EE2703: Assignment 6B
The Laplace Transform

Nandha Varman
EE19B043

1 Introduction

We analyse LTI systems in continuous time using Laplace transforms to find the output of the system to a given input with the help of a python library, namely `scipy.signal` toolbox.

2 Time Response of Spring Oscillator

Our goal is to find the response of a spring oscillator, governed by the equation:

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

where,

$$x(t) = \text{Displacement of spring}$$
$$f(t) = \text{Force applied on the spring}$$

We consider the case that the force applied on the spring is given by:

$$f(t) = e^{-at}cos(\omega t)u(t)$$

We shall do a case by case analysis for the different values of a and ω:

2.1 Use of Laplace transforms

The Laplace transform of $f(t) = e^{-at}cos(\omega t)u(t)$ is given as:

$$\mathcal{L}\{f(t)\} = \frac{s+a}{(s+a)^2+\omega^2}$$

From the property of Laplace transforms, we know:

$$\begin{aligned} x(t) &\longleftrightarrow \mathcal{X}(s) \\ \implies \dot{x}(t) &\longleftrightarrow s\mathcal{X}(s) - x(0^-) \\ \implies \ddot{x}(t) &\longleftrightarrow s^2\mathcal{X}(s) - sx(0^-) - \dot{x}(0^-) \end{aligned}$$

From the above equations, we get, for $a = 0.5$ and $\omega = 1.5$:

$$\mathcal{F}(s) = \mathcal{L}\{f(t)\} = \frac{s+0.5}{(s+0.5)^2+2.25}$$

So, the equation of the spring oscillator can be written as:

$$s^2\mathcal{X}(s) - sx(0^-) - \dot{x}(0^-) + 2.25\mathcal{X}(s) = \frac{s+0.5}{(s+0.5)^2+2.25}$$

Given that the ICs $x(0)$ and $\dot{x}(0)$ are 0, we get:

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

or,

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

Using `scipy.signal.impulse` to find the x(t), and plotting it (for $0 < t < 50s$), we get:

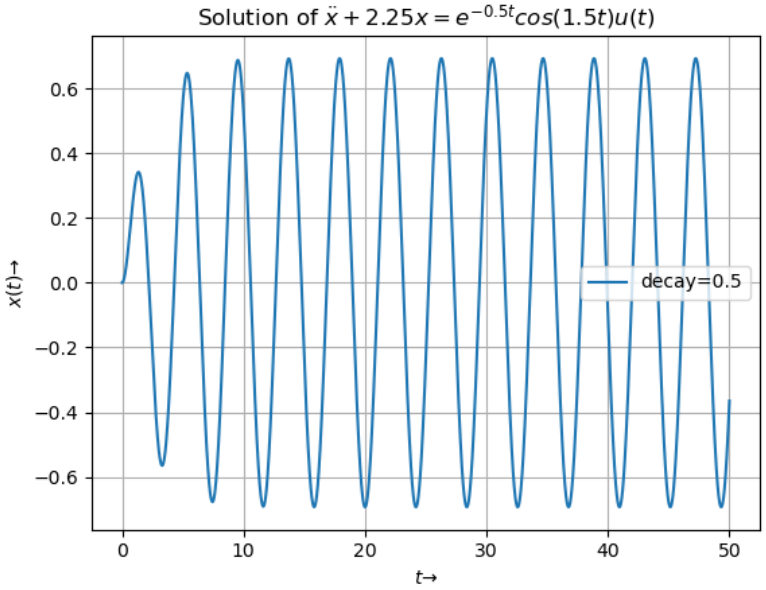

Figure 1: $x(t)$ for $a = 0.5$ and $\omega = 1.5$

If we use a smaller decay of $a = 0.05$, then we get:

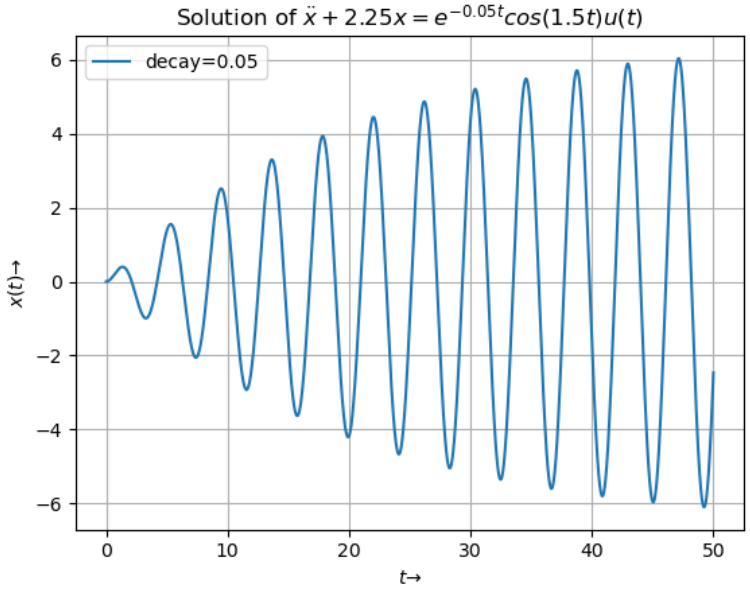

Figure 2: $x(t)$ for $a = 0.05$ and $\omega = 1.5$

2.2 Response for different frequencies

Modeling the system as a LTI system and computing the response for various frequencies, we get:

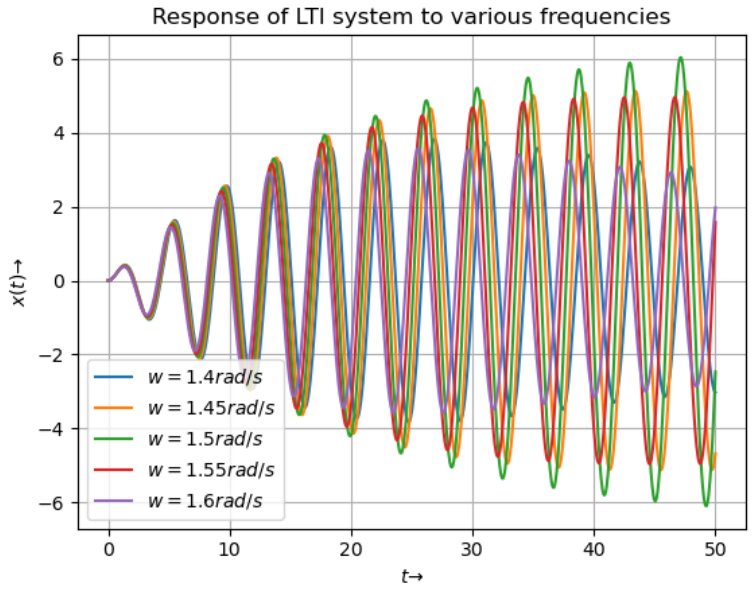

Figure 3: $x(t)$ for $a = 0.05$ and varying ω

From the given equation, we can see that the natural response of the system has the frequency $\omega = 1.5\ rad/s$. Thus, as expected, the maximum amplitude of oscillation is obtained when the frequency of $f(t)$ is 1.5 rad/s, as a case of **resonance**.

3 Coupled Spring Problem

The coupled equations we are interested in solving are:

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

Substituting for y from the 1st equation, into the 2nd, we get a 4th order differential equation in x:

$$\ddddot{x} + 3\ddot{x} = 0$$

Given the ICs $x(0) = 1$ and $\dot{x}(0) = y(0) = \dot{y}(0) = 0$, we can write the above differential equation in the Laplace domain as:

$$s^4\mathcal{X}(s) - s^3 + 3(s^2\mathcal{X}(s) - s) = 0$$
$$\implies \mathcal{X}(s) = \frac{s^2 + 3}{s^3 + 3s}$$
$$\implies \mathcal{Y}(s) = \frac{2}{s^3 + 3s}$$

Solving for $x(t)$ and $y(t)$ is now very simple - use `scipy.signal.impulse` with the above $\mathcal{X}(s)$ and $\mathcal{Y}(s)$. We get the following graph for $x(t)$ and $y(t)$.

