# Digital Signal Processing Lab
# Experiment 3
## DTMF (Dual Tone Multifrequency or Touch-Tone ) coder/ decoder

---

**Name: Anand Jhunjhunwala**
**Roll No: 17EC30041**
**Group No: 63**

---

## Objective:
Study and Analysis of DTMF (Dual-tone multifrequency) coder/decoder using Digital FIR Filter in MATLAB.

## Theory:
**DTMF Signal Synthesis:**
Telephone keypads generate Dual-Tone multifrequency signals to dial a telephone number. When any key is pressed, the sinusoids of the corresponding row and column frequencies are generated and summed producing two simultaneous or dual-tone.

A DTMF tone represents a single keypress on a telephone device, consists of two summed frequencies that have been chosen so that no harmonics occur i.e No frequency is an integer multiple of other and the difference and sum of any two frequency are not equal to any of the other frequency.

Table 1: shows the exact frequency used and shows the telephone touchpad. The tone for a key is generated when the frequency for the keys column and row are summed together as waveforms.

**Table1:**

| Hz | 1209 | 1336 | 1477 | 1633 |
|---|---|---|---|---|
| 697 | 1 | 2 | 3 | A |
| 770 | 4 | 5 | 6 | B |
| 852 | 7 | 8 | 9 | C |
| 941 | * | 0 | # | D |

### Matlab code for generating tone: [File name: gen_signal.m ]

- This code is responsible for generating tone corresponding to the key pressed by the user, it takes input as the key pressed and give output the generated tone signal.

Code snippet:

```
function [x] = gen_signal(value,L)
t = 1:L;
switch value
    case '1'
        f1 = 697;
        f2 = 1209;
    case '2'
        f1 = 697;
        f2 = 1336;
    case '3'
        f1 = 697;
        f2 = 1477;
    case 'A'
        f1 = 697;
        f2 = 1633;
    case '4'
        f1 = 770;
        f2 = 1209;
    case '5'
        f1 = 770;
        f2 = 1336;
    case '6'
        f1 = 770;
        f2 = 1477;
    case 'B'
        f1 = 770;
        f2 = 1633;
    case '7'
        f1 = 852;
        f2 = 1209;
    case '8'
        f1 = 852;
        f2 = 1336;
    case '9'
        f1 = 852;
        f2 = 1477;
    case 'C'
        f1 = 852;
        f2 = 1633;
    case '*'
        f1 = 941;
        f2 = 1209;
    case '0'
        f1 = 941;
        f2 = 1336;
    case '#'
        f1 = 941;
        f2 = 1477;
    case 'D'
        f1 = 941;
        f2 = 1633;
end
x = cos(f1*t) + cos(f2*t);
end
```

Clearly from the code gen_signal is a function who takes input the key pressed (i.e value) and generates output the required tone (i.e X) and return it.
This code plays the role of the coder in my experiment.

## Concept Used for Decoding:

---

Decoding of tone signal was done using bandpass filtering to determine which frequency was present in the tone signal.
Bandpass filter used was FIR bandpass filters and it is designed using the L-Point average filter.
In L-Point average filters design, the impulse response of the designed filter is given by:

**H[n] = b*cos(Wc*n)  where 0<=n<L**

The bandwidth of the bandpass filter design is controlled by L, the larger the L, the narrower the bandwidth.
Wc is the center frequency of the bandpass filter designed.

For decoding the tone, each tone is passed through 8 bandpass filters centered at all possible frequency component in tone signal, the output is an 8 filtered signal, then RMS power of these signal was calculated, and its value is largest for that two frequency which is present in tone and less for all other.

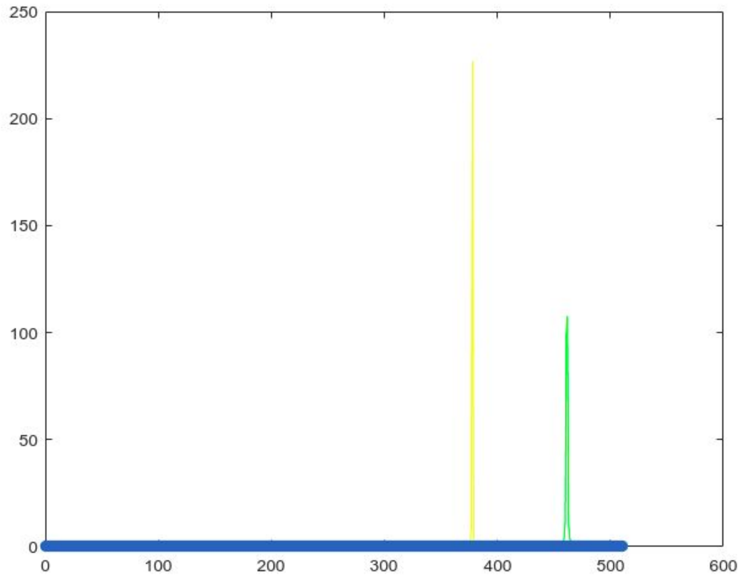
And thus finding the max two elements in power array we can decode the key pressed using Tabel1.

