Ahsanullah University of Science and Technology

Department of Electrical and Electronic Engineering

LAB PROJECT REPORT

Course no: EEE 3218

Course name: Digital Signal Processing Laboratory

Prepared by:

Group no:3

Student name and ID:	Indra Dewan	190105001	Israt Jahan Tasmia	190105003
	Fayzan Kowshik Khan	190105007	Abeer Khan	190105008
	Akash Barua	190105009	Nafisa Anjum Mahi	190105010

Session: 3rd year 2nd semester

Section: A1

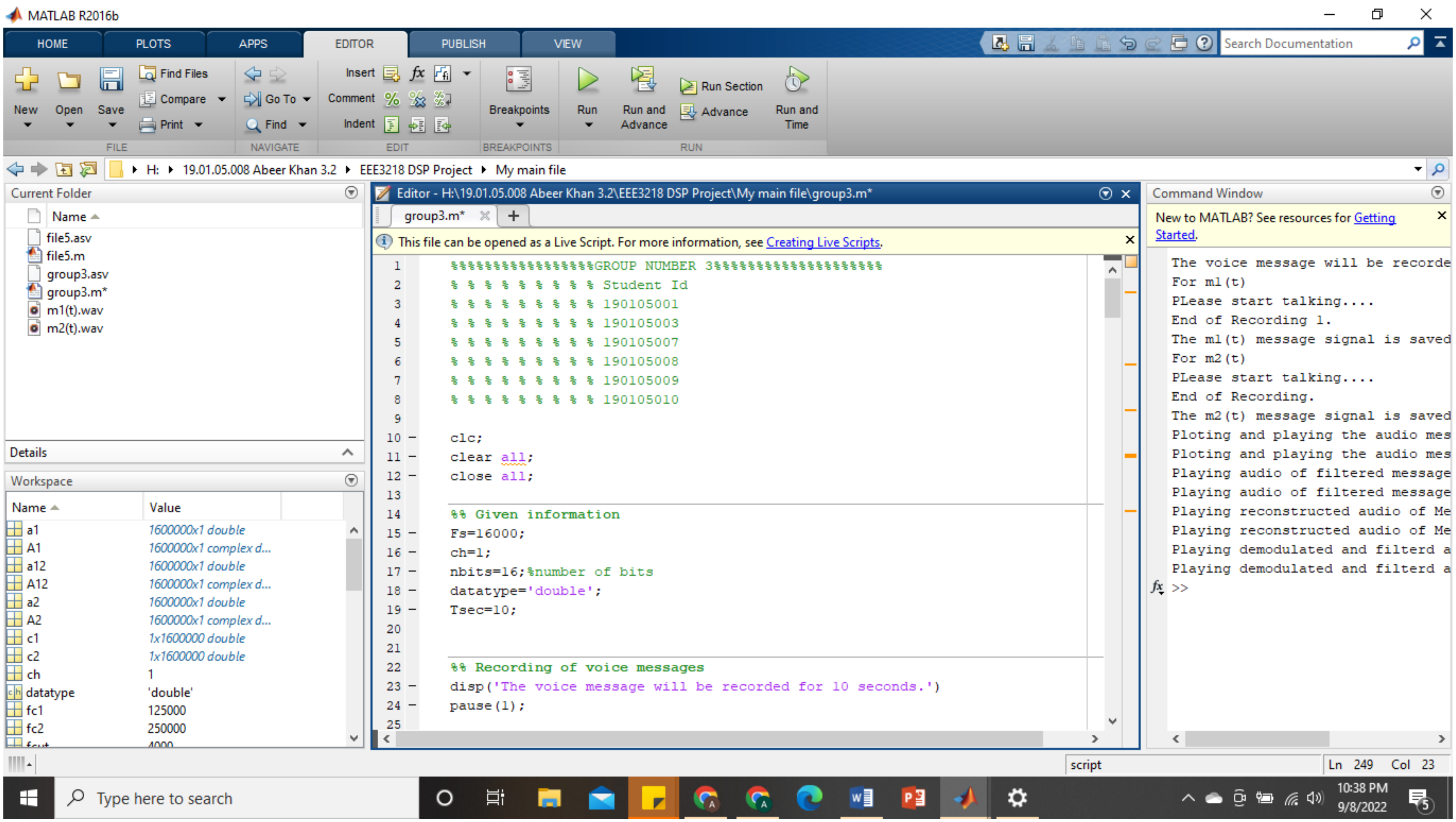
Department: EEE

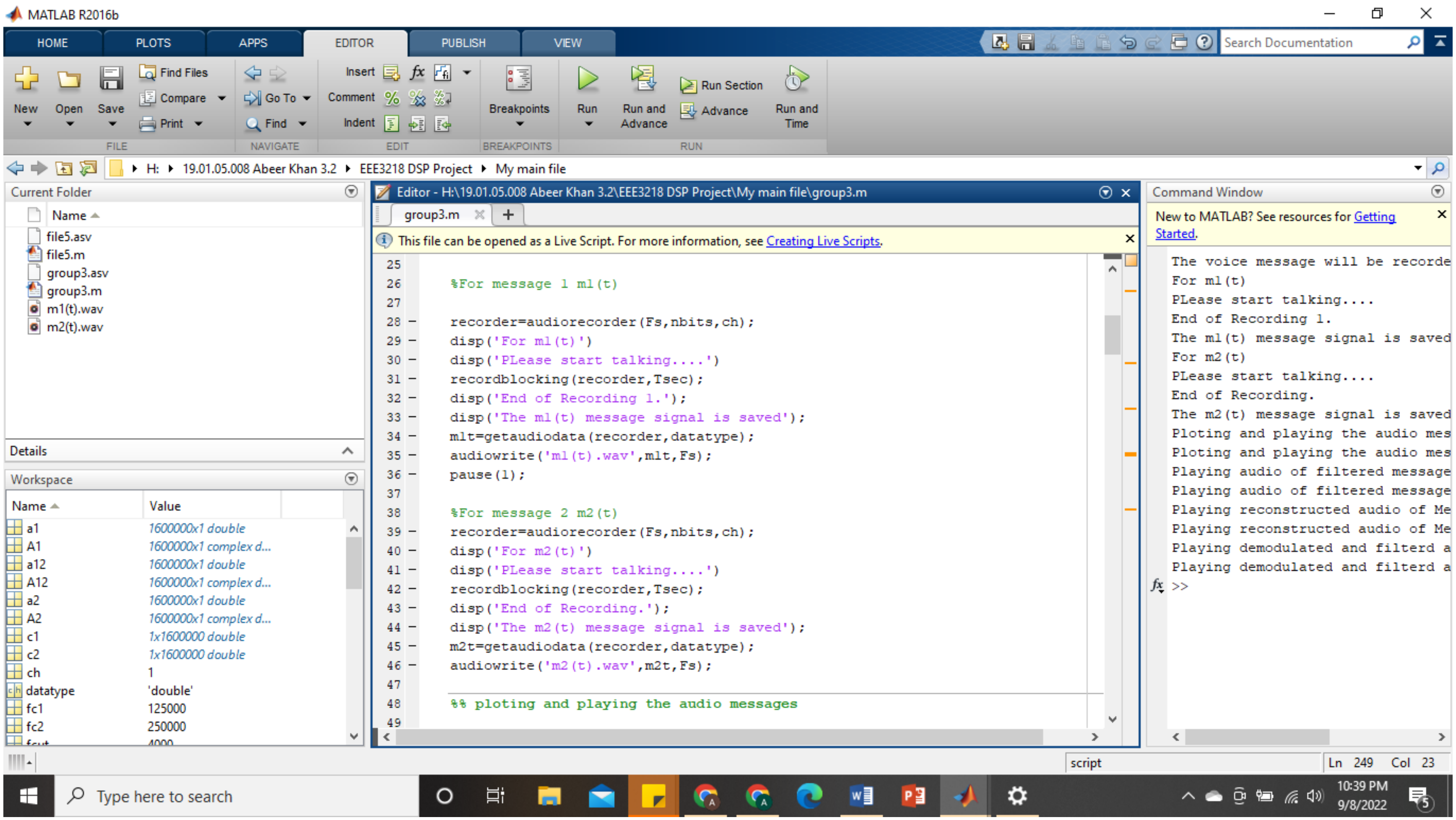
An introduction focusing the real life importance and application of this project

The project is mainly focusing on 2 channel FDM system. The importance and real life application of this is so many that it's impossible to mention every one of them here,but a few important and known ones are : 1)Radio Broadcasting 2)Television Broadcasting 3)Satelite Communication system 4)Digital Subscriber Line or DSL Modems 5)Multimedia data such as audio,video and image transmission. The above mentioned applications are of high importance in our daily lives, such as,we somewhat depend on radio and television broadcasting for entertainment and news.And we depend on DSL for large data transmission,and to extend it,we use multimedias for entertainment purpose,educational purpose.Satteltice communication is connecting the world so wide that the whole world can now be referred as 'one big village'. As we can see,the importance of this reigns from daily entertainment like watching YouTube videos to Military grade information transmission.

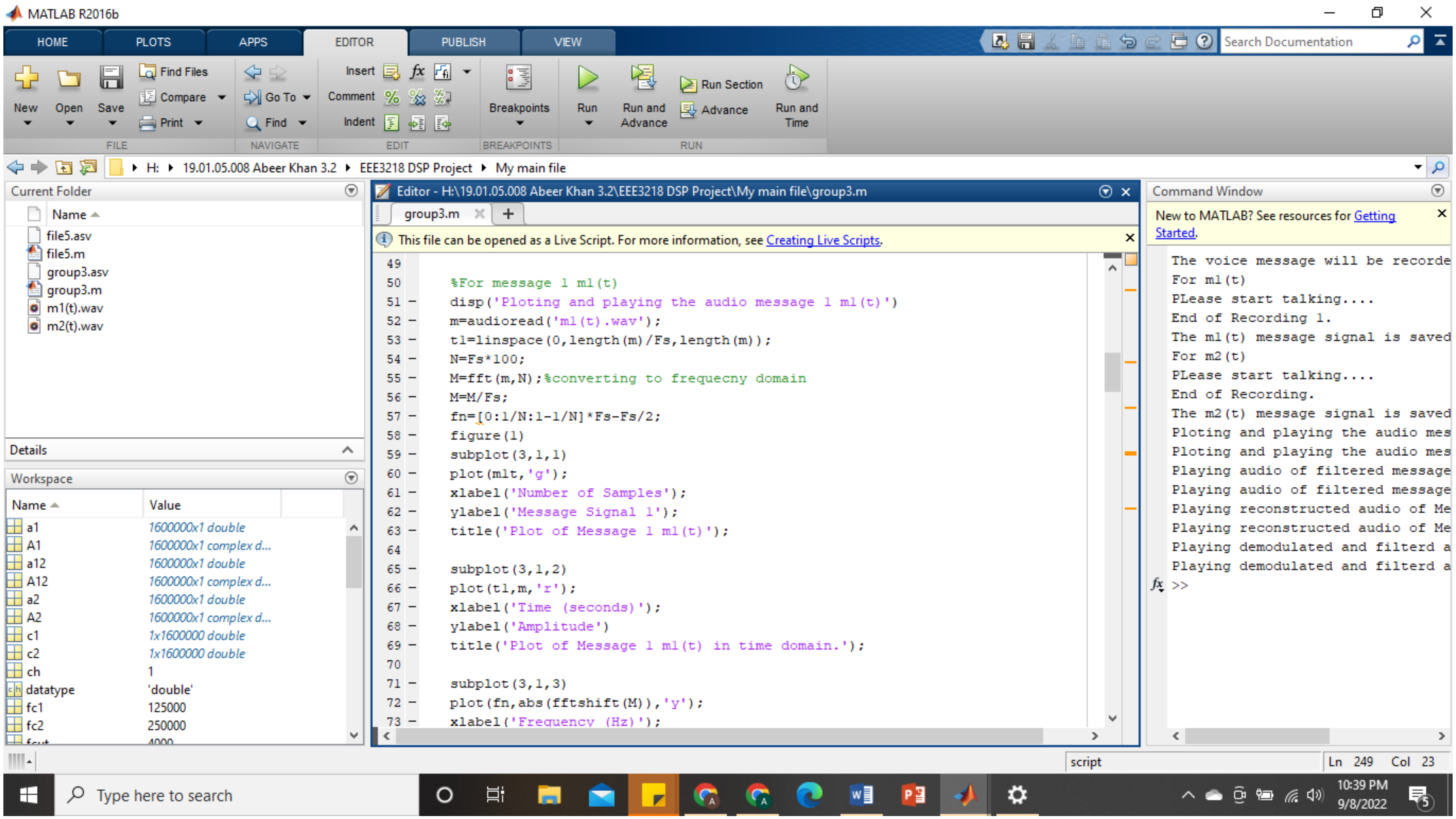
1. **Display at least 5 of the signal spectrums from A to I with proper labeling also display the input signals both in time and frequency domain.**

Screenshots of the MATLAB code and outputs.

