EE2703 : Applied Programming Lab Week7 : Analysis of circuits using Laplace Transform and symbolic algebra

Aditya Suresh
EE20B004

Friday 8th April, 2022

AIM

- To analyse give high pass and low pass filters via Laplace transform in python
- To use symbolic algebra via sympy module in python for the same.
- Plot Bode plot the High pass and low pass filter's impulse response.
- To plot the output in time domain for given input signals.

Theory

Laplace Transform

Laplace transform converts time domain signals into frequency domain.The best way to convert differential equations into algebraic equations is the use of Laplace transformation.

Because of this possibility of producing frequency domain signals which have algebraic relationship, helps solve systems which have rather complex tiime domain equations.

Circuit

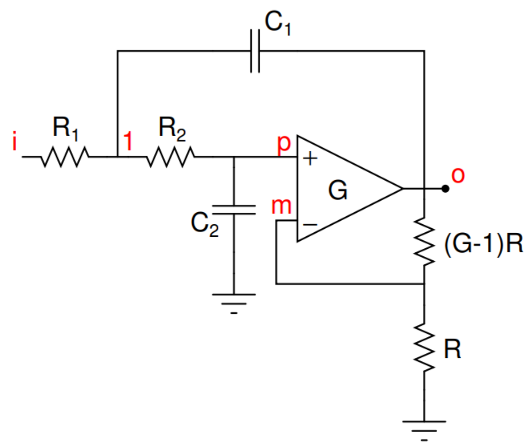

Figure 1: Circuit of low pass filter

Formula

Writing Nodal Equation of the above circuit

$$V_m = \frac{V_o}{G} \tag{1}$$

$$V_p = V_1 \frac{1}{1 + j\omega R_2 C_2} \tag{2}$$

$$V_o = G(V_p - V_m) \tag{3}$$

$$\frac{V_i - V_1}{R_1} + \frac{V_p - V_1}{R_2} j\omega C_1 (V_0 - V_1) = 0 \tag{4}$$

Solving for V_0 in 3, we get

$$V_0 = \frac{GV_1}{2} \frac{1}{1 + j\omega R_2 C_2}$$

Assignment Questions

Bode plot and phase plot of H(s) for LPF

The transfer function for the ow pass filter comes out as

$$V_{out} = \frac{0.0001(1.586 \cdot 10^{-15} \cdot s^3 + 4.758 \cdot 10^{-10} \cdot s^2 4.758 \cdot 10^{-5} \cdot s + 1.586)}{(2 \cdot 10^{-29} \cdot s^5 + 1.0414 \cdot 10^{-23} \cdot s^4 + 2.12 \cdot 10^{-18} \cdot s^3) + 2.12 \cdot 10^{-13} \cdot s^2 + 1.0414 \cdot 10^{-8} \cdot s +}$$

