Bengisu Münteha Koç

21801974

EEE 321 – 02

17.10.2024

EEE 321 Lab Work 2

Introduction

The aim of this assignment is to explore the behavior and response of a discrete Linear Time Invariant (LTI) system using convolution. The impulse response of the system is given as

$$h[\mathrm{n}] = \left(\frac{7}{8}\right)^n u[n-4]$$

The objective is to analyze the system's stability and causality. Additionally, the system's response to various input signals is computed both analytically and numerically.

The results are presented through MATLAB code that calculates and plots numerical convolution results. The lab aims to provide a comprehensive hands-on experience with system responses, convolution and compares the analytical and numerical convolution results.

Convolution is a fundamental mathematical operation used in signal processing to determine the output of a Linear Time-Invariant (LTI) system when subjected to a given input signal. In discrete time, the convolution of two sequences x[n] (input) and h[n] (impulse response) are defined as

$$y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Implementation and Results

1. The impulse response of a discrete linear time invariant is given as

$$h[n] = \left(\frac{7}{8}\right)^n u[n-4]$$

The impulse response has a unit step function part. Unit step function is

$$u[n] = \begin{cases} 0 \text{ if t } < 0 \\ 1 \text{ if t } \geq 0 \end{cases}$$

Hence

$$u[n\text{-}4] = \begin{cases} 0 \text{ if t } < 4 \\ 1 \text{ if t } \geq 4 \end{cases}$$

So, the given impulse response is zero for all n smaller than 4.

The impulse response is defined and plotted in MATLAB as follows

```
n = 0:50;                                         % Range for n


% Impulse response h[n]
h = (7/8).^(n) .* (n >= 4);                       % h[n] = (7/8)^(n-4) * u[n-4]


% Plot of h[n]
stem(n, h, 'filled');
title('Impulse Response h[n] = (7/8)^n u[n - 4]');
xlabel('n');
ylabel('h[n]');
grid on;
```

The range for the impulse response is set to 50, to observe a meaningful finite size segment of the signals.

The plotting of the impulse response is shown in figure 1.

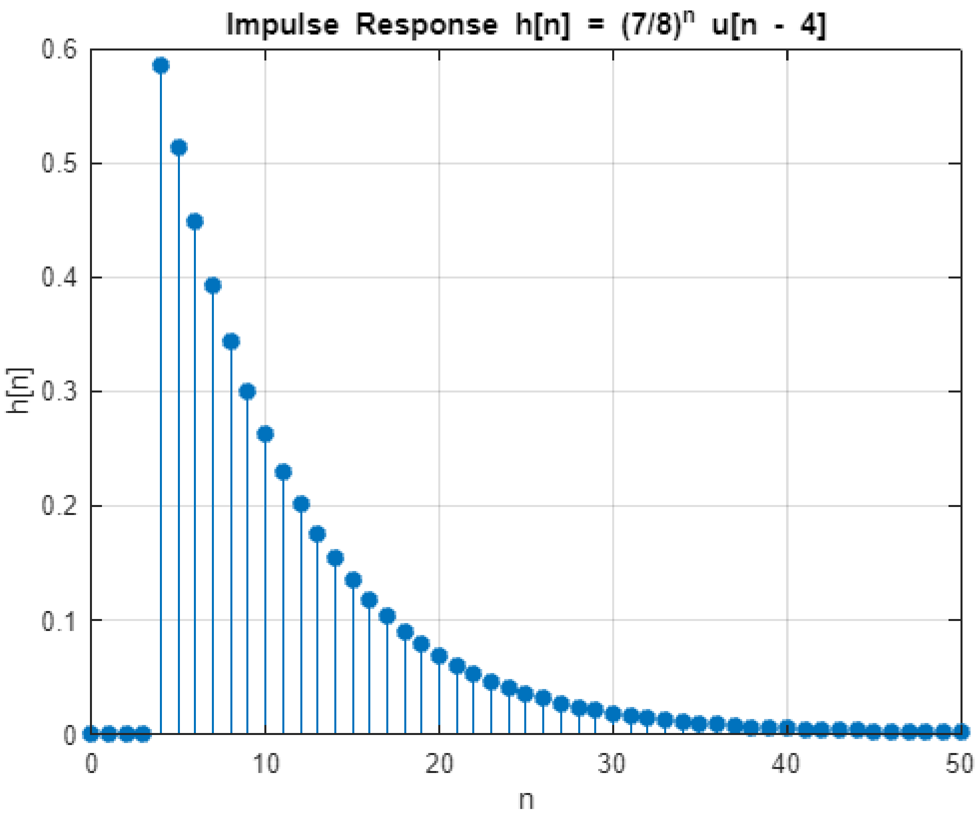

Figure 1: Impulse Response $h[n] = (7/8)^{n}$ u[n-4]

As mentioned before, the impulse response is zero for n < 4.

A system is said to be causal if its output for arbitrary n depends only on present and past inputs, and not future inputs. From the definition of unit response part of the impulse response, it can be concluded that the system does not rely on any future values of the input. Hence the system is causal.

