Análise do Experimento do Pêndulo Simples na Célula de Carga

Universidade Federal de Campina Grande
Disciplina: Instrumentação Científica
Professor Adriano de A. Batista

Sony Gonzaga de Melo Neto - 119110023

8 de abril de 2022

1 INTRODUÇÃO

Este relatório descreverá o processo experimental e a análise de dados de um pêndulo simples obtidos utilizando uma célula de carga, tendo como objetivo a extração da frequência de oscilação do pêndulo.

A célula de carga é um aparato eletrônico que tem como principal função a medição da intensidade de uma força aplicada sobre ele. No geral, as células de carga são compostas por um material que se deforma microscopicamente ao ser nele aplicada uma força. Essa deformação modifica as propriedades estruturais do material e comuta na alteração de sua resistência elétrica. A mudança na resistência provoca variação numa corrente atuando sobre ele, que pode ser interpretada como sinais num osciloscópio e traduzida como informações da força originalmente aplicada.

Esse instrumento, por possuir bastante exatidão nas medidas, é amplamente utilizado principalmente na fabricação de balanças de precisão.

2 DESCRIÇÃO DO EXPERIMENTO

No experimento em questão, uma extremidade de uma corda foi acoplada à uma célula de carga e na outra colocou-se uma massa para executar-se oscilações como as de um pêndulo simples.

O pêndulo foi deixado para balançar por um período de 20 segundos, enquanto os sinais, em Volts, foram registrados no computador num arquivo .csv (Comma-separated values), para cada ponto correspondente no domínio do tempo.

3 O PÊNDULO SIMPLES

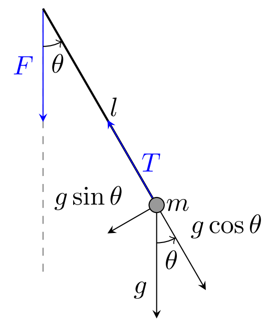

Figura 1: Diagrama de um Pêndulo Simples

O pêndulo simples é um sistema físico de um movimento harmônico, com sua trajetória obedecendo um arco de circuferência. Por conta disso, o ângulo θ varia com o tempo, fazendo sua aceleração tangencial $g \sin \theta$ também ser função do tempo, portanto, o movimento descrito é oscilatório.

A força de aceleração $g \cos \theta$ se equilibra com T, portanto, a única força atuante no sistema é $g \sin \theta$, com o único grau de liberdade θ. A equação de movimento, então, é:

$$\ddot{\theta} + \frac{g}{l} \sin \theta = 0. \tag{1}$$

Para ângulos muito pequenos, $sin\theta \approx \theta$, e a solução da equação será:

$$\theta(t) = \theta_0 \cos(\sqrt{\frac{g}{l}} t). \tag{2}$$

O termo $\sqrt{\frac{g}{l}}$ é denominado de frequência angular do pêndulo, denotada por ω. Como $\omega = 2\pi f$, sendo f a frequência de oscilação,

$$f = \frac{1}{2\pi} \sqrt{\frac{g}{l}}. \tag{3}$$

4 ANÁLISE DOS DADOS

Os dados registrados são exibidos por meio de um programa em Python, que utiliza a biblioteca matplotlib para esse propósito. São dois conjuntos de dados, um cru, formado pelas medições diretas da célula de carga:

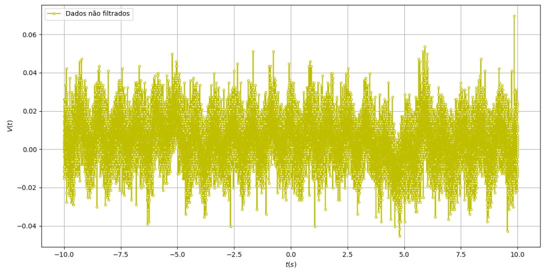

Figura 2: Dados não filtrados

e outro com um filtro aplicado:

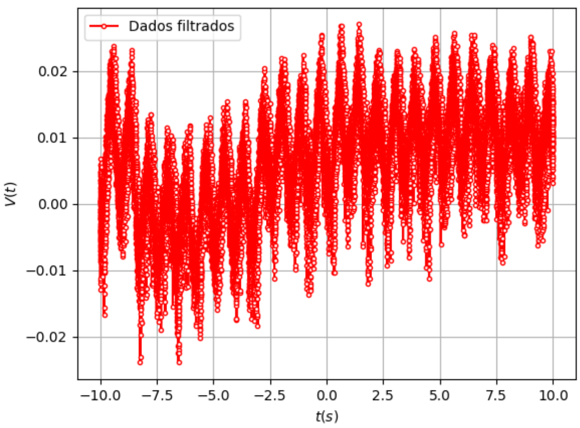

Figura 3: Dados filtrados

Para os dados não filtrados da Figura 2, utiliza-se um Filtro de Média Móvel, utilizando a biblioteca Pandas. O Filtro de Média Móvel calcula a média entre n sinais e a transforma em um novo sinal. Fazendo isso com todos os pontos, os ruídos serão atenuados. Caso n seja muito pequeno, pouco ruído sumirá, mas, caso n seja muito grande, as características originais do gráfico serão perdidas, como sua propriedade oscilatória.

O Filtro de Média Móvel segue a seguinte equação:

$$y[t] = \frac{1}{n}\sum_{k=0}^{n-1} x[t-k]. \quad (4)$$

Os dados, ao passar pelo filtro com $n = 649$, ficarão:

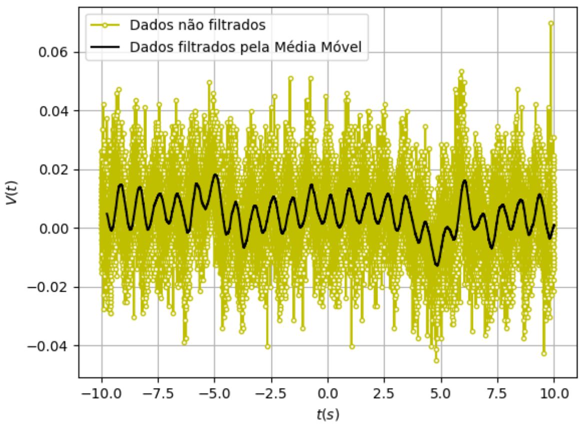

Figura 4: Dados filtrados com Filtro de Média Móvel

É importante observar que a frequência das oscilações obtidas nos sinais da célula de carga é o dobro da frequência de oscilação do pêndulo. Isso pode ser explicado ao verificar que na Figura 1, a força F medida pela célula de carga é uma componente da tensão T: $F = T\cos\theta$. Como T também é componente da força peso, $T = mg\cos\theta$, há a implicação $F = mg(\cos\theta)^2$. Utilizando a identidade trigonométrica $(\cos\theta)^2 = \frac{1}{2}(1+\cos 2\theta)$, conclui-se que $F = \frac{1}{2}mg(1+\cos 2\theta)$, que tem frequência duas vezes a do pêndulo.

A obtenção da frequência de oscilação é feita de duas maneiras, por medição direta e por Transfomada de Fourier.

4.1 MEDIÇÃO DIRETA

Para a medição direta da frequência, cria-se um programa que traça retas entre as posições no tempo entre os máximos e mínimos do gráfico com a biblioteca Scipy.

Nesse caso, o programa identificará os mínimos como os máximos do gráfico espelhado no eixo do horizontal, subtraído da diferença entre seus valores de maior e menor intensidade, para manter os pontos mínimos sempre abaixo dos máximos. Apesar da amplitude ser diferente, a frequência independe desta.

A partir desses pontos, mede-se a diferença entre um máximo e um mínimo consecutivos para todos os períodos, excluindo os primeiros e últimos máximos e mínimos (o programa pode identificar os primeiros e últimos pontos como um máximo ou um mínimo relativo). A média aritimética dessas diferenças dará metade do período médio. A frequência obtida por esse experimento, portanto, será função desse período médio. Para achar o melhor valor de n, faz-se um algoritmo que encontre o maior valor da frequência em função do n. Esta função terá um único pico máximo, portanto, executa-se o algoritmo com os valores da frequência num range razoável que contenha o melhor valor de n. O melhor valor encontrado é $n = 649$, já usado anteriormente.

