191IT234_Niraj-Nandish

September 15, 2020

1 Lab Week 5

1.0.1 Name: Niraj Nandish

1.0.2 Roll no.: 191IT234

1.0.3 Semester: 3

[1]:

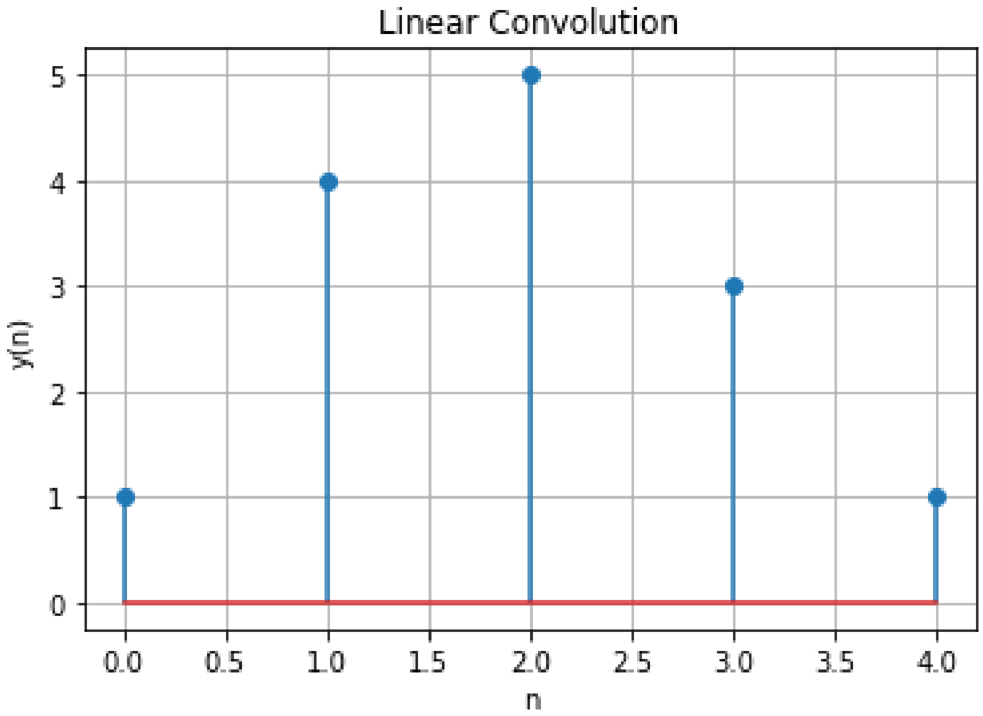
```
# Import all required libraries
import numpy as np                                      # Contains built-in␣
 ↪functions to work on arrays
import matplotlib.pyplot as plt                         # Used to plot graphs
```

