

Relatório do EP3 de MAC0209

João Pedro Feitosa
Marcelo Nascimento
Renan Ryu Kajihara

5 de julho de 2024

Resumo

O presente relatório descreve as atividades realizadas no terceiro Exercício-Programa da disciplina "Modelagem e Simulação (MAC0209)", que consistiu na coleta de dados de mobilidade pedestre seguida de modelagem, simulação e análise comparativa com base nos dados coletados.

Em adição, foi realizado um vídeo de divulgação da realização do experimento, que pode ser acessado por meio desta [referência](#).

Conteúdo

1	Introdução	3
2	Objetivos	3
3	Cronograma	3
4	Dados e métodos	3
4.1	Coleta e Tratamento de dados	3
4.2	Modelagem do deslocamento a partir dos dados do acelerômetro	4
4.3	Simulações de Monte Carlo e Erro Gaussiano	4
5	Resultados experimentais	4
5.1	Resultado dos dados coletados pelo acelerômetro	4
5.2	Resultado da simulação do deslocamento a partir dos dados do acelerômetro	6
6	Discussão e Conclusão	6

1 Introdução

Com o intuito de explorar o uso de sensores de dados multimodais em modelagem e simulação, realizou-se a coleta de dados de mobilidade pedestre para diferentes tipos de movimento. Os dados coletados foram utilizados para o desenvolvimento de modelos capazes de simular os experimentos realizados, por meio de técnicas analíticas e estocásticas. A fim de verificar a qualidade dos modelos desenvolvidos, por fim, comparou-se as simulações realizadas aos dados reais obtidos por meio da experimentação.

2 Objetivos

O exercício teve como objetivo a implementação de técnicas de modelagem e simulação em dados reais, bem como a comparação entre os dados coletados por meio dos sensores e àqueles resultantes dos processos de modelagem determinística e estocástica desenvolvidos.

3 Cronograma

O desenvolvimento deste Exercício Programa envolveu algumas etapas-chave, elencadas abaixo:

- (24/05-01/07) - Discussão, experimentos provisórios e desenvolvimento do protocolo de aquisição de dados para o Experimento Final;
- 02/07 - Experimento Final, coleta de dados utilizando o aplicativo MultiSensor;
- (02/07-04/07) - Análise de dados, tratamento e implementação de algoritmos de modelagem de dados;
- (04/07-05/07) - Redação do relatório do experimento;

4 Dados e métodos

4.1 Coleta e Tratamento de dados

Utilizou-se o aplicativo MultiSensor para a coleta de dados multimodais. Por meio deste, foi possível coletar, além de vídeos dos experimentos realizados, uma diversidade de sinais medidos por diferentes sensores através do tempo. Dentre esses sensores, o de maior importância foi o acelerômetro, capaz de medir a aceleração do aparelho em 3 diferentes eixos: x,y e z.

Os experimentos consistiram na coleta de dados de mobilidade pedestre para diferentes tipos de movimento. Ao todo, 4 cenários distintos foram considerados, cada um dos quais foi repetido 5 vezes:

1. Movimento uniforme (MU) em trajetória retilínea.
2. Movimento uniformemente acelerado (MUV) em trajetória retilínea.
3. Movimento uniforme (MU) em trajetória "zig zag".
4. Movimento uniformemente acelerado (MUV) em trajetória "zig zag".

Os experimentos foram desenvolvidos em um trecho de calçada aproximadamente plana com um comprimento total de 32 metros. Por meio de marcas no chão, esse trecho foi dividido em 4 subtrechos menores, de 8 metros cada. Em cada experimento, o estudante deveria percorrer o percurso definido segurando o aparelho celular, se atentando às marcas no chão. Ao ultrapassar cada marca, ele deveria realizar uma movimentação vertical acentuada, de forma a proporcionar um pico de aceleração a ser

medido por meio do acelerômetro. Dessa forma foi possível determinar, para cada experimento, os tempos necessários para se atingir cada uma das marcas do trajeto.

Utilizou-se a biblioteca "sidesseing_tools" do Python para o tratamento dos dados coletados, assim como as bibliotecas "NumPy", "Pandas" e "Matplotlib" para a análise dos dados e determinação dos tempos t_0 a t_4 , correspondentes à passagem do estudante por cada uma das marcas em um dado experimento. Para cada um dos cenários considerados, calculou-se a velocidade média e a aceleração média (para casos de MUV) ao longo do percurso.

4.2 Modelagem do deslocamento a partir dos dados do acelerômetro

Para modelar o deslocamento a partir dos dados do acelerômetro, foram utilizados dois métodos: o método de Euler e o método analítico.

Para a modelagem com o método de Euler, que, à partir da derivada de uma função, calcula uma aproximação para o valor de $f(t+dt)$ a partir do valor de $f(t)$, foram utilizadas as seguintes equações:

$$\frac{dv}{dt} = a(t) \rightarrow \frac{v(t+dt) - v(t)}{dt} = a(t) \rightarrow v(t+dt) = v(t) + dt.a(t),$$

$$\frac{ds}{dt} = v(t) \rightarrow \frac{s(t+dt) - s(t)}{dt} = v(t) \rightarrow s(t+dt) = s(t) + dt.v(t),$$

sendo s o deslocamento, v a velocidade, a a aceleração e dt o intervalo entre as observações.

Dessa forma, a partir dos vetores que continham a aceleração e o tempo decorrido que foram gerados pelos dados do acelerômetro, foi simulado a velocidade e o deslocamento a partir das equações acima.

Para a modelagem com o método analítico que, a partir da derivada de uma função, integra-a para determinar a função original, foi utilizada a seguinte equação:

$$S = S_0 + v_0.t + \frac{a.t^2}{2},$$

sendo S o deslocamento no tempo t , v_0 a velocidade inicial, a a aceleração média e S_0 o deslocamento inicial.

Assim, da mesma forma com que foi feito pelo método de Euler, a partir da média dos valores do vetor que continha a aceleração e o tempo decorrido que foram gerados pelos dados do acelerômetro, foi simulado a velocidade e o deslocamento a partir da equação acima.

4.3 Simulações de Monte Carlo e Erro Gaussiano

Definidas as estratégias de modelagem e em posse das estatísticas (médias e desvios padrão de velocidade e aceleração) obtidas pelos dados experimentais, realizou-se simulações de Monte Carlo das trajetórias para cada um dos 4 cenários considerados. No Movimento Uniforme (MU), considerou-se uma velocidade constante k , enquanto no Movimento Uniformemente Acelerado (MUV) considerou-se uma aceleração constante k , e velocidade inicial média v_0 . Em cada uma das simulações, o valor do parâmetro k era alterado, conforme uma distribuição gaussiana, com média igual ao valor médio da variável de interesse - velocidade ou aceleração - e o desvio padrão igual ao desvio padrão medido da variável para o determinado cenário.

5 Resultados experimentais

5.1 Resultado dos dados coletados pelo acelerômetro

A partir da visualização dos gráficos de aceleração no eixo y (que representa a aceleração vertical) medida pelo acelerômetro em função do tempo, na maioria dos experimentos, é possível observar notórios picos, que representam os pulos do pedestre de 8 em 8 metros. Nos gráficos que representam o Movimento Uniforme, é possível observar que a distância entre tais pulos é praticamente igual,

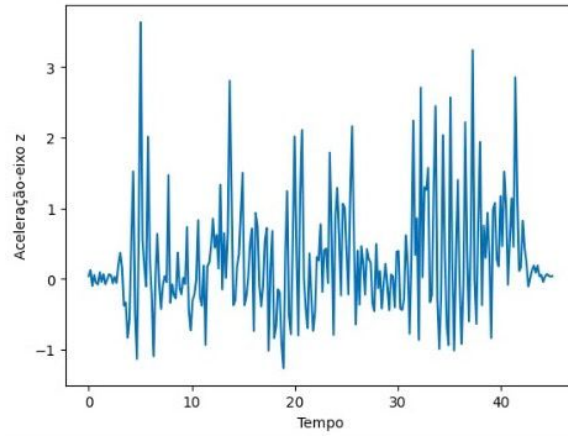


Figura 1: Gráfico do tempo em relação à aceleração do eixo z do MU na trajetória linear

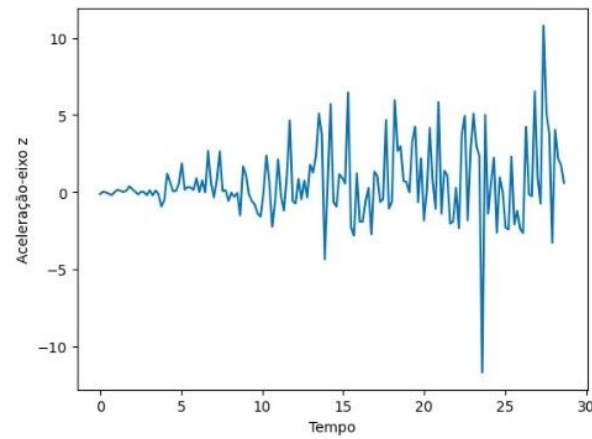


Figura 2: Gráfico do tempo em relação à aceleração do eixo z do MUA na trajetória linear

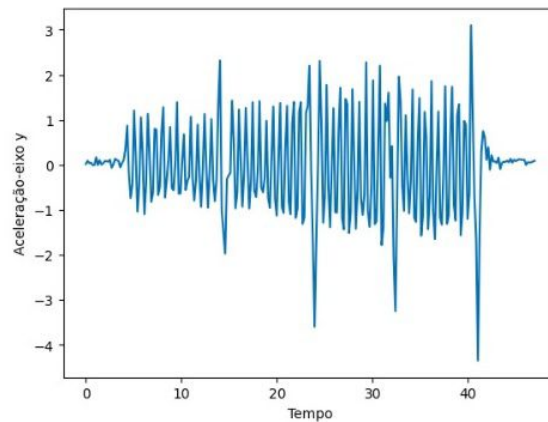


Figura 3: Gráfico do tempo em relação à aceleração do eixo y do MU na trajetória linear

enquanto no Movimento Uniformemente Acelerado, a distância entre tais puls diminui ao decorrer do tempo. Os casos em que os picos não são claramente marcados no gráfico podem ser explicados por uma falha no momento de coleta dos dados, não havendo esforço vertical de magnitude suficiente no momento da passagem pela marca.

