Q1- RL serie

November 28, 2020

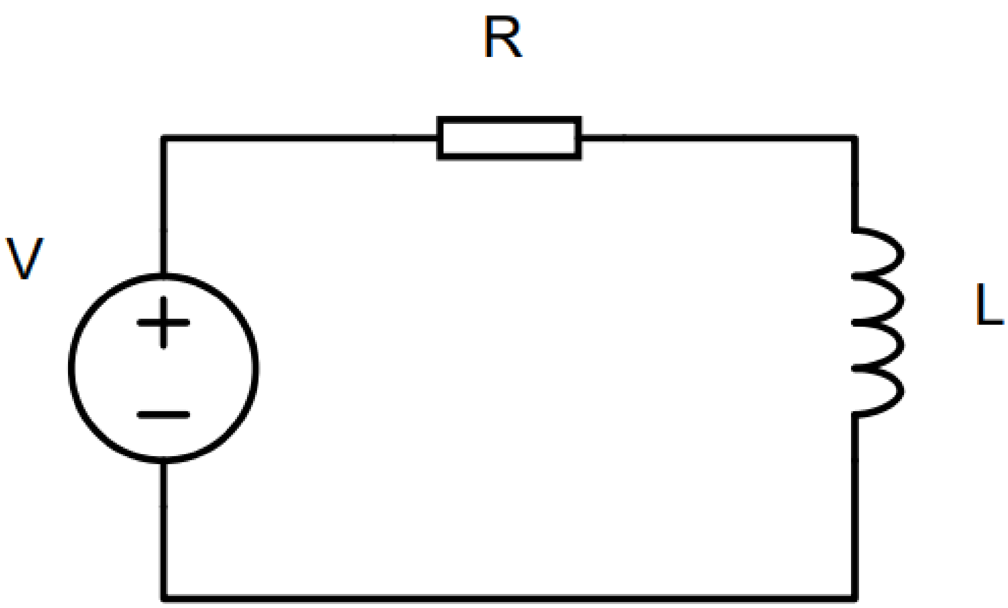
1 Q1

1.0.1 Reposta Forçada

Dado o Circuito RL em Serie abaixo

[15]:
```
from IPython.display import Image
Image(filename='Rlserie.png')
```

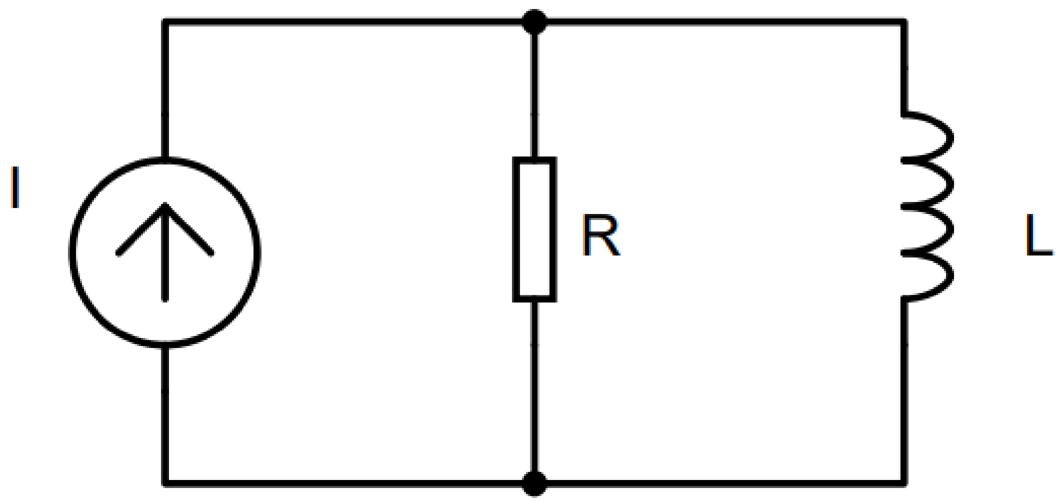
[15]:



Usando o equivalentes Norton e Thévinin ficamos com o Circuito abaixo:

[16]:
```
from IPython.display import Image
Image(filename='Rlserie2.png')
```

[16]:



Sabemos que em $t \to 0^+$ nosso indutor se comporta como circuito aberto, isso é assim que começarmos, ou pensarmos em chavear, teremos que

$$V_L(0) = RI$$

e

$$i_L(0) = 0$$

Análisando o nó superior e aplicando Kichhoff teremos

$$i_R + i_L = I$$

$$\frac{di_r}{dt} + \frac{di_L}{dt} = 0$$

lembrando que $V(0) = RI$ e $V_L(t) = L\frac{di}{dt}$

logo teremos a EDO

$$\frac{1}{R}\frac{dV}{dt} + \frac{1}{L}V = 0$$

[17]:
```
from math import *
import sympy as sp

R,t,L,I = sp.symbols("R t L I")
V = sp.Function('V_L')
iL = sp.Function('i_L')

v=sp.dsolve((1/R)*V(t).diff(t)+(1/L)*V(t),V(t))
```

```
v
```

