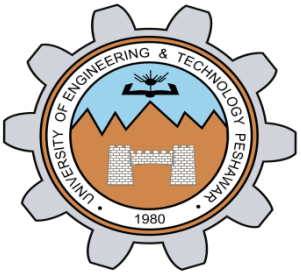
Lab Report No 8



Digital Signal Processing

Submitted By: Usman Yaqoob

Registration No: 19PWCSE1754

Section: B

"On my honor , as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work"

Student Signature:

Department of Computer Systems Engineering

University of Engineering and Technology Peshawar

Task:

Implement all the steps below in MATLAB.

STEP 1, the signals are reproduced as they arrive

Code:

```
%task 1
%Playingthe signals as they arrive
clc;
close all;
%reading audio files
%reading audio file of male 1
wait = input("Signal are played on the speaker firstly:\n");
[y1,Fs] = audioread('male1.wav');
sound(y1,Fs);
pause(6);   %pause of 6 seconds till the next audio will be
played

%reading female voice
[y2,Fs1] = audioread('female.wav');
sound(y2,Fs1);
pause(5);  %pasue of 5 seconds

%reading male voice 2
[y3,Fs2] = audioread('male2.wav');
sound(y3,Fs2);
pause(5);
```

STEP 2, plot the spectra of the signals as they arrive

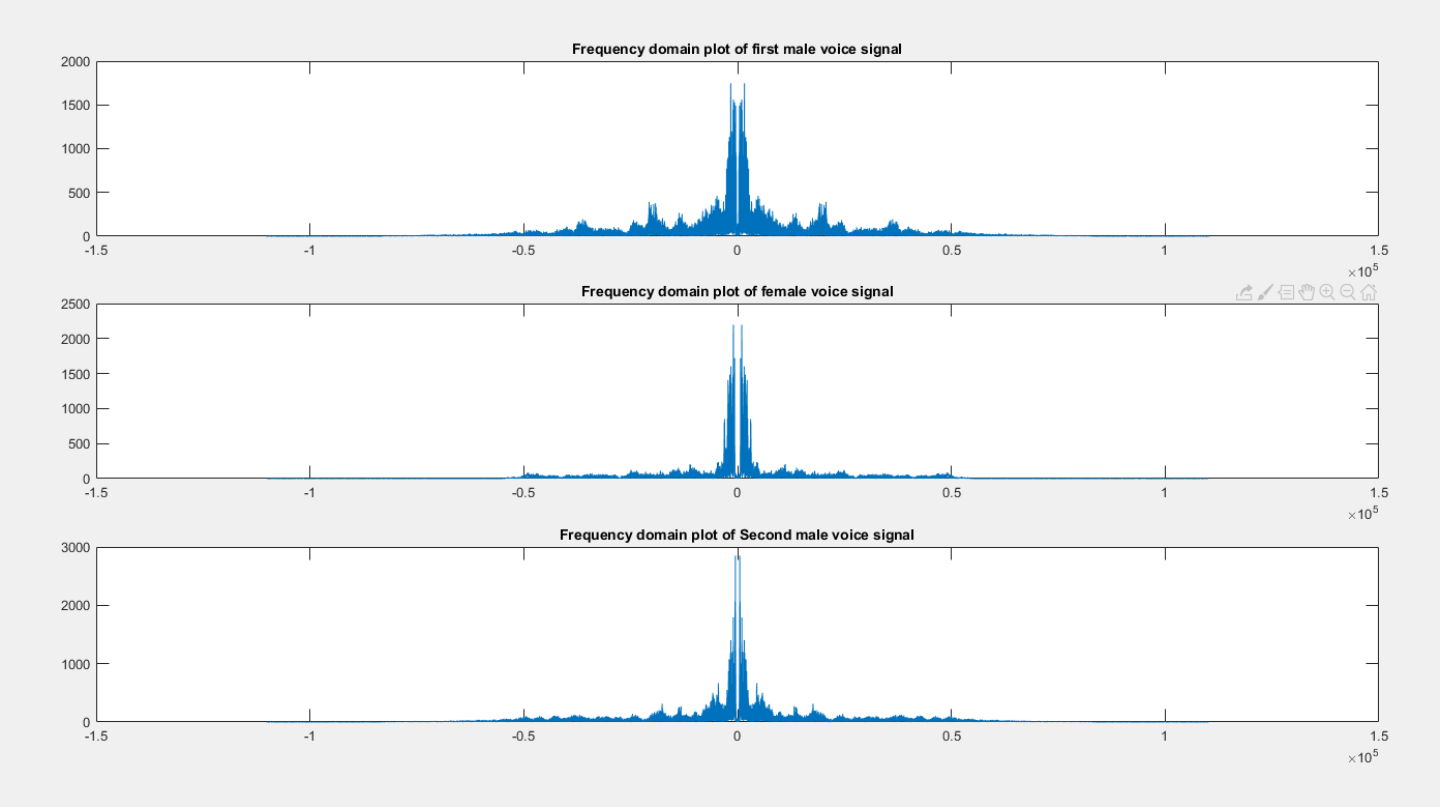
Code:

```
%task 2
%Converting to frequency domain and plotting them
%storing male signal1 in variabel signal1
```

```
signal1 = y1;
%storing female signal in variable signal2
signal2 = y2;
%storing male signal 2 in variabel signal 3
signal3 = y3;
%now we have to convert them into frequecny domain
wait1 = input("Plot in the frequency Domain:\n");
%This fft function used below returns the Discrete Fourier
Transform with
%the length equals to the length of signal given
fsig1 = fft(signal1,length(signal1));
len = length(fsig1);
fshift = (-len/2:len/2-1);
%basically ffshift function rearrange the fourier transform that
we found
%through fft by shifting zero frequency component center of
array
shifted = (fftshift(fsig1));    %for y-axis
subplot(3,1,1)
%plotting the frequency domain of male signal 1
plot(fshift,abs(shifted));
title("Frequency domain plot of first male voice signal")

%now converting female voice singal to frequecny domain

fsig2 = fft(signal2,length(signal2));
len = length(fsig2);
fshift = (-len/2:len/2-1);
shifted = (fftshift(fsig2));
subplot(3,1,2)
%plotting the frequency domain of female signal
plot(fshift,abs(shifted));
title("Frequency domain plot of female voice signal")



%now converting male voice singal 2 to frequecny domain

fsig3 = fft(signal3,length(signal3));
len = length(fsig3);
```

```
fshift = (-len/2:len/2-1);
shifted = (fftshift(fsig3));
subplot(3,1,3)
%plotting the frequency domain of female signal
plot(fshift,abs(shifted));
title("Frequency domain plot of Second male voice signal")
```

