ECE280 - Lab 4: Touch-Tone Phone System

Jerry Worthy (jdw102)

March 18, 2022

I have adhered to the Duke Community Standard in completing this assignment.

Contents

1 Objectives

The objective of this lab was to learn about DTMF phone dialing as well as how to create a simulated DTMF tone generator and software that will decode these tones into their respective number. This is done using MATLAB to create an encoder and Simulink to create a decoder, allowing students to practice using each program's variety of tools.

2 Background

In the past, telephone systems transmitted information using dual-tone multi-frequency signals. Each key on the telephone key pad corresponds to a combination of two sinusoids, known as a dual-tone. Each of the 12 frequencies were chosen so that no two are harmonically-related, as this makes decoding the dual-tone much easier. A MATLAB program was written to simulate a dual-tone encoder, utilizing a function called PhoneNum. The function takes a string of numbers and a fundamental frequency value and produces an array of dual-tones separated by 0.05 seconds of silence. This is done using a 4 by 3 matrix that contains every key form a telephone key pad, and the appropriate frequencies. Each number in the string then has the two corresponding sinusoids added together to produce the dual-tone. Each dual-tone is added to the final array along with a rest. The decoder is created using an array of XCORR blocks in simulink, which allows the array of dual-tone signals to be correlated with each of the 7 different frequencies to determine which numbers were used to encode the signal. This is done by passing the outputs of the XCORR blocks into multiplexers and then graphing the resulting table value onto an oscilloscope.

3 Results and Discussion

1. Include your dtmfdial.m file.

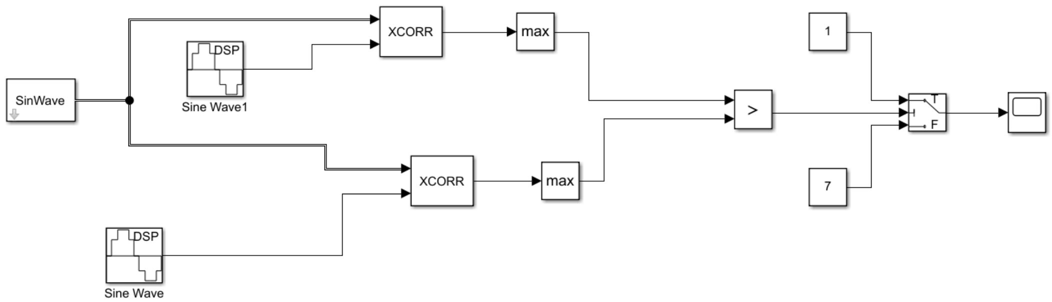
```
function PhoneNum = dtmfdial(KeyNames, fs)
    dtmf.Keys = ['1', '2', '3';
                 '4', '5', '6';
                 '7', '8', '9';
                 '*', '0', '#'];
    dtmf.colTones = ones(4, 1) * [1209, 1336, 1477];
    dtmf.rowTones = [697; 770; 852; 941] * ones(1, 3);
    duration = [0:1/fs:0.5-1/fs];
    rest = [0:1/fs:0.05-1/fs];
    PhoneNum = [];
    for k = 1:length(KeyNames)
        [R, C] = find(dtmf.Keys == KeyNames(k));
        f1 = dtmf.colTones(R, C);
        f2 = dtmf.rowTones(R, C);
        y = sin( 2 * pi * f1 * duration) + sin(2 * pi * f2 * duration);
        rest = 0 * sin(2 * pi * f1 * rest);
        PhoneNum = [PhoneNum, y, rest];
    end
end
```

