Project Report for ECE 351

Lab 4: System Step Response Using Convolution

Isaias Munoz

September 20, 2022

University of Idaho

Contents

1 Introduction

The purpose of this lab is to compute a systems step response as the forcing function and become familiar with convolution.

2 Methodology

2.1 Part 1

The first task that was assigned was to use Lab 3 user defined functions for a step function and write the following functions and plot them in a single figure.

$$h_1(t) = (e^-2t) * [u(t-2) - u(t-3)]$$

$$h_2(t) = u(t-2) - u(t-6)$$

$$h_3(t) = cos(w_0 * t) * u(t)$$

Figure 1 shows the code that was used to plot the functions. Where $w = 2 * pi * f$. It was simple enough since the user defined functions for step were already created in Lab 2 and therefore I just typed my equations and made them all as subplots to achieve one graph located in the results section.

```
###############################################################################
#Part 1 graphin the h(t)
steps = 1e-2 # Define step size
t = np . arange (-10 , 10 + steps , steps )
# q = np . arange (-1 , 14 + steps , steps )

def stepfunc ( t ) :  #creating step function
    y = np.zeros ( t.shape  ) #initializing y as all zeroes
    for i in range (len ( t ) ): #startingforloop
        if  t [ i ]<0:
            y[i]=0
        else:
            y[i]=1
    return y

def rampfunc ( t ) :
    y = np.zeros ( t . shape )
    for i in range (len ( t ) ):
        if  t [ i ]<0:
            y[i]=0
        else:
            y[i]=t[i]
    return y
####Better to make a function

#w=2pif
w=2*math.pi*.25

h1t=(np.exp(-t))*(stepfunc(t)-stepfunc(t-3))
h2t=stepfunc(t-2)-stepfunc(t-6)
h3t=np.cos(w*t)*stepfunc(t)
```

1

Figure 1: Using Lab 2 Step function code

2.2 Part 2

Part 2 task was to create a user define function in which it would convolve the functions created above with a unit step function. The unit step function would be a forcing function which the convolution of a random function with a step function is the integral of the random function.

```
def my_conv(f1,f2):

    Nf1=len(f1)
    Nf2=len(f2)   #Just defining a length  being passed into function

    x1extended=np.append(f1,np.zeros((1,Nf2-1))) #Combining both  and extendning
    x2extended=np.append(f2,np.zeros((1,Nf1-1))) #Combining botha nd extedning

    result=np.zeros(x1extended.shape) #iniitalizing new result array using one the new arra

    for i in range(Nf2+Nf1-2): #combin lengths of f1 nad f2
        result[i]=0#Is this initliazing the resultant array?
        for j in range(Nf1):
            #if(len(Nf1) and len(Nf2) > len(x1extended)):
            if(i-j+1>0):
                try:
                    result[i]+=x1extended[j]*x2extended[i-j+1]
                except:
                    print(i,j)
    return result
forcingfunc=stepfunc(t)

h1c = my_conv(h1t, forcingfunc)*steps
h2c = my_conv(h2t, forcingfunc) *steps
h3c = my_conv(h3t, forcingfunc)*steps

#not sure why times 2???
#the time axis should not matter where to starts
#my time axis needs to match with my new length of the convolution created which is doubled
#original is from -10 to 10 so a length of 20
#now after the convolution i double it and it reads -20 to 20 meaning length of 40
#now my starting point of my time array is still at -10 to 10 which is 20 therefore i
#have to double that bad boy as well or do it like below
tconvo = np . arange (2*t[0] , 2*t[len(t)-1]+steps , steps )
plt.figure(figsize=(10,7))
plt.subplot(3,1,1)
plt.plot(tconvo,h1c)
plt.grid()
plt.ylabel('Output')
plt.title('h1, h2, and h3 Step Response')
plt.subplot(3,1,2)
```

1

Figure 2: Using Lab 3 convolution code to convole with a step function

Calling the convolution function from Lab 3 and passing it my "random" function $h(t)$ and step-function non-shifted. I then continued to graph that. It's important to know I multiply by the steps so my output wasn't as tall and was scaled down. The trickiest part here was setting your time axis to be proper. Since before the convolution the time axis was defined and spans from -10 to 10 or a range of 20. After the convolution my axis needs to be double to make room for the shifts that are made and will span from -20 to

20 or a range of 40 which therefore require a new time range. Therefore *tconvo* is made and that way I am not plotting different ranges. This took me a while to grasp and was the most challenging task on this lab. The results of the convolution made with python are in Figure 6 under the result section.

