

Experiment No. 01

1.1 Experiment Name:

Experimental study on DTMF tone detection using MATLAB

1.2 Objectives:

- To become accustomed with the simulation of communication circuits in the MATLAB environment.
- Learn DTMF tone detection model and their operation.
- Learn how to code & run DTMF tone detection for future projects and sessions through MATLAB.

1.3 Theory:

Dual Tone Multi Frequency (DTMF) is a signal toning technique used in communication, ranging from voice mail and help centers to telephone banking and controlling robots' designs. Addition of two sinusoidal tones produces the DTMF signal. One is chosen from a set of 697 Hz, 770 Hz, 852 Hz, and 941 Hz known as the low frequency group, and the other is from a set of 1209 Hz, 1336 Hz, 1477 Hz, and 1633 Hz known as the high frequency group. By combining two sinusoidal tones and four frequencies from each group, 16 combinations were obtained, representing 10 decimal digits, alphabet characters A, B, C, D, and two special characters “*” and “#”.

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

Table 1.1: Frequencies of DTMF tones.

DTMF is a tonal technique for expressing digits in communications. All touch-tone phones employ DTMF tones to represent the digits on a touch tone keypad. Here, ‘Touching’ a button generates a tone which is a combination of two frequencies, one from the above-mentioned lower band, and other from the upper band.

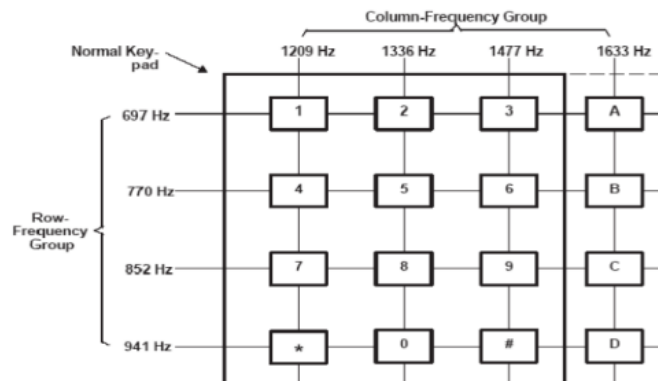


Table 1.2: Touch tone telephone keypad

1.4 Apparatus:

- MATLAB

1.5 MATLAB Code:

```

2  symbol = {'1','2','3','4','5','6','7','8','9','*','0','#'};
3  lfg = [697 770 852 941]; % Low frequency group
4  hfg = [1209 1336 1477]; % High frequency group
5  f = [];
6  for c=1:4
7      for r=1:3
8          f = [ f [lfg(c);hfg(r)] ];
9      end
10 end
11 Fs = 8000; % Sampling frequency 8 kHz
12 N = 800; % Tones of 100 ms
13 t = (0:N-1)/Fs; % 800 samples at Fs
14 pit = 2*pi*t;
15 tones = zeros(N,size(f,2));
16 n = input('Choose any key:');
17 for toneChoice=1:12
18     % Generate tone
19     tones(:,toneChoice) = sum(sin(f(:,toneChoice)*pit));
20     if toneChoice==n
21         keypad = tones(:,n);Fs;
22         sound(keypad);
23     end
24     % Plot tone
25     if toneChoice==n
26         subplot(4,3,toneChoice),plot(t*1e3,tones(:,toneChoice));
27         title(['Symbol "', symbol{toneChoice}, '" :',
28             [' ',num2str(f(1,toneChoice)),', ',num2str(f(2,toneChoice)),', ']]
29         set(gca, 'Xlim', [0 25]);
30         ylabel('Amplitude');
31     end
32     if toneChoice>9, xlabel('Time (ms)'); end
33 end
34 set(gcf, 'Color', [1 1 1], 'Position', [1 1 1280 1024])
35 annotation(gcf,'textbox','Position',[0.38 0.96 0.45 0.026],...
36     'EdgeColor',[1 1 1],...
37     'String','\bf Time response of each tone of the telephone pad', ...
38     'FitBoxToText','on');

