EEE 391

Basics of Signals and Systems

Fall 2021

Computer Assignment 2

due: 20 December 2021, Sunday, by 23:55 on Moodle

In this assignment, you will see two common daily life applications of the concepts you learned in the signals and systems course. Please follow the instructions for each part carefully and answer the questions.

1. DTMF Signal and Transceiver

Dual tone multi frequency (DTMF) is the name of the standard technique used over analog telephone lines to transmit and receive the information about the dialed phone number. Consider Figure 1:

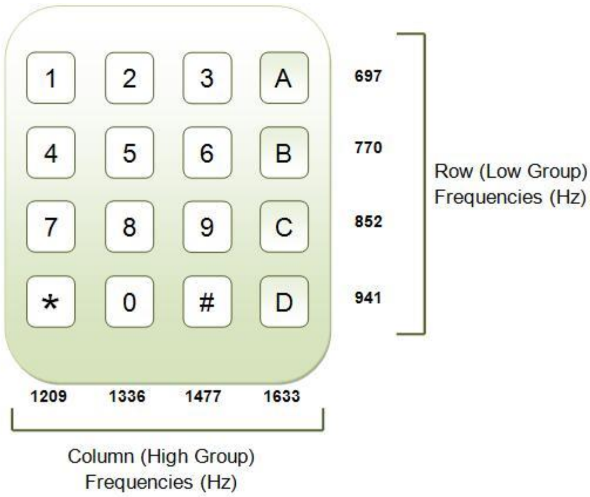

Figure 1. DTMF frequencies

Suppose you press the button for 3 on your phone for 0.5 seconds. Then, the DTMF transmitter on your phone sends the following signal:

$$x(t) = \begin{cases} cos(2\pi 697t) + cos(2\pi 1477t) & for 0 \leq t < 0.5 \\ 0 & otherwise \end{cases}$$

At the receiver, the DTMF receiver examines the incoming signal, tries to understand which frequencies were transmitted, so tries to decide which number is dialed.

1. Write a MATLAB function that prepares the analog signal to be transmitted when a phone number containing only numerical digits 0,1,...,9 is dialed. Assume that for each button, the duration of the transmitted signal is only 0.5 seconds.

Your function should look like:

function [x]=DTMFTRA(Number)

- **Number** of size 1 × N contains the phone number that is dialed. For instance, if the dialed number is 1453, you should have Number = [1 4 5 3].
- **x** contains the samples of the transmitted signal x(t).

Hint: As an example, consider the dialled number is: 2743, the signal should be:

$$x(t) = \begin{cases} cos(2\pi 697t) + cos(2\pi 1336t) & for 0 \leq t < 0.5 \\ cos(2\pi 852t) + cos(2\pi 1209t) & for 0.5 \leq t < 1 \\ cos(2\pi 770t) + cos(2\pi 1209t) & for 1 \leq t < 1.5 \\ cos(2\pi 697t) + cos(2\pi 1477t) & for 1.5 \leq t < 2 \\ 0 & otherwise \end{cases}$$

Note: If N digits are dialed, the duration of the final signal is 0.5N seconds. Of course, in MATLAB we can only compute the samples of x(t). Your code should compute the samples within 0 ≤ t < 0.5N using the sampling period Ts = 1/16384.

2. Test your code by using the signal for your own cellular phone number and listen to it using the MATLAB command **soundsc(x,16384)**. Is the sound familiar to you?

Now, we are going to look at the receiver part. Before starting this part, clear everything in the workspace by typing the command clear all. You will use two functions FT and IFT to find the Fourier transform and the inverse Fourier transform respectively. You need to save these m-files in the same folder that you saved your code.

Consider the last 5 digits of your student ID number as input to your function. First run **x=DTMFTRA(Number)**, where Number contains (i.e. for 21702859, choose Number = [0 2 5 8 9]). Now, suppose we are on the receiver side, x is the received signal and assume we do not know what x includes. You can listen to it typing **soundsc(x,16384)**. We know that the form of the signal that we receive is as follows:

$$x(t) = \begin{cases} cos(2\pi f_{r1}t) + cos(2\pi f_{c1}t) & for 0 \leq t < 0.5 \\ cos(2\pi f_{r2}t) + cos(2\pi f_{c2}t) & for 0.5 \leq t < 1 \\ cos(2\pi f_{r3}t) + cos(2\pi f_{c3}t) & for 1 \leq t < 1.5 \\ cos(2\pi f_{r4}t) + cos(2\pi f_{c4}t) & for 1.5 \leq t < 2 \\ cos(2\pi f_{r5}t) + cos(2\pi f_{c5}t) & for 2 \leq t < 2.5 \\ 0 & otherwise \end{cases}$$

where (f_{ri}, f_{ci}) determines the i th digit, i = 1, 2, 3, 4, 5. To understand the dialed phone number, we need to find the pairs (f_{r1}, f_{c1}), ..., (f_{r5}, f_{c5}).

The Fourier transform operation is a powerful tool to analyze the frequency content of signals, and we will make use of it to understand the frequency content of the received signal.

You have learnt the Fourier transform of a discrete signal so far. The Fourier transform or spectrum of a continuous signal x(t), denoted by X(ω) can be defined as:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)\, e^{-j\omega t} dt \qquad (1)$$

You will learn the details of the formula later in this course. In this assignment, you will use MATLAB functions that we provided. For a particular frequency ω, $X(\omega)$ denotes the contribution of the complex exponential $e^{j\omega t}$ to the signal x(t). As an analogy, $X(\omega)$ shows the weight of $e^{j\omega t}$ we need to use to form x(t).

