Assignment 6(b) - The Laplace Transform

Jahnavi Pragada

May 12, 2021

Introduction

In this assignment, we will look at how to analyze “Linear Time-invariant Systems” using the scipy.signal library in Python .All the problems are in continuous time and use Laplace Transforms. We considered systems: A forced oscillatory system, A coupled system of Differential Equations and an RLC low pass filter

Question 1,2

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.We found for two different decay constants 0.5,0.05

```
def func(decay, w):
    Xn = np.poly1d([1, decay])
    Xd = np.polymul([1, 0, 2.25], [1, 2*decay, (w**2 + decay**2)])
    Xs = sp.lti(Xn, Xd)
    t, x = sp.impulse(Xs, None, np.linspace(0, 50, 500))
    return Xs, t, x

X, t1, x1 = func(0.5, 1.5)
X, t2, x2 = func(0.05, 1.5)

plt.figure(1)
plt.plot(t1, x1,'r')
plt.title(r"$x(t)$ with decay=0.5")
```

```
plt.xlabel(r"$t \to $")
plt.ylabel(r"$x(t) \to $")
plt.show()

plt.figure(2)
plt.plot(t2, x2,'r')
plt.title(r"$x(t)$ with decay=0.05")
plt.xlabel(r"$t \to $")
plt.ylabel(r"$x(t) \to $")
plt.show()
```

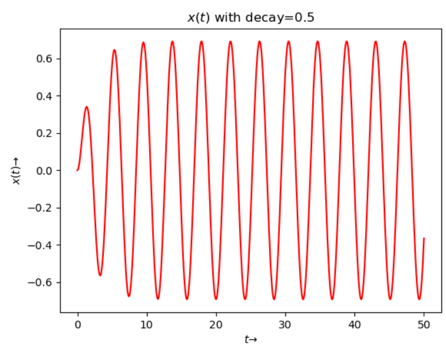

Figure 1: System Response with Decay = 0.5

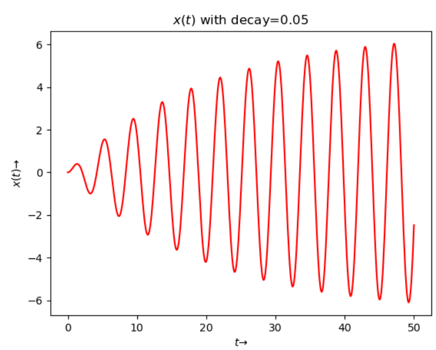

Figure 2: System Response with Decay = 0.05

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

Question 3

Considering the system with $f(t)$ input and $x(t)$ output.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

```
H = sp.lti([1],[1,0,2.25])
freq=np.linspace(1.4,1.6,5)
for w in freq:
t = np.linspace(0,50,500)
f = np.cos(w*t)*np.exp(-0.05*t)
t,x,svec = sp.lsim(H,f,t)

plt.figure(3)
plt.plot(t,x,label='w = ' + str(w))
plt.title("x(t) for different frequencies")
plt.xlabel(r'$t\rightarrow$')
plt.ylabel(r'$x(t)\rightarrow$')
plt.legend(loc = 'upper left')
plt.show()
```

