EE2703 : Applied Programming Lab
Assignment 7

Dharahas H
EE20B041

April 9, 2022

Introduction

Using SymPy to analyse Lowpass and Highpass filters.

Q1

The low pass filter circuit is as given below :

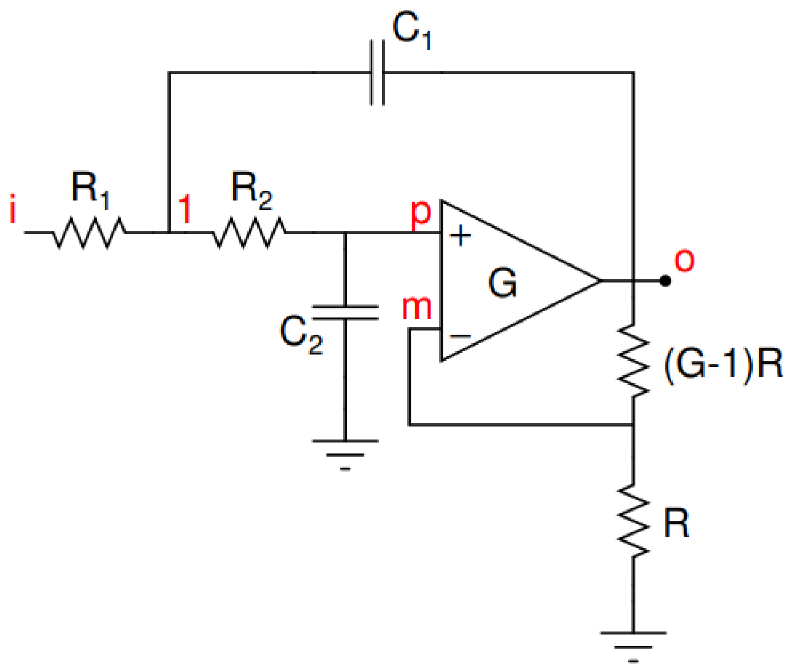

Figure 1: Lowpass Filter Circuit

The matrix equation for solving the circuit is :

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

Code

Defining a Lowpass filter :

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

    return (A,b,V)
```

Transfer function of LPF :

```
A,b,V = lowpass(10000,10000,1e-9,1e-9,1.586,1)
Vo = V[3]
ww=np.logspace(0,8,801)
ss=1j*ww
hf=sp.lambdify(s,Vo,'numpy')
v=hf(ss)

plt.figure()
plt.title("Transfer function of LPF")
plt.xlabel(r'$\omega\rightarrow$')
plt.ylabel(r'$|H(j\omega)|\rightarrow$')
plt.loglog(ww,abs(v),lw=2)
plt.grid(True)
plt.show()
```

Result :

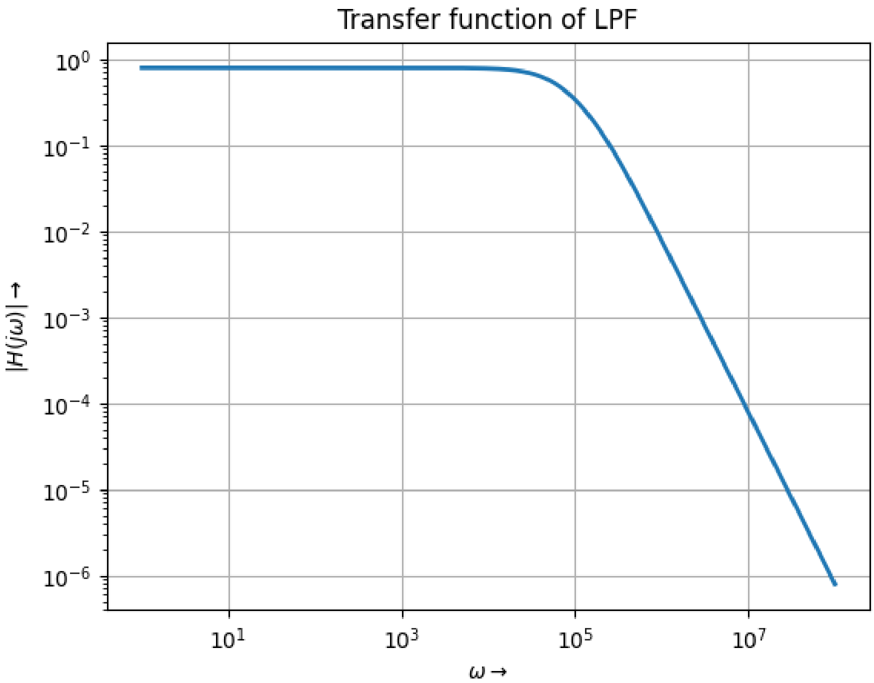

Figure 2: Frequecy response of LPF

For Step response:

```
H = sg.lti([-1.586e-4], [2e-14, 4.414e-9, 2e-4])
t = np.linspace(0, 0.001, 200000)
u = (t>0)
t,y,_ = sg.lsim(H,u,t)

plt.figure()
plt.title("Step response of LPF")
plt.xlabel(r'$t\rightarrow$')
plt.ylabel(r'$y\rightarrow$')
plt.plot(t,u, label="input")
plt.plot(t,y, label="output")
plt.legend()
plt.grid()
plt.show()
```

Result :

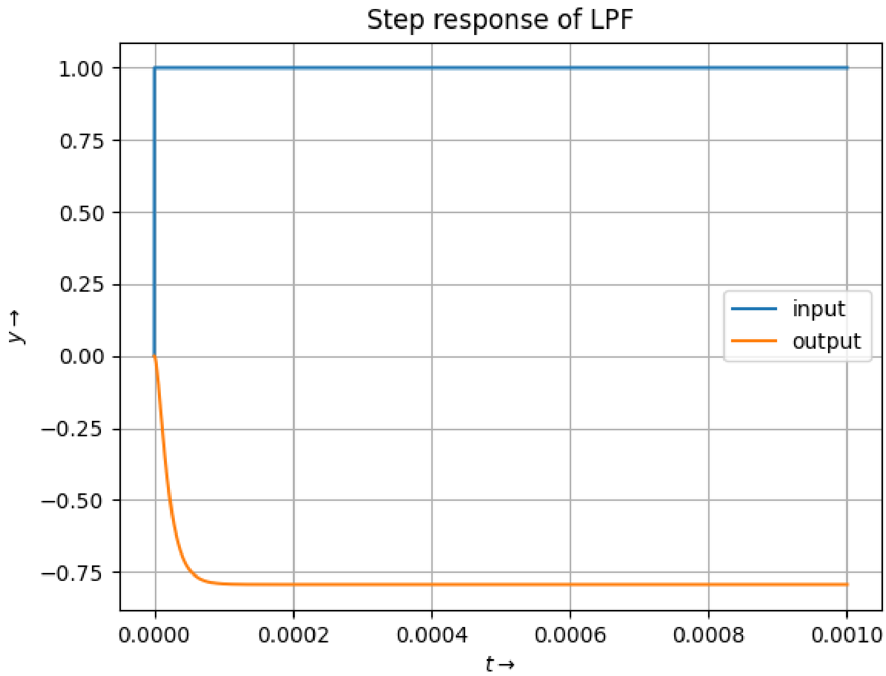

Figure 3: Step response of LPF

Q2

The input signal to LPF is given as :

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

The signal can be split into two parts
Low frequecy signal :

$$V_{il}(t) = sin(2000\pi t)u_o(t)$$

High frequency signal :

$$V_{ih}(t) = cos(2 * 10^6 \pi t)u_o(t)$$

Code

Plotting the given input :

```
t = np.linspace(0, 0.0005, 30000)
Vi_l = np.sin(2000*np.pi*t)*(t>0)
Vi_h = np.cos(2e5*np.pi*t)*(t>0)

