Signals Systems and Communication Lab

Laboratory report submitted for the partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

by

Mohit Akhouri - 19ucc023

Course Coordinator
Dr. Navneet Upadhyay

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

February 2021

Contents

Chapter *1*

Experiment - 1

1.1 Aim of the experiment

1. To get familiarity with basic commands in MATLAB.
2. To explore the connection between system impulse response and the solution of linear ordinary constant coefficient differential equation.
3. To understand and implement convolution routine for discrete time finite length sequences.

1.2 Software Used

MAT LAB

1.3 Theory

1.3.1 About Matlab :

MATLAB (Matrix Laboratory) is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

Figure 1.1 MATLAB Icon

1.3.2 About the function plot(y,x) :

plot(y,x) function takes as argument two sequences y and x and plots them as a graph. We can use other options also like **grid on** to display the grid , **title** to give title to graph and for giving labels to X and Y axis , use **xlabel** and **ylabel** functions.

1.3.3 About Discrete Convolution :

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output a system produces for a given input signal. It can be shown that a linear time invariant system is completely characterized by its impulse response. The sifting property of the discrete time impulse function tells us that the input signal to a system can be represented as a sum of scaled and shifted unit impulses. Thus, by linearity, it would seem reasonable to compute of the output signal as the sum of scaled and shifted unit impulse responses. That is exactly what the operation of convolution accomplishes. Hence, convolution can be used to determine a linear time invariant system's output from knowledge of the input and the impulse response.
The formula for Convolution of two infinite length sequences x and h is :

$$y(m) = \sum_{k=-\infty}^{\infty} x(k).h(m-k) \tag{1.1}$$

In this experiment, we have been given two finite length sequences x and h and we have to write a myconv(x,h) function to calculate the convolution of x and h. So the equation 1.1 transforms to the below equation :

$$y(m) = \sum_{k=1}^{n_x} x(k).h(m-k+1) \tag{1.2}$$

where $1 <= m <= ny$ and **nx** and **ny** are length of sequences x and output sequence y and **nh** is length of sequence h.
$ny = nx + nh - 1$
The above equation is implemented in MATLAB through the concept of **padding with zeros** and **Looping through FOR construct**.

1.4 Code and Results

1.4.1 Exercise 1 : Plotting the signals

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Code for generating plot of x1(t) = (e^-t)cos(2*pi*t)

