EE2703 : Applied Programming Lab
Assignment - 7

Jawhar S
EE20B049

April 8, 2022

Solving Circuits in Laplace Domain

In this assignment, we solve for two different circuits: Low Pass Filter and High Pass Filter.

An ideal low pass filter is a circuit that allows only the lower frequency components (i.e, input frequencies below a cut off frequency) to pass through and blocks all the higher frequency components. In reality, a low pass filter also allows high frequency components to pass through but their amplitude is highly attenuated.

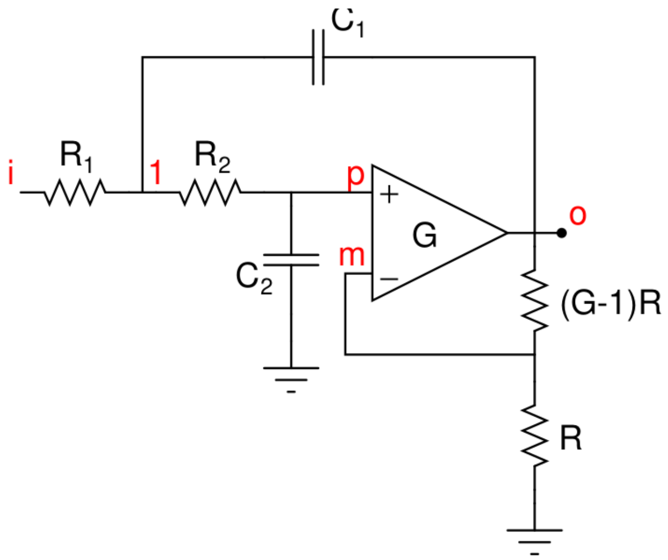

Figure 1: Low Pass Filter

Using nodal analysis in Laplace domain, we can obtain the set of equations which can be further solved to obtain the output to the corresponding input. This is done using *Sympy* which has powerful symbolic algebraic capabilities in Python. The code for the same is given below.

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

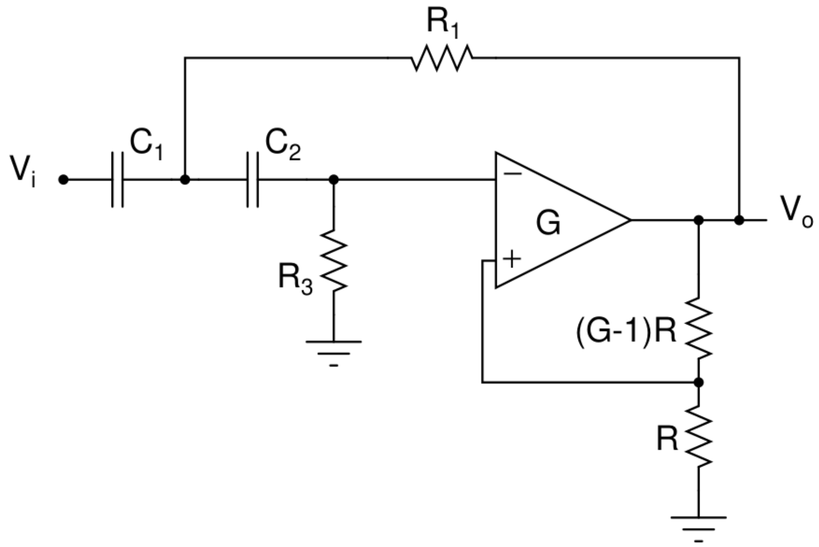

Figure 2: High Pass Filter

An ideal high pass filter is a circuit that allows only the higher frequency components (i.e, input frequencies above a cut off frequency) to pass through and blocks all the lower frequency components. In reality, a high pass filter also allows low frequency components to pass through but their amplitude is highly attenuated.

Similar approach as used for low pass filter is used to solve the high pass filter. The code for the same is given below.

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

Step Response of Low Pass Filter

```
A,b,V = lowpass(10000,10000,1e-9,1e-9,1.586,1/s)
H = sp_lti(V[3])
t = np.linspace(0,1.5e-4,int(1e5)+1)
t, lp_step = sp.impulse(H,None,t)
```

The step response of the low pass filter was found by applying $\frac{1}{s}$ as input. The function sp-lti is used to convert the Sympy expression to a Scipy compatible LTI system.

