Assignment 7

Sagnik Ghosh

April 2021

1 Introduction

In this week's assignment, we will explore Python's Symbolic Algebra Capabilities using the library called SymPy. We will use SymPy and SciPy's Signal Library to analyse active second order filters using Modified Nodal Analysis and Laplace Transforms.

2 Low Pass Filter

The Low Pass Filter Circuit provided to us in this assignment is governed by the following Modified Nodal Analysis Matrix Equation:

$$\begin{bmatrix} 0 & 0 & 1 & -1/G \\ \frac{-1}{sR_2C_2} & 1 & 0 & 0 \\ 0 & -G & G & 1 \\ \frac{-1}{R_1} - \frac{1}{R_2} - s*C_1 & \frac{1}{R_2} & 0 & sC_1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_p \\ V_m \\ V_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{-V_i(s)}{R_1} \end{bmatrix}$$

```
def lowpass(R1,R2,C1,C2,G,Vi):
    s=sympy.symbols('s')
    A=sympy.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=sympy.Matrix([0,0,0,Vi/R1])

    V=A.inv()*b

    return A, b, V
```

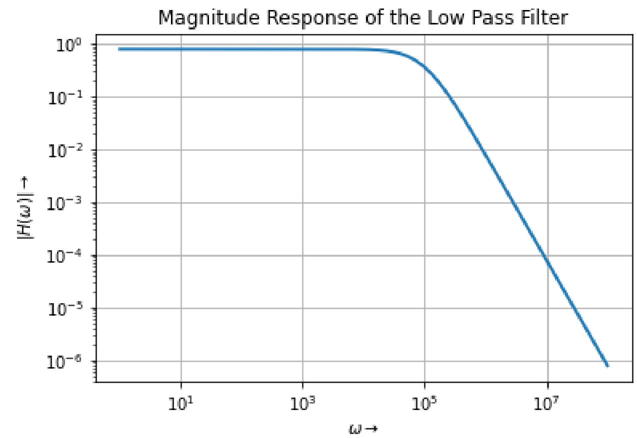

Figure 1: Low Pass Filter

3 Assignment

3.1 Problem 1

In this problem, we need to find the unit step response of the Low Pass Filter defined above.

```
# Question 1
A,b,V=lowpass(10000,10000,1e-9,1e-9,1.586,1)

print ('G=1000')
Vo=V[3]
Vo=sympy.simplify(Vo)
display(Vo)

s,t=sympy.symbols("s t")
t=sympy.Symbol("t",positive=True)
n,d = sympy.fraction(Vo)
n_sp,d_sp=(np.array(sympy.Poly(j,s).all_coeffs(),dtype=float) for j in (n,d))

print(n_sp,d_sp)
ts=np.linspace(0,0.001,8001)
t,x,svec=sp.lsim(sp.lti(n_sp,d_sp),np.ones(len(ts)),ts)
# Plot the absolute step response
plt.plot(t,np.abs(x),lw=2)
plt.grid(True)
plt.show()
```

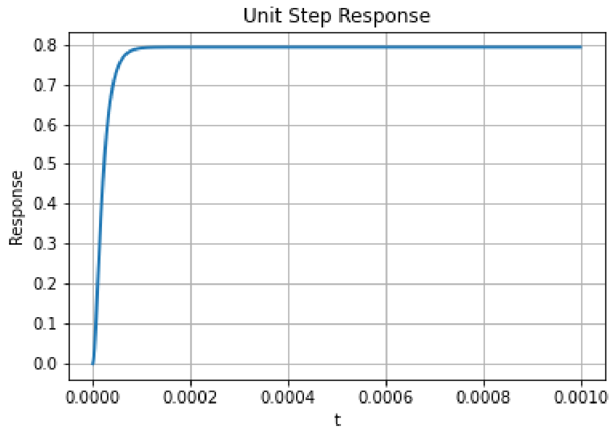

Figure 2: Unit Step Response of Low Pass Filter

3.2 Problem 2

In this section we consider an input given by the following equation and compute its output and plot it.

$$v_i(t) = \left(\sin(2000\pi t) + \cos\left(2\times 10^6\pi t\right)\right) u_0(t)$$

Let us first plot the input signal as follows,

```
ts=np.linspace(0,0.001,8001)
vi= np.sin(2000*np.pi*ts)+np.cos(2*10**6*np.pi*ts)
plt.plot(vi)
plt.xlabel('t')
plt.ylabel(r'$V_i$')
plt.title('Sum of Sinusoids')
plt.show()
```