A system is said to be stable if

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

Hence to check stability, the sum of the absolute values of h[n] for $n \geq 4$ should be evaluated as above

$$\sum_{n=4}^{\infty} \left(\frac{7}{8}\right)^{n}$$

This is a geometric series with a common ratio of 7/8, which is smaller than 1. The sum is found as

$$S = \frac{\left(\frac{7}{8}\right)^4}{1-\frac{7}{8}} = \frac{\left(\frac{7}{8}\right)^4}{\frac{1}{8}} = 8\left(\frac{7}{8}\right)^4 < \infty$$

Hence the system is stable.

From figure 1, the one can be forecast the stability of the system due the convergence of the plot to zero.

The impulse response will be convolved with six different input functions.

For convolution process in MATLAB, a function called "manual_convolution(x, h)" is constructed as shown below

```
function y = manual_convolution(x, h)

    Lx = length(x);          % Length of the input sequence x
    Lh = length(h);          % Length of the impulse response h
    Ly = Lx + Lh - 1;        % Length of the output sequence y
    y = zeros(1, Ly);        % Initializing y with zeros

    % Convolution
    for i = 1:Ly
        for j = 1:Lh
            if (i-j+1 > 0) && (i-j+1 <= Lx)
                y(i) = y(i) + h(j) * x(i-j+1);
            end
        end
    end
end
```

After the length of the output is set to the sum of the length of the impulse response end input function, the output is initialized with zero. Then the convolution process is conducted. The process consists of flipping and shifting input signals and adding the multiplication of the impulse response and input signal, point by point.

a) The given input function is

$$x_1[n] = \begin{cases} 3 \text{ if } 0 \leq n \leq 8 \\ 0 \text{ else} \end{cases}$$

The plot of the signal is shown in figure 2.

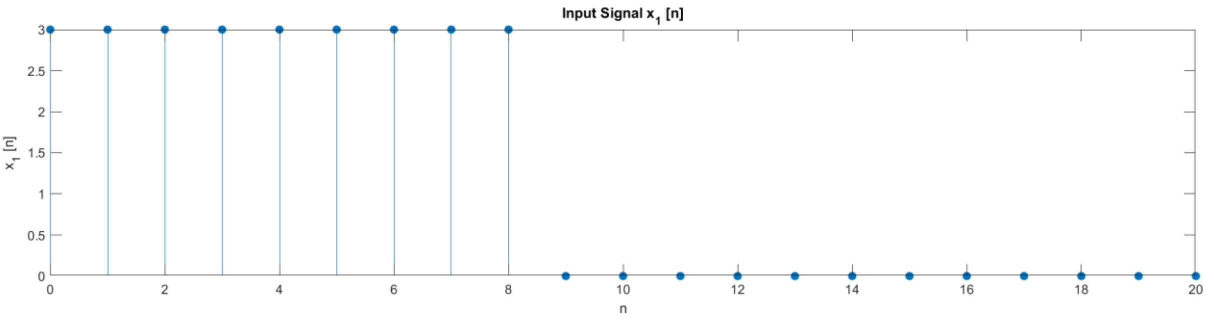

Figure 2: Plot of $x_1[n]$

The analytical calculation of the convolution is made by flipping the input function $x_1[n]$ and the result is found as follows

$$y_1[n] = x_1[n] * h[n] = h[n] * x_1[n] = \sum_{m=-\infty}^{\infty} h[m]x_1[n-m]$$

While making the convolution, the important part is to decide boundaries. Because, both impulse response and input signal are zero for some n values. Hence the convolution is also zero for some interval. For this input signal;

$$y_1[n] = \begin{cases} 0 \text{ if } n < 4 \\ 3\sum_{m=4}^{n} \left(\frac{7}{8}\right)^m \text{ if } 4 \leq n \leq 12 \\ 3\sum_{m=n-8}^{n} \left(\frac{7}{8}\right)^m \text{ if } n > 12 \end{cases}$$

The result is zero for n < 4, because the impulse response is zero for this interval. This boundary comes from impulse response u[n-4]. Since the nonzero value length of the input signal is 8, the interval is [4, 12].

The definition of the input signal, plotting process of input signal and output signal in MATLAB is implemented as follows

```
n = 0:20;
x1 = (n >= 0 & n <= 8) * 3;                    % x1[n] = 3 for 0 <= n <= 8,
else 0
```

```
% Convolution Function Call
y1 = manual_convolution(x1, h);


% Plot of the input signal x1[n]
figure;
subplot(2, 1, 1);
stem(n, x1, 'filled');
xlabel('n');
ylabel('x_1 [n]');
title('Input Signal x_1 [n]');


% Plot of the convolution result
n_output = 0:(length(y1)-1);
subplot(2, 1, 2);
stem(n_output, y1, 'filled');
xlabel('n');
ylabel('y_1 [n]');
title('Output y[n] = x_1 [n] * h[n] via manual convolution');
```

The output plot of the convolution is shown in figure 3.