Ouput:



STEP 3, The signals are passed through a low pass filter

Code:

```
%task 3
%passing the signals from low pass filter to get the desired
frequency
%range singal
wait2 = input("Pass the signals from the low pass filter: \n");
%Fs = 44100 for all three signals
```

```
%Assuming F = 2500
F = 1900;
%Making butter worth filter where F/(Fs/2) is the cut off
frequency
%and 4 is bsically n
%n is order of the filter and this butter will return transfer
function
%coeffiecient b and a
[b,a] = butter(4,F/(Fs/2));
%now passing signals one by one from the filter
%using transfer function coefficiects and first signal y1
filtered1 = filter(b,a,y1);
%using transfer function coefficiects and second signal y2
filtered2 = filter(b,a,y2);
%using transfer function coefficiects and third signal y3
filtered3 = filter(b,a,y3);
```

STEP 4, reproduce the signals after passing them through the filter
Code:

```
%task 4
%Playing the low pass filtered signal
wait3 = input("Play the low pass filtered signal: \n");
%playing the filtered signal 1 using the sound()
sound(filtered1,Fs);
%giving pause of 6 seconds
pause(6);

%playing the filtered signal 2
sound(filtered2,Fs);
pause(5);

%playing filtered signal 3
sound(filtered3,Fs);
pause(5);
```

