Transmission and Selective Retrieval of Audio Signals using FDM

Aadhitya S CB.EN.U4ECE18101
Aadityan E CB.EN.U4ECE18102
Aruneswari S CB.EN.U4ECE18107
Chandhana S CB.EN.U4ECE18112
Nirali M Dave CB.EN.U4ECE18136
R R Prathiksha CB.EN.U4ECE18144

***Abstract* — Multiple signals (ranging from 2-5) given as an input to a common channel will be modulated separately using different modulation techniques such as Double Side-Band Suppressed Carrier modulation (DSB-SC) Technique, Single Side-Band Technique and will be multiplexed using Frequency Division Multiplexing Algorithm (FDM) to ensure no mixing, overlapping, interference of any two or more signals takes place. Upon transmission, the user gets to choose the signal that he/she desires to listen to and the desired signal will be received at the receiver end.**

I. INTRODUCTION

Why multiplexing?

When we have multiple message signals, to transmit all of them, we'd require multiple channels but in order to reduce the number of channels, thereby also reducing the cost to materialise the channels, we come up with the concept of multiplexing.

Types of multiplexing:

1. Frequency Division Multiplexing (FDM)
2. Wavelength Division Multiplexing (WDM) and,
3. Time Division Multiplexing (TDM).

Frequency division multiplexing (FDM):

Signals of different frequencies are combined for concurrent transmission. The total bandwidth is divided into a set of frequency bands that are far apart from each other. This is done by modulating each message signal with different carrier signal frequencies. The modulated signals are combined and transmitted through a single communication channel, thus allowing multiple independent data streams to be transmitted simultaneously. Finally, it is demodulated at the receiver end and the original signal is extracted.

Applications of FDM are found in the working of television networks, FM and AM radio broadcasting, first generation cellular telephone, etc.

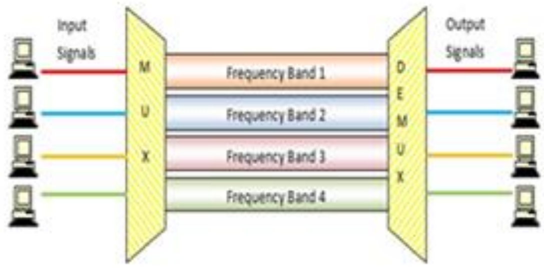

Fig. 1: FDM

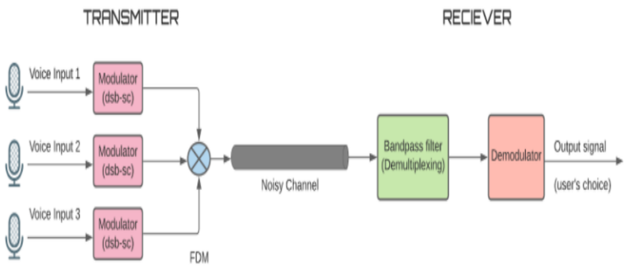

Fig. 2: Flowchart showing the steps involved in FDM

II. PROPOSED WORK

- We aim to transmit two or more different voice/audio signals through a single noise-less channel after modulating them individually with different carrier frequencies. The modulation techniques used here are DSB-SC and SSB.
- Making use of frequency division multiplexing technique, the voice signals are transmitted and received in the receiver end.
- To separate the modulated voice signals from the composite signal, band pass filters are used.
- Then, the signals are to be demodulated and passed through a low pass filter to obtain the original input signals. Here, the user will be given a choice to select the voice/audio that he/she wishes to listen to.

Audio extraction from the user:

- Object creation using the audiorecorder module
- Recording the user voice - recordblocking class function
- Storing the UserData to a local variable - s1 (getaudiodata module)
- Storing length s1 for truncation

```
noc= 1;
nob=16;%8,16,24
recObj = audiorecorder(Fs,nob,noc);
disp("Recording Voice 1 for 3- seconds ");
```

```
Recording Voice 1 for 3- seconds
```

```
recordblocking(recObj,3);
s1= getaudiodata(recObj);
longituds1 = length(s1);
```

