EE2703 Applied Programming Lab - Assignment 7

Name: Rajat Vadiraj Dwaraknath
Roll Number: EE16B033

March 17, 2018

1 Introduction

In this assignment, we deal with Linear Time Invariant systems and their responses to certain inputs. We use the scipy.signal module to perform analysis of systems with rational polynomial transfer functions. We look at a coupled system of differential equations, and also a linear electrical circuit which behaves like a low-pass filter.

2 Question 1

We solve for the response by using the property that the Laplace transform of the output $X(s)$ of a system with transfer function $H(s)$ to an input with Laplace transform $F(s)$ is given by:

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

We then use the sp.impulse function to find the inverse Laplace transform of the output over a certain range of times:

```
import scipy.signal as sp

def F_s(freq=1.5,decay=0.5):
    """Transfer function of the given system"""
    n = poly1d([1,decay])
    d = n*n+freq**2
    return n,d

print(F_s())
```

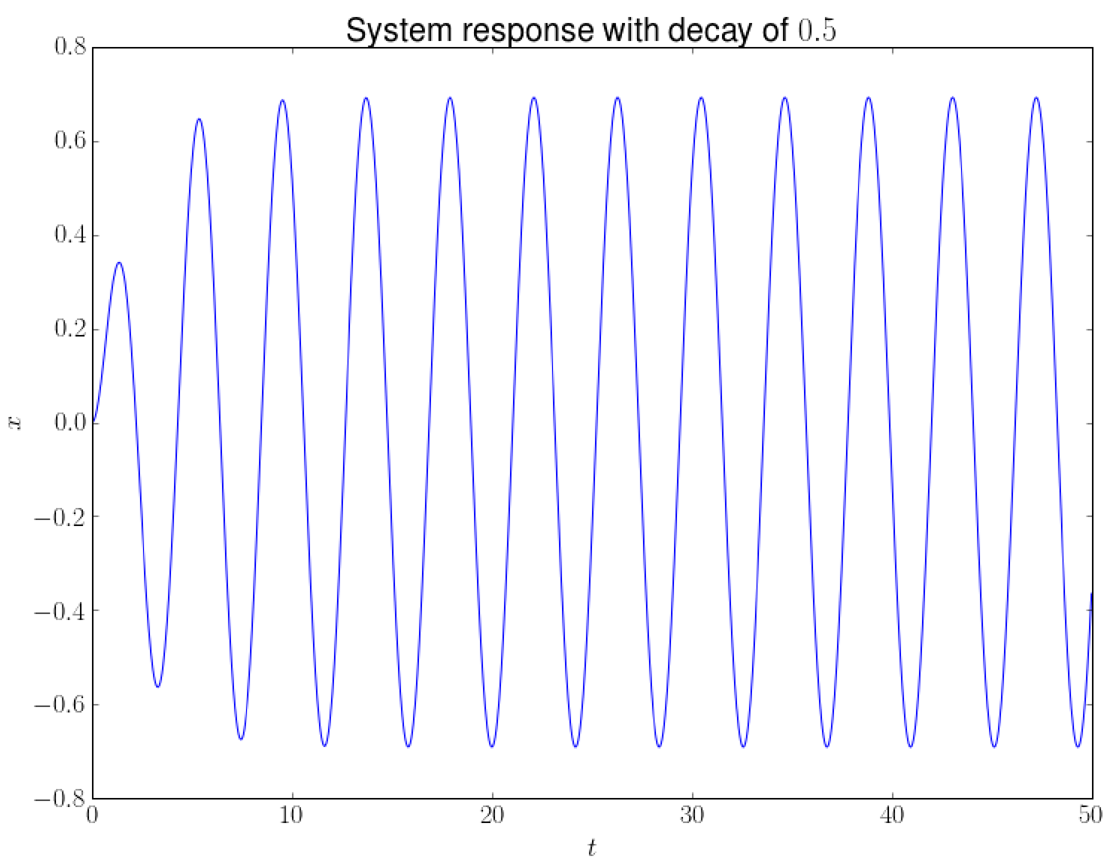
```
(poly1d([ 1. ,  0.5]), poly1d([ 1. ,  1. ,  2.5]))
```

```
def secondOrderH(wn=1.5,zeta=0,gain=1/2.25):
    """General second order all pole transfer function."""
    n = poly1d([wn**2*gain])
    d = poly1d([1,2*wn*zeta,wn**2])
    return n,d

print(secondOrderH())
```

```
(poly1d([ 1.]), poly1d([ 1.   ,  0.   ,  2.25]))
```

```
def solveProblem(decay):
    """Find the response to the given system to a decaying cosine."""
    inN, inD = F_s(decay=decay)
    HN, HD = secondOrderH()

    outN,outD = inN*HN, inD*HD

    out_s = sp.lti(outN.coeffs, outD.coeffs)

    t = linspace(0,50,1000)
    return sp.impulse(out_s,None,t)
```