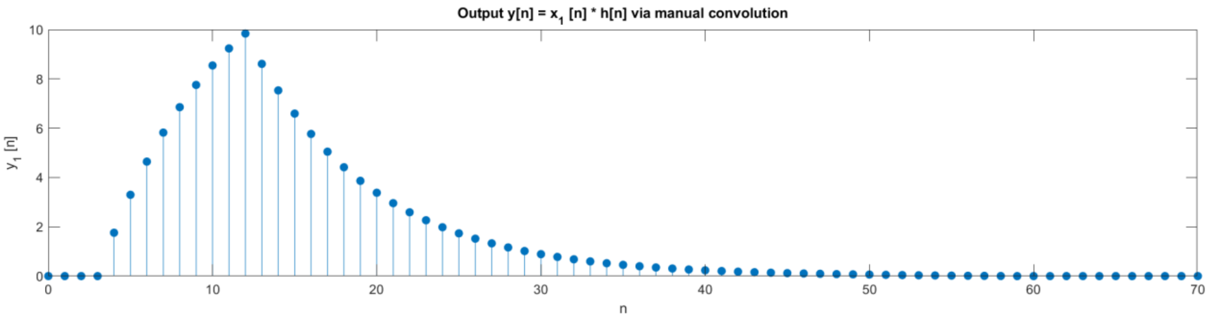

Figure 3: $y_1[n] = x_1[n] * h[n]$

When arbitrary n values tested in calculation result, the results match with he obtained plot.

b) The given input function is

$$x_2[n] = \begin{cases} 3 \text{ if } 0 \leq n \leq 4 \\ -3 \text{ if } 5 \leq n \leq 8 \\ -6 \text{ if } 9 \leq n \leq 18 \end{cases}$$

$$x_2[n] = 0 else$$

The plot of the input signal $x_2[n]$ is given in figure 4.

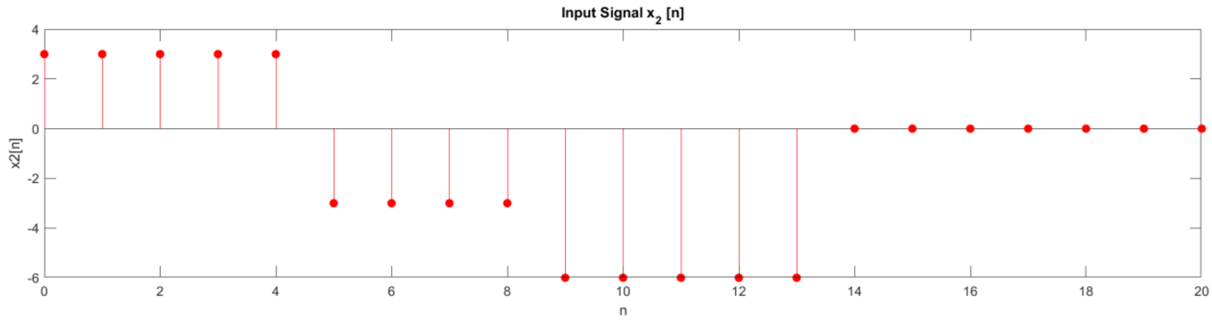

Figure 4: Plot of $x_2[n]$

The analytical calculation of the convolution is found as

$$y_2[n] = x_2[n] * h[n] = h[n] * x_2[n] = \sum_{m=-\infty}^{\infty} h[m]x_2[n-m]$$

By comparing plots of $x_1[n]$ and $x_2[n]$, and the definitions, it can be seen that

$$x_2[n] = x_1[n] - 2x_1[n-5]$$

Convolution of the impulse response and input signal is

$$y_2[n] = x_2[n] * h[n]$$

From the above relation, it can be written as

$$y_2[n] = (x_1[n] - 2x_1[n-5]) * h[n]$$

Using the distributive property of the convolution

$$y_2[n] = (x_1[n] * h[n]) - 2(x_1[n-5] * h[n])$$

Since convolution has shifting property, the second convolution can be written as

$$y_2[n] = y_1[n] - 2y_1[n\text{ - }5]$$

Hence the calculation result is

$$y_2[n] = 0 \; if \; n < 4$$

$$y_2[n] = 3\sum_{m=4}^{n}\left(\frac{7}{8}\right)^m \text{ if } 4 \leq n < 9$$

$$y_2[n] = 3\sum_{m=4}^{n}\left(\frac{7}{8}\right)^m \text{ - } 3.2\sum_{m=4}^{n\text{-}5}\left(\frac{7}{8}\right)^m \text{ if } 9 \leq n < 12$$

$$y_2[n] = 3\sum_{m=\text{n-8}}^{n}\left(\frac{7}{8}\right)^m \text{ - } 3.2\sum_{m=4}^{n\text{-}5}\left(\frac{7}{8}\right)^m \text{ if } 12 < n \leq 17$$

$$y_2[n] = 3 \sum_{m=n-8}^{n} \left(\frac{7}{8}\right)^m - 3.2 \sum_{m=n-13}^{n-5} \left(\frac{7}{8}\right)^m \quad \text{if } n > 17$$

Using the shifting property, the intervals are also shifted.