Fig. 3: Code snippet showing audio recording block

```
recObj =

  audiorecorder with properties:

      SampleRate: 44100
   BitsPerSample: 16
     NumChannels: 1
        DeviceID: -1
   CurrentSample: 1
    TotalSamples: 132300
         Running: 'off'
        StartFcn: []
         StopFcn: []
        TimerFcn: []
     TimerPeriod: 0.0500
             Tag: ''
        UserData: []
            Type: 'audiorecorder'
```

Fig. 4: audiorecorder properties

The truncation block is used to ensure that the audio-data array is of the same size.

```
%truncation_block
longitud_minima = min([longituds1 longituds2]);
t = linspace(0,3,longitud_minima);
s1 = s1(1:longitud_minima);
s2 = s2(1:longitud_minima);
```

Fig. 5: Truncation block

III. PROCEDURE 1: USING DSB-SC MODULATION

Plot of extracted audio data:

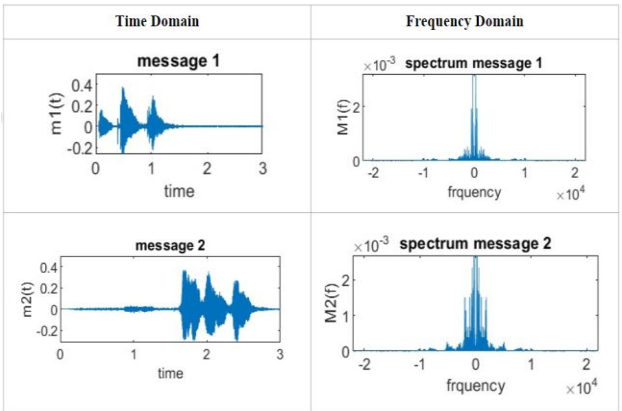

Fig. 6: Input audio signals

Modulation: Double Sideband Suppressed-Carrier AM

In the DSB-SC modulation, unlike in standard AM, the wave carrier is not transmitted and the modulated wave has the information only in the sidebands; thus, much of the power is distributed between the side bands.

A DSB-SC AM signal is obtained by multiplying the message signal *m(t)* with the carrier signal $c(t)=A_c \cos(2\pi f_c t)$

Thus, we have the amplitude-modulated signal $u(t) = m(t)\, c(t) = A_c\, m(t) \cos(2\pi f_c t)$

Bandwidth: $(f_c + f_m) - (f_c - f_m) = 2f_m$ or 2W (W is the highest frequency of the message signal).

Spectrum of the DSB-SC AM Signal:

