EE2703 : Applied Programming Lab
Assignment 6B
Laplace Transformation

Krishna Somasundaram
EE19B147

April 26, 2021

Objective

Most general systems around us can be modelled as **Linear Time-invariant Systems** (a.k.a LTI Systems) and is extensively used in Electrical Engineering. Example: Linear Circuit Analysis and Image processing. In this assignment, we try to attain the following objectives-

- Analysis of continuous time LTI systems in laplace domain numpy.poly functions
- Solve **LCCDE - Linear Constant Coefficient Differential Equations** in laplace domain using the scipy.signal library
- Explore various functions of the above mentioned library like `impulse, bode, lti, lsim`

Section 1

The time response of a lossless spring system is given by

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

where $f(t)$ is the forced input on the spring system. It is given by

$$f(t) = cos(1.5t)e^{-decay*t}u(t)$$

In Laplace domain

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

with $x(0) = 0$ and $\dot{x} = 0$ input conditions. This corresponds to

$$s^2X(s) + 2.25X(s) = F(s)$$

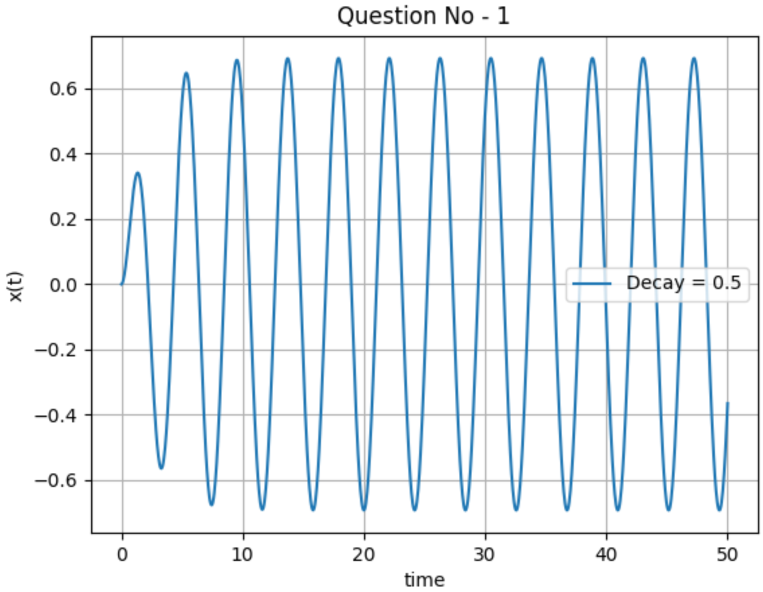
$$H(s) = s^2 + 2.25 = \frac{F(s)}{X(s)}$$

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

Now, we determine $X(s)$ using the `impulse` function

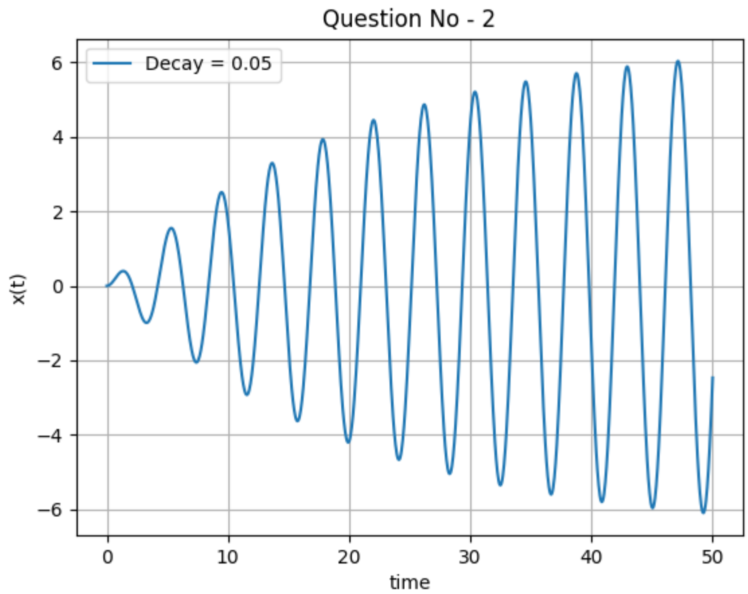
```
#Time vector going from 0 to 50 seconds
t=np.linspace(0,50,1000)
# X(s) using np.poly functions
X = sp.lti([1,decay],np.polymul([1,2*decay,2.25+decay*decay],[1,0,2.25]))
t,x = sp.impulse(X,None,t)

#Time domain function values
t, x = sp.impulse(X, None, t)
```



