Laboratórios de Matemática 2

Trabalho Curricular

Ano de 2016/2017

Licenciatura em Engenharia Electrotécnica e de Computadores
ISEP

1160775: João Neves
1160555: Erendiro Pedro
1161053: Tiago Ribeiro
1161312: João Costa
Versão: V1
Data: 31/05/2017 Turma: 1DH Nome do docente: João Matos

Conteúdo

1. Introdução geral

Este trabalho prático está dividido em três partes, sendo que, cada uma delas aborda um tema diferente.

A primeira parte consiste no estudo de um circuito elétrico RC, no qual iremos descobrir várias componentes do circuito em causa, como por exemplo, a constante de tempo $\boldsymbol{\tau}$ (tau), correntes, tensões, energias e mais algumas. A chegada a estes resultados era impossível sem utilizarmos os métodos da amostragem, derivação e integração numérica.

A segunda parte consiste no estudo de, mais uma vez, um circuito RC. No entanto, desta vez, iremos estudar o comportamento do circuito relativamente tensões máximas e mínimas, energias médias (ou potencias medias), e representação destes mesmos comportamentos. Para tal, iremos usar o método das Series de Fourier e caracterização do Sinal.

A terceira parte, de certa forma, diferencia-se das restantes, pois desta vez trata-se de um circuito RLC, sendo que o objetivo desta parte é descobrir a corrente e tensão aos terminais dos componentes do circuito, dependendo do comportamento deste. Para alcançarmos estes resultados iremos aplicar equações diferenciais à análise deste circuito, e o uso do método de Euler.

2. Resultados

Parte 1

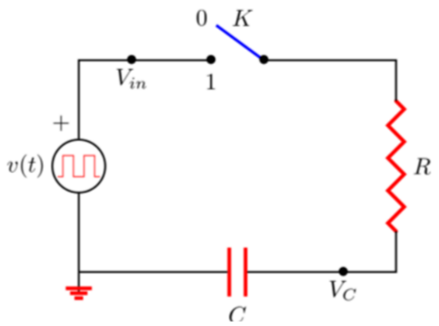


Fig. 1 Circuito RC

Todos os resultados obtidos estão na parte B dos anexos.
R=4,7 +/- 5% Ω
Em t=0s, K está na posição 1
O erro associado à leitura dos tempos é de 1ms
O erro associado à leitura do potencial é de 5mV.
Os valores de V_{in} e V_c estão guardados no ficheiro fornecido para a realização do trabalho.
Nesta parte do trabalho pretende-se calcular a tensão do condensador (V_c) e outras componentes do circuito como a constante de tempo **τ** (tau), a corrente do circuito i(t), a energia dissipada na resistência e os possíveis erros associados aos cálculos destes componentes.

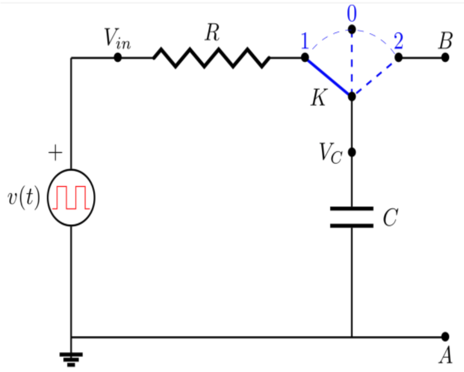
1./2. Na pergunta 1 iremos calcular o número de amostras, o intervalo de tempo entre amostras, a frequência de amostragem e o tempo no qual ocorreu a aquisição dos sinais V_{in} (tensão de entrada) e V_c (tensão aos terminais do condensador). A pergunta 2 trata-se da representação do gráfico desses sinais em relação ao tempo. Para tal, usamos funções que o python já contém como a função ("len(string)") que nos diz a quantidade de campos da string, assim obtivemos a quantidade de amostras. O intervalo de tempo entre amostras obtém-se através da divisão entre o tempo total de obtenção (100 segundos) e a quantidade de amostras (1301), ou seja, ($T = \frac{100}{Ndados}$). A frequência de amostragem obtém-se através da expressão que envolve o período ($f = \frac{1}{T}$). Finalmente, o tempo durante o qual ocorreu a aquisição dos sinais é dado através de um Δt, isto é, ($Taquisição = Tfinal - Tinicial$). Por outro lado, na pergunta 2, bastou-nos usarmos funções fornecidas pelo python para representarmos o gráfico de V_c e V_{in} em relação ao tempo, sendo que, neste caso a função utilizada foi ("*plot(t,Vin,t,Vc)*").
3. O objetivo da questão 3 era calcular os tempos em que ocorrem as descargas do Condensador e determinar a derivada progressiva (ou a direita) do sinal V_{in}(t). Para tal criamos um ciclo onde iriamos guardas as varias derivadas numa lista, dentro dessa lista fomos procurar as derivadas negativas com outro ciclo. Após isso só tivemos de apresentar os intervalos de tempo.
4. A pergunta 4 divide-se em quatro alíneas, mas todas referentes à primeira descarga completa do condensador.
A) A alínea **A** pede-nos para calcularmos a constante de tempo **τ** (tau), sendo que o tau corresponde ao tempo decorrido entre o inicio da descarga até ao instante em que a tensão do condensador desceu para 36,8%. Para descobrirmos essa grandeza, primeiro descobrimos o valor da tensão no instante tau: ($Vc_{\tau} = Vc_{descarga} \times 0{,}369$). Depois, através de um ciclo descobrimos o índice para calcular tau, e finalmente tínhamos tudo para calcular a constante de tempo.
B) A alínea **B** tem como objetivo calcular a capacidade do condensador e o valor do majorante do erro relativo associado a esse calculo, sabendo que: $\tau = R \times C$. Como já calculamos tau na alínea anterior e o valor da resistência é nos dado no enunciado, podemos alterar a formula para descobrirmos C (a capacidade do condensador), ou seja, $C = \frac{\tau}{R}$. Para descobrirmos o erro

associado a este calculo, temos de fazer as derivadas parciais em ordem ao tau e em ordem à resistência, e finalmente somar o produto dessas derivadas pelos respetivos erros associados.

C) Na alínea **C** temos de descobrir a corrente do circuito (i(t)), sabendo que: $i(t) = \frac{dq}{dt}$, $q(t) = C \times V_c(t)$. Temos também de representar os sinais q(t) e i(t) graficamente. Para a resolução deste problema, começamos por definir as funções i(t) e q(t), conforme as funções que foram fornecidas, isto é, através de ciclos obtivemos os valores das funções em cada instante e assim podemos formar uma lista com os pontos das funções, sendo que desta forma conseguimos obter os gráficos em relação ao tempo.

D) A ultima alínea a resolver neste trabalho é a alínea **D** onde é requisitado o cálculo da energia dissipada na resistência R e o majorante do erro absoluto desse cálculo sabendo que a formula da energia dissipada é: $\int R \times i^2 \times dt$ (Com a nota adicionada para obter estes valores através do método das somas de Riemann). Para a realização desta alínea temos de perceber bem o método de Riemann tal como o significado do cálculo de um integral, isto é, o calculo de um integral é o calculo da área sob uma curva no plano cartesiano. E, por outro lado, o método de Riemann trata-se de uma aproximação da área de funções ou linhas em um gráfico, mas também o comprimento das curvas e outras aproximações, isto é, o método de Riemann trata-se da soma de varias áreas calculadas que estão sob uma função. Assim se obtivermos o valor de $R \times i^2$em cada instante, obtemos a altura dos retângulos para os quais queremos calcular a área. Por outro lado, sabemos que essa função esta em ordem ao tempo, isto é, o comprimento dos retângulos será o intervalo de tempo entre cada amostra, podemos assim concluir que a área dos retângulos em cada instante é: $A(t) = (R \times i^2(t)) \times T$. Assim, o valor do integral será aproximado à soma de todas as áreas calculadas.

