ECE280 - Lab 4: Touch-Tone Phone System

Marc Chmielewski (msc68)

March 27, 2020

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

Marc Chmielewski

Contents

1 Objectives

The objective of this laboratory exercise is to explore the design and operation of a standard dual-tone multi-frequency (DTMF) phone dialing system. These systems were introduced in the mid-20th century as a way to allow telephone lines to be controlled using dial-pads as opposed to previous rotary systems, and remained in use up until the early 21st century. The goal of this lab to emulate a DTMF encoder and decoder within MATLAB by utilizing the MATLAB Audio System Toolbox and SIMULINK to create and interpret DTMF tones which can then be converted back and forth to actual telephone numbers.

2 Background

A DTMF system is designed to encode button-presses on the keypad of a telephone into different combinations of two tones which can be subsequently decoded by a DTMF decoder. These tones were specifically chosen by engineers to have no harmonic relationships (that is, they are not multiples of each other) nor two values that are integer multiples of each others. The use of two tones also ensures that caller's voice is not accidentally interpreted as a button press.

To use the system, the number that is to be encoded is entered into the phone which generates paired sinusoids, which are then sent down the line till the reach a switching station where they are decoded and the call is placed.

3 Results and Discussion

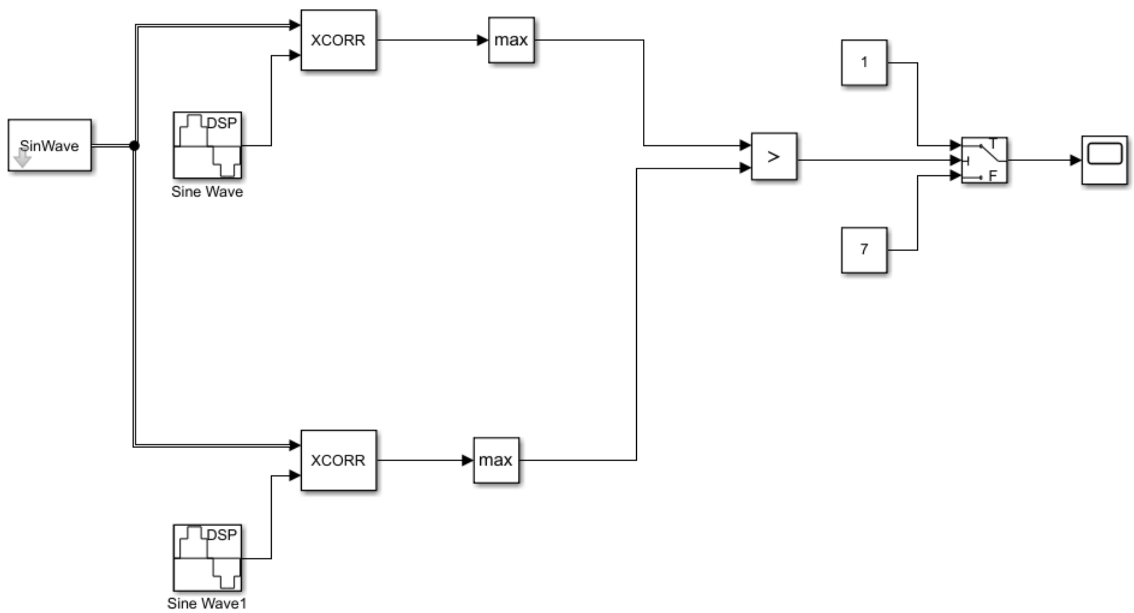
1. Include your dtmfdial.m file.

```
function PhoneNum = dtmfdial(KeyNames, fs)
% DTMFDIAL      Create a signal vector of tones which will dial a DTMF
%               telephone system.
%
%  usage: PhoneNum = dtmfdial(KeyNames, fs)
%  KeyNames = a vector of characters containing valid key names
%  fs = sampling frequency
%  PhoneNum = signal vector that is the concatenation of DTMF tones
%
tone = [0:(1/fs):.5-(1/fs)];
silence = [0:(1/fs):.05-(1/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);
PhoneNum = [];
for i = 1:length(KeyNames)
    RC = find(dtmf.Keys == KeyNames(i));
    curRowFreq = dtmf.rowTones(RC);
    curColFreq = dtmf.colTones(RC);
    % Create the tones
    CurNum = sin(2*pi*curRowFreq.*(tone))+ sin(2*pi*curColFreq.*(tone));
    % Add in the space
    CurNum = [CurNum, 0.*silence];
    PhoneNum = [PhoneNum, CurNum];
end
```