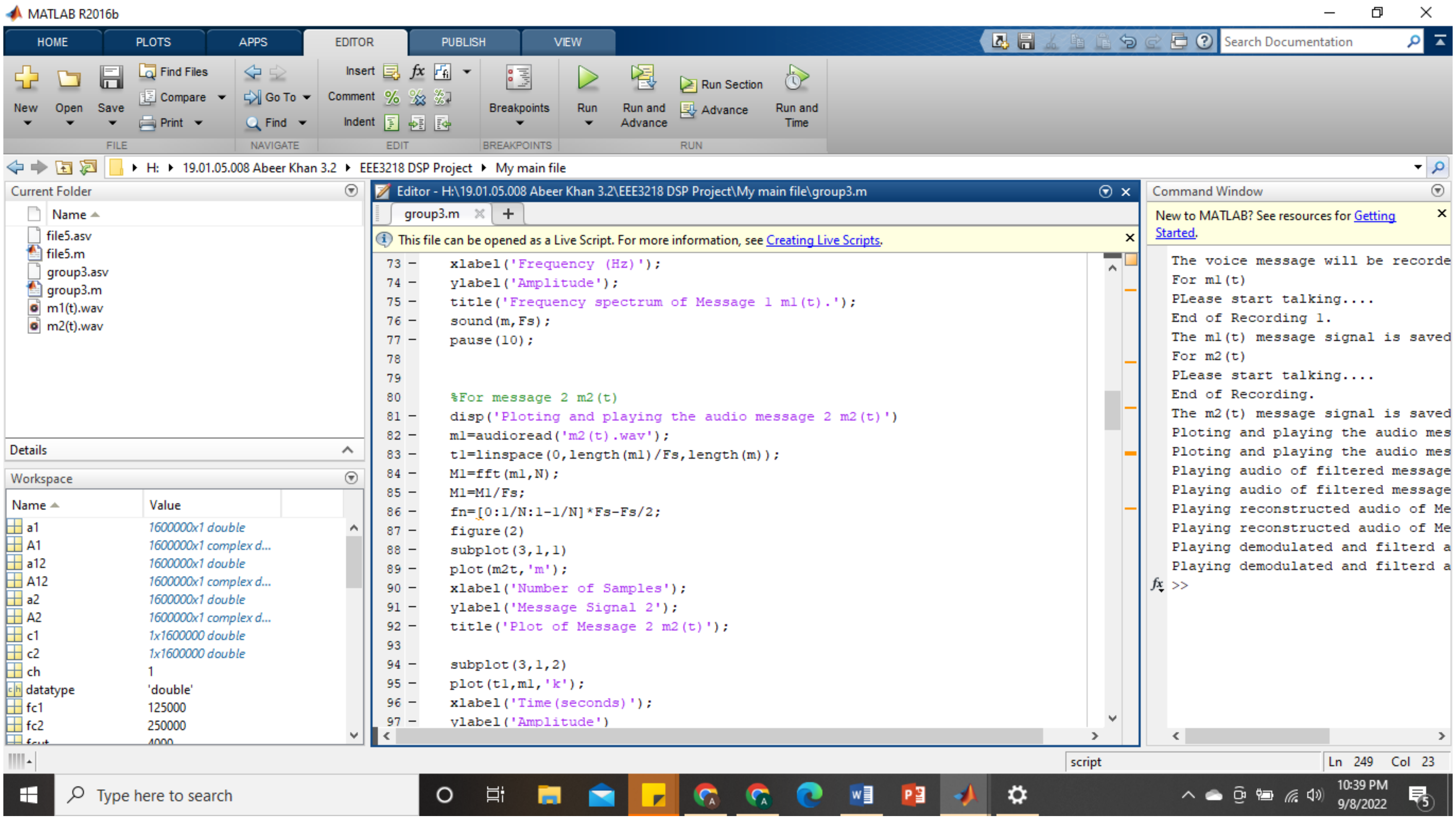


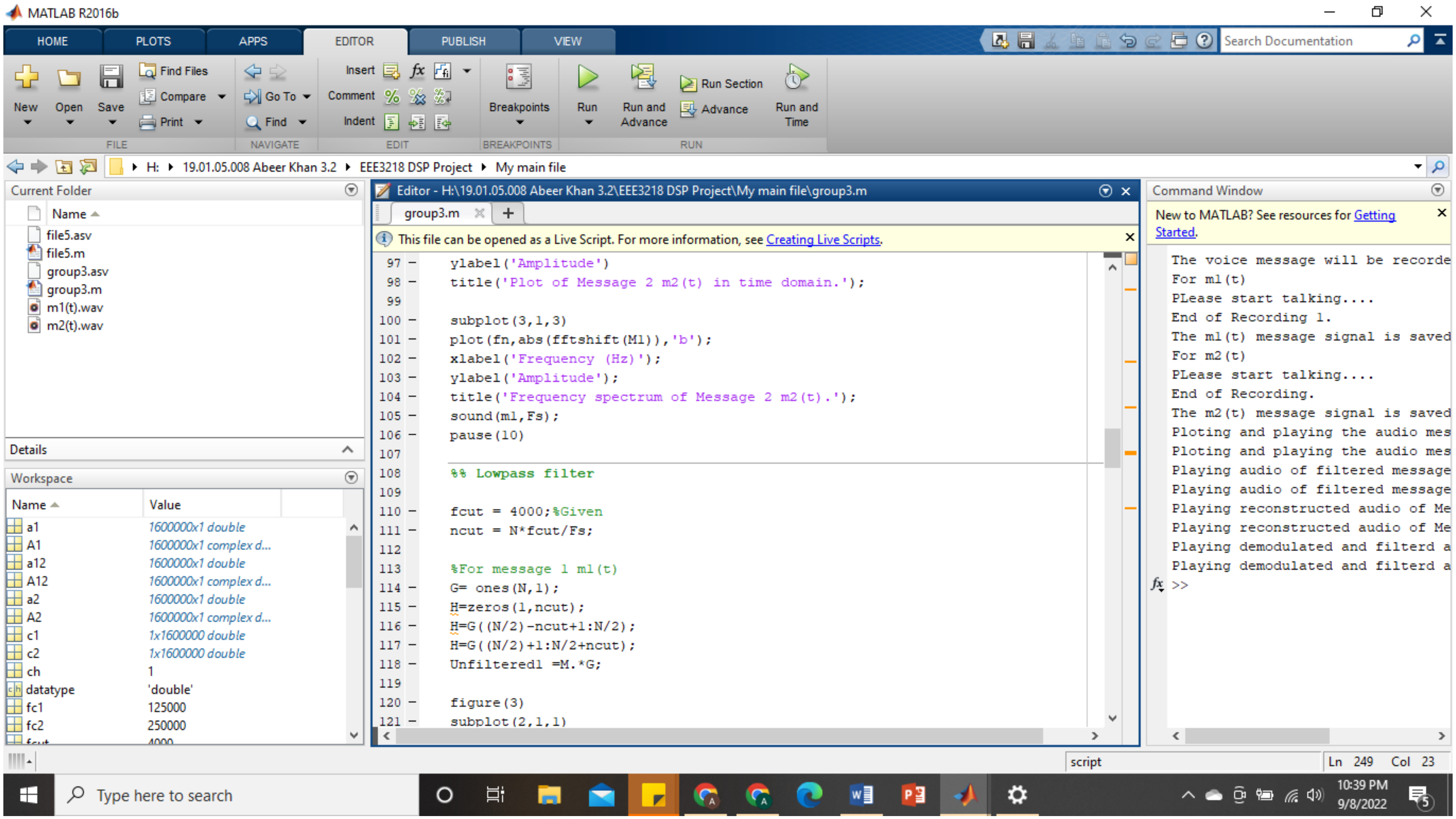


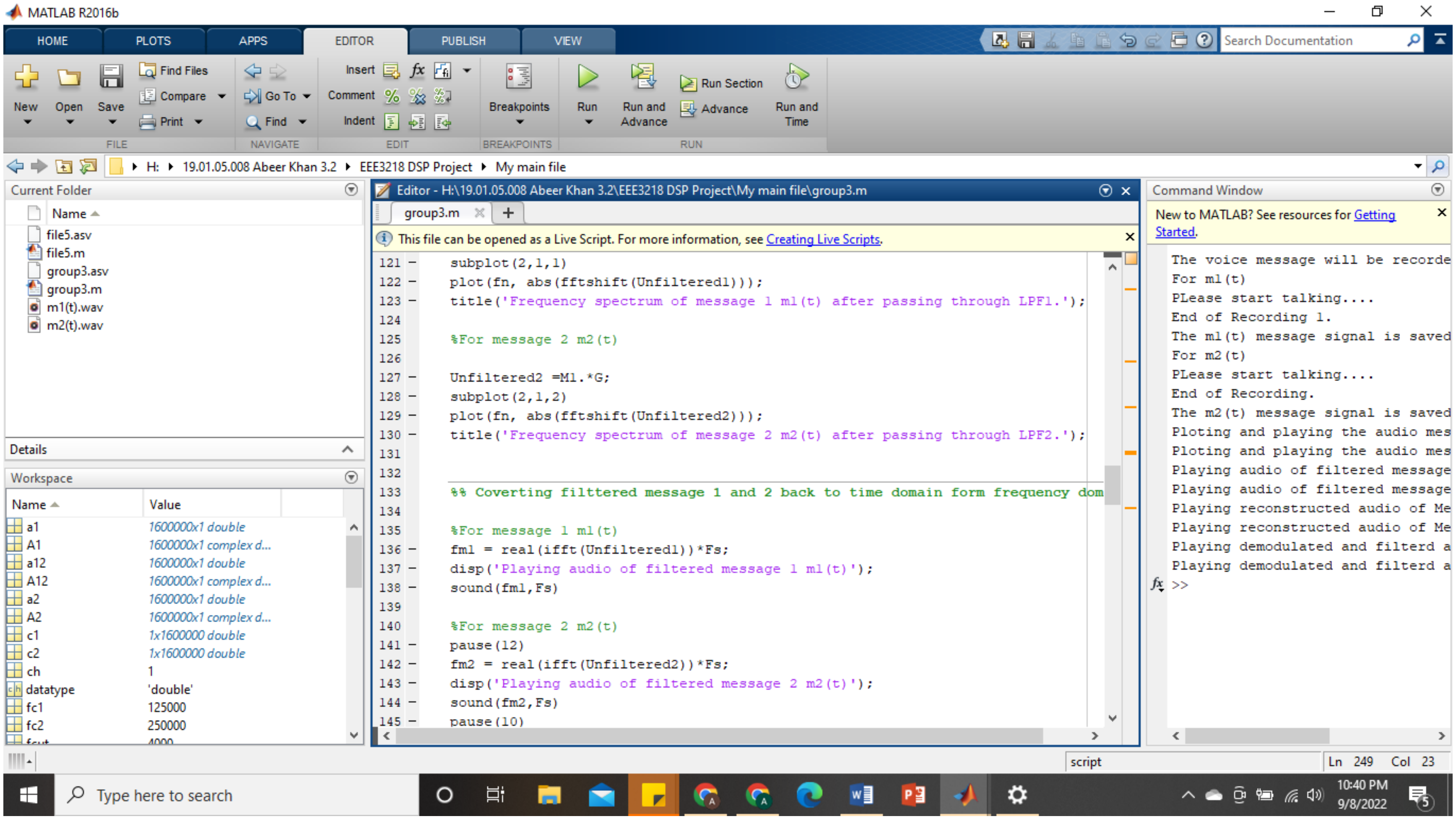
MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
%For message 1 m1(t)
recorder=audiorecorder(Fs,nbits,ch);
disp('For m1(t)')
disp('PLease start talking....')
recordblocking(recorder,Tsec);
disp('End of Recording 1.');
disp('The m1(t) message signal is saved');
m1t=getaudiodata(recorder,datatype);
audiowrite('m1(t).wav',m1t,Fs);
pause(1);
%For message 2 m2(t)
recorder=audiorecorder(Fs,nbits,ch);
disp('For m2(t)')
disp('PLease start talking....')
recordblocking(recorder,Tsec);
disp('End of Recording.');
disp('The m2(t) message signal is saved');
m2t=getaudiodata(recorder,datatype);
audiowrite('m2(t).wav',m2t,Fs);
%% ploting and playing the audio messages
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:39 PM
9/8/2022

