ECE 351 - Section 52 #

Lab 2 Report

Lab2

Submitted By :
Shujaea Aldousari

Contents

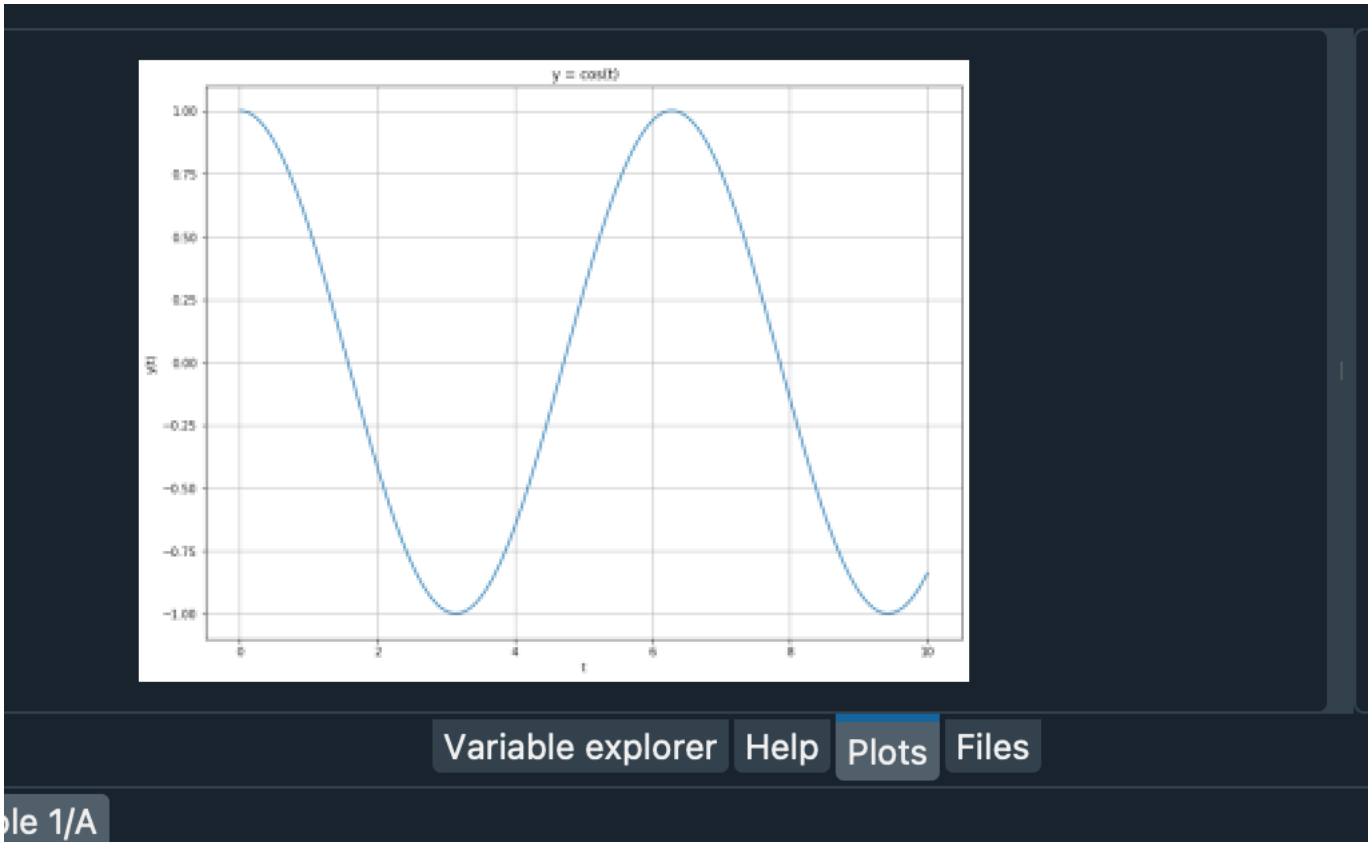
1 Introduction

In lab 2 we were introduced with new way of plotting signals which was if/else statements as well as for loops Also, we knew how to define a ramp function as well as a step function as well as adding and finding the derivative of the functions by using Python. At the very beginning of the lab, we were asked to follow the example code in order to work on the first task.

