EE2703: Applied Programming Lab

Week 6B: The Laplace Transform

Author: Surya Prasad S, EE19B121

April 18, 2021

1 Abstract:

In this week's assignment we shall analyse Linear Time-Invariant Systems with numerical tools in Python. We shall use SciPy's signal library for this assignment to solve all the differential equations. We shall only analyse rational transfer functions. More specifically, we consider 3 systems: Forced oscillatory system, Coupled System of Differential Equations and an RLC Lowpass Filter

2 The Assignment:

2.1 Question 1:

First system is a Forced oscillatory system whose LTI system is given by the following equation:

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

$f(t)$ is given by

$$f(t) = cos(1.5t)e^{-0.5t}u_0(t)$$

whose $F(s)$ is

$$F(s) = \frac{s+0.5}{(s+0.5)^2+2.25}$$

We solve for $X(s)$ and we get the following equation:

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

We then use SciPy's impulse() to get the corresponding $x(t)$. Code snippet for this question is:

```
# Question 1
## The rate of decay of the input is 0.5
p.figure(0)
p.title("1. Forced Damping Oscillator with decay=0.5")
func(0.5)

## Function to plot the impulse response. Decay takes the value of the rate at which decay
def func(decay):
    ### Frequency response of a spring is being calculated
    f_spring_num = p.poly1d([1, decay])
    f_den = p.poly1d([1, 2 * decay, 2.25 + decay**2])
    diff_coeff = p.poly1d([1, 0, 2.25])
    f_spring_den = p.polymul(f_den, diff_coeff)
```

```
### Converting the coefficients to LTI system type for generating impulse response
f_lti = sp.lti(f_spring_num, f_spring_den)
t, x = sp.impulse(f_lti, None, p.linspace(0, 50, 1000))

### Plotting the response
p.plot(t, x)
p.xlabel('time$\\rightarrow$')
p.ylabel('x$\\rightarrow$')
p.show()
```

The plot of the impulse response is as follows:

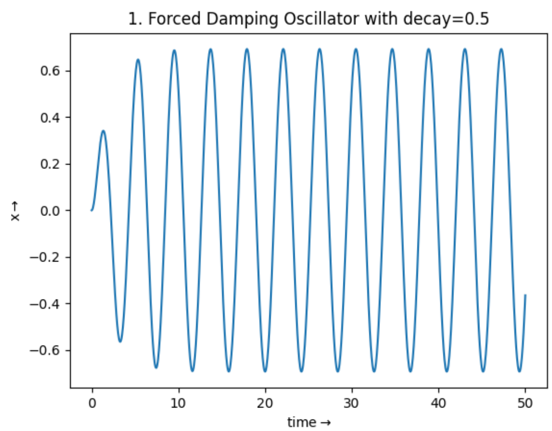


Figure 1: Impulse Response with decay=0.5

2.2 Question 2:

In the previous equation, the rate of decay was 0.5 and now we solve the same problem with a decay rate of 0.05. We shall use the same function used for the previous question and we get the following plot:

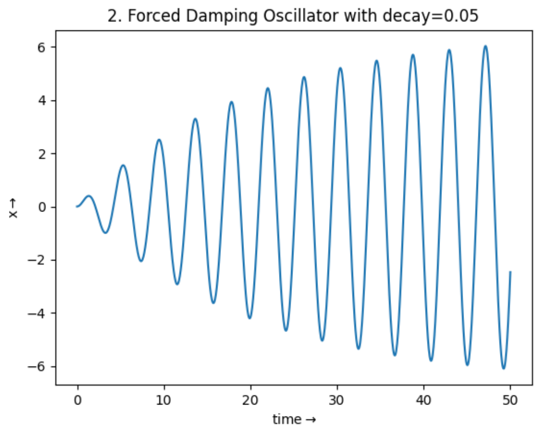


Figure 2: Impulse Response with decay=0.05

As expected, for lesser decay we can see the system taking more time to stabilise.

2.3 Question 3:

Now let's consider the previous problem to be an LTI system where $f(t)$ is the input and $x(t)$ is the output. We shall vary the frequency of the input signal from 1.4 to 1.6 in steps of 0.05 and plot the responses.