Bode Plot of LPF:- The above bode plot shows the circuit is **Low Pass**

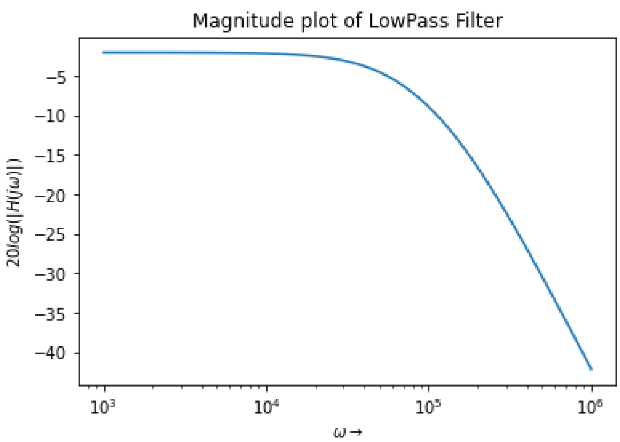

Figure 2: Magnitude Plot of LPF

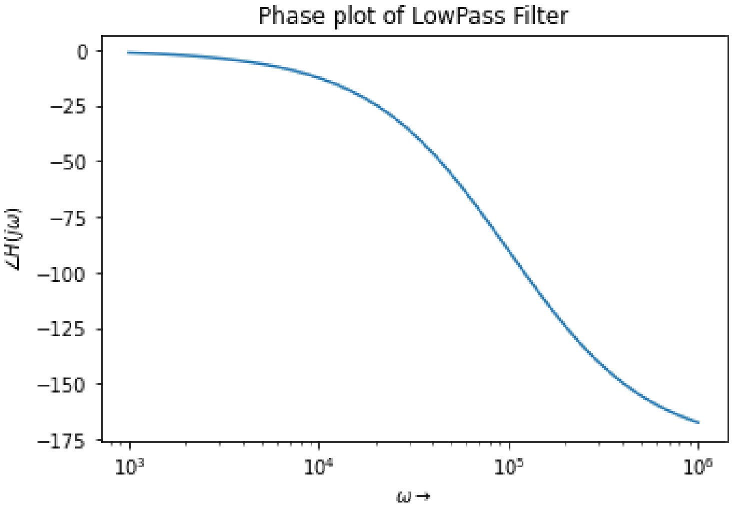

Figure 3: Phase plot of LPF

Filter.

Question 1 : Step response

1. The following code is used to generate and plot step response of Low pass filter

---

```
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
    b=Matrix([0,0,0,-Vi/R1])
    V = A.inv() *b
    return V[3]

Vo=Lowpass(10000,10000,1e-9,1e-9,1.586,1)

#Defining H for low pass filter
num, denom = get_rational_coeffs(Vo)
H = sp.lti(num, denom)

t2,y,svec = sp.lsim(H,u,t2)
plt.title("Step response of given Low pass filter")
plt.xlabel(r"$t \rightarrow$")
plt.ylabel(r"$V_{out}$")
```

```
plt.plot(t2,y)
plt.show()
```

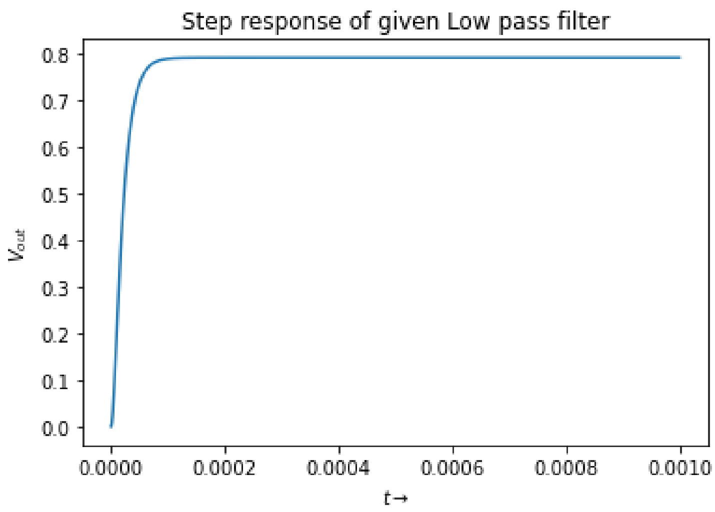

Figure 4: System step response

Question 2: Determine V_{out} for given V_{in}

The given input is : $V_{in} = (sin(2000\pi t) + cos(2x10^6\pi t))u_0(t)$

1. The following code is used to generate graph of V_{in}

```
plt.title("$V_{in}$ = [$sin(2000\pi t)$ + $cos(2x10^6\pi t)$]u(t)(Low pass 
plt.xlabel(r"$t \rightarrow$")
plt.ylabel(r"$V_{out}$")
plt.plot(t1,vi(t1))
plt.show()
```

