Assignment 7

Om Shri Prasath (EE17B113)

March 17, 2019

1 Introduction

- We analyse the LTI systems in continuous time using Laplace transform to find the solutions to the equations governing the system with the help of python tools such as Symbolic python known as *Sympy* and Signal toolbox.

- *symbols* $\rightarrow$ used to declare symbols using sympy.
- $lambdify$ $\rightarrow$ Converts the sympy expression into one numpy expression which can be evaluated.
- *system.impulse* $\rightarrow$ Computes the impulse response of the transfer function
- *sp.lti* $\rightarrow$ defines a transfer function from polynomial coefficients of numerator and denominator as inputs.
- In this assignment we solve all equations analytically using expressions with variables like we use to solve manually with variables which is unlike the way we generally follow to solve i.e by solving for some specific cases (Numerically)) rather than solving for general case with variables.
- So we use Symbolic python to do the above mentioned and we analyse the circuits by solving them analytically and analyse the systems finally using numpy.

2 Python Code

2.1 Question 1:

- We analyse a Low pass filter circuit given below using symbolic python and numpy.
- Observe and analyse the responses of the systems for various inputs.

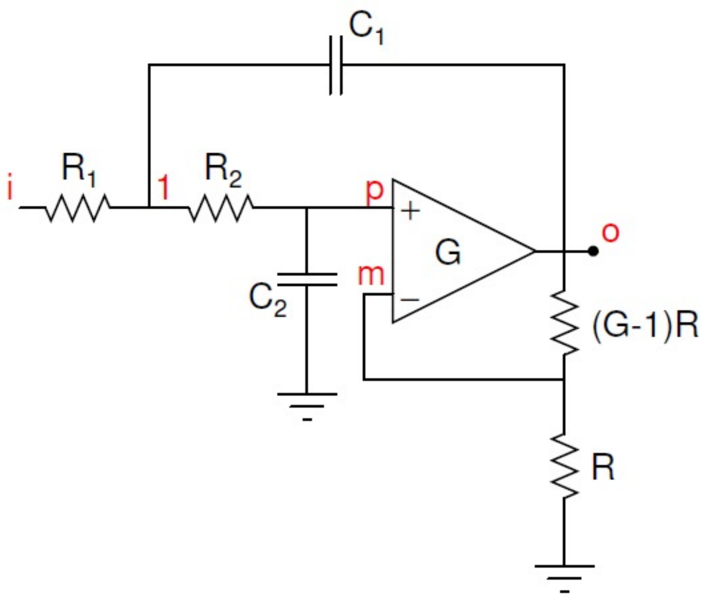


Figure 1: Active Lowpass Filter circuit

- Using sympy we can represent the nodal equations of the ciruit in the form of matrix and solve it to find Vo(t).

$$\begin{pmatrix} 0 & 0 & 1 & -\frac{1}{G} \\ -\frac{1}{1+sR2C2} & 1 & 0 & 0 \\ 0 & -G & G & 1 \\ -\frac{1}{R_1}-\frac{1}{R_2}-sC_1 & \frac{1}{R2} & 0 & sC_1 \end{pmatrix} \begin{pmatrix} V_1 \\ V_p \\ V_m \\ V_o \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -V_i(s)/R_1 \end{pmatrix}$$

- Obtain the Transfer function of the network, which is determined by finding laplace transform of impulse response ($V_i(t) = \delta(t)$ whose $Laplace\ Transform$ is $\mathcal{V}_i(s) = \frac{1}{s}$).
- From transfer function obtain the Quality factor of the system, which essentially says how sharp the system is certain range frequencies.

