Assignment 6

K.R.Srinivas - EE18B136

March 11, 2020

Abstract

This week's Python assignment focuses on the following topics.

- To analyse "Linear Time-invariant Systems"
- Learn to use Signal-toolbox available in Python
- Specifically look at how to plot frequency magnitude and phase response of transfer functions

1 Introduction

The goal of this assignment is to look at how to analyze "Linear Time-invariant Systems" using the scipy.signal library in Python .We limit our analysis to systems with rational polynomial transfer functions. More specifically we consider 3 systems: A forced oscillatory system, A coupled system of Differential Equations and an RLC low pass filter.

Since the given problems are in continuous time domain, we solve the differential equations governing the problems in Laplace domain. Some of the equations to follow while finding laplace transform

$$L\{x(t)\} \rightarrow \mathcal{X}(s) \tag{1}$$

$$L\{\frac{dx(t)}{dt}\} \rightarrow s\mathcal{X}(s) - x(0^-) \tag{2}$$

$$L\{\frac{d^2x(t)}{dt^2}\} \rightarrow s^2\mathcal{X}(s) - sx(0^-) - \dot{x}(0^-) \tag{3}$$

2 Assignment Tasks

2.1 Forced Oscillatory System

We have to solve for the time repsonse of the spring mass system,whose driving force varies as $f(t)$ given as

$$f(t) = \cos(1.5t)e^{-0.5t}u_0(t) \tag{4}$$

Laplace transform of $f(t)$ using equations (1),(2) & (3) given above

$$\mathcal{F}(s) = \frac{s + 0.5}{(s + 0.5)^2 + 2.25} \tag{5}$$

Spring satisfies the below equation with $x(0) = 0$ and $\dot{x}(0) = 0$ for $0 \le t \le 50s$.

$$\ddot{x} + 2.25x = f(t) \tag{6}$$

which inturn gives,

$$\mathcal{X}(s) = \frac{s + 0.5}{((s + 0.5)^2 + 2.25)(s^2 + 2.25)} \tag{7}$$

the python code to solve the Laplace equation is given below.

```
def func(t, w, alpha):
  return cos(w*t)*exp(-alpha*t)

def time_response( alpha, wo, k):
  Dr = polymul([1,0,pow(k,2)],[1,2*alpha,(pow(wo,2)
                +pow(alpha,2))])
  Nr = poly1d( [1,alpha] )
  H = sp.lti(Nr, Dr)
  t, x = sp.impulse(H, None, linspace(0, 100, 10000))
  return H, t, x

X, t1, x1 = time_response(0.5, 1.5, 1.5)
X, t2, x2 = time_response(0.05, 1.5, 1.5)
```

Also we plot the response of the system if the decay is 0.05 instead of 0.5.

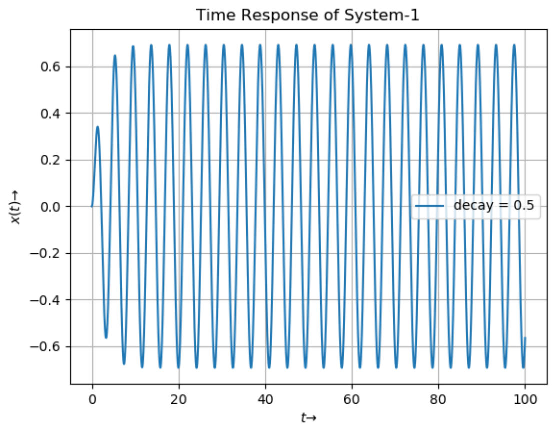

Figure 1: System response with decay $\alpha = 0.5$

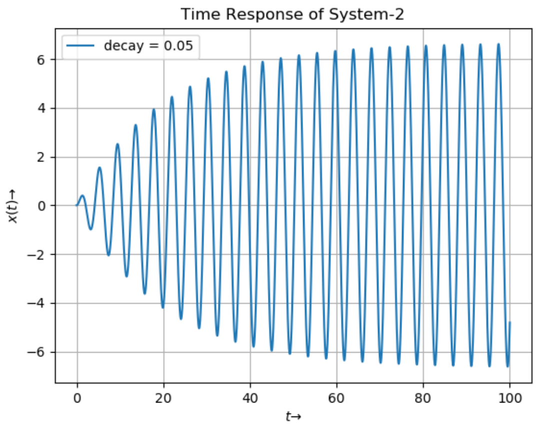

Figure 2: System response with decay $\alpha = 0.05$

We observe the plot that for smaller decay of $e^{-0.05t}$, $x(t)$ has large amplitude and its growing as time increases and oscillates,whereas the for higher decay value the amplitude of $x(t)$ is very small.

