EE 5410 Signal Processing

MATLAB Exercise 1

Telephone Touch-Tone Signal Encoding and Decoding

Intended Learning Outcomes:

On completion of this MATLAB laboratory exercise, you should be able to

- Generate and decode telephone touch-tone signals
- Understand the impact of additive noise in decoding touch-tone signals

Grouping and Schedule:

- One student per group
- Each student is required to submit a **hardcopy answer sheet** which contains answers to the questions in this manual on or before **28 June 2010**.

Background:

Telephone touch-tone pads generate dual tone multiple frequency (DTMF) signals to dial a telephone. When any key is pressed, the sinusoids of the corresponding row and column frequencies, which are depicted in Figure 1, are generated and summed to give dual tone. As an example pressing the "5" key generates a signal containing the sum of the two tones at 770 Hz and 1336 Hz together, and mathematically, it can be generated as

$$x(t) = \cos(2\pi \cdot 770t) + \cos(2\pi \cdot 1336t)$$

In fact, the frequencies in Figure 1 are chosen to avoid harmonics. No frequency is an integral multiple of another, the difference between any two frequencies does not equal any of the frequencies, and the sum of any two frequencies does not equal any of the frequencies. This makes it easier to detect exactly which tones are present in the dialled signal in the presence of non-linear line distortion.

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

Figure 1: DTMF encoding for touch-tone dialling

Decoding of DTMF signals can be achieved via using a simple finite impulse response (FIR) filter bank which is shown in Figure 2. The filter bank consists of 7 band-pass filters (BPFs) where each filter passes only one of the 7 possible DTMF frequencies.

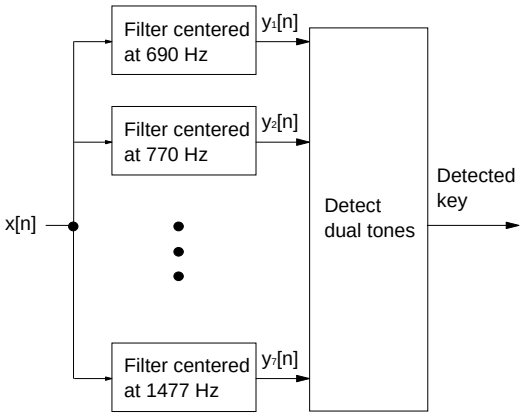

Figure 2: Block diagram for DTMF tone decoding

When the input $x[n]$ to the filter bank is a DTMF signal, the outputs from two of the BPFs should be larger than the rest. If we detect the two largest outputs, the two corresponding frequencies can be found. These frequencies are then used as row and column pointers to determine the key from the DTMF code. A good measure of the output levels can be the peak value at the filter outputs, because when the BPF is working properly it should pass only one sinusoidal signal and the peak value would be the amplitude of the sinusoid passed by the filter.

Procedure:

1. Create a file named “tones.m” with the following MATLAB code:

```
clear all;
fs = 4000;
Ts = 1/fs;
t = [0:Ts:2];
f1 = 400;
f2 = 600;
x=[cos(2*pi*f1*t), cos(2*pi*f2*t)];
soundsc(x,fs);
```

Type `tones` at the command line.

(a) What is the duration of the signal?

(b) What do you hear?

2. Create a file named “tones2.m” with the following MATLAB code:

```
clear all;
x=[];
fs = 4000;
Ts = 1/fs;
t = [0:Ts:0.0205];
f = 400;
for i=1:100
  x=[x,cos(2*pi*f*t)];
end
soundsc(x,fs)
```

and another file named “tones3.m” with the following MATLAB code:

```
clear all;
fs = 4000;
Ts = 1/fs;
t = [0:Ts:2.05];
f = 400;
x=cos(2*pi*f*t);
soundsc(x,fs)
```

Run the two MATLAB programs.
(a) State one difference between the two signals.
(b) Explain the difference.

3. Create a file named “tone.m” with the following MATLAB code:

```
function x = tone(frequency, observation_length);
% x=tone(frequency, observation_length) is used to generate
% a sinusoidal signal x with frequency and observation
% length specified in the arguments.
fs = 4000;
Ts = 1/fs;
t = [0:Ts:observation_length];
x = cos(2*pi*frequency*t);
```

Note that `tone` is a user-defined MATLAB function. Try the following commands: `help tone`, `x=tone(100,0.1)` and `y=tone(1000,0.01)`. Describe the usage for each of the three commands.

