American International University- Bangladesh (AIUB) Faculty of Engineering (FE)

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | DATA COMMUNICATION | **Course Code:** | COE3103 |
| **Semester:** | Summer 2024-25 | **Section:** | D |
| **Faculty:** | MOHAMMAD ASADUZZAMAN KHAN | **Group:** | 05 |

|  |  |
| --- | --- |
| **Experiment No:** | 08 |
| **Experiment Name:** | Study of Frequency Division Multiplexing (FDM) using MATLAB |

|  |  |  |  |
| --- | --- | --- | --- |
| **Group Members** | | **Name** | **ID** |
|  | 1. | Basudeb Kundu | 23-50856-1 |
| 2. | Prohlad Chandra Das | 23-50922-1 |
| 3. | Debashis Kumar Das | 23-50953-1 |
| 4. | Indronill Dutta Nill | 23-50974-1 |
| 5. | Nafiur Rahman Nirob | 23-50991-1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance Date:** | 01-9-25 | **Due Date:** | 07-9-25 |

**Marking Rubrics (to be filled by Lab Instructor)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Category | Proficient [6] | Good [4] | Acceptable [2] | Unacceptable [1] | Secured Marks |
| **Theoretical Background, Methods & procedures sections** | All information, measures and variables are provided and  explained. | All Information provided is sufficient, but more explanation is  needed. | Most information is correct, but some information may be  missing or inaccurate. | Much information is missing and/or inaccurate. |  |
| **Results** | All of the criteria are met; results are described clearly and accurately; | Most criteria are met, but there may be some lack of clarity and/or incorrect information. | Experimental results don’t match exactly with the theoretical values and/or analysis  is unclear. | Experimental results are missing or incorrect; |  |
| **Discussion** | Demonstrates thorough and sophisticated understanding.  Conclusions drawn are appropriate for  analyses; | Hypotheses are clearly stated, but some concluding statements not supported by data or data not well  integrated. | Some hypotheses missing or misstated; conclusions not supported by data. | Conclusions don’t match hypotheses, not supported by data; no integration of data from different sources. |  |
| **General formatting** | Title page, placement of figures and figure captions, and other  formatting issues all correct. | Minor errors in formatting. | Major errors and/or missing information. | Not proper style in text. |  |
| **Writing & organization** | Writing is strong and easy to understand; ideas are fully elaborated and connected; effective transitions between sentences; no  typographic, spelling, or grammatical errors. | Writing is clear and easy to understand; ideas are connected; effective transitions between sentences; minor typographic, spelling, or grammatical errors. | Most of the required criteria are met, but some lack of clarity, typographic, spelling, or grammatical errors are present. | Very unclear, many errors. |  |
| Comments: |  | | | Total Marks (Out of **30**): |  |

**Title**

Study of Frequency Division Multiplexing (FDM) 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.

**Introduction:**

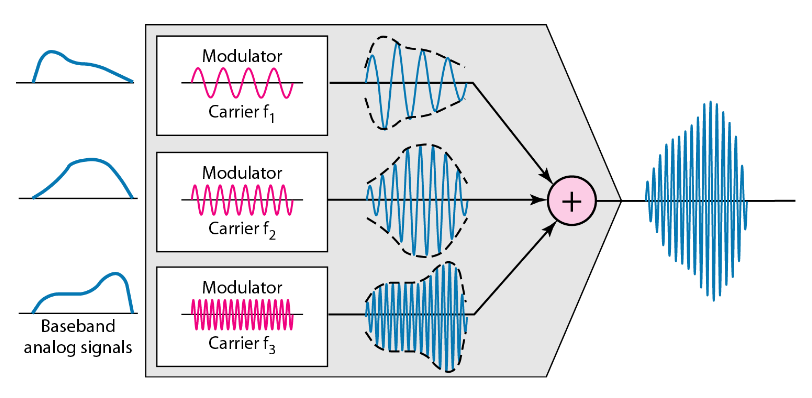
Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted. In FDM, signals generated by each sending device modulate different carrier frequencies. These modulated signals are then combined into a single composite signal that can be transported by the link. Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal. These bandwidth ranges are the channels through which the various signals travel. Channels can be separated by strips of unused bandwidth—guard bands— to prevent signals from overlapping. In addition, carrier frequencies must not interfere with the original data frequencies.