$$U(f) = \frac{A_c}{2}[M(f - f_c) + M(f + f_c)]$$

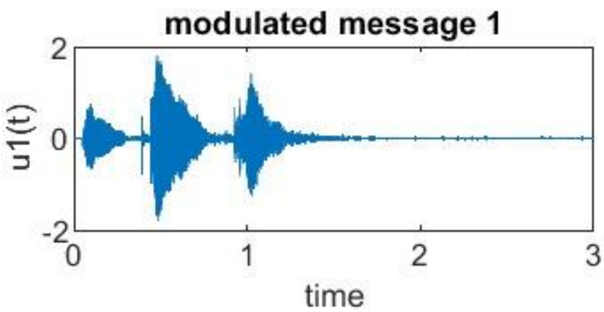

Fig. 7: Modulated input signal 1

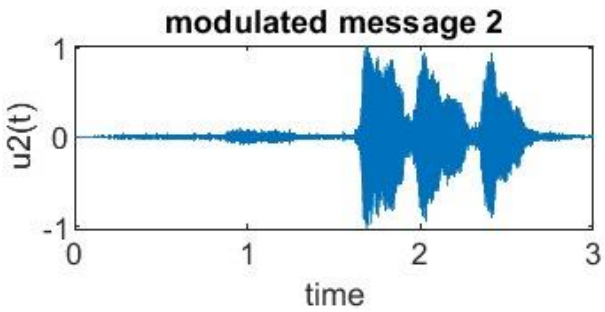

Fig. 8: Modulated input signal 2

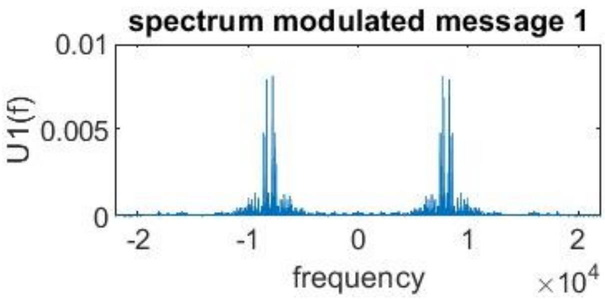

Fig. 9: Spectrum of modulated input signal 1

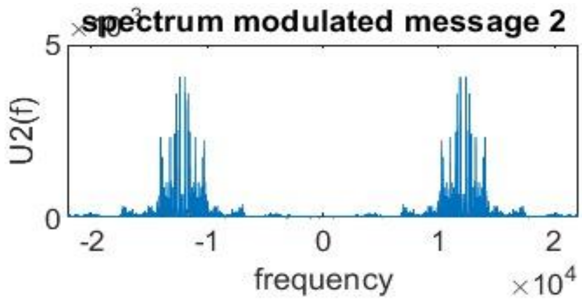

Fig. 10: Spectrum of modulated input signal 2

The carrier frequencies used to modulate the voice signals are selected in such a way that there is a minimum separation of 2W (where W is the bandwidth of the voice signal). This prevents any overlap between the modulated signals or any interference between their independent spectrums.

The carrier frequency used:

$$f_{c1} = 2 \times W$$

$$f_{c2} = 8 \times W$$

Therefore, the separation between the carrier frequencies is greater than 2W.

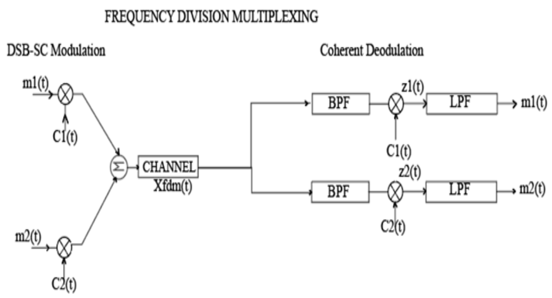

Fig. 11: FDM in detail

By using this technique, we combine the two DSB-SC signals to produce a composite signal $X_{FDM}(t)$.

$$X_{FDM}(t) = m_1(t) \times c_1(t) + m_2(t) \times c_2(t)$$

FDM implementation in MATLAB:

Composite signal:

```
net_sig = modsig2+modsig1
```

where modsig1 is the modulated signal 1 and modsig2 is the modulated signal 2 as mentioned before.

Conversion of composite signal from time domain to frequency domain:

```
fft_netsig= fftshift(fft(net_sig,N)/N)
```

The spectrum of the composite signal:

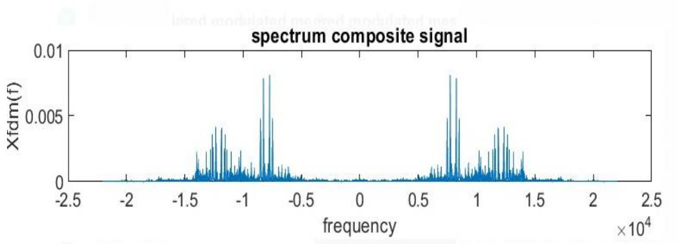

Fig. 12: Spectrum of composite signal

Filter the spectrum of the modulated signal-1 and signal-2 from the composite spectrum.

As shown in the figure, we need to separate the spectrum of the modulated signals, so that we can demodulate and retrieve both the signals separately.

Here, we use bandpass filters such that the frequency range of the filter lies within the given range of frequencies. The bandpass filter is designed using the inbuilt Butterworth function. It is given by:

Butter (order, [normalized cut off frequency range])

In our case, to extract the spectrum of modulated signal-1, the range of frequency of chosen:

(freq_carrier1-freq_cutoff) the upper band frequency and (freq_carrier1+freq_cutoff) the lower band cut off frequency.

The order chosen is 4. As the order increases the Butterworth bandpass filter is said to be more ideal. Hence the order chosen is 4.

Similarly, to extract the spectrum of modulated signal-2, the range of frequency chosen is the same as shown above with the carrier frequency_2.

(freq_carrier2-freq_cutoff) the upper band frequency and (freq_carrier2+freq_cutoff) the lower band cut off frequency.

