Assignment 5: Applied Programming Lab

Aman Singhal, EP17B002

5 March 2020

1 Abstract

In this assignment, we are analysing "Linear Time-invariant Systems" using python's scipy module.

2 My Code

```
from numpy import *
import matplotlib.pyplot as plt
import scipy.signal as sp

# f(t)
def damping(t,f,a=1):
    return cos(f*t)* exp(-t*a/2)*heaviside(t,2)

#values for t for Q1,Q2,Q3
t1 = linspace(0,50,501)


#Q1
function1 = ([1.0,0.5],[1.0,1.0,4.75,2.25,5.625])
x,y = sp.impulse(function1,None,t1)
plt.plot(x,y)
plt.xlabel(r't',size=15)
plt.ylabel(r'X(t)',size=15)
plt.title(r'plot system response vs t for damping coefficient = -0.5 ')
plt.show()


#Q2
function2 = ([1.0,0.05],[1.0,0.1,4.5025,0.225,5.068])
x,y = sp.impulse(function2,None,t1)
plt.plot(x,y)
plt.xlabel(r't',size=15)
plt.ylabel(r'X(t)',size=15)
plt.title(r'plot for system response vs t for damping coefficient = -0.05 '
    )
plt.show()


#Q3
```

```python
for i in arange(1.4,1.6,0.05):
    function3 = ([1.0,0.05],[1.0,0.1,2.25+i**2+0.05**2,0.1*2.25,2.25*(0.05**2+i**2)])
    x,y,svec = sp.lsim(function3,damping(t1,i),linspace(0,100,501))
    plt.plot(x,y)
    plt.plot(x,y)
    plt.xlabel(r't',size=15)
    plt.ylabel(r'X(t)',size=15)
    plt.title(r'plot for system response vs t for i = {}'.format(i))
    plt.show()


#Q4
t2 = linspace(0,20,201)

system_x = ([1,0,2,0],[1,0,3,0,0])
t2,x = sp.impulse(system_x,None,t2)
plt.plot(t2,x)
plt.xlabel(r't',size=15)
plt.ylabel(r'X(t)',size=15)
plt.title(r'plot for coupled system X vs t')
plt.show()

system_y = ([2],[1,0,3,0])
t,y = sp.impulse(system_y,None,t2)
plt.plot(t,y)
plt.xlabel(r't',size=15)
plt.ylabel(r'Y(t)',size=15)
plt.title(r'plot for coupled system Y vs t')
plt.show()


#Q5
H = sp.lti([1],[10**(-12),10**(-4),1])
w,S,phi = H.bode()

plt.semilogx(w,S)
plt.xlabel(r'frequency',size=15)
plt.ylabel(r'magnitude(Db)',size=15)
plt.title(r'magnitude plot')
plt.show()

plt.semilogx(w,phi)
plt.xlabel(r'frequency',size=15)
plt.ylabel(r'phase',size=15)
plt.title(r'phase plot')
plt.show()

t3 = arange(0,10**(-2),10**(-6))
#Q6
function6 = ([1],[10**(-12),10**(-4),1])
t3,x,svec = sp.lsim(function6,damping(t3,10**3,0)-damping(t3,10**6,0),t3)
plt.plot(t3,x)
plt.xlabel(r't',size=15)
plt.ylabel(r'V(t)',size=15)
plt.title(r'plot of voltage at the output for given input')
plt.show()
```

3 Plots

3.1 Question 1

Solving a differential equation and finding X(t) for a damped input

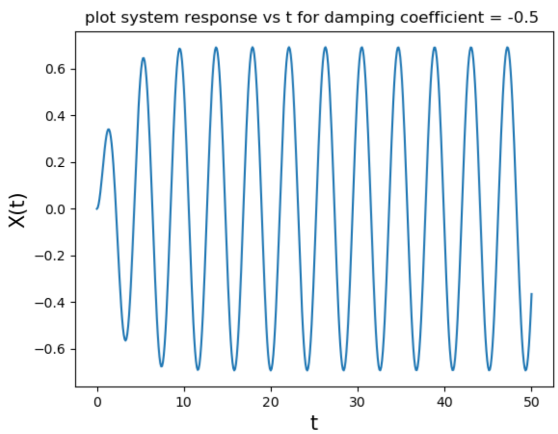


Figure 1

3.2 Question 2

Solving the same differential equation and finding X(t) for a lightly damped input

