Assignment No 6

Subhankar Chakraborty

March 11, 2019

1 The Laplace Transform

The Laplace transform is an integral transform perhaps second only to the Fourier transform in its utility in solving physical problems. The Laplace transform is particularly useful in solving linear ordinary differential equations such as those arising in the analysis of electronic circuits.The **Unilateral Laplace Transform** in the integral form for a function $f(t)$ can be stated as

$$\mathcal{L}_t[f(t)](s) \equiv \int_{0^-}^{\infty} f(t)e^{-st}dt$$

2 Questions

2.1 Question 1

The spring system given in the question satisfies the double differential equation

$$\frac{\partial^2 x}{\partial t^2} + x = f(t)$$

The input to the system, $f(t)$ is

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

Apart from this, we have the initial conditions on $x(t)$ given by $x(0) = 0$ and $\dot{x}(0) = 0$.From this we can find the system transfer function, $H(s) = \frac{X(s)}{F(s)}$.The system transfer function in this case, $H(s)$ is $\frac{1}{(s^2+2.25)}$.
The Laplace transform of the input, $F(s)$, is of the form

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

We know that the system transfer function($H(s)$) and the input($F(s)$) are related to the output($X(s)$) by

$$X(s) = F(s) * H(s)$$

After finding out $X(s)$ we can use the **inverse laplace transform** to calculate the value of $x(t)$. The following section of code does this:

```
f1_t = np.cos(1.5*t)*np.exp(-0.5*t)
transfer_function = sp.lti([1],[1,0,2.25])
t,h_t = sp.impulse(transfer_function,None,t)
t,y,svec = sp.lsim(transfer_function, f1_t, t)
```

The output($x(t)$) along with the input and the impulse response can be seen in Figure-1.

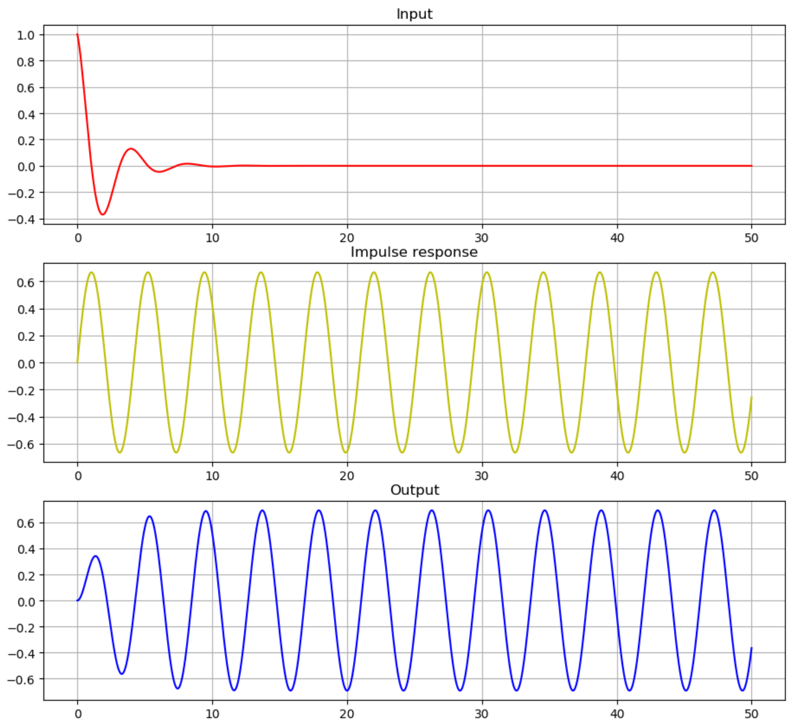

Figure 1: Plots for $f(t) = cos(1.5t)e^{-0.5t}u_0(t)$

2.2 Question 2

we are now required to plot the output through the same LTI system when the input has a much slower decay, ie,

$$f(t) = cos(1.5t)e^{-0.05t}u_0(t)$$

The code for plotting is almost the same and has hence been excluded from being mentioned here.The output can be seen in Figure-2.

