**Title: Frequency Division Multiplexing using MATLAB**

# Abstract:

## This experiment is designed to-

1.To understand the use of MATLAB for solving communication engineering problems. 2.To develop understanding of FDM concept and how to implement it in Matlab.

## An example code of doing FDM in MATLAB is given below:

%Lab 10

%Matlab Program for Frequency Division Multiplexing

%% Message Signal Generation fs = 4001; %Sampling Frequency

t = 0:1/fs:1-1/fs; %Generating Time axis Am1 = 2; %Amplitude of First Message Signal fm1 = 4; %Frequency of First Message Signal

m1 = Am1\*cos(2\*pi\*fm1\*t); % First Message Signal Am2 = 3; %Amplitude of Second Message Signal

fm2 = 5; %Frequency of Second Message Signal

m2 = Am2\*cos(2\*pi\*fm2\*t); % Second Message Signal Am3 = 4; %Amplitude of Third Message Signal

fm3 = 6; %Frequency of Third Message Signal

m3 = Am3\*cos(2\*pi\*fm3\*t); % Third Message Signal

%%

%% Carrier Signal Generation

Cm1 = 1; %Amplitude of First Carrier Signal fc1 = 100; %Frequency of First Carrier Signal

c1 = Cm1\*cos(2\*pi\*fc1\*t); % First Carrier Signal Cm2 = 1; %Amplitude of Second Carrier Signal

fc2 = 170; %Frequency of Second Carrier Signal

c2 = Cm2\*cos(2\*pi\*fc2\*t); % Second Carrier Signal Cm3 = 1; %Amplitude of Third Carrier Signal

fc3 = 250; %Frequency of Third Carrier Signal

c3 = Cm3\*cos(2\*pi\*fc3\*t); % Third Carrier Signal

%%

%% Composite Signal Generation x = (m1).\*c1+(m2).\*c2+(m3).\*c3;

%%%% Plotting the Signals in Time-Domain and Frequency-Domain figure

subplot(3,1,1) plot(t,m1) xlabel('time') ylabel('amplitude')

title('Message Signal 1 in Time Domain') ylim([-Am1 Am1])

subplot(3,1,2) plot(t,m2) xlabel('time') ylabel('amplitude')

title('Message Signal 2 in Time Domain') ylim([-Am2 Am2])

subplot(3,1,3) plot(t,m3) xlabel('time') ylabel('amplitude')

title('Message Signal 3 in Time Domain') ylim([-Am3 Am3])

M1 = abs(fftshift(fft(m1)))/(fs/2); %Fourier Transformation of m1 M2 = abs(fftshift(fft(m2)))/(fs/2); %Fourier Transformation of m2 M3 = abs(fftshift(fft(m3)))/(fs/2); %Fourier Transformation of m3 X = abs(fftshift(fft(x)))/(fs/2); %Fourier Transformation of x

f = fs/2\*linspace(-1,1,fs); figure

subplot(3,1,1) stem(f,M1) xlabel('frequency') ylabel('amplitude')

title('Message Signal 1 in Frequency Domain') axis([-10 10 0 2.5])

subplot(3,1,2) stem(f,M2) xlabel('frequency') ylabel('amplitude')

title('Message Signal 2 in Frequency Domain') axis([-10 10 0 3.5])

subplot(3,1,3) stem(f,M3) xlabel('frequency') ylabel('amplitude')

title('Message Signal 3 in Frequency Domain') axis([-10 10 0 4.5])

figure subplot(2,1,1) plot(t,x) xlabel('time') ylabel('amplitude')

title('Composite Signal in Time Domain') subplot(2,1,2)

stem(f,X) xlabel('frequency')

ylabel('amplitude')

title('Composite Signal in Frequency Domain') axis([-270 270 0 2.5])

%%

%% Passing the Composite Signal Through Bandpass Filter

[num1, den1] = butter(5, [(fc1-fm1-6)/(fs/2),(fc1+fm1+6)/(fs/2)]);

%Butterworth Filter Window Determining for Bandpass Filter bpf1 = filter(num1,den1,x); %Filtering is done here

[num2, den2] = butter(5, [(fc2-fm2-6)/(fs/2),(fc2+fm2+6)/(fs/2)]);

%Butterworth Filter Window Determining for Bandpass Filter bpf2 = filter(num2,den2,x); %Filtering is done here

[num3, den3] = butter(5, [(fc3-fm3-6)/(fs/2),(fc3+fm3+6)/(fs/2)]);

%Butterworth Filter Window Determining for Bandpass Filter bpf3 = filter(num3,den3,x); %Filtering is done here

%%

%% Mixing

z1 = 2\*bpf1.\*c1; z2 = 2\*bpf2.\*c2; z3 = 2\*bpf3.\*c3;