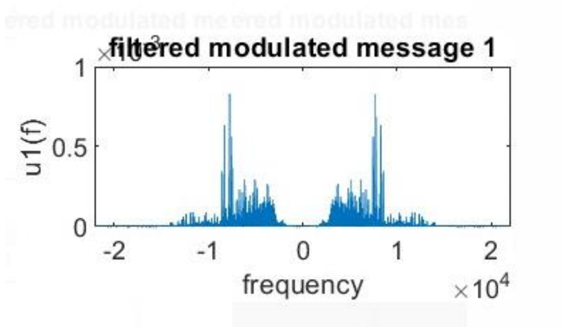

Fig. 13: Output after passing through bandpass filter -1

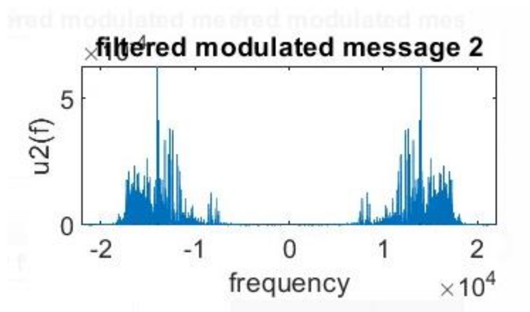

Fig. 14: Output after passing through band pass filter-2

Demodulation: Double Sideband Suppressed-Carrier AM

The process of extracting an original message signal from a DSB-SC wave is known as detection or demodulation of DSB-SC. Here we use a coherent detector demodulator for the demodulation process. The same carrier signal (which is used for generating DSB-SC signal) is used to detect the message signal. Hence, this process of detection is called coherent or synchronous detection. In this process, the message signal can be extracted from the DSBSC wave by multiplying it with a carrier, having the same frequency and the phase of the carrier used in DSBSC modulation. The resulting signal is then passed through a Low Pass Filter. Output of this filter is the desired message signal.

Let the DSBSC wave be $s(t)=A_c \cos(2\pi f_c t)\, m(t)$. The output of the local oscillator is $c(t)= A_c \cos(2\pi f_c t)$.

Therefore, the output of the product modulator is

$v(t) = s(t)\, c(t)$

$v(t)=(A_c^2/2)\, m(t) + (A_c^2/2)\cos(4\pi f_c t)\, m(t)$

Thus, the output of the low pass filter is

$V_o(t) = (A_c^2/2)\, m(t)$

These band pass filtered modulated message signals are demodulated using a coherent demodulator and passed through the low pass filter to retrieve the original message signals.

```
%coherent-DSBSC Demodulation
filter_input1=filter_s1.*c1;
%low pass filter
sig1_t=pass_band(filter_input1);
filter_input2=filter_s2.*c2;
%low pass filter
sig2_t=pass_band(filter_input2);
```

Fig. 15: Code showing demodulation

The output of these filters will give the message signal whose amplitude has been halved. Thus, the filters' output is plotted with twice its amplitude.

```
subplot(3,2,5);
plot(t,real(2*sig1_t));
title('output message 1');
xlabel('time');
ylabel('m1(t)');
subplot(3,2,6);
plot(t,real(2*sig2_t));
title('output message 2');
xlabel('time');
ylabel('m2(t)');
```

Fig. 16: Code for plotting the graphs

Low pass filter specifications:

The demodulated signal is passed through the low pass filter and the message signal is extracted. The low pass filter is designed using the inbuilt Butterworth filter in MATLAB.

Butter (order, normalized cut off frequency);

Cutt_off freq=2500Hz

This is because the frequency range of an audio signal is from 300-2500Hz. Hence the highest frequency component of the audio signal could be 2500 in this case. But the parameter is the normalized cut off frequency which would be, $freqcutoff\ /\ (fs/2)$

The output after passing demodulated message signal-1 through low pass filter:

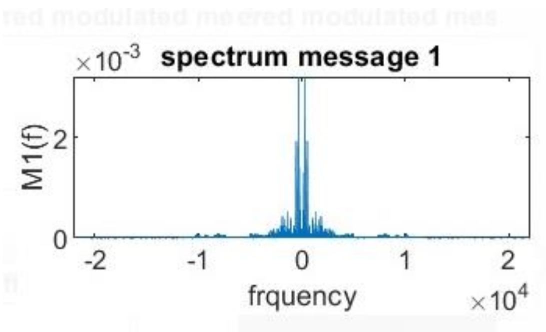

Fig. 17: Spectrum of demodulated signal 1

The output after passing demodulated message signal-2 through low pass filter:

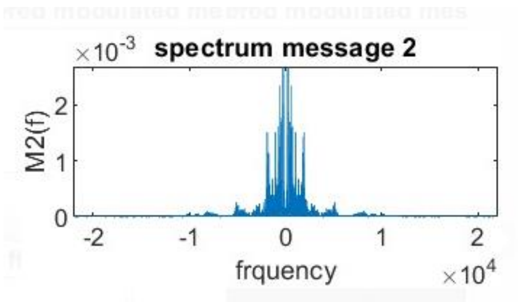

Fig. 18: Spectrum of demodulated signal 2

The final retrieved message signals:

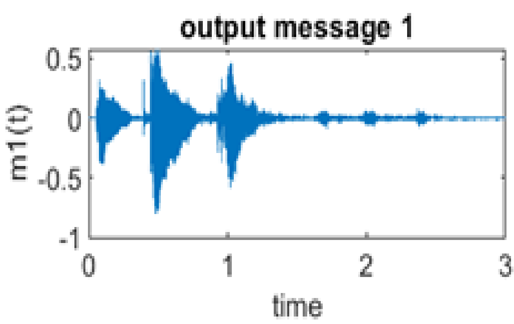

Fig. 19: Retrieved message signal 1

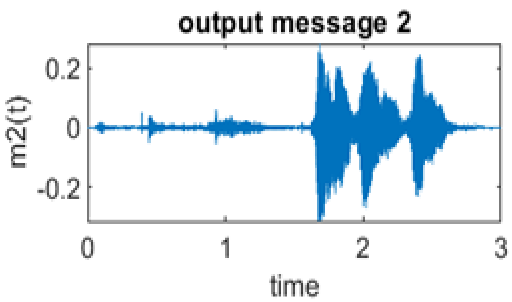

Fig. 20: Retrieved message signal 2

IV. PROCEDURE 2: USING SSB-SC MODULATION

In the SSB-SC modulation, unlike in DSB-SC modulation, either one of the sidebands is sufficient to reconstruct the message signal m(t) at the receiver. Thus, we reduce the bandwidth of the transmitted signal to that of the baseband message signal m(t).

(SSB) AM signal is representation:

$$u(t) = A_c m(t) \cos 2\pi f_c t \mp A_c \hat{m}(t) \sin 2\pi f_c t$$

Frequency Discrimination method:
Generation of an SSB-AM signal by filtering one of the sidebands of a DSB-SC AM signal.
The frequency range of the band pass filter is selected as the spectrum of the desired SSB-SC wave. It either can be tuned to upper sideband or lower sideband frequencies to get the respective SSBSC wave having upper sideband or lower sideband.

The separation between the Upper sideband and the lower sideband must be equal to twice the lowest frequency component of the message signal. So that it is wide enough to accommodate the transition band of Band Pass Filter.

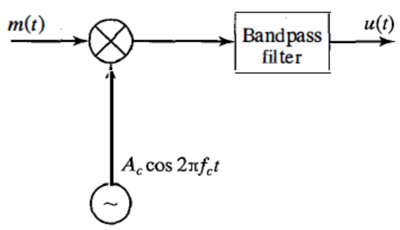

Fig. 21: Frequency discrimination method

Implementation of SSB-SC modulation in MATLAB:
Base parameters:

```
bandwidth = 4000
audio_min= 300
```

Bandwidth of the voice signal is 4k Hz.

The lowest frequency component of the message signal equals 300Hz.

Therefore, the separation between the Upper sideband and the lower sideband must be at least 600Hz.

Carrier frequency initialization:

```
frec_carrier1 = bandwidth*3
frec_carrier2 = bandwidth*4
```

The minimum separation between the carrier frequencies is W (Bandwidth of the voice signal).