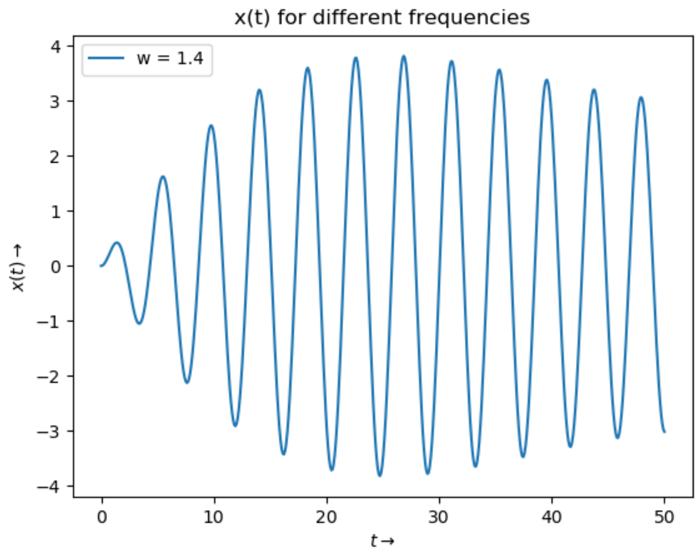

Figure 3: System Response with frequency = 1.4

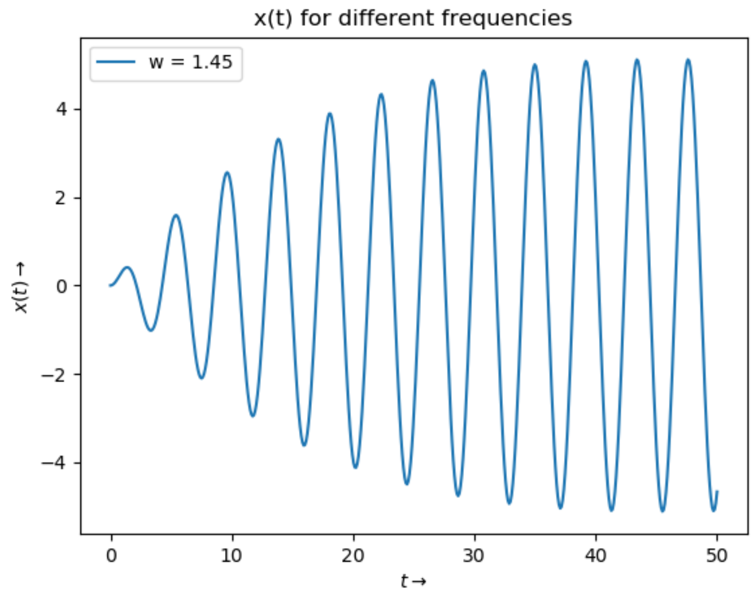

Figure 4: System Response with frequency = 1.45

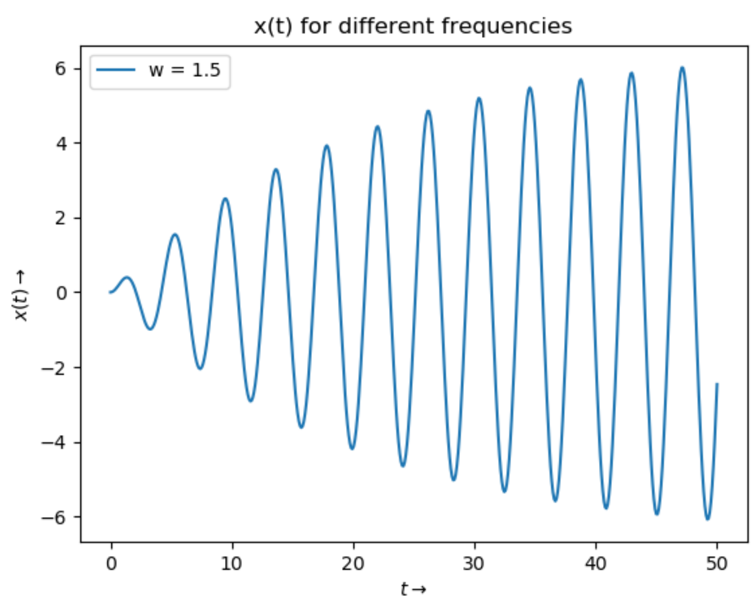

Figure 5: System Response with frequency = 1.5

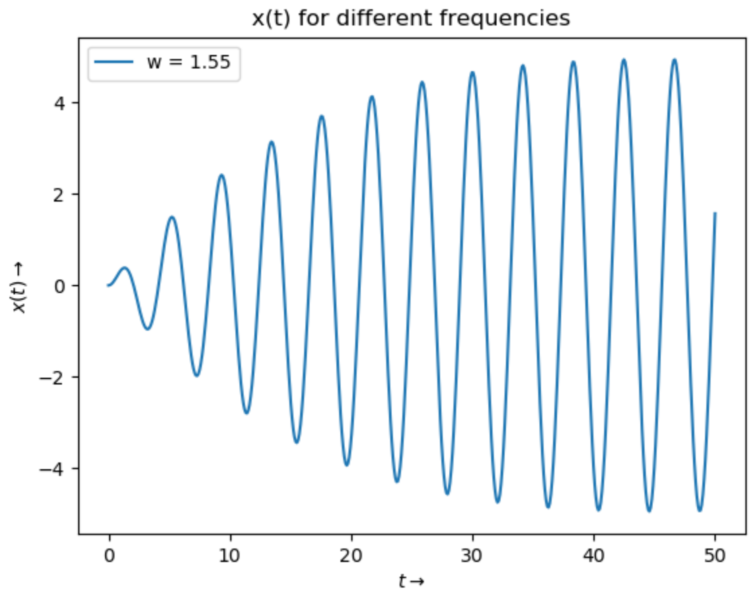

Figure 6: System Response with frequency = 1.55

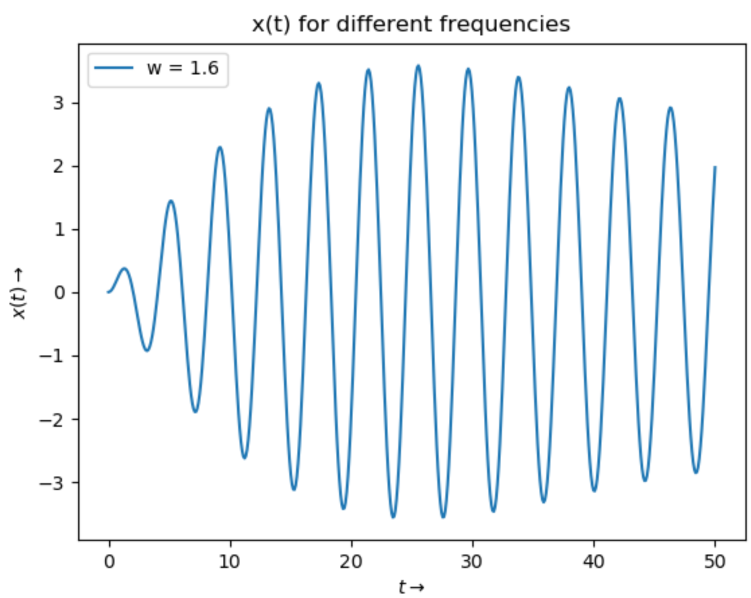

Figure 7: System Response with frequency = 1.6

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:

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

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

```
#solve for X in coupling equation
t4 = np.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)

plt.figure(4)
plt.plot(t4, x4,'r')
plt.title(r"Time evolution of $x(t)$ for Coupled spring system")
plt.xlabel(r"$t \to $")
plt.ylabel(r"$x(t) \to $")
plt.show()

plt.figure(5)
plt.plot(t4, y4,'r')
plt.title(r"Time evolution of $y(t)$ for Coupled spring system ")
plt.xlabel(r"$t \to $")
plt.ylabel(r"$y(t) \to $")
plt.show()
```

We notice that the outputs of this system are 2 sinusoids which are out of phase.The amplitude of y is greater than that of x.