Parte 2



Todos os resultados obtidos estão na parte B dos anexos.
R=4700Ω +/- 5%
C=330µF +/- 20%
No instante t=0s, o comutador K passa da posição 0 para a posição 1.
Os valores de V_C e V_{IN} estão guardados no ficheiro fornecido no enunciado da parte.
O erro associado à leitura do tempo é 1ms.
O erro associado à leitura do potencial é 5mv.

$$RMSE_{\%} = 100 \times \frac{\sqrt{P-P_N}}{\sqrt{P}};\ P_N = \left(\frac{a0}{2}\right)^2 + \frac{1}{2} \times \sum_{n=1}^{N} (a_n^2 + b_n^2)$$

Na parte 2 do trabalho iremos estudar o comportamento do circuito RC, a carga e a descarga do condensador, utilizando métodos analíticos e também tendo em conta as Series de Fourier.

1. A pergunta 1 divide-se em 5 alíneas todas em volta dos sinais V_{IN} e V_C.

a) A alínea **A** tem como objetivo definir as expressões analíticas de V_{IN} e V_C tendo em conta 3 fatores:

- Para $V_{IN}(t)$, o valor máximo e mínimo, $\mathbf{V_M}$ e $\mathbf{V_0}$ respetivamente, da tensão são os inteiros mais próximos dos valores máximo e mínimo lidos pelo sensor;
- Para $V_C(t)$, os instantes de inicio de carga e descarga, t_C e t_D respetivamente, são os inteiros mais próximos dos instantes registados, sendo a lei da carga e da descarga definidas por:
  i. $V_M \times (1 - e^{-(\frac{t-t_C}{\tau})})$;
  ii. $V_D \times e^{-(\frac{t-t_D}{\tau})}$, onde $V_D = V_C(t_D)$ e $\tau = R \times C$;
- O período de definição começa com o inicio da 1ª descarga completa.

Para a resolução deste exercício começamos por ter em conta o que fizemos no trabalho passado, pois na primeira parte trabalhamos com dados numéricos de um circuito RC, e nesta parte vamos usar estas equações para modelar o que observamos na parte numérica. Sendo assim, neste exercício usamos valores do trabalho passado para descobrir a tensão no momento de carga, descarga e o tau, assim com estes componentes conseguimos modelar as novas funções de V_C e V_{IN} periodicamente (tendo em conta os 3 fatores apresentados) e representá-las graficamente.

b) A alínea **B** pede para determinarmos a potência média de V_{IN} e V_C sabendo que, para um período ***T*** esta se define da seguinte forma: $P = \frac{1}{2} \times \int f^2(t) \times dt$. Sendo assim, na resolução desta alínea tivemos apenas de aplicar a forma dada diretamente no python e conseguimos obter os resultados desejados.

c) Na alínea **C** é pedido para determinarmos as Series de Fourier na forma trigonométrica dos sinais V_{IN} e V_C. Para tal, é necessário saber representar as Series de Fourier, que na forma trigonométrica

$$a_0 = \frac{1}{L}\int_c^{c+2L} f(t)\,dt,\ a_n = \frac{1}{L}\int_c^{c+2L} f(t)\cos\left(\frac{n\pi t}{L}\right)dt,\ n \geq 1 \text{ e } b_n = \frac{1}{L}\int_c^{c+2L} f(t)\ \text{sen}\left(\frac{n\pi t}{L}\right)dt,\ n \geq 1$$

têm o seguinte aspeto:

Sendo assim, só tivemos de aplicar os valores que obtivemos em alíneas anteriores a esta formula para obtermos as Series de Fourier na forma trigonométrica dos sinais V_{IN} e V_C.

d) A alínea **D** pede-nos para representarmos os sinais V_{IN} e V_C em conjunto com a correspondente aproximação dada pela soma até ao 10º harmónico. Então, começamos por definir a função das aproximações tanto de V_{IN} como de V_C, utilizando a expressão que deduzimos para o calculo dos harmónicos na alínea anterior, após isso só tivemos de representar os gráficos utilizando essas novas expressões que obtivemos.

e) O problema presente nesta alínea é a representação dos espetros de V_{IN} e V_C, tendo a certeza que as reconstruções dos sinais a partir das correntes somas parciais da Serie de Fourier têm um erro RMS inferior a 10% para V_{IN} e um erro inferior a 5% para V_C. Então através das formulas da serie de

$$A_0 = \frac{a_0}{2},\ A_n = \sqrt{a_n^2 + b_n^2} \text{ para } n \geq 1, \text{ e } w_n = \frac{2n\pi}{T}$$

Fourier conseguimos descobrir a amplitude de V_{IN} e V_C, as suas respetivas fases, sendo que a amplitude dos sinais é obtida através do seguinte:

Após termos obtido a amplitude e a fase de ambos os sinais, calculamos o erro e a potencia media, assim só nos restava apresentar os espetros.

2. A pergunta 2 pede-nos para repetirmos as alíneas **c)**, **d)** e **e)** para o sinal V_{IN} tendo em conta a leitura de um bloco de amostras correspondente ao intervalo de leituras em que ocorreu a 1ª descarga completa seguida de uma carga completa do condensador. Sendo que, devemos garantir que nessa leitura esse intervalo de tempo tem duração igual à do período dos sinais V_{IN} e V_C. Então, a primeira coisa a fazer foi descobrir o valor de tempo inicial e final e assim obtendo o Δt, depois disso usamos 3 ciclos para calcularem o integral de V_C e assim obtendo valores necessários para a definição de a_0 ,a_n e b_n, e para a definição da serie de harmónicos, depois repetimos o mesmo processo das alíneas que eram pedidas mas desta vez com os novos valores obtidos.
3. Na pergunta 3, decidimos responder à alínea **a)**, esta alínea pede para determinar os valores de V_{IN} e de V_C utilizando o conceito de taxa de distorção harmónica, **THD**, utilizando no máximo 10 harmónicos. Após pesquisarmos o que era essa tal taxa, descobrimos que ela é calculada da seguinte forma:

$$THD = \sqrt{\frac{a_2^{\,2} + a_3^{\,2} + a_4^{\,2} + \ldots + a_N^{\,2}}{a_1^{\,2}}}$$

Então, usamos um ciclo para descobrir a soma do numerador da raiz, que são os harmónicos de 1 ate 10 e depois aplicamos o resto da formula, obtendo assim, a taxa de distorção harmónica.

Parte 3

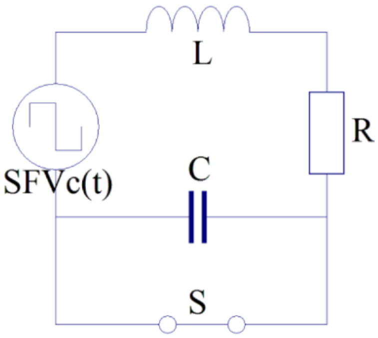


Fig. 1 Circuito Elétrico

Todos os resultados obtidos estão na parte B dos anexos.
R=1,2Ω
L=2H
C=1F

$$V_R(t) = R \times i(t)$$
$$V_L(t) = L \times (\frac{di}{dt})$$
$$V_C(t) = \frac{1}{C} \times \int_0^t i(t) \times dt$$

O sinal de entrada SFV_C corresponde à aproximação de Fourier do sinal $V_C(t)$, que foi obtido na alínea 2 da segunda parte do trabalho.