Audio signal generation:

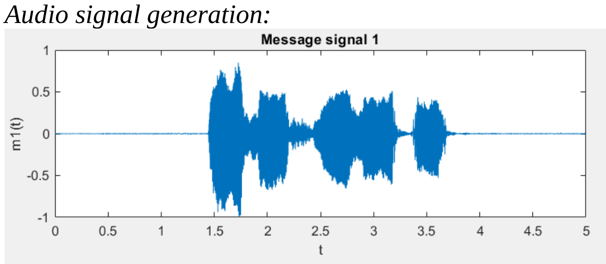

Fig. 22: Input signal 1

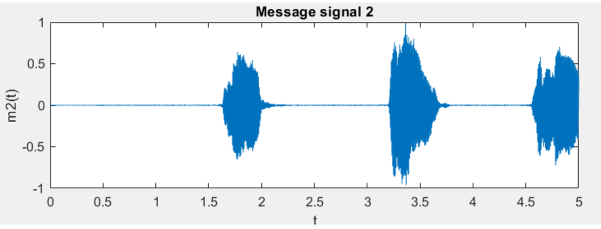

Fig. 23: Input signal 2

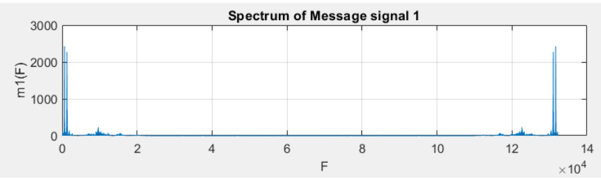

Fig. 24: Spectrum of input signal 1

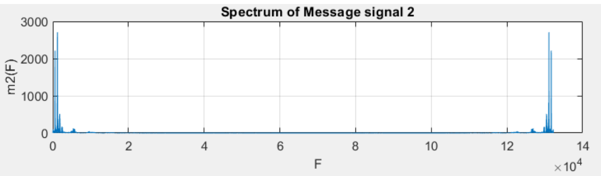

Fig. 25: Spectrum of input signal 1

```
s1mod = ssbmod(s1,frec_carrier1,Fs)
s2mod = ssbmod(s2,frec_carrier2,Fs)
```

The SSB-SC modulation is implemented using the inbuilt function in MATLAB called *ssbmod*.
The parameters of ssbmod:

- S1 =Voice signal 1 and S2 =Voice signal 2
```
s1= getaudiodata(recObj)
s2= getaudiodata(recObj)
```
- Carrier frequencies of the respective carrier signals.
```
frec_carrier1 = bandwidth*3
frec_carrier2 = bandwidth*4
```
- The carrier signal and the voice signal must have sample frequency Fs (Hz). *Fs= 44.1k Hz*

The modulated signal has zero initial phase and retains the lower side band and the upper side band is eliminated.

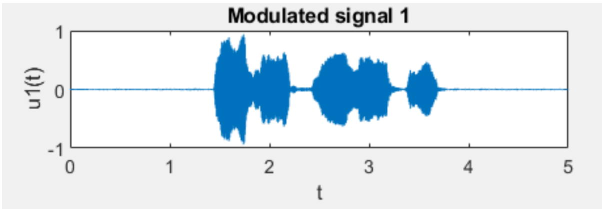

Fig. 26: SSB Modulated input signal 1

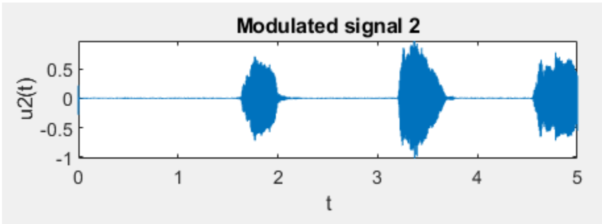

Fig. 27: SSB Modulated input signal 2

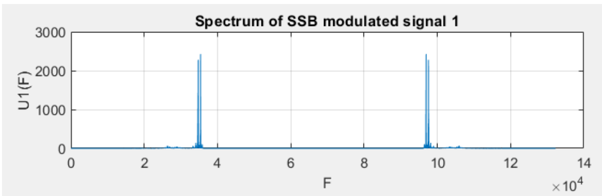

Fig. 28: Spectrum of modulated input signal 1

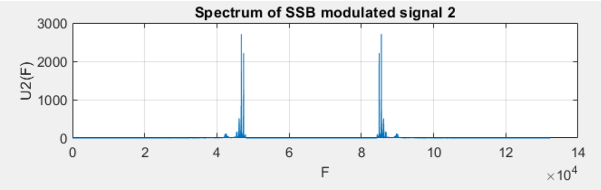

Fig. 28: Spectrum of modulated input signal 1

FDM implementation:

```
fs1 = s1mod;
fs2 = s2mod;
net_signal = fs1+fs2;
figure
net_signalf=abs(fft(net_signal));
```