1.1 Question 1

Calculate the linear convolution for the following

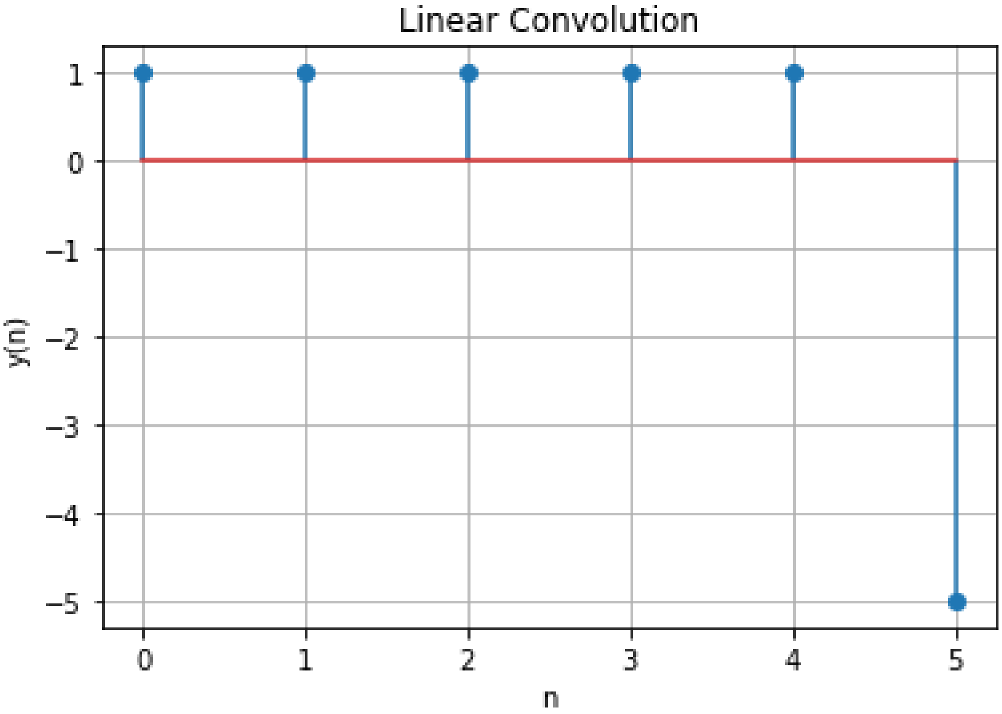
[2]:

```
# a. x(n) = [1 3 2 1],  h(n) = [1 1]

x = [1, 3, 2, 1]
h = [1, 1]
y = np.convolve(x, h)

plt.figure()
plt.stem(y, use_line_collection=True)
plt.grid()
plt.title('Linear Convolution')
plt.xlabel('n')
plt.ylabel('y(n)')
plt.show()
```

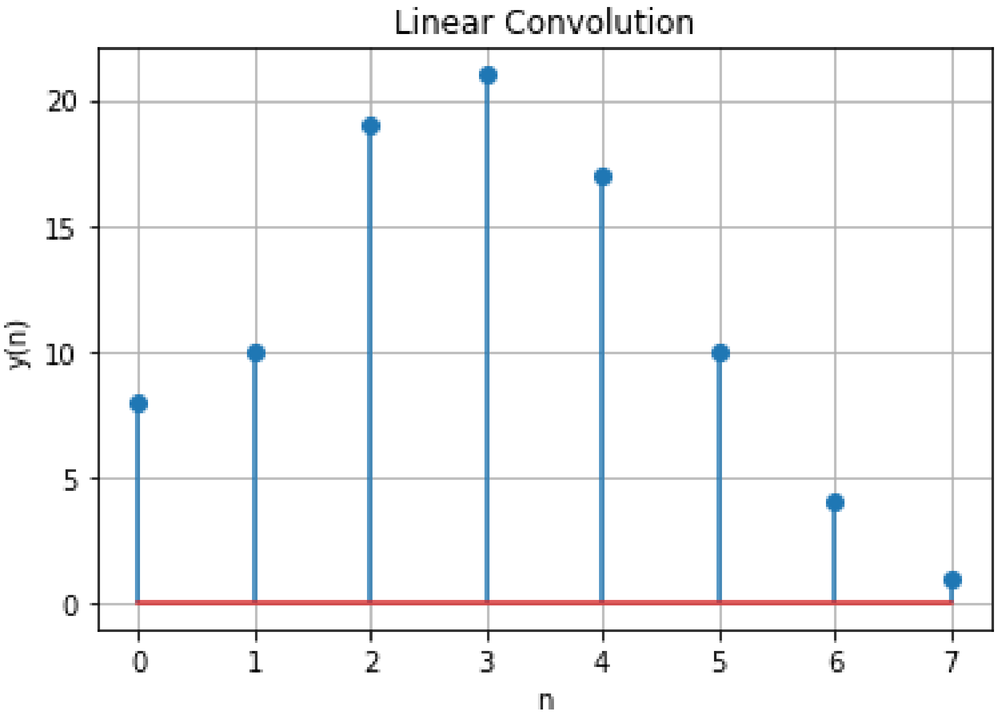


[3]:

```
# b. x(n) = [1 2 3 4 5],  h(n) = [1 -1]

x = [1, 2, 3, 4, 5]
h = [1, -1]
y = np.convolve(x, h)

plt.figure()
plt.stem(y, use_line_collection=True)
plt.grid()
plt.title('Linear Convolution')
plt.xlabel('n')
plt.ylabel('y(n)')
plt.show()
```