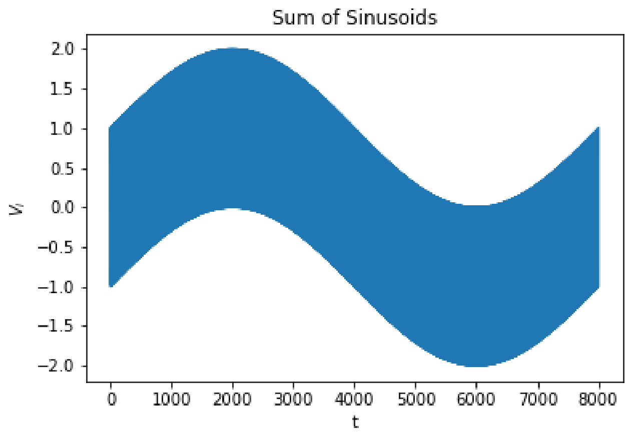

Figure 3: Input: Sum of Sinusoids

Let us now plot the output of the above input signal for the Low Pass Filter as follows,

```
t,x,svec=sp.lsim(sp.lti(n_sp,d_sp),vi,ts)
# Plot the lamdified values
plt.plot(t,x,lw=2)
plt.grid(True)
plt.xlabel('t')
plt.ylabel(r'$V_o$')
plt.title('Output for Low Pass Filter')
plt.savefig('Figure4.png')
plt.show()
```

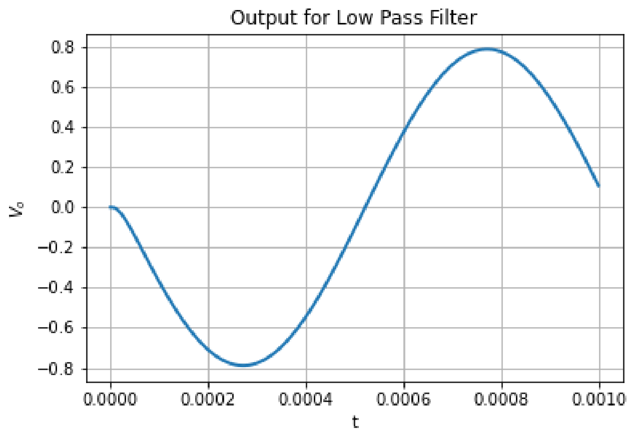

Figure 4: Output of Low Pass Filter

The output of the High Pass Filter can be plotted similarly and it looks as follows.

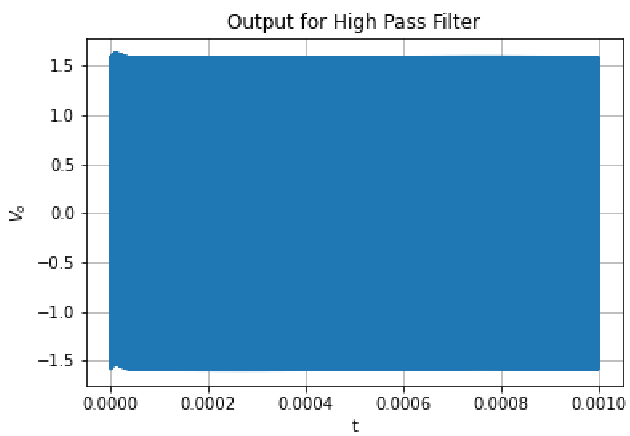

Figure 5: Output of the High Pass Filter

We can observe that the low pass filter nearly attenuates the 10^6 frequency component as expected and the high pass filter attenuates the 10^3 frequency component as expected.

