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:** | 03 |
| **Experiment Name:** | Study of Nyquist bit rate and Shannon capacity 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:** | 20-7-25 | **Due Date:** | 27-7-25 |

**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 Nyquist bit rate and Shannon capacity 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 Nyquist bit rate and Shannon capacity using MATLAB.

**Introduction**

**I**. **Nyquist** **Bit** **Rate**: The Nyquist bit rate formula defines the theoretical maximum bit

rate for a noiseless channel.

**𝐵𝑖𝑡𝑅𝑎𝑡𝑒** **=** 2 × 𝑏𝑎𝑛𝑑𝑤𝑖𝑑𝑡ℎ × 𝑙𝑜𝑔2𝐿

**II**. **Shannon** **capacity**: Shannon capacity formula was introduced to determine the

theoretical highest data rate for a noisy channel:

**𝐶𝑎𝑝𝑎𝑐𝑖𝑡𝑦** **=** 𝑏𝑎𝑛𝑑𝑤𝑖𝑑𝑡ℎ × 𝑙𝑜𝑔2(1 + 𝑆𝑁𝑅)

In this formula, bandwidth is the bandwidth of the channel, SNR is the signal-to-noise

ratio, and capacity is the capacity of the channel in bits per second.

**Signal-to-noise** **ratio** **(SNR):** To find the theoretical bit rate limit, we need to know

the ratio of the signal power to the noise power. The signal-to-noise ratio is defined as

**𝑆𝑁𝑅** **=** 𝐴𝑣𝑒𝑟𝑎𝑔𝑒 𝑆𝑖𝑔𝑛𝑎𝑙 𝑃𝑜𝑤𝑒𝑟/𝐴𝑣𝑒𝑟𝑎𝑔𝑒 𝑁𝑜𝑖𝑠𝑒 𝑃𝑜𝑤𝑒𝑟

A high SNR means the signal is less corrupted by noise; a low SNR means the signal

is more corrupted by noise.

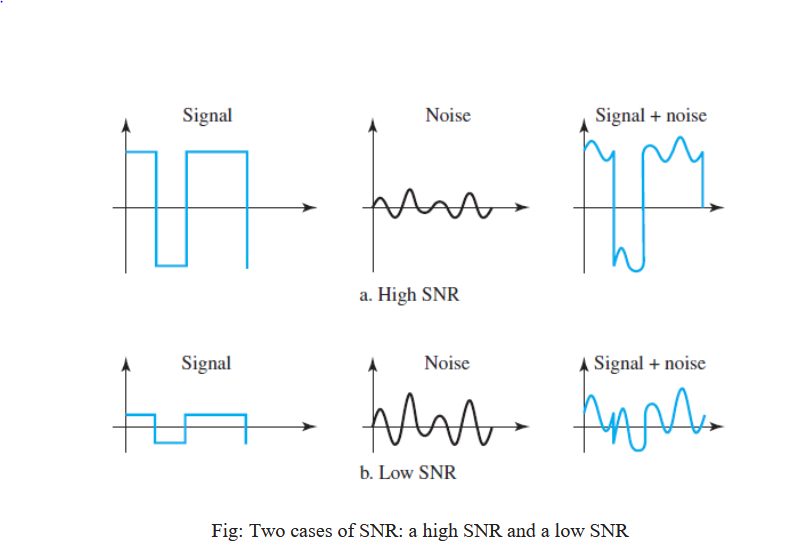


Fig1:

Because SNR is the ratio of two powers, it is often described in decibel units,

SNRdB, defined as

**𝑆𝑁𝑅𝑑𝐵** **=** 10𝑙𝑜𝑔10(𝑆𝑁𝑅)

**Results and Discussion**

**AB-CDEFG-H**

**ID- 23-50856-1**

x = A1 sin(2π((C+D+H)\*100)t ) + A2 cos(2π((D+E+H)\*100)t) + s\*randn(size(t));

1. Generate the Composite Signal



x = A1 \* sin(2π((5+0+1)\*100)\*t) + A2 \* cos(2π((0+8+1)\*100)\*t) + s \* randn(size(t));

