DTMF Decoder

Department of Electrical Engineering
Columbia University

Ang Zhang Hui Shao
az209@columbia.edu hs728@columbia.edu
www.columbia.edu/~az209

Abstract

Dual-tone-multi-frequency (DTMF, also known as touch-tone) is very commonly used. It is also the audible sounds you hear when you press keys on your phone. The technology is increasingly being employed worldwide with push-button telephone sets, offers a higher dialing speed than the traditional dial-pulse signaling used in rotary telephone sets and many other applications. In this project, we present some methods to use Matlab to decode the sound files containing DTMF symbols. We try to find a way that can be used to decode DTMF symbols recorded by various methods.

1. Introduction

DTMF is a tone composed of two sine waves of given frequencies. Individual frequencies are chosen so that it is quite easy to design frequency filters, and so that they can easily pass through telephone lines (where the maximum guaranteed bandwidth extends from about 300 Hz to 3.5 kHz). DTMF was not intended for data transfer; it is designed for control signals only. With standard decoders, it is possible to signal at a rate of about 10 "beeps" (=5 bytes) per second. DTMF standards specify 50ms tone and 50ms space duration. For shorter lengths, synchronization and timing becomes very tricky.

DTMF is the basis for voice communications control. Modern telephony uses DTMF to dial numbers, configure telephone exchanges (switchboards), and so on. Occasionally, simple floating codes are transmitted using DTMF - usually via a CB transceiver (27 MHz). It is used to transfer information between radio transceivers, in voice mail applications, etc. DTMF signaling has also found applications requiring interactive control like in voice main, e-mail, telephone banking, and ATM machines.

Frequency table :

	1209Hz	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

This table resembles a matrix keyboard. The X and Y coordinates of each code give the two frequencies that the code is composed of. Notice that there are 16 codes; however, common DTMF dialers use only 12 of them. The "A" through "D" are "system" codes. Most users won't need any of those; they are used to configure phone exchanges or to perform other special functions.

The frequency table shows how to compose any DTMF code. Each code, or "beep", consists of two simultaneous frequencies mixed together (added amplitudes). Standards specify 0.7% typical and 1.5% maximum tolerance. The higher of the two frequencies may have higher amplitude (be "louder") of 4 dB max. This shift is called a "twist". If the twist is equal to 3 dB, the higher frequency is 3 dB louder. If the lower frequency is louder, the twist is negative.

Transmitting DTMF

Most often, dedicated telephony circuits are used to generate DTMF (for example, MT8880). On the other hand, a microprocessor can do it, too. Just connect a RC filter to two output pins, and generate correct tones via software. However, getting the correct frequencies often requires usage of a suitable Xtal for the processor itself - at the cost of non-standard cycle length, etc. So, this method is used in simple applications only.

2. Decoding DTMF using Matlab

The DTMF decoder computes the DFT samples closest in frequency to the eight DTMF fundamental tones and their respective second harmonics. In addition, a practical DTMF also computes the DFT samples closest in frequency to the second harmonics corresponding to each of the fundamental tone frequencies to distinguish between the human voice and the pure DTMF signal. On the other hand, the DTMF signal generated by the handset or modem has negligible second harmonics.

The DFT length N determines the frequency spacing between the locations of the DFT

samples and the time it takes to compute the DFT sample. A large N makes the spacing smaller, providing higher resolution in the frequency domain but longer computation time. The frequency f_k in Hz corresponding to the DFT index k is given by:

$$f_k = \frac{kF_T}{N}, \qquad k = 0,1....N-1$$

where F_T is the sampling frequency. Then f_k can be checked with the frequency form to get the corresponding dialed number or symbol.

