EE2703 : Applied Programming Lab
Assignment 7

Rakshit Pandey
EE20B106

April 10, 2022

1. AIM OF THE ASSIGNMENT

In this week's assignment we have analyze low pass and high pass filter circuits using the Laplace Transforms of the transfer functions, while using Symbolic Python (Sympy) Library of Python.

2. INTRODUCTION TO THE ASSIGNMENT

- We first install the python library Sympy for doing symbolic algebra in Python.
- We will then analyze the given Low pass and High pass filter analog ciruits in Laplace domain while using the powerful libraries like scipy.signal and Sympy
- We will then plot the output responses of the filters for different inputs.
- And finally we will analyze those responses which we obtained .

3. ASSIGNMENT QUESTIONS

Importing important libraries

We first import some important python libraries which will enable us to do this assignment. This is done in python as follows:

Code

```
from sympy import *
import scipy.signal as sp
import numpy as np
import matplotlib.pyplot as plt

```

1. QUESTION 1

In this question we are supposed to find the step response for the given Low pass filter analog circuit

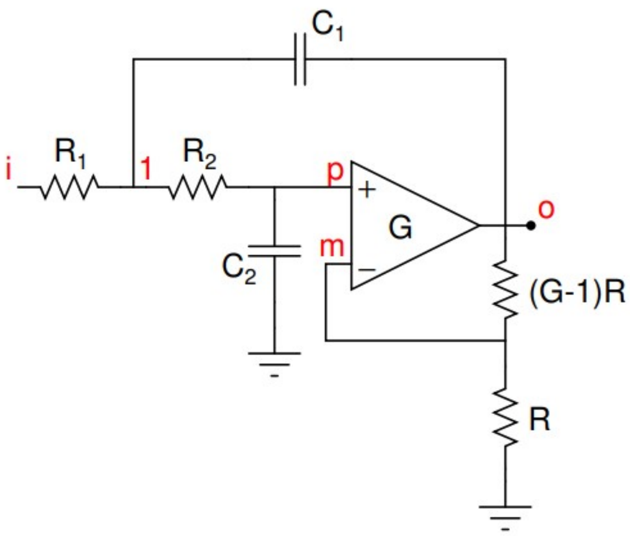

Figure 1: Low pass filter analog ciruit

We first define the lowpass filter circuit python function which takes R1,R2,C1,C2,G, and V_i(input voltage) as argument parameters.
The unit step function in time domain is equivalent to $\frac{1}{s}$ in the frequency domain.

We then create a function to convert sympy input in the form that can be used by scipy.signal as well and then we find the step response as follows:

Code

```
def lowpass(R1,R2,C1,C2,G,Vi):
  A = Matrix([[0,0,1,-1/G],[-1/(1+s*R2*C2),1,0,0],[0,-G,G,1],[-1/R1-1/R2-s*C1,1/
    R2,0,s*C1]])
  b = Matrix([0,0,0,-Vi/R1])
  V = A.inv()*b
  return A,b,V

#Creating a function to convert sympy function in a form so that it could be
    understood by scipy.signal
def num_den(expression):
  num,denom = expression.as_numer_denom()
  num = [float(i) for i in Poly(num,s).all_coeffs()]
  denom = [float(i) for i in Poly(denom,s).all_coeffs()]
  return num,denom

A1,b1,V1=lowpass(10000,10000,1e-9,1e-9,1.586,1/s)
Vo = V1[3]

num1,den1 = num_den(Vo)
H1 = sp.lti(num1,den1)
```

```python
t1,Vo1 = sp.impulse(H1,None,np.linspace(0,0.001,1000))

plt.figure(1)
plt.title("Vo vs t step function u(t)")
plt.xlabel("t")
plt.ylabel("Vo")
plt.plot(t1,Vo1)
plt.grid()
```