The definition of the input signal, plotting process of input signal and output signal in MATLAB is implemented as follows

```
y2_shifted = [zeros(1,5) y1];                    % Shifting y1[n] by 5 (adding
5 zeros at the beginning)
y2_shifted = y2_shifted(1:length(y1));           % Length shift
y2 = y1 - 2 * y2_shifted;


% Plot
figure;
subplot(2, 1, 1);
x2 = [3*ones(1, 5), -3*ones(1, 4), -6*ones(1, 5), zeros(1, 7)]; % x2[n]
definition
stem(n, x2, 'filled','r');
xlabel('n');
ylabel('x2[n]');
title('Input Signal x_2 [n]');


% Plot the output y2[n]
subplot(2, 1, 2);
n_output = 0:(length(y2)-1);
stem(n_output, y2, 'filled','r');
xlabel('n');
ylabel('y_2 [n]');
title('Output y_2 [n] = y_1 [n] - 2 * y_1 [n-5]');
```

The output plot of the convolution is shown in figure 5.

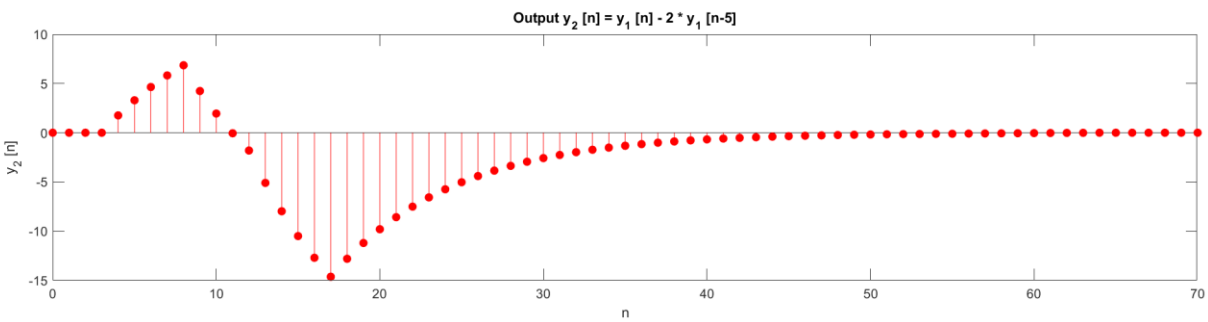

Figure 5: Plot of $y_2[n] = y_1[n] - 2y_1[n]$

MATLAB values and calculation results match.

c) The given input function is

$$x_3[n] = \begin{cases} e^{j(1/3)n} & \text{if } 2 \le n \le 20 \\ 0 & \text{else} \end{cases}$$

By Euler's Formula, the input function can be written as

$$x_3[n] = e^{j\left(\frac{1}{3}\right)n} = e^{j\left(\frac{n}{3}\right)} = \cos\left(\frac{n}{3}\right) + j\sin\left(\frac{n}{3}\right)$$

For $2 \le n \le 20$.

The real and imaginary parts of the input signal are shown in figures 6 and 7, respectively.

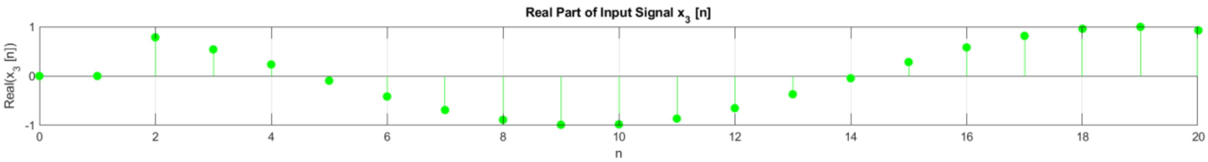

Figure 6: Real Part of x3[n]

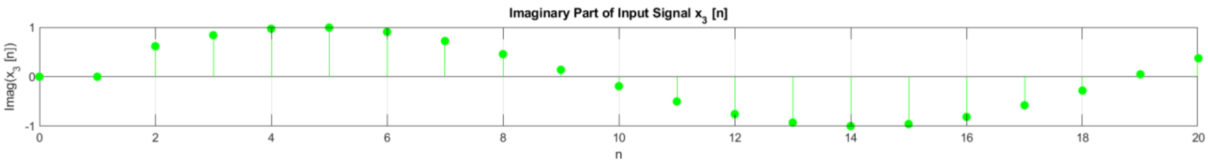

Figure 7: Imaginary Part of x3[n]

By using the associative property of convolution, the output of the system can be written as

$$y_3[n] = x_3[n] * h[n] = \left[x_{3R}[n] + jx_{3I}[n]\right] * h[n]$$

$$= (x_{3R}[n] * h[n]) + j(x_{3I} * h[n])$$

Hence the convolution is

$$y_3[n] = x_3[n] * h[n] = [y_{3R}[n] * h[n]] + j[y_{3I}[n] * h[n]]$$

The boundary for this convolution is 6, since the impulse response has unit step function u[n-4] and the input signal is nonzero for $2 \le n \le 20$.