t = 0 : 0.001 : 2;
a = exp(-t);
b = cos(2*pi*t);
x1 = a.*b;
plot(t,x1);
xlabel('time(t) -> ');
ylabel('x_{1}(t) -> ');
title('Mohit Akhouri-19ucc023 - Exercise1\_a','x_{1}(t) = e^{-
t}cos(2\pit)');
grid on;
```

Figure 1.2 Code for Exercise 1a - Plotting x_1(t)

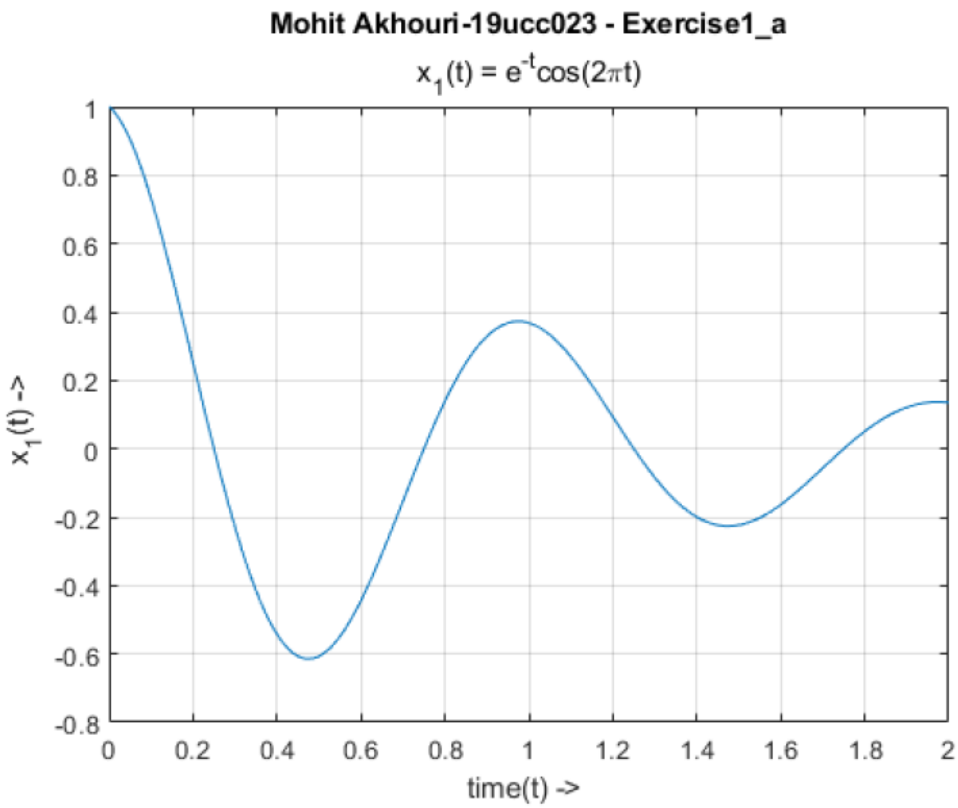

Figure 1.3 Graph for Exercise 1a - Plotting x_1(t)

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Code for generating plot of x2(t) = 1 + 1.5cos(2*pi*t) -
 0.6cos(4*pi*t)

t = 0 : 0.001 : 2;
x2 = 1 + 1.5*cos(2*pi*t) - 0.6*cos(4*pi*t);
plot(t,x2);
xlabel('time(t) -> ');
ylabel('x_{2}(t) -> ');
title('Mohit Akhouri-19ucc023 - Exercise1\_b','x_{2}(t) = 1 +
 1.5cos(2\pit) - 0.6cos(4\pit)');
grid on;
```

Figure 1.4 Code for Exercise 1b - Plotting x_2(t)