2. Include a printout of your completed

$$Simple_Corr.mdl$$

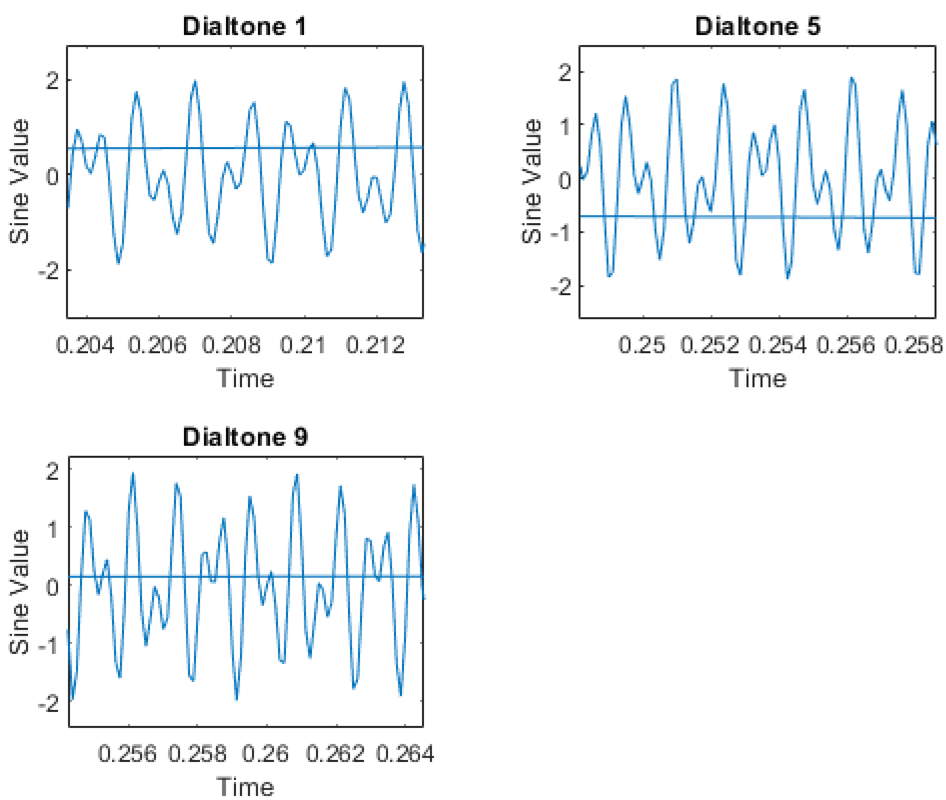
. Explain how this model works. This model works by branching the input signal to be correlated with both a 1477 Hz sine wave and a 697 Hz sine wave. The maximum of each XCORR output is taken to ensure that it is obvious which of the two sine waves has the greatest correlation with the input signal. Both XCORR outputs are then passed into the relational operator which connects to a switch. If the correlation of the top XCORR, the 697 Hz wave, is greater than the bottom XCORR, the 1477 Hz wave, the relational operator outputs a 1, otherwise it outputs a 0. The switch then outputs the top value, 1, if its input is 1, and the bottom value, 7, if its input is 0.

