EE2703 Applied Programming Lab Week 6

Shailesh Pupalwar,EE20B100

Mar 27, 2022

1 Introduction

In this assignment, we will look at how to analyze “Linear Time-invariant Systems” using the scipy.signal library in Python .We limit our analysis to systems with rational polynomial transfer functions. More specifically we consider 3 systems: A forced oscillatory system, A coupled system of Differential Equations and an RLC low pass filter

1.1 Assignment Questions

1.1.1 Question 1

We first consider the forced oscillatory system(with 0 initial conditions):

$$\ddot{x} + 2.25x = f(t) \tag{1}$$

We solve for $X(s)$ using the following equation, derived from the above equation.

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

We then use the impulse response of $X(s)$ to get its inverse Laplace transform.

```
# Time response of spring with decay 0.5

ply11 = poly1d([1,0.5])
ply21 = polymul([1,1,2.5],[1,0,2.25])
X1 = sp.lti(ply11,ply21) # X(s)
t1,x1 = sp.impulse(X1,None,linspace(0,50,500)) # Calculate x(t) using impulse function


# The plot x(t) vs t for decay 0.5
figure(0)
plot(t1,x1)
title("The solution x(t) for Q.1")
xlabel(r'$t\rightarrow$')
ylabel(r'$x(t)\rightarrow$')
grid(True)
```

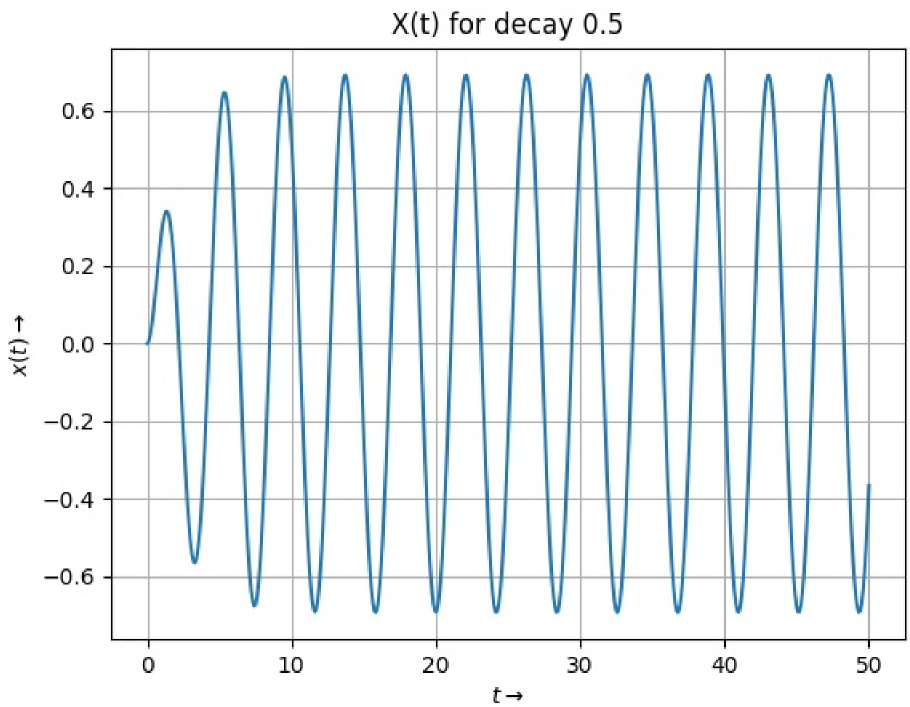

Figure 1: System Response with Decay = 0.5

1.1.2 Question 2

We now see what happens with a smaller Decay Constant.

```
# The plot x(t) vs t for decay 0.05
ply12 = poly1d([1,0.05]) # numerator
ply22 = polymul([1,0.1,2.2525],[1,0,2.25]) # denominator
X2 = sp.lti(ply12,ply22)
t2,x2 = sp.impulse(X2,None,linspace(0,50,500)) # x(t) for deacy = 0.05


# The plot x(t) vs t for decay 0.05
figure(1)
plot(t2,x2)
title("The solution x(t) for Q.2")
xlabel(r'$t\rightarrow$')
ylabel(r'$x(t)\rightarrow$')
grid(True)
```

