Trabalho 1

Metrologia e Instrumentação para Controle e Automação

Carlos Eduardo dos Santos Junior (16250645)

Professor: Dr. Ciro André Pitz

BLUMENAU /SC - Março, 2021

Questão 1:

Dado o circuito da Figura 1 e a partir das informações do mesmo (Figura 2), pede-se:

1 - Determinar valor da carga RL que permita a operação adequada do circuito de controle de corrente.

2 - Simulação do circuito com o software LTspice, e o valor máximo de RL utilizando o modelo ua741 para o amplificador operacional e 2N3904 para o transistor.

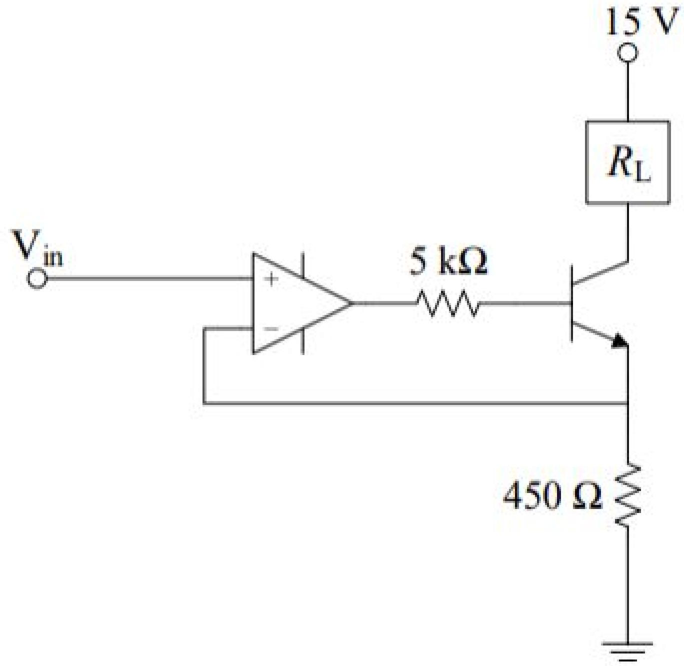


Figura 1: Modelo do Circuito.

- V_{cc} = 15V;
- 0V ≤ V_{in} ≤ 10V;
- amplificador operacional modelo uA741;
  - alimentado com ±15V;
  - limites da tensão na saída: ±13V;
- transistor BJT modelo 2N3904;
  - tensão de saturação V_{ce} = 0,12V;

Universidade Federal de Santa Catarina
Metrologia e Instrumentação para Automação
Professor: Ciro André Pitz

  - quando em condução, V_{be} = 0,7V.

Figura 2: Informações do Circuito.

Para facilitar os cálculos podemos dar nomes para as correntes e tensões do circuito, como segue na Figura 3.

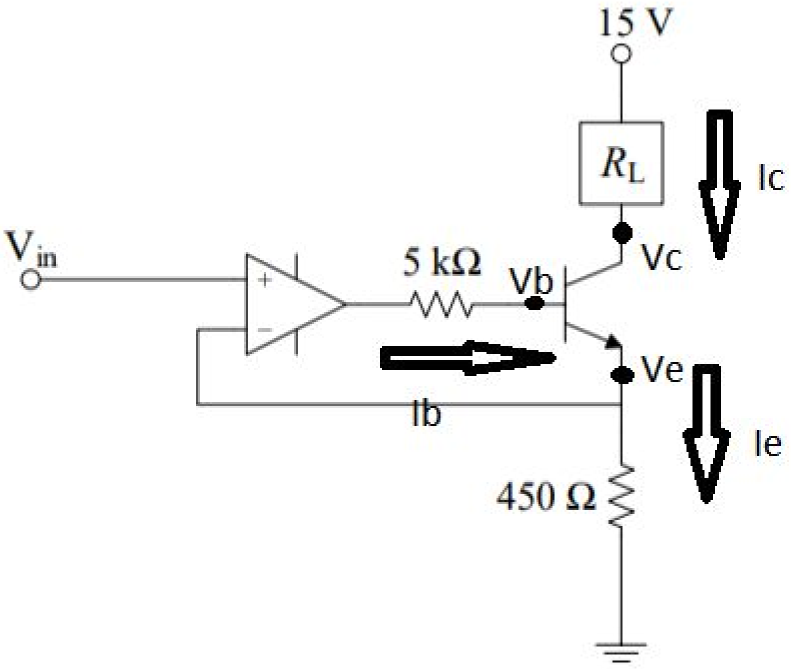


Figura 3: Correntes e pontos de tensão marcados.

1 - Para determinar o valor de RL precisamos saber qual o valor da corrente Ic e da tensão Vc, para podermos aplicar a lei de Ohm, e descobrir RL.

Pela LKC sabemos que $Ie = Ib + Ic$ então, $Ic = Ie - Ib$.

A corrente Ie pode ser encontrada a partir da lei de Ohm, dada por $Ie = \frac{Ve-0}{Re}$, como $Re = 450\,\Omega$ e Ve é a tensão no terminal negativo do amplificador operacional, Logo $Ve = Vin_{Max} = 10\ V$, Vin_{Max} é devido a encontrarmos a resistência máxima de RL.

Portanto, temos:

$$Ie = 10 / 450 = 22,22\ mA$$

Para corrente Ib, temos que considerar a saída máxima do amplificador operacional, chamaremos de $Vamp$ que é de $13\ V$, Vb será a tensão obtida através da expressão

$$Vb - Ve = Vbe\,(l)$$

Pelas informações do dispositivo sabemos que $Vbe = 0,7\ V$, Ve é a tensão no terminal negativo do amplificador operacional, como visto anteriormente, $Ve = 10\ V.$, logo pela equação (l)

$$Vb = Vbe + Ve = 0,7 + 10 = 10,7\ V$$

Considerando a resistência na saída do amplificador que chamaremos de $Ramp$ sendo de $5\ K\Omega$, com isso, temos:

$$Ib = Vamp - Vb\,/\,Ramp$$

Substituindo os valores, obtemos:

$$Ib = 13 - 10,7\,/\,5000 = 0,46\ mA$$

Com os valores de $Ib\ e\ Ie$ em mãos, podemos utilizar a relação da corrente Ic explicitada anteriormente

$$Ic = Ie - Ib = 22,22\ x\ 10^{-3} - 0,46\ x\ 10^{-3} = 21,76\ mA$$

Agora, basta obter o valor da tensão Vc, utilizando novamente as informações da Figura 2, sabemos que, $Vce = 0,12$ e Vc é dado por:

$Vc - Ve = Vce$ logo, $Vc = Ve + Vce$, então

$$Vc = 10 + 0,12 = 10,12\ V$$

Utilizando a lei de Ohm $V = RI$ e chamando a tensão de entrada da Figura 3 em cima de RL de $Vrl = 15\ V$, obtemos RL

$$RL = \frac{Vrl - Vc}{Ic} = \frac{15 - 10,12}{21,76\ x\ 10^{-3}} = 224,26\ \Omega$$

Concluída a parte 1 da questão 1, iremos agora simular com o Software LTspice para achar o valor de RL novamente, agora via simulação, como mostra de forma geral a Figura 4.

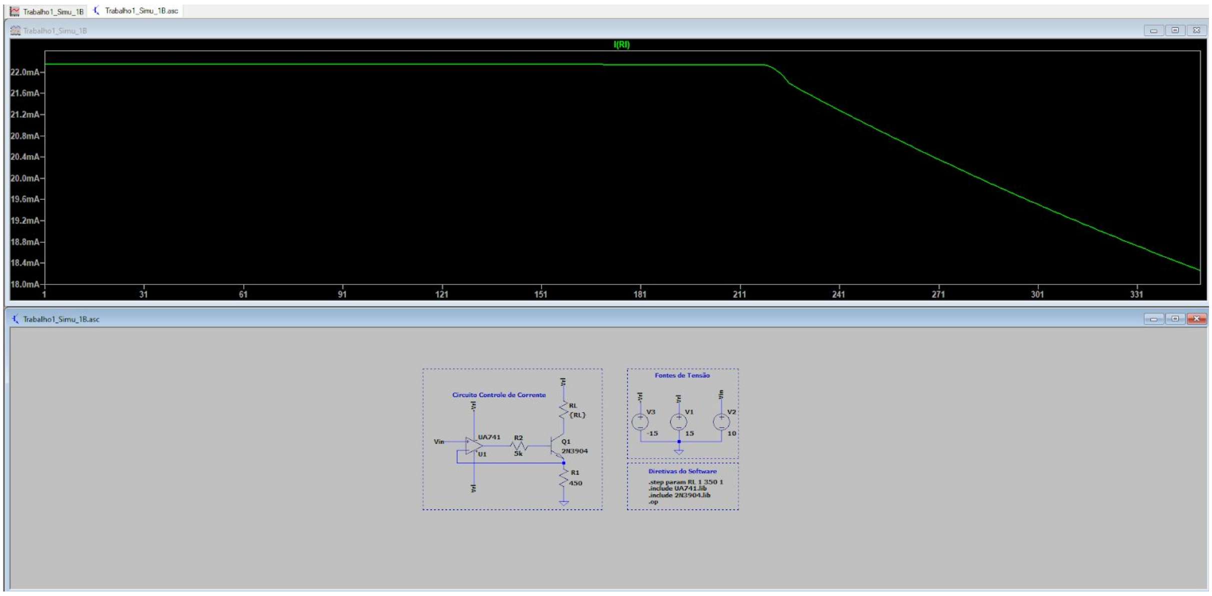


Figura 4: Simulação no LTspice do Circuito.

