Assignment 6: The Laplace Transform

Manognya Param Koolath

March 12, 2019

The task

We analyse "Linear Time-Invariant Systems" with numerical tools in python. We use the Signals toolbox of python for this..

Question 1

We have to solve for the time response of a spring satisfying

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

with $x(0) = 0$ and $\frac{dx}{dt} = 0$ for t going from 0 to 50 seconds where $f(t)$ is defined as follows:

$$f(t) = cos(1.5t)\exp -0.5tu(t)$$

To do this, we first calculate the transfer function $H(s)$ of this system as:

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

We obtain $h(t)$ using the `sp.impulse` function. Then we convolve $h(t)$ with $f(t)$ to get the required solution. The plot of x v/s t is as shown in Figure 1.

Question 2

We have to solve the same problem given above with $f(t)$ as follows:

$$f(t) = cos(1.5t)\exp -0.05tu(t)$$

We do this by following the same procedure as what we did for Question 1. The plot of x v/s t is as shown in Figure 2.

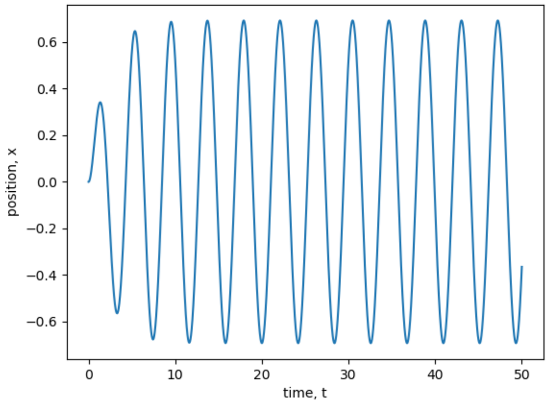

Figure 1: x v/s t in Question 1

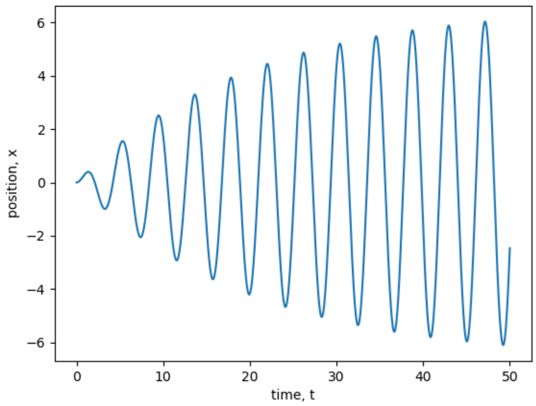

Figure 2: x v/s t in Question 2

Question 3

We have to solve the same problem as in Question 1 with $f(t)$ as follows, with w varying from 1.4 to 1.6 in steps of 0.05.

$$f(t) = cos(wt)\exp -0.05tu(t)$$

We do this by following the same procedure as what we did for Question 1. The plot of x v/s t for different frequencies is shown in Figure 3.

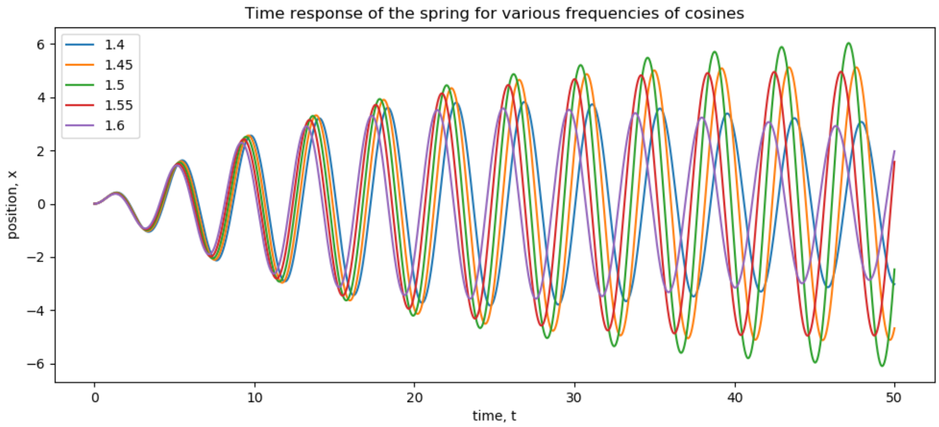

Figure 3: x v/s t for different frequencies (Question 3)

Question 4

We are given a coupled spring system governed by the following equations:

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

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

where the initial condition is $x(0) = 1$, $\frac{dx}{dt}(0) = \frac{dy}{dt}(0) = y(0) = 0$. To solve this, from the two equations using the initial conditions, we found $X(s)$ and $Y(s)$ to be the following:

$$X(s) = \frac{s(s^2 + 2)}{(s^2 + 1)(s^2 + 2) - 2}$$

$$Y(s) = \frac{2s}{(s^2 + 1)(s^2 + 2) - 2}$$

From these, we get the time domain functions for $x(t)$ and $y(t)$ using `sp.impulse` function. These are plotted in Figure 4.

