EE2703 : Applied Programming Lab

G.Rohith Kumar
EE18b008

March 12, 2020

Assignment 6

Abstract

Analysing Linear Time-invariant Systems Using NUmerical tools in python. Using signals toolbox in python we will use Laplace transforms to solve problems in continuous time.

Requirments:

Modules to be imported,

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

Q1,Q2:

Calculating the Laplace transform of x(t) and using sp.impulse to find response in time domain.

$$X(s) = \frac{F(s)}{s^2 + 2.25}$$

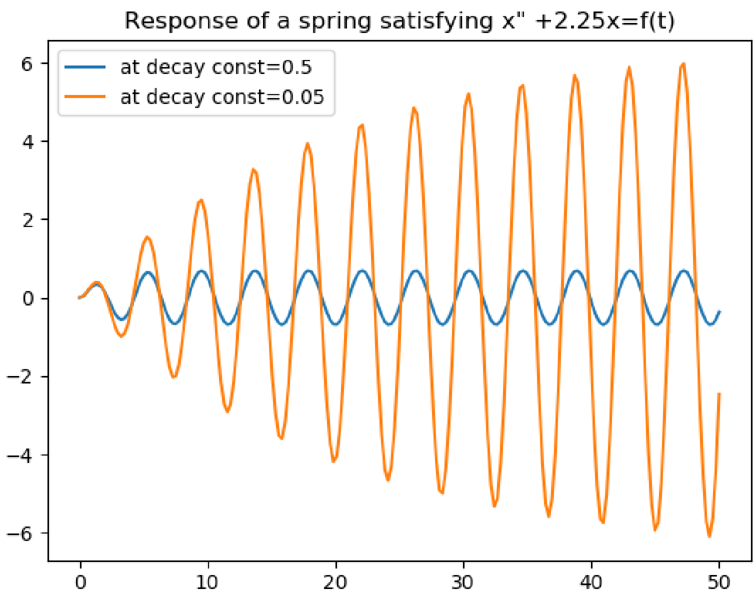
for decay constant 0.05,

$$F(s) = \frac{s + 0.05}{(s + 0.05)^2 + 2.25}$$

Code:

```
c2=0.05
c1=0.5
p1=poly1d([1,c1])
p2=poly1d([1,2*c1,2.25+c1**2])
p3=poly1d([1,0,2.25])
d1=poly1d([1,c2])
d2=poly1d([1,2*c2,2.25+c2**2])
d3=poly1d([1,0,2.25])
X1=sp.lti(p1,polymul(p2,p3))
X2=sp.lti(d1,polymul(d2,d3))
t1,x1=sp.impulse(X1,None,linspace(0,50,200))
```

```
t2,x2=sp.impulse(X2,None,linspace(0,50,200))
figure(1)
title('Response of a spring satisfying x" +2.25x=f(t)')
plot(t1,x1,label='at decay const=0.5')
plot(t2,x2,label='at decay const=0.05')
legend(loc='upper left')
```



Q3:

Calculating the Transfer function of LTI system and using sp.lsim to simulate the output of LTI system.

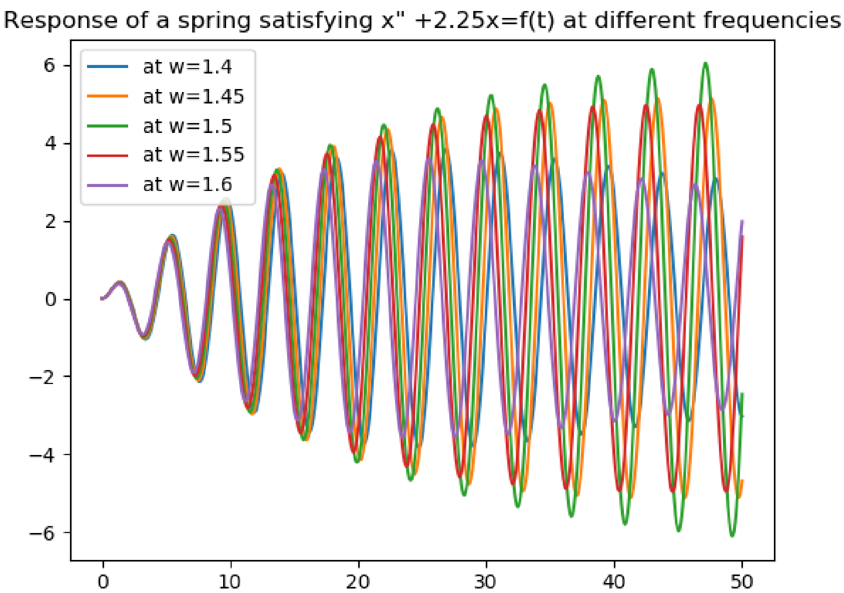
$$H(s) = \frac{1}{s^2 + 2.25}$$

$$f(t) = cos(wt)e^{-0.05t}u(t)$$

Analysing the LTI system by varying omega(w) from 1.4 to 1.6 in steps of 0.05.