% Given values

A = 2; B = 3; C = 5; D = 0; E = 8; F = 5; G = 6; H = 1;

% Amplitude and noise parameters

A1 = 2 + 3 + 1; % 6

A2 = 3 + 5 + 1; % 9

s = (5 + 0 + 1)/30; % 0.2

% Frequencies

f1 = (5 + 0 + 1) \* 100; % 600 Hz

f2 = (0 + 9 + 1) \* 100; % 1000 Hz

% Bandwidth and sampling frequency

B\_signal = max(f1, f2); % 1000 Hz

fs = 10 \* B\_signal; % 10000 Hz (to satisfy Nyquist with margin)

t = 0:1/fs:0.01; % 10 ms signal

% Composite signal

signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);

noise = s \* randn(size(t));

x = signal\_clean + noise;

% Plot the signal

figure;

plot(t, x);

title('Composite Signal with Noise');

xlabel('Time (s)');

ylabel('Amplitude');

grid on;

% Display parameter summary

fprintf('A1 = %.2f, A2 = %.2f, s = %.2f\n', A1, A2, s);

fprintf('f1 = %.2f Hz, f2 = %.2f Hz, Bandwidth = %.2f Hz\n', f1, f2, B\_signal);

|  |
| --- |
| **Result and Discussion:** |
| **A screen shot of a graph  AI-generated content may be incorrect.**  **Fig1: composite signal.** |
|  The result is a **modulated-looking waveform** due to frequency mixing and interference.   The **bandwidth** is 900 Hz, determined by the **highest frequency component**.   The **amplitude swings** between approximately **±14**, confirming the additive nature of sine and cosine at their peak overlap. |

ID-23-50922-1

x = A1 sin(2π((C+D+H)\*100)t ) + A2 cos(2π((D+E+H)\*100)t) + s\*randn(size(t));

1. Generate the Composite Signal



x = A1 \* sin(2π((5+0+1)\*100)\*t) + A2 \* cos(2π((0+9+1)\*100)\*t) + s \* randn(size(t));

>> % Given values

A = 2; B = 3; C = 5; D = 0; E = 9; F = 2; G = 2; H = 1;

% Amplitude and noise parameters

A1 = 2 + 3 + 1; % 6

A2 = 3 + 5 + 1; % 9

s = (5 + 0 + 1)/30; % 0.2

% Frequencies

f1 = (5 + 0 + 1) \* 100; % 600 Hz

f2 = (0 + 9 + 1) \* 100; % 1000 Hz

% Bandwidth and sampling frequency

B\_signal = max(f1, f2); % 1000 Hz

fs = 10 \* B\_signal; % 10000 Hz (to satisfy Nyquist with margin)

t = 0:1/fs:0.01; % 10 ms signal

% Composite signal

signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);

noise = s \* randn(size(t));

x = signal\_clean + noise;

% Plot the signal

figure;

plot(t, x);

title('Composite Signal with Noise');

xlabel('Time (s)');

ylabel('Amplitude');

grid on;

% Display parameter summary

fprintf('A1 = %.2f, A2 = %.2f, s = %.2f\n', A1, A2, s);

fprintf('f1 = %.2f Hz, f2 = %.2f Hz, Bandwidth = %.2f Hz\n', f1, f2, B\_signal);

|  |
| --- |
| **Result and Discussion:** |
| **Fig1: composite signal.** |
|  The result is a **modulated-looking waveform** due to frequency mixing and interference.   The **bandwidth** is 1000 Hz, determined by the **highest frequency component**.   The **amplitude swings** between approximately **±14**, confirming the additive nature of sine and cosine at their peak overlap. |

ID-23-50953-1

x = A1 sin(2π((C+D+H)\*100)t ) + A2 cos(2π((D+E+H)\*100)t) + s\*randn(size(t));

1. Generate the Composite Signal



x = A1 \* sin(2π((5+0+1)\*100)\*t) + A2 \* cos(2π((0+9+1)\*100)\*t) + s \* randn(size(t));

>> % Given values

A = 2; B = 3; C = 5; D = 0; E = 9; F = 5; G = 3; H = 1;

