

EE2703 : Applied Programming Lab
Assignment 7
Circuit Analysis using Sympy

Chagari Koushal Kumar Reddy
EE20B023

March 25, 2022

Contents

1	Aim	3
2	Theory: Active RC Filters	3
2.1	Butterworth Low Pass Filter	3
2.2	Butterworth High Pass Filter	4
2.3	Solving for the transfer function	5
3	Functions defined in Code	6
3.1	Function to generate the LPF transfer function	6
3.2	Function to generate the HPF transfer function	6
3.3	Function to convert transfer function expression to LTI form of signal toolbox .	6
3.4	Function to generate damped sinusoid responses	7
4	Observations and Plots	7
4.1	Magnitude Response Plots	7
4.2	Step Response Plots	8
4.3	Sum of Sinusoids Response Plots	10
4.4	Damped Sinusoid Signals	11
4.5	Damped Sinusoid Signal Response Plots	12
5	Conclusions	14

1 Aim

The aim to this assignment is to:

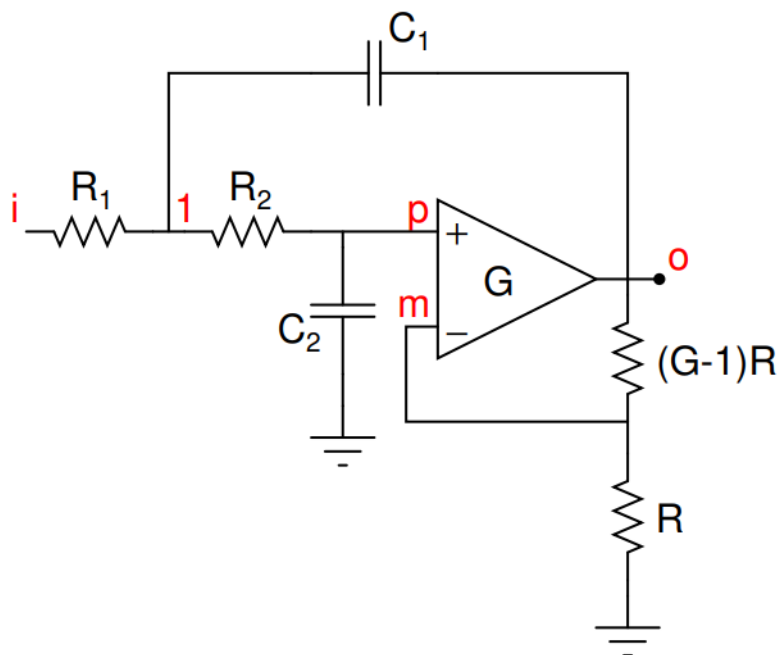
1. Get started with some basic SymPy (Symbolic python) commands
2. Solve for the output response of given active Low pass and High pass filters using SymPy and Scipy.signal toolbox
3. Analyze the frequency components in output responses

2 Theory: Active RC Filters

Active RC filters are electronic devices which are purely made using active operational amplifiers (OpAmps), resistors and capacitors. These filters help us to filter out certain frequency components present in an electric signal which may not be desirable for particular application. They are used in various applications such as noise reduction, radio tuning, selective amplification etc. In this assignment, two types of active RC filters are analyzed: Butterworth Low Pass Filter and Butterworth High Pass Filter.

2.1 Butterworth Low Pass Filter

A Low Pass Filter (LPF) is an electrical filter which allows only the low frequency components of an electric signal and attenuates the high frequency components. The electronic used to realize a Butterworth LPF is shown below:



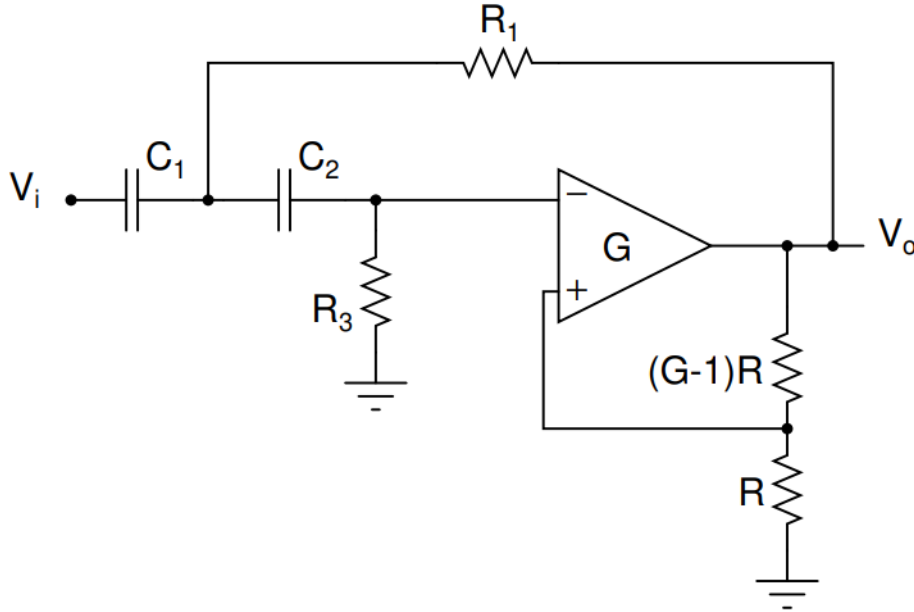
We need to solve for the transfer function $H(s) = \frac{V_o(s)}{V_i(s)}$. To do this, we need to solve for the node voltages using nodal analysis. Also, we know that in the laplace domain, a capacitor acts like resistor with its resistance as $R = \frac{1}{sC}$. With these things in mind, the nodal equations in matrix form are given by:

$$\begin{bmatrix} 0 & 0 & 1 & \frac{-1}{G} \\ -\frac{1}{1+sR_2C_2} & 1 & 0 & 0 \\ 0 & -G & G & 1 \\ -\frac{1}{R_1} - \frac{1}{R_2} - sC_1 & \frac{1}{R_2} & 0 & sC_1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_p \\ V_m \\ V_o \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{V_i(s)}{R_1} \end{bmatrix}$$

Clearly the source vector on the RHS contains only one non-zero term dependent on V_i . Hence all the other node voltages would be some multiple of it. We need the multiple relating V_o and that multiple is the transfer function $H(s)$.

2.2 Butterworth High Pass Filter

A high pass filter (HPF) is an electrical filter which allows only the high frequency components of an electric signal and attenuates the low frequency components. The electronic circuit used to realise a Butterworth HPF is shown below:



The circuit is almost similar to the LPF circuit except that the resistors are replaced by capacitors and vice-versa (Except for feedback resistors). Also, there isn't much change in the structure of the nodal matrix equation. With the understanding that whatever was initially a resistor is now replaced by a capacitor and vice-versa, we can apply the following transform:

$$R \rightarrow \frac{1}{sC}$$

$$sC \rightarrow \frac{1}{R}$$

With these changes in mind, the new matrix equation becomes:

$$\begin{bmatrix} 0 & 0 & 1 & \frac{-1}{G} \\ -\frac{sC_2R_3}{1+sC_2R_3} & 1 & 0 & 0 \\ 0 & -G & G & 1 \\ -\frac{1}{R_1} - sC_1 - sC_2 & sC_2 & 0 & \frac{1}{R_1} \end{bmatrix} \begin{bmatrix} V_1 \\ V_p \\ V_m \\ V_o \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -sC_1V_i(s) \end{bmatrix}$$

Again the output response V_o will be some multiple of V_i and that multiple is the HPF transfer function $H(s)$.

2.3 Solving for the transfer function

As we have seen, the matrix equation is not purely based on numbers. It is also based on mathematical variables. In our case, the variable is 's'. Unlike normal matrix equations, we can't solve this using packages such as numpy. We need to treat the variable 's' as an unknown entity and solve for the transfer function as a function of 's'. To do this, we need the Sympy library of python. Sympy is a special library which allows us to treat certain variables as mathematical/symbolic variables. These variables are different from the usual python variables. They necessarily need not store values but rather store symbols. These symbols don't have any value but can be treated like the variables we use in math (x,y,z) and complex expressions could be formed using these symbols. Finally the value of that expression can be found by providing these symbols some numerical values.

Sympy is very useful when it comes to obtain the expression from primitive inputs. In our filter case too, the primitive inputs are the resistor and capacitor values. However, the final expression is not easily derivable and hence, we can't write a user-defined python function without knowing the final expression. Sympy allows us to find the final expression while still treating the variable 's' as a mathematical variable.

```

1  import sympy as sp
2  s = sp.symbols('s')
3
4  def lowpass(R1,R2,C1,C2,G,Vi):
5      A = sp.Matrix([ [0,0,1,-1/G], [-1/(1+s*R2*C2),1,0,0], [0,-G,G,1], [-(1/R1)
6      )-(1/R2)-s*C1,1/R2,0,s*C1]])
7      b = sp.Matrix([0,0,0,-Vi/R1])
8      V = A.inv()*b
9      Vo = V[3]
10     return Vo

```

For example, in the python function provided above, the inputs are the resistor, capacitor and Opamp gain values. However, the final expression is not directly used to find the transfer function. Instead, the sympy variable 's' is used to denote the variable and the matrix equation is solved. Finally, the expression is obtained and returned out.