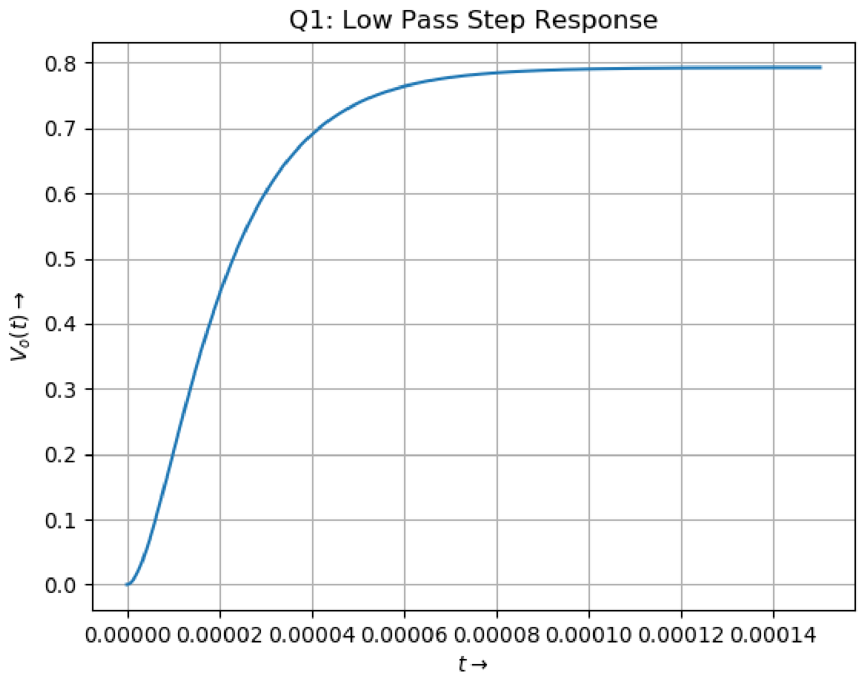

Figure 3: Low Pass Step Response

The response starts from zero, then increases to its steady state value around 0.8. The change in input at t=0 from 0 to 1 can be considered a high frequency component, and since this is a low pass filter, this component is attenuated. The steady input after this change can be considered low frequency, and it passes without attenuation. The saturation amplitude is 0.8 due to the construction of the circuit – it is the maximum gain of this filter across all frequencies.

Magnitude Response of High Pass Filter

```
A,b,V = highpass(10000,10000,1e-9,1e-9,1.586,1)
Vo = V[3]
w = np.logspace(0,8,801)
ss = 1j*w
hf = sm.lambdify(s,Vo,'numpy')
v = hf(ss)
```

The magnitude response of an ideal high pass filter should be equal to 1 for higher frequencies and 0 for lower frequencies.

