

EE2703: Assignment 7

Circuit analysis using Sympy

Nandha Varman
EE19B043

1 Introduction:

In this assignment, we will focus on two powerful capabilities of Python:

- Symbolic Algebra
- Analysis of Circuits using Laplace Transforms

2 Low-Pass Filter:

In this section, we analyse the following circuit:

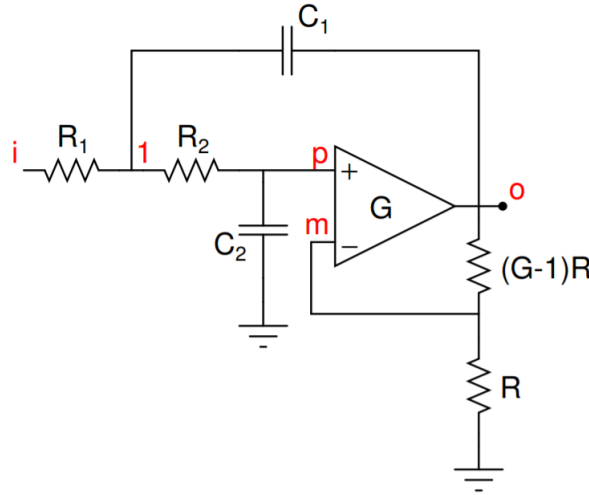


Figure 1: Op-Amp based Low-Pass Filter

Writing KCL equations (in s -domain) of the nodes marked on the figure, we get the following matrix:

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

Sympy allows us to create matrices with symbolic entries, and also perform mathematical operations on them, as if they were **numpy** arrays. In the above circuit, the values of R_1 , R_2 , C_1 , C_2 are $10k\Omega$, $10k\Omega$, 10pF , 10pF respectively.

Solving for $V_o(s)$, (with the above given values) we get:

$$V_o(s) = \frac{-0.0001586 \cdot V_i(s)}{2 \times 10^{-14}s^2 + 4.414 \times 10^{-9}s + 0.0002} \quad (2)$$

From (2), we get the step response of the circuit as:

$$V_o(s) = \frac{-0.0001586 \cdot \frac{1}{s}}{2 \times 10^{-14}s^2 + 4.414 \times 10^{-9}s + 0.0002} \quad (3)$$

Sympy allows us to convert a symbolic expression into functions that are compatible with other packages like `numpy`, `scipy` etc. This can be accomplished by converting the expression into a *lambda*¹ function. For this purpose, `sympy` has a method called *lambdify*, which takes in as arguments, the symbols in the expression (which are to be treated as variables), the expression itself, and also the package with which the resulting function has to be compatible with.

However, since we are required to use the `scipy.signal` toolbox, we have to convert the above the symbolic expression to a format with which we can easily create a `signal.lti` object. For that, we extract the coefficients of the numerator and denominator polynomials of $V_o(s)$ and create a `signal.lti` object using the same. The below piece of code accomplishes this:

```
def sympyToLTI(sym_expr):
    # extracting the numerator and denominator
    num, denom = sym_expr.as_numer_denom()
    # converting numerator and denominator to polynomials of 's'
    num = sym.Poly(num, s)
    denom = sym.Poly(denom, s)
    # extracting the coefficients
    num_coeffs = num.all_coeffs()
    denom_coeffs = denom.all_coeffs()
    # converting the numerator and denominator coefficients to float
    for i in range(len(num_coeffs)):
        x = float(num_coeffs[i])
        num_coeffs[i] = x
    for j in range(len(denom_coeffs)):
        x = float(denom_coeffs[j])
        denom_coeffs[j] = x
    return num_coeffs, denom_coeffs
```

Now, we can easily create a `signal.lti` object with the coefficients we got, and use `signal.bode` to obtain the Bode magnitude and phase plots, which are shown below.