3 Functions defined in Code

3.1 Function to generate the LPF transfer function

The following function returns the transfer function of the LPF.

```
1 def lowpass(R1,R2,C1,C2,G,Vi):
2     A = sp.Matrix([ [0,0,1,-1/G], [-1/(1+s*R2*C2),1,0,0], [0,-G,G,1], [-(1/R1)
3     )-(1/R2)-s*C1,1/R2,0,s*C1]])
4     b = sp.Matrix([0,0,0,-Vi/R1])
5     V = A.inv()*b
6     Vo = V[3]
7     return Vo
```

3.2 Function to generate the HPF transfer function

The following function returns the transfer function of the HPF.

```
1 def highpass(R1,R3,C1,C2,G,Vi):
2     A = sp.Matrix([ [0,0,1,-1/G], [-s*C2,(1/R3)+s*C2,0,0], [0,-G,G,1], [s*C1+
3     s*C2+(1/R1),-s*C2,0,-1/R1]])
4     b = sp.Matrix([0,0,0,Vi*s*C1])
5     V = A.inv()*b
6     Vo = V[3]
7     return Vo
```

3.3 Function to convert transfer function expression to LTI form of signal toolbox

This function takes a transfer function expression as input, converts it to the signal toolbox LTI form and returns it.

```
1 def sympy_tf2sig_tf(tf):
2     n,d = tf.as_numer_denom()
3     num = n.as_poly(s).all_coeffs()
4     den = d.as_poly(s).all_coeffs()
5     num = [float(i) for i in num]
6     den = [float(i) for i in den]
7     TF = sig.lti(num,den)
8     return TF
```

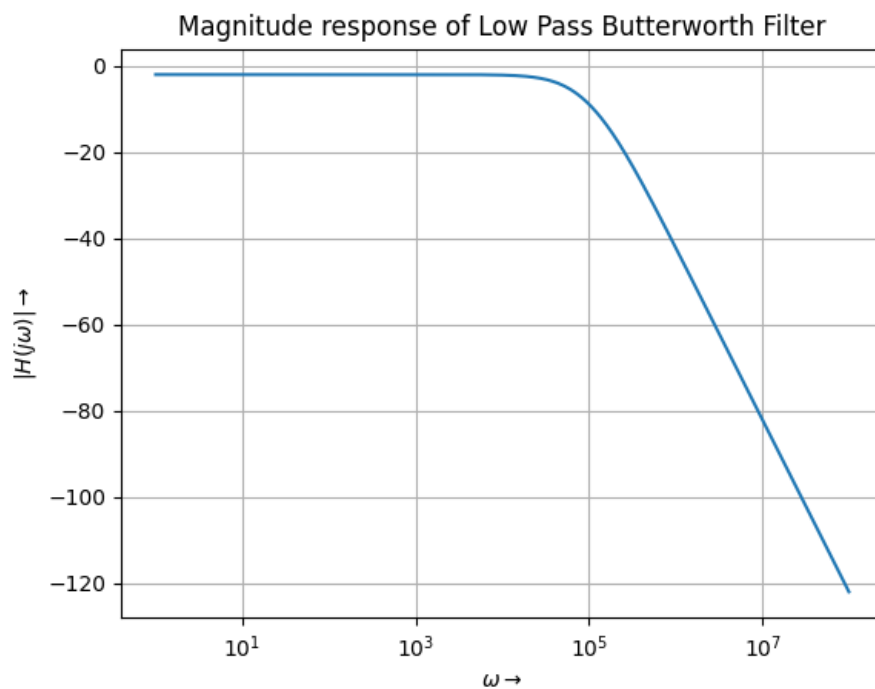
3.4 Function to generate damped sinusoid responses

This function takes the transfer function, damping factor, frequency and time as the inputs. It computes the response and returns it.

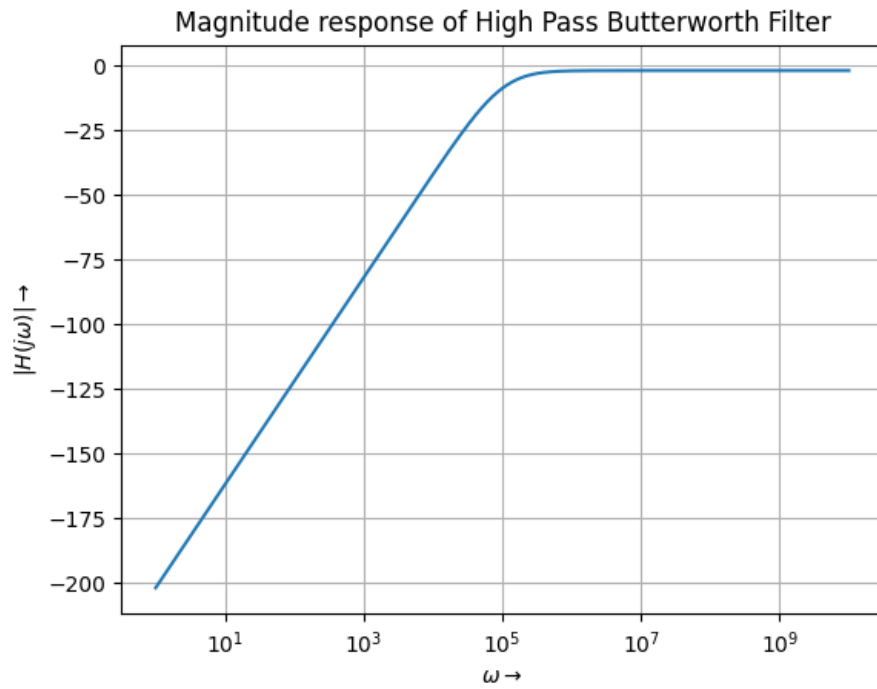
```
1 def damped_sinusoid_response(H,A,B,t):  
2     V_in = np.exp(-A*t)*np.cos(B*t)  
3     V_o = sig.lsim(H,V_in,t)[1]  
4     return V_o
```