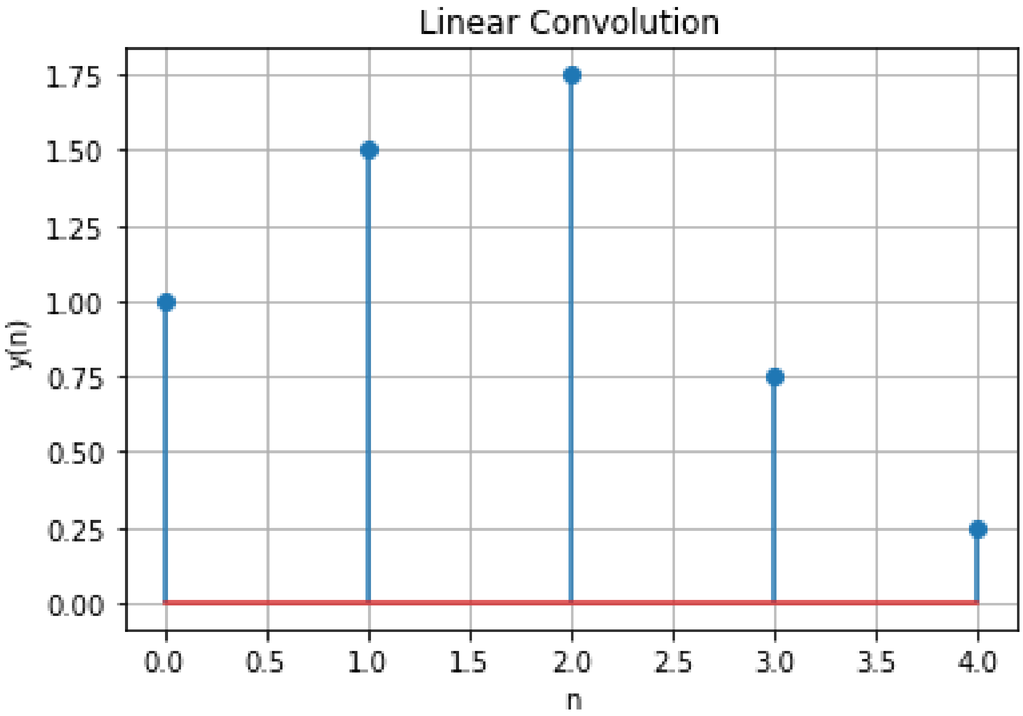


```
[4]: # c. x(n) = [2 1 3 2 1],  h(n) = [4 3 2 1]

     x = [2, 1, 3, 2, 1]
     h = [4, 3, 2, 1]
     y = np.convolve(x, h)

     plt.figure()
     plt.stem(y, use_line_collection=True)
     plt.grid()
     plt.title('Linear Convolution')
     plt.xlabel('n')
     plt.ylabel('y(n)')
     plt.show()
```