2 Part 1

In part one we were asked to defned a function named by used defined function with a good resolution and valued of y = cos(t). The difference between a good resolution and poor resolution is of its step size range. The function we were asked to find is between a range of 0 to 10. I had to create the code of the function to get the figure, which is in the next figure.

```
1
2
3 import numpy as np
4 import matplotlib . pyplot as plt
5
6
7 steps = 1e-2
8 t = np. arange (0, 10 + steps , steps )
9
10
11 def func1 (t):
12     y = np. zeros (t. shape )
13     for i in range ( len (t)):
14         y[i] = np.cos (t[i])
15     return y
16
17 y = func1 (t)
18 plt. figure ( figsize = (12 , 9))
19 plt. plot (t, y)
20 plt. grid ()
21 plt. ylabel ('y(t)')
22 plt. xlabel ('t')
23 plt. title (' y = cos(t)')
```

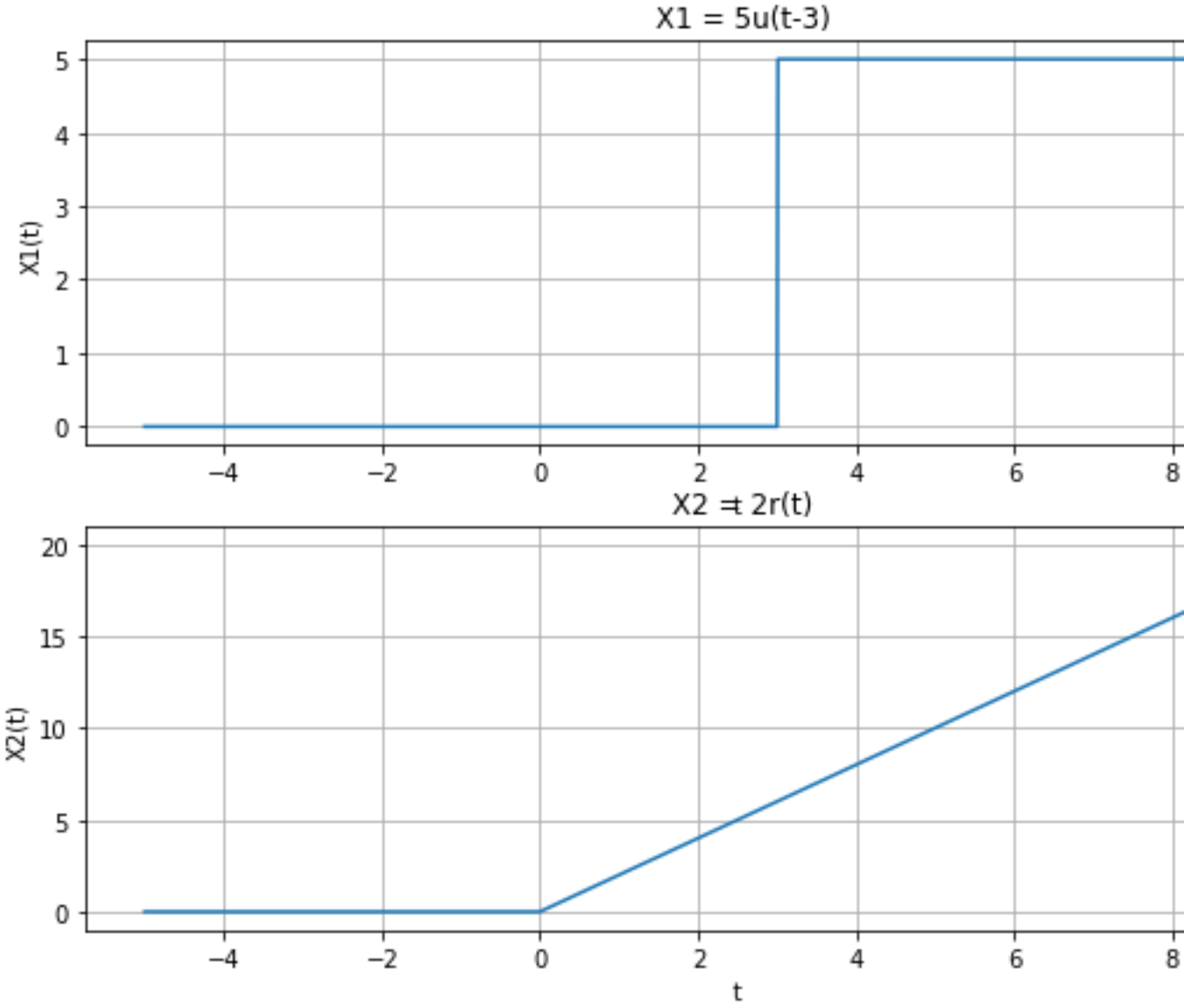


3 Part 2 Task 2

In the second part, we were asked to create a utilize a user-defined function to model a simple plot. We were asked to plot two types of function (ramp and step).