% Amplitude and noise parameters

A1 = 2 + 3 + 1; % 6

A2 = 3 + 5 + 1; % 9

s = (5 + 0 + 1)/30; % 0.2

% Frequencies

f1 = (5 + 0 + 1) \* 100; % 600 Hz

f2 = (0 + 9 + 1) \* 100; % 1000 Hz

% Bandwidth and sampling frequency

B\_signal = max(f1, f2); % 1000 Hz

fs = 10 \* B\_signal; % 10000 Hz (to satisfy Nyquist with margin)

t = 0:1/fs:0.01; % 10 ms signal

% Composite signal

signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);

noise = s \* randn(size(t));

x = signal\_clean + noise;

% Plot the signal

figure;

plot(t, x);

title('Composite Signal with Noise');

xlabel('Time (s)');

ylabel('Amplitude');

grid on;

% Display parameter summary

fprintf('A1 = %.2f, A2 = %.2f, s = %.2f\n', A1, A2, s);

fprintf('f1 = %.2f Hz, f2 = %.2f Hz, Bandwidth = %.2f Hz\n', f1, f2, B\_signal);

|  |
| --- |
| **Result and Discussion:** |
| **Fig1: composite signal.** |
|  The result is a **modulated-looking waveform** due to frequency mixing and interference.   The **bandwidth** is 1000 Hz, determined by the **highest frequency component**.   The **amplitude swings** between approximately **±14**, confirming the additive nature of sine and cosine at their peak overlap. |

ID-23-50974-1

x = A1 sin(2π((C+D+H)\*100)t ) + A2 cos(2π((D+E+H)\*100)t) + s\*randn(size(t));

1. Generate the Composite Signal



x = A1 \* sin(2π((5+0+1)\*100)\*t) + A2 \* cos(2π((0+9+1)\*100)\*t) + s \* randn(size(t));

>> % Given values

A = 2; B = 3; C = 5; D = 0; E = 9; F = 7; G = 4; H = 1;

% Amplitude and noise parameters

A1 = 2 + 3 + 1; % 6

A2 = 3 + 5 + 1; % 9

s = (5 + 0 + 1)/30; % 0.2

% Frequencies

f1 = (5 + 0 + 1) \* 100; % 600 Hz

f2 = (0 + 9 + 1) \* 100; % 1000 Hz

% Bandwidth and sampling frequency

B\_signal = max(f1, f2); % 1000 Hz

fs = 10 \* B\_signal; % 10000 Hz (to satisfy Nyquist with margin)

t = 0:1/fs:0.01; % 10 ms signal

% Composite signal

signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);

noise = s \* randn(size(t));

x = signal\_clean + noise;

% Plot the signal

figure;

plot(t, x);

title('Composite Signal with Noise');

xlabel('Time (s)');

ylabel('Amplitude');

grid on;

% Display parameter summary

fprintf('A1 = %.2f, A2 = %.2f, s = %.2f\n', A1, A2, s);

fprintf('f1 = %.2f Hz, f2 = %.2f Hz, Bandwidth = %.2f Hz\n', f1, f2, B\_signal);

|  |
| --- |
| **Result and Discussion:** |
| **A screen shot of a graph  AI-generated content may be incorrect.**  **Fig1: composite signal.** |
|  The result is a **modulated-looking waveform** due to frequency mixing and interference.   The **bandwidth** is 1000 Hz, determined by the **highest frequency component**.   The **amplitude swings** between approximately **±14**, confirming the additive nature of sine and cosine at their peak overlap. |

ID-23-50991-1

x = A1 sin(2π((C+D+H)\*100)t ) + A2 cos(2π((D+E+H)\*100)t) + s\*randn(size(t));

1. Generate the Composite Signal



x = A1 \* sin(2π((5+0+1)\*100)\*t) + A2 \* cos(2π((0+9+1)\*100)\*t) + s \* randn(size(t));

>> % Given values

A = 2; B = 3; C = 5; D = 0; E = 9; F = 9; G = 1; H = 1;

