EE2703 : Applied Programming Lab
Assignment 7
Report

Kesava Aruna Prakash R L
EE20B061

April 6, 2022

Abstract

This assignment is aimed at exploring circuit analysis using Laplace Transforms and numerical methods of analysing s-domain expressions. We will use the `sympy` module for its powerful symbolic capabilities in the process.

Question 1

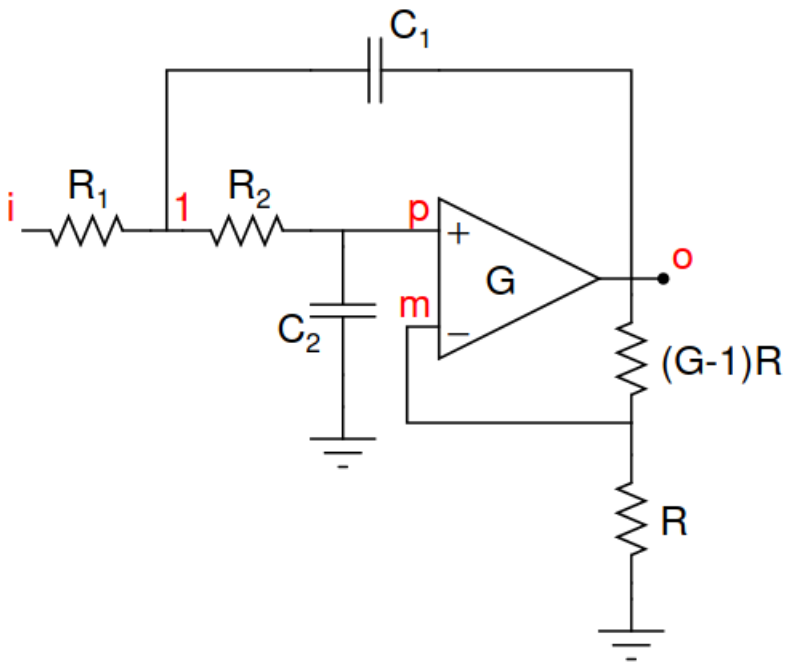

Figure 1: Butterworth Low Pass Filter

For the given circuit, the circuit equations are:

$$V_m = \frac{V_o}{G}$$

$$V_p = \frac{V_1}{1 + sR_2C_2}$$

$$V_o = G(V_p - V_m)$$

$$\frac{V_i - V_1}{R_1} + \frac{V_p - V_1}{R_2} + sC_1(V_o - V_i) = 0$$

Rearranging and writing in matrix form, the equations become :

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

If these matrices above are expressed as $A \cdot x = B$, then $x = A^{-1}B$ can be easily evaluated and V_o is the last element of x. Hence, we can easily find the response of the system to any input $V_i(s)$.The transfer function obtained(output at $V_i(s) = 1$) can be converted into the form which we can use in `scipy.signal` toolbox.
Python code for the above:

```
import sympy as spy
import scipy.signal as sg
def ltisys(v):
    num,denom = spy.fraction(v)
    num_coeff = spy.poly(num).all_coeffs()
    denom_coeff = spy.poly(denom).all_coeffs()
    num_coeff = [float(x) for x in num_coeff]
    denom_coeff = [float(x) for x in denom_coeff]

    return sg.lti(num_coeff,denom_coeff)
```

The converted transfer function is plotted:

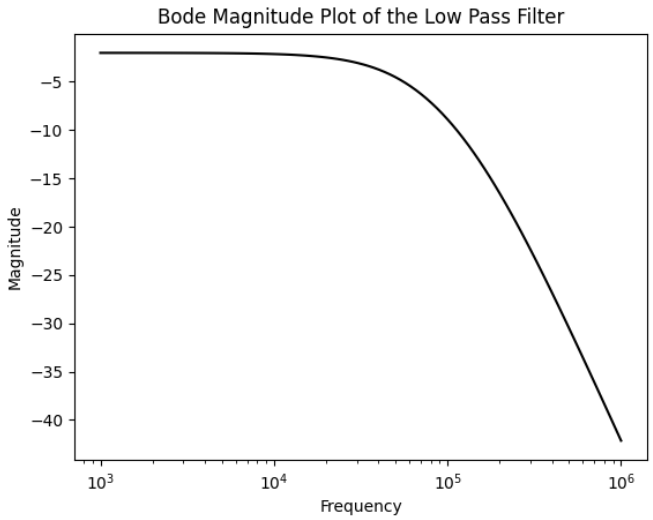

Figure 2: Bode Magnitude Plot for Transfer function of Low Pass Filter

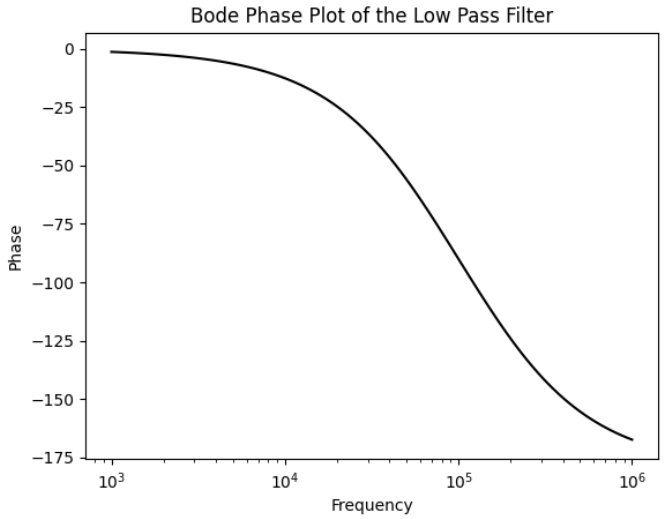

Figure 3: Bode Phase Plot for Transfer function of Low Pass Filter

Step Response for the system can be obtained by convolving the transfer function of the Low Pass Filter with $u(t)$. Convolution can be performed with `signal.lsim` module.

$$s(t) = h(t) * u(t)$$

Python code for the above:

```
H_LP = ltisys(Vo)

# Bode plots for the Transfer function of the circuit

w,H_mag,H_phase = H_LP.bode()

pt.plot(w,H_mag,'k')
pt.title("Bode Magnitude Plot of the Low Pass Filter")
pt.xlabel("Frequency")
pt.semilogx()
pt.ylabel("Magnitude")
pt.show()

pt.plot(w,H_phase,'k')
pt.title("Bode Phase Plot of the Low Pass Filter")
pt.xlabel("Frequency")
```

```
pt.semilogx()
pt.ylabel("Phase")
pt.show()

# Step response for the circuit

t = np.linspace(0,0.005,10000)
u_t = np.ones([10000])

t,y_t,svec = sg.lsim(H_LP,u_t,t)

pt.title("Step Response of the Low Pass filter")
pt.ylabel("Output Voltage Vo")
pt.xlabel("Time")
pt.plot(t,y_t,'r')
pt.show()
```

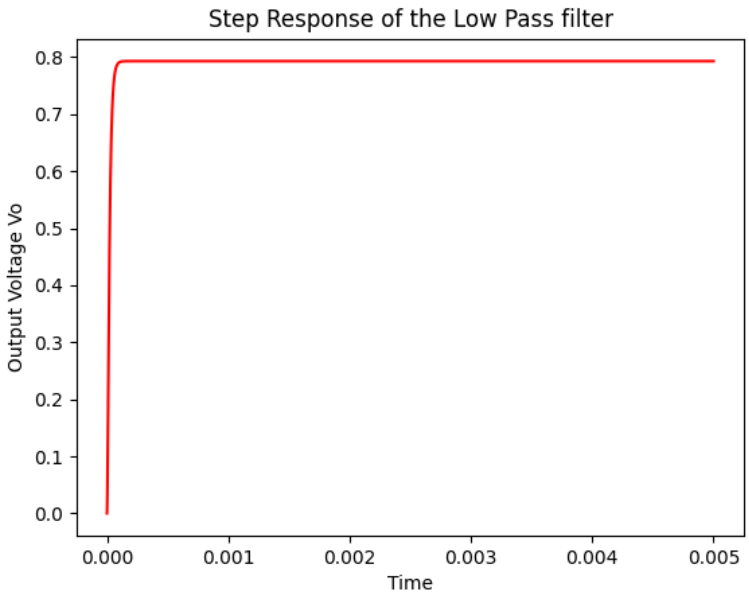

Figure 4: Step Response of the Low Pass Filter

Question 2

For the given input $v_i(t) = (sin(2000\pi t)) + cos(2 \times 10^6 \pi t))u_0(t)$, the output can be easily obtained using the `lsim()` from the `scipy.signal` module. Python code for the above:

```
x_t = np.sin(2000*np.pi*t)+np.cos(2*10**6*np.pi*t)

t,Vo_LP,svec = sg.lsim(H_LP,x_t,t)

pt.title("Output of the Low Pass filter for input Vi(t)")
pt.ylabel("Output Voltage Vo")
pt.xlabel("Time")
pt.plot(t,Vo_LP,'v')
pt.show()
```

