NAMAL UNIVERSITY MIANWALI
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 345 (L) – Digital Signal Processing (Lab)
LAB # 07
REPORT

Title :
Digital Processing of Continuous Signal (Sampling) in MATLAB

Name	Fahim Ur Rehman Shah
Roll No	NIM-BSEE-2021-24
Intructor	Zulaikha Kiran
Lab Engineer	Engr. Faizan Ahmad
Date Performed	17-April-2024
Marks	

Introduction

The purpose of this lab is to perform sampling, discrete time processing and up-sampling and down-sampling using MATLAB.

Course Learning Outcomes

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713

CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

Equipment

- Software
  - MATLAB

Instructions

1. This is an individual lab. You will perform the tasks individually and submit a report.
2. Some of these tasks are for practice purposes only while others (marked as 'Exercise') have to be answered in the report.
3. When asked to display an image/ graph in the exercise, either save it as jpeg or take a screenshot, in order to insert it in the report.
4. The report should be submitted on the given template, including:
   a. Code (copy and pasted, NOT a screenshot)
   b. Output values (from command window, can be a screenshot)
   c. Output figure/graph (as instructed in 3)
   d. Explanation where required
5. The report should be properly formatted, with easy to read code and easy to see figures.
6. Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, both (or all) students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken.

Background

Sampling of Continuous Signals

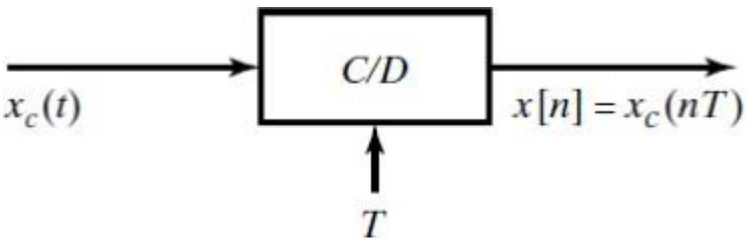
Sampling of continuous time signals is the process of converting a continuous time signal into a discrete time signal by taking samples at regular intervals. The result is a sequence of discrete values that can be stored in a digital format (discrete time signal) and processed using digital signal processing techniques.
The sampling process involves taking instantaneous snapshots of the continuous time signal at specific time intervals called the sampling period or sampling interval T.
T = sampling period; Fs = 1/ T = sampling frequency (samples/sec)
$\Omega_s = 2\pi/T$ = sampling frequency (radians/sec)

Nyquist-Shannon Sampling Theorem

The minimum sampling rate required to avoid aliasing and accurately reconstruct the original signal is called the Nyquist rate, which is given by the formula:

$$\Omega_s >= 2^* \Omega_{max}$$

Where Ω_s is the sampling frequency and Ω_{max} is the highest frequency component of the signal. If this condition does not meet, we can't reconstruct the original signal.



Practically sampling is implemented using Analog to Digital (A/D) converter, which also includes quantization step.

- From above figure, the continuous signal $x_c(t)$ is passed from the converter at sampling time T to generate discrete time signal x(n).
- After this step, we perform DTFT, and create multiple copies and perform some discrete time processing (filtering, modulation, etc.).
- After processing, we pass our processed signal from the Digital to Analog (D/A) converter to reconstruct the original signal.

