Assignment 7

Gali Ushasri EE20B033

April 9, 2022

1 Introduction

This week's assignment involves the analysis of filters using laplace transforms. Python's symbolic solving library, sympy is a tool we use in the process to handle our requirements in solving Modified Nodal Analysis equations. Besides this the library also includes useful classes to handle the simulation and response to inputs. Coupled with scipy's signal module, we are able to analyse both High pass and low pass filters, both second order, realised using a single op amp

2 Low Pass Filter

2.1 Step Response

The low pass filter we use gives the following matrix after simplification of Modified Nodal Equations.

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

```
def Lowpass_filter(R1, R2, C1, C2, G, Vi):#Lowpass filter
    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*C
    b = sympy.Matrix([0,0,0,-Vi/R1])
    V = A.inv()*b
    return (A,b,V)
```

Using the lowpass function, we can obtain the impulse response in s-domain for the given values of components as follows :

```
A,b,V=Lowpass_filter(10000,10000,1e-9,1e-9,1.586,1)
Vo = V[3]
```

```
H = sympy_to_lti(Vo)
t = np.linspace(0,0.001,1000)
Vo = sp.step(H,T=t)
plot_graph(Vo[0], Vo[1],r't$\rightarrow$',r'$V_{o}\rightarrow$')
```

After obtaining the response in its symbolic representation, the following function is used to convert it into the polynomial representation compatible with the Signals toolbox of scipy The following step response is obtained.

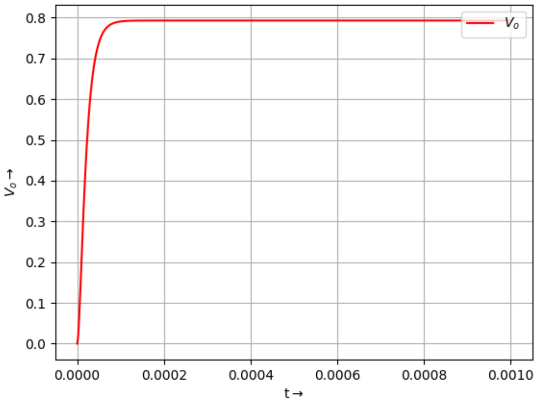

Figure 1: Step response of the lowpass filter

2.2 Response for a mixed frequency input

The input is, $Vi = (sin(2000\pi t) + cos(2 * 10^6 \pi t)) * u(t)$
Since the cutoff frequency $1/2\pi$ MHz, the system is expected to allow the low frequency sine component while attenuating the high frequency cosine component.

```
t = np.linspace(0,0.01,100000)
Vi = np.multiply((np.sin(2000*np.pi*t)+np.cos(2000000*np.pi*t)),np.heaviside(t,0.5))
Vo = sp.lsim(H,Vi,T=t)

p.plot(Vo[0],Vi,label=r'$V_{in}$')
plot_graph(Vo[0],Vo[1],r't$\rightarrow$',r'$V\rightarrow$')
```

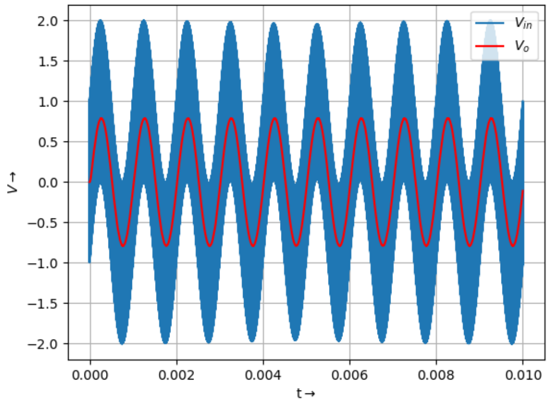

Figure 2: Output response to a mixed frequency input

3 High Pass Filter

The high pass filter we use gives the following matrix after simplification of Modified Nodal Equations.

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

The magnitude response, as expected, is that of a high pass filter, with cut-off frequency at $1/2\pi$ MHz is given below.

```
def Highpass_filter(R1,R3,C1,C2,G,Vi): #High pass filter
    s = sympy.symbols('s')
    A = sympy.Matrix([[0,0,1,-1/G],[-1/(1+1/(s*R3*C2)),1,0,0],[0,-G,G,1],[-s*C1-s*C2
    b = sympy.Matrix([0,0,0,-Vi*s*C1])
    V = A.inv()*b
    return (A,b,V)
```