We obtain the following step reponse for the given lowpass filter with parameters as:

$$R1 = R2 = 10000\Omega$$
$$C1 = C2 = 1nF$$
$$G = 1.586$$
$$V_i = \frac{1}{s}$$

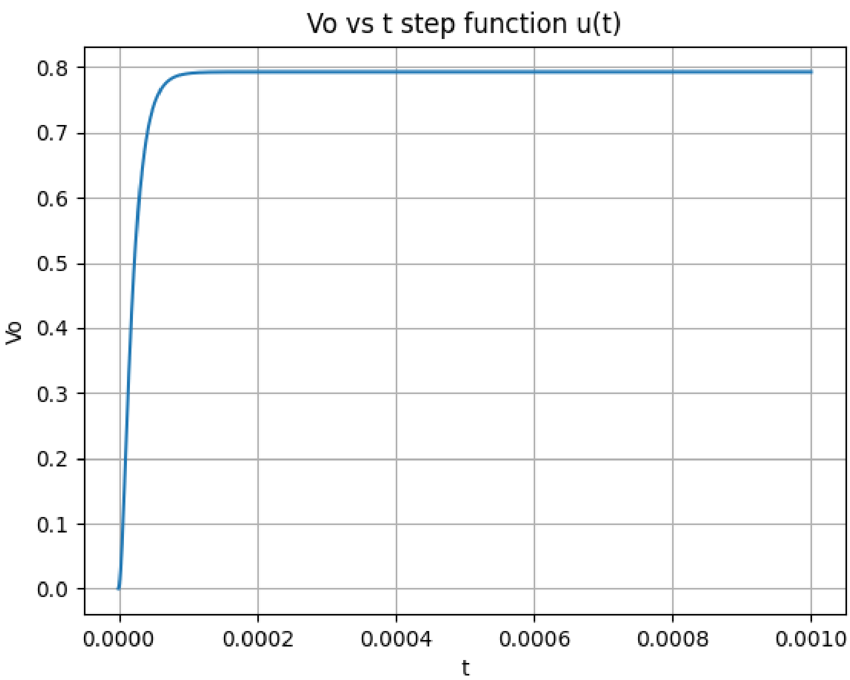

Figure 2: step response of the low pass filter

2. QUESTION 2

In this question we were given the input voltage as :

$$V_i(t) = (sin(2000\pi t) + cos(2 * 10^6 \pi t))u_0(t)$$

For this input voltage which is sum of two sinusoids with different frequencies, we were supposed to find the output response for the given low pass filter analog circuit. This is done in python as follows:

Code

```
A2,b2,V2 = lowpass(10000,10000,1e-9,1e-9,1.586,1)#For getting Low pass transfer function
Vo2 = V2[3]

num2,den2 = num_den(Vo2)
H2 = sp.lti(num2,den2) ##Low pass transfer function##
t2 = np.linspace(0,1e-2,100000)
Vi_t = np.sin(2000*PI*t2) + np.cos(2e6*PI*t2)
t2,Vo2,svec = sp.lsim(H2,Vi_t,t2)

plt.figure(2)
plt.title("Vo vs t for given Vi")
plt.xlabel("t")
plt.ylabel("Vo")
plt.plot(t2,Vi_t,label="Vin")
plt.plot(t2,Vo2,label="Vo")
plt.legend(loc="upper right")
plt.grid()
```

We obtain the following output response:

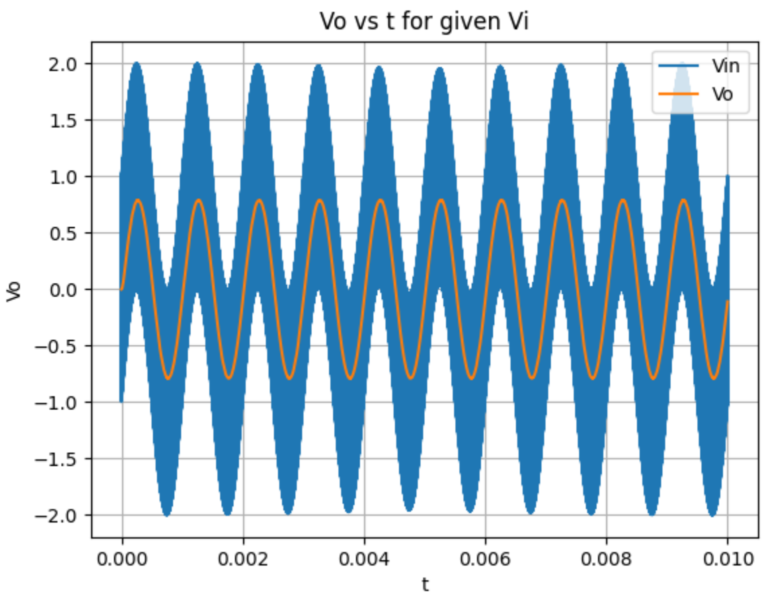

Figure 3: Output response of low pass filter for mixed sinusoid input

From the above plot we can clearly see that the output signal is a sinusoid with 1kHz frequency, which is obvious because the low pass filter attenuates the higher frequency component of the mixed sinusoid input and we get the output for the lower frequency component which in this case is $sin(2000\pi t)$.