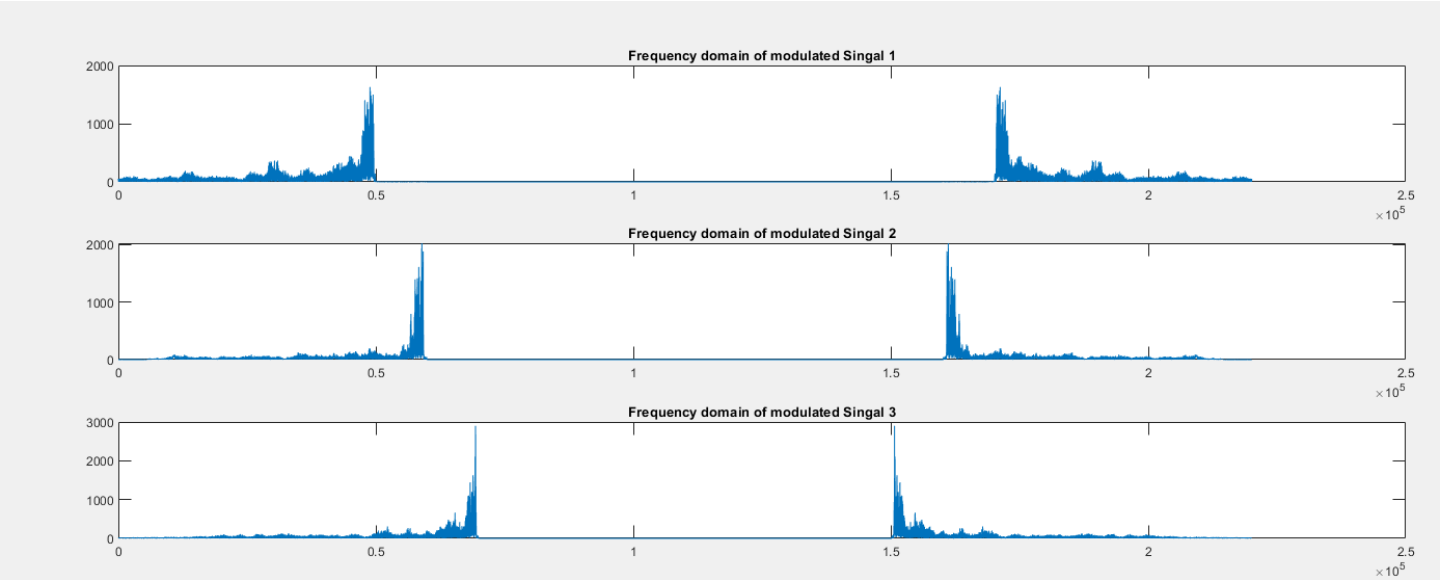
STEP 5, The signals are modulated to different carriers

Code:

```
%task 5
%The signals are modulated to different carriers
%Firstly for the modulation we need high frequency carrier
signals
%Carrier for the signal 1
fcarrier1 = 10000;
%carrier for the signal 2
fcarrier2 = 12000;
%carrier for the signal 3
fcarrier3 = 14000;

wait4 = input("The signals are modulated to different carriers
and added: \n");
%matrix was not matching while multiplexing so this to match the
matrix
length1 = length(y1);    %storing length of signal 1
length2 = length(y2);    %storing length of signal 2

%so we will find the minimum between 2 lengths
length_min = min([length1 length2]);
%making signal of length min
y1 = y1(1:length_min);
%making signal 2 of length min
y2 = y2(1:length_min);
%making signal 3 of length min
y3 = y3(1:length_min);

%here is the main part of modulation
%We will use ssbmod() for modulation
%ssbmod is basically single side band modulation that transmits
single band
%modulation of signal 1
modulated1 = ssbmod(y1,fcarrier1,Fs);
%modulation of signal 2
modulated2 = ssbmod(y2,fcarrier2,Fs);
```

```
%modulation of signal 3
modulated3 = ssbmod(y3,fcarrier3,Fs);

%note: Fs is same for all the signals
%Now plotting the frequency domain of the modulated signals
%frequency domain of ist modulated signal
fmodulated1 = abs(fft(modulated1));
subplot(4,1,1)
%plotting
plot(fmodulated1)
title("Frequency domain of modulated Singal 1");

%frequency domain of second modualted signal
fmodulated2 = abs(fft(modulated2));
subplot(4,1,2)
plot(fmodulated2)
title("Frequency domain of modulated Singal 2");

%frequency domain of third modualted signal
fmodulated3 = abs(fft(modulated3));
subplot(4,1,3)
plot(fmodulated3)
title("Frequency domain of modulated Singal 3");
```