Figure 1 gives a conceptual view of FDM. In this illustration, the transmission path is divided into three parts, each representing a channel that carries one transmission.



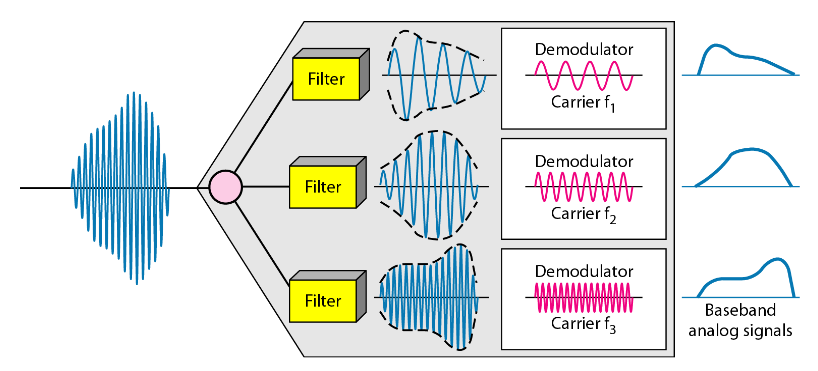
**Fig-1: Frequency Division Multiplexing (FDM)**

**Multiplexing Process:** Figure 2 is a conceptual illustration of the multiplexing process. Each source generates a signal of a similar frequency range. Inside the multiplexer, these similar signals modulate different carrier frequencies ( f1, f2, and f3). The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it



**Fig-2: Multiplexing Process in FDM**

**Demultiplexing Process:** The demultiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals. The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines. Figure 3 is a conceptual illustration of demultiplexing process.



**Fig-3: De-multiplexing Process in FDM**

**Results and Discussion**

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

**AB-CDEFG-H**

**23-50922-1**

Generates four message signals as:

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

where:

**F=2 and G=2**

**So,**

am1 = (F+2) =4;

am2 = (F+5) =7;

am3 = (F+8) =10;

am4 = (F+11) =13;

and

fm1 = (G+1) =3;

fm2 = (G+2) =4;

fm3 = (G+3) =5;

fm4 = (G+4) =6;

Frequency range is 50 Hz - 250 Hz.