The real and imaginary parts of the convolution is calculated separately as follows

$$y_{3R}[n] = \sum_{m=2}^{20} \cos\left(\frac{m}{3}\right)\left(\frac{7}{8}\right)^{n-m} u[n-m-4] \text{ for } n \ge 6$$

$$y_{3I}[n] = \sum_{m=2}^{20} \sin\left(\frac{m}{3}\right)\left(\frac{7}{8}\right)^{n-m} u[n-m-4] \text{ for } n \ge 6$$

Else $y_3[n] = 0$.

The definition of the input signal, plotting process of input signal and output signal in MATLAB is implemented as follows

```
% Defining the input signal x3[n] using Euler's formula
x3_real = cos((1/3) * n) .* (n >= 2 & n <= 20); % Real part of x3[n]
x3_imag = sin((1/3) * n) .* (n >= 2 & n <= 20); % Imaginary part of x3[n]


y3_real = manual_convolution(x3_real, h);        % Convolution of real part
y3_imag = manual_convolution(x3_imag, h);        % Convolution of imaginary
part


% Plot
figure;


subplot(4, 1, 1);                    % First subplot for real part of x3[n]
stem(n, x3_real, 'g', 'filled');
xlabel('n');
ylabel('Real(x_3 [n])');
title('Real Part of Input Signal x_3 [n]');
grid on;


subplot(4, 1, 2);                    % Second subplot for imaginary part of
x3[n]
stem(n, x3_imag, 'g', 'filled');
xlabel('n');
ylabel('Imag(x_3 [n])');
title('Imaginary Part of Input Signal x_3 [n]');
grid on;


% Real Part Convolution Plot
n_output_real = 0:(length(y3_real)-1);
subplot(4, 1, 3);                    % Third subplot for the convolution of
real part
stem(n_output_real, y3_real, 'g', 'filled');
xlabel('n');
ylabel('Real(y_3 [n])');
title('Convolution Result of Real Part');
grid on;


% Imaginary Part Convolution Plot
n_output_imag = 0:(length(y3_imag)-1);
subplot(4, 1, 4);                    % Fourth subplot for the convolution of
imaginary part
stem(n_output_imag, y3_imag, 'g', 'filled');
xlabel('n');
ylabel('Imag(y_3 [n])');
```

```
title('Convolution Result of Imaginary Part');
grid on;
```

Convolution plots of the real and imaginary parts are shown in figures 8 and 9, respectively.

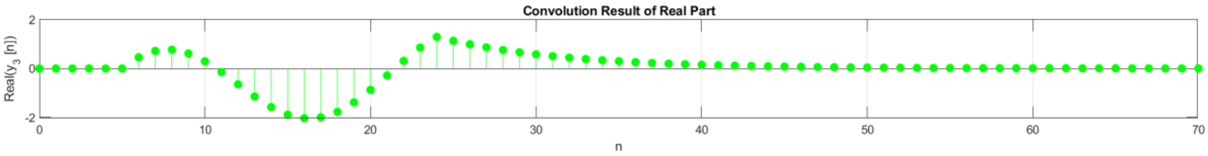

Figure 8: Plot of $y_{3R}[n]$

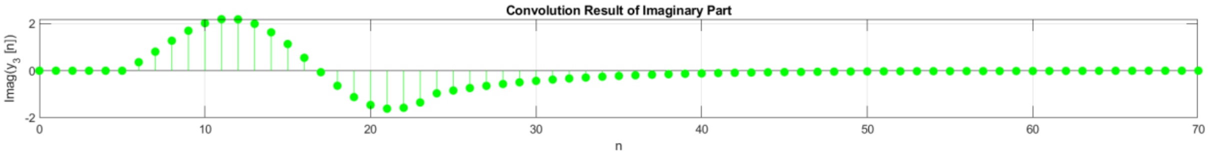

Figure 9: Plot of $y_{3I}[n]$

MATLAB values and calculation results match.

d) The given input function is

$$\mathrm{x}_4[n] = \begin{cases} -3sin[(1/3)n]\ if\ 2 \leq n \leq 20 \\ 0 \text{ else} \end{cases}$$

The plot of the input function is shown in figure 10.

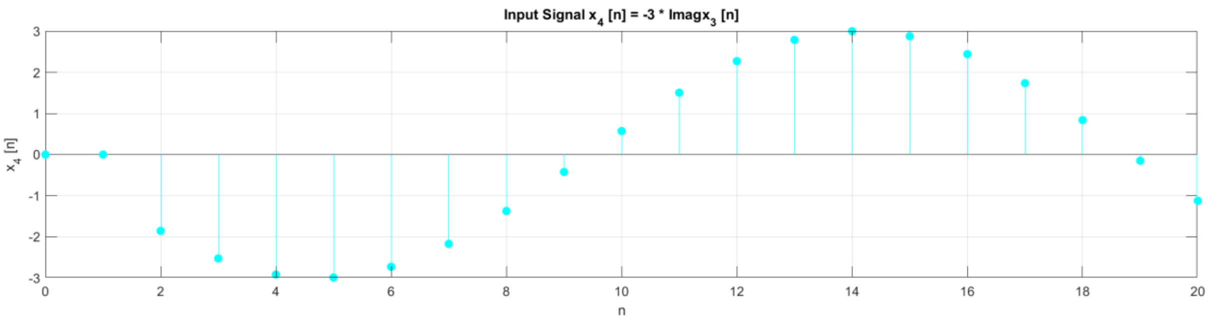

Figure 10: Plot of $\mathrm{x}_4[\mathrm{n}]$

From the previous input function $\mathrm{x}_3[\mathrm{n}]$, $\mathrm{x}_4[\mathrm{n}] = -3\mathrm{Im}\{\mathrm{x}_3[\mathrm{n}]\}$. Hence the convolution of the system can be written as