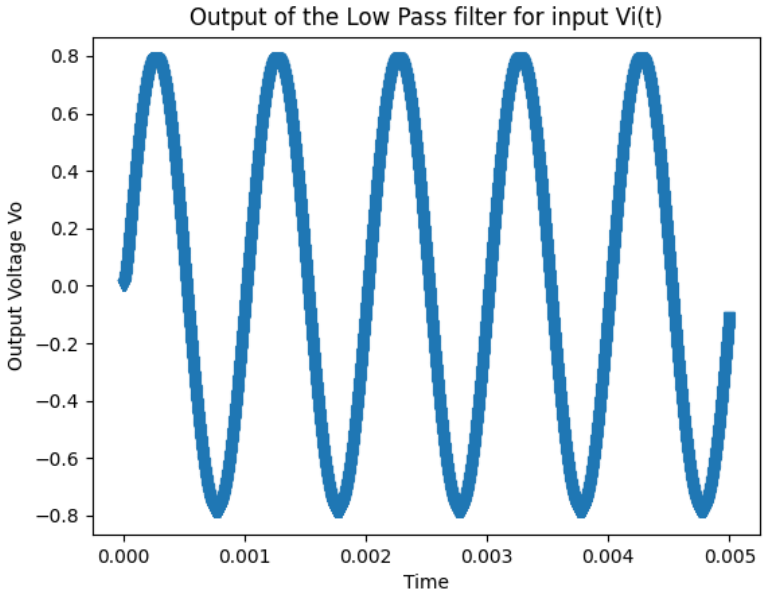

Figure 5: Output of the Low Pass Filter

It can be seen that the response due to the high-frequency cosine term is absent, and the entire output waveform is the response to the low-frequency sine term. This is due to the low-pass filter cutting off the high-frequency components.

Question 3

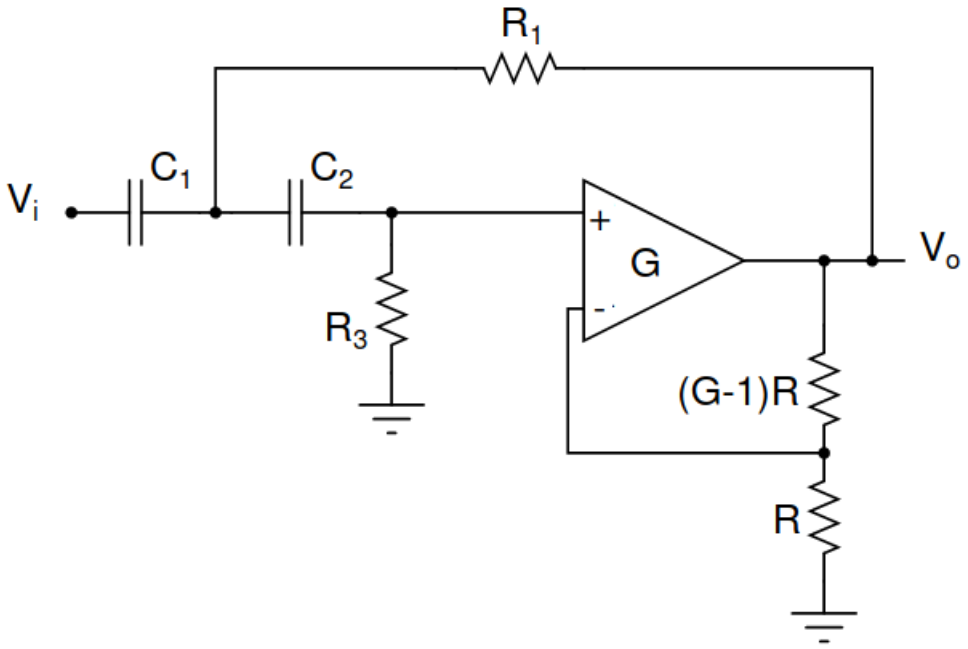

Figure 6: Butterworth High Pass Filter

For the given circuit, the circuit equations are:

$$V_m = \frac{V_o}{G}$$
$$V_p = V_1 \frac{sC_2R_3}{1 + sC_2R_3}$$
$$V_o = G(V_p - V_m)$$
$$sC_1(V_1 - V_i) + sC_2(V_1 - V_p) + \frac{V_1 - V_o}{R_1} = 0$$

