

# Experiment 3: Dual Tone Multi Frequency Encoder Decoder (DTMF)

**AIM: To study and Analyse Coder/Decoder using FIR Filters in MATLAB**

Dual Tone Multi Frequency is the basis for telephone system. Each key is mapped to a specified column and row frequency. As the key is pressed, two sinusoids (tones) are generated. These two tones are summed and this “Dual tone” signal now represents the pressed key. The two tones are carefully chosen so that no harmonics occur i.e. No frequency is an integral multiple of the other and the sum/ difference of these two frequencies does not equal to any of the frequencies.

Eg: Pressing the digit 2 will generate the tones 1336 Hz and 697 Hz.

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

Decoding: The dual tone signal is now passed through a filter-bank containing 8 filters. Each filter is a band pass filter with the central frequency tuned to a tone and appropriately sharp cut-off to prevent overlapping with the adjacent tones. As the signal passes through the filter bank, the outputs of the filters tuned to the tones of the input signal are high in amplitude, the rest of the filter outputs are low amplitude outputs. With a simple thresholding scheme, the outputs are now 1s and 0s (digital output). Now this can be decoded using appropriate digital logic to get back the numbers.

For successful decoding, since we do not know the strength of the signal and the output comes over a channel and is hence susceptible to noise, we need the following criteria:

1.  $(P_a + P_b) > 0.75 \cdot P_t$
2.  $\Delta f/f < 1.5\%$

The frequency content in the output is estimated using the Goertzel algorithm, which fills up frequency bins that are predetermined by the user with the frequency content. Taking the max. gives us the frequency content in one such bandpass filter output (since only one tone could possibly result as a filter output).

Saurabh Dash 14EC10050

Aditya Sinha 14EC10002

Goertzel algorithm proves computationally more efficient than FFT for small number of data points. For our consideration, each symbol lasts for 40ms and we have  $8000 \times 0.04 = 320$  data points, which is quite less. Further, it has been shown that to minimise the error, this may be chosen as low as  $N=205$ , rejecting the other data points.

Goertzel algorithm is most typically used as a hardware implemented replacement for the filter bank, as designing such sharp bandpass filters isn't always feasible. This block can directly find the two peaks and estimate the frequency content of the signal, making it an alternative DTMF decoder.

**Code:****Experiment3.m**

```

%clear;
%clc;
%% Definitions
f = [697 770 852 941 1209 1336 1447 1633];
S='123A456B789C*0#D';
Fs=8e3;
T=40e-3;
len=round(5+(30-5)*rand(1)); %sequence length between 5 and 30
sequence=datasample(S,len); %random sequence of key presses
t=0:1/Fs:(len*Fs-1)*T/Fs;
Nt=T*Fs;
freq=(Fs/2)/Nt*(0:Nt-1);
y=zeros(1,8);
keys=zeros(1,len);
power=zeros(1,8);

%SNR=20; %quantifies the input noise
%uncomment if run standalone

%% DTMF Generator and Decoder.
for i=1:len
    y(:)=0;
    signal=sumsinusoid(sequence,i,S,f,Fs); %signal corresponding to input
    key
    signal=signal+peak2peak(signal)/10^(SNR/20)*(-0.5+rand(1,Nt));
    %addition of AWGN

    for j=1:8
        A=fir1(95,[2*f(j)/Fs-20/Fs 2*f(j)/Fs+2*20/Fs]);
        out=filter(A,1,signal);
        power(j)=rms(out)^2;
    end
    [p1,j1]=max(power(1:4));
    y(j1)=1;
    [p2,j2]=max(power(5:8));
    j2=j2+4;
    y(j2)=1;

    if(p1+p2<0.75*sum(power)) %power criterion for valid detection
        y(:)=0;
    end

    % Goertzel algorithm for checking frequency deviation
    A=fir1(95,[2*f(j1)/Fs-20/Fs 2*f(j1)/Fs+2*20/Fs]);
    out=filter(A,1,signal);
    [~,f_est1]=max(goertzel(out,round(freq/Fs*Nt)+1));
    f_est1=freq(f_est1+1);
    A=fir1(95,[2*f(j2)/Fs-20/Fs 2*f(j2)/Fs+2*20/Fs]);
    out=filter(A,1,signal);
    [~,f_est2]=max(goertzel(out,round(freq/Fs*Nt)+1));
    f_est2=freq(f_est2+1);

    if(abs(f_est1-f(j1))>0.015*f(j1)||abs(f_est2-f(j2))>0.015*f(j2))
        y(:)=0; %frequency deviation criterion for valid
detection
end

```

Saurabh Dash 14EC10050

Aditya Sinha 14EC10002

```
% Decoder
index=0;
for j=1:4
    if(y(j))
        index=4*j-4;
    end
end
for j=1:4
    if(y(j+4))
        index=index+j;
    end
end

if(index)
    keys(i)=num2str(S(index));
else
    keys(i)='X'; %in case of invalid detection
end
end

keys=char(keys);
```

Saurabh Dash 14EC10050  
Aditya Sinha 14EC10002

```
function signal=sumsinusoid(sequence,i,S,fr,fc)
T=1;
t=0:T/10000:9999*T/10000;

index=find(S==sequence(i));
temp=mod(index,4);
temp2=floor(index/4)+1;

if (~temp)
    temp=temp+4;
    temp2=temp2-1;
end

f1=fr(temp2);
f2=fc(temp);

signal=sin(2*pi*f1*t)+sin(2*pi*f2*t);
end
```

