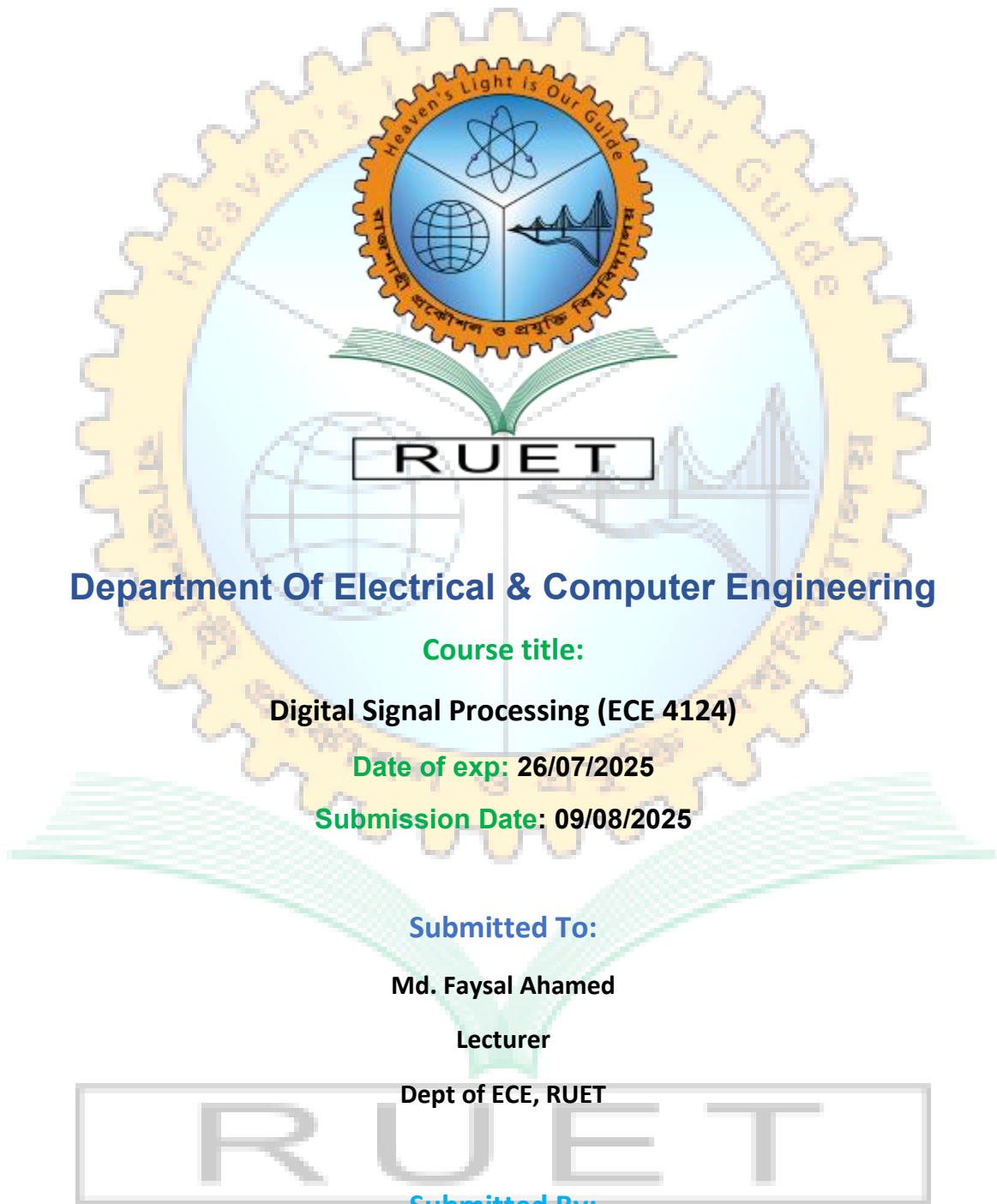


Heaven's light is our guide

RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY



Department Of Electrical & Computer Engineering

Course title:

Digital Signal Processing (ECE 4124)

Date of exp: 26/07/2025

Submission Date: 09/08/2025

Submitted To:

Md. Faysal Ahamed

Lecturer

Dept of ECE, RUET

Submitted By:

