Tone Encoding and Decoding

CSE1010 Project 3, Fall 2012

Date: 9/20/2012

Name: William Dickson

Section: 009L

TA: Levon Nazaryan

Instructor: Jeffrey A. Meunier

1. **Introduction**

In this assignment you will write a Matlab program to generate (or synthesize) and analyze DTMF sounds, otherwise known as Touch Tone® sounds. You will use Matlab vectors to hold the sound data, plot some of it, and then use the Fourier Transform to take a sound signal and extract its frequency components.

1. **Functions**
   1. **generateTone**

‘generateTone’ creates a sinusoidal tone given a frequency, number of samples per second, and the length of the tone (in that order).

>> tone = generateTone(261.626, 8192, 1);

>> tone(1:10)

ans =

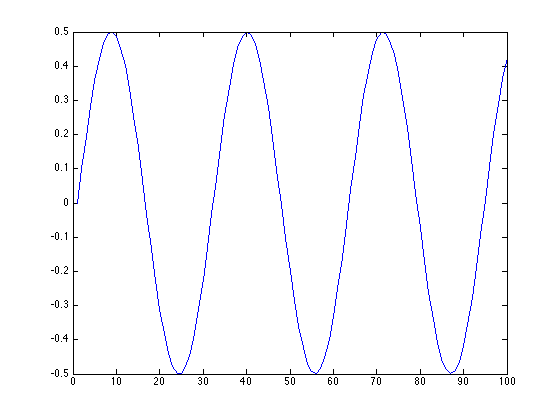
Columns 1 through 8

0 0.0997 0.1953 0.2832 0.3596 0.4217 0.4668 0.4931

Columns 9 through 10

0.4997 0.4862

>> plot(tone(1:100))



* 1. **synthDtmf**

‘synthDtmf’ uses two frequencies to generate two tones using the ‘generateTone’ function with the number of samples equal to 8192 and the frequency equal to 0.3 seconds, then combines the two tones to create a new sinusoidal wave.

>> tone = synthDtmf(261.626,300);

>> tone(1:10)

ans =

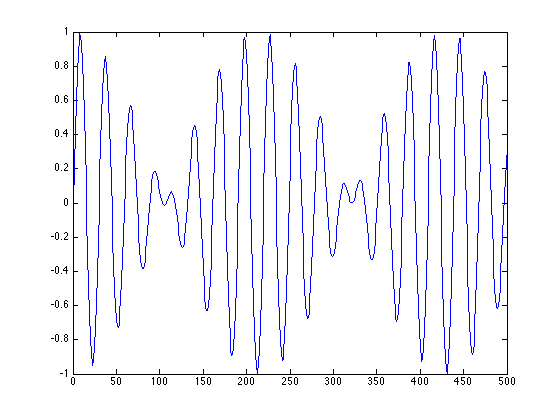
Columns 1 through 5

0 0.2138 0.4176 0.6019 0.7579

Columns 6 through 10

0.8784 0.9579 0.9928 0.9814 0.9245

>> plot(tone(1:500))



* 1. **nameToFrequency**

‘nameToFrequency’ takes the name of a key that you provide and converts it to the two frequencies that correspond to that key.

>> nameToFreq('1')

ans =

697

* 1. **touchTone**

‘touchTone’ uses an input, named ‘key’, and the function, ‘nameToFreq’, and the function ‘synthDtmf’ to change the ‘key’ defined by the user to the corresponding dual tone.

>> [a b] = nameToFreq('5')

a =

770

b =

1336

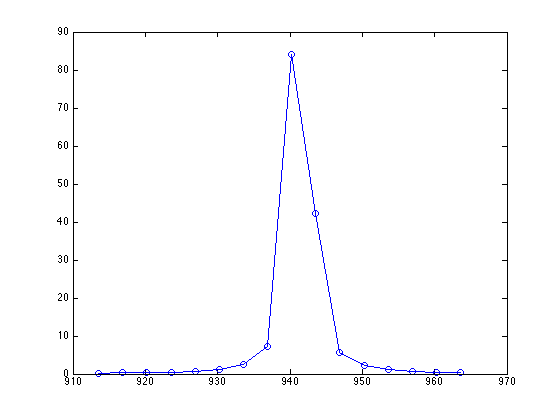
* 1. **timeToFreq**

‘timeToFreq’ uses the ‘singal’ (previously referred to as ‘tone’) and samplesPerSec defined by the user to output the frequency and power of the signal.

>> sig = touchTone('0');

>> [f p] = timeToFreq(sig, 8192);

>> plot(f(275:290), p(275:290), '-o')



* 1. **peaks**

‘peaks’ finds the maximums powers and then returns the frequencies for those maximums by comparing to the values to the left and right of each value.

>> sig = touchTone('0');

>> [f p] = timeToFreq(sig, 8192);

>> peaks(f,p)

ans =

940 1337

* 1. **closeTo**

‘closeTo’ figures out whether or not the number a is within 5 of number b

>> closeTo(5,10)

ans =

1

>> closeTo(5,11)

ans =

0

* 1. **freqToName**

Given frequencies (such as those from ‘peaks’) this function will find the corresponding DTMF key.

