**American International University- Bangladesh (AIUB)**

**Faculty of Engineering**

**Data Communications Lab**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Data Communications** | | |
| **Course Code:** | CoE 3201 | **Section:** | D |
| **Semester:** | Summer 2024-25 | **Group No:** | 5 |
| **Assignment Name:** | **Open Ended Lab** | | |
| **Assessed CO5:** | **Accepts and recognizes the role of shift keying (ASK) and multiplexing (FDM) to communicate binary bits as analog signals through multiple channels in society, health, safety, legal and culture** | | |
| **Assessed POI:** | **P.f.1.A3** | | |
| **Student Name:** | PROHLAD CHANDRA DAS | **Student ID:** | 23-50922-1 |
| **Student Name:** | DEBASHIS KUMAR DAS | **Student ID:** | 23-50953-1 |
| **Student Name:** | BASUDEB KUNDU | **Student ID:** | 23-50856-1 |
| **Student Name:** | INDRONILL DATTA NILL | **Student ID:** | 23-50974-1 |
| **Student Name:** | NAFIUR RAHMAN NIROB | **Student ID:** | 23-50991-1 |
| **Student Name:** |  | **Student ID:** |  |
| **Student Name:** |  | **Student ID:** |  |

**Mark distribution (to be filled by Faculty):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objectives | Proficient [9-10] | Good  [6-8] | Needs Improvement  [1-5] | Secured Marks |
| **Depth of knowledge displayed through**  **appropriate**  **research**  **(P1)** | Student was able to apply in- depth engineering knowledge achieved by appropriate research about digital/analog communication to design the  communication model  correctly and **fulfilled all design criteria**. | Design process is not completely supported by in-depth engineering knowledge achieved by appropriate research about digital/analog communication, **some but not all of the design criteria are fulfilled**. | Design process contains mistakes and does not display enough in- depth engineering knowledge achieved by appropriate research  about digital/analog  communication. **Most of the design criteria are not fulfilled**. |  |
| **Depth of analysis**  **(P3)** | Student defended the diversified approach taken to solve the problem with **well- justified in-depth analysis that demonstrated abstract thinking**. | Student’s attempts to analyze the diversified approach taken to solve the problem **is not enough in-depth, some of design choices do not demonstrate adequate abstract thinking** and are not properly justified. | Student **did not attempt any in- depth analysis** of the designed system and **displayed no abstract thinking**. |  |
| **Level of integration of multiple sections of design for**  **solution of high-level**  **problem**  **(P7)** | Student correctly identified all problems and successfully integrated the interdependent parts into a high-level design using a block diagram.  Block diagram was at best match with the given problem. | Student was able to identify some of the problems correctly and integrated the interdependent parts into a high-level design using a block diagram.  Some parts of the block diagram were not a good match for the given problem. | Student was able to identify only one/two of the problems correctly and could not properly integrate the interdependent parts into a high-level design using a block diagram.  Only one/two blocks were correct and/or block diagram was incomplete. |  |
| **Comments:** |  |  | **Total Marks (Out of 30):** |  |
|  |  |  | **Converted to (15)** |  |

Title

Binary Data Transmission using BASK Modulation and FDM with ASCII Encoding and Decoding.

**Objective:** The objective of this experiment is to implement and analyze a complete digital communication system using MATLAB. The study involves converting the surnames of group members into binary vectors, representing them in unipolar NRZ format, and applying Binary Amplitude Shift Keying (BASK) modulation. The modulated signals are combined using Frequency Division Multiplexing (FDM) with the addition of Gaussian noise to simulate a realistic transmission environment. Subsequently, the composite signal is demultiplexed and demodulated to recover the original binary sequences. Finally, the reconstructed digital signals are decoded back into ASCII characters to verify accuracy. This experiment provides practical insights into the concepts of digital signal transmission, modulation, multiplexing, noise effects, and signal recovery, reinforcing the understanding of fundamental digital communication techniques.

**Theory**

**1.Frequency Division Multiplexing (FDM)**Frequency Division Multiplexing (FDM) is a communication technique that allows multiple signals to be transmitted simultaneously over a single communication channel. Each signal is assigned a unique carrier frequency, ensuring that they do not interfere with each other. At the receiver, bandpass filters are used to separate the individual signals, allowing each one to be independently recovered. FDM is commonly used in telephone networks, radio broadcasting, and various digital communication systems to efficiently utilize bandwidth.

