Assignment-6 : The Laplace Transform

Name: Tadigoppala Yeswanth
Roll Number: EE19B122

april 2021

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 pass filter .
- To understand how to use the scipy signals toolkit.
- To plot graphs to understand the Laplace Transform.

Single Spring System

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

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

(Initial Conditions are all zero) whose Laplace transform is of the form

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

The input signal is of the form $f(t) = \cos(\omega t)\exp(-at)u(t)$, 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 these function these using numpy polynomials and multiply to get the output laplace transform. Finally we take the inverse laplace transform of the function using sp.impulse to get the time domain sequences and we plot these. We do this for ω=1.5 (natural frequency of the system), and decay of 0.5 and 0.05.
The python code snippet for this is shown below

```
# Q.1(a=0.5)
p11 = poly1d([1,0.5])
p21 = polymul([1,1,2.5],[1,0,2.25])
X1 = sp.lti(p11,p21)
t1,x1 = sp.impulse(X1,None,linspace(0,50,500))

# plot of x(t) vs t for Q.1
figure(0)
plot(t1,x1)
title("The solution x(t) for Q.1")
xlabel(r'$t\rightarrow$')
ylabel(r'$x(t)\rightarrow$')
grid(True)

# Q.2(a= 0.05)
p12 = poly1d([1,0.05])
p22 = polymul([1,0.1,2.2525],[1,0,2.25])
X2 = sp.lti(p12,p22)
t2,x2 = sp.impulse(X2,None,linspace(0,50,500))

# Plot of x(t) vs t for Q.2
figure(1)
plot(t2,x2)
title("The solution x(t) for Q.2")
xlabel(r'$t\rightarrow$')
ylabel(r'$x(t)\rightarrow$')
grid(True)
```

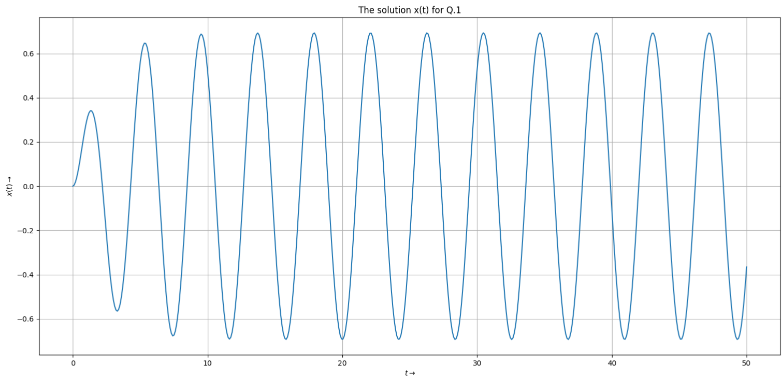

Figure 1: x(t) for a decay of 0.5

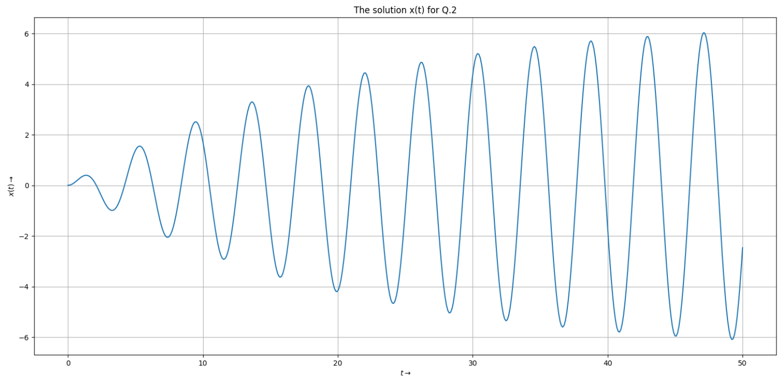

Figure 2: x(t) for a decay of 0.05

We observe that the osciallation amplitude settles to a fixed value in both the cases. We observe it takes longer to settle with decay being less. We also observe that the amplitude increases to a much larger amount in the case with lower decay. At zero (or negative) decay the amplitude increases to infinity, and at high decay it reaches the max amplitude almost instantaneously.