%%

%% Passing the Mixed Signals Through Lowpass Filter

[num4, den4] = butter(5, (fm1+3)/(fs/2)); %Low pass filter is made here rec1 = filter(num4,den4,z1); %Filtering is done here

[num5, den5] = butter(5, (fm2+3)/(fs/2)); %Low pass filter is made here rec2 = filter(num5,den5,z2); %Filtering is done here

[num6, den6] = butter(5, (fm3+3)/(fs/2)); %Low pass filter is made here rec3 = filter(num6,den6,z3); %Filtering is done here

%%

%% Plotting the Received Signals in Time-Domain and Frequency Domain figure

subplot(3,1,1) plot(t,rec1) xlabel('time') ylabel('amplitude')

title('received signal 1 in time domain') ylim([-Am1 Am1])

subplot(3,1,2) plot(t,rec2) xlabel('time') ylabel('amplitude')

title('received signal 2 in time domain') ylim([-Am2 Am2])

subplot(3,1,3) plot(t,rec3) xlabel('time') ylabel('amplitude')

title('received signal 3 in time domain') ylim([-Am3 Am3])

R1 = abs(fftshift(fft(rec1)))/(fs/2); %Fourier Transformation is done here R2 = abs(fftshift(fft(rec2)))/(fs/2); %Fourier Transformation is done here R3 = abs(fftshift(fft(rec3)))/(fs/2); %Fourier Transformation is done here figure

subplot(3,1,1) stem(f,R1) xlabel('frequency') ylabel('amplitude')

title('received signal 1 in frequency domain') xlim([-10 10])

subplot(3,1,2) stem(f,R2) xlabel('frequency') ylabel('amplitude')

title('received signal 2 in frequency domain') xlim([-10 10])

subplot(3,1,3) stem(f,R3) xlabel('frequency') ylabel('amplitude')

title('received signal 3 in frequency domain') xlim([-10 10])

%% End

**Performance Task for Lab Report: (your ID = AB-CDEFG-H)**

I have four message signals:

My id = 20-42139-1

1. mt1 = am1\*cos(2\*pi\*fm1\*t); = 6\*cos(2\*pi\*2\*t);
2. mt2 = am2\*cos(2\*pi\*fm2\*t); = 9\*cos(2\*pi\*3\*t);
3. mt3 = am3\*cos(2\*pi\*fm3\*t); = 12\*cos(2\*pi\*4\*t);
4. mt4= am4\*cos(2\*pi\*fm4\*t); = 15\*cos(2\*pi\*5\*t);

where,

am1=(F+2) =5

am2= (F+5) =8

am3=(F+8) = 11

am4=(F+11) = 14

fm1 = G+1= 9+1=10

fm2= G+2= 11

fm3 = G+3= 12

fm4 = G+4= 13

%Lab 10

%Matlab Program for Frequency Division Multiplexing

>> %% Message Signal Generation

fs = 4001; %Sampling Frequency

t = 0:1/fs:1-1/fs; %Generating Time axis

Am1 = 6; %Amplitude of First Message Signal

fm1 = 2; %Frequency of First Message Signal

m1 = Am1\*cos(2\*pi\*fm1\*t); % First Message Signal

Am2 = 9; %Amplitude of Second Message Signal

fm2 = 3; %Frequency of Second Message Signal

m2 = Am2\*cos(2\*pi\*fm2\*t); % Second Message Signal

Am3 = 12; %Amplitude of Third Message Signal

fm3 = 4; %Frequency of Third Message Signal

m3 = Am3\*cos(2\*pi\*fm3\*t); % Third Message Signal

%%

%% Carrier Signal Generation

Cm1 = 1; %Amplitude of First Carrier Signal

fc1 = 100; %Frequency of First Carrier Signal

c1 = Cm1\*cos(2\*pi\*fc1\*t); % First Carrier Signal

Cm2 = 1; %Amplitude of Second Carrier Signal

fc2 = 170; %Frequency of Second Carrier Signal

c2 = Cm2\*cos(2\*pi\*fc2\*t); % Second Carrier Signal

Cm3 = 1; %Amplitude of Third Carrier Signal

fc3 = 250; %Frequency of Third Carrier Signal

c3 = Cm3\*cos(2\*pi\*fc3\*t); % Third Carrier Signal

%%

%% Composite Signal Generation

x = (m1).\*c1+(m2).\*c2+(m3).\*c3;

%%%% Plotting the Signals in Time-Domain and Frequency-Domain

figure

subplot(3,1,1)

plot(t,m1)

xlabel('time')

ylabel('amplitude')

title('Message Signal 1 in Time Domain')

ylim([-Am1 Am1])

subplot(3,1,2)

plot(t,m2)

xlabel('time')

ylabel('amplitude')

title('Message Signal 2 in Time Domain 20-42139-1 sl 16')

ylim([-Am2 Am2])

subplot(3,1,3)

plot(t,m3)

xlabel('time')

ylabel('amplitude')

title('Message Signal 3 in Time Domain 20-42139-1 sl 16')

ylim([-Am3 Am3])