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


steps = 1e-2 # Define step size
t = np. arange (-5, 10 + steps , steps )

# user defined function
def step (t,tn):
    y = np. zeros (t. shape )
    for i in range ( len (t)):
        if t[i] >= -tn:
            y[i] = 1
    return y

```

```
16 def ramp (t, tn, m):
17     y = np. zeros (t. shape )
18     for i in range ( len (t)):
19         if t[i] >= -tn:
20             y[i] = m * (t[i] + tn)
21     return y
22
23 X1 = 5 * step (t,-3)
24 X2 = ramp(t, 0, 2)
25 plt. figure ( figsize = (10 , 7))
26 plt. subplot (2, 1, 1)
27 plt. plot (t, X1)
28 plt. grid ()
29 plt. ylabel ('X1(t)')
30 plt. xlabel ('t')
31 plt. title (' X1 = 5u(t-3)')
32
33 plt. subplot (2, 1, 2)
34 plt. plot (t, X2)
35 plt. grid ()
36 plt. ylabel ('X2(t)')
37 plt. xlabel ('t')
38 plt. title (' X2 = 2r(t)')
39 plt. show ()
```

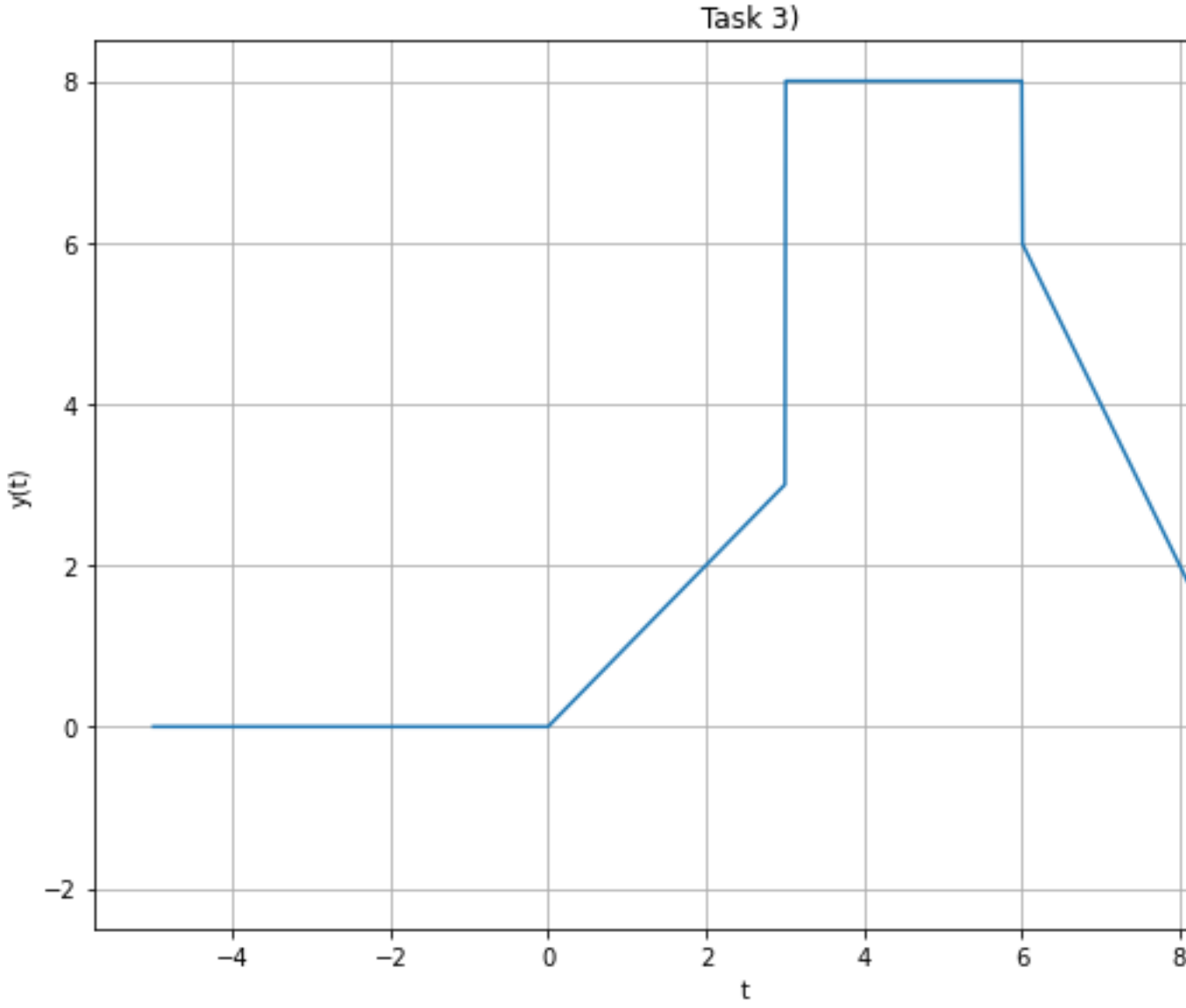


4 Part 2 Task 3

In task 3, we were asked to get/include the functions and plot them together as single graph which was the work from previous parts as well as some other new variables.

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

