Digital Signal Processing Lab

Expt. No. 3

DTMF (Dual Tone Multi Frequency or Touch Tone) coder or decoder



By Pranit Dalal

Roll no. 16EC10016

Group 22 (Tuesday)

**AIM:**

Study and analysis of DTMF coder/decoder using Digital FIR filter in MATLAB.

We will learn about:

(a)Functionality of DTMF and its different applications.

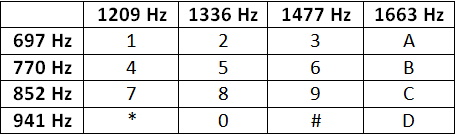
(b)Implementation of Bandpass Digital Filter

(c)DTMF coder/decoder

(d)How to characterise a filter by knowing how it reacts to different frequency components in the input.

**THEORY:**

Dual tone multiple frequency (DTMF) signalling is used in telephone dialling, digital answer machines, and interactive banking systems. DTMF signalling represents each symbol on a telephone touchtone keypad (0–9, \*, #) using two sinusoidal tones, as shown in Figure. When a key is pressed, a DTMF signal consisting of a row frequency tone plus a column frequency tone is transmitted. Keys A–D are not on commercial telephone sets but are used in military and radio signalling applications.



No frequency is an integer multiple of another and the difference or sum of any two frequencies does not equal any of the frequencies. Pressing 4 should generate a tone containing a 770 Hz and a 1209Hz frequency component.

**DTMF coder:**

The coder should give an output consisting 2 frequencies.

x(t) = A1sin(2\*f1\*t) + A2sin(2\*f2\*t)

where in this experiment we used A1 = A2 = 1 without losing the generality.

**DTMF decoder:**

There are several steps in decoding a DTMF signal

1.Divide the time signal into short time segments representing individual key presses

2. Filter the individual segments. Band Pass filters are used to isolate the sinusoidal components

3.Determine which two frequency components are present in each segment by measuring the size of the output signal from all the Band Pass Filters.

4 - Decode which key was pressed, 0-9, A - D, \* or # by converting frequency pairs back into key names according to above given table.

BAND PASS FILTER DESIGN:   
THE L-PONT AVERAGE FILTER IS A LOW PASS FILTER

Its pass bandwidth is inversely proportional to L. It is also possible to create a filter whose pass band is centred around some frequency other than zero. One simple way to do this is to define the impulse response of an L-point FIR filter as

Where L is the filter length, and is the centre frequency that defines the frequency location of the pass band and p is used to adjust the gain in the pass band. So, it is possible to, choose so that the maximum value of the frequency response magnitude will be one. The Band width of the band pass filter is controlled by L: The larger the value of L, the narrower the band width.

**CODE:**

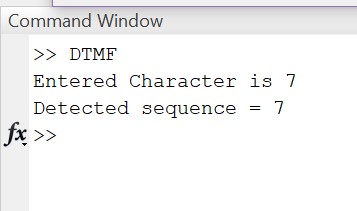
This code encodes a single bit entered in the command window and outputs the time domain signal output consisting of the frequency corresponding to the key pressed. Fourier transform of the signal is also obtained.

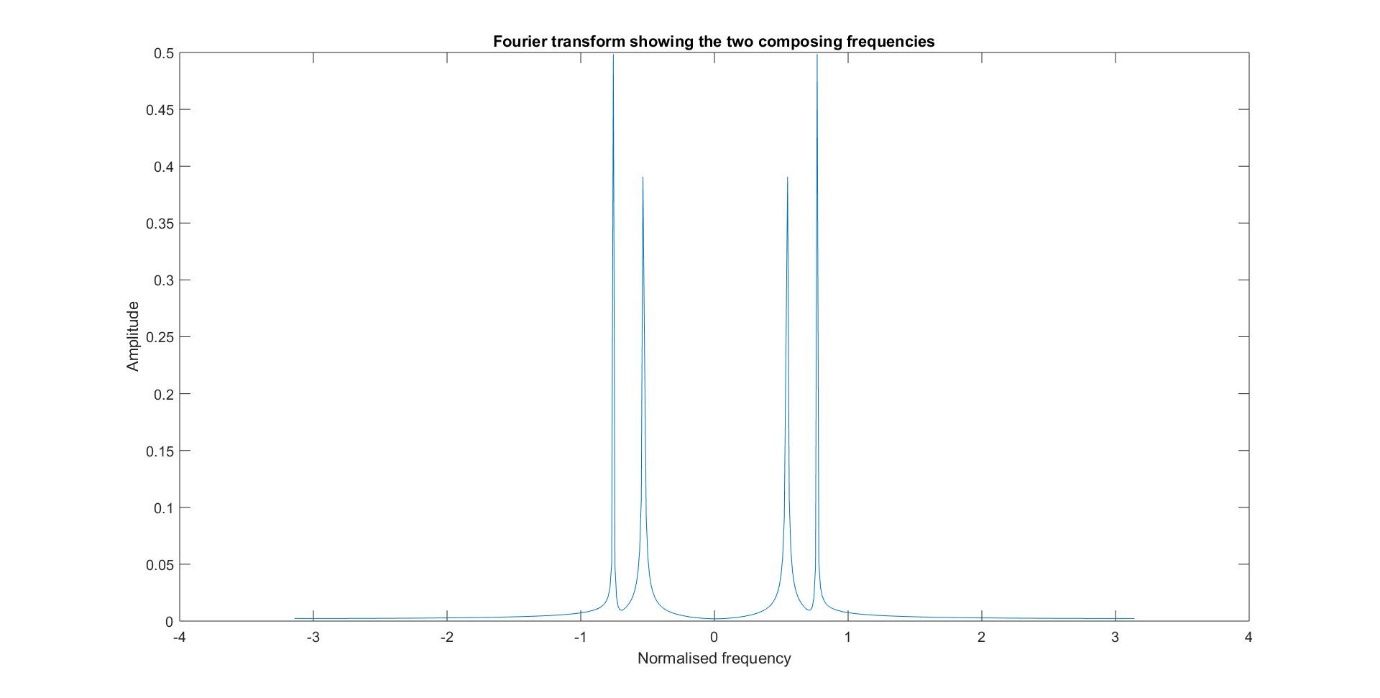
The decoder then returns the key pressed by passing the signal through filters and calculating the index of max rms value of output.

