Assignment 6: The Laplace Transform

Siddharth D P
EE18B072

March 10, 2020

1 Abstract

The goal of this assignment is the following.

- To analyze LTI systems using Laplace Transform.
- To see how RLC systems can be used as a low low pass filter.
- To understand how to use the Scipy signals toolkit.
- To plot Bode plots to visualise the Laplace Transform.

2 Assignment

2.1 Importing the standard libraries

```
import scipy.signal as sp
import numpy as np
import matplotlib as mp
import matplotlib.pyplot as plt
```

3 Single Spring System

3.1 Varying the Decay of the Input

We use the Laplace transform to solve a simple spring system. The system is characterized by the given differential equation-

$$\frac{d^2x}{dt^2} + 2.25x = f(t)$$

whose Laplace transform with all initial conditions zero is of the form

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

The input signal is of the form $f(t) = \cos(\omega t)e^{-at}u(t)$, where a is the decay factor and ω is the frequency of the cosine. The Laplace Transform of the input signal is

$$F(s) = \frac{s+a}{(s+a)^2+\omega^2}$$

First, we define the decay functions using **numpy** polynomials and multiply to get the output Laplace Transform. Finally, we take the Inverse Laplace Transform using the *sp.impulse* to get the time domain sequences and we plot these. We do this for $\omega = 1.5$ (which is the given natural frequency of the system) for a decay of a=0.5 and a=0.05.

```
#f=(np.cos(1.5*t))*(np.exp(-(0.5*t)))
F1=sp.lti([1,0.5],[1,1,2.5])
X = sp.lti([1,0.5],np.polymul([1,1,2.5],[1,0,2.25]))
H1=sp.lti(1,(1,0,2.25))
t,x=sp.impulse(X,None,np.linspace(0,50,501))

plt.figure(1)
plt.plot(t,x,'b-')
plt.title("Solution of differential equation with decay 0.5")
plt.grid('True')
plt.xlabel("t")
plt.ylabel("x(t)")
plt.show()

#f=(np.cos(1.5*t))*(np.exp(-(0.05*t)))
F2=sp.lti([1,0.05],[1,0.1,2.2525])
Y = sp.lti([1,0.05],np.polymul([1,0.1,2.2525],[1,0,2.25]))
H2=sp.lti(1,(1,0,2.25))
t,y=sp.impulse(Y,None,np.linspace(0,50,501))

plt.figure(2)
plt.plot(t,y,'b-')
plt.title("Solution of differential equation with decay 0.05")
plt.grid('True')
plt.xlabel("t")
plt.ylabel("x(t)")
plt.show()
```

3.2 Varying the frequency of the input

We vary the frequency of the cos function from 1.4 to 1.6 in steps of 0.05 and see what affect it has on the output. The responses are graphed for a better understanding.

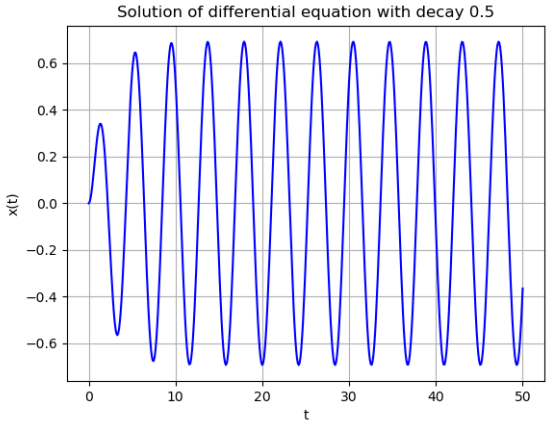

Figure 1:

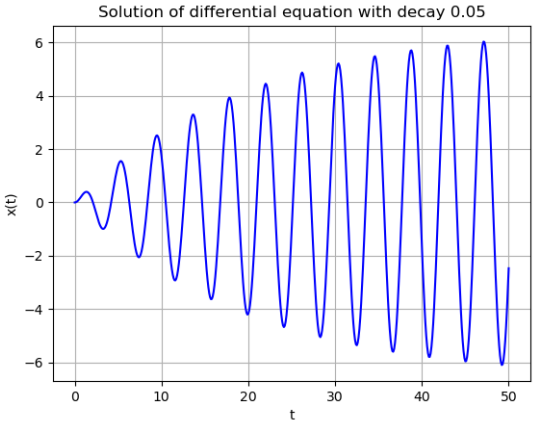

Figure 2:

```
plt.figure(3)
#defining the for loop for the variable frequencies from 1.4 to 1.6
#with decay remaining fixed at 0.05
for i in range(0,5,1):
    freq=np.around(1.4+i*0.05,2)
    p=0.0025+freq*freq
    F=sp.lti([1,0.05],[1,0.1,p])
    Z = sp.lti([1,0.05],np.polymul([1,0.1,p],[1,0,2.25]))
    t,z=sp.impulse(Z,None,np.linspace(0,50,501))
    plt.plot(t,z,label='Frequency = {}'.format(freq))

plt.title("Frequency varying from 1.4 to 1.6 with decay fixed at 0.05")
```

```
plt.grid('True')
plt.xlabel("t")
plt.ylabel("x(t)")
plt.legend()
plt.show()
```

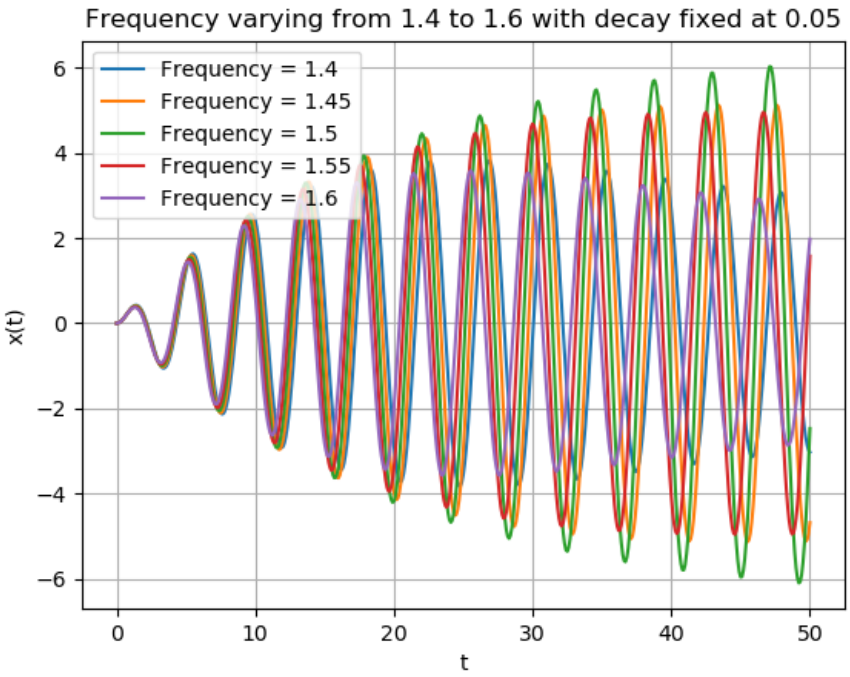

Figure 3:

When the input frequency is at the natural frequency, the output amplitude is maximum. In the other cases the output amplitude decreases. This phenomenon is known as resonance. If the Bode plot were to be plotted, we would see that there is a maximum at the natural frequency with a second order pole present there.

3.3 Coupled Spring Problem

There are two second order differential equations in two variables of the form -

$$\frac{d^2x}{dt^2} + (x - y) = 0$$

$$\frac{d^2y}{dt^2} + 2(y - x) = 0$$

With the given initial condition as x(0) = 1, we substitute for y in the second equation from the first and get a fourth order differential equation in terms of x. Simplifying this and substituting to find the y equation, we get -

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

and

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

Taking the Inverse Laplace Transform of these two expressions, we plot the time domain expressions for x(t) and y(t).

```
#Coupled spring displacements. Using the manually calculated vales of X(s)
#and Y(s)
V = sp.lti([1,0,2],[1,0,3,0])
t,v=sp.impulse(V,None,np.linspace(0,20,201))
W = sp.lti([2],[1,0,3,0])
t,w=sp.impulse(W,None,np.linspace(0,20,201))

plt.figure(4)
plt.plot(t,v, label='Horizontal x(t)')
plt.plot(t,w,label='Vertical y(t)')
plt.title("Displacement as a function of time")
plt.grid('True')
plt.xlabel("t")
plt.ylabel("Displacement")
plt.legend()
plt.show()
```

It can be clearly seen that the amplitude of y is greater than x. The phase of the two direction displacement are opposite. The offsets are the same for both the expressions. This models two masses attached to the ends of an ideal spring.