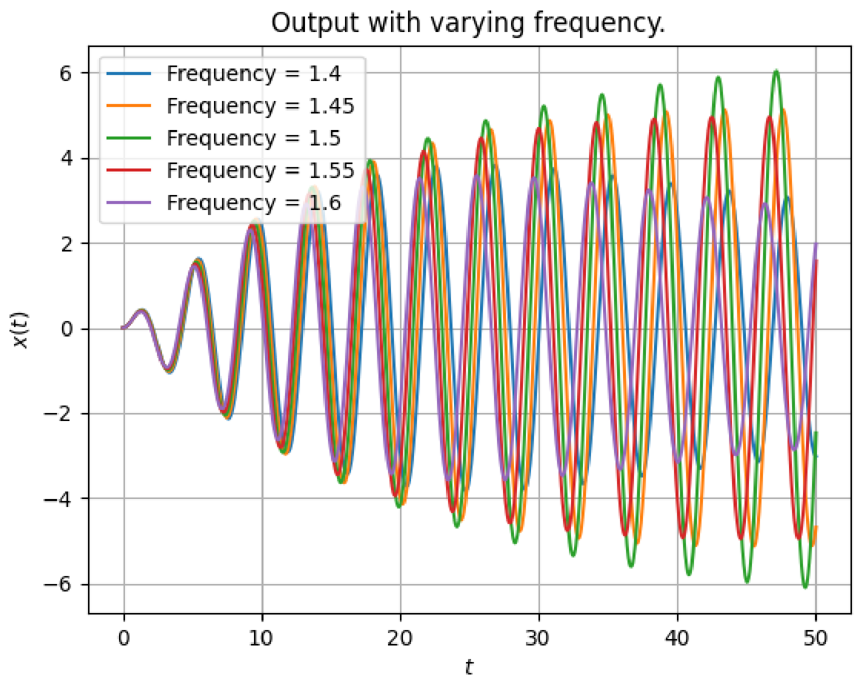
X(t) in time domain for decay coefficient 0.5

We can observe that the output of the system when the decay coefficient = 0.05 (in the next graph) has higher amplitude - since the energy supplied by the forced input in case of decay coefficient = 0.05 will be higher compared to decay coefficient = 0.5 as the former one decays slower than the latter.



X(t) in time domain for decay coefficient 0.05

Now, let us plot the output of the system for a decay of 0.05 at various frequencies ranging from 1.4 to 1.6.



Section 2

We try to solve the following coupled spring problem with:

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

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

with initial conditions $x(0) = 1$, $\dot{x}(0) = y(0) = \dot{y}(0) = 0$. Solving further