- If $Q < 0.5$ system is overdamped since damping factor $\zeta = \frac{1}{2Q} > 1$ for $Q < 0.5$ which means the unit step response will raise slowly from 0 to V_{max} exponentially unlike immediately changing from 0 $to V_{max}$.
- To observe this behaviour we give unit step as input and analyse the output.
- Obtain unit step response of the system i.e

$$V_i(t) = u(t)\ Volts \tag{1}$$

$$\mathcal{V}_i(s) = \frac{1}{s} \tag{2}$$

- Obtain and analyse the response for sinusoid with a low frequency and high frequency component of $\omega_1 = 2000\pi\ rads^{-1}\ and\ \omega_2 = 2 * 10^6\pi\ rads^{-1}$.

$$V_i(t) = (\ \sin(2000\pi t) + \cos(2 * 10^6\pi t)\ )u_o(t)\ Volts \tag{3}$$

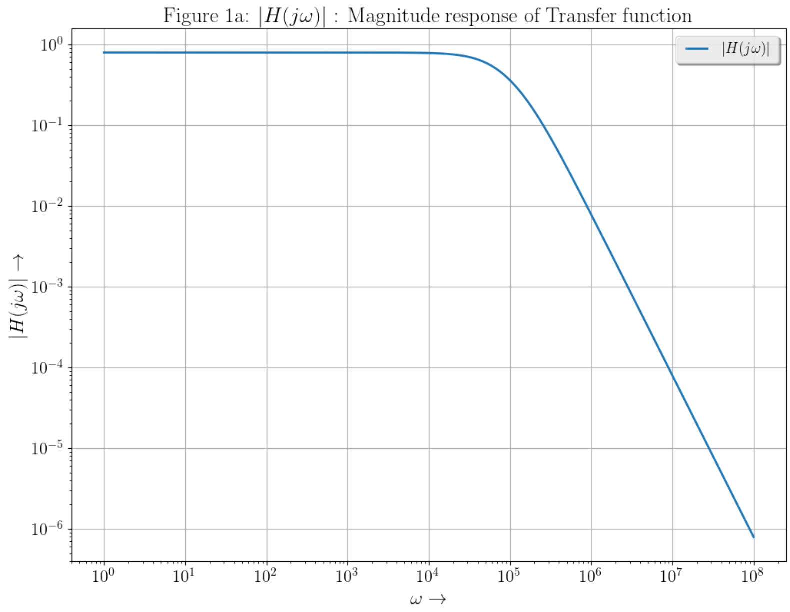
- *sp.lsim* is used for finding the output in time domain.
- Determine and Plot the output voltage $V_o(t)$ for the cases above and analyse them.

Code:

```
def lowpass(R1, R2, C1, C2, G, Vi):
  s = symbols('s')
  A = Matrix([[0, 0, 1, -1/G], [-1/(1+s*R2*C2), 1, 0, 0],
              [0, -G, G, 1], [-1/R1-1/R2-s*C1, 1/R2, 0, s*C1]])
  b = Matrix([0, 0, 0, Vi/R1])
  V = A.inv()*b
  return (A, b, V)

def sympytolti(n_coeff, d_coeff):
  n_coeff = np.array(n_coeff, dtype=float)
  d_coeff = np.array(d_coeff, dtype=float)
  H_n = poly1d(n_coeff)
  H_d = poly1d(d_coeff)
  Hs = sp.lti(H_n, H_d)
  return Hs
```

```
def solve_circuit(R1, R2, C1, C2, G, Vi, input_freqresponse):
    s = symbols('s')
    A, b, V = input_freqresponse(R1, R2, C1, C2, G, Vi)
    Vo = V[3]
    num, den = fraction(simplify(Vo))
    num_coeffs = Poly(num, s).all_coeffs()
    den_coeffs = Poly(den, s).all_coeffs()
    Vlti = sympytolti(num_coeffs, den_coeffs)
    w = logspace(0, 8, 801)
    ss = 1j*w
    hf = lambdify(s, Vo, "numpy")
    v = hf(ss)

    # Calculating Quality factor for the system
    if(Vi == 1):
        # Vi(s)=1 means input is impulse
        Q = sqrt(1/(pow(den_coeffs[1]/den_coeffs[2],
                        2) / (den_coeffs[0]/den_coeffs[2])))
        print("Quality factor of the system : %g" % (Q))
        return v, Vlti, Q
    else:
        return v, Vlti


# Declaring params of the circuit1
R1 = 10000
R2 = 10000
C1 = 1e-9
C2 = 1e-9
G = 1.586
# w is x axis of bode plot
s = symbols('s')
w = logspace(0, 8, 801)
Vi_1 = 1  # Laplace transform of impulse
Vi_2 = 1/s  # Laplace transform of u(t)

# Impulse response of the circuit
Vo1, Vs1, Q = solve_circuit(R1, R2, C1, C2, G, Vi_1, lowpass)
# To find Output Voltage in time domain
t1, Vot1 = sp.impulse(Vs1, None, linspace(0, 1e-2, 10000))
# Step response of circuit
Vo2, Vs2 = solve_circuit(R1, R2, C1, C2, G, Vi_2, lowpass)
# To find Output Voltage in time domain
t2, Vot2 = sp.impulse(Vs2, None, linspace(0, 1e-3, 10000))
```

```
# Magnitude response for impulse (in loglog)
loglog(w, abs(Vo1),
        label=r"$|H(j\omega)|$")
title(r"Figure 1a: $|H(j\omega)|$ : Magnitude response
of Transfer function")
legend()
xlabel(r"$\omega \to $")
ylabel(r"$ |H(j\omega)| \to $")
grid()
show()

# Plot of output for step input
step([t2[0], t2[-1]], [0, 1], label=r"$V_{i}(t) = u(t)$")
plt.plot(t2, abs(Vot2), label=r"Response for $V_{i}(t) = u(t)$")
legend()
title(r"Figure 1b: $V_{o}(t)$ : Unit Step response in time domain")
xlabel(r"$t \to $")
ylabel(r"$ V_{o}(t) \to $")
grid()
show()
```