Sampling Rate Conversion

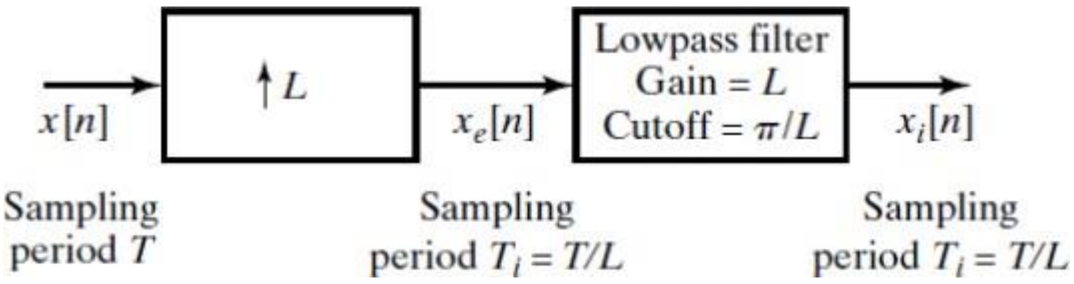
Sampling rate conversion is the process of changing the sampling rate of a digital signal from one value to another. This process can be necessary in various situations, such as when a signal needs to be transmitted or stored at a different sampling rate than its original rate, or when a signal needs to be synchronized with another signal at a different sampling rate.
There are two main methods of sampling rate conversion: up sampling/interpolation and down sampling/decimation.

Up Sampling/Interpolation

Up sampling is the process of increasing the sampling rate of a signal by inserting new samples between the existing samples. The below figure shows that the x(n) signal is up sampled/interpolated by factor of L.
Here are the steps for interpolation.

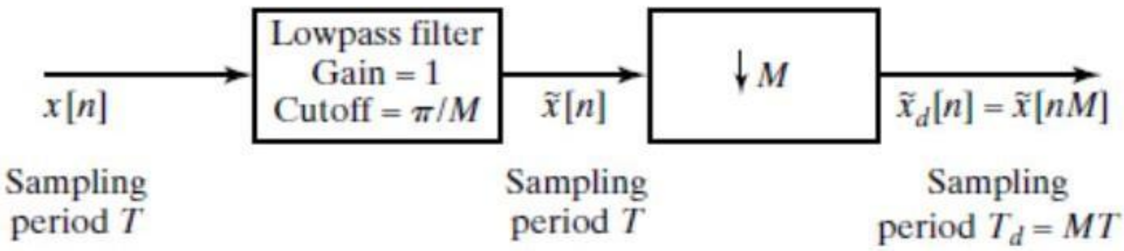
- Define the original signal which need to be up sampled.

- Define the up sampling factor and apply on original signal.
- Apply a low pass filter.
- Interpolate the filtered signal.



Down Sampling/Decimation

The down sampling is the process of reducing the sampling rate of a signal by discarding some of the samples. The below figure shows that the x(n) signal is down sampled/decimated by factor of M.



Here are the steps for decimation.

- Define the original signal which needs to be up sampled.
- Apply a low pass filter.
- Decimate the filtered signal by the decimation factor.