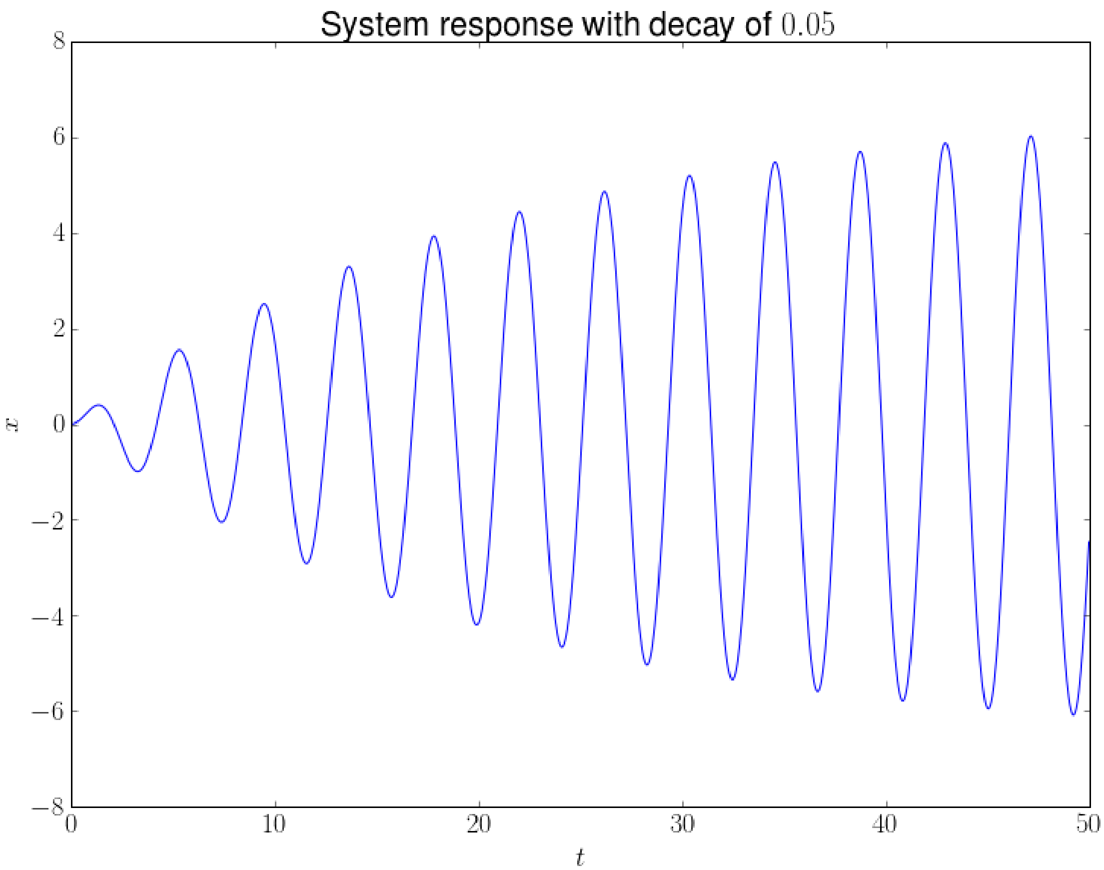
We plot the output for an input with a decay constant of 0.5:



We observe that the output of the system in steady state is a a sinusoid with the same frequency as the input, but with no decay. This is because the natural frequency of the system is equal to the frequency of the input.

3 Question 2

We repeat the plot with a decay constant of 0.05:



We observe that the steady state response follows the same trend as the previous case, except that it has a much larger amplitude. This is because the input excited the system for a longer duration due to its smaller decay constant. This resulted in a larger buildup of output due to resonance. We can see that, during the buildup of the output, the amplitude grows linearly. This is characteristic of resonance in a second order system. This will be made clear by exciting the system with slightly different frequencies:

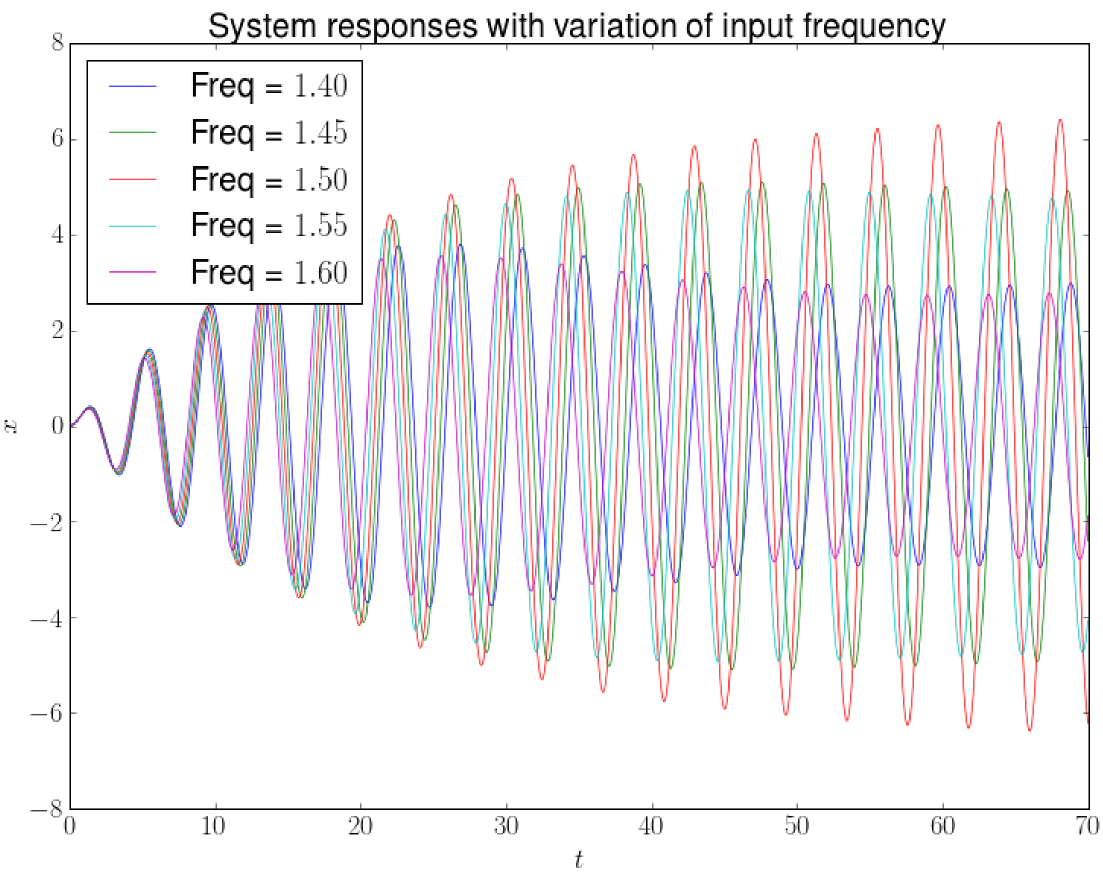
4 Problem 3

We find the response to inputs with slightly different frequencies around 1.5.

```
def input_f(t,decay=0.5,freq=1.5):
    """Exponentially decaying cosine function."""
    u_t = 1*(t>0)
    return cos(freq*t)*exp(-decay*t) * u_t
```

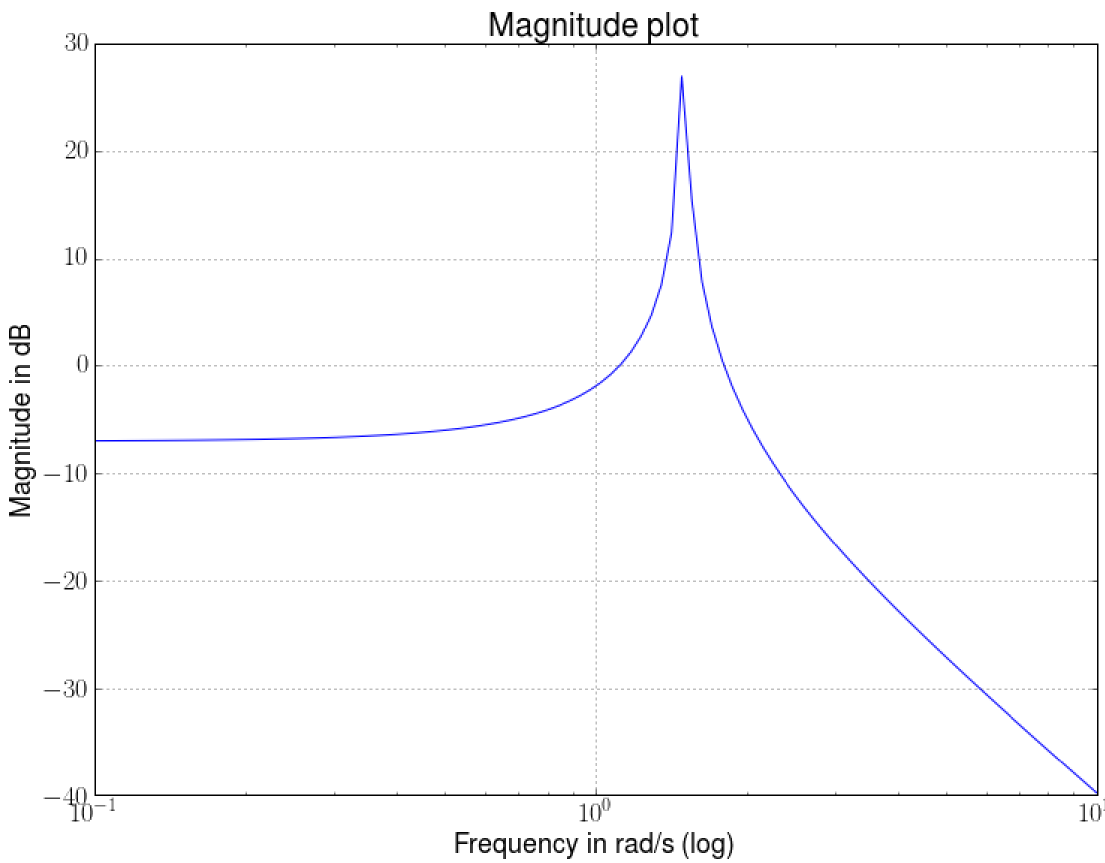
```
# get transfer function
system = secondOrderH()

# time range
t = linspace(0,70,1000)

# list of outputs
outs = []

# list of frequencies to iterate over
fs = linspace(1.4,1.6,5)


for freq in fs:
    # solve
    t,y,svec = sp.lsim(system,input_f(t,decay=0.05,freq=freq),t)

    # store
    outs.append(y)
```