Varying the frequency cosine in the input

We vary the frequency of the cosine and see what affect it has on the output. The following python code snippet is used to find x(t) and plot it for various frequencies:

```
# Q.3
H = sp.lti([1],[1,0,2.25])
for w in arange(1.4,1.6,0.05):
t = linspace(0,50,500)
f = cos(w*t)*exp(-0.05*t)
t,x,svec = sp.lsim(H,f,t)

# plot of x(t) for various frequencies vs time in Q.3
figure(2)
plot(t,x,label='w = ' + str(w))
title("x(t) for different frequencies")
xlabel(r'$t\rightarrow$')
ylabel(r'$x(t)\rightarrow$')
legend(loc = 'upper left', prop={'size':10})
grid(True)
```

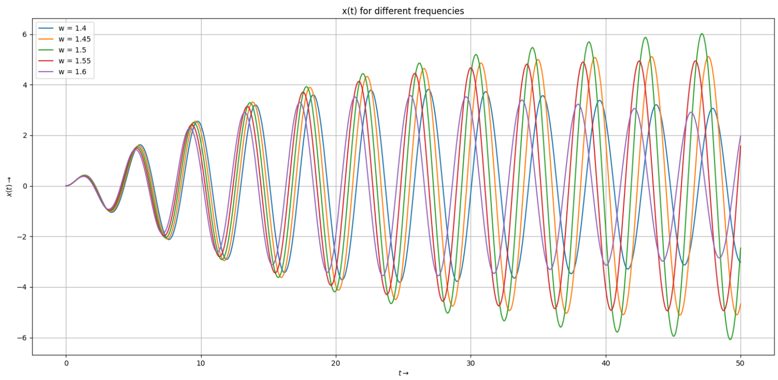

Figure 3: x(t) for different frequencies

When the input frequency is at the natural frequency(1.5) the output amplitude is maximum. In the other cases the output amplitude decreases. This phenomenon is known as resonance.

Coupled Spring Problem

In this problem we have two differential equations and two variables to solve for. The equations are

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

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

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

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

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

We can take the Inverse laplace transform of these two expressions to find the time domain expressions for x(t) and y(t).
The python code snippet used for this is

```
# Q.4
t4 = linspace(0,20,500)
X4 = sp.lti([1,0,2],[1,0,3,0])
Y4 = sp.lti([2],[1,0,3,0])
t4,x4 = sp.impulse(X4,None,t4)
t4,y4 = sp.impulse(Y4,None,t4)

# plots of x(t) and y(t) vs t for Q.4
figure(3)
plot(t4,x4,label='x(t)')
plot(t4,y4,label='y(t)')
title("x(t) and y(t)")
xlabel(r'$t\rightarrow$')
ylabel(r'$functions\rightarrow$')
legend(loc = 'upper right')
grid(True)
```

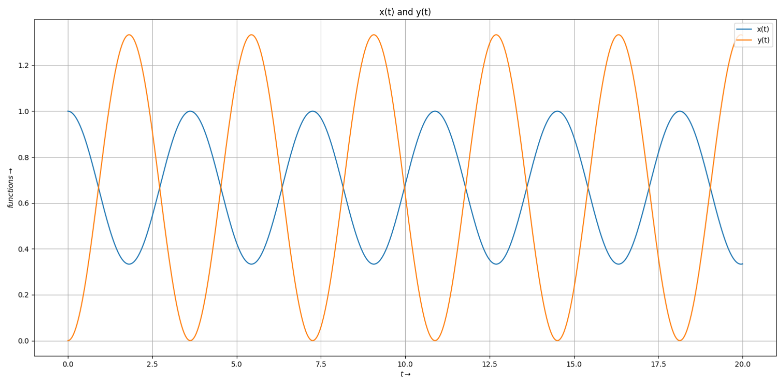

Figure 4: x(t) and y(t)

We observe that the amplitude of y is greater than x. The phase of the two are opposite. The offsets are the same for both the expressions. This models two masses attached to the ends of an ideal spring.

RLC Filter

We now consider the case of an RLC Filter with the transfer function as shown.