Além disso, pelos gráficos que representam a variação do eixo z (que representa a aceleração "para frente") em função do tempo, é possível observar que a variação da aceleração nos experimentos que

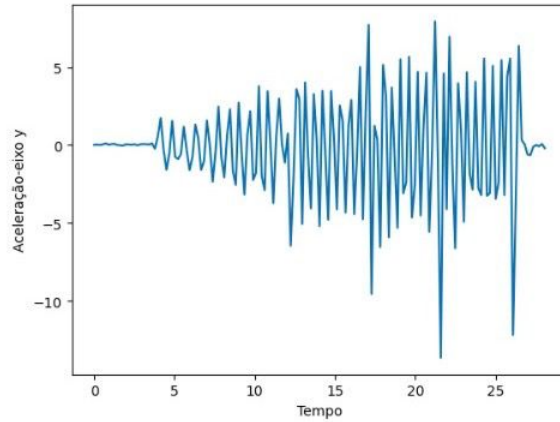


Figura 4: Gráfico do tempo em relação à aceleração do eixo y do MUA na trajetória linear

envolviam o Movimento Uniformemente Variável é muito maior do que nos experimentos que envolviam o Movimento Uniforme.

Por fim, as gravações geradas pelo aplicativo MultiSensor podem ser visualizadas no vídeo de divulgação do experimento referenciado na página inicial deste relatório.

5.2 Resultado da simulação do deslocamento a partir dos dados do acelerômetro

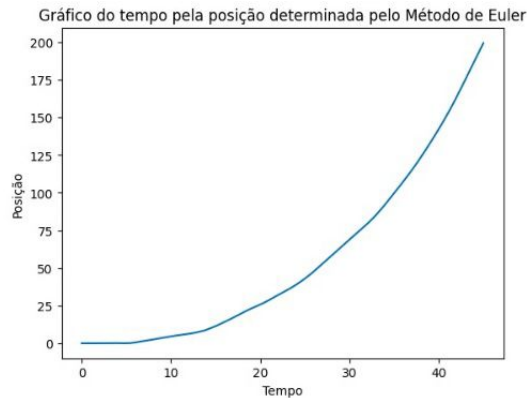


Figura 5: Gráfico do tempo pela posição determinada pelo Método de Euler do MU na trajetória linear

Primeiramente, é notório que os gráficos do deslocamento em função do tempo gerados pelo Método de Euler e pela forma analítica são bem parecidas, uma vez que as equações são bem parecidas entre si.

Nota-se, além disso, que a posição nos gráficos que representam o Movimento Uniformemente Acelerado aumentam mais rapidamente do que nos gráficos que representam o Movimento Uniforme.

Por fim, é notório que a simulação do deslocamento a partir dos dados do acelerômetro é bastante impreciso. Isso ocorre pois, os dados do acelerômetro variam muito quando o pedestre realiza os pulos, gerando uma grande imprecisão nos dados simulados. Além disso, na coleta dos dados houve uma certa demora (cerca de 3 segundos) para que o pedestre começasse a se movimentar, gerando ainda mais imprecisões nas simulações.

6 Discussão e Conclusão

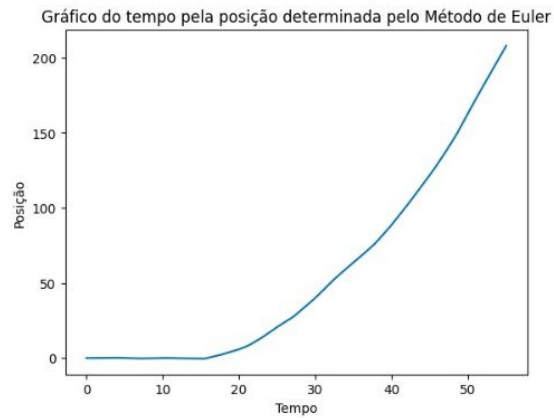


Figura 6: Gráfico do tempo pela posição determinada pelo Método de Euler do MRU na trajetória linear

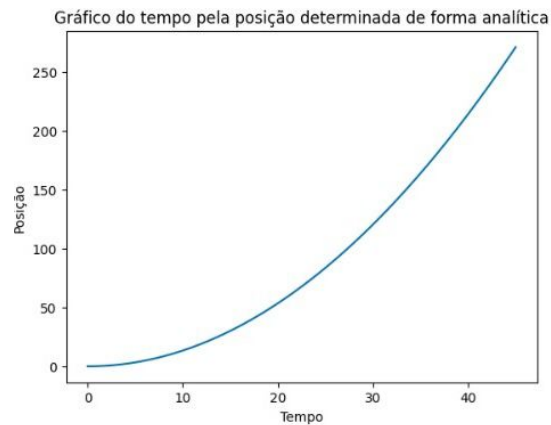


Figura 7: Gráfico do tempo pela posição determinada de forma analítica do MU na trajetória linear

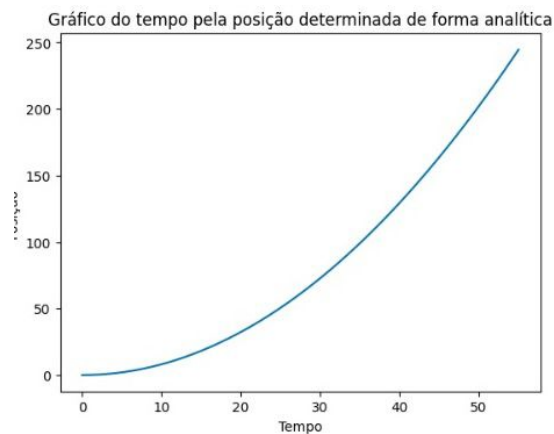


Figura 8: Gráfico do tempo pela posição determinada de forma analítica do MU na trajetória linear

Primeiramente, é notório que os experimentos relacionados ao Movimento Uniforme, na realidade, não possuem velocidade constante, uma vez que é impossível com que o pedestre que coletou os dados mantenha a mesma velocidade em qualquer trajeto. Além disso, da mesma forma, os experimentos relacionados ao Movimento Uniformemente Variado não apresentam aceleração constante. O que houve, na realidade, foi uma tentativa de se aproximar o máximo possível desses objetivos. Uma

sugestão para realizações futuras deste experimento é a utilização de marcações menores para medir o deslocamento realizado e controlar melhor a velocidade de passo, bem como a aceleração no MUV. Outra sugestão é a organização de trajetórias "zig-zag" mais acentuadas, a fim de diversificar a coleta de dados, o que não pode ocorrer devido à limitações ocasionais do experimento.

EP3_MAC0209

July 5, 2024

Exercício de MAC0209 - Modelagem e Simulação

João Pedro Feitosa - 10741569 (IME-USP)

Marcelo Nascimento - 11222012 (IME-USP)

Renan Ryu Kajihara - 14605762 (IME-USP)

EP3 - Mobilidade Pedestre ***

1 Setup

```
[1]: %%capture

# Monta o drive
from google.colab import drive
drive.mount('/content/drive')

# Instalação de pacotes
!pip install sideseeing-tools
!python -m pip install ruptures
!pip install scipy

# Imports
import numpy as np
import pandas as pd
import os
import ruptures as rpt
import math

from matplotlib import pyplot as plt
from sideseeing_tools import sideseeing
from sideseeing_tools import plot
```

2 Lib

```
[2]: def vetor_velocidade (aceleracao, tempo):
    velocidade=[]
    velocidade.append(0)
    for i in range (1,len(aceleracao)):
        velocidade.append(velocidade[i-1]+aceleracao[i-1]*(tempo[i]-tempo[i-1]))
    return velocidade

def vetor_posicao (velocidade, tempo):
    posicao=[]
    posicao.append(0)
    for i in range (1,len(velocidade)):
        posicao.append(posicao[i-1]+velocidade[i-1]*(tempo[i]-tempo[i-1]))
    return posicao

def media(vetor):
    vetor_ =np.array(vetor)
    return np.mean(vetor_)

def desvio_padrao(vetor, media):
    soma=0
    for i in range (len(vetor)):
        soma+= ((vetor[i]-media)**2)
    dp = (soma/len(vetor))*(1/2)
    return dp

def posicao_analitica(s0, v0, a, t):
    s = s0 + v0*t + (a/2)*t**2
    return s

def velocidade_analitica(v0, a, t):
    v = v0 + a*t
    return v

def monte_carlo(vetor):
    ks=[]
    for i in range(len(vetor)):
        k_s= vetor[i] + np.random.normal(0, desvio_padrao(vetor, media(vetor)))
        ks.append(k_s)# Plot velocidade por experimento
    return ks
```

