ECE 4755

Project 4

Encoding and Decoding Touch-Tone (DTMF) Signals

You must prepare a report containing the answers to each question, include the figures and results. Also, you will need to submit your code scripts.

Background: Telephone Touch Tone Dialing

Telephone touch-tone pads generate *dual tone multiple frequency* (DTMF) signals to dial a telephone. When any key is pressed, the sinusoids of the corresponding row and column frequencies (in Tab. 1) are generated and summed, hence dual tone. As an example, pressing the **5** key generates a signal containing the sum of the two tones at 770 Hz and 1336 Hz together.

The frequencies in Tab. 1 were chosen (by the design engineers) to avoid harmonics. No frequency is an integer multiple of another, the difference between any two frequencies does not equal any of the frequencies, and the sum of any two frequencies does not equal any of the frequencies. This makes it easier to detect exactly which tones are present in the dialed signal in the presence of non-linear line distortions.

Tab. 1: Extended DTMF encoding table for Touch Tone dialing. When any key is pressed the tones of the corresponding column and row are generated and summed. Keys A-D (in the fourth column) are not implemented on commercial and household telephone sets, but are used in some military and other signaling applications.

FREQS	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	**1**	**2**	**3**	**A**
770 Hz	**4**	**5**	**6**	**B**
852 Hz	**7**	**8**	**9**	**C**
941 Hz	*****	**0**	**#**	**D**

DTMF Decoding

There are several steps to decoding a DTMF signal:

1. Divide the time signal into short time segments representing individual key presses.
2. Filter the individual segments to extract the possible frequency components. Bandpass filters can beused to isolate the sinusoidal components.
3. Determine which two frequency components are present in each time segment by measuring the sizeof the output signal from all of the bandpass filters.
4. Determine which key was pressed, **0**–**9**, **A**–**D**, *****, or **#** by converting frequency pairs

back into keynames according to Fig. 1.

It is possible to decode DTMF signals using a simple FIR filter bank. The filter bank in Fig. 1 consists of eight bandpass filters which each pass only one of the eight possible DTMF frequencies. The input signal for all the filters is the same DTMF signal.

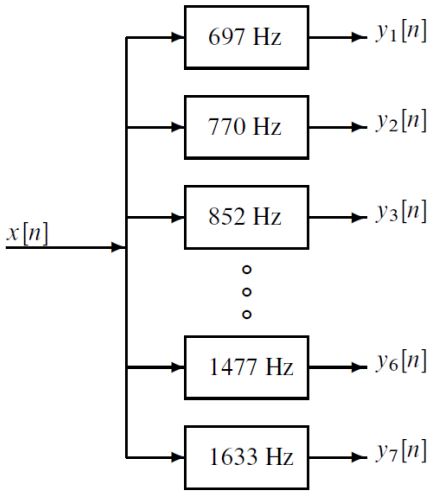

Fig. 1: Filter bank consisting of bandpass filters (BPFs) which pass frequencies corresponding to the eight DTMF component frequencies listed in Tab. 1. The number is each box is the *center frequency* of the BPF.

Tasks:

1- Generate the Touchtone Signals

In this section, you will learn to generate the touchtone signals using the information given in Tab. 1. As stated before, when any key is pressed, the **sinusoids** of the corresponding row and column frequencies (in Tab. 1) are generated and summed.

Problem 1: Run the script "TouchTone_project_part1.m". This script generates all the tones of the keypad. The time-length of each keypad signal is set to 2 seconds. The sampling frequency of Fs = 32768 Hz is used. It also plays the sounds and plots the corresponding signals in time domain. Explain what each line of this code does.

Next Page

2- Detect which Button is Pressed

As you see in the plots of Problem 1, it would be hard to detect the pressed keypad buttons by either listening to the corresponding sounds or looking at the plot of the signals in time domain. In order to detect the main frequency components of each button, we can study the signals in frequency domain (using Fast Fourier Transform) instead of the time domain analysis. The Fast Fourier Transform (FFT) will show what the main frequency components of the signal are. Then, we can refer to Tab. 1 and figure out what button have been pressed.

Problem 2: Run the script "TouchTone_project_part2.m". This script is a modified version of "TouchTone_project_part1.m", which also computes and plots the FFT of each signal. Explain what each line of this code does (you do not need to explain the lines appeared in "TouchTone_project_part1.m").

Problem 3: Determine and explain what each FFT plot shows and corresponds to which keypad button.

3- Understanding Spectrogram and Its Advantage Over FFT for Time-Frequency Varying Signals

A spectrogram is a visual way of representing the signal strength, or "loudness", of a signal over time at various frequencies present in a waveform. Not only can one see whether there is more or less energy at, for example, 2 Hz vs 10 Hz, but one can also see how energy levels vary over time.

Problem 3: Run the script "TouchTone_project_part2.m". Explain how the FFT of a signal is generated. Also, explain what you learned from this code. What is the advantage of using FFT here?

Problem 4: In this problem you will learn the main advantage of the spectrogram. A set of buttons has been pushed and we desire to figure out which buttons have been used. It would be hard to figure out the buttons by analyzing the signal in time domain. Even analyzing the FFT of the whole signal would be a daunting task unless you first divide the signal into different chunks and then take the FFT of each chunk. Run the script "TouchTone_project_part3.m". Explain how the FFT of a signal is generated. Explain what Figure 1 shows. Can you figure out the pushed buttons by just looking at the FFT of the whole signal? Now, take a look at Figure 2, which is the spectrogram of the whole signal. Can you figure out which buttons have been pushed? Explain.

Problem 5: Import the file "Test1.wav" using "audioread" command and the sampling frequency of Fs = 32768 Hz. Write a MATLAB code that plots the signal in time domain, frequency domain (FFT), and the spectrogram of the signal. Include you code and explain what numbers have been pushed in order.

Next Page

Problem 6: Repeat the tasks in Problem 5 for the file “Tone1.wav”.

Problem 7: Repeat the tasks in Problem 5 for the file “Tone2.wav”.

Remark: Include all your codes, reports, and figures.

Good Luck