# **Binary Amplitude Shift Keying (BASK) Modulation** BASK is a form of digital modulation in which the amplitude of a carrier wave is varied according to the binary data:

# Bit “1” is represented by a carrier signal of fixed amplitude.

# Bit “0” is represented by the absence of the carrier (zero amplitude). BASK is simple to implement and serves as a foundational technique in digital communication systems.

# **BASK Demodulation** BASK demodulation is the process of retrieving the original binary data from a BASK-modulated signal. The standard steps include:

# Bandpass Filtering: Isolate the desired carrier frequency from a composite or noisy signal (essential in FDM).

# Coherent Detection: Multiply the received signal by a synchronized carrier waveform to shift it back to baseband.

# Low-pass Filtering: Remove high-frequency components, leaving the original digital waveform.

# Threshold Decision: Sample the signal at bit intervals; voltages above a predefined threshold are interpreted as “1”, while those below are interpreted as “0”. This method allows accurate recovery of transmitted digital information even in the presence of noise, ensuring reliable communication.

# Codes and Results

***Code For Task One:***

We have selected 2 surnames of the group “DAS “ and “DATTA” :-

Code:

clc; clear; close all;

%% Step 1: Surnames of two group members

name1 = 'NILL'; name2 = 'DAS';

% Convert names to binary (8-bit ASCII, MSB first)

binMat1 = dec2bin(name1, 8)';

% Each column is one char

binMat2 = dec2bin(name2, 8)';

% Flatten column-wise to form binary stream (synchronous serial transmission)

x1 = binMat1(:) - '0';

% Convert char to numeric

x2 = binMat2(:) - '0';

% Display binary vectors

disp('Binary vector x1 (NILL):');

disp(x1');

disp('Binary vector x2 (DAS):');

disp(x2');

%% Optional: Digital waveform visualization

bp = 1e-3; samples\_per\_bit = 100; t\_bit = linspace(0, bp, samples\_per\_bit);

% Function to generate NRZ waveform

generateNRZ = @(bitstream) reshape(repmat(bitstream \* 5, samples\_per\_bit, 1), 1, []);

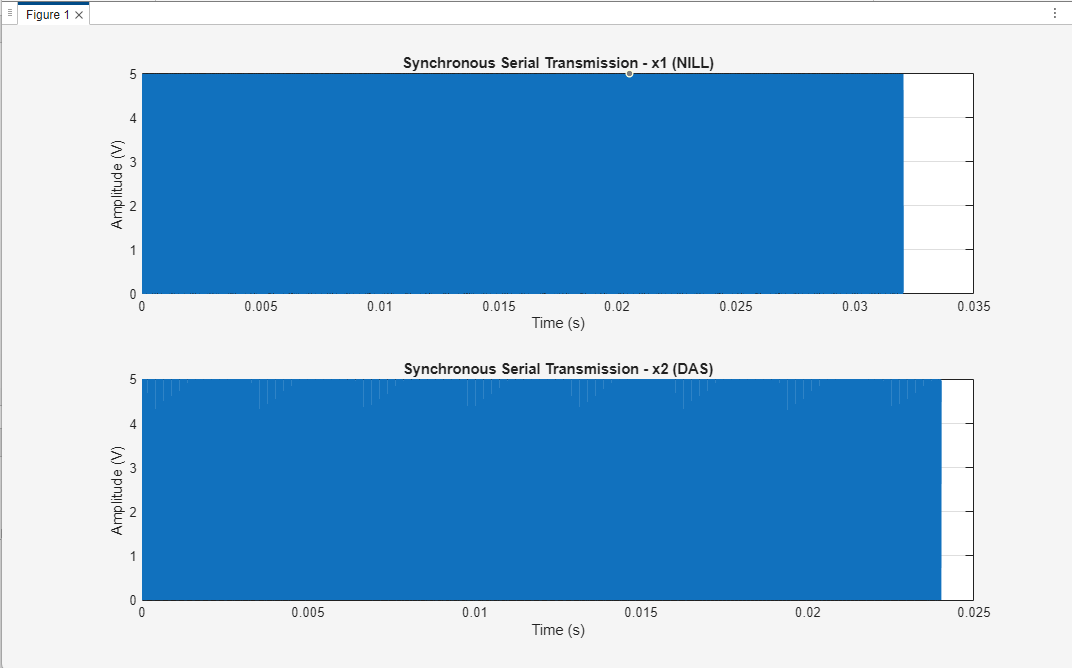
nrz1 = generateNRZ(x1); nrz2 = generateNRZ(x2);

t1 = linspace(0, length(x1)\*bp, length(nrz1)); t2 = linspace(0, length(x2)\*bp, length(nrz2));

figure;

subplot(2,1,1); plot(t1, nrz1, 'LineWidth', 2); grid on; xlabel('Time (s)'); ylabel('Amplitude (V)'); title('Synchronous Serial Transmission - x1 (NILL)');

subplot(2,1,2); plot(t2, nrz2, 'LineWidth', 2); grid on; xlabel('Time (s)'); ylabel('Amplitude (V)'); title('Synchronous Serial Transmission - x2 (DAS)');