3 Main

```
[3]: %%capture
# Copia os dados para a máquina virtual

!rm -r "/ep3"
!mkdir "/ep3"
!mkdir "/ep3/resultados"
!cp -r "/content/drive/MyDrive/Colab Notebooks/dados_1" "/ep3/"

# Carrega os dados
dados_experimentos = []

dir_root = "/ep3/dados_1"
dir_list = os.listdir(dir_root)
dir_list = sorted(dir_list)

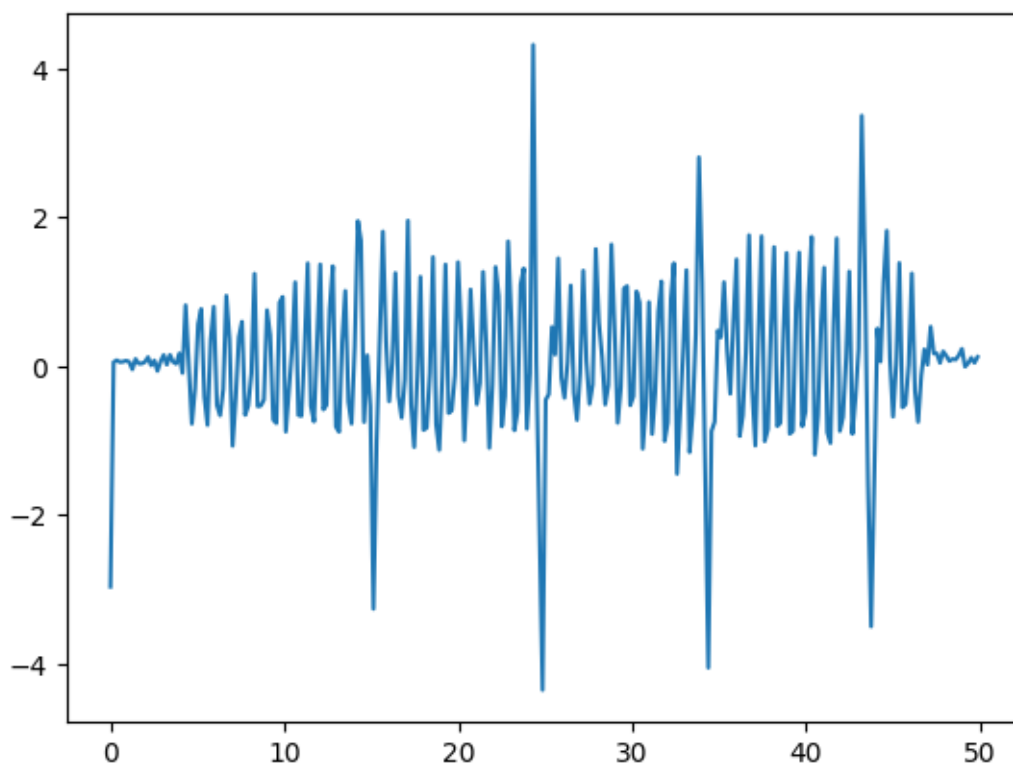
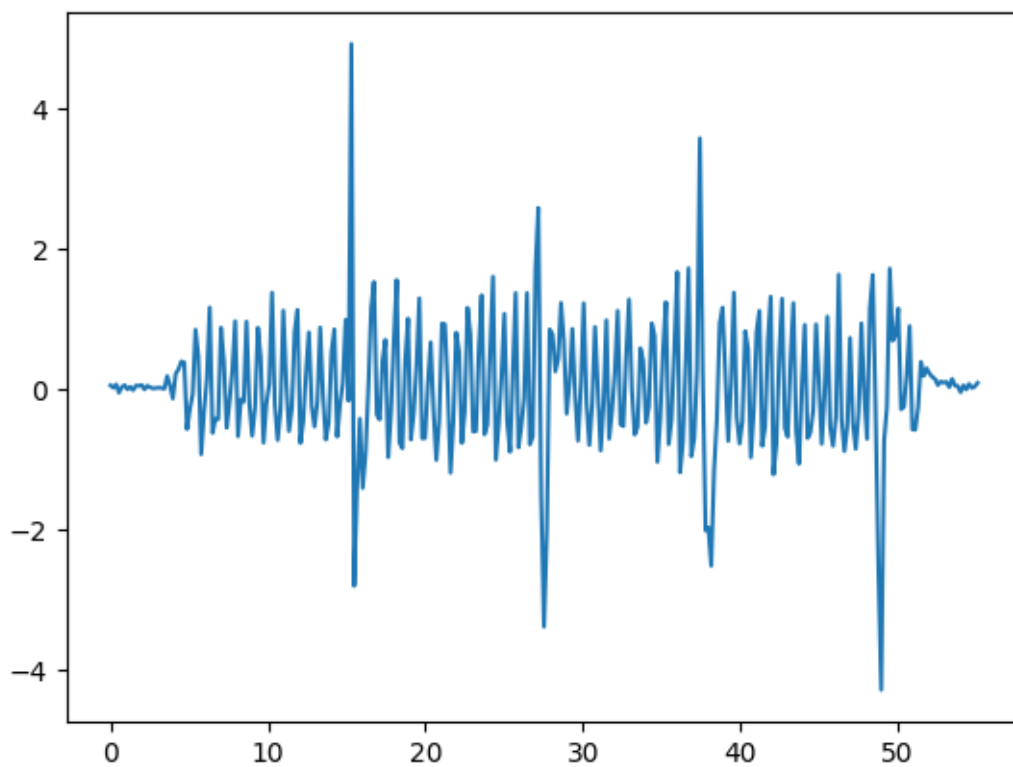
for dir in dir_list:
    ds = sideseeing.SideSeeingDS(root_dir=dir_root + "/" + dir)
    data = ds.instance.sensors3['Linear Accelerometer']
    dados_experimentos.append(data)
```

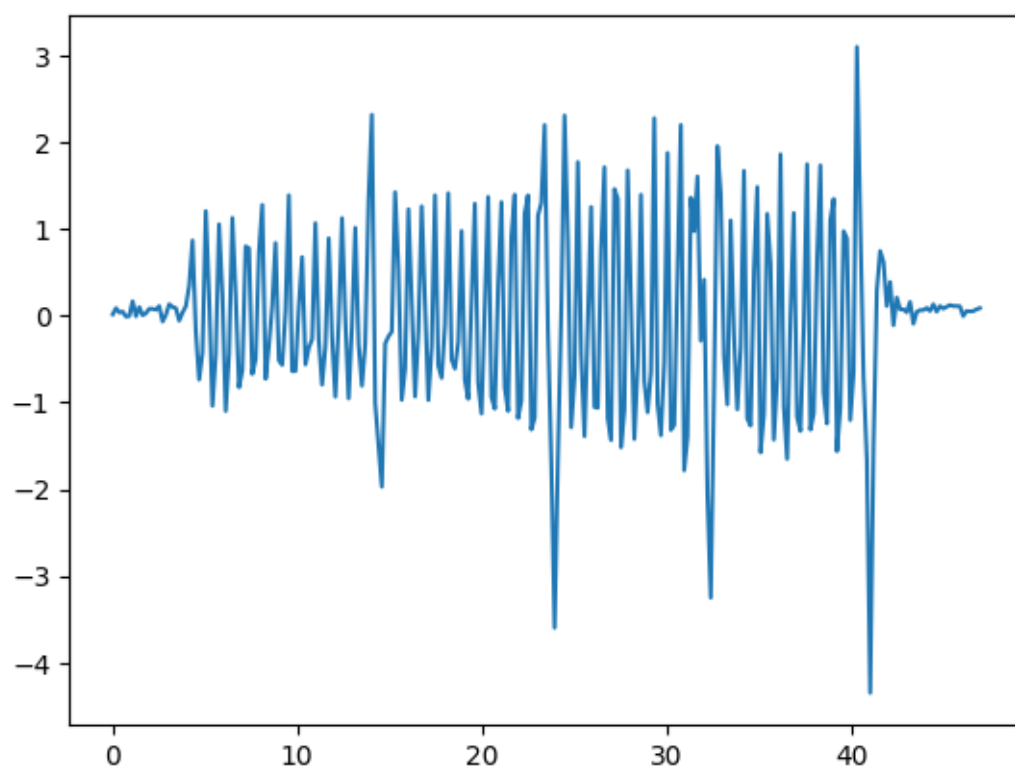
```
[4]: # Sensores
ds.sensors
```

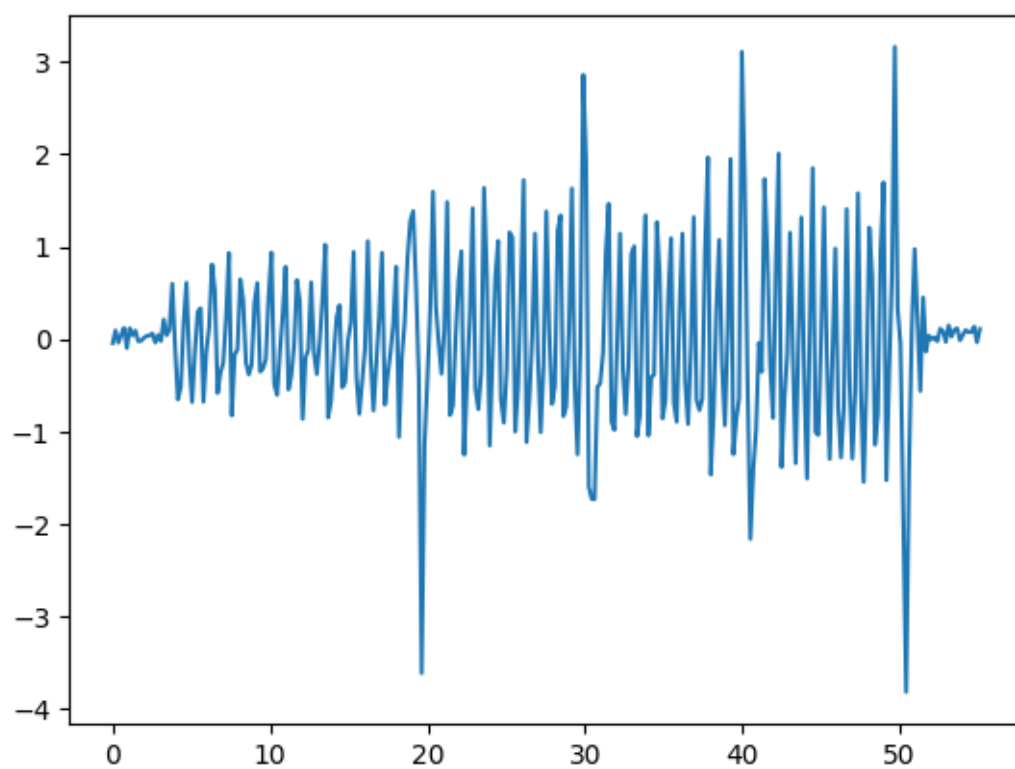
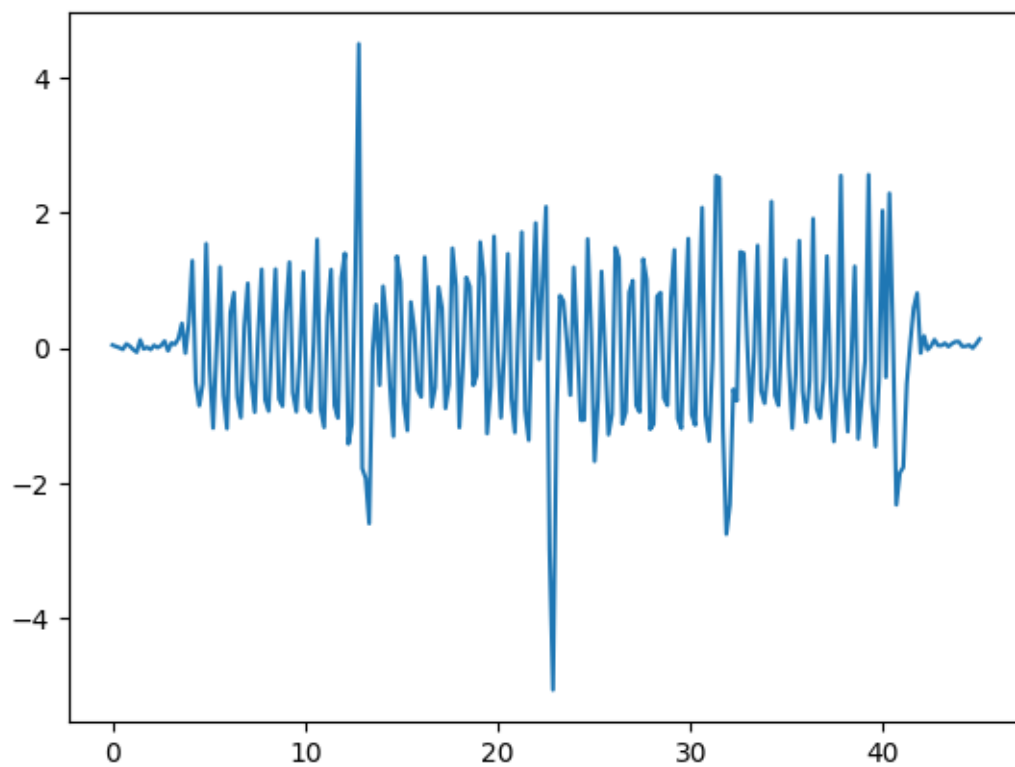
```
[4]: {'Accelerometer',
      'Gravity',
      'Gyroscope',
      'Gyroscope Uncalibrated',
      'Light Uncalibrated',
      'Linear Accelerometer',
      'Magnetometer',
      'Magnetometer Uncalibrated'}
```

