Assignment 1
Encoding and Decoding Touch-Tone Signals

Introduction

In this lab, we will be using the frequency response of filters to examine the problem solved by telephone touchtone dialing. The problem is this: given a noisy audio channel (like a telephone connection), how can we reliably transmit and detect phone numbers? The solution, which was developed at AT&T, involves the transmission of a sum of sinusoids with particular frequencies. In order for this solution to be feasible, we must be able to easily decode the resulting signal to determine which numbers were dialed. We will see that we can do this easily by considering filters in the frequency domain.

1. Encoding

Whenever you hit a number on a telephone touch pad, a unique tone is generated. Each tone is actually a sum of two sinusoids, and the resulting signal is called a dual-tone multifrequency (or DTMF) signal. Table 1 shows the frequencies generated for each button. For instance, if the "6" button is pressed, the telephone will generate a signal which is the sum of a 1336 Hz and a 770 Hz sinusoid.

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**

These frequencies were chosen to minimize the effects of signal distortions. Notice that none of the DTMF frequencies is a multiple of another. 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.

Your program must do the following

1. Ask the user at first how many keys will be pressed.
2. Then, ask the user to enter the keys.
3. Plot the generated signal in both time domain and frequency domain. The waveform for each key should last 1 sec (sampling frequency = 8 KHz).
4. Play the sound of the generated tones

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. In this step, bandpass filters can be used to isolate the sinusoidal components.
3. Determine which two frequency components are present in each time segment by measuring the size of 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 key names according to Table. 1.

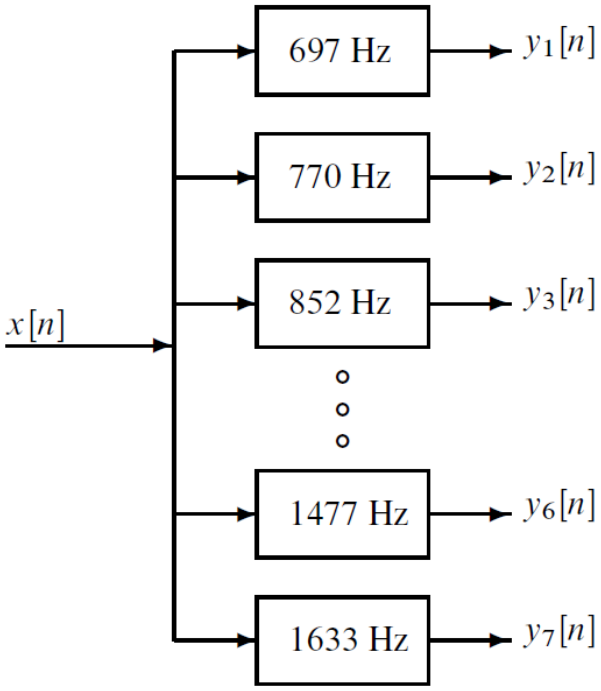
It is possible to decode DTMF signals using a simple filter bank. The filter bank in Fig. consists of eight bandpass filters which each pass only one of the eight possible DTMF frequencies. The input signal for all the filters is the same DTMF signal.

Here is how the system should work:

When the input to the filter bank is a DTMF signal, the outputs from two of the bandpass filters (BPFs) should be larger than the rest. If we detect (or measure) which two outputs are the large ones, then we know the two corresponding frequencies. These frequencies are then used as row and column pointers to determine the key from the DTMF code. A good measure of the output levels is the peak value at the filter outputs, because when the BPF is working properly it should pass only one sinusoidal signal and the peak value would be the amplitude of the sinusoid passed by the filter.

For a simple BPF:

$$h[n] = \beta \cos(\frac{2\pi f_b}{f_s} n) \quad 0 \leq n \leq L-1$$



Where

- L is the filter length
- f_b center frequency of BPF
- f_s sampling frequency
- β is the filter gain : Devise a MATLAB strategy for picking the constant β so that the maximum value of the frequency response will be equal to one.

➢ **When you have completed your filter design function, you should plot the frequency response of all filters for L=40 and L= 80 cases, and then illustrate the effect of changing filter length.**

After passing the DTMF signal through filter bank, the program should detect whether or not a tone is presented after the filter output in order to find out which key was pressed.

Guidelines

- Each Group should consist of 3-4 students.
- Each Group should submit a report showing the running of the Matlab program and explaining any obtained results.
- Discussion will be held during the week starting **Saturday 6th November 2021**.