IT204 Assignment 1

Nithin S

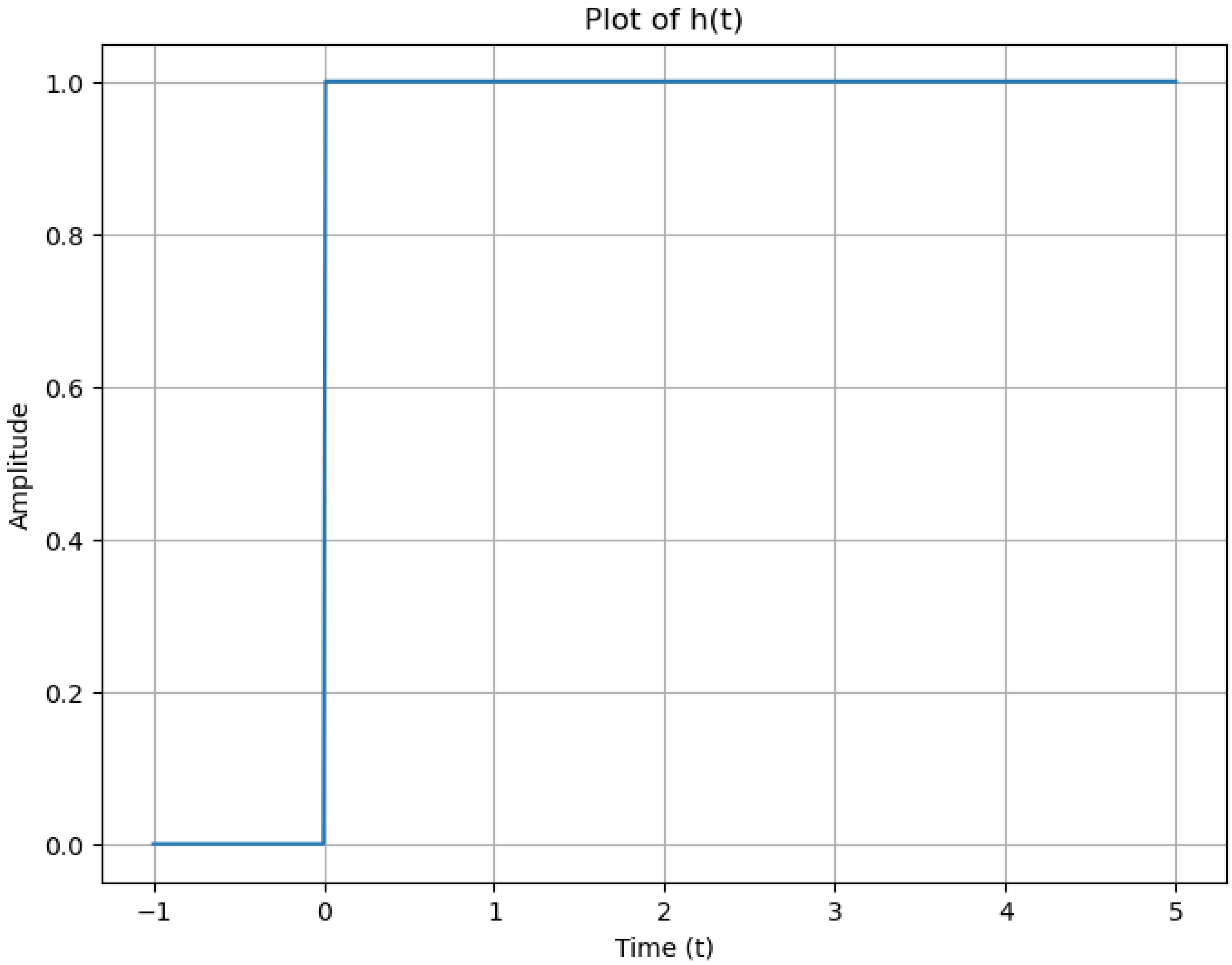
221IT085

Experiment 2: Impulse response of a system

Continous Signal Impulse Response

In [25]:

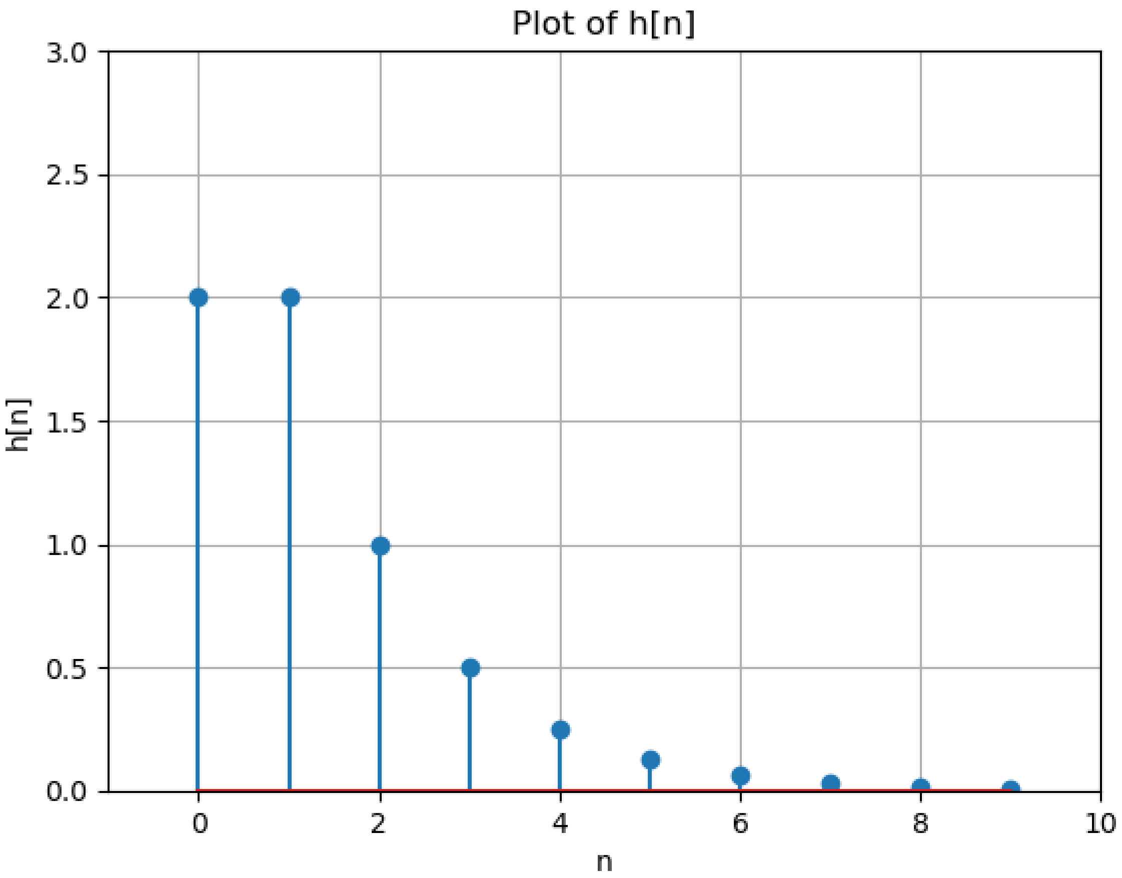
```
import numpy as np
import matplotlib.pyplot as plt


t = np.linspace(-1, 5, 500)

# Define the signal h(t)
h = 3 * np.where(t == 0, 1, 0) + np.heaviside(t, 1)

plt.figure(figsize=(8, 6))
plt.plot(t, h)
plt.xlabel('Time (t)')
plt.ylabel('Amplitude')
plt.title('Plot of h(t)')
plt.grid(True)
plt.show()
```