$$s^2X(s) - sx(0^-) - \dot{x}(0^-) = Y(s)$$

$$s^2Y(s) - sy(0^-) - \dot{y}(0^-) + 2Y(s) = X(s)$$

Substituting and solving further, we arrive at

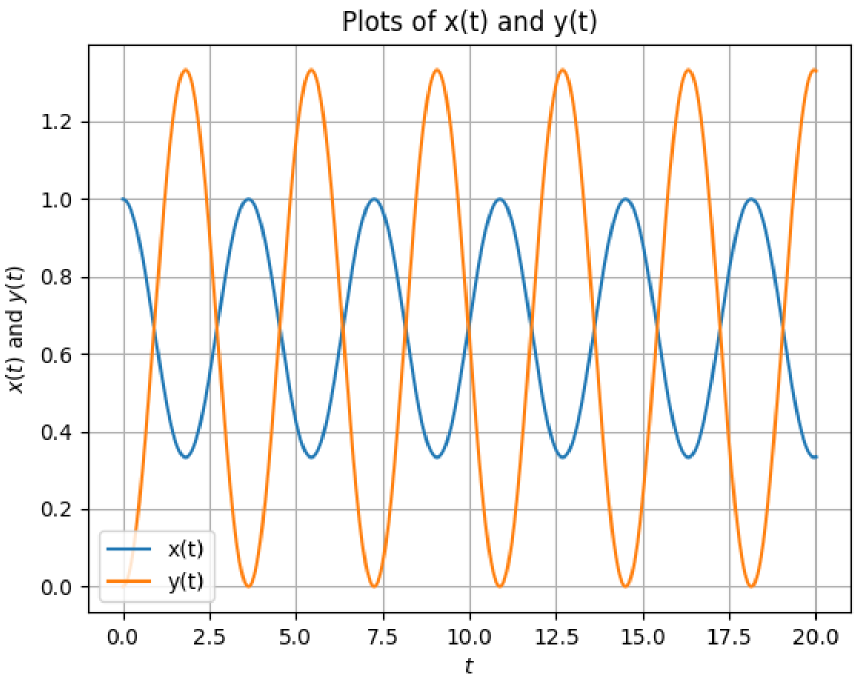
$$X(s) = \frac{(0.5s^3 + s)}{s^4 + 1.5s^2}$$

$$Y(s) = \frac{s}{s^4 + 1.5s^2}$$

```
t = np.linspace(0,20,1000)

X = sp.lti([0.5,0,1,0],[0.5,0,1.5,0,0])
Y = sp.lti([1,0],[0.5,0,1.5,0,0])

t,x = sp.impulse(X,None,t)
t,y = sp.impulse(Y,None,t)
```



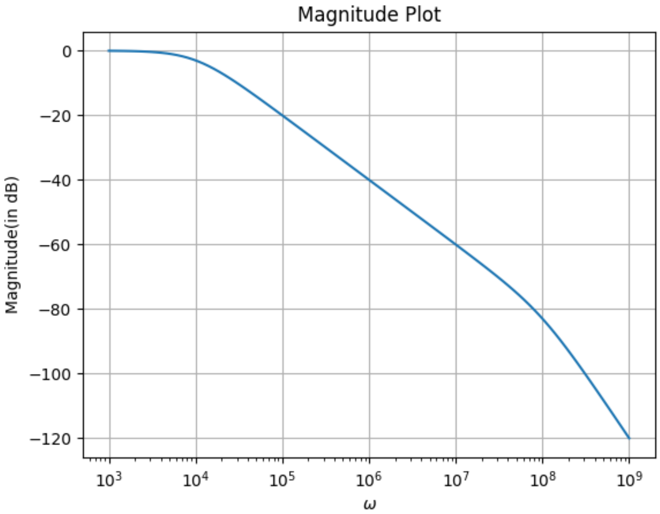
Section 3

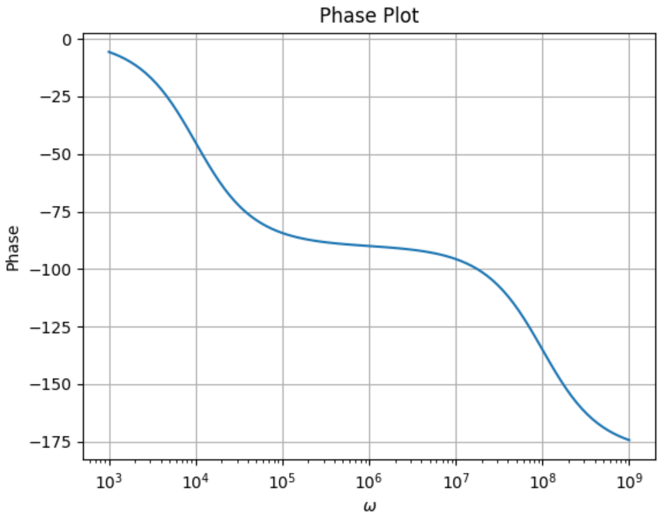
The goal is to obtain the magnitude and phase response of the steady state transfer function of an RLC Circuit.