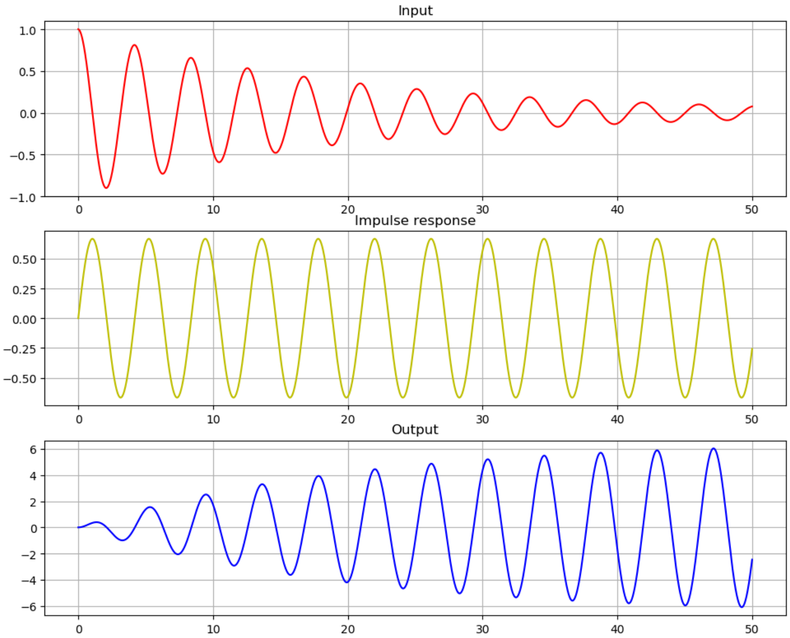

Figure 2: Plots for $f(t) = cos(1.5t)e^{-0.05t}u_0(t)$

2.3 Question 3

For this part of the assignment, we change the value of ω in

$$f(t) = cos(\omega t)e^{-0.05t}u_0(t)$$

and see how the system responds to the change in input.ω is varied from 1.4 to 1.6 in steps of 0.05. Figure-3 shows how the output changes according to varying ω.

The main inference that can be drawn from these graphs is as the driving frequency gets closer and closer to the natural frequency of oscillation(ω=1.5) here, the steady state oscillation amplitude keeps on increasing. It reaches the peak at ω=1.5 from ω=1.4 to ω=1.6. All of the above is done using a simple for loop.

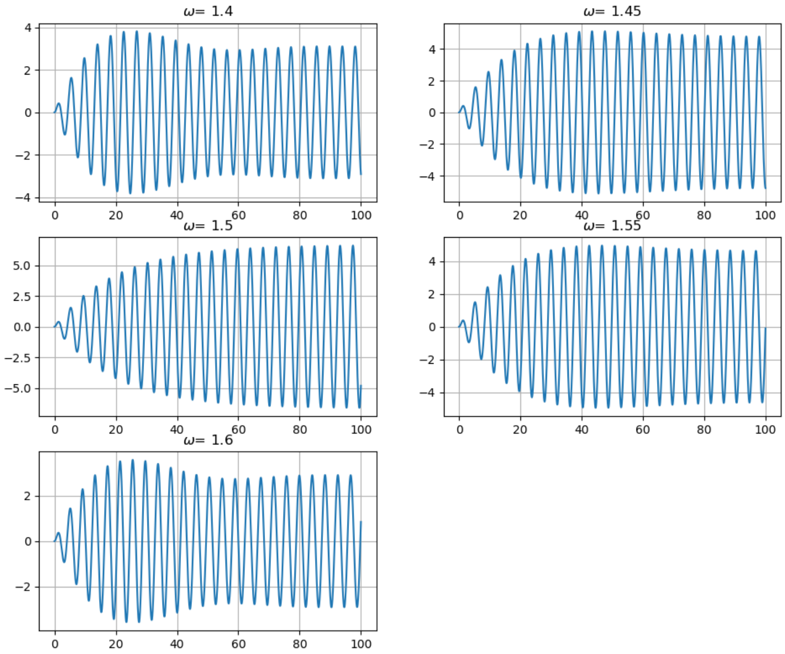

Figure 3: Plots for $f(t) = cos(\omega t)e^{-0.05t}u_0(t)$

```
w_vals = np.arange(1.4, 1.62, 0.05)
for w in w_vals:
    F_num = np.poly1d([1,0.05])
    f_t = np.cos(w*t)*np.exp(-0.05*t)
    t,y_temp,svec = sp.lsim(transfer_function_temp, f_t, t)
```

2.4 Question 4

The given equations are

$$\ddot{x} + (x - y) = 0 \tag{1}$$

and

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

We have also been given the initial conditions that $x(0) = 0, \dot{y}(0) = y(0) = \dot{x}(0) = 0$.Solving (1) and (2) coupled together in the Laplace Domain,

$$s^2X(s) - s + X(s) - Y(s) = 0$$

$$s^2Y(s) + 2(Y(s) - X(s)) = 0$$

from the above two equations we get for $X(s)$ and $Y(s)$,

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

and

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

We know the convolution of a function with the dirac delta($\delta(t)$) gives the function back. Using this property, the following piece of code finds the value of $x(t)$ and $y(t)$ from $x(s)$ and $y(s)$ respectively. The plots are shown in Figure-4.