2 - A Figura 5 representa nosso circuito em questão simulado, como se pode ver, foram utilizadas as mesmas nomenclaturas de variáveis da parte 1 deste presente trabalho para ele.

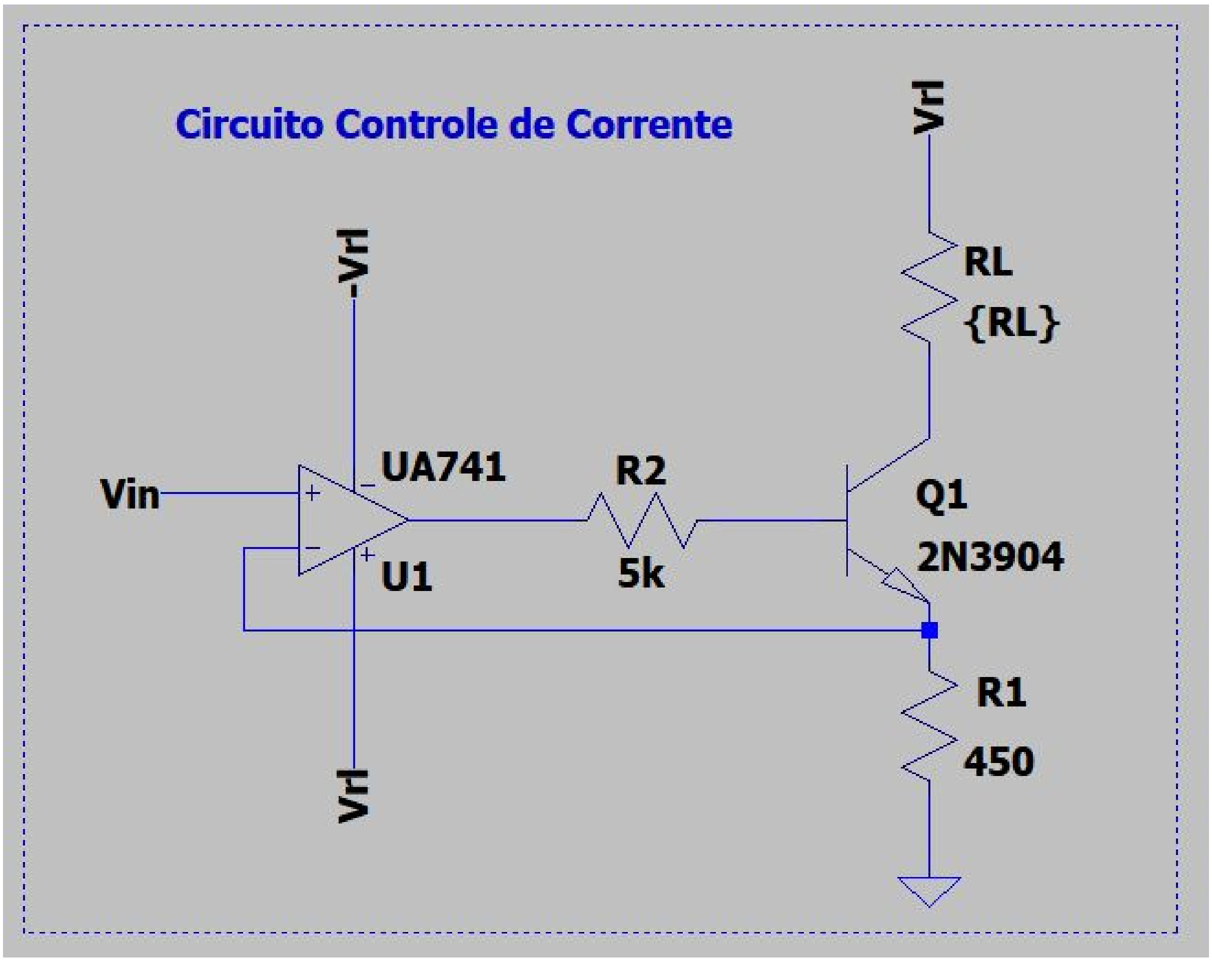


Figura 5: Circuito simulado.

Na Figura 6 estão as fontes de tensão utilizadas de acordo com as informações obtidas na Figura 2 da parte 1.

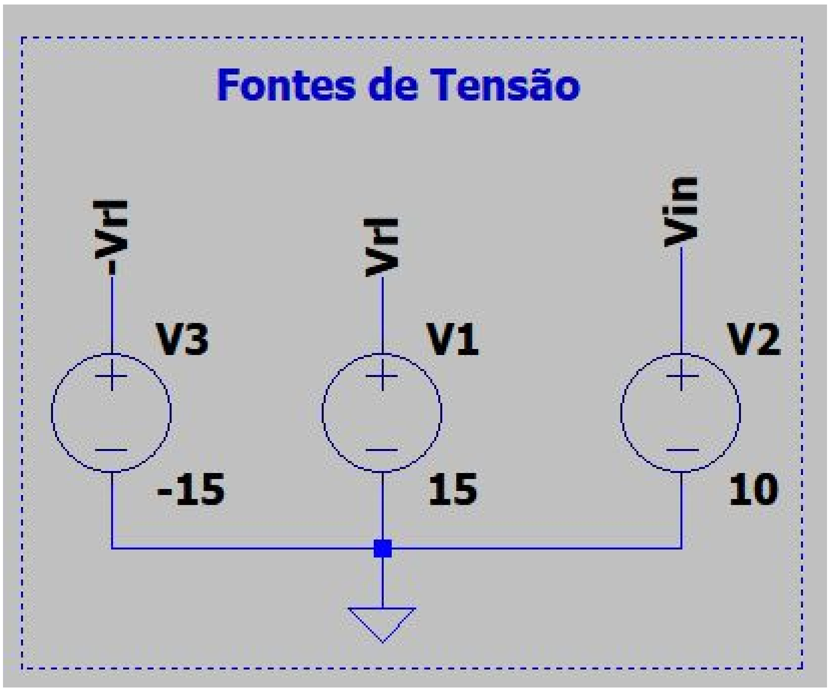


Figura 6: Fontes de tensão utilizadas no Software.

Na Figura 7 estão as diretivas utilizadas para a simulação, sendo:

1. ".step para RL 1 350 1" - O parâmetro de simulação de nosso resistor, onde o primeiro 1 da série " 1 350 1 " significa o valor inicial, 350 o valor final de que nossa resistência poderá variar na simulação e o último 1 o incremento da resistência, ou seja de quanto obtemos valores no gráfico a partir de nossa amostragem de corrente, vale frisar que foi escolhido o valor de 350 em nossa amostragem devido ao valor teórico obtido na primeira parte.

2. ".include UA741.lib" - Diretiva para incluir a biblioteca do amplificador operacional UA741, como estava na mesma pasta de simulação que o circuito, não foi preciso descrever todo o caminho.

3. ".include 2N3904.lib" - Assim como o ampop, esta diretiva faz a mesma coisa com o transistor 2N3904.

4. ".op" - É uma modalidade de simulação do software que mostra o gráfico, onde o usuário pode clicar (no circuito) ou adicionar a variável desejada para análise ou mostra os resultados dos pontos importantes do circuito.