Output:

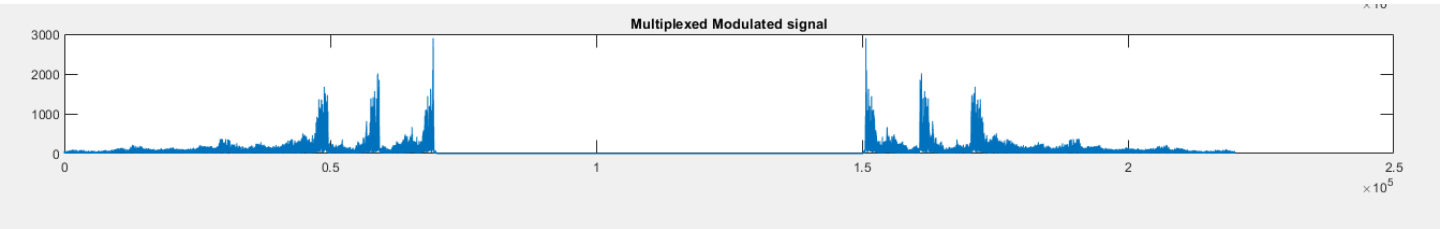


STEP 6, The modulated signals are filtered in the defined bands and added

Code:

```
%task 6
%multiplexing of the signals
%so we will modulated the signals just by adding the signals
multiplexed  = modulated1  + modulated2 + modulated3;
subplot(4,1,4)
%converting the multiplexed signal 2 frequency domain
fmultiplexed = abs(fft(multiplexed));
plot(fmultiplexed)
```

Output:



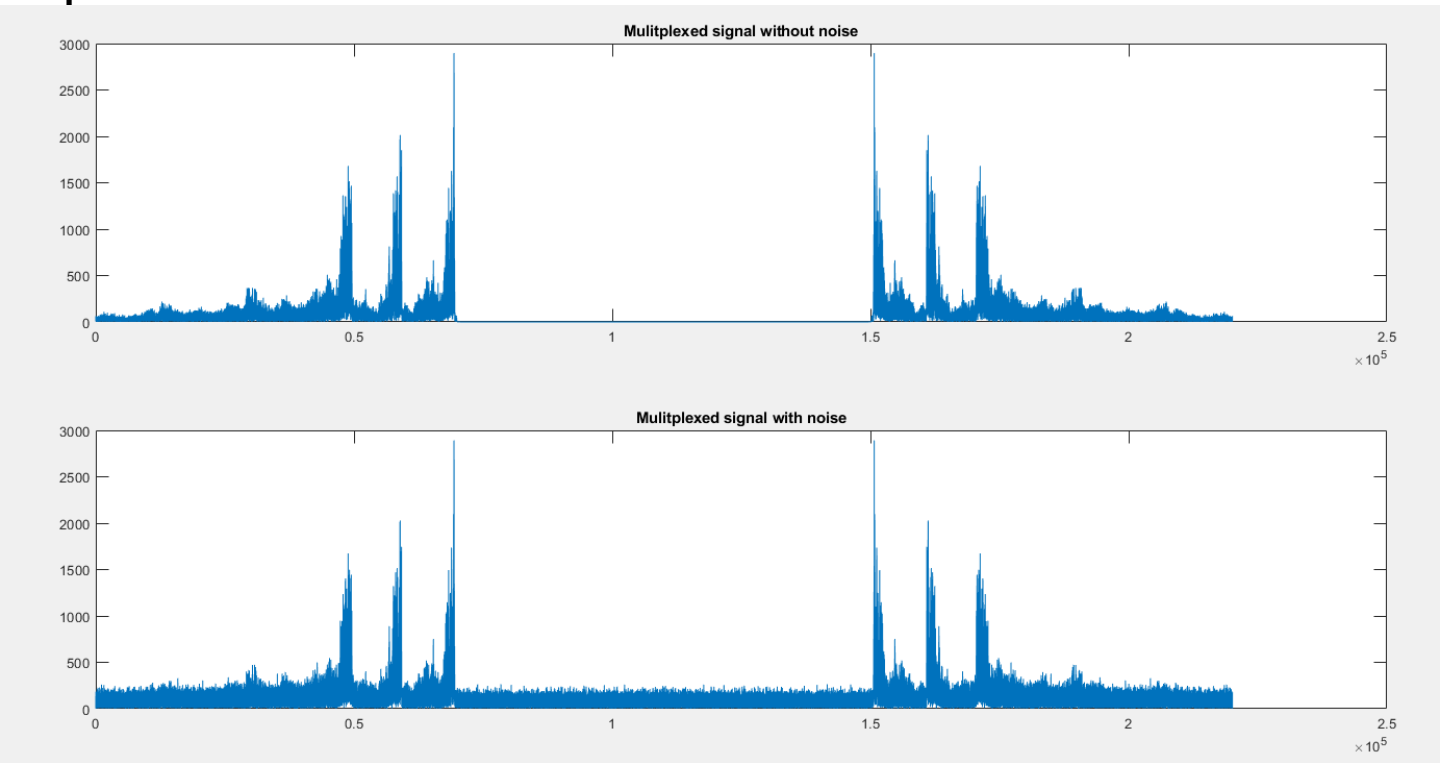
STEP 7, some noise is added to the transmitted signal