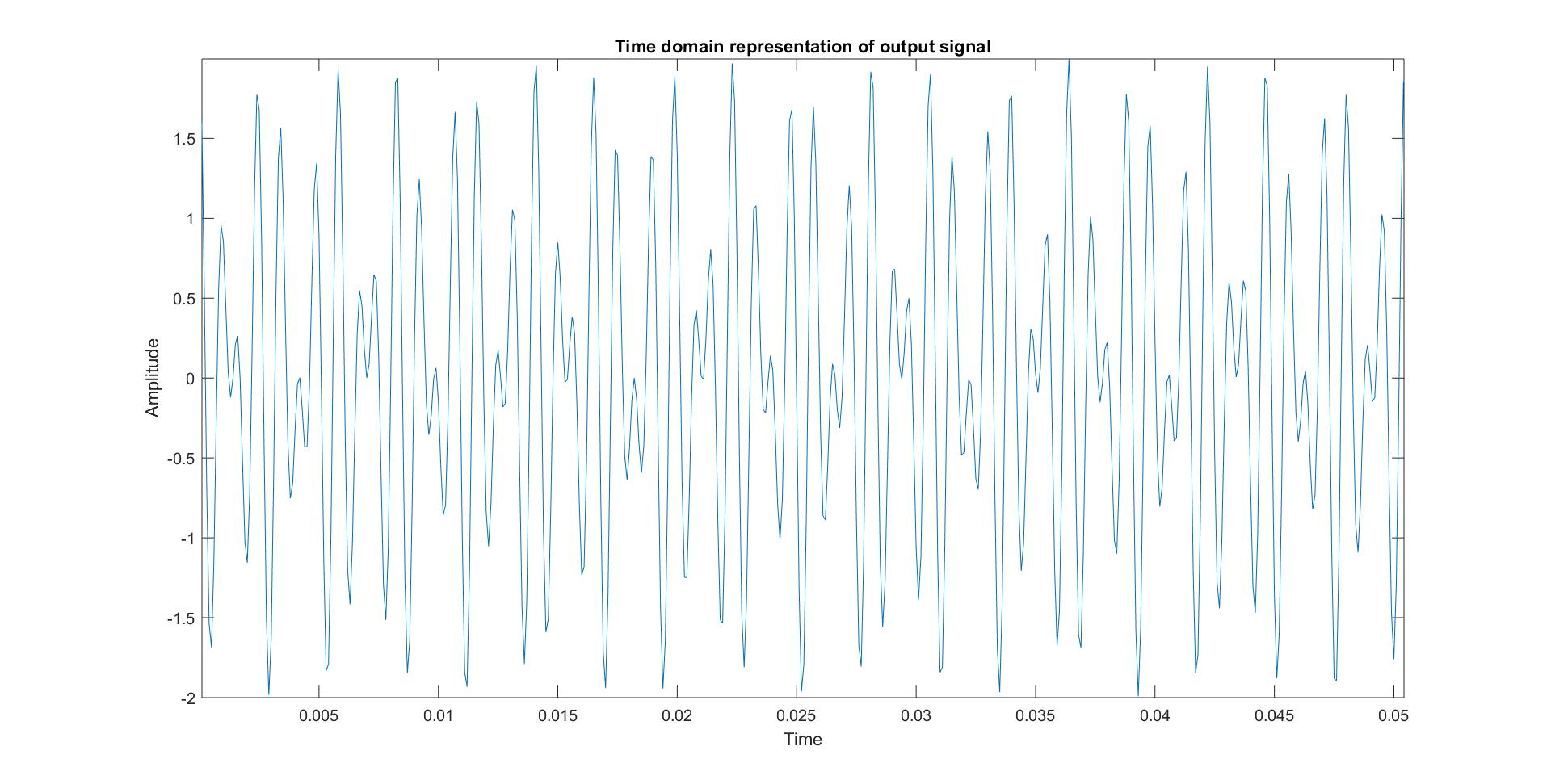
clear all;  
N = 512;  
Fs = 10000;  
T = 1/Fs;  
t = 0:T:(N-1)\*T\*10;  
F = [697,770,852,941;1209,1336,1477,1633];  
  
