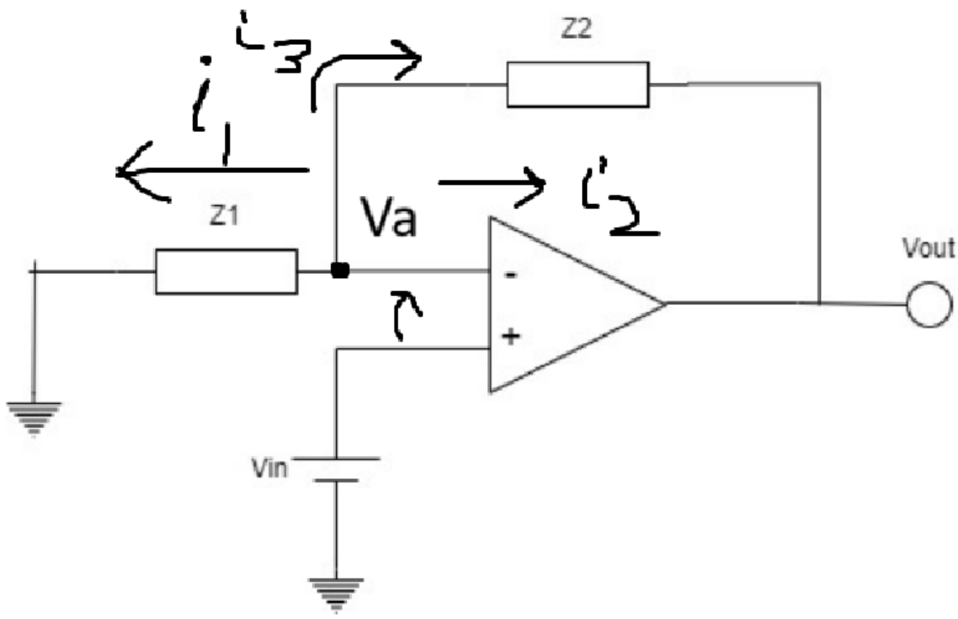
Aluno: Raphael Henrique Braga Leivas

Email: rapha.lei8@gmail.com

Atividade 6 - Capítulo 7

O código completo usado nessa atividade se encontra no ANEXO A.

Exercício 2



Aplicando análise nodal em V_a, temos

$$i_1 + i_2 + i_3 = 0$$

Note que $i_2 = 0$ pois o AmpOp é ideal e possui impedância de entrada infinita.

$$\frac{V_a - 0}{Z_1} + 0 + \frac{V_a - V_{out}}{Z_2} = 0$$

Pelo curto circuito virtual entre as entradas inversora e não-inversora, temos $V_a = V_{in}$. Logo,

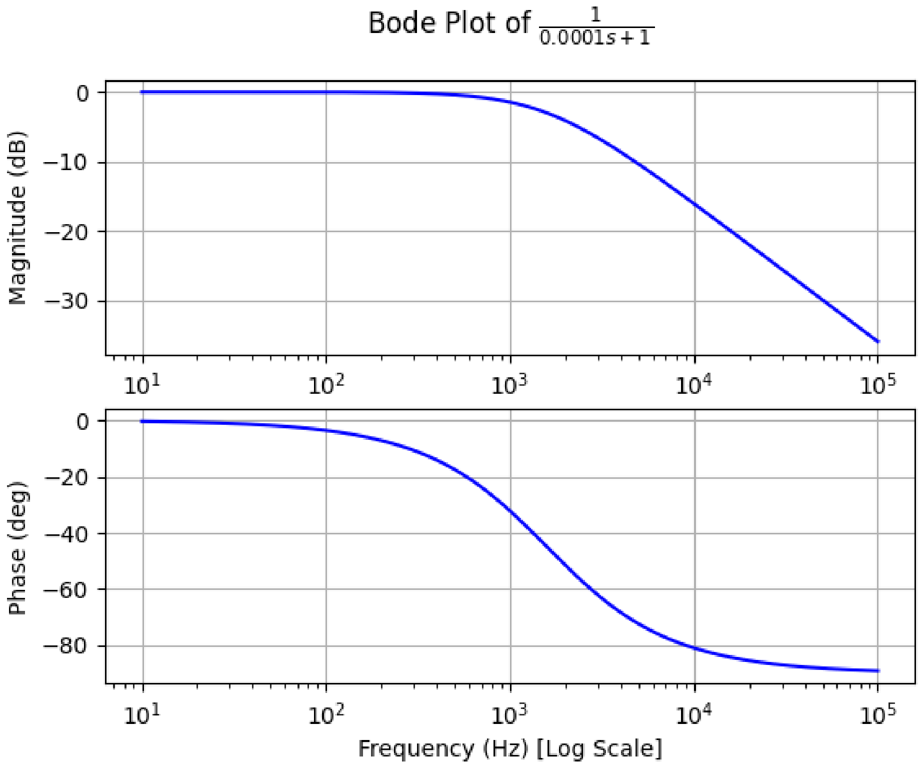
$$\frac{V_{in} - 0}{Z_1} + 0 + \frac{V_{in} - V_{out}}{Z_2} = 0$$