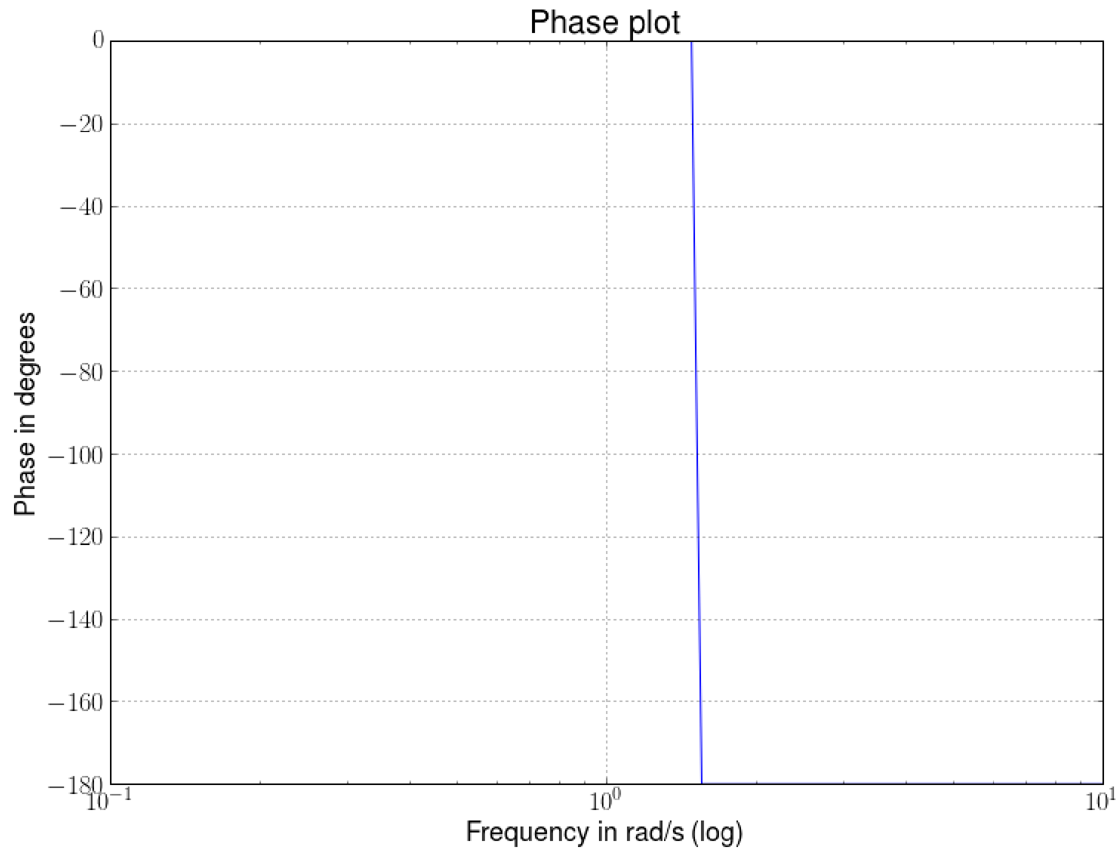
The response is plotted for frequencies around 1.5:



We observe that an input with frequency of exactly 1.5 reaches the largest steady state amplitude. This is because of the aforementioned resonance. Nearby frequencies are not tuned to the natural response of the system, so their amplitudes die down after the initial rise before reaching a steady state. This can also be understood by looking at the magnitude of the transfer function at these frequencies. Let us find the Bode plots of this transfer function:

```
w,S,phi=sp.lti(*system).bode()
```





From the above Bode plots, it is clear that this system is a second order system with complex conjugate poles. The quality factor of this system is actually infinte, but is not visible in the graph of the magnitude due to lack of resolution. It is much more evident in the phase plot, in which it drops from 0 degrees to -180 degrees like a step, which is indicative of infinite Q. This infinte quality means that the magnitude decays sharply around the resonant frequency of 1.5. This is why we observe a much lesser amplitude in output for frequencies which are slightly detuned from the resonant frequency.

5 Question 4

There are two ways to approach a set of coupled differential equations. One way is to solve them by eliminating one equation by substitution. Let us first use this approach. The given equations are:

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

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

We substitute for y in the second equation using the first and obtain the following:

$$x^{(4)} + 3x^{(2)} = 0$$

where the superscript in parenthesis denotes the order of the derivative.