4 Observations and Plots

4.1 Magnitude Response Plots

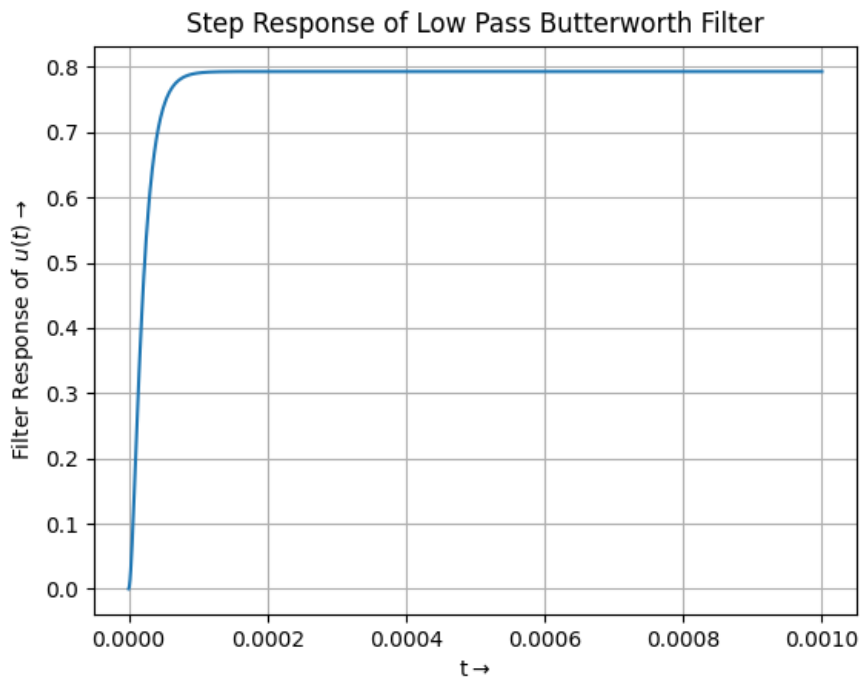


The LPF seems to have a single pole at frequency = 10^5 rad/s

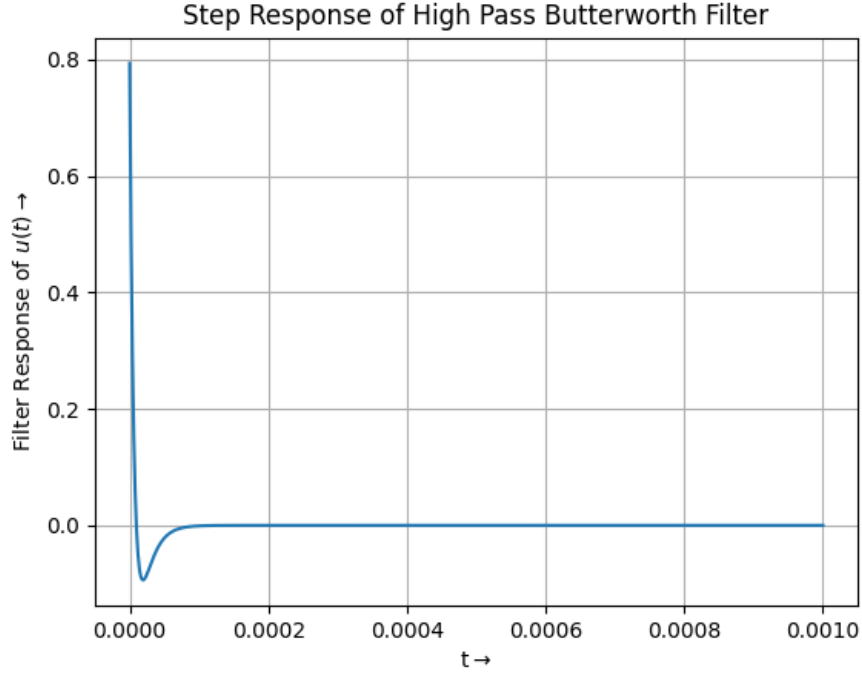


The HPF seems to have DC zero and a single pole at frequency = 10^5 rad/s

4.2 Step Response Plots



The DC gain of the LPF is $\frac{G}{2} = 0.793$. Hence the step response is looking a like a DC signal after some time



The step response of HPF is a DC signal with value close to 0. The initial values cause the transient behaviour.