In matrix form, the equations become :

$$\begin{pmatrix} 0 & 0 & 1 & -\frac{1}{G} \\ -\frac{sC_2R_3}{1+sC_2R_3} & 1 & 0 & 0 \\ 0 & -G & G & 1 \\ sC_1 + sC_2 + \frac{1}{R_1} & -sC_2 & 0 & -\frac{1}{R_1} \end{pmatrix} \begin{pmatrix} V_1 \\ V_p \\ V_m \\ V_o \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ sC_1V_i \end{pmatrix}$$

Following the same procedure as in **Question 1**, the circuit can be analysed and its transfer function can be obtained.

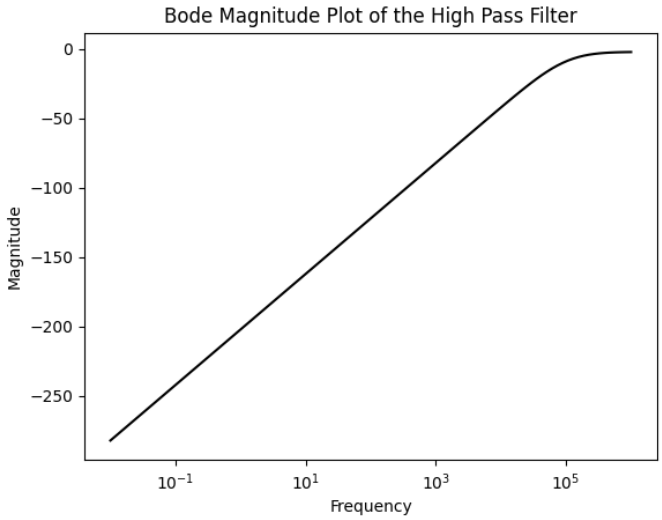

Figure 7: Bode Magnitude Plot for Transfer function of High Pass Filter

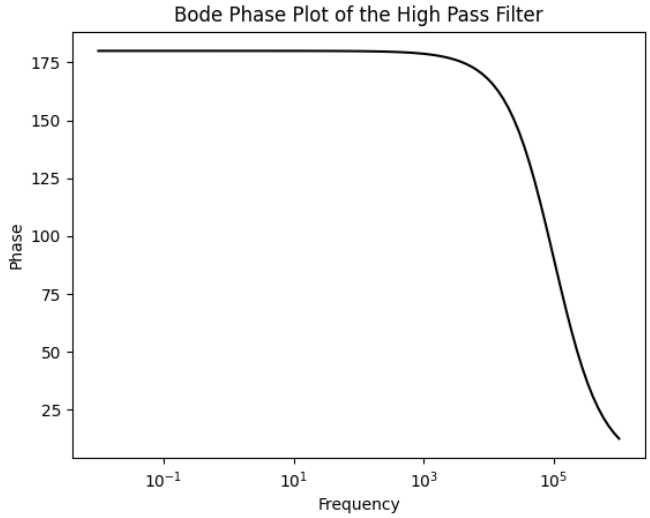

Figure 8: Bode Phase Plot for Transfer function of High Pass Filter

Question 4

The high-pass filter gives different responses to damped sinusoids of low and high frequency. Here, we take the input as $V_i(t) = e^{-100t}sin(2\pi ft)$, where f is either 1 kHz for low frequency or 1 MHz for high frequency.
Python code for the above:

```
# For a decay of 100 second^-1 and frequency of 1e6 * 2pi(High)

t1 = np.linspace(0,2e-5,10000)
vi_t_1 = 1*np.exp(-100*t1)*np.sin(2e6*np.pi*t1)

t1,Vo_HP_1,svec  = sg.lsim(H_HP,vi_t_1,t1)

pt.title("Output of the High Pass Filter for input(Damped Sinusoid)-High Frequen
pt.ylabel("Output Voltage Vo")
pt.xlabel("Time")
pt.plot(t1,Vo_HP_1,'r')
#pt.show()

# For a decay of 100 second^-1 and frequency of 1e3 * 2pi(Low)

vi_t1_2 = 1*np.exp(-100*t1)*np.sin(2e3*np.pi*t1)

t1,Vo_HP_2,svec  = sg.lsim(H_HP,vi_t1_2,t1)

pt.title("Output of the High Pass Filter for input(Damped Sinusoid)-Low Frequenc
pt.ylabel("Output Voltage Vo")
pt.xlabel("Time")
pt.plot(t1,Vo_HP_2,'k')
pt.show()
```

