

# 1 Introduction

The goal of this lab is to become familiar with using convolution to compute a system's step response.

## 2 Equations

For this lab, we create the following signals as user-defined functions and also we had convolve this function manually:

$$h1(t) = e^{2t}u(1-t) \quad (1)$$

$$h2(t) = u(t-2) - u(t-6) \quad (2)$$

$$h3(t) = \cos(w_0 t)u(t) \quad (3)$$

The step responses, respectively:

$$f1(t) = \frac{1}{2}(e^{2t} - e^2)u(1-t) \quad (4)$$

$$f2(t) = (t-3)u(t-2) - (t-6)u(t-6) \quad (5)$$

$$f3(t) = \frac{1}{w_0} \sin(w_0 t)u(t) \quad (6)$$

## 3 Methodology and Results

Lab 4 has 2 parts :

- Part 1:

In this part, we plot the three functions in a single figure (separate subplots) from -10 to 10 with time steps small enough to achieve appropriate resolution:

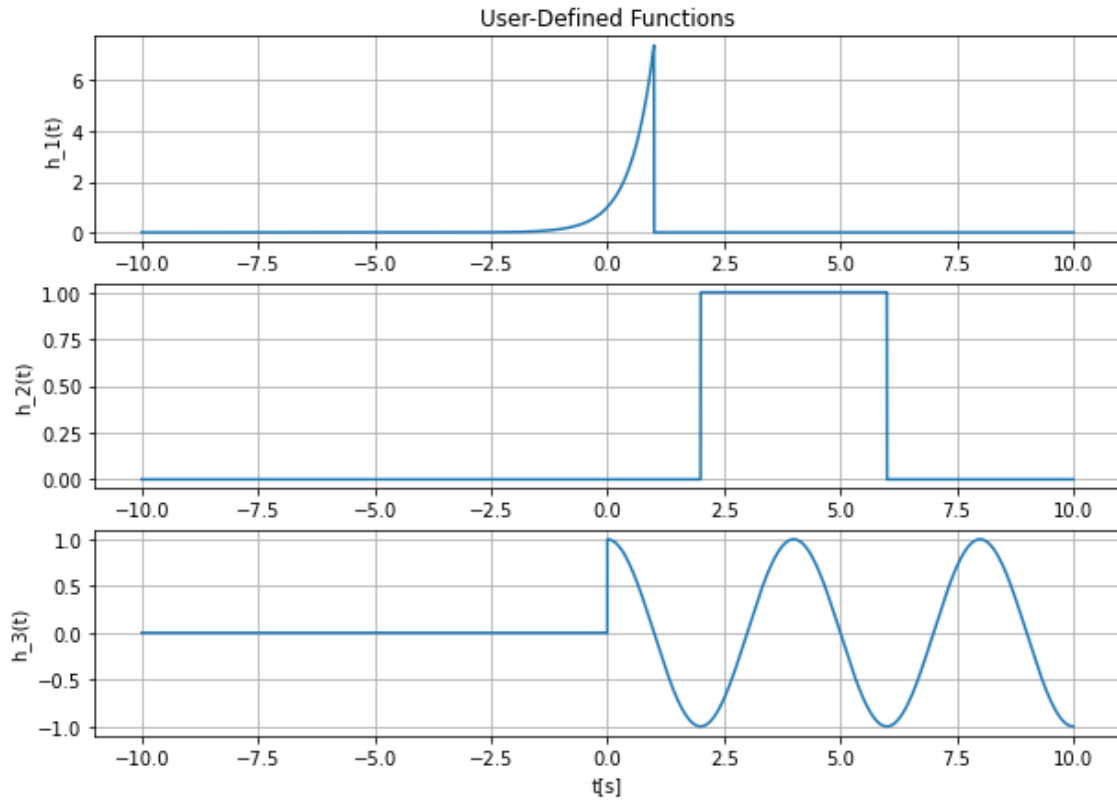
```
1      %% part 1 task 1 & 2
2
3
4      def u(t):
5          y = np.zeros(t.shape)
6          for i in range(len(t)):
7              if t[i] >= 0:
8                  y[i] = 1
9              else:
10                 y[i] = 0
11         return y
12
13     def h_1(t):
14         h1 = u(1-t)*np.exp(2*t)
15         return h1
16
```

```

17
18     def h_2(t):
19         h2 = u(t-2)-u(t-6)
20         return h2
21
22     def h_3(t):
23         h3 = np.cos((2*np.pi*0.25)*t)*u(t)
24         return h3
25
26     steps = 1e-3
27     t = np.arange(-10, 10 + steps, steps)
28
29     plt.figure(figsize=(10,7))
30     plt.subplot(3, 1, 1)
31     plt.plot(t,h_1(t))
32     plt.ylabel ('h_1(t)')
33     plt.title ('User-Defined Functions')
34     plt.grid()
35     plt.subplot(3, 1, 2)
36     plt.plot(t,h_2(t))
37     plt.ylabel ('h_2(t)')
38     plt.grid()
39     plt.subplot(3, 1, 3)
40     plt.plot(t,h_3(t))
41     plt.ylabel ('h_3(t)')
42     plt.grid()
43     plt.xlabel('t[s]')
44     plt.show()
45
46
47
48