|  |
| --- |
| **Code:** |
| clc;  clear all;  close all;  fs = 4001; %Sampling Frequency  t = 0:1/fs:1-1/fs; %Generating Time axis  Am1 = 4; %Amplitude of First Message Signal  fm1 = 3; %Frequency of First Message Signal  m1 = Am1\*cos(2\*pi\*fm1\*t); % First Message Signal  Am2 = 7; %Amplitude of Second Message Signal  fm2 = 4; %Frequency of Second Message Signal  m2 = Am2\*cos(2\*pi\*fm2\*t); % Second Message Signal  Am3 = 10; %Amplitude of Third Message Signal  fm3 = 5; %Frequency of Third Message Signal  m3 = Am3\*cos(2\*pi\*fm3\*t); % Third Message Signal  Am4 = 13; %Amplitude of Third Message Signal  fm4 = 6; %Frequency of Third Message Signal  m4 = Am4\*cos(2\*pi\*fm4\*t); % 4th Message Signal  %% Carrier Signal Generation  Cm1 = 1; %Amplitude of First Carrier Signal  fc1 = 50; %Frequency of First Carrier Signal  c1 = Cm1\*cos(2\*pi\*fc1\*t); % First Carrier Signal  Cm2 = 1; %Amplitude of Second Carrier Signal  fc2 = 100; %Frequency of Second Carrier Signal  c2 = Cm2\*cos(2\*pi\*fc2\*t); % Second Carrier Signal  Cm3 = 1; %Amplitude of Third Carrier Signal  fc3 = 150; %Frequency of Third Carrier Signal  c3 = Cm3\*cos(2\*pi\*fc3\*t); % Third Carrier Signal  Cm4 = 1; %Amplitude of Third Carrier Signal  fc4 = 250; %Frequency of Third Carrier Signal  c4 = Cm4\*cos(2\*pi\*fc4\*t); % 4th Carrier Signal  %% Composite Signal Generation  x = (m1).\*c1+(m2).\*c2+(m3).\*c3+(m4).\*c4;    % Plotting the Signals in Time-Domain and Frequency-Domain figure  subplot(4,1,1)  plot(t,m1)  xlabel('time')  ylabel('amplitude')  title('Message Signal 1 in Time Domain')  ylim([-Am1 Am1])  subplot(4,1,2)  plot(t,m2)  xlabel('time')  ylabel('amplitude')  title('Message Signal 2 in Time Domain')  ylim([-Am2 Am2])  subplot(4,1,3)  plot(t,m3)  xlabel('time')  ylabel('amplitude')  title('Message Signal 3 in Time Domain')  ylim([-Am3 Am3])  subplot(4,1,4)  plot(t,m4)  xlabel('time')  ylabel('amplitude')  title('Message Signal 4 in Time Domain')  ylim([-Am4 Am4])  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  M4 = abs(fftshift(fft(m4)))/(fs/2); %Fourier Transformation of m4  X = abs(fftshift(fft(x)))/(fs/2); %Fourier Transformation of x  f = fs/2\*linspace(-1,1,fs);  figure  subplot(4,1,1)  stem(f,M1)  xlabel('frequency')  ylabel('amplitude')  title('Message Signal 1 in Frequency Domain')  axis([-10 10 0 10])  subplot(4,1,2)  stem(f,M2)  xlabel('frequency')  ylabel('amplitude')  title('Message Signal 2 in Frequency Domain')  axis([-10 10 0 15])  subplot(4,1,3)  stem(f,M3)  xlabel('frequency')  ylabel('amplitude')  title('Message Signal 3 in Frequency Domain')    axis([-10 10 0 15])  subplot(4,1,4)  stem(f,M4)  xlabel('frequency')  ylabel('amplitude')  title('Message Signal 4 in Frequency Domain')  axis([-10 10 0 20])  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([50 250 0 10])  %% Passing the Composite Signal Through Bandpass Filter  [num1, den1] = butter(5, [(fc1-fm1-4)/(fs/2),(fc1+fm1+4)/(fs/2)]);  %Butterworth Filter Window Determining for Bandpass Filter  bpf1 = filter(num1,den1,x); %Filtering is done here  [num2, den2] = butter(5, [(fc2-fm2-4)/(fs/2),(fc2+fm2+4)/(fs/2)]);  %Butterworth Filter Window Determining for Bandpass Filter  bpf2 = filter(num2,den2,x); %Filtering is done here  [num3, den3] = butter(5, [(fc3-fm3-4)/(fs/2),(fc3+fm3+4)/(fs/2)]);  %Butterworth Filter Window Determining for Bandpass Filter  bpf3 = filter(num3,den3,x); %Filtering is done here  [num7, den7] = butter(5, [(fc4-fm4-4)/(fs/2),(fc4+fm4+4)/(fs/2)]);  %Butterworth Filter Window Determining for Bandpass Filter  bpf4 = filter(num7,den7,x); %Filtering is done here  % Mixing  z1 = bpf1.\*c1; z2 = 2\*bpf2.\*c2; z3 = 2\*bpf3.\*c3; z4 = 2\*bpf4.\*c4;  %% 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  [num8, den8] = butter(5, (fm4+3)/(fs/2)); %Low pass filter is made here  rec4 = filter(num8,den8,z4); %Filtering is done here  % Plotting the Received Signals in Time-Domain and Frequency Domain  figure  subplot(4,1,1)  plot(t,rec1)  xlabel('time')  ylabel('amplitude')  title('received signal 1 in time domain')  ylim([-10 10])  subplot(4,1,2)  plot(t,rec2)  xlabel('time')  ylabel('amplitude')  title('received signal 2 in time domain')  ylim([-Am2 Am2])  subplot(4,1,3)  plot(t,rec3)  xlabel('time')  ylabel('amplitude')  title('received signal 3 in time domain')  ylim([-Am3 Am3])  subplot(4,1,4)  plot(t,rec4)  xlabel('time')  ylabel('amplitude')  title('received signal 4 in time domain')  ylim([-Am4 Am4])  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  R4 = abs(fftshift(fft(rec4)))/(fs/2); %Fourier Transformation is done here  figure  subplot(4,1,1)    stem(f,R1)  xlabel('frequency')  ylabel('amplitude')  title('received signal 1 in frequency domain')  xlim([0 10])  ylim([0 15])  subplot(4,1,2)  stem(f,R2)  xlabel('frequency')  ylabel('amplitude')  title('received signal 2 in frequency domain')  xlim([0 10])  ylim([0 15])  subplot(4,1,3)  stem(f,R3)  xlabel('frequency')  ylabel('amplitude')  title('received signal 3 in frequency domain')  xlim([0 10])  ylim([0 15])  subplot(4,1,4)  stem(f,R4)  xlabel('frequency')  ylabel('amplitude')  title('received signal 4 in frequency domain')  xlim([0 10])  ylim([0 20]) |