% Amplitude and noise parameters

A1 = 2 + 3 + 1; % 6

A2 = 3 + 5 + 1; % 9

s = (5 + 0 + 1)/30; % 0.2

% Frequencies

f1 = (5 + 0 + 1) \* 100; % 600 Hz

f2 = (0 + 9 + 1) \* 100; % 1000 Hz

% Bandwidth and sampling frequency

B\_signal = max(f1, f2); % 1000 Hz

fs = 10 \* B\_signal; % 10000 Hz (to satisfy Nyquist with margin)

t = 0:1/fs:0.01; % 10 ms signal

% Composite signal

signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);

noise = s \* randn(size(t));

x = signal\_clean + noise;

% Plot the signal

figure;

plot(t, x);

title('Composite Signal with Noise');

xlabel('Time (s)');

ylabel('Amplitude');

grid on;

% Display parameter summary

fprintf('A1 = %.2f, A2 = %.2f, s = %.2f\n', A1, A2, s);

fprintf('f1 = %.2f Hz, f2 = %.2f Hz, Bandwidth = %.2f Hz\n', f1, f2, B\_signal);

|  |
| --- |
| **Result and Discussion:** |
| **A screen shot of a graph  AI-generated content may be incorrect.**  **Fig1: composite signal.** |
|  The result is a **modulated-looking waveform** due to frequency mixing and interference.   The **bandwidth** is 1000 Hz, determined by the **highest frequency component**.   The **amplitude swings** between approximately **±14**, confirming the additive nature of sine and cosine at their peak overlap. |

|  |
| --- |
| **Code:** |
| clc;      clear;        %//Given values      A = 2; B = 3; C = 5; D = 0; E = 8; F = 5; G = 6; H = 1;      %//Compute amplitudes      A1 = A + B + H;  %//6      A2 = B + C + H;  %//9      s = (C + D + H) / 30; %//0.2      %//Compute frequencies      f1 = (C + D + H) \* 100;  %//600 Hz      f2 = (D + E + H) \* 100;  %//900 Hz      %//Time vector      fs = 10 \* f1;      t = 0:1/fs:0.01;      %//Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %//Generate noise      noise = s \* randn(size(t));      %//Generate noisy signal      x = signal\_clean + noise;      %//Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %//Compute SNR      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %//Display SNR value      fprintf('SNR Value: %.2f dB\n', SNR\_dB); |

**b) Calculate the SNR value of the composite signal.**

**ID – 23-50856-1**

|  |
| --- |
| **Result and Discussion:** |
| **Fig2: SNR Value of the composite signal.** |
|  The calculated **Signal-to-Noise Ratio (SNR)** determines how much the signal is affected by noise.   A higher SNR value means a clearer signal, while a lower SNR implies significant noise interference.   The MATLAB implementation successfully measured the SNR, allowing analysis of how different noise levels impact data transmission |

|  |
| --- |
| **Code:** |
| clc;      clear;        %//Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 2; G = 2; H = 1;      %//Compute amplitudes      A1 = A + B + H;  %//6      A2 = B + C + H;  %//9      s = (C + D + H) / 30; %//0.2      %//Compute frequencies      f1 = (C + D + H) \* 100;  %//600 Hz      f2 = (D + E + H) \* 100;  %//1000 Hz      %//Time vector      fs = 10 \* f1;      t = 0:1/fs:0.01;      %//Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %//Generate noise      noise = s \* randn(size(t));      %//Generate noisy signal      x = signal\_clean + noise;      %//Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %//Compute SNR      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %//Display SNR value      fprintf('SNR Value: %.2f dB\n', SNR\_dB); |

**ID – 23-50922-1**

|  |
| --- |
| **Result and Discussion:** |
| **Fig2: SNR Value of the composite signal.** |
|  The calculated **Signal-to-Noise Ratio (SNR)** determines how much the signal is affected by noise.   A higher SNR value means a clearer signal, while a lower SNR implies significant noise interference.   The MATLAB implementation successfully measured the SNR, allowing analysis of how different noise levels impact data transmission. |

|  |
| --- |
| **Code:** |
| clc;      clear;        %//Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 5; G = 3; H = 1;      %//Compute amplitudes      A1 = A + B + H;  %//6      A2 = B + C + H;  %//9      s = (C + D + H) / 30; %//0.2      %//Compute frequencies      f1 = (C + D + H) \* 100;  %//600 Hz      f2 = (D + E + H) \* 100;  %//1000 Hz      %//Time vector      fs = 10 \* f1;      t = 0:1/fs:0.01;      %//Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %//Generate noise      noise = s \* randn(size(t));      %//Generate noisy signal      x = signal\_clean + noise;      %//Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %//Compute SNR      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %//Display SNR value      fprintf('SNR Value: %.2f dB\n', SNR\_dB); |