We can observe that the circuit acts as a voltage divider. Thus it reduces to the following transfer equation

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

```
H = sp.lti([1], [1e-12, 1e-4, 1])
w, S, phi = H.bode()
```





We can observe that there are two major deflections in both graphs indicating the two poles of the system. The magnitude plot remains constant until it sees a pole after which it decreases rapidly at -20 dB/decade. It further sloped down to -40 dB/decade after it encounters the second pole. In the phase plot, we can observe that there is a phase drop of approximately 90 degrees at each pole.

In the given circuit, if the applied input voltage is of form,

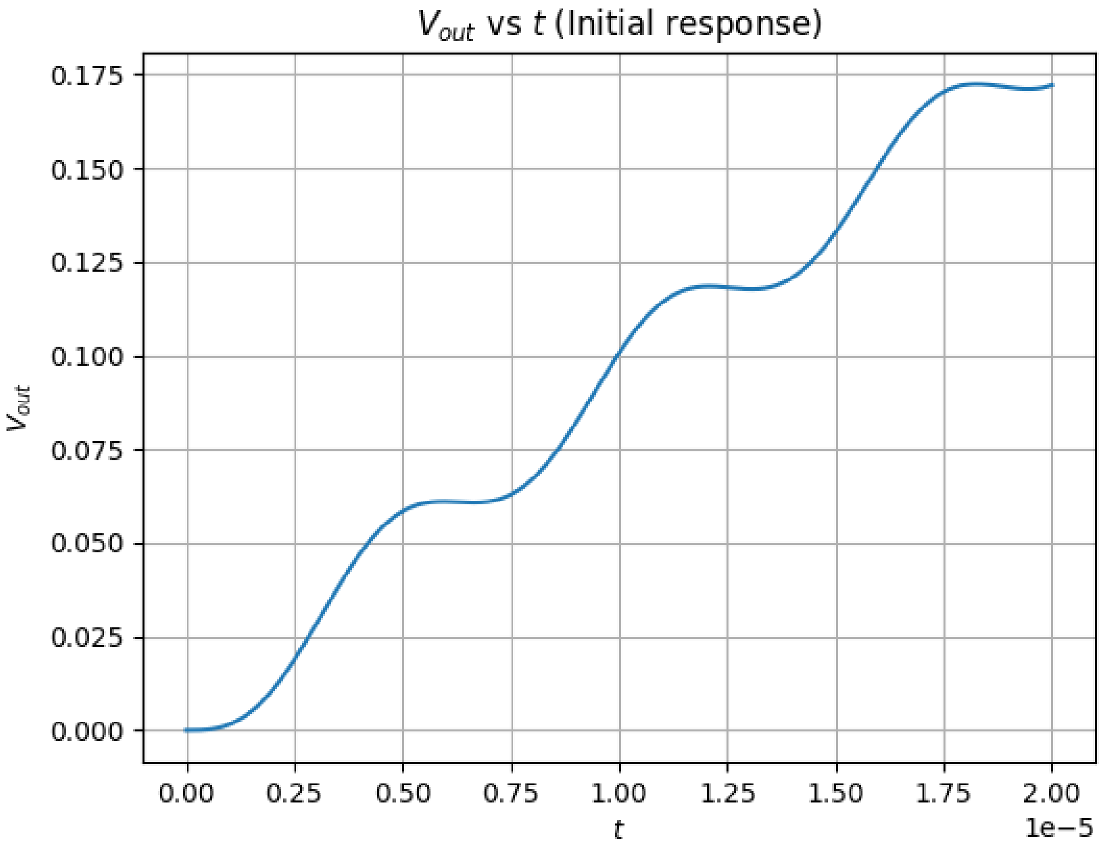
$$V_{in}(t) = cos(10^3 t)u(t) - cos(10^6 t)u(t)$$