2.1 Decoding using length-N DFT

For an 8-kHz sampling frequency, the best value of the DFT length N to detect the eight fundamental DTMF tones has been found to be 205. So we try to decode the sound file containing DTMF symbols using length-205 DFT. Firstly we separate the whole sound samples into segments of 205 samples each. Then we try to find whether there is a DTMF symbol in each segment. This method is easy to implement and is useful in real-time DTMF processing. Using Goertzel s method to calculate a single DFT sample is the simplest and fast way to get the value of the DFT of desired k value. It works very well in a good environment (relatively high amplitude of dial symbol, high signal to noise ratio). But in many environments, the length-205 DFT cannot get the right result. Because using 205-length DFT, the k value of each fundamental tone frequencies is very close. DTMF decoder are widely used in various environments where the propriety of the channel and the methods used in creating DTMF signals may vary much and the frequencies of the fundamental tone may not strictly match the frequency table. So we must increase the ability of error correction of the decoder. Increase the length of the DFT calculation maybe a good method. Using longer DFT calculation, we can use the maximum value of DFT sample of several k values next to the value calculated using the equation above instead of a single value. Therefore we can correct most of the errors due to the shift of the frequency of the fundamental tone. Although the cost of the calculation will increase as the length of the DFT calculation increases, carefully selecting the length of the DFT calculation according to the environment decoder used in can reduce the cost to the least. We found that when the length of DFT calculation is more than 400, the decoder can decode all of the dial signals correctly in all environments we worked in.

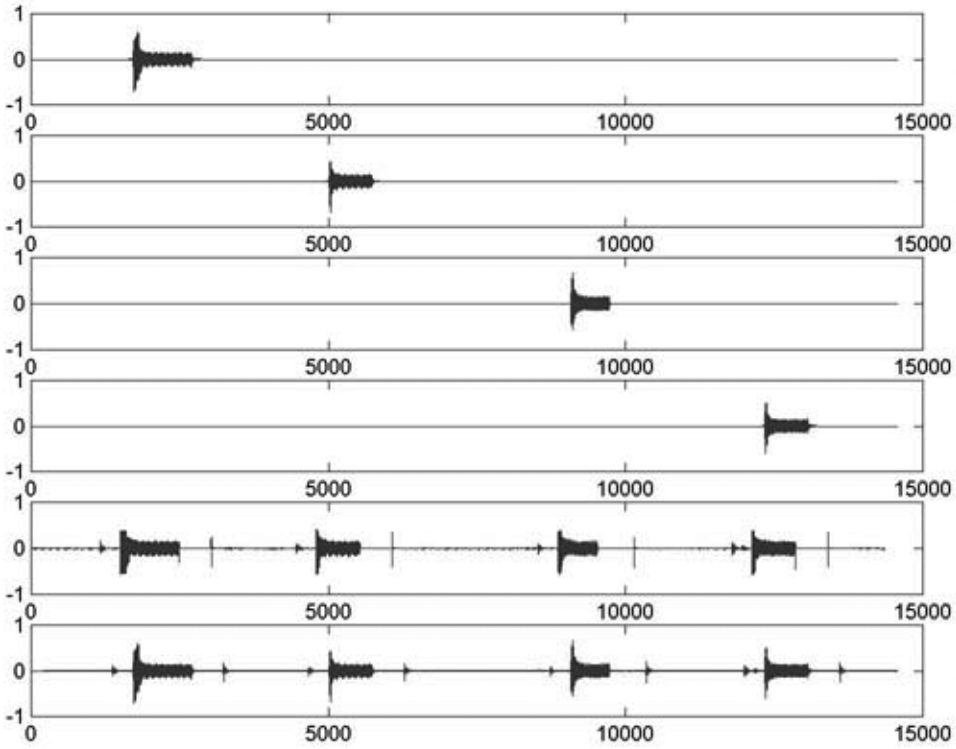
2.2 Detecting the dial signal in time domain