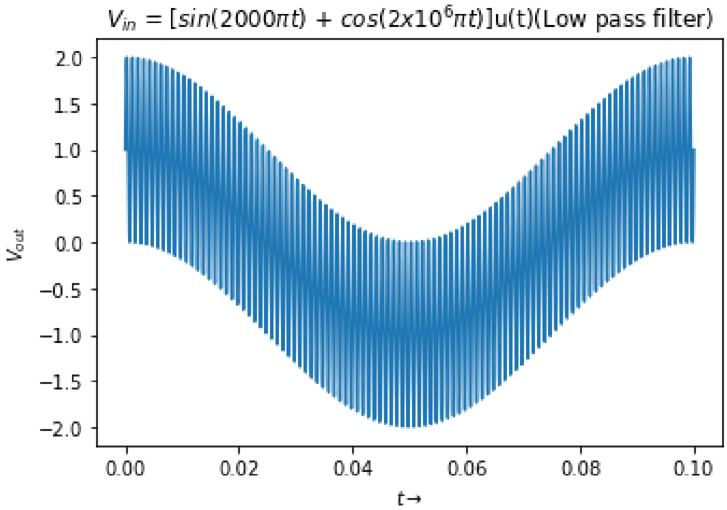

Figure 5: Graph of V_{in}

2. The following code is used to plot the output (zoomed in version is given as well)

```
fig, (ax1, ax2) = plt.subplots(2,1)
t1, vo, svec = sp.lsim(H, vi(t1), t1)
ax1.set_title("Vout for Low pass filter$")
ax1.set_xlabel(r"$t \rightarrow$")
ax1.set_ylabel(r"$V_{out}$")
ax1.plot(t1,vo)
#zoomed in version ie t value in 1e-5
t2, vo, svec = sp.lsim(H, vi(t2), t2)
ax2.set_title("Vout for Low pass filter (zoomed in)$")
ax2.set_xlabel(r"$t \rightarrow$")
ax2.set_ylabel(r"$V_{out}$")
ax2.plot(t2,vo)
plt.show()
```