Code:

```
%task 7
%adding noise
%awgn() will add white gaussian noise to signal
%it taskes the signal as input and value of snr
%basically snr is singal to noise ratio is ratio of signal power
and noise
wait5 = input("Adding noise to singal: \n");
noise = awgn(multiplexed,15);
%plotting normal multiplexed signal
subplot(2,1,1)
plot(fmultiplexed)
title("Mulitplexed signal without noise");
%now converted noised signal to frequency domain
fnoise = abs(fft(noise));
```

```
subplot(2,1,2)
%plotting noisy signal
plot(fnoise)
title("Mulitplexed signal with noise");
```

Output:



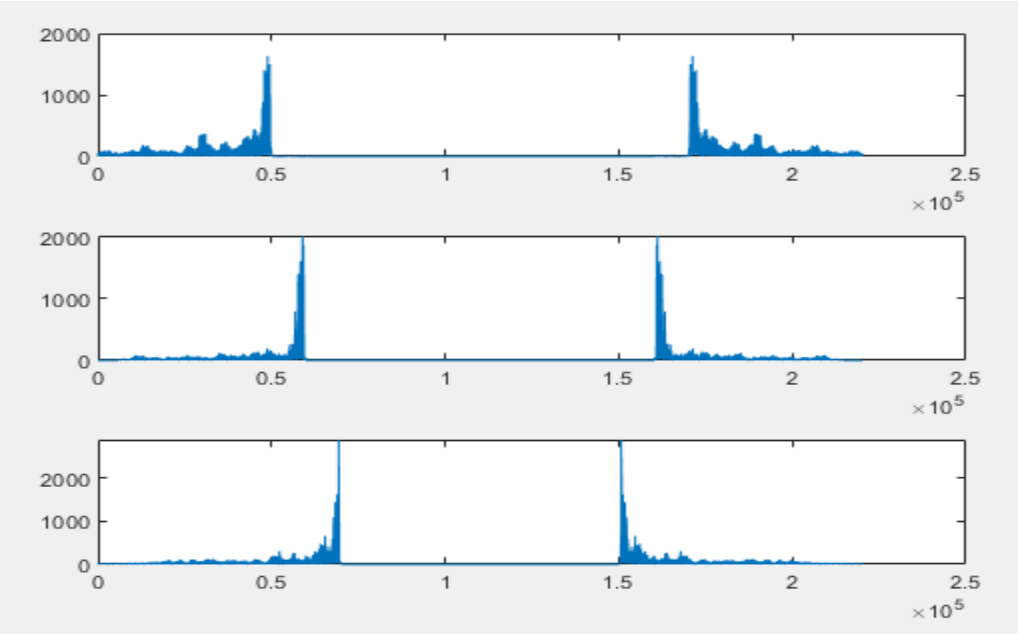
STEP 8, upon arrival each band is filtered