Nesta parte do trabalho é pedido para recorrer a modelização do circuito a partir de equações diferenciais e usando o método de Euler para determinar as soluções, representar graficamente o comportamento da corrente de todas as tensões nos terminais dos componentes do circuito antes e apos a abertura do comutador S, apenas no período de definição T.

1. O exercício **1** trata-se da importação de dados que estão em ficheiros para o script.
2. O exercício **2** pede-nos a para identificarmos os instantes de inicio fim e do período de definição T, e também pede para identificarmos o instante em que o comutador troca de posição. Na resolução deste exercício começamos por definir o instante de tempo inicial e o final, sendo estes o primeiro e o ultimo elemento do array do ficheiro, depois através de um ciclo que iria percorrer o ficheiro fornecido encontramos o instante em que o comutador muda de posição.
3. O exercício **3** tem como objetivo apresentar as equações diferenciais que modelizam o circuito antes e depois da comutação. Na resolução deste exercício começamos por definir **t** como variável independente, **i** como variável dependente, e definimos a função da tensão (que é uma equação diferencial). Como antes da comutação o condensador não tem efeito no circuito então $SFV_C(t)$ só irá ter as componentes da resistência e da bobine, por outro lado, depois da comutação o condensador já tem efeito no circuito, assim só tivemos de ajustar a função a estas condições e conseguimos obter os resultados.
4. Na pergunta **4** é necessário apresentar e atribuir as condições iniciais a cada uma das equações diferenciais do exercício anterior. Sendo assim, antes da comutação a corrente no instante de inicio do período de definição é igual a **0**. Por outro lado, depois da comutação a corrente no instante em que o comutador troca de posição irá ter o ultimo valor que surgiu antes de ele trocar de posição e a carga do condensador nesse instante é **0**.
5. Na questão 5 temos de obter a resolução numérica das equações diferenciais usando o método de Euler, sendo que este deve estar devidamente adaptado as equações diferenciais obtidas. Assim, antes da comutação estabelecemos as condições iniciais e calculamos a diferença entre duas amostras consecutivas, e assim aplicamos o método de Euler onde iremos somar a área de cada instante. Para o outro instante, tentamos evitar o calculo de um integral numericamente então definimos $\frac{dq}{dt} = i$, mais uma vez, definimos as equações iniciais e fizemos um ciclo para obter um sistema de equações diferenciais, a partir do qual podemos retirar duas listas, uma com os valores de carga ao longo do tempo e outra com os valores da corrente ao longo do tempo. Com todos estes valores podemos apresentar a representação gráfica da corrente antes do instante de comutação e a representação gráfica da corrente e da carga após o instante de comutação.
6. No exercício **6** temos de construir e concatenar os vetores de corrente e tensão nos componentes antes e depois da comutação, no período de definição T do sinal. Começamos por concatenar os vetores de corrente, usando os valores do exercício anterior, depois definimos vetores para guardarem os valores com as tensões dos componentes e da corrente. Depois, fizemos um ciclo para encontrar a tensão na resistência durante esse instante sabendo que $V_R = R \times i(t)$, sabemos que inicialmente a tensão do condensador é nula e sabemos também que $V_C = q \times C$, então

fizemos um ciclo que iria multiplicar esses valores em cada instante. Por outro lado, a tensão da bobine será obtida pela Lei de Kirchoff para tensão em malhas fechadas, onde a soma das quedas de tensão nos componentes é igual à tensão fornecida pela fonte, ou seja, $SFV_c = V_R + V_C + V_L$, da qual obtemos que a tensão na bobine é: $V_L(t) = SFV_c(t) - V_R(t) - V_C(t)$. Então só nos resta fazermos um ciclo para obter o valor da bobine em cada instante e guarda-los no vetor.

7. A pergunta **7** pede a representação gráfica da corrente e faz tensões no período de definição T do sinal. Como já obtivemos as funções no exercício anterior, basta aplicar as funções do python para obtermos os gráficos.

3. Conclusão

1. Na parte 1, podemos considerar os resultados mais importantes, pois foram os mais difíceis de obter, os instantes de descarga do condensador, os valores dos gráficos de q(t) e i(t) e os valores da soma de Riemann, isto é, o minorante, majorante, valor médio e área.

   Na parte 2, os resultados a marcar são a apresentação dos sinais V_{IN} e V_C na forma trignometrica das series de Fourier a apresentação dos espetros e os valores obtidos para o exercício 2, pois tivemos de, basicamente, repetir o mesmo processo para os valores anteriores, mas com condições novas.

   Na parte 3, os resultados mais importantes são os valores obtidos pelo método de Euler adaptado as condições propostas e a concatenação dos vetores de corrente e tensão antes e depois da comutação.

2. Ao longo do desenvolvimento deste trabalho enfrentamos vários problemas, sendo que estes nunca foram de enorme escala, mas mesmo assim, por momentos impediram-nos o progresso deste mesmo. Exemplos destes problemas são erros de código e aplicação errada de erros matemáticos.

3. Reflexão sobre os pontos marcantes do trabalho:
Podemos dizer, como grupo, que com este trabalho conseguimos aprender mais sobre as várias formas de analisar circuitos, bem como os vários comportamentos destes, e fundamentalmente, aprendemos a dominar os métodos matemáticos ensinados nesta unidade curricular, como por exemplo, o método de Euler, as somas de Riemann, as series de Fourier nas suas varias formas, e também a dominar mais o pythonxy. Esta foi uma oportunidade ótima para progredirmos no nosso desenvolvimento como engenheiros tendo por muitas vezes que procurar soluções ortodoxas. Podemos dizer que se possível começar de novo não mudaríamos praticamente nada, pois o desenvolvimento deste trabalho foi quase perfeito como equipa.

4. Referências/Bibliografia

Wikipédia- https://pt.wikipedia.org/wiki/S%C3%A9rie_de_Fourier
https://pt.wikipedia.org/wiki/Soma_de_Riemann
https://pt.wikipedia.org/wiki/Integral