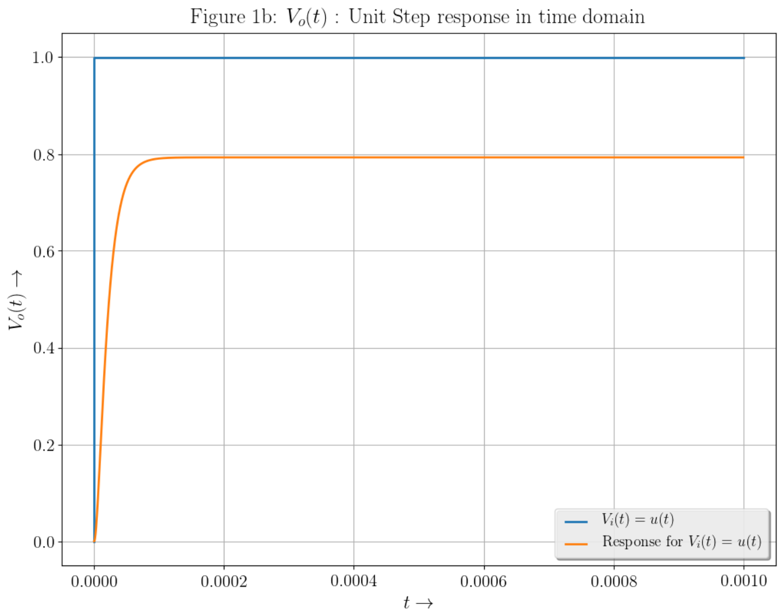


Figure 2: Plots of impulse response(Bode plot) and step response of circuit

2.1.1 Results and Discussion:

- As we observe the Bode plot and the circuit that we know it is a low pass filter with bandwidth
$0<\omega<10^4$.

- So the circuit will only pass input with frequencies which are in range of bandwidth only and attenuates other frequencies largely since its second order filter with -40dB/dec drop in gain

- As we observe the step response plot that $V_o(t)$ increases quickly from 0 to 0.8 and settles at 0.8 for after some time and remains constant.

- Because since the network is lowpass filter, the output must be dominated by DC gain at steady state.