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

(Initial Conditions are all zero) The input is of the form

$$x(t) = \cos(10^3 t) - \cos(10^6 t)$$

which is basically the superposition of two sinusoids with low and high frequencies. First we plot the bode plot 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 25ms.

```
# Q.5
denom = poly1d([1e-12,1e-4,1])
H5 = sp.lti([1],denom)
w,S,phi = H5.bode()

# magnitude bode plot for Q.5
figure(4)
subplot(2,1,1)
semilogx(w,S)
title("Magnitude plot")
ylabel(r'$20\log|H(j\omega)|\rightarrow$')
grid(True)
# phase bode plot for Q.5
subplot(2,1,2)
semilogx(w,phi)
title("Phase plot")
xlabel(r'$\omega\rightarrow$')
ylabel(r'$\angle H(j\omega)\rightarrow$')
grid(True)

# Q.6
t6 = arange(0,25e-3,1e-7)
vin = cos(1e3*t6) - cos(1e6*t6)
t6,vo,svec = sp.lsim(H5,vin,t6)
```

```
# The plot of Vo(t) vs t for large time interval.
figure(5)
plot(t6,vo)
title("Long term response")
xlabel(r'$t\rightarrow$')
ylabel(r'$V_o(t)\rightarrow$')
grid(True)

# The plot of Vo(t) vs t for small time interval.
figure(6)
plot(t6[0:300],vo[0:300])
title("Short term response")
xlabel(r'$t\rightarrow$')
ylabel(r'$V_o(t)\rightarrow$')
grid(True)
```

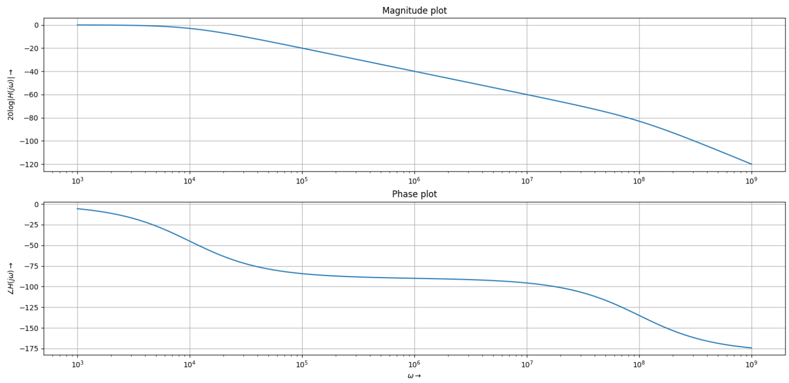

Figure 5: Bode plot

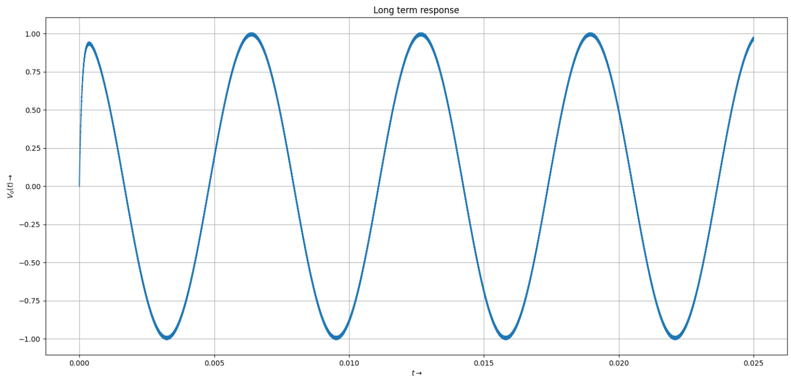

Figure 6: Long term response

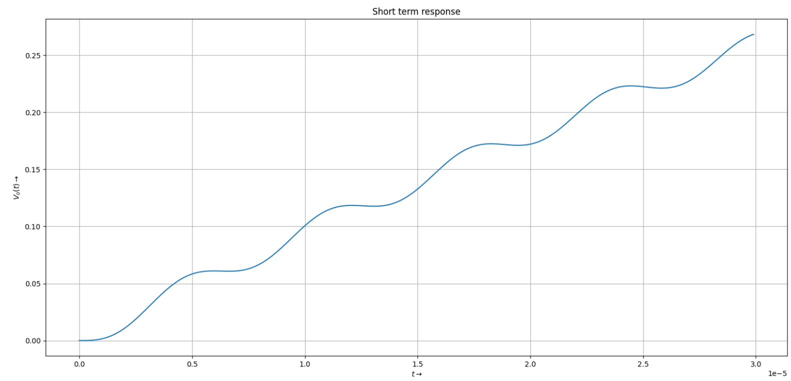

Figure 7: Short term response

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 the $\omega = 10^3 \frac{rad}{s}$ is well within the 3-dB bandwidth ($\omega_{3dB} = 10^4 \frac{rad}{s}$) of the system. Clearly this reiterates the fact that this is a low pass filter with bandwidth $\omega_{3dB} = 10^4 \frac{rad}{s}$.

1 Conclusions

- We analyzed LTI Systems using Laplace Transform.

- We saw a low pass filter constructed from an RLC circuit.
- We used the scipy signals toolkit to calculate the time domain response and the Bode Plot.
- We plotted graphs to understand the above