[17]: $\mathrm{V_L}\left(t\right)=e^{R\left(C_1-\frac{t}{L}\right)}$

Simplifcando ainda mais a edo resolvida, temos que a Tensão no indutor é dada pela função

$$V_L(t) = C_1 \cdot e^{-\frac{R}{L}t}$$

Jogando as condições iniciais $V(0) = RI$, ficamos com

$$V_L(t) = RI \cdot e^{-\frac{R}{L}t}$$

sabendo que $V_L(t) = L\frac{di}{dt}$ basta integrar e dividir por L

[18]:
```
V = lambda t: R*I*sp.exp(-(R/L)*t)

iL = lambda t: sp.integrate((1/L)*V(t),(t,0,t))

iL(t)
```

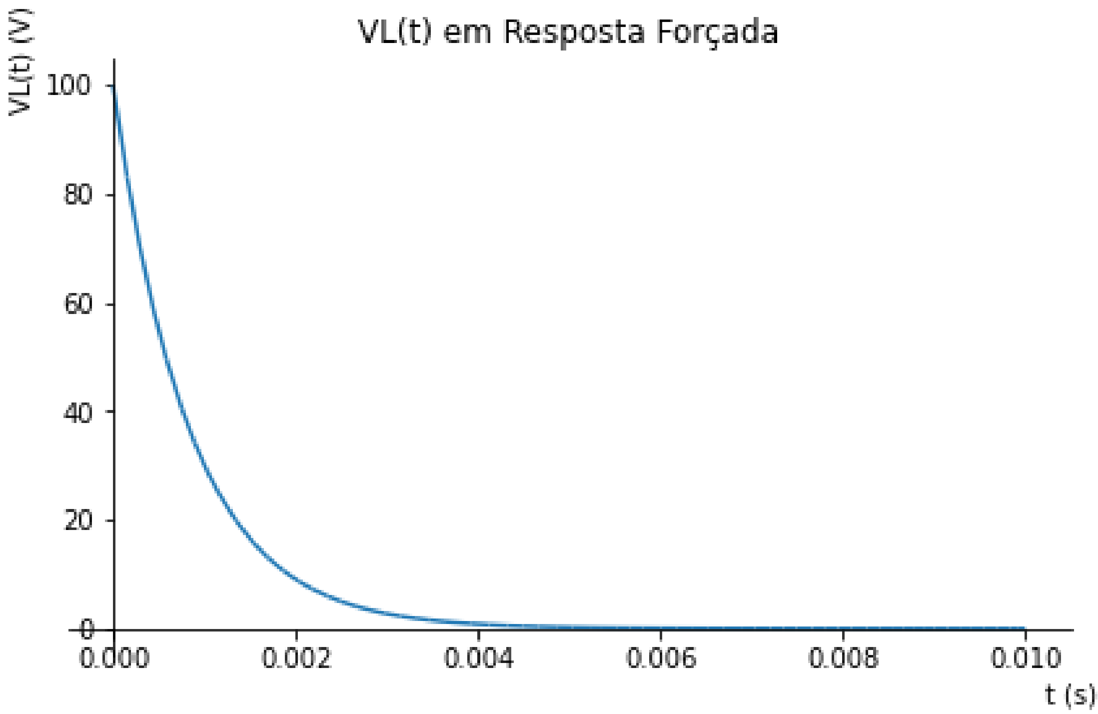
[18]: $I - Ie^{-\frac{Rt}{L}}$

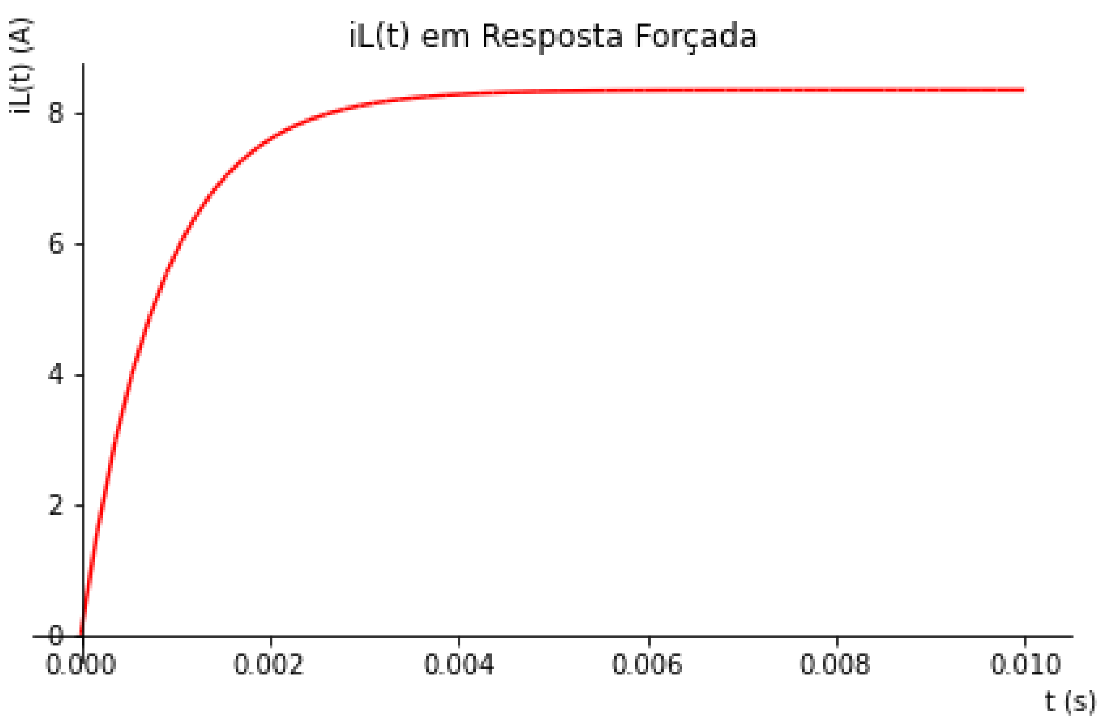
Logo botando em evidência temos

$$i_L(t) = I \cdot (1 - e^{-\frac{R}{L}t})$$

Uma vez com as equações bem estabelecidas, vamos considerar um circuito RL em serie com uma fonte V onde as considereções para as constantes foram estabelecidas no código abaixo.

[19]:
```
R = 12
L = 10e-3
V = 100
I = V/R

iL = lambda t: I*(1-sp.exp(-(R/L)*t))
Vl = lambda t: R*I*sp.exp(-(R/L)*t)

sp.plot(Vl(t),(t,0,0.01),xlabel = "t (s)",ylabel="VL(t) (V)", title="VL(t) em␣
 ↪Resposta Forçada")
sp.plot(iL(t),(t,0,0.01),xlabel = "t (s)",ylabel="iL(t) (A)", title="iL(t) em␣
 ↪Resposta Forçada", line_color = 'red')
```


VL(t) em Resposta Forçada
VL(t) (V)
100
80
60
40
20
0
0.000
0.002
0.004
0.006
0.008
0.010
t (s)