D\_string = 'Entered Character is ';  
i = input(D\_string, 's');  
  
if i == '1'  
 f1 = F(1,1);  
 f2 = F(2,1);  
elseif i == '2'  
 f1 = F(1,1);  
 f2 = F(2,2);  
elseif i == '3'  
 f1 = F(1,1);  
 f2 = F(2,3);  
elseif i == 'A'  
 f1 = F(1,1);  
 f2 = F(2,4);  
elseif i == '4'  
 f1 = F(1,2);  
 f2 = F(2,1);  
elseif i == '5'  
 f1 = F(1,2);  
 f2 = F(2,2);  
elseif i == '6'  
 f1 = F(1,2);  
 f2 = F(2,3);  
elseif i == 'B'  
 f1 = F(1,2);  
 f2 = F(2,4);  
elseif i == '7'  
 f1 = F(1,3);  
 f2 = F(2,1);  
elseif i == '8'  
 f1 = F(1,3);  
 f2 = F(2,2);  
elseif i == '9'  
 f1 = F(1,3);  
 f2 = F(2,3);  
elseif i == 'C'  
 f1 = F(1,3);  
 f2 = F(2,4);  
elseif i == '\*'  
 f1 = F(1,4);  
 f2 = F(2,1);  
elseif i == '0'  
 f1 = F(1,4);  
 f2 = F(2,2);  
elseif i == '#'  
 f1 = F(1,4);  
 f2 = F(2,3);  
elseif i == 'D'  
 f1 = F(1,4);  
 f2 = F(2,4);  
else  
 fprintf('Invalid Selection\n');  
end  
  
y = cos(2\*pi\*f1\*t) + cos(2\*pi\*f2\*t);  
  
figure;  
plot(t,y);  
y1 = fft(y,N);  
y1 = fftshift(y1);  
y1 = abs(y1/N);  
w2 = -pi:2\*pi/(N-1):pi;  
figure;  
plot(w2,y1);  
  
y\_rms = zeros(2,4);  
  
L=128;  
n = 0:1:L-1;  
  
BPF11 = cos(2\*pi/Fs\*F(1,1)\*n);  
Y11 = conv(y,BPF11);  
  
y\_fft11 = fft(Y11,N);  
y\_fft11 = fftshift(y\_fft11);  
y\_fft11 = abs(y\_fft11/N);  
y\_rms(1,1) = rms(y\_fft11);  
  
  
BPF12 = cos(2\*pi/Fs\*F(1,2)\*n);  
Y12 = conv(y,BPF12);  
  
y\_fft12 = fft(Y12,N);  
y\_fft12 = fftshift(y\_fft12);  
y\_fft12 = abs(y\_fft12/N);  
y\_rms(1,2) = rms(y\_fft12);  
  
  
BPF13 = cos(2\*pi/Fs\*F(1,3)\*n);  
Y13 = conv(y,BPF13);  
  
y\_fft13 = fft(Y13,N);  
y\_fft13 = fftshift(y\_fft13);  
y\_fft13 = abs(y\_fft13/N);  
y\_rms(1,3) = rms(y\_fft13);  
  
  
BPF14 = cos(2\*pi/Fs\*F(1,4)\*n);  
Y14 = conv(y,BPF14);  
  
y\_fft14 = fft(Y14,N);  
y\_fft14 = fftshift(y\_fft14);  
y\_fft14 = abs(y\_fft14/N);  
y\_rms(1,4) = rms(y\_fft14);  
  
  
BPF21 = cos(2\*pi/Fs\*F(2,1)\*n);  
Y21 = conv(y,BPF21);  
  
y\_fft21 = fft(Y21,N);  
y\_fft21 = fftshift(y\_fft21);  
y\_fft21 = abs(y\_fft21/N);  
y\_rms(2,1) = rms(y\_fft21);  
  
  
BPF22 = cos(2\*pi/Fs\*F(2,2)\*n);  
Y22 = conv(y,BPF22);  
  
y\_fft22 = fft(Y22,N);  
y\_fft22 = fftshift(y\_fft22);  
y\_fft22 = abs(y\_fft22/N);  
y\_rms(2,2) = rms(y\_fft22);  
  
  
BPF23 = cos(2\*pi/Fs\*F(2,3)\*n);  
Y23 = conv(y,BPF23);  
  
y\_fft23 = fft(Y23,N);  
y\_fft23 = fftshift(y\_fft23);  
y\_fft23 = abs(y\_fft23/N);  
y\_rms(2,3) = rms(y\_fft23);  
  
  
BPF24 = cos(2\*pi/Fs\*F(2,4)\*n);  
Y24 = conv(y,BPF24);  
  
y\_fft24 = fft(Y24,N);  
y\_fft24 = fftshift(y\_fft24);  
y\_fft24 = abs(y\_fft24/N);  
y\_rms(2,4) = rms(y\_fft24);  
  
  
[M,I] = max(y\_rms.');  
  
if I(1)==1&&I(2)==1  
 fprintf('Detected sequence = 1');  
elseif I(1)==1&&I(2)==2  
 fprintf('Detected sequence = 2');  
elseif I(1)==1&&I(2)==3  
 fprintf('Detected sequence = 3');  
elseif I(1)==1&&I(2)==4  
 fprintf('Detected sequence = A');  
elseif I(1)==2&&I(2)==1  
 fprintf('Detected sequence = 4');  
elseif I(1)==2&&I(2)==2  
 fprintf('Detected sequence = 5');  
elseif I(1)==2&&I(2)==3  
 fprintf('Detected sequence = 6');  
elseif I(1)==2&&I(2)==4  
 fprintf('Detected sequence = B');  
elseif I(1)==3&&I(2)==1  
 fprintf('Detected sequence = 7');  
elseif I(1)==3&&I(2)==2  
 fprintf('Detected sequence = 8');  
elseif I(1)==3&&I(2)==3  
 fprintf('Detected sequence = 9');  
elseif I(1)==3&&I(2)==4  
 fprintf('Detected sequence = C');  
elseif I(1)==4&&I(2)==1  
 fprintf('Detected sequence = \*');  
elseif I(1)==4&&I(2)==2  
 fprintf('Detected sequence = 0');  
elseif I(1)==4&&I(2)==3  
 fprintf('Detected sequence = #');  
elseif I(1)==4&&I(2)==4  
 fprintf('Detected sequence = D');  
end  
fprintf("\r");