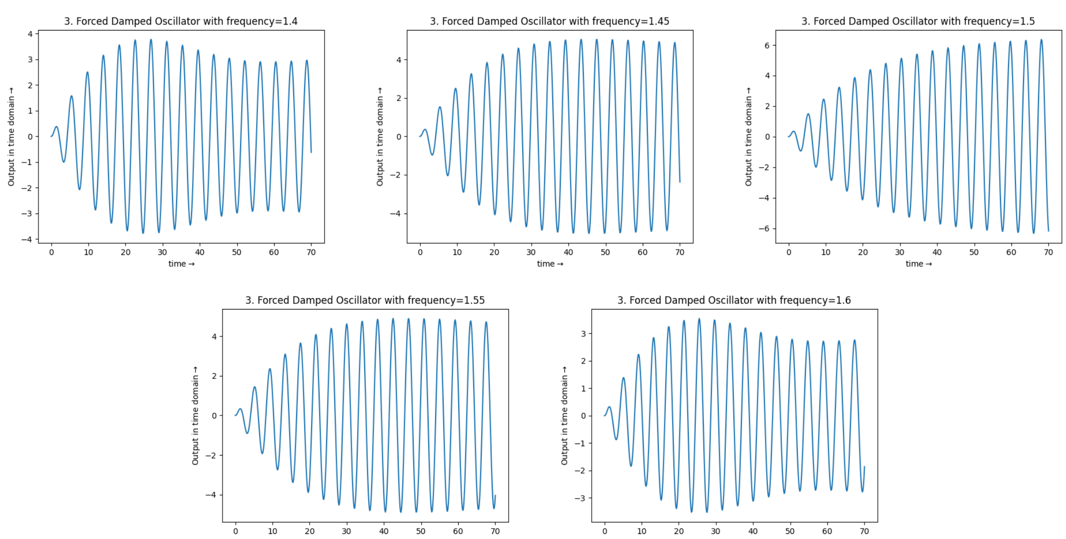


Figure 3: Impulse Response for varying frequencies

As we can see, the system responds with maximum amplitude at 1.5 as that is the resonant frequency.

2.4 Question 4:

We now consider a Coupled Differential System. The equations are the following:

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

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

Here we take the initial conditions to be $x(0) = 1$, $\dot{x}(0) = y(0) = \dot{y}(0) = 0$. To solve the equations we subsitute y from the first equation into the second to get a fourth order equation. Taking Laplace Transform and solving for $X(s)$ and $Y(s)$, we get:

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

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

We shall again use SciPy's impulse() to compute the signals in the time domain. Code snippet for this question is:

```
### Computing the signals in time domain
X = sp.lti([1, 0, 2], [1, 0, 3, 0])
t, x = sp.impulse(X, None, p.linspace(0, 20, 5001))

Y = sp.lti([2], [1, 0, 3, 0])
t, y = sp.impulse(Y, None, p.linspace(0, 20, 5001))
```

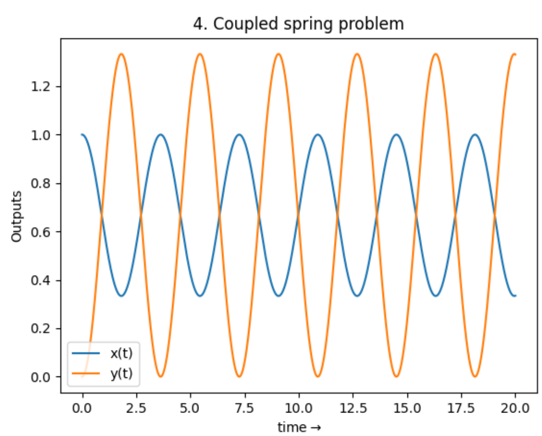
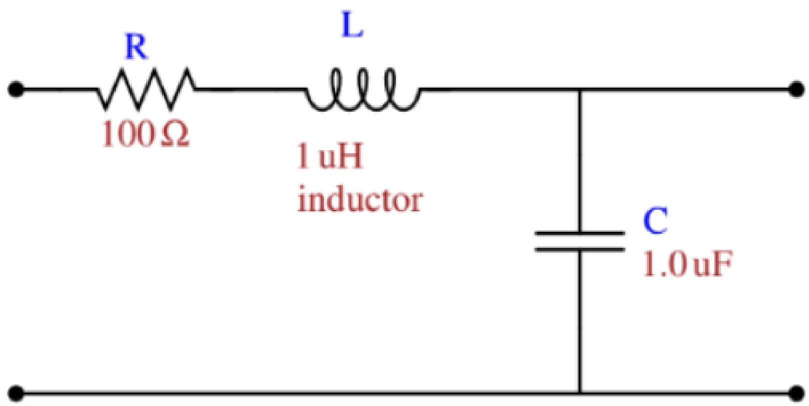


Figure 4: Coupled Oscillations

We notice that the outputs of this system are 2 sinusoids which are out of phase and have different magnitudes. The difference in the magnitudes is because of the fact that there is no symmetry between $x(t)$ and $y(t)$ for them to have the same magnitude.