3. QUESTION 3

In this question we were supposed to analyze the given high pass analog filter ciruit:

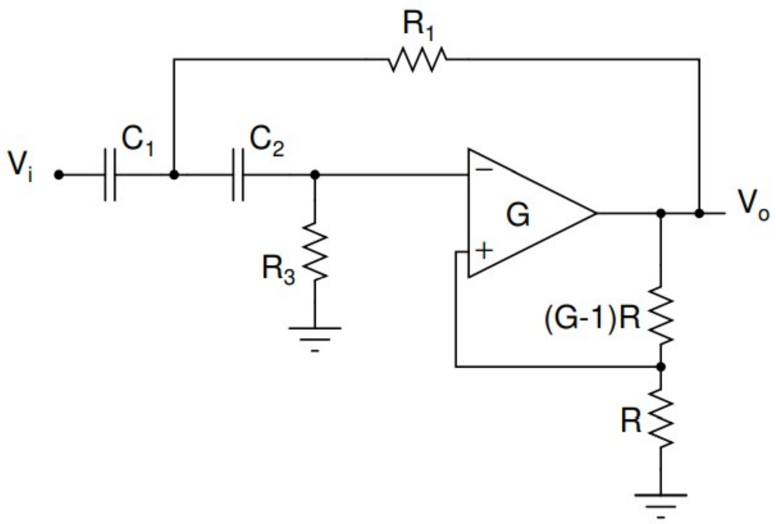

Figure 4: High pass filter analog circuit

for the following parameters:

$$R1 = R2 = 10000\Omega$$
$$C1 = C2 = 1nF$$
$$G = 1.586$$
$$V_i = 1$$

with the help of given parameters we obtain the transfer function for the given high pass filter circuit. Then we plot the magnitude response of the transfer function and phase response of the transfer function. This is done in python as follows:

Code

```
 #Creating a function for highpass filter
def highpass(R1,R3,C1,C2,G,Vi):
  A = Matrix([[0,-1,0,1/G],[s*C2*R3/(s*C2*R3+1),0,-1,0],[0,G,-G,1],[-s*C2-1/R1-s*
    C1,0,s*C2,1/R1]])
```

```
    b = Matrix([0,0,0,-Vi*s*C1])
    V = A.inv()*b
    return A,b,V

A3,b3,V3 = highpass(10000,10000,1e-9,1e-9,1.586,1) #For getting High pass
    transfer function
Vo3 = V3[3]
num3,den3 = num_den(Vo3)
H3 = sp.lti(num3,den3) ##High pass transfer function##
w,S,phi = H3.bode()

# Magnitude response of the given high pass filter
plt.figure(3)
plt.title("Magnitude response of a High Pass filter")
plt.xlabel("frequency")
plt.ylabel("Magnitude response")
plt.semilogx(w,S)
plt.grid()

#Phase response of the given high pass filter
plt.figure(4)
plt.title("Phase response of a High Pass filter")
plt.xlabel("frequency")
plt.ylabel("Phase response")
plt.semilogx(w,phi)
plt.grid()