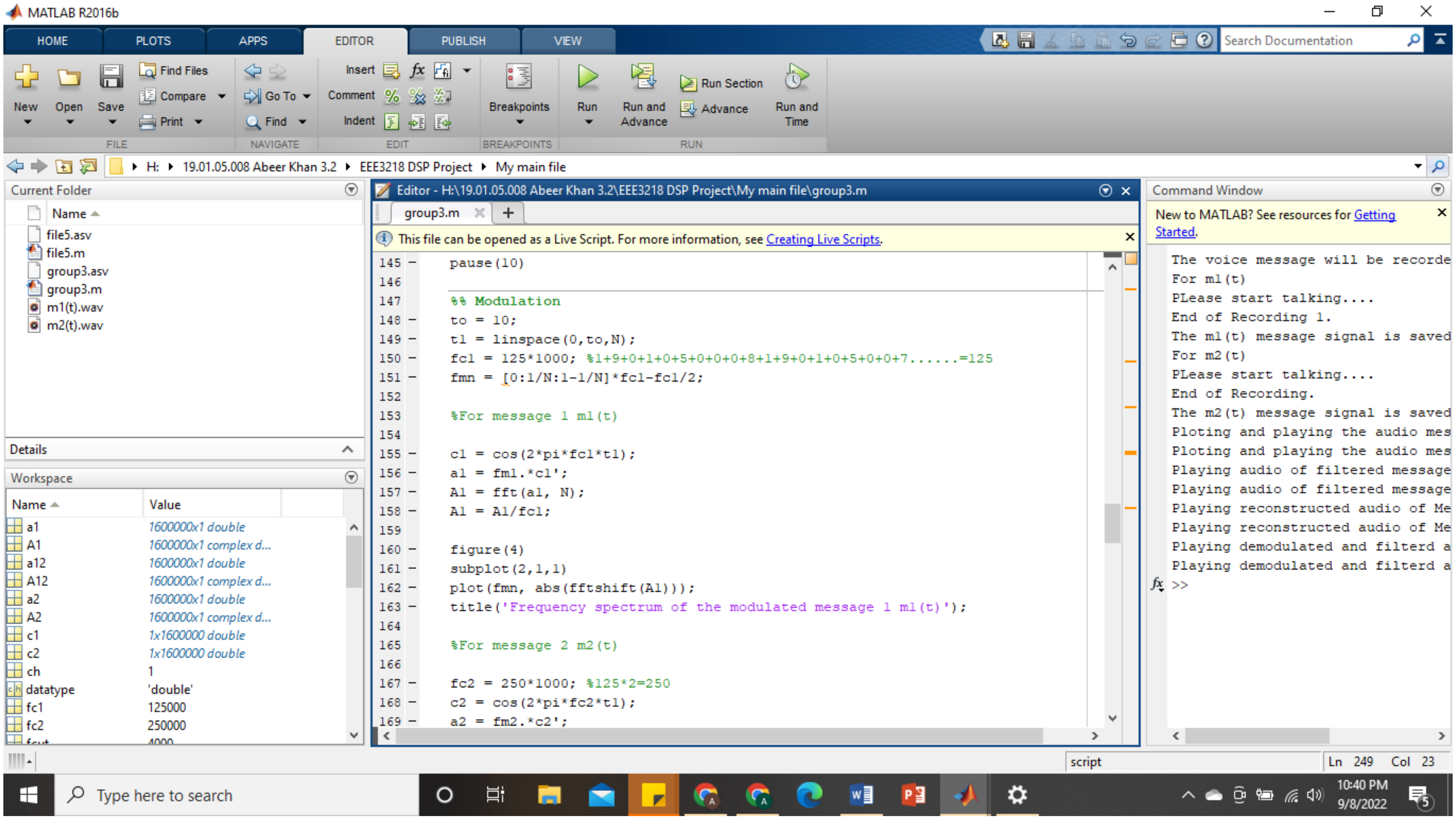


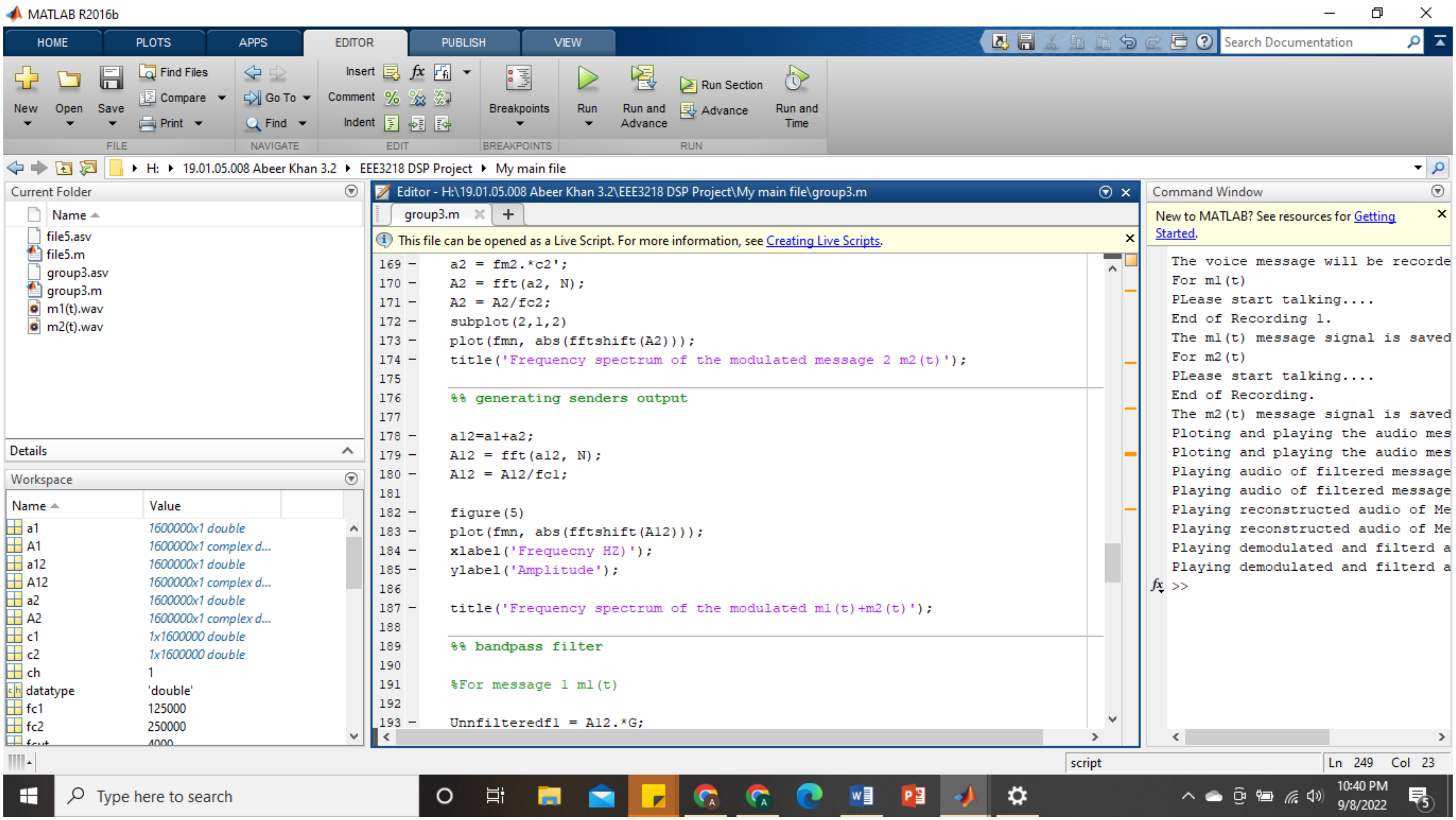
MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Workspace
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
%For message 1 m1(t)
disp('Ploting and playing the audio message 1 m1(t)')
m=audioread('m1(t).wav');
t1=linspace(0,length(m)/Fs,length(m));
N=Fs*100;
M=fft(m,N);%converting to frequecny domain
M=M/Fs;
fn=[0:1/N:1-1/N]*Fs-Fs/2;
figure(1)
subplot(3,1,1)
plot(m1t,'g');
xlabel('Number of Samples');
ylabel('Message Signal 1');
title('Plot of Message 1 m1(t)');
subplot(3,1,2)
plot(t1,m,'r');
xlabel('Time (seconds)');
ylabel('Amplitude')
title('Plot of Message 1 m1(t) in time domain.');
subplot(3,1,3)
plot(fn,abs(fftshift(M)),'y');
xlabel('Frequency (Hz)');
Command Window
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a

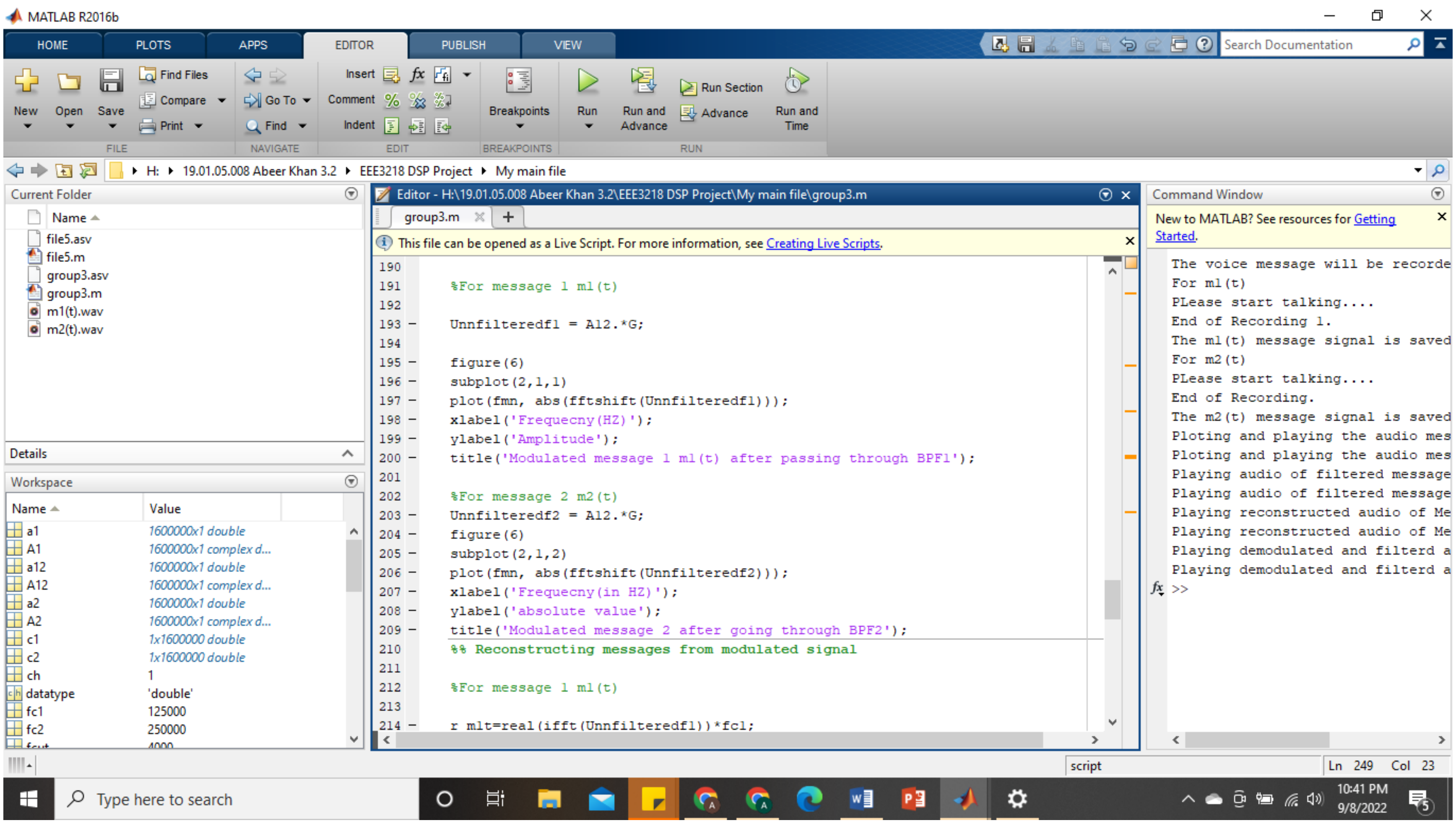

MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Find Files
Compare
Print
New
Open
Save
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
73 xlabel('Frequency (Hz)');
74 ylabel('Amplitude');
75 title('Frequency spectrum of Message 1 m1(t).');
76 sound(m,Fs);
77 pause(10);
80 %For message 2 m2(t)
81 disp('Ploting and playing the audio message 2 m2(t)')
82 m1=audioread('m2(t).wav');
83 t1=linspace(0,length(m1)/Fs,length(m));
84 M1=fft(m1,N);
85 M1=M1/Fs;
86 fn=[0:1/N:1-1/N]*Fs-Fs/2;
87 figure(2)
88 subplot(3,1,1)
89 plot(m2t,'m');
90 xlabel('Number of Samples');
91 ylabel('Message Signal 2');
92 title('Plot of Message 2 m2(t)');
94 subplot(3,1,2)
95 plot(t1,m1,'k');
96 xlabel('Time(seconds)');
97 ylabel('Amplitude')
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:39 PM
9/8/2022