3.4 RLC Filter

We now consider the case of an RLC Filter with the transfer function hand calculated and obtained as follows -

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

With all initial conditions zero, the input is of the form

$$x(t) = cos(10^3 t) + cos(10^6 t)$$

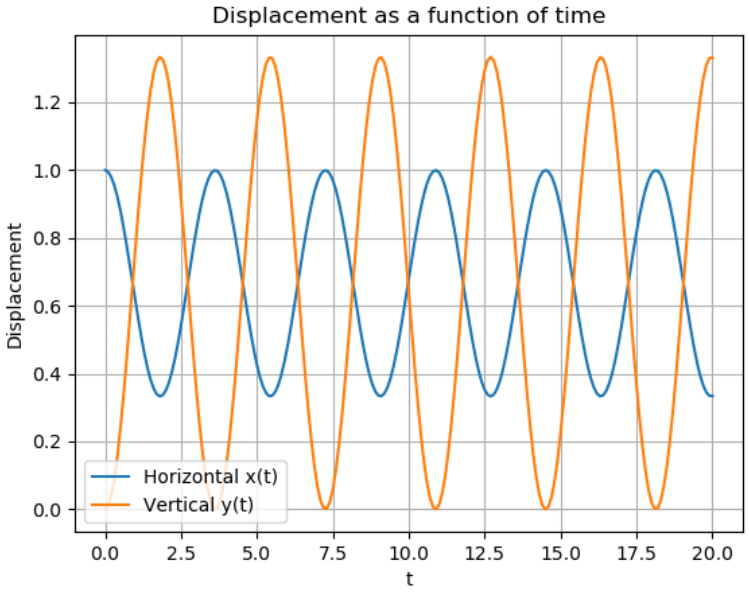

Figure 4:

which is basically the superposition of two sinusoids with low and high frequencies. First, we plot the Bode magnitude and phase plots of the transfer function. Then we use *sp.lsim* to find the output of the filter to the input system. We plot the output from 0 to 30µs as well as from 0 to 10ms.

```
#Transfer function of the given RLC circuit
plt.figure(5)
Hc=sp.lti([10**12],[1,10**8,10**12])
w,mod,phi=Hc.bode()
plt.subplot(2,1,1)
plt.title('Bode plot of magnitude')
plt.xlabel("\u03C9")
plt.ylabel("|H(j\u03A9|")
plt.semilogx(w,mod)
plt.subplot(2,1,2)
plt.title('Bode plot of phase')
plt.xlabel("\u03C9")
plt.ylabel("<H(j\u03C9)")
plt.semilogx(w,phi)
plt.tight_layout()
plt.show()

#Now, given the input sinusoidal input, finding the output by convolving it
```

```
#with the transfer function
plt.figure(6)
plt.subplot(2,1,1)
t=np.arange(0,10e-3,10e-8)
u=np.cos(1e3*t)-np.cos(1e6**t)
t,vo,svec=sp.lsim(Hc,u,t)
plt.plot(t,vo,'b-')
plt.title('Output response till 10ms')
plt.xlabel("Time")
plt.ylabel("Vout")
plt.grid('True')
```