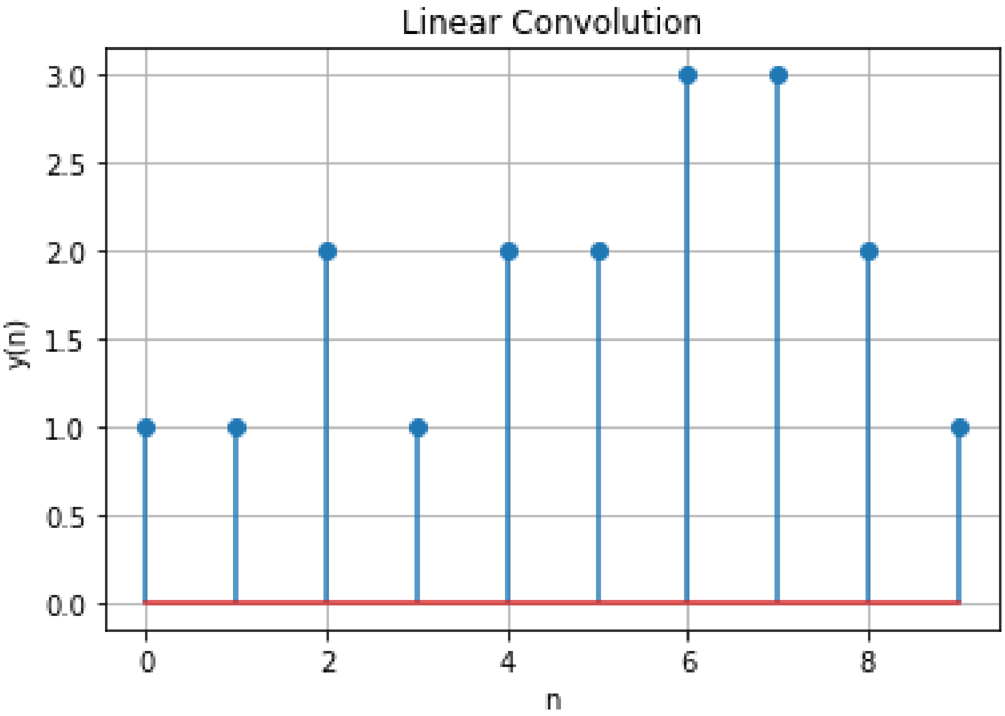
[5]:

```
# d. x(n) = [1 1 1],  h(n) = [1 0.5 0.25]

x = [1, 1, 1]
h = [1, 0.5, 0.25]
y = np.convolve(x, h)

plt.figure()
plt.stem(y, use_line_collection=True)
plt.grid()
plt.title('Linear Convolution')
plt.xlabel('n')
plt.ylabel('y(n)')
plt.show()
```



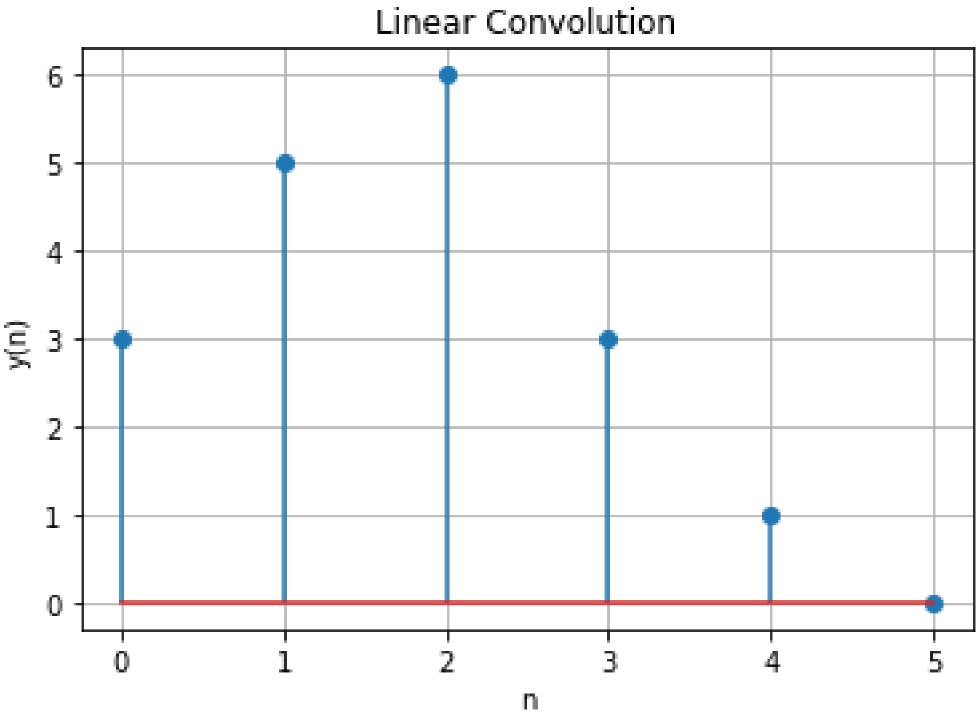
[6]:

```
# e. x(n) = [1 0 1 0 1 1 1 1],  h(n) = [1 1 1]

x = [1, 0, 1, 0, 1, 1, 1, 1]
h = [1, 1, 1]
y = np.convolve(x, h)

plt.figure()
plt.stem(y, use_line_collection=True)
plt.grid()
plt.title('Linear Convolution')
plt.xlabel('n')
plt.ylabel('y(n)')
plt.show()
```



