

# **Review of Convolution**

Lab Exercises on January 2, 2022

*Department of Electronics and Computer Engineering*

*Pulchowk Campus, Lalitpur*

**Ashlesh Pandey**

**PUL074BEX007**

## Contents

<b>List of Figures</b>	<b>ii</b>
<b>Listings</b>	<b>iii</b>
<b>1 Objectives</b>	<b>1</b>
<b>2 Functions Used</b>	<b>1</b>
<b>3 Packages Used</b>	<b>1</b>
<b>4 Background Theory</b>	<b>1</b>
<b>5 Lab Exercises</b>	<b>4</b>
5.1 Convolution using basic convolution formula . . . . .	4
5.2 Convolution using conv and convolve function . . . . .	6
5.2.1 DT convolution . . . . .	7
5.2.2 CT convolution . . . . .	9
5.3 LTI system response in DT . . . . .	11
5.4 LTI system response in CT with impulse response as sinc functions . . . .	13
<b>6 Discussion and Conclusion</b>	<b>16</b>

## List of Figures

1	Example convolution using graphical, tabular and conv . . . . .	4
2	Observation of convolution using basic convolution formula (Matlab) . . .	5
3	Observation of convolution using basic convolution formula (Python) . . .	6
4	Observation of DT convolution using conv function (Matlab) . . . . .	8
5	Observation of DT convolution using convolve function (Python) . . . . .	9
6	Observation of CT convolution using conv function (Matlab) . . . . .	10
7	Observation of CT convolution using convolve function (Python) . . . . .	11
8	Observation of response of LTI system (Matlab) . . . . .	12
9	Observation of response of LTI system (Python) . . . . .	13
10	Observation of response of LTI system with sinc function as impulse response (Matlab) . . . . .	14
11	Observation of response of LTI system with sinc function as impulse response (Python) . . . . .	15

## Listings

1	Matlab function for folding . . . . .	2
2	Matlab function for shifting . . . . .	2
3	Matlab function for multiplication . . . . .	2
4	Matlab script to find convolution of example using graphical, tabular and conv	3
5	Matlab script for convolution using basic convolution formula . . . . .	5
6	Python script for convolution using basic convolution formula . . . . .	6
7	Matlab script for DT convolution using conv function . . . . .	7
8	Python script for DT convolution using convolve function . . . . .	8
9	Matlab script for CT convolution using conv function . . . . .	10
10	Python script for CT convolution using convolve function . . . . .	11
11	Matlab script for finding response of LTI system . . . . .	12
12	Python script for finding response of LTI system . . . . .	13
13	Matlab script for finding response of LTI system with sinc function as im- pulse response . . . . .	14
14	Python script for finding response of LTI system with sinc function as im- pulse response . . . . .	15

## 1 Objectives

- To be able to perform convolution of two given signal using basic formula.
- To be able to perform convolution of two given signals using Python (NumPy) functions.

## 2 Functions Used

- `conv`: `conv` is an in-built Matlab function to find convolution of input arrays.
- `sinc`: `sinc` is an in-built Matlab function to return the *sinc* response of the input.

## 3 Packages Used

- NumPy: NumPy is a fundamental package for scientific computing with Python.
- Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

## 4 Background Theory

The output of any Linear Time Invariant (LTI) system is some sort of operation between input and system response; the operation is nothing but convolution, denoted by symbol  $\otimes$ , and defined as,

$$y(t) = x(t) \otimes h(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau \quad (1)$$

$$y[n] = x[n] \otimes h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n - k] \quad (2)$$

Equation 1 is known as convolution integral and is defined for continuous time signals where as Equation 2 is known as convolution sum and is defined for discrete time signals.

For a causal LTI system, the convolution sum is given by,

$$y[n] = \int_{k=0}^n x[k]h[n-k] \quad (3)$$

The process of computing the convolution between  $x[k]$  and  $h[k]$  involves the following four steps:

**1. Folding:** Fold  $h[k]$  about  $k = 0$  to obtain  $h[-k]$ .

```

1 function [y,n]=sigfold(x,n)
2     y=flipr(x);
3     n=-flipr(n);
4 end

```

Listing 1: Matlab function for folding

**2. Shifting:** Shift  $h[-k]$  by  $n_0$  to the right (left if  $n_0$  is positive (negative), to obtain  $h[n_0-k]$ .

```

1 function [y,n]=sigshift_m(x,m,n0)
2     n=m+n0;
3     y=x;
4 end

```

Listing 2: Matlab function for shifting

**3. Multiplication:** Multiply  $x[k]$  by  $h[n_0-k]$  to obtain the product sequence  $V_{n_0}[k]$  where,  $V_{n_0}[k] = x[k].h[n_0-k]$ .

```

1 function [y,n]=sigmulti(x1,n1,x2,n2
2     n=min(min(n1),min(n2)):max(max
3     (n1),max(n2));
4     y1=zeros(1,length(n));
5     y1(find((n>=min(n1))&(n<=max(
6     n1))==1))=x1;
7     y2(find((n>=min(n2))&(n<=max(
8     n2))==1))=x2;
9     y=y1.*y2;
10    end

```

Listing 3: Matlab function for multiplication

**4. Summation:** Sum all the values of the product sequence  $V_{n_0}$  to obtain the value of the output at times  $n = n_0$ .