Os gráficos correspondentes à cada conjunto de dados, com cada frequência do pêndulo encontrada, seguem:

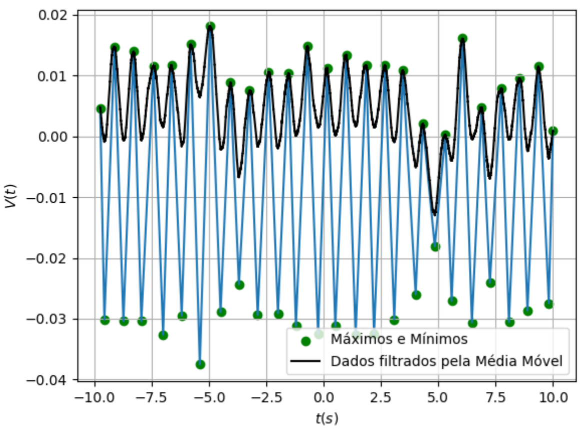

Figura 5: Medida Direta com os Dados Filtrados pela Média Móvel. $f = 0,602119Hz$

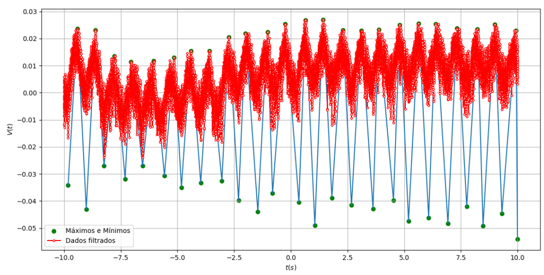

Figura 6: Medida Direta com os Dados Filtrados. $f = 0,612355Hz$

4.2 TRANSFOMADA DE FOURIER

Para a obtenção da frequência de oscilação utilizando a Transformada de Fourier, utiliza-se um programa escrito com base na biblioteca Scipy que plote o gráfico da T.F. de um gráfico de oscilação no domínio das frequências.

O gráfico da T.F. apresentará vários pontos discretos que correspondem às frequências que compõem o gráfico de oscilações e seus respectivos pesos. A frequência mais distinta diferente de zero, a do pêndulo, será a mais proeminente, enquanto as outras são resultantes dos ruídos, podendo ser descartadas.

Para os dados não filtrados, é mais precisa a aplicação da T.F. direta, sem passar pelo filtro. Seguem, portanto, as T.F. com as frequências do pêndulo encontradas:

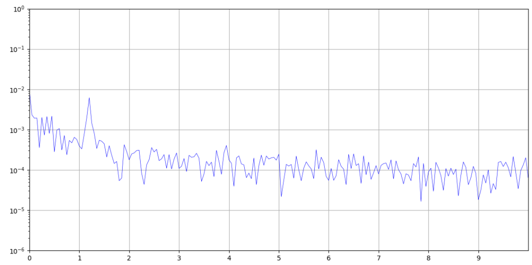

Figura 7: Transformada de Fourier com os Dados Não Filtrados. $f = 0,60003Hz$

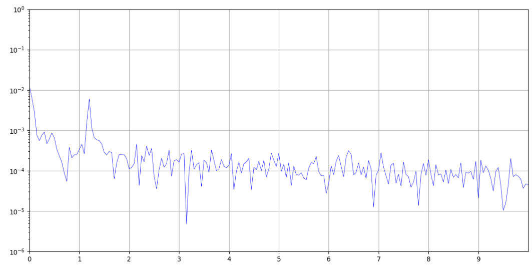

Figura 8: Transformada de Fourier com os Dados Filtrados. $f = 0,60001Hz$

5 CONCLUSÃO

Da análise por medição direta, conclui-se que a frequência de oscilação do pêndulo é $f_1 = (0,6072 \pm 0,0051)Hz$, enquanto a encontrada pela análise das Transformadas de Fourier é $f_2 = (0,60002 \pm 0,00001)Hz$.

