Monitoria em Sistemas de Controle para Engenharia

Coordenador: Cleison Silva

Monitor - Bolsista: Felipe Piano (Turma 2017)

Resposta Sistema de Primeira Ordem

Objetivo:

- Encontrar, analisar e validar uma função de transferência de primeira ordem para a Planta Motor-Gerador.

Introdução

Nesta experiência, analisaremos o comportamento dinâmico e estático de um motor de corrente contínua real. Verificaremos que o mesmo apresenta características não-lineares e veremos como modelar matematicamente o motor por uma Função de Transferência de primeira ordem em torno de um ponto de operação. Compararemos a dinâmica do motor com a do modelo determinado, tanto no domínio do tempo quanto da frequência

Encontrar um modelo matemático que capture as características dinâmicas e estáticas relevantes de um sistema real, é de fundamental importância para a análise e controle do sistema. Muitas vezes o modelo pode ser obtido a partir das leis da física (mecânica, eletromagnetismo, etc), e os valores dos parâmetros dependem de constantes e coeficientes físicos (coeficiente de atrito, indutância, etc). Em situações reais, não conheceremos o valor exato dos parâmetros do modelo, tendo na melhor da hipóteses uma estimativa para os mesmos. Por exemplo, todo resistor possui um valor nominal e uma faixa de tolerância percentual (e.g. $R = 100\Omega \pm 5\%$). Além disso, muitas vezes a determinação de um modelo matemático para um sistema a partir de leis naturais é extremamente difícil e, mesmo no caso em que isso é possível, o modelo obtido pode ser demasiadamente complexo para ser estudado matematicamente.

Devido às dificuldades que acabamos de expor, em geral buscamos um modelo matemático relativamente simples, mas que capture, ao menos aproximadamente, as características dinâmicas e estáticas relevantes do sistema. Assim, primeiramente fixamos um modelo (modelagem do sistema), e em seguida determinamos de maneira aproximada o valor de seus parâmetros (identificação dos parâmetros).

Idealmente, um motor de corrente contínua (CC) é um sistema linear. Na prática, no entanto, o mesmo é um sistema não-linear devido ao fato de que os elementos elétricos e mecânicos que

o constituem não serem ideais, e também por causa da existência de outros fenômenos físicos que foram desprezados na equação diferencial que descreve sua dinâmica. Desse modo, em geral, desejamos obter um modelo linear (Função de Transferência) para um motor CC que capte as principais características de sua dinâmica em torno de um ponto de operação.

Revisão Teórica

Toda Funcao de Transferéncia $G(s)$ de primeira ordem pode ser escrita como

$$G(s) = \frac{K}{\tau s + 1} \tag{1}$$

Suponha que $G(s)$ é estavel, ou seja, $\tau > 0$. Considere uma entrada $u(t) = A$ do tipo degrau de magnitude A. Temos que a saida correspondente é

$$y(t) = KA(1 - e^{-\frac{t}{\tau}}),\ \forall t \geq 0 \tag{2}$$

O valor da saída em regime permanente é

$$y(\infty) = KA \tag{3}$$

e o instante de tempo $t(63\%)$ para que se atinja 63% de $y(\infty)$ é dado por

$$0,6321KA = KA(1 - e^{-\frac{\tau}{\tau}}) = \tau \tag{4}$$

Logo,

$$K = \frac{y(\infty)}{A}$$

e

$$\tau = t(63\%).$$

Exercício

1. Mostre que para uma entrada degrau de amplitude A a resposta do sistema (1) é dada pela equação (2).
2. Assuma que $y(\infty) = 3$ para um degrau de 0.5 de amplutude. Determine o o valor do ganho do sistema ($K = \ ?$).
3. Para $K = 1,63$ e $A = 2$. Determine a constante de tempo do sistema.

Identificação de um modelo de primeira ordem em torno de um ponto de operação

Considere que a entrada $u(t)$ de um sistema normalmente opera em torno de $\bar{u}$, e que a saída $y(t)$ correspondente opera em torno de $\bar{y}$. Denominamos $\bar{u}$ de entrada de operação, $\bar{y}$ de saída

de operação, e o par $(\bar{u}, \bar{y})$ de ponto de operação.

A Figura 2 ilustra o procedimento para realizarmos a identificação experimental de um modelo $G(s)$ de primeira ordem para o sistema em torno do ponto de operação $(\bar{u}, \bar{y})$. Isto significa que tal modelo é válido desde que $u(t) \approx \bar{u}$ e $y(t) \approx \bar{y}$, ou seja, $\Delta u(t) = u(t) - \bar{u} \approx 0$, $\Delta y(t) = y(t) - \bar{y} \approx 0$ (pequenos variações na entrada e na saída!). Temos então que

$$\boxed{G(s) = \frac{\Delta Y(s)}{\Delta U(s)} = \frac{K}{\tau s + 1}, \; K = \frac{\delta y}{\delta u}, \; \tau = t(63\%)}$$