```

The output of this part is:



- Part 2:

Task 1: We convolve the three transfer functions with step function to find step response using the convolution created in lab 3 and plot the response:

```

1      %% part 2 task 1
2
3
4
5      def conv(h1, h2):
6          Nh1 = len(h1)
7          Nh2 = len(h2)
8          h1Extended = np.append(h1, np.zeros((1, Nh2 - 1)))
9          h2Extended = np.append(h2, np.zeros((1, Nh1 - 1)))
10         result = np.zeros(h1Extended.shape)
11
12         for i in range(Nh2 + Nh1 - 2):
13             result[i] = 0
14             for j in range(Nh1):
15                 if(i - j + 1 > 0):
16                     try:
17                         result[i] = result[i] + h1Extended[j]*h2Extended[i - j + 1]
18                     except:
19                         print(i, j)
20             return result
21

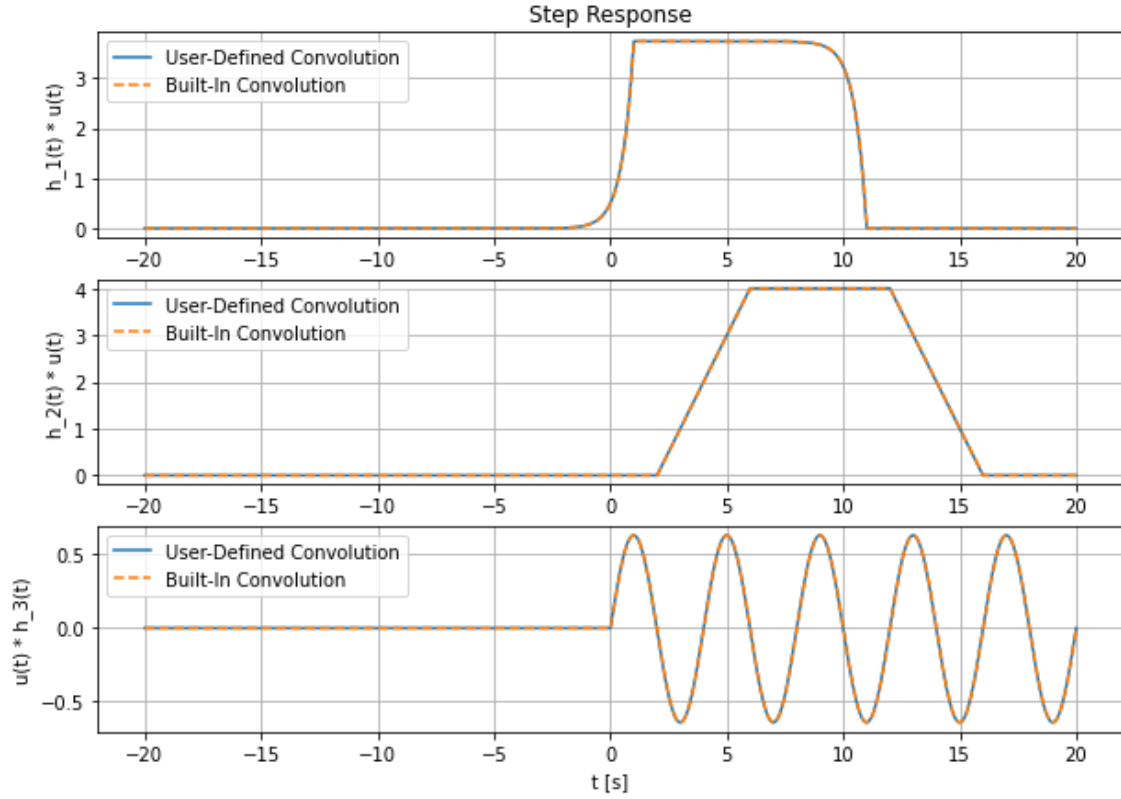
```

```

22     steps = 1e-2
23     t = np.arange(-10, 10 + steps, steps)
24     NN = len(t)
25     tExtended = np.arange(2*t[0], 2*t[NN - 1] + steps, steps)
26
27     h1 = h_1(t)
28     h2 = h_2(t)
29     h3 = h_3(t)
30
31
32     conv12 = conv(h1, u(t))*steps
33     conv12Check = sig.convolve(h1, u(t))*steps
34
35     plt.figure(figsize = (10, 7))