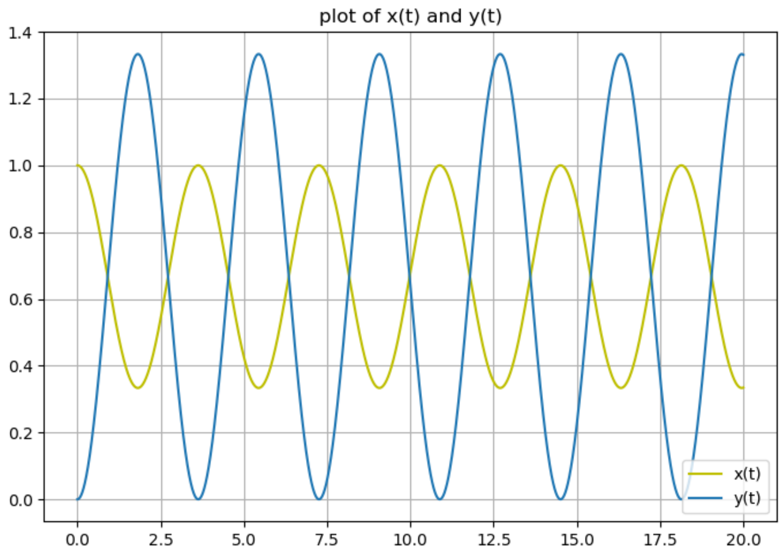

Figure 4: Values of $x(t)$ and $y(t)$

2.5 Question 5

The frequency response of the LTI system given in the figure in the assignment pdf is

$$H(j\omega) = \frac{\frac{1}{j\omega C}}{R + j\omega L + \frac{1}{j\omega C}}$$

or

$$H(j\omega) = \frac{1}{1 - \omega^2 LC + j\omega RC}$$

substituting the values of R, L and C, we get

$$H(j\omega) = \frac{1}{1 - \omega^2 10^{-12} + j\omega 10^{-4}}$$

Figure-5 shows the magnitude and phase response of the transfer function. This is taken care of by the following piece of code.

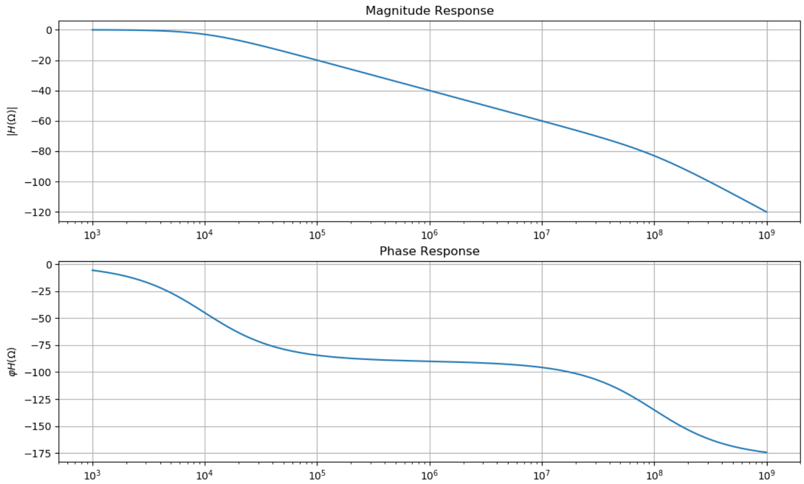

Figure 5: Bode plot of $H(j\omega)$

```
RLC_transfer_function = sp.lti([1e12], [1,1e8,1e12])
w,S,phi = RLC_transfer_function.bode()
```

2.6 Question 6

We pass an input voltage $v_i(t)$ through the RLC circuit. $v_i(t)$ is given by

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

We plot $v_o(t)$, which is the optput voltage as a function of time.

```
time_vec = np.linspace(0,30e-6, int(1e5))
v_i = np.cos(1000*time_vec)-np.cos(1e6*time_vec)
time_vec,v_o, svec = sp.lsim(RLC_transfer_function,v_i,time_vec)
plt.plot(time_vec, v_o)
```

$v_o(t)$ has been plotted in two time scales, one upto $30\mu s$ and another upto $10ms$. Figures 6 and 7 respectively show the plots.

