

Digital Signal Processing Laboratory

Experiment 3

By: Hardik Tibrewal (18EC10020)

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

Objective:

- (a) Study and analysis of DTMF coder/decoder using Digital FIR Filter in MATLAB

Theoretical Background:

DTMF coder/decoder was used in telephone keypads as a way of communicating the information about the button pressed on the telephone between different equipment and switching centres. It uses a unique pair of two frequencies to represent each symbol on the keypad. The frequencies are chosen such that neither of them is an integer multiple of the other and the sum or difference of any two does not result in another chosen frequency. This helps to uniquely identify each symbol and makes the process of decoding easier.

When a button is pressed, two cosine waves are generated having the frequency representing the signal. The sum of the two cosine waves is the encoded symbol. This is transmitted through a channel where noise contaminates the signal, and reaches the receiving end.

To decode the signal at the receiving end, a filter bank with 8 band-pass filters is present, each centred at one of the 8 unique frequencies which make up the pairs representing the symbols. The output power will be highest in the two filters centred at the frequencies present in the received wave thus giving us the unique pair of frequency components, which can be used to identify the symbol.

To design band-pass filters in MATLAB, we define the impulse response of an L-point FIR Filter as

$$h[n] = \beta \cos(\widehat{\omega}_c \cdot n), \quad 0 \leq n < L.$$

Here L is the length of the filter and $\widehat{(\omega_c)}$ is the centre frequency defining the location of the pass band. β is used to adjust the gain in the pass band. Usually, we choose it such that the pass-band gain is 1.

Pseudocode:

The length of each filter was taken to be 512, and all signals were sampled at 6 kHz, which is sufficiently above the Nyquist rate for the highest frequency component, which is 1633 Hz. A 2-D matrix with 8 rows and 512 columns is generated to represent the filter bank, with each row representing a filter centred at our desired frequency. All the frequency values are stored in an array, **fc**, in ascending order. The impulse response of each filter is then defined using the formula mentioned previously.