Question 5

We are given a circuit for which we have to plot the bode plots. From the circuit, we get the transfer function as follows:

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{10^6}{10^{-6}s^2 + 100s + 10^6}$$

From this, we have plotted the bode plots in Figure 5.

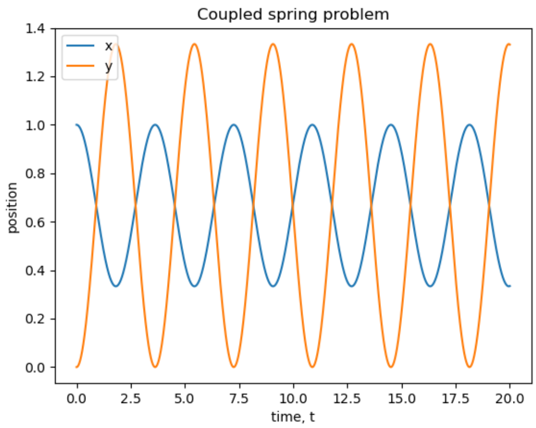

Figure 4: Plots of x and y vs t for the coupled spring problem as explained in Question 4

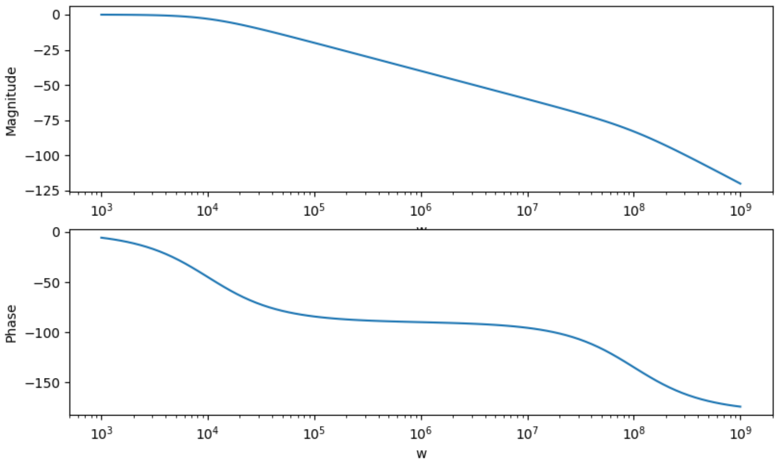

Figure 5: Bode plot for the circuit in Question 5.

Question 6

We are given the input signal for the system defined by the transfer function from Question 5 to be:

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

From this, we obtain the output signal by convolving $h(t)$ and $v_i(t)$ using `sp.lsim` function. We have plotted the plot of $v_o(t)$ v/s t in Figures 6 and 7. Figure 6 shows the plot till a few miliseconds range while Figure 7

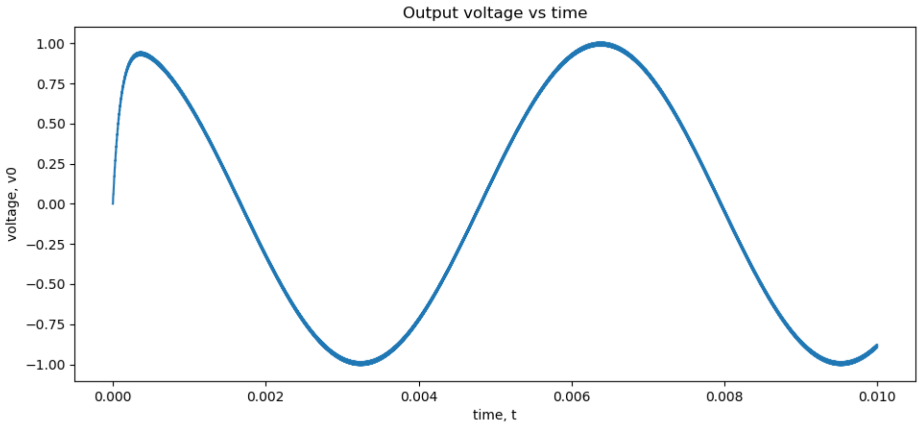

Figure 6: $v_o(t)$ v/s t (Question 6)