Continuing task 2 we were also assigned to integrate $h(t)$ by hand and plot those results and compare them as well. I continued to solve by hand. Figure 3 and 4 shows the hand calculations and code to graph the hand calculations. .

$$Equation1 = .5[1 - e^-2t]u(t) - .5[1 - e^-2(t-3)]u(t-3)$$

$$Equation2 = (t-2)u(t-2) - (t-6)u(t-6)$$

$$Equation3 = (1/w)sin(wt)u(t)$$

1

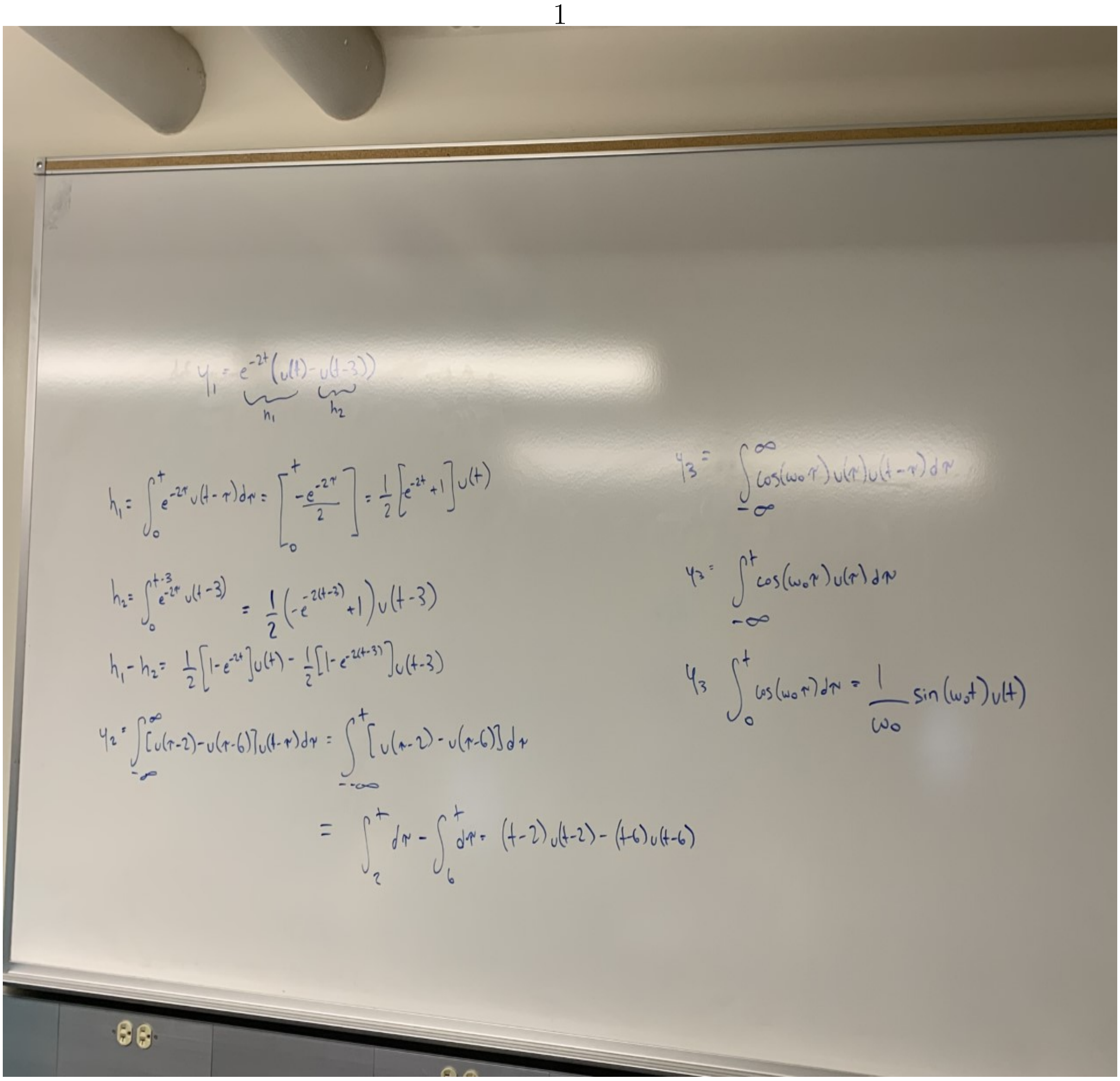


Figure 3: Solving integrals by hand

1

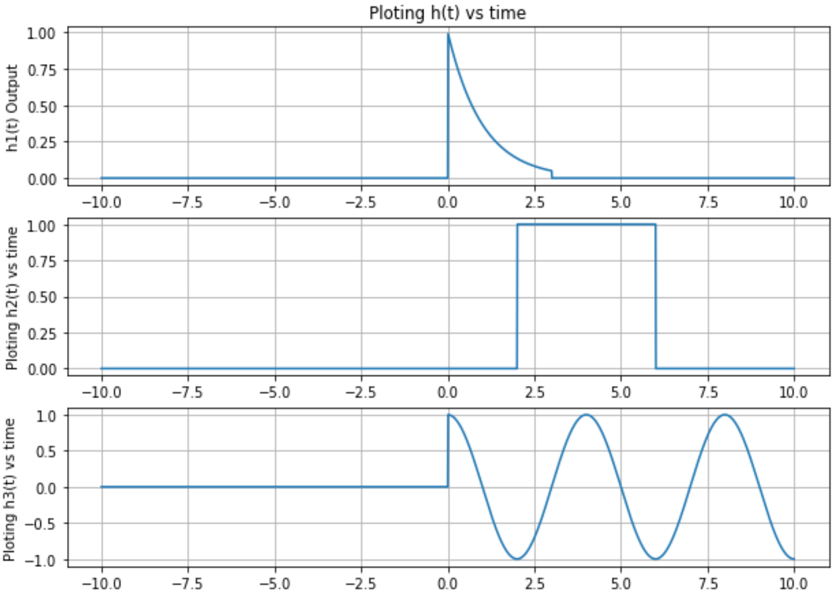
```
handh1= .5*(1-np.exp(-2*t))*stepfunc(t)-.5*(1-np.exp(-2*(t-3)))*stepfunc(t-3)
handh2=(t-2)*stepfunc(t-2)-(t-6)*stepfunc(t-6)
handh3=(1/w)*np.sin(w*t)*stepfunc(t)



plt.figure(figsize=(10,7))
plt.subplot(3,1,1)
plt.plot(t,handh1)
plt.grid()
plt.ylabel('Output')
plt.title('h1, h2, and h3 Step Response by hand')

plt.subplot(3,1,2)
plt.plot(t,handh2)
plt.grid()
plt.ylabel('Output')


plt.subplot(3,1,3)
plt.plot(t,handh3)
plt.grid()
plt.ylabel('Output')
```

Figure 4: Code for hand calculation graphs

The code shows the plotting of the functions that were derived from the hand integrals. For the time-step I used the original with a range-span of 20 from -10 to 10 and plotted. The results are below. Both graphs match like they should.