$$y_4[n] = x_4[n] * h[n] = \left[\text{-}3x_{3I}[n]\right] * h[n] = -3y_3[n]$$

This means that the result is -3 times of the amplitude of the previous convolution. Hence the boundary for previous convolution holds.

The convolution calculation is

$$y_4[n] = \text{-}3y_{3R}[n] = -3\sum_{m=2}^{20} \cos\left(\frac{m}{3}\right)\left(\frac{7}{8}\right)^{n-m} u[n-m-4] \text{ for } n \geq 6$$

Else $y_4[n] = 0$.

The definition of the input signal, plotting process of input signal and output signal in MATLAB is implemented as follows

```
% y4[n] = -3 * y3_imag[n]
y4 = -3 * y3_imag;


% Plot the input signal x4[n] = -3 * Im{x3[n]}
figure;
subplot(2, 1, 1);                    % First subplot for input signal x4[n]
x4 = -3 * x3_imag;
stem(n, x4, 'c', 'filled');
xlabel('n');
ylabel('x_4 [n]');
title('Input Signal x_4 [n] = -3 * Imag{x_3 [n]}');
grid on;


% Plot the output y4[n] = -3 * y3_imag[n]
n_output = 0:(length(y4)-1);
subplot(2, 1, 2);                    % Second subplot for output signal y4[n]
stem(n_output, y4, 'c', 'filled');
xlabel('n');
ylabel('y4[n]');
title('Output y_4 [n] = -3 * y_3I [n]');
grid on;
```

Convolution plot of the system is shown in figure 11.

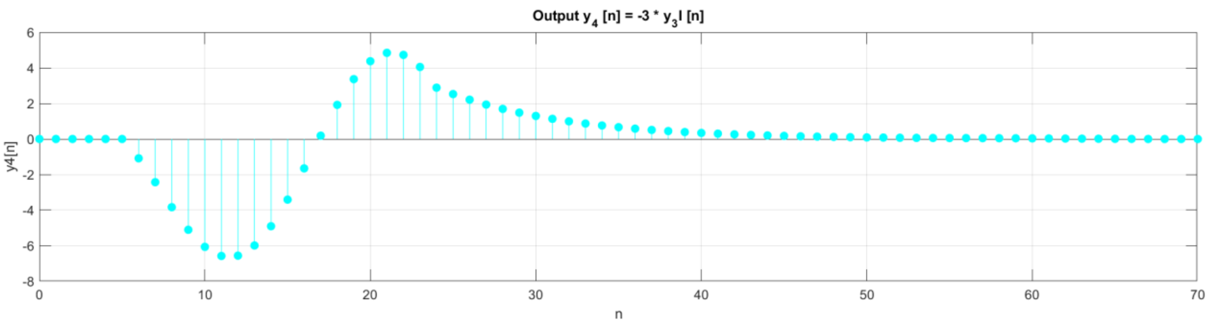

Figure 11: Plot of $y_4[n]$

MATLAB values and calculation results match.

e) The given input signal is

$$x_5[n] = \begin{cases} 2cos[(1/3)n] if 2 \leq n \leq 20 \\ 0 \text{ else} \end{cases}$$

The plot of the input function is shown in figure 12.

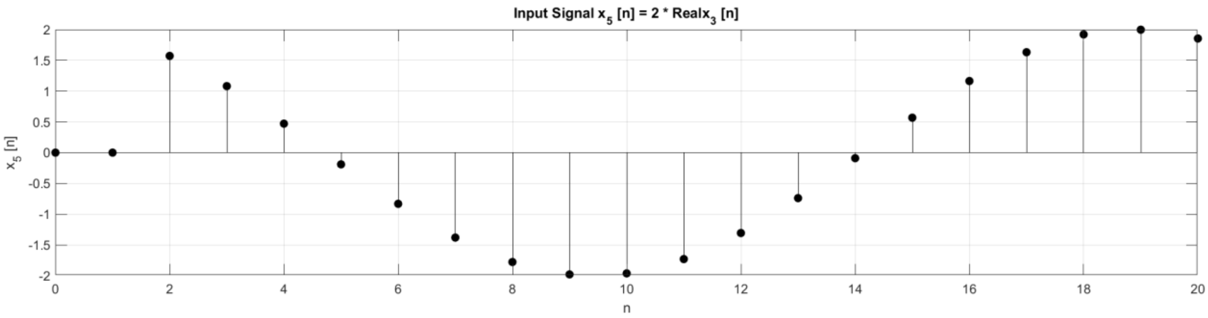

Figure 12: Plot of $x_5[n]$

From the input function $x_3[n]$, $x_5[n] = 2\mathrm{Re}\{x_3[n]\}$. Hence the convolution of the system can be written as

$$y_5[n] = x_5[n] * h[n] = [2x_{3R}[n]] * h[n] = 2y_3[n]$$

This means that the result is 2 times the amplitude of the previous convolution. Hence the boundary for previous convolution holds.

