Chapter 8

This chapter will discuss the most complicated of the three provided in this text. It involves a classical communications system and relies on many of the things that were learned in the previous two projects.

8.1 Dual-Tone Multi-Frequency (DTMF) Signaling

Using buttons that you press to dial a telephone number was introduced in the mid-1960s by Bell Telephone. Prior to the use of so-called “touch tone” phones the standard dialing method was the *rotary dial*. The rotary dial was rotated to the desired number, and as it relaxed to the un-dialed position, N pulses were generated. The telephone switching equipment counted those pulses to determine the desired digit. Touch tone technology was introduced as a replacement technology because it could operate effectively on lower signal strengths, was more robust, and more user-friendly (i.e. quicker to dial the numbers). As the engineers at Bell Telephone were designing the touch tone system they worked diligently to develop a signaling scheme that was robust and immune to confusion between symbols. The dual-tone multi-frequency standard was the result of their work. This standard uses two sets of frequencies, four high frequencies and four low frequencies in combinations of two (one low and one high). This provides up to 2^4 (or 16) separate symbols, which was enough to handle the 12 buttons that were on the standard touch tone phones of the late 20^{th} century. The specific telephone-related DTMF table is shown in Table 8.1. Although modern cell phones do not use DTMF to encode the desired number being dialed, they still incorporate DTMF on the virtual keypad so that users can navigate automated answering systems that request number presses for menu selection.

Table 8.1. *Telephone DTMF Frequency Assignments*

	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

For example, the number "4" is pressed and the sound recorded in the presence of additive noise. This is shown in the upper panel of Figure 8.1. In this case, the sound was sampled at 8 KHz, meaning that the highest frequency that can be represented in the Discrete Fourier Transform is 4 KHz. The frequency magnitude response is shown in the lower panel of Figure 8.1. The magnitude of the Fourier Transform is simply the square root of the complex magnitude squared. In this case, the peaks are identified at indices that correspond to the frequencies 771 Hz and 1211 Hz. These are not specifically in the table, but we know that the closest values are 770 Hz and 1209 Hz, and this represents the number "4." Because of the sample rate, it is not likely that the computer's processor can identify the frequencies precisely. However, this is one of the primary reasons for developing the set of frequencies chosen by Bell Labs.

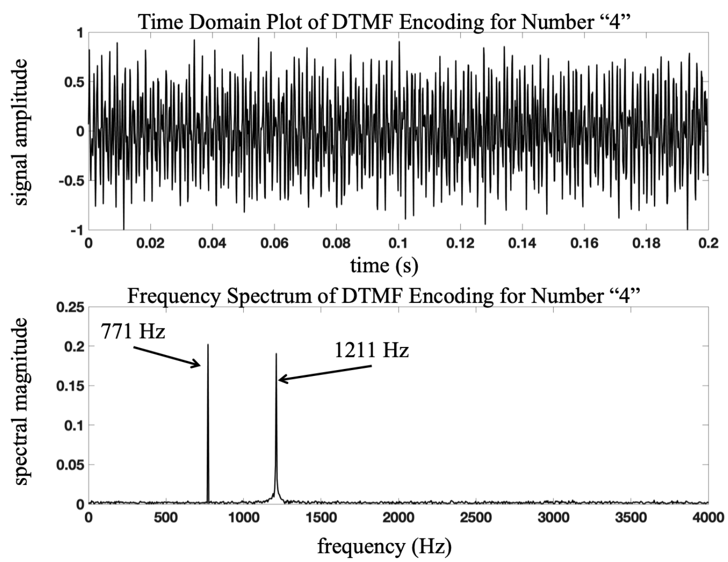

Figure 8.1. ***Time Domain and Frequency Domain for DTMF Encoding of "4":*** *When the number "4" is encoded using DTMF, 770 Hz and 1209 Hz sinusoids combine with the ubiquitous additive noise to produce the voltage signal shown in the upper panel. Sampling 0.2 seconds of this signal at 8000 Hz and computing the spectrum (via FFT), produces the spectrum shown in the lower panel. Searching the frequency magnitude vector shows peaks at 771 Hz and 1211 Hz.*

The reader should consider whether spectral analysis is the only way to determine the two frequencies and, thus, the intended symbol (number). Computationally, using this method involves computing the FFT of some short segment of sampled data, implementing some peak detection algorithm to locate the indices of the two peaks, locating the two corresponding frequencies based on the vector indices of the peaks, and then finally coerce the detected frequency value into a known DTMF value.