Figure 1: Simple Correlation Model



3. Include plots of the signals corresponding to the keys 1, 5, and 9.

Figure 2: Dial Tone Plots



4. Include plots of the correlation of the keys 1, 5, and 9 to the seven individual DTMF tones.

Figure 3: Dial Tone 1 Correlation

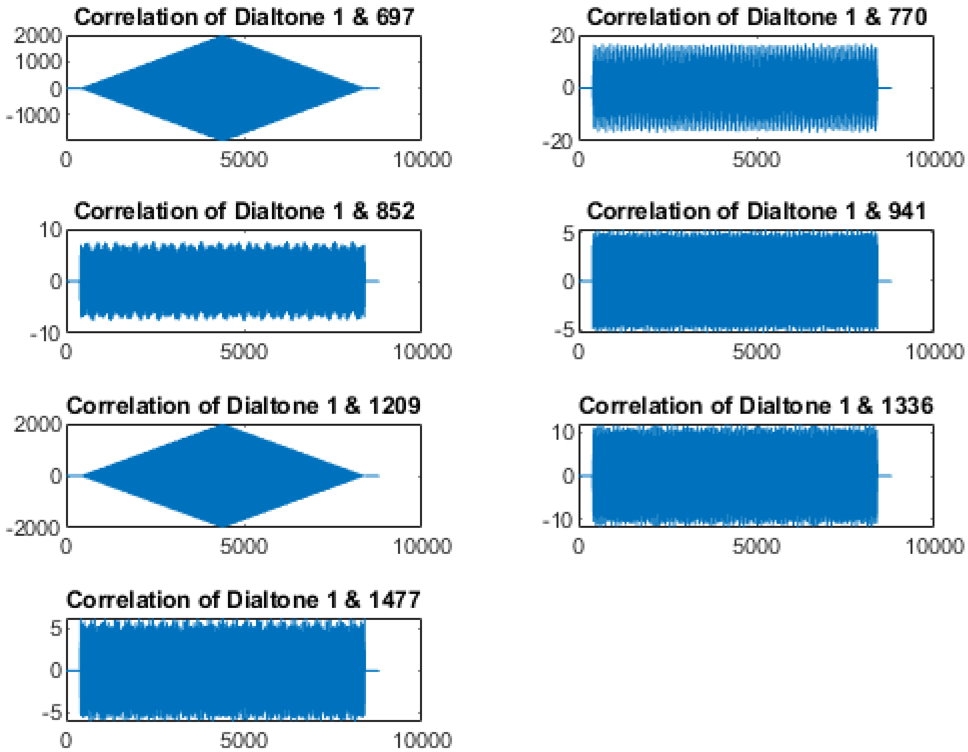


Figure 4: Dial Tone 5 Correlation

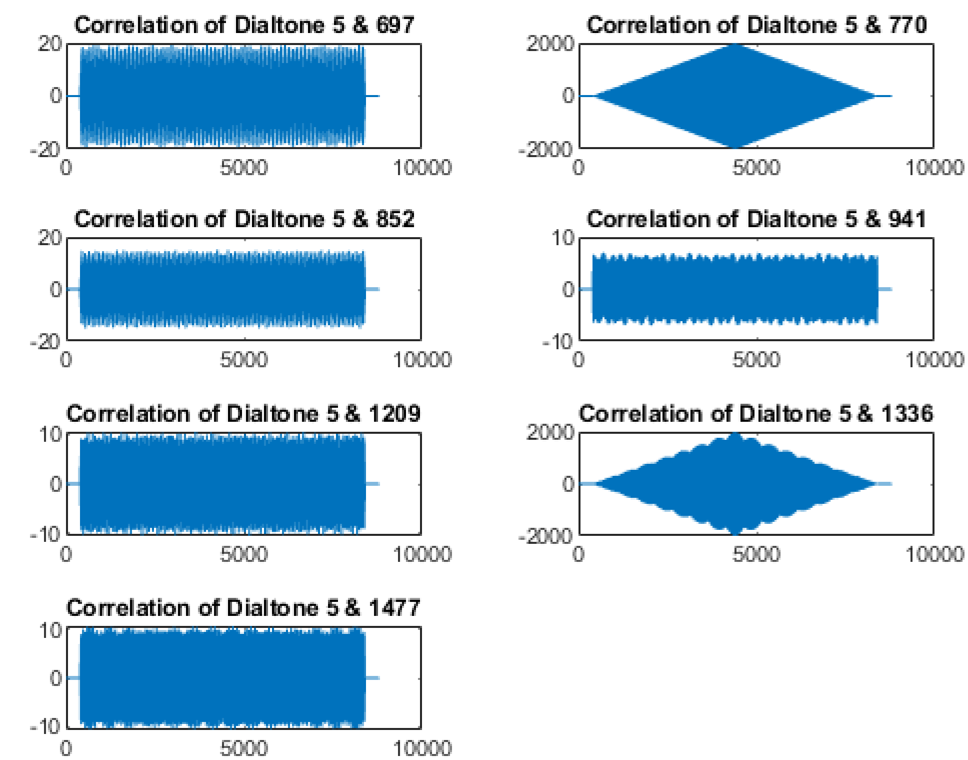
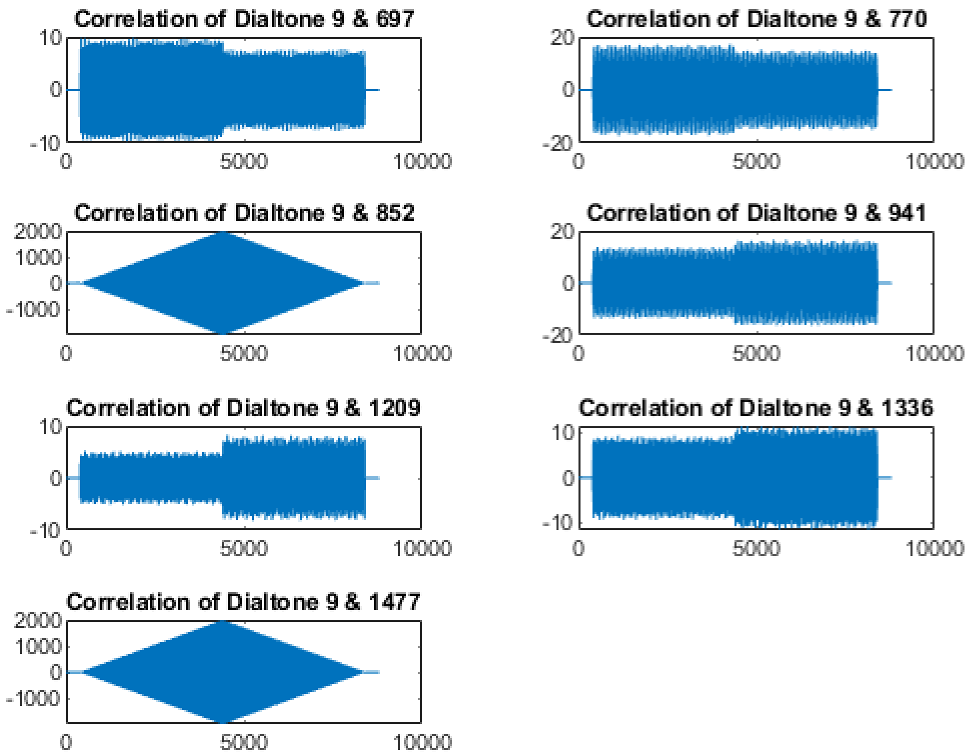