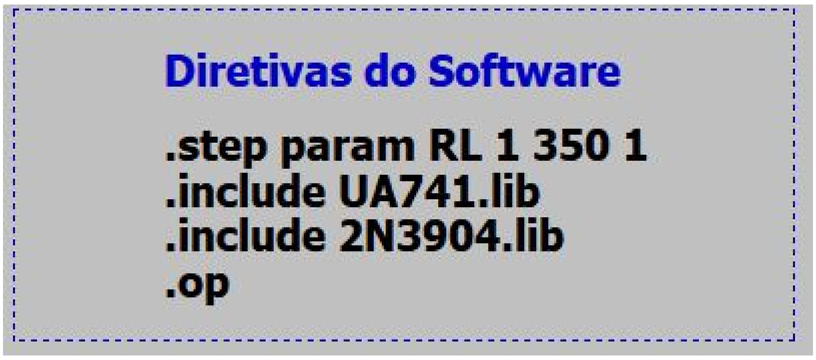


Figura 7: Diretivas de simulação.

Resultados:

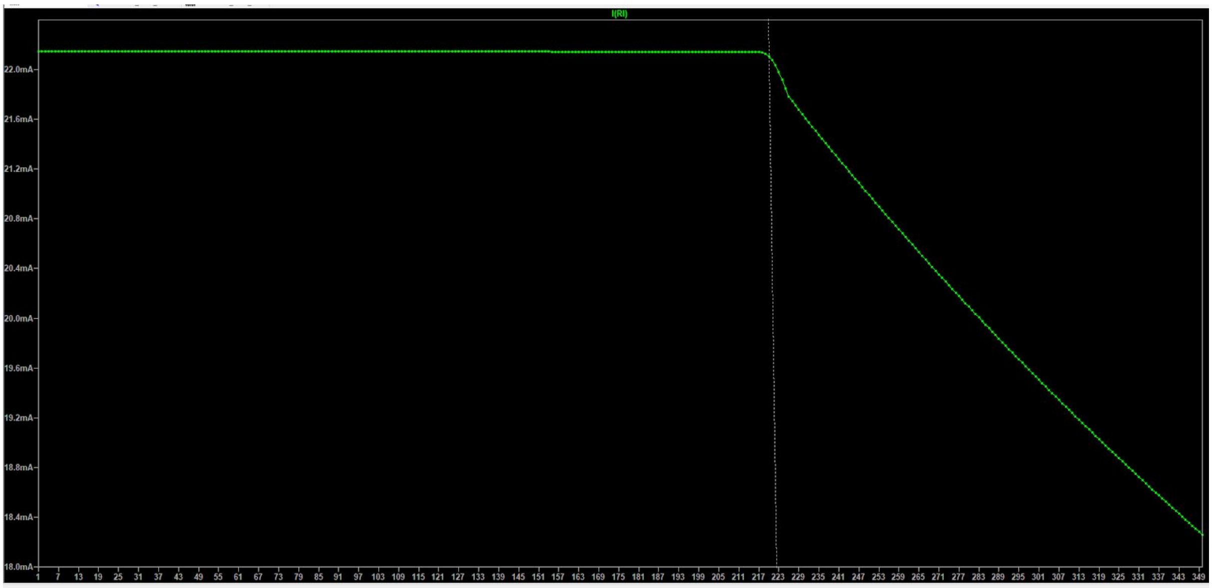


Figura 8: Gráfico valor da corrente sobre $RL\ (Ic)$ x RL.

No gráfico da Figura 8 podemos ver de forma geral o comportamento da corrente elétrica conforme a resistência de nossa carga RL aumenta, tendendo essa corrente a 0, a partir desse gráfico iremos gerar outros para que o leitor possa obter uma melhor visualização dos resultados.

No gráfico da Figura 9, está o valor da corrente inicial que é constante até o valor de RL, que já era esperada devido aos cálculos feitos na parte 1.

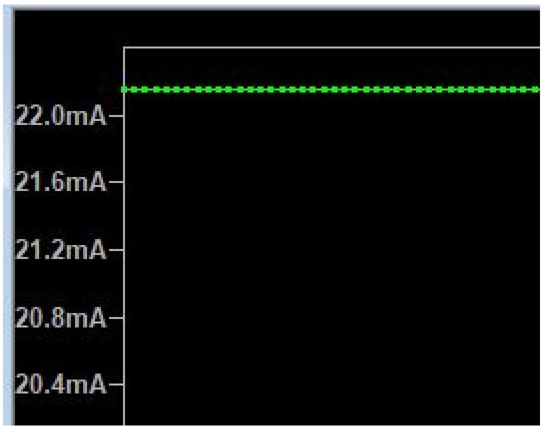


Figura 9: Corrente Ic em cima da Carga RL.

Na Figura 10 podemos ver o valor de resistência da carga esperado coincidindo com o valor calculado, pela linha “reta” do gráfico com o valor de aproximadamente $222\,\Omega$, a partir desse valor de carga, podemos perceber pela Figura 8, que o circuito já não mantém a corrente desejada.

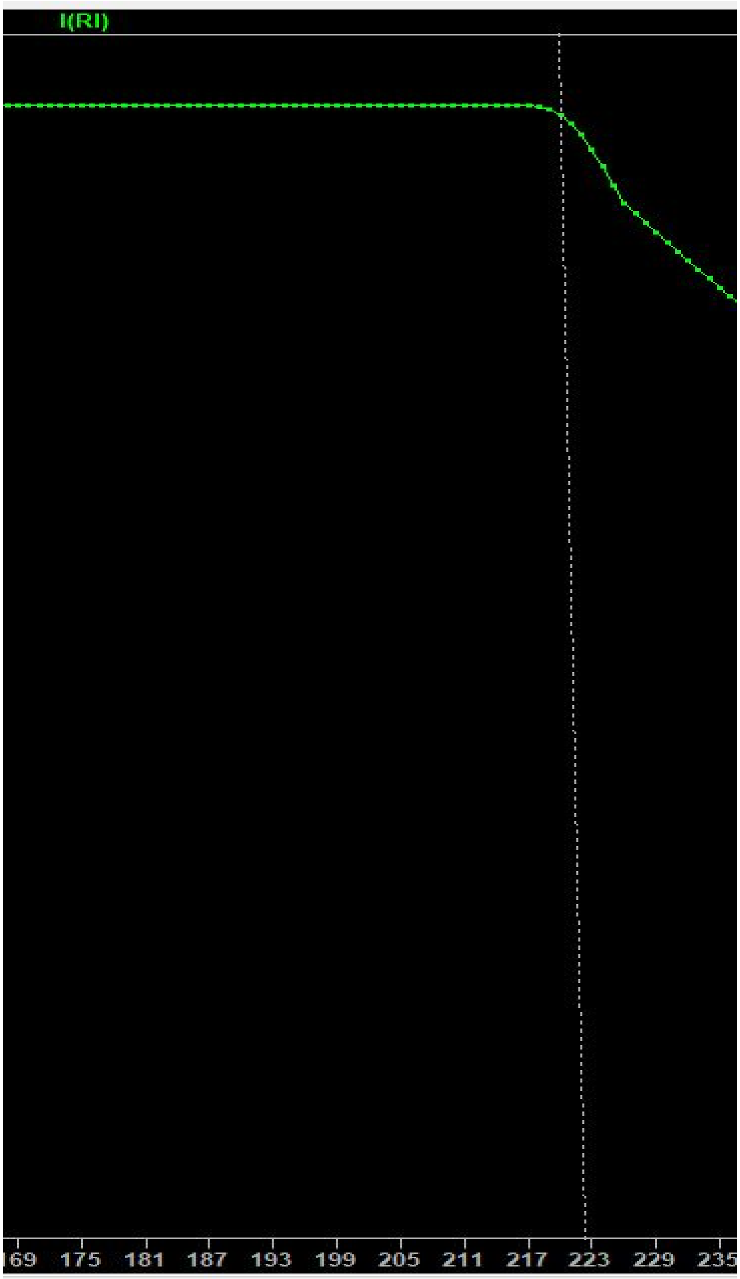


Figura 10: Momento da queda de corrente Ic.

Enfim, podemos concluir sob essas condições um valor máximo de carga de aproximadamente $RL = 222\,\Omega$.

Questão 2:

Na questão 2 é dado um circuito integrado muito utilizado em instrumentação modelo INA128, sendo sua principal função proporcionar um alto valor de CMRR, onde este índice define a tensão de saída do circuito, a problemática está quando os valores de resistência dentro deste circuito possui incerteza em seus valores diminuindo o valor de CMRR, pede-se então que seja simulado o circuito da Figura 11 utilizando o software LTspice e mesmo modelo de amplificador operacional da Questão 1.