Filtering is an alternate method that could be used to determine which number is represented in the signal. One might consider a parallel set of narrow, bandpass filters. If the signal from Figure 8.1 is passed through a bandpass filter centered on 770 Hz, and another bandpass filter centered at 697 Hz, the output of the 770 Hz bandpass filter will have more signal power than the output of the 697 Hz bandpass filter. So in this case, the signal would be passed through seven parallel bandpass filters, the signal power at the output of the filters would be computed, and then the two maximum signal powers would indicate the two frequencies. It may seem a bit excessive using seven parallel bandpass filters, but the difficult to implement coercion step is removed by this method. As an example of the process, there would be four narrow band filters used to identify the low frequency (from the possibilities of: 697 Hz, 770 Hz, 852 Hz, and 941 Hz). The DTMF signal for the 4 button is passed into all four narrow band filters, as shown in Figure 8.2, in parallel. Since the low frequency of the digit 4 is 770 Hz, the power (estimated using the RMS measure) in the 697 Hz filter, the 852 Hz filter, and the 941 Hz filter will all be significantly lower than the power in the 770 Hz filter. A similar approach can be used to determine the high frequency.

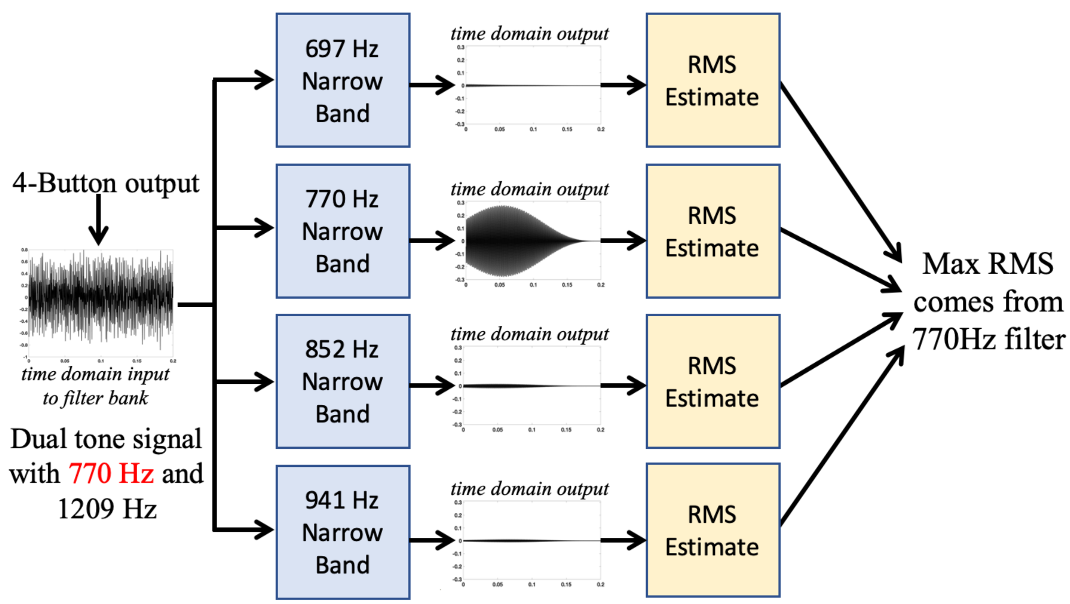

Figure 8.2. ***Frequency Filter Bank to Determine Low Frequency:*** *Using parallel narrow band filters can help identify which low frequency is present. The power of the remaining signal after filtering will be much greater from the narrow band filter that is associated with the tone that is present.*

Another possible detection algorithm implements the *correlation* process. In this algorithm, there is a *gold standard* version of each symbol (12 different gold standards for a DTMF symbol), and the incoming signal is cross-correlated with each of the twelve gold standards. The cross-correlation that produces the highest value corresponds to the encoded symbol. For example, in the upper panel of Figure 8.3 the ideal dual-sinusoid signal for the symbol "4" is cross-correlated with the recorded encoding of the symbol "2."

