Analysis of Circuits using Laplace Transforms

Mohammed Muqeeth
EE16B026
Electrical Engineering
IIT Madras

April 5, 2018

Abstract

We discuss symbolic algebra and analysis of circuits using laplace transforms.We get introduced to sympy module and how to convert expressions in sympy to pylab functions.

1 Introduction

- Sympy helps us do symbolic work.These symbols are defined as below.The symbols are not python variables.

$$w = symbols('omega')$$

- It has Matrix function built into it which helps to define a matrix.M = Matrix([1 2],[3 4]) gives $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$.The inv function helps us in computing inverse of a matrix .It is done as M.inv().
- Sympy has evalf function to evaluate value for the symbol as expr = sqrt(2) ,expr.evalf() returns 1.414.
- For magnitude response of laplace transform we use lambdify function.

2 Low pass filter

2.1 Writing the nodal equations:

$$\begin{pmatrix} 0 & 0 & 1 & \frac{-1}{G} \\ \frac{-1}{1+sR_2C_2} & 1 & 0 & 0 \\ 0 & -1000 & 1000 & 1 \\ \frac{-1}{R_1} - \frac{1}{R_2} - sC_1 & \frac{1}{R_2} & 0 & sC_1 \end{pmatrix} \begin{pmatrix} V_1 \\ V_P \\ V_m \\ V_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\frac{V_i(S)}{R1} \end{pmatrix}$$

- The transfer of the system $\frac{V_o(s)}{V_i(s)}$ comes from solving for x in Ax=b ie x = inv(A).b
- $V_o(s)$ is in x[3].
- For calculating impulse response of system we give $V_i(s) = 1$ and for step response of system we give $V_i(s) = \frac{1}{s}$.
- Sympy helps in defining matrix with symbols. inv function calculates inverse of matrix.The lambdify takes $V_o(s)$ and converts it into a python function.

```
s = symbols('s')
def lowpass(R1,R2,C1,C2,G,Vi):
    A=Matrix([[0,0,1.0,-1.0/G],[-1.0/(1.0+s*R2*C2),1.0,0,0],
    [0,-1000.0,1000.0,1.0],[-1.0/R1-1.0/R2-s*C1,1.0/R2,0,s*C1]])
    b=Matrix([0,0,0,-Vi/R1])
    V=A.inv()*b
    return (A,b,V)
```

2.2 Plot magnitude response

- we define range of w for which magnitude response to be calculated using logspace.
- Since we have transfer function as function of s. For bode plot we put s=jw.The corresponding plot is in figure 1

```
A,b,V=lowpass(10000.0,10000.0,1e-9,1e-9,1.586,1.0)
Vo=V[3]
print simplify(Vo)
w=p.logspace(0,8,801)
ss= 1j*w
hf=lambdify(s,Vo,'numpy')
v=hf(ss)
p.loglog(w,abs(v),lw=2)
p.title('Low pass filter')
p.xlabel('$\omega(rad/s)$')
p.ylabel('$Magnitude$')
p.grid(True)
p.show()
```

2.3 Unit step response of low pass filter.

- For unit step response since we have H(s) , output Y(s) = H(s)X(s). For unit step X(s) = 1/s.Therefore Y(s)= H(s)/s.
- We can find the corresponding coefficients of Y(s) and give as input to sp.lti function which gives y(t).
- Next we plot output y(t) vs t in figure 2

```
#unit step response
H=sp.lti([0.1],[6.31517e-12,8.9455e-7,0.0631517,0])
t,x=sp.impulse(H,None,p.linspace(0,0.0001,1001))
p.title('unit step response of low pass filter')
p.xlabel('t')
p.ylabel('$V_o(t)$')
p.plot(t,x)
p.grid(True)
p.show()
```