3. Compute the Fourier transform of x by typing **X=FT(x)**. Then create a frequency array using the following code:

omega=linspace(-16384*pi,16384*pi,16384*2.5+1);

omega=omega(1:end-1);

This piece of code will create a frequency array in angular frequency. Then type **plot(omega,abs(X))**. It will be the magnitude of the Fourier transform of x(t) computed over the grid specified by omega. Include the plot to the report. Examine the figure, in particular, determine the frequencies where you see the peaks. Are the frequencies where the peaks occur the ones used by DTMF transceivers? (Here you should consider the conversion of cyclic frequency (with units of Hertz) to angular frequency (with units of rad/sec)). If yes, can you understand ONLY from this figure what the dialed number is?

4. Define a new signal as:

$$x_3(t) = \begin{cases} x(t) & for 1 \leq t < 1.5 \\ 0 & otherwise \end{cases}$$

This operation can be seen as a multiplication of x(t) by a rectangular signal (shifted properly). Also, the subscript 3 indicates that $x_3(t)$ corresponds to the DTMF signal of the 3th digit of the number sequence. In MATLAB, generate this rectangular signal and by multiplying by x(t), create $x_3(t)$. Make sure that the size of the array for $x_3(t)$ is the same as the size of the array for x(t). Include your code to the report.

Hint: you can define rectangular function as:

$$rect(t) = \begin{cases} 1 & |t| < 0.5 \\ 0 & otherwise \end{cases}$$

5. Compute $X_3(\omega)$ using the **FT.m** function and plot its magnitude against ω. Include the plot to the report. Look again to the frequencies where the peaks occur. This time, can you understand what is the third digit that is dialed?

6. Repeat part 5 for other digits.

7. What is your comments for these two methods? Which one is more practical? (first method means part 3 and the second method means part 4)

2. Echo cancellation

Note: Before starting this part, clear everything in the workspace by typing the command clear all. You will use two functions FT and IFT to find the Fourier transform and the inverse Fourier transform respectively. You need to save these m-files in the same folder that you saved your code.

You are familiar with echo phenomenon and experienced it in your daily life. The aim of this part is to eliminate these echoes from the recording.

1. Create a MATLAB array which includes your speech. For this purpose, you will use a microphone, which can be a webcam microphone or another one. By using this microphone, record your speech for 10 seconds. You can record your voice directly in MATLAB by creating an array, or use another program to save your voice to a file and then read it in MATLAB. For this recording issue, you may need to do some google search. During your recording, the number of the samples per second, in other words, the sampling rate f_s, should be 8192. Generate this array with the name x. Note that the length of the array should be 8192 × 10 = 81920. If the length of the array that you produced exceeds 81920, you can crop it to have a length of 81920. What you speak to the microphone is up to you. After creating x, listen to it in MATLAB in order to make sure that your recording is successful. Include your code, the spoken sentence and the plot of your speech to the report.

Now you will artificially create the signal **y** which suffers from echo. It can be represented as:

$$y(t) = x(t) + \sum_{i=1}^{M} A_i x(t - t_i) \qquad (2)$$

where the summation simulates the environment which causes the echo. M represent the number of the echo, A_i denotes the amplitude of the i^{th}echo and t_i denotes the time delay for the i^{th} echo with $A_i > 0$ and $t_i > 0$. (If there is no amplifier in the environment, we expect $A_i < 1$ due to the propagation power loss.)

In a linear time-invariant (LTI) system, if you if you know the response of the system to an impulse, you can find the response of the system to ANY input. In the time domain, the system can be described as below:

$$y(t) = x(t) * h(t) \qquad (3)$$

where $*$ denotes the convolution operation and h(t) is the impulse response of this LTI system. The Convolution theorem states that the Convolution in time domain equals multiplication in frequency domain. Therefore, in the frequency domain, the equation 3 can be written as below:

$$Y(\omega) = X(\omega)H(\omega) \qquad (4)$$

Where X(ω), H(ω), and Y (ω) are the Fourier Transform of $x(t)$, $h(t)$, and $y(t)$ respectively.

First, generate y from x by assuming M = 5, A_i = 0.8, 0.6, 0.4, 0.2, 0.05 and ti = 0.5, 1, 1.5, 2, 3 seconds. To do so, you must first generate the time variable t by issuing the command t=0: 1/8192 : T-1/8192 ;, where T is the duration of y(t). Don't forget to adjust the lengths of your signals accordingly.

1. Plot x(t) vs. t, y(t) vs. t in separate figures. Also, plot the delayed signals in the same figure (make sure they are distinguishable). Clearly indicate the titles and labels. Listen to y(t) and describe the sound that you listened.

Now, crop the duration of y(t) to 10 seconds. To extract the original signal from y(t), Fourier domain relations (equation 4) will be used. Use the command **Y=FT(y)** to Find the Fourier transform of y(t).

Compute H(ω) over the grid specified by **omega**, which will be generated:

omega=linspace(-8192*pi,8192*pi,8192*10+1);

omega=omega(1:end-1);

2. Compute h(t) typing **h=IFT(H)**. Plot h(t) vs. t and |H(ω)| vs. ω in separate figures. (label your plots)

3. By considering equation 4, compute $X_e(\omega)$, where e indicates estimated X. Compute $x_e(t)$ from $X_e(\omega)$ *by* typing **xe=IFT(Xe)**.

4. Listen to $x_e(t)$. Is the estimated speech different than your original speech? Plot $x_e(t)$ and include the plot to your report. Also include your comments and observations to the report.