LABORATORY REPORT

ELEC 342 Discrete Time Signals and Systems

FALL 2023

Course: ELEC 342 **Lab Section:** UN-X

Experiment No.: 2 **Date Performed:** 2023 – 10 – 03
YYYY – MM – DD

Experiment Title: The Discrete Time Fourier Transform and Introduction to Simulink

Name: BAYAN ALSALEM **ID No.:** 40105034

I certify that this submission is my original work and meets the Faculty's Expectations of Originality

Signature: **Date:** 2023 – 10 –04
YYYY – MM – DD

Table of Contents

Objectives

In this laboratory, the primary objectives are twofold. In Part I, the focus is on implementing the discrete time Fourier transform of a pulse input through the utilization of loops and arrays and in-built functions. This portion of the lab aims to provide hands-on experience with signal processing techniques and algorithmic implementations. In Part II, the lab transitions to introduce the fundamental concepts of Simulink, a powerful simulation environment integrated within MATLAB.

Theory

The Discrete Time Fourier Transform (DTFT) is a fundamental mathematical tool used in the field of signal processing and digital signal analysis. It plays a pivotal role in analyzing and understanding the frequency content of discrete-time signals. The DTFT is defined by the equation:

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{n=+\infty} x[n]e^{-j\omega n}$$

In this equation, X(e^{j\omega}) represents the DTFT of the discrete-time signal x[n], which is a complex-valued function of the continuous frequency variable omega. The DTFT essentially characterizes how the signal x[n] is composed of various sinusoidal components at different frequencies, and it provides a continuous spectrum of these frequency components.

The summation in the equation extends over all possible values of n, making the DTFT applicable to both finite and infinite sequences. The complex exponential term e^ {-j*omega*n} serves as a basis function, and by taking the inner product of x[n] with these basis functions for all values of n\, we obtain the representation of the signal in the frequency domain.

The DTFT is a powerful tool for analyzing the frequency characteristics of signals and is used in various applications, including filter design, spectrum analysis, and modulation techniques. It provides valuable insights into how a discrete-time signal is distributed across the frequency spectrum, making it an essential concept in the field of digital signal processing. [1]

Simulink is a powerful graphical simulation and modeling tool integrated with MATLAB. It provides engineers and scientists with a user-friendly environment for designing, simulating, and analyzing complex systems and control algorithms. Simulink allows users to create block diagrams that represent dynamic systems visually, with each block representing a specific function or component of the system. These blocks can be interconnected to define the system's behavior, and Simulink provides various tools for simulating the dynamics, observing the system's response, and tuning control parameters. It is widely used in fields such as control systems, signal processing, communications, and more, making it a versatile tool for system modeling and simulation in various engineering disciplines.[2]

Tasks, Results, Discussion and Questions

Part 1

Question 1

Obtain the DTFT of a pulse $x[n]$ shown in Figure 1 over the interval $-10 \leq n \leq 10$

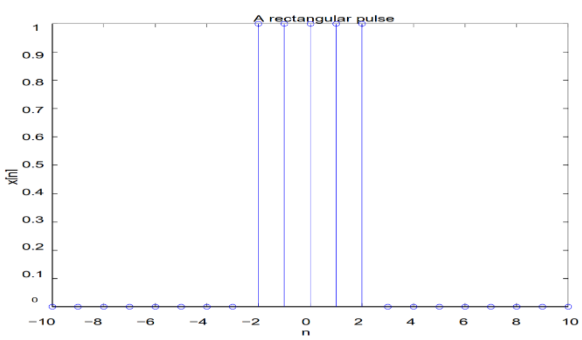

Figure 1: a rectangular pulse [3]

Compute the transform over the frequency interval $-\pi \leq w \leq \pi$ using a user-defined input value for the step size. Plot the signal x[n] (using stem) together with its transform (using plot). Use the length function to control the number of iterations of the outer for loop given in the pseudocode algorithm.

Please refer to the MATLAB code in figure 1 and the plots in figure 2.

Question 2

MATLAB has a built-in function called fft which computes the discrete Fourier transformation of an input vector x(n). Re-write your script for Question 1 such that the transform is computed over the interval of $0 \leq w \leq 2\pi$ and compare the results of your transform with that produced by the MATLAB fft command. Note that if the input signal x[n] is a vector of length 21 (21 values of n ranging from -10 to +10), then x_fft = fft(x) will be a 1 x 21 array. In order to compare the results of plot (w, x_fft) with the results of your transform, it will be necessary to choose a step size for w such that the size of the array holding your transform values also has length 21.