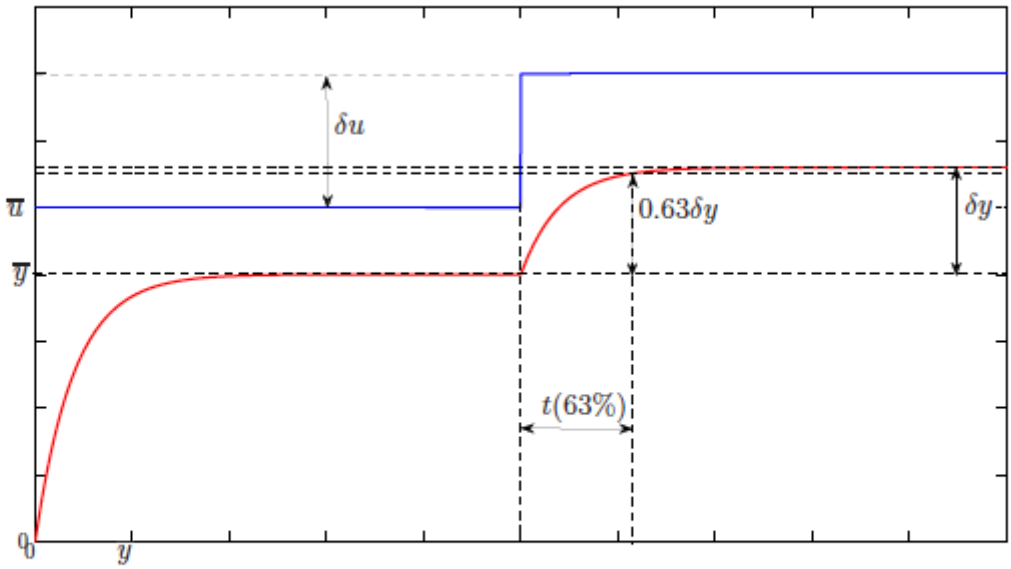


Figura 2: Identificação de primeira ordem em torno do ponto de operação $(\overline{u}, \overline{y})$.

1. Abra o arquivo *Motor_Gerador.py no VSCode program*. Aplique no Motor-Gerador as tensões correspondentes a 20%, 50% e 80% do valor fornecido pela Fonte de Alimentação, por exemplo, para uma Fonte de Alimentação de $20V$, tem-se:

$$\bar{u}_{20\%} = 4V, \bar{u}_{50\%} = 10V, \bar{u}_{80\%} = 16V$$

durante 10 segundos. Observe que Motor-Gerador real é não-linear, apresentado uma zona morta. Note também a presença de ruídos no sinal medido.

2. Identifique um modelo linear de primeira ordem para o Motor-Gerador em torno do ponto de operação

$$\bar{u}_{50\%} = 10.$$

Para isto, aplique

$$\delta u_1 = 1$$

em

$$t = 10.$$

Para identificar os parâmetros do modelo $G(s)$ de primeira ordem com base na Figura 1 e em (2). Determine também $\bar{y}$ (saída de operação). Compare a dinâmica de $G(s)$ com a do Motor-Gerador. Devido à imprecisão na medição e ao modelo simplificado de primeira ordem, talvez seja necessário fazer um "ajuste fino" no valor dos parâmetros de $G(s)$ com o objetivo de melhorar a identificação.

3. Compare a dinâmica do Motor-Gerador com a de $G(s)$ identificada para

$$\delta u_1 = 0.9, \delta u_1 = 1.1, \delta u_1 = 2.$$

Analise os resultados.

4. Compare a dinâmica do Motor-Gerador com a de $G(s)$ identificada para $\delta u_1 = 1$, mas agora aplicando também $\delta u_2 = -1$ em $t = 20$.

Verifique se a dinâmica do Motor-Gerador é diferente da de $G(s)$ quando variamos a entrada em $+1V$ $(0 \rightarrow 1)$, e depois em $-1V(1 \rightarrow 0)$.

5. Repita os Itens $2, 3$ e 4 para o seguinte ponto de operação:

$$\boxed{\bar{u}_{30\%} = 6}$$

.

6. Repita os Itens $2, 3$ e 4 para o seguinte ponto de operação:

$$\boxed{\bar{u}_{80\%} = 16}$$

.

Resposta Senoidal

Considere a Função de Transferência $G(s)$ obtida para o ponto de operação $\bar{u}_{50\%} = 10$. Relembre que se a entrada $u(t)$ de uma Função de Transferência $G(s) = Y(s)/U(s)$ é

$$u(t) = Acos(\omega t), t \geq 0,$$

então a saída em regime permanente (steady-state) é

$$y_{ss}(t) = A|G(j\omega)|cos(\omega t + \angle G(j\omega)).$$

Relembre também o diagrama de Bode de um sistema de primeira ordem. Para isto, utilize os seguintes comandos:

=> G = ct.tf([K],[tau 1])

=> ct.bode(G)

1. Utilizando o arquivo *Motor_Gerador.py no VSCode program*, aplique uma variação sinusoidal $\delta u(t) = 0.5cos(\omega t)$ na entrada com:

$\omega = 0, \omega = 0.5/\tau, \omega = 1/\tau, \omega = 10/\tau, \omega = 12/\tau, \omega = 14/\tau, \omega = 20/\tau.$

Para cada frequência ω, compare a resposta em regime permanente de $G(s)$ com a do Motor-Gerador, fazendo uma medida aproximada da defasagem de fase.