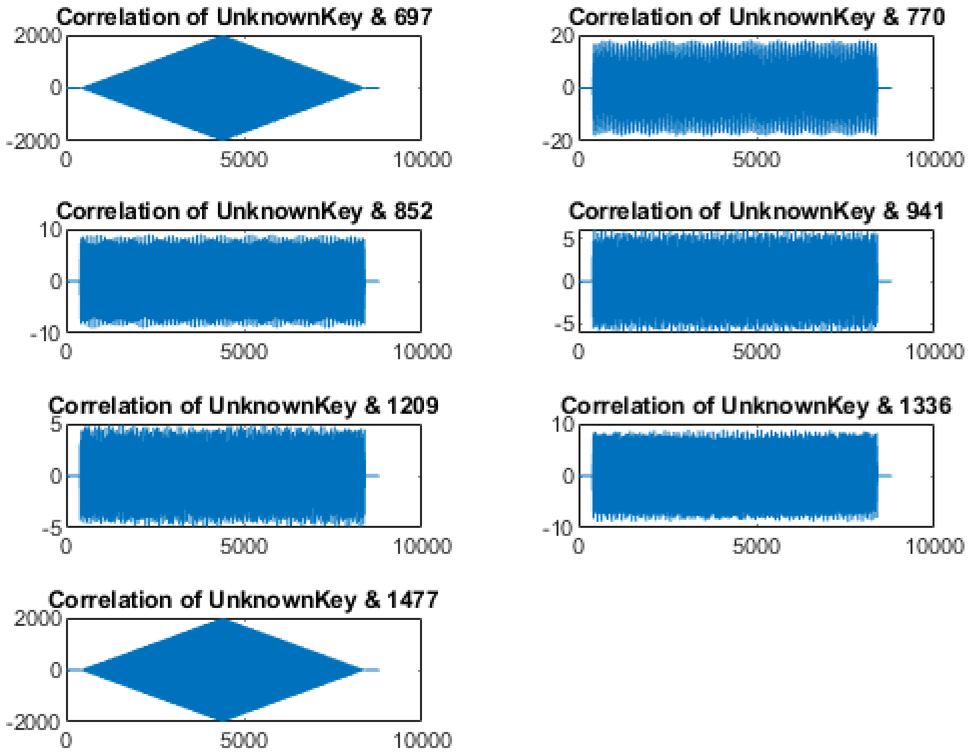
Figure 5: Dial Tone 9 Correlation



5. Discuss what you observed about the correlation plots you generated for the three keys (1, 5, 9). Each of the 3 plots shows that there is a greater correlation with the frequencies that comprise the dual tone. For example, in figure 3, the dual tone 1 is comprised of the frequencies 697 and 1209, and the graph of the XCORR outputs shows that those frequencies have the greatest correlation with it. Their amplitudes go up to 2000 as opposed to the other correlation plots that only go up to around 20.