A rectangular white rectangular object with black dots

AI-generated content may be incorrect.

Figure 1 :

***Code for task two:***

clc; clear; close all;

%% Step 1: Group member surnames

surname1 = 'NILL';

surname2 = 'DAS';

% Convert each name to 8-bit ASCII binary (MSB first)

binMat1 = dec2bin(surname1, 8)';

% % Transpose so it's column-wise

binMat2 = dec2bin(surname2, 8)';

% Convert binary strings to numerical arrays

x1 = binMat1(:) - '0';

% Binary bitstream for 'NILL'

x2 = binMat2(:) - '0';

% Binary bitstream for 'DAS'

%% Step 2: NRZ Signal Parameters

bit\_duration = 1e-3;

% 1 ms per bit

samples\_per\_bit = 500;

% 500 samples per bit (good resolution)

fs = samples\_per\_bit / bit\_duration;

% Sampling frequency (Hz)

% Total number of samples for each bitstream

n1 = length(x1) \* samples\_per\_bit; n2 = length(x2) \* samples\_per\_bit;

% Generate time vectors of matching length

t1 = linspace(0, length(x1)\*bit\_duration, n1); t2 = linspace(0, length(x2)\*bit\_duration, n2);

% Voltage levels

max\_voltage = 5;

min\_voltage = 0;

%% Step 3: Build NRZ signal for x1 (RAHUL)

for i = 1:length(x1)

idx\_start = (i-1)\*samples\_per\_bit + 1; idx\_end = i\*samples\_per\_bit;

if x1(i) == 1

nrz1(idx\_start:idx\_end) = max\_voltage;

else

nrz1(idx\_start:idx\_end) = min\_voltage;

end

end

%% Step 4: Build NRZ signalfor x2 (CHAITI)

for i = 1:length(x2)

idx\_start = (i-1)\*samples\_per\_bit + 1; idx\_end = i\*samples\_per\_bit;

if x2(i) == 1

nrz2(idx\_start:idx\_end) = max\_voltage;

else

nrz2(idx\_start:idx\_end) = min\_voltage;

end

end

%% Step 5: Plot both NRZ signals using subplot

figure(1);

subplot(2,2,1); plot(t1, nrz1, 'b', 'LineWidth', 1.5); grid on; xlabel('Time (s)'); ylabel('Voltage (V)'); ylim([-1 6]);

title('Unipolar NRZ - x1 (NILL)');

subplot(2,2,2); plot(t2, nrz2, 'r', 'LineWidth', 1.5); grid on; xlabel('Time (s)'); ylabel('Voltage (V)'); ylim([-1 6]); title('Unipolar NRZ - x2 (DAS)');

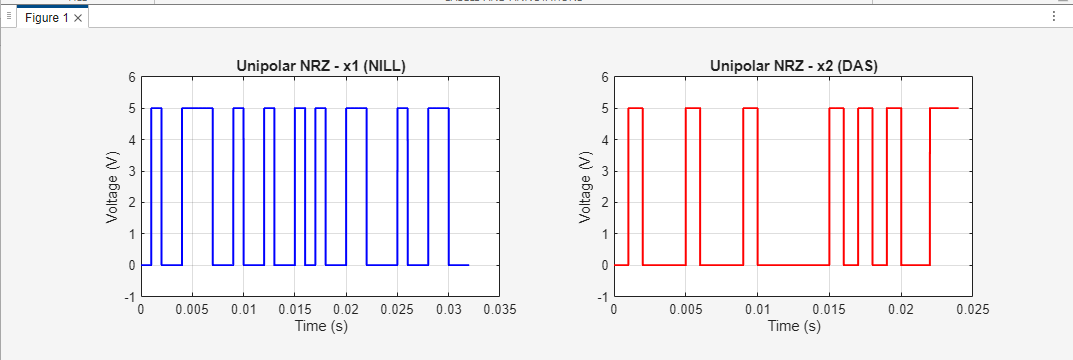


Figure 2

***Code for task three:***

clc; clear; close all;