**Example:** *Convolution of signals  $x[n]$  and  $h[n]$  by graphical, tabular and conv function.*

```

1  x=[1,0,-1,1,2,1];
2  n1=[-2,-1,0,1,2,3];
3  h=[1,1,1,1,1];
4  n2=[0,1,2,3,4];
5  nmin=min(min(n1),min(n2));
6  nx=length(x);
7  nh=length(h);
8  n=nmin:nx+nh+nmin-2;
9  y_graphical=zeros(1,length(n));
10 [x_folded,n_folded]=sigfold(x,n1);
11 temp=1;
12 for i=n
13     [x_shift,n_shift]=sigshift_m(
14         x_folded,n_folded,i);
15     [x_multi,n_multi]=sigmulti(
16         x_shift,n_shift,h,n2);
17     for j=1:length(n_multi)
18         y_graphical(temp)=
19             y_graphical(temp)+x_multi(j);
20     end
21     temp=temp+1;
22 end
23 y_tabular=zeros(1,length(n));
24 prod=x'.*h;
25 for i = 1:nx
26     y_tabular(i:i+nh-1)=y_tabular(
27         i:i+nh-1)+prod(i,:);
28 end
29 y_conv=conv(x,h);
30 l= tiledlayout(3,1);
31 title(l,{'Example convolution from
32         labsheet','(PUL074BEX007)'})
33 xlabel(l,'n')
34 ylabel(l,'Value')
35 nexttile
36 stem(n,y_graphical,'Linewidth'
37     ,1.5)
38 title('Graphical method')
39 nexttile
40 stem(n,y_tabular,'Linewidth',1.5)
41 title('Tabular method')
42 nexttile
43 stem(n,y_conv,'Linewidth',1.5)
44 title('Conv function')
45 print(' ../Figures/
46         lab_3_example_all','-depsc')

```

Listing 4: Matlab script to find convolution of example using graphical, tabular and conv

Example convolution from labsheet  
(PUL074BEX007)

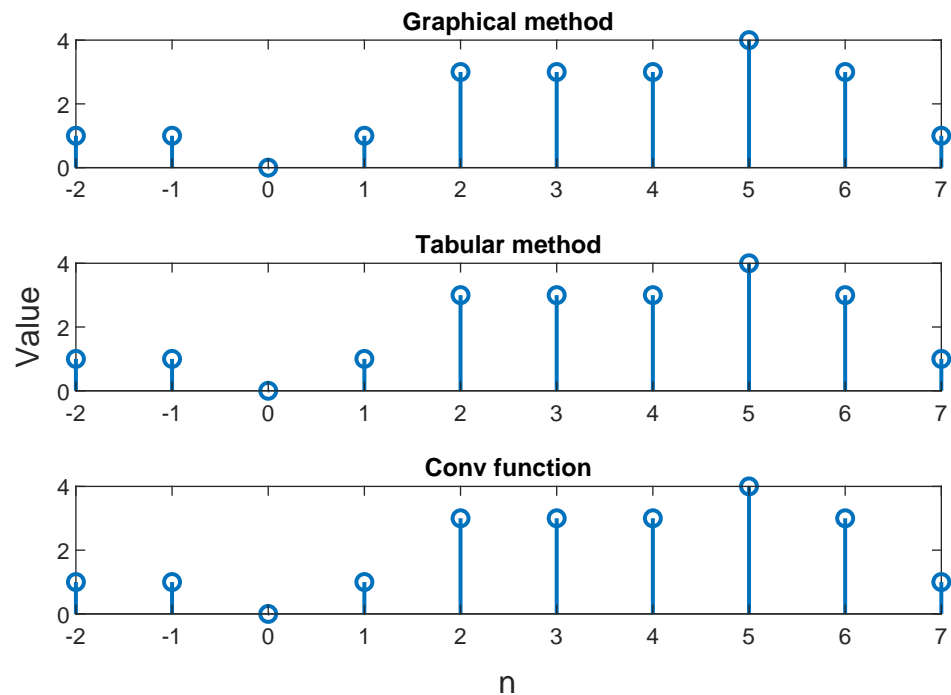


Figure 1: Example convolution using graphical, tabular and conv

## 5 Lab Exercises

### Problem 1

Find the convolution result of the following signal using basic convolution formula.

$$X1(n1)=[1,1,1,1,1]$$

$$n1=[-2,-1,0,1,2]$$

$$X2(n2)=[1,0,0,0,0,0,0,0,0,0]$$

$$n2=[-4,-3,-2,-1,0,1,2,3,4,5]$$

$X2$  is a periodic signal.