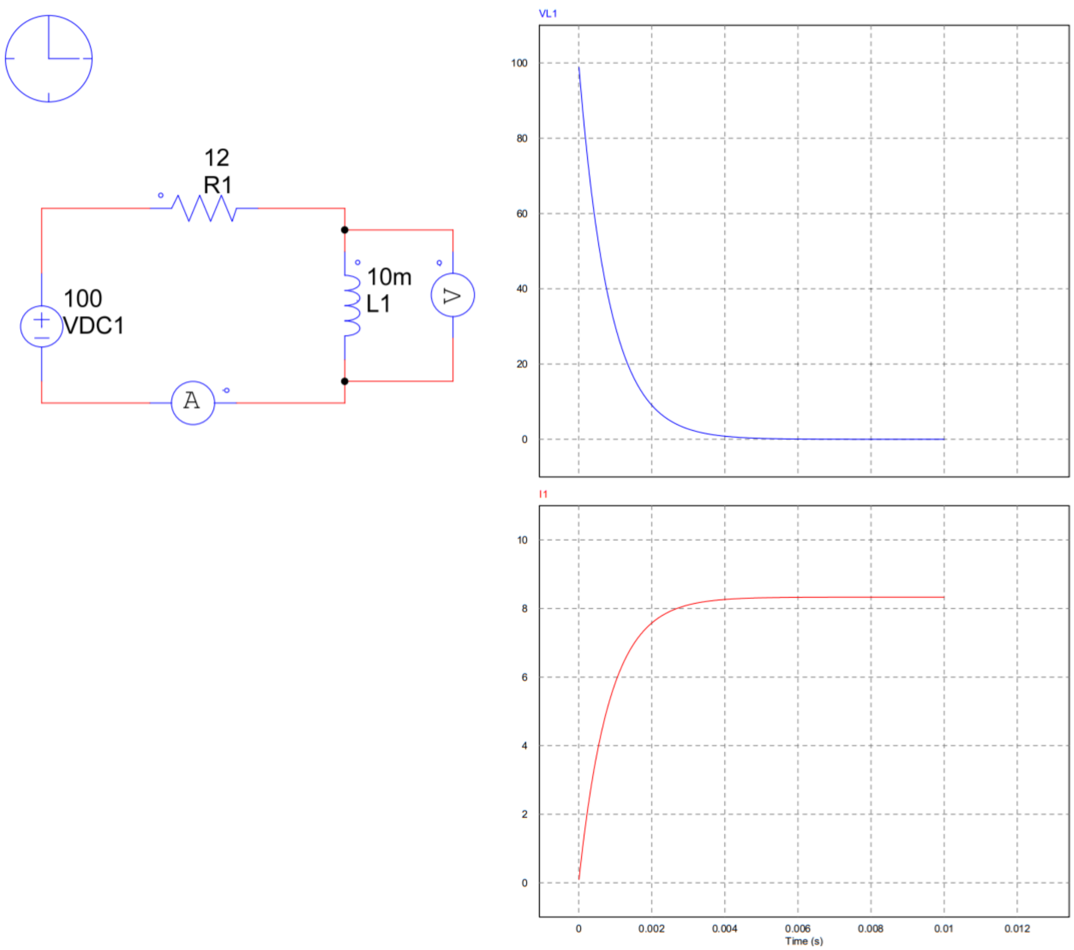

iL(t) em Resposta Forçada
iL(t) (A)
8
6
4
2
0
0.000
0.002
0.004
0.006
0.008
0.010
t (s)

[19]: <sympy.plotting.plot.Plot at 0x2002c0f2190>

Comparando agora na simulação configurada com os mesmo parâmetros temos que o Osciloscopico estipula a mesma curva

[20]:
```python
from IPython.display import Image
Image(filename='CircuSimuPsim.png')
```

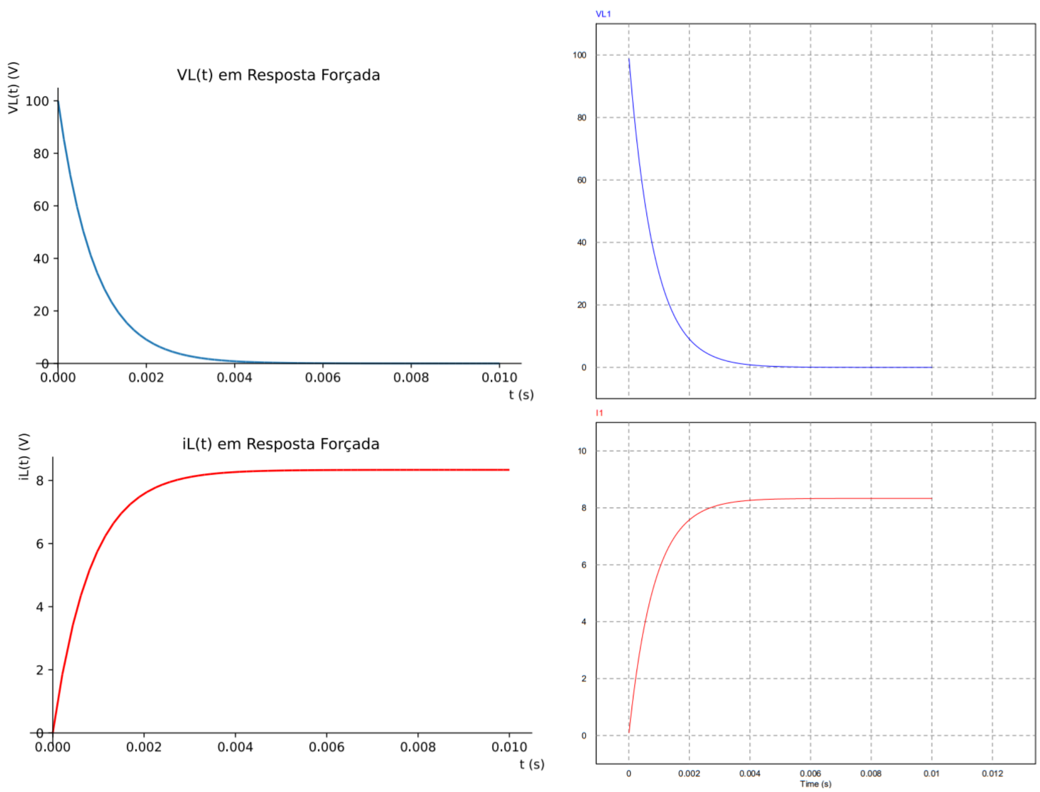
[20]:



Podemos agora fazer uma comparação da resposta forçada através das soluções algébricas para com a simulação do circuito RL em serie no software PSIM, no qual os gráficos a esquerda representam a solução algébrica e já há direita a simulação.