The convolution calculation is

$$y_5[n] = 2y_{3I}[n] = 2\sum_{m=2}^{20} \cos\left(\frac{m}{3}\right)\left(\frac{7}{8}\right)^{n-m} u[n-m-4] \text{ for } n \geq 6$$

Else $y_5[n] = 0$.

The definition of the input signal, plotting process of input signal and output signal in MATLAB is implemented as follows

```
y5 = 2 * y3_real;


% Plot the input signal x5[n] = 2 * Re{x3[n]}
figure;
subplot(2, 1, 1);                    % First subplot for input signal x5[n]
x5 = 2 * x3_real;                    % x5[n] is defined as 2 * real part of
x3[n]
stem(n, x5, 'k', 'filled');
xlabel('n');
ylabel('x_5 [n]');
title('Input Signal x_5 [n] = 2 * Real{x_3 [n]}');
grid on;


% Plot the output y5[n] = 2 * y3_real[n]
n_output = 0:(length(y5)-1);
subplot(2, 1, 2);                    % Second subplot for output signal y5[n]
stem(n_output, y5, 'k', 'filled');
xlabel('n');
```

```
ylabel('y_5 [n]');
title('Output y_5 [n] = 2y_3R [n]');
grid on;
```

Convolution plot of the system is shown in figure 13.

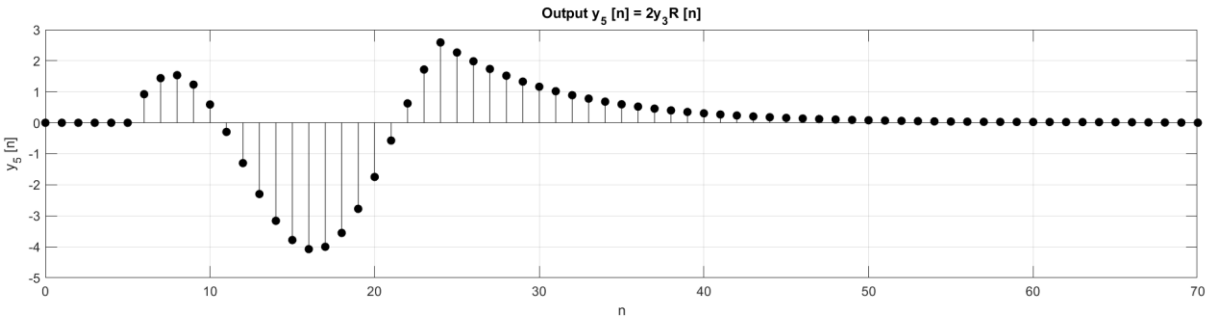

Figure 13: Plot of $y_5[n]$

MATLAB values and calculation results match.

f) The given input function is

$$x_6[n] = x_1[n] + j2x_2[n]$$

The plot of the real and imaginary parts of the input signal are shown in figure 14 and 15, respectively.

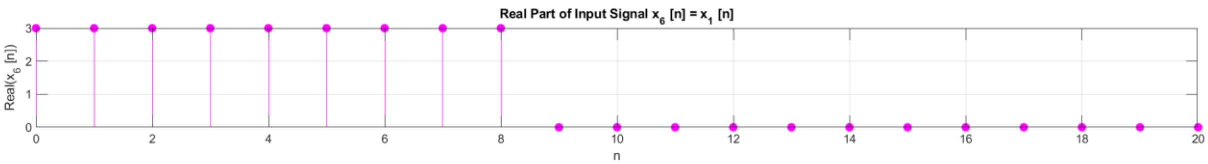

Figure 14: Real Part of $x_4[n]$

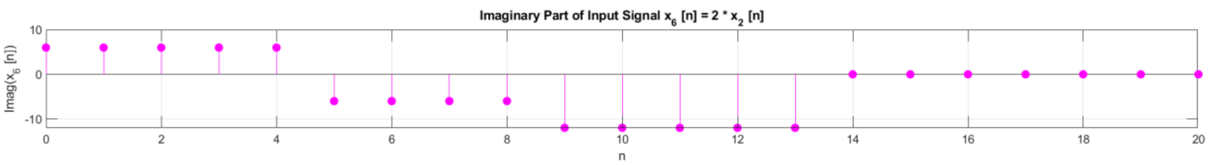

Figure 15: Imaginary Part of $x_4[n]$

By using the associative property of the convolution, the system can be written as

$$y_6[n] = x_6[n] * h[n] = (x_1[n] + j2x_2[n]) * h[n] = (x_1[n] * h[n]) + j2(x_2[n] * h[n])$$

