EE2703 : Applied Programming Lab
Week6 : The Laplace Transform

Aditya Suresh
EE20B004

March 27, 2022

AIM

- to analyse Linear Time-invariant Systems with numerical tools in Python.
- Plot the system response in time domain
- Plot Bode plot the system's impulse response
- To plot the output in time domain for a given input signal

Theory

Laplace Transform

Laplace transform converts time domain signals into frequency domain.The best way to convert differential equations into algebraic equations is the use of Laplace transformation.

Because of this possibility of producing frequency domain signals which have algebraic relationship, helps solve systems which have rather complex tiime domain equations.

Formula

Laplace transform is the integral transform of the given derivative function with real variable t to convert into a complex function with variable s. For $t0$, let f(t) be given and assume the function satisfies certain conditions.

$$F(s) = \int_0 f(t) \cdot e^{-s \cdot t} \cdot dt$$

Assignment Questions

Question 1

Given equation:

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

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

Solving using Laplace transform we get :

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

1. The following code is used to generate and plot $x(t)$

```
F = sp.lti([1, 0.5], np.polymul([1,0,2.25], [1,1,2.5]))
t = np.linspace(0, 50, 501)

t, x = sp.impulse(F, None,t )
plt.plot(t,x)
plt.xlabel(r"Time(s)$\rightarrow$")
plt.ylabel(r"x(t)$\rightarrow$")
plt.title("Forced oscillator with decay = 0.5")
plt.show()
```

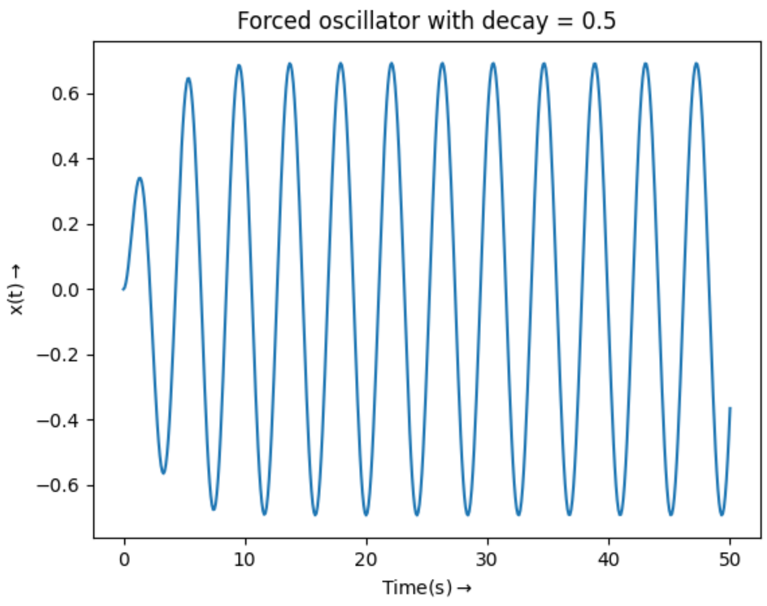

Figure 1: System response with decay = 0.5

Question 2

The equations remain same but decay factor has been changed to 0.05 So, $f(t) = cos(1.5t)e^{0.05t}u_0(t)$

1. The following code is used to generate and plot $x(t)$ for decay of 0.05

```
F = sp.lti([1, 0.05], np.polymul([1,0,2.25], [1,0.1,2.2525]))

t, x = sp.impulse(F, None,t )
plt.figure(2)
plt.xlabel(r"Time(s)$\rightarrow$")
plt.ylabel(r"x(t)$\rightarrow$")
plt.title("Forced oscillator with decay = 0.05")
plt.plot(t,x)
plt.show()
```

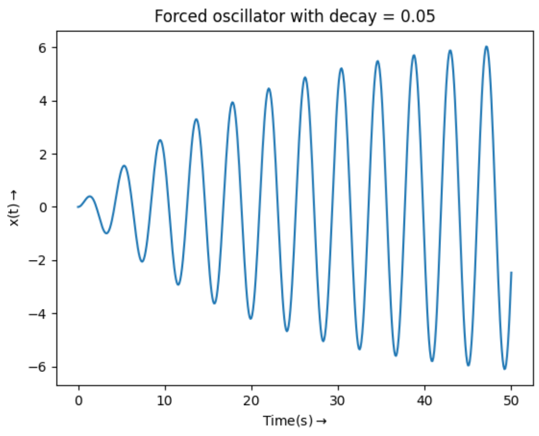

Figure 2: System response with decay = 0.05

We see that the system now has some delay before it reaches the steady state value. This is due to effect of smaller decay factor