[21]:
```python
from IPython.display import Image
Image(filename='Comparacao.png')
```

[21]:

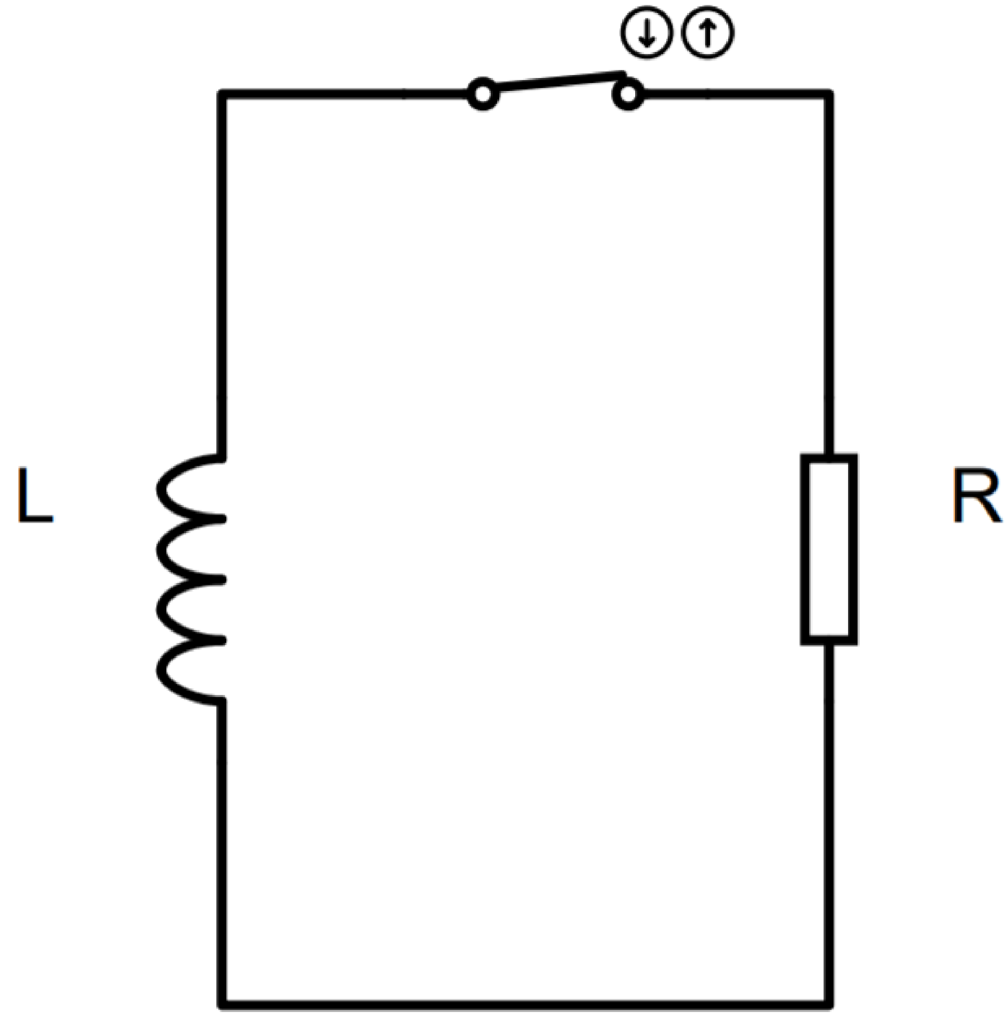


1.0.2 Resposta Natural

Agora para a resposta natural vamos admitir as seguintes condições iniciais $i_L(0) = -I_0$ e $V_L(0) = RI_0$

[22]:
```python
from IPython.display import Image
Image(filename='RLnatura.png')
```

[22]:



Aplicando novamente Kirchhoff porém agora no switch (desprezando qualquer resistência nele) teremos que

$$V_L(t) + V_R(t) = 0$$

$$L\frac{di(t)}{dt} + R\ i(t) = 0$$

Resolvendo essa EDO

[23]:
```
i = sp.Function('i')
R,L = sp.symbols("R L ")
sp.dsolve( L*i(t).diff(t) + R*i(t), i(t))
```