2.5 Question 5:

Now we analyse the given two-port network and plot the Magnitude and Phase response of the Steady State transfer function:



This is a very simple LC oscillator with natural frequency, w_0, and Quality Factor (Q) as follows:

$$w_0 = \frac{1}{\sqrt{LC}}$$

$$Q = \frac{1}{R}\sqrt{\frac{L}{C}}$$

The transfer function of this system is given by:

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

Code snippet for converting this into time domain is given by:

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

Plots of Magnitude and Phase response:

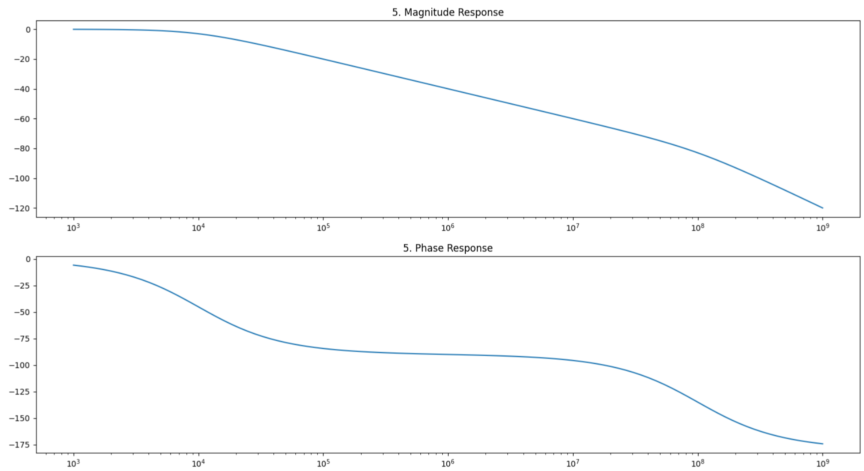


Figure 5: Bode Plots for RLC LPF

2.6 Question 6:

For the previous system, now we shall give the following input signal $v_i(t)$:

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

We shall analyse the output in two levels: for $0 < t < 30\mu s$ and $0 < t < 10ms$

Code snippet for finding the output in time domain is given by:

```
## First we shall find the output upto time 30myus
t = p.linspace(0, 30e-6, 500)
vi = p.cos(1e3 * t) - p.cos(1e6 * t)
t, x, _ = sp.lsim(H, vi, t)

## First we shall find the output upto time 100ms
t = p.linspace(0, 10e-3, 100001)
vi = p.cos(1e3 * t) - p.cos(1e6 * t)
t, x, _ = sp.lsim(H, vi, t)
```

Plots of the output in time domain are:

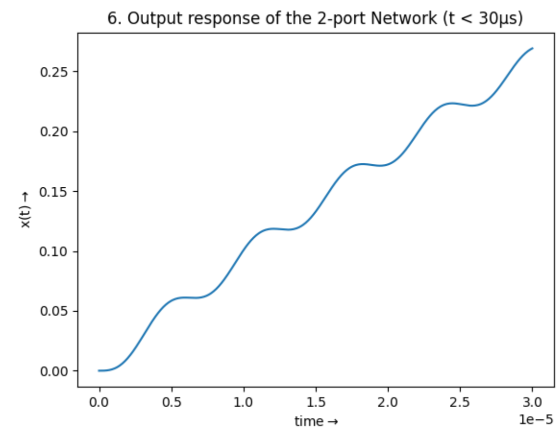


Figure 6: System Response for $t < 30\mu s$

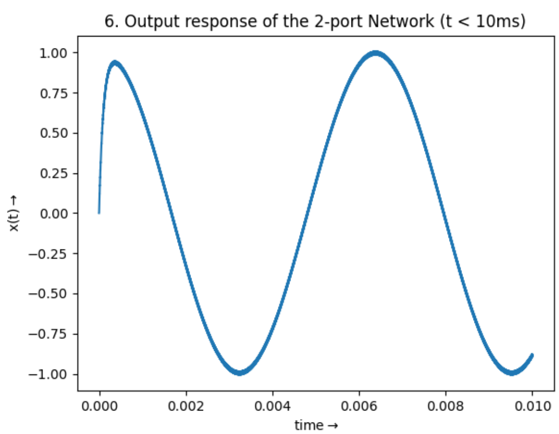


Figure 7: System Response for $t < 10ms$

From the graphs, we can see that the output of the system is almost a sinusoidal signal with frequency $10^3 rad/s$. This is because the system is a Lowpass Filter and higher frequencies can be seen as a noise.

3 Conclusion:

Thus we have plotted and analysed various LTI systems which are very crucial for engineering. We have used SciPy's signal processing library to analyse a wide range of LTI systems and have plotted their responses.