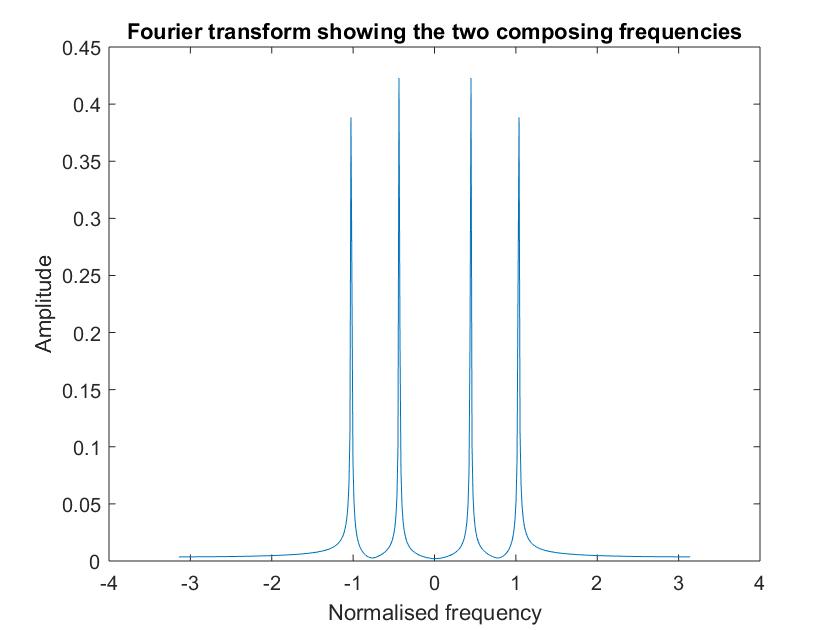
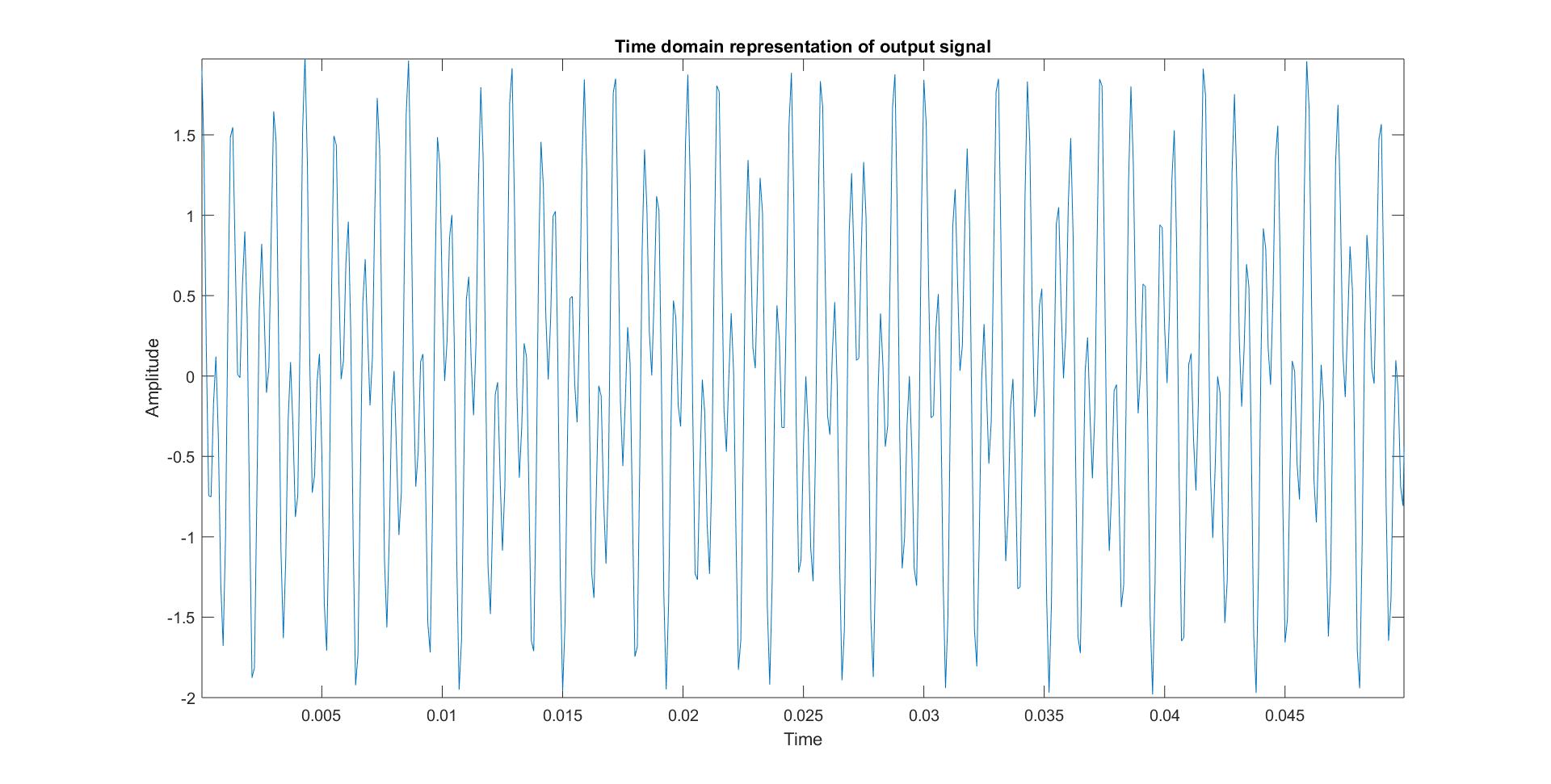
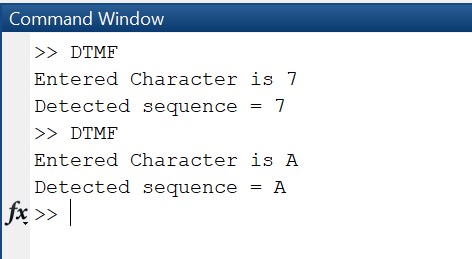
Outputs:

Checking the working of the code for input 7 and the input A. The following plots were obtained:







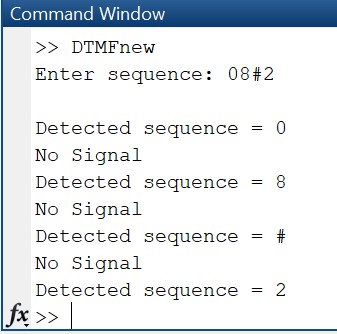
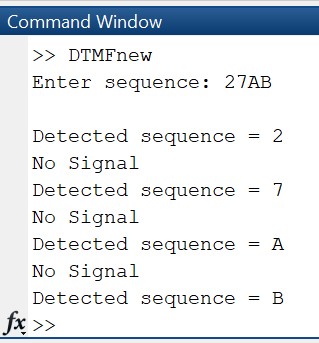


**Code for string input:**

The input of 4 characters is taken and concatenation of the signal is done and passed through the filters. The output in command window gives the 4 characters as the output along with the spaces generated implying that no signal was sent during that duration.

clear all;  
N = 512;  
Fs = 10000;  
T = 1/Fs;  
t = 0:T:70\*N\*T+6\*T;  
t1 = 0:T:10\*N\*T;  
F = [697,770,852,941;1209,1336,1477,1633];  
j = 1:1:4;  
f1 = zeros(1,4);  
f2 = zeros(1,4);  
  
prompt = 'Enter sequence: ';  
x = input(prompt,'s');  
  
for j=1:4  
if x(j) == '1'  
 f1(j) = F(1,1);  
 f2(j) = F(2,1);  
elseif x(j) == '2'  
 f1(j) = F(1,1);  
 f2(j) = F(2,2);  
elseif x(j) == '3'  
 f1(j) = F(1,1);  
 f2(j) = F(2,3);  
elseif x(j) == 'A'  
 f1(j) = F(1,1);  
 f2(j) = F(2,4);  
elseif x(j) == '4'  
 f1(j) = F(1,2);  
 f2(j) = F(2,1);  
elseif x(j) == '5'  
 f1(j) = F(1,2);  
 f2(j) = F(2,2);  
elseif x(j) == '6'  
 f1(j) = F(1,2);  
 f2(j) = F(2,3);  
elseif x(j) == 'B'  
 f1(j) = F(1,2);  
 f2(j) = F(2,4);  
elseif x(j) == '7'  
 f1(j) = F(1,3);  
 f2(j) = F(2,1);  
elseif x(j) == '8'  
 f1(j) = F(1,3);  
 f2(j) = F(2,2);  
elseif x(j) == '9'  
 f1(j) = F(1,3);  
 f2(j) = F(2,3);  
elseif x(j) == 'C'  
 f1(j) = F(1,3);  
 f2(j) = F(2,4);  
elseif x(j) == '\*'  
 f1(j) = F(1,4);  
 f2(j) = F(2,1);  
elseif x(j) == '0'  
 f1(j) = F(1,4);  
 f2(j) = F(2,2);  
elseif x(j) == '#'  
 f1(j) = F(1,4);  
 f2(j) = F(2,3);  
elseif x(j) == 'D'  
 f1(j) = F(1,4);  
 f2(j) = F(2,4);  
else  
 fprintf('Invalid Selection\n');  
end  
end  
  
  
no\_sig = zeros(1,5121);  
a = cos(2\*pi\*f1(1)\*t1) + cos(2\*pi\*f2(1)\*t1);  
b = cos(2\*pi\*f1(2)\*t1) + cos(2\*pi\*f2(2)\*t1);  
c = cos(2\*pi\*f1(3)\*t1) + cos(2\*pi\*f2(3)\*t1);  
d = cos(2\*pi\*f1(4)\*t1) + cos(2\*pi\*f2(4)\*t1);  
  
z = [a no\_sig b no\_sig c no\_sig d];  
  