%Surname

surname1 = 'NILL';

surname2 = 'DAS';

% Convert surnames to binary vectors (8 bits per char)

binMat1 = dec2bin(surname1, 8)';

binMat2 = dec2bin(surname2, 8)';

x1 = binMat1(:) - '0';

% binary vector for 'NILL'

x2 = binMat2(:) - '0';

% binary vector for 'DAS'

% Parameters

bp = 0.001;

% bit period (1 ms)

A1 = 5;

% amplitude for bit 1

A0 = 0;

% amplitude for bit 0

f1 = 2;

% carrier frequency for x1 (Hz)

f2 = 4;

% carrier frequency for x2 (Hz)

% Time vector for one bit (make 100 samples per bit)

samples\_per\_bit = 100; t\_bit = linspace(0, bp, samples\_per\_bit);

% BASK for surname1 ('NILL')

m1 = [];

for i = 1:length(x1)

if x1(i) == 1

y = A1 \* cos(2 \* pi \* f1 \* t\_bit);

else

y = A0 \* cos(2 \* pi \* f1 \* t\_bit);

end

m1= [m1 y];

end

% Time vector for entire signal

t1 = linspace(0, bp\*length(x1), length(m1));

% BASK for surname2 ('DAS')

m2 = [];

for i = 1:length(x2)

if x2(i) == 1

y = A1 \* cos(2 \* pi \* f2 \* t\_bit);

else

y = A0 \* cos(2 \* pi \* f2 \* t\_bit);

end

m2 = [m2 y];

end

% Time vector for entire signal

t2 = linspace(0, bp\*length(x2), length(m2));

% Plotting

figure;

subplot(2,1,1); plot(t1, m1, 'b', 'LineWidth', 1.2); axis([0 bp\*length(x1) -6 6]); grid on; xlabel('Time (s)'); ylabel('Amplitude (V)');

title('BASK Modulated Signal (NILL, 2 Hz)');

subplot(2,1,2); plot(t2, m2, 'r', 'LineWidth', 1.2); axis([0 bp\*length(x2) -6 6]); grid on; xlabel('Time (s)'); ylabel('Amplitude (V)');

title('BASK Modulated Signal (DAS, 4 Hz)');

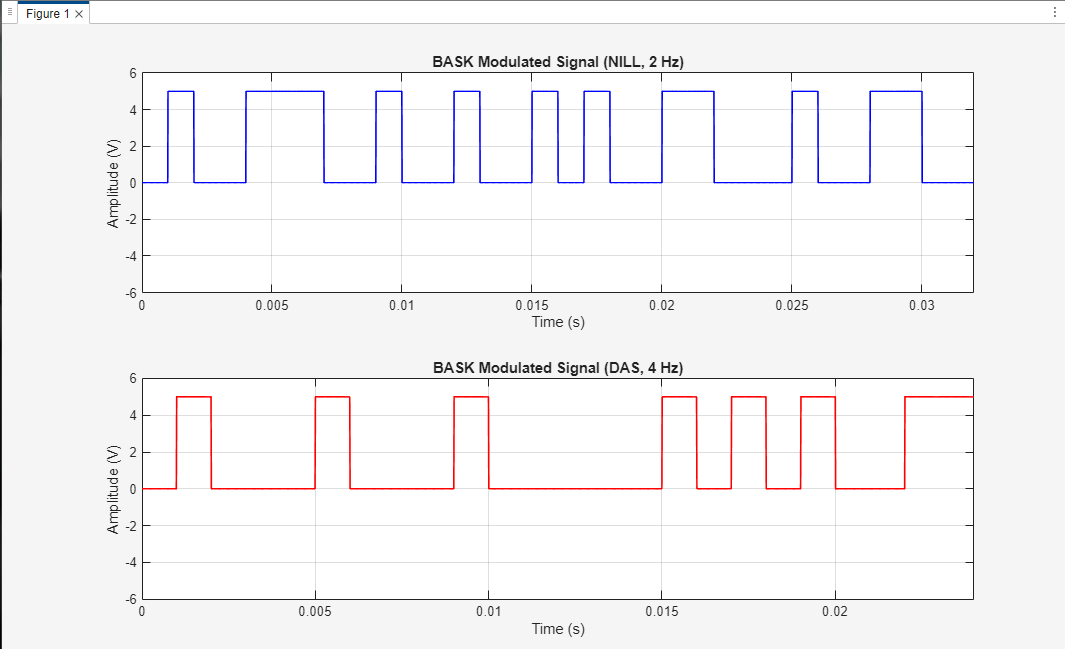


Figure 3

***Code for task four:***clc; clear;

close all;