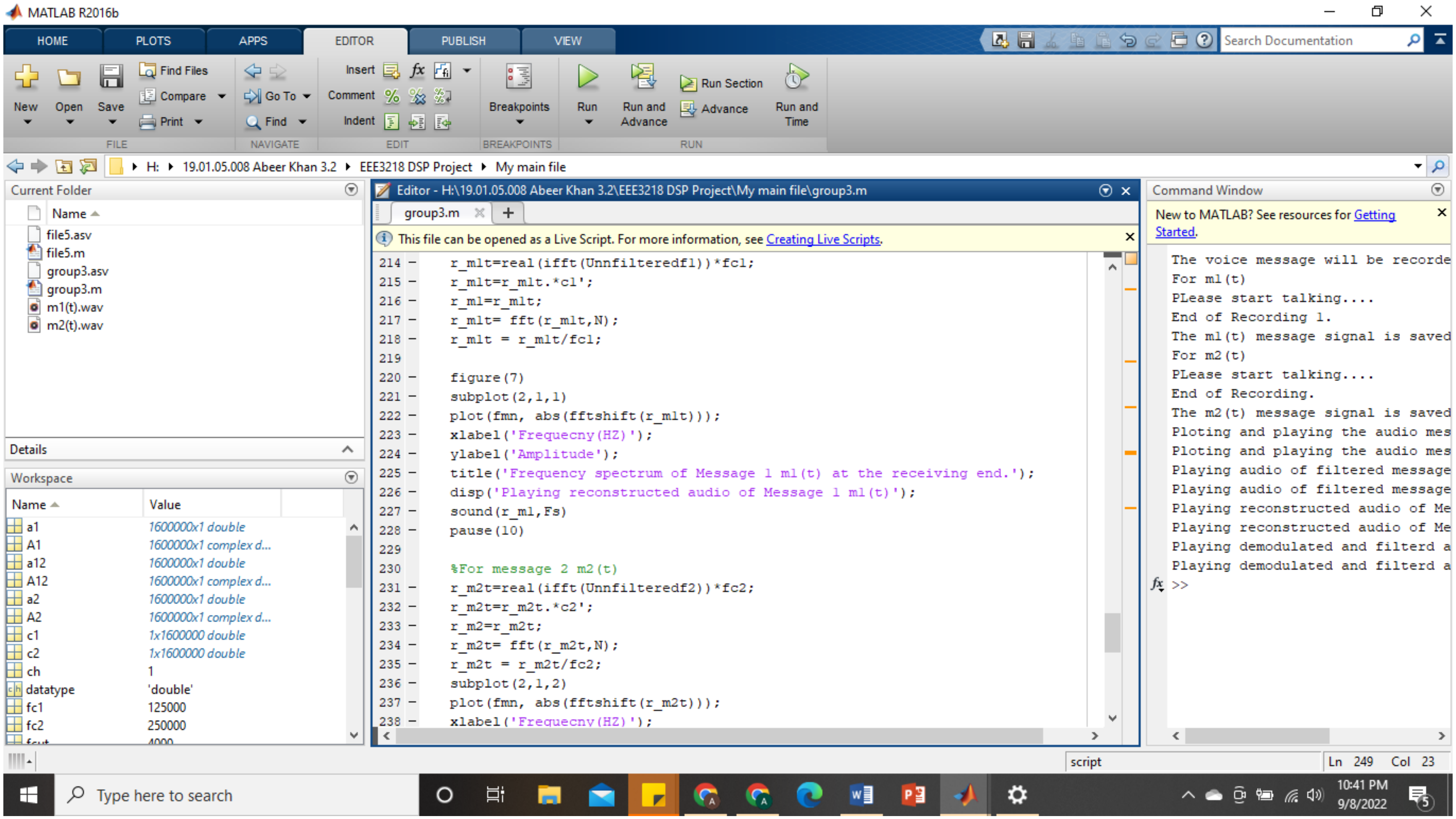

MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
ylabel('Amplitude')
title('Plot of Message 2 m2(t) in time domain.');
subplot(3,1,3)
plot(fn,abs(fftshift(M1)),'b');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Frequency spectrum of Message 2 m2(t).');
sound(m1,Fs);
pause(10)
%% Lowpass filter
fcut = 4000;%Given
ncut = N*fcut/Fs;
%For message 1 m1(t)
G= ones(N,1);
H=zeros(1,ncut);
H=G((N/2)-ncut+1:N/2);
H=G((N/2)+1:N/2+ncut);
Unfiltered1 =M.*G;
figure(3)
subplot(2,1,1)
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:39 PM
9/8/2022

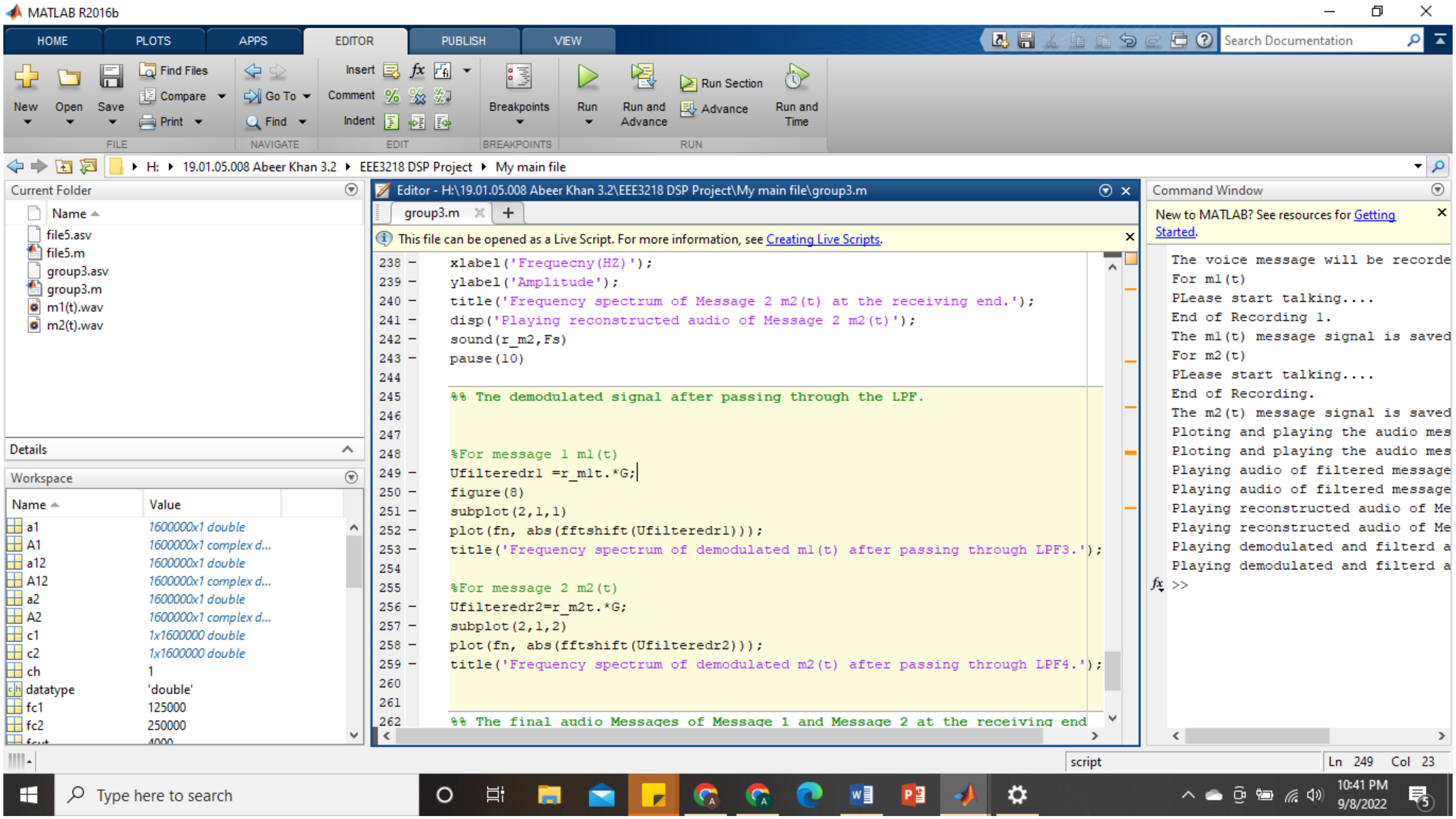

MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
subplot(2,1,1)
plot(fn, abs(fftshift(Unfiltered1)));
title('Frequency spectrum of message 1 m1(t) after passing through LPF1.');
%For message 2 m2(t)
Unfiltered2 =M1.*G;
subplot(2,1,2)
plot(fn, abs(fftshift(Unfiltered2)));
title('Frequency spectrum of message 2 m2(t) after passing through LPF2.');
%% Coverting filttered message 1 and 2 back to time domain form frequency dom
%For message 1 m1(t)
fm1 = real(ifft(Unfiltered1))*Fs;
disp('Playing audio of filtered message 1 m1(t)');
sound(fm1,Fs)
%For message 2 m2(t)
pause(12)
fm2 = real(ifft(Unfiltered2))*Fs;
disp('Playing audio of filtered message 2 m2(t)');
sound(fm2,Fs)
pause(10)
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:40 PM
9/8/2022

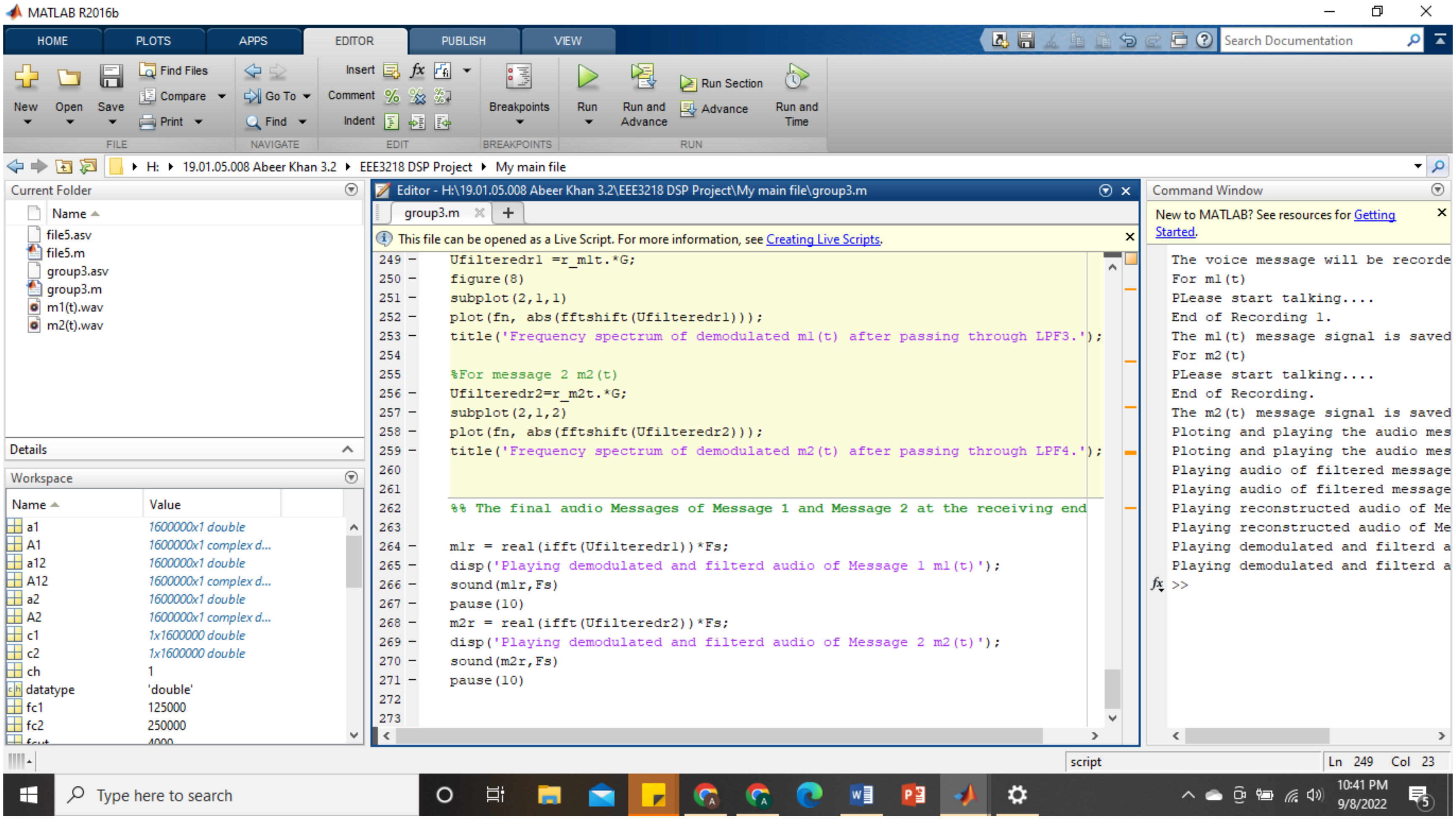

MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
pause(10)
%% Modulation
to = 10;
t1 = linspace(0,to,N);
fc1 = 125*1000; %1+9+0+1+0+5+0+0+0+8+1+9+0+1+0+5+0+0+7......=125
fmn = [0:1/N:1-1/N]*fc1-fc1/2;
%For message 1 m1(t)
c1 = cos(2*pi*fc1*t1);
a1 = fm1.*c1';
A1 = fft(a1, N);
A1 = A1/fc1;
figure(4)
subplot(2,1,1)
plot(fmn, abs(fftshift(A1)));
title('Frequency spectrum of the modulated message 1 m1(t)');
%For message 2 m2(t)
fc2 = 250*1000; %125*2=250
c2 = cos(2*pi*fc2*t1);
a2 = fm2.*c2';
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:40 PM
9/8/2022


MATLAB R2016b
HOME PLOTS APPS EDITOR PUBLISH VIEW
Search Documentation
New Open Save Find Files Compare Print
Go To Find
Insert Comment Indent
Breakpoints
Run Run and Advance Run Section Advance Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN
H: 19.01.05.008 Abeer Khan 3.2 EEE3218 DSP Project My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
a2 = fm2.*c2';
A2 = fft(a2, N);
A2 = A2/fc2;
subplot(2,1,2)
plot(fmn, abs(fftshift(A2)));
title('Frequency spectrum of the modulated message 2 m2(t)');
%% generating senders output
a12=a1+a2;
A12 = fft(a12, N);
A12 = A12/fc1;
figure(5)
plot(fmn, abs(fftshift(A12)));
xlabel('Frequecny HZ)');
ylabel('Amplitude');
title('Frequency spectrum of the modulated m1(t)+m2(t)');
%% bandpass filter
%For message 1 m1(t)
Unnfilteredf1 = A12.*G;
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script Ln 249 Col 23
Type here to search
10:40 PM
9/8/2022