Discrete Signal Time Impulse

In [22]:

```
import numpy as np
import matplotlib.pyplot as plt


n = np.arange(0, 10)

# Define the sequence h[n]
h = 2*(0.5)**n * (n >= 0) + (0.5)**(n - 1) * (n >= 1)


plt.stem(n, h)

# Label the axes
plt.xlabel('n')
plt.ylabel('h[n]')


plt.xlim(-1, 10)
plt.ylim(0, 3)


plt.title('Plot of h[n]')


plt.grid(True)
plt.show()
```

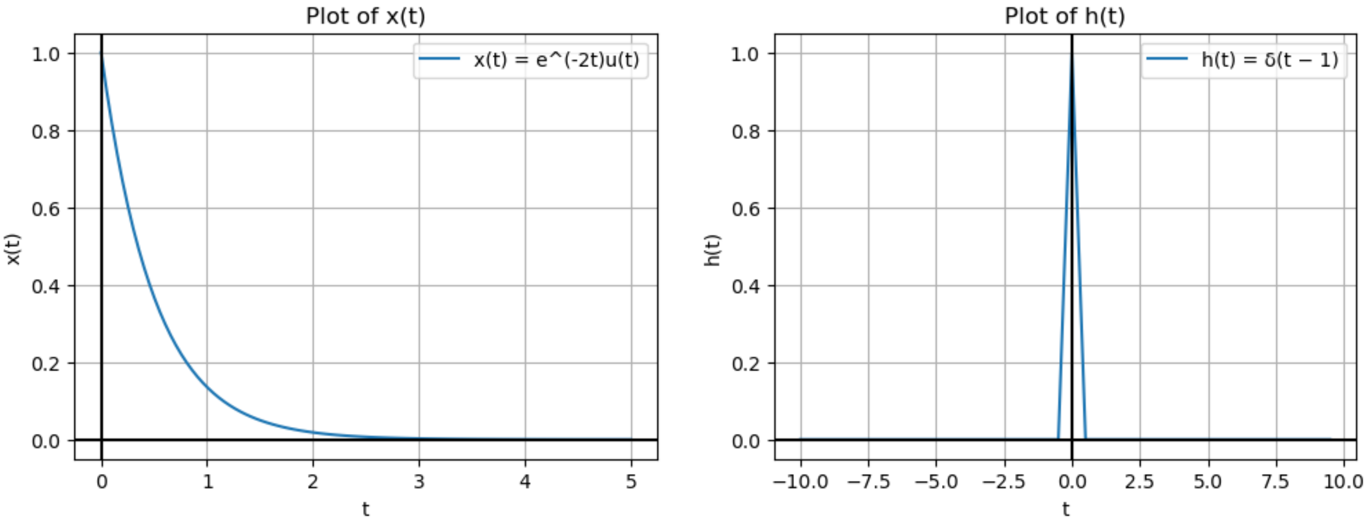


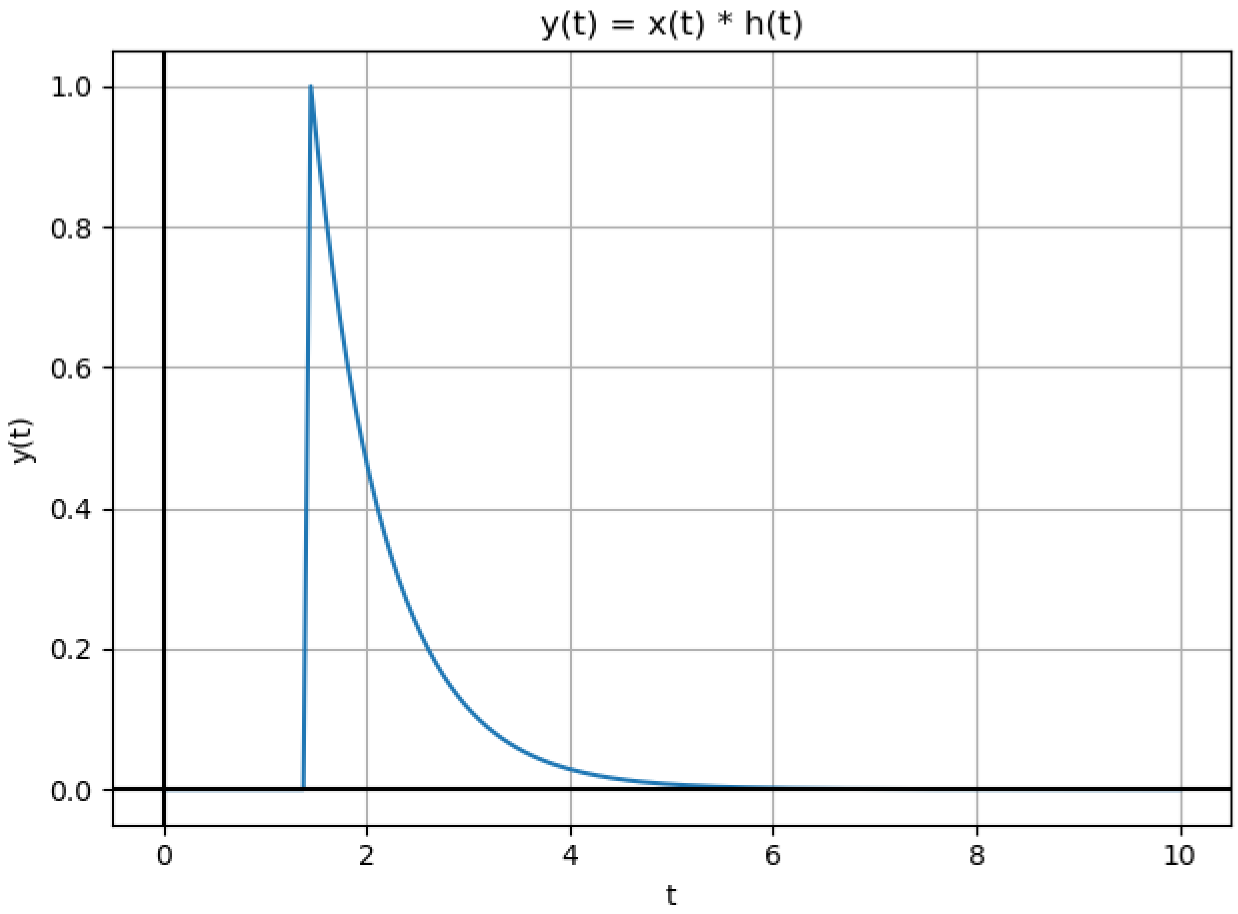
Experiment 3: Convolution

Continuous Convolution Problem:

In [34]:

```
import numpy as np
import matplotlib.pyplot as plt

# Define the first signal x(t)
t = np.linspace(0, 5, 100)
x = np.exp(-2 * t) * (t >= 0)

# Define the second signal, which is the impulse signal
n = np.arange(-10, 10, 0.5)
delta = np.array([1 if x == 0 else 0 for x in n])

# Convolve the two signals
convolution_result = np.convolve(x, delta, 'full')
t_conv = np.linspace(0, 10, len(convolution_result))

# Plot the original signals and the convolution result
plt.figure(figsize=(12, 4))
plt.subplot(121)
plt.plot(t, x, label='x(t) = e^(-2t)u(t)')
plt.xlabel('t')
plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')
plt.ylabel('x(t)')
plt.title('Plot of x(t)')
plt.grid(True)
plt.legend()

plt.subplot(122)
plt.plot(n, delta,label='h(t) = δ(t − 1)')
plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')
plt.grid()
plt.title("Plot of h(t)")
plt.xlabel("t")
plt.ylabel("h(t)")
plt.legend()
plt.show()


plt.plot(t_conv, convolution_result)
plt.title("y(t) = x(t) * h(t)")
plt.xlabel("t")
plt.ylabel("y(t)")
plt.axhline(y=0, color='k')
plt.axvline(x=0, color='k')
plt.grid(True)
plt.tight_layout()
plt.show()
```