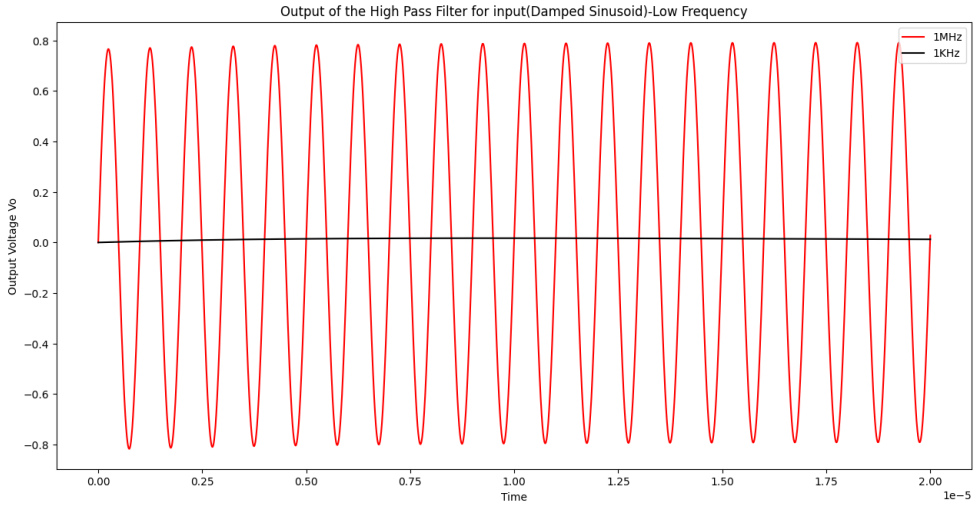

Figure 9: Output for damped sinusoids of different frequencies as input.

As we can see from the figure, high frequency input is reflected in the output without any attenuation but low frequency input was almost reduced to zero amplitude.

Question 5

The Step Response of the High Pass Filter can be found out by replacing $V_i(s) = 1/s$ and finding the impulse response of V_o. Python code for the above:

```
A,b,V = highpass(10000,10000,1e-9,1e-9,1.586,1/s)
Vo_HP = V[3]
H_HP = ltisys(Vo_HP)
t,Vo_HP_Step = sg.impulse(H_HP,None,t)

pt.title("Step Response of the High Pass filter")
pt.ylabel("Output Voltage Vo")
pt.xlabel("Time")
pt.plot(t,Vo_HP_Step,'g')
pt.show()
```

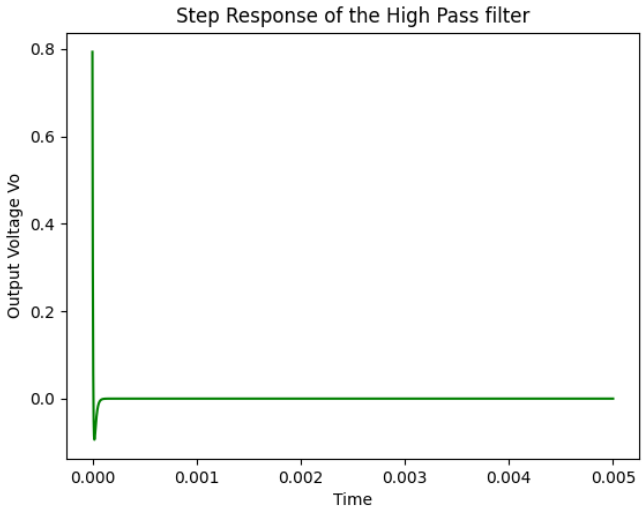

Figure 10: Step Response of the High Pass Filter

Clearly, the step response goes to 0 very quickly. This is because the high-pass filter blocks low-frequency components, and the step response essentially has frequency 0.