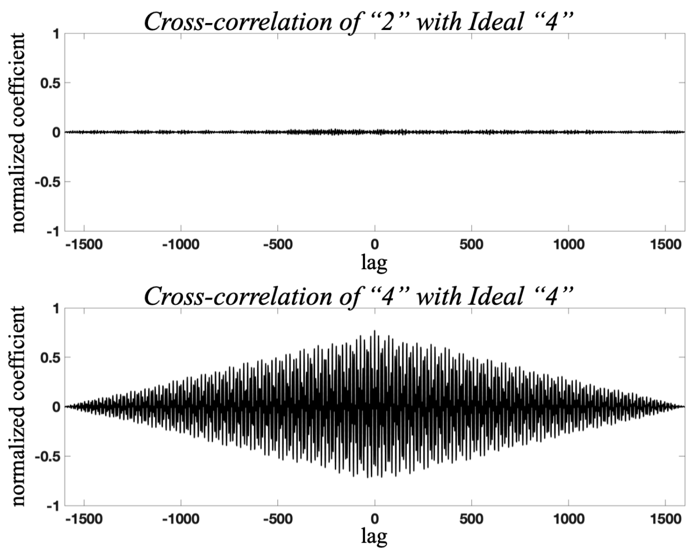

Figure 8.3. ***Cross-Correlation with Gold Standard to Identify Symbol:*** *In the upper panel the incoming signal encodes the symbol 2. It is correlated with the ideal representation of the symbol 4, and the resulting correlation is near 0. In the lower panel, the incoming signal encodes the symbol 4, and is correlated to the ideal representation of the symbol 4. The correlation is significantly higher (0.77). Correlation is not perfect due to the additive noise in the signal.*

The correlation coefficients are normalized (so perfect correlation is one), and it is clear that there is no correlation. In the lower panel of Figure 8.3 the ideal dual-sinusoid for the symbol "4" is cross-correlated with the signal from Figure 8.1. At the *lag* 0 point, the correlation coefficient is about 0.77. The signal-to-noise ratio for both signals is -2 dB. Cross-correlation has the advantage that one does not have to separate low and high frequencies, nor does one have to execute the cross-correlation across the entire signal (meaning a reduced number of computations for each cross-correlation). Of course, there is the memory needed to store 12 separate gold-standard signals and the 12 partial correlations to be executed in order to detect the encoded symbol.

8.2 Project

This project provides an analog, DTMF-encoded, seven-digit telephone number that the student must be able to create and decode. An example of the transmitted signal is shown in Figure 8.4. The higher amplitude sections are the phone numbers. There are smaller amplitude sections between each of the numbers and a longer section of smaller amplitude before the first number is sent.

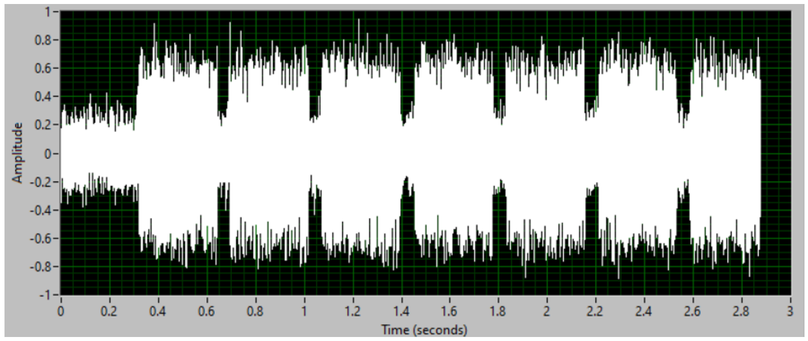

Figure 8.4. ***Example of Transmitted DTMF Signal:*** *The reader should observe that the periods when no button is being pressed still have some level of additive noise. However, the button-press occurrences clearly have greater power and can easily be distinguished.*

The following are some of the requirements for the implementation of this project in the Measurement and Automation course at OU.