2. Verifique também se a resposta de amplitude realmente cai $3dB \Leftrightarrow 0,707$ quando $\omega \sim= 1/\tau$ (frequência de corte). Os valores obtidos estão de acordo com o esperado? Justifique sua resposta.
3. Em termos da resposta em frequência do Motor-Gerador, o modelo de primeira ordem está adequado? Qual seria uma ordem mais adequada para modelarmos o motor em relação às altas frequências que foram aplicadas? Justifique sua resposta.

Conclusão: O modelo $G(s) = \Delta Y(s)/\ \Delta\ U(s)$ de primeira ordem é válido enquanto $\Delta u(t)$ for de baixa magnitude e de variação lenta.

Clique duas vezes (ou pressione "Enter") para editar

A maioria dos métodos determinísticos de identificação de sistemas fornece modelos contínuos, ainda que os dados sejam obtidos em instantes específicos no tempo, como é o caso de dados amostrados. O método determinístico caracteriza-se por não considerar que o sinal coletado pode estar contaminado com ruído (ruíde de medição). O que implica na hipótese de que a relação sinal-ruído deve ser suficientemente alta para se obter resultados adequados. Ao observar a figura, nota-se que a resposta é típica de sistemas de primeira ordem ou sistemas de ordem maior, no entanto somente com pólos reais. Fato já esperado, pois de acordo com a modelagem do tipo caixa branca. Para se determinar o comportamento desse sistema é necessário estimar dois parâmetros: o ganho, K_m , e a constante de tempo do motor, τ . O modelo considerado é dado pela função de transferência mostrada na equação (1).

Resposta ao Degrau

Revisão Teórica

Considere o sistema de 1º Ordem:

$$G(s) = \frac{\Delta Y(s)}{\Delta U(s)} = \frac{K_m}{\tau s + 1} \quad (1)$$

Em que $\Delta Y(s)$ é a variação do sinal de saída e $\Delta U(s)$ é a variação do sinal de entrada. As constantes K_m e τ representam o ganho e a constante de tempo do sistema, respectivamente.

A resposta do sistema para uma variação do tipo degrau com amplitude A no domínio do tempo, é dada pela expressão:

$$\Delta y(t) = AK_m(1 - e^{-\frac{t}{\tau}})u(t) \quad (2)$$

Observe que a resposta do sistema para o instante de tempo correspondente a uma constante de tempo é, $t = \tau$

$$\Delta y(\tau) = AK_m(1 - e^{-\frac{\tau}{\tau}}) = 0,6321AK_m \quad (3)$$

O que implica na seguinte interpretação, para uma variação do tipo degrau de amplitude A na entrada do sistema de primeira ordem, a amplitude correspondente na saída decorridos uma constante de tempo após a variação na entrada é $63,21\%$ do ganho do sistema K_m vezes a amplitude da entrada A.