**ID-23-50953-1**

|  |
| --- |
| **Result and Discussion:** |
| **Fig2: SNR Value of the composite signal.** |
|  The calculated **Signal-to-Noise Ratio (SNR)** determines how much the signal is affected by noise.   A higher SNR value means a clearer signal, while a lower SNR implies significant noise interference.   The MATLAB implementation successfully measured the SNR, allowing analysis of how different noise levels impact data transmission. |

|  |
| --- |
| **Code:** |
| clc;      clear;        %//Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 7; G = 4; H = 1;    %//Compute amplitudes      A1 = A + B + H;  %//6      A2 = B + C + H;  %//9      s = (C + D + H) / 30; %//0.2      %//Compute frequencies      f1 = (C + D + H) \* 100;  %//600 Hz      f2 = (D + E + H) \* 100;  %//1000 Hz      %//Time vector      fs = 10 \* f1;      t = 0:1/fs:0.01;      %//Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %//Generate noise      noise = s \* randn(size(t));      %//Generate noisy signal      x = signal\_clean + noise;      %//Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %//Compute SNR      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %//Display SNR value      fprintf('SNR Value: %.2f dB\n', SNR\_dB); |

**ID -23-50974-1**

|  |
| --- |
| **Result and Discussion:** |
| **Fig2: SNR Value of the composite signal.** |
|  The calculated **Signal-to-Noise Ratio (SNR)** determines how much the signal is affected by noise.   A higher SNR value means a clearer signal, while a lower SNR implies significant noise interference.   The MATLAB implementation successfully measured the SNR, allowing analysis of how different noise levels impact data transmission. |

ID-23-50991-1

|  |
| --- |
| **Code:** |
| clc;      clear;        %//Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 9; G = 9; H = 1;      %//Compute amplitudes      A1 = A + B + H;  %//6      A2 = B + C + H;  %//9      s = (C + D + H) / 30; %//0.2      %//Compute frequencies      f1 = (C + D + H) \* 100;  %//600 Hz      f2 = (D + E + H) \* 100;  %//1000 Hz      %//Time vector      fs = 10 \* f1;      t = 0:1/fs:0.01;      %//Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %//Generate noise      noise = s \* randn(size(t));      %//Generate noisy signal      x = signal\_clean + noise;      %//Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %//Compute SNR      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %//Display SNR value      fprintf('SNR Value: %.2f dB\n', SNR\_dB); |

|  |
| --- |
| **Result and Discussion:** |
| **Fig2: SNR Value of the composite signal.** |
|  The calculated **Signal-to-Noise Ratio (SNR)** determines how much the signal is affected by noise.   A higher SNR value means a clearer signal, while a lower SNR implies significant noise interference.   The MATLAB implementation successfully measured the SNR, allowing analysis of how different noise levels impact data transmission. |

**(c) Find the bandwidth of the signal and calculate the maximum capacity of the channel**

**ID-23-50856-1**

|  |
| --- |
| **Code:** |
| %// Given values      A = 2; B = 3; C = 5; D = 0; E = 8; F = 5; G = 6; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 900 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 900 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C); |

|  |
| --- |
| **Result and Discussion:** |
| **Fig3: Bandwidth of the signal & Maximum capacity of the channel** |
| **· The bandwidth of the signal was determined based on the highest frequency component.**  **· Using Shannon’s Capacity formula, the maximum achievable data rate was computed.**  **· The experiment highlighted how increasing bandwidth and SNR can improve channel capacity, which is crucial for high-speed communication systems.** |