THD- http://www.inf.ufrgs.br/~juan/tutorial/My_tutorial/tutorial3/THD/THD.htm

A. Anexos

Scripts desenvolvidos

Parte 1

Parte 2

Parte 3

B. Anexos

Resultados globais obtidos

Parte 1

```
# -*- coding: utf-8 -*-
"""
Laboratórios de Matemática 2
Parte 1 - Versão 1

Data: 2017/04/08

Autores
1. <1160555> <Erendiro Sangueve Njunjuvili Pedro>
2. <1160775> <João Pedro Freitas Neves>
3. <1161053> <Tiago Miguel Pereira Ribeiro>
4. <1161312> <João Pedro Gomes da Costa>
"""

from __future__ import division
from __future__ import print_function
from LabMat1_Lib import *
import pylab as pl


clc()

#%% Bloco de identificação do trabalho
print('\n\nTrabalho curricular\nPARTE 1\nVERSÃO 1\n\n')
print('AUTORES:')
print('\t<1160555>\t<Erendiro Pedro>')
print('\t<1161053>\t<Tiago Ribeiro>')
print('\t<1160775>\t<João Neves>')
print('\t<1161312>\t<João Costa>')

#%% Leitura do ficheiro de dados
# Estrutura de dados: dados=array(Nlinhas,3colunas)
#  t/s |  Vin/V  | Vc/V
#
Ficheiro="Dados_TC_P1_V1.txt"
print('\n\nLeitura do ficheiro de dados:',Ficheiro,'...\n\n')
dados=loadtxt(Ficheiro)
print('      t/s       Vin/V      Vc/V ')
print(35*'-')
for i in range(0,5):
    print((3*"%10.3f") % tuple(dados[i,:]))
print('     .....      .....      .....')
for i in range(-5,0):
    print((3*"%10.3f") % tuple(dados[i,:]))
print(35*'-','\n\n')

#%% Resolução do trabalho

# Dados

 #Resistencia
R=4700#'-+ 5% = T_R
erro_R=4700*0.05

 #Condensador
# C=??? (um objectivo 'conhecer seu valor')


# Em t=0 (s), K está na posição 1
# Ao longo do tempo foram registados os valores de V_in e V_c  (volt)

 #Erros de leitura
T_r=0.001 #(s)
V_r=0.005 #(v)

#%% Pergunta 1
```

```
print('Pergunta 1\n\n')

# nº de amostras
ndados=len(dados)
print(u'- Foram feitas ', ndados,'amostras')


# intervalo de tempo entre amostras
T=100/ndados
print (u'- Intervalo de tempo entre amotras = ',format(T,'.3f'),' +- ',T_r,' (s) ')


# frequncia de amostragens
f=1/T  #valor da frequência

t_=sym('t')
f_=1/t
f_r=diff(f_,t_)*T_r  #erro associado ao calculo da frequência
f_r=abs(subs(f_r,t_,T))

print (u'- Frequência de amostragem = ', format(f,'.7f'),' +- ', format(f_r,'.7f'), '(Hz) ')


# tempo durante ocorreu aquisição de  V(in) e V(c)
tempaq=dados[:,0][-1]-dados[:,0][0]
print (u'- Tempo de aquisição = ', tempaq,' +- ', T_r,' (s)')

print('\n\t -----//-----')



#%% Pergunta 2

print('\n\nPergunta 2\n\n')

t=dados[:,0]
Vin=dados[:,1]
Vc=dados[:,2]

plot(t,Vin,t,Vc)
xlabel('t (s)'); ylabel('V (V)')
title(u'Deteção de ocorrências')
show()

print('\n\t -----//-----')

#%% Pergunta 3

print('\n\nPergunta 3\n\n')

l=[]#Lista de derivadas
l1=[]#lista de indices onde a derivada é negativa
l2=[]
tempdes=[]
#definicao de uma lista de derivadas
for i in range(0,len(dados)):
    if(i<len(dados)-1):
        l.append((Vin[i+1]-Vin[i])/T)
    else:
        l.append((Vin[i]-Vin[i-1])/T)

#Pesquisar na lista de derivadas, as derivadas negativas
#Lista de indices e instantes onde a derivade é negativa



for i in range(0,len(l)):
    if (l[i]<-1):  #neste caso temos <-1 para desprezar as derivadas de muito pequeno valor que representam erros de leitura.
        l1.append(i)
        tempdes=tempdes+[t[i]]
    if(l[i]>1):
        l2.append(i)
        tempdes=tempdes+[t[i]]

tempdes=tempdes+[t[-1]]
```

```
print(u'As descargas do condesador ocorrem nos seguintes intervalos: \n')
for i in range(0,len(tempdes)-1):
    if ((i%2)==0):
        print('[',tempdes[i],',', tempdes[i+1],']')
print(u'\nTodas com um erro = ', '+- ', T_r, ' (s)')


print('\n\t -----//-----')


#%% Pergunta 4

print('\n\nPergunta 4\n\n')

#%% A)
Vdc=Vc[l1[0]] #Vc no momento da descarga
Vtau=0.369*Vdc #Vc para calcular o tau

for i in range(l1[0],l2[0]):    #ciclo para descobrir o indice para calcular tau
    if(Vc[i]<Vtau):
        itau=i
        break

tau=t[i]-t[l1[0]]   #calculo do tau

print(u'O Valor de tau e: \n')
print(tau, '(s) +-', T_r)

print('\n\t -----//-----')

#%% B)

#Equacao para calcular C
C1=tau/R
C=FIX(C1,5)
#derivada em ordem a R
dCr=-(tau/(R*R))
#derivada em ordem a tau
dCtau=1/R

merro=abs(dCtau*T_r)+abs(dCr*erro_R)
merro=2*10**(-5)

print(u'O Valor de C e: \n')
print(C, 'f  +-', merro)

#%% C)

#Definicao dos arrays
dQt=[]
Qt=[]

#definir de Q(t)
for i in range(0,len(dados)):
    Qt.append(C1*Vc[i])

#Definir um array com as derivadas de Vc(t)
dQt.append((Qt[0+1]-Qt[0])/T)#Primeira derivada

for i in range(1,len(dados)-1): #defenicao das derivadas, ou TVMC
    if(i>0):
        if(i<len(dados)):
            dQt.append((Qt[i+1]-Qt[i-1])/2*T)

dQt.append((Qt[i]-Qt[i-1])/T)

plot(t,Qt)
axis([10,25,0,0.0020])
xlabel('t (s)'); ylabel('Q (t)')
title(u'Grafico de Q(t)')
show()

plot(t,dQt)
```

```
axis([10,25,-0.00005,0.00005])
xlabel('t (s)'); ylabel('I (t)')
title(u'Grafico de i(t)')
grid(0.5)
show()

#%% D)

# Valor de inicio da descarga.
z1=sym('z1');z1=0;
z2=sym('z2');z2=0;

for i in range(0,len(dados)):
    if(dados[i,0]==10.000):
        z1=i
for i in range(0,len(dados)):
    if(dados[i,0]==25.000):
        z2=i
# Como a função está em ordem ao tempo, a largura dos retangulos será o intervalo de tempo entre amostras.
# E as alturas dos retangulos serão os valores de i**2*R, logo a area sera (i**2*R)*T
#Defínir as variaveis
z=[]
zM=sym('zM');zM=0;
zm=sym('zm');zm=0;
#obter os valores da função i**2*R
for i in range(z1,z2):
    z.append((dQt[i]**2)*R)
# Calculo da soma de Riemann a direita (Majorante)
tz=len(z)
for i in range(1,tz):
    zM=(z[i]*T)+zM

print("O valor do Majorante é: ",zM)
# Calculo da soma de Riemann a esquerda (Minorante)
for i in range(0,tz-1):
    zm=(z[i]*T)+zm
print("O valor do minorante é: ",zm)
# Calculo do Erro
erroz=abs(zM-zm)/2
print("O valor do erro é: ",erroz)
# Calculo do Valor Medio
areaz=(zM+zm)/2
print("O valor médio é: ",areaz)
```

Trabalho curricular
PARTE 1
VERSÃO 1

AUTORES:
<1160555> <Erendiro Pedro>
<1161053> <Tiago Ribeiro>
<1160775> <João Neves>
<1161312> <João Costa>

Leitura do ficheiro de dados: Dados_TC_P1_V1.txt ...