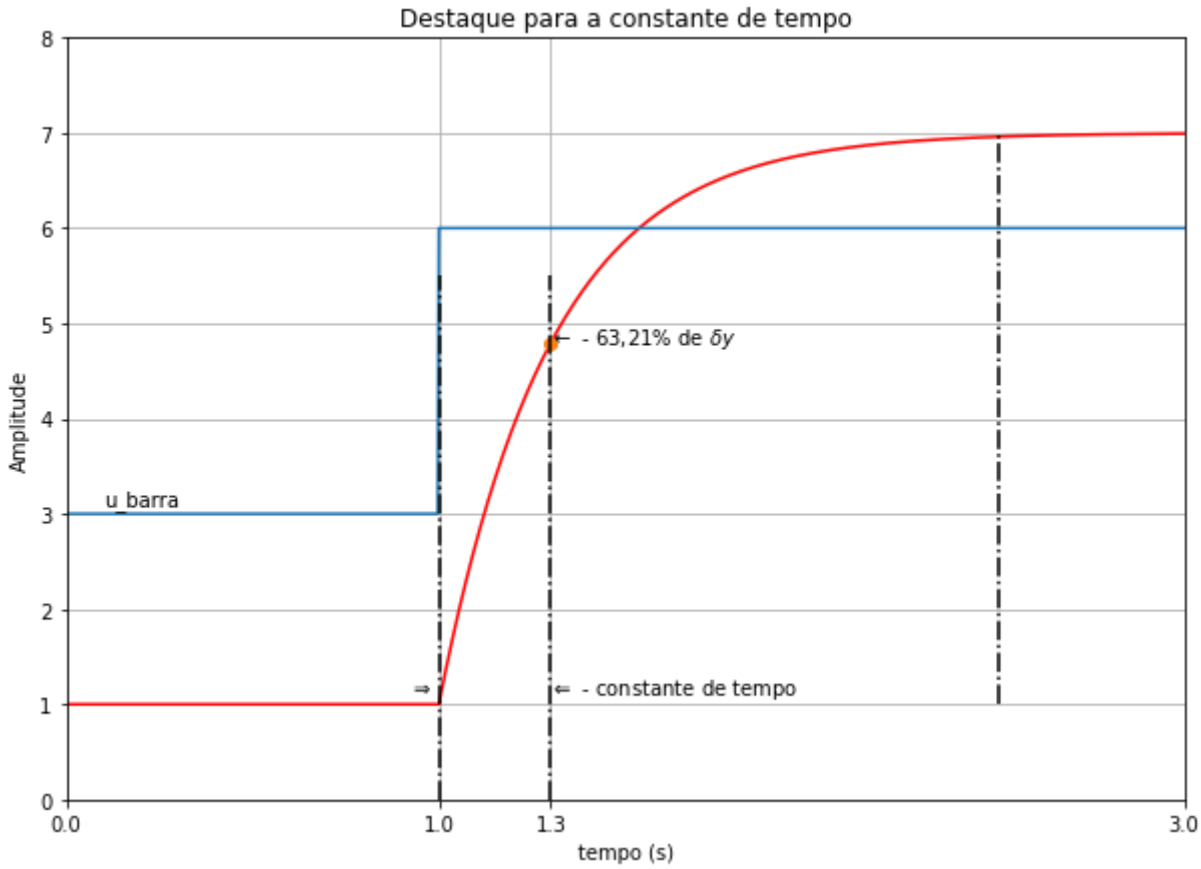
Código 1

O código gera o gráfico no tempo da resposta do sistema .

```
# _________________ CARREGA BIBLIOTECAS __________#

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

```
# _________________ DEFINE AS CONSTANTES __________#
A = 3       # Define a amplitude do sinal de entrada
Km = 2      # Define o ganho do sistema de 1 ordem
tau = 0.3   # Define a constante de tempo do sistema de 1 ordem
t0 = 1      # Instante em que o degrau é aplicado
setpoint = 1 # off set ou ponto de operação

degrau = lambda t: 1*(t>=0)
delta_y = lambda t: A*Km*(1-np.exp(-t/tau))*(t>=0)
t = np.linspace(0,3,1000)


plt.figure(figsize=(10,7))
plt.plot(t, setpoint + delta_y(t - t0), c = "r")
plt.plot(t, 3 + A*degrau(t-t0))

plt.plot(tau + t0, setpoint + delta_y(tau),'o', label = 'constante de tempo')

plt.plot((tau + t0)*np.ones(100), np.linspace(0,5.5,100),'-.', c="k")
plt.plot((t0)*np.ones(100), np.linspace(0,5.5,100),'-.', c="k")
plt.plot((t0+5*tau)*np.ones(100), np.linspace(1,7,100),'-.', c="k")

plt.title("Destaque para a constante de tempo")
plt.ylabel('Amplitude')
plt.xlabel('tempo (s)')
plt.axis([0,10*tau,0, A*Km + 2])
plt.xticks([0, t0, t0 + tau, t[-1]])
plt.text(t0+tau,setpoint+0.1, "$\Leftarrow$ - constante de tempo ")
```

```
plt.text(t0-0.1, 1.1, " $\Rightarrow$ ")
plt.text(t0+tau,setpoint + delta_y(tau), "$\leftarrow$ - 63,21% de $\delta y$ ")

plt.text(0.1, 3.1, "u_barra")

plt.grid()
plt.show()
```



Ganho do Sistema

Observa-se que, em regime permanente, a resposta $\Delta y(\infty) \rightarrow AK_m$. Assim, o ganho do sistema pode ser estimado a partir de uma resposta ao degrau.

$$K_m = \frac{\Delta y(t)}{\Delta u(t)} = \frac{y(\infty) - y(0^-)}{A} \tag{4}$$

em que $\Delta y(t)$ corresponde a variação obervada no sinal de saída, com $y(\infty)$ sendo a resposta em regime permanente e $y(0^-)$ o valor da resposta antes da excitação. O $\Delta u(t)$ é a variação de amplitude do sinal de entrada.

Constante de tempo

A resposta do sistema para uma constante de tempo assume o valor de $\Delta y(\tau) = AK_m 0,6321$. Logo, de posse da resposta ao degrau é possível estimar o valor da constante de tempo.

Basta seguir os seguintes passos:

1) Identifique na resposta ao degrau o valor de regime permanente;

2) Calcule $63,21\%$ do valor de regime permanente;

3) Some ao valor da resposta antes do grau ter sido aplicado o valor calculado no item 2)

Exemplo No gráfico gerado pelo Código 1 tem-se:

$A = 1$

$\Delta y(\infty) = 3$

```
!pip install control
```

```
# _________________ CARREGA BIBLIOTECAS __________#

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import control as ct
import scipy.signal as sg
from control.matlab import *
```