1 - A partir da simulação obter o $CMRR_{dB}$ do circuito para obter A_{cm} considerando as equações da Figura 12, utilizando uma tensão DC igual em V_{in}^{+} e V_{in}^{-} para isto.

$$\mathrm{CMRR}_{\mathrm{dB}} = 20\log_{10}\left(\frac{\mathrm{A_d}}{|\,\mathrm{A_{cm}}\,|}\right) \quad (1)$$

com A_d e A_{cm} denotando, respectivamente, o ganho em modo diferencial e o ganho em modo comum. Tais valores de ganho definem a tensão de saída do circuito conforme segue:

$$V_0 = \mathrm{A_d} V_d + \mathrm{A_{cm}} V_{cm} \quad (2)$$

com

$$V_d = V_{in}^{+} - V_{in}^{-} \quad (3)$$

e

$$V_{cm} = \frac{V_{in}^{+} + V_{in}^{-}}{2}. \quad (4)$$

Figura 12: Equações para questão 2.

2 - Considerando 4 resistores de $40k\Omega$ apresentando incerteza no valor de resistência, fazer uma análise do pior caso de $CMRR_{dB}$ nos valores de 1% e 5%, seguindo a dica fornecida expressa na Figura 13.

Dica: a análise de pior caso nesta simulação pode ser realizada de duas diferentes maneiras: 1 – via simulação de Monte Carlos (com várias realizações, acima de 500); 2 – usando os valores limites dos resistores conforme sugerido no link https://www.analog.com/en/technical-articles/ltspice-worst-case-circuit-analysis-with-minimal-simulations-runs.html. Em ambos os casos use o modo de simulação DC *operation point* para facilitar a análise.

Figura 13: Dica parte 2 Questão 2.

1 - Para obter o valor de $CMRR_{dB}$ precisamos considerar a equação abaixo:

$$CMRR_{dB} = 20\ log(\frac{A_d}{|A_{cm}|})\ (ll)$$

Pela equação acima teremos que calcular os valores de A_d e A_{cm} , a relação de A_d é dada pela expressão abaixo:

$$A_d = 1 + \frac{50k}{R_g}\ (lll)$$

Onde R_g é o resistor de precisão conectado pelo usuário para controlar o ganho do amplificador de instrumentação.

Conforme pedido no enunciado e ilustrado na Figura 14, utilizaremos $R_g = 150\Omega$.

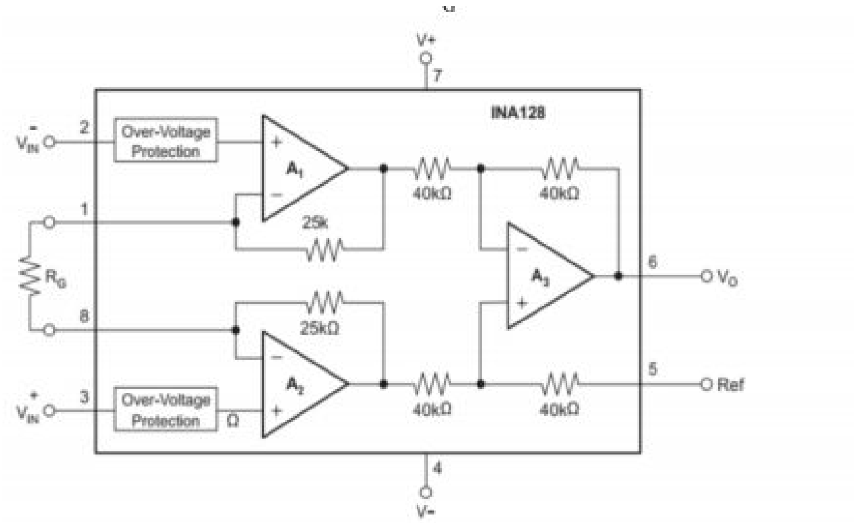


Figura 2 – diagrama esquemático do amplificador de instrumentação INA128.
Fonte: https://www.ti.com/product/INA128

Simule o circuito da Figura 2 no LTspice. Considere o amplificador operacional uA741 (modelo disponível no Moodle), R_G = 150 Ω e tensão de alimentação ±15 V. A simulação pode ser configurada para o modo DC *operation point*. Em seguida, determine via simulação os itens a seguir.

Figura 14: Circuito a ser simulado e valores importantes a serem utilizados.

Obtendo valor de R_g e utilizando-se da equação (lll) e substituindo os valores, conseguiremos encontrar A_d.

$$A_d = 1 + \frac{50\,x\,10^3}{150} = 334,33$$

Com A_d, agora precisamos achar A_{cm}, iremos utilizar a equação abaixo para isto.

$$V_o = A_d V_d + A_{cm} V_{cm}$$

Isolando A_{cm}, temos:

$$A_{cm} = \frac{V_o - A_d V_d}{V cm}$$

Sabemos que $V_d = V_{in}^{+} - V_{in}^{-}$, mas como iremos utilizar $V_{in}^{+} = V_{in}^{-}$ conforme o enunciado, então, podemos ter uma fórmula simplificada para A_{cm}, demonstrada abaixo.

$$A_{cm} = \frac{V_o}{Vcm}$$

Onde Vcm é dado por:

$$Vcm = \frac{V_{in}^{+} + V_{in}^{-}}{2}$$

Arbitrariamente iremos utilizar a tensão de entrada como $10\ V$, logo, $V_{in}^{+} = V_{in}^{-} = 10\ V$, então, $Vcm = \frac{10+10}{2} = 10\ V$, portanto:

$$A_{cm} = \frac{V_o}{10} \quad (Iv)$$

Restando apenas a variável V_o para ser encontrada, onde para isso, iremos simular o circuito da Figura 14, considerando, $V_{in}^{+} = V_{in}^{-} = 10\ V$ e a tensão de entrada no amplificador operacional de $Vamp = 15\ V$, seguindo a especificação da Figura 14.

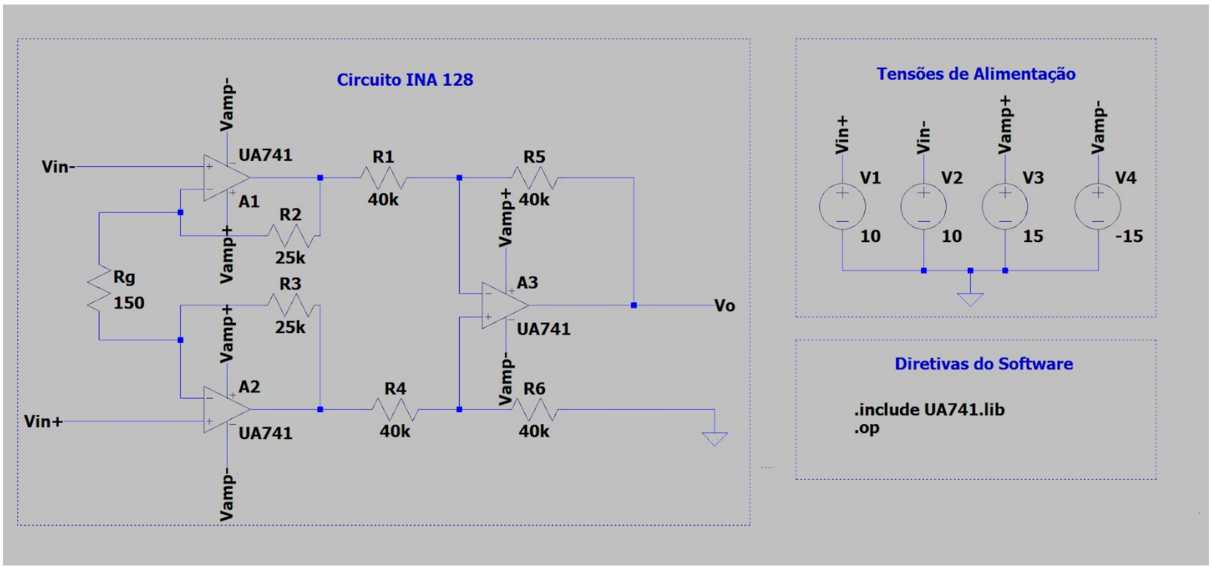
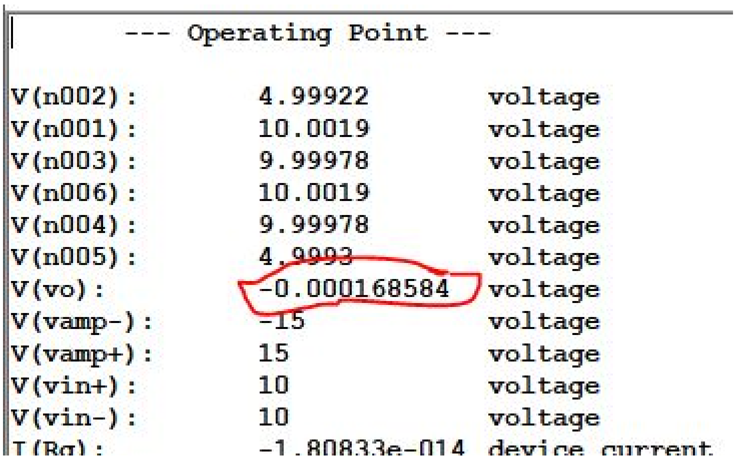


Figura 15: Simulação do Circuito.