$$Y2=X1*X2$$



```

1 x=[1,1,1,1,1];
2 n1=[-2,-1,0,1,2];
3 h=[1,0,0,0,0,0,0,0,0,0];
4 n2=[-4,-3,-2,-1,0,1,2,3,4,5];
5 nmin=min(min(n1),min(n2));
6 nx=length(x);
7 nh=length(h);
8 n=nmin:nx+nh+nmin-2;
9 y=zeros(1,length(n));
10 prod=x'.*h;
11 for i = 1:nx
12     y(i:i+nh-1)=y(i:i+nh-1)+prod(i
    ,:);
13 end
14 l=tiledlayout(1,1);
15 title(l,{ 'Convolution using basic
    convolution formula','(
    PUL074BEX007)'});
16 nexttile
17 stem(n,y,'Linewidth',1.5)
18 xlabel('n')
19 ylabel('Value')
20 print('../Figures/lab_3_1_ml','-
    depsc')

```

Listing 5: Matlab script for convolution using basic convolution formula

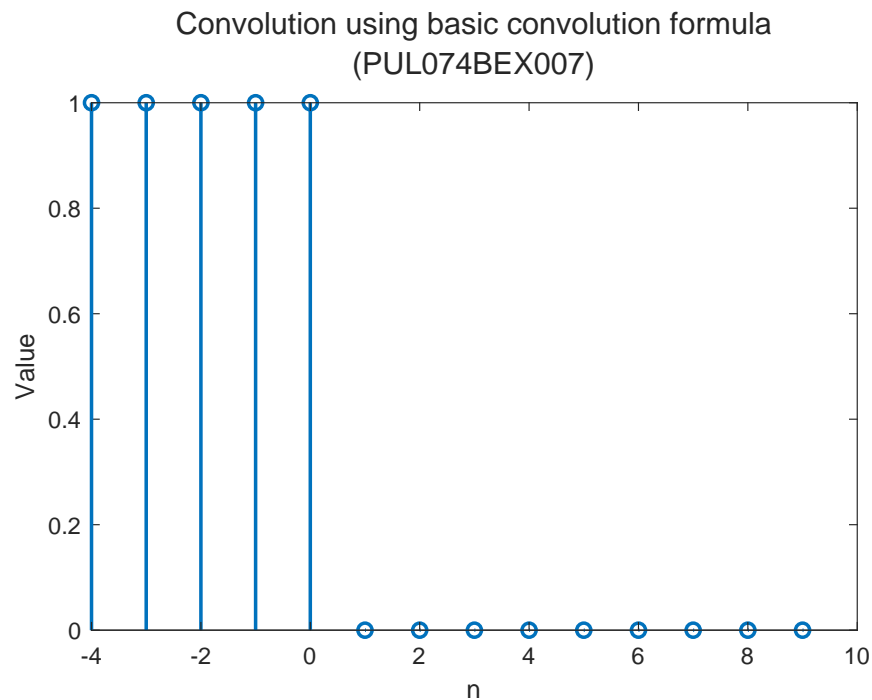


Figure 2: Observation of convolution using basic convolution formula (Matlab)

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3 x1 = np.array([1, 1, 1, 1, 1])
4 n1 = np.arange(start=-2, stop=3)
5 x2 = np.array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0])
6 n2 = np.arange(start=-4, stop=6)
7 n_min = min(np.amin(n1), np.amin(n2))
8 n = np.arange(start=n_min, stop=n_min+n1.size+n2.size-2+1)
9 prod = np.transpose(np.column_stack(x1))*x2
10 y = np.zeros(n.size)
11 for i in range(0, x1.size):
12     y[i:i+x2.size] = y[i:i+x2.size]+prod[i, :]
13 plt.stem(n, y)
14 plt.title('Convolution using basic convolution formula\n(PUL074BEX007)')
15 plt.xlabel('n')
16 plt.ylabel('Value')
17 plt.savefig('../Figures/lab_3_1_py.eps', format='eps')
18 plt.show()

```

Listing 6: Python script for convolution using basic convolution formula

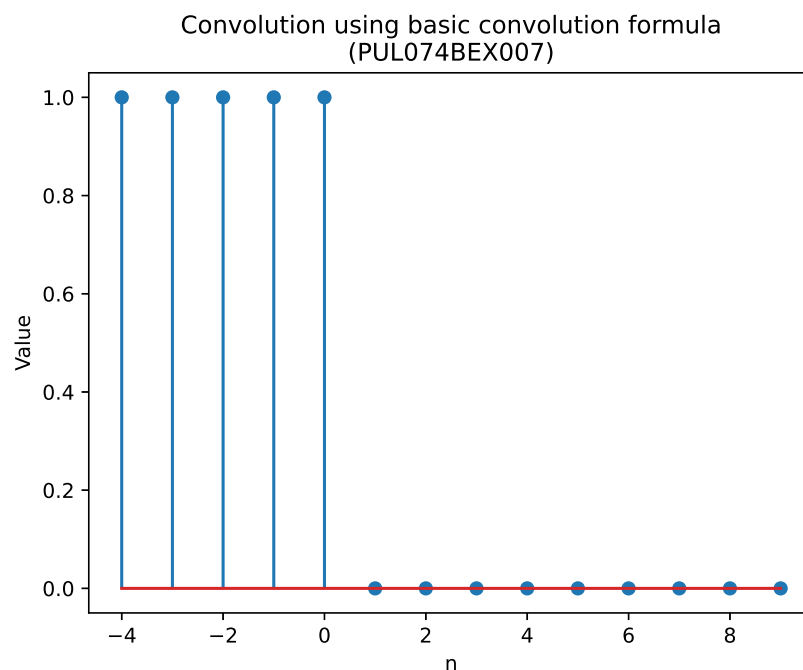


Figure 3: Observation of convolution using basic convolution formula (Python)

## Problem 2

Find the convolution using `conv` and `convolve` function.

a.

$$x[n] = \begin{cases} \frac{1}{3}n & \text{for } 0 \leq n \leq 6 \\ 0 & \text{else} \end{cases} \quad \text{and,} \quad h[n] = \begin{cases} 1 & \text{for } -2 \leq n \leq 2 \\ 0 & \text{else} \end{cases}$$