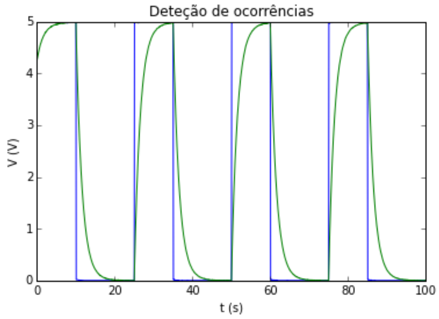
```
t/s Vin/V Vc/V
-----------------------------------
0.000 4.990 4.258
0.077 4.995 4.292
0.154 4.995 4.326
0.231 4.995 4.355
0.308 4.995 4.380
..... ..... .....
99.691 0.000 0.000
99.768 0.000 0.000
99.845 0.000 0.005
99.922 0.000 0.000
99.999 0.000 0.000
-----------------------------------
```

Pergunta 1

- Foram feitas 1301 amostras
- Intervalo de tempo entre amotras = 0.077 +- 0.001 (s)
- Frequência de amostragem = 13.0100000 +- 0.1692601 (Hz)
- Tempo de aquisição = 99.999 +- 0.001 (s)

-----//-----

Pergunta 2



-----//-----

Pergunta 3

As descargas do condesador ocorrem nos seguintes intervalos:

[10.0 , 25.0]
[35.0 , 50.0]
[60.0 , 74.999]
[84.999 , 99.999]

Todas com um erro = +- 0.001 (s)

-----//-----

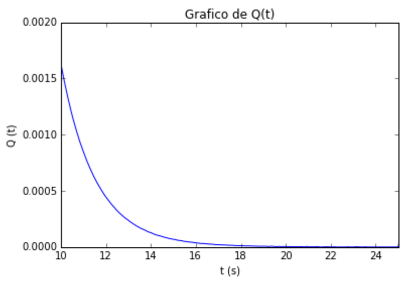
Pergunta 4

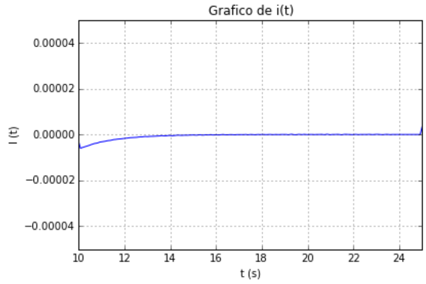
O Valor de tau e:

1.538 (s) +- 0.001

-----//-----
O Valor de C e:

0.00033 f +- 2e-05





O valor do Majorante é: 1.3518979614e-07
O valor do minorante é: 1.38732392125e-07
O valor do erro é: 1.77129799285e-09
O valor médio é: 1.36961094132e-07