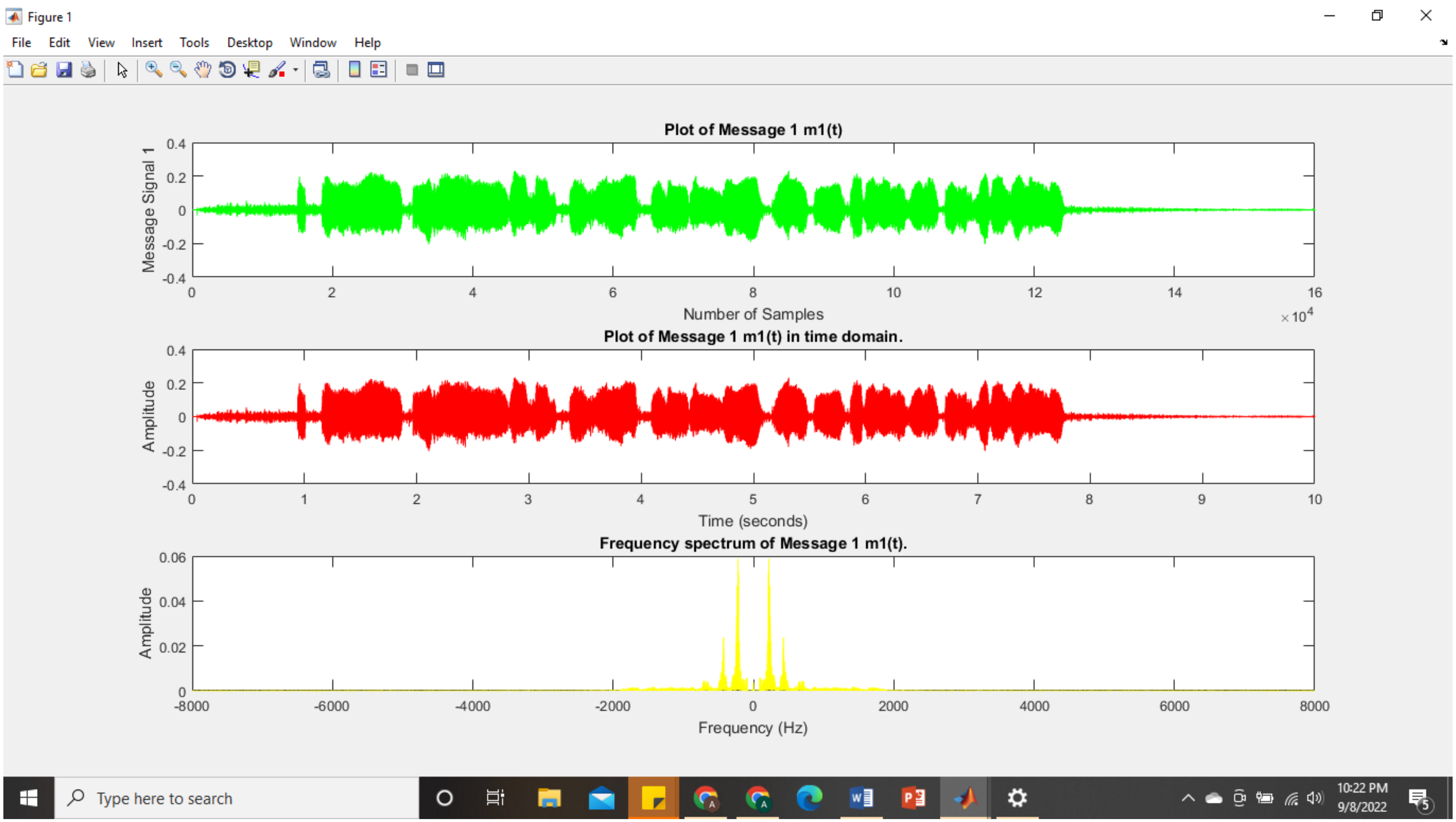

MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
%For message 1 m1(t)
Unnfilteredf1 = A12.*G;
figure(6)
subplot(2,1,1)
plot(fmn, abs(fftshift(Unnfilteredf1)));
xlabel('Frequecny(HZ)');
ylabel('Amplitude');
title('Modulated message 1 m1(t) after passing through BPF1');
%For message 2 m2(t)
Unnfilteredf2 = A12.*G;
figure(6)
subplot(2,1,2)
plot(fmn, abs(fftshift(Unnfilteredf2)));
xlabel('Frequecny(in HZ)');
ylabel('absolute value');
title('Modulated message 2 after going through BPF2');
%% Reconstructing messages from modulated signal
%For message 1 m1(t)
r_m1t=real(ifft(Unnfilteredf1))*fc1;
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:41 PM
9/8/2022


MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
r_m1t=real(ifft(Unnfilteredf1))*fc1;
r_m1t=r_m1t.*c1';
r_m1=r_m1t;
r_m1t= fft(r_m1t,N);
r_m1t = r_m1t/fc1;
figure(7)
subplot(2,1,1)
plot(fmn, abs(fftshift(r_m1t)));
xlabel('Frequecny(HZ)');
ylabel('Amplitude');
title('Frequency spectrum of Message 1 m1(t) at the receiving end.');
disp('Playing reconstructed audio of Message 1 m1(t)');
sound(r_m1,Fs)
pause(10)
%For message 2 m2(t)
r_m2t=real(ifft(Unnfilteredf2))*fc2;
r_m2t=r_m2t.*c2';
r_m2=r_m2t;
r_m2t= fft(r_m2t,N);
r_m2t = r_m2t/fc2;
subplot(2,1,2)
plot(fmn, abs(fftshift(r_m2t)));
xlabel('Frequecny(HZ)');
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:41 PM
9/8/2022


MATLAB R2016b
HOME PLOTS APPS EDITOR PUBLISH VIEW
Search Documentation
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
xlabel('Frequecny(HZ)');
ylabel('Amplitude');
title('Frequency spectrum of Message 2 m2(t) at the receiving end.');
disp('Playing reconstructed audio of Message 2 m2(t)');
sound(r_m2,Fs)
pause(10)
%% The demodulated signal after passing through the LPF.
%For message 1 m1(t)
Ufilteredr1 =r_m1t.*G;
figure(8)
subplot(2,1,1)
plot(fn, abs(fftshift(Ufilteredr1)));
title('Frequency spectrum of demodulated m1(t) after passing through LPF3.');
%For message 2 m2(t)
Ufilteredr2=r_m2t.*G;
subplot(2,1,2)
plot(fn, abs(fftshift(Ufilteredr2)));
title('Frequency spectrum of demodulated m2(t) after passing through LPF4.');
%% The final audio Messages of Message 1 and Message 2 at the receiving end
Command Window
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
Workspace
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
script Ln 249 Col 23
10:41 PM 9/8/2022


MATLAB R2016b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
H: ▸ 19.01.05.008 Abeer Khan 3.2 ▸ EEE3218 DSP Project ▸ My main file
Current Folder
Name
file5.asv
file5.m
group3.asv
group3.m
m1(t).wav
m2(t).wav
Details
Workspace
Name
Value
a1 1600000x1 double
A1 1600000x1 complex d...
a12 1600000x1 double
A12 1600000x1 complex d...
a2 1600000x1 double
A2 1600000x1 complex d...
c1 1x1600000 double
c2 1x1600000 double
ch 1
datatype 'double'
fc1 125000
fc2 250000
Editor - H:\19.01.05.008 Abeer Khan 3.2\EEE3218 DSP Project\My main file\group3.m
group3.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
Ufilteredr1 =r_m1t.*G;
figure(8)
subplot(2,1,1)
plot(fn, abs(fftshift(Ufilteredr1)));
title('Frequency spectrum of demodulated m1(t) after passing through LPF3.');
%For message 2 m2(t)
Ufilteredr2=r_m2t.*G;
subplot(2,1,2)
plot(fn, abs(fftshift(Ufilteredr2)));
title('Frequency spectrum of demodulated m2(t) after passing through LPF4.');
%% The final audio Messages of Message 1 and Message 2 at the receiving end
m1r = real(ifft(Ufilteredr1))*Fs;
disp('Playing demodulated and filterd audio of Message 1 m1(t)');
sound(m1r,Fs)
pause(10)
m2r = real(ifft(Ufilteredr2))*Fs;
disp('Playing demodulated and filterd audio of Message 2 m2(t)');
sound(m2r,Fs)
pause(10)
Command Window
New to MATLAB? See resources for Getting Started.
The voice message will be recorde
For m1(t)
PLease start talking....
End of Recording 1.
The m1(t) message signal is saved
For m2(t)
PLease start talking....
End of Recording.
The m2(t) message signal is saved
Ploting and playing the audio mes
Ploting and playing the audio mes
Playing audio of filtered message
Playing audio of filtered message
Playing reconstructed audio of Me
Playing reconstructed audio of Me
Playing demodulated and filterd a
Playing demodulated and filterd a
>>
script
Ln 249 Col 23
Type here to search
10:41 PM
9/8/2022

Screenshots of Output at different stages



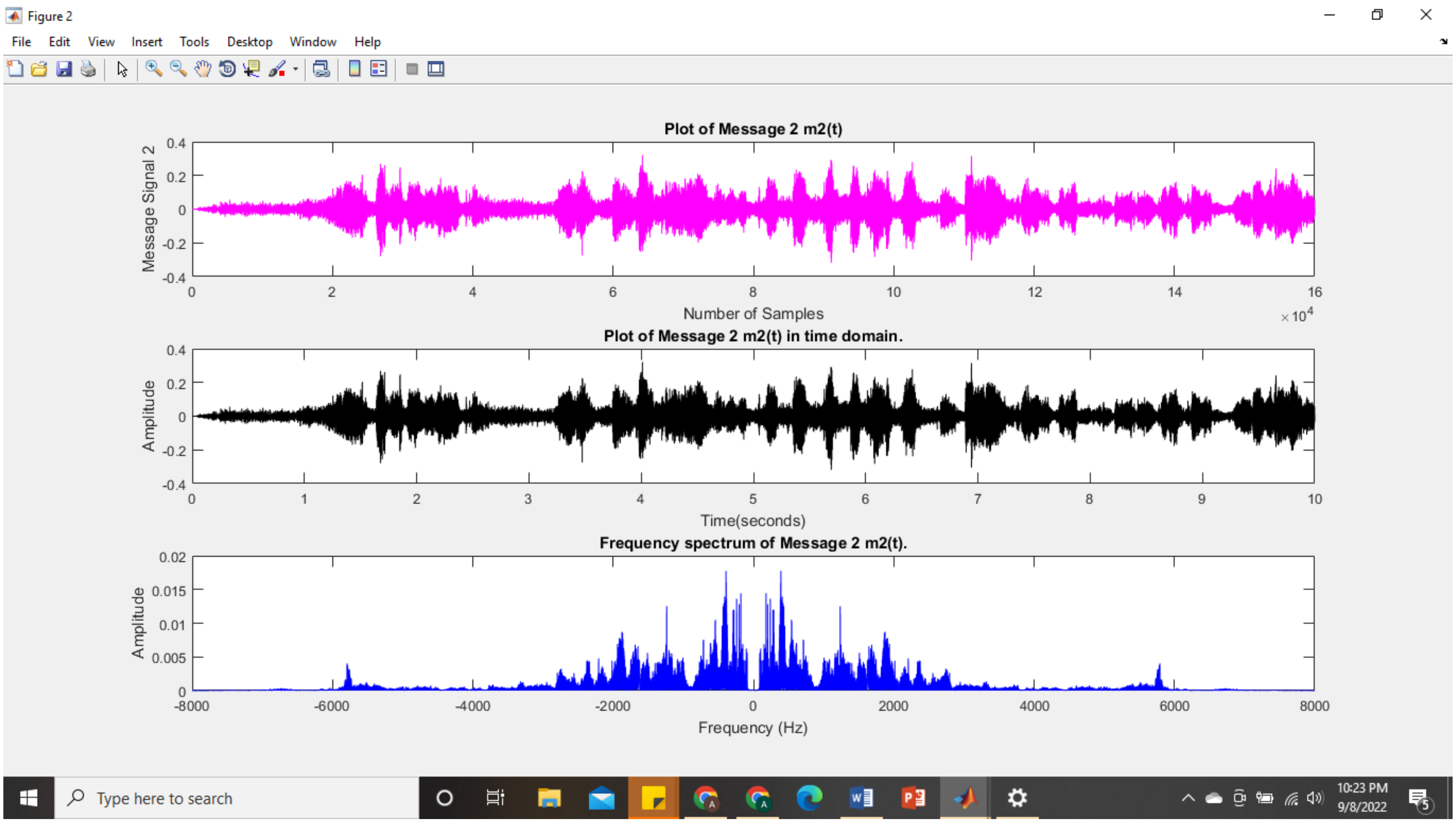


Figure 2
File Edit View Insert Tools Desktop Window Help
Plot of Message 2 m2(t)
Message Signal 2
Number of Samples
× 10^4
Plot of Message 2 m2(t) in time domain.
Amplitude
Time(seconds)
Frequency spectrum of Message 2 m2(t).
Amplitude
Frequency (Hz)
Type here to search
10:23 PM
9/8/2022