3.3 Problem 3

In this section, we obtain the transfer function of the High Pass Filter using the Modified Nodal Analysis Matrix and plot its magnitude response. The MNA Matrix is as follows,

$$\begin{bmatrix} 0 & -1 & 0 & 1/G \\ \frac{s*C_2*R_3}{1+s*C_2*R_3} & 0 & -1 & 0 \\ 0 & G & -G & 1 \\ -s*C_2-\frac{1}{R_1}-s*C_1 & 0 & s*C_2 & \frac{1}{R_1} \end{bmatrix} \begin{bmatrix} V_1 \\ V_p \\ V_m \\ V_o \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -V_i(s)sC_1 \end{bmatrix}$$

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

# Question 3
A,b,V=highpass(10000,10000,1e-9,1e-9,1.586,1)

print ('G=1000')
Vo=V[3]
print("Vo (transfer function)")
display(Vo)
Vo=sympy.simplify(Vo)
print("Vo (transfer function) Simplified")
display(Vo)
w=np.logspace(0,8,801)
ss=1j*w
hf=sympy.lambdify(s,Vo,"numpy")
v=hf(ss)

plt.loglog(w,abs(v),lw=2)
plt.grid(True)
plt.xlabel(r'$\omega$')
plt.ylabel(r'Magnitude')
plt.title('Magnitude Response of High Pass Filter')
plt.show()
```

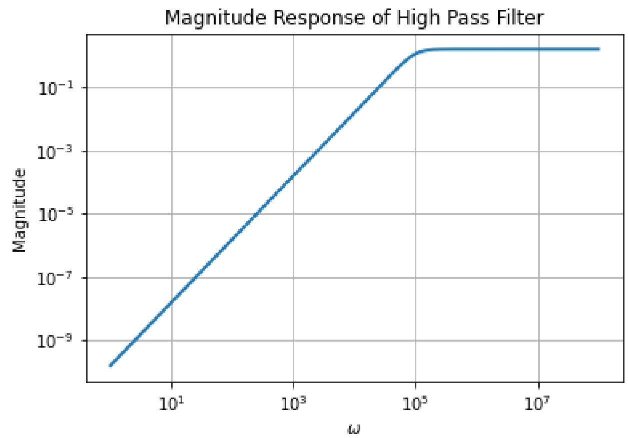

Figure 6: Magnitude Response of the High Pass Filter

3.4 Problem 4

In this section, we will consider the damped sinusoids and their response to both the high pass and the low pass filter. First, we will consider the High Frequency Damped Sinusoid given by,

$$f(t) = \cos\left(10^7 t\right) * e^{-3000t}$$

This is what the High Frequency Damped Sinusoid looks like when plotted.

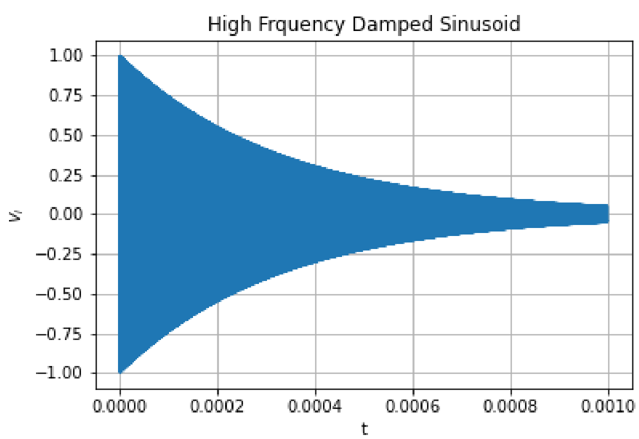

Figure 7: High Frequency Damped Sinusoid

Let us now plot this signal's output for the High Pass Filter. This can be done as follows,

```
A,b,V=highpass(10000,10000,1e-9,1e-9,1.586,1)

Vo=V[3]
Vo=sympy.simplify(Vo)
n,d = sympy.fraction(Vo)
n_sp,d_sp=(np.array(sympy.Poly(j,s).all_coeffs(),dtype=float) for j in (n,d))

t,x,svec=sp.lsim(sp.lti(n_sp,d_sp),vi,t)
plt.plot(t,x,lw=2)
plt.grid(True)
plt.xlabel('t')
plt.ylabel(r'$V_o$')
plt.title('Output for High Pass Filter')
plt.show()
```

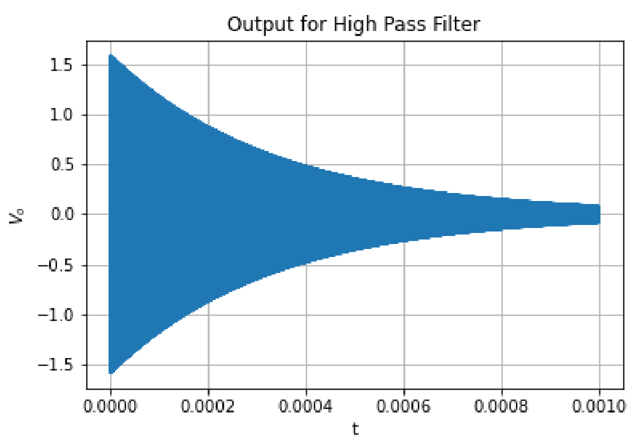

Figure 8: Output of High Pass Filter

Let us now plot its output for the Low Pass Filter. This can be done as follows,

