## EXPERIMENT-2

**AIM:** To generate DTMF Signals

**SOFTWARE AND HARDWARE USED:** Matlab 24a and Personal PC

**THEORY:**

Dual-Tone Multi-Frequency (DTMF) signaling is a method used in telecommunication to send information through a telephone line using audible tones. Each DTMF signal is composed of two simultaneous tones, one from a low-frequency group and one from a high-frequency group, representing a specific key on a telephone keypad. This combination of tones allows for the unique identification of each number (0-9) and other keys such as *, #, A, B, C, and D. The frequencies used in DTMF are standardized: the low-frequency group includes 697 Hz, 770 Hz, 852 Hz, and 941 Hz, while the high-frequency group includes 1209 Hz, 1336 Hz, 1477 Hz, and 1633 Hz. For example, pressing the '1' key generates a combination of 697 Hz and 1209 Hz, while the '0' key combines 941 Hz and 1336 Hz. DTMF is widely used in telecommunication systems, especially in interactive voice response (IVR) systems, where users can navigate menus or input information by pressing keys on their phone. The signals are transmitted over the voice channel and are easily recognized by electronic equipment, enabling automated systems to process the inputs. Originally developed for touch-tone telephones, DTMF has become a standard in the telecommunications industry due to its reliability and simplicity. It is still used today in various applications, including remote control of systems, secure data entry, and even in amateur radio for control purposes.

**CODE:**

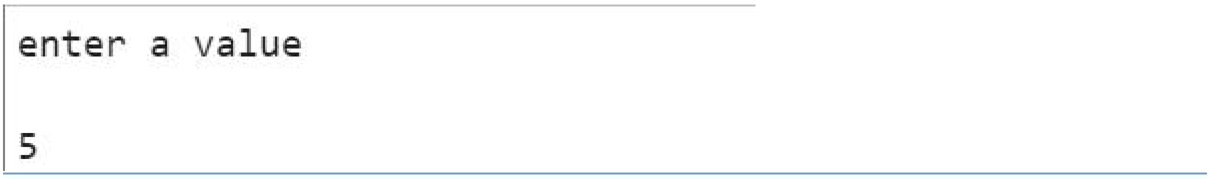
```
function [y]=sines_filter(fs,f)

t=0:1/fs:0.002;

y=sin(2*pi*f*t);

subplot(4,1,1)

plot(t,y)

%sgtitle("SINE SIGNAL USING FILTERING-22071A04B4")

%xlabel("time")

%ylabel("amplitude")

%title("sine signal")
```

```
n=-2:1:200;

x=zeros(1,length(n));

x(n==0)=1;

w=2.*pi.*(f/fs);

b=[0 sin(w) 0];

a=[1 -2.*cos(w) 1]';

p=filter(b,a,x);

%subplot(4,1,3)

plot(p)

axis([0 200 -1 1])

%title("sinusoidal signal using inbuilt command")

end

clc;

close all;

fs=8000;

fr1=697;

fr2=770;

fr3=852;

fr4=941;

fc1=1209;

fc2=1336;

fc3=1477;
```

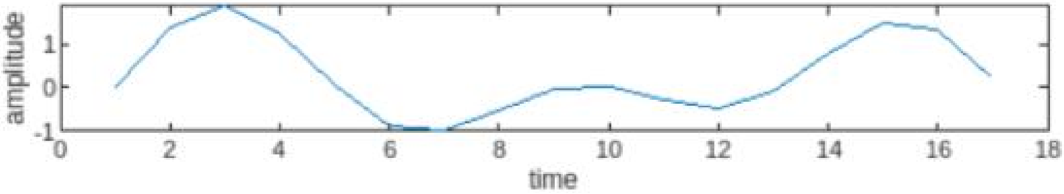
```
while(1)

m=input("enter a value");

if(m==1)

    y1=sines_filter(fs,fr1);

    y2=sines_filter(fs,fc1);

end

if(m==2)

    y1=sines_filter(fs,fr1);

    y2=sines_filter(fs,fc2);

end

if(m==3)

 y1=sines_filter(fs,fr1);

    y2=sines_filter(fs,fc3);

end

if(m==4)

    y1=sines_filter(fs,fr2);

    y2=sines_filter(fs,fc1);

end

if(m==5)

    y1=sines_filter(fs,fr2);

    y2=sines_filter(fs,fc2);

end
```

```
if(m==6)

    y1=sines_filter(fs,fr2);

    y2=sines_filter(fs,fc3);

end

if(m==7)

    y1=sines_filter(fs,fr3);

    y2=sines_filter(fs,fc1);

end

if(m==8)

    y1=sines_filter(fs,fr3);

    y2=sines_filter(fs,fc2);

end

if(m==9)

    y1=sines_filter(fs,fr3);

    y2=sines_filter(fs,fc3);

end

if(m==0)

    y1=sines_filter(fs,fr4);

    y2=sines_filter(fs,fc1);

end

if(m=='*')

    y1=sines_filter(fs,fr4);
    y2=sines_filter(fs,fc2);

end

if(m=='#')

    y1=sines_filter(fs,fr4);

    y2=sines_filter(fs,fc3);

end

y=y1+y2

plot(y)

soundsc(y)

end
```

**OUTPUT:**



```
enter a value

5
```



**RESULT:** Hence the sinusoidal signal using filtering is generated and verified.