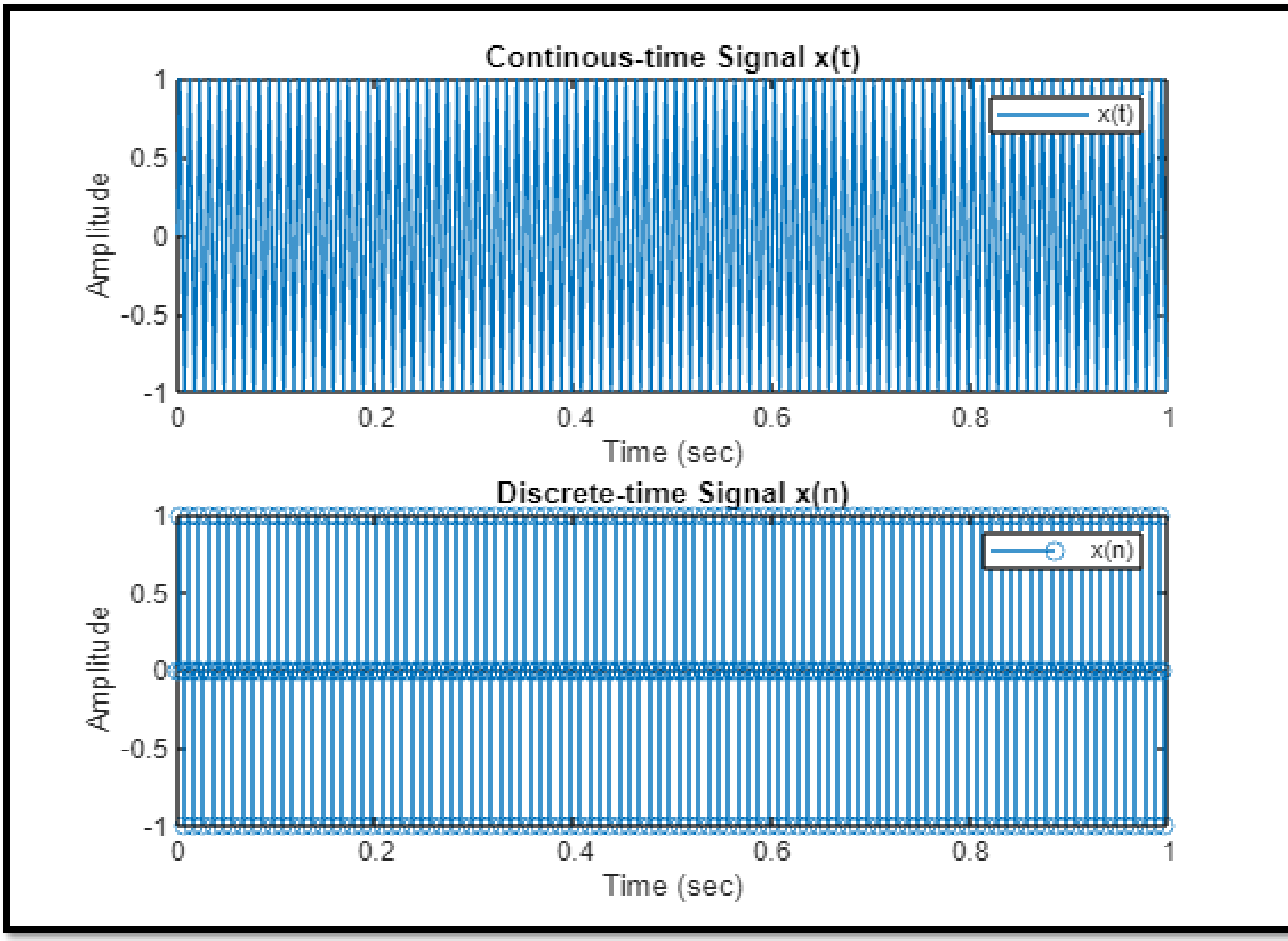
Exercise 1

The signal was sampled with sampling period T = 1/400 secs to obtain a discrete time signal x(n) .

$$x_c(t) = \sin(2\pi(100)t)$$

What will be the resulting sequence x(n)? Plot it with proper labels.

```
% Sampling period
T = 1/400;

% Sampling frequency
fs = 1/T;
% Time array
t = 0:T:1-T;
% Discrete-time signal
x = sin(2*pi*100*t);

subplot(2,1,1)
% Plot the discrete-time signal
plot(t, x, 'DisplayName', 'x(t)');
xlabel('Time (sec)');
ylabel('Amplitude');
title('Continous-time Signal x(t)');
legend;

% Plot the discrete-time signal
subplot(2,1,2)
stem(t, x, 'DisplayName', 'x(n)');
xlabel('Time (sec)');
ylabel('Amplitude');
title('Discrete-time Signal x(n)');
legend;
```

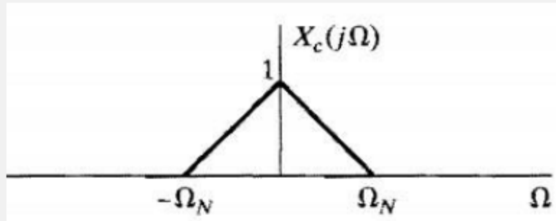


Explanation:

In this code, I set the sampling period T and calculate the sampling frequency fs. Then, I create a time array from 0 to just before 1 with intervals of T. Using this time array, I generate a discrete-time signal x which represents a sine wave with a frequency of 100 Hz. I plot this signal in two subplots: in the first subplot, I use a continuous line plot to show the signal over time t, libelling the axes and giving it a title; in the second subplot, I use a stem plot to display the discrete values of the signal x at each time point t, again labelling the axes and giving it a title.