```

1 n1=0:6;
2 x=n1./3;
3 n2=-2:2;
4 h=n2 >= -2 & n2 <= 2;
5 nmin=min(min(n1),min(n2));
6 nx=length(x);
7 nh=length(h);
8 n=nmin:nx+nh+nmin-2;
9 y=conv(x,h);
10 l= tiledlayout(1,1);

11 title(1,{ 'DT convolution using
            conv function','(PUL074BEX007)'
            })
12 nexttile
13 stem(n,y,'Linewidth',1.5)
14 xlabel('n')
15 ylabel('Value')
16 print(' ../Figures/lab_3_2_a_m1','-
        depsc')

```

Listing 7: Matlab script for DT convolution using `conv` function

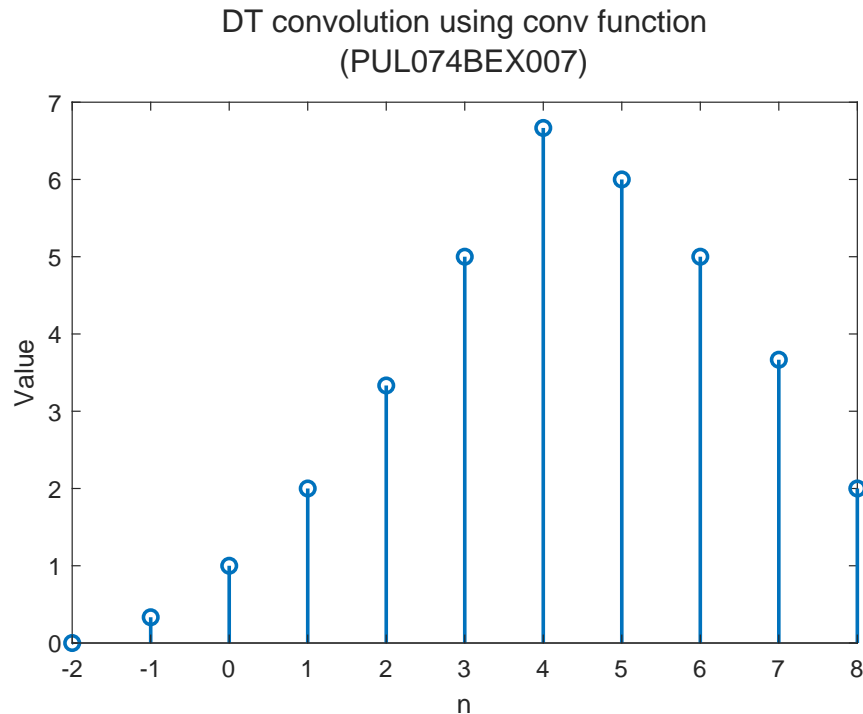


Figure 4: Observation of DT convolution using conv function (Matlab)

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3 n1 = np.arange(start=0, stop=7)
4 x = np.where(np.logical_and(n1 >=
    0, n1 <= 6), n1/3, 0)
5 n2 = np.arange(start=-2, stop=3)
6 h = np.where(np.logical_and(n2 >=
    -2, n2 <= 2), 1, 0)
7 n_min = min(np.amin(n1), np.amin(
    n2))
8 n = np.arange(start=n_min, stop=
    n_min+n1.size+n2.size-2+1)
9 y = np.convolve(x, h)
10 plt.stem(n, y)
11 plt.title('DT convolution using
    convolve function\n(
    PUL074BEX007)')
12 plt.xlabel('n')
13 plt.ylabel('Value')
14 plt.savefig('../Figures/
    lab_3_2_a_py.eps', format='eps'
    )
15 plt.show()