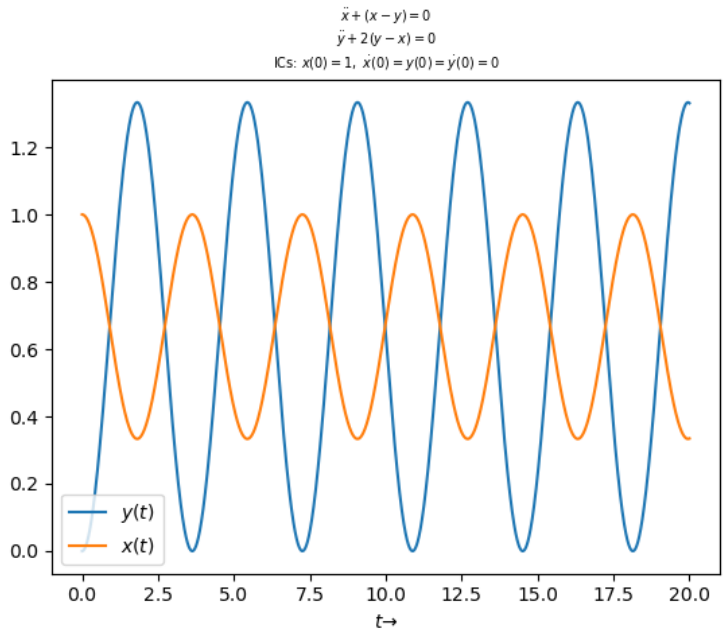

Figure 4: $x(t)$ and $y(t)$ for the coupled spring problem

We can see that $x(t)$ and $y(t)$ are sinusoids of the same frequency, but of different phase and magnitude.

4 Two-port Network

The transfer function of the given two-port network can be written as:

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

The Bode magnitude and phase plots can be found using the method `scipy.signal.bode()`. The plots are shown below:

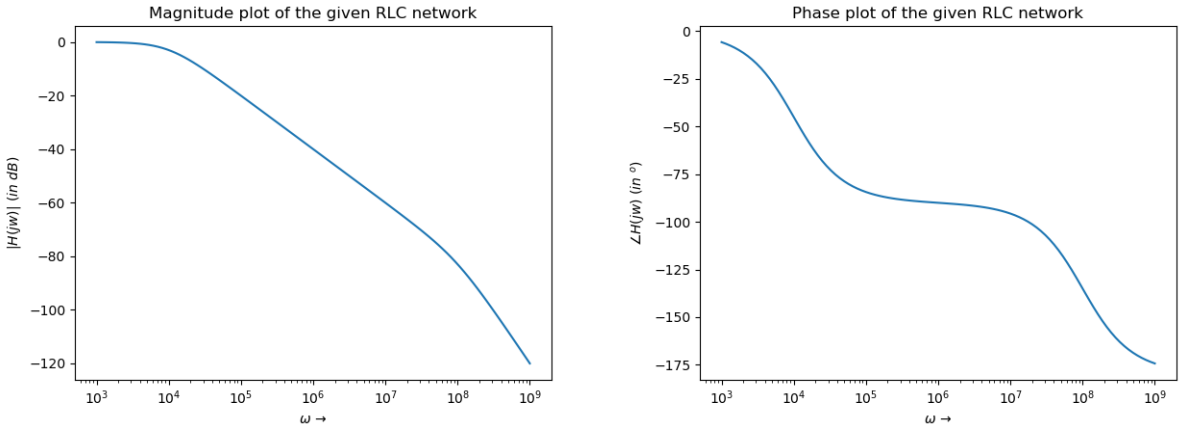

Figure 5: Bode Plots of the RLC Network's Transfer function

Now, when the input to the network, $v_i(t)$ is $(cos(10^3 t) - cos(10^6 t))u(t)$, the output is given by:

$$V_o(s) = V_i(s)\mathcal{H}(s)$$

Since we have already found out $\mathcal{H}(s)$ and $V_i(s)$ can be easily found by using a lookup table (or by substituting $a = 0$ in the equations used before), finding $v_o(t)$ is a simple task, thanks to `scipy.signal.lsim`. Plotting the obtained $v_o(t)$, we get:

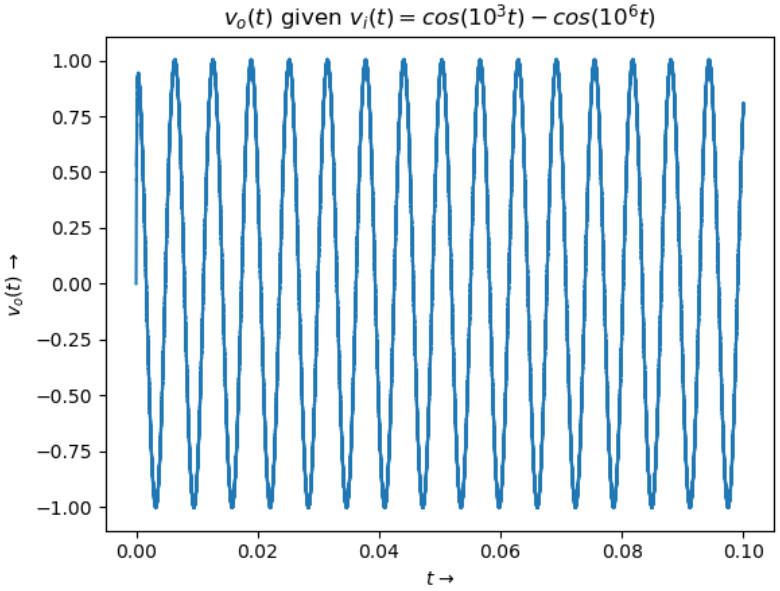

Figure 6: $v_o(t)$ of the RLC network, when $v_i(t) = (cos(10^3t) - cos(10^6t))u(t)$