2. Include a printout of your completed SimpleCorr.mdl. Explain how this model works.

Figure 4-1 SimpleCorr.mdl:



This model operates by correlating the sine wave in question with two sine waves. The sine waves which are being correlated against are the DTMF representations of the numbers 1 and 7 and correlation itself is performed by the inbuilt XCORR utility provided in the DSP Toolbox in MATLAB. The max blocks then take the maximum value of the correlation rhombus (assuming the signal in question is well correlated to one of the test signals) and inputs these values into a comparator block which then selects the closer match and graphs the results on the oscilloscope.

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

Figure 4-2 DTMF-1:

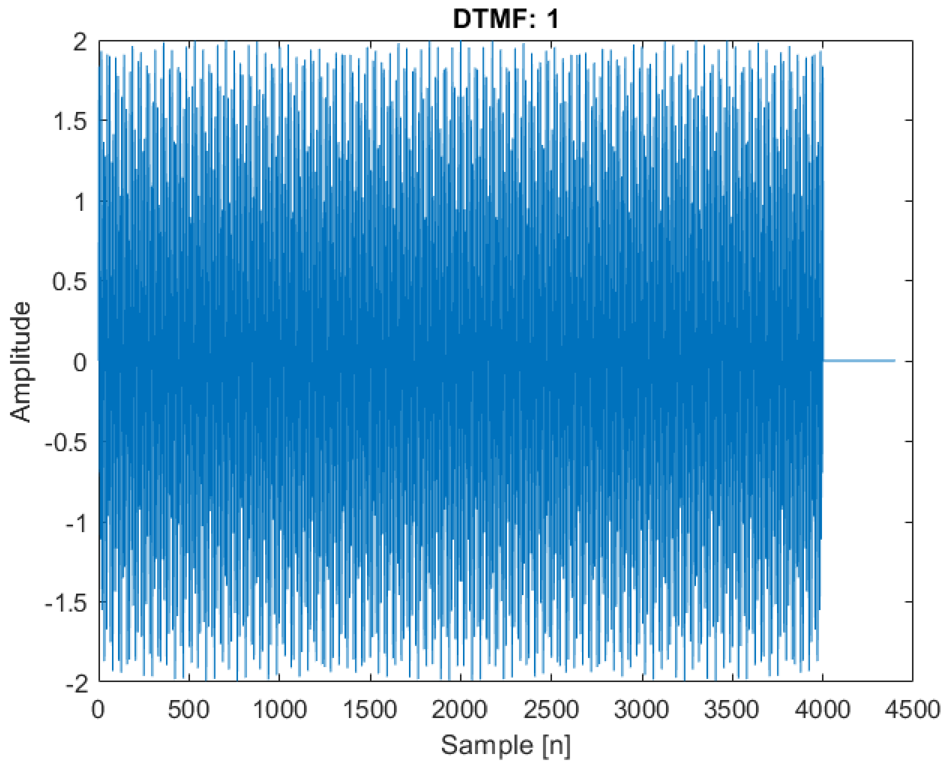


Figure 4-3 DTMF-5:

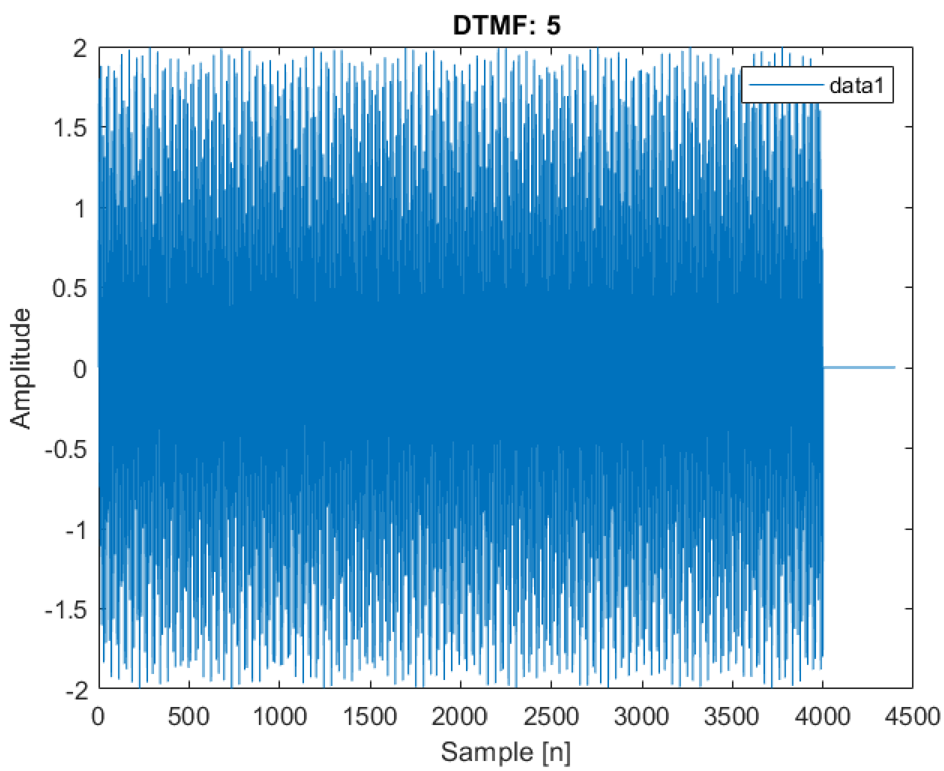
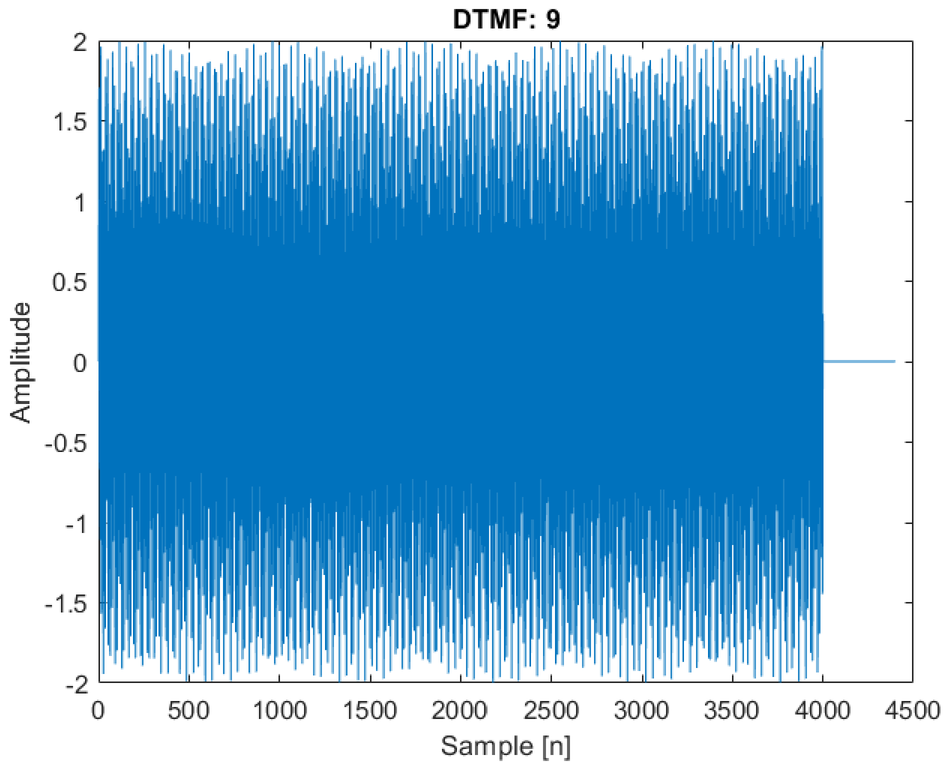


Figure 4-4 DTMF-9:



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

Figure 4-5 DTMF Correlation-1 :

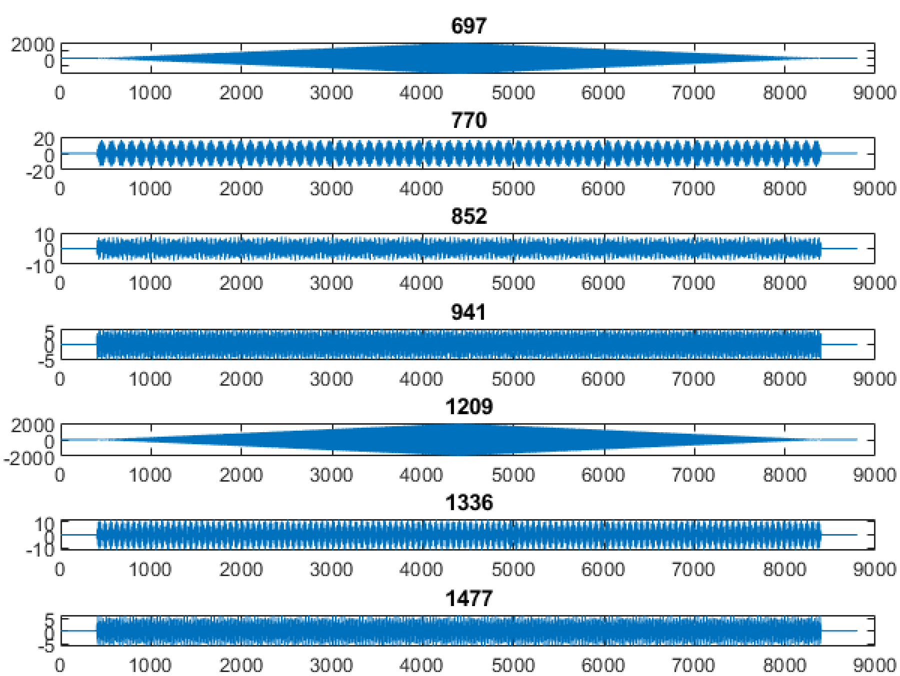


Figure 4-6 DTMF Correlation-5 :