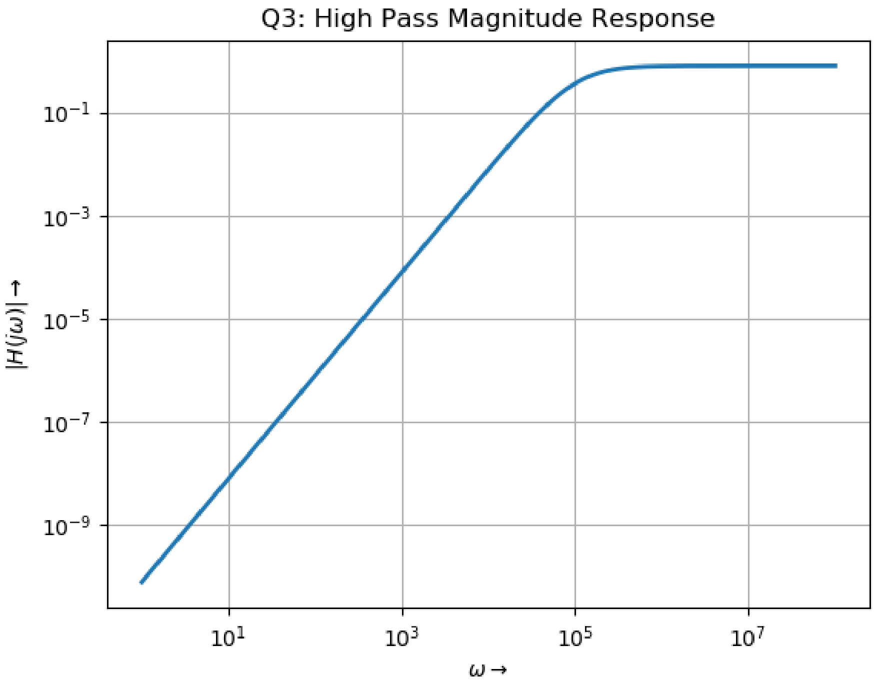

Figure 4: High Pass Magnitude Response

In the above figure, we observe that after ω = 1e5, the magnitude of H(jω) is 1 i.e. the output will be equal to the input for frequencies above 1e5. For ω ¡ 1e4, the magnitude of H(jω) increases by 20dB per decade (if increases by a factor of 10, then the magnitude of H(jω) increases by a factor of 10).

The plot has a slope of 20db/decade at low frequencies, and flat at high frequencies, indicating the presence of a single pole around 1e5.

High Pass Filter Response to Mixed Sinusoids

A mixed sinusoid of two frequencies, 2e3 and 2e6, is passed into the highpass filter. Since the attenuation of the low frequency was too much, the output has been plotted on the high frequency timescale.

We can use *sp.lsim()* for simulating the output to given input after obtaining the transfer function of the system. The code for the same is given below.

```
A,b,V = highpass(10000,10000,1e-9,1e-9,1.586,1)
H = sp_lti(V[3])
t = np.linspace(0,0.00001,int(1e5)+1)
vi = np.sin(2e3 * PI * t) + np.cos(2e6 * PI * t)
t,hp_out,svec = sp.lsim(H,vi,t)
```

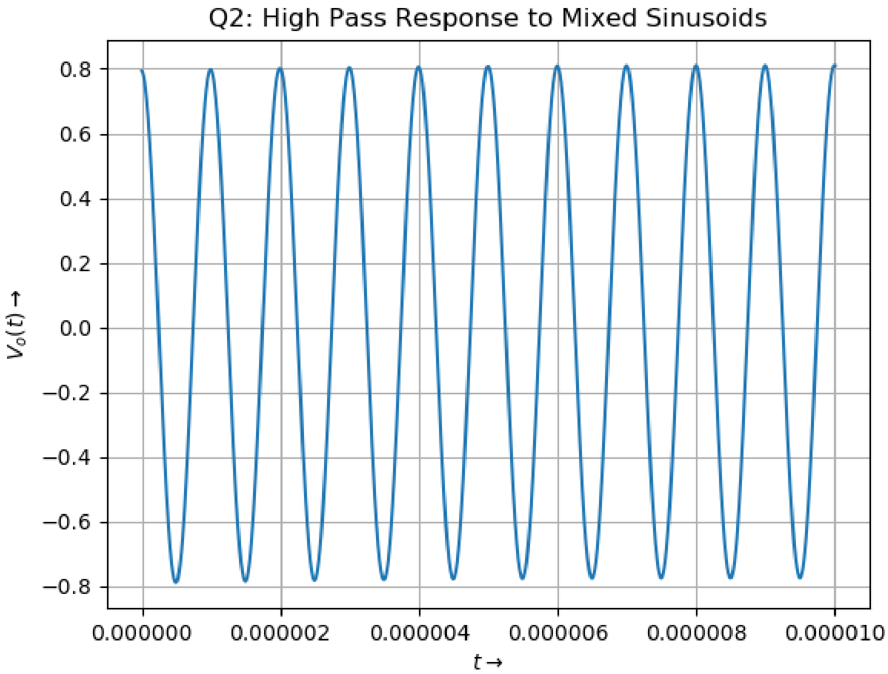

Figure 5: High Pass Magnitude Response

As we can observe from the above plot, only the sinusoid with high frequency (2e6) is passed and the sinusoid with low frequency (2e3) is attenuated.