2.4 Response for sinusoidal input.

- Given input is $v_i(t) = sin(2000\pi t) + cos(2\times106\pi t)$.The h(t) can be calculted from coefficients of H(s) using sp.lti function.
- sp.lsim gives us output by convolving h(t) with input $v_i(t)$.The plot of response is in figure 3

```
#for sinusoidal input
H=sp.lti([0.1],[6.31517e-12,8.9455e-7,0.0631517])
t=p.linspace(0.0,1e-2,10001)
u=p.sin(2e3*p.pi*t)+p.cos(2e6*p.pi*t)
t,y,svec=sp.lsim(H,u,t)
p.title('response of low pass filter for sinusoidal input')
p.xlabel('t')
p.ylabel('$V_o(t)$')
p.plot(t,y)
p.grid(True)
p.show()
```

2.5 Plots and discussion

Figure 1: Magnitude response of second order low pass filter

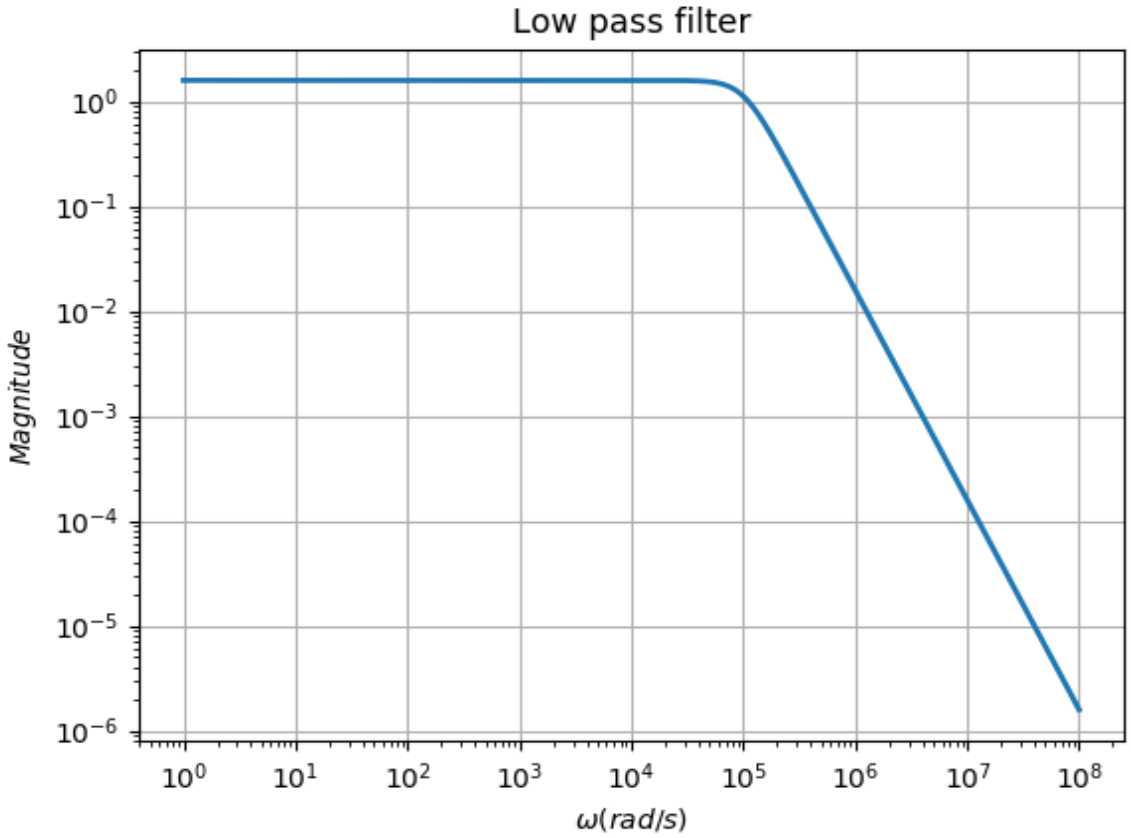


Figure 2: Unit step response of Low pass filter

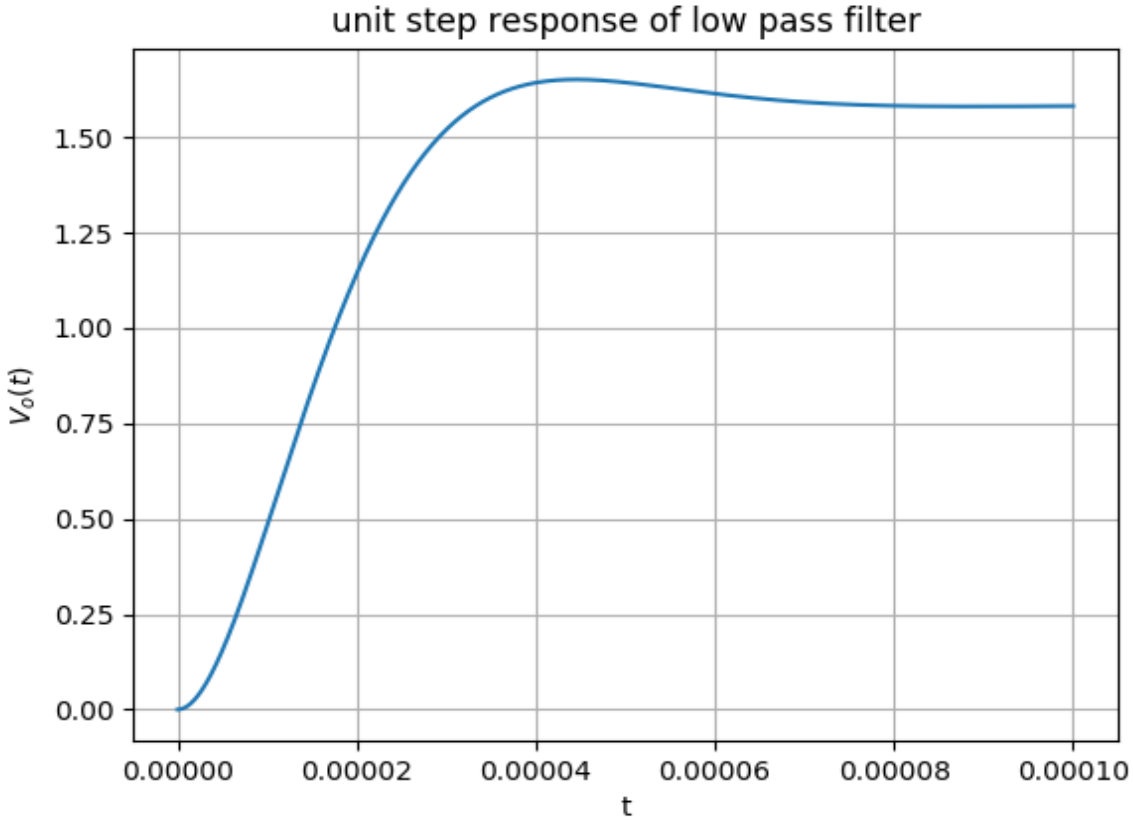
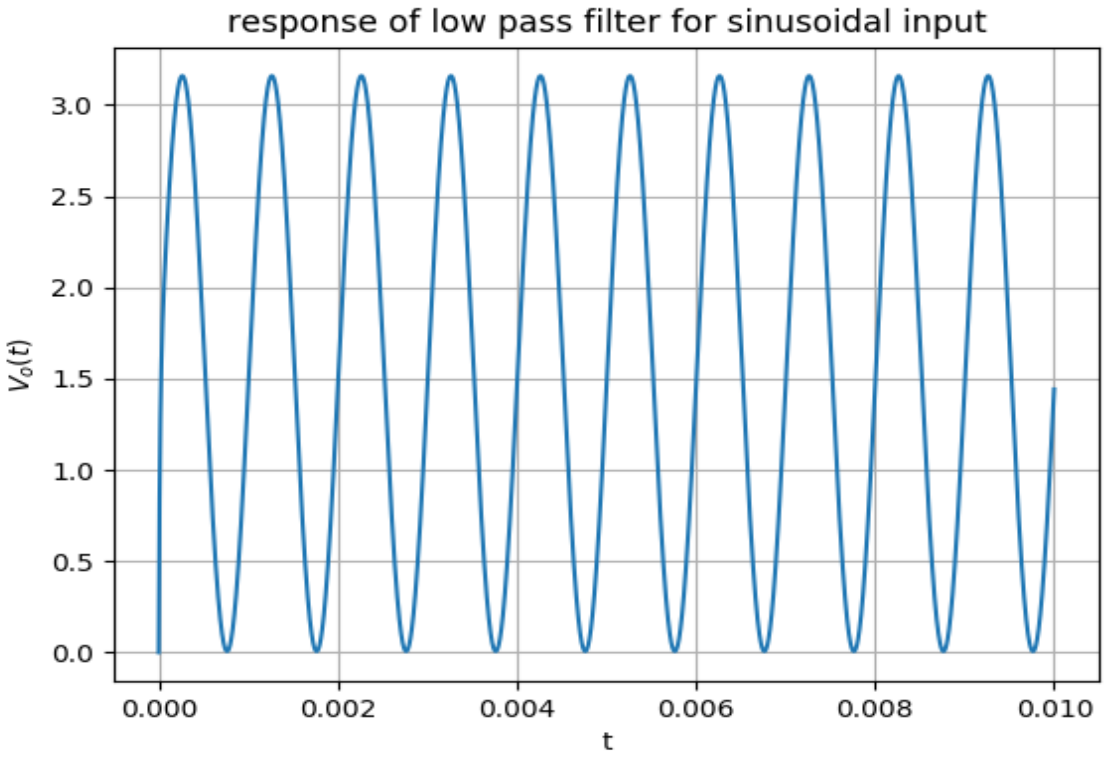