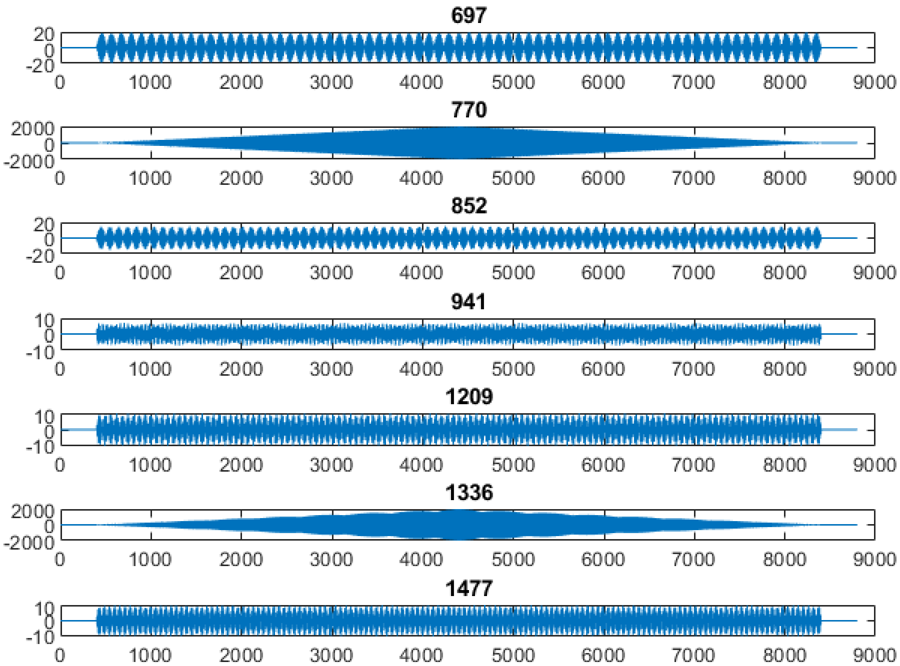
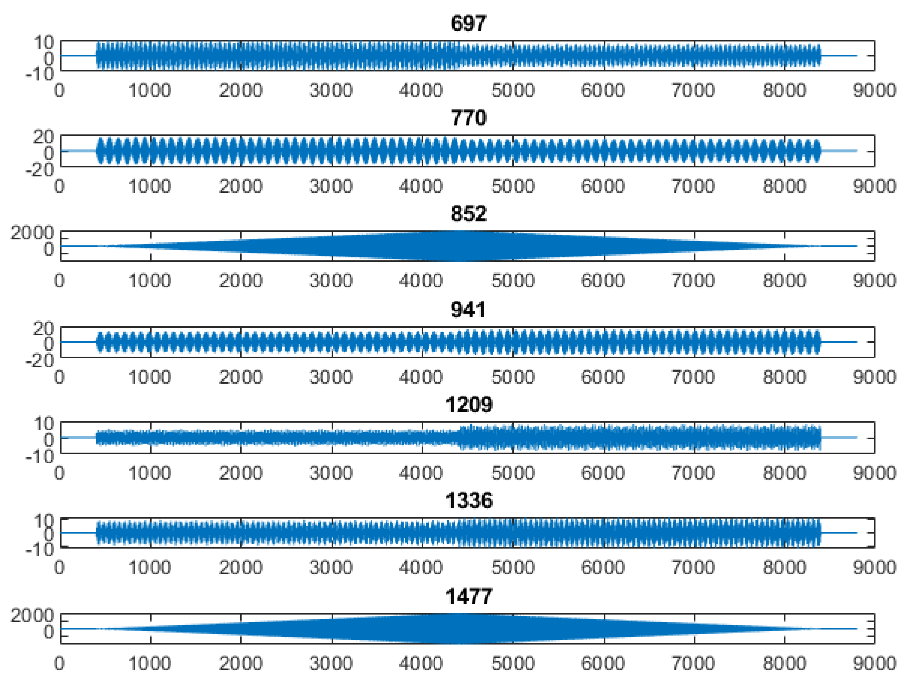


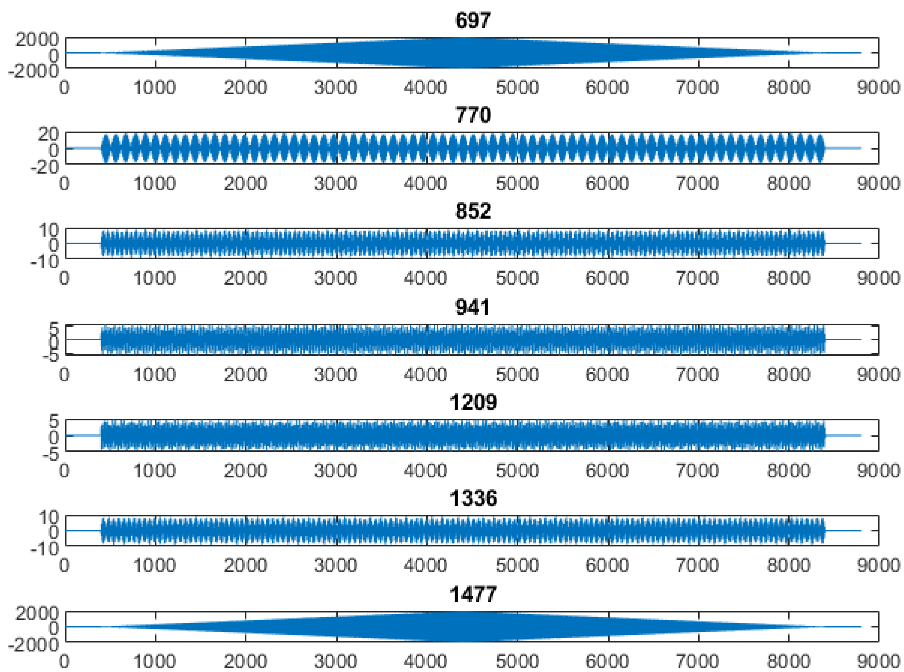
Figure 4-7 DTMF Correlation-9 :



5. Discuss what you observed about the correlation plots you generated for the three keys (1, 5, 9).
   I observed that the two tones which composed the DTMF representation of the number in question formed rhombuses, while the remaining tones formed relatively dissonant, noisy patterns. From this information I was able to verify that the key in question's signal was in fact composed of the two proposed frequencies.