Saurabh Dash 14EC10050  
Aditya Sinha 14EC10002

Now, next, following the ITU standard, we have taken 8kHz sampling with 40ms time for each key press. The DTMF generated is with respect to random sequence of random length. To account for input noise, we have also added AWGN in the input signal (Additive White Gaussian Noise). This corrupts the input signal and pushes up the level of Power Spectral Density equally everywhere. As a result, the power criterion may fail as if  $P_a + P_b > 0.75 * P_t$ , it is not necessary that

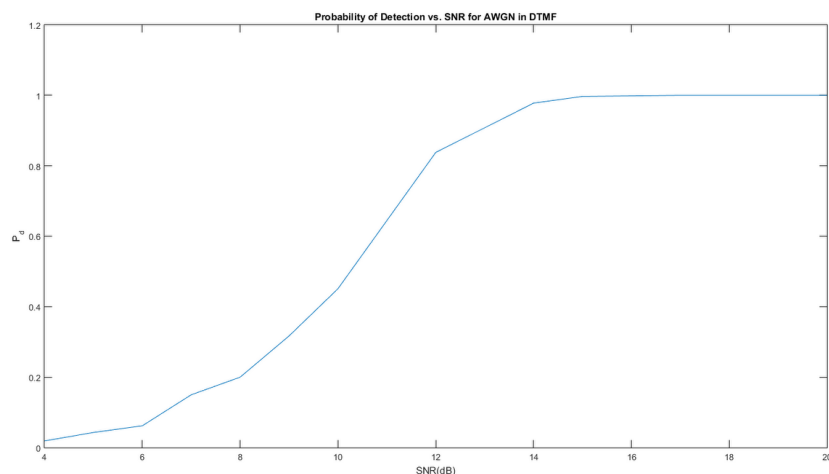
$$(P_a + P_b + 2P) > 0.75 * (P_t + 8P)$$

Here, we do not have a possibility of frequency deviation, as AWGN does not corrupt relative frequency content, but pushes the level at each frequency equally. However, if we modelled channel non-linearity and distortion, we would also get some frequency deviation and the frequency criterion could also be violated.

For plotting the Probability of Detection vs. SNR, we take a particular value and find the statistical probability of correct detection (all keys of input sequence should match decoded output). For this, for each value of SNR, run ~1000 trials with random input sequences. Here is the wrapper code around **Experiment3.m** that accomplishes this.

#### **ProbabilityDetectionVsSNR.m**

```
arr=[4 5 6 7 8 9 10 12 14 15 17 20];  
Pd=zeros(1,length(arr));  
  
for k=1:length(arr)  
    SNR=arr(k);  
    for test=1:1000                                %compute statistical probability for  
        given SNR  
            Experiment3;  
            if (~any(keys-sequence))  
                Pd(k)=Pd(k)+1/1000;  
            end  
        end  
    end  
end  
  
plot(arr,Pd);  
xlim=[4 20];  
title('Probability of Detection vs. SNR for AWGN in DTMF');  
xlabel('SNR(dB)');  
ylabel('P_d');
```



## Discussions:

### Aditya Sinha (14EC10002)

1. Due to inherent non-linearity of the channel, we have intermodulation distortion, which causes second and higher order terms in the spectrum of output. So, for successful decoding, we have to ensure that the two input tones are such that the output when band pass filtered does not give an erroneous decoded value. For this we ensure that the second and higher order terms do not coincide with any of the other frequencies. So, we choose  $f$ 's such that no frequency is an integral multiple of the other and the difference or sum of any two frequencies should not equal any of the frequencies.
2. Here the sampling frequency is chosen as 10kHz, and as  $10/2 > \max(f) = 1633$ , it satisfies the Nyquist criterion and the band pass filtered output will be as desired. In practice, since the channel on which the dual tone is transmitted is the same as the audio channel (telephone line routed through operator), the sampling frequency will be the same as the standard for audio.
3. Depending upon the proximity of the frequencies, the required sharpness of bpf and threshold applied for decoding will change. If they are closer, we'll need higher order bpf and a higher thresholding for calculating binary  $y$ .
4. Depending on the selectivity of the bandpass filter, we'll have different power distributions. So, to meet the power criterion, it is imperative that we use a high order filter (95 in our case). However, if we have a frequency deviation, we would need to estimate this as we want to see if frequency criterion is satisfied. For this, we cannot have a very sharp filter. This high order and contradiction is further motivation for us to ditch the filter bank and go for the Goertzel algorithm implementation.
5. We have taken the sequence length varying from 5 to 30. If we took a larger length, we would have lesser probability of correct detection of the sequence for a given SNR.

### Saurabh Dash (14EC10050)

1. In this experiment while choosing the sampling frequency, it was ensured that the sampling frequency was greater than twice the maximum frequency, which in this case was 1633Hz.
2. We could have used on one tone per symbol but then that would lead to complication in the band pass filter design as the spectrum would now be more crowded and a sharper and greater order of filter would be required for decoding.
3. The frequencies for DTMF are chosen such that none of them have a harmonic relation with the others and that mixing the frequencies would not produce sum or product frequencies that could mimic another valid tone.