[23]:
$i(t) = e^{\frac{C_1 - Rt}{L}}$

Jogando as condições iniciais podemos destacar que $C_1 = -I_0$ ficando assim com

$$i(t) = -I_0 \cdot e^{-\frac{R}{L}t}$$

sabendo que

$$V_L(t) = L\frac{di}{dt}$$

podemos derivar a função $i(t)$ e multiplicar por L para encontrar a tensão

[24]:
```
I0 = sp.Symbol("I_0 ")
Vl1 = sp.Function('V_L')

i = lambda t: -I0*sp.exp(-(R/L)*t)

L*i(t).diff(t)
```

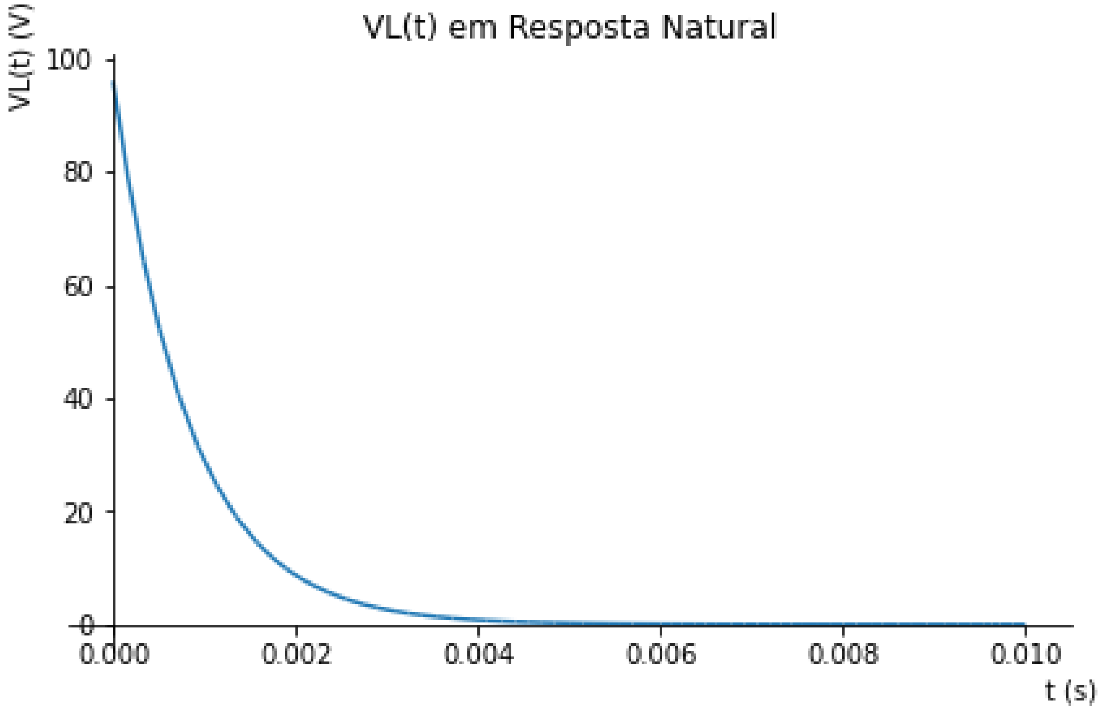
[24]:
$I_0 Re^{-\frac{Rt}{L}}$

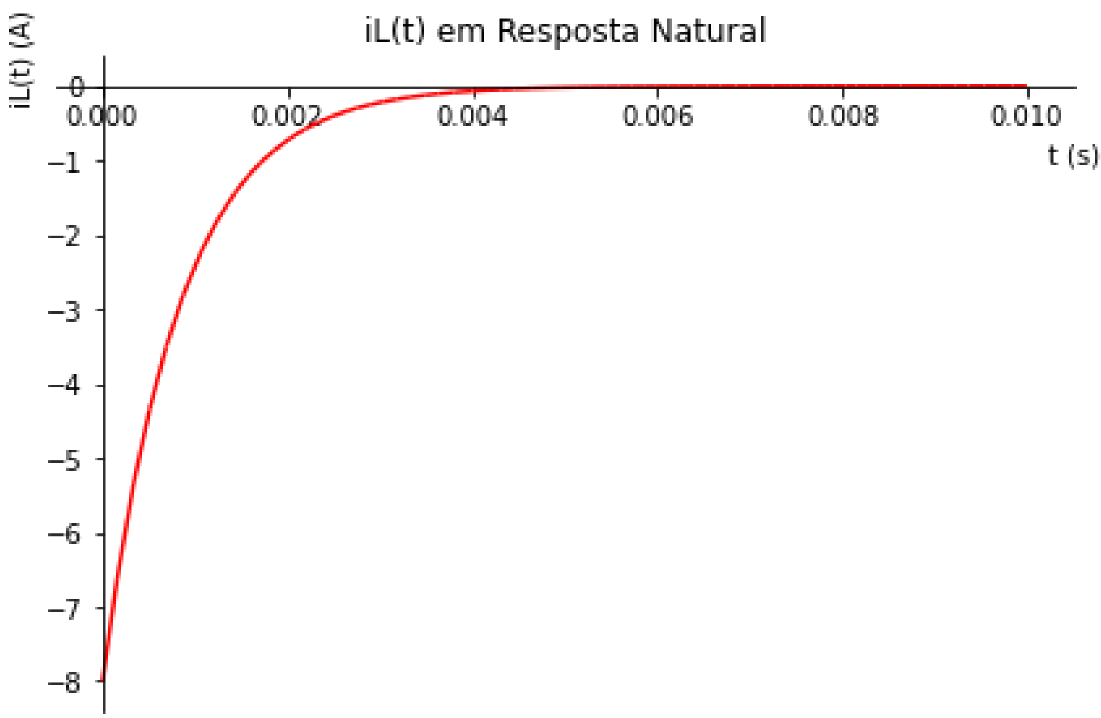
[25]:
```
Vl1 = lambda t: I0*R*sp.exp(-(R/L)*t)

Vl1(t)
```