```
# _________________ MONTAGEM DO GOOGLE DRIVE __________#
# _________________ PERMITE ACESSA DADOS ARNAZENADOS __________#

from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Ponto de Operação

```
# dados_rampa = pd.read_csv("/content/drive/MyDrive/Monitoria-2020-ERE-Sistemas_
# dados_rampa = np.asarray(dados_rampa)[:,0]

# plt.figure()
# plt.plot(dados_rampa)
# plt.show()
```

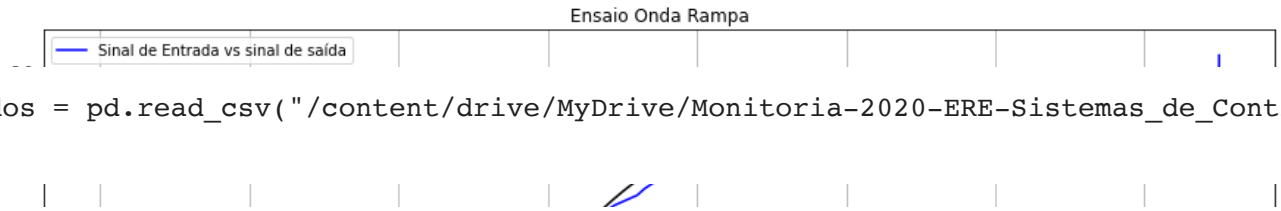
```
# dados_rampa = pd.read_csv("/content/drive/MyDrive/Monitoria-2020-ERE-Sistemas_
# tempo = dados_rampa[0,:]
# sinal_entrada  = dados_rampa[1,:]
# sinal_saida = dados_rampa[2,:]
# toc = dados_rampa[3,:]

# janela = (tempo<3.5)

# plt.figure(figsize=(15,7))
# plt.plot(sinal_entrada[janela], sinal_saida[janela], c = 'b', label = "Sinal d

```

```
# plt. title('Ensaio Onda Rampa')
# plt.ylabel('Tensão (V)')
# plt.xlabel('Tensão (V)')
# plt.legend()
# plt.grid()
# plt.show()

# plt.figure(figsize=(15,7))
# plt.plot(tempo[janela],sinal_entrada[janela], c = 'b', label = "Sinal de Entra
# plt.plot(tempo[janela],sinal_saida[janela], c = 'r', label = "Sinal de Saída")

# plt. title('Ensaio Onda Rampa')
# plt.ylabel('Tensão (V)')
# plt.xlabel('Tensão (V)')
# plt.legend()
# plt.grid()
# plt.show()
```

```
nolinearTransform  = lambda A, a, t: A*(1 - np.exp(-a*t))*(t>=0)

tensao_entrada = np.linspace(0,15, 750)
tensao_saida = nolinearTransform(3, 0.5,tensao_entrada-4.2)
```

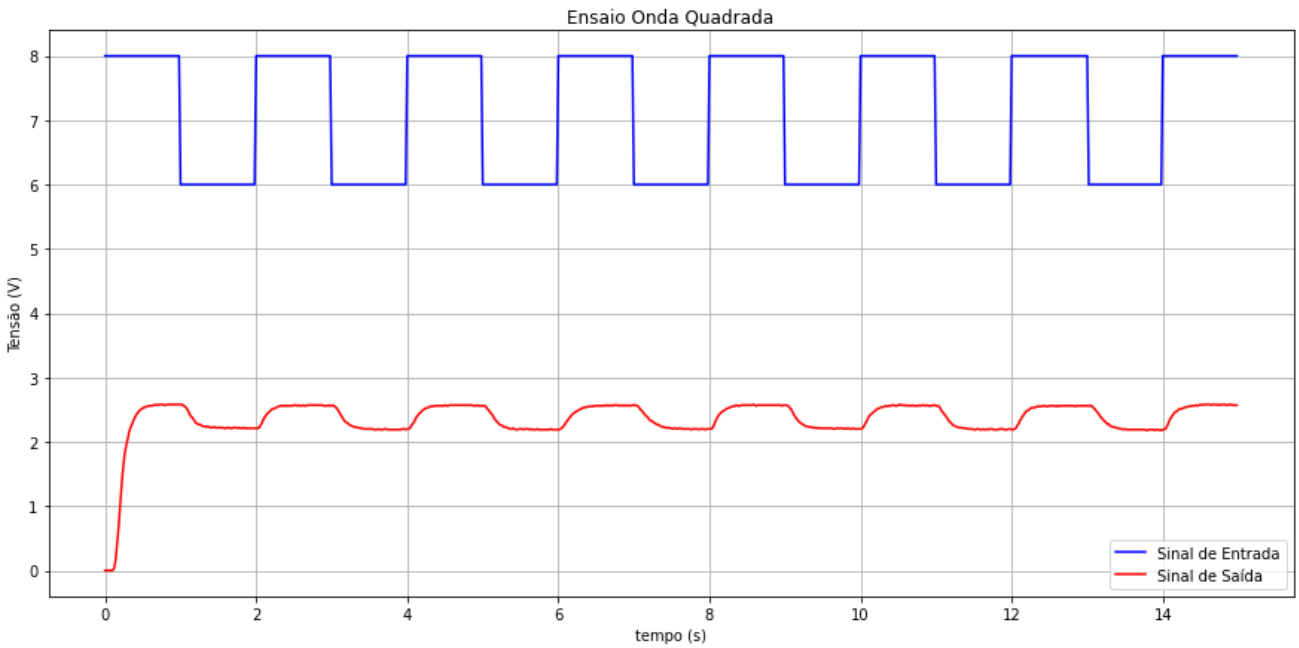
```
dados_rampa = pd.read_csv("/content/drive/MyDrive/Monitoria-2020-ERE-Sistemas_de
tempo = dados_rampa[0,:]
sinal_entrada  = dados_rampa[1,:]
sinal_saida = dados_rampa[2,:]
toc = dados_rampa[3,:]

janela = (tempo<3.5)

plt.figure(figsize=(15,7))
plt.plot(sinal_entrada[janela], sinal_saida[janela], c = 'b', label = "Sinal de

plt.plot(tensao_entrada,tensao_saida,'k')

plt. title('Ensaio Onda Rampa')
plt.ylabel('Tensão (V)')
plt.xlabel('Tensão (V)')
plt.legend()
plt.grid()
plt.show()

plt.figure(figsize=(15,7))
plt.plot(tempo[janela],sinal_entrada[janela], c = 'b', label = "Sinal de Entrada
plt.plot(tempo[janela],sinal_saida[janela], c = 'r', label = "Sinal de Saída")

plt. title('Ensaio Onda Rampa')
plt.ylabel('Tensão (V)')
plt.xlabel('Tensão (V)')
plt.legend()
plt.grid()
plt.show()
```

```
31
32 # plt.figure()
33 # plt.plot(tempo,toc)
34 # plt.show()
35
36 Ts = np.mean(toc)
37 print('\n','Periodo de Amostragem:', Ts, janela.shape )
```




```
dados = pd.read_csv("/content/drive/MyDrive/Monitoria-2020-ERE-Sistemas_de_Contr