In this question we are asked whether we understood the response

The reason for the initial peak is the initial conditions. The capacitor does not allow instantaneous changes in the voltage across it. Before $t = 0$, both capacitors have no charge, and the voltage across both is 0, and this will not change instantaneously.

Referring to the circuit diagram, the instant the 1V of the unit step is applied ($t = 0^+$), the drop across the capacitors C_1 and C_2 will still be zero. Therefore $V_m = 1V$.

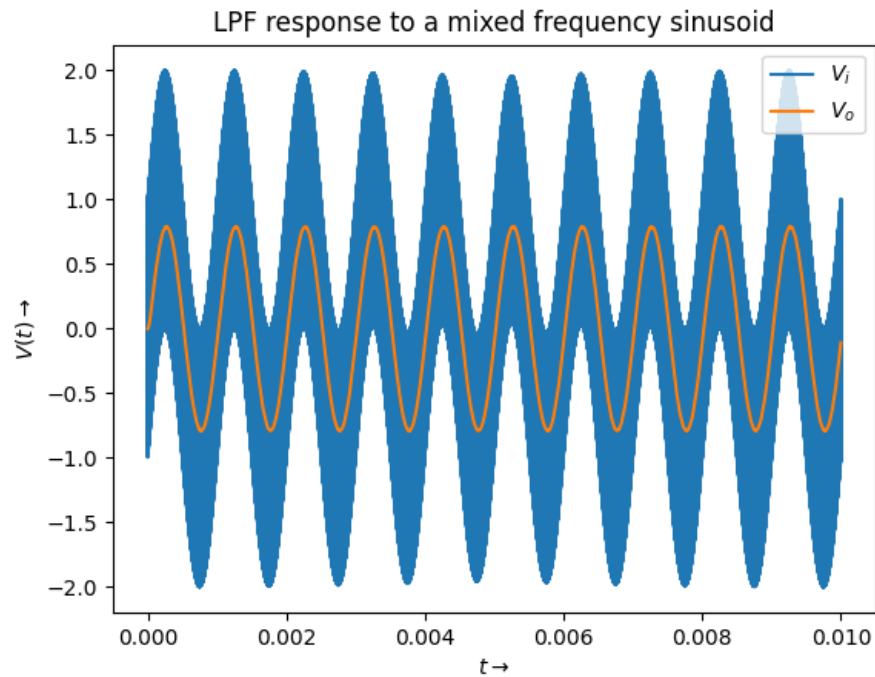
$V_m = \frac{V_o}{G}$ and $V_o = G(V_p - V_m)$, therefore we get:

$$V_o = G\left(1 - \frac{V_o}{G}\right)$$

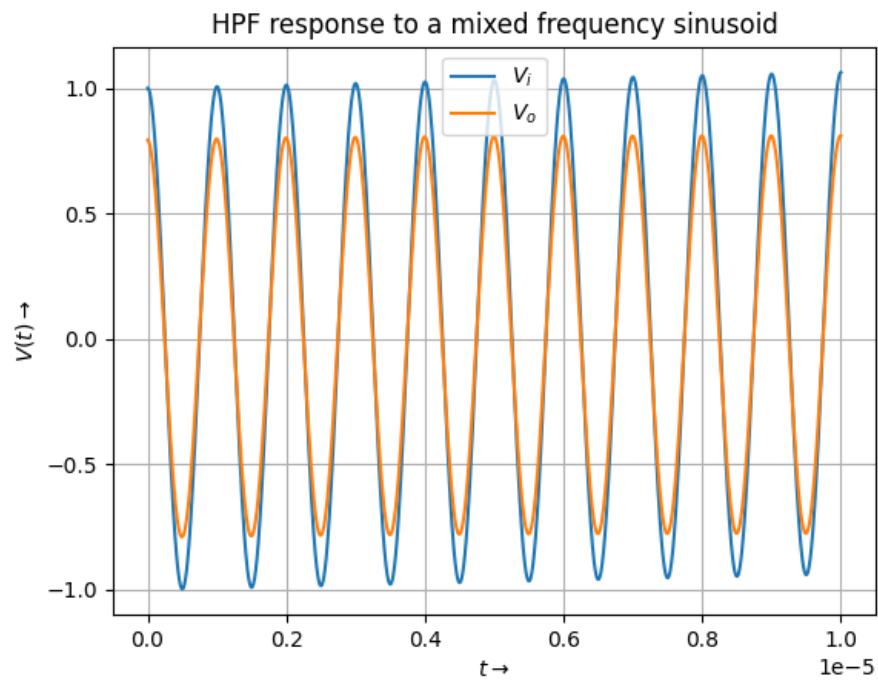
$$V_o = \frac{G}{2} = \frac{1.586}{2} = 0.793V \approx 0.8V$$

And hence, we have an initial peaking around 0.8V and then it quickly decays and settles down to the expected 0V since it is a high pass filter and DC output should be 0V.