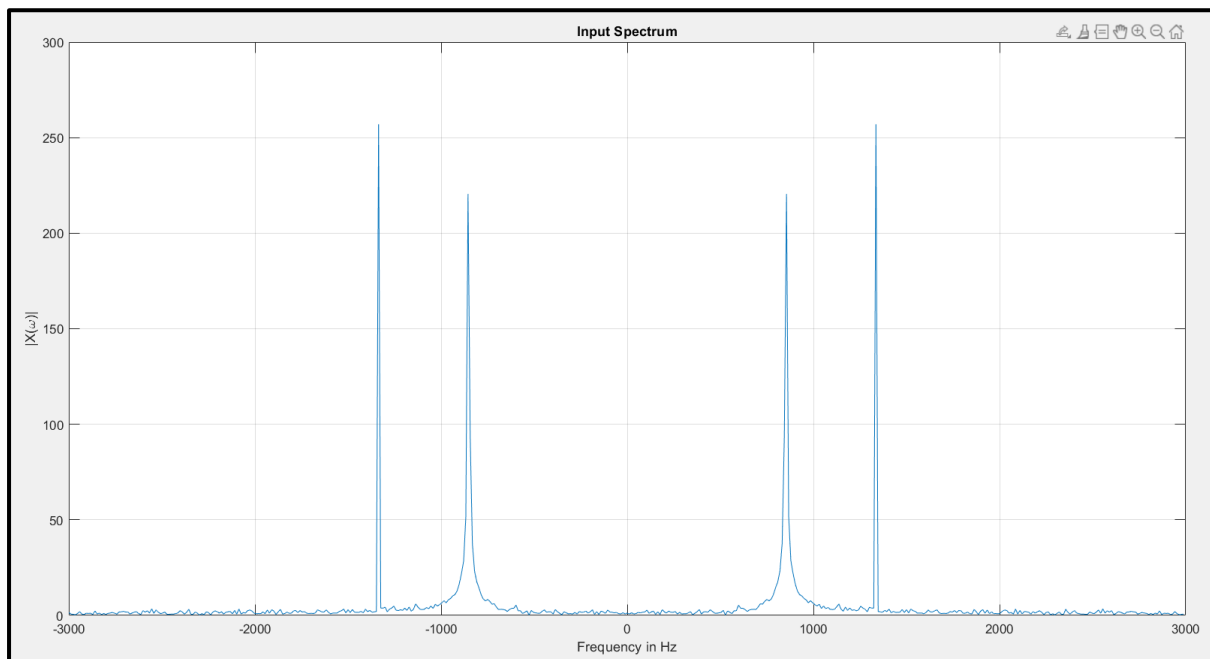
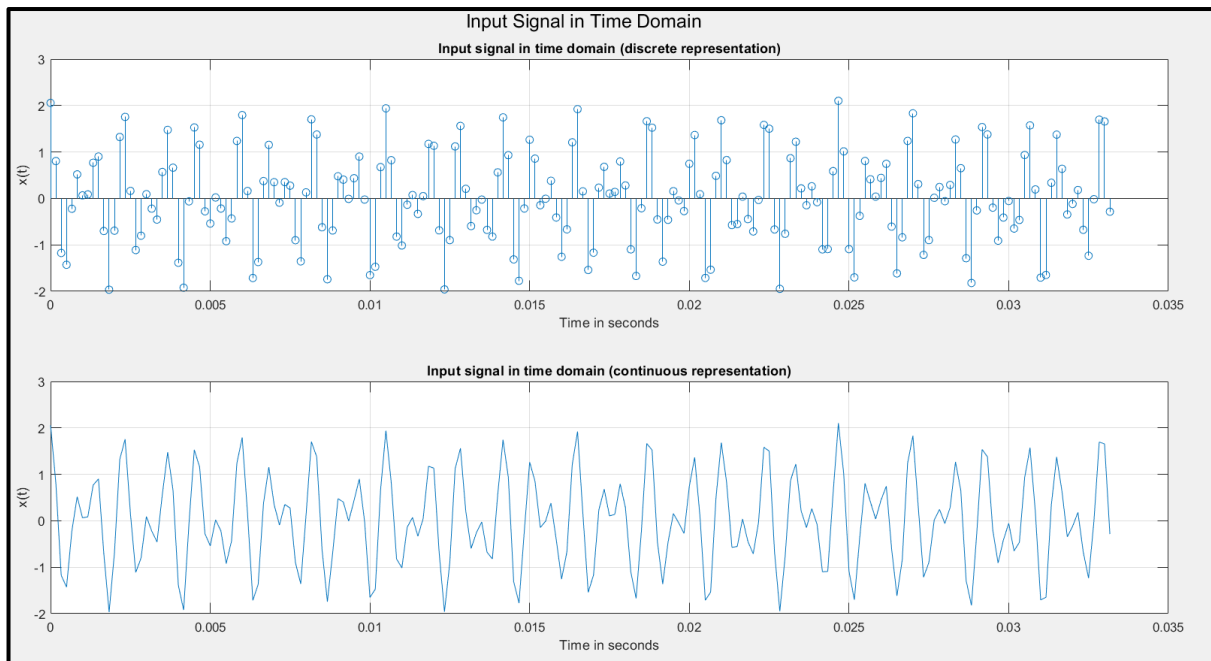
A look-up table was created with the elements placed in such a manner that access expressions could be created to easily identify the frequency components from the symbol and vice versa. (The expressions can be seen in the code added in the appendix). A signal is generated which is the sum of two cosine waves of the chosen frequencies, and noise is added to simulate its transmission through a physical channel. The input signal is then plotted in the time and frequency domains.