```

Listing 8: Python script for DT convolution using convolve function

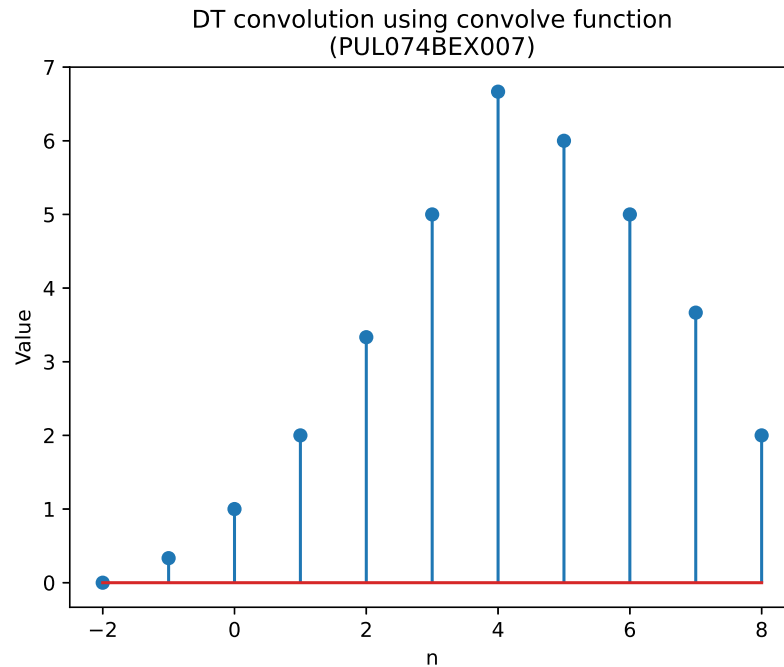


Figure 5: Observation of DT convolution using convolve function (Python)

**b.**

$$x(t) = u(t) \quad \text{and,} \quad h(t) = e^{-at}u(t), \quad \text{where } a > 0$$

```

1 a=1;
2 t=linspace(-5,5,1000);
3 x=t>=0;
4 h=x.*exp(-a.*t);
5 y=conv(x,h,'same');
6 l=tilde(3,1);
7 title(1,{ 'CT convolution using
            conv function','(PUL074BEX007)'
            })
8 xlabel(1,'t')

9 ylabel(1,'Amplitude')
10 nexttile
11 plot(t,x,'Linewidth',1.5)
12 ylabel('x(t)')
13 nexttile
14 plot(t,h,'Linewidth',1.5)
15 ylabel('h(t)')
16 nexttile
17 plot(t,y,'Linewidth',1.5)
18 ylabel('y(t)')

```

```
19 print('../Figures/lab_3_2_b_ml', '- depsc')
```

Listing 9: Matlab script for CT convolution using conv function

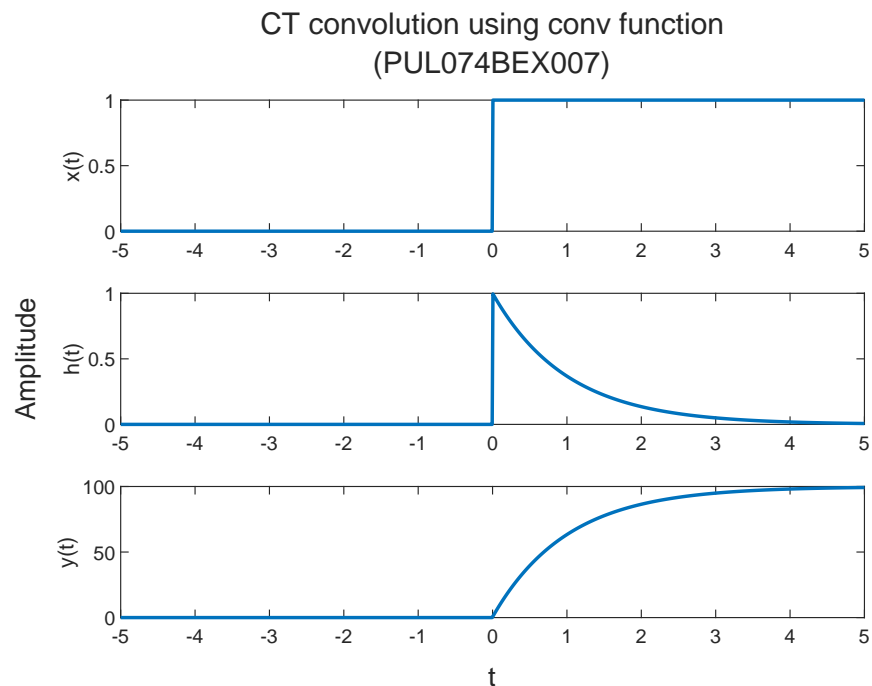


Figure 6: Observation of CT convolution using conv function (Matlab)