plt.figure()
plt.title("The input signal")
plt.xlabel(r'$t\rightarrow$')
plt.ylabel(r'$y\rightarrow$')
plt.plot(t, Vi_l, label=r'$Low\;\omega$')
plt.plot(t, Vi_h, label=r'$High\;\omega$')
plt.legend()
plt.grid()
plt.show()
```

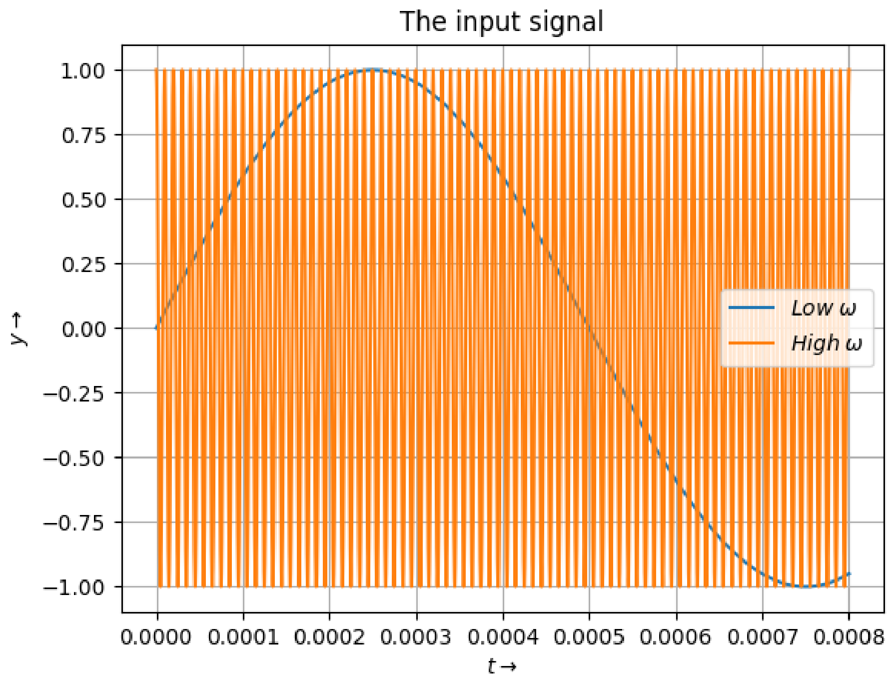

Figure 4: Vi at Low and High frequencies

For getting Vo :

```
H = sg.lti([-1.586e-4], [2e-14, 4.414e-9, 2e-4])
t,y,_ = sg.lsim(H,Vi_l + Vi_h, t)
```

```
plt.figure()
plt.title("Time domain response to the input signal")
plt.xlabel(r'$t\rightarrow$')
plt.ylabel(r'$Vo\rightarrow$')
plt.grid()
plt.plot(t,y)
plt.show()
```

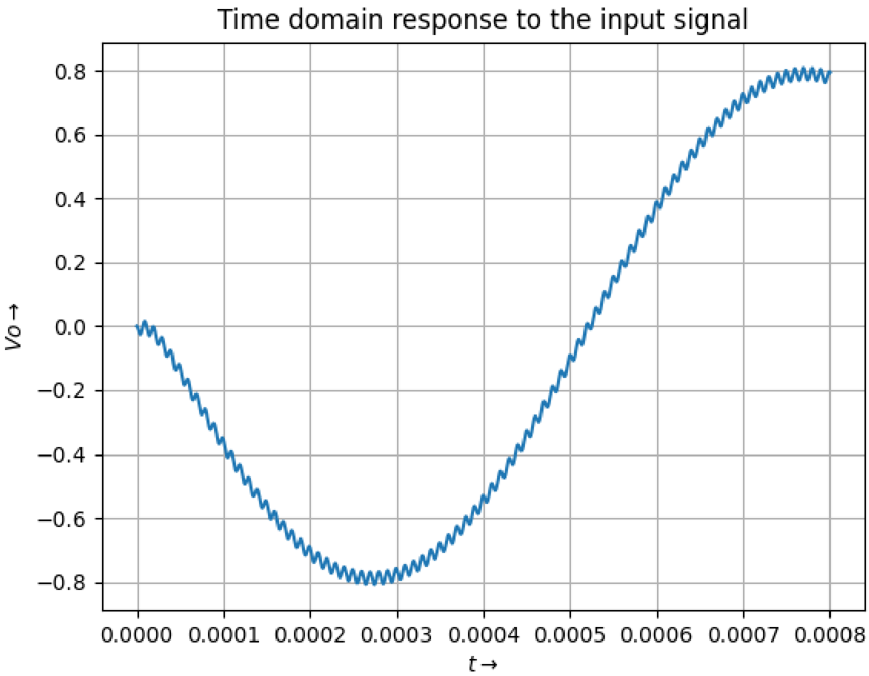

Figure 5: Vo for the given Vi

Q3

The High pass filter circuit is as follows :

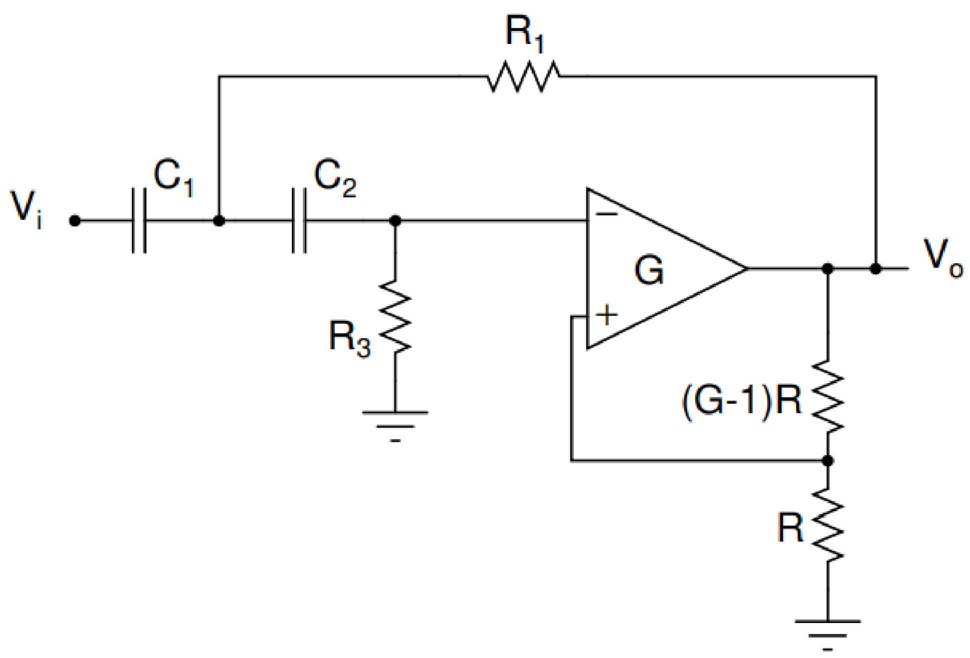

Figure 6: Highpass Filter circuit

Code

Defining the Highpass filter:

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

    return (A,b,V)
```

