EE2703 : Applied Programming Lab

G.Rohith Kumar
EE18b008

April 9, 2020

Assignment 7

Abstract

Analysis of circuits using Laplace transforms and this assignment focus on solving circuits using symbolic algebra.

We also try to Realise Lowpass and Highpass filters.

Requirments:

Modules to be imported,

```
from sympy import *
import scipy.signal as sp
import pylab as p
```

Lowpass filter:

Lowpass filter is a filter which is used to separate lower frequencies from higher frequencies.It will pass the lower frequency signals and attenuate higher frequency signals to zero.
Writing equations for given lowpass circuit and solving it using matrix equation Ax=B will give you variable matrix x.The matrix contains Vout in terms of Vi.

By giving Vi as impulse signal(i.e.,delta) you get the impulse response of this lowpass filter as Vout.Using this transfer function we can simulate the output of any given input with sp.lsim.

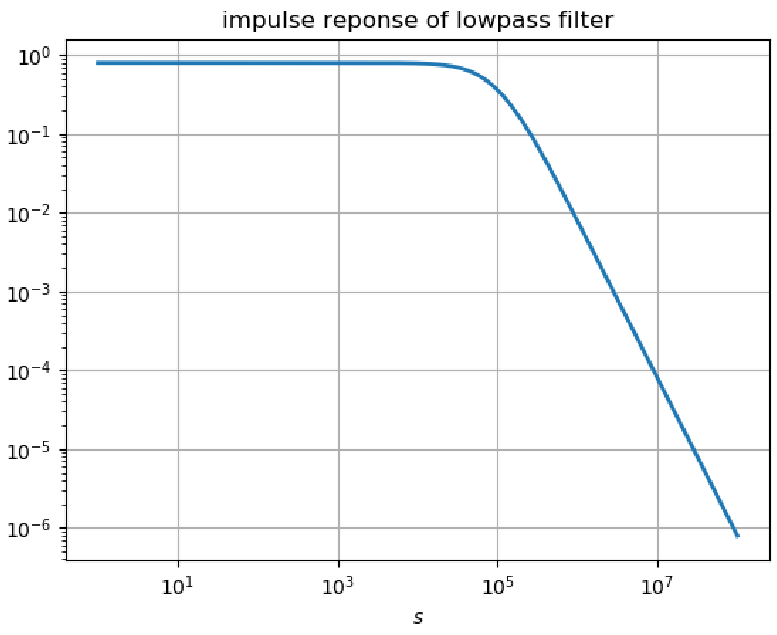
$$H(s) = \frac{-0.0001586}{2 * 10^{-14}s^2 + 4.414 * 10^{-9}s + 0.0002}$$

Code:

```
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*C1,1/R2,0,s*C1]])
        b=Matrix([0,0,0,Vi/R1])
        V=A.inv()*b
        return (A,b,V)
A,b,V=lowpass(10000,10000,1e-9,1e-9,1.586,1)
Vo=V[3]
ww=p.logspace(0,8,801)
ss=1j*ww
hf=lambdify(s,Vo,'numpy')
v=hf(ss)
h2=simplify(Vo)
numer,denom=h2.as_numer_denom()
num=poly(numer,s)
den=poly(denom,s)

Hlp=sp.lti([float(i) for i in num.all_coeffs()],
                    [float(i) for i in den.all_coeffs()])
                    p.figure(1)
p.title("impulse reponse of lowpass filter")
p.xlabel(r'$s$',size=10)
p.loglog(ww,abs(v),lw=2)
p.grid(True)
```



Highpass filter

Highpass filter is a filter which is used to separate higher frequencies from lower frequencies.It will pass the higher frequency signals and attenuate lower frequency signals to zero.