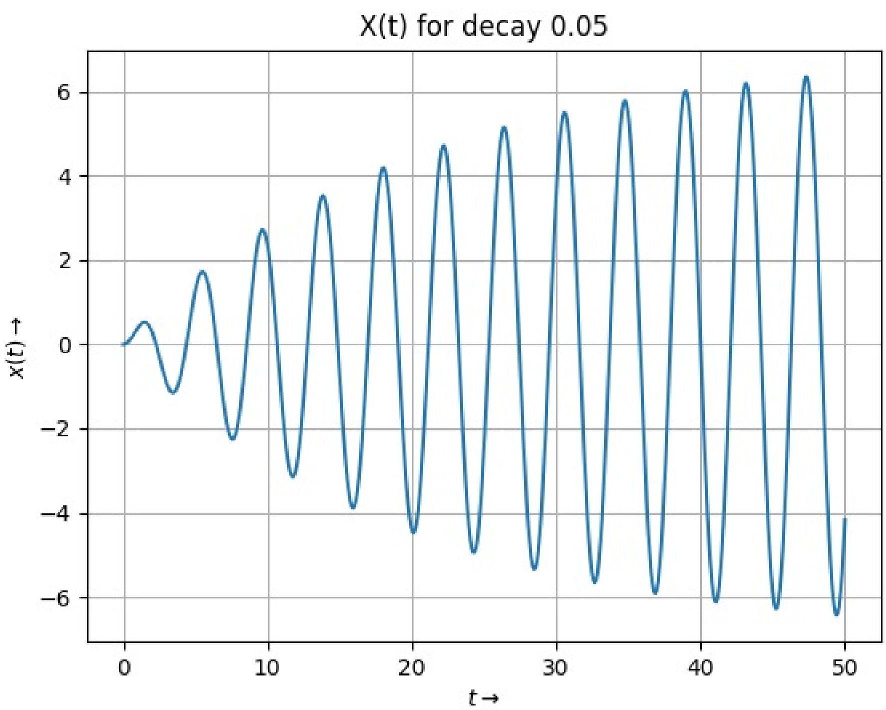

Figure 2: System Response with Decay = 0.05

We notice that the result is very similar to that of question 1, except with a different amplitude. This is because the system takes longer to reach a steady state.

1.1.3 Question 3

We now see what happens when we vary the frequency. We note the the amplitude is maximum at frequency = 1.5, which is the natural frequency of the given system

```
# Transfer function and responces by varying frequencies from 1.4 to 1.6
H = sp.lti([1],[1,0,2.25])
for w in arange(1.4,1.6,0.05): # w = frequency and range is [1.4, 1.6]
        t = linspace(0,50,500)
        f = cos(w*t)*exp(-0.05*t)
        t,x,svec = sp.lsim(H,f,t) # Calculate y(t) from x(t) and H(s)

# The plot of x(t) for various frequencies vs time.
    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')
        grid(True)
```