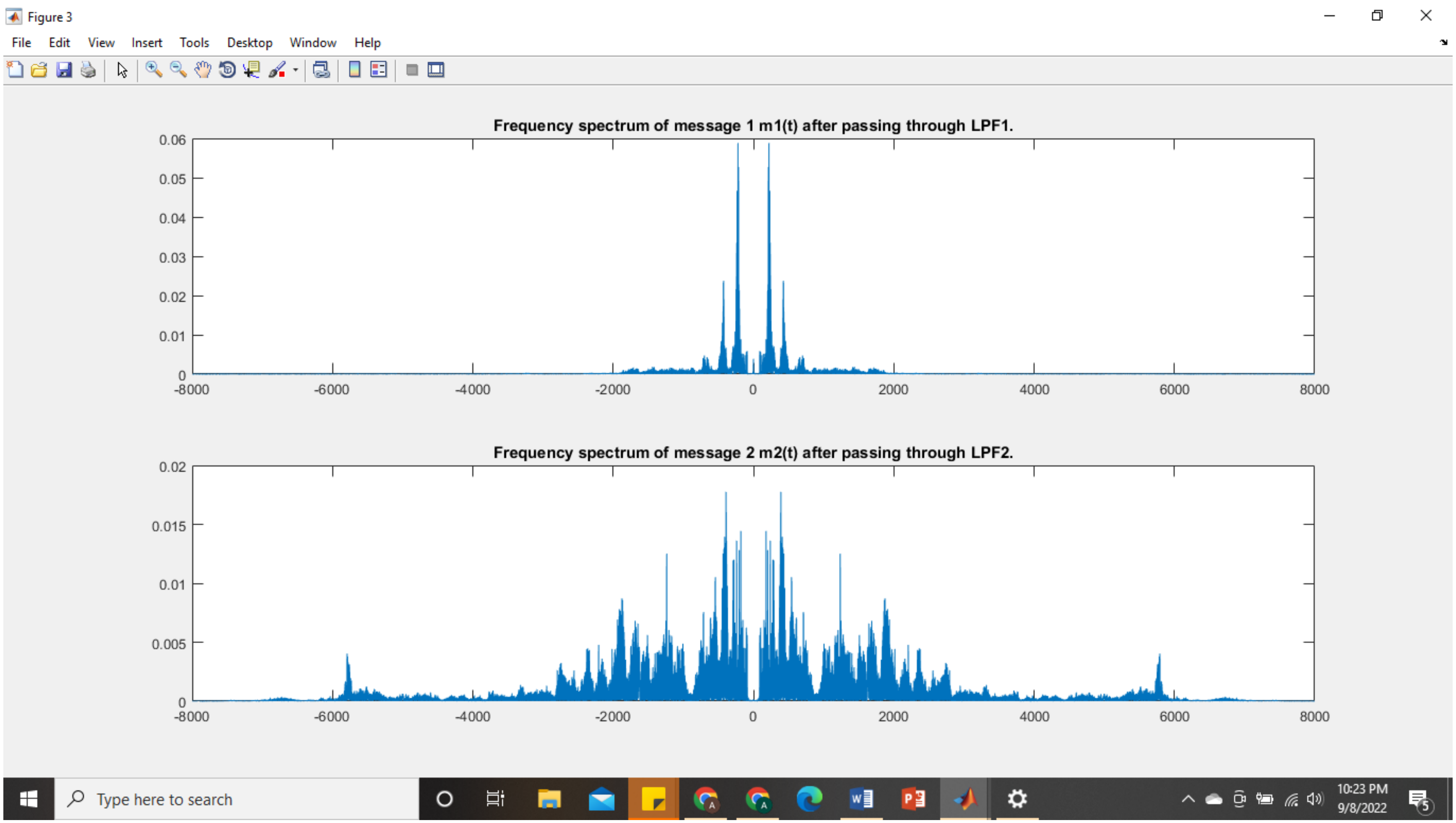


Figure 3
File Edit View Insert Tools Desktop Window Help
Frequency spectrum of message 1 m1(t) after passing through LPF1.
0.06
0.05
0.04
0.03
0.02
0.01
0
-8000 -6000 -4000 -2000 0 2000 4000 6000 8000
Frequency spectrum of message 2 m2(t) after passing through LPF2.
0.02
0.015
0.01
0.005
0
-8000 -6000 -4000 -2000 0 2000 4000 6000 8000
Type here to search
10:23 PM
9/8/2022

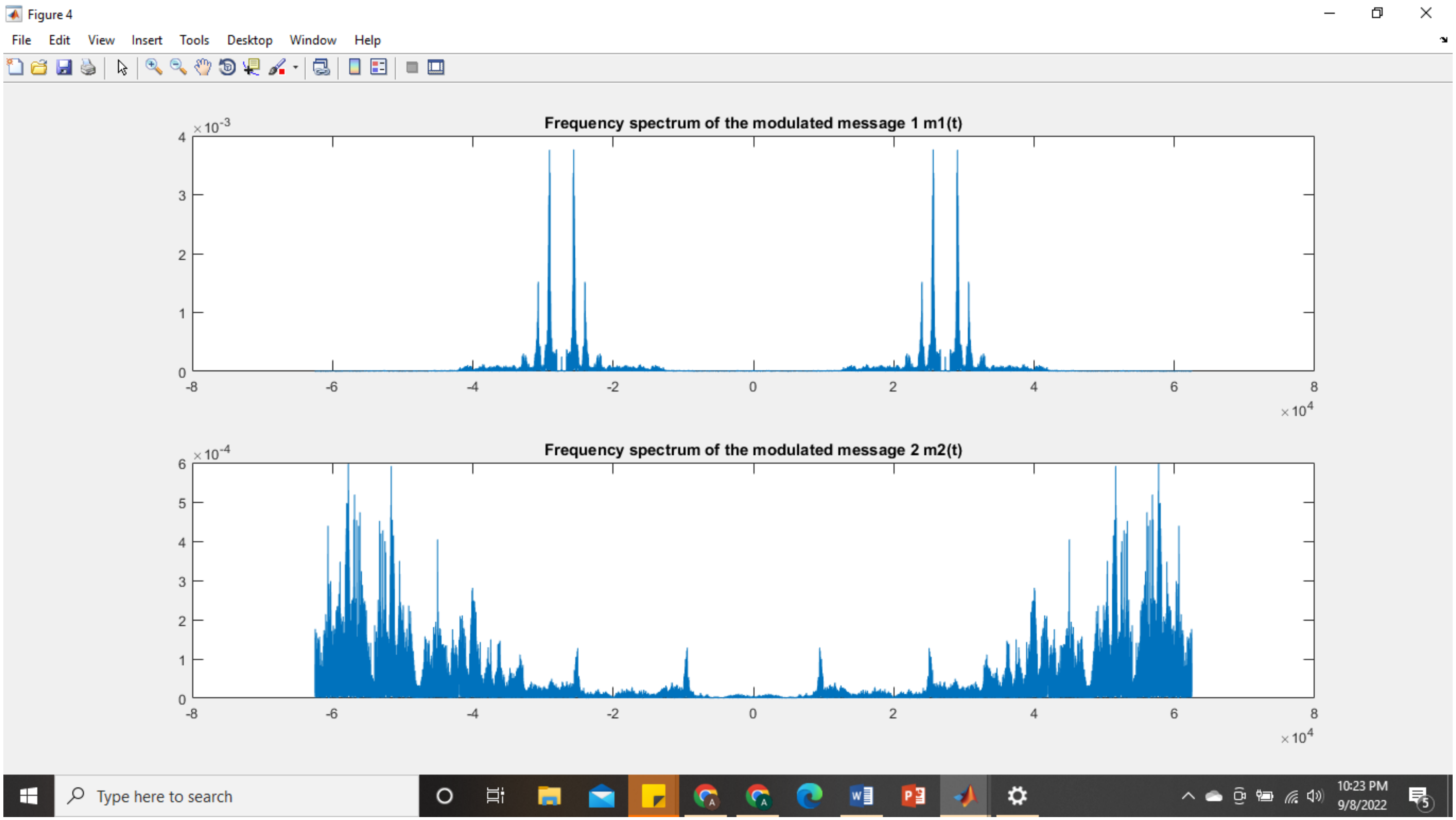


Figure 4
File Edit View Insert Tools Desktop Window Help
Frequency spectrum of the modulated message 1 m1(t)
×10^-3
0 1 2 3 4
-8 -6 -4 -2 0 2 4 6 8
×10^4
Frequency spectrum of the modulated message 2 m2(t)
×10^-4
0 1 2 3 4 5 6
-8 -6 -4 -2 0 2 4 6 8
×10^4
Type here to search
10:23 PM
9/8/2022

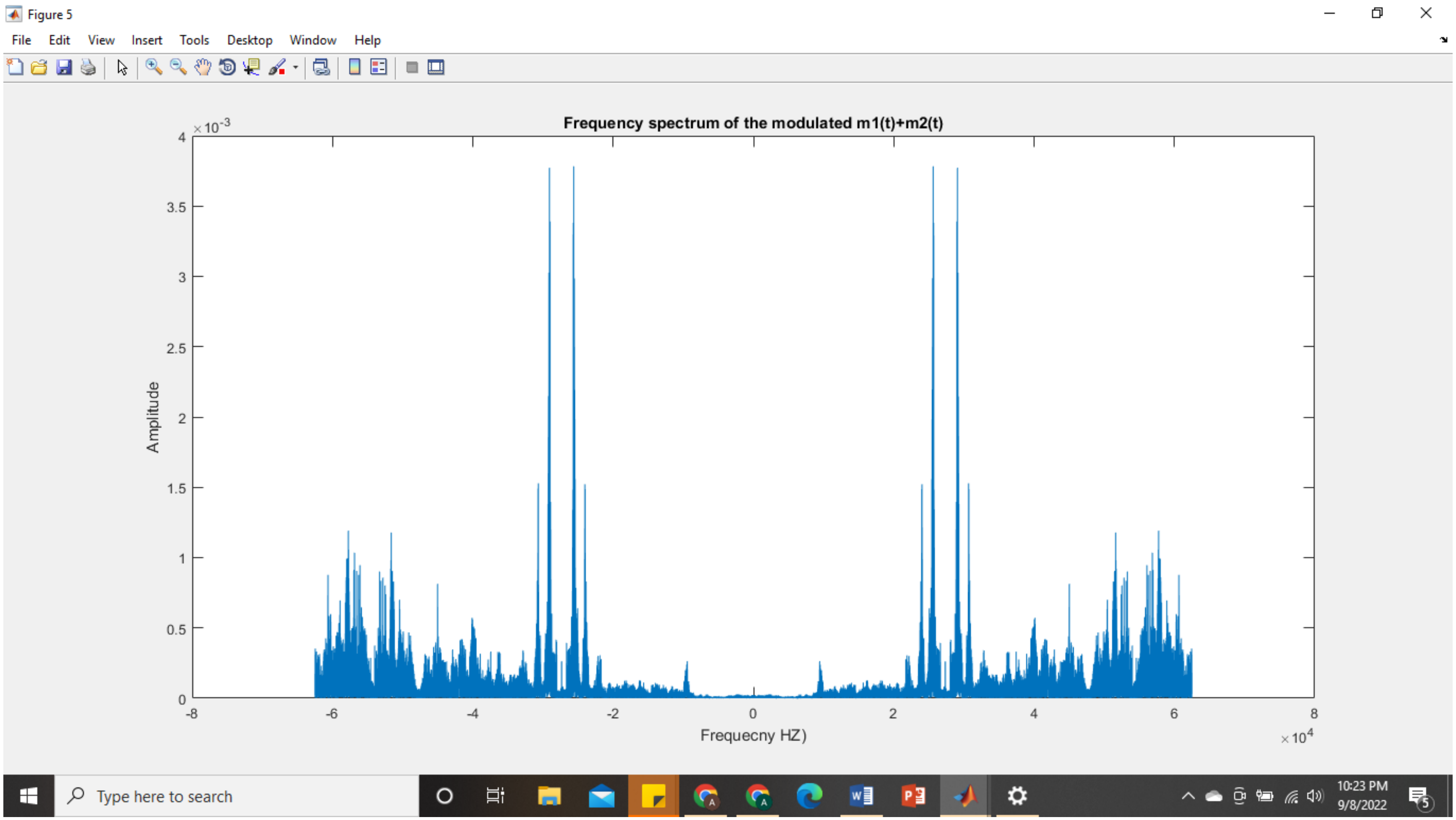


Figure 5
File Edit View Insert Tools Desktop Window Help
Frequency spectrum of the modulated m1(t)+m2(t)
Amplitude
Frequecny HZ)
×10-3
×10^4
Type here to search
10:23 PM
9/8/2022