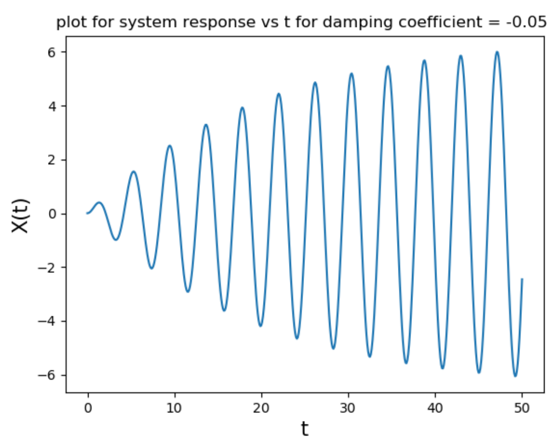
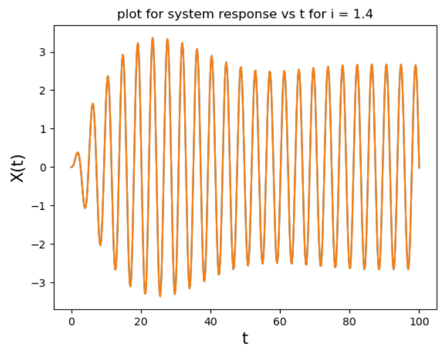
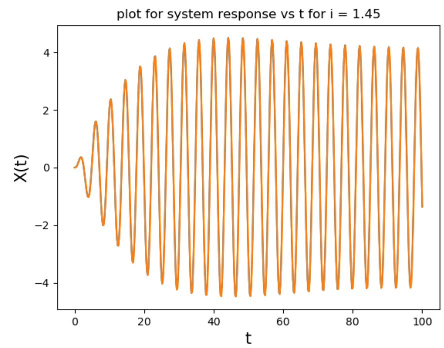


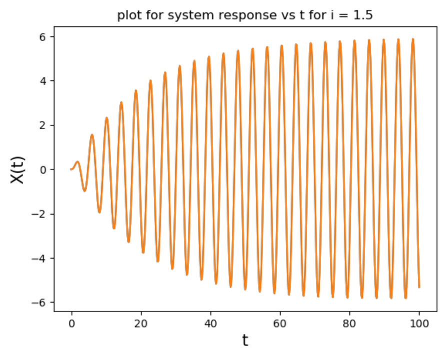
Figure 2

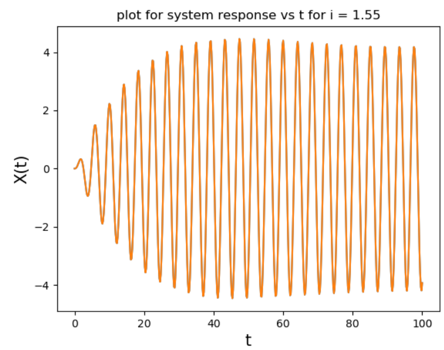
3.3 Question 3

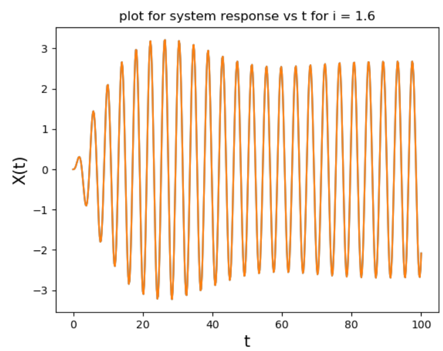
Using convolution to solve the the above problem for different input frequencies.