We also need to solve for the initial conditions. We obtain the following set of initial conditions on only derivatives of x:

$$x(0) = 1$$
$$x^{(1)}(0) = 0$$
$$x^{(2)}(0) = -1$$
$$x^{(3)}(0) = 0$$

Using these conditions, we take the Laplace transform of the above equation and solve for $X(s)$ to get:

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

Substituting back, we find for $Y(s)$:

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

We can now invert these Laplace transforms to find the solutions in the time domain. However, a more interesting approach to the problem would be to decouple the set of differential equations. This is done by finding the Jordan decomposition of the coefficient matrix. We obtain the following change of variables for decoupling:

$$u = x - y$$
$$v = 2x + y$$

The system of equations reduces to two independent equations:

$$\ddot{u} + 3u = 0$$
$$\ddot{v} = 0$$

With the following initial conditions:

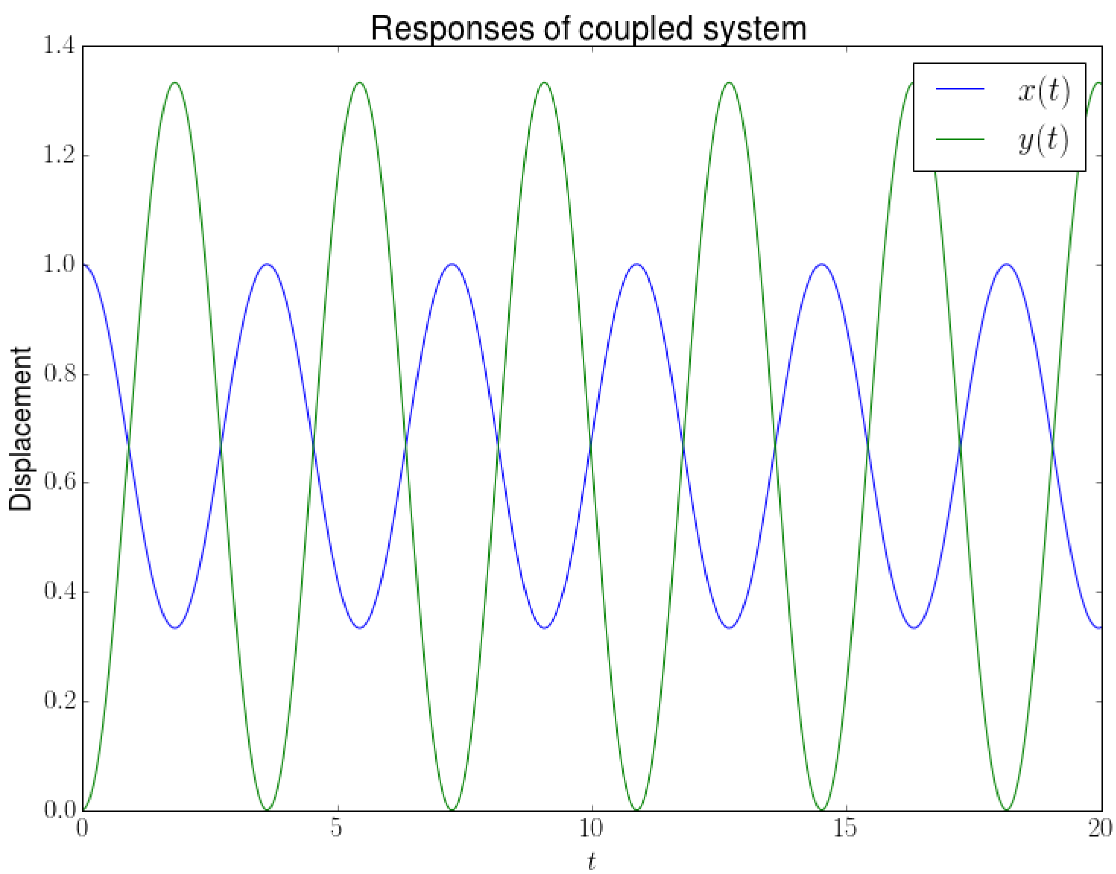
$$u(0) = 1$$
$$\dot{u}(0) = 0$$
$$v(0) = 2$$
$$\dot{v}(0) = 0$$

These can be easily solved independently, as they are just second order systems as opposed to a fourth order system. The solutions obtained for u and v can then be transformed back to solutions for x and y using the change of variables described above. We obtain the same results for $X(s)$ and $Y(s)$ using this method as before.

We now invert these Laplace transforms to obtain the solutions in time domain:

```
X_s = sp.lti([1,0,2],[1,0,3,0])

Y_s = sp.lti([2],[1,0,3,0])

t = linspace(0,20,1e3)

t, x = sp.impulse(X_s,None,t)
t, y = sp.impulse(Y_s,None,t)
```

The solutions are plotted below:



- We observe that the solutions are sinusoidal with a certain DC offset.
- This is evident from the expressions of the Laplace Transforms of the solutions as the denominator contains a factor of s.
- We observe that these two DC offsets are the same for both x and y.
- This can be confirmed by noticing that the differential equation in $u = x - y$ has no forcing function, so the DC offset of u must be 0. Therefore, x and y must have the same DC offset.
- They also oscillate with the same frequencies because the coefficient of the second derivative term is the same in both differential equations.