For this question, I employed the MATLAB built-in function named fft to obtain the Fourier transform of a given input vector x(n). Using this functionality, my objective was to compare the result of this built-in function with the manually computed function defined in Question 1. However, the question required me to define an input signal x[n] as a vector of length 21. I used a step size of (2pi) / 21 to accommodate the 21 values, and the range of frequencies was defined as 0 <= (2pi) / 21 <= ((2pi) - (2pi) / 21) to ensure that the transform was computed over the interval 0 <= w <= 2*pi.

The plot clearly demonstrates that we obtain **identical results.** This indicates that both manual computation of the DTFT and utilizing the built-in MATLAB function yield the same outcome.

Please refer to the MATLAB code in figure 3 and the plots in figure 4.

Question 3

MATLAB has a function called ifft which computes the inverse discrete Fourier transform. Compute the fft of the input signal $x[n]$ given in Question 1, and then use it as input to the ifft function to obtain the original signal $x[n]$. Plot the original signal $x[n]$ together with the signal obtained from the ifft command.

Please refer to the MATLAB code in figure 5 and the plots in figure 6.

Part 2

Question 1

Use Simulink to simulate the behavior of the difference equation $y[n] = x[n] + 1\ 4\ y[n-1]$. Use a sine wave as the input $x[n]$. Decide upon a frequency and sample time of your choice.

Please refer to figures 7, 8, and 9.

Conclusion

In summary, our primary goal is to explore different built-in MATLAB functions like fft and ifft. We begin by examining the Fourier transform and its inverse for an input signal, x[n], across various frequency ranges and step sizes. Additionally, we utilize Simulink to model these input signals and simulate them, with the results visible in the Scope block.

References

(1) S. W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing - Chapter 10: Frequency Sampling Filter Design - 6. The Frequency Spectrum," [Online]. Available: https://www.dspguide.com/ch10/6.htm. [Accessed: Oct. 04, 2023].

(2) "What is Simulink in Matlab? | How Simulink work in Matlab with Examples," Educba, 24-Sep-2023. [Online]. Available: https://www.educba.com/what-is-simulink-in-matlab/. [Accessed: 04-Oct-2023].

(3) Obuchowicz Ted, ELEC 342 Lab 3

Appendix

Lab2_part_1_Q_1.m

```matlab
% Bayan Alsalem
% ID: 40105034


% Define the time domain (-10 to 10)
n = -10:10;

% Initialize the frequency domain
w = -pi:0.01:pi;            % User-defined step size, here 0.01

% Define the pulse signal x[n] (input signal)
x = zeros(size(n));         % Initialize x[n] with zeros
x(n >= -2 & n <= 2) = 1;    % Set x[n] to 1 where specified

% Plot the signal x[n]
subplot(2, 1, 1);
stem(n, x, 'b', 'filled');
title('Signal x[n]');
xlabel('n');
ylabel('x[n]');


% Initialize the DTFT result
Xw = zeros(size(w));

% Compute the DTFT
for index_w = 1:length(w)
    sum = 0;
    for index_n = 1:length(n)
        sum = sum + (x(index_n) * exp(-1i * w(index_w) * n(index_n)));
    end
    Xw(index_w) = sum;
end

% Plot the DTFT magnitude using plot
subplot(2, 1, 2);
plot(w, abs(Xw), 'r', 'LineWidth', 2);
title('Magnitude of DTFT');
xlabel('Frequency (w)');
ylabel('Magnitude');
grid on;
```