We can see it to be varying as a sinusoid of frequency approximately 160 Hz, which is expected, as the RLC network acts as a low pass filter - it allows low frequencies to pass through unchanged, while damping high frequencies to huge extent. We can see this in the graph. If we zoom in Figure (6), we can see the high frequency signal:

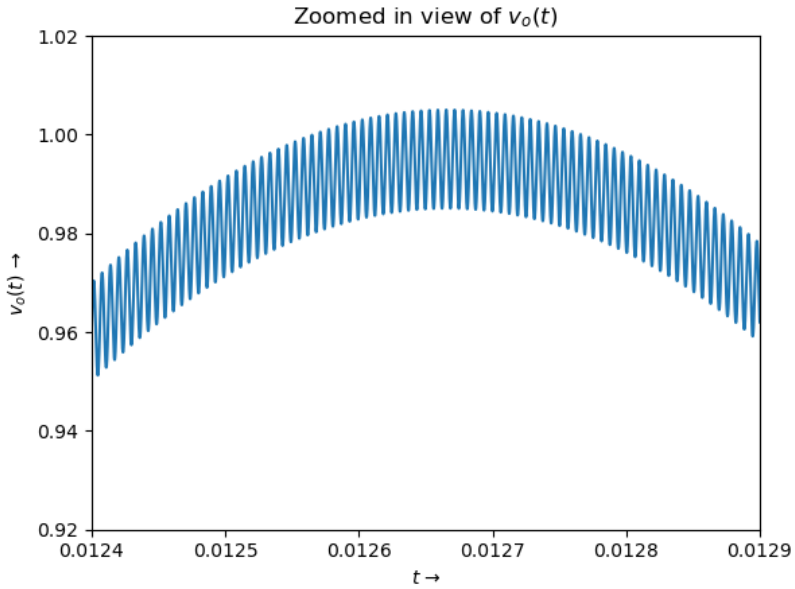

Figure 7: High frequency signal in the output

We observe that the peak-to-peak amplitude of this high frequency variation is very less, approximately 0.02 V, compared to the initial 2 V. This is expected as we get a gain of -40dB at $\omega = 10^6$ rad/sec from the bode plot, which corresponds to gain factor of 0.01.

Another peculiarity of Figure (6) is the initial variation. When zoomed in, for $0 < t < 30$ us, we get:

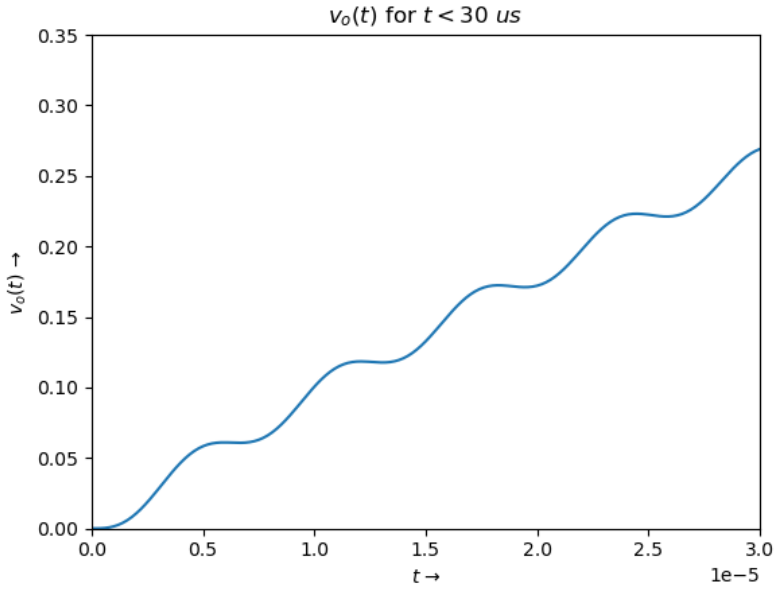

Figure 8: Initial transients

This is due to application of the step input, i.e., the input is suddenly turned on at $t = 0$.

5 Conclusion

To conclude, we analysed the solution of various continuous time LTI systems using Laplace transforms with help of `scipy.signal` toolbox.