[25]:
$I_0 Re^{-\frac{Rt}{L}}$

Determida as funções iremos agora plotar o gráfico da resposta natural, considerando a simulação no software PSIM e comparando-as e admitindo que o indutor está fornescendo $I_0 = 8A$ inicialmente

[26]:
```
I0 = 8
R = 12
L = 10e-3
sp.plot(Vl1(t),(t,0,0.01),xlabel = "t (s)",ylabel="VL(t) (V)", title="VL(t) em␣
↪Resposta Natural")
sp.plot(i(t),(t,0,0.01),xlabel = "t (s)",ylabel="iL(t) (A)", title="iL(t) em␣
↪Resposta Natural", line_color = 'red')
```


VL(t) em Resposta Natural
VL(t) (V)
100
80
60
40
20
0
0.000
0.002
0.004
0.006
0.008
0.010
t (s)

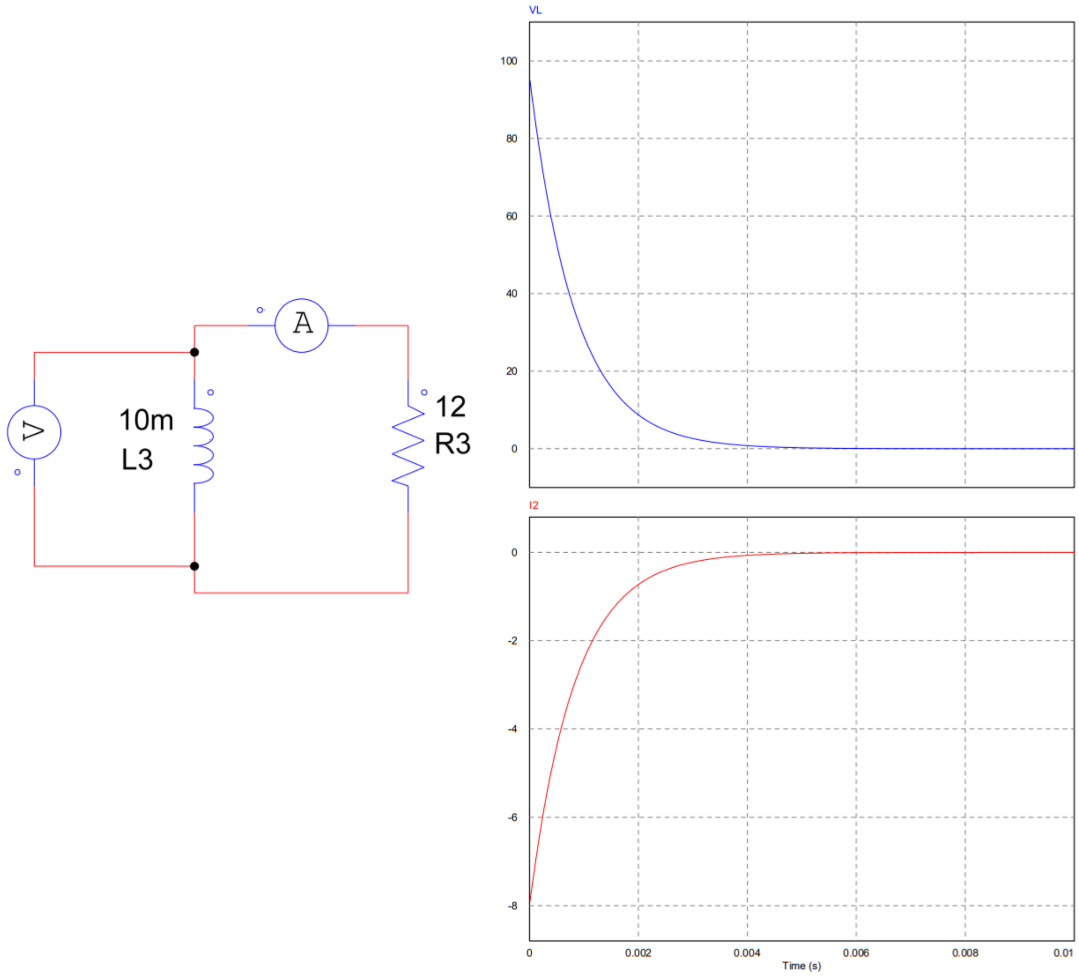

iL(t) em Resposta Natural
iL(t) (A)
0
−1
−2
−3
−4
−5
−6
−7
−8
0.000
0.002
0.004
0.006