```

```
steps = 1e-2
t = np. arange (-5, 10 + steps , steps )
N=len (t)

def step (t,tn):
    y = np. zeros (t. shape )
    for i in range ( len (t)):
        if t[i] >= -tn:
            y[i] = 1
    return y

def ramp (t, tn, m):
    y = np. zeros (t. shape )
    for i in range ( len (t)):
        if t[i] >= -tn:
            y[i] = m * (t[i] + tn)
    return y #
ym= np. zeros ([5, N] )
ym[0,:]= ramp(t,0,1)
ym[1,:]= ramp(t,-3,-1)
ym[2,:]= 5 * step (t,-3)
ym[3,:]= -2 * step (t,-6)
ym[4,:]= ramp(t,-6,-2)

y= [sum(x) for x in zip(*ym)]


plt. figure ( figsize = (10 , 7))
plt. plot (t, y)
plt. grid ()
plt. ylabel ('y(t)')
plt. xlabel ('t')
plt. title (' Task 3)')

plt. show ()
```



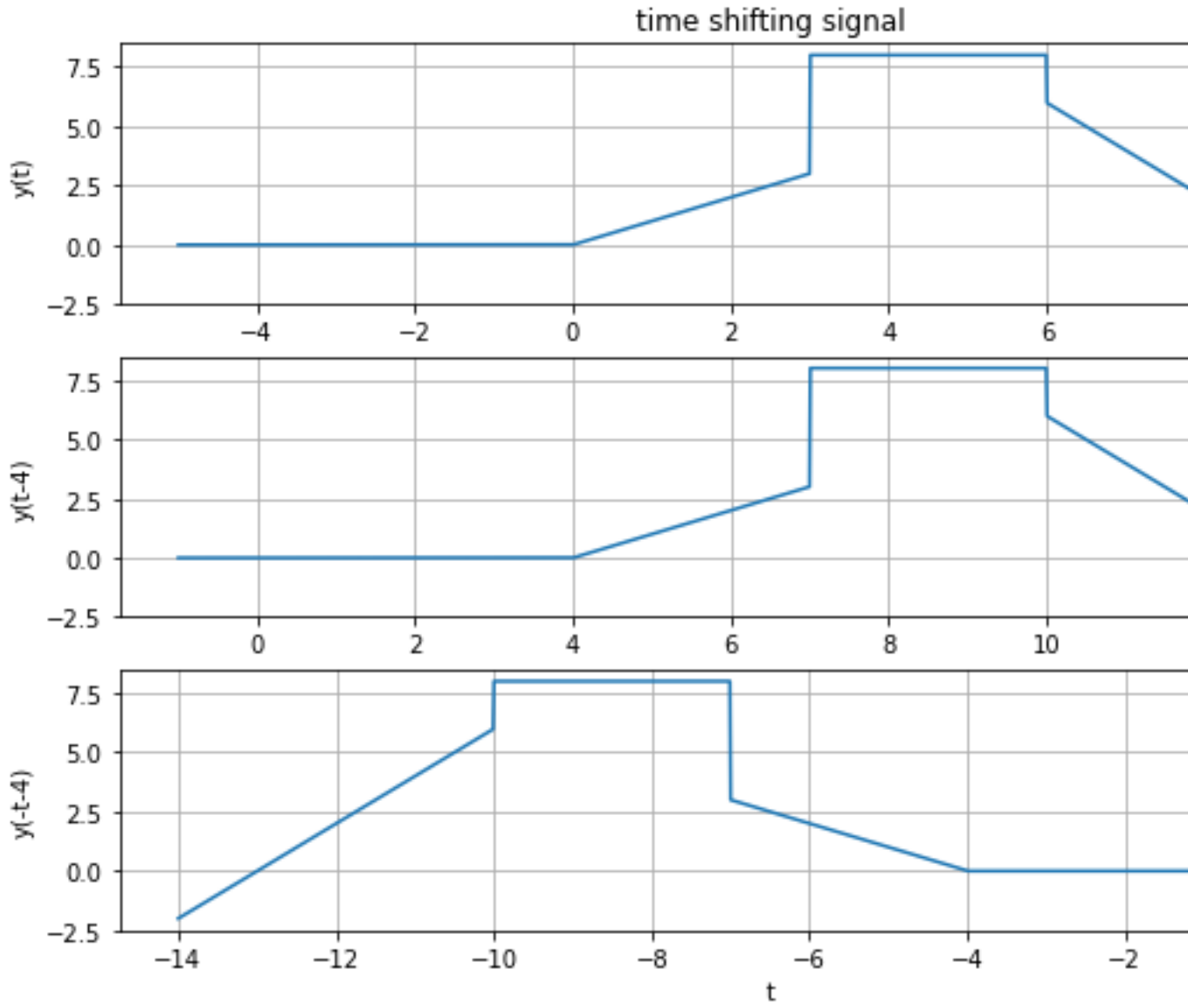
5 Part 3

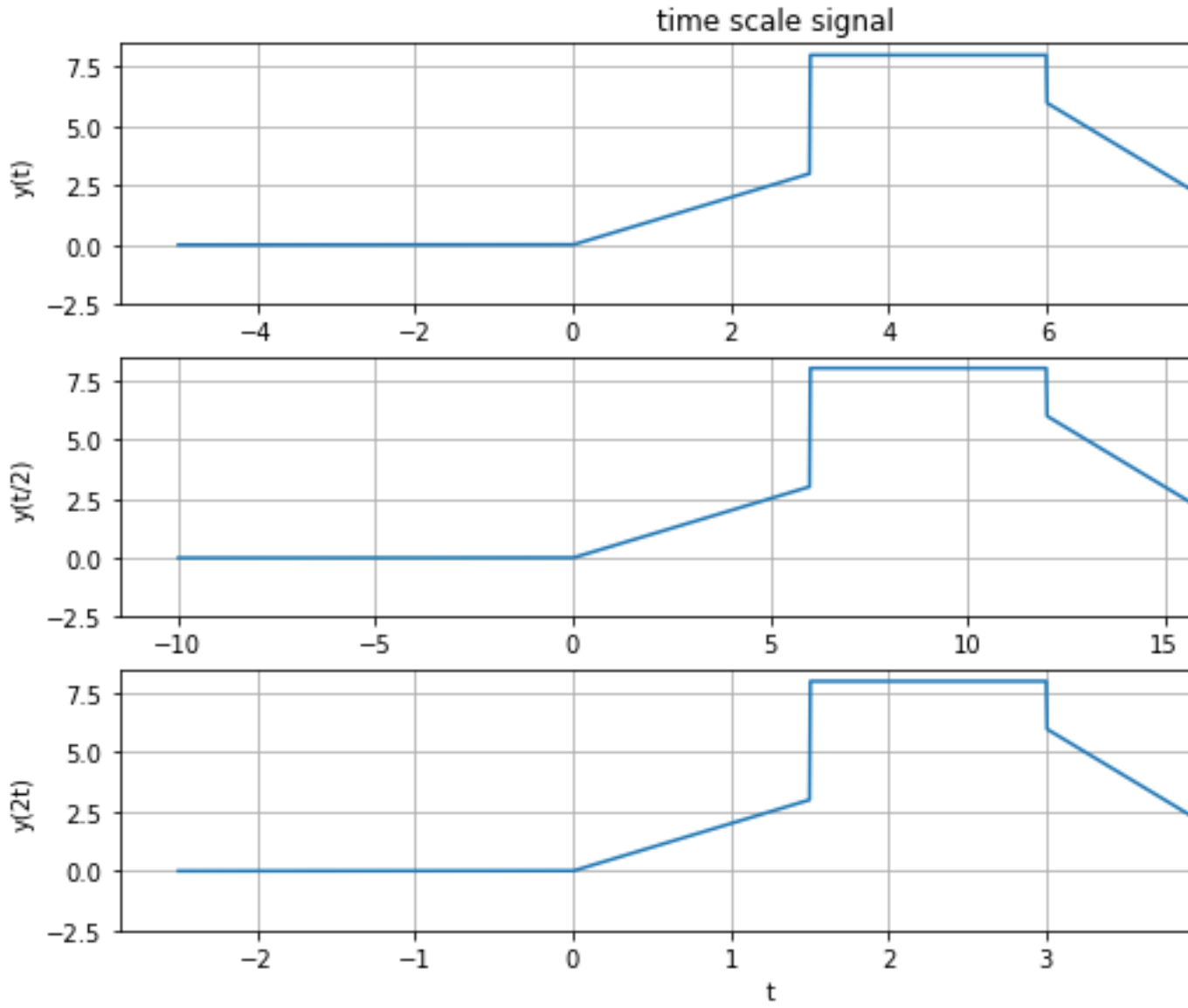
At the last part we were asked to Apply a time reversal, time-shift, time scale as well as a derivative with respect to time for the functions, which the plots will be included, however the derivative on the ramp function was not very accurate.