Feito a simulação do circuito, conforme a Figura 15, e os detalhes já explicados na Questão 1, conseguimos obter o valor de V_o mostrado na Figura 16 e circulado em vermelho.



--- Operating Point ---

V(n002):	4.99922	voltage
V(n001):	10.0019	voltage
V(n003):	9.99978	voltage
V(n006):	10.0019	voltage
V(n004):	9.99978	voltage
V(n005):	4.9993	voltage
V(vo):	-0.000168584	voltage
V(vamp-):	-15	voltage
V(vamp+):	15	voltage
V(vin+):	10	voltage
V(vin-):	10	voltage
I(Rg):	-1.80833e-014	device current

Figura 16: Resultado de V_o.

Agora, obtendo V_o, podemos finalmente calcular o valor de A_{cm}, utilizando a expressão (lv), logo.

$$A_{cm} = \frac{V_o}{10} = \frac{-0{,}000168584}{10} = -0{,}0000168854 = -16{,}8854\ x\ 10^{-6}$$

Usando a equação (ll) conseguiremos obter $CMRR_{dB}$, logo:

$$CMRR_{dB} = 20\ log\left(\frac{334{,}33}{16{,}8854\ x\ 10^{-6}}\right) = 145{,}94$$

2 - Para esta segunda etapa iremos utilizar o método 1, utilizando a função de monte carlos, fazendo 601 iterações e com tolerância de 0,01 %,sendo assim, temos a primeira simulação demonstrada na Figura 17 abaixo:

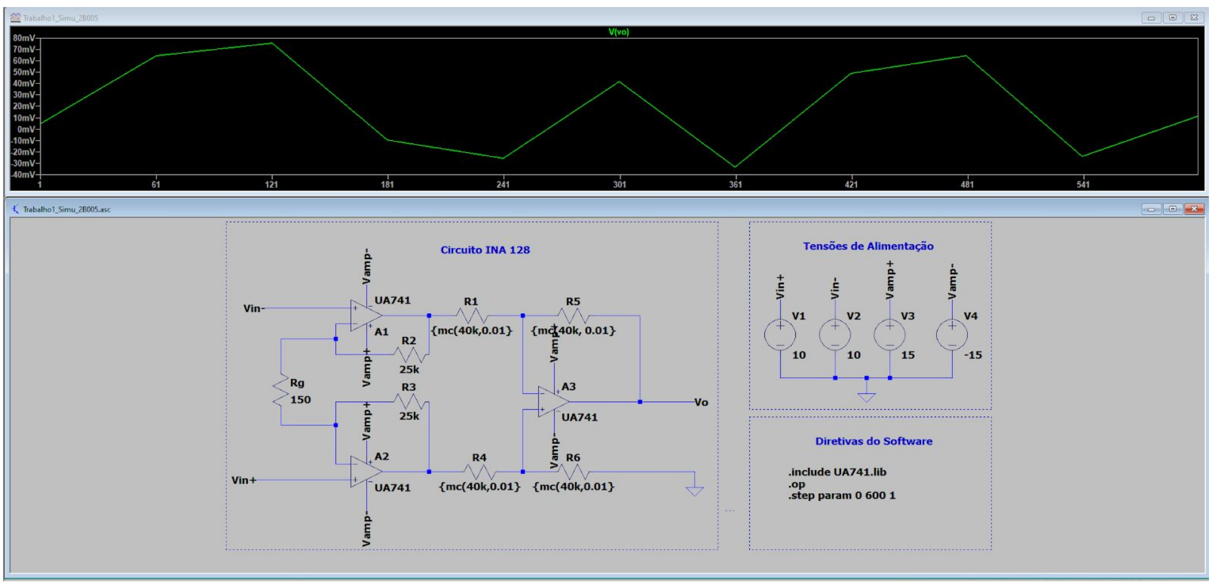


Figura 17: Visão geral da simulação 0,01% de tolerância.

A partir desta simulação conseguimos obter o maior número em módulo para V_o e assim, conseguiremos obter o $CMRR_{dB}$.

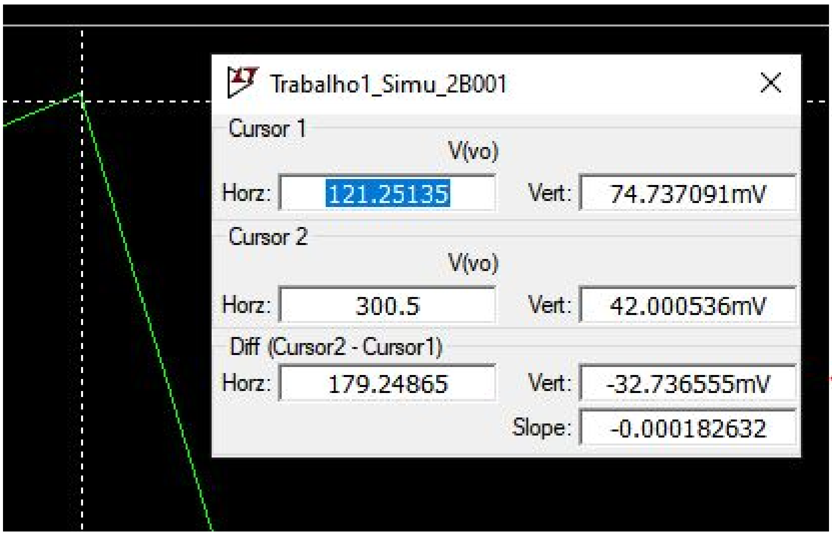


Figura 18: Valor de V_o com tolerância de 0,01 %.

Logo, conseguimos obter o $CMRR_{dB}$, utilizando o mesmo procedimento da primeira parte.

$$A_{cm} = \frac{V_o}{10} = \frac{-74,737091\ x\ 10^{-3}}{10} = 74,737091\ x\ 10^{-4}$$

$$CMRR_{dB} = 20\ log(\frac{334,33}{74,737091\ x\ 10^{-4}}) = 93,01$$

O circuito deste procedimento está demonstrado na Figura 19 abaixo.

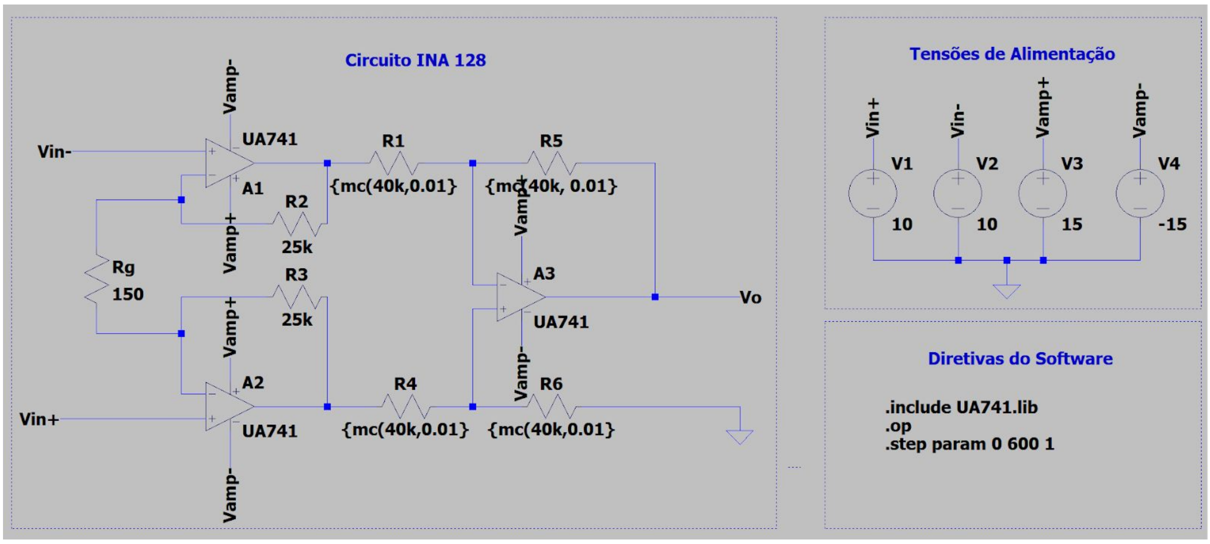


Figura 19: Circuito de tolerância 0,01% ampliado.