Figure 3: Response of filter for sinusoidal input with two frequencies



- From figure 1 the frequencies greater than 10^5rad/s are attenuated by filter.The lower frequencies are passed with gain of 1.58.The transfer function of this second order filter $\frac{0.1}{6.31517e^{-12}s^2+8.9455e^{-7}s+0.06315}$ clearly dc gain is 1.58 when we put s=0.It is second order low pass filter with cutoff close to

10^5rad/s.These filters are butterworth filters as they are maximally flat with quality factor close to $\frac{1}{\sqrt{2}}$.

- From figure 2 ,since system has damping factor close to $\frac{1}{\sqrt{2}}$,it is underdamped system therefore the response to unit step peaks and then settles to 1.58(dc gain).
- From figure 3 since input has two frequencies $2\pi e^3$and $2\pi e^6$rad/s and our system has cutooff close to 10^5rad/s. The higher frequency is filtered and response will be sinusoid of lower frequency with ripples on it.

3 High pass filter

3.1 Writing the nodal equations:

$$\begin{pmatrix} 0 & 0 & 1 & \frac{-1}{G} \\ \frac{-1}{1+\frac{1}{sC_2R_2}} & 1 & 0 & 0 \\ 0 & 1000 & -1000 & 1 \\ sC_1 - sC_2 - \frac{1}{R_1} & sC_2 & 0 & \frac{1}{R_1} \end{pmatrix} \begin{pmatrix} V_1 \\ V_P \\ V_m \\ V_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -V_i(s)sC_1 \end{pmatrix}$$

- The transfer of the system $\frac{V_o(s)}{V_i(s)}$ comes from solving for x in Ax=b ie x = inv(A).b
- $V_o(s)$ is in x[3].
- For calculating impulse response of system we give $V_i(s) = 1$ and for step response of system we give $V_i(s) = \frac{1}{s}$.
- Sympy helps in defining matrix with symbols. inv function calculates inverse of matrix.The lambdify takes $V_o(s)$ and converts it into a python function.

```
def highpass(R1,R2,C1,C2,G,Vi):
    A=Matrix([[0,0,1.0,-1.0/G],[-1.0/(1.0+1.0/(s*R2*C2)),1.0,0,0],
    [0,1000.0,-1000.0,1.0],[-1.0/R1-s*C1-s*C2,s*C2,0,1.0/R1]])
    b=Matrix([0,0,0,-Vi*s*C1])
    V=A.inv()*b
    return (A,b,V)
```