Fig. 29: Code showing FDM implementation

The carrier frequencies used to modulate the voice signals are selected in such a way that there is a minimum separation of W(where W is the bandwidth of the voice signal). This prevents any overlap between the modulated signals or any interference between their independent spectrums.

By using this technique, we combine the two SSB-SC signals to produce a composite signal net_signalf.

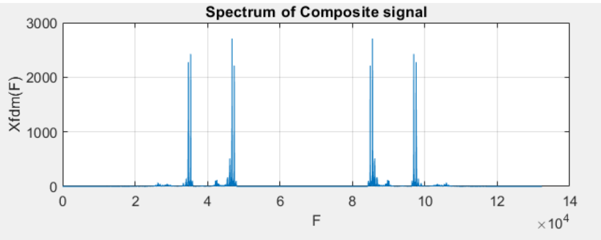

Fig. 30: Spectrum of composite signal

Filters and their specifications:
The FDM'ed signal now has message components at different frequencies, namely at the two bandwidths across the two carrier frequencies, fc1 and fc2. To now separate the two messages, passing the FDM'ed signal through two bandpass filters is required. Hence, the bandpass filters are designed with different passband and stopband frequencies such that only one of the two signals is extracted in both bandpass filters.

Filter coefficients:
A filter is defined using its filter coefficients. As an analogy, a circle has infinite points. So, one way to define a circle is to say all the infinite points present on the circumference of the circle but that's inhumane. Hence a clever way of defining it would be to fix its radius and the centre and the same circle can now be drawn. Similarly, the filter is defined by its coefficients. They are a set of values arranged in a row.

Filter frequencies:
In the case of SSB-SC, the bandwidth was 4kHz and the carrier frequencies were 2*BW and 3*BW respectively. Therefore, the passband and stopband of the two filters are separated by at least 600Hz if the BW of the audio signal is 2700Hz, which ensures that there's no overlap of signals.

Upon extracting the two message signals, it's still modulated and upon demodulating, the message lies in the baseband and in 2fc. Therefore, a requirement of a lowpass filter arises and the signals are passed through them to extract the message effectively.

Filter order:
A Butterworth filter is used in the project to realize LPFs and BPFs.

An interesting specification of the Butterworth filter is its order. The lowpass filter is of order 4 while that of the BPF is of order 2. Why?

As the order increases, the filter approaches ideality (as in, the steepness from pass to stopband increases) but at the cost of ripples near the passband and stopband.

So, for the design of a LPF, care must be taken to ensure it's as ideal as possible since even the tiniest amount of higher extra frequency components can corrupt the original signal. Hence, order 4 is used.

In the latter case of the BPF, the design is such that there is a minimum of 600Hz gap between the two bands of signals. Hence, even if the ideality of the filter reduces, there isn't going to be any overlap since the gap is high. Hence, a 2nd order filter will suffice.

Demodulation of SSB-SC:
Upon extracting the original message signal after passing them through the bandpass filter, demodulation is required. Hence, to implement it, we have to multiply the filtered modulated output with the carrier signal again. The resulting signal now has to be passed through the above mentioned and designed low-pass filter to retrieve the original signal since upon demodulation, there will be message components at two frequency components, one near baseband and one at 2*fc. Hence, the requirement of the LPF arises.

Shown below are the demodulated signals after passing them through the two bandpass filters. As mentioned above, we have to now pass these two demodulated signals through lowpass filters to retrieve the message lying in the baseband.