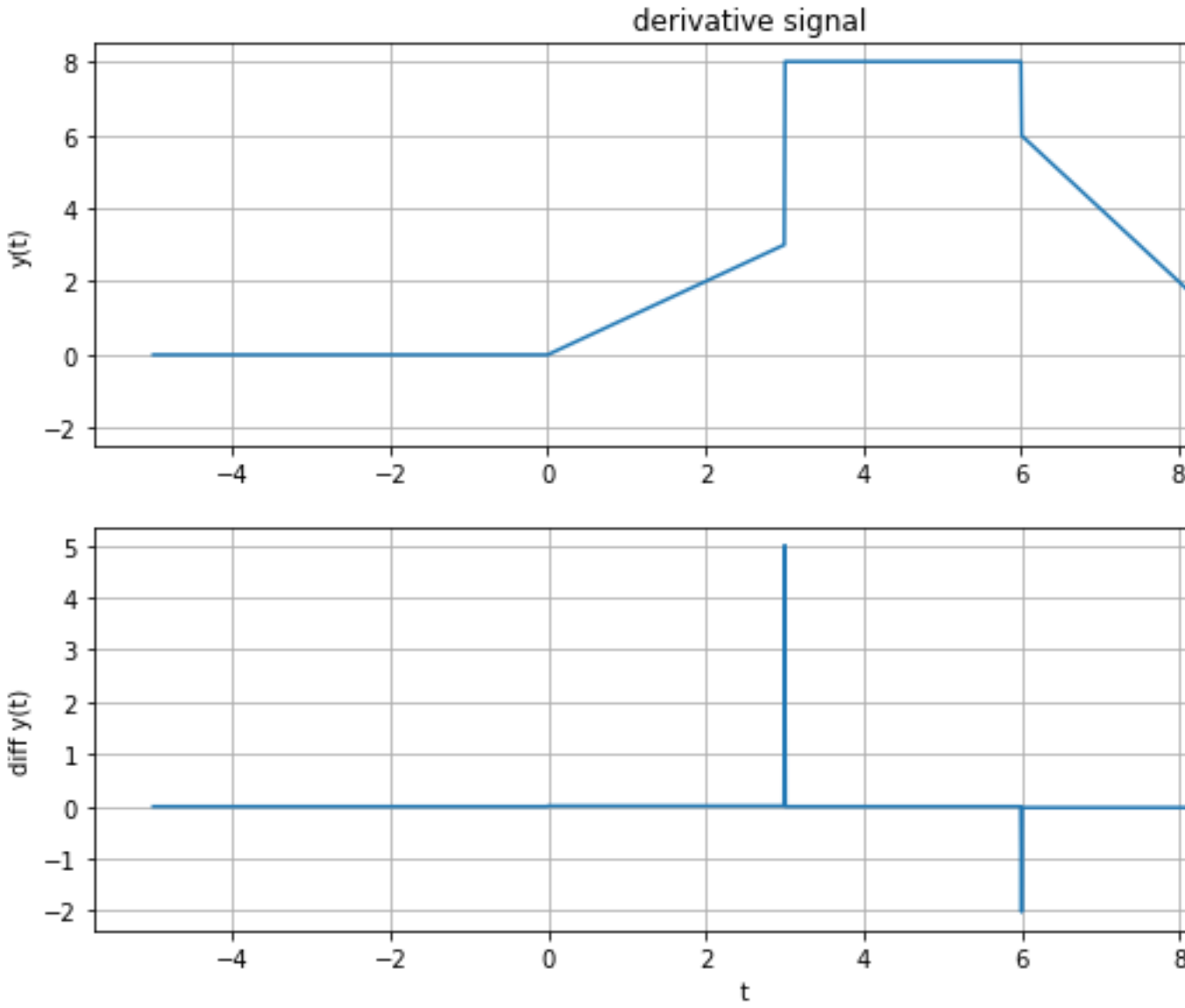
```
1
2
3 import numpy as np
4 import matplotlib . pyplot as plt
```