%plot(t,z);  
  
fprintf('\r');  
for i = 1:7  
 y = z((i-1)\*5121+1:i\*5121-1);  
 y\_rms = zeros(2,4);  
  
L=128;  
n = 0:1:L-1;  
  
BPF11 = cos(2\*pi/Fs\*F(1,1)\*n);  
Y11 = conv(y,BPF11);  
  
y\_fft11 = fft(Y11,N);  
y\_fft11 = fftshift(y\_fft11);  
y\_fft11 = abs(y\_fft11/N);  
y\_rms(1,1) = rms(y\_fft11);  
  
  
BPF12 = cos(2\*pi/Fs\*F(1,2)\*n);  
Y12 = conv(y,BPF12);  
  
y\_fft12 = fft(Y12,N);  
y\_fft12 = fftshift(y\_fft12);  
y\_fft12 = abs(y\_fft12/N);  
y\_rms(1,2) = rms(y\_fft12);  
  
  
BPF13 = cos(2\*pi/Fs\*F(1,3)\*n);  
Y13 = conv(y,BPF13);  
  
y\_fft13 = fft(Y13,N);  
y\_fft13 = fftshift(y\_fft13);  
y\_fft13 = abs(y\_fft13/N);  
y\_rms(1,3) = rms(y\_fft13);  
  
  
BPF14 = cos(2\*pi/Fs\*F(1,4)\*n);  
Y14 = conv(y,BPF14);  
  
y\_fft14 = fft(Y14,N);  
y\_fft14 = fftshift(y\_fft14);  
y\_fft14 = abs(y\_fft14/N);  
y\_rms(1,4) = rms(y\_fft14);  
  
  
BPF21 = cos(2\*pi/Fs\*F(2,1)\*n);  
Y21 = conv(y,BPF21);  
  
y\_fft21 = fft(Y21,N);  
y\_fft21 = fftshift(y\_fft21);  
y\_fft21 = abs(y\_fft21/N);  
y\_rms(2,1) = rms(y\_fft21);  
  
  
BPF22 = cos(2\*pi/Fs\*F(2,2)\*n);  
Y22 = conv(y,BPF22);  
  
y\_fft22 = fft(Y22,N);  
y\_fft22 = fftshift(y\_fft22);  
y\_fft22 = abs(y\_fft22/N);  
y\_rms(2,2) = rms(y\_fft22);  
  
  
BPF23 = cos(2\*pi/Fs\*F(2,3)\*n);  
Y23 = conv(y,BPF23);  
  
y\_fft23 = fft(Y23,N);  
y\_fft23 = fftshift(y\_fft23);  
y\_fft23 = abs(y\_fft23/N);  
y\_rms(2,3) = rms(y\_fft23);  
  
  
BPF24 = cos(2\*pi/Fs\*F(2,4)\*n);  
Y24 = conv(y,BPF24);  
  
y\_fft24 = fft(Y24,N);  
y\_fft24 = fftshift(y\_fft24);  
y\_fft24 = abs(y\_fft24/N);  
y\_rms(2,4) = rms(y\_fft24);  
  
  
[M,I] = max(y\_rms.');  
if M(1)==0&&M(2)==0  
 fprintf('No Signal');  
elseif I(1)==1&&I(2)==1  
 fprintf('Detected sequence = 1');  
elseif I(1)==1&&I(2)==2  
 fprintf('Detected sequence = 2');  
elseif I(1)==1&&I(2)==3  
 fprintf('Detected sequence = 3');  
elseif I(1)==1&&I(2)==4  
 fprintf('Detected sequence = A');  
elseif I(1)==2&&I(2)==1  
 fprintf('Detected sequence = 4');  
elseif I(1)==2&&I(2)==2  
 fprintf('Detected sequence = 5');  
elseif I(1)==2&&I(2)==3  
 fprintf('Detected sequence = 6');  
elseif I(1)==2&&I(2)==4  
 fprintf('Detected sequence = B');  
elseif I(1)==3&&I(2)==1  
 fprintf('Detected sequence = 7');  
elseif I(1)==3&&I(2)==2  
 fprintf('Detected sequence = 8');  
elseif I(1)==3&&I(2)==3  
 fprintf('Detected sequence = 9');  
elseif I(1)==3&&I(2)==4  
 fprintf('Detected sequence = C');  
elseif I(1)==4&&I(2)==1  
 fprintf('Detected sequence = \*');  
elseif I(1)==4&&I(2)==2  
 fprintf('Detected sequence = 0');  
elseif I(1)==4&&I(2)==3  
 fprintf('Detected sequence = #');  
elseif I(1)==4&&I(2)==4  
 fprintf('Detected sequence = D');  
end  
fprintf("\r");  
  
end

Command Prompt:

**Graphical User Interface:**

CODE:

classdef app1 < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

Button matlab.ui.control.Button

Button\_2 matlab.ui.control.Button

Button\_3 matlab.ui.control.Button

AButton matlab.ui.control.Button

Button\_4 matlab.ui.control.Button

Button\_5 matlab.ui.control.Button

Button\_6 matlab.ui.control.Button

BButton matlab.ui.control.Button

Button\_7 matlab.ui.control.Button

Button\_8 matlab.ui.control.Button

Button\_9 matlab.ui.control.Button

CButton matlab.ui.control.Button

Button\_10 matlab.ui.control.Button

Button\_11 matlab.ui.control.Button

Button\_12 matlab.ui.control.Button

DButton matlab.ui.control.Button

Button\_13 matlab.ui.control.StateButton

PressanykeyLabel matlab.ui.control.Label

ThePressedkeyisLabel matlab.ui.control.Label

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 640 480];

app.UIFigure.Name = 'UI Figure';