Code:

```
w=linspace(1.4,1.6,5)
t=t2
y=[]
figure(2)
title('Response of a spring satisfying x" +2.25x=f(t) at different frequencies')
for i in w:
    H=sp.lti([1],[1,0,2.25])
    f=cos(i*t)*(e**(-0.05*t))
    y.append(sp.lsim(H,f,t)[1])
    plot(t,y[-1],label='at w='+str(i))
legend(loc='upper left')
```



Q4:

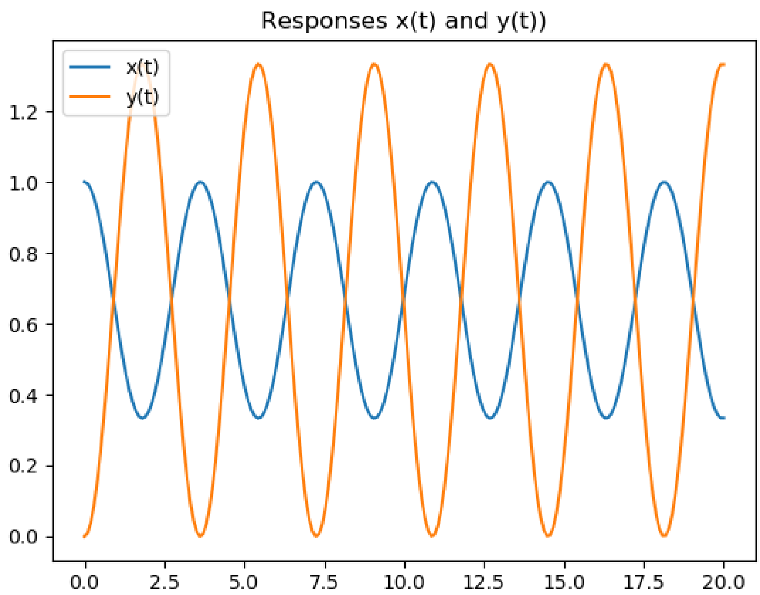
Similar to Question 1,Get Laplace transforms for x and y and use sp.impulse.

$$X(s) = \frac{s^2+2}{s^3+3s}$$

$$Y(s) = \frac{2}{s^3+3s}$$

Code:

```
Xs=sp.lti([1,0,2],[1,0,3,0])                                          # laplace tr
Ys=sp.lti([2],[1,0,3,0])                                                # laplac
tx,x=sp.impulse(Xs,None,linspace(0,20,200))
ty,y=sp.impulse(Ys,None,linspace(0,20,200))
#subplot(3,1,3)
figure(3)
title('Responses x(t) and y(t))')
plot(tx,x,label='x(t)')
plot(ty,y,label='y(t)')
legend(loc='upper left')
```



Q5,Q6:

Calculating

$$H(s) = \frac{V_o(s)}{V_i(s)}$$

$$H(s) = \frac{1}{s^2LC + sRC + 1}$$

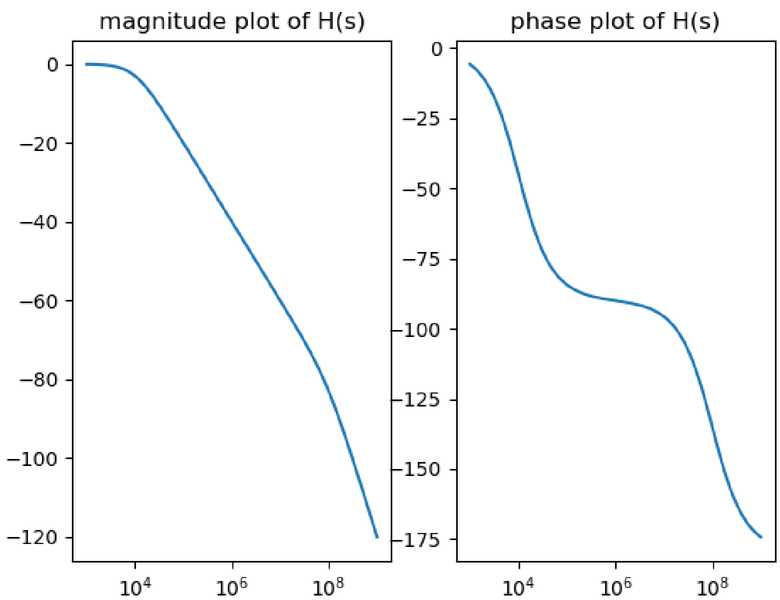
From given values of R,L,C

$$H(s) = \frac{1}{10^{-12}s^2 + 10^{-4}s + 1}$$

from given circuit and Using H.bode() to get magnitude and phase plot of transfer function.