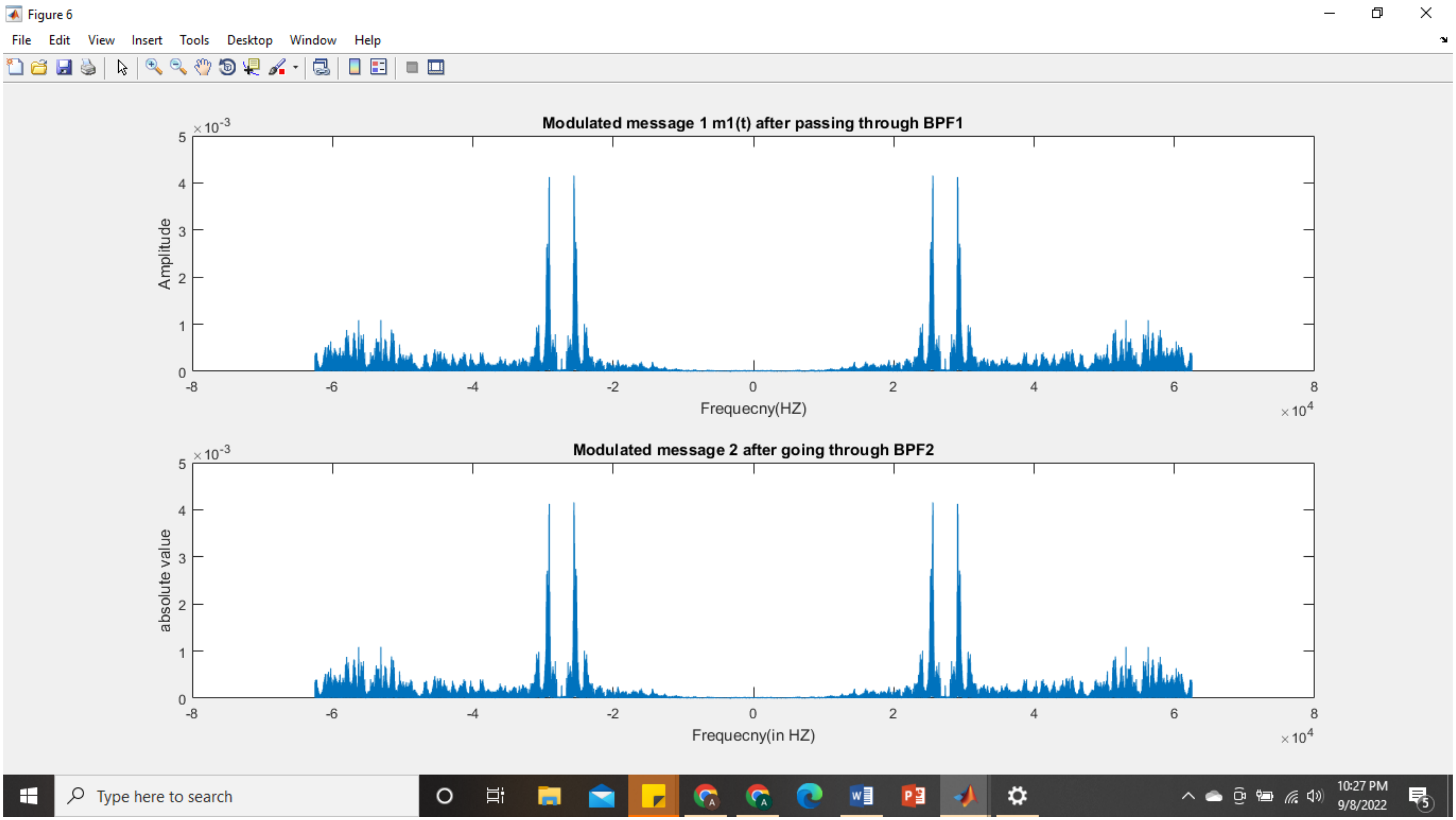


Figure 6
File Edit View Insert Tools Desktop Window Help
Modulated message 1 m1(t) after passing through BPF1
Amplitude
Frequecny(HZ)
Modulated message 2 after going through BPF2
absolute value
Frequecny(in HZ)
Type here to search
10:27 PM
9/8/2022

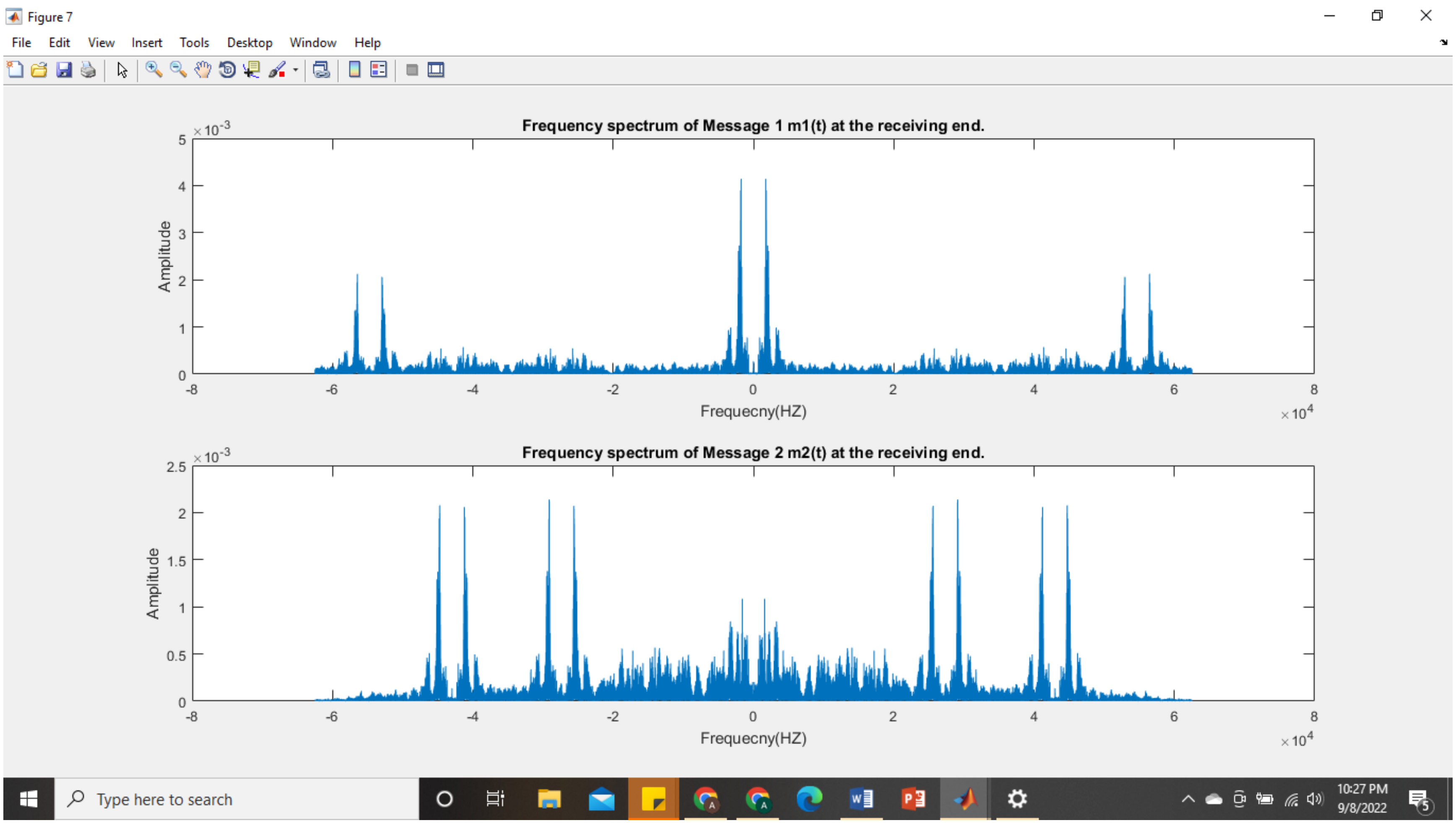


Figure 7
File Edit View Insert Tools Desktop Window Help
Frequency spectrum of Message 1 m1(t) at the receiving end.
×10^-3
Amplitude
Frequecny(HZ)
×10^4
Frequency spectrum of Message 2 m2(t) at the receiving end.
×10^-3
Amplitude
Frequecny(HZ)
×10^4
Type here to search
10:27 PM
9/8/2022

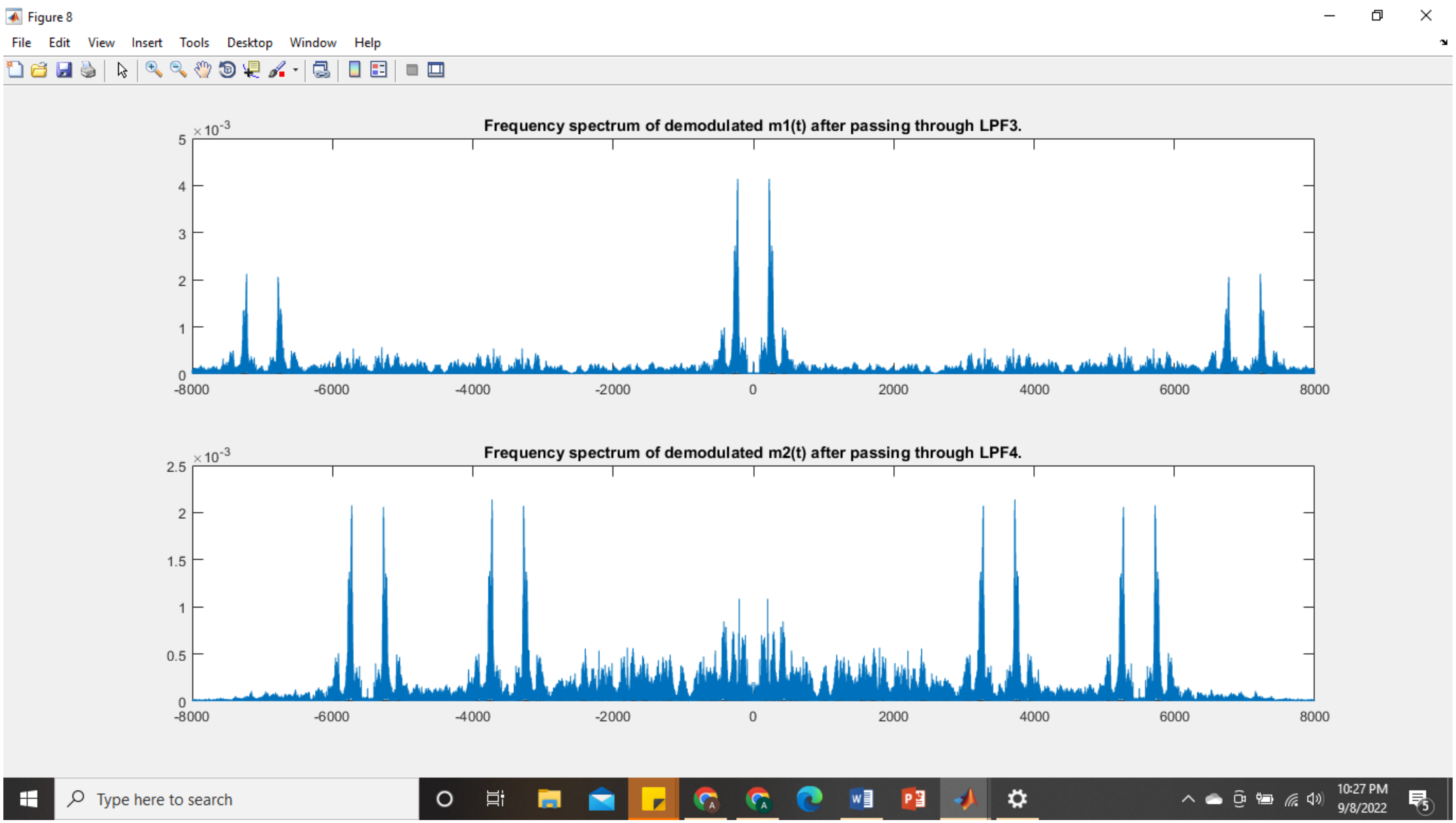


Figure 8
File Edit View Insert Tools Desktop Window Help
Frequency spectrum of demodulated m1(t) after passing through LPF3.
×10^-3
5
4
3
2
1
0
-8000 -6000 -4000 -2000 0 2000 4000 6000 8000
Frequency spectrum of demodulated m2(t) after passing through LPF4.
×10^-3
2.5
2
1.5
1
0.5
0
-8000 -6000 -4000 -2000 0 2000 4000 6000 8000
Type here to search
10:27 PM
9/8/2022

2. How the Ideal filter design techniques are applied?

The feature of an ideal filter is to pass a limited band of frequencies unchanged up to a certain frequency ω_c while totally eliminate frequencies outside the pass band from the signal. As the $H(j\omega)$ is an even function, it is typically only shown for $\omega > 0$ in the filter.
If one calculates the inverse Fourier transform of the product of the filter characteristic H(jw) and the Fourier transform of the unit impulse function $\delta(t)$, one obtains the unit impulse response $h(t)$.
In the digital world, other problems are associated with implementing an ideal filter.There are oscillations in the impulse response $h(t)$ from $-\infty$ to ∞, and its frequency domain equivalent has an infinitely steep slope. For an ideal filter with a square wave input (with zero-mean and an equal duty cycle), the first oscillation is an overshoot with an amplitude that is always 18% of the expected step amplitude.Techniques Applied are:

Signal Processing and Pattern Recognition for Structural Health Monitoring with PWAS Transducers

Denoising Using Digital Filters:
The simplest filter is an ideal filter with zero phase. Four commonly used ideal-filter frequency responses are the (1) lowpass filter, (2) highpass filter, (3) bandpass filter, and (4) bandstop filter.
In actual implementations, the ideal filters need to be approximated with a realizable shape. That is to say, the sharp cutoff edges need to be replaced with *transition bands* in which the designed frequency response would change smoothly from one band to another. Therefore, the cutoff edges are replaced with nonzero-width transition bands located around the ideal cutoff edges. Here, ωp is the passband cutoff frequency and ωs is the stopband cutoff frequency. The resulting transition band has a width of $\Delta\omega=\omega s-\omega p$. In the transition band, the frequency response is desired to change smoothly, i.e., without fluctuations or overshoots. This requirement may be satisfied by a design with constraints on such transition bands., δp is the passband ripple and the maximum allowable error in the passband, whereas δs is the stopband ripple and the maximum allowable error in the stopband.