>> freqToName(697, 1209)

ans =

1

>> freqToName(697, 1477)

ans =

3

* 1. **encodeAndDecode**

‘encodeAndDecode’ encodes the key\_name (provided by user) to the tone using ‘touchTone’ then converts it to frequency and power using ‘timeToFreq’, then finds the ‘peaks’ and converts the frequencies back to the key name. I also added the option to play the sound if you uncomment a line in this function.

>> encodeAndDecode('5')

ans =

5

>> encodeAndDecode('3')

ans =

3

1. **Source code**

function result = closeTo(a, b)

epsilon = 5;

result = abs(a - b) <= epsilon;

end

function answer = encodeAndDecode(key\_name)

% Set tT equal to the touch tone of the signal from the

% key name. In this order key\_name -> frequencies -> convert

% frequencies to tones -> combine tones

tT = touchTone(key\_name);

% Uncomment following line to create sound of key in this function

% sound(touchTone(key\_name));

% Set f and p equal to the frequency and power of the signal from tT

% with 8192 samples per second

[f p] = timeToFreq(tT,8192);

% Take the frequencies of the peak power values and convert them back

% to the name of the key.

peaks\_out = peaks(f,p);

answer = freqToName(peaks\_out(1),peaks\_out(2));

end

function name = freqToName(f1, f2)

if closeTo(f1, 697)

if closeTo(f2, 1209)

name = '1';

elseif closeTo(f2, 1336)

name = '2';

elseif closeTo(f2, 1477)

name = '3';

elseif closeTo(f2, 1633)

name = 'A';

end

elseif closeTo(f1, 770)

if closeTo(f2, 1209)

name = '4';

elseif closeTo(f2, 1336)

name = '5';

elseif closeTo(f2, 1477)

name = '6';

elseif closeTo(f2, 1633)

name = 'B';

end

elseif closeTo(f1, 852)

if closeTo(f2, 1209)

name = '7';

elseif closeTo(f2, 1336)

name = '8';

elseif closeTo(f2, 1477)

name = '9';

elseif closeTo(f2, 1633)

name = 'C';

end

elseif closeTo(f1, 941)

if closeTo(f2, 1209)

name = '\*';

elseif closeTo(f2, 1336)

name = '0';

elseif closeTo(f2, 1477)

name = '#';

elseif closeTo(f2, 1633)

name = 'D';

end

end

end

function tone = generateTone( frequency , samples\_sec , tone\_length )

seconds = tone\_length;

samplesPerSecond = samples\_sec;

timeDomain = linspace(0, seconds, seconds \* samplesPerSecond);

freq=frequency;

tone = sin(2 \* pi \* freq \* timeDomain);

tone = tone \* 0.5;

end

function [f1,f2] = nameToFreq( key\_name )

if key\_name == '0'

f1 = 941;

f2 = 1336;

elseif key\_name == '1'

f1 = 697;

f2 = 1209;

elseif key\_name == '2'

f1 = 697;

f2 = 1336;

elseif key\_name == '3'

f1 = 697;

f2 = 1477;

elseif key\_name == '4'

f1 = 770;

f2 = 1209;

elseif key\_name == '5'

f1 = 770;

f2 = 1336;

elseif key\_name == '6'

f1 = 770;

f2 = 1477;

elseif key\_name == '7'

f1 = 852;

f2 = 1209;

elseif key\_name == '8'

f1 = 852;

f2 = 1336;

elseif key\_name == '9'

f1 = 852;

f2 = 1477;

elseif key\_name == 'A' || key\_name == 'a'

f1 = 697;

f2 = 1633;

elseif key\_name == 'B' || key\_name == 'b'

f1 = 770;

f2 = 1633;

elseif key\_name == 'C' || key\_name == 'c'

f1 = 852;

f2 = 1633;

elseif key\_name == 'D' || key\_name == 'd'

f1 = 941;

f2 = 1633;

elseif key\_name == '\*'

f1 = 941;

f2 = 1209;

elseif key\_name == '#'

f1 = 941;

f2 = 1477;

else

disp('The key you entered does not exist on the DTMF keypad.')

pause

end

end

function [freq\_values] = peaks(freq, power)

Left = [0 power(1:end-1)];

Right = [power(2:end) 0];

% Determine when the power is greater than both the value to the left

% and right, then get the frequency of those 'peak' locations

locations = power > Left & power > Right;

freq\_values = freq(locations);

freq\_values = round(freq\_values);

end

function dual\_tone = synthDtmf( freq1 , freq2 )

tone1 = generateTone(freq1, 8192, 0.3);

tone2 = generateTone(freq2, 8192, 0.3);

dual\_tone = tone1 + tone2;

end

function [freq power] = timeToFreq(signal, samplesPerSec)

count = length(signal);

y = fft(signal, count);

power = y.\*conj(y)/count;

power = power(1:floor(count/2));

freq = samplesPerSec/count\*(0:(floor(count/2)-1));

end

function dual\_tone = touchTone( key )

[f1 f2] = nameToFreq(key);

dual\_tone = synthDtmf(f1, f2);

end