Syed Mahmudul Imran

Roll: 2010058

Experiment no.: 1

Experiment name: Signal Operations - **Folding, Shifting, and Rotating**

“

Perform the following Operations in the given signals:

Given signals	Operations
$\sin(x)$	folding $x(-n)$
$\cos(x)$	shifting $(n-2)$
$x(n) = \{1, 2, 3, 0, 4, -3\}$ [where 3 is 0th value]	rotating $[-x(-n)]$, $[-x(n)]$

Theory:

Signal operations are fundamental in digital signal processing. The main operations performed in this experiment are:

1. **Signal Folding (Time Reversal):** $x(-n)$

- This operation reflects the signal about the y-axis ($n=0$)
- For a discrete signal $x(n)$, folding produces $x(-n)$
- If $x(n)$ has values at indices n , then $x(-n)$ has the same values at indices $-n$

2. **Signal Shifting (Time Shifting):** $x(n-k)$

- This operation delays the signal by k samples when $k > 0$
- For $x(n-2)$, the signal is delayed by 2 samples
- The signal maintains its shape but appears at different time indices

3. **Signal Rotating:**

- Combination of folding and negation operations
- $-x(-n)$: First fold the signal, then negate all amplitude values
- $-x(n)$: Simply negate all amplitude values without folding

Procedure:**1. For $\sin(x)$ - Folding Operation $x(-n)$:**

- A discrete sine wave signal is generated
- Time reversal is applied by replacing n with $-n$
- Both original and folded signals are plotted for comparison using stem plots

2. For $\cos(x)$ - Shifting Operation $(n-2)$:

- A discrete cosine wave signal is generated
- Time shifting is applied by replacing n with $(n-2)$
- Both original and shifted signals are plotted using stem plots

3. For discrete signal $x(n) = \{1, 2, 3, 0, 4, -3\}$ - Rotating Operations:

- The discrete signal is defined with given values
- $-x(-n)$ operation is applied: First the signal is folded, then negated
- $-x(n)$ operation is applied: Signal values are simply negated
- All three signals (original, $-x(-n)$, $-x(n)$) are plotted for comparison

4. Analysis:

- The transformed signals are compared with original signals
- The effects of each operation on signal characteristics are observed

