EE2703 : Applied Programming Lab
Assignment 5

Sasanapuri Uttam Raj
EE20B118

March 2022

Abstract

The aim of the assignment is to use the signal toolbox in python in analysing LTI (Linear-Time Invariant) systems.

Time response of a spring system

Given time variant signal f(t):

$$f(t) = cos(1.5t)e^{-0.5t}u_0(t)$$

Its Laplace transform is known to be:

$$F(s) = \frac{s + 0.5}{(s + 0.5)^2 + 2.25}$$

To obtain the time response of a spring system given by the equation

$$\ddot{x} + 2.25x = f(t)$$

we find X(s) from the time response of the spring and use *system.impulse* to find the inverse x(t)

```
num1 = np.poly1d([1,0.5])
den1 = np.polymul([1,1,2.5],[1,0,2.25])
X1 = sp.lti(num1,den1)
t1,x1 = sp.impulse(X1,None,np.linspace(0,50,500))
pl.plot(t1, x1)
pl.xlabel(r'$t$')
pl.ylabel(r'$x(t)$')
pl.show()
```

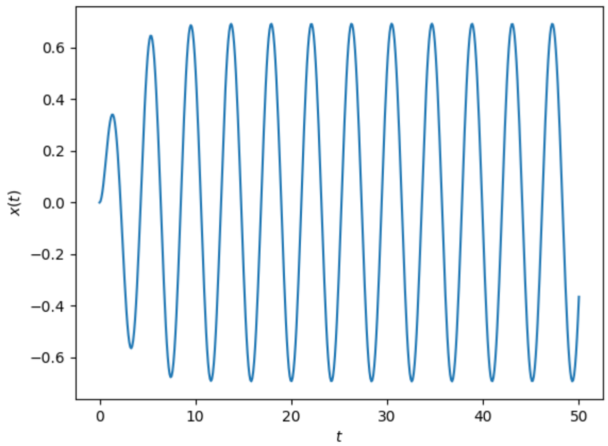


Figure 1: System Response with Decay = 0.5

Under the initial conditions x(0) = 0 and ẋ(t) = 0, The plot of x(t) vs time can be seen as shown above.
For a smaller decay of 0.05,the equation is:

$$f(t) = cos(1.5t)e^{-0.05t}u_0(t)$$

We get the following plot:

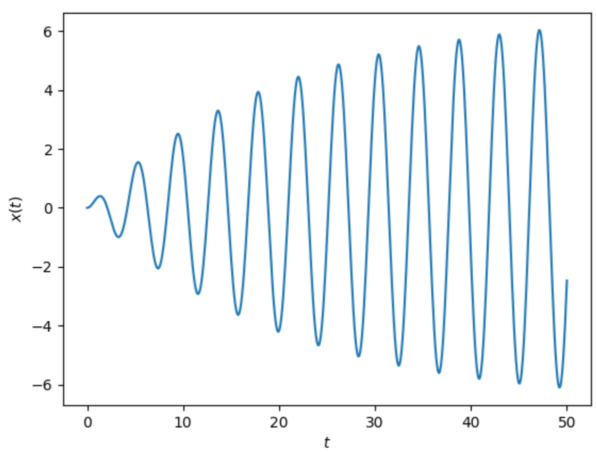
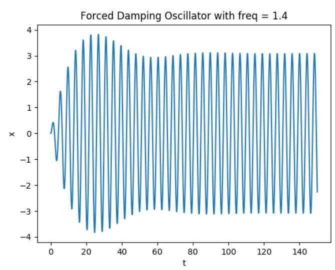


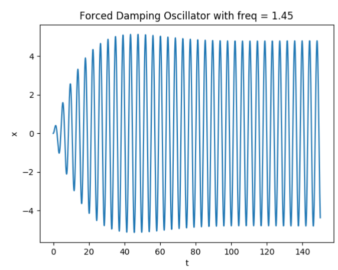
Figure 2: System Response with Decay = 0.05

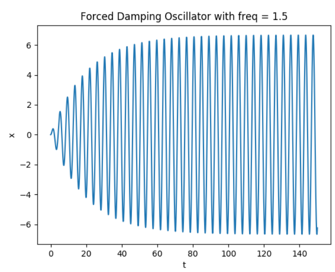
Response over different frequencies

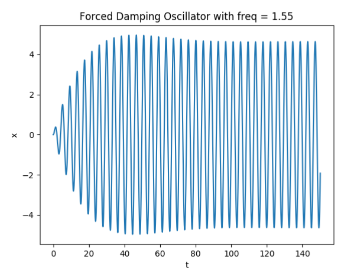
For a fixed delay coefficient of 0.05 and a set of frequencies between 1.4 and 1.6 in steps of 0.05, we try to plot the corresponding time responses of the spring.

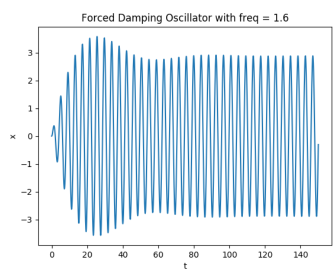
```
for w in np.arange(1.4,1.65,0.05):
t = np.linspace(0,50,500)
f = np.cos(w*t)*np.exp(-0.05*t)
t,x,svec = sp.lsim(H1,f,t)
pl.figure(3)
pl.plot(t,x,label='w = ' + str(w))
```











Now looking all the frequency plots together on a single graph, we get:

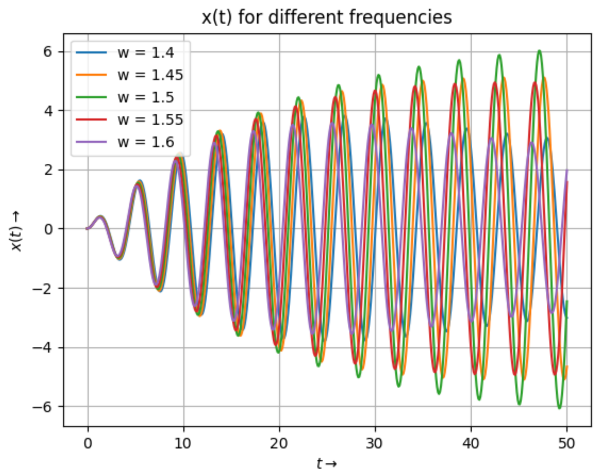


Figure 3: System Response with frequency various frequencies

For the equation, we can see that the natural response of the has the frequency $\omega = 1.5$ rad/sec. Thus, the maximum amplitude of oscillation is obtained when frequency of f(t) is 1.5 rad/sec.

Coupled spring problem

For a set of coupled springs related by the response equations as:

$$\ddot{x} + (x - y) = 0$$

$$\ddot{y} + 2(y - x) = 0$$

We try to find the individual x(t) and y(t) responses.

```
t = np.linspace(0,20,1000)
H_x = sp.lti([1,0,2],[1,0,3,0])
t3,x = sp.impulse(H_x,T=t)
H_y = sp.lti([2],[1,0,3,0])
t4,y = sp.impulse(H_y,T=t)
pl.figure(4)
pl.plot(t3, x)
pl.plot(t4, y)
pl.legend([r'$x(t)$', r'$y(t)$'])
pl.xlabel(r'$t$')
pl.show()
```

Under the initial conditions

$$x(0) = 1, \dot{x}(0) = y(0) = \dot{y}(0) = 0$$

, we get the transfer functions

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

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

We notice that the outputs of this system are 2 sinusoids which are out of phase. This system can be realized by creating a single spring double mass system.

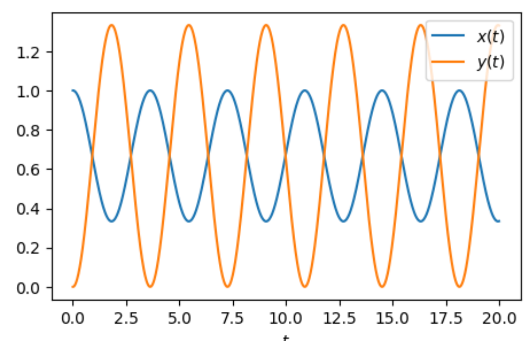


Figure 4: Coupled Oscillations

RLC Filter

Now we try to create the bode (magnitude and phase) plots for the RLC filter defined in the question

$$H(s) = \frac{10^6}{s^2 + 100s + 10^6}$$

```
L = 1e-6; R = 100; C = 1e-6
H2 = sp.lti([1], [L*C, R*C, 1])
w,S,phi = H2.bode()
```

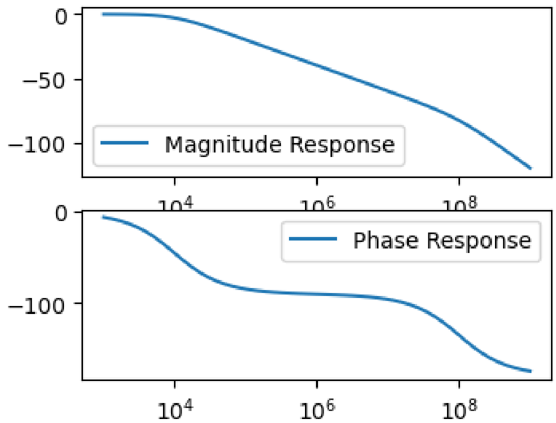


Figure 5: Bode plot of H(s)

We now plot the response of the RLC filter to the input:

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

for 0< t< 30µs and 0< t< 30ms

```
time = np.linspace(0, 10e-3, 10001)
vi = np.cos(1e3*time) - np.cos(1e6*time)
time, vo, svec = sp.lsim(H2, vi, time)
```

The system behaves as a low pass filter. The lower frequencies get slightly damped whereas the higher frequencies gets highly damped.

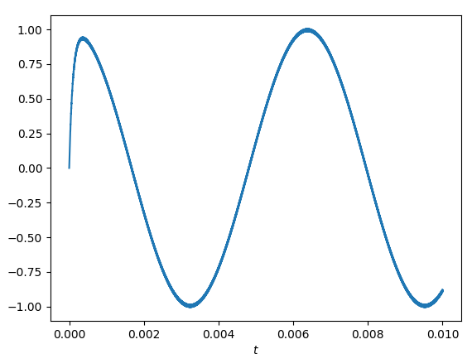


Figure 6: Output for a large time interval

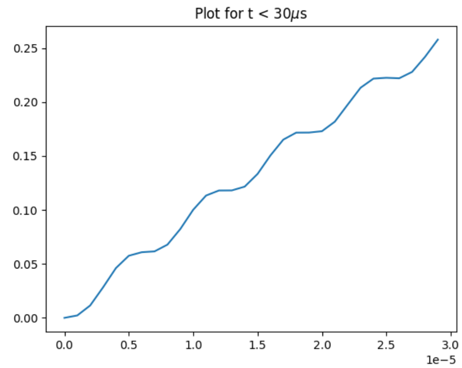


Figure 7: System Response for t<30ms

Conclusion

LTI systems are observed in all fields of engineering and are very important. In this assignment, we have used scipy's signal processing library to analyze a wide range of LTI systems. Specifically we analyzed forced oscillatory systems, single spring, double mass systems and Electric filters.