Code:

```
Hs=sp.lti([1],[1e-12,1e-4,1])
freq,S,phi=Hs.bode()
figure(4)
subplot(1,2,1)
title('magnitude plot of H(s)')
semilogx(freq,S)
subplot(1,2,2)
title('phase plot of H(s)')
semilogx(freq,phi)
```



By giving signal input as

$$V_i(t) = cos(10^3t)u(t) - cos(10^6t)u(t)$$

and

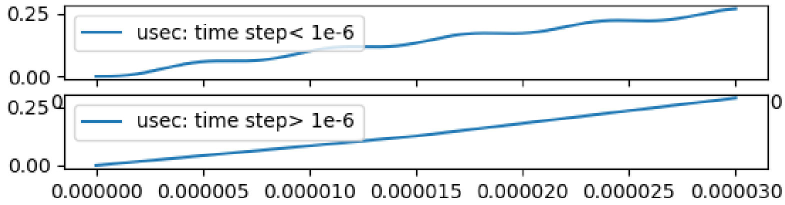
$$H(s) = \frac{1}{10^{-12}s^2 + 10^{-4}s + 1}$$

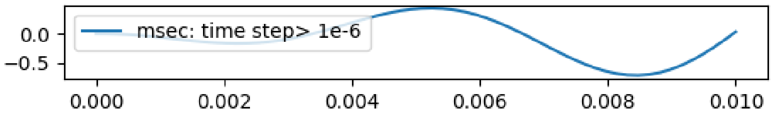
Using sp.lsim to obtain output signal.

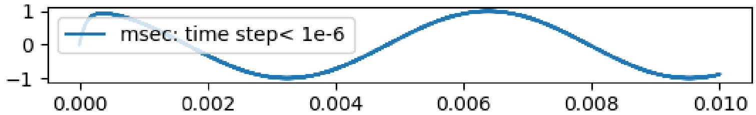
Vary the time scale and time step and check the output signal to know the Behaviour of it.

Code:

```
timee=[linspace(0,30e-6,60),linspace(0,30e-6,3),
            linspace(0,30e-3,60),linspace(0,30e-3,120e4)]
vo=[]
for i in timee:
    vi=cos((10**3)*i)-cos((10**6)*i)
    vo.append(sp.lsim(Hs,vi,i)[1])
    plot(i,vo[-1],label='time step='+str(i[-1]/len(i)))
legend(loc='upper left')
figure(6)
subplot(6,1,1)
plot(timee[0],vo[0],label='usec: time step< 1e-6')
legend(loc='upper left')
subplot(6,1,2)
plot(timee[1],vo[1],label='usec: time step> 1e-6')
legend(loc='upper left')
subplot(6,1,4)
plot(timee[2],vo[2],label='msec: time step> 1e-6')
legend(loc='upper left')
subplot(6,1,6)
plot(timee[3],vo[3],label='msec: time step< 1e-6')
legend(loc='upper left')
```

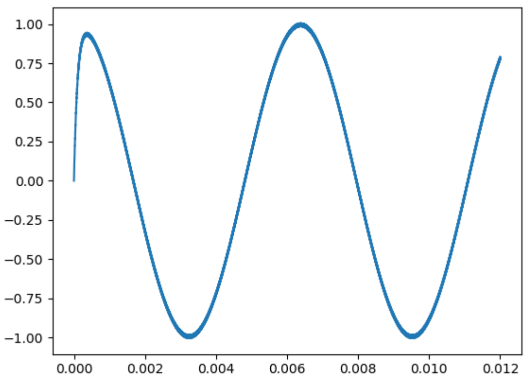


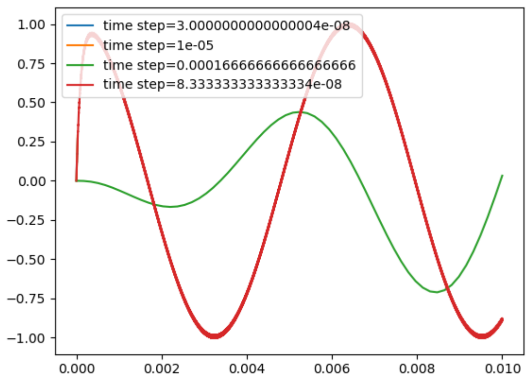




Code:

```
figure(7)
tm=linspace(0,12*(10**-3),50000)
vi=cos((10**3)*tm)-cos((10**6)*tm)
plot(tm,sp.lsim(Hs,vi,tm)[1])
```





Conclusion

- In this assignment we learnt how to use signal toolbox in python.
- We learnt to simulate output of LTI systems using signal.lsim().
- We also Observed the behaviour of output signal of an RLC circuit for inputs with different frequencies.