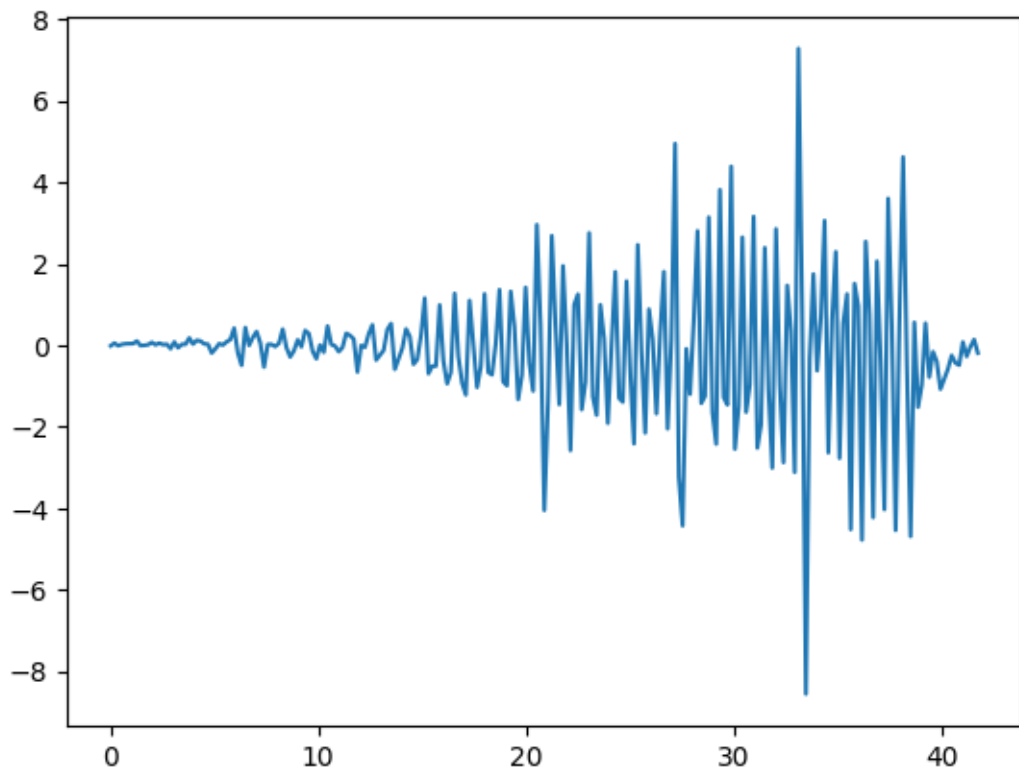
```
[5]: # Plot eixo y

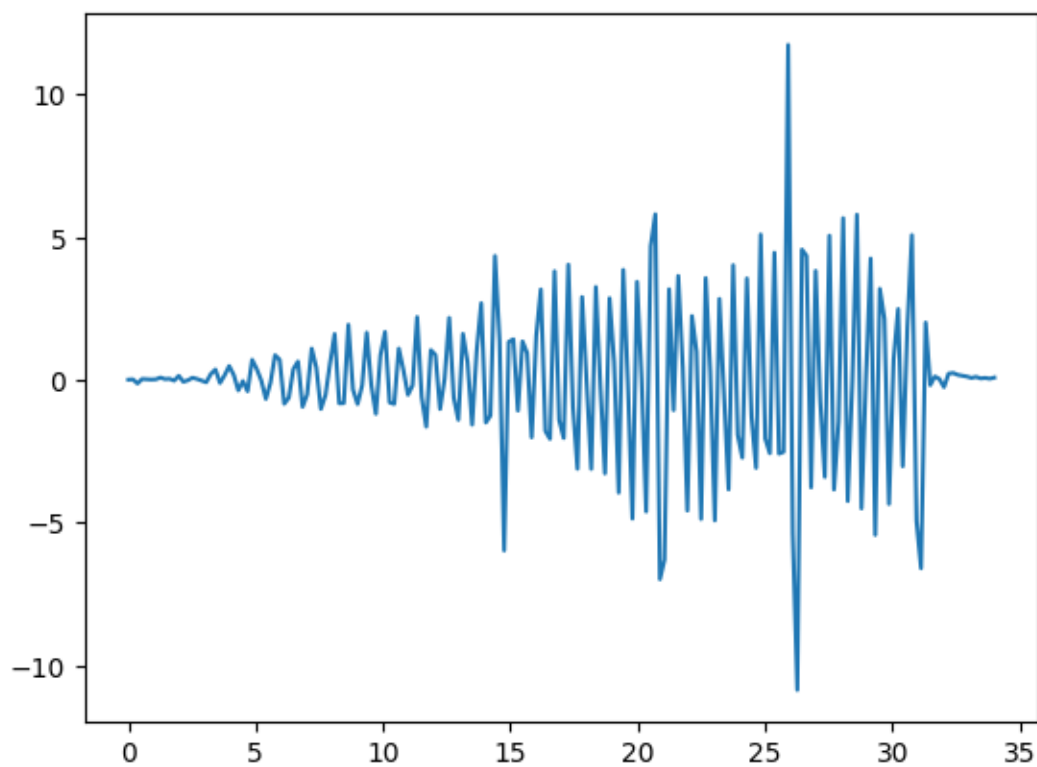
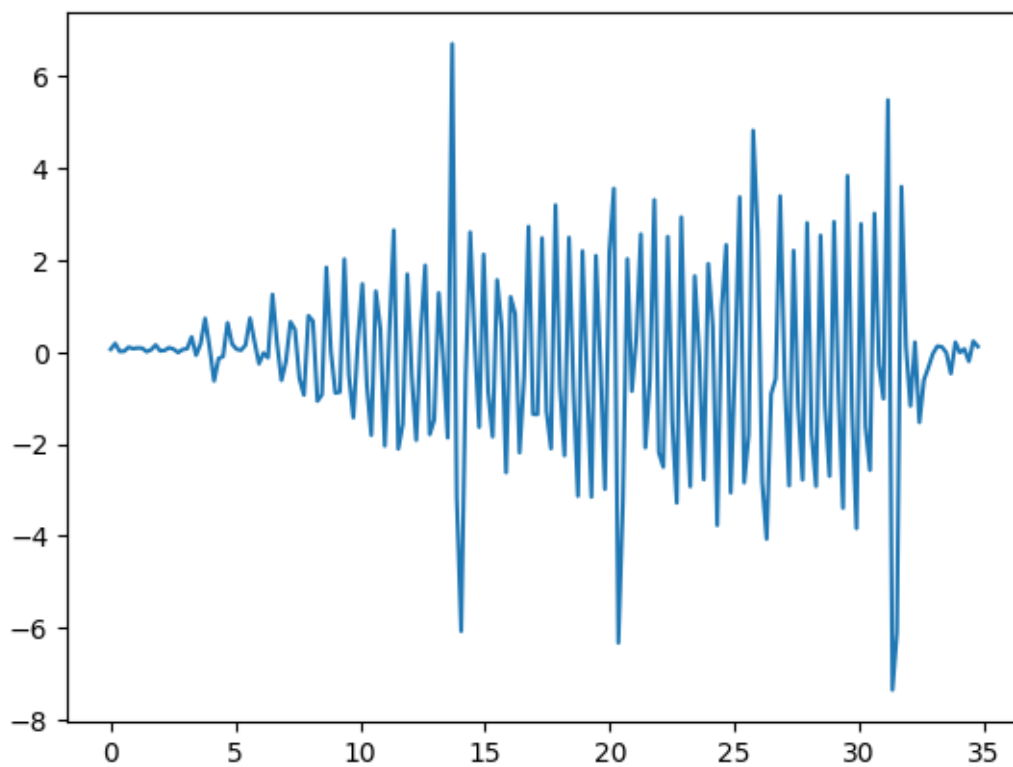
for experimento in dados_experimentos:
    t = np.array(experimento['Time (s)'])
    y = np.array(experimento['y'])
    plt.figure()
    plt.plot(t, y)
    plt.show()
```