Question 3

Solving $\frac{X(s)}{f(s)}$, we get H(s) as:

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

No we shall vary frequency of f(t) $f(t) = cos(frequency * t)e^{0.05t}u_0(t)$

where frequency $\in$ [1.4, 1.45, 1.5, 1.55, 1.6]

1. The following code is used to generate and plot $x(t)$ for various frequencies

```
H = sp.lti([1], [1,0,2.25])
frequencies = np.linspace(1.4,1.6,5)
t = np.linspace(0, 150, 1000)
for freq in frequencies:
    f = (np.cos(freq*t)) * (np.exp(-0.05*t))
    t, x, svec = sp.lsim(H, f,t )
    plt.xlabel(r"Time(s)$\rightarrow$")
    plt.ylabel(r"x(t)$\rightarrow$")
    string = "Forced oscillator with frquency = {}".format(freq)
    plt.title(string)
    plt.plot(t,x)
    plt.show()
```

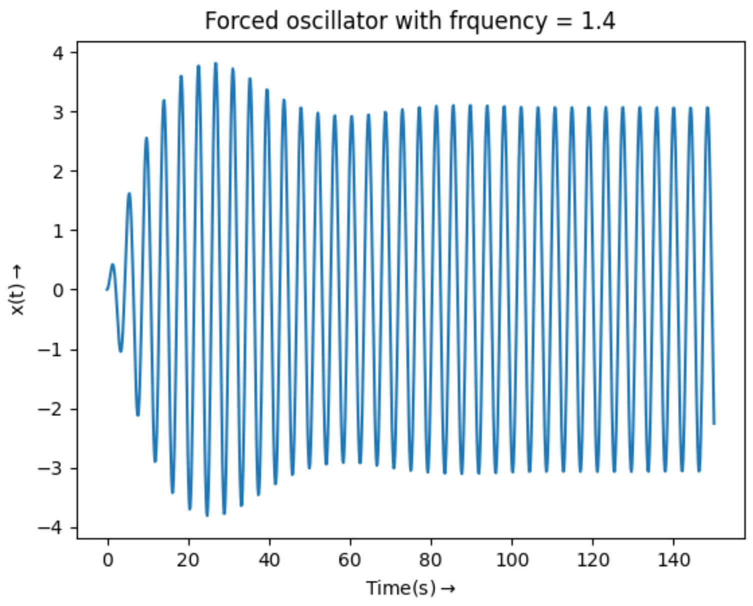

Figure 3: System response for frequency = 1.4

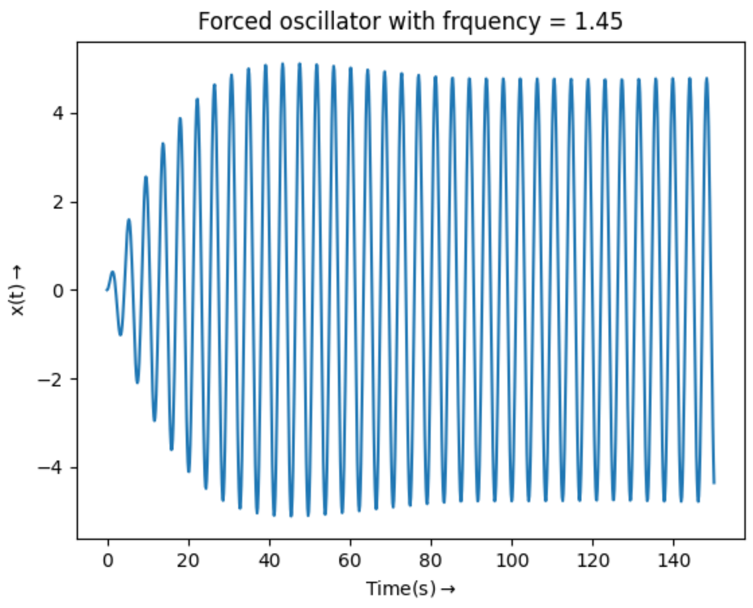

Figure 4: System response for frequency = 1.45

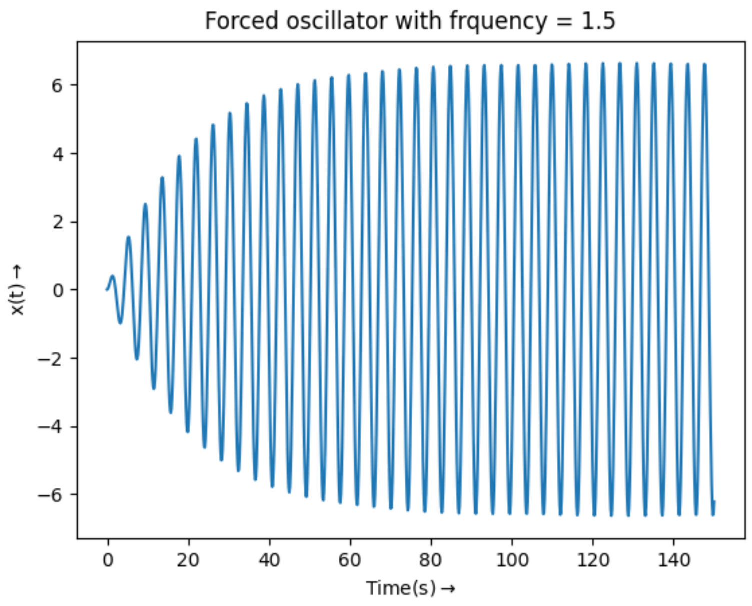