Parte 2

```
# -*- coding: utf-8 -*-
"""
Laboratórios de Matemática 2
Parte 2 - Versão 1.2.teste

Data: 2017/04/08

Autores
1. <1160555> <Erendiro Sangueve Njunjuvili Pedro>
2. <1160775> <João Pedro Freitas Neves>
3. <1161053> <Tiago Miguel Pereira Ribeiro>
4. <1161312> <João Pedro Gomes da Costa>
"""
from __future__ import division
from __future__ import print_function
from LabMat1_Lib import *
import pylab as pl


clc()

#%% Bloco de identificação do trabalho
print('\n\nTrabalho curricular\nPARTE 1\nVERSÃO 1\n\n')
print('AUTORES:')
print('\t<1160555>\t<Erendiro Pedro>')
print('\t<1161053>\t<Tiago Ribeiro>')
print('\t<1160775>\t<João Neves>')
print('\t<1161312>\t<João Costa>')

#%% Leitura do ficheiro de dados
# Estrutura de dados: dados=array(Nlinhas,3colunas)
#  t/s |  Vin/V  | Vc/V
#
Ficheiro="Dados_TC_P1_V1.txt"
print('\n\nLeitura do ficheiro de dados:',Ficheiro,'...\n\n')
dados=loadtxt(Ficheiro)
print('      t/s       Vin/V     Vc/V ')
print(35*'-')
for i in range(0,5):
    print((3*"%10.3f") % tuple(dados[i,:]))
print('     .....     .....     .....')
for i in range(-5,0):
    print((3*"%10.3f") % tuple(dados[i,:]))
print(35*'-','\n\n')

#%% Resolução do trabalho

# Dados

 #Resistencia
R=4700#'-+ 5% = T_R
erro_R=4700*0.05
```

```
C=330*10**-6 # 330 microfardas

# Em t=0 (s), K está na posição 1
# Ao longo do tempo foram registados os valores de V_in e V_c  (volt)

 #Erros de leitura
T_r=0.001 #(s)
V_r=0.005 #(v)


#%% Pergunta 1

print("Pergunta 1\n\n")
print("a)\n")

tt=dados[:,0]
Vin=dados[:,1]
Vc=dados[:,2]


td=10     #inicio da primeira descarga
tc=25  #inicio da primeira carga

for i in range(0,len(dados)):
    if (tt[i]==td): break; #valor obtido atravez da primeira parte do trabalho
for j in range(0,len(dados)):
    if (tt[j]==tc): break; #valor obtido atravez da primeira parte do trabalho

Vd=Vc[i]    #tencao no momento da descarga
Vcarg=Vc[j]

print("Valores extremos da tensão:")
print("Tensão no momento de descarga: ", Vd)
print("Tensão no momento de carga: ", Vcarg)

print("\nE como tal, os valores para VM e V0, serão respectivamente: ")
VM=round(Vd); V0=round(Vcarg)
print(VM,",",V0," (V)")

t,T,tau,n=sym('t T tau n')
n,N=symbols('n,N',positive=True,integer=True)
m=inline(t-T*floor((t-tau)/T),(t,tau,T))  # definir função modulus

tau1=round(R*C,1)  #(tau=RC=~1.551=~1.5) aproximacao de tau para nao dar erro nos harmonicos
T=25    #periodo da primeira descarga completa
tau=10       #inteiros mais proximos dos valores resgistrados

for w in range(0,len(tt)): #Procurar posição do em que o tempo assum valor de 10
    if(tt[w]==td):
        break

LVc=VM*(1-e**(-(t-tc)/tau1)); #Defenicao de Vc no momento de carga
LVd=Vc[w]*e**(-(t-td)/tau1);     #Defenicao de Vc no momento de descarga

VC=inline(Piecewise((LVc,t>25),(LVd,t<=25)),t) # definir os 2 ramos da funcao Vc
Vcp=inline(VC(m(t,tau,T)),t)    #defenicao da funcao periodicamente


Vi=inline(Piecewise((VM,t>25),(V0,t<=25)),t) # definir os 2 ramos da Funcao Vin
Vip=inline(Vi(m(t,tau,T)),t)    #defenicao da funcao periodicamente

print("\n\nRepresentação gráfica dos sinais Vin e Vc:")

h1=ezplot(Vcp(t),[0,100])
h2=ezplot(Vip(t),[0,100])
title("Vin e Vc");xlabel("t (s)");ylabel("Vin / Vc (v)")
setp(h2,color='b')
setp(h1, color='r')
axis([0,100,0,8])
grid(1)
show()

print("\t\t--------//--------\n\n\n")
#%% b)
print("b)\n")
#Calculo da potencia media
```

```
print("Aplicando a formula que nos é apresentada no formulário, segue-se:\n ")
PVin=intc(Vi(t)**2,t,tau,(tau+T))/T
print('PVin=', FIX(PVin,2),'(W)\n')

PVc=intc(VC(t)**2,t,tau,(tau+T))/T
print('PVc=', FIX(PVc,2),'(W)')
print("\n\n\t\t--------//--------\n\n\n")
#%% c)
print("c);d)\n")
#series na forma tringonometricas dos sinais

W0=2*pi/T #frequencia angular do sinal

#para vin

a0=2*intc(Vi(t),t,tau,(tau+T))/T
an=2*intc(Vi(t)*cos(n*W0*t),t,tau,(tau+T))/T
bn=2*intc(Vi(t)*sin(n*W0*t),t,tau,(tau+T))/T

#Harmonicos de ordem n
Hvin=inline(an*cos(n*W0*t)+bn*sin(n*W0*t),(n,t))

#para vin

A0=2*intc(VC(t),t,tau,(tau+T))/T
An=2*intc(VC(t)*cos(n*W0*t),t ,tau,(tau+T))/T
Bn=2*intc(VC(t)*sin(n*W0*t),t,tau,(tau+T))/T

#Harmonicos de ordem n
Hvc=inline(An*cos(n*W0*t)+Bn*sin(n*W0*t), (n,t))

#%%d)

#Grafico das aproximacoes

SFVin=a0/2+symsum(Hvin(n,t),n,1,10)
h1=ezplot(SFVin,[0,100])
h2=ezplot(Vip(t),[0,100])
setp(h2,color='b')
setp(h1, color='r')
xlabel('t (s)'); ylabel('Vin (v)')
title(u'Vin + Aproximação pela Serie de fourier')
grid('on')
show()

SFVc=a0/2+symsum(Hvc(n,t),n,1,10)
h1=ezplot(SFVc,[0,100])
h2=ezplot(Vcp(t),[0,100])
setp(h2,color='b')
setp(h1, color='r')
axis([0,100,0,8])
xlabel('t (s)'); ylabel('Vc (v)')
title(u'Vc + Aproximação pela Serie de fourier')
grid('on')
show()

print("\n\n\t\t--------//--------\n\n\n")
#%% e)
#Espectros de Vin
print("e)\n")
Ani=(an**2+bn**2)
AVi=sqrt(Ani) # calculo das amplitudes de Vi
phin=atan2(-bn,an)  #calculo das fases de Vi

Pn=(a0/2)**2+(1/2)*symsum(Ani,n,1,26) #Valor de potencia media Pn
RMSE=100*sqrt((PVin-Pn)/PVin)       #Calculo da percentagem do erro
print('Valor de Pn', FIX(Pn,2),'Watts')
print('Valor de RMSE:', FIX(RMSE,2),'%')



W=[i*W0 for i in range(27)] # calculo da frequencia angular
HVi1=[abs(a0/2)]+[subs(AVi,n,i) for i in range(1,27)]
PHI=[subs(phin,n,i) for i in range(1,27)]
subplot(1,2,1);stem(W,HVi1);axis([0,9,0,3.5]);grid(1)
title(u'Amplitude de Vi')
```

```
subplot(1,2,2);stem(W[1:],PHI);axis([0,9,-4,4]);grid(1)
title(u'Fases de Vi')
show()



#%% Espectros de Vc

Anc=An**2+Bn**2
AVc=sqrt(Anc) # calculo das amplitudes de Vc
phin1=atan2(-Bn,An) #calculo das fases de Vc

Pnvc=(A0/2)**2+(1/2)*symsum(Anc,n,1,6) #Valor de potencia media Pn
RMSE1=100*sqrt((PVc-Pnvc)/PVc)         #Calculo da percentgem do erro
print('Valor de Pn', FIX(Pnvc,2),'Watts')
print('Valor de RMSE:', FIX(RMSE1,2),'%')


W=[i*W0 for i in range(6)] # calculo da frequencia angular
HVc1=[abs(A0/2)]+[subs(AVc,n,i) for i in range(1,6)]
PHI1=[subs(phin1,n,i) for i in range(1,6)]
subplot(1,2,1);stem(W,HVc1);axis([0,2,0,3.5]);grid(1)
title(u'Amplitude de Vc')
subplot(1,2,2);stem(W[1:],PHI1);axis([0,2,-4,4]);grid(1)
title(u'Fases de Vc')
show()


print("\n\n\t\t------------------//------------------\n\n\n")

#%% Resolucao numerica da Serie de fourier para Vc

#Descobrir os indices correrpondentes ao tempo a analizar
taui=10
tf=35

for i in range(0,len(dados)):
    if (tt[i]==taui): break; #Valor de indice do tempo de inicio
for j in range(0,len(dados)):
    if (tt[j]==tf): break; #valor do tempo final


dt=round(tt[len(dados)-1]/len(dados),4) #calculo do intrevalo de tempo dt

Vcdt=[Vc[k]*dt for k in range(i,j)] #calculo numerico do integral de Vc para o calculo de a0
Vcdta=[Vc[k]*cos(n*W0*tt[k])*dt for k in range(i,j)] # integral para an
Vcdtb=[Vc[k]*sin(n*W0*tt[k])*dt for k in range(i,j)] #integral para bn

Na0=(2/T)*sum(Vcdt) #defenicao de a0

Nan=(2/T)*sum(Vcdta) #defenicao de an

Nbn=(2/T)*sum(Vcdtb) #defenicao de bn

NHN=inline(Nan*cos(n*W0*t)+Nbn*sin(n*W0*t),(n,t)) # defenicao da serie de harmonicos

#%% Grafico de arpoximacao numeirca da serie de fourier

SFNVc=Na0/2+symsum(NHN(n,t),n,1,10) #serie de fourier
h1=ezplot(SFNVc,[0,100])
h2=ezplot(Vcp(t),[0,100])
setp(h2,color='b')
setp(h1, color='r')

xlabel('t (s)'); ylabel('Vc (v)')
title(u'Vc + Aproximação pela Serie de fourier')
grid('on')
show()

#%%

NAnc=(Nan**2+Nbn**2)
NAVc=sqrt(NAnc)
Nphin1=atan2(-Nbn,Nan)
```

```
NPnvc=(Na0/2)**2+1/2*symsum(NAnc,n,1,6)
RMSE2=100*sqrt((PVc-NPnvc)/PVc)         #Calculo da percentgem do erro
print('Valor de Pn', FIX(Pnvc,2),'Watts')
print('Valor de RMSE:', FIX(RMSE1,2),'%')

W=[i*W0 for i in range(6)] # calculo da frequencia angular
NHVc1=[abs(Na0/2)]+[subs(NAVc,n,i) for i in range(1,6)]
NPHI1=[subs(Nphin1,n,i) for i in range(1,6)]
subplot(1,2,1);stem(W,NHVc1);axis([0,2,0,3.5]);grid(1)
title(u'Amplitude de Vc')
subplot(1,2,2);stem(W[1:],NPHI1);axis([0,2,-4,4]);grid(1)
title(u'Fases de Vc')
show()



print("\n\n\t\t------------------//------------------\n\n\n")

#%% 3.
print("/n  3.")

# estudo de Vin

# THD = sqrt(h2**2+h3**2+...+h10**2)/h1
# SFVin=a0/2+symsum(Hvin(n,t),n,1,10)
# SFVc=a0/2+symsum(Hvc(n,t),n,1,10)
x_3_vin=0
for i in range(1,10):
    x_3_vin=x_3_vin+(a0/2+symsum(Hvin(n,t),n,i,i))**2

THD_vin=sqrt(x_3_vin)/(a0/2+symsum(Hvin(n,t),n,i,i))
disp(THD_vin)
print("/noi/n")
disp(simplify(THD_vin))
```

Trabalho curricular
PARTE 1
VERSÃO 1

AUTORES:
<1160555> <Erendiro Pedro>
<1161053> <Tiago Ribeiro>
<1160775> <João Neves>
<1161312> <João Costa>

Leitura do ficheiro de dados: Dados_TC_P1_V1.txt ...