We obtain the final output V_{out} of the RLC circuit by

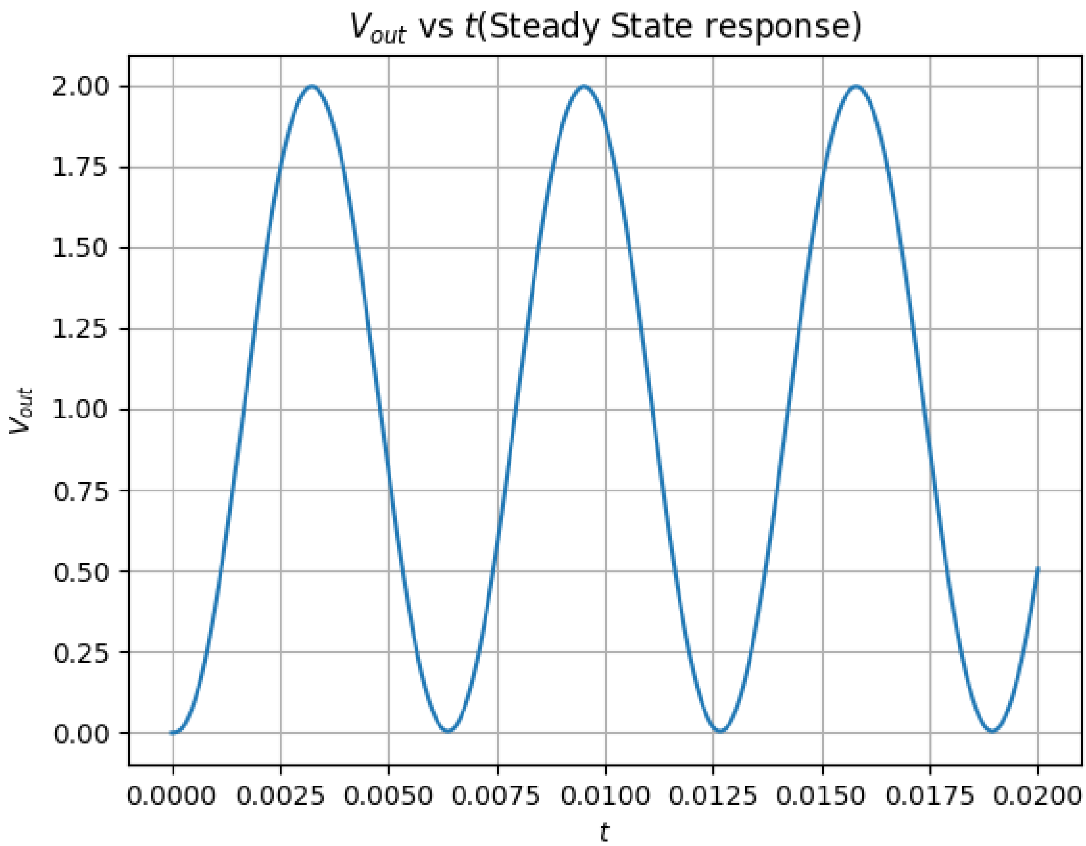
```
#For short term response
t = np.linspace(0, 20e-6, 10000)
```

```
#Input signal
Vin = np.cos(1e3*t)-np.cos(1e6*t)
#Convolution
t, Vout, svec = sp.lsim(H, Vin, t)

#For long term response
t = np.linspace(0, 2e-2, 10000)
#Convolution
t, Vout, svec = sp.lsim(H, Vin, t)
```



The above graph corresponds to the initial transient response of the system. We can observe that the sinusoidal component `cos(1e6t)` is showed up as ups and downs in the graph.



But in steady state, we can observe that the output is primarily composed of 1000 rad/s frequency while the frequency 10^6 rad/s is almost attenuated. This is because the later frequency experiences attenuation of about 100 times as shown in the magnitude plot of the transfer function. Essentially the given circuit acts as a low pass filter supporting frequencies upto 1000 rad/s.

Conclusion

We explored the idea of using laplace transformations to solve equations of various systems like spring system, coupled spring problem and an RLC system using the scipy.signal library in python. We used bode plots to determine and understand the phase and magnitude response of systems. We also saw how an RLC circuit can behave like a low pass filter.