Plotting frequency response:

```
A,b,V = highpass(1e4,1e4,1e-9,1e-9,1.586,1)
Vo = V[3]
ww=np.logspace(0,8,801)
ss=1j*ww
```

```
hf=sp.lambdify(s,Vo,'numpy')
v=hf(ss)

plt.figure()
plt.title("Frequency response of HPF")
plt.xlabel(r'$\omega\rightarrow$')
plt.ylabel(r'$|H(j\omega)|\rightarrow$')
plt.semilogx(ww,abs(v),lw=2)
plt.grid(True)
plt.show()
```

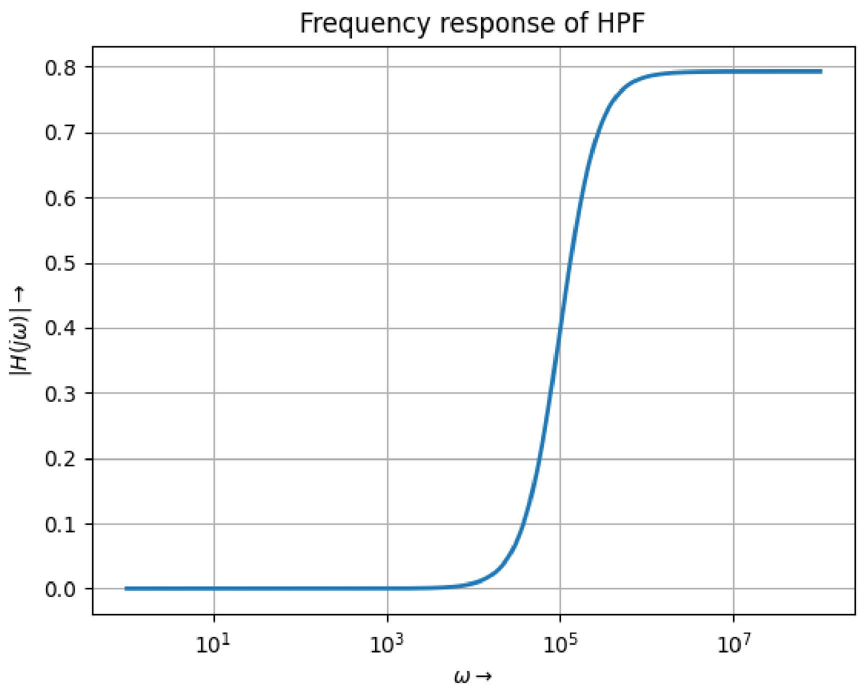

Figure 7: Frequency response

Q4

The damping input is taken as:

$$V_i(t) = (sin(2000\pi t) + cos(200000\pi t)) \exp^{-50000t}$$

Code

Plotting the input signal :

```
t = np.linspace(0,0.0001, 20000)
Vi_l = np.sin(2000*np.pi*t)*np.exp(-50000*t)
Vi_h = np.cos(2e5*np.pi*t)*np.exp(-50000*t)

plt.figure()
plt.title("Input signal")
plt.xlabel(r'$t\rightarrow$')
plt.ylabel(r'$y\rightarrow$')
plt.plot(t, Vi_l, label=r'$Low\;\omega$')
plt.plot(t, Vi_h, label=r'$High\;\omega$')
plt.legend()
plt.grid()
plt.show()
```

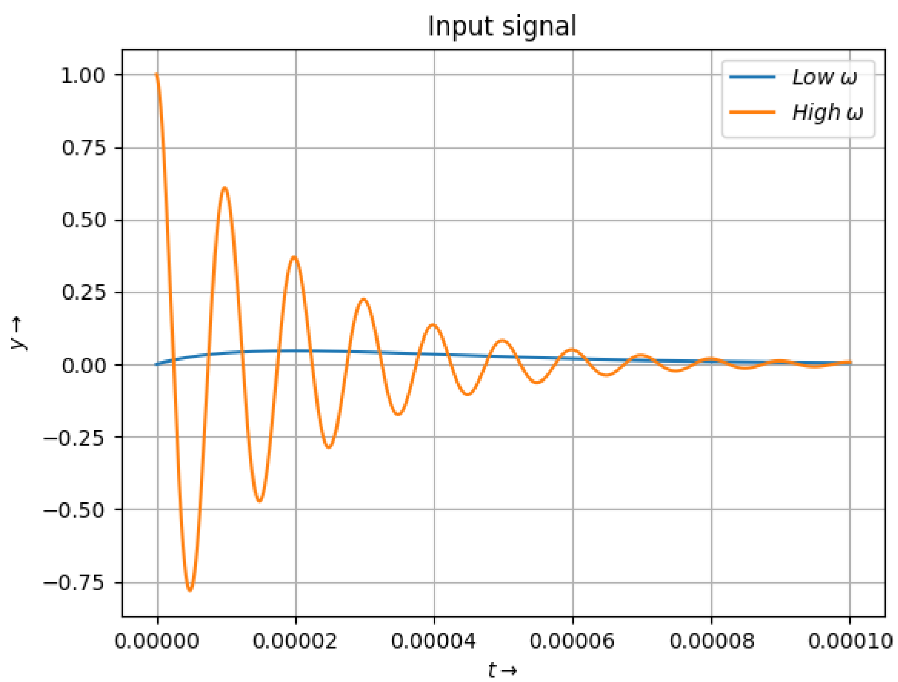

Figure 8: The damping input signal Vi

For response to the given input :

```
H = sg.lti([-1.586e-9,0,0],[2e-9,4.414e-4,20])
t,y,_ = sg.lsim(H, Vi_l + Vi_h, t)

plt.figure()
plt.title("Response to the damping signal")
plt.xlabel(r'$t\rightarrow$')
plt.ylabel(r'$Vo\rightarrow$')
plt.plot(t,y)
plt.grid()
plt.show()
```