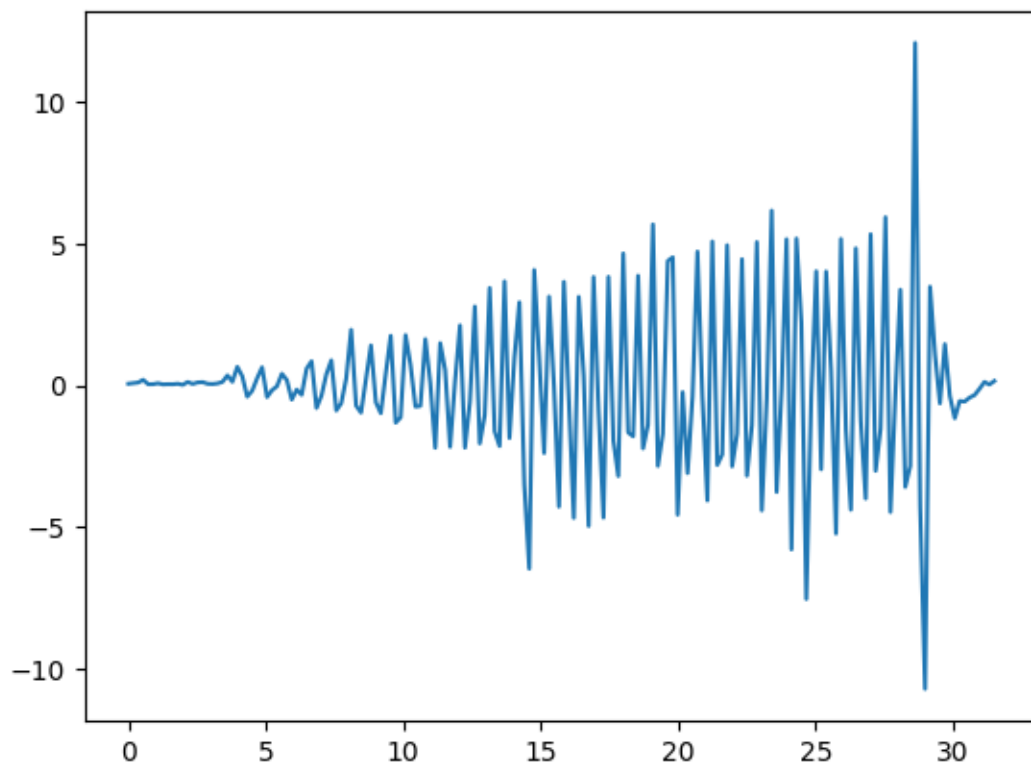


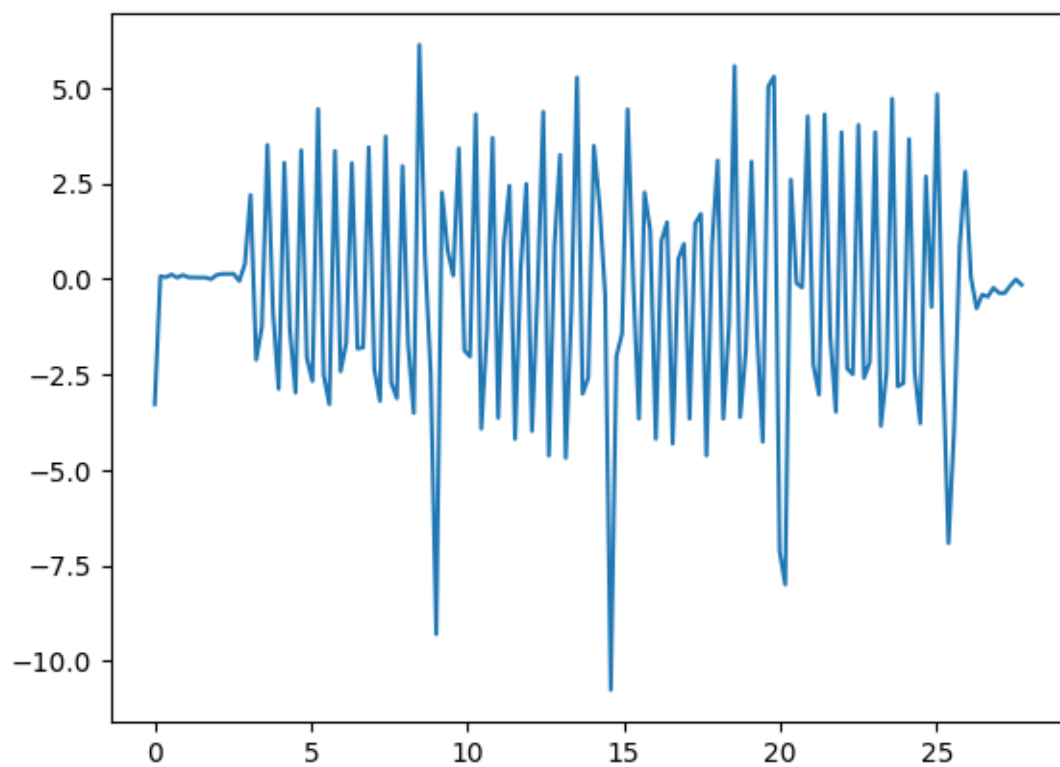
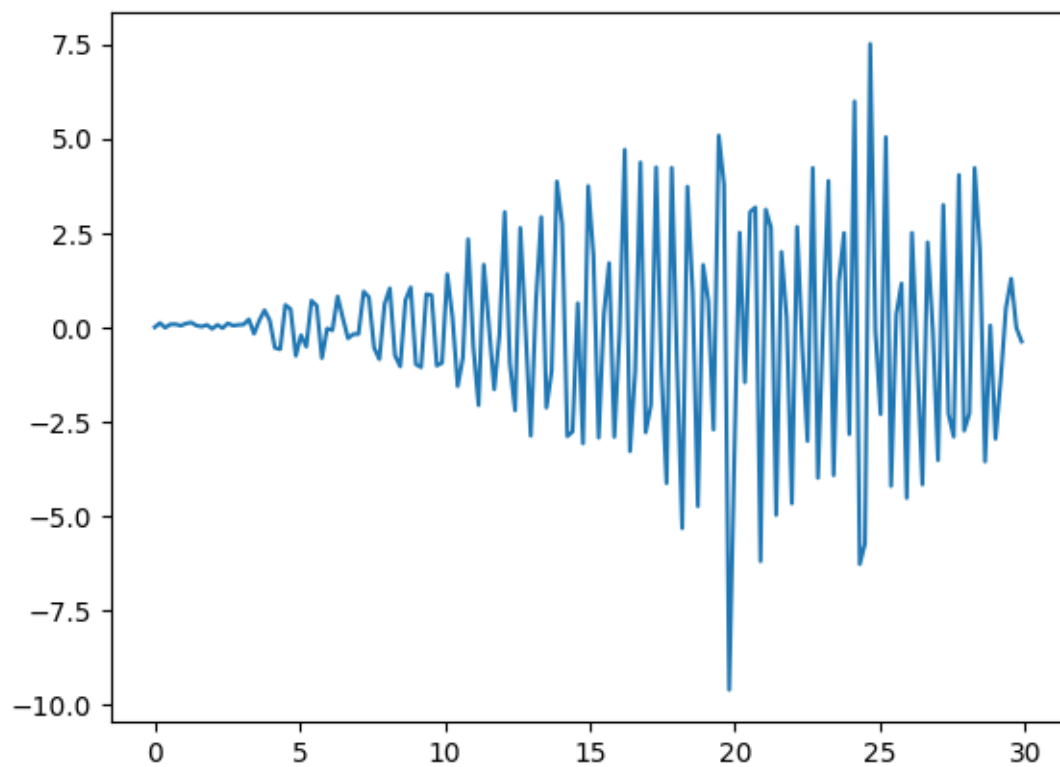