|  |
| --- |
| **Result and Discussion:** |
| **Fig4: Message Signals 1–4 in Time Domain** |
| **Fig5: Message Signals 1–4 in Frequency Domain** |
| **Fig6:** : **Composite Signal in Time Domain and Frequency Domain** |
| **Fig7: Received Signals 1–4 in Time Domain** |
| **Fig8: Received Signals 1–4 in Frequency Domain** |
| This experiment demonstrated the principle of **Frequency Division Multiplexing (FDM)** using MATLAB by transmitting multiple signals over a single communication channel. Four message signals with different amplitudes and frequencies were generated and modulated using DSB-SC amplitude modulation, each assigned to a unique carrier frequency.  The selected carrier frequencies (50 Hz, 100 Hz, 150 Hz, and 250 Hz) were spaced sufficiently apart to avoid spectral overlap, ensuring proper separation during multiplexing. The modulated signals were then combined into a composite signal, representing the multiplexed data stream.  Time-domain and frequency-domain analysis of the composite signal confirmed the presence of distinct frequency components corresponding to each modulated channel. At the receiver, bandpass filters were employed to extract individual signals, followed by coherent demodulation and lowpass filtering to recover the original message signals.  The recovered signals closely resembled the original inputs, showing minimal distortion. Minor discrepancies observed could be attributed to filter imperfections or insufficient frequency spacing. This highlights the importance of appropriate carrier frequency selection and filter design in FDM systems. The experiment further demonstrated how MATLAB can serve as a powerful simulation tool for analyzing and visualizing communication processes. |

**Conclusion**

The experiment successfully demonstrated the concept of Frequency Division Multiplexing (FDM) using MATLAB. Four different message signals were assigned to separate carrier frequencies, combined into a composite signal, and later recovered at the receiver through appropriate filtering and demodulation.

The outcome showed that FDM can reliably transmit multiple signals simultaneously, provided that carrier spacing and filter design are carefully managed. This work not only reinforced the theoretical principles of multiplexing but also highlighted MATLAB’s effectiveness in modeling and analyzing communication systems.