$$y_6[n] \;=\; y_1[n] \;+\; \mathrm{j}2\; y_2[n]$$

Hence the convolution of the system is calculated as

$$y_6[n] = 0 \; if \; n < 4$$

$$y_6[n] = 3\sum_{m=4}^{n}\left(\frac{7}{8}\right)^m \text{ if } 4 \le n < 9$$

$$y_6[n] = 3\sum_{m=4}^{n}\left(\frac{7}{8}\right)^m + \mathrm{j}2.3\sum_{m=4}^{n\text{-}5}\left(\frac{7}{8}\right)^m \text{ if } 9 \le n < 12$$

$$y_6[n] = 3\sum_{m=n\text{-}8}^{n}\left(\frac{7}{8}\right)^m + \mathrm{j}\,3.2\sum_{m=4}^{n\text{-}5}\left(\frac{7}{8}\right)^m \text{ if } 12 < n \le 17$$

$$y_6[n] = 3\sum_{m=n\text{-}8}^{n}\left(\frac{7}{8}\right)^m + \mathrm{j}\,3.2\sum_{m=n\text{-}13}^{n\text{-}5}\left(\frac{7}{8}\right)^m \text{ if n} > 17$$

Else $y_6[n] = 0$.

The definition of the input signal, plotting process of input signal and output signal in MATLAB is implemented as follows

```
x6_real = x1;          % Real part of x6[n] is x1[n]
x6_imag = 2 * x2;      % Imaginary part of x6[n] is 2 * x2[n]


y6 = y1 + 1i * 2 * y2;


figure;


subplot(4, 1, 1);                   % First subplot for the real part of x6[n]
stem(n, x6_real, 'm', 'filled');
xlabel('n');
ylabel('Real(x_6 [n])');
title('Real Part of Input Signal x_6 [n] = x_1 [n]');
grid on;


subplot(4, 1, 2);                   % Second subplot for the imaginary part of
x6[n]
stem(n, x6_imag, 'm', 'filled');
xlabel('n');
ylabel('Imag(x_6 [n])');
title('Imaginary Part of Input Signal x_6 [n] = 2 * x_2 [n]');
grid on;




subplot(4, 1, 3);                   % Third subplot for the real part of the
output y6[n]
n_output = 0:(length(y6)-1);
```

```
stem(n_output, real(y6), 'm', 'filled');
xlabel('n');
ylabel('Real(y_6 [n])');
title('Real Part of Output y_6 [n]');
grid on;


subplot(4, 1, 4);                    % Fourth subplot for the imaginary part of
the output y6[n]
stem(n_output, imag(y6), 'm', 'filled');
xlabel('n');
ylabel('Imag(y_6 [n])');
title('Imaginary Part of Output y_6 [n]');
grid on;
```

Convolution plots of the real and imaginary parts are shown in figure 16 and 17.

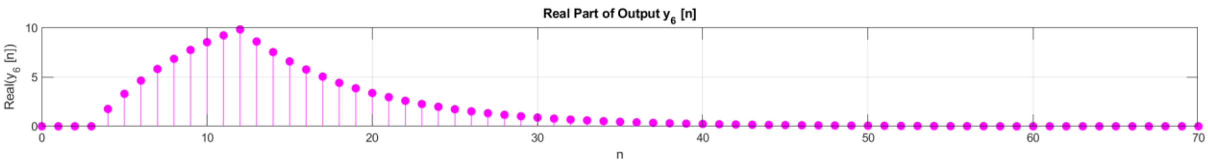

Figure 16: Real Part of $y_6[n]$

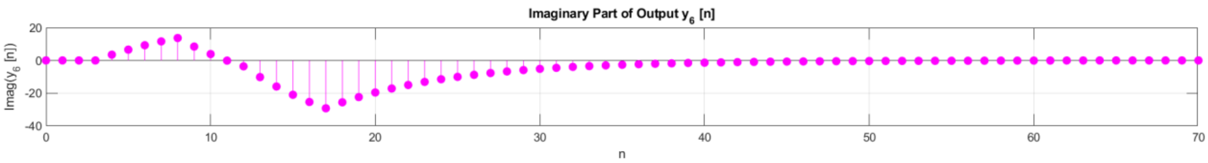

Figure 17: Imaginary Part of $y_6[n]$

MATLAB values and calculation results match.

Conclusion

This assignment focused on manually computing convolutions for a discrete Linear Time-Invariant (LTI) system using explicit additions and multiplications, avoiding built-in MATLAB functions. The system's impulse response was analyzed, and the outputs were computed for various input signals, both real and complex.

The convolution process was applied using analytical methods, with results confirmed by numerical computations. Convolution properties, such as linearity and time-invariance, were used to simplify the calculations, especially for signals involving shifts and scaling. Additionally, the system's causality and stability were verified, showing that the system is causal and BIBO stable due to the nature of the impulse response.

This assignment emphasized the importance of understanding the convolution operation and its role in signal processing. By computing convolutions manually, a deeper insight into how LTI systems respond to different inputs was gained, reinforcing both theoretical and practical knowledge in the field.