```

1.7 Result:



```

>> Communication_exp01_final
Choose any key:9
>> Communication_exp01_final
Choose any key:2
>> Communication_exp01_final
Choose any key:1
fx >>

```

Fig. 1.1: Command window(Input)

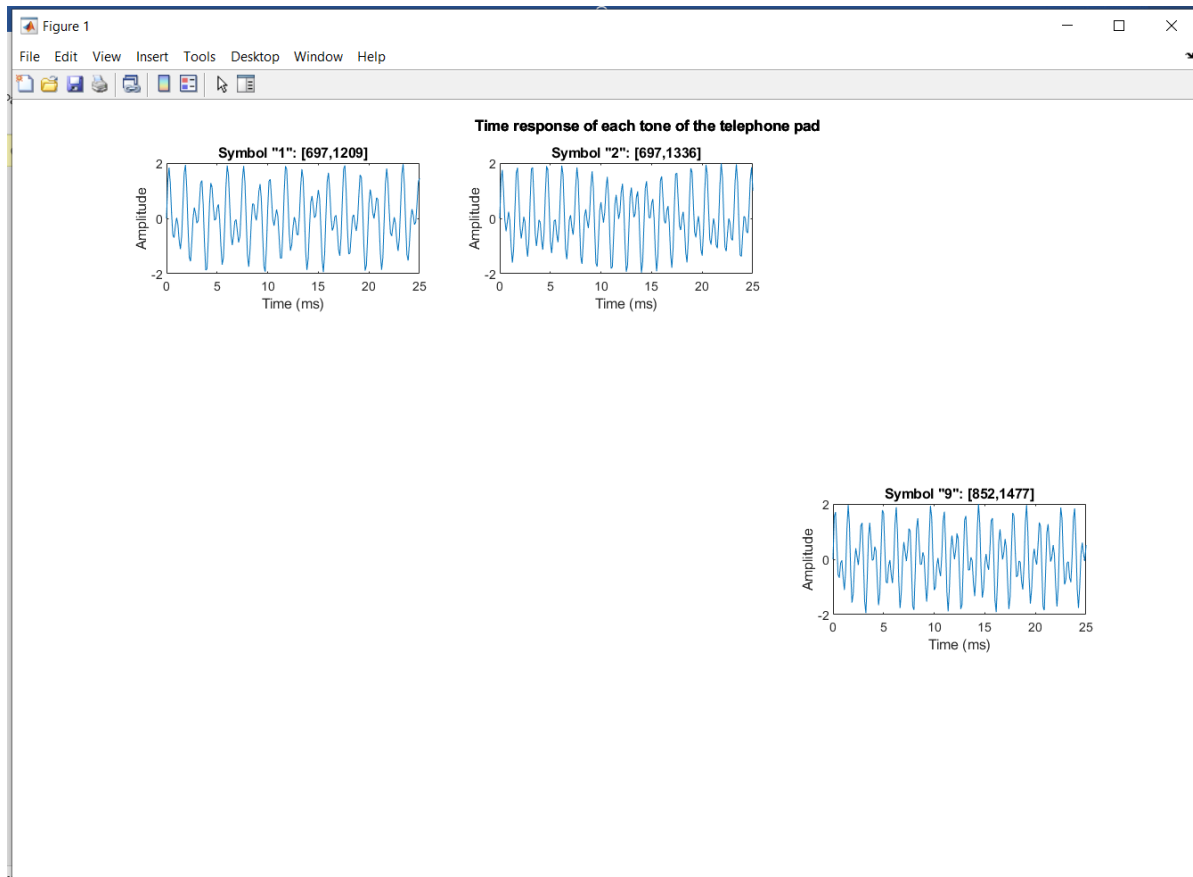


Fig. 1.2: Time response window(output)

1.8 Discussion & Conclusion:

We implemented DTMF signal detection in this study using the MATLAB application. In MATLAB, it can generate the DTMF signals for each of the 16 digits by running the code. In this case, we used this code to obtain signal for digit 1, 2, 9 respectively. As we run the code, we get proper audio tone and waveform for those respective keys.

The DTMF technique is implementable in MATLAB/Simulink, and they can use the fixed-point code generated by MATLAB in real-time telecommunication applications.