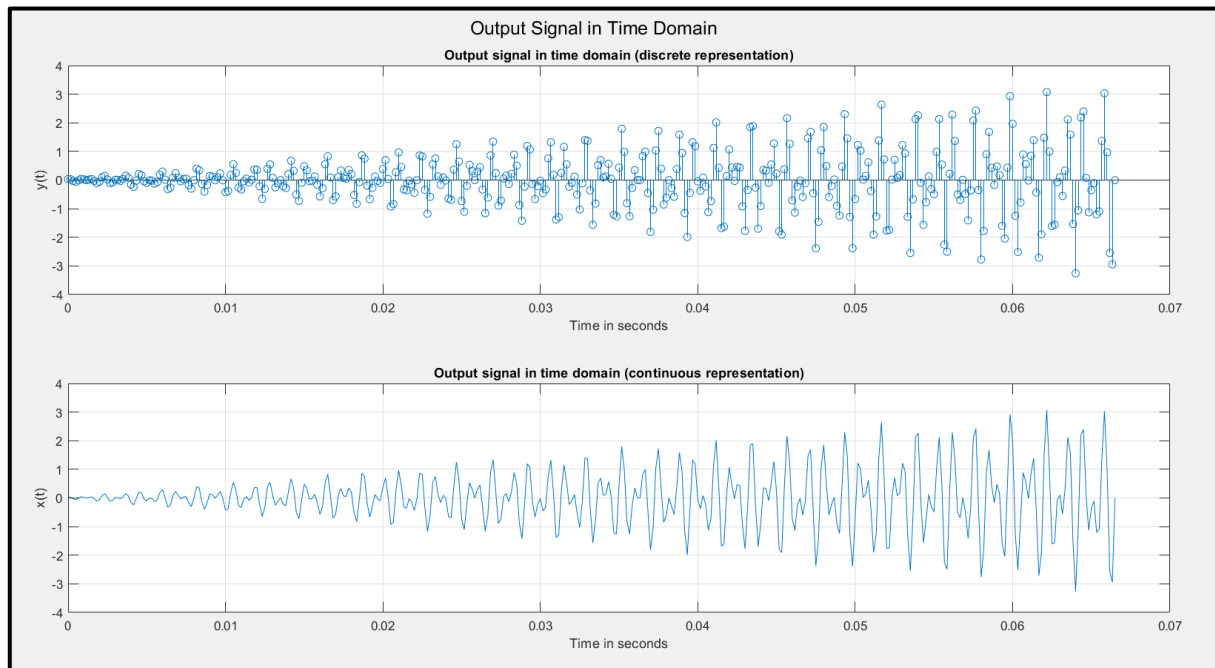
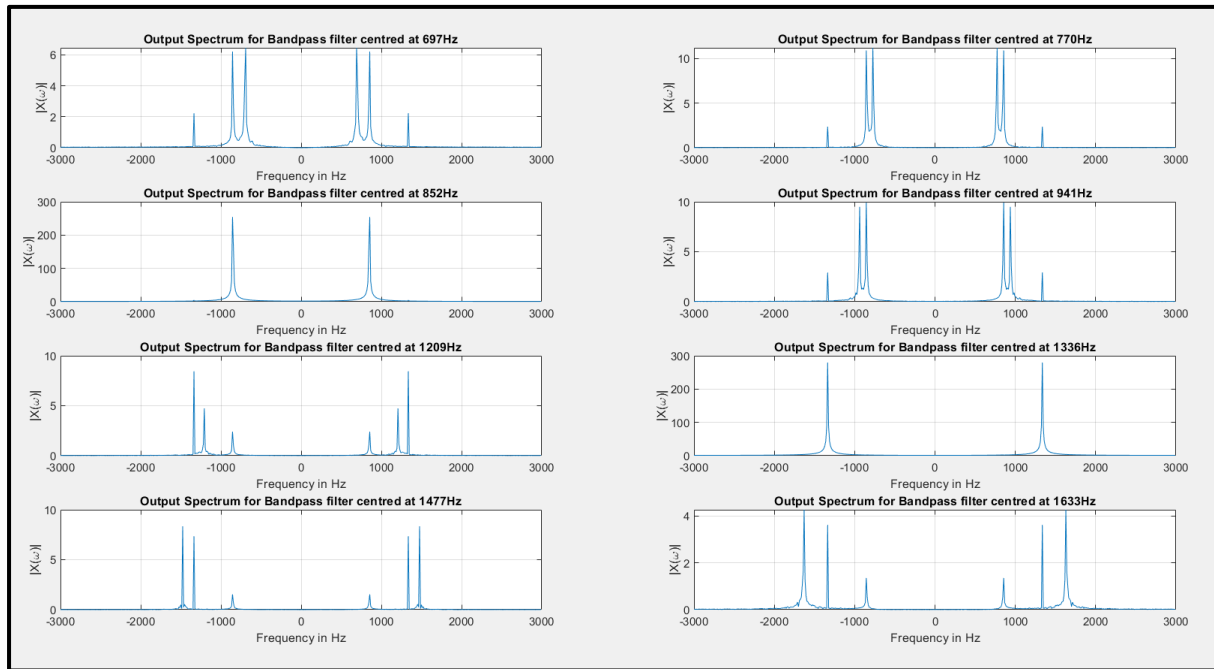
The signal is then convolved with each filter impulse response, simulating the process of filtering. The output spectrum of each filter is plotted, and the value of the output power is stored in an array.

To decode the signal, we find the indices of the maximum two output powers in the array. They will correspond to the frequency components in **fc**, which can easily be used to identify the symbol from the look-up table.

The output signal and spectrum corresponding to the two frequency components is also plotted to compare with the input signal.