Realizando o mesmo procedimento da tolerância de 0,01%, temos Visão geral da simulação para tolerância de 0,05%, então, foi feita a mesma simulação para tolerância de 0,05 %, mostrada de forma geral na Figura 20.

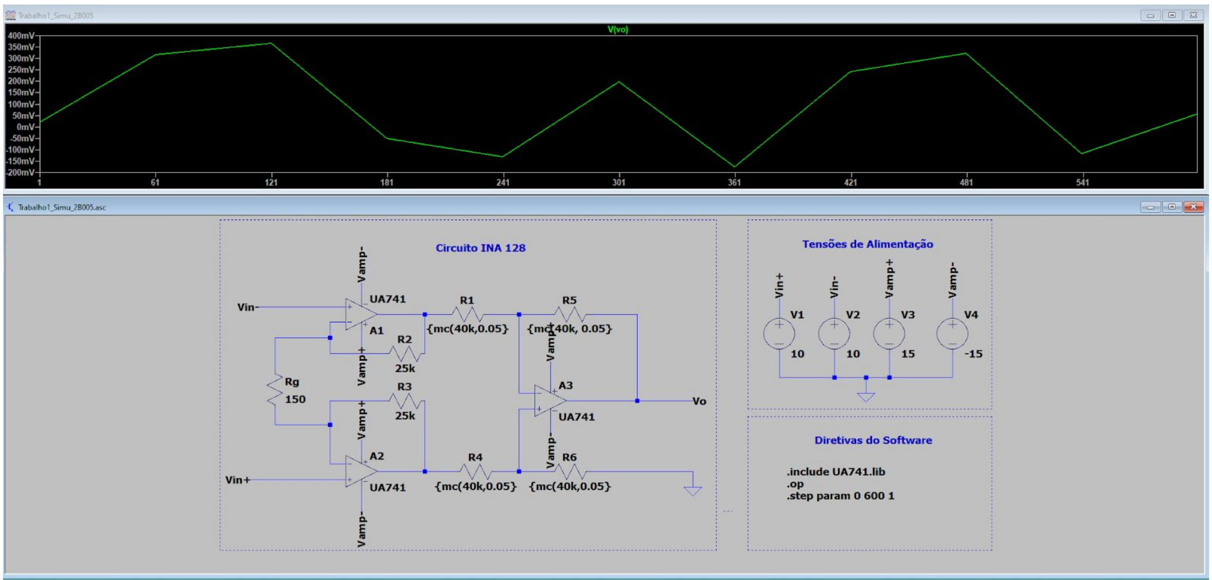


Figura 20: Simulação geral tolerância 0,05%.

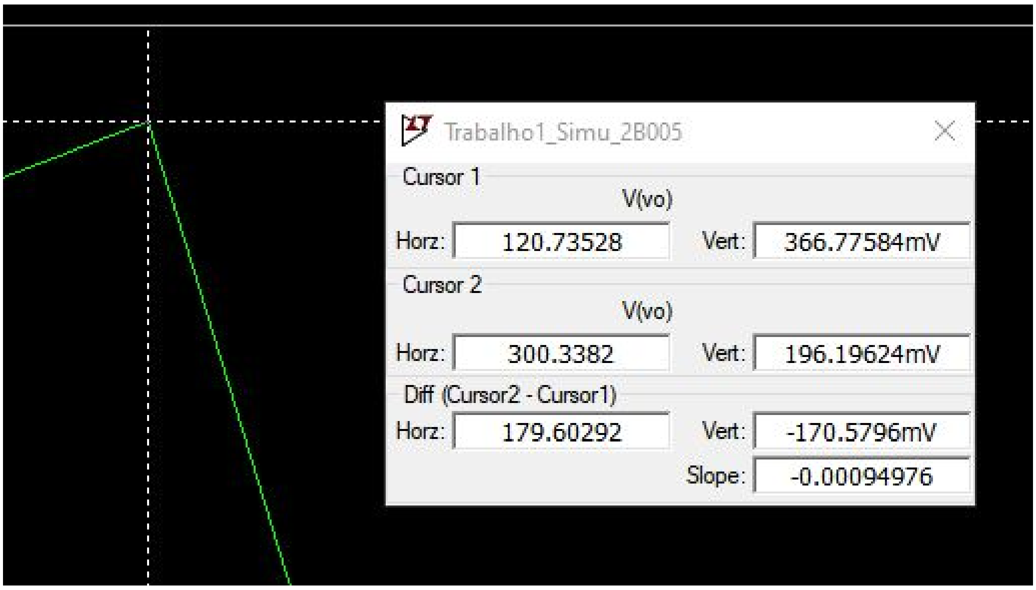


Figura 21: Máximo ponto da V_o para tolerância de 0,05%.

Com isso podemos calcular $CMRR_{dB}$.

$$A_{cm} = \frac{V_o}{10} = \frac{366{,}77584 \; x \; 10^{-3}}{10} = 36,677584 \; x \; 10^{-3}$$

$$CMRR_{dB} = 20 \; log(\frac{334{,}33}{36{,}677584 \; x \; 10^{-3}}) = 79,19$$

Segue o circuito de forma mais ampla utilizando a função de Monte Carlos, na Figura 22.

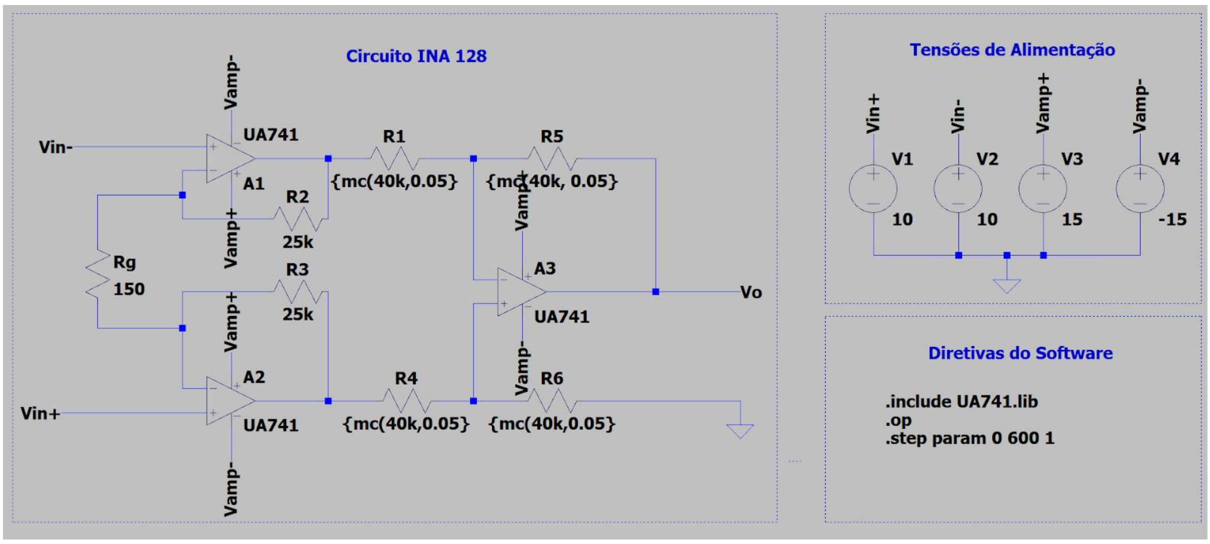


Figura 22: Circuito ampliado tolerância 0,05%.

Conclusão:

Pode-se concluir que quanto maior a incerteza do resistor, menor será o nosso $CMRR_{dB}$, fazendo com que o amplificador em questão seja menos eficiente, isto explica a afirmação inicial do trabalho, em dizer que estes circuitos vem em circuitos integrados com resistores de alta precisão, justamente para diminuir esse efeito que acarreta em um maior $CMRR_{dB}$.

Trabalho Parte 2:

Nesta etapa iremos trabalhar com quantizadores de sinal utilizando a linguagem Python para descrever estes sinais.

Para o presente trabalho foi utilizado a IDE Thonny e as bibliotecas numpy e matplotlib, e uma função disponibilizada pelo professor “quantize”, demonstrado isto na Figura 23.