Code:

```
%task 8
%demultiplexing the signal now
%signal 1 will be addition of modulated 2 and 3
wait6 = input("Now demultiplexing the multiplexed signal: \n");
signal1 = modulated2+modulated3;
%so we can get our demultilplexed signal 1 by subtracting
signal1 from
%total multiplexed signal
```

```
demultiplexed_signal1 = multiplexed - signal1;
subplot(3,1,1)
%shifting to frequency doamin and plotting
plot(abs(fft(demultiplexed_signal1)))
%same logic to obtain rest 2
%signal 2 will be addition of modulated 1 and 3
signal2 = modulated1+modulated3;
%so we can get our demultilplexed signal 2 by subtracting
signal2 from
%total multiplexed signal
demultiplexed_signal2 = multiplexed - signal2;
subplot(3,1,2)
plot(abs(fft(demultiplexed_signal2)))
%signal 2 will be addition of modulated 1 and 2
signal3 = modulated1+modulated2;
%so we can get our demultilplexed signal 3 by subtracting
signal3 from
%total multiplexed signal
demultiplexed_signal3 = multiplexed - signal3;
subplot(3,1,3)
plot(abs(fft(demultiplexed_signal3)))
```

Output:



STEP 9, each recovered band is demodulated to return the signal to the baseband frequency

Code:

```
%task 9
%demodulation of the singals
%as we have done the single side band modulation so demodulation
will also
%be single side band using ssbdemod()
%demodulation of ist demultiplexed signal
wait7 = input("Demodulation to get the orignal signal: \n");
demodulated_signal1 =
ssbdemod(demultiplexed_signal1,fcarrier1,Fs);
%demodulation of second demultiplexed signal
demodulated_signal2 =
ssbdemod(demultiplexed_signal2,fcarrier2,Fs);
%demodulation of third demultiplexed signal
demodulated_signal3 =
ssbdemod(demultiplexed_signal3,fcarrier3,Fs);
```

STEP 10, the recovered signal is passed through a low pass filter

Code:

```
%task 10
%passig through low pass filter
%we have to remove noise
%making butter worth filter again
Wait8 = input("passing from low pass filter to remove noise:
\n");
[b,a] = butter(4,F/(Fs/2));
%passing the ist demodulated signal from the filter
noise_removed1 = filter(b,a,demodulated_signal1);
%passing the second demodulated signal from the filter
noise_removed2 = filter(b,a,demodulated_signal2);
%passing the third demodulated signal from the filter
noise_removed3 = filter(b,a,demodulated_signal3);
```

STEP 11, play the reproduced signal after transmission

Code:

```
%task 11
%playing signals after removing noise
Wait9 = input("Signal are played on the speaker at the end
without noise: \n");
%for playing signal we will use sound()
%playing signal 1
sound(noise_removed1,Fs);
pause(6);    %pause of 9 seconds till the next audio will be
played

%playing signal 2
sound(noise_removed2,Fs1);
pause(5);   %pasue of 19 seconds

%palying signal 3
sound(noise_removed3,Fs2);
pause(5);
```