Portanto, vê-se que a Transformada de Fourier é um método bem mais eficiente e preciso para a análise dos dados (nesse caso, estacionários e, no caso contrário, não seria tão adequado), pois possui menor desvio padrão, o que pode ser justificado pela menor interferência dos ruídos por causa da disposição dos pontos de frequências separadamente, havendo a possibilidade de ignorar aqueles com picos baixos. Isso não acontece no caso da medida direta, que vai depender da qualidade do filtro utilizado e do tamanho da amostragem, visto que, quanto mais períodos medidos, mais estável tende a ser o valor da frequência medida.

Utilizando a Equação 3 para a f_2, $\pi = 3,14159$ e $g = 9,81m/s$, o comprimento da corda do pêndulo é de $l = 0,690m$ ou $l = 6,90cm$.

Referências

Eduardo O Cerqueira, Ronei J Poppi, Lauro T Kubota, and Cesar Mello. Utilização de filtro de transformada de fourier para a minimização de ruídos em sinais analíticos. *Química Nova*, 23(5): 690–698, 2000. https://www.scielo.br/j/qn/a/hTR8VbLNqbFDPCKcgkfzJMb/?lang=pt.

Carlos E. Rufino da Silva. Relatório do experimento do pêndulo físico física não-linear (tef 2018.1), 2018.

MatPlotLib. Matplotlib documentation. https://matplotlib.org/3.5.0/index.html.

Numpy. Numpy documentation. https://numpy.org/doc/stable/index.html.

Pandas. Pandas documentation. https://pandas.pydata.org/docs/.

Scipy. Scipy documentation. https://docs.scipy.org/doc/scipy/.

Mathuranathan Viswanathan. Understand moving average filter with python & matlab. https://www.gaussianwaves.com/2010/11/moving-average-filter-ma-filter-2/, 2010.