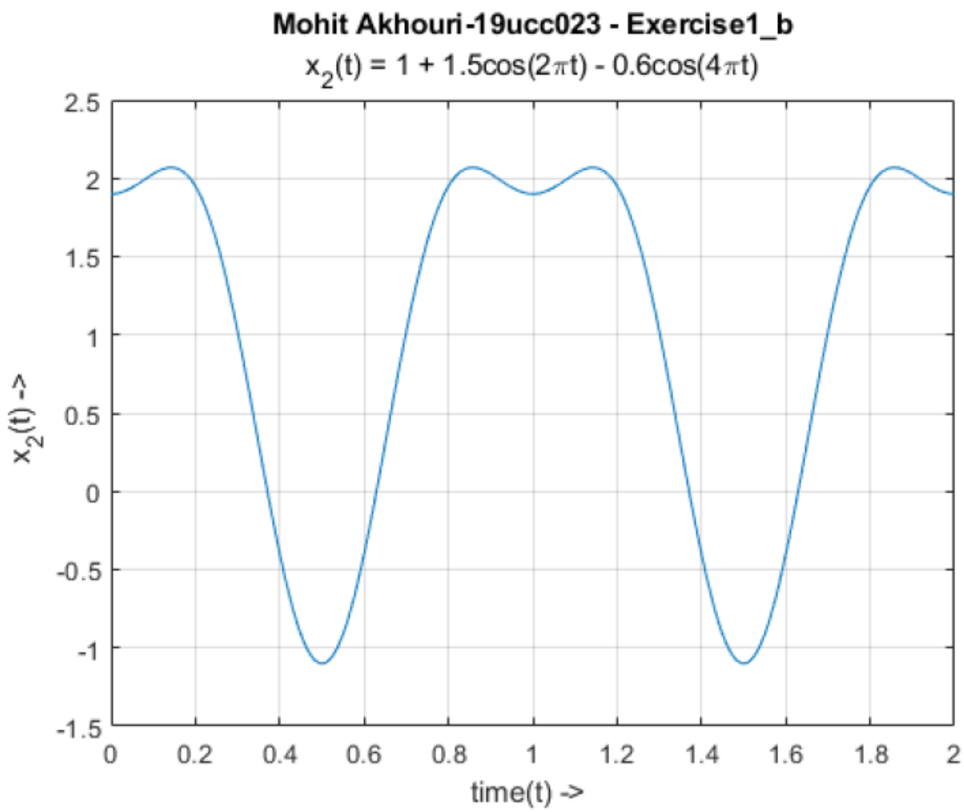

Figure 1.5 Graph for Exercise 1b - Plotting x_2(t)

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Code for generating plot of x3(t) = |cos(2*pi*t)|

t = 0 : 0.001 : 2;
x3 = abs(cos(2*pi*t));
plot(t,x3);
xlabel('time(t) -> ');
ylabel('x_{3}(t) -> ');
title('Mohit Akhouri-19ucc023 - Exercise1\_c','x_{3}(t) = |
cos(2\pit)|');
grid on;
```

Figure 1.6 Code for Exercise 1c - Plotting x_3(t)

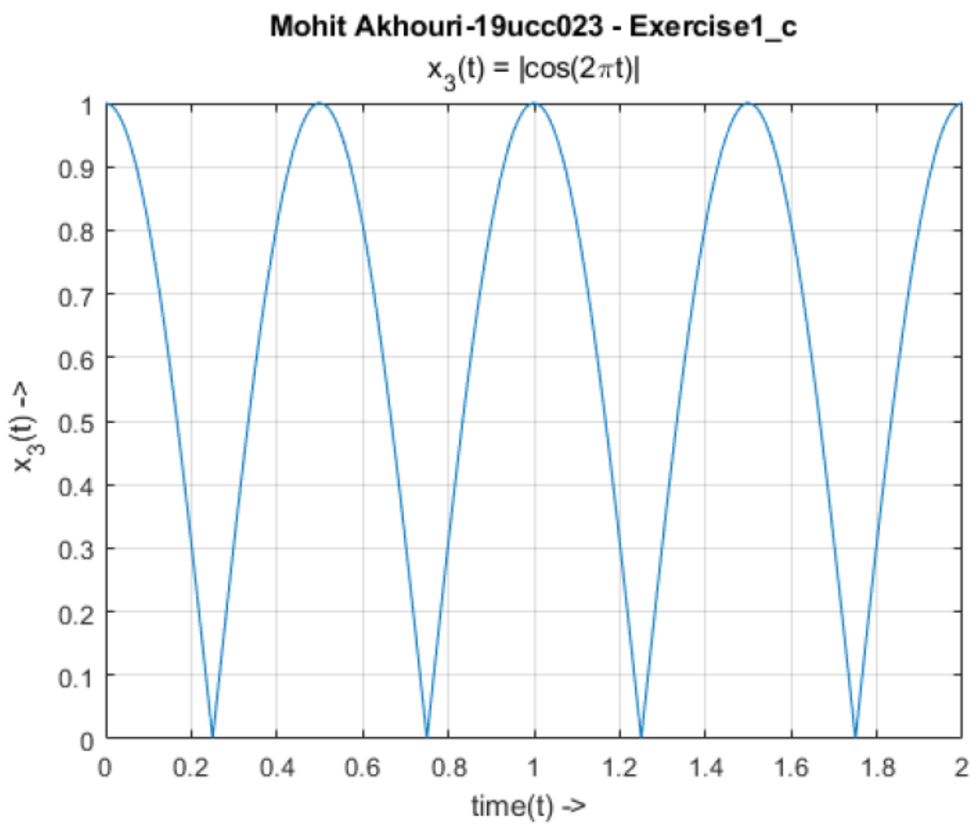

Figure 1.7 Graph for Exercise 1c - Plotting x_3(t)

1.4.2 Exercise 2 : Computing convolution of two finite length sequences