- And we determined Quality factor of the system as $Q = 0.453.. < \frac{1}{\sqrt{2}}$ Which implies that the gain of the system never exceeds DC Gain and always less than that. This observation comes by analysing the general form of second order transfer function.

- So with this we see that unit step response is always less that the DC Gain of 0.8 which is obtained by putting $s = 0$ in the $\mathcal{V}_o(s)$.

- Also If $Q < 0.5$ system is overdamped since damping factor $\zeta = \frac{1}{2Q} > 1$ for $Q < 0.5$ which means the unit step response will raise slowly from 0 to V_{max} exponentially unlike immediately changing from 0 toV_{max}

- So this is also observed in the plot as it slowly raises from 0 to 0.8 and settles.

2.2 Question 2:

- Now we give different input to the same circuit given above : $V_i(t)$ as follows

$$V_i(t) = (\ \sin(2000\pi t) + \cos(2x10^6 \pi t)\ )u_o(t)\ Volts \tag{4}$$

- Determine the output voltage $V_o(t)$ using sp.lsim.
- Plot $V_o(t)$ and analyse the results.

Code:

```
# Input sinusoid frequencies in rad/s
w1 = 2000*np.pi
w2 = 2*1e6*np.pi

ts = np.linspace(0, 0.005, 8001)

vi = np.sin(w1*ts)+np.cos(w2*ts)
t, Vout, svec = sp.lsim(Vs1, vi, ts)

#Plotting the output for sinusoid
plt.plot(ts, Vout, label=r"Response for $V_{i}(t) = $ Sinusoid")
legend()
title(r"Figure 2: $V_{o}(t)$ : Output Voltage for sinusoidal input")

xlabel(r"$t \to $")
ylabel(r"$ V_{o}(t) \to $")
plt.ylim([-1.1, 1.1])
grid()
show()
```