6. How did you go about determining the value of the UnknownKey? This was done using the plotting method from the previous dial tones. The unknown key was correlated with every frequency and then plotted. This showed that it had the highest correlation with the 697 Hz and 1477 Hz frequencies, which corresponds to the number 3.

Figure 6: Unknown Key Correlation



7. Explain, step-by-step, how the DTMF decoding algorithm presented in

$$DTMF.mdl$$

works. The decoding algorithm works by first correlating the signal with every possible frequency and taking the maximum of this output. These correlation outputs are divided into two multiplexers according to the table, which combines them into a single output for each multiplexer. This allows them to be sorted by a sorting block, with the greatest value of each multiplexer being passed into a 2 dimensional look up table. Because both the indices and values are sorted, the indices of the two highest-correlated frequencies correspond to the row and column of the look up table, which corresponds to the numbers on the key pad. Essentially, the two indices of the frequencies with the highest correlation are chosen using the sorting block, and then used to pick out the number from the lookup table and graph it on the oscilloscope.

8. When you used the Scope to decode the phone number, you should have noticed erroneous values that did not correspond to the numbers you dialed. Comment on why those values occur and how you could improve the system so that they are eliminated. When looking at the previous figures, the frequencies that highly correlated with the signal had amplitudes that varied in a diamond-shape. That is, its amplitude peaks in the middle but dies down near the beginning and end of the signal. The errors likely occur when the correct frequency's correlation amplitude falls below the amplitude of the others, leading to an incorrect key being chosen from the sorting block. This could be avoided by instead just sustaining the index for half a second before checking again instead of continuously checking the signal's correlation with the different frequencies.

9. Answer all questions asked in the instructions.

4 Conclusions

This lab was great in showing a practical application of signal analysis. It provided a lot of good practice in writing MATLAB functions and creating simulink models to assist in the analysis and transformation of signals. Learning how simpler models can be combined to create a more complex model with greater function is a useful skill and is evidently applicable to real-world problems such as the dual-tone decoder in phones. The introduction to the XCORR block in simulink was particularly interesting, and there are likely many other applications using this and similar methods of signal analysis.

5 Extension

The use of cross-correlation was particularly interesting in the course of this lab, and there are many other applications of this operation outside of dual-tone decoding. One example is its use in drawing conclusions from brain wave signals. In one study, oxygenation levels of hemoglobin in the frontal lobes of children were measured while they played computer games to measure how their brains react to positive and negative stimuli. The researchers used cross-correlation and a similar Fourier series methods between the different oxygenation measurements to determine how closely related to one another they are, and any errors in the data-collection method. Plots of the oxygenation levels and the corresponding cross-correlation plot can be seen in Figures 7 and 8 respectively.