```
function  myconv(x,h)
% Summary of the function myconv(x,h) is :
%   This function takes as input two finite length sequences 'x' and
 'h'
%   and calculate the convulution of x and h

% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

nx = length(x); % length of sequence x
nh = length(h); % length of sequence h
ny = nx+nh-1; % length of output convulated sequence y
y = zeros(1,ny); % output convulated sequence y
h = [h zeros(1,ny-nh+1)]; % padding h with extra zeros

for i=1:ny
    sum=0; %to calculate sum at each iteration
    for j=1:nx
        if (i-j+1>0)
            sum = sum + (x(j)*h(i-j+1));
        end
    end
    y(i)=sum;
end
stem(y);
xlabel('time(t) ->');
ylabel('x(n) conv h(n) ->');
title('Mohit Akhouri-19ucc023 - Exercise2','Discrete convolution');
grid on;
end
```

Published with MATLAB® R2020b

Figure 1.8 Function myconv(x,h) designed for Experiment 2

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Code for using myconv(x,h) function to calculate the convulution
% of two finite length sequences

x = [1 1 1 1 -1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1];
h1 = [1 1];
myconv(x,h1);
```

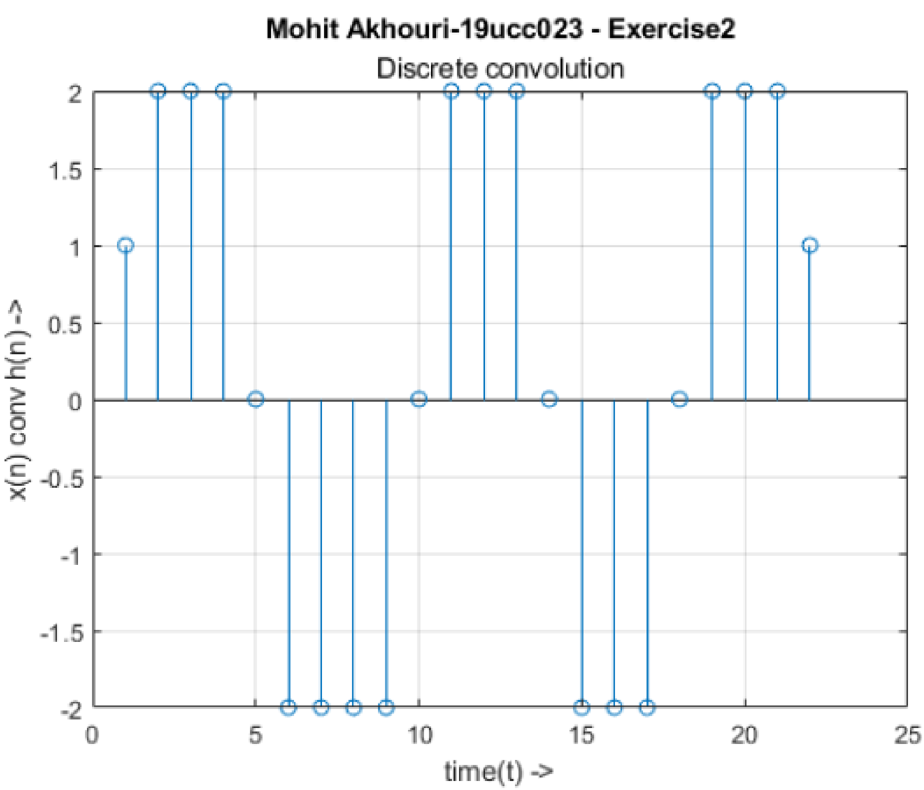

Figure 1.9 Main code and graph for convolution x and h1

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Code for using myconv(x,h) function to calculate the convulution
% of two finite length sequences