Our input $f(t)$ to the system has natural frequency that is $w = w_0$, so its a resonant case,so the solution of differential equation for sinusoidal inputs from observing the plot can be of the form $te^{-dt}\cos(w_0 t)$ so for smaller decay value the graph takes more time neutralise the growing effect of t in the solution.So to conclude for small decay , $x(t)$ has large amplitude and the time required for it settle or saturate to a certain maximum amplitude is higher compared to large decay case.

The Magnitude and phase of the output frequency response for the given input is plotted below.

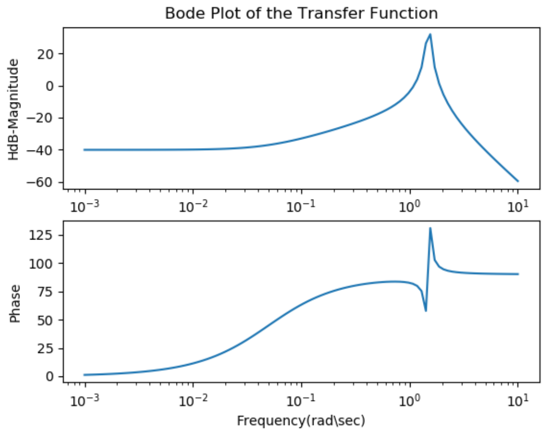

Figure 3: Frequency Response of the system for the given Input

The corresponding time domain response of the system at discrete frequencies ranging between 1.4 to 1.6 rad/s is plotted below. As expected from frequency we see that at resonant frequency the amplitude of output signal is maximum.

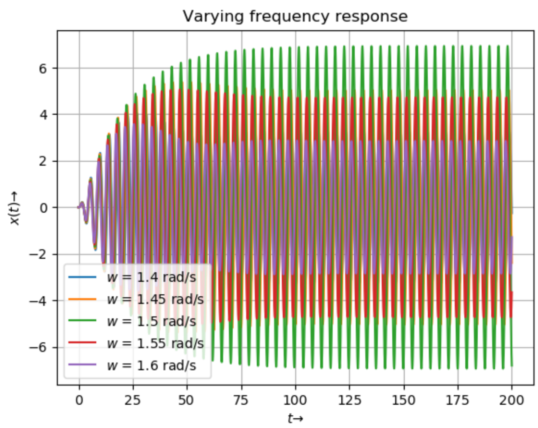

Figure 4: Transient Response of the system for the given Input

2.2 Coupled Spring Problem

The System satisfies the below equation with $x(0) = 1$ and $\dot{x}(0) = y(0) = \dot{y}(0) = 0$.

$$\ddot{x} + (x - y) = 0 \tag{8}$$

$$\ddot{y} + 2(y - x) = 0 \tag{9}$$

We have to solve for $x(t)$ and $y(t)$ for $0 \leq t \leq 20s$ by taking laplace transform of both equations given above and solve for $\mathcal{X}(s)$ and $\mathcal{Y}(s)$ using substitution method.Now from $\mathcal{X}(s)$ and $\mathcal{Y}(s)$ we can find $x(t)$ and $y(t)$ using *system.impulse*.

```
X_s  = sp.lti([1,0,2],[1,0,3,0])
t, x = sp.impulse(X_s, None, linspace(0, 100, 10000))

Y_s = sp.lti([2],[1,0,3,0])
t1, x1 = sp.impulse(Y_s, None, linspace(0, 100, 10000))
```

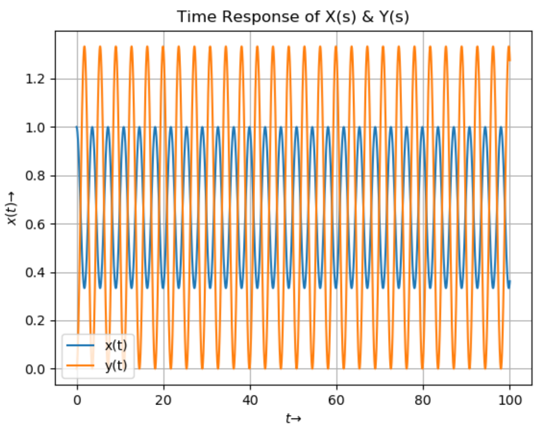

Figure 5: Transient Response for Coupled System

We observe that, the $x(t)$ and $y(t)$ obtained satisfies the given initial conditions,and oscillating sinusoidally with 180^o out of phase.