Although the decoding of the DTMF signal is all in frequency domain, if we can firstly detect a dial signal in time domain and only calculate the DFT of the sound clips containing the dial signal, we can use our DTMF decoder in much more Bad environments. For example, if there is a noise signal with high power and unfortunately

it s frequency is in one of the frequencies of the fundamental tone, the decoder we used in 2.2 may not decode the DTMF signal correctly. And the frequency of some noise is in the frequency region of the DTMF tone. When the signal to noise ratio is not high enough, these noise may disturb the decoding of the DTMF signal. These noises are hard to filtered in frequency domain, but they may have obviously different proprieties in time domain. Most kinds of such noise last much less time than dial signals. So we may separate the sound samples into clips containing dial signals according to the energy difference and filter these noise. Another reason we want to detect the incoming DTMF signals is that we can reduce the cost of the calculation of DFT because we only calculate the DFT of the DTMF signal. And detecting of the incoming event is an important subject in sound and speech processing. Once we can detect the events of incoming DTMF signals, we can separate the DTMF signals from the file. Then the record, decode or other processing may be much easier.

The method we use here to detect the DTMF signals sense the change of energy when a DTMF signal comes or leaves. Firstly, we separate sound samples into slots (S_1 S_2 S_3 ... S_N), each of which containing M samples, and then calculate the energy of the S_1 and S_3, S_2 and S_4 ... S_{n-1} and S_{n+1} ... We use a gap here because the energy difference of neighbor slot may be too weak to detect. If the ratio of the energy of S_{n+1} and S_{n-1} is more than a threshold a we can say that a DTMF signal comes and if the ratio is less than 1/a, we can say that a DTMF signal ends. Then we have a clip of sound samples. The length of the clip must reach the minimum length of the DTMF signals. Otherwise we discard it. We only calculate the DFT of the clips to find the dial number. The length of each clip may have different length and we can use constant or various lengths DFT. In 2.1 we use constant length DFT, now we use various length DFT. It may be hard to implement in DSP chip, but it can be more accurate. We may also construct a length 2^N length clip from the original clip and use FFT.

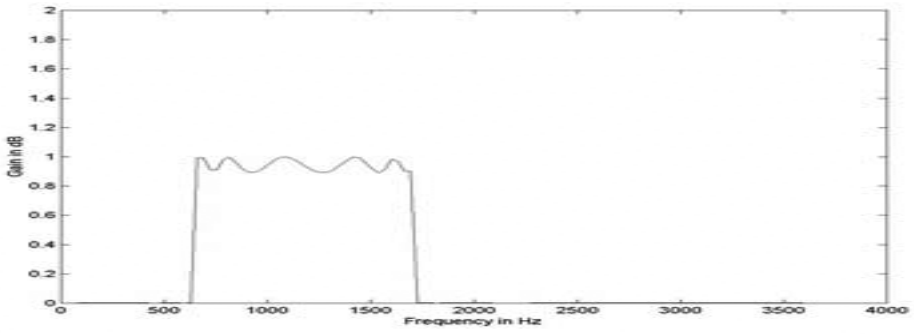
We successfully detecting each DTMF signal and decode it in several sound samples recorded in different kinds of environment.



More result can be found in web site www.columbia.edu/~az209/dspproject

3. Using bandpass filter in DTMF decoder

Bandpass filter is used in analog DTMF decoder to detect the fundamental tone, but in Digital DTMF decoder we can use the methods mentioned above to decode the dial signal. The bandpass filter we used here is to preprocess the sound samples so that we can filter some noises before we detect and decode DTMF signals. We use bandpass elliptical filter here to pass only 650 to 1700 Hz signals. Some outband can be filtered using this method.



4. Conclusion

Applying the methods described above, we tried the sound samples provided on the project web page, which are real recordings from various methods of hooking the computer to the phone line. All of them are decoded successfully. (Some typical results are in the web site www.columbia.edu/~az209/dspproject .)

During the implementation of the DTMF tone decoder, we learned how to apply the DSP concepts from class to the practical requirements. The real cases are not as simple as what we had imagined at the beginning. But by making changes of our model and employing the knowledge in different ways, we solve the problems and realize our design at last.