```

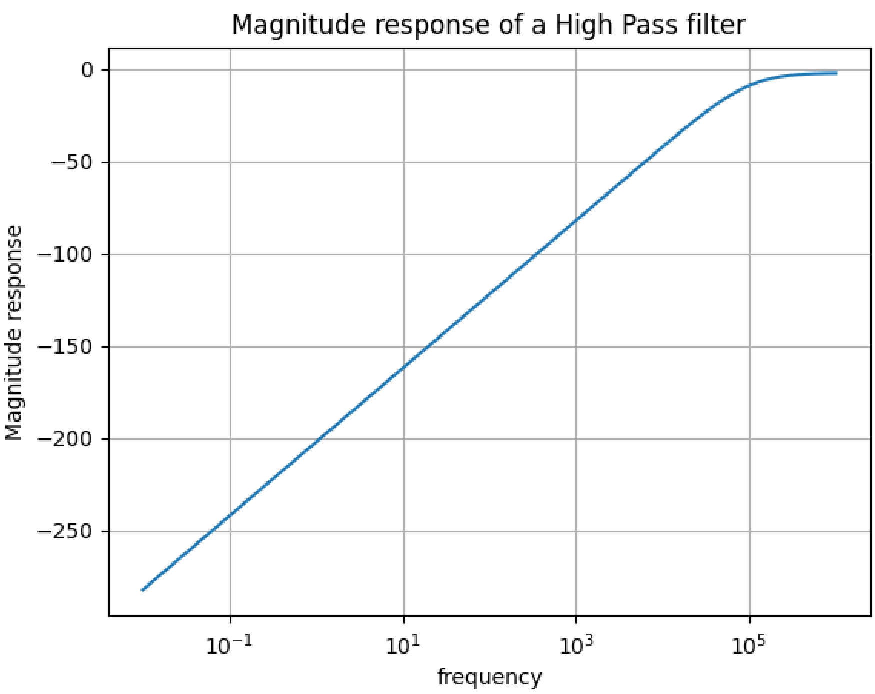

Figure 5: Magnitude response of High pass ciruit

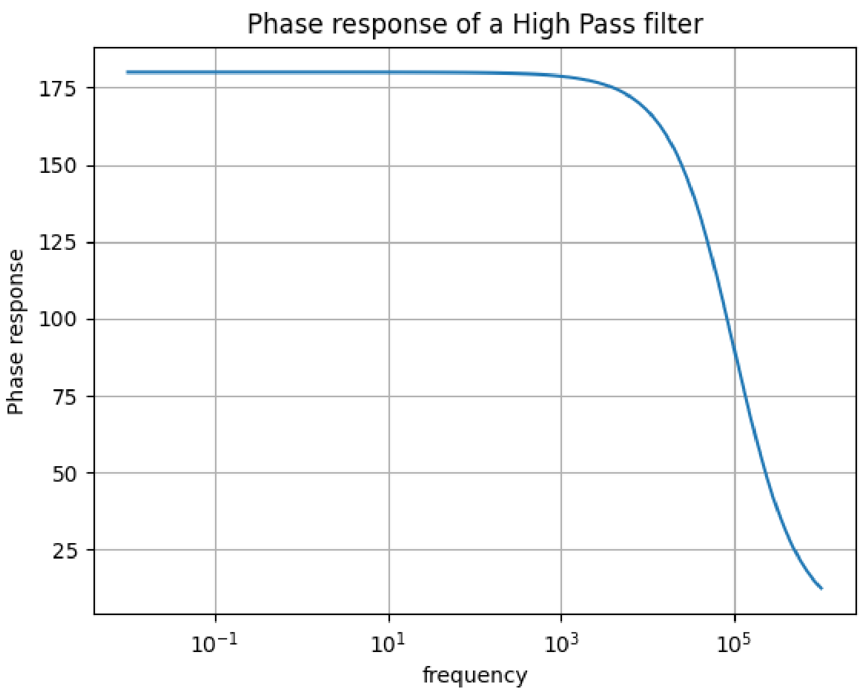

Figure 6: Phase response of High pass circuit

We now find the output response of this high pass filter circuit for the mixed sinusoid input signal.