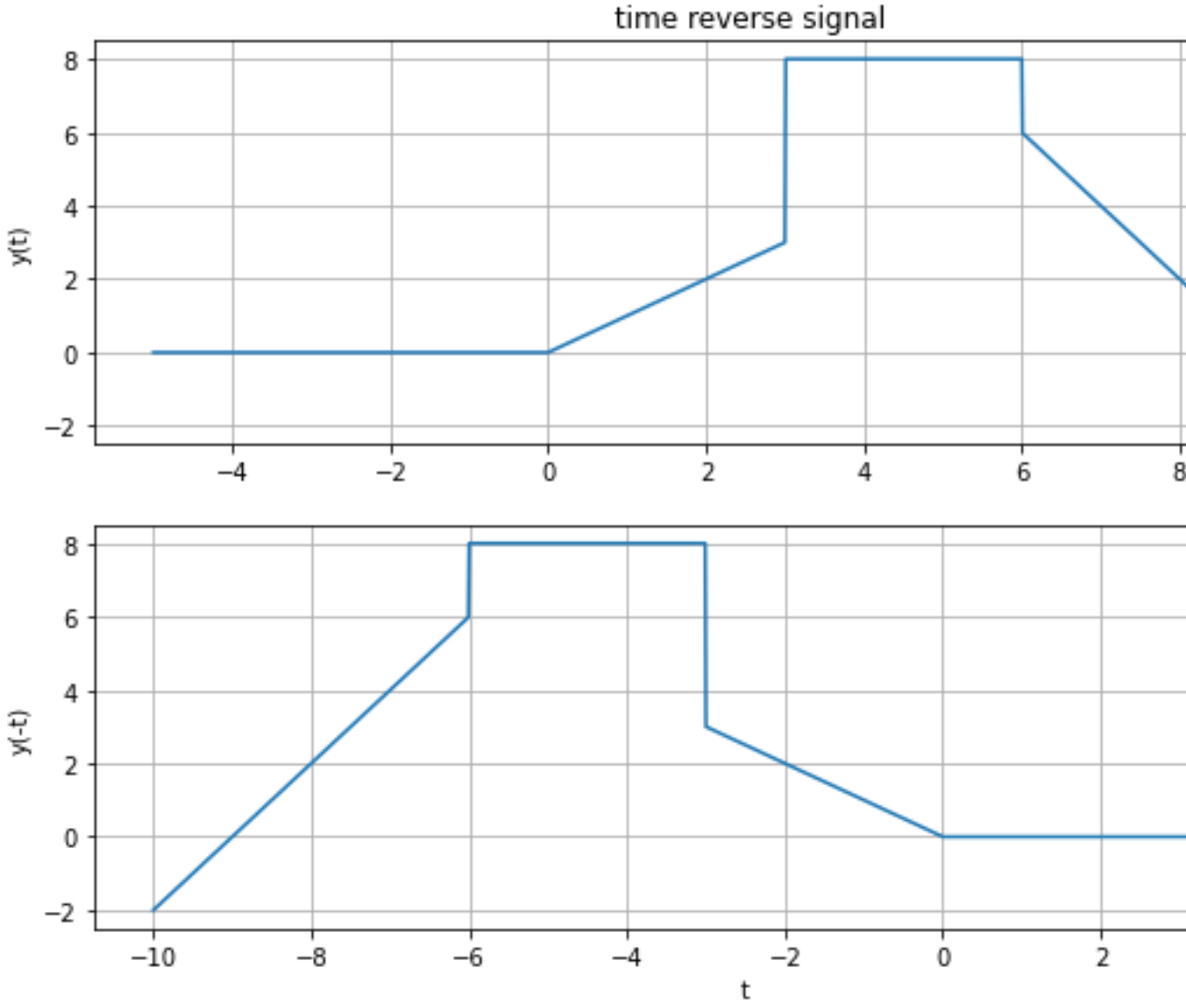
```
5
6
7  steps = 1e-2
8  t = np. arange (-5, 10 + steps , steps )
9
10
11 def step (t,tn):
12     y = np. zeros (t. shape )
13     for i in range ( len (t)):
14         if t[i] >= -tn:
15             y[i] = 1
16     return y
17
18 def ramp (t, tn, m):
19     y = np. zeros (t. shape )
20     for i in range ( len (t)):
21         if t[i] >= -tn:
22             y[i] = m * (t[i] + tn)
23     return y
24
25 y= ramp(t,0,1) + ramp(t,-3,-1) + 5 * step (t,-3) -2 * step (t,-6) +
       ramp(t,-6,-2)
26
27 #Task 1
28 yf = np.flip(y)
29 tf = np.flip(-t)
30 plt. figure ( figsize = (10 , 7))
31 plt. subplot (2, 1, 1)
32 plt. plot (t, y)
33 plt. grid ()
34 plt. ylabel ('y(t)')
35 plt. title (' time reverse signal')
36 plt. subplot (2, 1, 2)
37 plt. plot (tf, yf)
38 plt. grid ()
39 plt. ylabel ('y(-t)')
40 plt. xlabel ('t')
41
42
43 #Task 2
44 td = t + 4
45 tdf = np.flip(-td)
46 plt. figure ( figsize = (10 , 7))
47 plt. subplot (3, 1, 1)
48 plt. plot (t, y)
49 plt. grid ()
50 plt. ylabel ('y(t)')
51 plt. title ('time shifting signal')
52 plt. subplot (3, 1, 2)
```

```
plt. plot (td, y)
plt. grid ()
plt. ylabel ('y(t-4)')
plt. subplot (3, 1, 3)
plt. plot (tdf, yf)
plt. grid ()
plt. ylabel ('y(-t-4)')
plt. xlabel ('t')

#Task 3
t1 = t*2
t2 = t/2
plt. figure ( figsize = (10 , 7))
plt. subplot (3, 1, 1)
plt. plot (t, y)
plt. grid ()
plt. ylabel ('y(t)')
plt. title (' time scale signal')
plt. subplot (3, 1, 2)
plt. plot (t1, y)
plt. grid ()
plt. ylabel ('y(t/2)')
plt. subplot (3, 1, 3)
plt. plot (t2, y)
plt. grid ()
plt. ylabel ('y(2t)')
plt. xlabel ('t')

#Task 4
yd = np.diff(y)
t3 = np. arange (-5, 10  , steps )
plt. figure ( figsize = (10 , 7))
plt. subplot (2, 1, 1)
plt. plot (t, y)
plt. grid ()
plt. ylabel ('y(t)')
plt. title (' derivative signal')
plt. subplot (2, 1, 2)
plt. plot (t3 , yd)
plt. grid ()
plt. ylabel ('diff y(t)')
plt. xlabel ('t')

plt. show ()
```