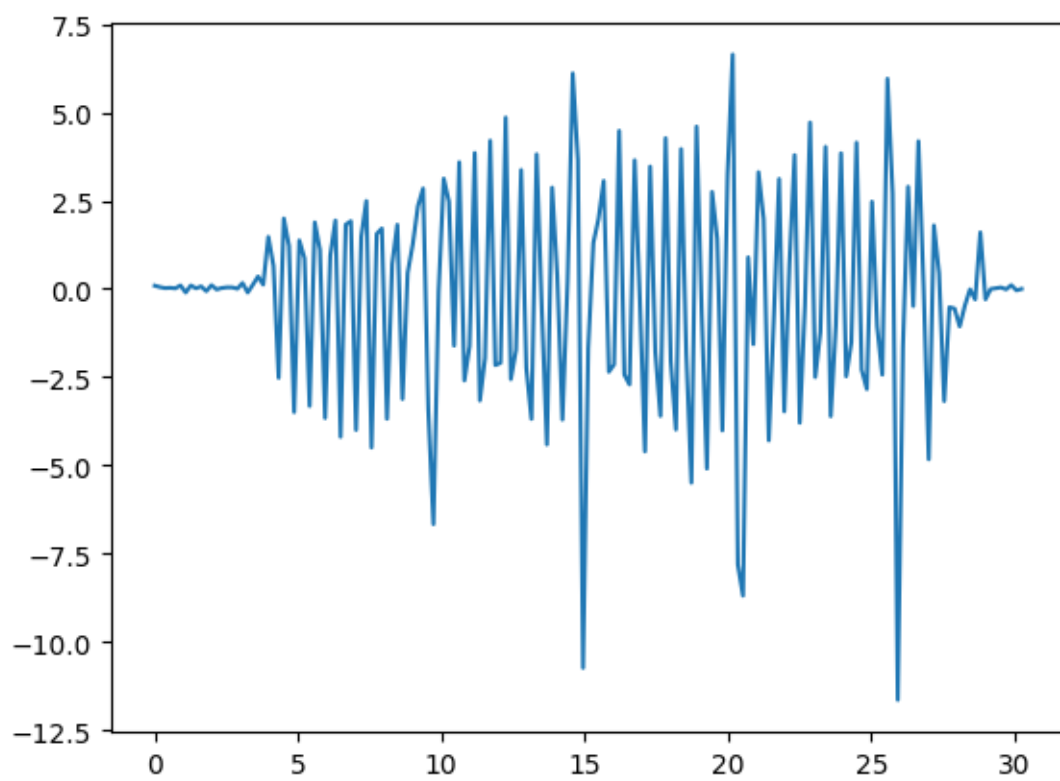


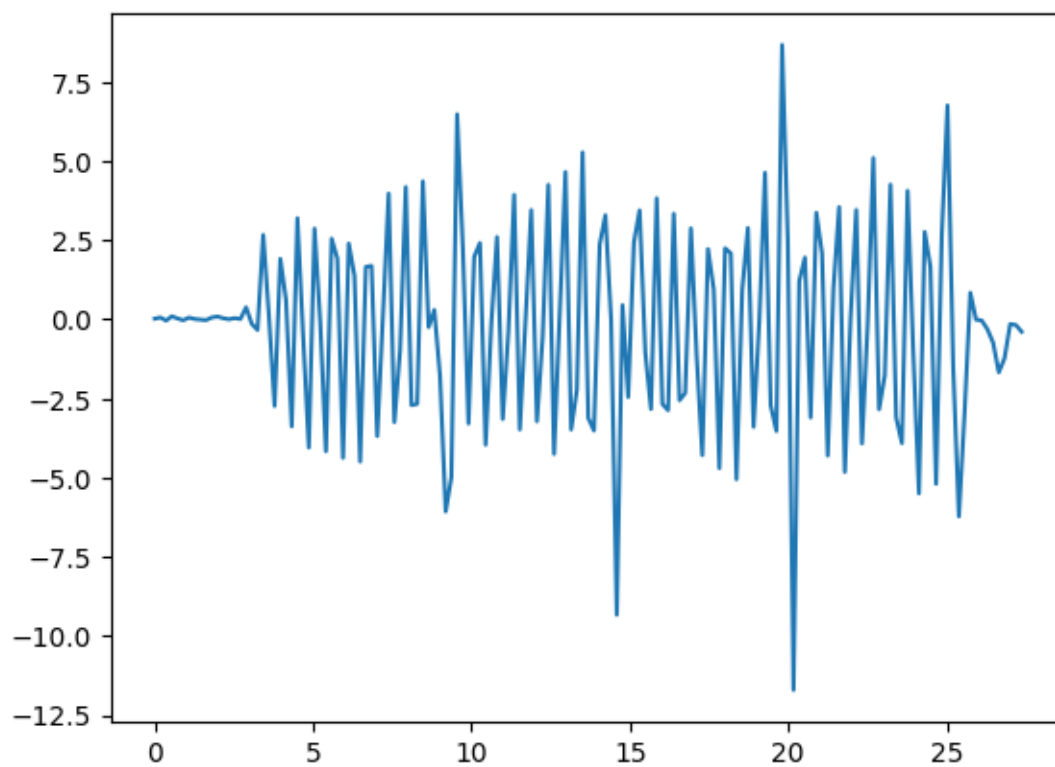
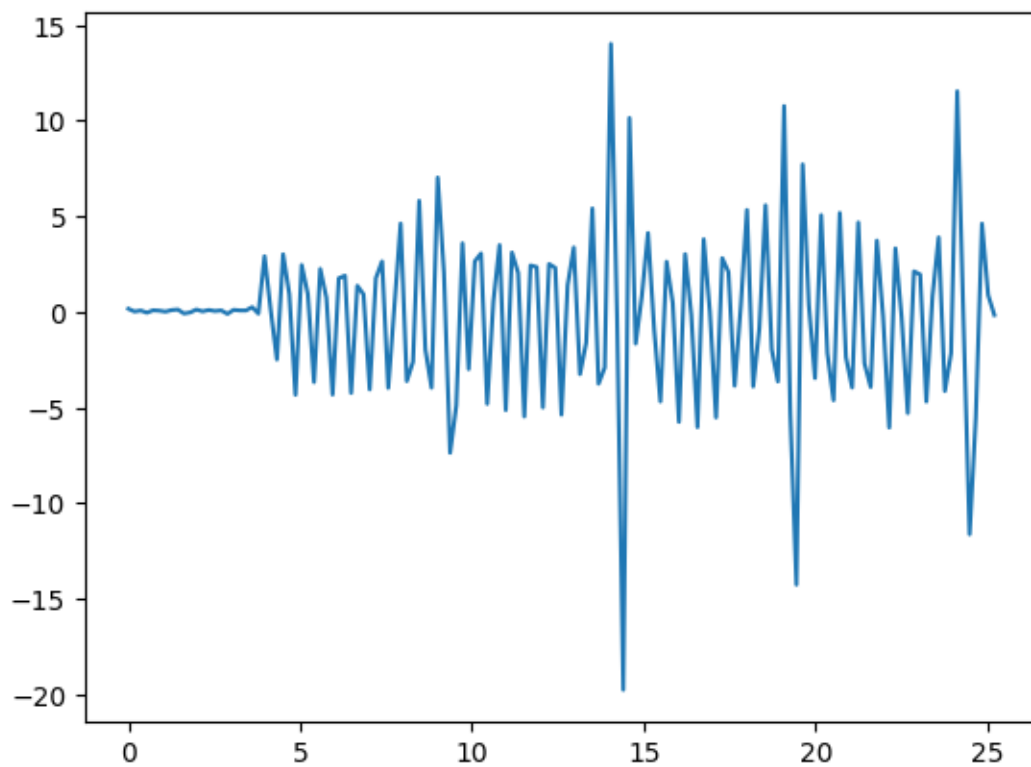


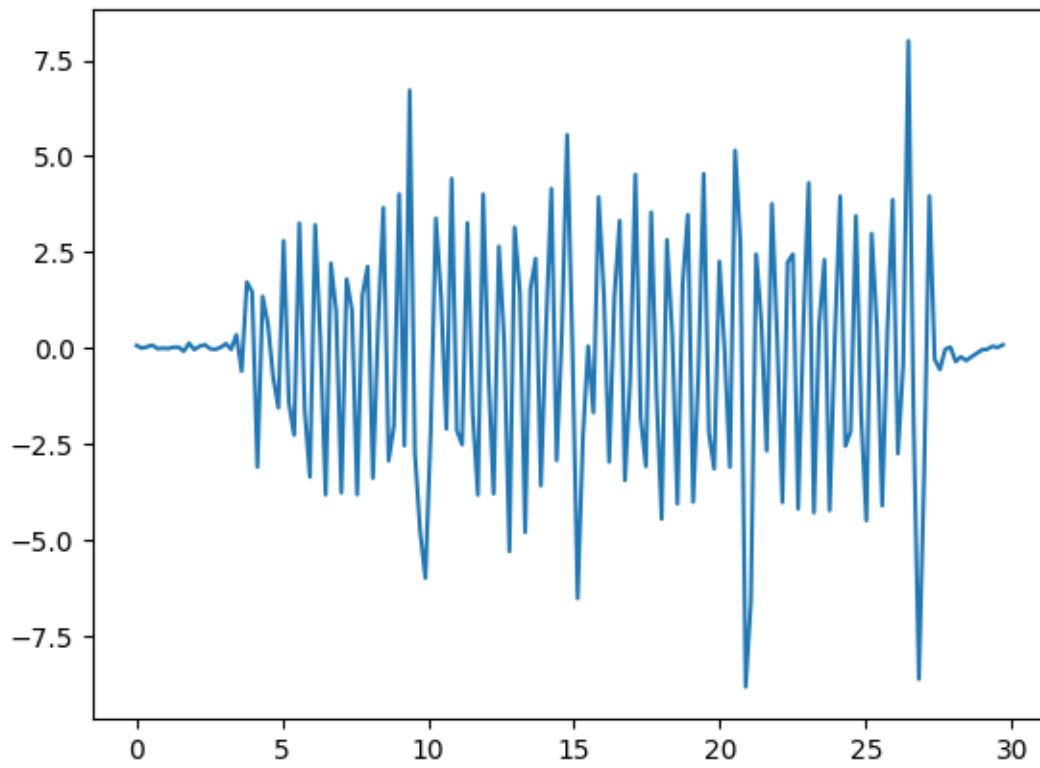


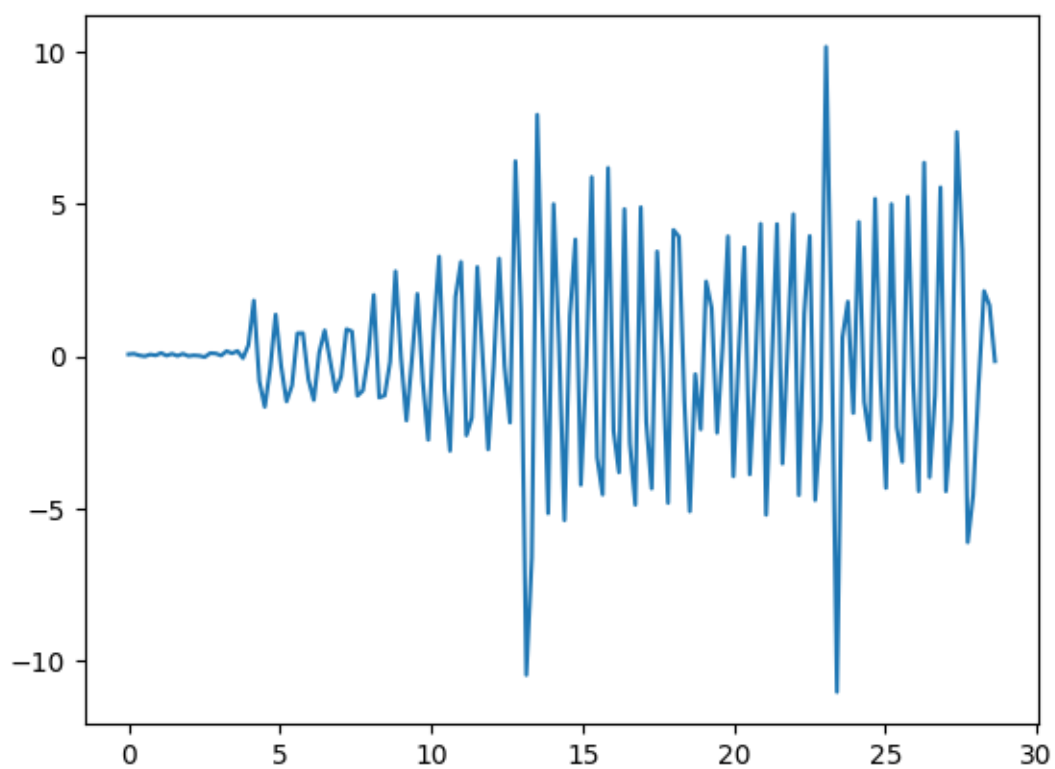
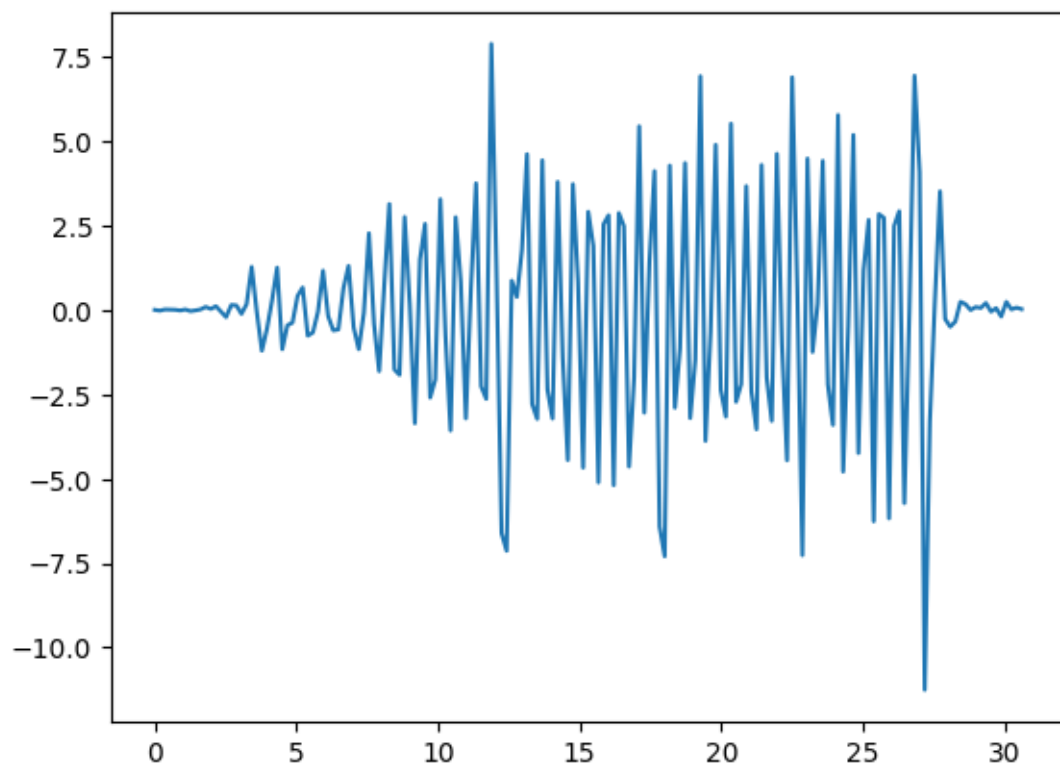