6 APÊNDICE

Programas utilizados:

```
###CÓDIGO DO FILTRO DE MÉDIA MÓVEL E PLOTAGEM DOS GRÁFICOS###
import sys
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import scipy.signal

##Função para o Algoritmo do melhor valor de n da Janela do Filtro Móvel
#def freqop(s):
s = int(sys.argv[2]) #Valor n da Janela do Filtro Móvel
df = pd.read_csv(sys.argv[1], sep =',', names=['t', 'theta'], header=2) #Leitura dos Dados para DataFrame
data = (pd.merge(df['t'], df['theta'].rolling(s).mean(), right_index=True, left_index=True)).dropna()#Filtro Móvel

datat = data['t'].to_numpy()#Conversão do DataFrame de Dados de Tempo para um Array

datam = data['theta'].to_numpy()#Conversão do DataFrame de Dados de Sinais para um Array
datam1 = (-data['theta']-(np.max(datam) - np.min(datam))).to_numpy()#Conversão dos Dados de Sinais Espelhados para um Array

p = scipy.signal.find_peaks(datam, distance=1500)[0] #Achando os máximos
p1 = scipy.signal.find_peaks(datam1, distance=1500)[0] #Achando os mínimos

dt = pd.DataFrame(datat[p], columns=['t'])#Conversão de volta dos tempos de máximos para um DataFrame
dt1 = pd.DataFrame(datat[p1], columns=['t'])#Conversão de volta dos tempos de mínimos para um DataFrame

d = pd.DataFrame(datam[p], columns=['peaks'])#Conversão de volta dos máximos para um DataFrame
d1 = pd.DataFrame(datam1[p1], columns=['peaks'])#Conversão de volta dos mínimos um DataFrame
dtm = pd.merge(dt, d, right_index=True, left_index=True)#DataFrame dos Dados de máximos
dtm1 = pd.merge(dt1, d1, right_index=True, left_index=True)#DataFrame dos Dados de mínimos

dtmf = pd.concat([dtm, dtm1]).sort_values(by=['t','peaks'], ascending=[1,1])#Concatenação dos Dados de máximos e mínimos
#print(dtmf)
## Cálculo da Frequência de oscilação do pêndulo
n = 1
l = 0
while n + 2 < len(p):
    l_n = abs(datat[p1[n]] - datat[p[n]])
    l = l+l_n
    n = n+1
  # print(l)
periodo = 2*l/(n-2)
frequencia = float(1/periodo)
freq_s=frequencia/2
  # return freq_s
#print(freq)

## Cálculo do melhor valor de n
#a = np.array([0])
#for s in range(1,1024):
 #   a = np.append(a, freqop(s))
#print(np.max(a))
#print(np.where(a == np.max(a)))
print('A frequência é: ', freq_s)

#Exportação dos dados de máximos e mínimos
arq = sys.argv[1].replace('.csv','') + ' Pontos máximos e mínimos.csv'
dtmf.to_csv(arq, index=False)

#def func(t,a,b,c):
   # return a*np.sin(b*t+c)
#x = data['t'].to_numpy
#y = data['theta'].to_numpy
#scipy.optimize.curve_fit(func, x, y)

#pd.display(df)

#x = np.linspace(-10,10,1000)
#y = 0.009*np.cos(2*np.pi*1.19*x + 1.5) + 0.009  - 0.006*np.cos(2*np.pi*0.06*x + 2) - 0.005

#Plotagem dos gráficos
plt.figure()
plt.scatter(dtmf['t'], dtmf['peaks'], c='green', label='Máximos e Mínimos')
plt.plot(dtmf['t'], dtmf['peaks'])
#plt.title('Frequência de oscilação do pêndulo = %f Hz'%freq)
#plt.plot(df['t'], df['theta'], 'r-o', mfc='w', mec='r', ms=3, label='Dados filtrados')
#plt.plot(x, y, 'r-o', mfc='w', mec='g', ms=1)
plt.plot(data['t'], data['theta'],'r-o', mfc='w', mec='r', ms=3, label='Dados filtrados')
plt.xlabel('$t (s)$')
```

```python
plt.ylabel(r'$V(t)$')
plt.grid()
plt.legend()
plt.show()
```

```python
###CÓDIGO DA TRANSFORMADA DE FOURIER###
import sys
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import scipy.fftpack
plt.rcParams['axes.formatter.use_locale'] = True

df = pd.read_csv(sys.argv[1], sep =',', names =['t', 'x'], header=2) #Leitura dos Dados para DataFrame
#pd.display(df)

#Plotagem dos gráficos dos Dados
plt.figure()
plt.plot(df['t'], df['x'], mec='black', ms=1, label = 'Dados não filtrados')
plt.xlabel('$t (s)$')
plt.ylabel(r'$V(t)$')
plt.legend()
plt.grid()
#print(df)
#figura = sys.argv[1].replace('.csv', '.pdf')
#plt.savefig(figura)
#print(figura)

#Transformada de Fourier
def FFT(signal, time):
    N_fft = len(signal)
    fft = 2*np.abs(scipy.fftpack.fft(signal))/N_fft
    dt = time[1]-time[0]
    freqs = scipy.fftpack.fftfreq(signal.size, dt)
    return [freqs, fft]

#Taxa de amostragem para FFT
sampRate = 1
time = np.array(df['t'])
signal = np.array(df['x'])
time = time[::sampRate]
signal = signal[::sampRate]

#Plotar serie temporal
fig, ax = plt.subplots(1, figsize=(8, 8))

#Plotagem do Gráfico da Transformada de Fourier
freqs, FFT = FFT(signal, time)
N_fft = len(freqs)
fft_max = np.amax(FFT[:N_fft//2])
n_max = np.argmax(FFT[:N_fft//2])
f_max = n_max*(freqs[1]-freqs[0])
print ("f_max = ", f_max, freqs[1]-freqs[0])
ax.set_yscale('log')
ax.set_xticks(np.arange(0., 10., 1.))
#ax.set_title('Transformada de Fourier', fontsize=14)
ax.set_xlim(0., 10.)
ax.set_ylim([10**-6, 10**0])
ax.plot(freqs[:N_fft//2], FFT[:N_fft//2], 'b-', linewidth = 0.5)
ax.grid()
plt.show()
```