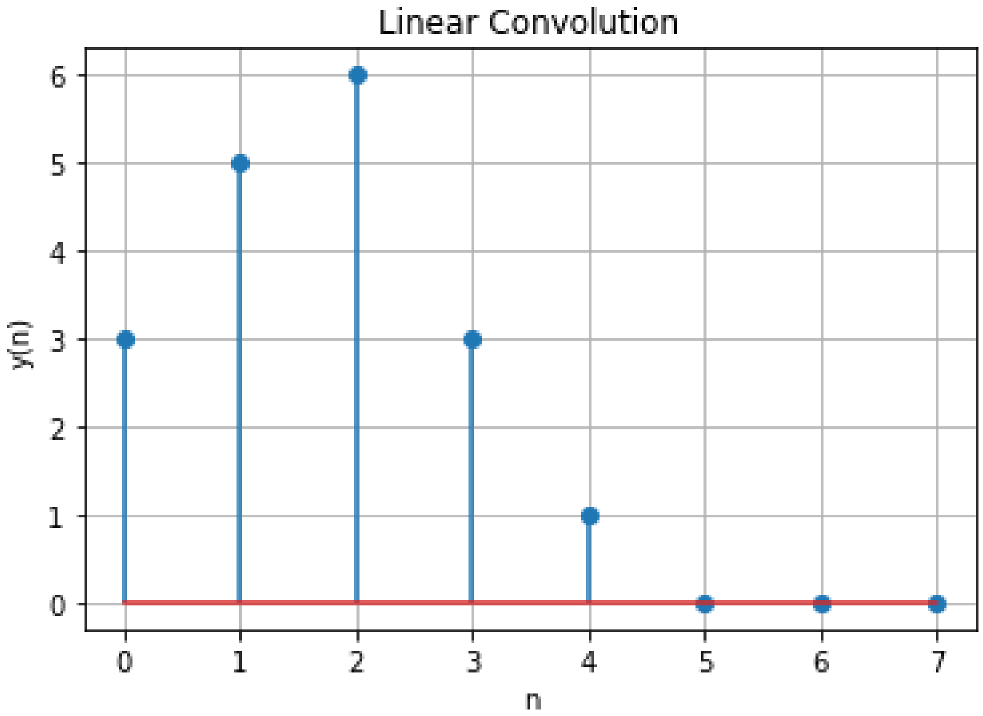
[7]:

```
# f. x(n) = [3 2 1 0],  h(n) = [1 1 1]

x = [3, 2, 1, 0]
h = [1, 1, 1]
y = np.convolve(x, h)

plt.figure()
plt.stem(y, use_line_collection=True)
plt.grid()
plt.title('Linear Convolution')
plt.xlabel('n')
plt.ylabel('y(n)')
plt.show()
```