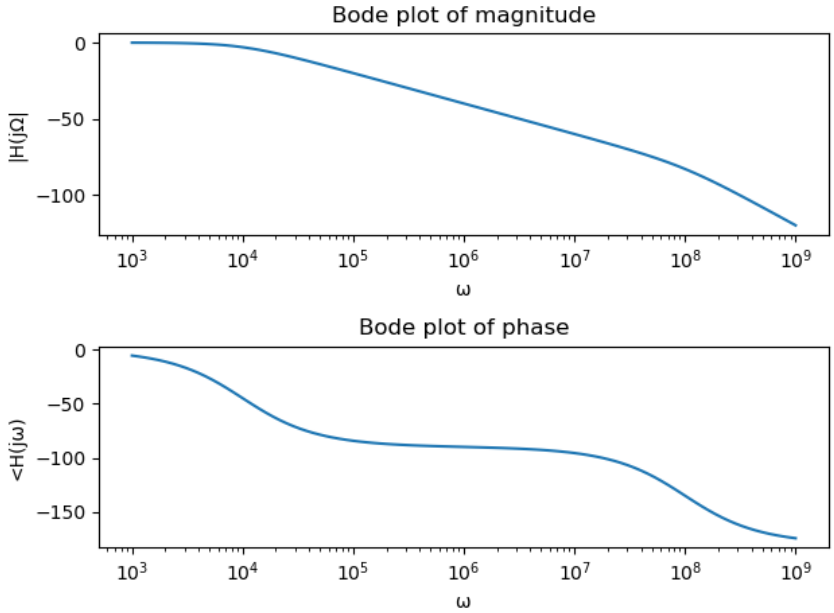

Figure 5:

From the Bode plot, it is clear that the RLC System is a second order low pass filter. The slow time plot shows that the capacitor is charging up to meet the input amplitude. The high frequency component can be seen as a ripple in the slow time plot. This component is highly attenuated and hence not visible in the fast time plot. In the fast time plot, we see that the low frequency component passes almost unchanged, the amplitude is almost 1. The reason is that $\omega = 10^3$ rad/s is well within the 3-dB bandwidth $\omega_{3dB} = 10^4$ rad/s.

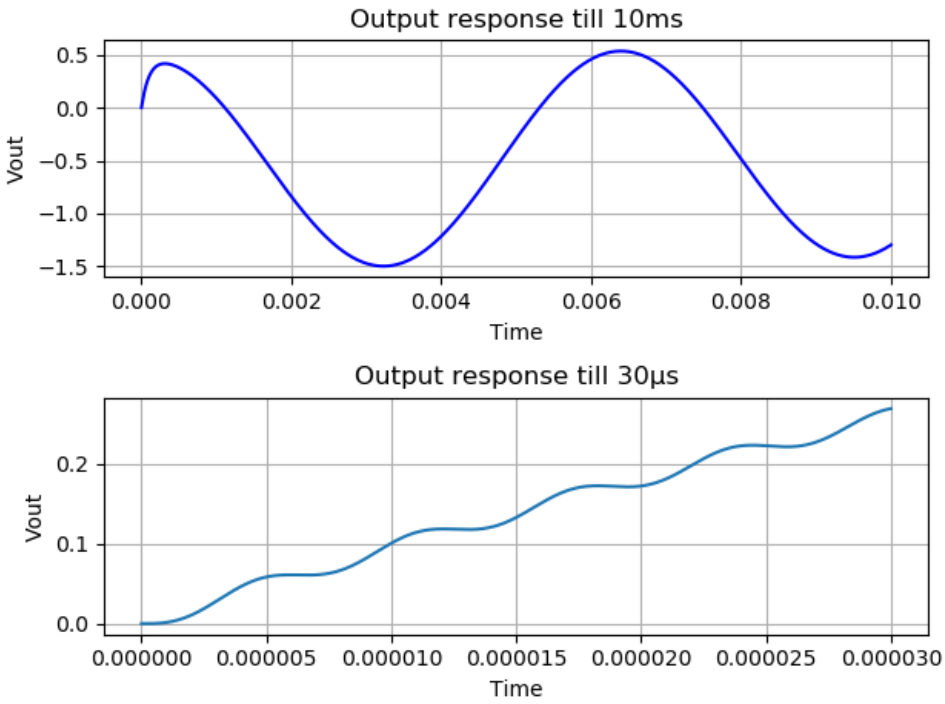

Figure 6:

4 Conclusion

- We can perform signal analysis in the form of Laplace Transforms and their Inversions in the frequency and time domains using Python.
- We saw a low pass filter constructed from an RLC circuit.