The main inferences drawn from the plots of $v_o(t)$ are

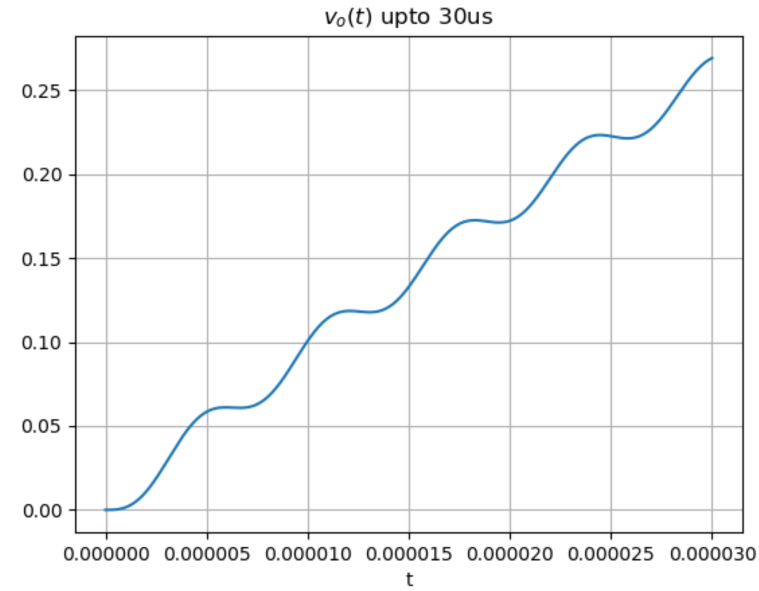

Figure 6: Values of $v_o(t)$ upto $30\mu s$

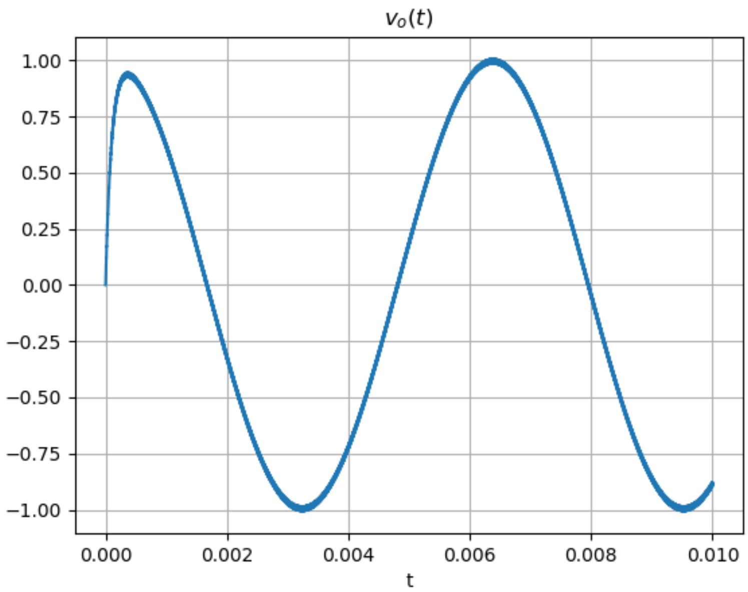

Figure 7: Values of $v_o(t)$ upto $10ms$

- The magnitude of the bode plot of the RLC filter at ω=1000 is about 0dB. So the component of $v_i(t)$ corresponding to $\omega = 1000$ remains almost unaffected in magnitude.

- The magnitude of the bode plot corresponding to $\omega = 10^6$ is about -40dB. So this component is highly attenuated.
- So overall we have a highly attenuated sinusoid of $\omega = 10^6$ riding on top of a sinusoid of $\omega = 1000$. So on the whole we still see the structure of $v_o(t)$ as a sinusoid of $\omega = 1000$. However, on zooming in, the small ripples corresponding to the small $\omega = 10^6$ component are visible(as in the graph plotted upto $30\mu s$).