Exercise 2

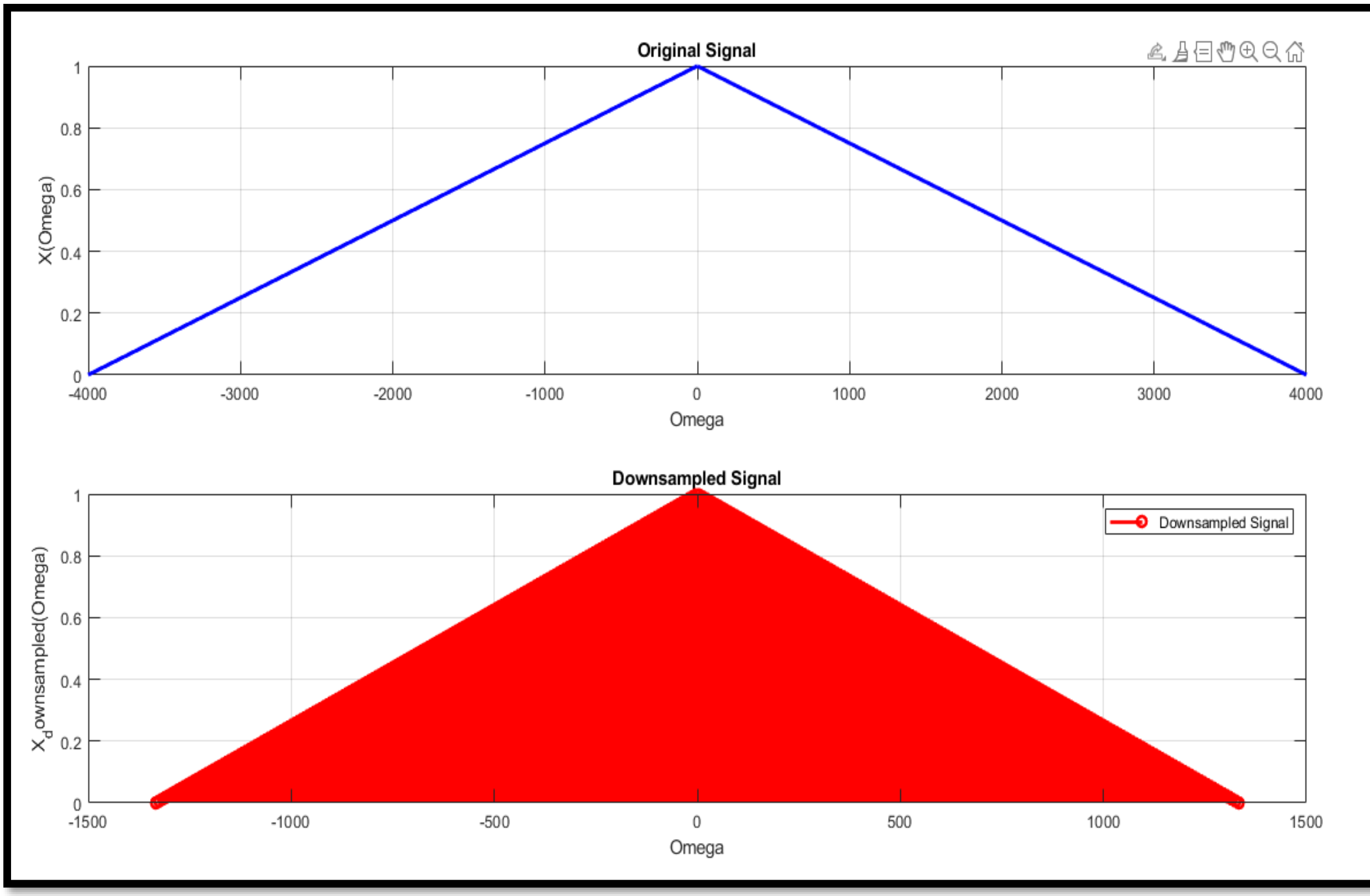
Perform down sampling ($M=3$) on given signal below. Where the value of maximum frequency $\Omega_N = 4000$Hz.