4.3 Sum of Sinusoids Response Plots

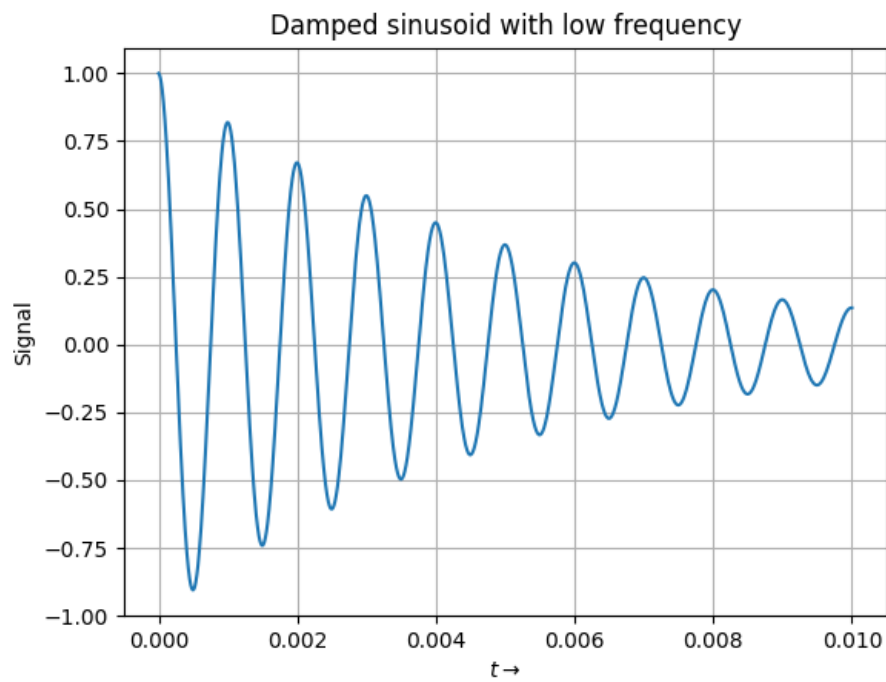


The output response of the sum of sinusoids, when passed through an LPF purely contains only the $f = 1\text{kHz}$ component

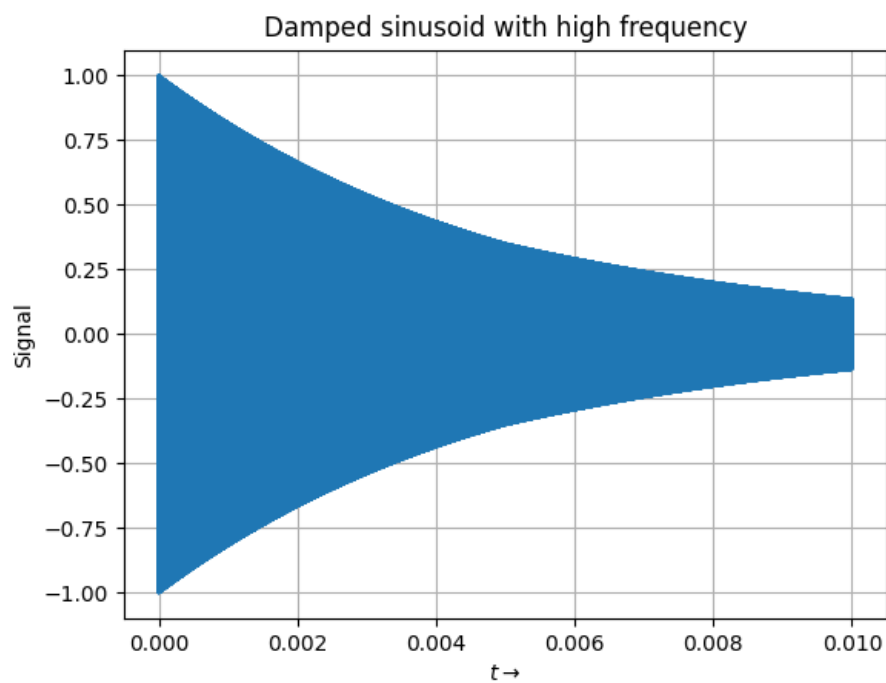


The output response of the sum of sinusoids, when passed through HPF purely contains only the $f = 1\text{MHz}$ component