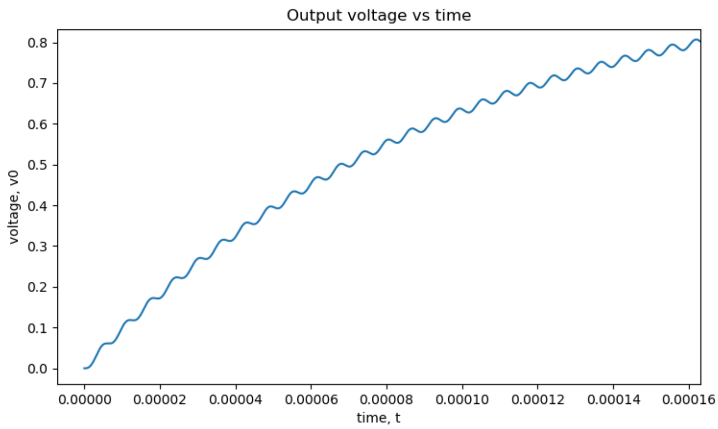

Figure 7: $v_o(t)$ v/s t (Question 6)

shows a few microseconds with high precision. Initially, due to the sudden impulsive input, we get the initial output which looks different from the rest for $0 < t < 30\mu s$. The long term sinusoid is due to the system attaining a steady state configuration due to the driving sinusoid.

Python code

The code is properly commented and completely vectorised.

Assignment 6 Code

```
#Assignment 6
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sp
from scipy.linalg import lstsq
import scipy.integrate as spint
import mpl_toolkits.mplot3d.axes3d as p3
from scipy import ndimage

def f_t(t,decay,freq):
        ans=np.cos(freq*t)*np.exp((-1*decay)*t)
        ii=np.where(t<0)
        ans[ii]=0
        return ans

def v_t(t):
        ii=np.where(t<0)
        ans=np.cos(1000*t)-np.cos(1e6*t)
        ans[ii]=0
        return ans

#Q1:
num=np.poly1d([1])
den=np.poly1d([1,0,2.25])
H=sp.lti(num,den)
t,h=sp.impulse(H,None,np.linspace(0,50,10001))
# plt.plot(t,h)
# plt.show()
f1=f_t(t,0.5,1.5)
t,y1,svec=sp.lsim(H,f1,t)
plt.plot(t,y1)
# plt.title('Time response of the spring in Q1')
```

```
plt.xlabel('time, t')
plt.ylabel('position, x')
plt.show()

#Q2:
f2=f_t(t,0.05,1.5)
t,y2,svec=sp.lsim(H,f2,t)
plt.plot(t,y2)
# plt.title('Time response of the spring in Q2')
plt.xlabel('time, t')
plt.ylabel('position, x')
plt.show()

#Q3:
rng=np.arange(1.4,1.6,0.05)
for i in rng:
        f=f_t(t,0.05,i)
        t,y,svec=sp.lsim(H,f,t)
        plt.plot(t,y,label=i)
        # plt.show()
plt.legend(loc='upper left')
plt.title('Time response of the spring for various frequencies of cosines
plt.xlabel('time, t')
plt.ylabel('position, x')
plt.show()

#Q4:
num=np.poly1d([1,0,2,0])
den=np.poly1d([1,0,3,0,0])
X=sp.lti(num,den)
t,x=sp.impulse(X,None,np.linspace(0,20,10001))
plt.plot(t,x)
plt.title('Plot of x vs t in the coupled spring problem')
plt.xlabel('time, t')
plt.ylabel('position, x')
plt.show()

num=np.poly1d([2,0])
den=np.poly1d([1,0,3,0,0])
Y=sp.lti(num,den)
t,y=sp.impulse(Y,None,np.linspace(0,20,10001))
plt.plot(t,y)
plt.title('Plot of y vs t in the coupled spring problem')
plt.xlabel('time, t')
```

```
plt.ylabel('position, y')
plt.show()

#Plotting x and y together:
plt.plot(t,x,label='x')
plt.plot(t,y,label='y')
plt.title('Coupled spring problem')
plt.legend(loc='upper left')
plt.xlabel('time, t')
plt.ylabel('position')
plt.show()

# Q5:
num=np.poly1d([1e6])
den=np.poly1d([1e-6,100,1e6])
H=sp.lti(num,den)
w,S,phi=H.bode()
plt.subplot(2,1,1)
# plt.title('Magnitude plot')
plt.semilogx(w,S)
plt.xlabel('w')
plt.ylabel('Magnitude')
plt.subplot(2,1,2)
# plt.title('Phase plot')
plt.semilogx(w,phi)
plt.xlabel('w')
plt.ylabel('Phase')
plt.show()

#Q6:
t,h=sp.impulse(H,None,np.linspace(0,10*1e-3,100001))
v=v_t(t)
t,out,svec=sp.lsim(H,v,t)
plt.plot(t,out)
plt.title('Output voltage vs time')
plt.xlabel('time, t')
plt.ylabel('voltage, v0')
plt.show()
```