Simulation Results: (Plots for an input of '8')





Filtered output here means the sum of outputs of the two filters with maximum power output.

```
>> Assignment3a_18EC10020
Which symbol do you want to send?
1
The output power was highest for bandpass filters centered at 697 Hz and 1209 Hz.
The chosen symbol, based on the output power of each filter, was 1.
>> Assignment3a_18EC10020
Which symbol do you want to send?
2
The output power was highest for bandpass filters centered at 697 Hz and 1336 Hz.
The chosen symbol, based on the output power of each filter, was 2.
>> Assignment3a_18EC10020
Which symbol do you want to send?
3
The output power was highest for bandpass filters centered at 697 Hz and 1477 Hz.
The chosen symbol, based on the output power of each filter, was 3.
>> Assignment3a_18EC10020
Which symbol do you want to send?
4
The output power was highest for bandpass filters centered at 770 Hz and 1209 Hz.
The chosen symbol, based on the output power of each filter, was 4.
>> Assignment3a_18EC10020
Which symbol do you want to send?
5
The output power was highest for bandpass filters centered at 770 Hz and 1336 Hz.
The chosen symbol, based on the output power of each filter, was 5.
>> Assignment3a_18EC10020
Which symbol do you want to send?
6
The output power was highest for bandpass filters centered at 770 Hz and 1477 Hz.
The chosen symbol, based on the output power of each filter, was 6.
```

```
>> Assignment3a_18EC10020
Which symbol do you want to send?
7
The output power was highest for bandpass filters centered at 852 Hz and 1209 Hz.
The chosen symbol, based on the output power of each filter, was 7.
>> Assignment3a_18EC10020
Which symbol do you want to send?
8
The output power was highest for bandpass filters centered at 852 Hz and 1336 Hz.
The chosen symbol, based on the output power of each filter, was 8.
>> Assignment3a_18EC10020
Which symbol do you want to send?
9
The output power was highest for bandpass filters centered at 852 Hz and 1477 Hz.
The chosen symbol, based on the output power of each filter, was 9.
>> Assignment3a_18EC10020
Which symbol do you want to send?
0
The output power was highest for bandpass filters centered at 941 Hz and 1336 Hz.
The chosen symbol, based on the output power of each filter, was 0.
>> Assignment3a_18EC10020
Which symbol do you want to send?
*
The output power was highest for bandpass filters centered at 941 Hz and 1209 Hz.
The chosen symbol, based on the output power of each filter, was *.
>> Assignment3a_18EC10020
Which symbol do you want to send?
#
The output power was highest for bandpass filters centered at 941 Hz and 1477 Hz.
The chosen symbol, based on the output power of each filter, was #.
```

```
>> Assignment3a_18EC10020
Which symbol do you want to send?
A
The output power was highest for bandpass filters centered at 697 Hz and 1633 Hz.
The chosen symbol, based on the output power of each filter, was A.
>> Assignment3a_18EC10020
Which symbol do you want to send?
B
The output power was highest for bandpass filters centered at 770 Hz and 1633 Hz.
The chosen symbol, based on the output power of each filter, was B.
>> Assignment3a_18EC10020
Which symbol do you want to send?
C
The output power was highest for bandpass filters centered at 852 Hz and 1633 Hz.
The chosen symbol, based on the output power of each filter, was C.
>> Assignment3a_18EC10020
Which symbol do you want to send?
D
The output power was highest for bandpass filters centered at 941 Hz and 1633 Hz.
The chosen symbol, based on the output power of each filter, was D.
```

Result:

The dual tone was generated with the frequency components selected based on the input symbol. For the input symbol of '8', the plots of the output spectrum of the filter bank clearly show that the maximum power is present at 852 Hz and 1336 Hz, which are the correct frequencies and were uniquely mapped to the symbol, leading to correct decoding at the receiving end. The output for each input symbol is also shown above, which shows that always the correct filters gave the maximum output power which led to proper decoding.

Discussion:

The DTMF coder/decoder was observed to work as expected, with the proper frequencies being chosen to represent the symbol, and the correct symbol being decoded from the received frequencies with most power. Look-up tables were made with proper access expressions to eliminate if-else statements in the code for encoding and decoding so that the code is more elegant.

The length of the filter was taken to be sufficiently high so get close to the ideal response while keeping the matrix dimensions reasonable for computational efficiency. We prefer higher values of L since that corresponds to a lower pass bandwidth, but more samples means heavier computation. Hence, a compromise has to be made.

The value of β ($= 0.0085$) was chosen such that the gain after receiving is roughly 1. A delay was observed in the output, which is expected in filtering. This is why we generally discard the initial samples of the filtered output. The output here is the sum of the outputs of the filters with maximum power, since that is the signal corresponding to the transmitted symbol.