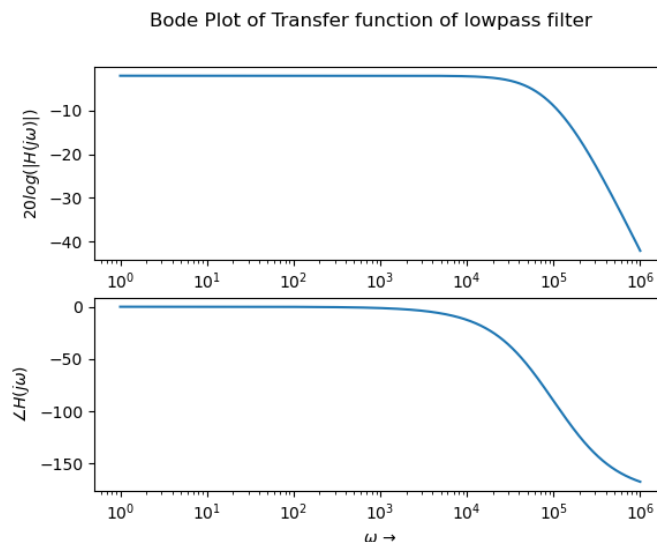


Figure 2: Bode Magnitude and Phase plots of step response of LPF

To see the step response of the system, we can use `signal.step`. The step response of the system is shown below.

¹Lambda functions are, in a nutshell, *one-time use* functions. They are short-lived functions, very useful for quick manipulations, especially inside other functions.

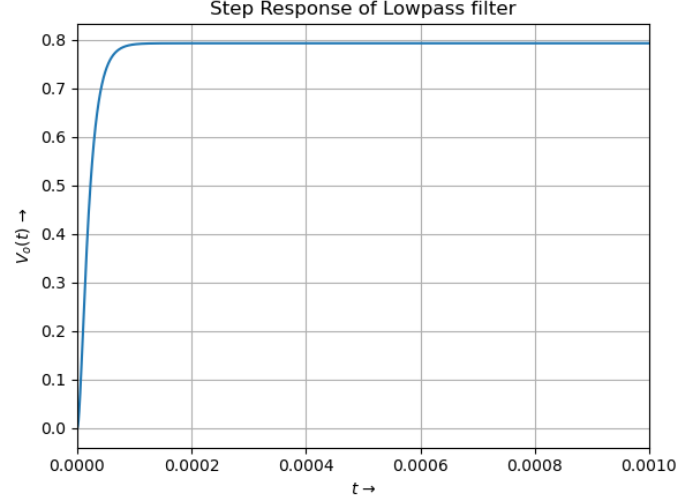


Figure 3: Step response of LPF

Now, we shall see that the circuit is indeed a low-pass filter by plotting the output for a mixed-frequency input, which has both high frequency and low frequency components.

We shall give the following input to the filter:

$$v_i(t) = (\sin(2\pi \times 10^3 t) + \cos(2\pi \times 10^6 t))u(t) \quad (4)$$

We shall use `signal.lsim` to calculate the time-domain response of the system.