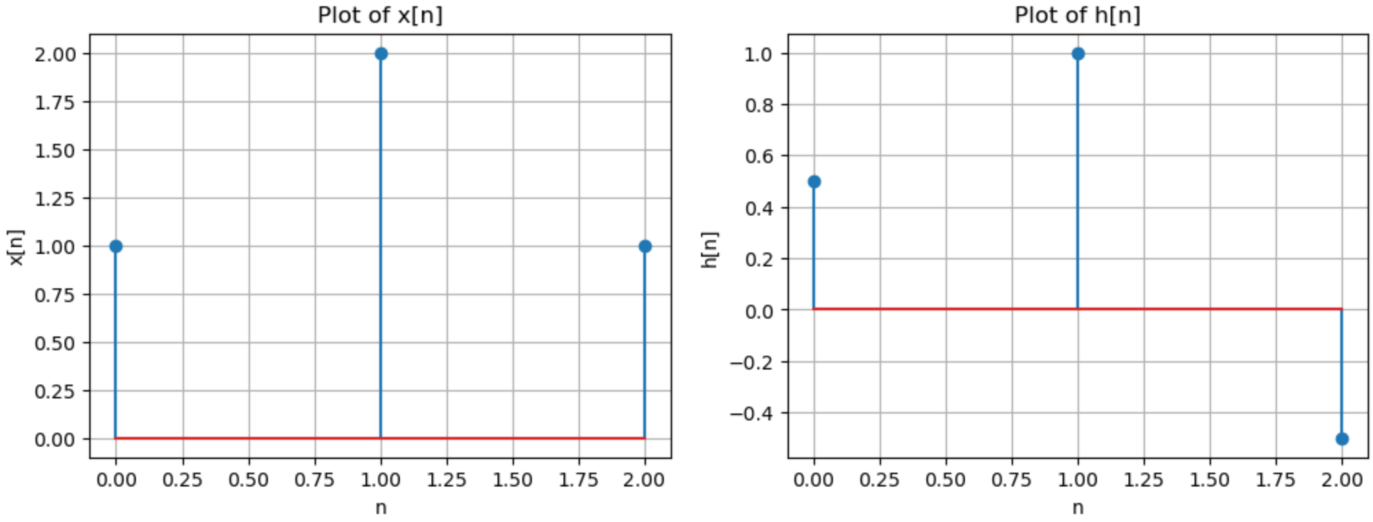


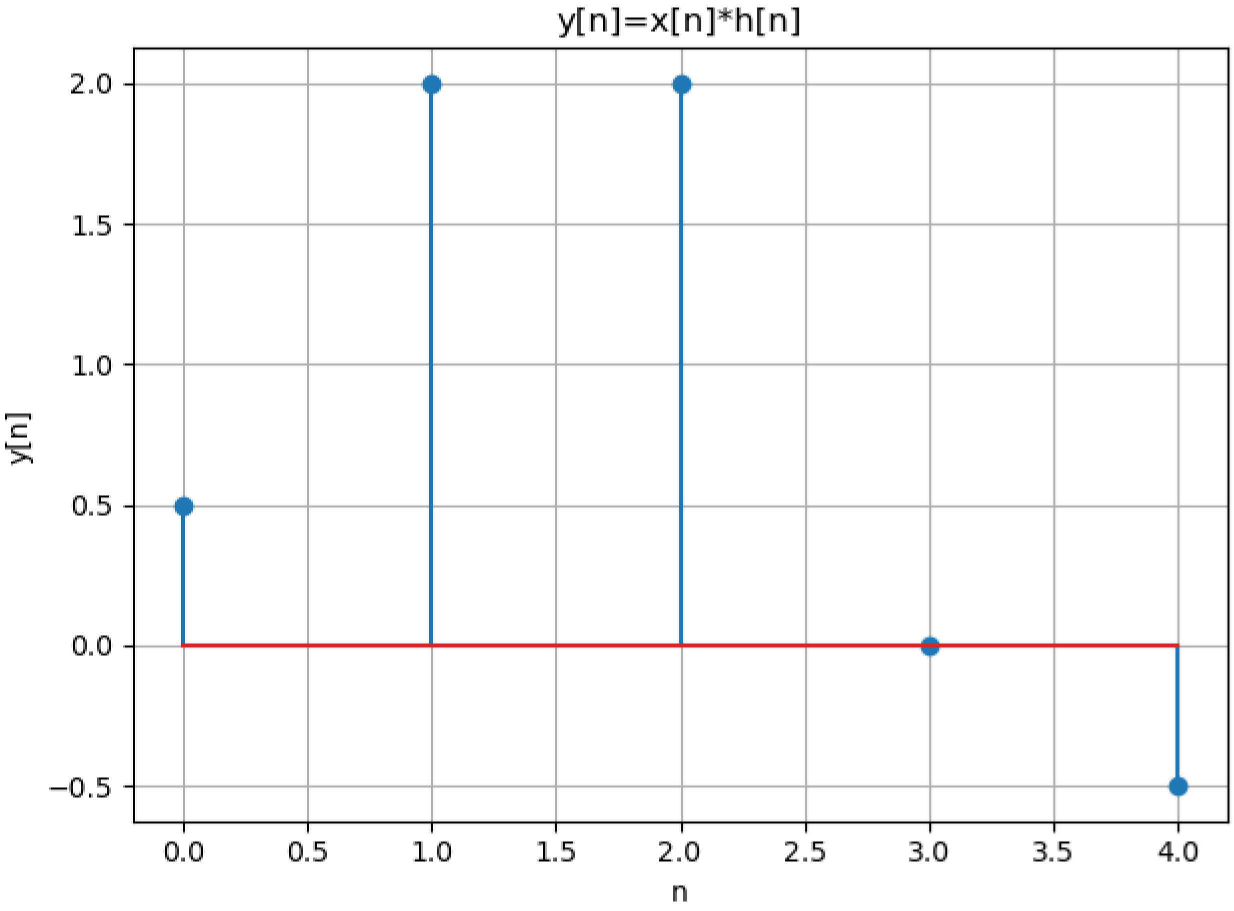


Discrete Convolution Problem:

In [63]:

```
import numpy as np
import matplotlib.pyplot as plt

# Define the values of x[n] and h[n]
n = np.arange(0, 3)
x = np.array([1, 2, 1])
h = np.array([0.5, 1, -0.5])

# Compute the convolution of x[n] and h[n]
convolution_result = np.convolve(x, h, 'full')
n_conv = np.arange(0, len(convolution_result))
plt.figure(figsize=(12, 4))
# Subplot 1: Plot of x[n]
plt.subplot(121)
plt.stem(n, x)
plt.xlabel('n')
plt.ylabel('x[n]')
plt.title('Plot of x[n]')
plt.grid(True)

# Subplot 2: Plot of h[n]
plt.subplot(122)
plt.stem(n, h)
plt.xlabel('n')
plt.ylabel('h[n]')
plt.title(('Plot of h[n]'))
plt.grid(True)
plt.show()

# Subplot 3: Plot of the convolution result
plt.stem(n_conv, convolution_result)
plt.xlabel('n')
plt.ylabel('y[n]')
plt.title('y[n]=x[n]*h[n]')
plt.grid(True)

plt.tight_layout()
plt.show()
```


Plot of x[n]
x[n]
n
Plot of h[n]
h[n]
n


y[n]=x[n]*h[n]
y[n]
2.0
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
n