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3 a = 1
4 t = np.linspace(-5, 5, 1000)
5 x = np.where(t >= 0, 1, 0)
6 h = x*np.exp(-a*t)
7 y = np.convolve(x, h, mode='same')
8 fig, (ax1, ax2, ax3) = plt.
    subplots(3,1,constrained_layout
    =True)
9 fig.suptitle('CT convolution using
    convolve function\n(
        PUL074BEX007)')
10 fig.supxlabel('t')
11 fig.supylabel('Amplitude')
12 ax1.plot(t, x)
13 ax1.set_ylabel('$x(t)$')
14 ax2.plot(t, h)
15 ax2.set_ylabel('$h(t)$')
16 ax3.plot(t, y)
17 ax3.set_ylabel('$y(t)$')
18 plt.savefig('../Figures/
    lab_3_2_b_py.eps', format='eps'
    )
19 plt.show()
```

Listing 10: Python script for CT convolution using convolve function

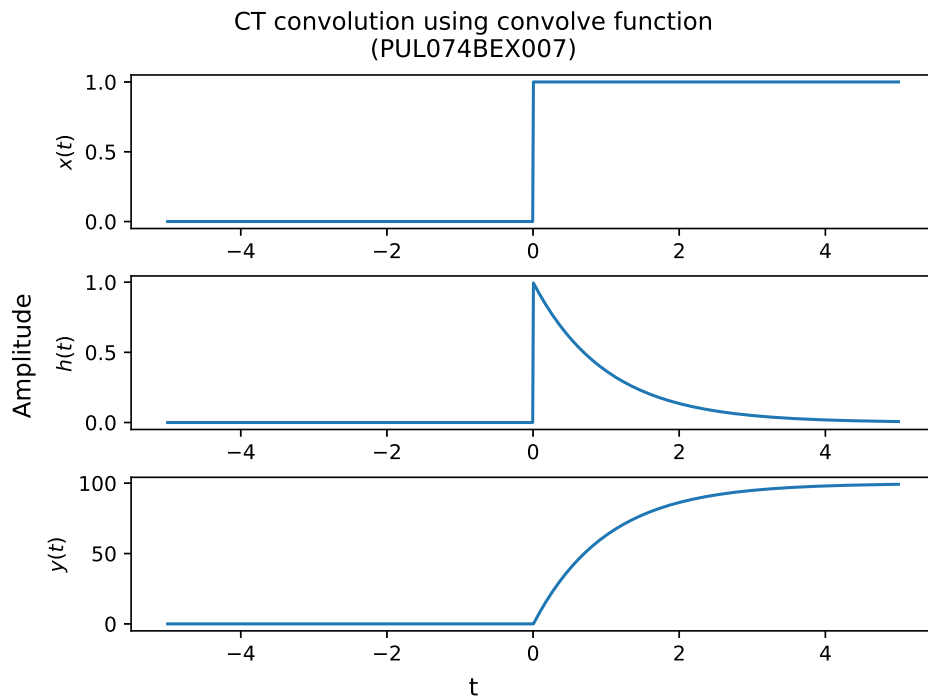


Figure 7: Observation of CT convolution using convolve function (Python)

### Problem 3

Consider two discrete time sequences  $x[n]$  and  $h[n]$  given by

$$x[n] = \begin{cases} 1 & \text{for } 0 \leq n \leq 4 \\ 0 & \text{elsewhere} \end{cases} \quad \text{and,} \quad h[n] = \begin{cases} 2^n & \text{for } 0 \leq n \leq 6 \\ 0 & \text{elsewhere} \end{cases}$$

- Find the response of the LTI system with impulse response  $h[n]$  to input  $x[n]$ .
- Plot the signals and comment on the result.

```

1 n1=0:4;
2 x=n1>=0&n1<=4;
3 n2=0:6;
4 h=(n2>=0&n2<=6).*(2.^n2);
5 nmin=min(min(n1),min(n2));
6 nx=length(x);

```

```

7 nh=length(h);
8 n=nmin:nx+nh+nmin-2;
9 y=conv(x,h);
10 l=tiledlayout(1,1);
11 title(l,{'Response of LTI system',
           '(PUL074BEX007)'});
12 nexttile
13 stem(n,y,'Linewidth',1.5)
14 xlabel('n')
15 ylabel('y[n]')
16 print('../Figures/lab_3_3_m1','-
           depsc')

```

Listing 11: Matlab script for finding response of LTI system

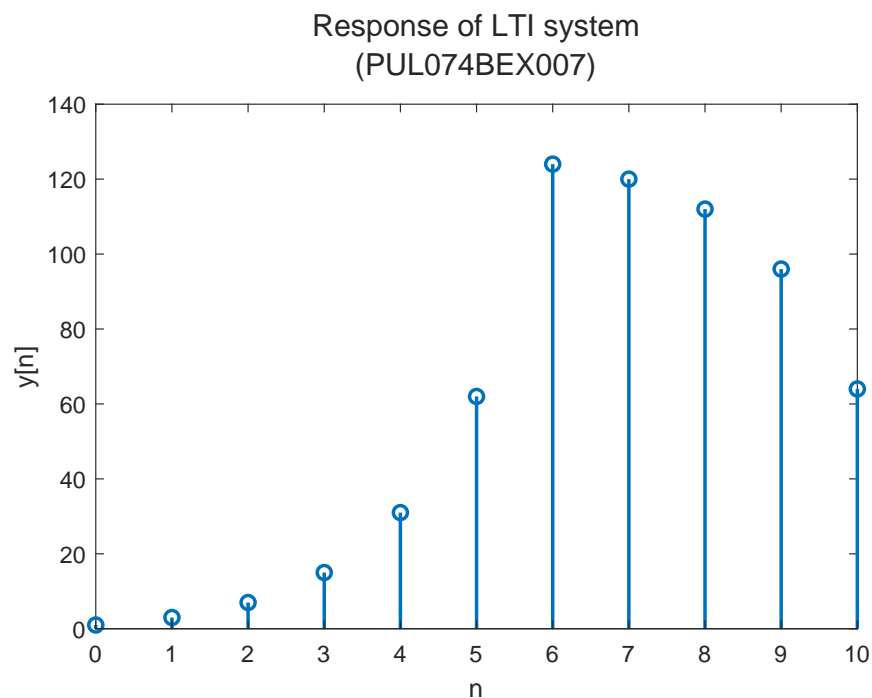


Figure 8: Observation of response of LTI system (Matlab)

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3 n1 = np.arange(start=0, stop=5)
4 x = np.where(np.logical_and(n1 >=
    0, n1 <= 4), 1, 0)
5 n2 = np.arange(start=0, stop=7)
6 h = np.where(np.logical_and(n2 >=
    0, n2 <= 6), 2**n2, 0)
7 n_min = min(np.amin(n1), np.amin(
    n2))
8 n = np.arange(start=n_min, stop=
    n_min+n1.size+n2.size-2+1)
9 y = np.convolve(x, h)