```

```
tempo = dados[0,:]
sinal_entrada  = dados[1,:]
sinal_saida = dados[2,:]
toc = dados[3,:]

plt.figure(figsize=(15,7))
plt.plot(tempo,sinal_entrada, c = 'b', label = "Sinal de Entrada")
plt.plot(tempo,sinal_saida, c = 'r', label = "Sinal de Saída")

plt. title('Ensaio Onda Quadrada')
plt.ylabel('Tensão (V)')
plt.xlabel('tempo (s)')
plt.legend()
plt.grid()
plt.show()

# plt.figure()
# plt.plot(tempo,toc)
# plt.show()

Ts = np.mean(toc)
print('\n','Periodo de Amostragem:', Ts)
```




```
Periodo de Amostragem: 0.027902403831481886
```

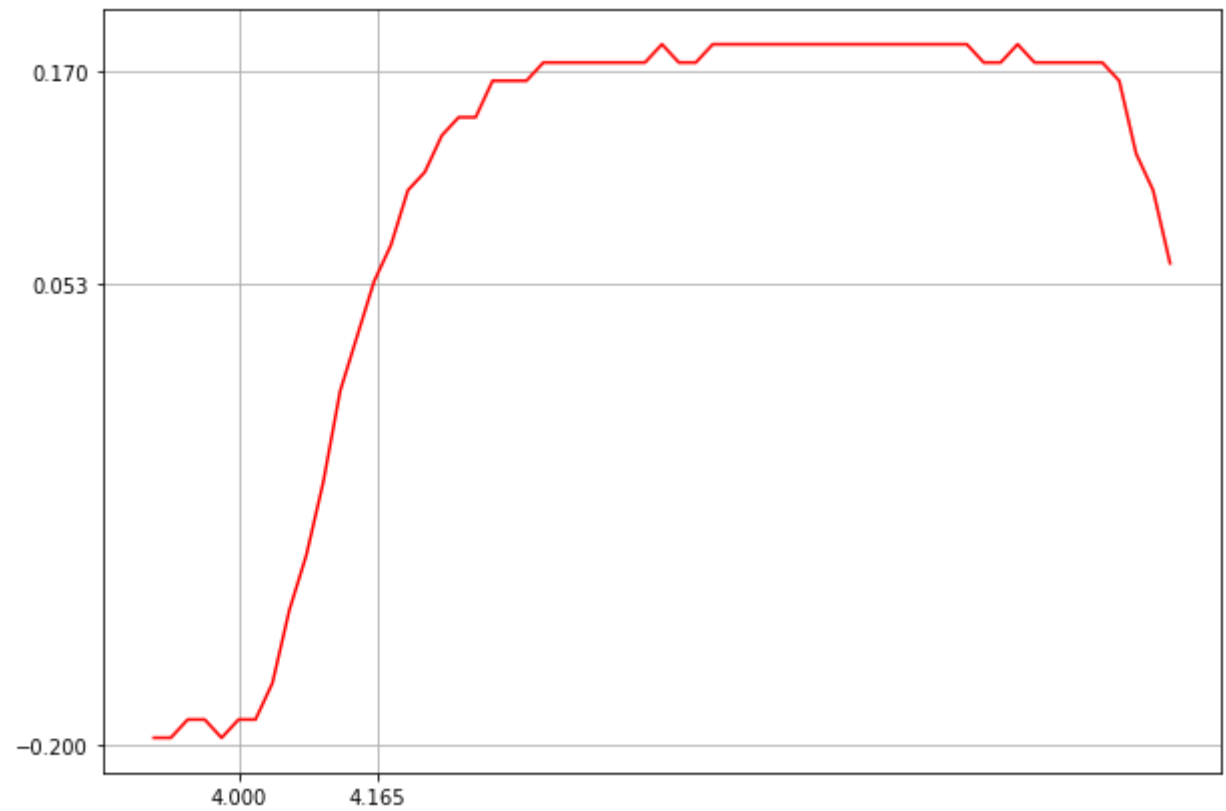
Exercício 01

Responda o que se pede:

1. Para o sinal de entrada, defina a excursão, o nível DC e a frequência.
2. Para o sinal de saída, defina a excursão, o nível DC e a frequência.

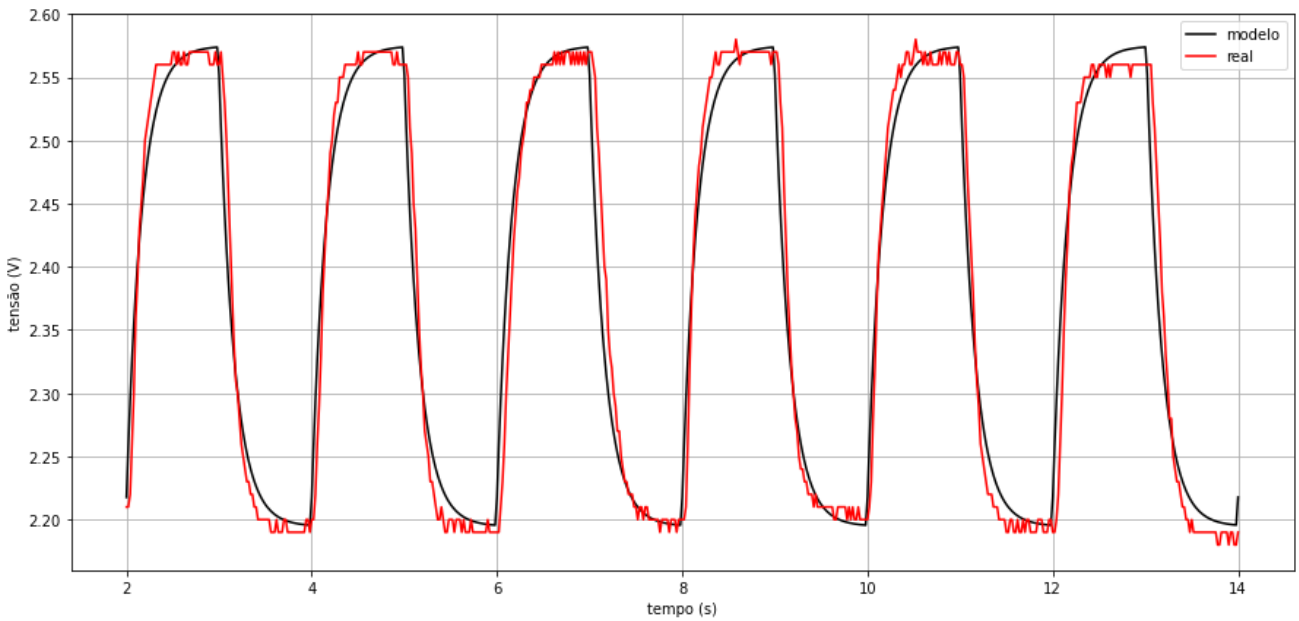
```
# _________________ Define janela que despreza os primeiros instantes do ensaio _

janela = (tempo>2) & (tempo<14)



nivel_dc_entrada = np.mean(sinal_entrada[janela])