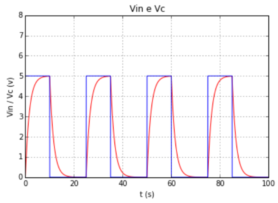
t/s Vin/V Vc/V

0.000 4.990 4.258
0.077 4.995 4.292
0.154 4.995 4.326
0.231 4.995 4.355
0.308 4.995 4.380
....
99.691 0.000 0.000
99.768 0.000 0.000
99.845 0.000 0.005
99.922 0.000 0.000
99.999 0.000 0.000

Pergunta 1

a)

Valores extremos da tensão:
Tensão no momento de descarga: 4.98
Tensão no momento de carga: 0.0



E como tal, os valores para VM e V0, serão respectivamente:
5.0 , 0.0 (V)

Representação gráfica dos sinais Vin e Vc:

--------//--------

b)

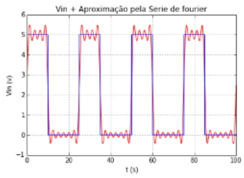
Aplicando a formula que nos é apresentada no formulário, segue-se:

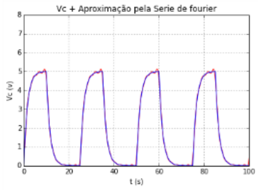
PVin= 10.00 (W)

PVc= 8.40 (W)

--------//--------

c);d)

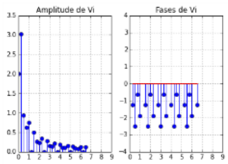




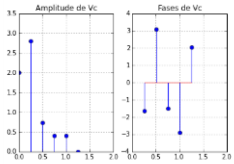
--------//--------

e)

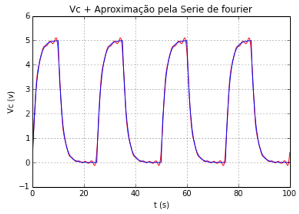
Valor de Pn 9.91 Watts
Valor de RMSE: 9.72 %



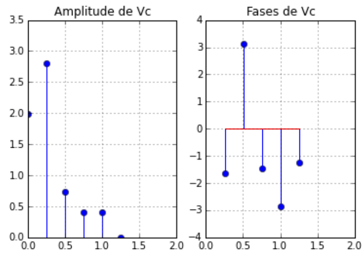
Valor de Pn 8.38 Watts
Valor de RMSE: 4.28 %



------------------//------------------



Valor de Pn 8.38 Watts
Valor de RMSE: 4.28 %



------------------//------------------

/n 3.
sqrt(((-2.5*(-1/4 + sqrt(5)/4)/pi + 2.5/pi)*sin(4*pi*t/25) - 2.5*sqrt(sqrt(5)/8 + 5/8)*cos(4*pi*t/25)/pi + 2.0)**2 + ((-1.66666666666667*(-1/4
+ sqrt(5)/4)/pi + 1.66666666666667/pi)*sin(6*pi*t/25) + 1.66666666666667*sqrt(sqrt(5)/8 + 5/8)*cos(6*pi*t/25)/pi + 2.0)**2 + ((-
0.714285714285714*(-1/4 + sqrt(5)/4)/pi + 0.714285714285714/pi)*sin(14*pi*t/25) - 0.714285714285714*sqrt(sqrt(5)/8 +
5/8)*cos(14*pi*t/25)/pi + 2.0)**2 + ((-0.625*(-1/4 + sqrt(5)/4)/pi + 0.625/pi)*sin(16*pi*t/25) + 0.625*sqrt(sqrt(5)/8 +
5/8)*cos(16*pi*t/25)/pi + 2.0)**2 + ((-5.0*(-sqrt(5)/4 - 1/4)/pi + 5.0/pi)*sin(2*pi*t/25) + 5.0*sqrt(-sqrt(5)/8 + 5/8)*cos(2*pi*t/25)/pi +
2.0)**2 + ((-1.25*(-sqrt(5)/4 - 1/4)/pi + 1.25/pi)*sin(8*pi*t/25) - 1.25*sqrt(-sqrt(5)/8 + 5/8)*cos(8*pi*t/25)/pi + 2.0)**2 + ((-
0.833333333333333*(-sqrt(5)/4 - 1/4)/pi + 0.833333333333333/pi)*sin(12*pi*t/25) + 0.833333333333333*sqrt(-sqrt(5)/8 +
5/8)*cos(12*pi*t/25)/pi + 2.0)**2 + ((-0.555555555555556*(-sqrt(5)/4 - 1/4)/pi + 0.555555555555556/pi)*sin(18*pi*t/25) -
0.555555555555556*sqrt(-sqrt(5)/8 + 5/8)*cos(18*pi*t/25)/pi + 2.0)**2 + 4.0)/((-0.555555555555556*(-sqrt(5)/4 - 1/4)/pi +
0.555555555555556/pi)*sin(18*pi*t/25) - 0.555555555555556*sqrt(-sqrt(5)/8 + 5/8)*cos(18*pi*t/25)/pi + 2.0)
/noi/n

Parte 3

-*- coding: utf-8 -*-
"""
Laboratórios de Matemática 2
Parte 2 - Versão 1.2.teste