Figure 1: MATLAB code for Part 1 Question 1

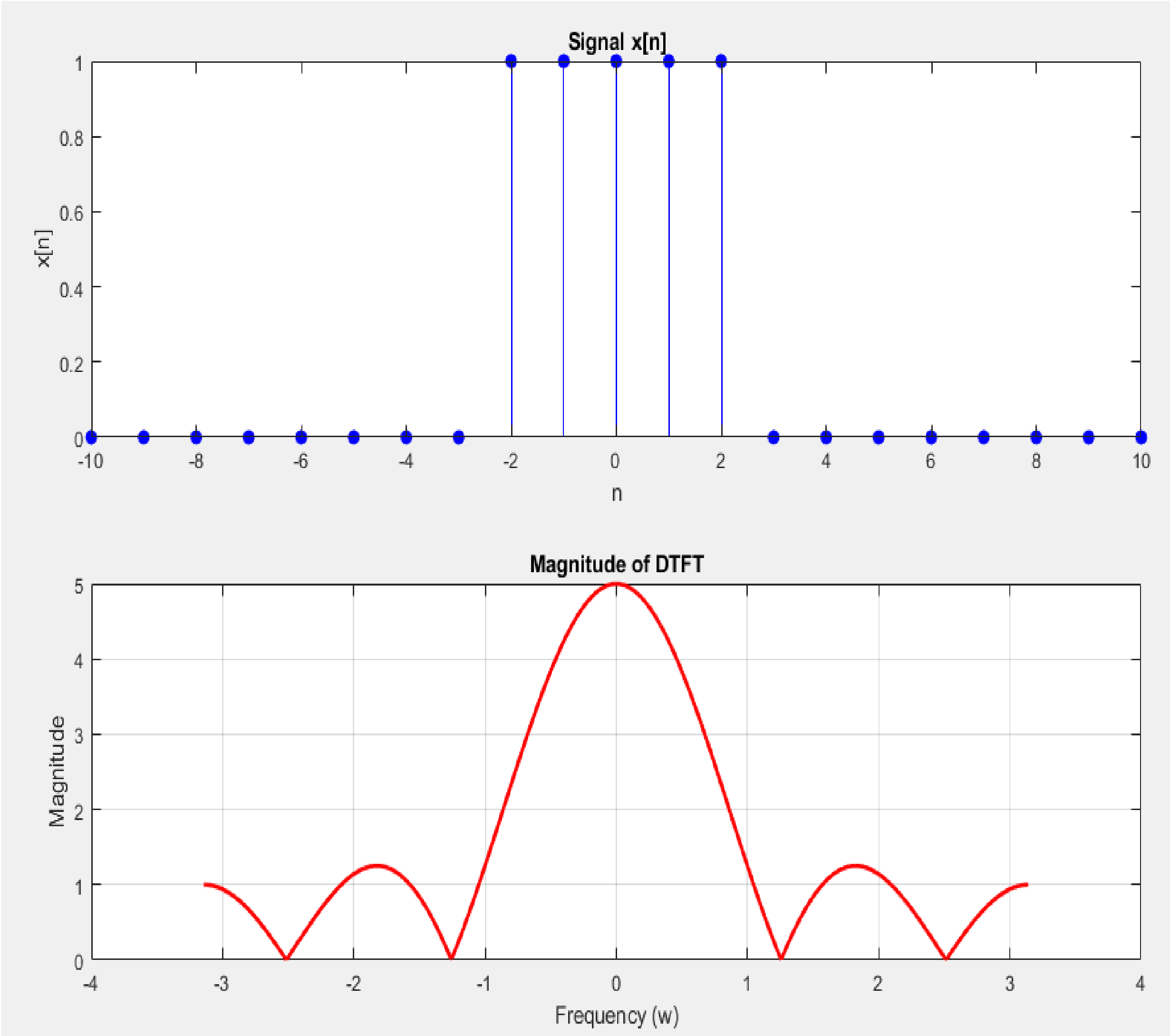

Figure 2: MATLAB Plot for Part 1 Question 1

Lab2_part_1_Q_2.m

```matlab
% Bayan Alsalem
% ID: 40105034

% Define the time domain (-10 to 10)
n = -10:10;

% Initialize the frequency domain
w = 0:2*pi/21:2*pi-2*pi/21;  % User-defined step size to make the length 21

% Define the pulse signal x[n] (input signal)
x = zeros(size(n));        % Initialize x[n] with zeros
x(n >= -2 & n <= 2) = 1;   % Set x[n] to 1 where specified

% Create a figure
figure;
```

```matlab
% Initialize the DTFT result
Xw = zeros(size(w));

% Compute the DTFT manually
for index_w = 1:length(w)
    sum = 0;
    for index_n = 1:length(n)
        sum = sum + (x(index_n) * exp(-1i * w(index_w) * n(index_n)));
    end
    Xw(index_w) = sum;
end

% Plot the DTFT magnitude using plot
subplot(2, 1, 1);
plot(w, abs(Xw), 'r', 'LineWidth', 2);
title('Magnitude of DTFT');
xlabel('Frequency (w)');
ylabel('Magnitude');
grid on;
```

```matlab

% Compute the DTFT using MATLAB's fft
Xw_fft = fft(x);

% Plot the magnitude of DTFT in the second subplot
subplot(2, 1, 2);
plot(w, abs(Xw_fft), 'b', 'LineWidth', 2);
title('Magnitude of DTFT (MATLAB fft)');
xlabel('Frequency (w)');
ylabel('Magnitude');
grid on;