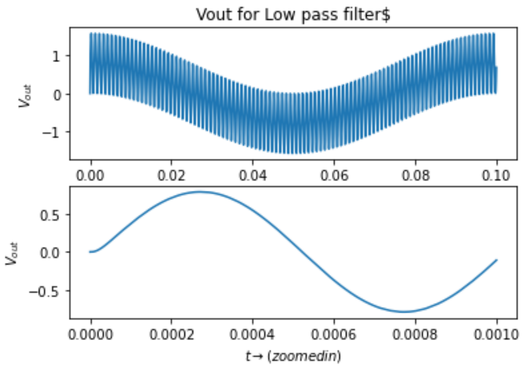

Figure 6: System response with decay = 0.05

3. From the above we see V_{out} to be following v_{in}, with an smaller amplitude

Question 3: High Pass Filter

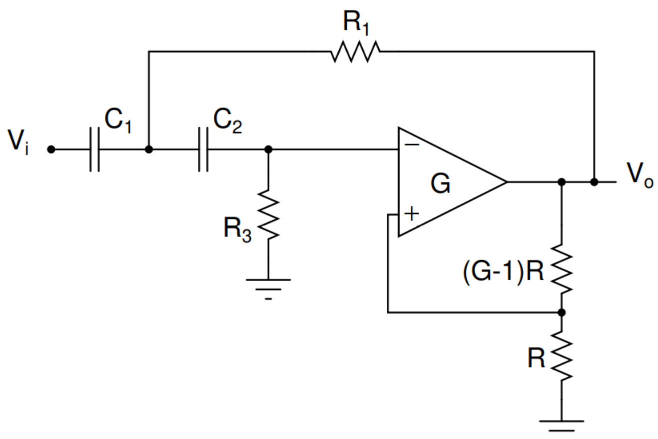

Figure 7: Circuit of High pass filter

1. The Bode plot of the high pass filter given above

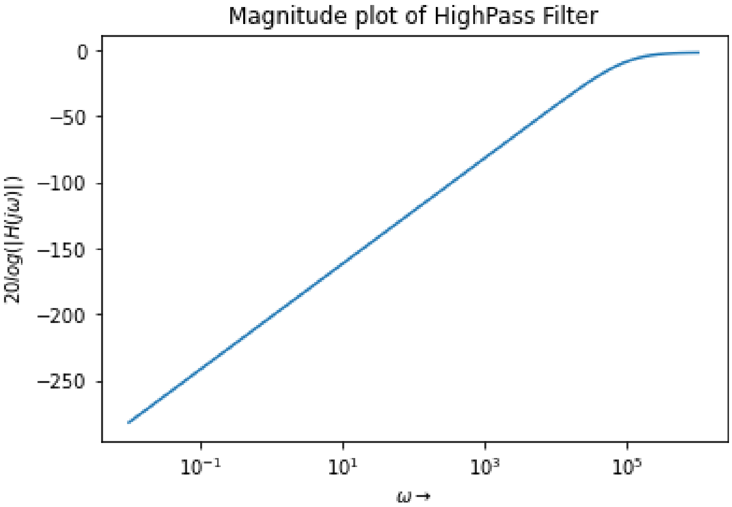

Figure 8: Magnitude response of HPF

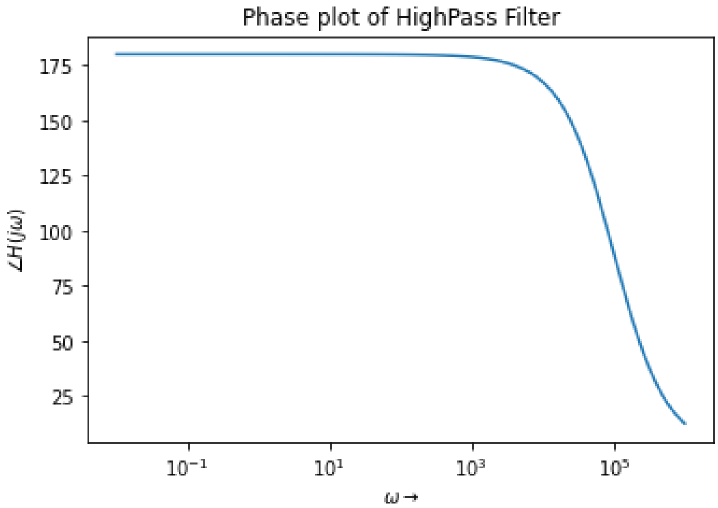

Figure 9: Phase response of HPF

2. Corresponding V_{out} when V_{in} is passed through he given HPF.

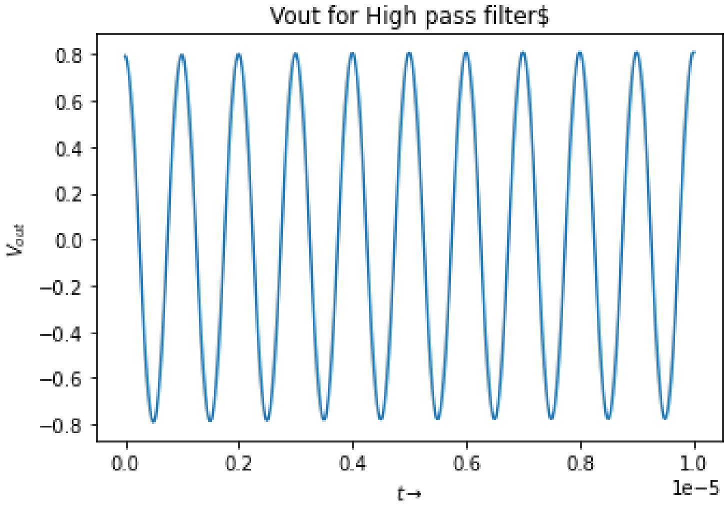

Figure 10: V_{out} for given V_{in}

3. The above plot of V_{out} too following V_{in} with amplitude being enlarged

Question 4: Passing Damped sinusoid through LPF and HPF

1. Plot of High frequency V_{in}

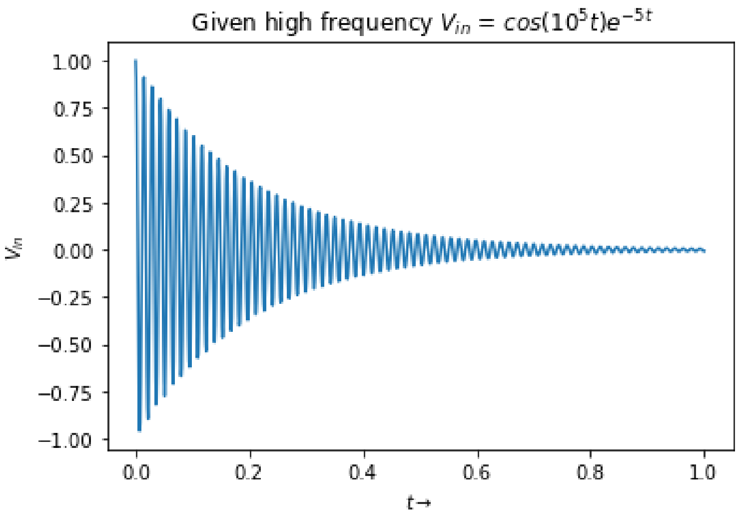