The frequency values are chosen such that they aren't integral multiples of one another, or addition or subtraction of any two doesn't give a third chosen frequency. They are also sufficiently far apart in terms of magnitude so as to avoid any sort of ambiguity in decoding due to filtering, noise, or randomness. This is important since any ambiguity will defeat the purpose of the DTMF coder/decoder, which is to transmit symbols reliably.

Appendix

```
clc
clear all
close all

L = 512; %Length of filter
Fs = 6000; %Sampling frequency > 2*1633
fc = [697, 770, 852, 941, 1209, 1336, 1477, 1633];
len = 0:L-1;
t = 0:1/Fs:0.1-1/Fs;

%initialising zero matrices
h_n = zeros(8,L);
y_n = zeros(8,L+Fs/10-1);
H = zeros(8,L);
W = zeros(8,L);
pwr = zeros(1,8);

for ii = 1:8
    h_n(ii,:) = 0.0085*cos(len*(2*pi*fc(ii)/Fs)); %defining filter response in time domain
end

%Look-up table for symbols. Indices correspond to index of corresponding
%frequencies in fc array
%Arranging matrix like this according to MATLAB indexing convention
mat_symbol = ['1' '4' '7' '*'; '2' '5' '8' '0'; '3' '6' '9' '#'; 'A' 'B' 'C' 'D'];

symbol = input('Which symbol do you want to send?\n', 's');

%%% Encoder %%%
idx = find(mat_symbol == symbol);
%Access expressions to find corresponding frequencies
f1 = fc(floor((idx-1)/4)+1);
f2 = fc(4+mod(idx-1,4)+1);

% Generating the signal and adding noise
x = cos(2*pi*f1*t)+cos(2*pi*f2*t);
noise = randn(size(x)); %noise to simulate channel effects
noise = ((max(x)/max(noise))/10)*noise;
x = x + noise;
%%% Encoder End %%%

figure();
sgtitle('Input Signal in Time Domain')
subplot(211);
stem(t(1:50*floor(Fs/f2)),x(1:50*floor(Fs/f2)))
grid on
xlabel('Time in seconds');ylabel('x(t)');title('Input signal in time domain (discrete representation)');
subplot(212);
plot(t(1:50*floor(Fs/f2)),x(1:50*floor(Fs/f2)))
grid on
xlabel('Time in seconds');ylabel('x(t)');title('Input signal in time domain (continuous representation)');

figure();
freq = -Fs/2:Fs/L:Fs/2-Fs/L;
plot(freq, abs(fftshift(fft(x, L))))
grid on
xlabel('Frequency in Hz');ylabel('|X(\omega)|');title('Input Spectrum');
```

```

%% Passing the input through filter bank
figure()
for ii = 1:8
    y_n(ii,:) = conv(x, h_n(ii,:));
    [H(ii,:), W(ii,:)] = freqz(y_n(ii,:), L);
    H(ii,:) = abs(H(ii,:));
    pwr(ii) = rms(y_n(ii,:))^2;
    subplot(4,2,ii)
    plot(freq, (abs(fftshift(fft(y_n(ii,:), L))))))
    grid on
    xlabel('Frequency in Hz');ylabel('|X(\omega)|');
    title('Output Spectrum for Bandpass filter centred at ' + num2str(fc(ii)) + 'Hz')
end
%% End of filtering

%% Decoder
[max, idx] = maxk(pwr, 2);
% Sorting to ensure that the higher and lower frequency components are in
% ascending order for access expressions to work correctly.
idx = sort(idx);
fprintf('The output power was highest for bandpass filters centered at %d Hz and %d Hz.\n', fc(idx(1)),
fc(idx(2)));

% Access expressions based on chosen table
result = mat_symbol(idx(2)-4, idx(1));
fprintf('The chosen symbol, based on the output power of each filter, was %c.\n', result)

figure();
sgtitle('Output Signal in Time Domain')
subplot(211);
f1 = fc(idx(1)); f2 = fc(idx(2));
stem(t(1:100*floor(Fs/f2)), y_n(idx(1), 1:100*floor(Fs/f2)) + y_n(idx(2), 1:100*floor(Fs/f2)))
grid on
xlabel('Time in seconds');ylabel('y(t)');title('Output signal in time domain (discrete representation)');
subplot(212);
plot(t(1:100*floor(Fs/f2)), y_n(idx(1), 1:100*floor(Fs/f2)) + y_n(idx(2), 1:100*floor(Fs/f2)))
grid on
xlabel('Time in seconds');ylabel('x(t)');title('Output signal in time domain (continuous representation)');
%% Decoder end

```