**ENGR-301L: Digital Signal Processing and Applications**
**Trinity College, Instructor: Professor T. Ning**

**Laboratory 5:** Dual-Tone Multi-Frequency (DTMF) Signals Encoding and Decoding
Report due (10/25/2018)

**Lab Description**:

***Dual-Tone Multi-Frequency (DTMF)*** is the generic name for the equivalent of pushbutton telephone of the Bell System's TouchTone. DTMF signaling has been replacing dial-pulse signaling in telephone networks worldwide. In addition to telephone call signaling, DTMF is also becoming popular in various applications of interactive, remote control, such as the telephone banking or electronic mail systems where users can select options by sending DTMF signals through telephone communication. The DTMF signaling standard is also known as ***TouchTone*** or ***MFPB*** (Multiple Frequency, Push Button). TouchTone was originally developed by Bell Labs and used by AT&T and it has become the current standard. The current standard tone frequencies were designated by the International Telegraphy and Telephony Consultative Committee (CCITT). They are specified by the following matrix

| | | | | |
|---|---|---|---|---|
| **row-fr1** | **1** | **2** | **3** | **A** |
| **row-fr2** | **4** | **5** | **6** | **B** |
| **row-fr3** | **7** | **8** | **9** | **C** |
| **row-fr4** | ***** | **0** | **#** | **D** |
| | **fc1** | **fc2** | **fc3** | **fc4** |

The row and column frequencies are given below:

**fr1 = 697 Hz** **fc1 = 1209 Hz**
**fr2 = 770 Hz** **fc2 = 1336 Hz**
**fr3 = 852 Hz** **fc3 = 1477 Hz**
**fr4 = 941 Hz** **fc4 = 1633 Hz**

That is, each of the DTMF signal can be generated by combining two sinusoids with one row frequency and one column frequency listed above, e.g., the dial tone of digit-4 is coded by two sinusoids (fr2 and fc1), *770 Hz* and *1209 Hz*.

To decode a given DTMF signal, one needs to detect the presence of a valid combination of a row frequency and a column frequency (i.e., **fr** and **fc)**. DTMF signals are usually interfaced via ***codec*** (coder/

decoder) chips or linear analog-to-digital (A/D) converters and digital-to-analog (D/A) converters within the analog domain. Codec chips contain all the necessary A/D, D/A, sampling and filtering circuitry for a bidirectional analog/digital interface.

The resultant DTMF tones are generated mathematically and added together. Actual signals are logarithmically compressed and converted to analog signals. ***Companding*** is the process of logarithmically compressing a signal at the source and expanding it at the destination. This can increase the signal-to-noise ratio (SNR) during transmission. In the digital domain, *companding* can reduce the quantization error (hence increasing the signal to quantization error ratio). The improved SNR index also provides an alternative trade-off for reduced bandwidth requirement. Signals are coded either by *μ-law* or *A-law* compressor or expander to maintain a high *end-to-end* dynamic range while reducing the dynamic range requirement within the communication channel.

The ***μ-law*** algorithm (mu-law) is a companding algorithm, primarily used in North America, Japan and Australia. In practice, $\mu$=255 is used. The equation is given below:

$$F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \qquad -1 \le x \le 1 \tag{1}$$

The *μ-law* expansion is given below*:*

$$F^{-1}(y) = \operatorname{sgn}(y)(1/\mu)[(1+\mu)^{|y|} - 1] \qquad -1 \le y \le 1 \tag{2}$$

The ***A-law*** algorithm is a standard companding algorithm used in Europe, where A=87.7 or 87.6 is used. The equation for A-law encoding is as follows,

$$F(x) = \operatorname{sgn}(x)\begin{cases} \frac{A|x|}{1+\ln(A)}, & |x| < \frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A} \le |x| \le 1 \end{cases} \tag{3}$$

The A-law expansion is performed as the following:

$$F^{-1}(y) = \operatorname{sgn}(y)\begin{cases} \frac{|y|(1+\ln(A))}{A}, & |y| < \frac{1}{1+\ln(A)} \\ \frac{\exp(|y|(1+\ln(A))-1)}{A}, & \frac{1}{1+\ln(A)} \le |y| < 1 \end{cases} \tag{4}$$

**Procedure to recover the signals on the receiving end**:

- Load a file with logarithmically compressed digital 8-bit data
- logarithmically expand it to 16-bit linear format
- compute the periodogram to determine the tonal frequencies
- compare the tone frequencies to determine the corresponding DTMF digit

**Purpose:** To become familiar with the theory of coding/decoding of DTMF signals and utilize Matlab and multimedia equipment to gain hands-on experience of implementing DTMF signal coding/decoding.

**Steps:**

1. Use Matlab to generate desired DTMF signals and perform DFT to confirm the correct tone frequencies. Repeat this step for all possible DTMF signals.

   e.g.,

   ```
   tone1=sin(2*pi*fr*t);
   tone2=sin(2*pi*fc*t);
   dtmfSig=tone1+tone2;
   dtmfMax=max(abs(dtmfsig));
   dtmfSig=dtmfsig/(dtmfmax+0.05); % normalize
   % save dtmfSig as a wave file with sampling frequency and the number of bits in each sample.
   % e.g., wavwrite(dtmfsig,fsamp,nbits,outwave);
   ```

2. Write a program that allows the user to enter a 10-digit (area code) + (local access) telephone dial signal, perform *companding* (with compand function) using $\mu$-*255* law compression, and save the result in a wave file with a sampling frequency of 8000 Hz and 8-bit per sample (T1 standard).
   *note: companding converts 16-bit data into 8-bit length and expanding performs the opposite*.

3. Determine the dial numbers of the following DTMF signals.
   - <u>single digit dialing tone:</u>
     dtmf001.wav
     dtmf002.wav
     dtmf003.wav

- multi-digit dialing tone:

*note: each dial-digit lasts 0.25 sec long (2000 samples)*

testfun1.wav

testfun2.wav

testfun3.wav

testfun4.wav