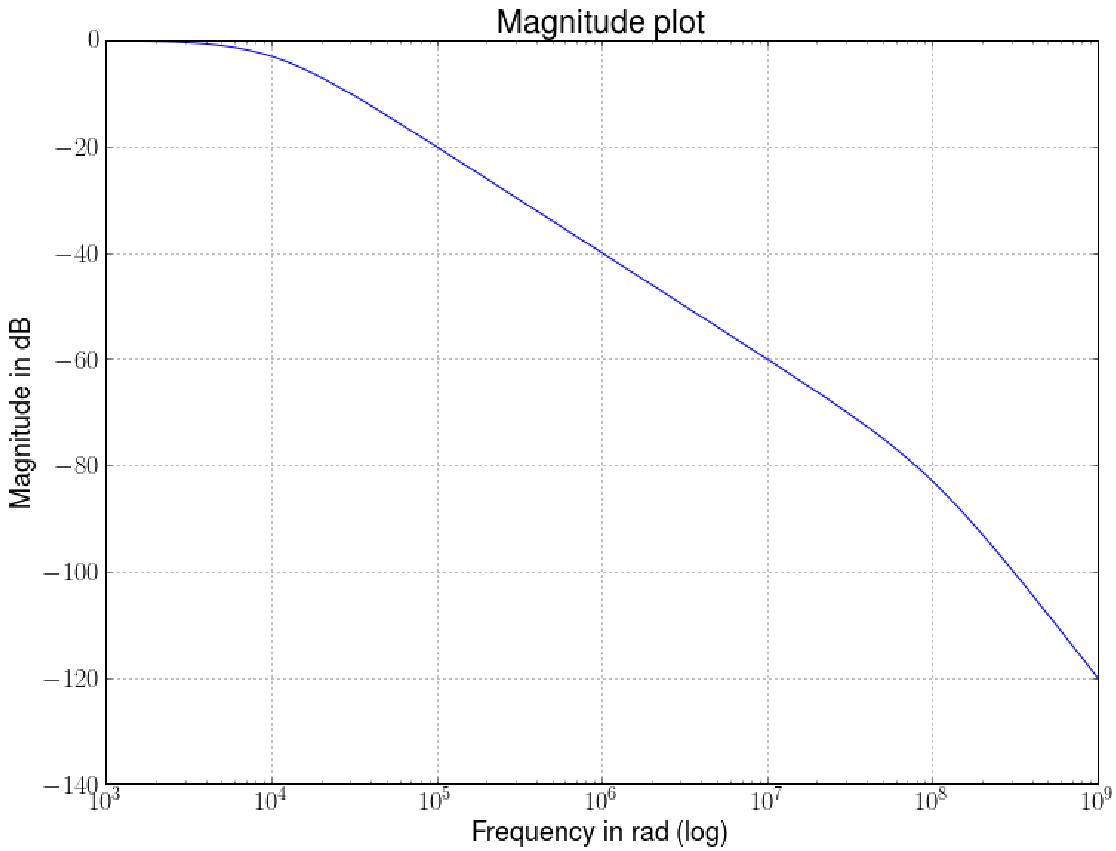
- This system of equations models two masses attached to the two ends of an ideal spring with no damping. x and y are the positions of the masses in a reference frame moving at the same speed as the centre of mass, but offset from the centre of mass by some amount.