Figure 5: System response for frequency = 1.5

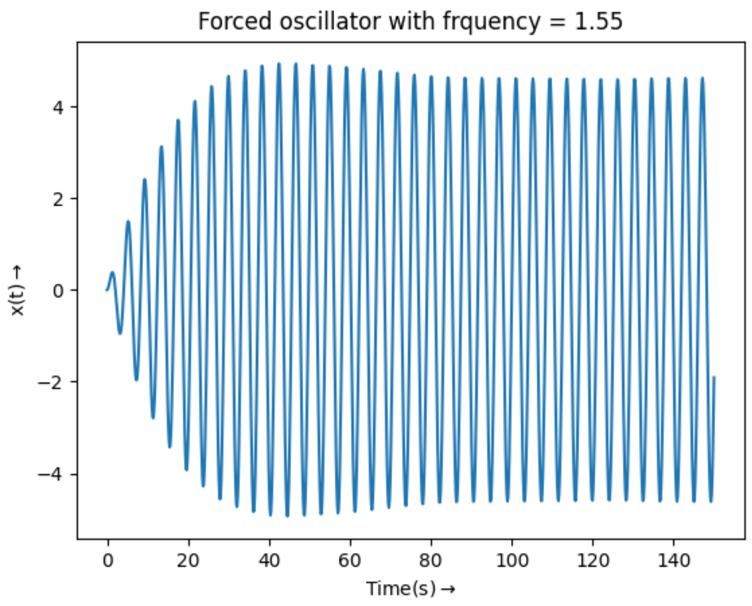

Figure 6: System response for frequency = 1.55

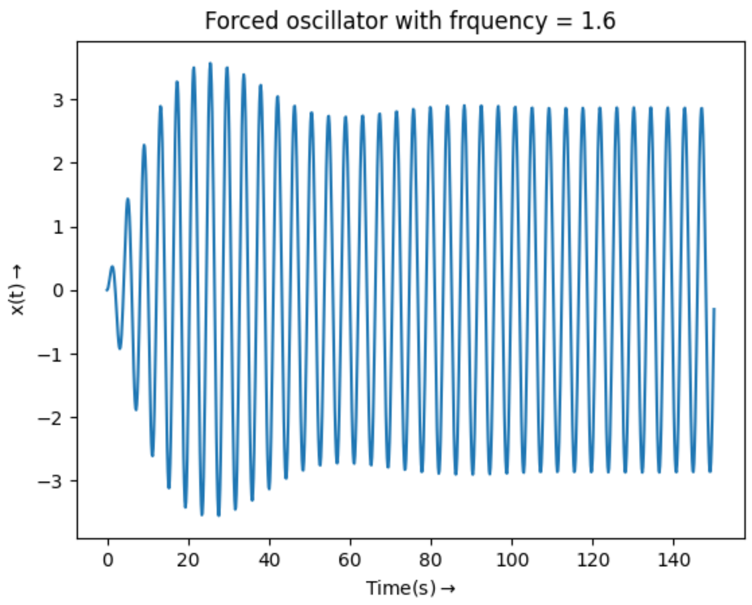

Figure 7: System response for frequency = 1.6

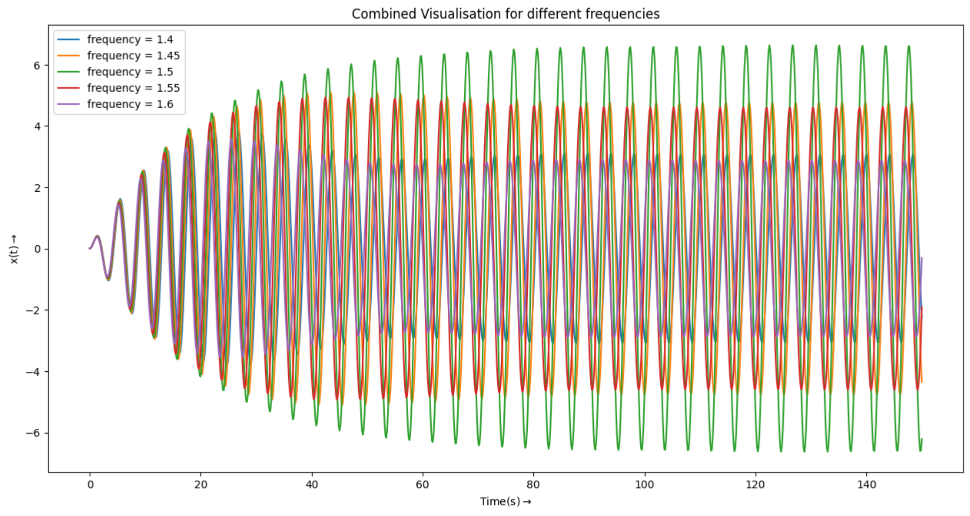

Figure 8: System response for all frequencies plotted in one

It is obvious from the plots that at frequency =1.5 , which is systems natural frequency, the response has maximum amplitude and at either side of it the amplitude decreases.

Question 4

Given equation:

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

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

where , $x(0) = 1, (0) = y(0) = (0) = 0$. Solving using Laplace transform we

get :

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

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

1. The following code is used to generate and plot $x(t) and y(t)$

```
X = sp.lti ( [1,0,2], [1,0,3,0] )
Y = sp.lti ( [2], [1,0,3,0] )