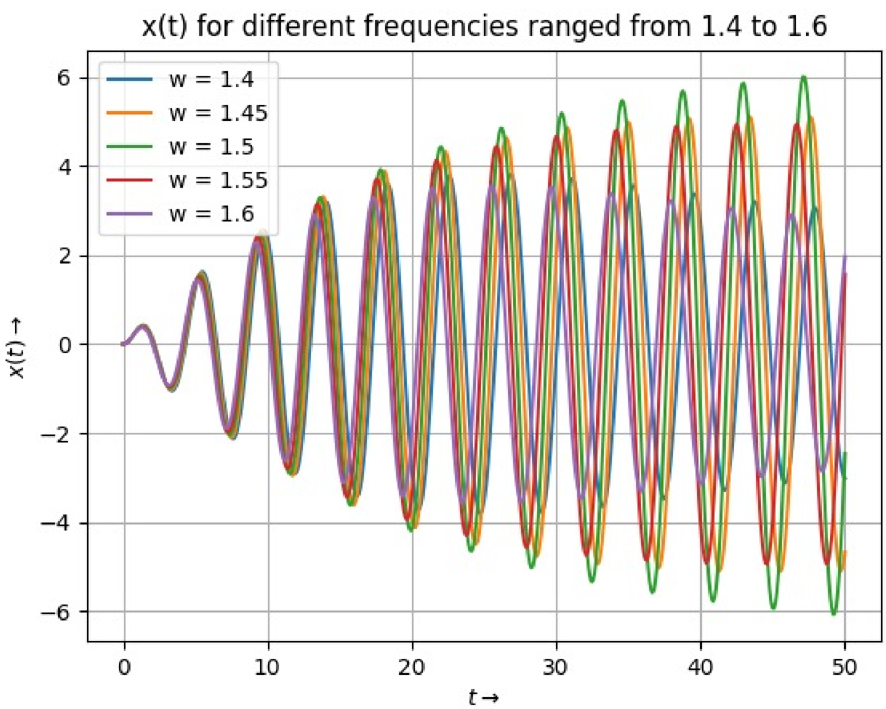

Figure 3: System Response with frequency ranging from 1.4 to 1.6

1.1.4 Question 4

We now consider a coupled Differential system

$$\ddot{x} + (x - y) = 0 \tag{3}$$

and

$$\ddot{y} + 2(y - x) = 0 \tag{4}$$

with the initial conditions: $\dot{x}(0) = 0, \dot{y}(0) = 0, x(0) = 1, y(0) = 0$. Taking Laplace Transform and solving for $X(s)$ and $Y(s)$, We get:

$$X(s) = \frac{s^2 + 2}{s^3 + 3s} \tag{5}$$

$$Y(s) = \frac{2}{s^3 + 3s} \tag{6}$$

```
# The python code snippet for Q.4: Coupled spring problem
t4 = linspace(0,20,500)
X4 = sp.lti([1,0,2],[1,0,3,0]) # X(s)
Y4 = sp.lti([2],[1,0,3,0])      # Y(s)
t4,x4 = sp.impulse(X4,None,t4) # Calculate x(t)
t4,y4 = sp.impulse(Y4,None,t4) # Calculate y(t)

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)
```

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

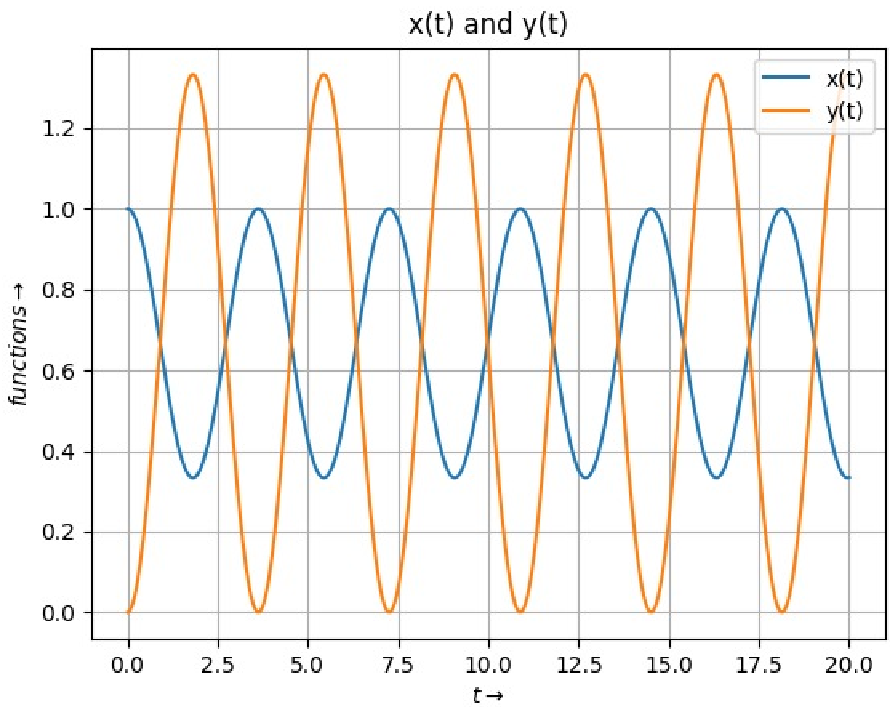

Figure 4: Coupled Oscillations

1.1.5 Question 5

Now we try to create the bode plots for the low pass filter defined in the question

```
# The python code snippet for Q.5: RLC circuit
temp = poly1d([1e-12,1e-4,1])
H5 = sp.lti([1],temp) # H(s)
w,S,phi = H5.bode() # Bode plot of transfer function


# The magnitude bode plot for Q.5
figure(4)
semilogx(w,S)
title("Magnitude Bode plot")
xlabel(r'$\omega\rightarrow$')
ylabel(r'$20\log|H(j\omega)|\rightarrow$')
grid(True)

# The phase bode plot for Q.5
figure(5)
semilogx(w,phi)
title("Phase Bode plot")
xlabel(r'$\omega\rightarrow$')
ylabel(r'$\angle H(j\omega)\rightarrow$')
grid(True)
```