% --- BASK modulation parameters ---

bp = 0.001;

% bit period (1 ms)

A1 = 5;

% amplitude for bit 1

A0 = 0;

% amplitude for bit 0

f1 = 2;

% carrier frequency for x1 (Hz)

f2 = 4;

% carrier frequency for x2 (Hz)

t\_bit = bp/99 : bp/99 : bp;

% time vector for one bit (99 samples)

% --- Example binary sequences for x1 and x2 (from previous code) ---

x1 = [ 0 1 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0];

% sample bits for signal 1

x2 = [0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 1];

% sample bits for signal 2

% --- Make sure x1 and x2 have same length by padding zeros to shorter one ---

if length(x1) < length(x2)

x1 = [x1 zeros(1, length(x2) - length(x1))];

else

length(x2) < length(x1)

x2 = [x2 zeros(1, length(x1) - length(x2))];

end

% --- BASK modulation for x1 ---

m1 = [];

for i = 1:length(x1)

if x1(i) == 1

y = A1\*cos(2\*pi\*f1\*t\_bit);

else

y = A0\*cos(2\*pi\*f1\*t\_bit);

end

m1 = [m1 y];

end

% --- BASK modulation for x2 ---

m2 = [];

for i = 1:length(x2)

if x2(i) == 1

y = A1\*cos(2\*pi\*f2\*t\_bit);

else

y = A0\*cos(2\*pi\*f2\*t\_bit);

end

m2 = [m2 y];

end

+

% --- Time vector for the entire signal ---

t = bp/99 : bp/99 : bp\*length(x1);

% --- Frequency Division Multiplexing (FDM) ---

composite\_signal = m1 + m2;

% --- Add Gaussian noise ---

SNR\_dB = 20;

% Signal-to-noise ratio in dB

noisy\_signal = awgn(composite\_signal, SNR\_dB, 'measured');

% --- Plotting --- figure(2);

subplot(3,1,1);

plot(t, m1, 'b'); title('BASK Modulated Signal x1 (f = 2 Hz)');

xlabel('Time (s)'); ylabel('Amplitude (V)'); axis([0 t(end) -6 6]); grid on;

subplot(3,1,2);

plot(t, m2, 'r'); title('BASK Modulated Signal x2 (f = 4 Hz)');

xlabel('Time (s)'); ylabel('Amplitude (V)'); axis([0 t(end) -6 6]); grid on;

subplot(3,1,3); plot(t, noisy\_signal, 'k'); title(['Composite FDM Signal with AWGN (SNR = ' num2str(SNR\_dB) ' dB)']); xlabel('Time (s)'); ylabel('Amplitude (V)'); axis([0 t(end) -10 10]); grid on;