6. How did you go about determining the value of the UnknownKey?
   I used a process similar to the one previously described to determine the constituent frequencies of UnknownKey, and from this information, was able to use the table of DTMF tones to determine its actual value. To decompose this signal, I used the previously established correlation script, and graphed the results, as depicted below in Figure 4-8. From this, I was able to determine that the frequencies 697 and 1477 were the two constituent frequencies. Upon consulting with the table, I was able to determine that the UnknownKey was 3.

Figure 4-8 DTMF Correlation-Unknown :



7. Explain, step-by-step, how the DTMF decoding algorithm presented in DTMF.mdl works.
   The model first takes in the DTMF signal in question and correlates it with each of the constituent frequencies in our DTMF table. This correlation is then analyzed for its maximum value, which is fed into a multiplexer which combines all the signals into a single bus signal. This bus signal is then sorted to determine which component of the mux-ed signal is best correlated and that value is then de-muxed and passed into a direct look up table. This is then repeated for the second set of frequencies and the DTMF input that matches the two frequencies in the table is passed to the scope and graphed as output.

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.
   The correlation artifact is likely a result of the relatively low sampling rate (8000 Hz) that is utilized

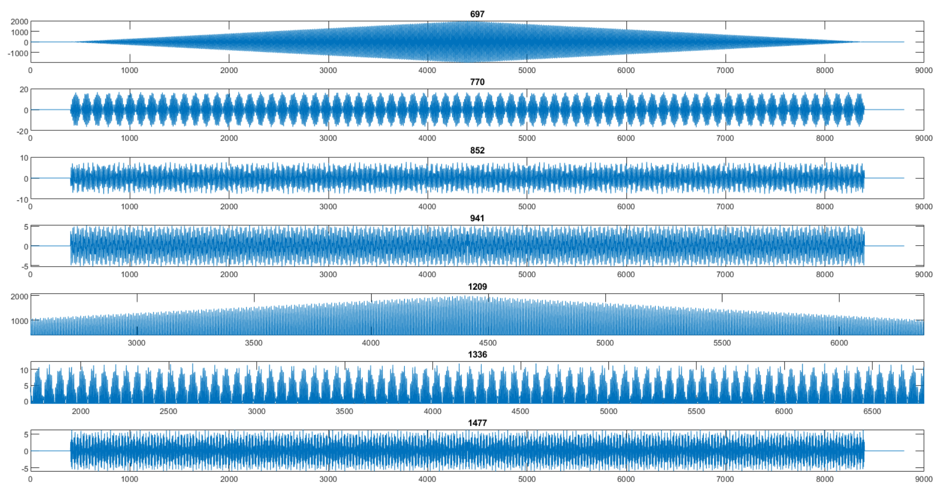
by the system. With more samples, the artifact should disappear as the likelihood of being caught on the edge of a correlation change will decrease dramatically.

9. Answer all questions asked in the instructions.

   (a) For which tones do you expect a large correlation for each key (1, 5, and 9)? Are your results what you expect?
   1: 697 and 1209 Hz
   5: 770 and 1336 Hz
   9: 852 and 1477 Hz
   These values are expected as they are the tones that make up each of the DTMF key values for the keys in question. As demonstrated in figures, 4-5, 4-6, and 4-7, the correlation diamonds are consistent with these expectations.

   (b) Zoom in on one (or more) of the correlation plots. What do you observe? Why does this happen? Include a zoomed-in plot in your report.

Figure 4-9 DTMF Correlation-1 Zoomed :



As depicted in Figure 4-9 the frequencies that compose the signal, in this case the DTMF representation of 1, form diamonds when correlated. Having zoomed in on 1209 Hz in this particular image, you can see that the signal is more or less periodic with a linearly increasing and then decreasing amplitude.