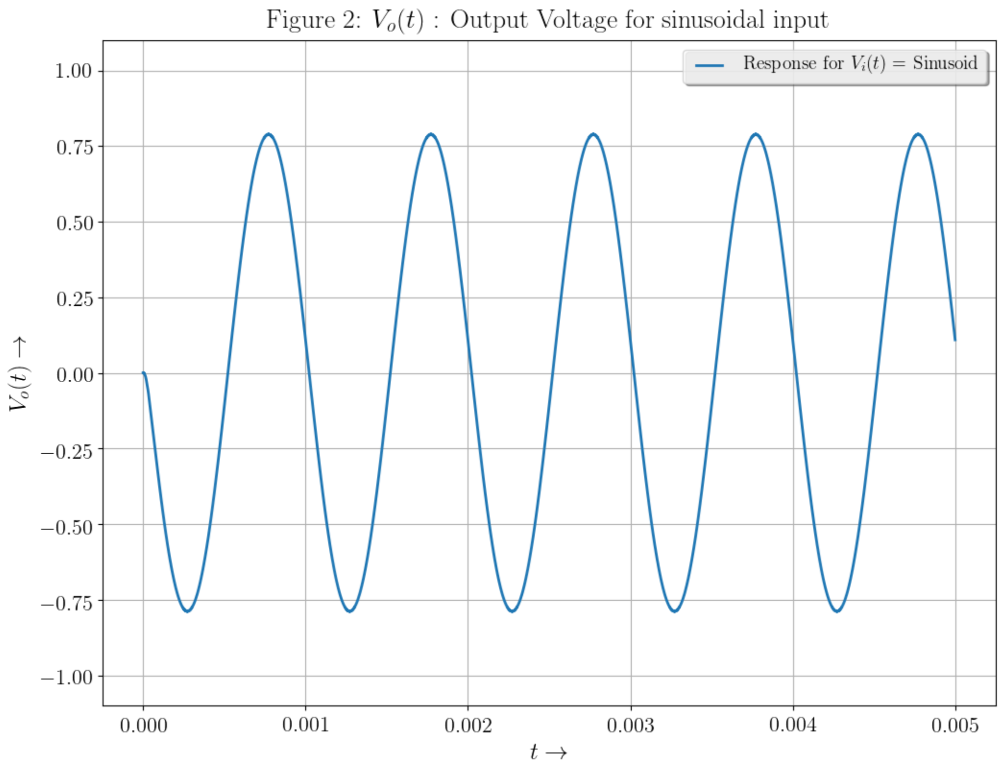


Figure 3: Plots of sinusoid response

2.2.1 Results and Discussion:

- From the plot, we can see that the circuit will only pass input with frequencies which are in range of its 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.
- So the output $V_o(t)$ will be mainly of $\sin(2000\pi t)$ with higher frequencies attenuated largely since its second order filter so gain drops 40dB/dec.
- Since its sin function, we can observe that $V_o(t)$ starts from 0

3 Question 3 ,4 & 5:

- Now we analyse a High pass filter circuit given below using symbolic python.
- We will observe the responses of the systems for various inputs.
- Using sympy we can represent the nodal equations of the ciruit in the form of matrix and solve it to find Vo(t).

$$\begin{pmatrix} 0 & 0 & 1 & -\frac{1}{G} \\ -\frac{-sR_3C_2}{1+sR_3C_2} & 1 & 0 & 0 \\ 0 & -G & G & 1 \\ -1-(sR_1C_1)-(sR_3C_2)) & sC_2R_1 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_1 \\ V_p \\ V_m \\ V_o \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -V_i(s)sR_1C_1 \end{pmatrix}$$

- $R_1 = 10k$,$R_2 = 10k,\ C1 = C2 = 10^{-9}F$

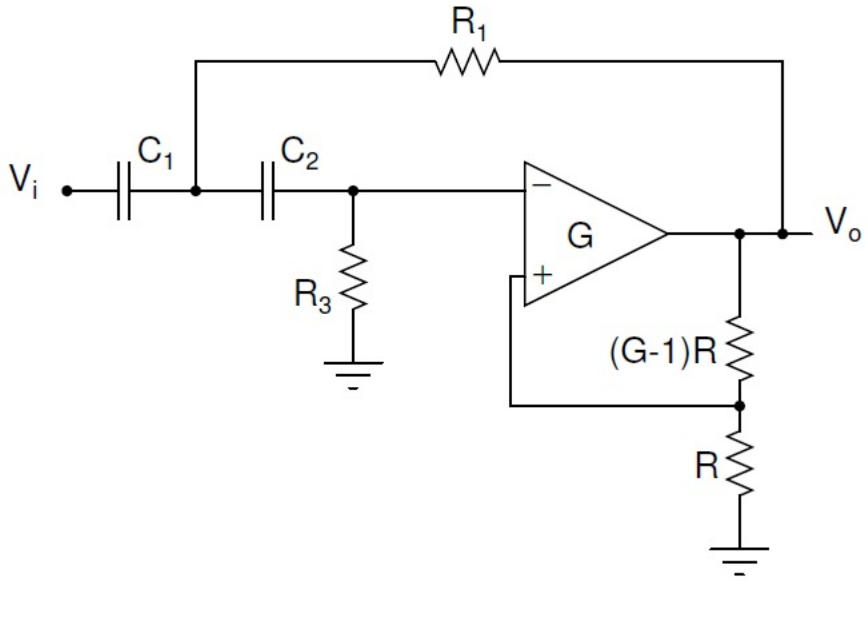


Figure 4: Active Lowpass Filter circuit

- Obtain the Transfer function of the network, which is determined by finding laplace transform of impulse response($V_i(t) = \delta t$).
- Obtain and analyse the response for undamped and damped sinusoids.

$$V_i(t) = (\ \sin(2000\pi t) + \cos(2x10^6\pi t)\ )u_o(t)\ Volts \tag{5}$$

$$V_i(t) = e^{-10^5 t}(\ \sin(2000\pi t) + \cos(2x10^6\pi t)\ )u_o(t)\ Volts \tag{6}$$