$$V_{in}\left(\frac{1}{Z_1} + \frac{1}{Z_2}\right) = V_{out}\frac{1}{Z_2}$$

$$\frac{V_{in}}{V_{out}} = G = H(s) = \frac{Z_2}{Z_1} + 1$$

Suponha que em Z_2 temos um capacitor de capacitância $C = 100nF$ e em Z_1 um resistor $R = 1k\Omega$. Temos

$$H(s) = \frac{\frac{1}{Cs}}{R} + 1 = \frac{1}{RCs} + 1$$



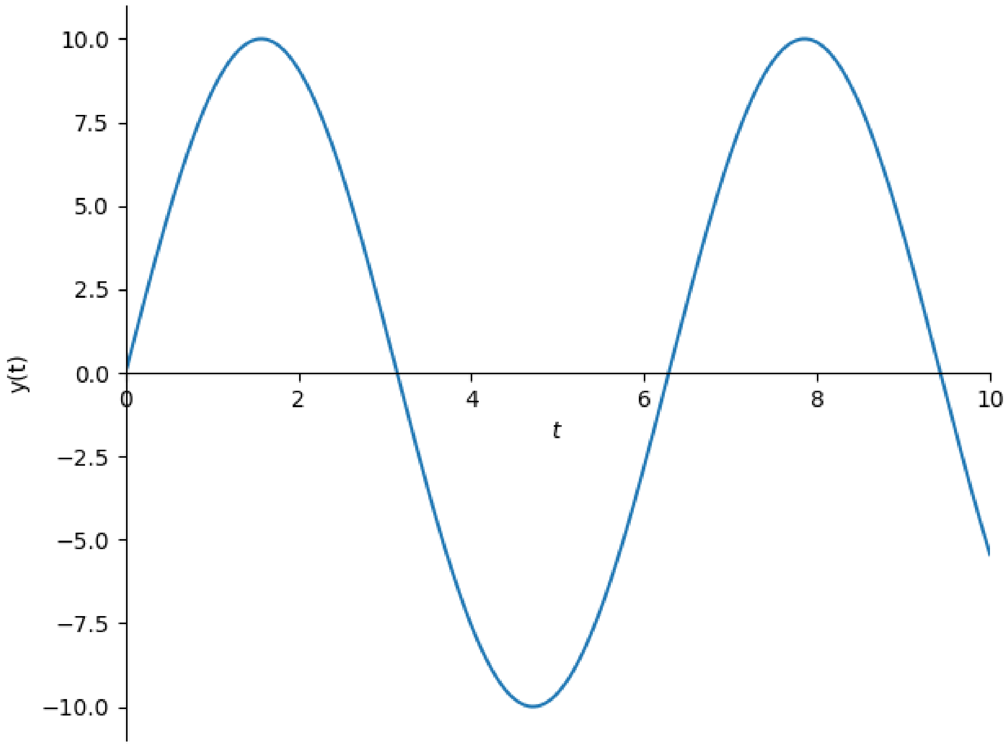
A resposta em frequência do circuito é dada pelo Diagrama de Bode abaixo. Ele atenua frequências maiores que 1 kHz.
Assumindo que a entrada é uma senoide dada por