What will be the value of sampling period? You can use built-in functions (downsample or decimate or resample) or your own function to perform downsampling. Plot the signal before and after the down sampling.

```
% Given parameters
Omega_N = 4000;  % Maximum frequency in Hz
M = 3;           % Downsampling factor

% Calculate sampling period
Ts = 1 / Omega_N;

% Define the range of Omega
Omega = linspace(-Omega_N, Omega_N, 1000);

% Define the original signal X(Ω) as a triangular pulse
X_Omega = zeros(size(Omega));
X_Omega(Omega >= -Omega_N & Omega <= 0) = (Omega(Omega >= -Omega_N & Omega <= 0) +
Omega_N) / Omega_N;
X_Omega(Omega > 0 & Omega <= Omega_N) = (Omega_N - Omega(Omega > 0 & Omega <= Omega_N)) /
Omega_N;

% Perform downsampling
X_downsampled = downsample(X_Omega, M);
Omega_downsampled = downsample(Omega, M)/3;

% Plot the original and downsampled signals
figure;
subplot(2, 1, 1);
plot(Omega, X_Omega, 'b', 'LineWidth', 2);
title('Original Signal');
xlabel('Omega');
ylabel('X(Omega)');
grid on;
subplot(2, 1, 2);
stem(Omega_downsampled, X_downsampled, 'r', 'LineWidth', 2);
```

```
title('Downsampled Signal');
xlabel('Omega');
ylabel('X_downsampled(Omega)');
grid on;

legend('Downsampled Signal');
```



Explanation:

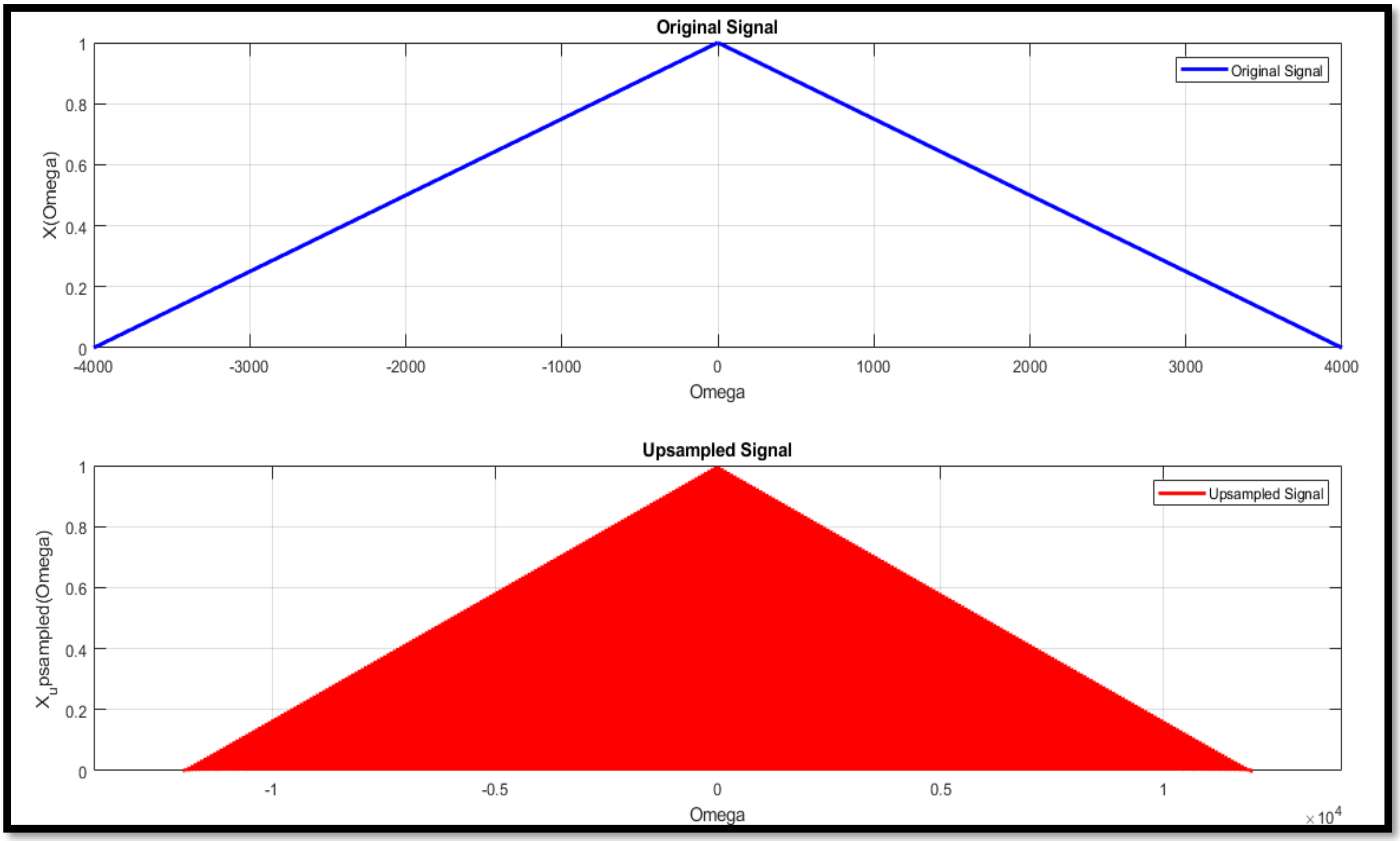
In this code, I start with given parameters: Ω N, representing the maximum frequency, and M, the downsampling factor. Then, I calculate the sampling period T s. I def ne a range of frequencies Ω from negative Ωn to positive Ωn Next, I create the original signal (Ω) X(Ω) as a triangular pulse, with values based on the frequency range. Af er that, I downsample the original signal by a factor of M, and adjust the frequency range accordingly. Finally, I plot both the original and downsampled signals on separate subplots, labeling the axes, giving titles, and adding a legend to the downsampling plot.

Exercise 3

Perform up sampling (L = 2) of above given signal in Exercise 2 You can use built-in functions (upsample or interp1 or resample) or your own function to perform up sampling. Plot the signal before and after the up sampling.

```
% Given parameters
Omega_N = 4000;  % Maximum frequency in Hz
M = 3;           % Downsampling factor
L = 2;           % Upsampling factor

% Calculate sampling period
Ts = 1 / Omega_N;

% Define the original signal X(Ω) as a triangular pulse (same as Exercise 2)
Omega = linspace(-Omega_N, Omega_N, 1000);
X_Omega = zeros(size(Omega));
X_Omega(Omega >= -Omega_N & Omega <= 0) = (Omega(Omega >= -Omega_N & Omega <= 0) +
Omega_N) / Omega_N;
X_Omega(Omega > 0 & Omega <= Omega_N) = (Omega_N - Omega(Omega > 0 & Omega <= Omega_N)) /
Omega_N;

% Perform downsampling (same as Exercise 2)
X_downsampled = X_Omega(1:M:end);
Omega_downsampled = Omega(1:M:end);

% Perform upsampling by inserting zeros
upsampled_length = length(X_downsampled) * L;
X_upsampled = zeros(1, upsampled_length);
X_upsampled(1:L:end) = X_downsampled*3;

% Upsample the Omega array accordingly
Omega_upsampled = linspace(-Omega_N, Omega_N, upsampled_length)*3;

% Plot the original and upsampled signals
figure;
subplot(2, 1, 1);
plot(Omega, X_Omega, 'b', 'LineWidth', 2);
title('Original Signal');
xlabel('Omega');
ylabel('X(Omega)');
grid on;
legend('Original Signal');
```

```
subplot(2, 1, 2);
plot(Omega_upsampled, X_upsampled, 'r', 'LineWidth', 2);
xlim([-14000 14000])
title('Upsampled Signal');
xlabel('Omega');
ylabel('X_upsampled(Omega)');
grid on;
legend('Upsampled Signal');
```