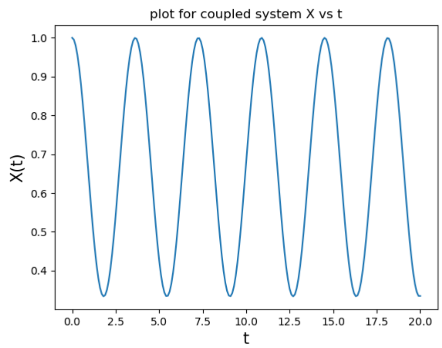


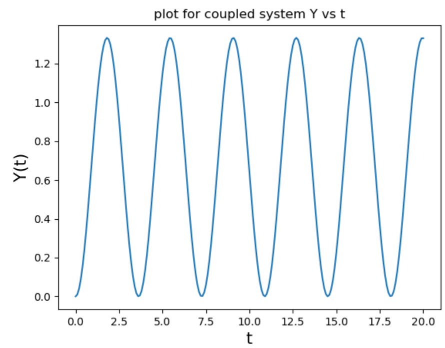




3.4 Question 4

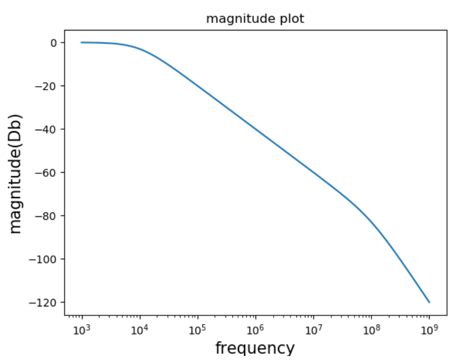
In this question we have to solve a coupled spring system.

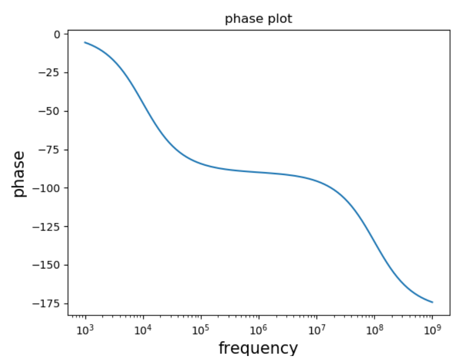




3.5 Question 5

In this question we are given an RLC circuit system. We have to find the transfer function and get the Bode plot.





3.6 Question 6

In this question we are given the same RLC circuit system. We have to give an input voltage to and plot the output voltage. We have the transfer function of the system we have to multiply it with the laplace transform of the input function and take the laplace inverse of the product.

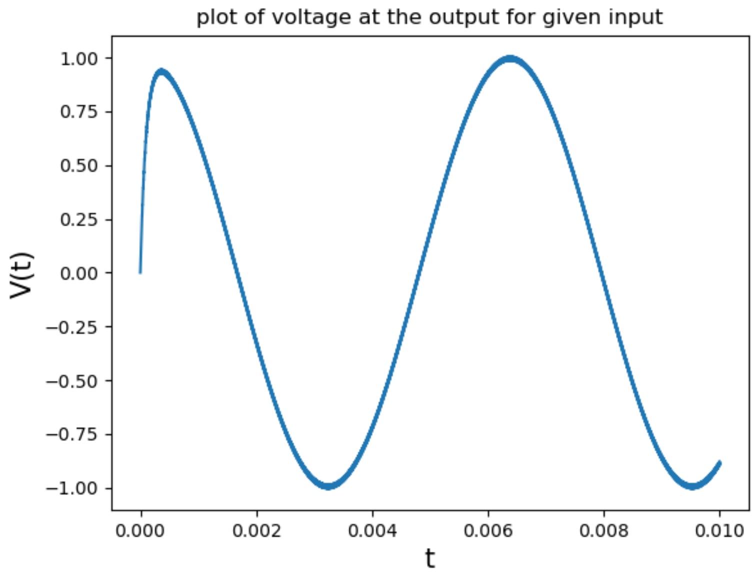


Figure 2

4 Conclusion

1) In questions 1 and 2 we observe a periodic but stable function with complex conjugate poles and a negative zero.

2) In question 3 we get largest response closest to w = 1.5

3) In question 4 we plotted the coupled system separately. They had the same frequency but the opposite phase.

4) In question 5 and 6 we observe that there are two negative poles and it behaves like a low pass filter.