The order N of the IIR filter determines the number of previous output samples that need to be stored and then fed back to compute the current output samples.
In addition, many practical filter designs are specified in terms of constraints on the magnitude responses and no constraints on the phase response. Then, the objective of filter design is to find a functional filter (either IIR or FIR) whose magnitude frequency response $H(\omega)$ approximates the given specified design constraints. There are four classic

filter functions available in most simulation software: (1) Butterworth, (2) Chebyshev I, (3) Chebyshev II, (4) Elliptic. Each filter has its own characteristics in terms of performance.

Dust and fume control

The ideal filter is permeable to the airstream but not to the dust requiring separation. Separation occurs by impaction of the dust particles upon fabric fibres, resulting in a dust cake forming on the fabric. Although aiding filtration, this cake does increase the resistance to air flow. If allowed to go unchecked it would result in a reduction of the total air being exhausted in the system.

Two methods are used to remove the dust cake, both of which require interruption of the air flow. The difference in dust-cake removal conveniently divides filters into intermittent and continuous rating. In the intermittent type the pressure increases (with time) up to a pre-arranged level. The air flow is then stopped and the fabric is mechanically shaken. In the continuously rated filter the pressure drop rises to a low set point, after which it remains constant across the filter as a whole. The cleaning is done by isolating a part of the filter from the airstream and that section is cleaned.

Intermittent filters are best suited to small applications which will allow the process to be stopped at intervals. The interval used is 4 h (i.e. a morning or afternoon shift). Shaking is done by either hand or electric motor. The application of these filters is limited to the incoming dust burden of the order of 5 g/m^3 and is known as nuisance dust.

Continuously rated filters have whole sections of the filter shut off from the air flow and then those sections are shaken or cleaned. Shaking is carried out in sequence, usually by electric motor. Where the filter is cleaned it is done by a jet of compressed air being blown in reverse to the air flow through the fabric. This system does not require whole sections to be shut down, as the reverse blow is carried out when the filter is on-stream. The time of blow is very small and are measured in parts of a second rather than in minutes, as in the case of shaking filters. The application of these filters is in continuous processes and where the dust burdens are high (in excess of 100 g/m^3).

The shaking and continuous filters are regenerative, but there is a third group usually associated with ventilation work rather than dust and fume. These are throw-away filters which, as the name implies, means that when they become too caked with dust to operate correctly the filters are removed and replaced with new ones. They will only handle low incoming dust burdens, but their efficiencies are the highest of any filter. Typical applications are fresh air input plants, clean-room filtration and nuclear processes.

Theory of Subband Decomposition

A two-band filter bank, serves as an example for the application of the theory just developed. demonstrate the connection between the shape of the desired band split and the decimation lattice and to illustrate how the subbands propagate through the configuration.
Suppose that the desired band split is that shown in Fig. 3.73 where the low-frequency region B_0 is the shaded interior of the diamond-shaped region, and the high-frequency subband B_1 is the complement of B_0 as indicated, such that

Figure 3.74 shows the spectral bands at the various nodes in the subband structure. For illustrative purposes, the input signal spectrum is represented by eight bands, four belonging to B_0, and four to B_1. The ideal filter H′0(ω_) passes bands 1, 2, 3, and 4 to give the spectrum at node (2). Similarly, H′1(ω_) passes bands 5 through 8, yielding the high-pass spectrum at point (6). Next, we select a subsampling lattice compatible with the subband split. In this respect, we want $D^T \omega$ to partition B into the diamond-shaped region. The decimation matrix and coset vectors that achieve this are

This spectrum is shown in Fig. 3.74(3). Note that the subbands 1 through 4 occupy the full band, and because of the ideal filters, there is no aliasing at this point. Also note the rotation and stretching induced by *D.* The up-sampling compresses the spectrum V′0(ω_) and creates the images outside the diamond as shown in Fig. 3.74(4). This is also evident from

The images are due to the term X′0(ω1-π,ω2-π). The ideal synthesis filter $G_0(z) = H_0(z)$ removes these images, leaving us with the subbands shown at Fig. 3.74(5). In a similar way, we can trace the signals through the lower branch of the two-band structure. These spectra are also illustrated in Fig. 3.74, (6) through (9). Finally adding the signals at points (5) and (9) gives xˆ(n_)=x(n_), or perfect reconstruction. A detailed discussion of admissible passbands and their relationship to the subsampling lattice is provided by

Design of Digital Filters

An ideal filter is considered to have a specified, nonzero magnitude for one or more bands of frequencies and is considered to have zero magnitude for one or more bands of frequencies. On the other hand, practical implementation constraints require that a filter be causal. Inputs can be delayed for the implementation of a discrete time system that only uses input samples (FIR filter, for example). The delay itself leads to a phase change which is inconsistent with the frequency response specification for an ideal filter. On the other hand, a discrete time system that uses previous values of the output for inputs cannot use output values that have not been computed (IIR filter, for example). This restriction also leads to a different type of phase change and it is also inconsistent with the frequency response specification for an ideal filter.

If $H(\omega)$ is zero over any finite band of frequencies, then this integral is not finite. Thus, the Paley–Wiener Theorem leads to the conclusion that the magnitude function for a given causal filter $|H(\omega)|$ can be zero at some frequencies but it cannot be zero over any finite band of frequencies. It follows that practical filters, which are required to be causal, cannot have ideal frequency specifications.

The design of IIR filters

Before we move on to the design of digital filters, it is probably worth having a *very* brief recap on filters.
An *ideal* filter will have a constant gain of at least unity in the passband and a constant gain of zero in the stopband. Also, the gain should increase from the zero of the stopband to the higher gain of the passband at a single frequency, i.e. it should have a 'brick wall' profile.
It is impossible to design a practical filter, either analogue or digital, that will have these profiles.

As the gain of a real filter does not drop vertically between the passband and stopband, we need some way of defining the 'cut-off' frequencies of filters, i.e. the effective end of the passband.
Some keywords to look for are: lowpass, highpass, bandstop, bandpass, cut-off frequency, roll-off, first, second (etc.) order, passive and active filters, Bode plots and dB. Howatson (1996) is just one of an abundance of circuit theory and analysis texts which will be relevant.
Much work has been carried out into the design of analogue filters and, as a result, standard design equations for analogue filters with very high specifications are available. Howeverthe characteristics of all analogue systems alter due to temperature changes and ageing. It is also impossible for two analogue systems to perform identically. Digital filters do not have these defects. They are also much more versatile than analogue filters in that they are programmable.

Image Enhancement

The Butterworth high pass filter (BHPF) is an approximation to the ideal filter.
The parameter *n* is a user-defined positive integer called the order of the filter. As the value of *n* increases, the BHPF approaches the ideal filter. Image enhancement by homomorphic filtering is usually carried out using the BHPF. In practice, a 2D FFT is used to carry out the filtering operation and the user may repeat this enhancing process for different values of *K* and *n*.

Linear and Nonlinear Control of Switching Power Converters

Assume now the fourth-order Chebyshev low-pass filter, as an ideal filter removing the high-frequency content of the *vPWM* voltage. Then, the v_{PWM} voltage can be considered as the amplifier output. However, the discontinuous voltage $v_{PWM} = \gamma V_{dd}$ is not a state variable and cannot follow the almost continuous reference vPWMr. The new error variable evPWM=vPWMr−kvγVdd is always far from the zero value.
Practical implementation of this control strategy can be done using an integrator with gain κ ($\kappa \approx 1800$) and a comparator with hysteresis ε ($\varepsilon \approx 6$ mV) (Fig. 35.43A). Such an arrangement is called a first-order sigma-delta (ΣΔ) modulator.
The square input, shows much less overshoot and oscillations . However, the v_{PWM}, msubvomsub, and v_o/10 waveforms, for a 20 kHz sine input presented in show increased output voltage loss, compared to the first-order sigma-delta modulator, since the second-order modulator was designed to eliminate the v_o output voltage ringing (therefore reducing the amplifier bandwidth). The obtained performances with these and other sigma-delta structures are inferior to the sliding-mode performances. Sliding mode brings definite advantages as the system order is reduced, flatter passbands are obtained, power supply rejection ratio is increased, and the nonlinear effects, together with the frequency-dependent phase delays, are canceled out.

Digital Filter Design Using an Adaptive Modelling Approach

The design process can be modelled as a system that consists of two parts: An ideal filter that meets the design specifications exactly, and an adaptive filter that gives an approximation to the specifications. The difference between the ideal filter output and the adaptive filter output is an error signal can be used to adjust the filter weights. The process is illustrated in Figure 1. Let the input signal *x(n)* be a sum of K-sinusoids, where each sinusoid is sampled at a frequency, f_s In addition, each sinusoid is of unit amplitude.
The output of the ideal filter is also a sum of sinusoids that exhibits a phase difference from the input. For the ideal filter, some of the output sinusoids will be attenuated and others will pass through the filter as per the design specifications.

Techniques for Noise Suppression for ECG Signal Processing

Linear and Time-Invariant Filtering: FIR and IIR Filters

A number of techniques may be considered for solving the issues related to the poor performance and efficiency of FIR-filters for suppressing baseline wander, such as forward–backward IIR filtering and insertion of zeros into an FIR filter. The use of forward–backward filtering remedies the disadvantage of nonlinear phase response of IIR filters, since the overall result is filtering with a zero-phase transfer function. Due to the freedom of positioning its poles, an IIR filter achieves magnitude frequency-response requirements more easily with a much lower filter order. a forward–backward filtering involves three steps. The first stage refers to the processing of the input signal, named x[n], with an IIR filter impulse response h[n]. Then the corresponding filter output is time reversed, that is, we create a reflected version over time axis. The third stage refers to another processing over the time-reflected filter output, also with h[n]. Finally, the second output (doubly filtered signal) is again time reversed to produce the output signal y[n].

The downsides of the approach related to forward–backward filtering are that the initial transients may be larger, making it inappropriate for many FIR filters that depend on phase response for proper operation. Also, this approach is nonapplicable for strictly real-time ECG signal processing, but indeed is a scheme for off-line processing, since the requirement of causality is not attended when a time-reversed signal is processed. However, forward–backward IIR filtering can be implemented with a relatively short (time) delay as a result of processing consecutive and overlapping signal segments, which may be considered by real-time applications.

Another technique for improving filtering efficiency related to attenuation of baseline wandering is the insertion of zeros (or zero-padding) into a finite impulse response h0[n], which should be originally designed for a much lower sampling rate .

3. How the filters are built and used to get the outputs ? [Hint : Which modern software tool used and why,Which functions used].

3) Some of the softwares and tools used to build filters :
i)FilterLab(https://www.microchip.com/developmenttools/ProductDetails/filterlabdesignsoftware)

ii)Filter Wizard(http://www.analog.com/designtools/en/filterwizard)

iii)Filter Design and Analysis(http://sim.okawa-denshi.jp/en/Fkeisan.htm)

iv)RF Filter Design(http://www.iowahills.com/9RFFiltersPage.html)

v) Matlab and Simulink (https://matlab.mathworks.com/)

vi)Filter Design Tool(https://webench.ti.com/filter-design-tool/)

4. How the investigation is carried out to control the band gap in the FDM spectrum ? [some reference from communication book or lecture should be added as reference in support of the literature review]

A narrow unused band gap that separates two ranges of wider frequency is called a guard band. Guard bands in frequency division multiplexing (FDM) isolate the inter-channel interferences which increases quality for both transmissions. During designing, waste of spectrum occurs in large guard bands. On the other hand, if the guard band is narrow, interference becomes a problem particularly on the edges of the carrier spectrum. At first, we determine a quantity of frequency spectrum available for usage in the guard band; and then expand or reduce bandwidth of the two carriers to increase or decrease the separation between the two carriers. Generally, it is adjusted by beginning an active modification request of the guard band between the two carriers. This allows flexible spectrum use by adjusting the acceptable guard band overlap. The data is distributed in the carrier spectrum band as a function of the Quality of Service so that the lowest Quality of Service constrained data are located at the edges of the carrier spectrum band and the highest Quality of Service constrained data are placed in the middle of the carrier spectrum band to minimize any effect from the increased adjacent channel interference or noise due to the narrow or overlapped guard band.

5. Clearly present a discussion on the overall system development highlighting mainly the challenges faced along with the assumptions made during code implementation.

Discussion:

The primary task of this project is to generate two message signals, m1(t) and m2(t) which are voice recorded in this matlab software. These two signals are passed through Low Pass filter to avoid any aliasing. Then, they are modulated by carrier frequency fc1 and fc2 respectively. These modulated signals are added by a linear adder and transmitted to the receiver side. This signal is then carried to band pass filter of the desired frequencies. Then these two signals are demodulated by the same carrier frequency ,fc1 and fc2 which were used previously. At the end, both the signals travel through Low pass filter for reconstruction and return back to their original form. Challenges: Although we have completed our task, the original signal could not be reconstructed precisely because of noise. The intensity of the given voice was reduced while some noises were added. Sampling frequency given was 16kHz. But we think, the output could have been much better if the sampling frequency was increased more than the provided one. Lastly, while passing through LPF, noise was reduced a bit and the voice was good enough to be understood but not fully noise free. In audio recorder function, if the rate was increased (instead of 16bit/second), the output may have been better. Assumptions: 1) cutoff frequency of LPF and BPF were considered 4KHz. 2) fc2 was considered double of carrier frequency fc1.