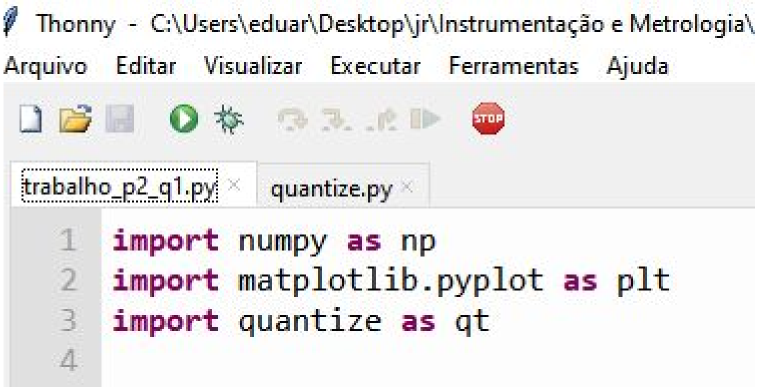


Figura 23: Bibliotecas e IDE.

1 - Nesta primeira etapa foi pedido a partir de um sinal disponibilizado pelo professor “sample.npy” fosse obtido um gráfico de comparação escolhendo a quantização midtread ou midrise e utilizando número de bits = 3.

Foi-se escolhido o midtread pelo fato do gráfico ficar mais amigável com o sinal original (após feito o teste com os 2), na Figura 24, está o resultado dessa primeira simulação e na Figura 25 o código fonte utilizado para tal.

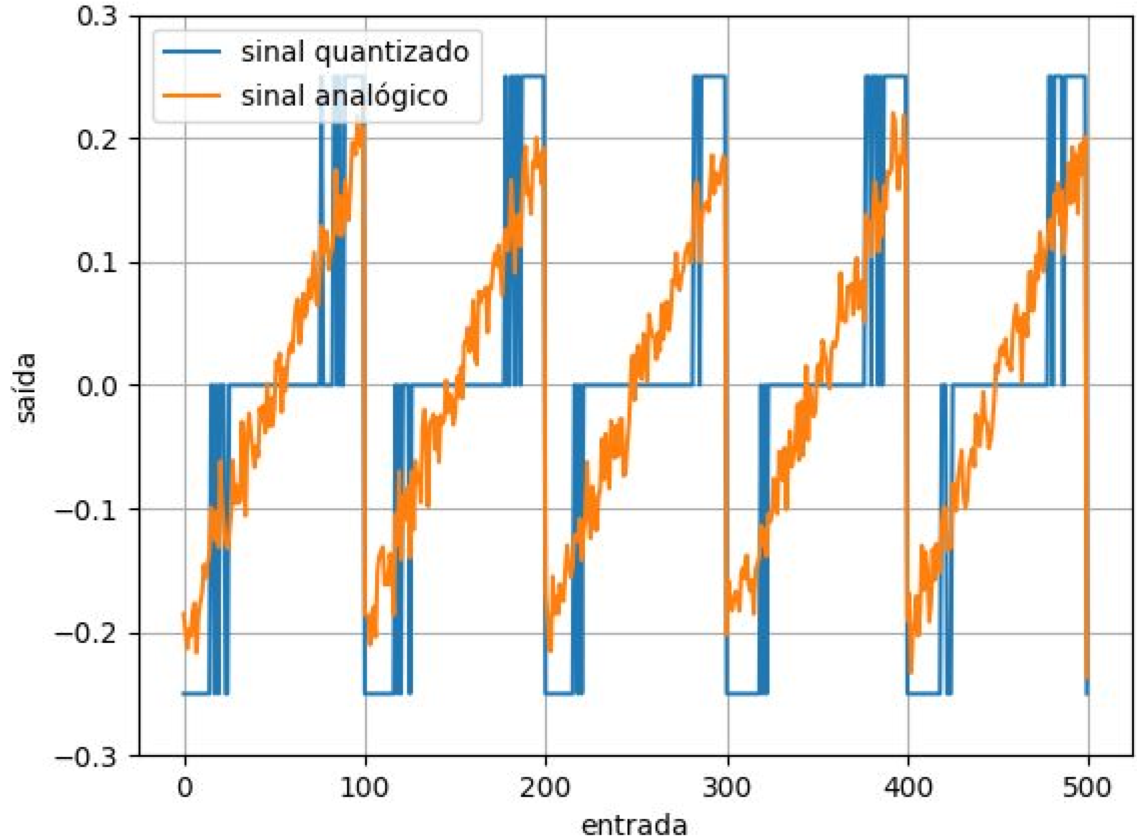


Figura 24: Sinal quantizado e analógico.

```
import numpy as np
import matplotlib.pyplot as plt
import quantize as qt

s_analog = np.load ('sample.npy') #importa o sinal análogico

n = 3 # número de bits
s_analog_midrise = qt.quantize(s_analog, n, 'midtread')
plt.plot(s_analog_midrise, label = 'sinal quantizado')
plt.plot (s_analog, label = 'sinal análogico')

plt.legend(loc='best')
plt.grid()
plt.xlabel('entrada')
plt.ylabel('saída')
plt.ylim(-0.3, 0.3)
plt.show()
```

Figura 25: Código fonte.

2 - Na segunda etapa, foi pedido para ajustar o sinal para que ocupasse a todo eixo Y entre -1 e 1.

Para isso, bastava a gente olhar o pico do sinal anterior e definir um multiplicador para todo o sinal que fizesse com que o novo pico fosse até + ou - 1, pelo gráfico da Figura 24, cheguei a conclusão em número próximo a ± 22. portanto, foi-se multiplicado o sinal por 4,5, com o resultado explicitado na Figura 26 e código fonte na Figura 27.

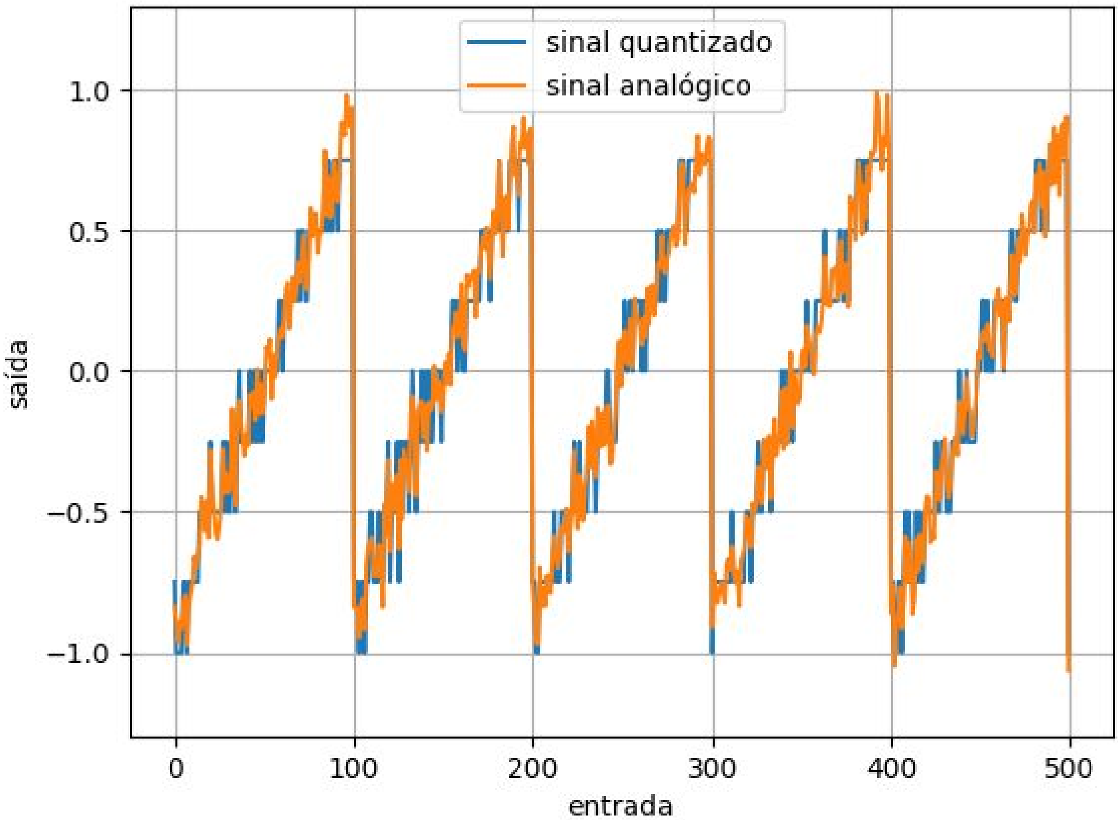


Figura 26: Sinais da etapa 2.

```python
import numpy as np
import matplotlib.pyplot as plt
import quantize as qt

s_analog = 4.5 * np.load ('sample.npy') #importa o sinal análogico

n = 3 # número de bits
s_analog_midrise = qt.quantize(s_analog, n, 'midtread')
plt.plot(s_analog_midrise, label = 'sinal quantizado')
plt.plot (s_analog, label = 'sinal analógico')

plt.legend(loc='best')
plt.grid()
plt.xlabel('entrada')
plt.ylabel('saída')
plt.ylim(-1.3, 1.3)
plt.show()
```

Figura 27: Código fonte etapa 2.