4.4 Damped Sinusoid Signals

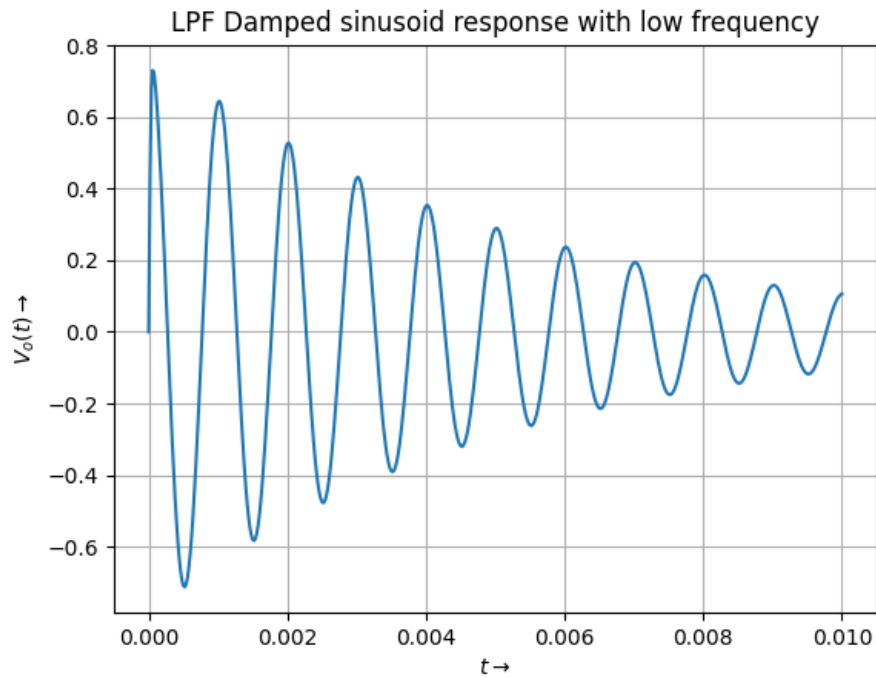


This is a damped sinusoid signal with $f = 1\text{kHz}$ and decay constant $= 200\text{s}^{-1}$

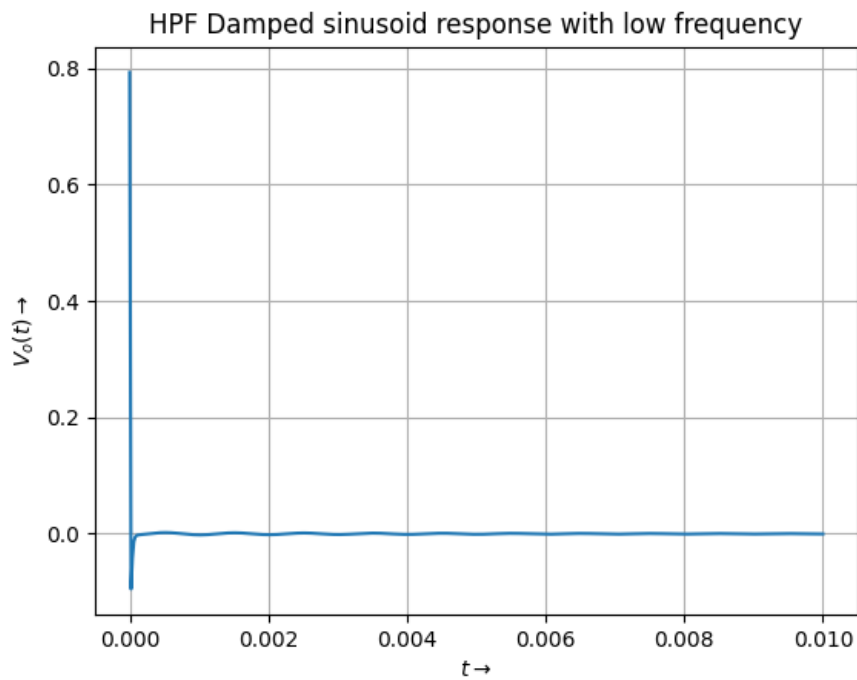


This is a damped sinusoid signal with $f = 1\text{MHz}$ and decay constant $= 200\text{s}^{-1}$