```
A,b,V=lowpass(10000,10000,1e-9,1e-9,1.586,1)

Vo=V[3]
Vo=sympy.simplify(Vo)
n,d = sympy.fraction(Vo)
n_sp,d_sp=(np.array(sympy.Poly(j,s).all_coeffs(),dtype=float) for j in (n,d))

t,x,svec=sp.lsim(sp.lti(n_sp,d_sp),vi,t)
plt.plot(t,x,lw=2)
plt.grid(True)
```

```
plt.xlabel('t')
plt.ylabel(r'$V_o$')
plt.title('Output for Low Pass Filter')
plt.show()
```

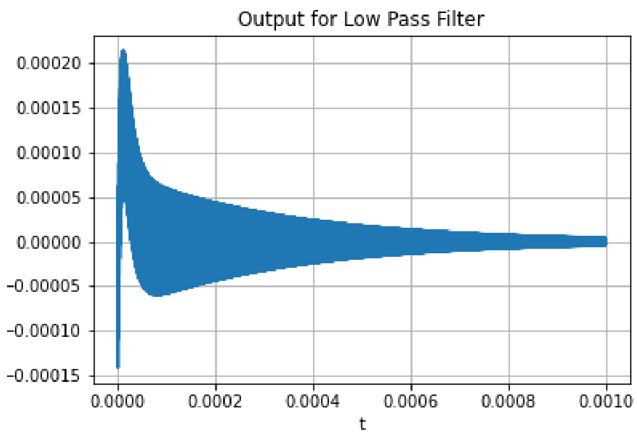

Figure 9: Output of Low Pass Filter

It can be noticed that the input signal has not been attenuated by the high pass filter but it has been almost completely attenuated by the low pass filter. This is to be expected as the input signal is of a high frequency.

Let us now take the input signal as a **low frequency** damped sinusoid. This is of the form,

$$f(t) = \cos\left(10^3 t\right) * e^{-1000t}$$

When we plot this signal, it looks like,

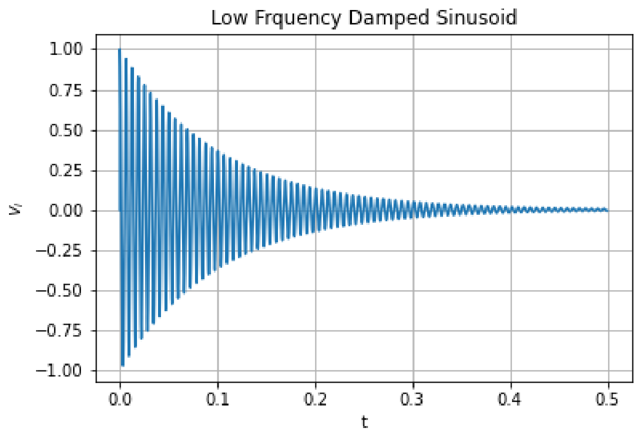

Figure 10: Low Frequency Damped Sinusoid

Let us now plot its output using the High Pass Filter.

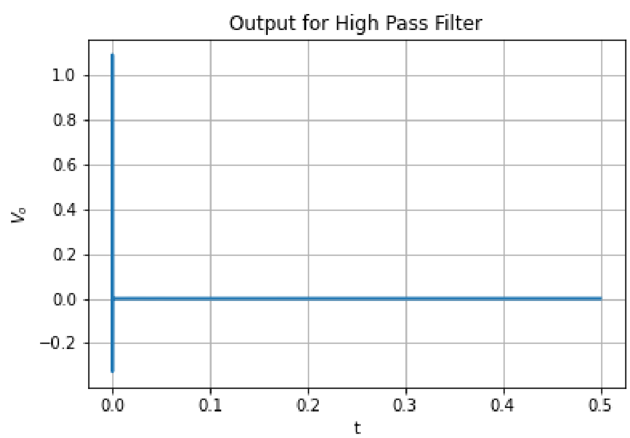

Figure 11: Output of High Pass Filter

Let us now plot its output using the Low Pass Filter.

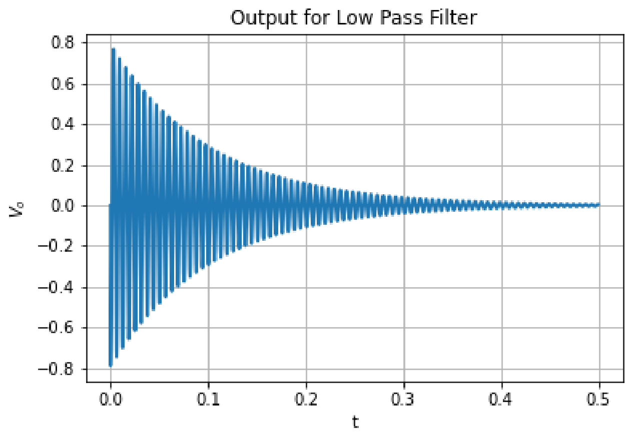

Figure 12: Output of Low Pass Filter

It can be noticed that the input signal has not been attenuated by the low pass filter but it has been almost completely attenuated by the high pass filter. This is to be expected as the input signal is of a low frequency.

3.5 Problem 5

In this problem, we will plot the step response of the High Pass Filter. This can be done as follows,