Explanation:

In this code, I start with given parameters: Ω N, representing the maximum frequency, M, the downsampling factor, and L, the upsampling factor. I calculate the sampling period T s. Then, I define the original signal (Ω) X(Ω) as a triangular pulse, similar to Exercise 2. After that, I perform downsampling of the original signal and adjust the frequency range accordingly. Next, I upsample the downsampled signal by inserting zeros and extend the frequency range accordingly. Finally, I plot both the original and upsampled signals on separate subplots, labeling the axes, giving titles, and adding legends to each plot. Additionally, I set the x-axis limit for the upsampled signal plot to ensure visibility.

Conclusion:

In our Digital Signal Processing (DSP) lab tasks, we focused on fundamental concepts such as sampling, downsampling, and upsampling. Through the provided codes, we gained practical insights into these processes. Initially, we explored sampling and discrete signal representation, visualizing the signal both in its continuous and discrete forms. Subsequently, we delved into downsampling, reducing the sample rate of a signal by a specified factor. This process involved discarding samples at regular intervals, effectively decreasing the signal's bandwidth. Finally, we investigated upsampling, increasing the sample rate of a signal by inserting zeros between existing samples and interpolating new values. These exercises helped solidify our understanding of sampling theory and its implications in digital signal processing applications, preparing us for more advanced topics in the field. Through hands-on coding and visualization, we gained practical experience in manipulating signals in both the time and frequency domains, essential skills for any aspiring DSP engineer.

Evaluation Rubric

- **Method of Evaluation**: In-lab marking by instructors, Report submitted by students
- **Measured Learning Outcomes**:

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713
CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

	Excellent 10	Good 9-7	Satisfactory 6-4	Unsatisfactory 3-1	Poor 0	Marks Obtained
Tasks (CLO1)	All tasks completed correctly. Correct code with proper comments.	Most tasks completed correctly.	Some tasks completed correctly.	Most tasks incomplete or incorrect.	All tasks incomplete or incorrect.	
Output (CLO1)	Output correctly shown with all Figures/Plots displayed as required and properly labelled	Most Output/Figures/Plots displayed with proper labels	Some Output/Figures/Plots displayed with proper labels OR Most Output/Figures/Plots displayed but without proper labels	Most of the required Output/Figures/Plots not displayed	Output/Figures/Plots not displayed	
Answers (CLO1)	Meaningful answers to all questions. Answers show the understanding of the student.	Meaningful answers to most questions.	Some correct/ meaningful answers with some irrelevant ones	Answers not understandable/ not relevant to questions	Not Written any Answer	
Report (CLO3)	Report submitted with proper grammar and punctuation with proper conclusions drawn and good formatting	Report submitted with proper conclusions drawn with good formatting but some grammar mistakes OR proper grammar but not very good formatting	Some correct/ meaningful conclusions. Some parts of the document not properly formatted or some grammar mistakes	Conclusions not based on results. Bad formatting with no proper grammar/punctuation	Report not submitted	
Total						