3.2 Plot magnitude response

- we define range of w for which magnitude response to be calculated using logspace.
- Since we have transfer function as function of s. For bode plot we put s=jw.The corresponding plot is in figure 4

```
A,b,V=highpass(10000.0,10000.0,1e-9,1e-9,1.586,1.0)
Vo=V[3]
print simplify(Vo)
w=p.logspace(0,8,801)
ss= 1j*w
hf=lambdify(s,Vo,'numpy')
v=hf(ss)
p.loglog(w,abs(v),lw=2)
p.title('High pass filter')
p.xlabel('$\omega(rad/s)$')
p.ylabel('$Magnitude$')
p.grid(True)
p.show()
```

3.3 Unit step response of low pass filter.

- For unit step response since we have H(s) , output Y(s) = H(s)X(s). For unit step X(s) = 1/s.Therefore Y(s)= H(s)/s.
- We can find the corresponding coefficients of Y(s) and give as input to sp.lti function which gives y(t).
- Next we plot output y(t) vs t in figure 5

```
#unit step response
H=sp.lti([1e-9,0],[6.2951e-10,8.885e-5,6.29517])
t,x=sp.impulse(H,None,p.linspace(0,0.0001,1001))
p.title('unit step response of high pass filter')
p.xlabel('t')
p.ylabel('$V_o(t)$')
p.plot(t,x)
p.grid(True)
p.show()
```

3.4 Response for sinusoidal input.

- Let input be $V_i(t) = exp(-1e^4t)cos(2e^5\pi t)$.The h(t) can be calculted from coefficients of H(s) using sp.lti function.
- sp.lsim gives us output by convolving h(t) with input $v_i(t)$.The plot of response is in figure 6

```
#for sinusoidal input
H=sp.lti([1e-9,0,0],[6.2951e-10,8.885e-5,6.29517])
t=p.linspace(0,5e-4,1001)
u=p.exp(-1e4*t)*p.cos(2e5*p.pi*t)
t,y,svec=sp.lsim(H,u,t)
p.title('response of high pass filter for damped sinusoidal input')
p.xlabel('t')
p.ylabel('$V_o(t)$')
p.plot(t,y)
p.grid(True)
p.show()
```