t = np.linspace(0,20, 501)

tx,x = sp.impulse(X, None, t)
ty, y = sp.impulse(Y, None, t)
plt.plot(tx,x, label ="x(t)")
plt.plot(ty, y, label = "y(t)")
plt.xlabel(r"Time(s)$\rightarrow$")
plt.ylabel(r"x(t) and y(t)$\rightarrow$")
plt.title("Coupled oscillations x and y")
plt.legend()
plt.show()
```

Question 5

Solving the system we get the transfer function to be

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

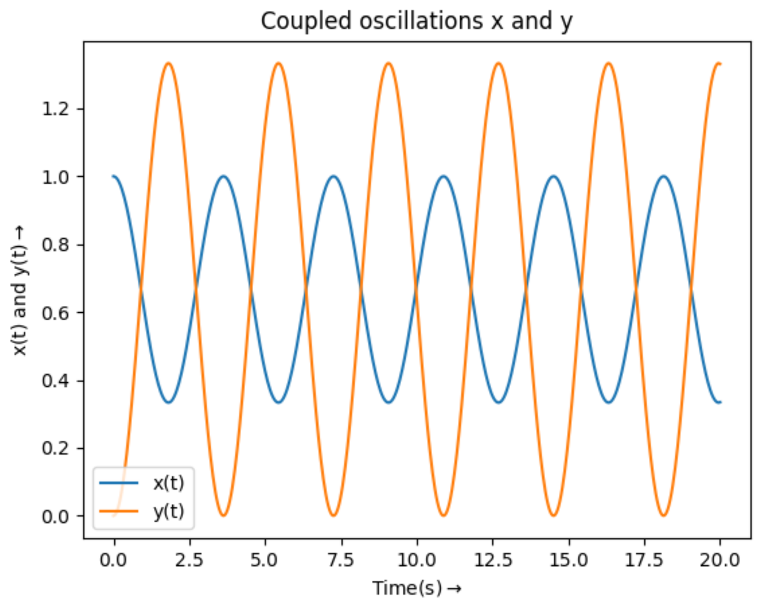

Figure 9: Coupled Oscillations

1. The following code is used to generate and plot Magnitude and phase response of the above function

```
H = sp.lti([1],[10**-12, 10**-4, 1])

w,S,phi=H.bode()
fig = plt.figure()
ax1 = fig.add_subplot(211)
ax1.semilogx(w,S)
ax2 = fig.add_subplot(212)
ax2.semilogx(w,phi)
ax1.title.set_text('Magnitude response')
ax2.title.set_text('Phase response')
ax1.set_xlabel(r'$\omega$')
ax1.set_ylabel('|H(jw)|l')
ax2.set_ylabel("/_H(jw)")
ax2.set_xlabel(r'$\omega$')
```

```
plt.show()
```

