequency (Hz)');
ylabel('Spectrum');
```

Exercise:

1. Change the frequency f of the cosine wave and observe the shifting of the impulse in the frequency domain. (Hint: Use *hold on* command to plot multiple graphs in the same axes)

1.1.2 Example 02: Rectangular pulse

```
Fs = 200; % Sampling frequency
t = -0.5:1/Fs:0.5; % Time vector of 1 second
T0 = .1; % width of rectangle
x  = rectpuls(t, T0);    % Generate Square Pulse
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that length(X) is equal to $nfft
X = fft(x,nfft);
X = X(1:nfft/2); % FFT is symmetric, hence taking the positive %half
mx = abs(X/nfft); % magnitude of fft of x
f = (0:nfft/2-1)*Fs/nfft; % Frequency vector
% Generate the plots
```

```
figure(1);
plot(t,x);
title('Rectangular Pulse Signal');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Rectangular Pulse');
xlabel('Frequency (Hz)');
ylabel('Spectrum');
```

<u>Exercise:</u>

1. Change the width of the rectangular pulse T_0 in the time domain and observe the corresponding change of the bandwidth in the frequency domain.

1.1.3 Example 03: Exponential decaying pulse

```
Fs = 200; % Fs is Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
a = 10; x = 2*exp(-a*t); % exponential decaying pulse
nfft = 2048; % Length of FFT
X = fft(x,nfft); % evaluate fft, padding with zeros so that length(X) ...
    is equal to fft
X = X(1:nfft/2); % FFT is symmetric, hence taking the positive half
mx = abs(X/nfft); % magnitude of fft of x
f = (0:nfft/2-1)*Fs/nfft;  % Frequency vector
% Generate the plots
figure(1);
plot(t,x);
title('Exponential decaying pulse');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Exponential decaying pulse');
xlabel('Frequency (Hz)');
ylabel('Power');
```

<u>Exercise:</u>

1. Change the variable a of the exponential decaying pulse in the time domain and observe the corresponding change of the bandwidth in the frequency domain.

2. Usning the same Matlab code, generate an exponential pulse. Change the variable a and observe the changes in time and frequency domains. (Hint: Exponential pulse can be written as $\exp(at)$)

1.1.4 Example 04: Symmetric double exponential pulse

```
Fs = 200; % Fs is Sampling frequency
t = -1:1/Fs:1; % Time vector of 2 second
a = 10; x = 2*exp(-a*abs(t)); % Symmetric double exponential pluse
nfft = 2048; % Length of FFT
X = fft(x,nfft); % evaluate fft, padding with zeros so that length(X) ...
% equal to fft
X = X(1:nfft/2); % FFT is symmetric, hence taking the positive half
mx = abs(X/nfft); % Magnitude of fft of x
f = (0:nfft/2-1)*Fs/nfft; % Frequency vector
% Generate the plots
figure(1);
plot(t,x);
title('Symmetric double exponential pulse');
xlabel('Time (s)');
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a Symmetric double exponential pulse');
xlabel('Frequency (Hz)');
ylabel('Spectrum');
```

Exercise:

1. Change the variable a of the symmetric double exponential pulse in the time domain and observe the corresponding change of the bandwidth in the frequency domain.
2. Then, change the code to plot asymmetric double exponential pulse and its' spectrum. (Hint: Asymmetric double exponential function can be written as $\exp(-a|t|)\mathrm{sgn}(\mathrm{t})$)

1.1.5 Example 04: Frequency shifting

```
Fs = 200; % Sampling frequency
t = -0.5:1/Fs:0.5; % Time vector of 1 second
T0 = .1; % Width of rectangle
fc=50; % Carrier frequency
A=1; % Amplitude
x = A*cos(2*pi*fc*t).*rectpuls(t, T0); % Generate RF Pulse
nfft = 1024; % Length of FFT
X = fft(x,nfft);% Take fft, padding with zeros so that length(X)
%is equal to $nfft
X = X(1:nfft/2); % FFT is symmetric, hence taking the positive half
mx = abs(X/nfft); % Magnitude of fft of x
f = (0:nfft/2-1)*Fs/nfft; % Frequency vector
% Generate the plots
figure(1);
plot(t,x);
title('RF Pulse');
xlabel('Time (s)');
```

```
ylabel('Amplitude');
figure(2);
plot(f,mx);
title('Power Spectrum of a RF Pulse');
xlabel('Frequency (Hz)');
ylabel('Spectrum');
```

Exercise:

1. Change the variable f_c of the RF pulse in the time domain and observe the corresponding frequency shift in the frequency domain.

Lab Exercise: Design of RF pulses for wireless communication systems

In this lab exercise, you are asked to use the knowledge that you acquired by conducting the aforementioned Matlab exercises to design a particular pass-band RF pulse, which is suitable for wireless communication purposes.
Suppose that you are given the following base-band signal, which is termed the information-bearing signal.

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

1. Develop a Matlab script to plot the time-domain signal $\Lambda(t)$ and the corresponding amplitude spectrum in frequency-domain.

2. By using this amplitude spectrum, determine the minimum bandwidth requirement of a wireless channel that this base-band information-bearing signal $\Lambda(t)$ would be sent over.

3. Now suppose that you are asked to design a modulation scheme that shifts this base-band signal into an upper frequency of f_c. To this purpose, we learned in the class that we can translate a base-band signal into pass-band by multiplying the former by a cosine-wave with a carrier frequency of f_c as follows:

$$y(t) = \Lambda(t)\cos(2\pi f_c t).$$

   By assuming that $f_c = 1680\,\text{kHz}$, you are asked to develop a Matlab script to plot the time-domain signal $y(t)$ and the amplitude spectral of $Y(f)$ in frequency-domain.

4. By using the amplitude spectral of $Y(f)$, determine the minimum bandwidth requirement for sending $y(t)$ over a communication system.

5. Recall that frequency-spectrum is a scarce resource, it is expensive and shared/allocated among many service providers. To this end, the federal communication commission (FCC) is responsible for allocating frequency-spectrum for communication service providers. These service providers need to purchase a frequency spectrum license with

a minimum bandwidth requirement for a particular communication service from the FCC as frequency-spectrum is extremely expensive.

Thus, an engineer, who is assigned to design a communication system, must pay a special attention to bandwidth requirement of RF pulses. Discuss the global, social, and economic factors if you would choose to design the following RF pulse

$$y'(t) = \Lambda(t/2)\cos(2\pi f_c t),$$

instead of the pulse given by $y(t) = \Lambda(t)\cos(2\pi f_c t)$ in Part 3.

To answer this question, make sure to develop a Matlab script to plot the amplitude spectrum of $Y'(f)$, determine the bandwidth requirement and its implications to global, social, and economic factors as per the guidelines given by FCC.