**Code snippet: [File name: assign3.m]**

```
L =1000;
wc = [697, 770, 852, 941, 1209, 1336, 1477, 1633];
h_n = [;];
k = 1:L;
y_n = [;];
H=[;];
W=[;];
power = [];
t = 0:511;
for i=1:8
    temp = [];
    temp = cos(wc(i)*k);
    h_n(i,:)= temp;
end
value = input('Enter the number you want to send \n', 's');
x_n = gen_signal(value,L);

fprintf('\nProcessing convolution with 8 filters\n');
for i=1:8
    y_n(i,:) = conv(x_n,h_n(i,:));
end
for i =1:8
    [H(i,:), W(i,:)] = freqz(y_n(i,:), 1024);
    H(i,:) = abs(H(i,:));
end
plot(t,H(1,:),'b',t,H(2,:),'g',t,H(3,:),'r',t,H(4,:),'c',t,H(5,:),'m',t,H(6,:),'y',t,H(7,:),'k',t,H(8,:),'o' );
ylim([0,250]);
fprintf('\nProcessing power of signal\n');
for i=1:8
    power(i) = rms(y_n(i,:))^2;
end

power

fprintf('\nDecoding the value\n');
[max,I] = maxk(power,2);
I = sort(I);
result = decode(I);
fprintf('\nEntered value is: %s\n',result);
```

- This code is responsible for coding the key pressed using gen_signal function (discussed earlier) and passing it to 8 bandpass filters and computing RMS power for each of the output.
  Then max two-element was found using maxk function and key pressed was decoded using decode function (Explained later).

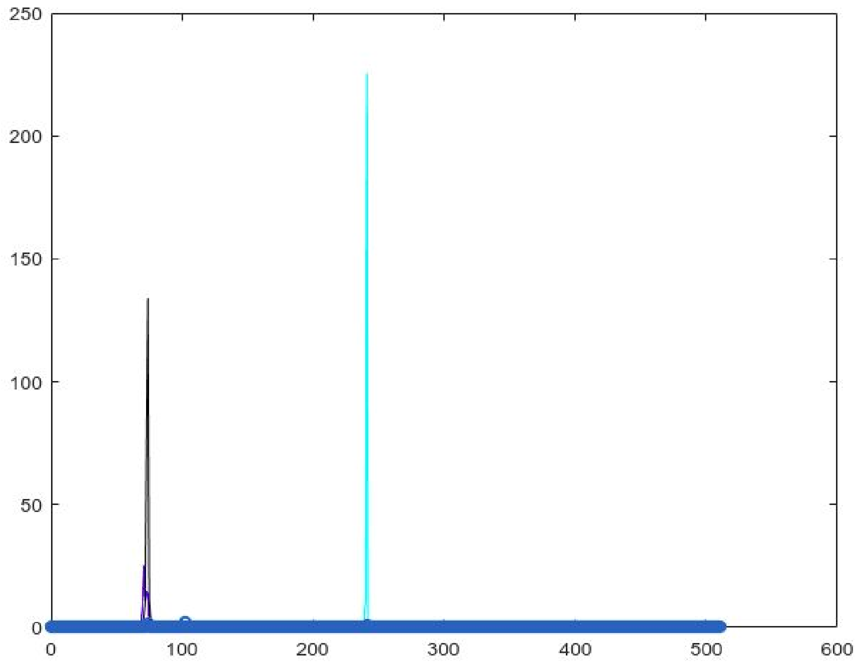
**Results:**
**Key pressed** 5:
Frequency response after passing it to 8 filters:




Power:

```
power =

   1.0e+04 *

    0.0001    4.1301    0.0007    0.0001    0.0007    4.1540    0.0001    0.0001
```

Key Pressed #:
Frequency response after passing it to 8 filters:



Power:

```
power =

   1.0e+04 *

    0.0821    0.0000    0.0001    4.1581    0.0001    0.0001    4.1485    0.0009
```

Clearly from the result, for key 5, maximum power is obtained at index 2 and 6 and for key # maximum power is obtained at index 4 and 7.
This index is then used to decode the actual key pressed using the code **decode.m.**

**Code snippet: [File name: decode.m]**

```
function [x] = decode(I)
if I(1) == 1
    if I(2) == 5
        x = '1';
    elseif I(2) == 6
        x = '2';
    elseif I(2) == 7
        x = '3';
    elseif I(2) == 8
        x = 'A';
    end
end
if I(1) == 2
    if I(2) == 5
        x = '4';
    elseif I(2) == 6
        x = '5';
    elseif I(2) == 7
        x = '6';
    elseif I(2) == 8
        x = 'B';
    end
end
if I(1) == 3
    if I(2) == 5
        x = '7';
    elseif I(2) == 6
        x = '8';
    elseif I(2) == 7
        x = '9';
    elseif I(2) == 8
        x = 'C';
    end
end
if I(1) == 4
    if I(2) == 5
        x = '*';
    elseif I(2) == 6
        x = '0';
    elseif I(2) == 7
        x = '#';
    elseif I(2) == 8
        x = 'D';
    end
end
end
```

This code takes the input the index of 2 maximum powers and uses that to calculate the key pressed and returns it as its output which is printed from assign3.m code.

This link contains all the original Matlab code:

https://drive.google.com/open?id=1oXQK7oYaBE_VSUya_Zovpupyz7iWjCyf

## Discussions:

- A DTMF tone representing a single key selected consists of two summed frequencies that have been chosen such that no harmonics occur.

- The sampling frequency used is much more than the maximum frequency component listed, so there will not arise a case of aliasing which reduces the probability of the wrong detection of the key.

- No frequency is an integer multiple of one another so that no overlap occurs at the receiver side.

- Difference or sum of any two frequencies does not equal any of the defined frequencies.

- To decode a DTMF signal, 8 bandpass filters each centered at one of the DTMF frequencies implemented by an L point FIR BPF.

- Larger the value of L, narrower is the bandwidth of the bandpass filter as the width of the main lobe is inversely proportional to the length of the filter(L).

- The maximum of them among row frequencies(magnitude of FFT) and column frequencies are determined to detect the key selected from the signal received.

- As the value of L is increased in the power array, the smaller values go to zero as the filter becomes more and more narrow and ideal.

---