M1 = abs(fftshift(fft(m1)))/(fs/2); %Fourier Transformation of m1

M2 = abs(fftshift(fft(m2)))/(fs/2); %Fourier Transformation of m2

M3 = abs(fftshift(fft(m3)))/(fs/2); %Fourier Transformation of m3

X = abs(fftshift(fft(x)))/(fs/2); %Fourier Transformation of x

f = fs/2\*linspace(-1,1,fs);

figure

subplot(3,1,1)

stem(f,M1)

xlabel('frequency')

ylabel('amplitude')

title('Message Signal 1 in Frequency Domain')

axis([-10 10 0 2.5])

subplot(3,1,2)

stem(f,M2)

xlabel('frequency')

ylabel('amplitude')

title('Message Signal 2 in Frequency Domain 20-42139-1 sl 16')

axis([-10 10 0 3.5])

subplot(3,1,3)

stem(f,M3)

xlabel('frequency')

ylabel('amplitude')

title('Message Signal 3 in Frequency Domain 20-42139-1 sl 16')

axis([-10 10 0 4.5])

figure

subplot(2,1,1)

plot(t,x)

xlabel('time')

ylabel('amplitude')

title('Composite Signal in Time Domain 20-42139-1 sl 16')

subplot(2,1,2)

stem(f,X)

xlabel('frequency')

ylabel('amplitude')

title('Composite Signal in Frequency Domain 20-42139-1 sl 16')

axis([-270 270 0 2.5])

%%

%% Passing the Composite Signal Through Bandpass Filter

[num1, den1] = butter(5, [(fc1-fm1-6)/(fs/2),(fc1+fm1+6)/(fs/2)]);

%Butterworth Filter Window Determining for Bandpass Filter

bpf1 = filter(num1,den1,x); %Filtering is done here

[num2, den2] = butter(5, [(fc2-fm2-6)/(fs/2),(fc2+fm2+6)/(fs/2)]);

%Butterworth Filter Window Determining for Bandpass Filter

bpf2 = filter(num2,den2,x); %Filtering is done here

[num3, den3] = butter(5, [(fc3-fm3-6)/(fs/2),(fc3+fm3+6)/(fs/2)]);

%Butterworth Filter Window Determining for Bandpass Filter

bpf3 = filter(num3,den3,x); %Filtering is done here

%%

%% Mixing

z1 = 2\*bpf1.\*c1;

z2 = 2\*bpf2.\*c2;

z3 = 2\*bpf3.\*c3;

%%

%% Passing the Mixed Signals Through Lowpass Filter

[num4, den4] = butter(5, (fm1+3)/(fs/2)); %Low pass filter is made here

rec1 = filter(num4,den4,z1); %Filtering is done here

[num5, den5] = butter(5, (fm2+3)/(fs/2)); %Low pass filter is made here

rec2 = filter(num5,den5,z2); %Filtering is done here

[num6, den6] = butter(5, (fm3+3)/(fs/2)); %Low pass filter is made here

rec3 = filter(num6,den6,z3); %Filtering is done here

%%

%% Plotting the Received Signals in Time-Domain and Frequency Domain

figure

subplot(3,1,1)

plot(t,rec1)

xlabel('time')

ylabel('amplitude')

title('received signal 1 in time domain')

ylim([-Am1 Am1])

subplot(3,1,2)

plot(t,rec2)

xlabel('time')

ylabel('amplitude')

title('received signal 2 in time domain')

ylim([-Am2 Am2])

subplot(3,1,3)

plot(t,rec3)

xlabel('time')

ylabel('amplitude')

title('received signal 3 in time domain')

ylim([-Am3 Am3])

R1 = abs(fftshift(fft(rec1)))/(fs/2); %Fourier Transformation is done here

R2 = abs(fftshift(fft(rec2)))/(fs/2); %Fourier Transformation is done here

R3 = abs(fftshift(fft(rec3)))/(fs/2); %Fourier Transformation is done here

figure

subplot(3,1,1)

stem(f,R1)

xlabel('frequency')

ylabel('amplitude')

title('received signal 1 in frequency domain')

xlim([-10 10])

subplot(3,1,2)

stem(f,R2)

xlabel('frequency')

ylabel('amplitude')

title('received signal 2 in frequency domain')

xlim([-10 10])

subplot(3,1,3)

stem(f,R3)