3.5 Plots and discussion

Figure 4: Magnitude response of High Pass filter

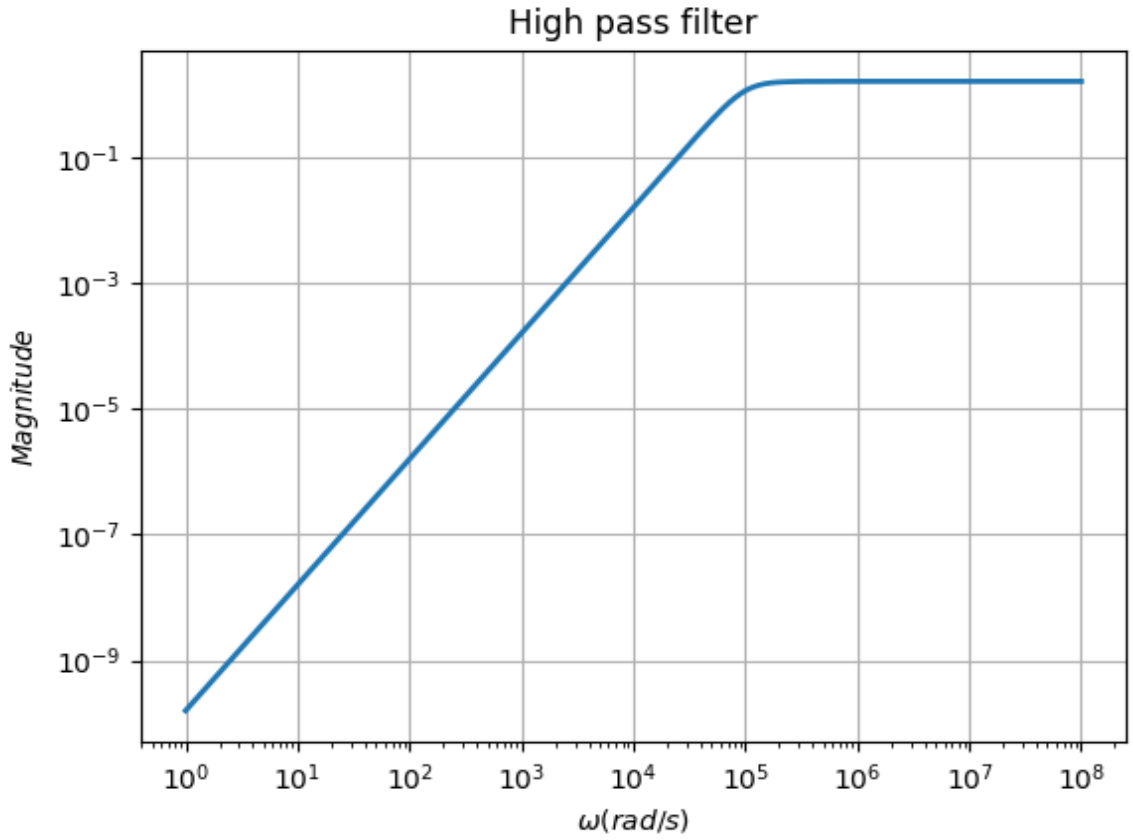


Figure 5: Unit step response of High pass filter

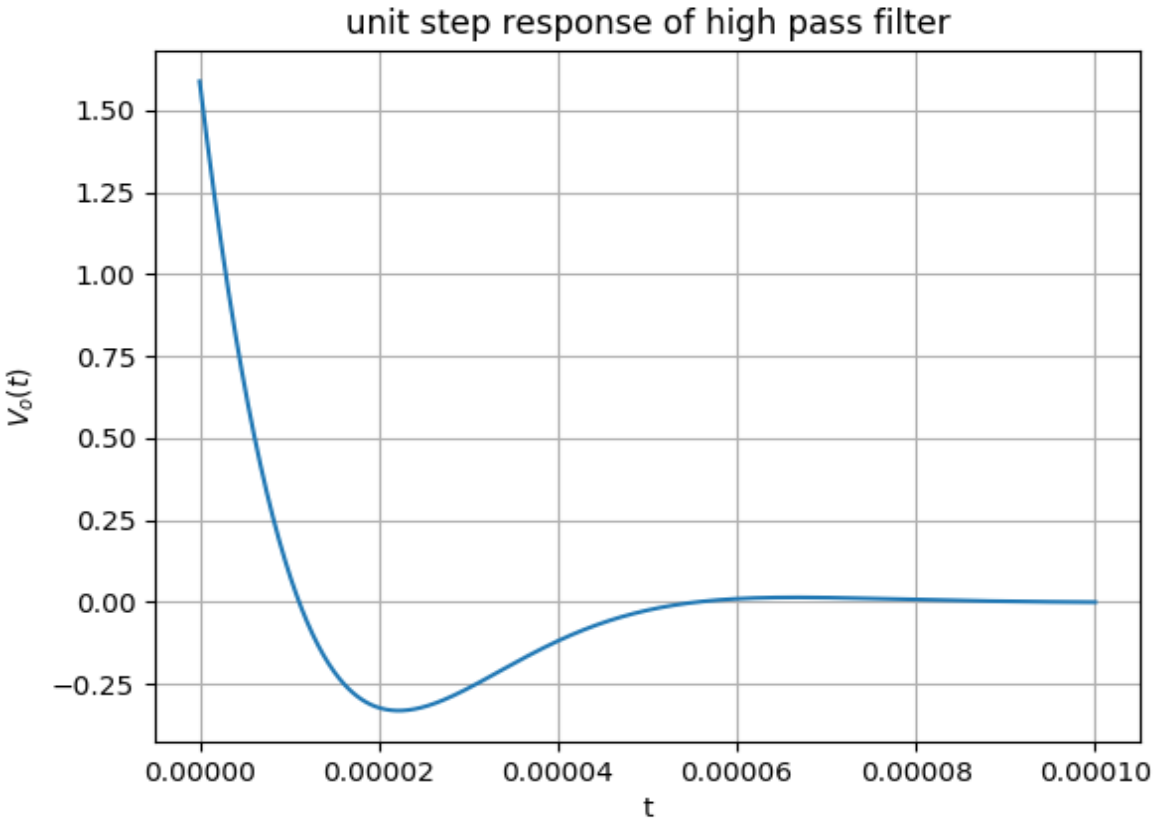
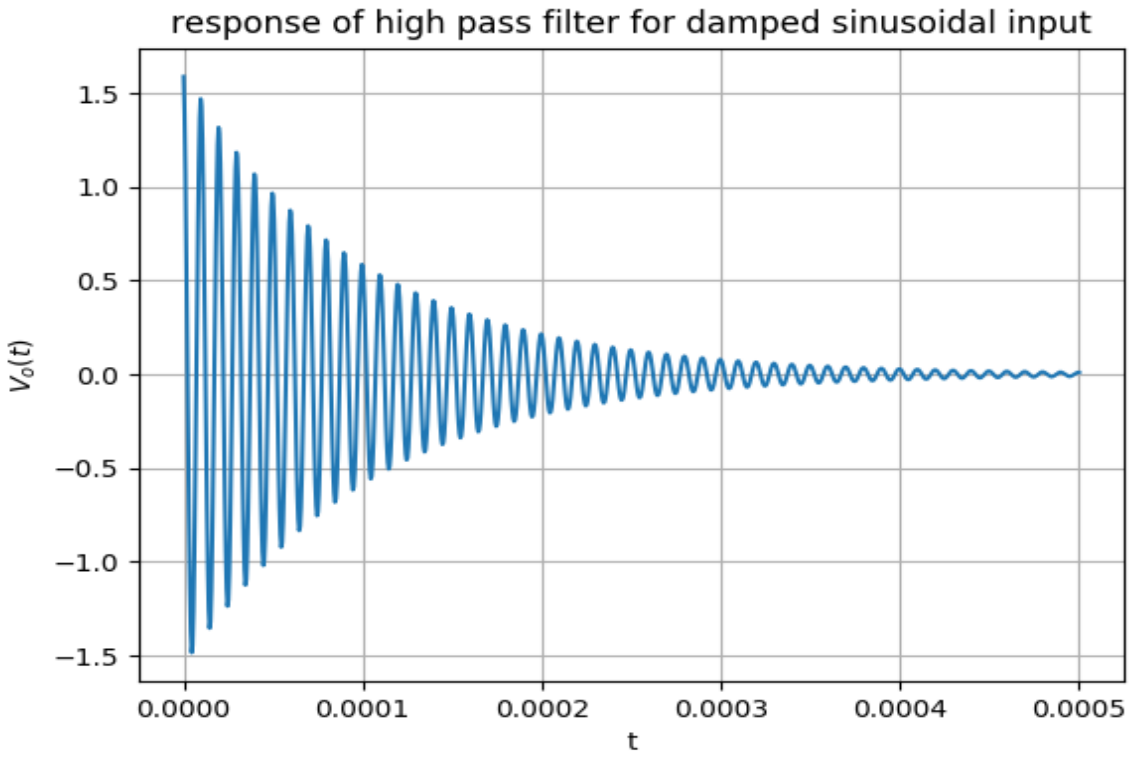


Figure 6: Response of filter for damped sinusoid input $V_i(t) = exp(-1e^4 t)cos(2e^5 \pi t)$



- From figure 4, it is high pass since it filters out low frequencies. It allow frequencies greater than 10^5 with gain of 1.58.The transfer function of second order filter is $\frac{1e-9s^2}{6.2951e^{-10}s^2+8.885e^{-5}s+6.29517}$.
- From figure 5, The output to step response is exactly inverted to that of low pass filter.The output starts from high value and decays to zero with a minimum peak as t increases which is expected because in steady state since input is dc and high pass filters out dc, output should be zero.
- From figure 6, The filter passes out input damped sinusoid as cutoff is less than input frequency but since it decays with time . The output is decaying sinusoid.