[8]:

```
# f. x(n) = [3 2 1 0 0 0],  h(n) = [1 1 1]

x = [3, 2, 1, 0, 0, 0]
h = [1, 1, 1]
y = np.convolve(x, h)

plt.figure()
plt.stem(y, use_line_collection=True)
plt.grid()
plt.title('Linear Convolution')
plt.xlabel('n')
plt.ylabel('y(n)')
plt.show()
```



1.2 Question 2

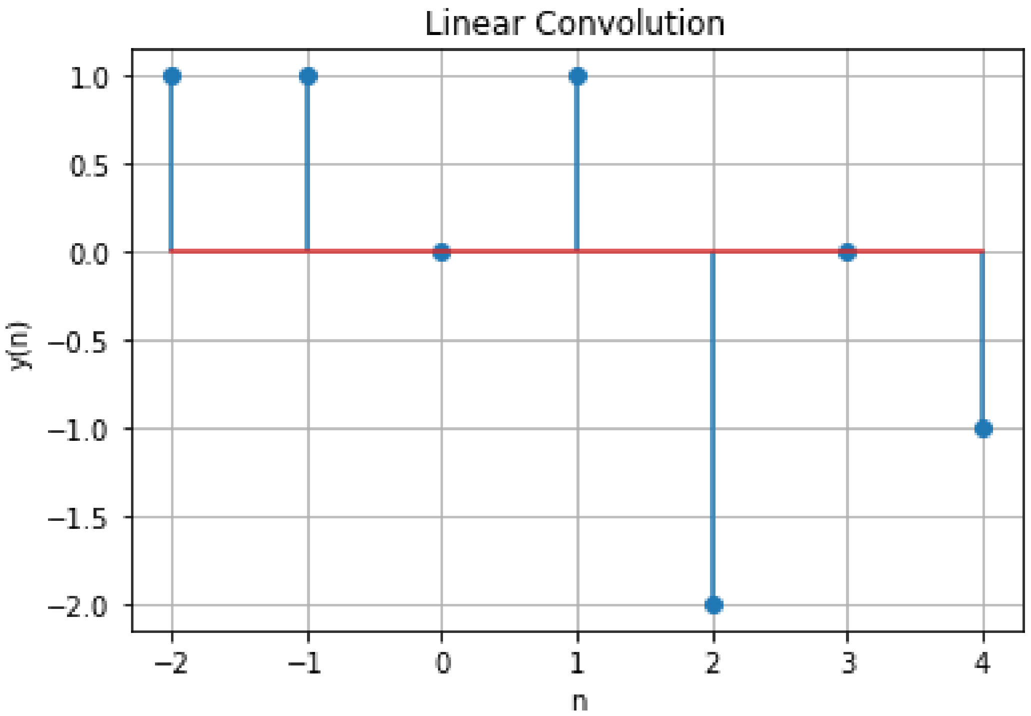
Find the output of the system if input x(n) and impulse reaction h(n) are given by:

x(n) = 1 n = -2, 0, 1 = 2 n = -1 = 0 otherwise

$h(n) = \delta(n) - \delta(n - 1) + \delta(n - 2) - \delta(n - 3)$

[9]:

```
n = np.arange(-2, 5)

x = []
for i in n:
    if i in [-2, 1, 0]:
        x.append(1)
    elif i == -1:
        x.append(2)
    else:
        x.append(0)

def impulse(n):
    h = []
    for i in n:
        if i == 0:
            h.append(1)
        else:
            h.append(0)
    return np.array(h)

h = impulse(n) - impulse(n-1) + impulse(n-2) - impulse(n-3)

y = np.trim_zeros(np.convolve(x, h))

plt.figure()
plt.stem(n, y, use_line_collection=True)
plt.grid()
plt.title('Linear Convolution')
plt.xlabel('n')
plt.ylabel('y(n)')
plt.show()
```


Linear Convolution
y(n)
n