```



```

10 plt.stem(n, y)
11 plt.title('Response of LTI system\
    n(PUL074BEX007)')
12 plt.xlabel('n')
13 plt.ylabel('Value')
14 plt.savefig('../Figures/lab_3_3_py
    .eps', format='eps')
15 plt.show()

```

Listing 12: Python script for finding response of LTI system

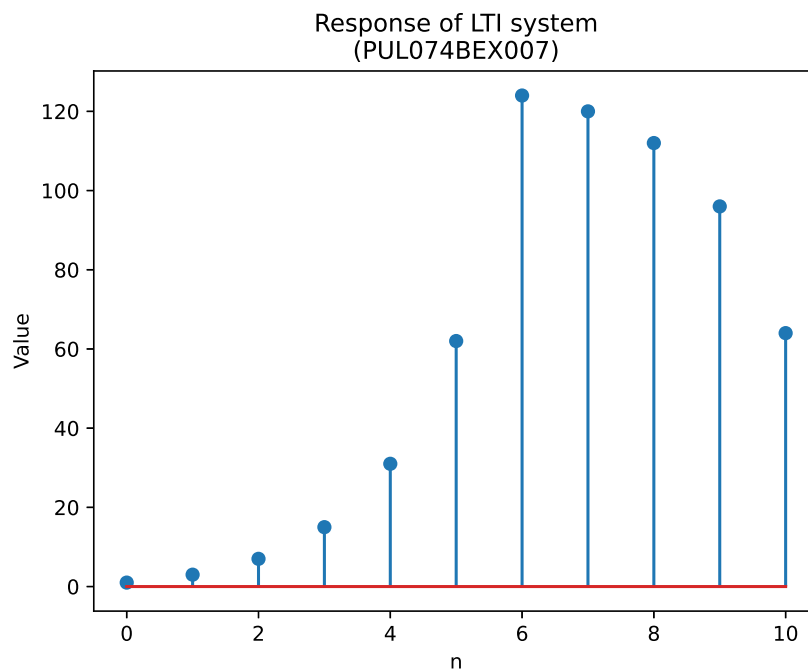


Figure 9: Observation of response of LTI system (Python)

The output response for the given LTI system goes on increasing up to  $n = 6$  and then decays for the rest of the observable time.

#### Problem 4

If the impulse response  $h(t)$  of a LTI system is given by sinc function and input signal  $x(t)$  is a rectangular wave given as,

$$h(t) = \frac{2\tau}{T_p} \text{sinc}\left(\frac{2\tau t}{T_p}\right) \quad \text{and,} \quad x(t) = \begin{cases} 1 & \text{for } 1 \leq t \leq 100 \\ 0 & \text{elsewhere} \end{cases}$$

**Find output of the system for different values of  $\tau$ . Comment on the result.**

```

1 t=linspace(-100,100,1000);
2 x=t>=1;
3 tau=double(input('Enter value of :
    '));
4 Tp=50;
5 h=2*tau/Tp.*sinc(t.*2*tau/Tp);
6 y=conv(x,h,'same');
7 l=tiledlayout(1,1);
8 title(1,['Response of LTI system
    with sinc function as impulse
    response'],sprintf('for %d',tau
    ),'(PUL074BEX007)')}
9 nexttile
10 plot(t,y,'Linewidth', 1.5)
11 xlabel('x')
12 ylabel('y(t)')
13 print('../Figures/lab_3_4_m1','-
    depsc')

```

Listing 13: Matlab script for finding response of LTI system with sinc function as impulse response