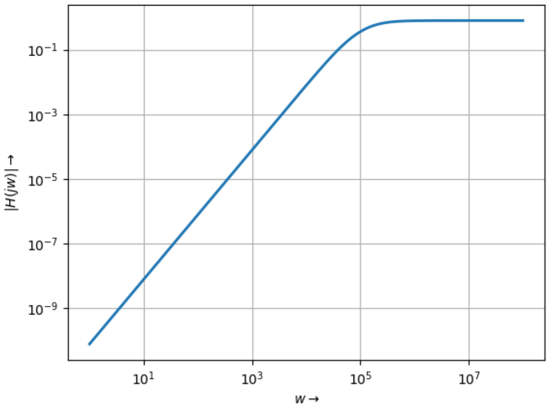

Figure 3: $|H(jw)|$ vs w in loglog scale

3.1 Response to damped sinusoid

Consider the following damped sinusoids, $v_in = e^{-0.5t}sin(2\pi t)$ and $v_in = e^{-0.5t}sin(2\pi 10^5 t)$.

The high pass filter is expected to attenuate the low frequency sinusoid. The system should allow frequencies such as $2 * 10^5$ Hz as they are above the cut-off frequency.

```
t = np.linspace(0,10,1000)
Vi = np.multiply(np.multiply(np.exp(-0.5*t),np.sin(2*np.pi*t)),np.heaviside(t,0.5))
Vo = sp.lsim(H,Vi,T=t)
p.plot(Vo[0],Vi,label=r'$V_{in}$')
plot_graph(Vo[0],Vo[1],r't$\rightarrow$',r'$V\rightarrow$')
```

```
     t = np.linspace(0,0.0001,10000)
    Vi = np.multiply(np.multiply(np.exp(-0.5*t),np.sin(2*np.pi*200000*t)),np.heavisi
    Vo = sp.lsim(H,Vi,T=t)
    p.plot(Vo[0],Vi,label=r'$V_{in}$')
    plot_graph(Vo[0],Vo[1],r't$\rightarrow$',r'$V\rightarrow$')
```

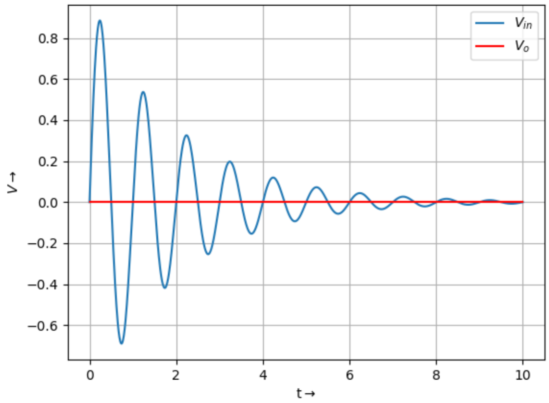

Figure 4: High-pass filter response for 1Hz sinusoid input

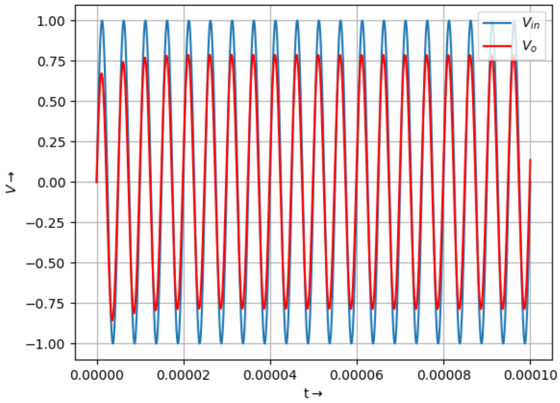

Figure 5: High-pass filter response for $2 * 10^5$ Hz sinusoid input

3.2 Step Response

```
  t = np.linspace(0,0.001,1000)
Vo = sp.step(H,T=t)
plot_graph(Vo[0],Vo[1],r't$\rightarrow$',r'$V_{o}\rightarrow$')
```

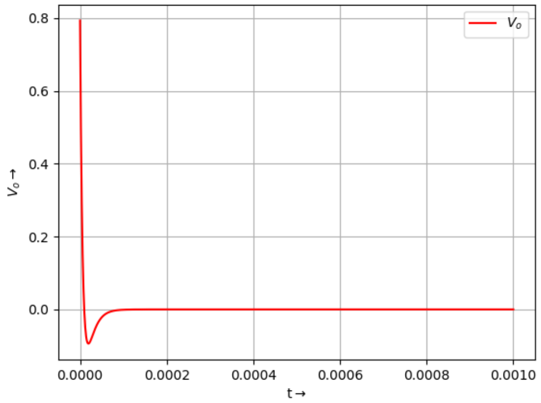

Figure 6: Step response of high pass filter

4 Conclusion

For a mixed frequency sinusoid as input, it was found that the filter suppressed the high frequencies while allowing the low frequency components. Similarly, a high pass filter was implemented using an op-amp with the same gain. The magnitude response of the filter was plotted and its output was analysed for damped sinusoids.