

## EE432 Fall 2011 Project 10: The DFT and DTMF Decoding (Due: 11/28/2011)

In this project, you will use the DFT (FFT) to help you determine which DTMF tone is being transmitted in a given DTMF signal. You may work this project in groups of no more than 2 people. The signal files you will decipher will each contain only one of the 16 DTMF tones, which are comprised of two sinusoids as in the following table.

		High-Group Frequencies			
		1209Hz	1336Hz	1477Hz	1633Hz
Low-Group Frequencies	697Hz	1	ABC 2	DEF 3	A
	770Hz	GHI 4	JKL 5	MNO 6	B
	852Hz	PRS 7	TUV 8	WXY 9	C
	941Hz	*	OPER 0	#	D

### I. MATLAB Function

Write a MATLAB function called *decodeDTMF* which will decode a DTMF tone. The inputs to the function are the DTMF signal itself (which may have periods of silence before and after, and may be noisy), and the sample frequency of the signal. These periods of silence should be REMOVED in your function, so that the FFT is as accurate as possible (that is, the FFT is only taken of the actual DTMF signal, not including any silence).

Usage:

[digit,duration,fresolution]=decodeDTMF(x,fs)

Here, *x* is the input signal and *fs* is its sample frequency. The function returns three values: (1) *digit*, a character (that is, a 1-character **string**) that is the decoded DTMF tone; (2) *duration*, the time duration of the tone (in sec); and (3) *fresolution*, the frequency resolution of the FFT that is performed...this depends on the sample frequency and the number of samples used in your FFT.

Some important considerations:

- Assume that each signal input to your function contains a single DTMF tone (there are no signals without a DTMF tone in it comprised of two sinusoids, and the signal does not contain more than one DTMF tone).
- Each DTMF signal contains one of the high frequency sinusoids, and one of the low frequency sinusoids. All the low frequency sinusoids are lower than all the high frequency sinusoids (so the two groups of frequencies are separated).
- The FFT of a DTMF signal should have a low frequency peak, and a high frequency peak. The frequencies associated with the two peaks will give indication of which DTMF tone is present.
- The *find* command can be used to find the index of peaks.
- Due to the sample frequency used for the DTMF tones, there most definitely will be leakage in the FFT, so the peaks in the FFT magnitudes will probably NOT correspond exactly to the true frequencies that make up a DTMF tone (but should be close).
- The function will be more accurate if the silence is removed prior to computing the FFT.

## II. Decoding DTMF Tones

1. On the course website, there are 3 files containing known DTMF tones: DTMF=7.wav, DTMF=D.wav and DTMF=1.wav. Download these files and test your function...your function should return a value of '7' for the 1<sup>st</sup> signal, 'D' for the 2<sup>nd</sup> signal and '1' for the third signal (all returned values should be in single quotes). There is silence before and after each of these signals, and they are noisy. They all have different sample frequencies. Compare the signal duration you determine with the true durations for each of these (these are posted on the course website).
2. Your function will now be tested using files that contain 8 unknown DTMF tones. Download the 8 files from the course website ("File01.wav" through "File08.wav"), and use your *FindSignalStart* and *FindSignalEnd* functions to locate the beginning and end of the actual DTMF tone, to fill in the actual time duration of the DTMF tone in seconds. Fill in the time duration in the table below. Fill in the rest of the table with the DTMF tone that your function decoded, as well as the sample frequency of the signal, N, and the FFT resolution. Do NOT assume that they all have the same sample frequency, or last for the same duration. You can only assume that there is a single DTMF tone in each file. There may be silence before and after, and there may be a little bit of noise.

<b>Filename</b>	<b>Duration of actual DTMF portion of signal (sec)</b>	<b>Number of samples for FFT</b>	<b>Sample Freq. (Hz)</b>	<b>FFT Resolution (Hz)</b>	<b>DTMF Symbol</b>
<b>File01.wav</b>					
<b>File02.wav</b>					
<b>File03.wav</b>					
<b>File04.wav</b>					
<b>File05.wav</b>					
<b>File06.wav</b>					
<b>File07.wav</b>					
<b>File08.wav</b>					

3. Your professor will randomly choose one of the unknown DTMF tone files. Demonstrate on your computer that your function correctly decodes this file and returns the decoded value, and have him initial here to show that this has been completed: \_\_\_\_\_

**For this project write up, turn in the well-commented code for your function, and the table above.**