3 - É pedido uma comparação entre 1 e 2.

A partir das duas experiências, pudemos observar que o gráfico do sinal quando expandido representou melhor o sinal analógico, onde pode-se concluir que quanto maior a expansão do sinal, maior será sua representatividade quando quantizado.

4 - É pedido a partir do sinal original a potência descrita pela Figura 28.

$$P_a = \frac{1}{K}\sum_{k=0}^{K-1} a^2(k)$$

onde K denota o número de amostras de $a(k)$.

Figura 28: Fórmula da potência.

O processo de quantização gera um erro em relação ao sinal original, podendo ser modelado como um sinal de ruído dado por

$$z(k) = y(k) - x(k) \tag{2}$$

onde $x(k)$ e $y(k)$ denotam, respectivamente, a amostra na entrada e na saída do quantizador. Dessa forma, é possível determinar a SNR (*signal-to-noise ratio*) resultante do processo de quantização, a qual é definida como segue:

$$SNR_{\mathrm{dB}} = 10\log_{10}\left(\frac{P_x}{P_z}\right) \tag{3}$$

com P_x e P_z representando a potência média de $x(k)$ e $z(k)$, respectivamente.

Figura 29: Dica da questão.

Para calcular a potência, utilizamos as dicas da questão descritas no trabalho, conforme mostrado em Figura 29, primeiramente foi-se calculado a potência média dos sinais (entrada e saída) e depois, achado o erro, com tudo isso em mãos, aplicamos a relação entre SNR_{dB} e as potências médias do erro e da entrada, obtendo o seguinte resultado, mostrado na Figura 30 com código fonte demonstrado na Figura 31.

```
35
36
37
Shell ×
>>> %Run trabalho_p2_q4.py

 SNRdB do bit 2 = 0.0
 SNRdB do bit 3 = 8.75717952370265
 SNRdB do bit 4 = 10.579560308558209
 SNRdB do bit 5 = 12.28159697540838
 SNRdB do bit 6 = 13.1555327827095
 SNRdB do bit 7 = 13.974308844702268
 SNRdB do bit 8 = 14.697570189473328

>>>
```

Figura 30: Resultados de SNR_{dB} por Bit.

```python
import numpy as np
import quantize as qt

s_analog = np.load ('sample.npy') #importa o sinal análogico

x = 0 #sinal de entrada
y = 0 #sinal de saída
xfinal = 0 #efeito somatório
yfinal = 0 #efeito somatório
xfinal2 = 0 #efeito somatório
yfinal2 = 0 #efeito somatório


lista_bits = [2, 3, 4, 5, 6, 7, 8]

for n in lista_bits:
    s_analog_qnt = qt.quantize(s_analog, n, 'midtread')

    for t1 in range(0,100,1): #Considerando o número de amostras igual a 100
        y = np.square(s_analog_qnt)
        yfinal = yfinal + y

    for t2 in range(0,100,1): #Considerando o número de amostras igual a 100
        x = np.square(s_analog)
        xfinal = xfinal + x

    xfinal2 = np.sum(xfinal)/100 # potência media do sinal de entrada
    yfinal2 = np.sum(yfinal)/100 # potência media do sinal de saída
    z = abs(yfinal2 - xfinal2)  # potência media do erro (pois log não admite valores negativos)
    print("SNRdB do bit " + str(n) + " = " + str(10*np.log10(xfinal2/z)))
```

Figura 31: Código fonte dos resultados.

5 - Obter os mesmos resultados, porém o sinal amplificado, e comentar os resultados.

Para isto, foi amplificado o nosso sinal em 5x novamente obtendo o resultado demonstrado pela Figura 32 e código fonte Figura 33, tendo um diferencial mostrado na Figura 32, do sinal original entre o range [-1 e 1].

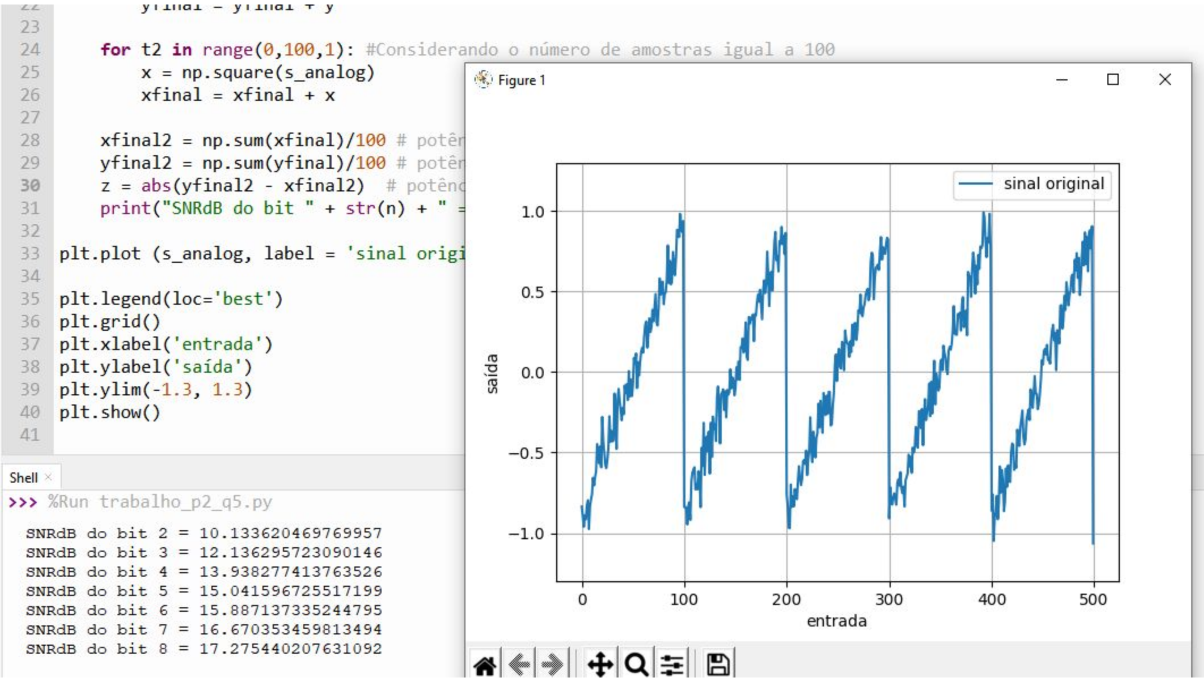


Figura 32: Resultados da Questão 5.

```python
import numpy as np
import quantize as qt
import matplotlib.pyplot as plt

s_analog = 4.5 * np.load ('sample.npy') #importa o sinal análogico

x = 0 #sinal de entrada
y = 0 #sinal de saída
xfinal = 0 #efeito somatório
yfinal = 0 #efeito somatório
xfinal2 = 0 #efeito somatório
yfinal2 = 0 #efeito somatório

lista_bits = [2, 3, 4, 5, 6, 7, 8]

for n in lista_bits:
    s_analog_qnt = qt.quantize(s_analog, n, 'midtread')

    for t1 in range(0,100,1): #Considerando o número de amostras igual a 100
        y = np.square(s_analog_qnt)
        yfinal = yfinal + y

    for t2 in range(0,100,1): #Considerando o número de amostras igual a 100
        x = np.square(s_analog)
        xfinal = xfinal + x

    xfinal2 = np.sum(xfinal)/100 # potência media do sinal de entrada
    yfinal2 = np.sum(yfinal)/100 # potência media do sinal de saída
    z = abs(yfinal2 - xfinal2)  # potência media do erro (pois log não admite valores negativos)
    print("SNRdB do bit " + str(n) + " = " + str(10*np.log10(xfinal2/z)))

plt.plot (s_analog, label = 'sinal original')

plt.legend(loc='best')
plt.grid()
plt.xlabel('entrada')
plt.ylabel('saída')
plt.ylim(-1.3, 1.3)
plt.show()
```

Figura 33: Código Fonte da Questão 5.

Comentários:

Comparando as Figuras 32 e 30, percebemos um nível de SNR_{dB} maior, conforme ampliamos o nível do sinal, pela fórmula de SNR_{dB} , podemos concluir que quanto maior o nosso erro (z), menor será o SNR_{dB} , logo, como os resultados de SNR_{dB} foram maiores para o segundo teste, podemos concluir que o efeito do ruído é mais notório em sinais menos ampliados.