Complete .m File of Task:

```
%task 1
%Playingthe signals as they arrive
clc;
close all;
%reading audio files
%reading audio file of male 1
wait = input("Signal are played on the speaker firstly:\n");
[y1,Fs] = audioread('male1.wav');
sound(y1,Fs);
pause(6);   %pause of 9 seconds till the next audio will be played

%reading female voice
[y2,Fs1] = audioread('female.wav');
sound(y2,Fs1);
pause(5);  %pasue of 19 seconds

%reading male voice 2
[y3,Fs2] = audioread('male2.wav');
sound(y3,Fs2);
pause(5);



%task 2
%Converting to frequency domain and plotting them
%storing male signal1 in variabel signal1
signal1 = y1;
%storing female signal in variable signal2
signal2 = y2;
%storing male signal 2 in variabel signal 3
signal3 = y3;
%now we have to convert them into frequecny domain
wait1 = input("Plot in the frequency Domain:\n");
%This fft function used below returns the Discrete Fourier Transform with
%the length equals to the length of signal given
fsig1 = fft(signal1,length(signal1));
len = length(fsig1);
fshift = (-len/2:len/2-1);
%basically ffshift function rearrange the fourier transform that we found
%through fft by shifting zero frequency component center of array
shifted = (fftshift(fsig1));    %for y-axis
subplot(3,1,1)
%plotting the frequency domain of male signal 1
plot(fshift,abs(shifted));
```

```
title("Frequency domain plot of first male voice signal")

%now converting female voice singal to frequecny domain

fsig2 = fft(signal2,length(signal2));
len = length(fsig2);
fshift = (-len/2:len/2-1);
shifted = (fftshift(fsig2));
subplot(3,1,2)
%plotting the frequency domain of female signal
plot(fshift,abs(shifted));
title("Frequency domain plot of female voice signal")



%now converting male voice singal 2 to frequecny domain

fsig3 = fft(signal3,length(signal3));
len = length(fsig3);
fshift = (-len/2:len/2-1);
shifted = (fftshift(fsig3));
subplot(3,1,3)
%plotting the frequency domain of female signal
plot(fshift,abs(shifted));
title("Frequency domain plot of Second male voice signal")

%task 3
%passing the signals from low pass filter to get the desired frequency
%range singal
wait2 = input("Pass the signals from the low pass filter: \n");
%Fs = 44100 for all three signals
%Assuming F = 2500
F = 2500;
%Making butter worth filter where F/(Fs/2) is the cut off frequency
%and 4 is bsically n
%n is order of the filter and this butter will return transfer function
%coeffiecient b and a
[b,a] = butter(4,F/(Fs/2));
%now passing signals one by one from the filter
%using transfer function coefficiects and first signal y1
filtered1 = filter(b,a,y1);
%using transfer function coefficiects and second signal y2
filtered2 = filter(b,a,y2);
%using transfer function coefficiects and third signal y3
filtered3 = filter(b,a,y3);
```

```
%task 4
%Playing the low pass filtered signal
wait3 = input("Play the low pass filtered signal: \n");
%playing the filtered signal 1 using the sound()
sound(filtered1,Fs);
%giving pause of 6 seconds
pause(6);

%playing the filtered signal 2
sound(filtered2,Fs);
pause(5);

%playing filtered signal 3
sound(filtered3,Fs);
pause(5);

%task 5
%The signals are modulated to different carriers
%Firstly for the modulation we need high frequency carrier signals
%Carrier for the signal 1
fcarrier1 = 10000;
%carrier for the signal 2
fcarrier2 = 12000;
%carrier for the signal 3
fcarrier3 = 14000;

wait4 = input("The signals are modulated to different carriers and added:
\n");
%matrix was not matching while multiplexing so this to match the matrix
length1 = length(y1);    %storing length of signal 1
length2 = length(y2);    %storing length of signal 2

%so we will find the minimum between 2 lengths
length_min = min([length1 length2]);
%making signal of length min
y1 = y1(1:length_min);
%making signal 2 of length min
y2 = y2(1:length_min);
%making signal 3 of length min
y3 = y3(1:length_min);

%here is the main part of modulation
%We will use ssbmod() for modulation
```

```
%ssbmod is basically single side band modulation that transmits single
band
%modulation of signal 1
modulated1 = ssbmod(y1,fcarrier1,Fs);