x = [1 1 1 1 -1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1];
h2 = [1 -1];
myconv(x,h2);
```

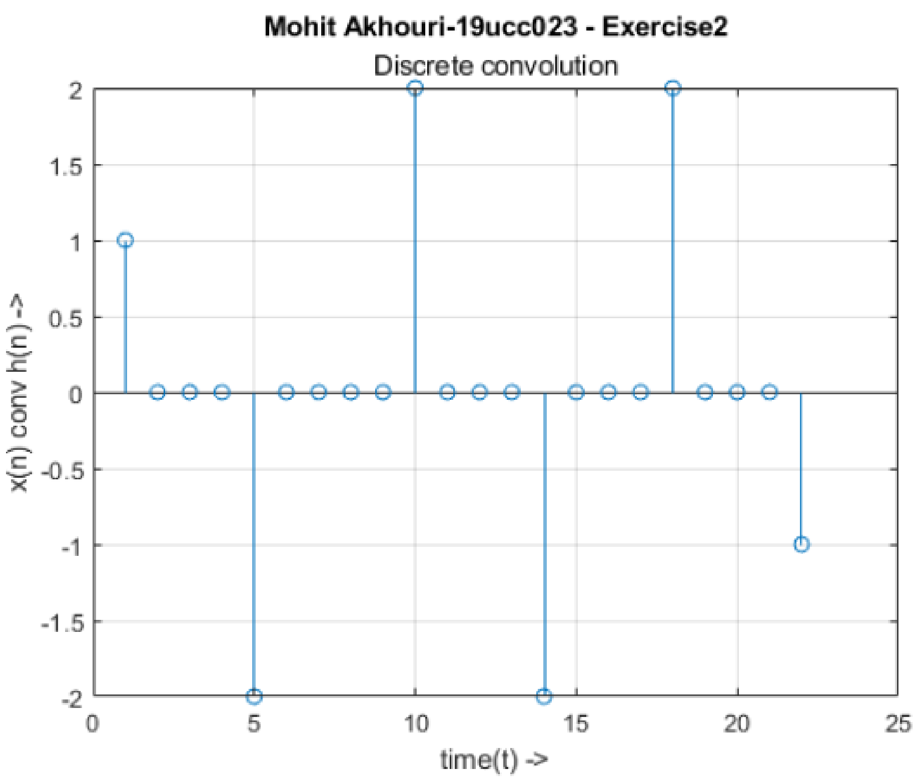

Figure 1.10 Main code and graph for convolution x and h2

1.4.3 Exercise 3 : Solving the Difference equation of RC Filter

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% MATLAB code to solve the difference equation with
% input specified as x(t)

RC=1; %defining time constant = 1
T=0.001;
t=0:T:2-T; %defining range for t
a=1/(1+(RC/T)); %defining 'a'
b=-1/(1+(T/RC)); %defining 'b'
x=(t>=0)+(-1*(t>=1));
y=zeros(1,2000); %filling y with zeros
for k=1:2000 %implementing the algorithm using for loop
    if k==1
        y(k)=a*x(1);
    else
        y(k)=a*x(k)-b*y(k-1);
    end
end
plot(t,y);
ylabel('y(k) ->');
xlabel('time(t) ->');
title('Mohit Akhouri-19ucc023 - Exercise3','Solving Difference
 Equation');
grid on;
```