- Using the Sine Waveform VI that is in the Signal processing–Wfm Generation Palette or the Simulate Signal Express VI create a VI that can add two sinusoids together and build a 7-digit telephone number. An example of the structure of the 7-digit phone number is shown in Figure 8.4. Approximate the amplitudes of the numbers and spaces so that it looks similar to the number in Figure 8.4. Also, set the length of samples of each number and spaces between each number so that it looks similar to Figure 8.4. The beginning dead zone should be a random length of time that ranges between 0.1 and 0.5 seconds. Since in an actual system you wouldn't know how long it would take for someone to start dialing this is more realistic. Once the person begins to dial the amount of time the buttons are pressed and the time the buttons are released are likely similar in length so you are allowed to set them all the same length and use the approximate length shown in Figure 8.4.

- Have a program that can decode a live phone DTMF signal like the one shown in Figure 8.4 that is sent through the DAQ and also decode a DTMF signal from a file. Have a switch that toggles between live and from a file. The phone number needs to be displayed as a string with a – sign between the 3rd and 4th digit (i.e. 555-1234)

- For this project (both the DTMF creation VI and the decoding VI), the students are required to use a State Machine for this project. The task could be accomplished by dividing the detection steps into distinct processes and developing a state for each process.

In general, the following items should be considered by the student to complete this project:

- Sampling rate, bin size, dynamic range, and channel configuration need to be set appropriately.
- Discriminate between button presses and the time between button presses
- Determine what algorithm (method) to use to decode each button press (correlation, bandpass filters, FFT peak detection and coercion, etc). Multiple algorithms could also be used to minimize the likelihood of an error by looking for agreement between methods.
- Determine a method to decode a button press *only once* and then ignore the button press until it is released.
- Develop a user interface that clearly informs the user of the decoded telephone number

One way to know you have moved from a space (or dead zone) to a number is by breaking up the signal into smaller arrays and looking for a change in RMS. Make sure you don't have too many points in each array as you are searching for the beginning or end of a number. Once you know where the number begins you should take just enough data points from the signal to accurately represent the number so that your algorithm can determine what number it is. Then, you need to start looking for an RMS drop that will signify the next space. Then, the cycle repeats.

References

[1] A. J. Wheeler and A. Ganji, *Introduction to Engineering Experimentation*. Upper Saddle River, NJ: Pearson Education, Inc., 2004.

[2] A. Oppenheim, A. Willsky, and S. Nawab, *Signals and Systems*. Upper Saddle River, NJ: Prentice Hall, Inc., 1983.

[3] N. Bose, *Digital Filters: Theory and Applications*. New York, NY: Elsevier Science Publishing, Inc., 1985.

[4] M. Hayes, *Statistical Digital Signal Processing and Modeling*. New York, NY: John Wiley & Sons, Inc., 1996.

[5] J. Proakis and D. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications, 1st Ed.* New York, NY: Macmillan, 1992.

[6] D. Halliday and R. Resnick, *Fundamentals of Physics, 3rd Extended Ed.* New York, NY: John Wiley & Sons, Inc., 1988.

[7] A. Sedra and K. Smith, *Microelectronic Circuits, 3rd Ed.* New York, NY: Oxford University Press, 1991.

[8] C. Davis, *Electromechanical Systems 1st Ed.* OK: Creative Commons License Oklahoma, 2018.

[9] J. Dyer, "Course notes developed for teaching at the University of Oklahoma from 2008-2018," 2018.