36     plt.subplot(3, 1, 1)
37     plt.plot(tExtended, conv12, label = 'User-Defined Convolution')
38     plt.plot(tExtended, conv12Check, '--', label = 'Built-In Convolution')
39     plt.grid()
40     plt.legend()
41     plt.xlabel('t [s]')
42     plt.ylabel('h_1(t) * u(t)')
43     plt.title('Step Response')
44
45
46     conv23 = conv(h2, u(t))*steps
47     conv23Check = sig.convolve(h2, u(t))*steps
48     plt.subplot(3, 1, 2)
49     plt.plot(tExtended, conv23, label = 'User-Defined Convolution')
50     plt.plot(tExtended, conv23Check, '--', label = 'Built-In Convolution')
51     plt.grid()
52     plt.legend()
53     plt.xlabel('t [s]')
54     plt.ylabel('h_2(t) * u(t)')
55
56
57     conv13 = conv(u(t), h3)*steps
58     conv13Check = sig.convolve(u(t), h3)*steps
59     plt.subplot(3, 1, 3)
60     plt.plot(tExtended, conv13, label = 'User-Defined Convolution')
61     plt.plot(tExtended, conv13Check, '--', label = 'Built-In Convolution')
62     plt.grid()
63     plt.legend()
64     plt.xlabel('t [s]')
65     plt.ylabel('u(t) * h_3(t)')
66
67

```

The output of task 1 is:



Task 3: We convolved the transfer functions with step function manually and found the step response equations as follows:

$$f1(t) = \frac{1}{2}(e^{2t} - e^2)u(t - 1) \quad (7)$$

$$f2(t) = (t - 3)u(t - 2) - (t - 6)u(t - 6) \quad (8)$$

$$f3(t) = \frac{1}{w_0} \sin(w_0 t)u(t) \quad (9)$$

Then, we used these equations in Python to get their plots:

```

1  def f_1(t):
2      f1 = (1/2)*u(1-t)*(np.exp(2*t)-np.exp(2))
3      return f1
4      plt.figure(figsize=(10,7))
5      plt.subplot(3, 1, 1)
6      plt.plot(tExtended,f_1(tExtended))
7      plt.grid()
8      plt.ylabel ('f_1(t)')
9      plt.xlabel('t [s]')
10     plt.title ('Step Response Manually')

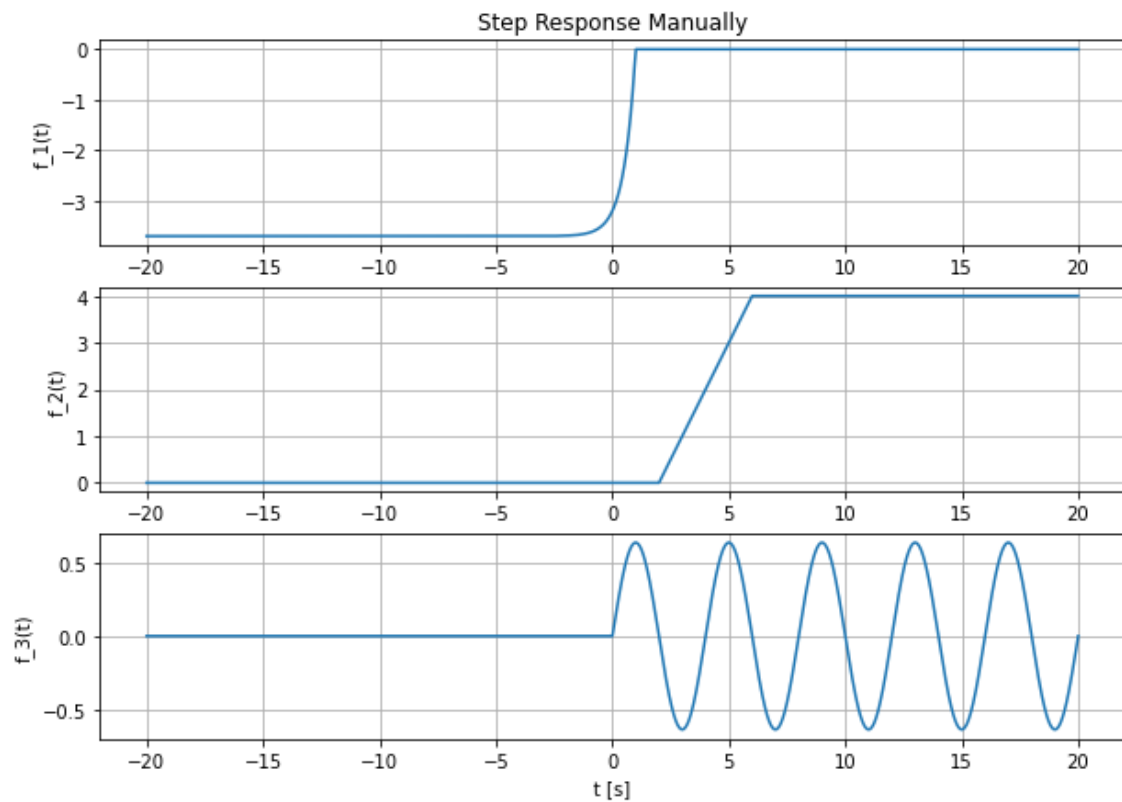
```

```

11
12
13
14     def f_2(t):
15         f2 = (t-2)*u(t-2)-(t-6)*u(t-6)
16         return f2
17     plt.subplot(3, 1, 2)
18     plt.plot(tExtended,f_2(tExtended))
19     plt.grid()
20     plt.ylabel ('f_2(t)')
21     plt.xlabel('t [s]')
22
23
24
25     def f_3(t):
26         f3 = (1/(2*np.pi*0.25))*np.sin((2*np.pi*0.25)*t)*u(t)
27         return f3
28     plt.subplot(3, 1, 3)
29     plt.plot(tExtended,f_3(tExtended))
30     plt.grid()
31     plt.ylabel ('f_3(t)')
32     plt.xlabel('t [s]')
33
34

```

The output of this task is:



## 4 Questions

1. Leave any feedback on the clarity of lab tasks, expectations, and deliverables.

The Lab is very clear.

## 5 Conclusion

At the end of this lab, we became more familiar with using convolution and learned how to compute a system's step response by Python.