Figure 11: V_{in}

2. Corresponding V_{out} for LPF

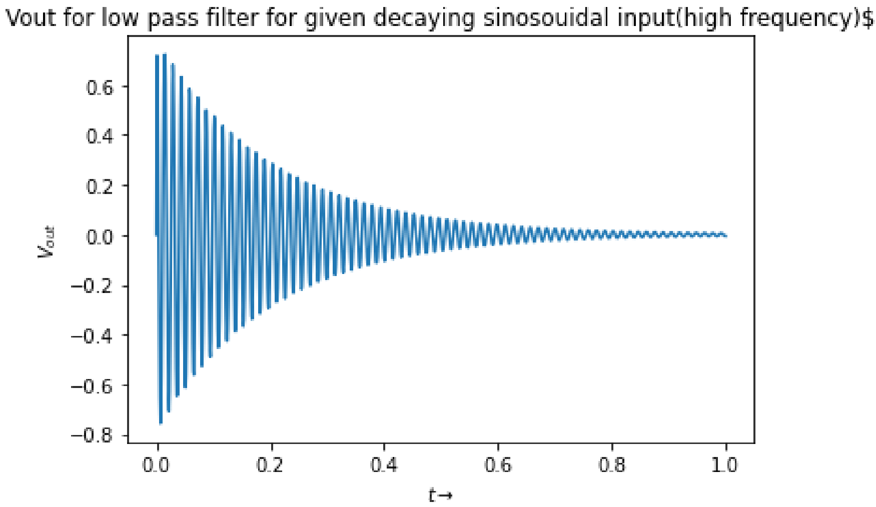

Figure 12: V_{out} for LPF for high frequency

3. Corresponding V_{out} for HPF

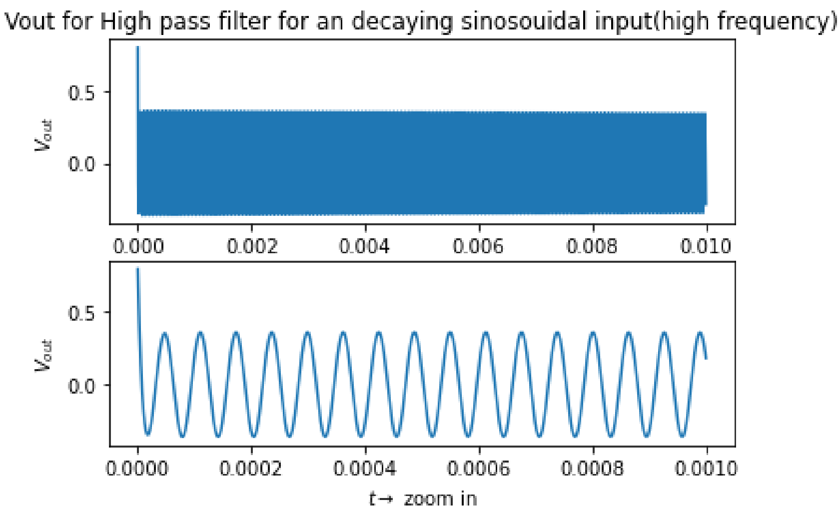

Figure 13: V_{out} for LPF for high frequency

4. We get a response varying sinusoidally around zero with amplitude being dampened

5. V_{in} having low frequency

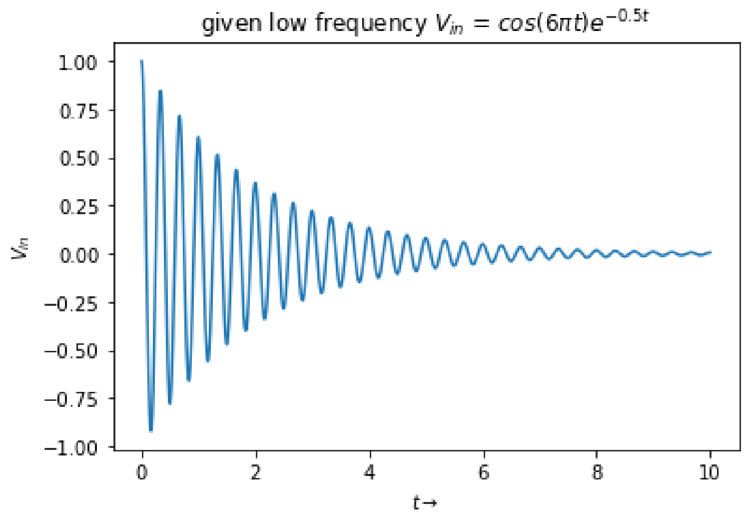

Figure 14: V_{in} for low frequency

6. V_{out} for LPF

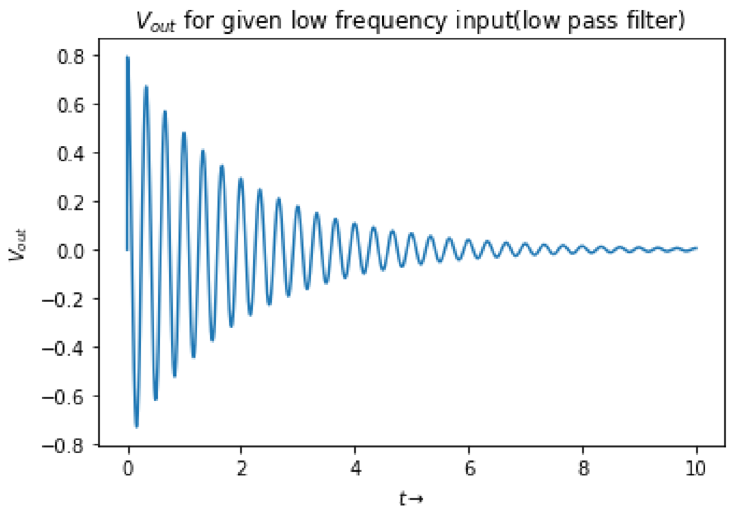

Figure 15: V_{out} for LPF for Low frequency

7. V_{out} for HPF

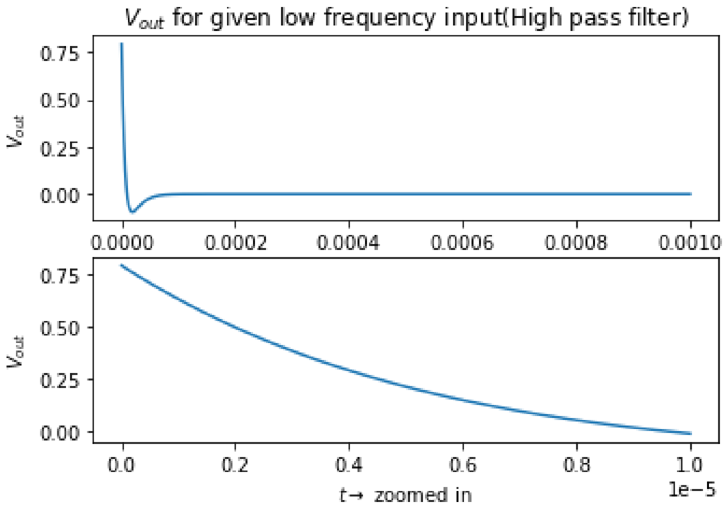

Figure 16: V_{out} for HPF for Low frequency

Question 5 : Step Response of HPF

1. The following code is used to generate and plot step response of the above given HPF circuit

---

```
t1 = np.linspace(0, 0.0001,1000)