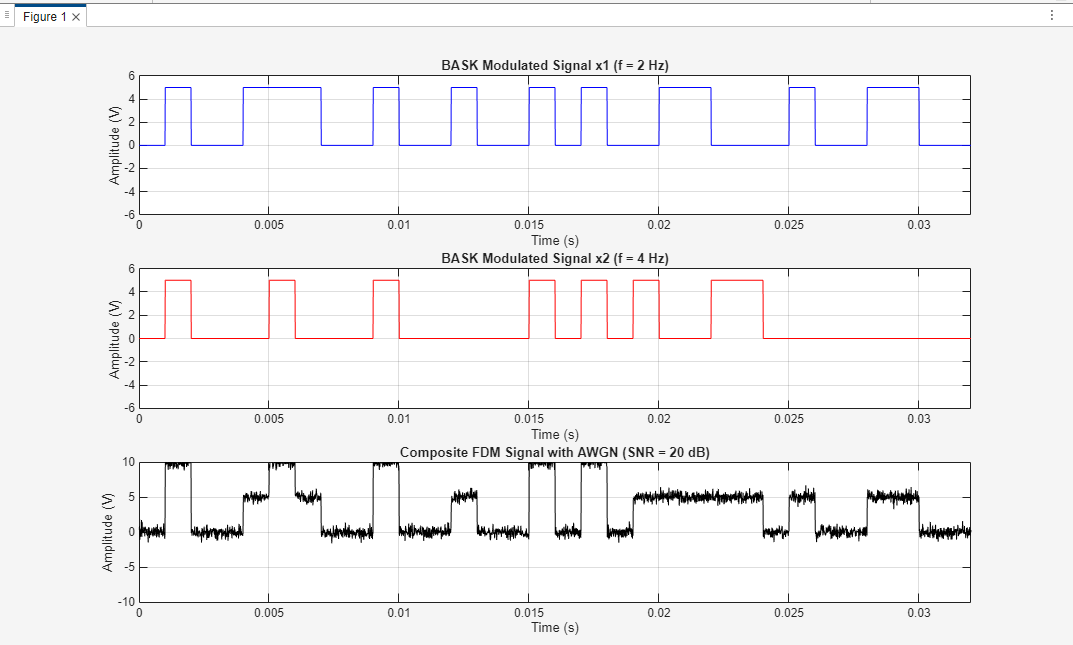


Figure 4

clc; clear;

close all;

% ----- Original bit streams (from ASCII of two names) -----

surname1 = 'NILL';

surname2 = 'DAS';

% Convert ASCII strings to binary vectors

x1 = reshape(dec2bin(surname1,8).'-'0',1,[]);

% bits for surname1

x2 = reshape(dec2bin(surname2,8).'-'0',1,[]);

% bits for surname2

bp = 0.001;

% bit period (1 ms)

A1 = 5;

% amplitude for bit 1

A0 = 0;

% amplitude for bit 0

f1 = 2;

% carrier freq for x1 (Hz)

f2 = 4;

% carrier freq for x2 (Hz)

% Time vector for one bit

t\_bit = linspace(0,bp,100);

% BASK Modulation of x1

m1 = [];

for i = 1:length(x1)

if x1(i) == 1

y = A1\*cos(2\*pi\*f1\*t\_bit);

else

y = A0\*cos(2\*pi\*f1\*t\_bit);

end

m1 = [m1 y];

end

% BASK Modulation of x2

m2 = [];

for i = 1:length(x2)

if x2(i) == 1

y = A1\*cos(2\*pi\*f2\*t\_bit);

else

y = A0\*cos(2\*pi\*f2\*t\_bit);

end

m2 = [m2 y];

end

% Make lengths equal by zero-padding the shorter signal

len1 = length(m1); len2 = length(m2);

if len1 < len2

m1 = [m1 zeros(1,len2 - len1)];

elseif len2 < len1

m2 = [m2 zeros(1,len1 - len2)];

end

% Composite signal by Frequency Division Multiplexing (FDM)

composite\_signal = m1 + m2;

% Add Gaussian noise (SNR = 20 dB)

SNR = 20;

noisy\_signal = awgn(composite\_signal, SNR, 'measured');

% Time vector for full signal

t\_full = linspace(0, bp\*max(length(x1),length(x2)), length(composite\_signal));

% Plot composite signal -figure(1); subplot(3,1,1);

plot(t\_full, m1, 'b');

title('BASK Modulated Signal x1 (2 Hz)'); xlabel('Time (s)'); ylabel('Amplitude'); axis([0 bp\*length(x1) -6 6]); grid on;

subplot(3,1,2);

plot(t\_full, m2, 'r');

title('BASK Modulated Signal x2 (4 Hz)'); xlabel('Time (s)'); ylabel('Amplitude'); axis([0 bp\*length(x2) -6 6]); grid on;

subplot(3,1,3); plot(t\_full, noisy\_signal, 'k'); title('Composite Signal with Noise'); xlabel('Time (s)'); ylabel('Amplitude'); axis([0 bp\*max(length(x1),length(x2)) -10 10]); grid on;

% Demultiplexing & Demodulation

fs = length(t\_bit)/bp;

% sampling frequency based on time vector

% Bandpass filter parameters

order = 6; bw = 1.5;

% bandwidth around carrier freq (Hz)

% Design Bandpass Filter for f1 = 2 Hz

[b1,a1] = butter(order, [(f1-bw)/(fs/2) (f1+bw)/(fs/2)], 'bandpass');

% Design Bandpass Filter for f2 = 4 Hz

[b2,a2] = butter(order, [(f2-bw)/(fs/2) (f2+bw)/(fs/2)], 'bandpass');

% Filter the noisy composite signal to separate signals

sep1 = filter(b1,a1,noisy\_signal); sep2 = filter(b2,a2,noisy\_signal);