4. Write a MATLAB function named `dtmfdial.m`, to implement a DTMF dialer based on the frequency table in Figure 1. A skeleton of `dtmfdial.m` is given as follows:

```
function xx=dtmfdial(keyName)
%DTMFDIAL   Create a DTMF tone
%usage: xx=dtmfdial(keyName)
% keyName = character which is one of the valid key names
% xx = signal vector that corresponds to the DTMF
dtmf.keys = ['1','2','3';
    '4','5','6';
    '7','8','9';
    '*','0','#'];
ff_cols = [1209,1336,1477];
ff_rows = [697;770;852;941];
dtmf.colTones = ones(4,1)*ff_cols;
dtmf.rowTones = ff_rows*ones(1,4);
```

Complete `dtmfdial.m` so that it implements the following:

(i) The input to the function is one of the valid key names.

(ii) The output should be a vector of samples at sampling frequency $f_s = 8000$ Hz containing the DTMF tone. Each DTMF signal is the sum of a pair of unity amplitude sinusoidal signals and the time duration is 0.2s.

(iii) The frequency information is given in two 4×3 matrices, namely, `dtmf.colTones` and `dtmf.rowTones`. To translate a key into the correct locations of the two matrices, the `find` function can be used. An example of using `find` is:

```
[ii,jj] = find('3'==dtmf.keys)
```

(iv) Play the sound of the DTMF tone using `soundsc`.

5. One simple way to implement a band-pass FIR filter is to use the following impulse response:

$$h[n] = \frac{1}{L}\cos(\omega n), \qquad 0 \le n < L$$

where ω is the center frequency of the band-pass filter and L is the filter length. Use MATLAB to generate a band-pass filter with $\omega = 0.2\pi$.

(a) Try the cases of $L = 50$ and $L = 500$. Plot the magnitudes of the frequency spectra of the two filters using `freqz`. An example of using `freqz` is:

```
[a,b] = freqz(h);       %h is the impulse response
plot(b,abs(a));
```

(b) Compute the energies of $h[n]$ for $L = 50$ and $L = 500$. The energy of $h[n]$ is defined as

$$E_h = \sum_{n=0}^{L-1} |h[n]|^2$$

(c) Which filter will give a better DTMF decoding performance, $h[n]$ with $L = 50$ or $L = 500$? Explain your answer.

Note that in general, the impulse response for this simple band-pass FIR filter is

$$h[n] = \frac{1}{L}\cos\left(\frac{2\pi f_b n}{f_s}\right), \qquad 0 \le n < L$$

where f_b is the center frequency of the filter and f_s is the sampling frequency, both in Hz.

6. Write a MATLAB function named `dtmfdetect.m`, to implement a DTMF encoder and decoder in a noisy environment. The requirements of the `dtmfdetect` function are given as follows:

(i) The input to the function consists of one of the valid key names, filter length of the band-pass filters and noise power. That is, `dtmfdetect('1',50,1)` will generate a DTMF tone '1' with $L = 50$ and the tone is corrupted by a zero-mean white Gaussian noise with power of

1. The output will show the result of the detection, namely, displaying a message of `The detected key is 1`.

(ii) Each DTMF signal is the sum of a pair of unity amplitude sinusoidal signals and the time duration is 0.2s with sampling frequency $f_s = 8000$.

(iii) To add a zero-mean white Gaussian noise to the noise-free DTMF tone, you can use the `randn` command. An example of using `randn` is:

```
noise = sqrt(0.1)*randn(1,10);
```

where a zero-mean Gaussian noise sequence of length 10 with power of $\sigma^2 = 0.1$ will be generated.

(iv) To detect the DTMF tone frequencies, you first need to pass the signal to a filter bank of 7 band-pass filters whose center frequencies are 697 Hz, 770 Hz, 852 Hz, 941 Hz, 1209 Hz, 1336 Hz and 1477 Hz. The DTMF tone can then be deduced from the two outputs with the largest energy. An example of producing the output signal given the input and FIR filter coefficients is

```
y=conv(x,h);      % x is the input and h is the filter
                  % impulse response
```

An example of computing the energy of a signal is

```
energy = sum(y.*y);
```

(a) Try your `dtmfdetect` function with various keys, different L ($L = 50$ and $L = 500$) and noise powers ($\sigma^2 = 0$, $\sigma^2 = 1$ and $\sigma^2 = 50$). <u>For each key, perform 10 trials and record the number of correct detection with the following table.</u>

Key	$L = 50$ $\sigma^2 = 0$	$L = 500$ $\sigma^2 = 0$	$L = 50$ $\sigma^2 = 1$	$L = 500$ $\sigma^2 = 1$	$L = 50$ $\sigma^2 = 50$	$L = 500$ $\sigma^2 = 50$
1						
2						
3						
4						
5						
6						
7						
8						
9						
0						
*						
#						

(b) <u>Can your `dtmfdetect` function detect the DTMF tone accurately for all cases? Why?</u>