2.3 RLC - Lowpass Filter

In this section we are suppose to solve for the steady state transfer function of a two-port RLC network, whose transfer function is given as below.

$$\frac{\mathcal{V}_0(s)}{\mathcal{V}_i(s)} = \mathcal{H}(s) = \frac{1}{s^2LC + sRC + 1} \tag{10}$$

For the given values of $R = 100\Omega$, $L = 1\mu H, C = 1\mu F$, we get

$$\mathcal{H}(s) = \frac{1}{s^2 10^{-12} + s10^{-4} + 1} \tag{11}$$

```
R = 100
C = 1e-6
L = 1e-6

H_s = sp.lti( [1], [L*C,R*C,1] )
```

The magnitude and phase response are plotted below.

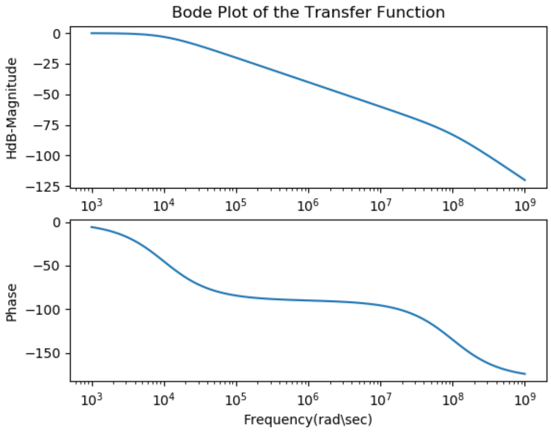

Figure 6:

We observe the system has poles on left half s plane, that too real poles with $s = -10^4, -10^8$ $rads^{-1}$.since all the poles are in left half s plane RLC Network given is unconditionally stable for given values.

Now,If the input signal $v_i(t)$ is given by

$$v_i(t) = \cos(10^3 t)u(t)\cos(10^6 t)u(t) \tag{12}$$

We need to obtain the output voltage $v_0(t)$ using the transfer function of the system obtained for $0<t<30\mu s$

```
def input(t):
    return cos(1e3*t)-cos(1e6*t)
t = linspace( 0, 30e-6, 10000)

t, output, svec = sp.lsim(H_s, input(t), t)
plot( t, output)
xlabel(r"$t \to $")
ylabel(r"$y(t) \to $")
grid()
show()
```

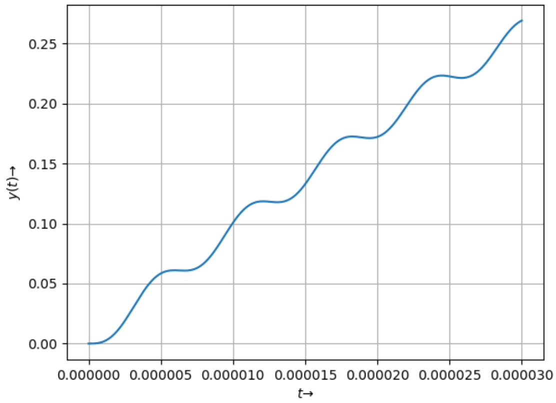

Figure 7:

The long term response is given below.

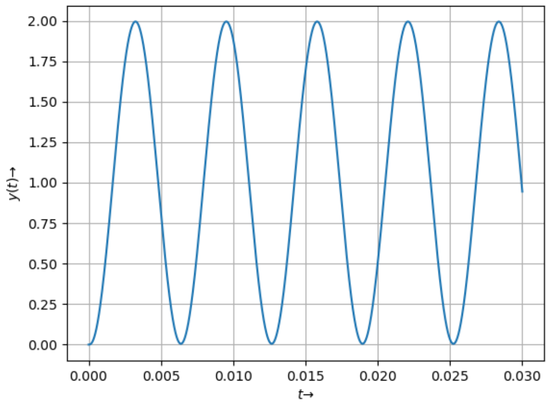

Figure 8:

We observe the plot and the circuit that we know it is a Low pass filter with bandwidth $0 < \omega < 10^4$.So when the circuit will only pass input with frequencies which are in range of bandwidth only. But since its not a ideal low pass filter as its gain doesn't drop abruptly at 10^4 rather gradual decrease which is observed from magnitude response plot.

3 Conclusion

So to conclude we analysed a way to find the solution of continuous time LTI systems using laplace transform with help of Python signals toolbox and got familiarised with solving of differential equations by taking laplace transform instead of doing arduous time domain analysis.