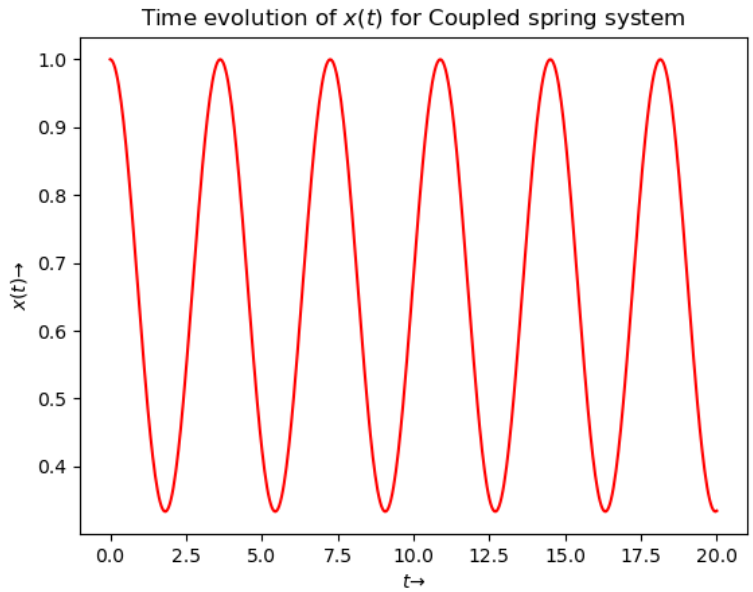

Figure 8: Coupled Oscillations

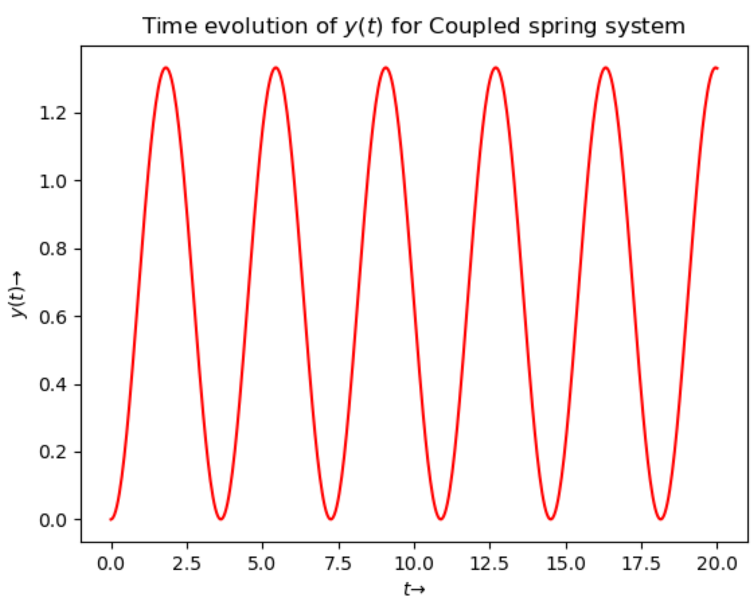

Figure 9: Coupled Oscillations

Question 5

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

```
def RLC_transfunc(R, C, L):
    Hnum = np.poly1d([1])
    Hden = np.poly1d([L*C, R*C, 1])
    Hs = sp.lti(Hnum, Hden)
    w, mag, phi = Hs.bode()
    return w, mag, phi, Hs

w, mag, phi, H = RLC_transfunc(100, 1e-6, 1e-6)

# plot Magnitude Response
plt.figure(6)
plt.semilogx(w, mag)
plt.title(r" Magnitude Response of $H(jw)$ of Series RLC network")
plt.xlabel(r"$ w \to $")
plt.ylabel(r"$ 20\log|H(jw)|  \to $")
plt.show()

# Plot of phase response
plt.figure(7)
plt.semilogx(w, phi, 'r', label="$Phase Response$")
plt.title(r"Phase response of the $H(jw)$ of Series RLC network")
plt.xlabel(r"$  w \to $")
plt.ylabel(r"$ \angle H(j\omega)$ $\to $")
plt.show()
```