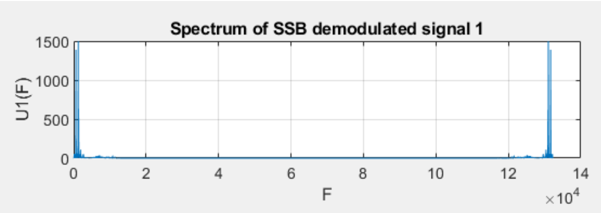

Fig. 31: Spectrum of SSB demodulated signal1

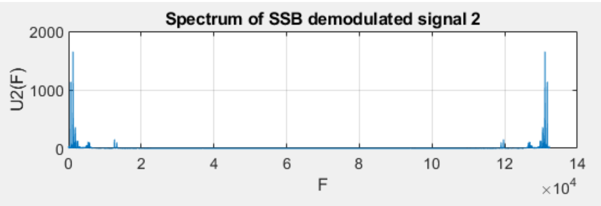

Fig. 32: Spectrum of SSB demodulated signal2

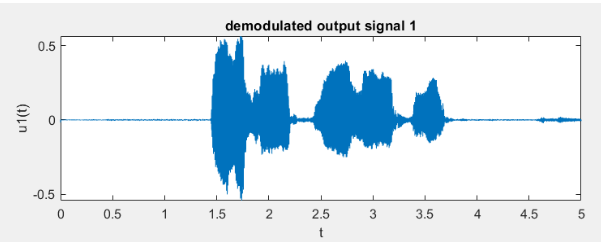

Fig. 33: Demodulated output signal 1

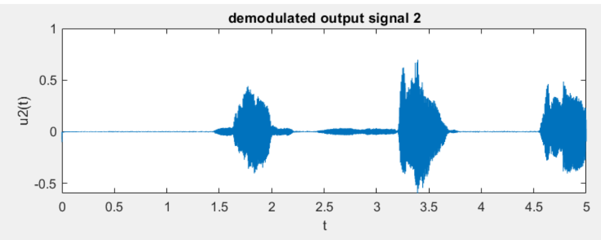

Fig. 34: Demodulated output signal 2

Notice that upon demodulating and passing them through the lowpass filter, the output graph is similar to the original input graph shown previously. Therefore, while playing the final audio, we will be able to successfully listen to the original input without any distortion or noise of any form.

To reiterate, this distortion less, noise free output was possible because of the appropriate choice of the Butterworth filter order as even the slightest error in choosing the order of the Butterworth filter would have led to a highly distorted, corrupted output.

V. RESULT

- As mentioned in the proposed work, the two output signals have been retrieved from the FDM'ed composite signal after passing it through a band pass filter and demodulation. In both DSB-SC and SSB procedures, this has been achieved.
- As per the user's choice, he/she is given an option to select the audio file that he/she wants to hear. When the user enters the choice, the respective audio file is played.

```
while k==5
choice = input("Whose voice you want to extract");
if choice==1
player4 = audioplayer(demods1,44100);
playblocking(player4);
end

if choice==2
player5 = audioplayer(demods2,44100);
playblocking(player5);
end
k = input("do you want to cotinue? (y=5/n=6)");
end
```

VI. REFERENCES

[1] John G. Proakis and Masoud Salehi, *Communication Systems Engineering (2nd edition)*
[2] Simon Haykin and Michael Moher, *Introduction to Analog and Digital Communications (2nd edition)*
[3] John G. Proakis and Masoud Salehi, *Fundamentals of Communication Systems (2nd edition)*

Code	Course Outcomes
15ECE301.2.CO01	Able to understand the basic principles of signal modulation and demodulation.
15ECE301.2.CO02	Able to analyze the time domain and frequency domain representations of amplitude and angle modulated signals.
15ECE301.2.CO04	Able to apply the concepts of modulation and demodulation for the design of communication systems.

DSB-SC modulation and demodulation, FDM, filter design for DSB-SC and debugging are done by:
Aadityan E CB.EN.U4ECE18102
Aruneswari S CB.EN.U4ECE18107
Chandhana S CB.EN.U4ECE18112

SSB-SC modulation and demodulation, FDM, filter design for SSB-SC and debugging of MATLAB code are done by:
Aadhitya S CB.EN.U4ECE18101
Nirali M Dave CB.EN.U4ECE18136
R R Prathiksha CB.EN.U4ECE18144