% Create Button

app.Button = uibutton(app.UIFigure, 'push');

app.Button.Position = [63 408 100 22];

app.Button.Text = '1';

% Create Button\_2

app.Button\_2 = uibutton(app.UIFigure, 'push');

app.Button\_2.Position = [162 408 100 22];

app.Button\_2.Text = '2';

% Create Button\_3

app.Button\_3 = uibutton(app.UIFigure, 'push');

app.Button\_3.Position = [261 408 100 22];

app.Button\_3.Text = '3';

% Create AButton

app.AButton = uibutton(app.UIFigure, 'push');

app.AButton.Position = [360 408 100 22];

app.AButton.Text = 'A';

% Create Button\_4

app.Button\_4 = uibutton(app.UIFigure, 'push');

app.Button\_4.Position = [63 387 100 22];

app.Button\_4.Text = '4';

% Create Button\_5

app.Button\_5 = uibutton(app.UIFigure, 'push');

app.Button\_5.Position = [162 387 100 22];

app.Button\_5.Text = '5';

% Create Button\_6

app.Button\_6 = uibutton(app.UIFigure, 'push');

app.Button\_6.Position = [261 387 100 22];

app.Button\_6.Text = '6';

% Create BButton

app.BButton = uibutton(app.UIFigure, 'push');

app.BButton.Position = [360 387 100 22];

app.BButton.Text = 'B';

% Create Button\_7

app.Button\_7 = uibutton(app.UIFigure, 'push');

app.Button\_7.Position = [63 366 100 22];

app.Button\_7.Text = '7';

% Create Button\_8

app.Button\_8 = uibutton(app.UIFigure, 'push');

app.Button\_8.Position = [162 366 100 22];

app.Button\_8.Text = '8';

% Create Button\_9

app.Button\_9 = uibutton(app.UIFigure, 'push');

app.Button\_9.Position = [261 366 100 22];

app.Button\_9.Text = '9';

% Create CButton

app.CButton = uibutton(app.UIFigure, 'push');

app.CButton.Position = [360 366 100 22];

app.CButton.Text = 'C';

% Create Button\_10

app.Button\_10 = uibutton(app.UIFigure, 'push');

app.Button\_10.Position = [63 345 100 22];

app.Button\_10.Text = '\*';

% Create Button\_11

app.Button\_11 = uibutton(app.UIFigure, 'push');

app.Button\_11.Position = [162 345 100 22];

app.Button\_11.Text = '0';

% Create Button\_12

app.Button\_12 = uibutton(app.UIFigure, 'push');

app.Button\_12.Position = [261 345 100 22];

app.Button\_12.Text = '#';

% Create DButton

app.DButton = uibutton(app.UIFigure, 'push');

app.DButton.Position = [360 345 100 22];

app.DButton.Text = 'D';

% Create Button\_13

app.Button\_13 = uibutton(app.UIFigure, 'state');

app.Button\_13.Text = '7';

app.Button\_13.Position = [215 252 100 22];

% Create PressanykeyLabel

app.PressanykeyLabel = uilabel(app.UIFigure);

app.PressanykeyLabel.HorizontalAlignment = 'center';

app.PressanykeyLabel.Position = [224 429 81 22];

app.PressanykeyLabel.Text = 'Press any key';

% Create ThePressedkeyisLabel

app.ThePressedkeyisLabel = uilabel(app.UIFigure);

app.ThePressedkeyisLabel.HorizontalAlignment = 'center';

app.ThePressedkeyisLabel.Position = [210 273 111 22];

app.ThePressedkeyisLabel.Text = 'The Pressed key is:';

end

end

methods (Access = public)

% Construct app

function app = app1

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

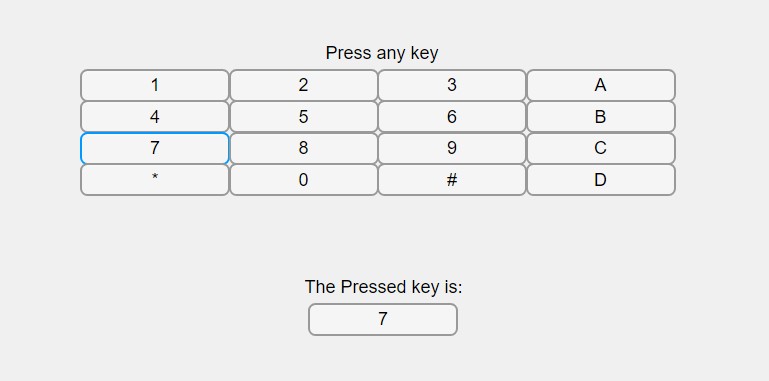
delete(app.UIFigure)