```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> Lab2_part_1_Q_2
>> Lab2_part_1_Q_2
>> length(w)

ans =

    21

>> length(Xw_fft)

ans =

    21
```

Figure 3: MATLAB code for Part 1 Question 2

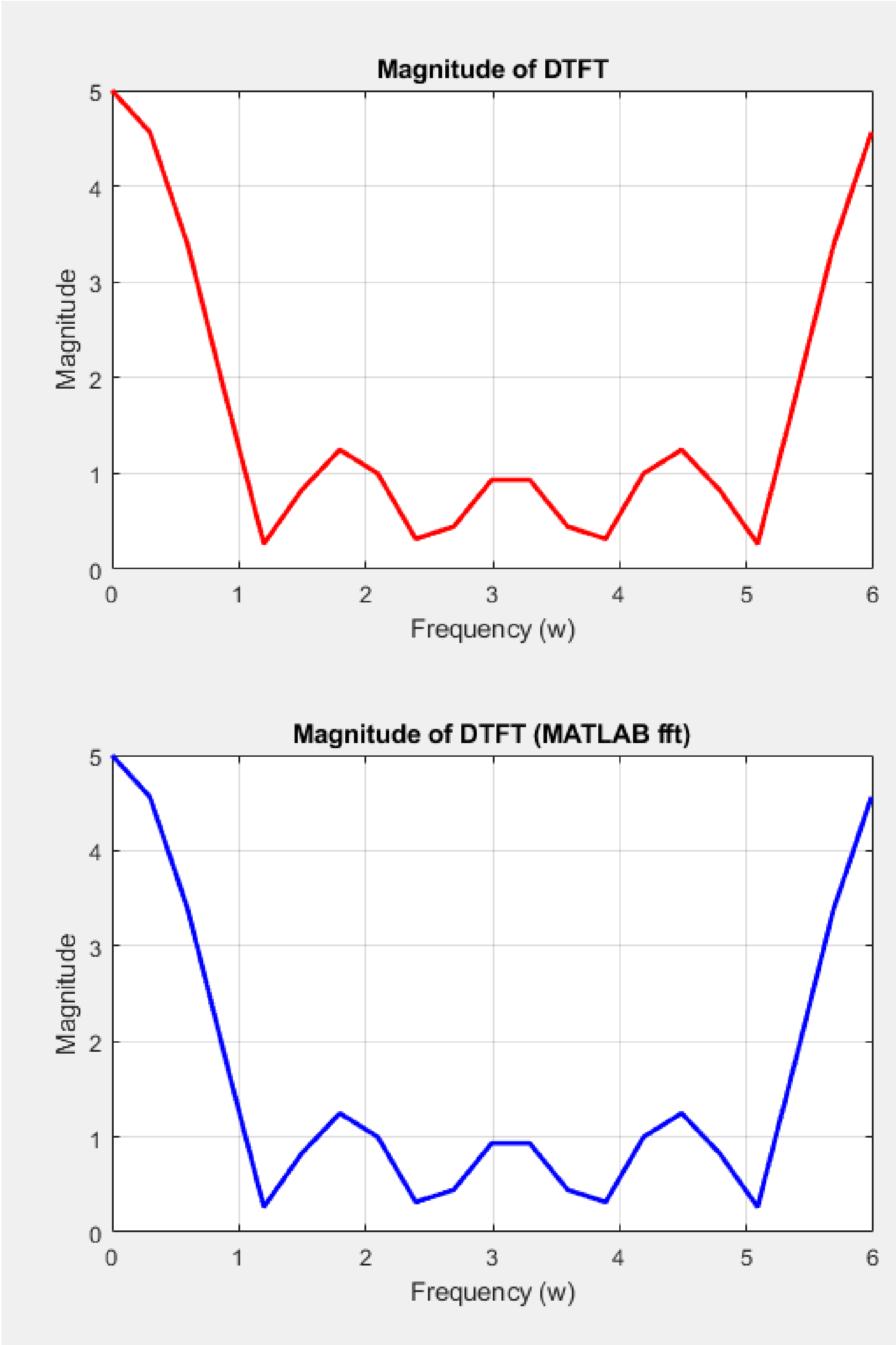

Figure 4: MATLAB Plot for Part 1 Question 2

Lab2_part_1_Q_3.m