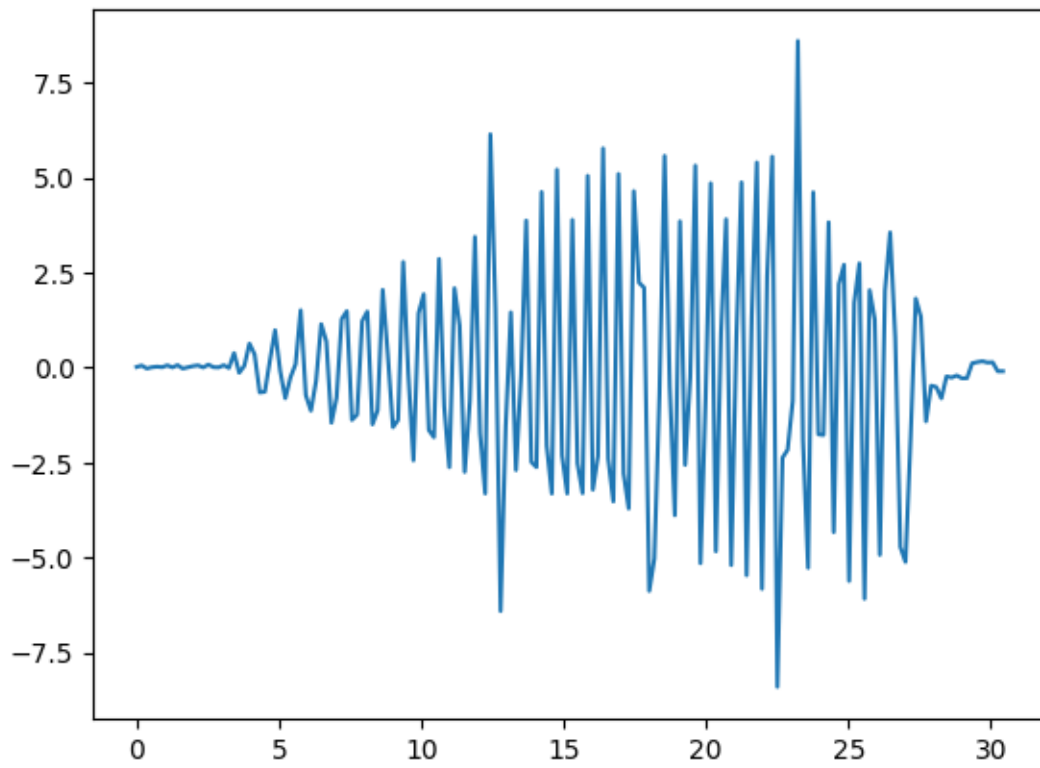


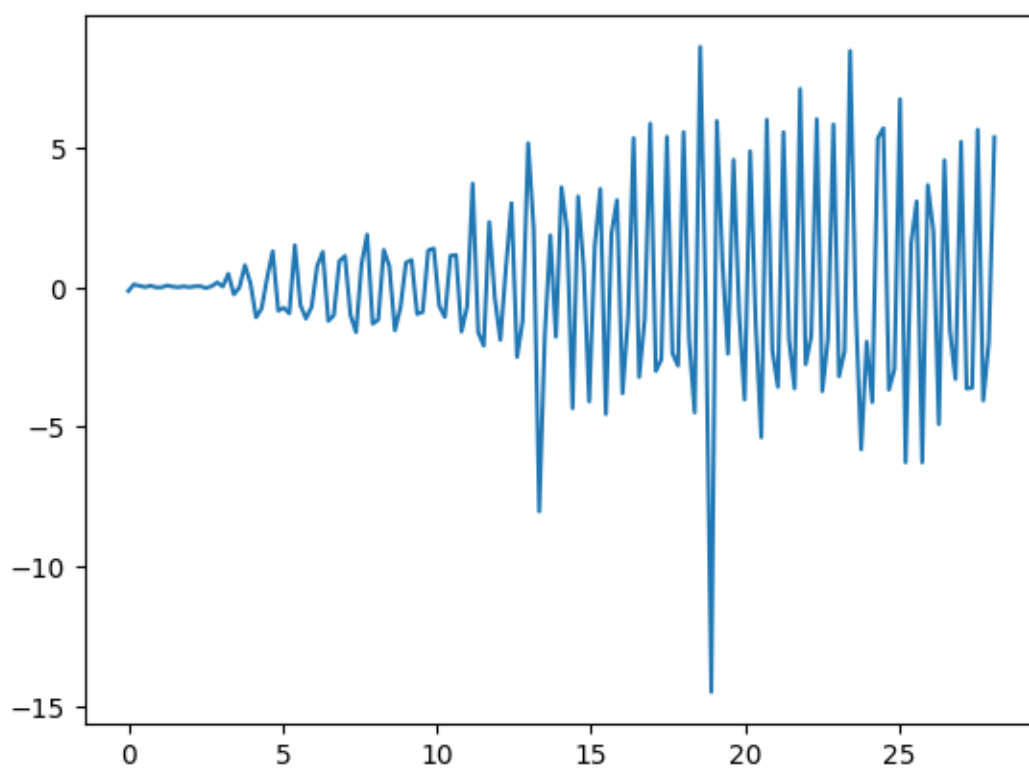
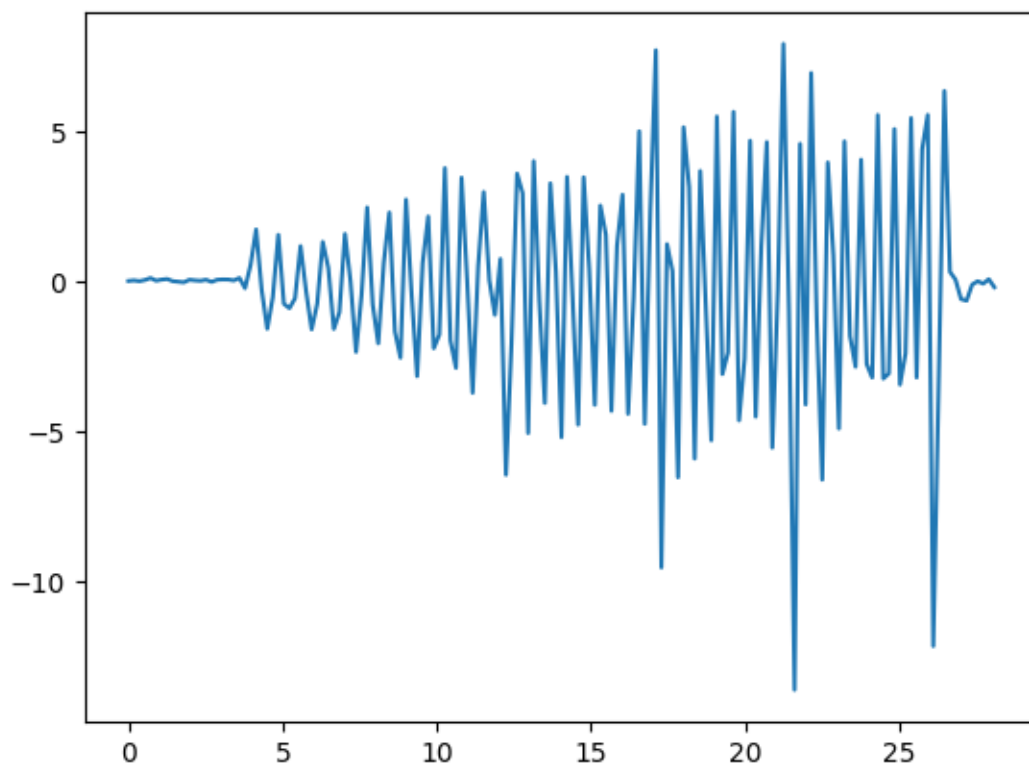













```
[6]: # Visualização dos vídeos
```

```
'''
from IPython.display import HTML
from base64 import b64encode
mp4 = open('/ep3/video_0_9.mp4', 'rb').read()
data_url = "data:video/mp4;base64," + b64encode(mp4).decode()
HTML("""
<video width=400 controls>
    <source src="%s" type="video/mp4">
</video>
""" % data_url)
'''
```

```
[6]: '\nfrom IPython.display import HTML\nfrom base64 import b64encode\nmp4 =\nopen(\'/ep3/video_0_9.mp4\',\'rb\').read()\ndata_url = "data:video/mp4;base64," +\nb64encode(mp4).decode()\nHTML("""\n<video width=400 controls>\n    <source\nsrc="%s" type="video/mp4">\n</video>\n"" % data_url)\n'
```

```
[7]: # Detecta os picos min
```

```
print("Tempos")
c = 5

pontos_mins = []

intervalo_min = [
    [90, 150, 210],
    [80, 140, 190],
    [80, 130, 180],
    [70, 130, 180],
    [110, 170, 225],
    [120, 150, 190],