High Pass Filter Response to Decaying Sinusoids

A low frequency and a high frequency sinusoid were passed as input.

```
H = sp_lti(Vo)
t_low = np.linspace(0,1,int(1e5)+1)
t_high = np.linspace(0,1e-5,int(1e5)+1)
vi_high = np.sin(1e7 * PI * t_high)*np.exp(-1.5e5*t_high)
vi_low = np.sin(1e2 * PI * t_low)*np.exp(-1.5*t_low)
t_high,hp_hf_out,svec = sp.lsim(H,vi_high,t_high)
t_low,hp_lf_out,svec = sp.lsim(H,vi_low,t_low)
```

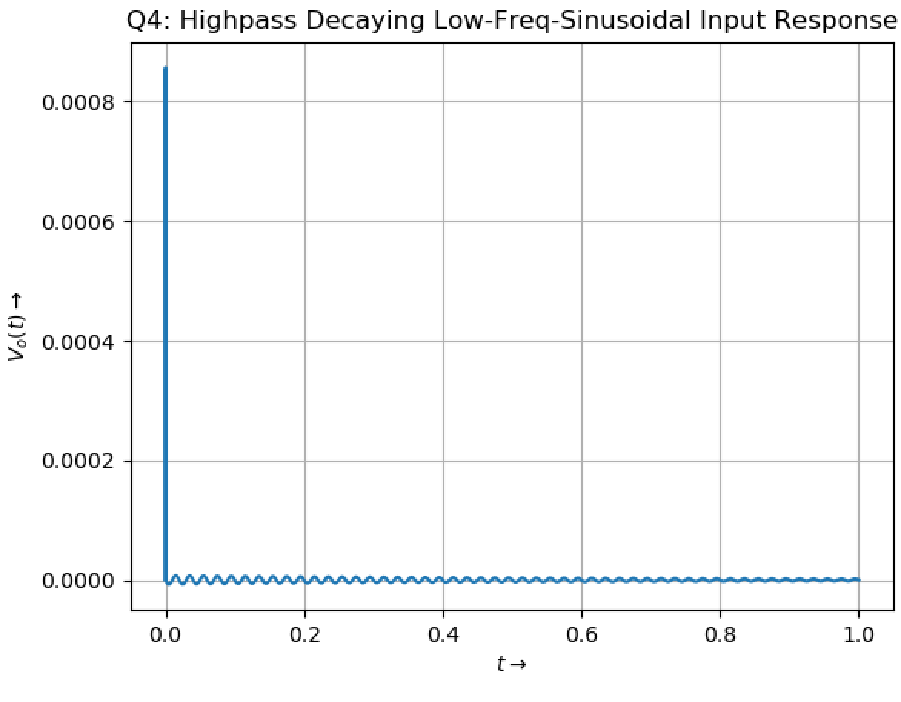

Figure 6: High Pass Response to low freq-decaying sinusoid

The low frequency response has been attenuated by about 1e-5, while the high frequency response is unattenuated. The amplitudes of both decay exponentially. The low frequency plot has been zoomed to show decaying nature. The un-zoomed plot is to show the initial peak. The low frequency plot has a peak near the beginning, much higher than the amplitude. This is due to the transients in the system.

