

ECE 351-51

FEBRUARY 1, 2022

Lab 2

Submitted By :
Kennedy Beach

Contents

1	Introduction	2
2	Equations	2
3	Methodology	2
4	Results	3
5	Error Analysis	6
6	Questions	6
7	Conclusion	7

1 Introduction

This lab introduced user-defined functions in Python. The goal of this lab was to create our own ramp and step functions, and then use those functions to recreate a given plot. Shifting and scaling the time of the plot as well as plotting the derivative was further explored with our functions.

2 Equations

An equation for the given plot is stated below with $r(t)$ being the ramp function and $u(t)$ being the step function.

$$f(t) = r(t) - r(t - 3) + 5u(t - 3) - 2u(t - 6) - 2r(t - 6)$$

3 Methodology

The first part was to create a cosine function from time 0 to 10. The code is shown in listing 1 and the plot itself in Figure 1.

```
1 import numpy as np
2
3 steps = 1e-2
4 t1 = np.arange(0, 10 + steps, steps)
5
6 def func1(t):
7     y = np.cos(t)
8     return y
9
10 y = func1(t1)
```

Listing 1: Defining the function to plot $\cos(t)$ from $0 < t < 10$

The ramp and step functions were then defined as seen in listing 2. Another function was created to utilize them over a period of time as seen in listing 3. The plots of the functions are seen in Figure 2.

```
1 def u(t):
2     if t < 0:
3         return 0
4     if t >= 0:
5         return 1
6
7 def r(t):
8     if t < 0:
9         return 0
```

```

10     if t >= 0:
11         return t
12
13 def function(t, func):
14     y = np.zeros((len(t), 1))
15     for i in range(len(t)):
16         y[i] = func(t[i])
17     return y
18
19 t = np.arange(-5, 10 + steps, steps)
20 y = function(t, r)
21 z = function(t, u)

```

Listing 2: Defining and implementing step and ramp functions

The function to graph the given plot was then created and is seen in listing 3. The recreated plot is seen in Figure 3.

```

1 def func2(t):
2     y = np.zeros((len(t), 1))
3     for i in range(len(t)):
4         y[i] = r(t[i]) - r(t[i]-3) + 5*u(t[i]-3) - 2*u(t[i]-6) - 2*
5         r(t[i]-6)
6
7     return y

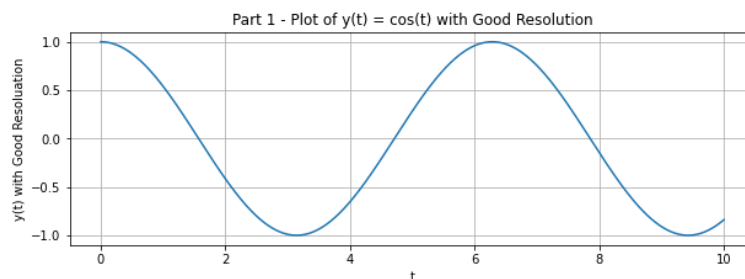
```

Listing 3: Defining the given plot function

The plot function was then shifted, scaled, and reversed. The plots for these are seen below in Figure 4.

Next, the derivative of $f(t)$ was drawn and then plotted through Python as seen in Figure 5.