%modulation of signal 2
modulated2 = ssbmod(y2,fcarrier2,Fs);
%modulation of signal 3
modulated3 = ssbmod(y3,fcarrier3,Fs);

%note: Fs is same for all the signals

%Now plotting the frequency domain of the modulated signals

%frequency domain of ist modulated signal
fmodulated1 = abs(fft(modulated1));
subplot(4,1,1)
%plotting
plot(fmodulated1)
title("Frequency domain of modulated Singal 1");

%frequency domain of second modualted signal
fmodulated2 = abs(fft(modulated2));
subplot(4,1,2)
plot(fmodulated2)
title("Frequency domain of modulated Singal 2");



%frequency domain of third modualted signal
fmodulated3 = abs(fft(modulated3));
subplot(4,1,3)
plot(fmodulated3)
title("Frequency domain of modulated Singal 3");



%task 6
%multiplexing of the singals
%so we will modulated the signals just by adding the signals
multiplexed  = modulated1  + modulated2 + modulated3;
subplot(4,1,4)
%converting the multiplexed signal 2 frequency domain
fmultiplexed = abs(fft(multiplexed));
plot(fmultiplexed)
title("Multiplexed Modulated signal");
```

```
%task 7
%adding noise
%awgn() will add white gaussian noise to signal
%it taskes the signal as input and value of snr
%basically snr is singal to noise ratio is ratio of signal power and noise
wait5 = input("Adding noise to singal: \n");
noise = awgn(multiplexed,15);
%plotting normal multiplexed signal
subplot(2,1,1)
plot(fmultiplexed)
title("Mulitplexed signal without noise");

%now converted noised signal to frequency domain
fnoise = abs(fft(noise));
subplot(2,1,2)
%plotting noisy signal
plot(fnoise)
title("Mulitplexed signal with noise");



%task 8
%demultiplexing the signal now
%signal 1 will be addition of modulated 2 and 3
wait6 = input("Now demultiplexing the multiplexed signal: \n");
signal1 = modulated2+modulated3;
%so we can get our demultilplexed signal 1 by subtracting signal1 from
%total multiplexed signal
demultiplexed_signal1 = multiplexed - signal1;
subplot(3,1,1)
%shifting to frequency doamin and plotting
plot(abs(fft(demultiplexed_signal1)))

%same logic to obtain rest 2
%signal 2 will be addition of modulated 1 and 3
signal2 = modulated1+modulated3;
%so we can get our demultilplexed signal 2 by subtracting signal2 from
%total multiplexed signal
demultiplexed_signal2 = multiplexed - signal2;
subplot(3,1,2)
plot(abs(fft(demultiplexed_signal2)))

%signal 2 will be addition of modulated 1 and 2
```

```
signal3 = modulated1+modulated2;
%so we can get our demultilplexed signal 3 by subtracting signal3 from
%total multiplexed signal
demultiplexed_signal3 = multiplexed - signal3;
subplot(3,1,3)
plot(abs(fft(demultiplexed_signal3)))



%task 9
%demodulation of the singals
%as we have done the single side band modulation so demodulation will also
%be single side band using ssbdemod()
%demodulation of ist demultiplexed signal
wait7 = input("Demodulation to get the orignal signal: \n");
demodulated_signal1 = ssbdemod(demultiplexed_signal1,fcarrier1,Fs);
%demodulation of second demultiplexed signal
demodulated_signal2 = ssbdemod(demultiplexed_signal2,fcarrier2,Fs);
%demodulation of third demultiplexed signal
demodulated_signal3 = ssbdemod(demultiplexed_signal3,fcarrier3,Fs);




%task 10
%passig through low pass filter
%we have to remove noise
%making butter worth filter again
wait8 = input("passing from low pass filter to remove noise: \n");
[b,a] = butter(4,F/(Fs/2));
%passing the ist demodulated signal from the filter
noise_removed1 = filter(b,a,demodulated_signal1);
%passing the second demodulated signal from the filter
noise_removed2 = filter(b,a,demodulated_signal2);
%passing the third demodulated signal from the filter
noise_removed3 = filter(b,a,demodulated_signal3);



%task 11
%playing signals after removing noise
wait9 = input("Signal are played on the speaker at the end without noise:
\n");
%for playing signal we will use sound()
%playing signal 1
sound(noise_removed1,Fs);
pause(6);   %pause of 9 seconds till the next audio will be played
```

```
%playing signal 2
sound(noise_removed2,Fs1);
pause(5);  %pasue of 19 seconds

%palying signal 3
sound(noise_removed3,Fs2);
pause(5);
```