Code

```
t3 = np.linspace(0,1e-5,1000)
Vi_t2 = np.sin(2000*PI*t3) + np.cos(2e6*PI*t3)
t3,Vo3,svec = sp.lsim(H3,Vi_t2,t3)

#Output response of the high pass filter to the sum of sinusoids input
plt.figure(5)
plt.title("Vo vs t for given Vi")
plt.xlabel("t")
plt.ylabel("Vo")
plt.plot(t3,Vo3,label="Vo")
plt.legend(loc="upper right")
plt.grid()
```

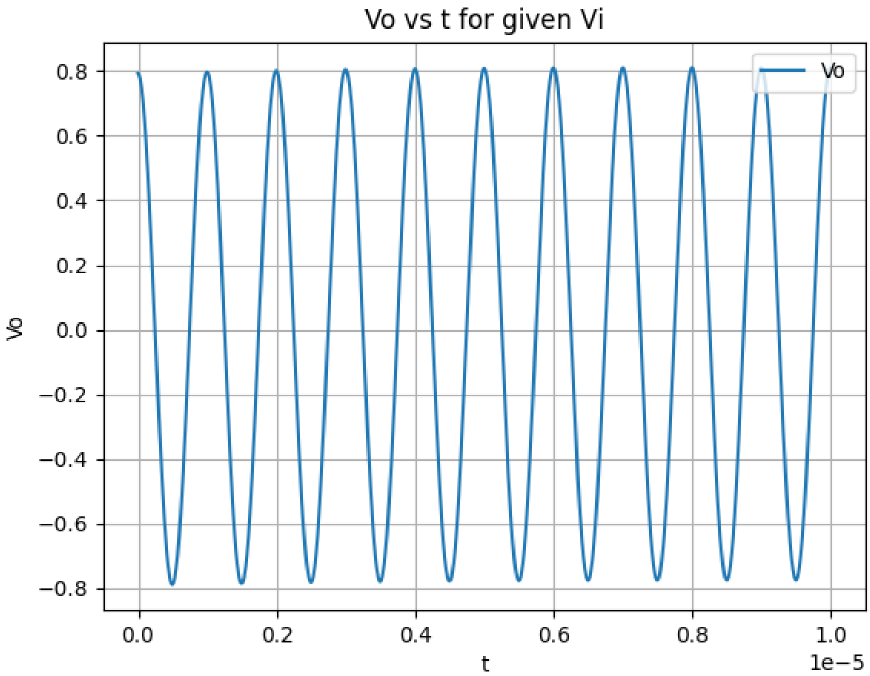

Figure 7: Output response of high pass filter for the mixed sinusoid input

From the above graph we can clearly observe that only the higher frequency component (1MHz frequency component) could pass through the given high pass filter while the lower frequency component got attenuated by the high pass filter.

4. QUESTION 4

In this question we were required to find the output response from the filters for a damping sinusoidal input signal.

For this we first define the function with frequency of sinusoid, the decay factor and the time as arguments and then we define the two damping sinusoid functions for the further analysis, This is done in python as follows:

Code

```
def damped_sin(freq,alpha,t):
  return (np.sin(freq*t)*np.exp(-alpha*t))

Vi_decay1 = damped_sin(2*PI,0.5,np.linspace(0,10,1000)) #Low frequency decaying
    sinusoid
Vi_decay2 = damped_sin(2*PI*1e5,0.5,np.linspace(0,1e-4,1000)) # High frequency
    decaying sinusoid
```

The two defined functions are:

$$V_{i1}(t) = (sin(2\pi t) * e^{-0.5t})u_0(t)$$

$$V_{i2}(t) = (sin(2\pi * 10^5 t) * e^{-0.5t})u_0(t)$$

We will start by analyzing the outputs for the first damped sinusoid function with the frequency of 1Hz.

Code

```
#low pass response
t5,Vo5,svec = sp.lsim(H2,Vi_decay1,np.linspace(0,10,1000))

#High pass response
t7,Vo7,svec = sp.lsim(H3,Vi_decay1,np.linspace(0,10,1000))

#response to decaying sinusoids for lowpass filter for lower frequency sinusoid
plt.figure(6)
plt.title("Low pass filter response of decaying sinusoids")
plt.xlabel("t")
plt.ylabel("Response")
plt.plot(t5,Vi_decay1,label="1Hz frequency input")
plt.plot(t5,Vo5,label="Vo for 1Hz frequency")
plt.legend()
plt.grid()

#response to decaying sinusoids for highpass filter for lower frequency sinusoid
plt.figure(8)
plt.title("High pass filter response of decaying sinusoids")
plt.xlabel("t")
plt.ylabel("Response")
plt.plot(t7,Vi_decay1,label="1Hz frequency input")
plt.plot(t7,Vo7,label="Vo for 1Hz frequency")
plt.legend()
plt.grid()
```