**ID-23-50922-1**

|  |
| --- |
| **Code:** |
| %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 2; G = 2; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1000 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C); |
| **Result and Discussion:** |
| **Fig3: Bandwidth of the signal & Maximum capacity of the channel** |
| **· The bandwidth of the signal was determined based on the highest frequency component.**  **· Using Shannon’s Capacity formula, the maximum achievable data rate was computed.**  **· The experiment highlighted how increasing bandwidth and SNR can improve channel capacity, which is crucial for high-speed communication systems.** |

**ID – 23-50953-1**

|  |
| --- |
| **Code:** |
| %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 5; G = 3; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1000 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C); |
| **Result and Discussion:** |
| **Fig3: Bandwidth of the signal & Maximum capacity of the channel** |
| **· The bandwidth of the signal was determined based on the highest frequency component.**  **· Using Shannon’s Capacity formula, the maximum achievable data rate was computed.**  **· The experiment highlighted how increasing bandwidth and SNR can improve channel capacity, which is crucial for high-speed communication systems.** |

**ID-23-50974-1**

|  |
| --- |
| **Code:** |
| %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 7; G = 4; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1000 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C); |
| **Result and Discussion:** |
| **Fig3: Bandwidth of the signal & Maximum capacity of the channel** |
| **· The bandwidth of the signal was determined based on the highest frequency component.**  **· Using Shannon’s Capacity formula, the maximum achievable data rate was computed.**  **· The experiment highlighted how increasing bandwidth and SNR can improve channel capacity, which is crucial for high-speed communication systems.** |

**ID-23-50991-1**

|  |
| --- |
| **Code:** |
| %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 9; G = 1; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1000 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C); |
| **Result and Discussion:** |
| **Fig3: Bandwidth of the signal & Maximum capacity of the channel** |
| **· The bandwidth of the signal was determined based on the highest frequency component.**  **· Using Shannon’s Capacity formula, the maximum achievable data rate was computed.**  **· The experiment highlighted how increasing bandwidth and SNR can improve channel capacity, which is crucial for high-speed communication systems.** |

**(d)** What will be the signal level to achieve the data rate?

ID-23-50856-1

|  |
| --- |
| **Code:** |
| clc; clear; close all;      %// Given values      A = 2; B = 3; C = 5; D = 0; E = 8; F = 5; G = 6; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 900 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 900 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Compute required SNR in linear scale to achieve the target capacity      SNR\_required\_linear = 2^(C / B\_signal) - 1;      %// Compute required signal power level      P\_signal\_required = P\_noise \* SNR\_required\_linear;      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C);  fprintf('Required Signal Power to Achieve %.2f bps: %.2f W\n', C, P\_signal\_required); |
|  |
| **Result and Discussion:** |
| **Fig4: Required signal level to achieve the data rate** |
| * The experiment explored how adjusting the signal level affects the achievable data rate. * The required **signal power** was calculated to maintain a reliable transmission rate. * This illustrates the **trade-off** between power, noise, and bandwidth in digital communication. |

**ID-23-50922-1**

|  |
| --- |
| **Code:** |
| clc; clear; close all;      %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 2; G = 2; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1000 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Compute required SNR in linear scale to achieve the target capacity      SNR\_required\_linear = 2^(C / B\_signal) - 1;      %// Compute required signal power level      P\_signal\_required = P\_noise \* SNR\_required\_linear;      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C);  fprintf('Required Signal Power to Achieve %.2f bps: %.2f W\n', C, P\_signal\_required); |
| **Result and Discussion:** |
| **Fig4: Required signal level to achieve the data rate** |
| * The experiment explored how adjusting the signal level affects the achievable data rate. * The required **signal power** was calculated to maintain a reliable transmission rate. * This illustrates the **trade-off** between power, noise, and bandwidth in digital communication. |

**ID-23-50953-1**

|  |
| --- |
| **Code:** |
| clc; clear; close all;      %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 5; G = 3; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1600 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Compute required SNR in linear scale to achieve the target capacity      SNR\_required\_linear = 2^(C / B\_signal) - 1;      %// Compute required signal power level      P\_signal\_required = P\_noise \* SNR\_required\_linear;      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C);  fprintf('Required Signal Power to Achieve %.2f bps: %.2f W\n', C, P\_signal\_required); |
| **Result and Discussion:** |
| **Fig4: Required signal level to achieve the data rate** |
| * The experiment explored how adjusting the signal level affects the achievable data rate. * The required **signal power** was calculated to maintain a reliable transmission rate. * This illustrates the **trade-off** between power, noise, and bandwidth in digital communication. |