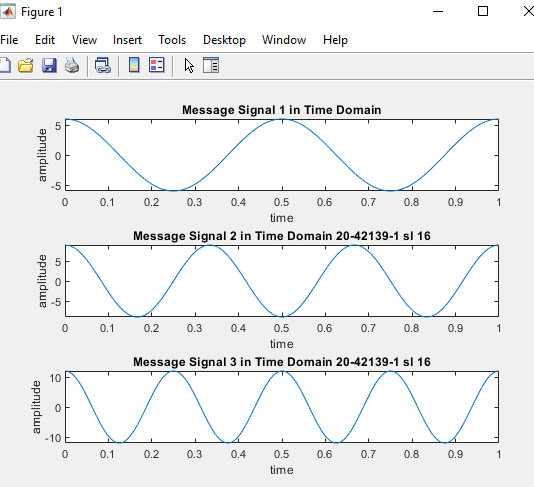
xlabel('frequency')

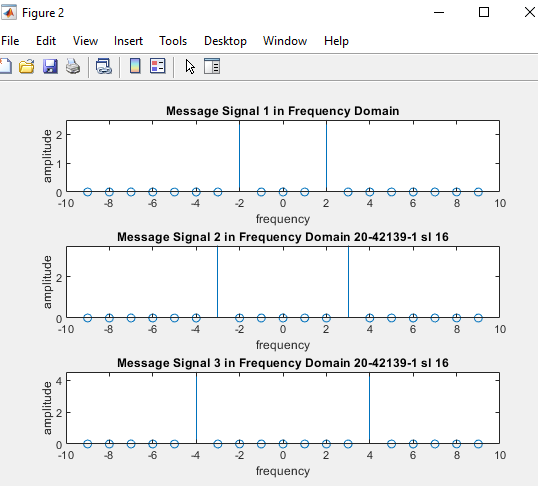
ylabel('amplitude')

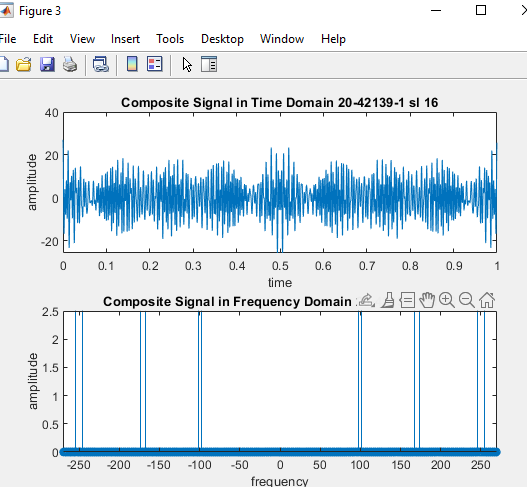
title('received signal 3 in frequency domain 20-42139-1 sl 16')

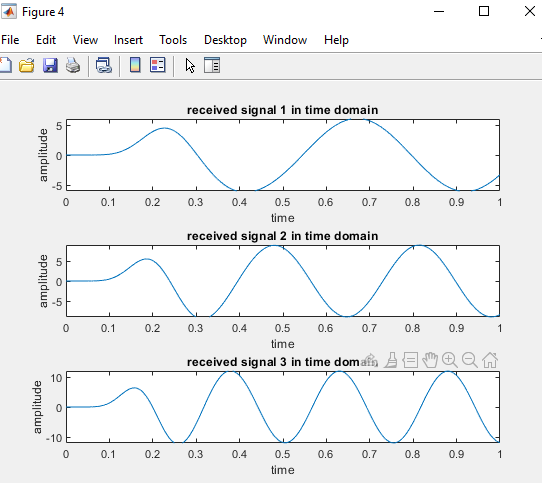
xlim([-10 10])

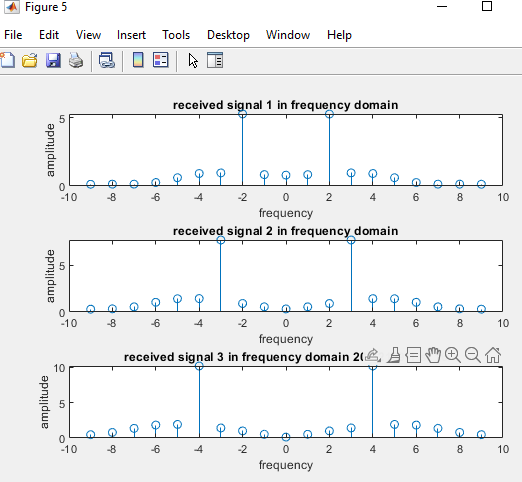
%% End











We want to simultaneously transmit these four signals through a single data link that can support a frequency range of 50 Hz to 250 Hz.

\*\*\* Write a code that can modulate and multiplex the four given message signals in transmitting side (use appropriate carrier signals for amplitude modulation as required) and de-multiplex (use appropriate cut- off frequencies in your bandpass filters) and de-modulate (use appropriate cut-off frequencies in your lowpass filters) to recover the four message signals in receiving side