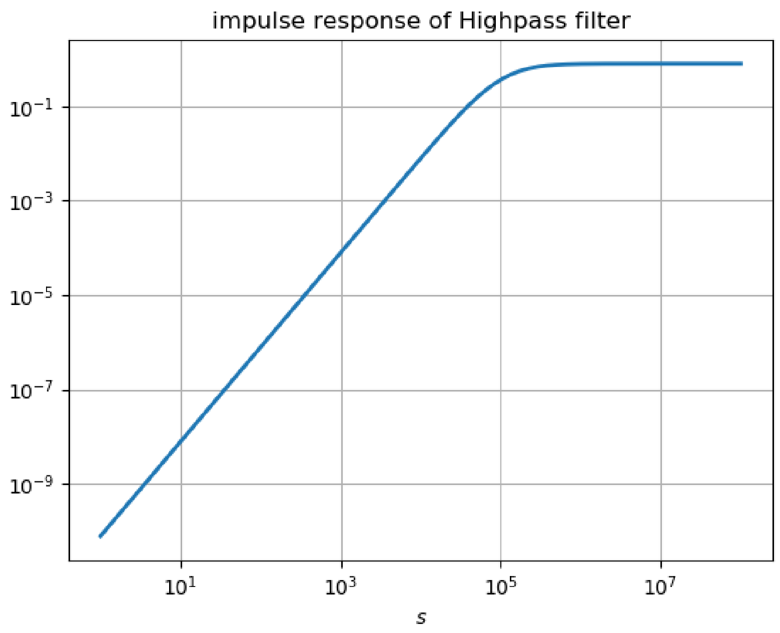
Similarly, By solving equations for given highpass circuit we get the impulse response of highpass filter.

$$H(s) = \frac{1.586 * 10^{-14}s^2}{2 * 10^{-14}s^2 + 4.414 * 10^{-9}s + 0.0002}$$

Code:

```
def highpass(R1,R3,C1,C2,G,Vi):

    A=Matrix([[0,0,1,-1/G],[0,G,-G,-1],
            [s*C2*R3/(1+s*C2*R3),-1,0,0],[(1/R1)+s*C1,1/R3,0,-1/R1]])
    b=Matrix([0,0,0,Vi*s*C1])
    V=A.inv()*b
    return (A,b,V)
A,b,V=highpass(10000,10000,1e-9,1e-9,1.586,1)
Vo=V[3]
hf=lambdify(s,Vo,'numpy')
v=hf(ss)
h2=simplify(Vo)
numer,denom=h2.as_numer_denom()
num=poly(numer,s)
den=poly(denom,s)
Hhp=sp.lti([float(i) for i in num.all_coeffs()],
                        [float(i) for i in den.all_coeffs()])
                        p.figure(2)
p.title("impulse response of Highpass filter")
p.xlabel(r'$s$',size=10)
p.loglog(ww,abs(v),lw=2)
p.grid(True)
```



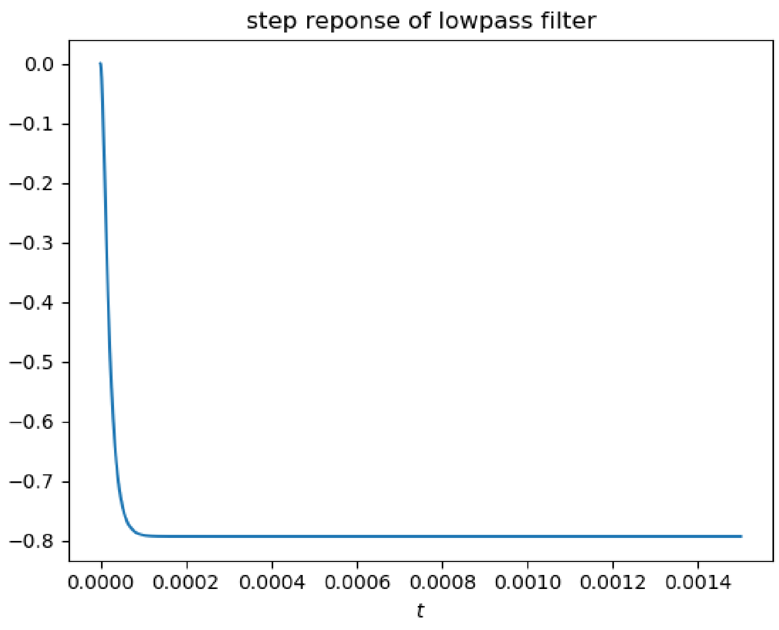
Q1:

Giving Step response as input to Lowpass filter.Using sp.lsim to simulate the output.

Code:

```
tm=p.linspace(0,1.5e-3,50000)

v1=p.array([1 for i in tm])                          # Step function
y1=sp.lsim(Hlp,v1,tm)[1]
p.figure(3)
p.title("step reponse of lowpass filter")
p.xlabel(r'$t$',size=10)
p.plot(tm,y1)
```



Q2,Q3:

Giving input as

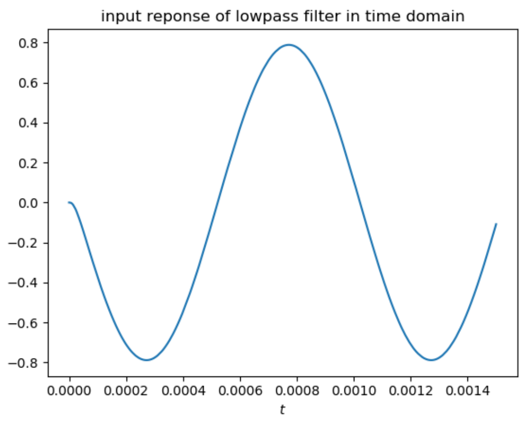
$$V_i(t) = (sin(2000\pi t) + cos(2 * 10^6 \pi t))u_o(t)$$

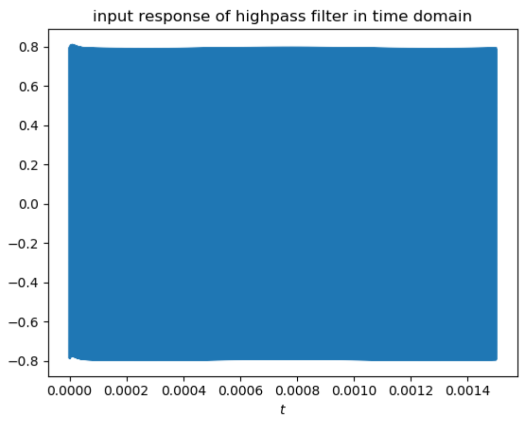
to both highpass and lowpass filters.

Code:

```
v2=p.sin(2000*p.pi*tm)+p.cos(2*1e6*p.pi*tm)      # given input
y2=sp.lsim(Hlp,v2,tm)[1]
p.figure(4)
p.title("input reponse of lowpass filter in time domain")
p.xlabel(r'$t$',size=10)
p.plot(tm,y2)

y3=sp.lsim(Hhp,v2,tm)[1]
p.figure(5)
p.title("input response of highpass filter in time domain")
p.xlabel(r'$t$',size=10)
p.plot(tm,y3)
```





