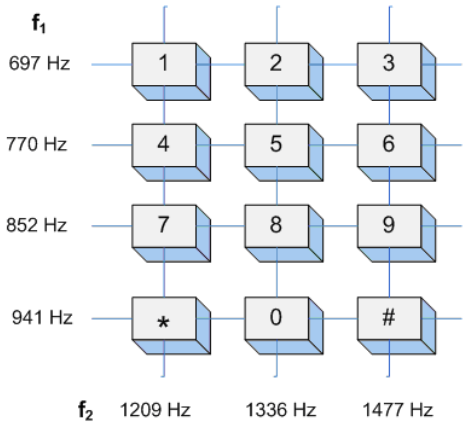
EE 40471: Project 2

Due 27 March, 2017



Telephones communicate dialed numbers through the dual-tone multifrequency (DTMF) scheme, illustrated above. Each 50 millisecond signal includes two sinusoids, one specifying one of the four rows on the keypad and the other specifying one of the three columns. A less frequently used fourth column signifies 'A', 'B','C' and 'D'. Detecting sinusoids in a DT signal is a fundamental task in DSP, with DTMF but one of many applications in which information is encoded in frequency. Your task is to write a Matlab program which will robustly detect the frequencies in each segment of the signal, and write the coded telephone number to the screen.

On the website, you will be supplied with a series of files, each of which contains a coded 10-digit phone number with preceding area code. The digits' signals are of uniform length, and they are sampled at 8192 Hz. Experiment with techniques for finding the two sinusoids. The most direct may be using the fft() function, which will provide a number of samples of the DTFT of your segment equal to the length of the signal. These complex DTFT values are uniformly spaced in $[0, 2\pi)$, and your knowledge of the sampling interval T will allow you to translate the frequency axis into Hertz, or radians/second. You will want to try creative methods, especially when noise corrupts the signal. You may feel free to look into more sophisticated methods of spectral estimation, but with the exception of fft(), you need to write your own code for everything, and are not allowed to simply transcribe any of Matlab's functions for this sort of signal processing problem. Your programs will be tested and graded on a set of clear signals, and several corrupted with white noise, heavy quantization noise, interfering dial tones, interfering voice, etc. The program should prompt the user for a filename, print the decoded number to the screen, and prompt for another file until the user enters 'exit' or 'q'.

The files will be stored in raw binary format ('float'), so you'll want to re-familiarize yourself with fopen(), fread(), etc., in Matlab.