% Coherent Demodulation (Multiply by carrier)

t\_demod = linspace(0, bp\*max(length(x1),length(x2)), length(sep1)); carrier1 = cos(2\*pi\*f1\*t\_demod); carrier2 = cos(2\*pi\*f2\*t\_demod);

demod1 = sep1 .\* carrier1; demod2 = sep2 .\* carrier2;

% Lowpass filter to get baseband signals

cutoff = 3;

% cutoff frequency in Hz (slightly above bit rate)

[b\_lp,a\_lp] = butter(order, cutoff/(fs/2), 'low');

baseband1 = filter(b\_lp,a\_lp,demod1); baseband2 = filter(b\_lp,a\_lp,demod2);

% Sampling and bit decision (sample at center of each bit)

samples\_per\_bit = length(t\_bit); sample\_points = round((samples\_per\_bit/2):samples\_per\_bit:length(baseband1));

rec\_bits1 = baseband1(sample\_points) > (A1/2);

% threshold at A1/2

rec\_bits2 = baseband2(sample\_points) > (A1/2);

% -------- Plot recovered signals in Unipolar NRZ --------

% Prepare unipolar NRZ signals for plotting

nrz1 = zeros(1, samples\_per\_bit\*length(rec\_bits1)); nrz2 = zeros(1, samples\_per\_bit\*length(rec\_bits2));

for i = 1:length(rec\_bits1)

if rec\_bits1(i) == 1

nrz1((i-1)\*samples\_per\_bit + (1:samples\_per\_bit)) = A1;

else

nrz1((i-1)\*samples\_per\_bit + (1:samples\_per\_bit)) = 0;

end

end

for i = 1:length(rec\_bits2)

if rec\_bits2(i) == 1

nrz2((i-1)\*samples\_per\_bit + (1:samples\_per\_bit)) = A1;

else

nrz2((i-1)\*samples\_per\_bit + (1:samples\_per\_bit)) = 0;

end

end

t\_nrz = linspace(0, bp\*length(rec\_bits1), length(nrz1));

figure(2); subplot(3,1,1); plot(t\_nrz, nrz1, 'b', 'LineWidth', 1.5); title('Recovered Signal x1 (Unipolar NRZ)');

xlabel('Time (s)'); ylabel('Voltage'); ylim([-0.5 A1+1]);

grid on;

subplot(3,1,2); plot(t\_nrz, nrz2, 'r', 'LineWidth', 1.5); title('Recovered Signal x2 (Unipolar NRZ)');

xlabel('Time (s)'); ylabel('Voltage'); ylim([-0.5 A1+1]);

grid on;

% Also plot the original bit streams for reference

subplot(3,1,3);

stairs(0:length(x1)-1, x1, 'b','LineWidth',1.5);

hold on;

stairs(0:length(x2)-1, x2, 'r','LineWidth',1.5);

hold off; title('Original Bit Streams');

xlabel('Bit index');

ylabel('Bit value');

ylim([-0.5 1.5]);

legend('x1','x2');

grid on;