Figure 7: Oxygenation Plot

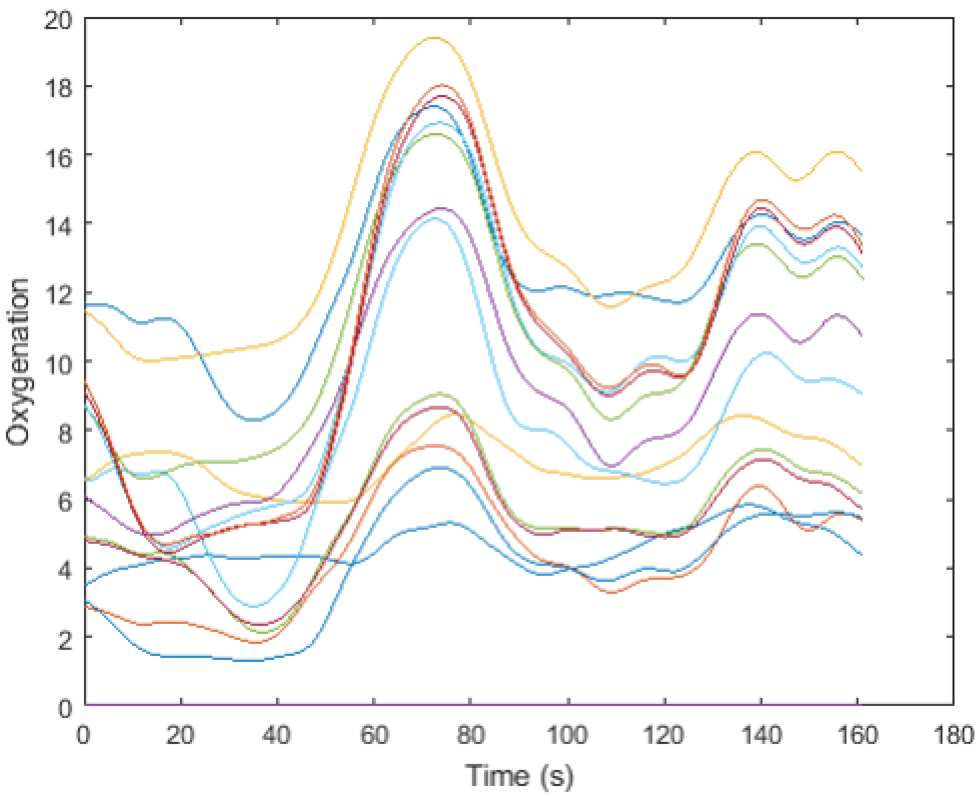
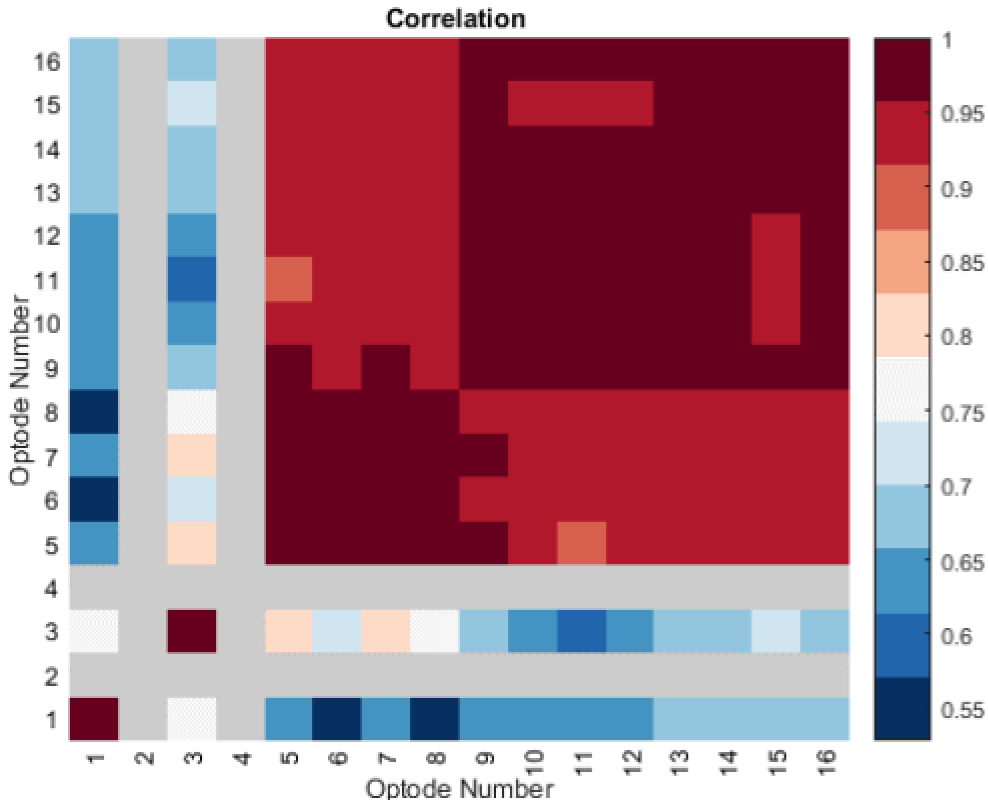


Figure 8:

Oxygenation Correlation Plot


Correlation
Optode Number
Optode Number
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55