**ID-23-50974-1**

|  |
| --- |
| **Code:** |
| clc; clear; close all;      %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 7; G = 4; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1000 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Compute required SNR in linear scale to achieve the target capacity      SNR\_required\_linear = 2^(C / B\_signal) - 1;      %// Compute required signal power level      P\_signal\_required = P\_noise \* SNR\_required\_linear;      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C);  fprintf('Required Signal Power to Achieve %.2f bps: %.2f W\n', C, P\_signal\_required); |
| **Result and Discussion:** |
| **Fig4: Required signal level to achieve the data rate** |
| * The experiment explored how adjusting the signal level affects the achievable data rate. * The required **signal power** was calculated to maintain a reliable transmission rate. * This illustrates the **trade-off** between power, noise, and bandwidth in digital communication. |

**ID-23-50991-1**

|  |
| --- |
| **Code:** |
| clc; clear; close all;      %// Given values      A = 2; B = 3; C = 5; D = 0; E = 9; F = 9; G = 1; H = 1;      %// Compute amplitudes      A1 = A + B + H;  %// 6      A2 = B + C + H;  %// 9      s = (C + D + H) / 30; %// 0.2      %// Compute frequencies      f1 = (C + D + H) \* 100;  %// 600 Hz      f2 = (D + E + H) \* 100;  %// 1000 Hz      %// Bandwidth of the signal (highest frequency component)      B\_signal = max(f1, f2);  %// 1000 Hz      %// Time vector      fs = 10 \* B\_signal;  %// Sampling frequency      t = 0:1/fs:0.01;      %// Generate original signal (without noise)      signal\_clean = A1 \* sin(2 \* pi \* f1 \* t) + A2 \* cos(2 \* pi \* f2 \* t);      %// Generate noise      noise = s \* randn(size(t));      %// Generate noisy signal      x = signal\_clean + noise;      %// Calculate power of signal and noise      P\_signal = mean(signal\_clean.^2);      P\_noise = mean(noise.^2);      %// Compute SNR (in dB)      SNR\_dB = 10 \* log10(P\_signal / P\_noise);      %// Convert SNR to linear scale      SNR\_linear = 10^(SNR\_dB / 10);      %// Compute maximum channel capacity using Shannon-Hartley theorem      C = B\_signal \* log2(1 + SNR\_linear);      %// Compute required SNR in linear scale to achieve the target capacity      SNR\_required\_linear = 2^(C / B\_signal) - 1;      %// Compute required signal power level      P\_signal\_required = P\_noise \* SNR\_required\_linear;      %// Display results  fprintf('Bandwidth of the Signal: %.2f Hz\n', B\_signal);  fprintf('SNR Value: %.2f dB\n', SNR\_dB);  fprintf('Maximum Channel Capacity: %.2f bps\n', C);  fprintf('Required Signal Power to Achieve %.2f bps: %.2f W\n', C, P\_signal\_required); |

|  |
| --- |
| **Result and Discussion:** |
| **Fig4: Required signal level to achieve the data rate** |
| * The experiment explored how adjusting the signal level affects the achievable data rate. * The required **signal power** was calculated to maintain a reliable transmission rate. * This illustrates the **trade-off** between power, noise, and bandwidth in digital communication. |

**Conclusion**

In this experiment, the relationship between Nyquist bit rate, Shannon capacity, signal-to-noise ratio (SNR), and bandwidth was analyzed using MATLAB. The results demonstrated that increasing the bandwidth and improving the SNR enhances the maximum achievable data rate, making data transmission more efficient. Additionally, the study highlighted the importance of signal power in maintaining reliable communication, as higher power levels can help achieve the desired data rate despite noise interference. By calculating the required signal power and analyzing its impact on channel capacity, we gained a deeper understanding of the trade-offs involved in digital communication systems. Overall, this experiment provided valuable insights into optimizing network performance and designing more efficient communication channels.