```
# Question 5
A,b,V=highpass(10000,10000,1e-9,1e-9,1.586,1)
Vo=V[3]
Vo=sympy.simplify(Vo)
s,t=sympy.symbols("s t")
t=sympy.Symbol("t",positive=True)
n,d = sympy.fraction(Vo)
n_sp,d_sp=(np.array(sympy.Poly(j,s).all_coeffs(),dtype=float) for j in (n,d))

print(n_sp,d_sp)
ts=np.linspace(0,0.001,8001)
t,x,svec=sp.lsim(sp.lti(n_sp,d_sp),np.ones(len(ts)),ts)
# Plot the lamdified values
plt.plot(t,x,lw=2)
plt.grid(True)
plt.xlabel('t')
plt.ylabel(r'$V_o$')
plt.title('Unit Step Response')
plt.savefig('Figure13.png')
```

```
plt.show()
```

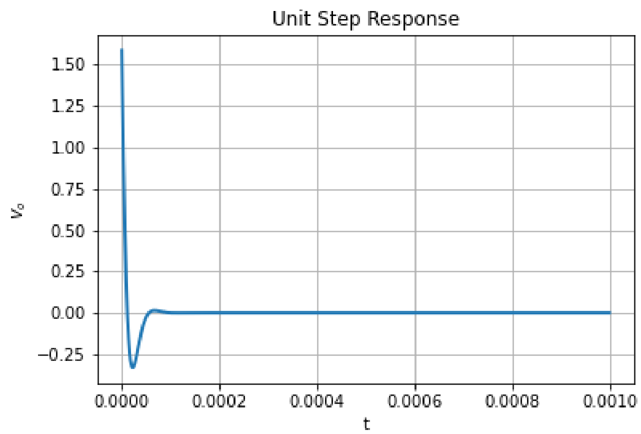

Figure 13: Step Response of High Pass Filter

4 Conclusion

- We used the SymPy Library to represent the transfer functions of both, a Low Pass Filter and a High Pass Filter and analyse them using several input signals.
- We first represented a Low Pass Filter symbolically using the MNA equation and plotted its transfer function. We noted that its Bandwidth was at around $\omega = 10^5$.
- Next, we found the unit step response of the Low Pass Filter. There was no ringing effect and hence we can conclude that the Quality Factor $Q < 1$.
- We then found the response of both, the Low Pass Filter and High Pass Filter to a sinusoid consisting of both, a low frequency and a high frequency component. As expected, the low pass filter attenuated the high frequency component and vice versa.
- We the plotted the response of a high frequency damped sinusoid to the low pass filter and the high pass filter. As expected, the high pass filter did not attenuate it while the low pass filter attenuated it significantly.
- We the plotted the response of a low frequency damped sinusoid to the low pass filter and the high pass filter. As expected, the low pass filter did not attenuate it while the high pass filter attenuated it significantly.

- Finally, we plotted the step response of the High Pass Filter. The Fourier Transform of the unit step function is mainly concentrated around the zero frequency and hence the high pass filter almost completely attenuates it.