Vo = highpass(10000,10000,1e-9,1e-9,1.586,1/s)

num , denom = get_rational_coeffs(Vo)
Vo = sp.lti(num, denom)
t1,vo = sp.impulse(Vo, None, t1)
plt.title("Step response of the given High Pass filter")
plt.xlabel(r"$t \rightarrow$")
plt.ylabel(r"$V_{out}$")
plt.plot(t1, vo)
plt.show()
```

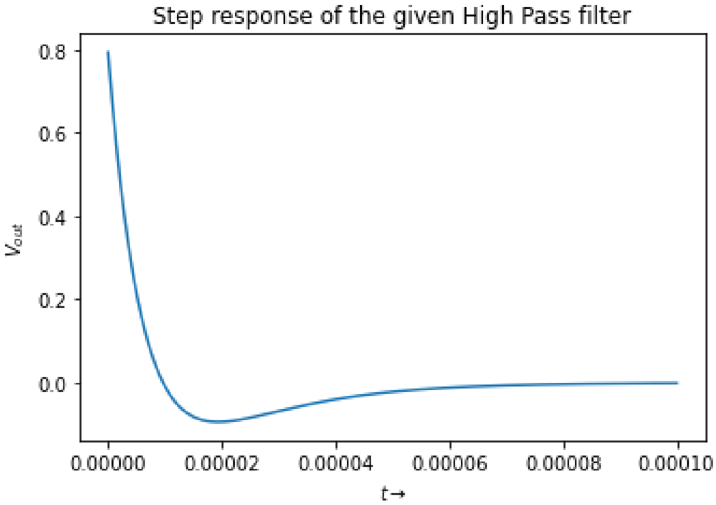

Figure 17: Step Response of High pass filter

2. The response tends to zero as system tends to stability. This can be seen by open circuiting capacitors(comes from the fact that $u_0(t)$ is sinusoidal). Apply this, virtual short and resistor divide formula we get V_{out} as 0.

Conclusions

We verified the behaviour of High pass filter and low pass filter by passing various high and low frequency input. We took the help of powerful python modules such as symy and matplotlib to visualise the same.