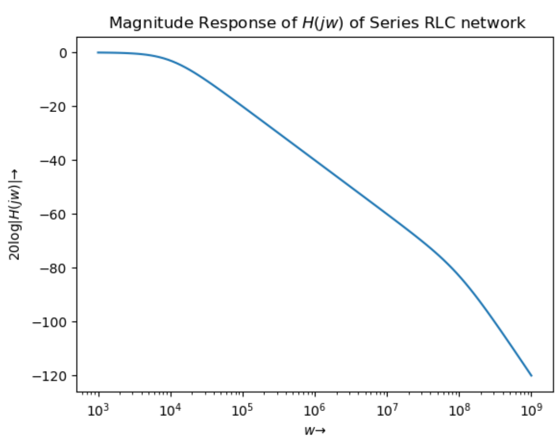

Figure 10: Magnitude Plot For RLC filter

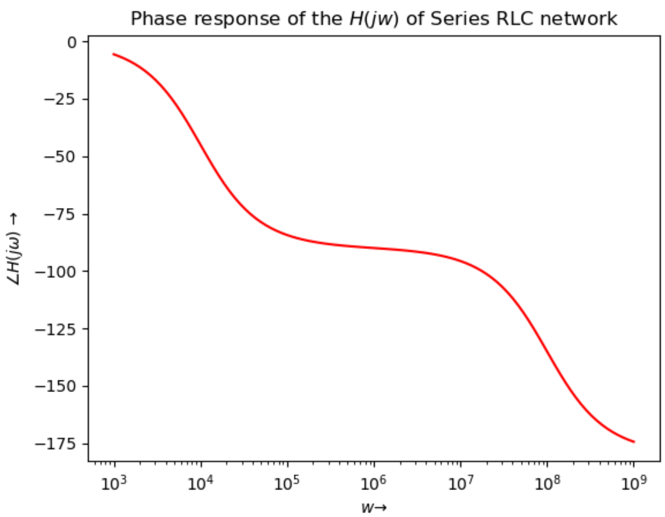

Figure 11: Phase Plot For RLC filter

Question 6

We now plot the response of the low pass filter to the input for different time variations:

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

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

```
t = np.arange(0, 90*pow(10, -3), pow(10, -6))
vi = np.cos(t*pow(10, 3))-np.cos(t*pow(10, 6))
t, vo, svec = sp.lsim(H, vi, t)

# plot of Vo(t) for large time i.e at steady state
# Long term response
plt.figure(8)
plt.plot(t, vo, 'r')
plt.title(r"Output voltage $v_0(t)$ at Steady State")
plt.xlabel(r"$ t \to $")
plt.ylabel(r"$ y(t) \to $")
plt.show()

# Plot of Vo(t) for 0<t<30usec
plt.figure(9)
plt.plot(t, vo, 'r')
```

```
plt.title(r"Output voltage $v_0(t)$ for $0<t<30\mu sec$")
plt.xlim(0, 30*(10**(-6)))
plt.ylim(-1e-5, 0.3)
plt.xlabel(r"$ t \to $")
plt.ylabel(r"$ v_0(t) \to $")
plt.show()
```

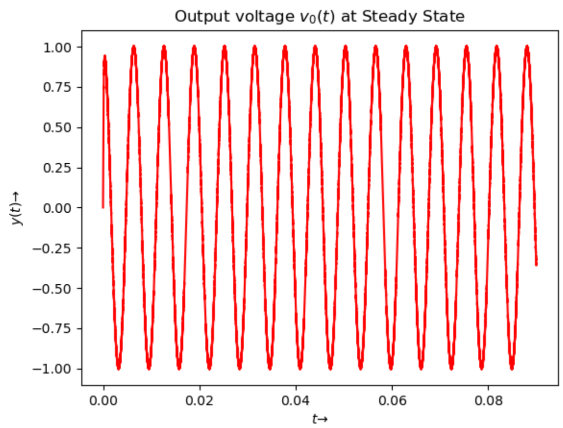

Figure 12: System response for t¡30us

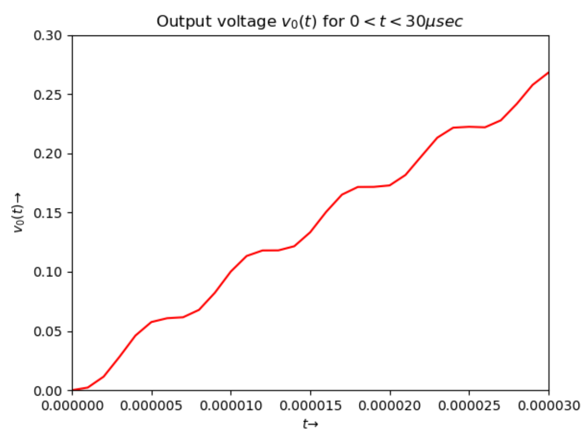

Figure 13: System response for t¡30ms

Conclusion

In this assignment, we have used scipy's signal processing library to analyze some LTI systems. In the first system we noticed that oscillation amplitude settles to afixed value in both cases but it takes longer to settle if the decay is less.And we also observed that amplitude is maximum when input frequency is equal to natural frequency.