We obtain the following output responses for the given low frequency decaying sinusoid:

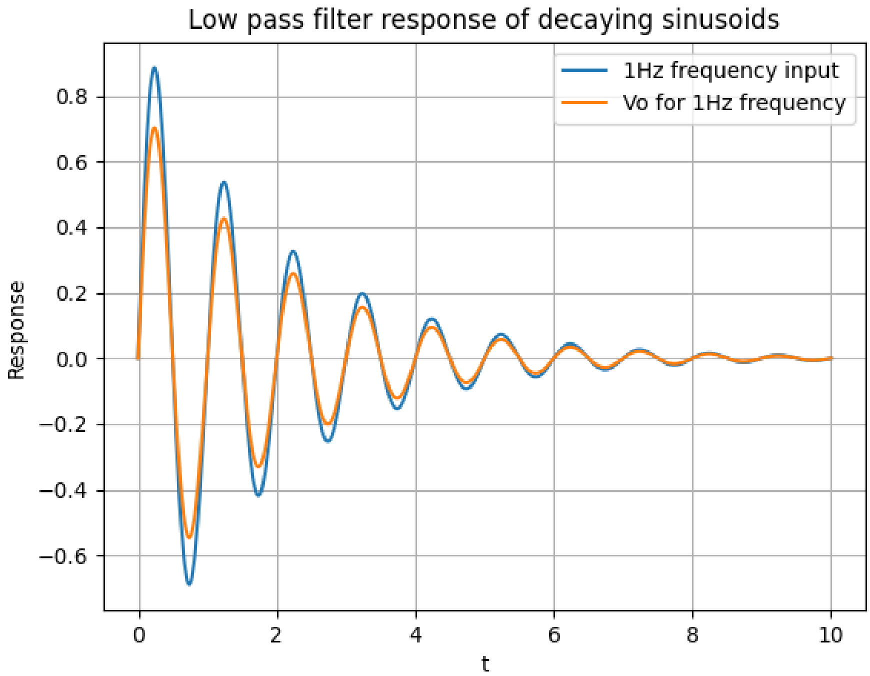

Figure 8: Output response of low pass filter for low frequency decaying sinusoid

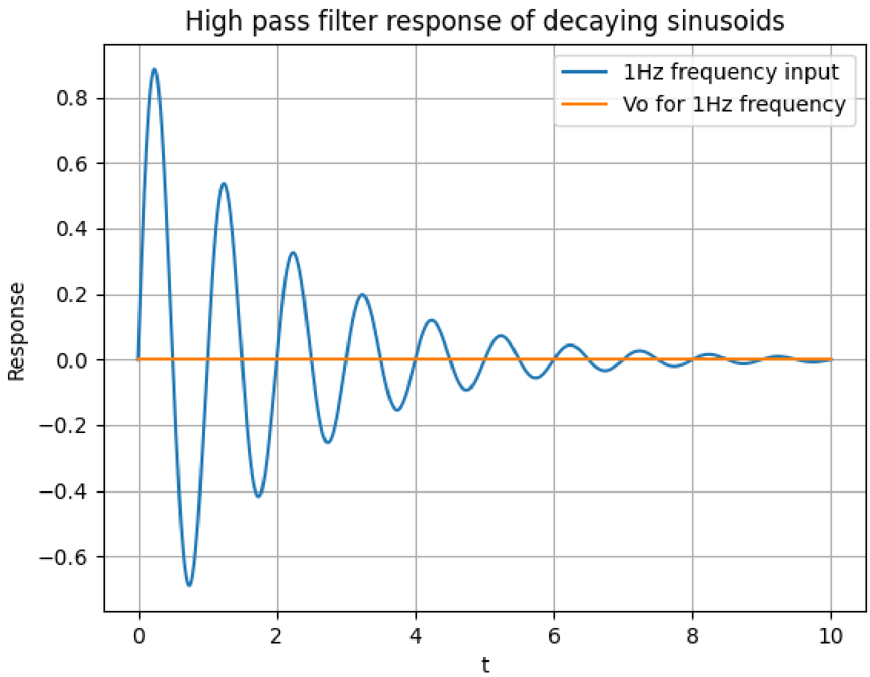

Figure 9: Output response of high pass filter for low frequency decaying sinusoid

From the output graphs we can clearly see that in first case where we were using a low pass filter, low frequency signal results in the given output, whereas in second case when low frequency signal is passed through a high pass filter, the low frequency component gets completely attenuated.