time shifting signal
y(t)
y(t-4)
y(-t-4)
t
7.5
5.0
2.5
0.0
−2.5
−4
−2
0
2
4
6
8
10
−14
−12
−10
−8
−6


time scale signal
y(t)
y(t/2)
y(2t)
t
7.5
5.0
2.5
0.0
−2.5
−4
−2
0
2
4
6
−10
−5
0
5
10
15
−2
−1
0
1
2
3


derivative signal
y(t)
8
6
4
2
0
−2
−4
−2
0
2
4
6
diff y(t)
5
4
3
2
1
0
−1
−2
−4
−2
0
2
4
6
t



6 Questions

1. Are the plots from Part 3 Task 4 and Part 3 Task 5 identical? Is it possible for them to match? Explain why or why not. 2. How does the correlation between the two plots (from Part 3 Task 4 and Part 3 Task 5) change if you were to change the step size within the time variable in Task 5? Explain why this happens. 3. Leave any feedback on the clarity of lab tasks, expectations, and deliverables.

1. They were not very accurate especially for diff for the ramp function.

2. If we change the step size of the function, we'll have an error as follows from python (ValueError: x and y must have same first dimension, but have shapes (150000,) and (1500,))

7 conclusion

Discuss briefly what you learned in this lab and whether or not you feel the lab was successful. Include any recommendations for future labs as this is a learning experience for all of us. Discuss any insights you gained from this lab and how that will affect future work. *Note: The bibliograhpy needs to be on its own page.*