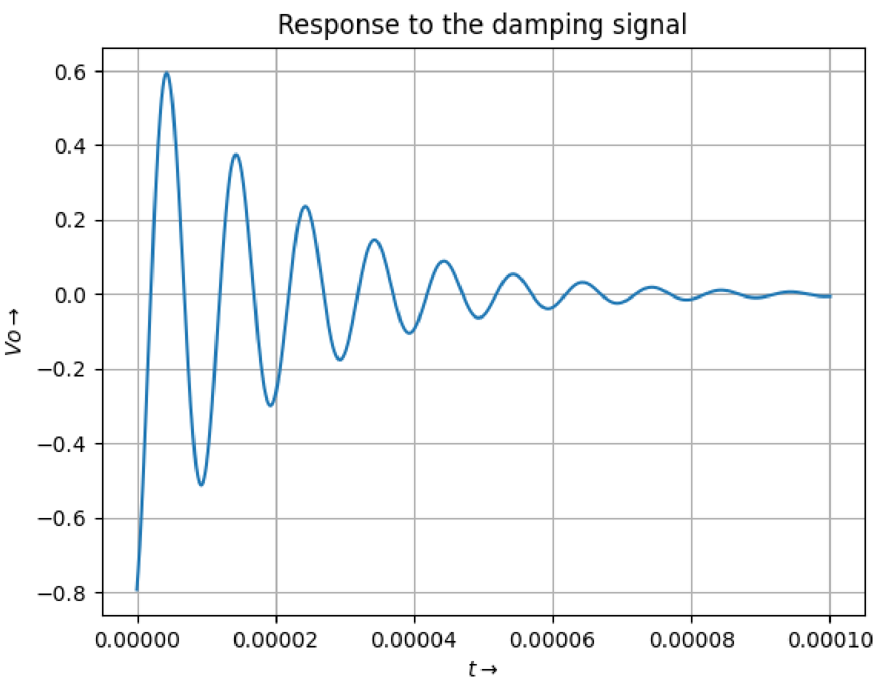

Figure 9: Time domian response corresponding to Vi

Q5

For step response of Highpass filter : Vi = 1

Code

```
u = (t>0)
```

```
t,y,_ = sg.lsim(H, u, t)

plt.figure()
plt.title("Step response of the HPF")
plt.xlabel(r'$t\rightarrow$')
plt.ylabel(r'$y\rightarrow$')
plt.plot(t,u, label="input" )
plt.plot(t,y, label="output")
plt.legend()
plt.grid()
plt.show()
```

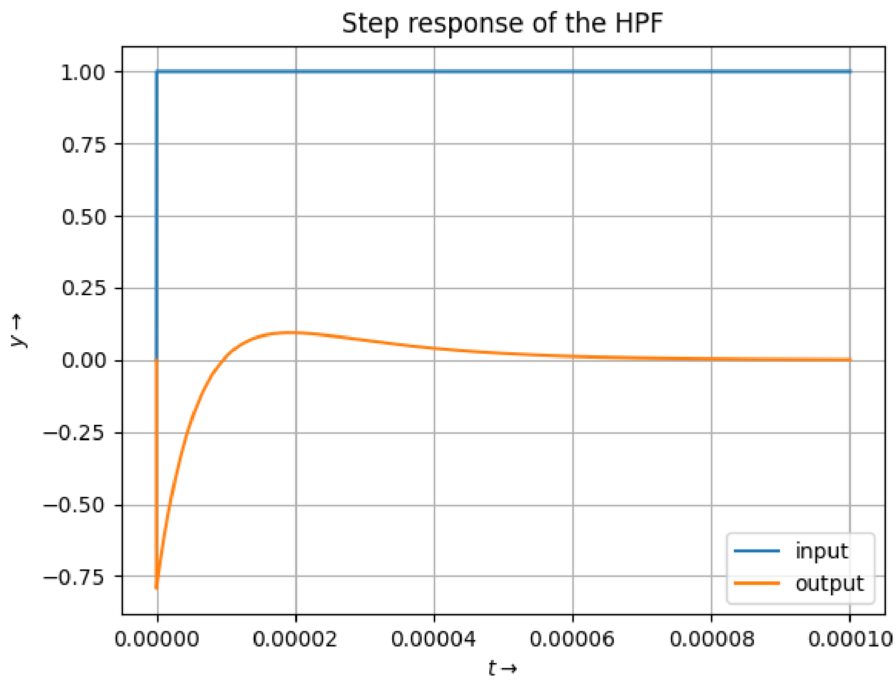

Figure 10: Step Response

The Response curve at a later time goes to zero as at a later instance the input is steady at 1 and the since it is a highpass filter, response is zero. But we can observe a spike initially because the input goes from zero to one abruptly.