Now we will follow the same procedure in case of high freuency decaying sinusoid signal.

Code

```
t6,Vo6,svec = sp.lsim(H2,Vi_decay2,np.linspace(0,1e-4,1000))
t8,Vo8,svec = sp.lsim(H3,Vi_decay2,np.linspace(0,1e-4,1000))

#response to decaying sinusoids for lowpass filter for higher frequency sinusoid
plt.figure(7)
plt.title("Low pass filter response of decaying sinusoids")
plt.xlabel("t")
plt.ylabel("Response")
plt.plot(t6,Vi_decay2,label="10kHz frequency input")
plt.plot(t6,Vo6,label="Vo for 10kHz frequency")
plt.legend()
plt.grid()

#response to decaying sinusoids for highpass filter for higher frequency sinusoid
plt.figure(9)
plt.title("High pass filter response of decaying sinusoids")
plt.xlabel("t")
plt.ylabel("Response")
plt.plot(t8,Vi_decay2,label="10kHz frequency input")
plt.plot(t8,Vo8,label="Vo for 10kHz frequency")
plt.legend()
plt.grid()
```

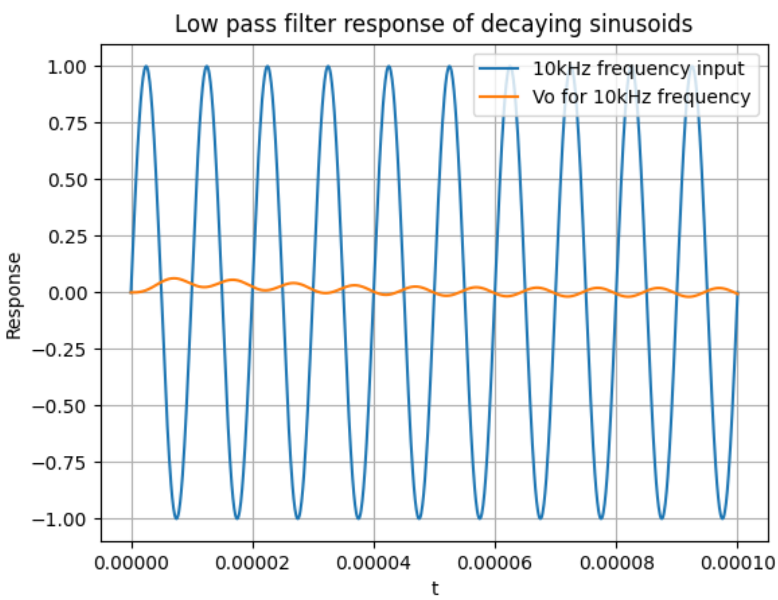

Figure 10: Output response of low pass filter for high frequency decaying sinusoid

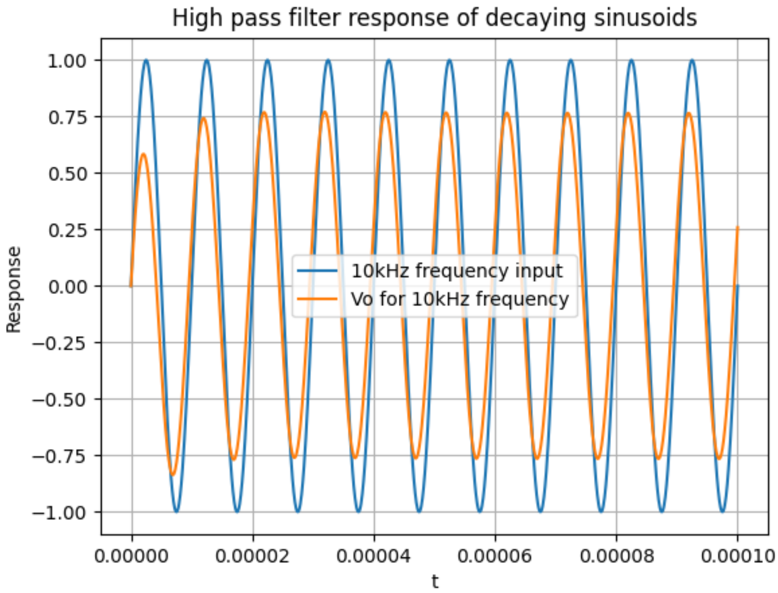

Figure 11: Output response of high pass filter for high frequency decaying sinusoid