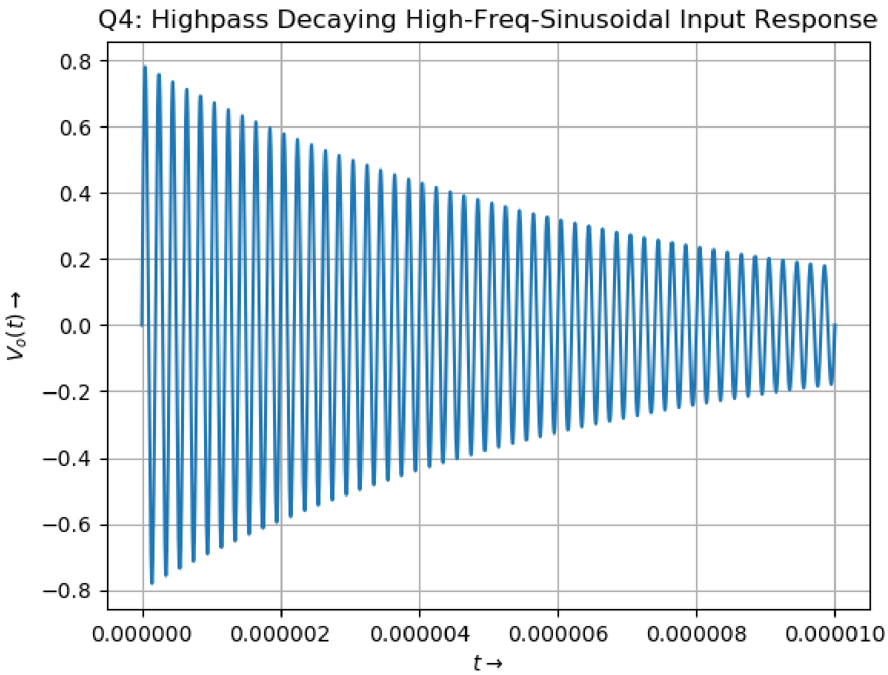

Figure 7: High Pass Response to high freq-decaying sinusoid

Step Response of High Pass Filter

```
A,b,V = highpass(10000,10000,1e-9,1e-9,1.586,1/s)
Vo = V[3]
H_step = sp_lti(Vo)
t = np.linspace(0,0.001,int(1e5)+1)
t, hp_step = sp.impulse(H_step,None,t)
```

The step response starts with a peak, then overshoots to negative, and then settles at 0 steady state. One way to think of this is – the change from 0 to 1 can be considered as a high frequency component, while the constant 1 after the switch can be considered a low frequency component. Since it is a high pass filter, it allows the 0-1 change component to pass, but attenuates the constant-1 component since it is low frequency.

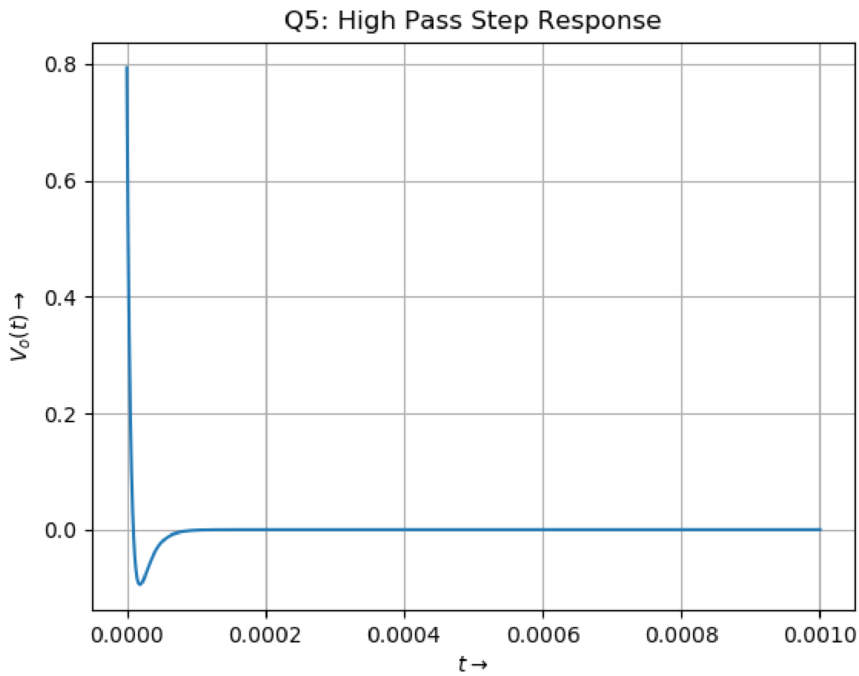

Figure 8: High Pass Step Response

Conclusion

This assignment explored the analysis of LTI circuits using Sympy in Python. Briefly, the following inferences were made.

- Q1: In step response of LPF, the start is low, it increases like exponential curve, then settles to a steady state value (0.8).
- Q2: When a sum of high and low frequency sinusoids are passed to HPF, the high frequency component is preserved while the low frequency is attenuated.
- Q3: The HPF was analyzed using Bode plot. Since the slope was +20dB/dec till 1e5, and then flat, the system has one pole around 1e5 and one zero around 0.
- Q4: The response to high frequency damped sinusoid is as expected – exponentially decaying amplitudes over time. The initial amplitude is unattenuated since it is HPF. The low frequency sinusoid also shows

the same features, but the amplitude is attenuated by the filter by around 1e-5. There is also an initial peak reaching 0.8, this is due to the transient response of the system.

- Q5: Step response starts high, and then overshoots, and then decays till 0 at steady state. With similar logic as LPF step response, the 0-1 change passes without attenuation since it is high frequency, while the constant-1 after is low frequency, and is attenuated.