Data: 2017/04/08

Autores
1. <1160555> <Erendiro Sangueve Njunjuvili Pedro>
2. <1160775> <João Pedro Freitas Neves>
3. <1161053> <Tiago Miguel Pereira Ribeiro>
4. <1161312> <João Pedro Gomes da Costa>

```
"""
from __future__ import division
from __future__ import print_function
from LabMat1_Lib import *
from LabMat2_Lib import *
import pylab as pl


clc()

#%% Bloco de identificação do trabalho
print('\n\nTrabalho curricular\nPARTE 1\nVERSÃO 1\n\n')
print('AUTORES:')
print('\t<1160555>\t<Erendiro Pedro>')
print('\t<1161053>\t<Tiago Ribeiro>')
print('\t<1160775>\t<João Neves>')
print('\t<1161312>\t<João Costa>')

#%% Leitura do ficheiro de dados
data_p3=loadtxt('data_TC_P3_versao_1.txt')
tt=data_p3[:,0] #discretização do tempo no periodo de definição T.
SFVct=data_p3[:,1] #Aproximação de Fourier de SFVc(t) no periodo de definição T.
print(35*'-','\n\n')

#%% Resolução do trabalho

#Constantes:
R= 1.2 #Ohm
L=2 #H
C_=1 #F


print("Trabalho curricular-Lmat2-Parte 3\n\n")

#%%
print('1)\n')
print(u"*Sem output necessário. Verificar código*")

#%%
print('\n\n2)\n')

ti=tt[0]; tf=tt[-1]

for i in range(0,len(data_p3)):
    if(SFVct[i]==min(SFVct)) :
        break;
icum=i     #mais tarde será utilizado
tcum=tt[i]

print('O instante em que o comutador muda de posição é: ', tcum,' (s)' )
print('O inicio do periodo de definição = ', ti,' (s)' )
print('O final do periodo de definição = ', tf,' (s)' )

#%%
print('\n\n3)\n')


t=sym('t')           # definir t como v. ind.
i=Function('i')         # definir i como v. dep
C=symbols('C{}'.format(1))    # definir a constante C1
EDO=Eq(L*Diff(i(t),t),-R*i(t))  # definir a EDO

#Antes da comutação
#Antes da comutação é como se o condensador não estivesse no circuito, e assim sendo:
#SFVc(t)=Vl+Vr
print("-Antes da comutação o circuito é modelado pela seguinte equação diferencial:" )
display(Eq(sym('SFVct'),L*Diff(i(t),t)+R*i(t)))


#Após a comutação
#Após a comutação o condensador já tem efeito no circuito e portanto:
#SFVc(t)=Vl+Vr+Vc
print("\n\n-Depois da comutação o circuito é modelado pela seguinte equação diferencial:" )
display(Eq(sym('SFVct'),L*Diff(i(t),t)+R*i(t)+(1/C_)*Intc(i(t),t,tcum,tf)))

#%%
```

```
print('\n\n4)\n')

print("Condições iniciais para a equação diferencial antes da comutação: " )
print("i(ti)=0")

print("\nCondições iniciais para a equação diferencial depois da comutação: " )
print("i(tcum)=in e q(tcum)=0")
print("\nLegenda:")

print("ti: instante de inicio do periodo de definição ")
print("tf: intante final do periodo de definição")
print("in: último valor da corrente antes do interruptor abrir ")

#%%
print('\n\n5)\n')

    #Antes do instante de comutação
ii=[]
ii.append(0) #Condição inicial
h=tt[1]-tt[0] #Diferença entre duas amostras consecutivas

for k in range(1,icum):
    ii.append(ii[k-1]+h*(1/L)*(SFVct[k-1]-R*ii[k-1])) #método de Euler

#gráfico
plot(tt[0:len(ii)],ii)
title(u'i(t) antes do instante de comutação')
grid('on');show()

    #Depois do instante de comutação

#De modo a facilitar o calculo pretendido e evitar o calculo de um integral numericamente
#fizemos a seguinte substituição: (dq/dt)=i

qq=[]
qq.append(0)    #Condição inicial
iid=[]
iid.append(ii[-1]) #Condição inicial
ttd=[]
ttd.append(tcum)

j=0;

i1,q=sym('i1 q')

#resolução
for z in range(icum+1,len(tt)):

    j+=1
    #método de Euler
    f=inline(i1,i1)
    g=inline((1/L)*(SFVct[z]-R*i1-(1/C_)*q),(i1,q))

    iid.append(iid[j-1]+h*g(iid[j-1],qq[j-1])) #1 equação do sistema de EDOs
    qq.append(qq[j-1]+h*f(iid[j-1]))    #2 equação do sistema de EDOs

#Deste sistema obtemos duas listas, uma com os valores da carga ao longo do tempo e outra com os valores da corrente

plot(tt[icum:len(tt)],iid)
title(u'i(t) depois do instante de comutação')
grid('on');show()

plot(tt[icum:len(tt)],qq)
title(u'q(t) depois do instante de comutação')
grid('on');show()



#%%
print('\n\n6)\n')


I=ii+iid #Concatenação dos vetores de corrente

Vr=[]   #Vetor com as tensões na resistência
Vl=[]   #Vetor com as tensões na bobina
```

```
Vc=[]   #Vetor com as tensões no condensador
di_dt=[]    #Vetor a derivada da corrente

#Tensão na resistência
k=0
while (k<len(tt)):
    Vr.append(R*(I[k]))
    k+=1

#Tensão no condensador
k=0
for k in range(0,len(ii)):
    Vc.append(0)    #Inicialmente a tensão no condensador é nula

p=0
for k in range(icum,len(tt)):
#Segundo leis fisicas sabemos que a tensão no condesador é dada por: Vc=q*C logo
    Vc.append(qq[p]*C_) #Tensão no condensador após a comutação
    p+=1

#Tensão na bobina
k=0

#Aplicando a lei de Kirchoff para tensões em malhas fechadas, sabemos que a soma das
#quedas de tensão nos componentes do circuito tem que ser igual a tensão debitada pela fonte,
#Assim sendo, após a comutação teremos que Vc=SFVct-Vr-Vl

for k in range(0,len(tt)):
    Vl.append(SFVct[k]-Vr[k]-Vc[k])

print(u"*Sem output necessário. Verificar código*")

#%%
print('\n\n7)\n')


#Representação gráfica da corrente durante o periodo de definição
plot(tt[0:len(tt)],I)
title(u'I(t) durante o periodo de definição do sinal')
grid('on');show()

#Tensão na resistência duarnte o periodo de definição
plot(tt[0:len(tt)],Vr)
title(u'Vr(t) durante o periodo de definição do sinal')
grid('on');show()

#Tensão na bobina duarnte o periodo de definição
plot(tt[0:len(tt)],Vl)
title(u'Vl(t) durante o periodo de definição do sinal')
grid('on');show()

#Tensão no condensador duarnte o periodo de definição
plot(tt[0:len(tt)],Vc)
title(u'Vc(t) durante o periodo de definição do sinal')
grid('on');show()

#%%
print('\n\n\n\t\t',32*'-')
print("\t\t\tFim do trabalho!")


Trabalho curricular
PARTE 1
VERSÃO 1


AUTORES:
<1160555> <Erendiro Pedro>
<1161053> <Tiago Ribeiro>
<1160775> <João Neves>
<1161312> <João Costa>
-----------------------------------


Trabalho curricular-Lmat2-Parte 3
```

1)

Sem output necessário. Verificar código

2)

O instante em que o comutador muda de posição é: 24.154 (s)
O inicio do periodo de definição = 10.077 (s)
O final do periodo de definição = 35.0 (s)

3)

-Antes da comutação o circuito é modelado pela seguinte equação diferencial:

$$SFVct = 1.2 i(t) + 2 \frac{d}{dt} i(t)$$

-Depois da comutação o circuito é modelado pela seguinte equação diferencial:

$$SFVct = 1.2 i(t) + 2 \frac{d}{dt} i(t) + 1.0 \int_{24.154}^{35.0} i(t)\, dt$$

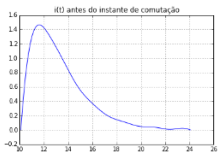
4)

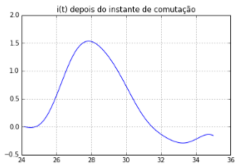
Condições iniciais para a equação diferencial antes da comutação:
i(ti)=0

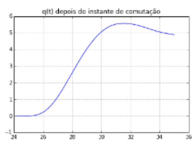
Condições iniciais para a equação diferencial depois da comutação:
i(tcum)=in e q(tcum)=0

Legenda:
ti: instante de inicio do periodo de definição
tf: intante final do periodo de definição
in: último valor da corrente antes do interruptor abrir

5)



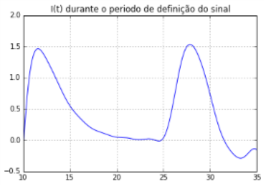


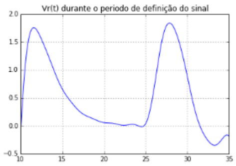


6)

Sem output necessário. Verificar código

7)





Fim do trabalho!

Autoria, Responsabilidade e Autoavaliação

Os trabalhos curriculares devem ser elaborados exclusivamente pelos elementos que se declaram como seus autores, os quais devem ter iguais contribuições. Se tal não tiver assim ocorrido, as contribuições de cada elemento devem constar nesta secção, indicando claramente o que foi feito por cada elemento. Devem ainda qualificar ou quantificar (em percentagem) o contributo de cada elemento do grupo para o sucesso do trabalho apresentado (podem e devem realizar a autoavaliação de forma negociada).

<u>Os plágios serão punidos com a anulação integral do relatório apresentado e esse resultado será refletido diretamente na nota final do trabalho curricular.</u>

Erendiro Pedro 1160555: 24%

Tiago Ribeiro 1161053: 24%

João Costa 1161312: 28%

João Neves 1160775: 24%

Assinaturas dos autores:

Erendiro Pedro 1160555: Autoavaliação(17)

Tiago Ribeiro 1161053: Autoavaliação(17)

João Costa 1161312: Autoavaliação(19)

João Neves 1160775: Autoavaliação(17)