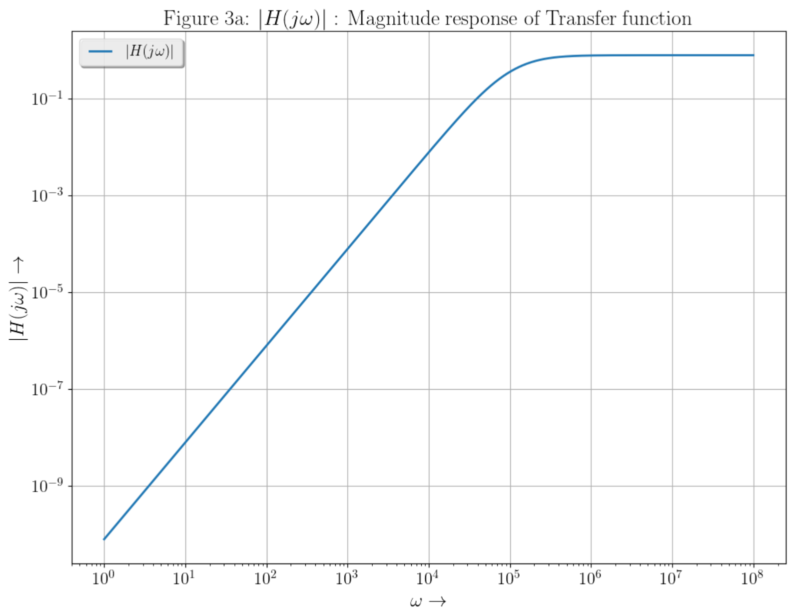
- *sp.lsim* is used for finding the output in time domain
- Determine and Plot the output voltage $V_o(t)$ for both the cases above and analyse them.
- Using results obtained from this network and previous network compare them and analyse the differences.

Code:

```
def highpass(R1, R3, C1, C2, G, Vi):
s = symbols('s')
A = Matrix([[0, 0, 1, -1/G],
            [(-s)*C2*R3/(1+s*R3*C2), 1, 0, 0],
            [0, -G, G, 1],
            [(-1-(s*R1*C1)-(s*R3*C2)), s*C2*R1, 0, 1]])
b = Matrix([0, 0, 0, -Vi*s*C1*R1])
V = A.inv()*b
return (A, b, V)

R1b = 10000
R3b = 10000
C1b = 1e-9
C2b = 1e-9
Gb = 1.586
# input frequencies for damped sinusoids
w1 = 2000*np.pi
w2 = 2e6*np.pi
# Decay factor for damped sinusoid
a = 1e5

# Laplace transform of impulse
Vi_1b = 1
# Laplace of unit step
Vi_2b = 1/s

# Solve for step response
Vo1b, Vs1b, Qb = solve_circuit(R1b, R3b, C1b, C2b, Gb, Vi_1b, highpass)
t1b, Vot1b = sp.impulse(Vs1b, None, linspace(0, 1e-2, 10000))
```

```
# Solve for impulse response
Vo2b, Vs2b = solve_circuit(R1b, R3b, C1b, C2b, Gb, Vi_2b, highpass)
t2b, Vot2b = sp.impulse(Vs2b, None, linspace(0, 5e-4, 1000001))

ts1 = np.linspace(0, 0.000005, 800001)
ts2 = np.linspace(0, 0.00003, 80001)

V_i3b = np.sin(w1*ts1)+np.cos(w2*ts1)

V_i4b = np.exp(-a*ts2)*(np.sin(w1*ts2)+np.cos(w2*ts2))

t3b, Vot3b, svec = sp.lsim(Vs1b, V_i3b, ts1)

t4b, Vot4b, svec = sp.lsim(Vs1b, V_i4b, ts2)

loglog(w, abs(Vo1b), label=r"$|H(j\omega)|$")
legend()
title(r"Figure 3a: $|H(j\omega)|$ : Magnitude response of Transfer functi
xlabel(r"$\omega \to $")
ylabel(r"$ |H(j\omega)| \to $")
grid()
show()

step([t2b[0], t2b[-1]], [0, 1], label=r"$V_{i}(t) = u(t)$")
plt.plot(t2b, Vot2b, label=r"Unit Step Response for $V_{i}(t) = u(t)$")
legend()
title(r"Figure 3b: $V_{o}(t) $ : Unit step response in time domain")
xlabel(r"$ t (seconds) \to $")
ylabel(r"$ V_{o}(t) \to $")
grid()
show()

plt.plot(t3b, (Vot3b), label=r"Response for $V_{i}(t) = $ Undamped Sinuso
legend()
title(r"Figure 4: $V_{o}(t)$ : Output Voltage for undamped sinusoid input
xlabel(r"$t \to $")
ylabel(r"$ V_{o}(t) \to $")
plt.ylim([-1.1, 1.1])
grid()
show()

plt.plot(t4b, (Vot4b), label=r"Response for $V_{i}(t) = $ Damped Sinusoid
legend()
title(r"Figure 5: $V_{o}(t)$ : Output Voltage for damped sinusoid input")
```

```
xlabel(r"$t \to $")
ylabel(r"$ V_{o}(t) \to $")
grid()
show()
```