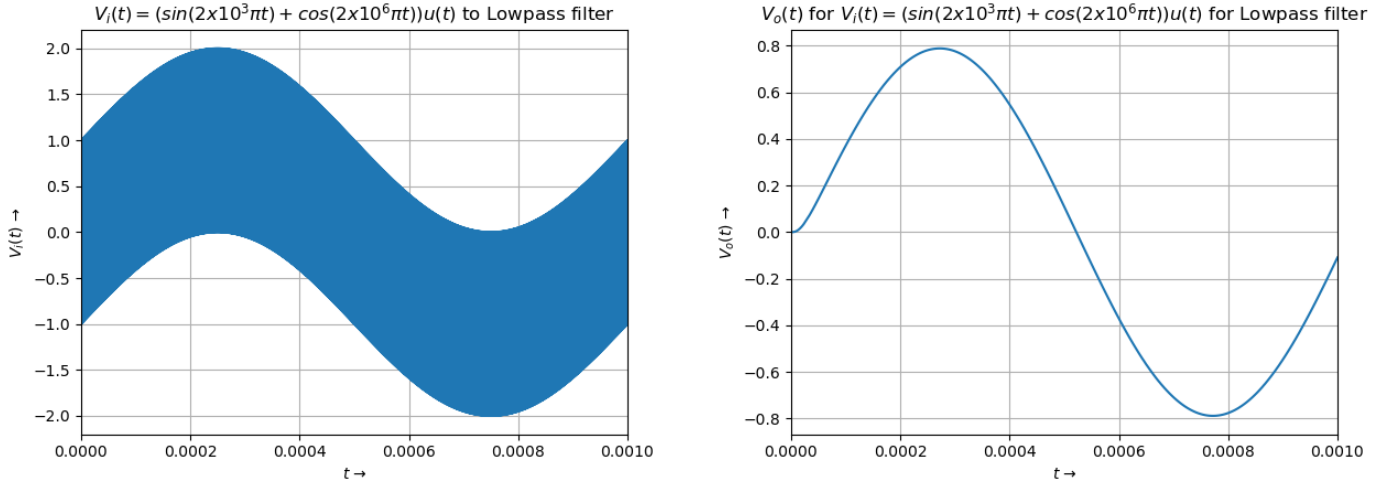


Figure 4: *Left*: Mixed frequency input
Right: Filtered Output

We can easily see that the output contains only the low frequency component of the input (1 KHz sinusoid). Thus, the circuit is a low-pass filter. It's cut-off frequency for the values of R_1, R_2, C_1, C_2 used is $\frac{1}{2\pi}$ MHz .

3 High-Pass Filter:

We shall now look at a slightly modified version of the above circuit.

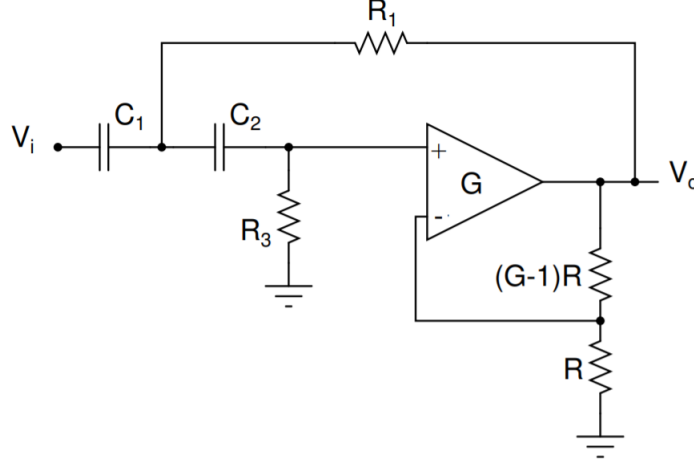


Figure 5: Op-amp based High-Pass Filter

Performing a similar procedure like before, we get the KCL matrix as:

$$\begin{pmatrix} 0 & 0 & 1 & \frac{-1}{G} \\ \frac{sR_3C_2}{1+sR_3C_2} & 0 & -1 & 0 \\ 0 & -G & G & 1 \\ -\frac{1}{R_1} & -sC_2 - sC_1 & 0 & sC_2 \frac{1}{R_1} \end{pmatrix} \begin{pmatrix} V_1(s) \\ V_p(s) \\ V_m(s) \\ V_o(s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -sC_1 V_i(s) \end{pmatrix} \quad (5)$$

Solving it for $V_o(s)$, we get,

$$V_o(s) = \frac{1.586 \times 10^{-14} s^2 \cdot V_i(s)}{2 \times 10^{-14} s^2 + 4.414 \times 10^{-9} s + 0.0002} \quad (6)$$

The Bode magnitude and phase plots of the transfer function as:

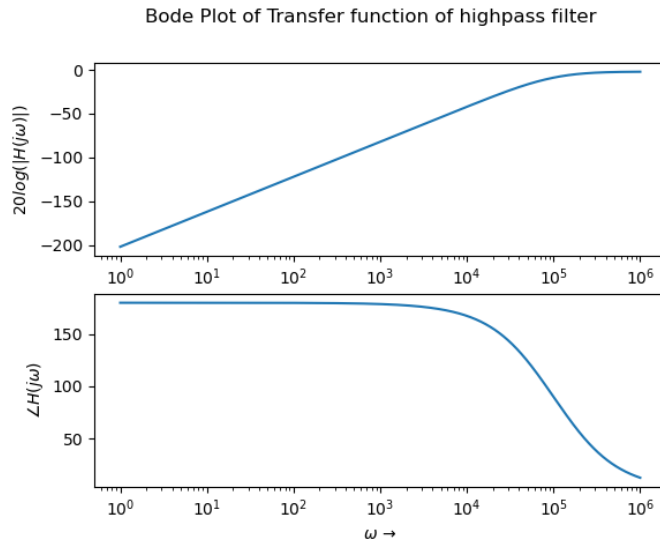


Figure 6: Bode Magnitude and Phase plots of step response of HPF

The step response of the circuit was obtained as:

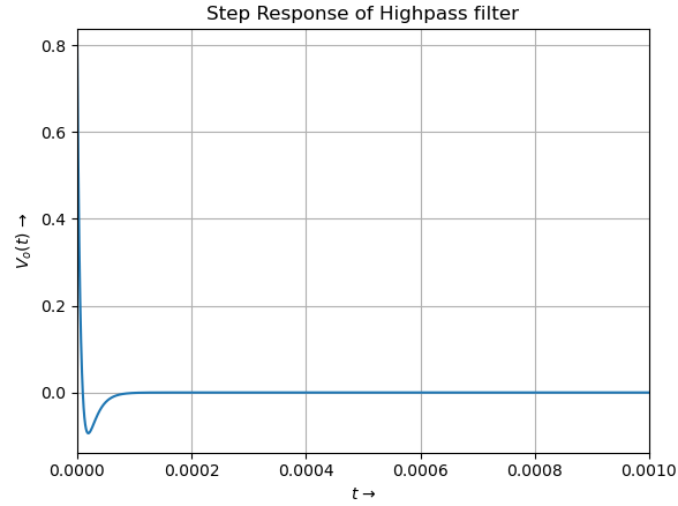


Figure 7: Step Response of HPF

The response to the mixed frequency input in Eq (4) is:

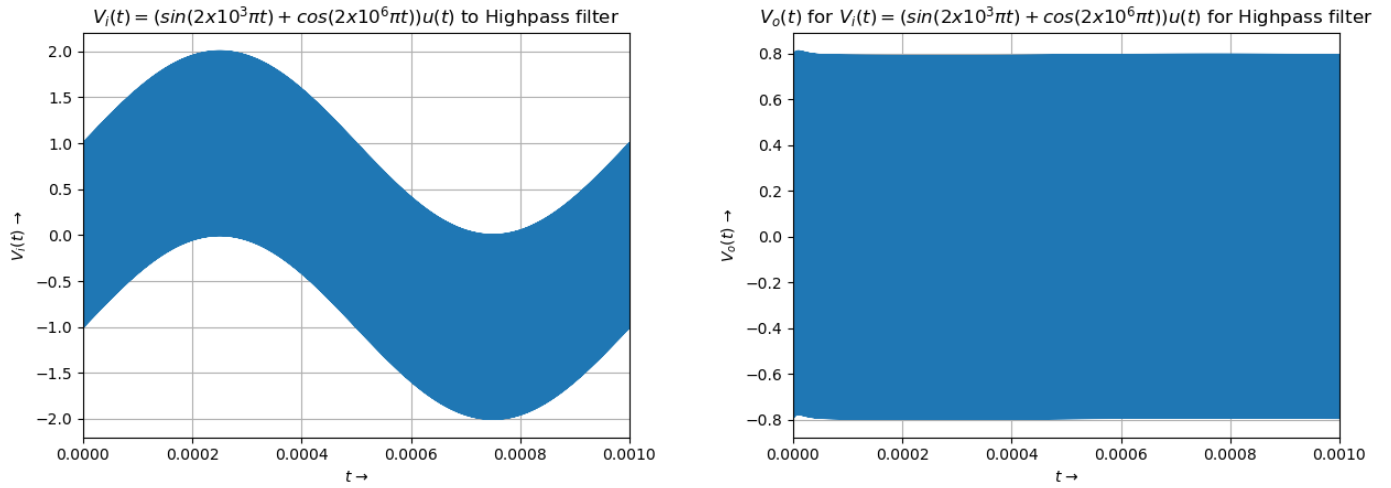


Figure 8: *Left*: Mixed frequency input
Right: Filtered Output

We shall look at the response of the system to a damped sinusoid. We shall consider two of them - one of high frequency (0.1 MHz) and the other of low frequency (1 Hz).

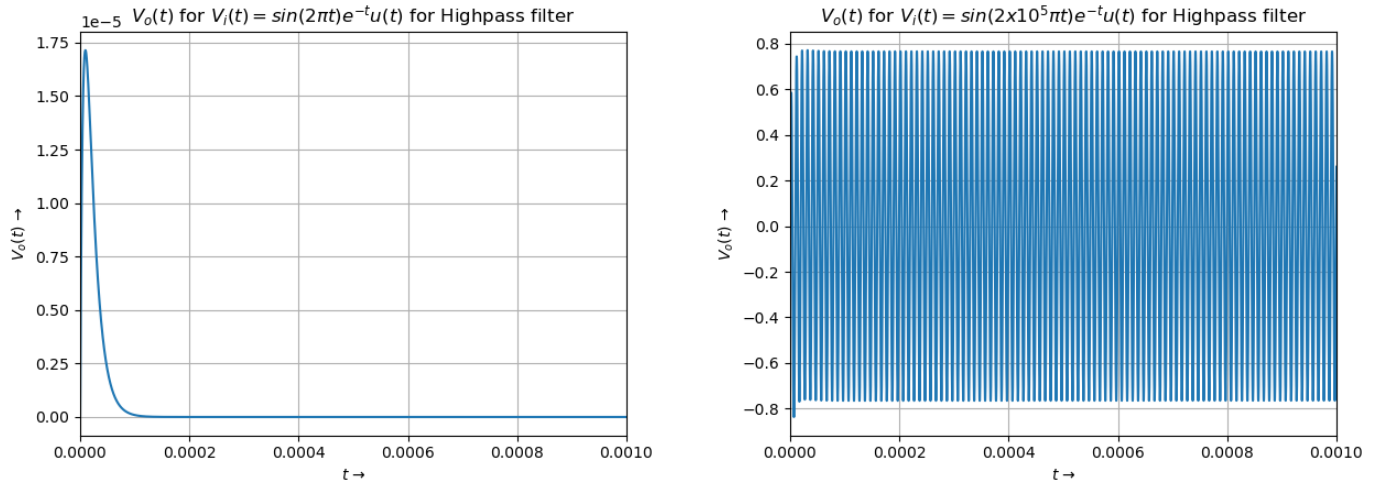


Figure 9: *Left*: Low frequency input
Right: High frequency input

We get that the output of the filter if the input is a low frequency damped sinusoid is 0, except for initial transients. This is expected due to the inherent nature of the circuit to act as a high-pass filter. This is why we can see that it allows the high-frequency input to pass through.

4 Conclusions:

Sympy provides a convenient way to analyse LTI systems using their Laplace transforms. The toolbox was used to study the behaviour of a low pass filter, implemented using an op-amp of gain G . For a mixed frequency sinusoid as input, it was found that the filter suppressed the high frequencies while allowing the low frequency components. Similarly, a high pass filter was implemented using an op-amp with the same gain. The magnitude response of the filter was plotted and its output was analysed for damped sinusoids. The step response of the filter was found to have a non-zero peak at $t = 0$, due to the sudden change in the input voltage.