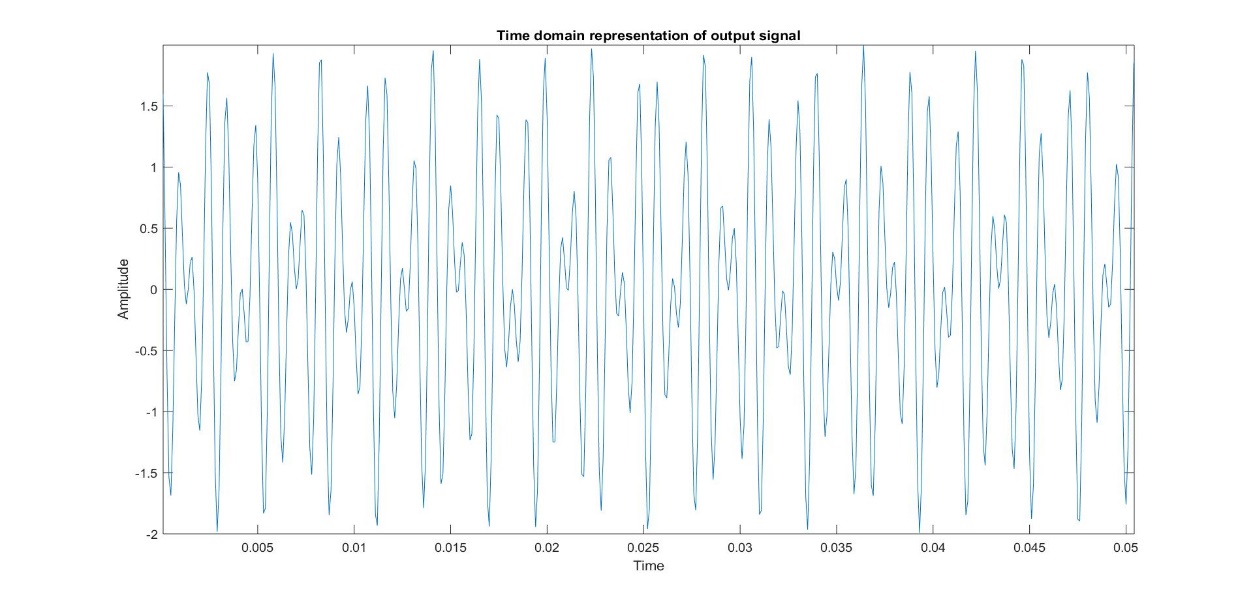
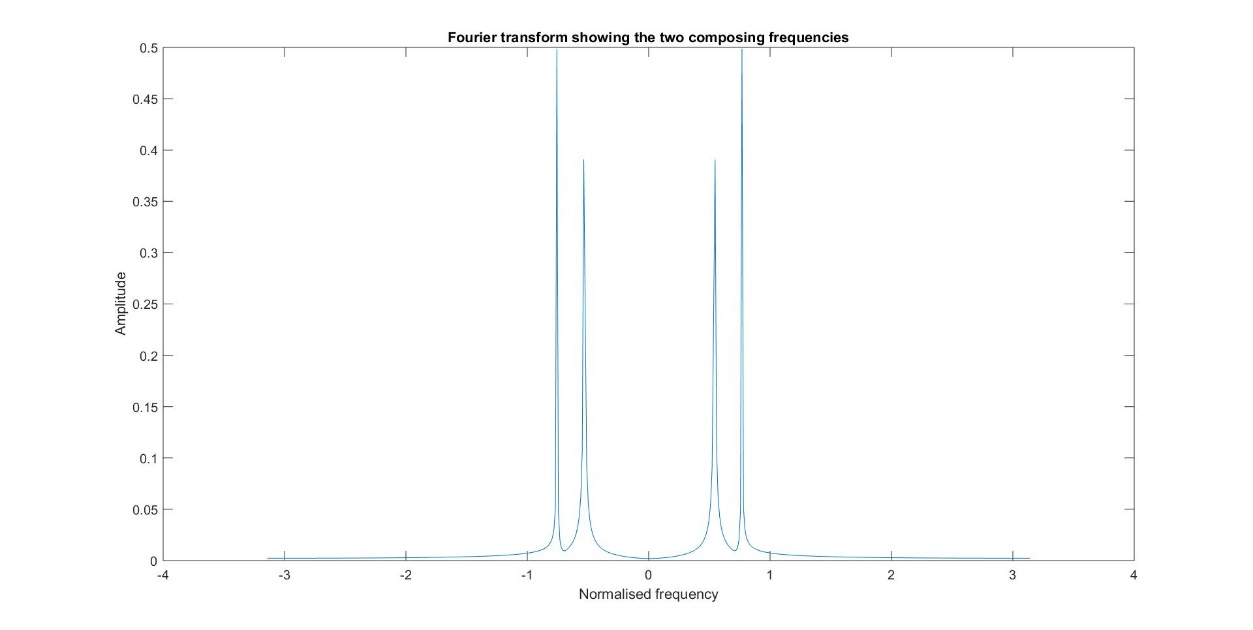
end

end

end

Output:





**Discussion:**

* These frequency values are not one’s multiples so that no overlap will occur between at the receiver side. Multi-frequency signalling is a group of signalling methods that use a mixture of two pure sine waves sounds.
* The band pass filter we used here is impulse response given is:

where N is the length of filter & is the centre frequency.

* We used the sampling frequency far more than the maximum frequency component listed, so that the error we expected will be lessen.
* The output of the filter bank consists of frequency components which the bandpass filter for which the amplitude when compared to the other is higher.
* The GUI helps in easy visualisation of the code and how the telephonic DTMF works.

*[Published with MATLAB® R2018a](https://www.mathworks.com/products/matlab" \o "https://www.mathworks.com/products/matlab)*