Figure 1.11 Code for Exercise 3

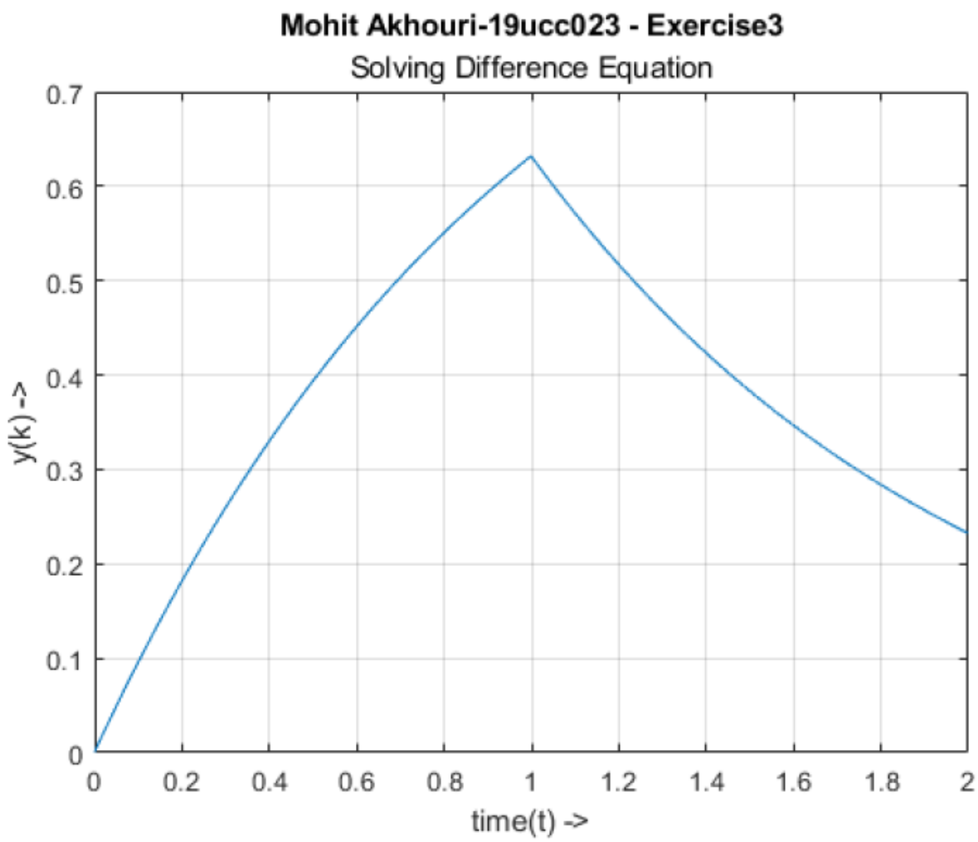

Figure 1.12 Graph of Exercise 3

1.4.4 Exercise 4 : Calculation of Output of RC Filter

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Calculation output of LTI system

RC=1; %defining time constant = 1
T=0.001;
t=0:T:2-T; %defining range for t
a=1/(1+(RC/T)); %defining 'a'
b=-1/(1+(T/RC)); %defining 'b'
x=[1 zeros(1,1999)]; %defining impulse response del(t)
y=zeros(1,2000); %initializing y
for k=1:2000 %algorithm to find convolution
    if k==1
        y(k)=a*x(1);
    else
        y(k)=a*x(k)-b*y(k-1);
    end
end
%using conv function on h and x
h=y;
x=(t>=0)+(-1*(t>=1));
y1=conv(x,h);
stem(y1);
xlabel('time(t) ->');
ylabel('x(n) conv h(n) ->');
title('Mohit Akhouri-19ucc023 - Exercise4','Output of the LTI system
 (RC Filter)');
grid on;
```

Figure 1.13 Code for Exercise 4

Figure 1.14 Graph of Exercise 4

1.5 Conclusion

In this experiment, We have learnt MATLAB basics and tried to implement basic plots in MATLAB. We have also implemented myconv(x,h) to calculate convolution of two finite length sequences using **for loop** and **concept of padding with zeros** . In exercise 3 and 4, We have analysed the LTI systems and implemented a code to solve difference equation of RC Filter and finding output of RC Filter using convolution of impulse signal and input signal.