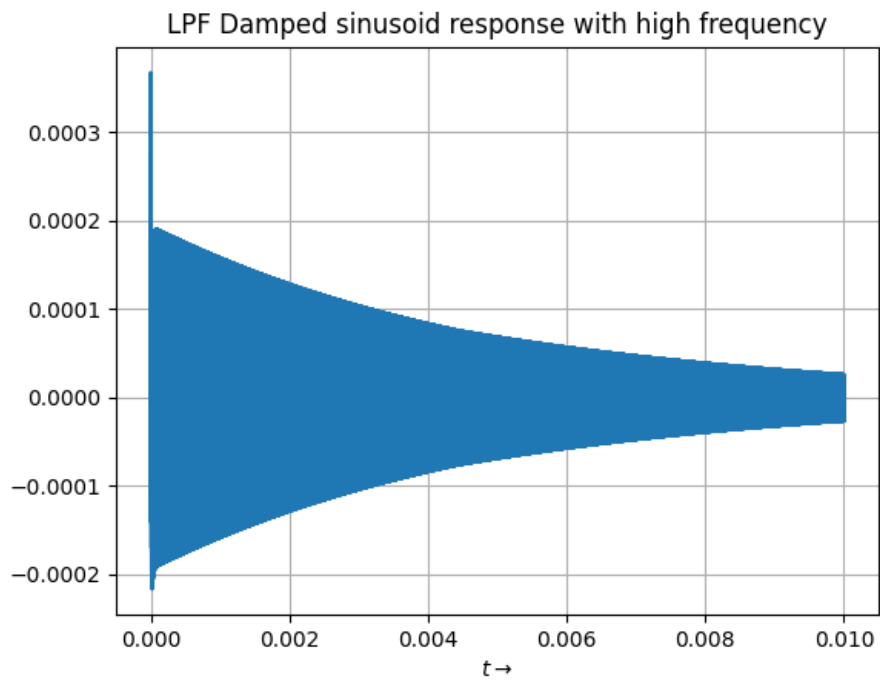
4.5 Damped Sinusoid Signal Response Plots



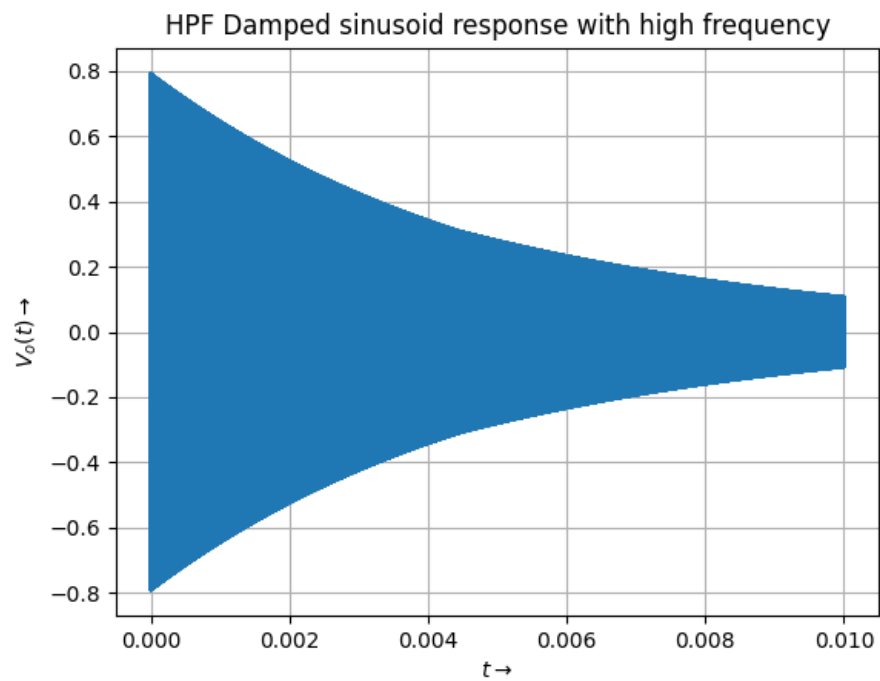
When the low frequency damped sinusoid is passed through an LPF, the output isn't much different from the input. The amplitude has reduced to 0.793



When the low frequency damped sinusoid is passed through a HPF, the output is highly attenuated as seen from the image. The line at $t=0$ is due to initial conditions



When the high frequency damped sinusoid is passed through an LPF, the output is highly attenuated as seen from the image(Y-axis scale is very low)



When high frequency damped sinusoid is passed through a HPF, the output isn't much different from the input. The amplitude has reduced to 0.8

5 Conclusions

1. Both the filters have a pole frequency of 10^5 rad/s and a constant gain of 0.8 in their operating region. This is evident from their magnitude response plots.
2. The step response of the low pass filter starts from zero (due to initial conditions) and stabilises at a DC value of 0.8
3. The step response of the high pass filter starts from 0.8 (due to initial conditions) and stabilises to 0.
4. When the sum of sinusoids signal is passed through the LPF, the 1kHz component gets filtered out with a gain of 0.8.
5. When the sum of sinusoids signal is passed through the HPF, the 1MHz component gets filtered out with a gain of 0.8.
6. When the low frequency damped sinusoid is passed through the LPF, the output is not so much different from the input.
7. When the low frequency damped sinusoid is passed through the HPF, the output is highly attenuated. It starts from 0.8 due to initial conditions.
8. When the high frequency damped sinusoid is passed through the LPF, the output is highly attenuated (Evident from the Y-Axis scale)
9. When the high frequency damped sinusoid is passed through the HPF, the output is not so much different from the input.