MATLAB CODE:

```
% Signal Operations Lab - LR2
clear all; close all; clc;

%% Create single figure with 3 columns for
all tasks
figure(1);
set(gcf, 'Position', [100, 100, 1200,
800]); % Make figure larger

%% 1. Sin(x) - Folding Operation x(-n) -
Column 1
n = -10:10; % discrete time indices
x_sin = sin(n);
x_sin_folded = sin(-n); % x(-n)

% Original sin(n) - Top left
subplot(3,3,1);
stem(n, x_sin, 'bo-', 'LineWidth', 2,
'MarkerSize', 6);
title('Task 1: Original sin(n)');
xlabel('n'); ylabel('sin(n)');
grid on;

% Folded sin(-n) - Bottom left
subplot(3,3,4);
stem(n, x_sin_folded, 'ro-', 'LineWidth',
2, 'MarkerSize', 6);
title('Folded sin(-n)');
xlabel('n'); ylabel('sin(-n)');
grid on;

%% 2. Cos(x) - Shifting Operation (n-2) -
Column 2
x_cos = cos(n);
% For shifting (n-2), we need to create
shifted indices
n_shifted = n + 2; % shift the time axis
x_cos_shifted = cos(n); % x(n-2) means the
signal appears 2 samples later

% Original cos(n) - Top middle
subplot(3,3,2);
stem(n, x_cos, 'bo-', 'LineWidth', 2,
'MarkerSize', 6);
title('Task 2: Original cos(n)');
xlabel('n'); ylabel('cos(n)');
grid on;
```

```
% Shifted cos(n-2) - Bottom middle
subplot(3,3,5);
stem(n_shifted, x_cos_shifted, 'go-',
'LineWidth', 2, 'MarkerSize', 6);
title('Shifted cos(n-2)');
xlabel('n'); ylabel('cos(n-2)');
grid on;

%% 3. Discrete Signal x(n) = {1, 2, 3, 0,
4, -3} - Rotating Operations - Column 3
% 3 is the center value (0th value), so
indices should be -2 to 3
n_discrete = -2:3; % indices for the given
signal with 3 at n=0
x_discrete = [1, 2, 3, 0, 4, -3];

% Original signal - Top right
subplot(3,3,3);
stem(n_discrete, x_discrete, 'bo-',
'LineWidth', 2, 'MarkerSize', 8);
title('Task 3: Original x(n) [3 at n=0]');
xlabel('n'); ylabel('x(n)');
grid on;

% -x(-n) operation: fold then negate -
Middle right
% For folding x(-n), we need to reverse
both indices and values
n_folded = -fliplr(n_discrete); % flip
indices then negate: [3, 2, 1, 0, -1, -2]
x_folded = fliplr(x_discrete); % flip the
values: [-3, 4, 0, 3, 2, 1]
x_neg_folded = -x_folded; % then
negate: [3, -4, 0, -3, -2, -1]

subplot(3,3,6);
stem(n_folded, x_neg_folded, 'ro-',
'LineWidth', 2, 'MarkerSize', 8);
title('Rotated -x(-n)');
xlabel('n'); ylabel('-x(-n)');
grid on;

% -x(n) operation: just negate - Bottom
right
x_negated = -x_discrete;

subplot(3,3,9);
stem(n_discrete, x_negated, 'go-',
'LineWidth', 2, 'MarkerSize', 8);
title('Negated -x(n)');
xlabel('n'); ylabel('-x(n)');
grid on;
```

```
% Add overall title
sgtitle('DSP Lab 2: Signal Operations -
Folding, Shifting, and Rotating',
'FontSize', 16, 'FontWeight', 'bold');

%% Display results
disp('Signal Operations Results:');
disp('=====');
disp(' ');
disp('1. Discrete Sine Signal:');
disp(['Original sin(n) at n=0: ',
num2str(sin(0))]);
disp(['Folded sin(-n) at n=0: ',
num2str(sin(0))]);
disp(' ');
```

```
disp('2. Discrete Cosine Signal:');
disp(['Original cos(n) at n=0: ',
num2str(cos(0))]);
disp(['Shifted cos(n-2) appears at n=2: ',
num2str(cos(0))]);
disp(' ');
disp('3. Discrete Signal x(n):');
disp(['n = ', num2str(n_discrete)]);
disp(['x(n) = ', num2str(x_discrete)]);
disp('Note: Center value 3 is at n=0');
disp(' ');
disp('Folded and negated signal -x(-n):');
disp(['n = ', num2str(n_folded)]);
disp(['-x(-n) = ', num2str(x_neg_folded)]);
disp(' ');
disp('Negated signal -x(n):');
disp(['n = ', num2str(n_discrete)]);
disp(['-x(n) = ', num2str(x_negated)]);
```

Output:

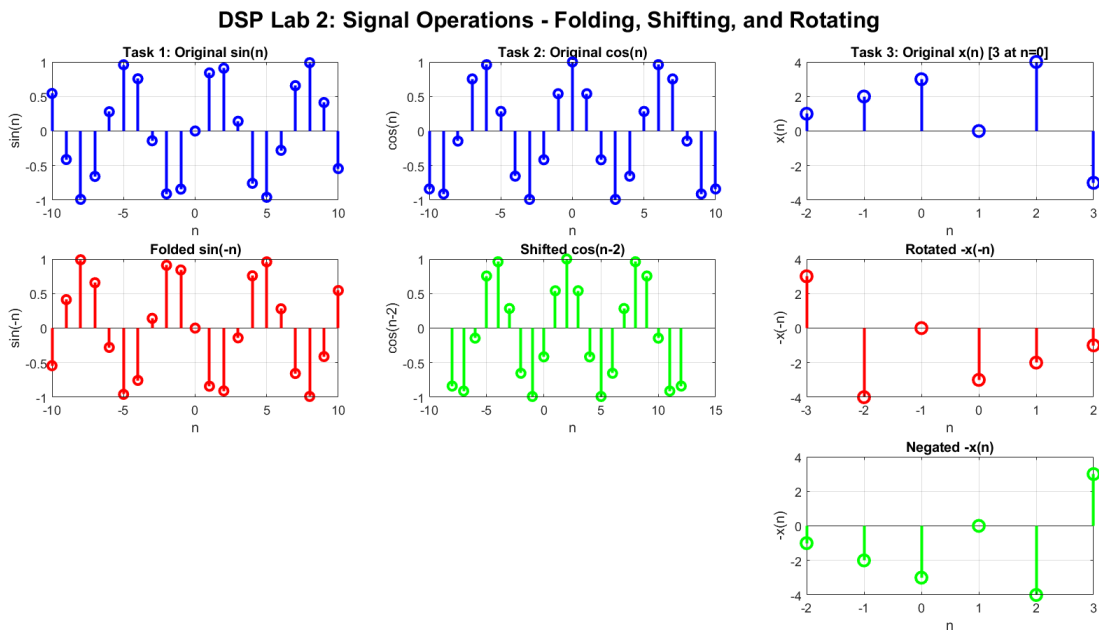


Figure 1: signal output of folding, shifting and rotating

Discussion:

1. Folding Operation on $\sin(n)$:

- The folded discrete sine signal $\sin(-n)$ is equivalent to $-\sin(n)$ due to the odd symmetry property of sine function
- For discrete signals, the folding operation reflects the signal samples about $n=0$
- This operation demonstrates the time-reversal property in discrete signal processing

2. Shifting Operation on $\cos(n)$:

- The shifted discrete cosine signal $\cos(n-2)$ represents a delayed version of the original signal
- The delay of 2 samples shifts the entire discrete signal to the right by 2 positions
- The amplitude and shape remain unchanged, only the time indices are affected

3. Rotating Operations on Discrete Signal:

- **Original signal:** $x(n) = \{1, 2, 3, 0, 4, -3\}$ at indices $n = \{-2, -1, 0, 1, 2, 3\}$ with center value 3 at $n=0$
- **$-x(-n)$ operation:** Results in $\{3, -4, 0, -3, -2, -1\}$ at indices $\{3, 2, 1, 0, -1, -2\}$
 - First the signal is folded (time-reversed): $\{-3, 4, 0, 3, 2, 1\}$
 - Then all amplitude values are negated: $\{3, -4, 0, -3, -2, -1\}$
- **$-x(n)$ operation:** Results in $\{-1, -2, -3, 0, -4, 3\}$ at the same indices $\{-2, -1, 0, 1, 2, 3\}$
 - Only amplitude values are negated without any time manipulation

Key Observations:

- All signals are now treated as discrete-time signals using stem plots
- Folding changes the time axis representation for discrete samples
- Shifting translates discrete signal samples in the time domain
- Negation affects only the amplitude values of discrete samples
- Combined operations ($-x(-n)$) affect both time and amplitude domains for discrete signals

Conclusion:

This experiment successfully demonstrated fundamental discrete signal operations in digital signal processing:

1. **Folding (Time Reversal):** Successfully implemented $x(-n)$ operation on discrete $\sin(n)$, showing how discrete signals can be reflected about the time axis. The operation revealed the odd symmetry property of discrete sine function.
2. **Time Shifting:** Implemented shifting operation $x(n-2)$ on discrete $\cos(n)$, demonstrating how discrete signals can be delayed in time domain while preserving their shape and amplitude characteristics.
3. **Rotating Operations:** Applied both $-x(-n)$ and $-x(n)$ operations on the discrete signal, showing the difference between:
 - Combined folding and negation: $-x(-n)$
 - Simple amplitude negation: $-x(n)$

Learning Outcomes:

- Understanding of basic discrete signal transformations
- Practical implementation of discrete signal operations in MATLAB
- Visualization of discrete signal behavior under different operations using stem plots
- Recognition of how time and amplitude domains can be independently manipulated in discrete systems

These discrete operations form the foundation for more complex digital signal processing techniques such as discrete convolution, correlation, and digital filtering operations used in various engineering applications.

References:

1. Oppenheim, A. V., & Schaffer, R. W. (2010). *Discrete-Time Signal Processing* (3rd ed.). Pearson Education.
2. Proakis, J. G., & Manolakis, D. G. (2013). *Digital Signal Processing: Principles, Algorithms, and Applications* (4th ed.). Pearson.
3. Mitra, S. K. (2011). *Digital Signal Processing: A Computer-Based Approach* (4th ed.). McGraw-Hill Education.
4. MATLAB Documentation. (2023). *Signal Processing Toolbox User's Guide*. MathWorks.