4 Conclusions

In the process of working through this lab, I learned a great deal about the practical applications of correlation in modern telecommunications technology. Correlation is a very useful way to perform spectral analysis on signals with a finite, but potentially large, number of constituent frequencies and provide easily digestible visual feedback regarding the signal's composition. Additionally, correlation offers a way of decoding complex information from a relatively simple encoding scheme, allowing for the efficient transmission of information over a relatively small number of data lines.

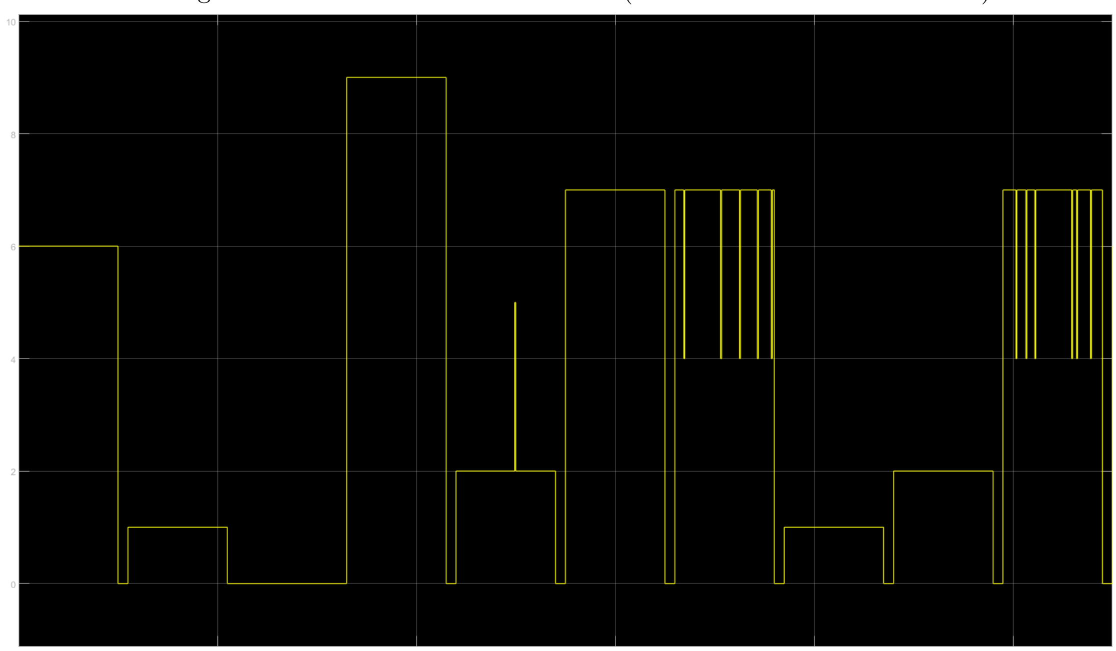
5 Extension

Correlation has a wide variety of applications within the realm of signal processing, oftentimes in the context of audio analysis. One such example of this is musical pitch detection. Many digital tuners utilize correlation circuits to analyze the constituent tones of a note by determining how well cross-correlated it is with a reference pitch.[1] The musician can then utilize this information to make appropriate corrections to the tuning of their instrument. Other musical applications of this technique include autotune, which can use correlation to detect off pitch notes and use software to correct them accordingly.

In a more general communications context, correlation can be used to determine the timing of a digital signal. When a stream of binary information is being sent along a wire more or less continuously, it can be very difficult to discern when a broadcast is beginning or ending. A potential solution to this would be to include a specified start or end sequence, much like the start and end codons in DNA transcription within the cell, and using correlation to determine when the signal matches the sequence the best. Once alignment is achieved, the signal can be processed accordingly. [2]

Additionally, I increased the sampling rate of the DTMF Audio Toolbox SIMULINK model to 16000 Hz in an attempt to remove some of the correlation artifact from the output. While this was partially successful, some artifact did still remain as depicted in Figure 4-10. Future experimentation with additional parameter modulation is recommended for additional improvement long term.

Figure 4-10 DTMF Toolbox 16000 Hz (Tested Number: 610-927-7127):



[1]https://en.wikipedia.org/wiki/Cross-correlation

[2]https://dsp.stackexchange.com/questions/13525/applications-of-correlation-in-signal-processing