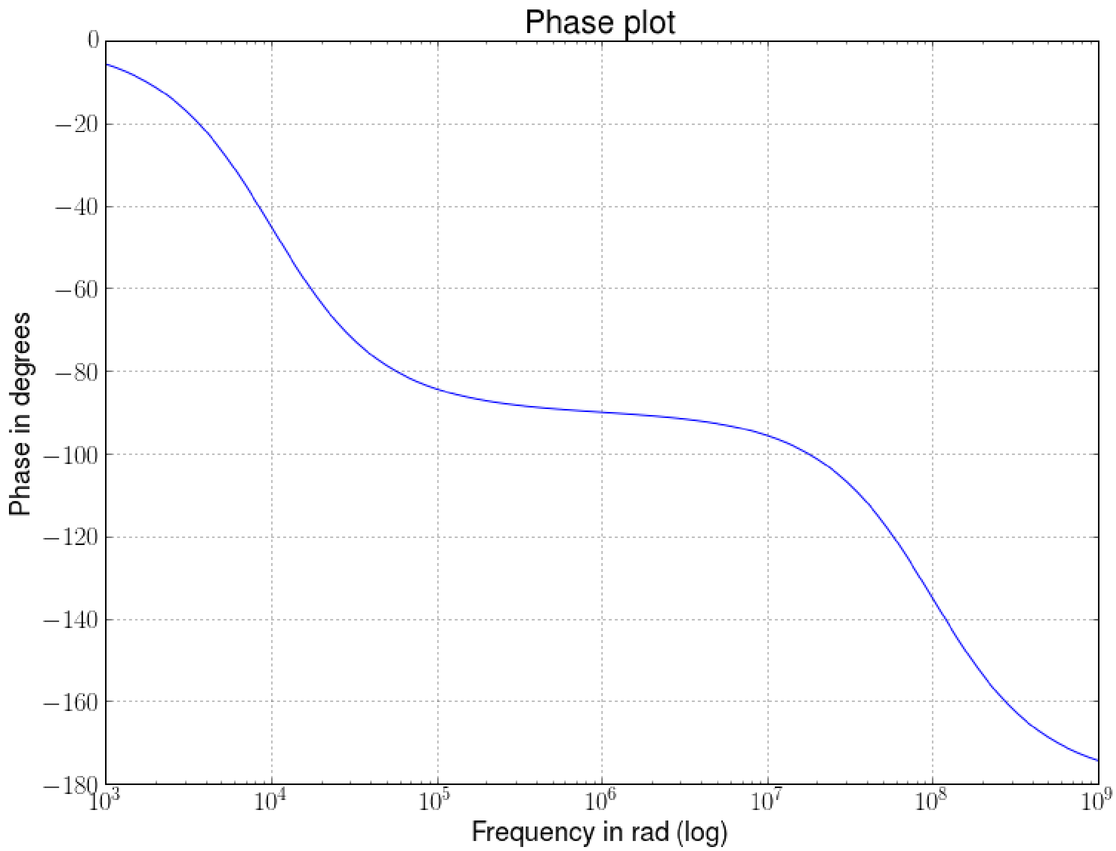
6 Problem 5

We find the transfer function by finding the natural frequency and the damping constant of the circuit.

```
# Find the transfer function of the given circuit
R = 100
L = 1e-6
C = 1e-6

wn = 1/sqrt(L*C) # natural frequency
Q = 1/R * sqrt(L/C) # quality factor
zeta = 1/(2*Q) # damping constant

# transfer function
n,d = secondOrderH(gain=1,wn=wn,zeta=zeta)

# make system
H = sp.lti(n,d)

# get bode plots
w,S,phi=H.bode()
```

We plot the magnitude and phase of the transfer function on log plots:


Magnitude plot
0
−20
−40
−60
−80
−100
−120
−140
Magnitude in dB
10^3
10^4
10^5
10^6
10^7
10^8
10^9
Frequency in rad (log)



- It is clear that there are two poles, one at around 10^4 rad/s and another at around 10^8 rad/s.
- Since the poles are not coincident, this means that the system is overdamped, with a quality factor < 1. The quality factor can be estimated using

$$Q = \frac{\omega_{p1}}{\omega_n}$$

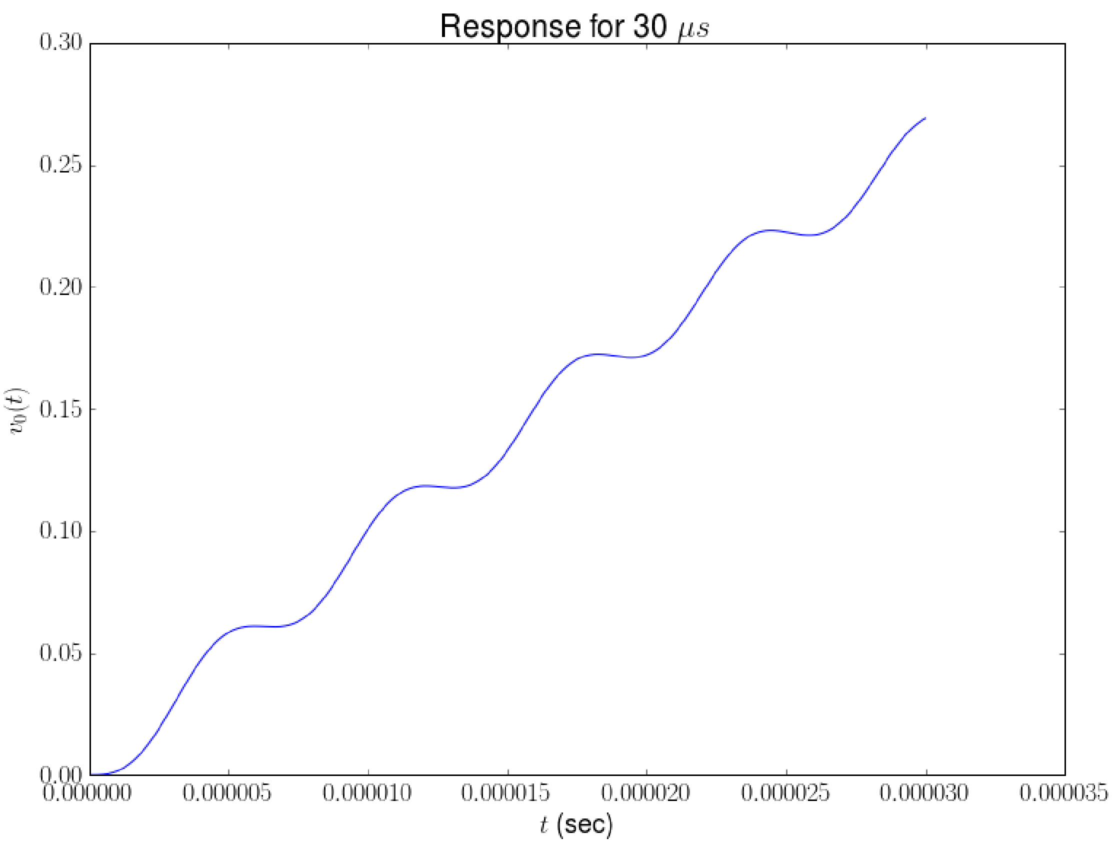
  where ω_{p1} is the first pole. We get $Q \approx 0.01$
- Since the system has no zeros, it acts as an all-pole low pass filter.
- Since the two poles are quite far apart, we can approximate the 3-dB bandwidth of the filter to be at the first pole, i.e., 10^4 rad/s.
- We therefore expect the system to pass frequencies lower than 10^4 rad/s and attenuate higher frequencies. We see this effect in the next part.