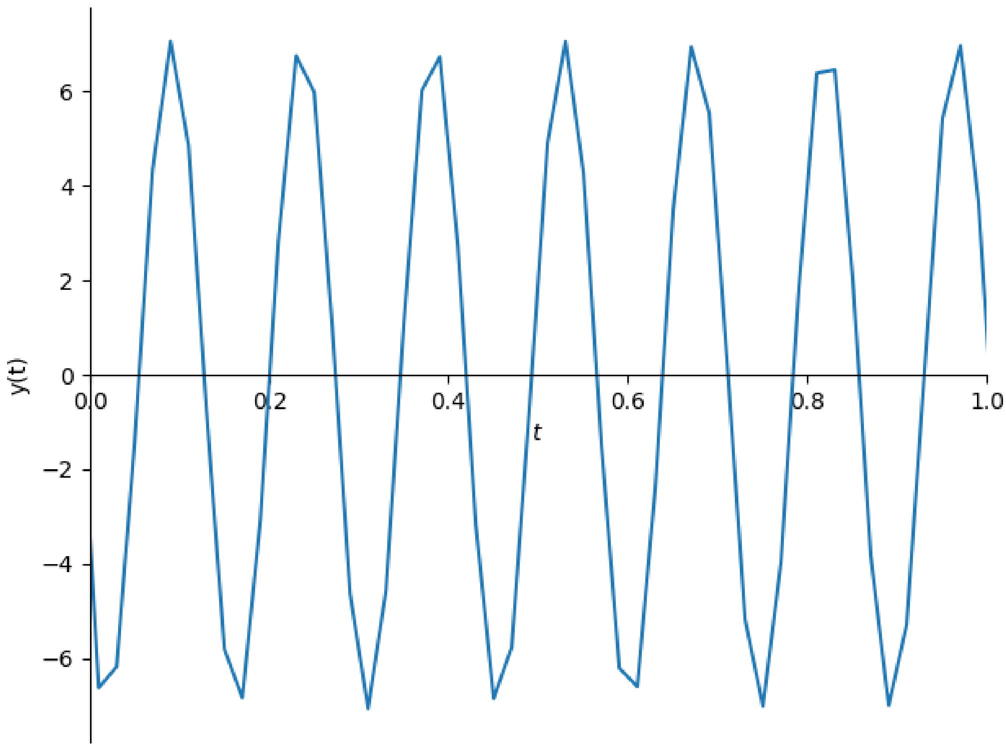
$$x(t) = A\sin(\omega t) \Longrightarrow X(s) = \frac{A\omega}{\omega^2 + s^2}$$

Logo, a saída do sistema é

$$Y(s) = X(s)H(s) = \frac{A\omega}{(\omega^2 + s^2)\,(CRs + 1)}$$

Assumindo $\omega = 1rad/s$ e $A = 10$, temos a saída com forma de onda abaixo. O sinal possui a mesma amplitude da entrada.
Se a entrada tivesse frequência maior, $\omega = 10krad/s$, temos a forma de onda abaixo. O sinal é atenuado em alta frequência, deformando-se e possuindo amplitude menos que a entrada.


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
y(t)
0
2
4
6
8
10
t


6
4
2
0
−2
−4
−6
y(t)
0.0
0.2
0.4
0.6
0.8
1.0
t

ANEXO A - Código

```
import numpy as np
import matplotlib as plt
from sympy import *
from sympy.physics.control.control_plots import pole_zero_plot
from sympy.physics.control.control_plots import bode_plot
from sympy.physics.control.lti import TransferFunction

## Exercicio 2

t, s = symbols('t, s')
omega, A = symbols('omega, A')
xt = A * sin(omega*t)

Xs = laplace_transform(xt, t, s, noconds=True)

print(latex(Xs))

C, R = symbols('C, R')

Hs_num = 1
Hs_den = R * C * s + 1

ft1 = TransferFunction(Hs_num, Hs_den, s)

# bode_plot(ft1.subs({ C: 100e-9, R: 1e3 }), initial_exp=1, final_exp=5, phase_unit='deg',
    freq_unit='Hz')

Ys = Xs * (Hs_num / Hs_den)

yt = inverse_laplace_transform(Ys.subs({ C: 100e-9, R: 1e3, A: 10, omega: 10000 }), s, t,
    noconds=True)

print(latex(yt))

plot(yt, ylabel = "y(t)", xlim=(0, 1))
```