```
1   % Bayan Alsalem
2   % ID: 40105034
3
4   % Define the time domain (-10 to 10)
5   n = -10:10;
6
7   % Initialize the frequency domain
8   w = 0:2*pi/21:2*pi-2*pi/21;  % User-defined step size to make the length 21
9
10  % Define the pulse signal x[n] (input signal)
11  x = zeros(size(n));        % Initialize x[n] with zeros
12  x(n >= -2 & n <= 2) = 1;   % Set x[n] to 1 where specified
13
14
15  % Compute the FFT of the input signal x[n]
16  X_fft = fft(x);
17
18  % Compute the IDFT to obtain the original signal x[n]
19  x_inverse = ifft(X_fft);
20
21  % Plot the original signal x[n]
22  subplot(3, 1, 1);
23  stem(n, x, 'b', 'filled');
24  title('Original Signal x[n]');
25  xlabel('n');
26  ylabel('x[n]');
27
28  % Plot FFT of the input signal x[n]
29  subplot(3, 1, 2);
30  plot(w, abs(X_fft), 'g');
31  title('Magnitude of DTFT');
32  xlabel('Frequency (w)');
33  ylabel('Magnitude');
34  grid on;
35
36  % Plot the inverse signal obtained from ifft
37  subplot(3, 1, 3);
38  plot(w, abs(x_inverse), 'r');
39  title('Inverse Signal (from ifft)');