3 Results

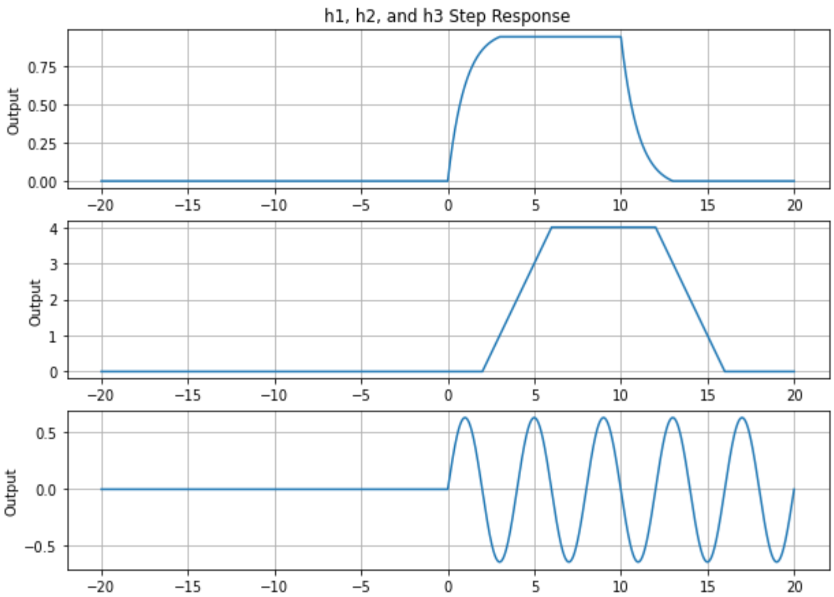
3.1 Part 1



1

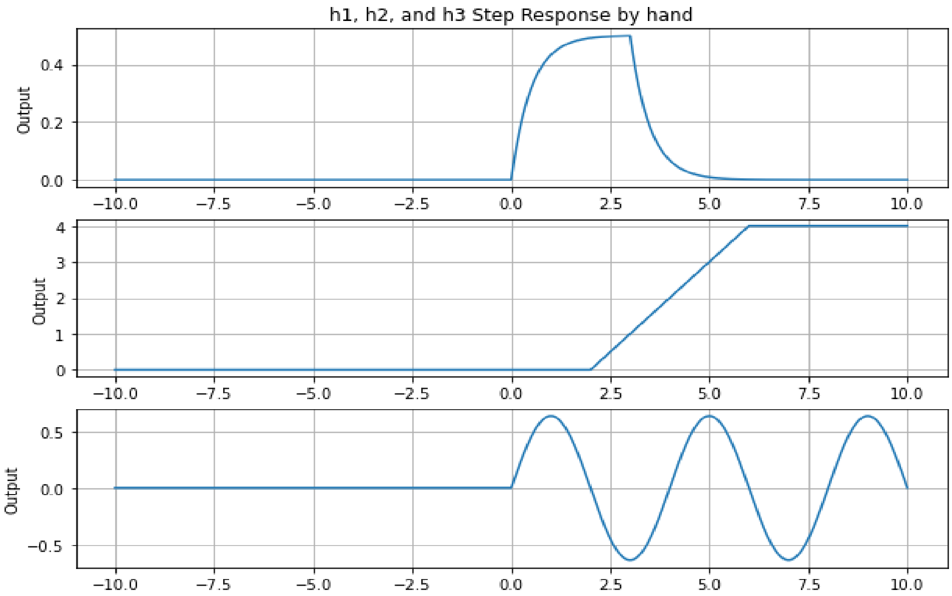
Figure 5: Combination of $h_1(t)$,$2_2(t)$,$h_3(t)$

3.2 Part 2



1

Figure 6: $h_1(t)$,$h_2(t)$,$h_3(t)$ step response



1

Figure 7: Hand calculation graphs of $h_1(t)$,$2_2(t)$,$h_3(t)$

4 Questions

The lab was straightforward after figuring out the time range axis needs to be the same for whatever you are plotting. I think stating the quantity number of graphs total needed in the deliverable overview would be helpful. A number given so we can double check we have the correct amount of graphs at least because I kind of got lost trying to find how many graphs we needed to have in our report but maybe it was just me. Aside from that everything was clear.