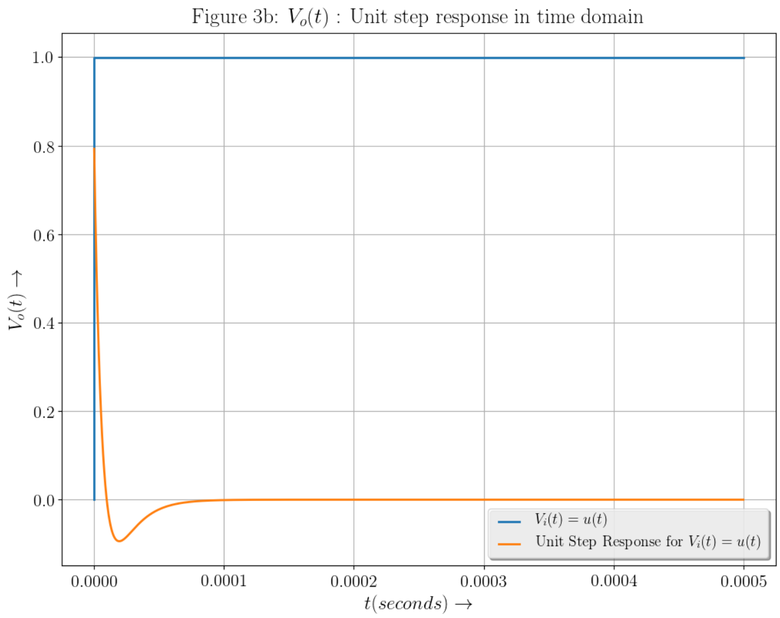


Figure 5: Plots of impulse(Bode plot) and step response

Figure 4: $V_o(t)$: Output Voltage for undamped sinusoid input

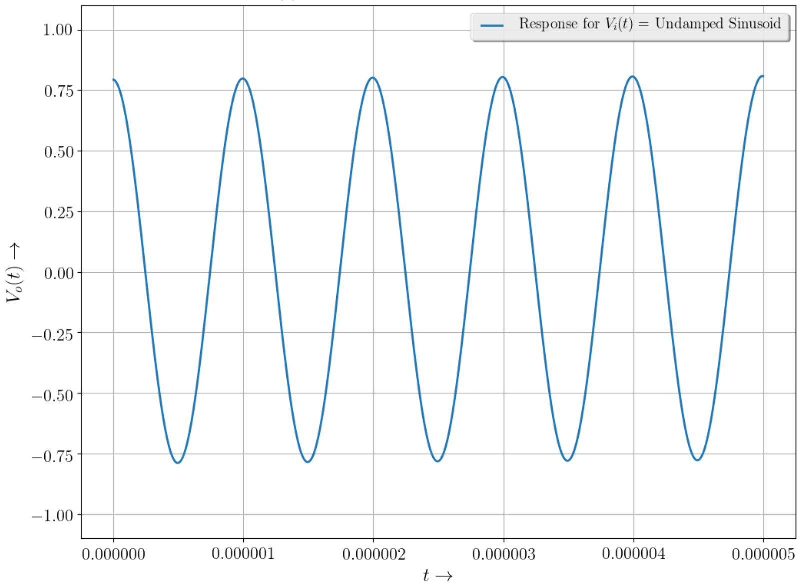


Figure 5: $V_o(t)$: Output Voltage for damped sinusoid input

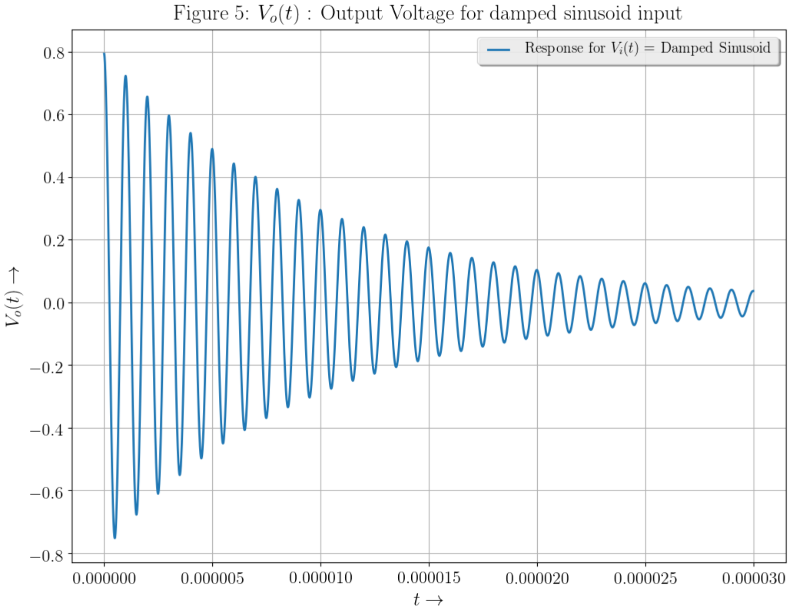


Figure 6: Plots of sinusoid responses

3.0.1 Results and Discussion:

- As we observe the plot and the circuit that we know it is a high pass filter with bandwidth $\omega{>}10^5$.
- So the circuit will only pass input with frequencies which are in range of bandwidth only.
- As we observe the impulse response plot and the circuit that we know it is a high pass filter with bandwidth
$\omega > 10^5$.
- As we observe the step response plot that $V_o(t)$ decreases quickly from 0.8 to 0 and settles at 0 for after some time and remains constant.
- But interestingly it **crosses zero and goes negative** for some period of time.
- Because since the network is high pass filter, the output must not allow DC at steady state and since the input is unit step which is constant for $t > 0$ so at steady state $V_0(t)$ should be zero. Also since its Highpass filter its $Mean = 0$ which means $\int_0^\infty V_o(t)dt = 0$.
- So thats why we observe that the voltage becomes negative to make averge zero.
- And we determined Quality factor of the system as $Q = 0.453.. < \frac{1}{\sqrt{2}}$ Which implies that the gain of the system never exceeds DC Gain and always less than that. This observation comes by analysing the general form of second order transfer function.
- Also If $Q < 0.5$ system is overdamped since damping factor $\zeta = \frac{1}{2Q} > 1$ for $Q < 0.5$ which means the unit step response will decrease slowly from 0.8 to 0 exponentially since its high pass filter and steady state voltage must be zero ,unlike immediately changing from 0.8 $to0$
- So this is also observed in the plot as it slowly decays from 0.8 to 0 and settles.
- Also, we can see from the sinusoidal plots that only the frequency component more than 10^4 rad/s are passed through, thus we see both the damped and non-damped sinusoids have frequency of nearly 10^6 rad/s, i.e. the cosine component

4 Conclusion :

- We successfully analysed circuits using laplace transform by solving analytically instead of numerical solutions using Symbolic python.