40  xlabel('Frequency (w)');
41  ylabel('Inverse x[n]');
42
```

*** Line 38 supposed to be: stem(w, abs(x_inverse), 'r','filled');

Figure 5: MATLAB code for Part 1 Question 3

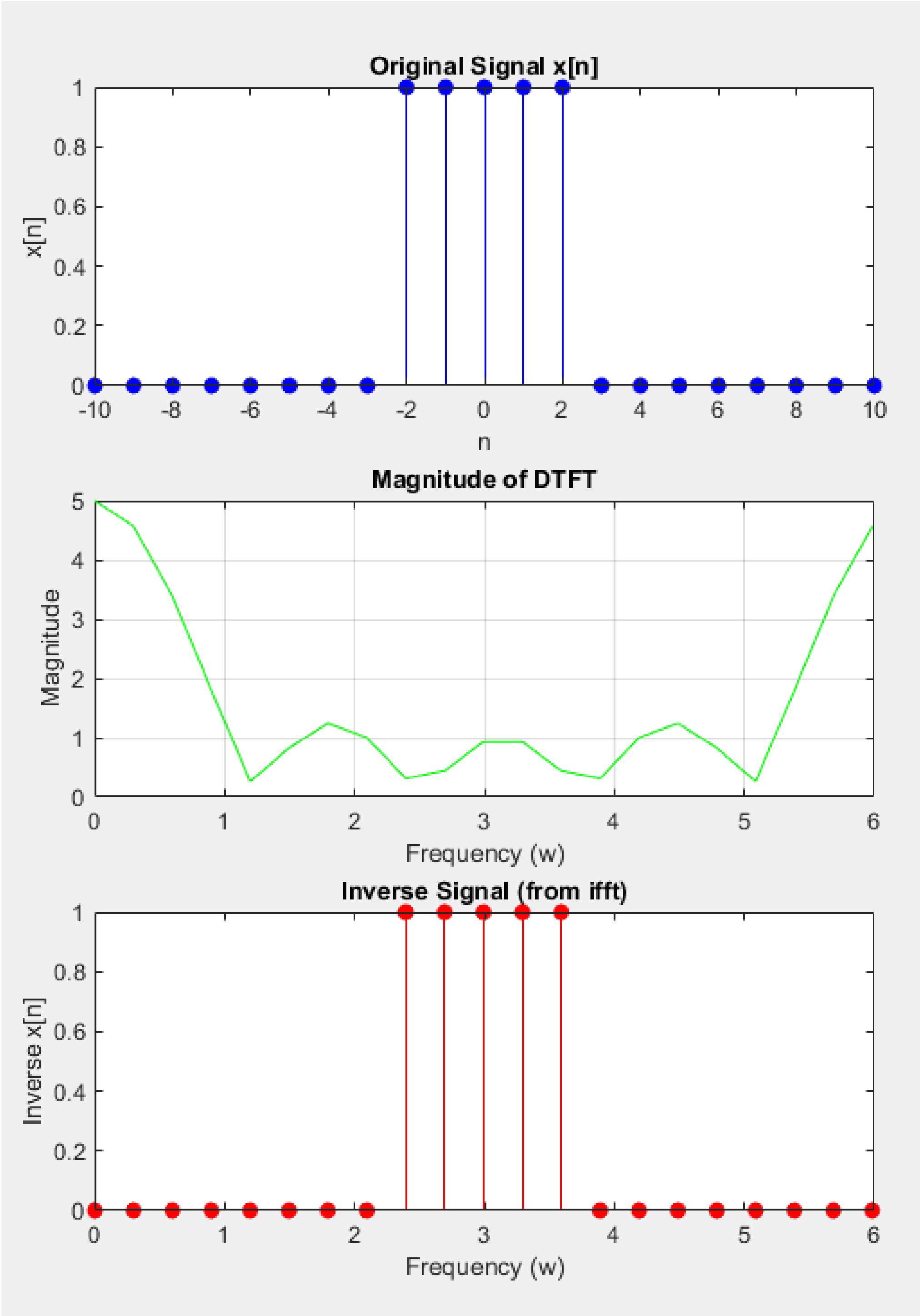

Figure 6: MATLAB Plot for Part 1 Question 3

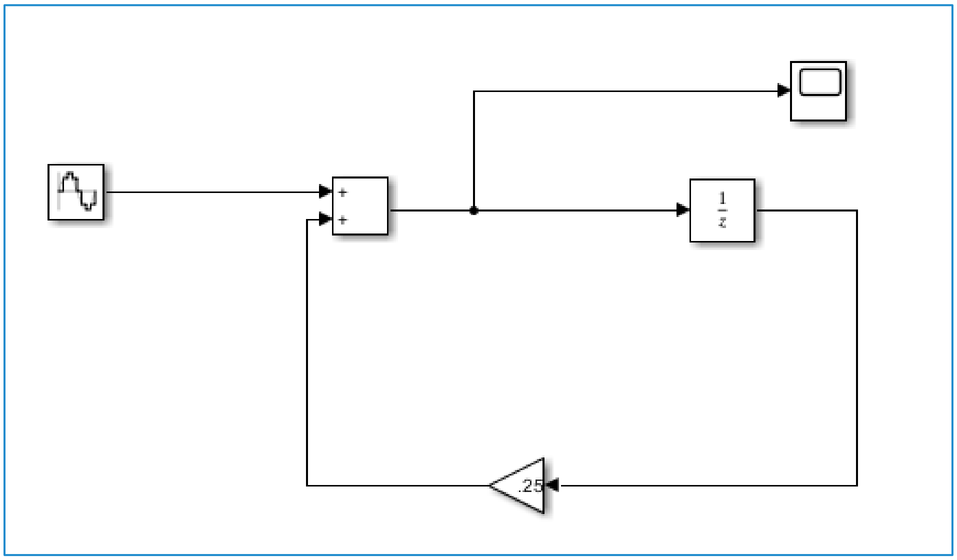

Figure 7: The Simulink model of $y[n] = x[n] + 1\ 4\ y[n-1]$

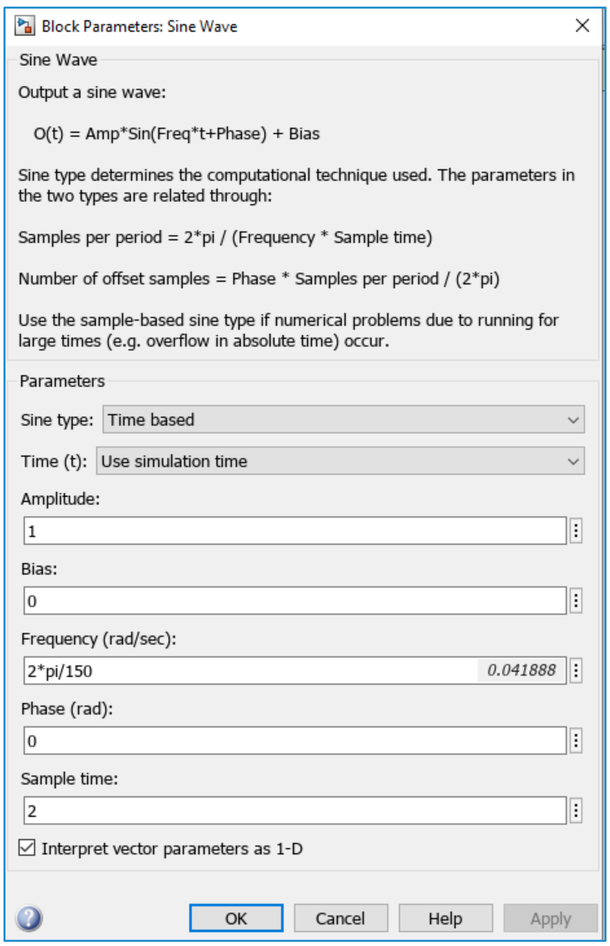

Figure 8: The Block Parameters for the Sine Wave Block

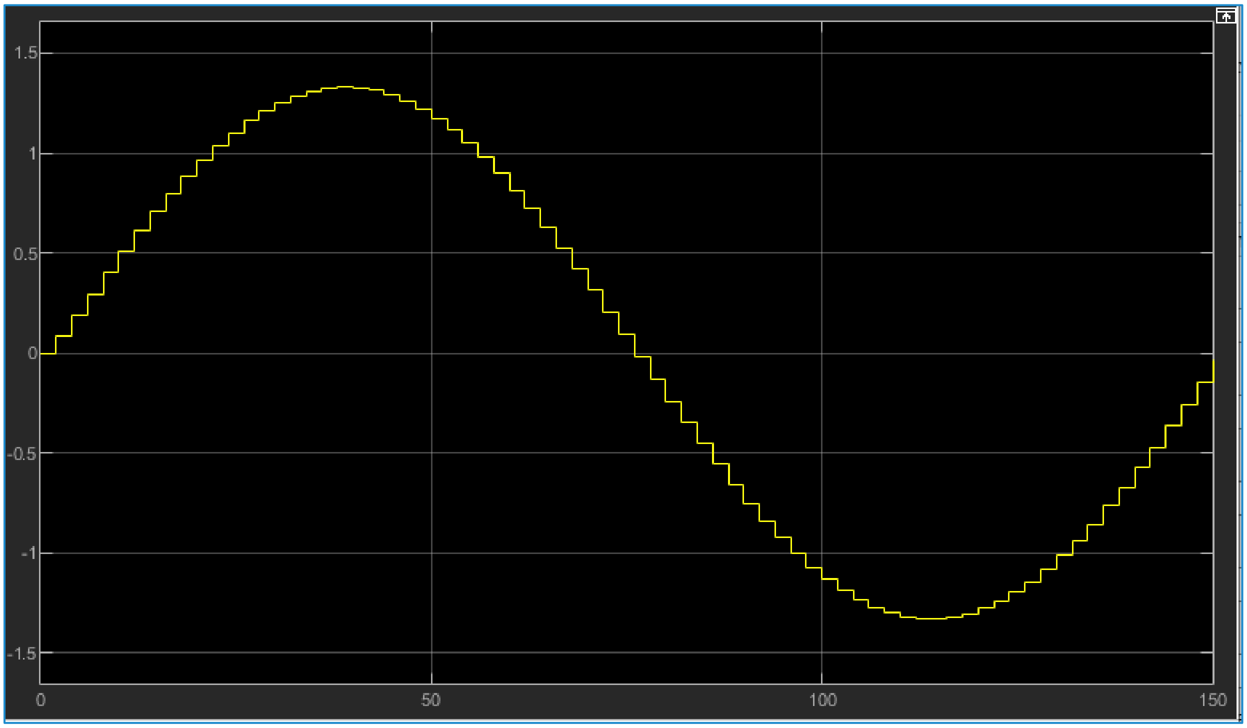

Figure 9: The Scope