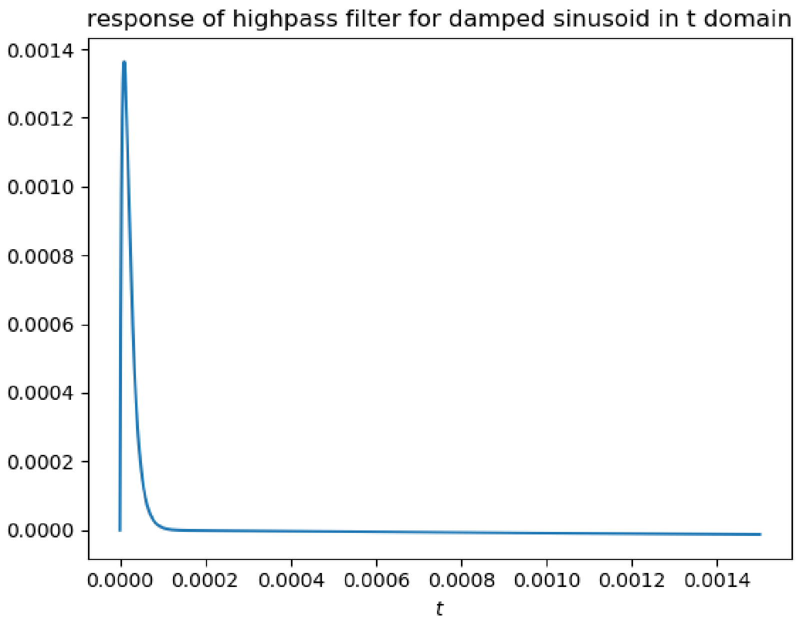
if you observe the input has two frequencies, one higher and one lower. When you passed this input through both filter, only low frequency component comes out from lowpass filter and high frequency component comes from highpass filter.

Q4:

Giving a damped sinusiod signal as input to highpass filter and Using sp.lsim to simulate the output.

Code:

```
omega=500
decay=0.05
v4=(p.e**(-decay*tm))*p.sin(omega*tm)                          # Damped sinusoid
y4=sp.lsim(Hhp,v4,tm)[1]
p.figure(6)
p.title("response of highpass filter for damped sinusoid in t domain")
p.xlabel(r'$t$',size=10)
p.plot(tm,y4)
```

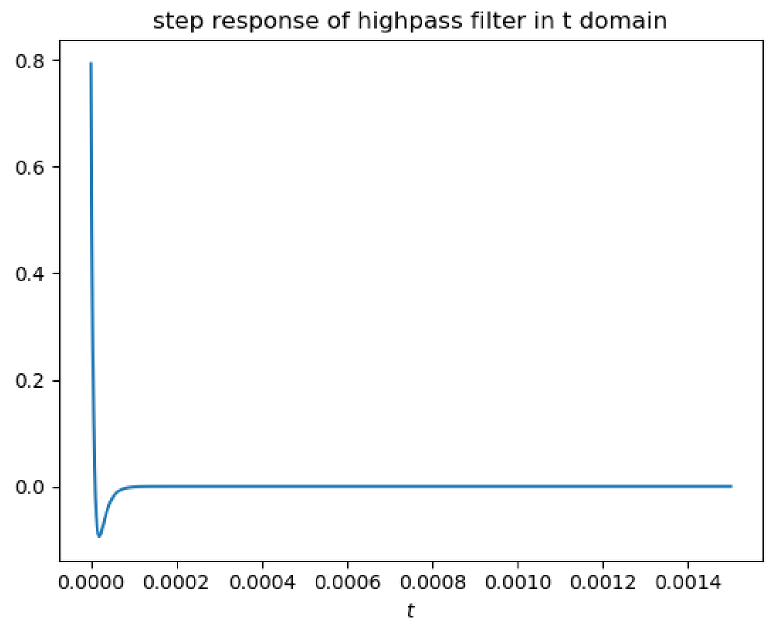


Q5:

Giving Step response as input to highpass filter and Using sp.lsim to simulate the output.

Code:

```
v1=p.array([1 for i in tm])                       # Step function
y5=sp.lsim(Hhp,v1,tm)[1]
p.figure(7)
p.title("step response of highpass filter in t domain")
p.xlabel(r'$t$',size=10)
p.plot(tm,y5)
```



Conclusion

- In this assignment we learnt how to use sympy in python.
- We learnt how to Analyse circuits using Laplace transforms.
- We also Observed the behaviour of output signal from Lowpasss and Highpass filters for the given input.