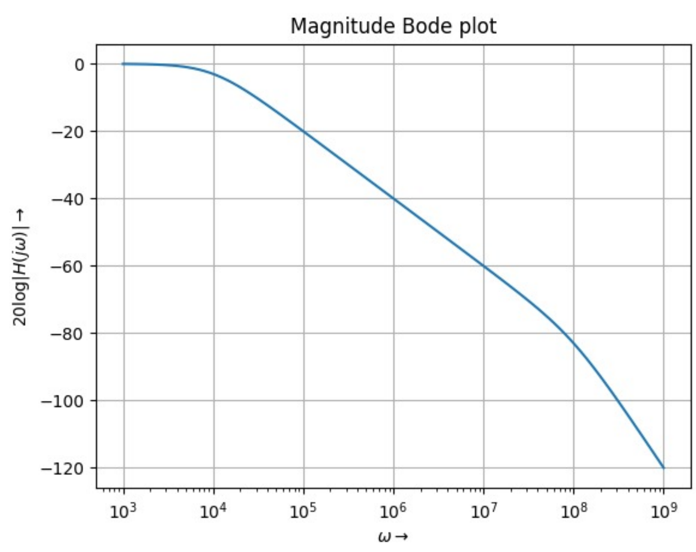

Figure 5: Magnitude Bode plot

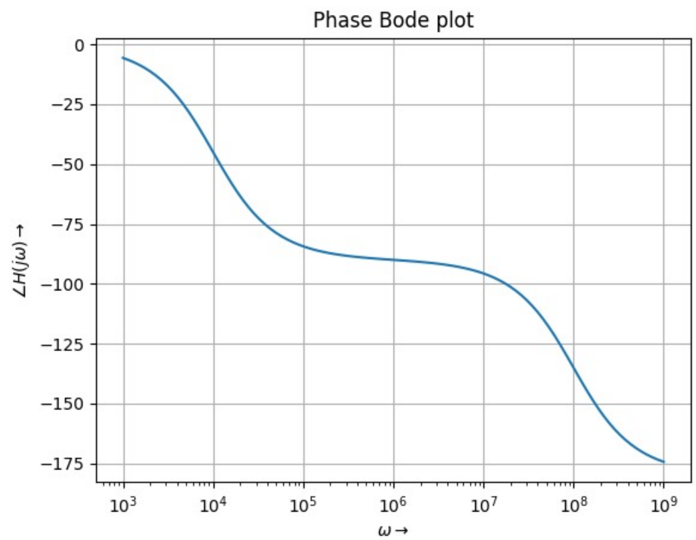

Figure 6: Phase Bode plot

1.1.6 Question 6

We know plot the response of the low pass filter to the input:

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

for $0 < t < 30\mu s$ and $0 < t < 30ms$

```
# The python code snippet for Q.6: Output for given input in RLC circuit
t6 = arange(0,25e-3,1e-7)
vi = cos(1e3*t6) - cos(1e6*t6) # Input function f(t)
t6,vo,svec = sp.lsim(H5,vi,t6) # Output voltage Vo(t)

# The plot of Vo(t) vs t for large time interval.
figure(6)
plot(t6,vo)
title("The Output Voltage for large time interval")
xlabel(r'$t\rightarrow$')
ylabel(r'$V_o(t)\rightarrow$')
grid(True)

# The plot of Vo(t) vs t for small time interval.
figure(7)
plot(t6[0:300],vo[0:300])
title("The Output Voltage for small time interval")
xlabel(r'$t\rightarrow$')
ylabel(r'$V_o(t)\rightarrow$')
grid(True)
```