From the output graphs we can clearly see that in first case where we were using a low pass filter, high frequency signal is almost completely attenuated, whereas in second case when high frequency signal is passed through a high pass filter, the high frequency component produced the output.

5. QUESTION 5

In this question we were supposed to obtain the step input response for the high pass circuit. In order to obtain the step input we just give unit step input to our high pass circuit transfer and obtain the output.

Code

```
A4,b4,V4=highpass(10000,10000,1e-9,1e-9,1.586,1/s)
Vo4 = V4[3]

num4,den4 = num_den(Vo4)
H4 = sp.lti(num4,den4)
t4,Vo4 = sp.impulse(H4,None,np.linspace(0,1e-3,1000))

#Step response of the high pass filter
plt.figure(10)
plt.title("Step response of the highpass filter")
plt.xlabel("t")
plt.ylabel("Vo")
```

```
plt.plot(t4,Vo4)
plt.grid()
plt.show()
```

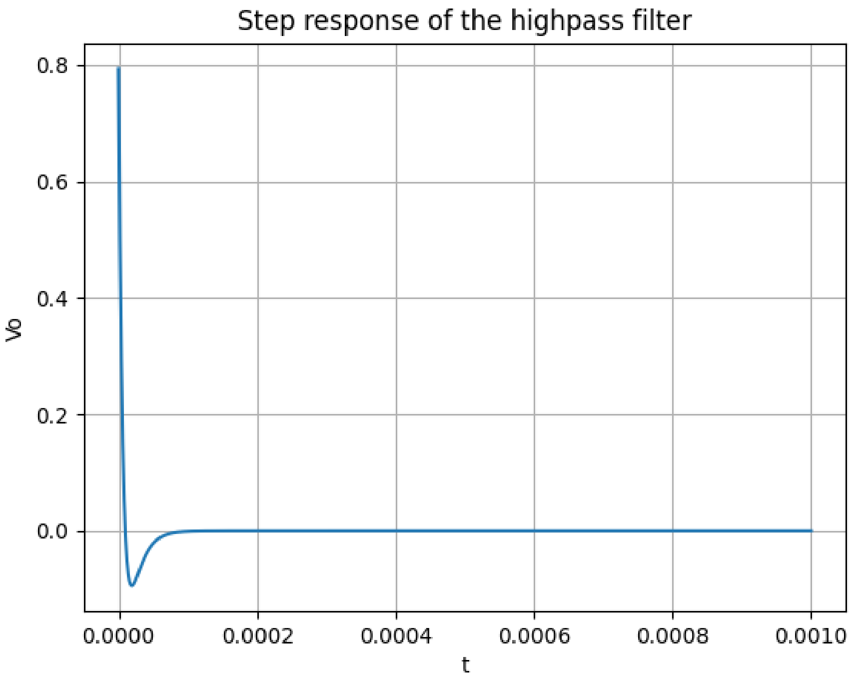

Figure 12: Step response of the high pass analog filter circuit

Initially at t=0, the capacitors will act as short, and hence a high non-zero value of output voltage is obtained. For, the steady state value of the V_i, the capacitor C_1 in the circuit will act as open circuit and allows no current to pass through. As a result of which we have a zero voltage at the output node.

CONCLUSIONS

Sympy (Symbolic Python) module provided us with a very convenient and powerful method to analyze the active High pass and low pass filters by using the Laplace transforms and analyzing their transfer functions. With the help of scipy.signal toolbox, sympy and matplotlib we analyzed the behaviour of high pass and low pass filters by plotting their step responses and their output responses to the mixed sinusoidal inputs. With the help of the plotted curves and the mixed sinusoidal inputs we could verify the fact that the low pass filter attenuates the high frequency component of input and allows only the low frequency components to pass through it, and vice-versa is also true.