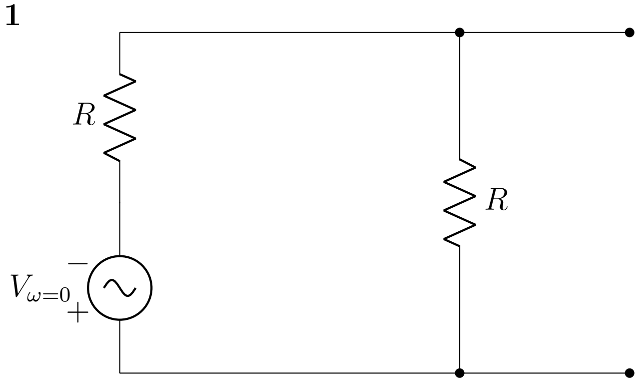
HW Set 3

Johannes Byle

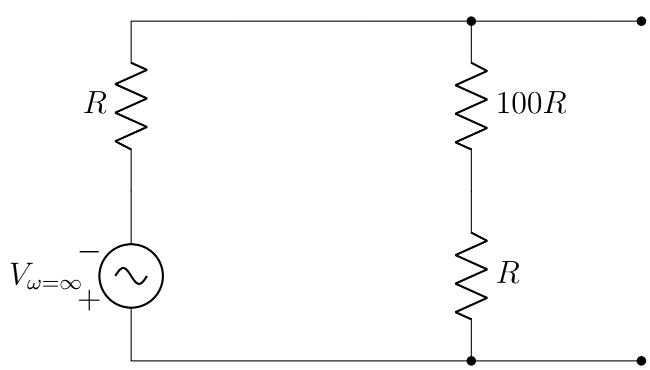
November 11, 2020

HW Set 3

1



$$\frac{V_{out}}{V_{in}} = \frac{1}{2} \approx -3.01 \text{ dB}$$

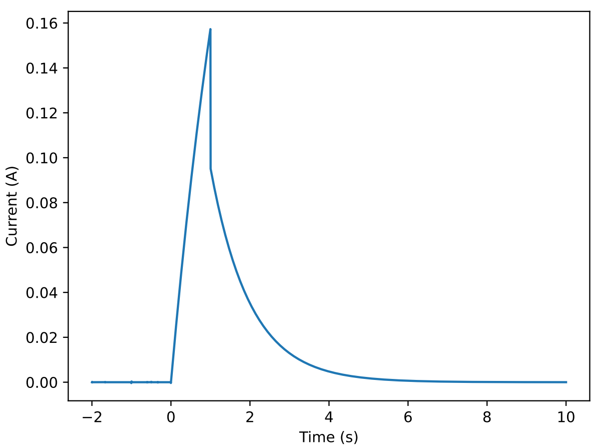


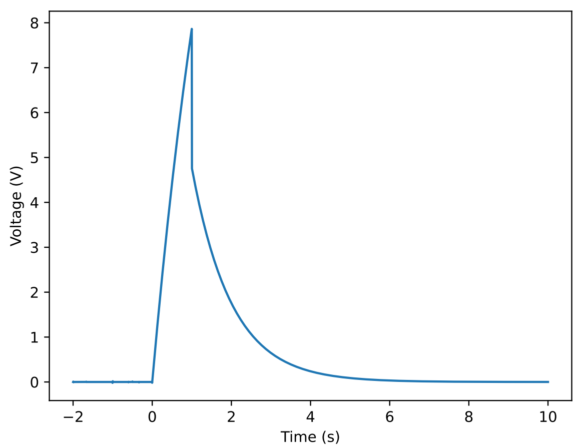
$$\frac{V_{out}}{V_{in}} = \frac{101}{102} \approx -0.04 \text{ dB}$$

2 Its a low pass filter.

$$H = \frac{\frac{1}{j\omega C}}{j\omega L + \frac{1}{j\omega C}}$$
$$|H| = \frac{1}{(CL\omega^2 - 1)^2}$$
$$\omega = \sqrt{1+\sqrt{2}}\sqrt{\frac{1}{CL}} = \sqrt{\frac{1-\sqrt{2}}{1\cdot 1\cdot 10^{-6}}} \approx 1554$$
$$f = 247 \text{ Hz}$$

3





```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad

t = np.linspace(-2, 10, 10000)
v = np.zeros(np.shape(t))
v[(t < 1) & (0 < t)] = 40
r = np.zeros(np.shape(t))
r[(t <= 1) & (0 < t)] = 100
r[t > 1] = 200
L = 200


def integral(time):
    def f(x):
        if x < 0:
            V = 0
            R = 0
        elif 0 <= x < 1:
            V = 40
            R = 100
        else:
            V = 0
            R = 200
        return (np.exp((R * x) / L) * V) / L

    y, err = quad(f, 1, time)
    return y


f = np.array([integral(n) for n in t])
i = np.exp(-(r * t) / L) * (f - f[0])
plt.plot(t, i)
plt.plot(t, i * 50)
plt.show()
```

This code simply plots the following equation for the given parameters.

$$e^{-\frac{Rt}{L}}\left(\int_1^t \frac{e^{\frac{R\xi}{L}}V\left(\xi\right)}{L}d\xi + C\right)$$

The code could have been a lot shorter and simpler, but I had to write my own function for the integral portion of the equation.

4.a

$$\omega_r = \frac{1}{\sqrt{LC}}$$

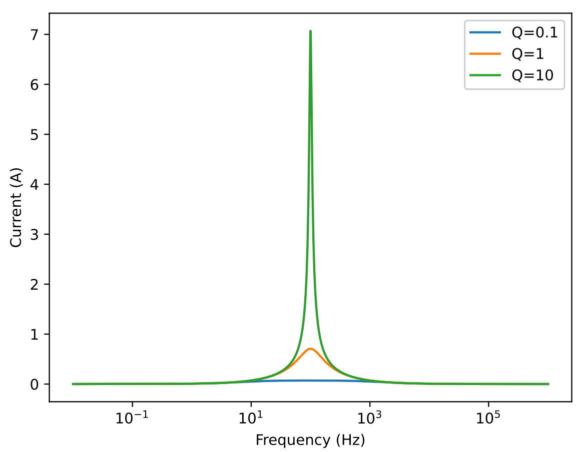
$$C = \frac{1}{L\omega_r^2} = \frac{1}{0.1 \cdot 100^2} = 0.001 \text{ F}$$

At this condition the imaginary component is zero, and its called the resonance frequency.

4.b

$$Q = \frac{\omega_0 L}{R}$$

$$R = \frac{\omega_0 L}{Q}$$




```
import numpy as np
```

```python
import matplotlib.pyplot as plt

C = 0.001
L = 0.1
R_1 = (100 * L) / 0.1
R_2 = (100 * L) / 1
R_3 = (100 * L) / 10


def z(w, R):
    val = R + 1 / (1j * w * C) + 1j * w * L
    return (10 / np.sqrt(np.real(val * np.conj(val)))) / (np.sqrt(2))


w = np.linspace(10 ** (-2), 10 ** 6, 1000000)
plt.plot(w, z(w, R_1), label="Q=0.1")
plt.plot(w, z(w, R_2), label="Q=1")
plt.plot(w, z(w, R_3), label="Q=10")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Current (A)")
plt.xscale("log")
# plt.yscale("log")
plt.legend()
plt.show()
```