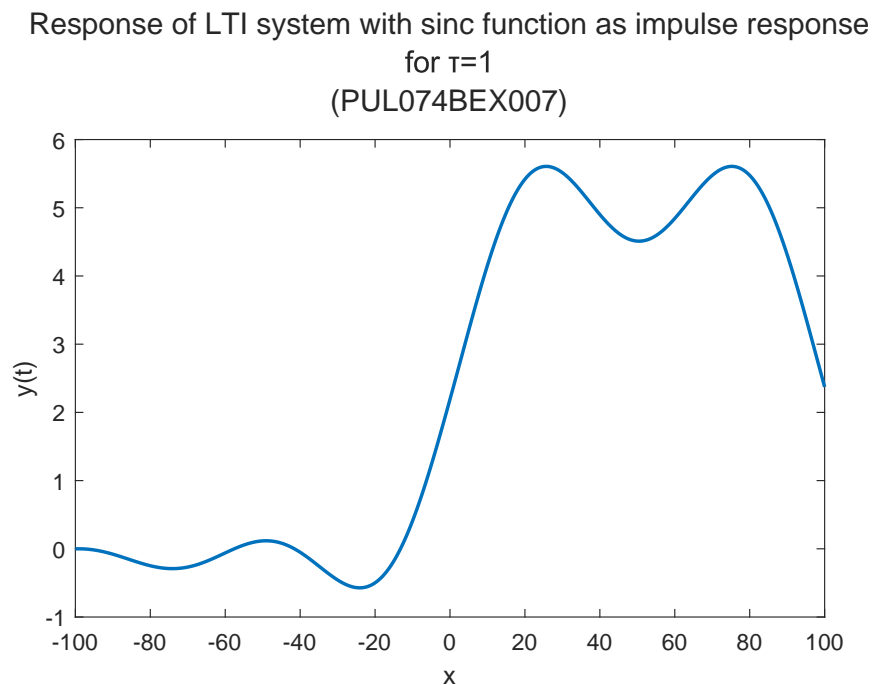


Figure 10: Observation of response of LTI system with sinc function as impulse response (Matlab)

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3 t = np.linspace(-100, 100, 1000)
4 x = np.where(np.logical_and(t>=1,t
    <=100), 1, 0)
5 tau=float(input('Enter value of :
    '))
6 Tp=50
7 h = 2*tau/Tp*np.sinc(t*2*tau/Tp)
8 y = np.convolve(x, h, mode='same')
9 plt.plot(t, y)
10 plt.title('Response of LTI system
    with sinc function as impulse
    response\nwith ={: .2f} \n(
    PUL074BEX007)'.format(tau))
11 plt.xlabel('t')
12 plt.ylabel('y(t)')
13 plt.savefig('../Figures/lab_3_4_py
    .eps', format='eps')
14 plt.show()

```

Listing 14: Python script for finding response of LTI system with sinc function as impulse response

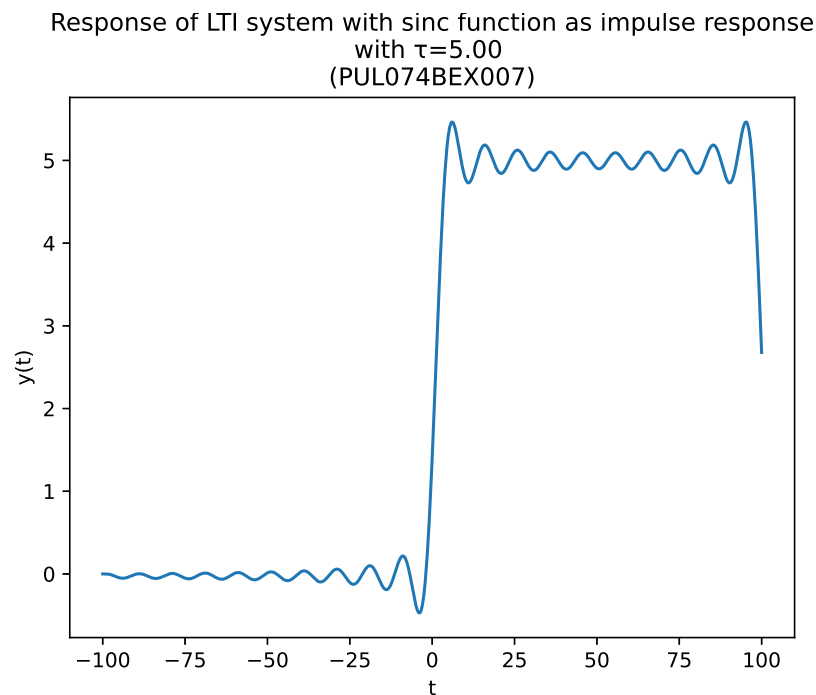


Figure 11: Observation of response of LTI system with sinc function as impulse response (Python)

The output response for the given LTI system where the impulse response is a sinc function

changes with change in the value of  $\tau$ .

## 6 Discussion and Conclusion

In this lab experiment, we performed convolution using different techniques. Initially, with the example problem in the lab sheet, convolution using graphical, tabular and in-built conv function was visualized in Matlab. Then, we performed convolution with the basic formula in both Matlab and Python. Following that, we performed convolution of DT and CT signals with conv function in Matlab and convolve function in Python using NumPy. The output response of an LTI system in DT was also visualized, which was the convolution sum of the input  $x[n]$  and impulse response  $h[n]$ . Finally, output response of an LTI system in CT with the sinc function as the impulse response was visualized.

Hence, the objectives of the lab experiment were fulfilled.