4 Results

Figure 1: $y(t) = \cos(t)$

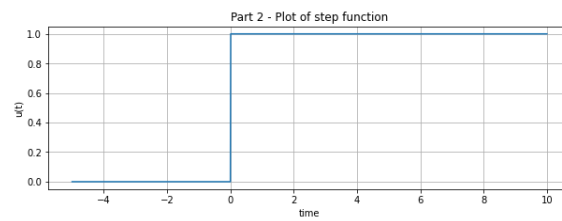
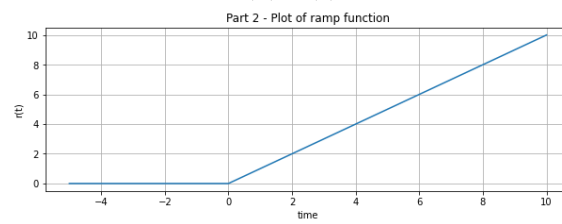
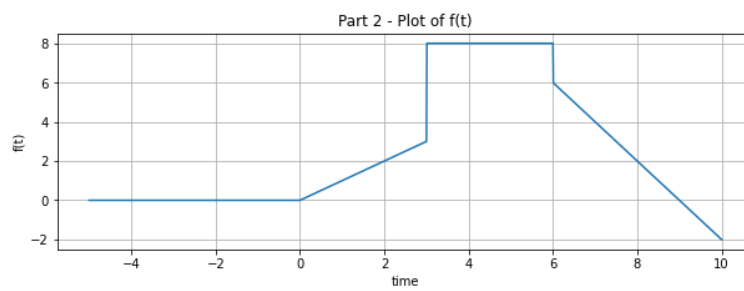
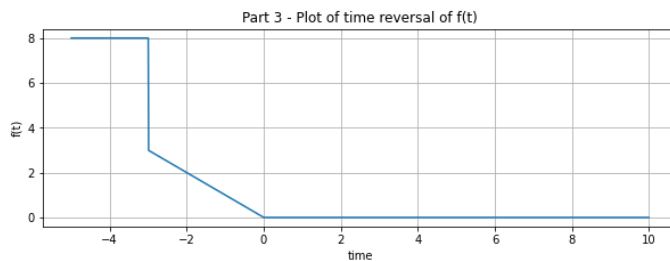
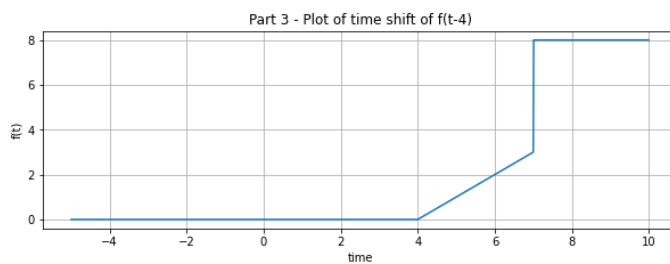
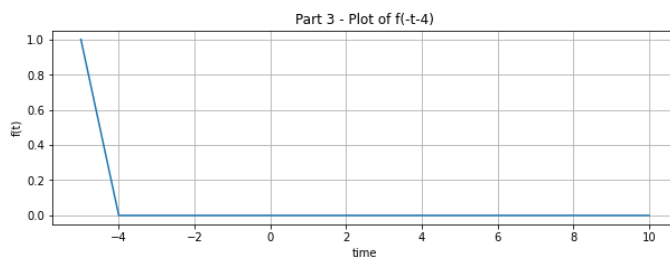
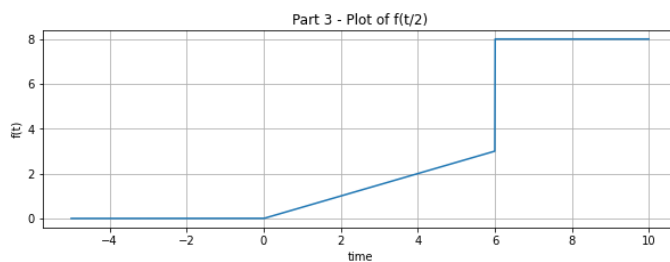
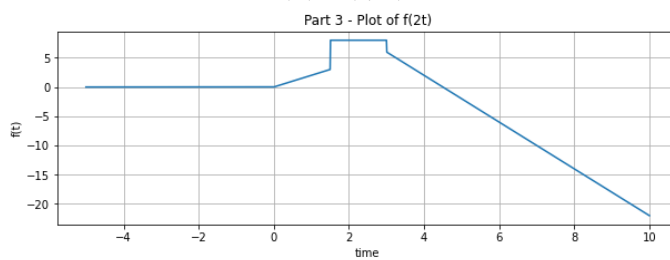
(a) $u(t)$ (b) $r(t)$

Figure 2: Step and Ramp function plots

Figure 3: $f(t)$



(a) Time Reversal

(b) $f(t - 4)$ (c) $f(-t - 4)$ (d) $f(t/2)$ (e) $f(2t)$ Figure 4: Time negation, shift, and scaling of $f(t)$

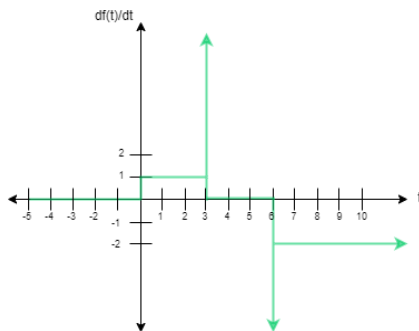
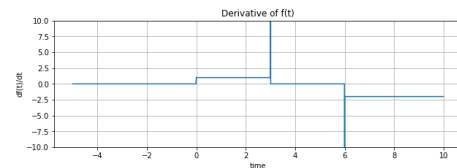
(a) Drawn derivative of $f(t)$ (b) $d(f(t))/dt$

Figure 5: Step and Ramp function plots

5 Error Analysis

I had difficulty plotting the derivative in Python at first. I just had to read through more documentation regarding the `numpy.diff()` function in order to use it properly. I also had some difficulty getting the step and ramp functions to work properly until I looked at their definitions more. I then added that before t was greater than or equal to zero, the function itself would be zero.

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.

They are not completely identical, but they could be if the scaling and resolution on both graphs were adjusted so that they are.

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.

If the steps become larger, then the resolution gets worse. The function between steps is "skipped" when plotting function with larger steps. This would make the plot from Task 5 look worse and more jagged.

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

Plotting the derivative of $f(t)$ was kind of confusing since `numpy.diff` had to be

used twice and I didn't realize that at first. Other than that, the lab was simple and easy to understand.

7 Conclusion

I learned how to define my own functions in Python and how to plot figures better. Overall, the lab was successful with a few obstacles that were eventually overcome. The Python and \LaTeX code are seen in https://github.com/Eniac618/ECE351_Code and https://github.com/Eniac618/ECE351_Reports respectively.