# nivel_dc_entrada = 7.

nivel_dc_saida = np.mean(sinal_saida[janela])

# _________________ Remove Nivel DC da Entrada e da Saída __________#

r = sinal_entrada - nivel_dc_entrada
y = sinal_saida - nivel_dc_saida

plt.figure(figsize=(15,7))
plt.plot(tempo[janela],r[janela], c = 'b', label = "Sinal de Entrada")
plt.plot(tempo[janela],y[janela], c = 'r', label = "Sinal de Saída")

plt. title('Ensaio Onda Quadrada')
plt.ylabel('Tensão (V)')
plt.xlabel('tempo (s)')
plt.legend()
plt.grid()
plt.show()

print("Nivel DC entrada:" , nivel_dc_entrada )
print("Nivel DC saída:" , nivel_dc_saida )
```


```
# _________________ Define o intervalo de tempo entre os instantes 3.9 e 5.1 segu

index = (tempo>3.9)&(tempo<5.1)

plt.figure(figsize=(10,7))
plt.plot(tempo[index],y[index],'r')
# plt.plot(tempo[index],r[index],c = 'b')
# plt.yticks([-.5,-.4,0.3097,0.4,0.45,0.5])
plt.yticks([-0.20,0.053,0.17])
plt.xticks([4,4.165])
plt.grid()
plt.show()
```