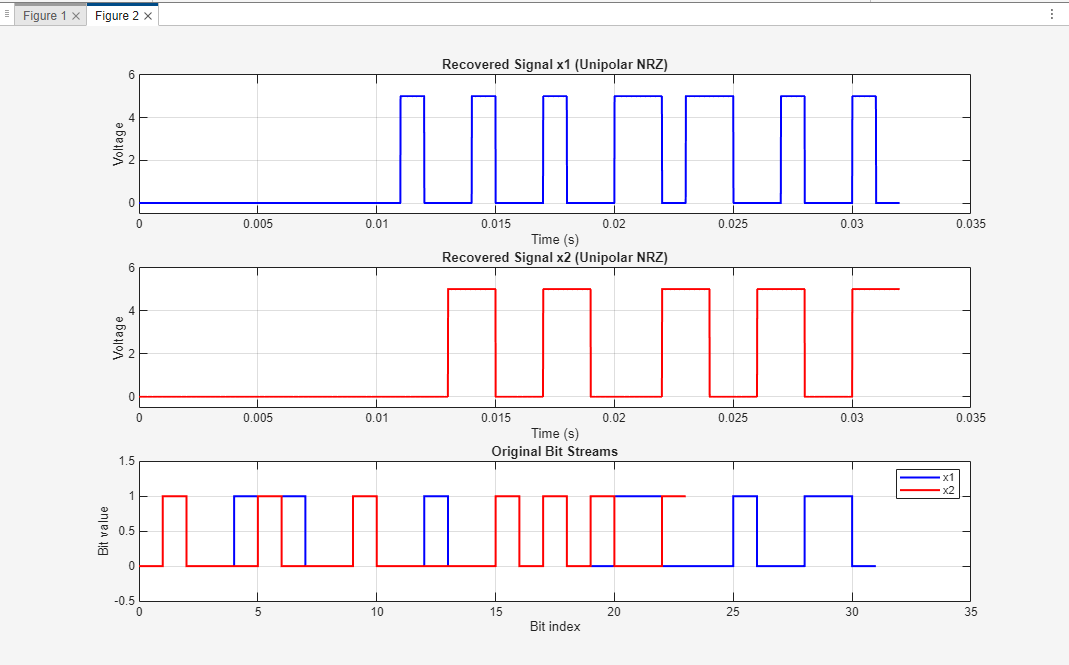


Figure 5

***Code for task six:***

clc;

clear;

x1 = [ 0 1 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0];

x2 = [0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 1];

% Convert to character strings

bin\_chars1 = reshape(char(x1 + '0'), 8, [])';

bin\_chars2 = reshape(char(x2 + '0'), 8, [])';

% Binary to decimal

ascii\_recovered1 = bin2dec(bin\_chars1)';

ascii\_recovered2 = bin2dec(bin\_chars2)';

% Decimal to characters

recovered\_surname1 = char(ascii\_recovered1); recovered\_surname2 = char(ascii\_recovered2);

% Display

disp('Recovered Surnames:');

disp(['x1 => ', recovered\_surname1]);

disp(['x2 => ', recovered\_surname2]);

% Expected

expected1 = 'NILL';

expected2 = 'DAS';

% Verification

if strcmp(recovered\_surname1, expected1) && strcmp(recovered\_surname2, expected2)

disp('Success: Surnames recovered correctly!');

else

disp(' Error: Surnames not recovered correctly');

end

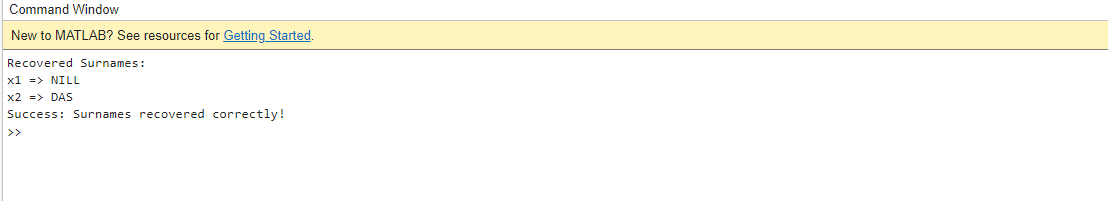


Figure 6

# Conclusion

The entire digital communication system was simulated successfully in MATLAB in this experiment. The steps included ASCII encoding of the last names, NRZ representation, BASK modulation, FDM multiplexing, addition of Gaussian noise, demultiplexing, demodulation, and ultimate ASCII decoding. The recovered digital data were verified to be similar to original last names, thereby establishing the correctness of the developed system.

The main advantage of this experiment is that it provides a very accurate, step-by-step account of how digital information is converted, transmitted, and recovered in real communication systems. It highlights the effectiveness of BASK modulation in simple digital transmission and shows how FDM allows multiple signals to be conveyed simultaneously without interference.

A recognized limitation is that as an amplitude-based BASK, it is less immune to noise compared to other modulation systems (for example, PSK or FSK). This was observed when introducing Gaussian noise, which had a minimal degrading impact on the signals, although filtering and thresholding allowed proper recovery in this case. Another practical limitation is that extremely low carrier frequencies (2 Hz and 4 Hz) were used for simplicity in the simulation, while real systems use significantly higher frequency levels.

One of the challenges in implementation was proper synchronization of bit duration and sample rate to avoid bit errors at demodulation. The selection of correct filter parameters and sample points had to be performed to ensure successful recovery.

Overall, the experiment provided hands-on experience with basic concepts of digital communication, demonstrating end-to-end data transmission chain from source coding to final verification.

# Reference

1. American International University–Bangladesh (AIUB), *Data Communication Laboratory Manual*, Department of Computer Science and Engineering, 2025.
2. Simon Haykin, *Communication Systems*, 4th Edition, Wiley, 2001.
3. Behrouz A. Forouzan, *Data Communications and Networking*, 5th Edition, McGraw-Hill, 2012.