0.008
0.010
t (s)

[26]: <sympy.plotting.plot.Plot at 0x2002bf67610>

no PSIM

[27]:
```
from IPython.display import Image
Image(filename='PSIM_natural.png')
```

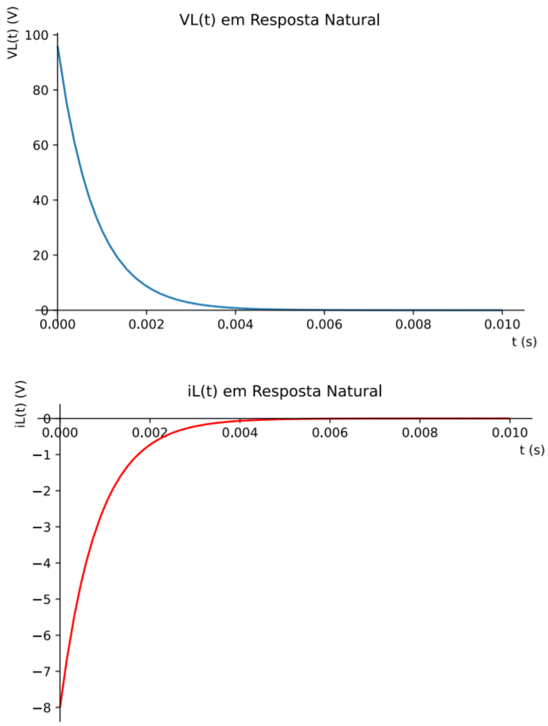
[27]:

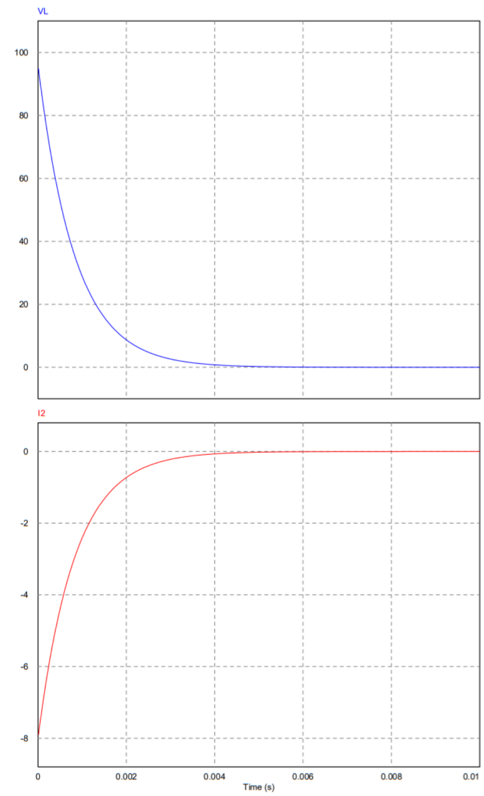


Comparando agora a simulação no PSIM há esquerda, com o Cálculo em python a direita

[28]:
```
from IPython.display import Image
Image(filename='comparaNatural.png')
```

[28]:


VL(t) em Resposta Natural
VL(t) (V)
100
80
60
40
20
0
0.000
0.002
0.004
0.006
0.008
0.010
t (s)
iL(t) em Resposta Natural
iL(t) (V)
0
−1
−2
−3
−4
−5
−6
−7
−8
0.000
0.002
0.004
0.006
0.008
0.010
t (s)


VL
100
80
60
40
20
0
I2
0
-2
-4
-6
-8
0
0.002
0.004
0.006
0.008
0.01
Time (s)