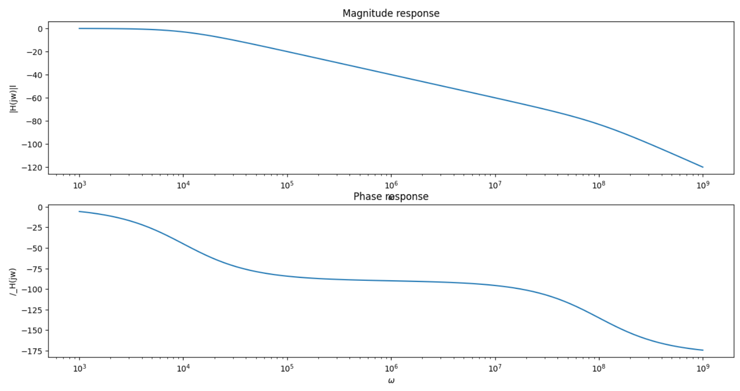

Figure 10: Bode Plot of the Transfer Function

As can be seen the system is low pass filter. Above the frequency of 10^3 hz the amplitude response falls.

Question 6

Given Input $v_i(t) = cos(10^3 t)u_0(t) - cos(10^6 t)u_0(t)$
:

1. The following code is used to generate and plot $v_o(t)$

```
t= np.linspace(0, 30e-6,10000)
vi = np.cos( (1e3) *t) -np.cos( (1e6) *t )
t, x, svec = sp.lsim(H, vi,t )
plt.xlabel(r"Time(s)$\rightarrow$")
plt.ylabel(r"$v_{o}(t)\rightarrow$")
plt.title("RLC response for time in microseconds")
plt.plot(t,x)
plt.show()

t= np.linspace(0, 30e-3,10000)
```

```
vi = np.cos( (1e3) *t) -np.cos( (1e6) *t )
t, x, svec = sp.lsim(H, vi,t )
plt.xlabel(r"Time(s)$\rightarrow$")
plt.ylabel(r"$v_{o}(t)\rightarrow$")
plt.title("RLC response for time in milliseconds")
plt.plot(t,x)
plt.show()
```

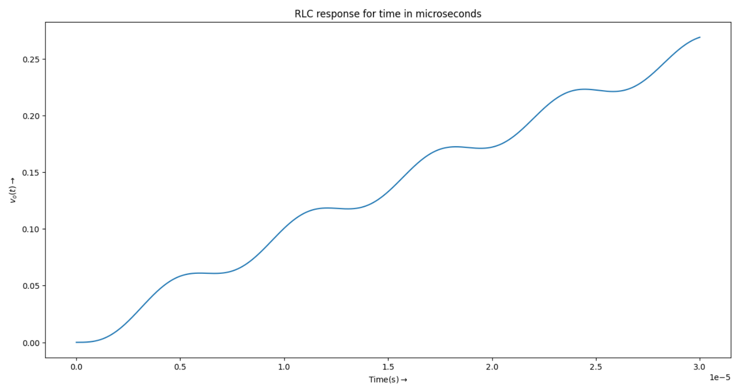

Figure 11: System response for t <30µm

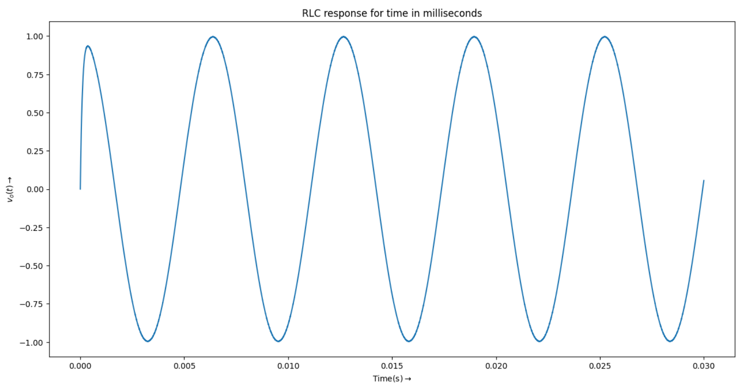

Figure 12: System response for t <30mm

While the timescale in milliseconds gives an output which looks like amplified version of input, The smaller timescale plot seems to be not so. This can be explained by seeing that the input contains 2 frequencies : 10^3 and 10^6. On the millisecond timescale the larger frequency is finely attenuated while on the microseconds scale , the larger frequency is not completely attenuated and contributes to h output

Conclusions

From this assignment we learnt the various methods used for signal processing. We also came to the conclusion that the system has highest amplitude when frequency matches its natural response. We analyzed and plotted forced oscillatory systems and checked on how various parameters affect the response