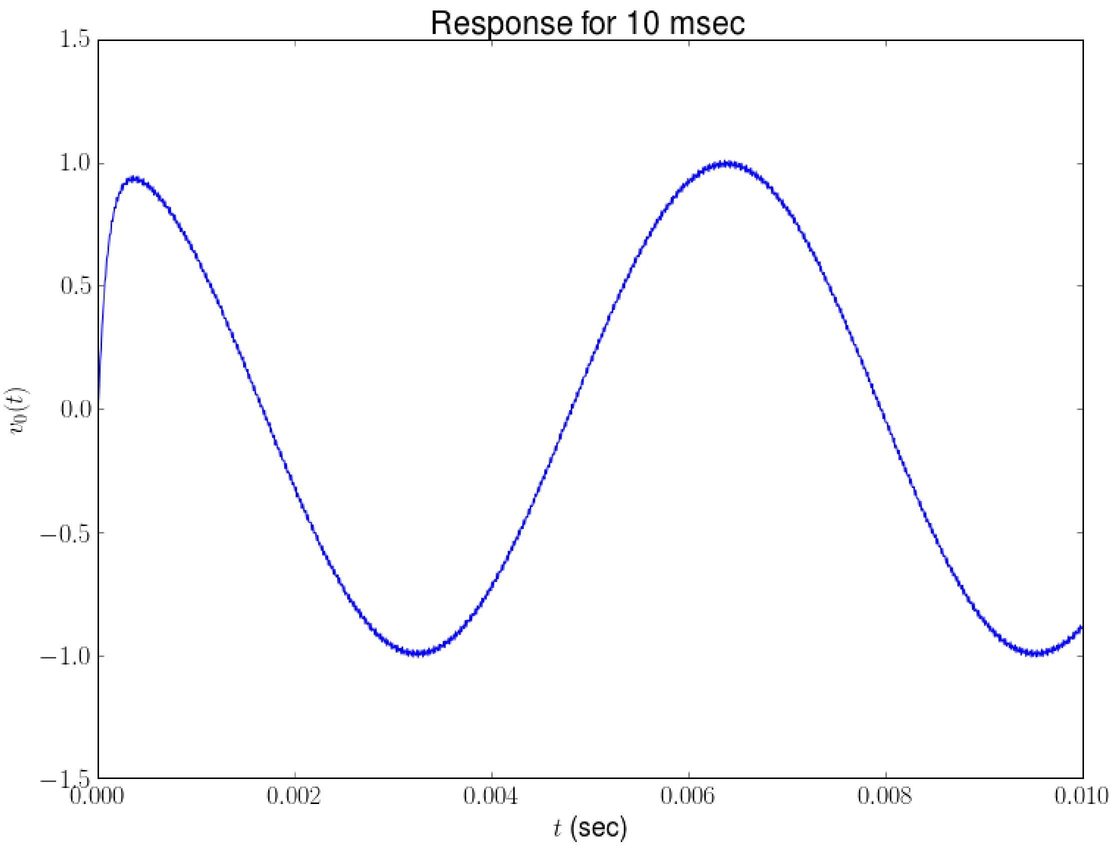
7 Question 6

We excite the system in Question 5 with two sinusoids, one whose frequency is below the 3-dB bandwith and one whose frequency is higher.

```python
def input_2(t,w1=1e3,w2=1e6):
    """Two cosines of different frequencies"""
    u_t = 1*(t>0)
    return (cos(w1*t)-cos(w2*t)) * u_t

# early response
t1=linspace(0,30e-6,1e3)
t1,y1,svec = sp.lsim(H,input_2(t1),t1)

# steady state response
t2=linspace(0,10e-3,1e3)
t2,y2,svec = sp.lsim(H,input_2(t2),t2)
```

We plot the time domain response in two parts, one for the first 30 μs, to observe transient effects, and one for 10 msec, to observe the steady state response.





8 Conclusions

- The transient response of the system is rapidly increasing. This is because the system has to charge up to match the input amplitude. This results in a phase difference between the input and the output. This can also be interpreted as a delay between the input and the output signals.
- The response can be broken up into a low frequency component and a high frequency one, after the transient effect has died down.
- The high frequency component is extremely attenuated (by -40 dB infact), so in the 10 msec plot, it is almost not visible.
- The low frequency component passes through almost unaffected with an amplitude of slightly less than 1. This is because its frequency is below the 3-dB bandwidth of the system.
- Thus, it is clear that the system behaves like a low-pass filter.