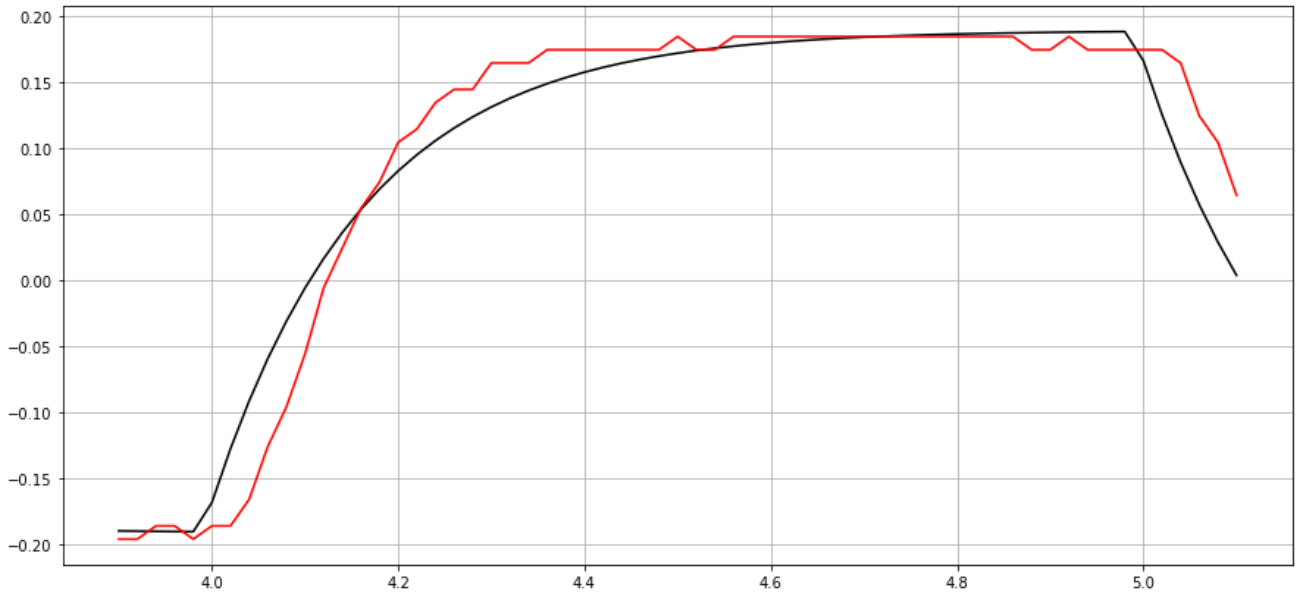

```
# _________________ Simulação do Modelo Determinado __________#

num = [0.19]
den = [0.165, 1]

G = ct.tf(num,den)
```

```
_,ym = ct.forced_response(G,T = tempo, U = r)

plt.figure(figsize=(15,7))
plt.plot(tempo[janela], ym[janela] + nivel_dc_saida ,'k', label = 'modelo')
plt.plot(tempo[janela], sinal_saida[janela] ,'r', label = 'real')
# plt.axis([0,14,-.5,0.5])
plt.xlabel('tempo (s)')
plt.ylabel('tensão (V)')
plt.legend()
plt.grid()
plt.show()

erro_rms = (np.sum((ym[janela] - sinal_saida[janela])**2)/np.size(sinal_saida[ja
print("Erro RMS:", np.sqrt(erro_rms))
```




```
Erro RMS: 2.3863939508311494
```

```
# _____ Define o intervalo de tempo entre os instantes 3.9 e 5.1 segu

index = (tempo>3.9)&(tempo<5.1)

plt.figure(figsize=(15,7))
plt.plot(tempo[index],ym[index],'k',tempo[index],y[index],'r')
plt.grid()
plt.show()
```




```
Kc = 5
a = 10

num = np.array([0, Kc*a])
# den = np.array([1,a])
den = np.poly([-1,-2])

Ts = 0.01

Gc = ct.tf(num,den)
Gz = ct.c2d(Gc,Ts,'zoh')


# Gz.zero()

# G = ct.tf([0.2],[0.16,1])
# Gp = ct.c2d(G,Ts,'zoh')


print("Modelo Contínuo:", Gc,"\n","Modelo Discreto:", Gz)



```

```
Modelo Contínuo:
     50
-------------
s^2 + 3 s + 2

 Modelo Discreto:
0.002475 z + 0.002451
---------------------
z^2 - 1.97 z + 0.9704

dt = 0.01
```