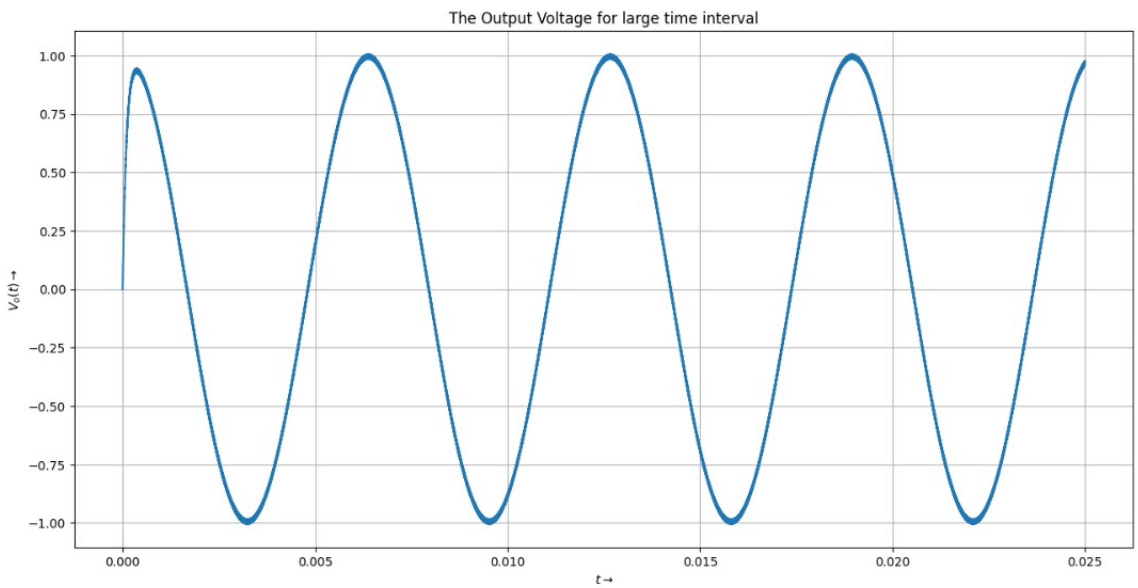

Figure 7: The output voltage for large time interval

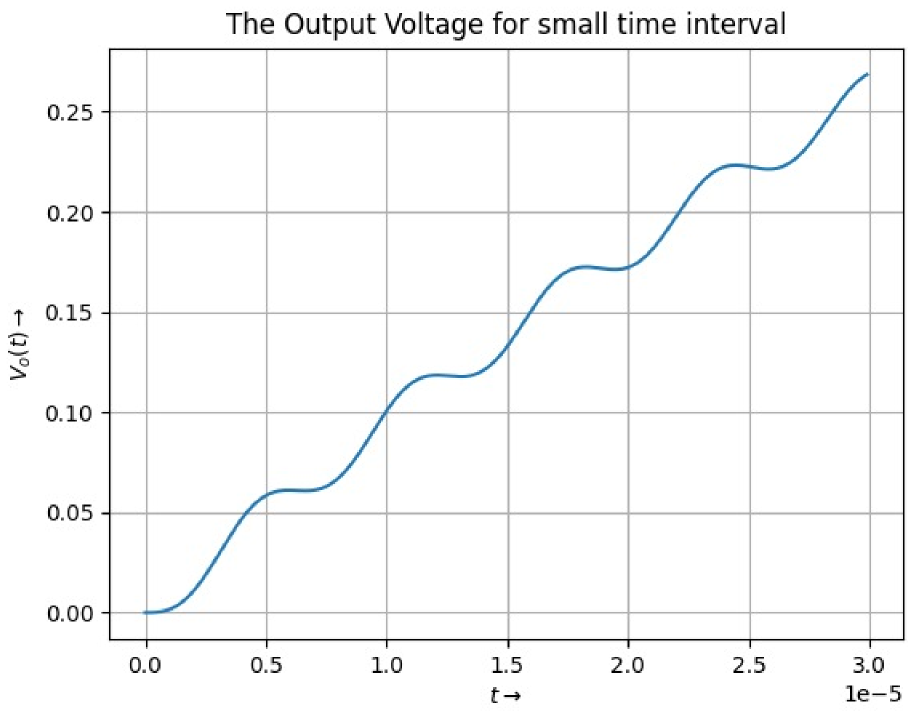

Figure 8: The output voltage for small time interval

2 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