    [75, 115, 150],
    [80, 115, 150],
    [80, 115, 135],
    [80, 115, 135],
    [50, 80, 115],
    [50, 80, 115],
    [50, 80, 110],
    [50, 80, 110],
    [55, 80, 115],
    [70, 100, 130],
    [70, 100, 130],
    [70, 100, 130],
```

```

    [70, 100, 120],
    [70, 105, 135]
]

index_experimento = 0
for experimento in dados_experimentos:
    data = experimento['y']
    t = experimento['Time (s)']
    plt.figure()
    plt.plot(data, label = 'raw')

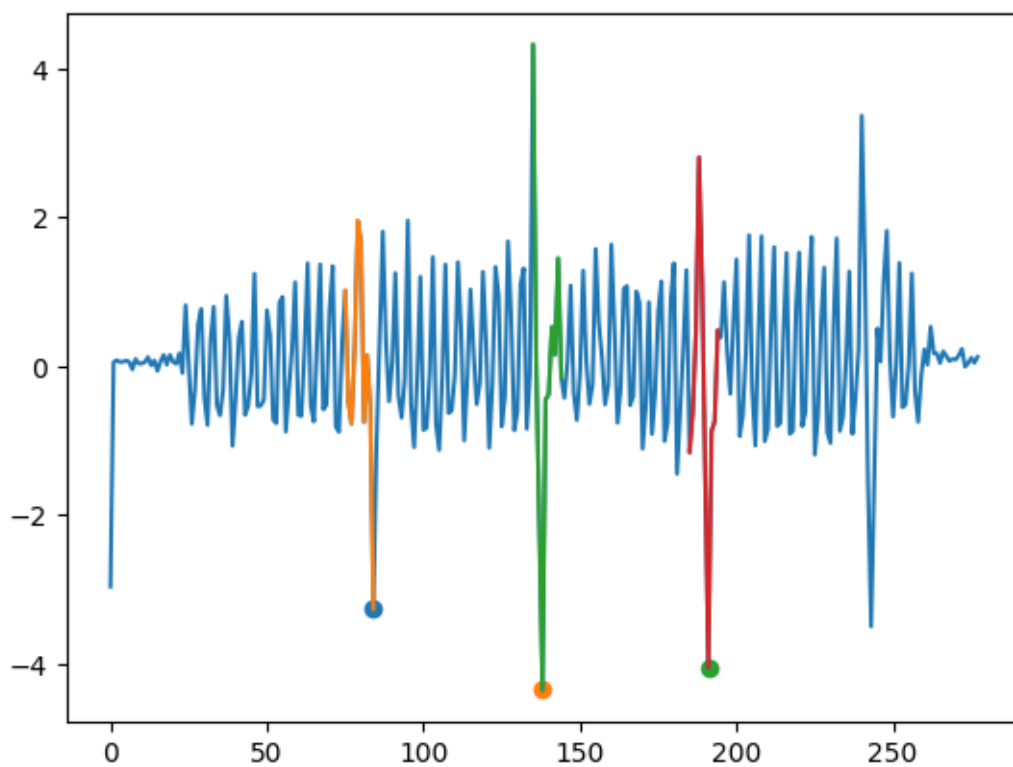
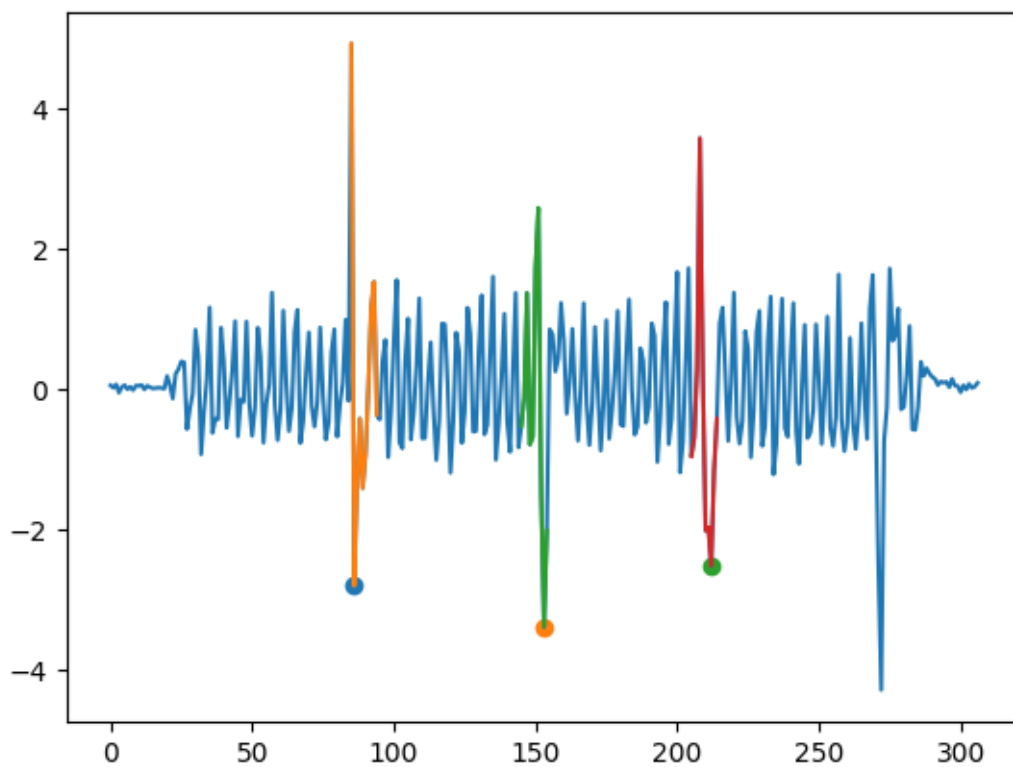
    if (index_experimento >= len(intervalo_min) ):
        break

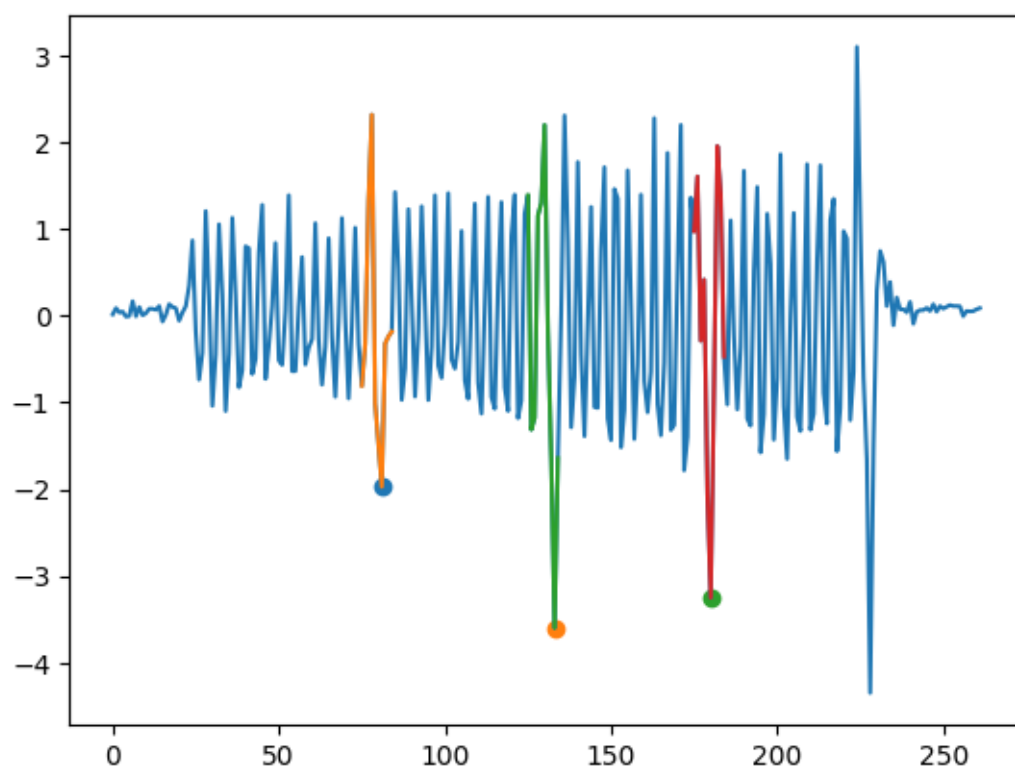
    t_mins = []
    pontos = intervalo_min[index_experimento]
    for ponto in pontos:
        data_p = data[(ponto-c) : (ponto+c)]
        ponto_min = np.argmin(data_p, axis=-1) + ponto - c
        t_mins.append(ponto_min)
        plt.plot(data_p, label = 'raw')
        plt.scatter(ponto_min, data[ponto_min])

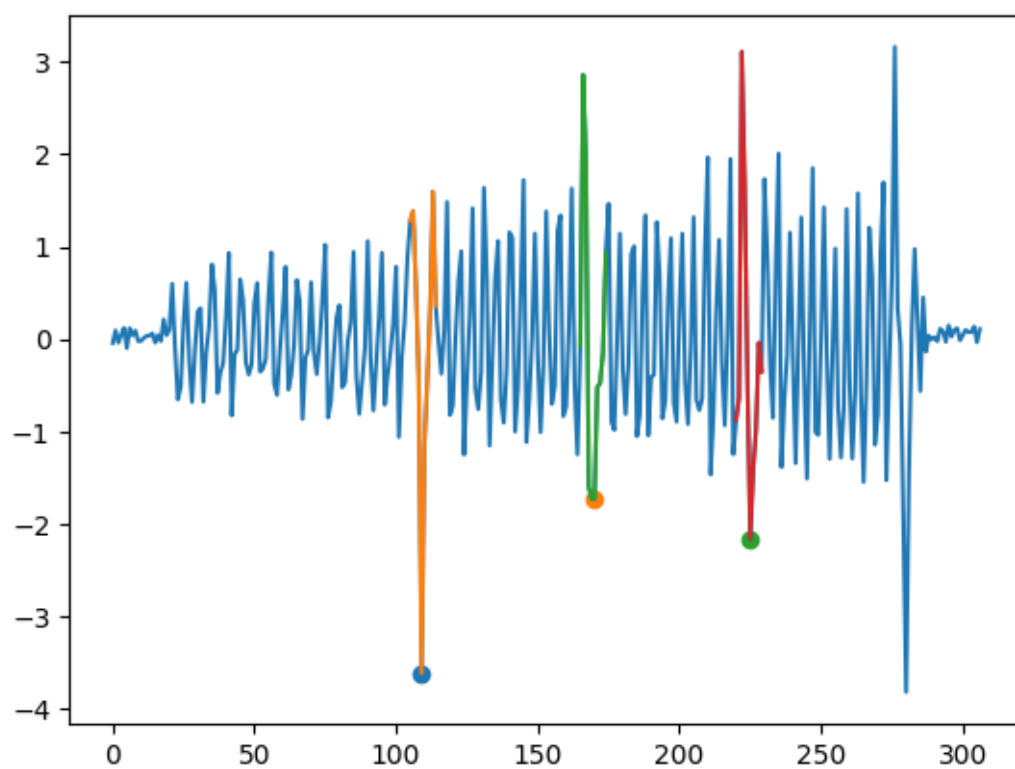
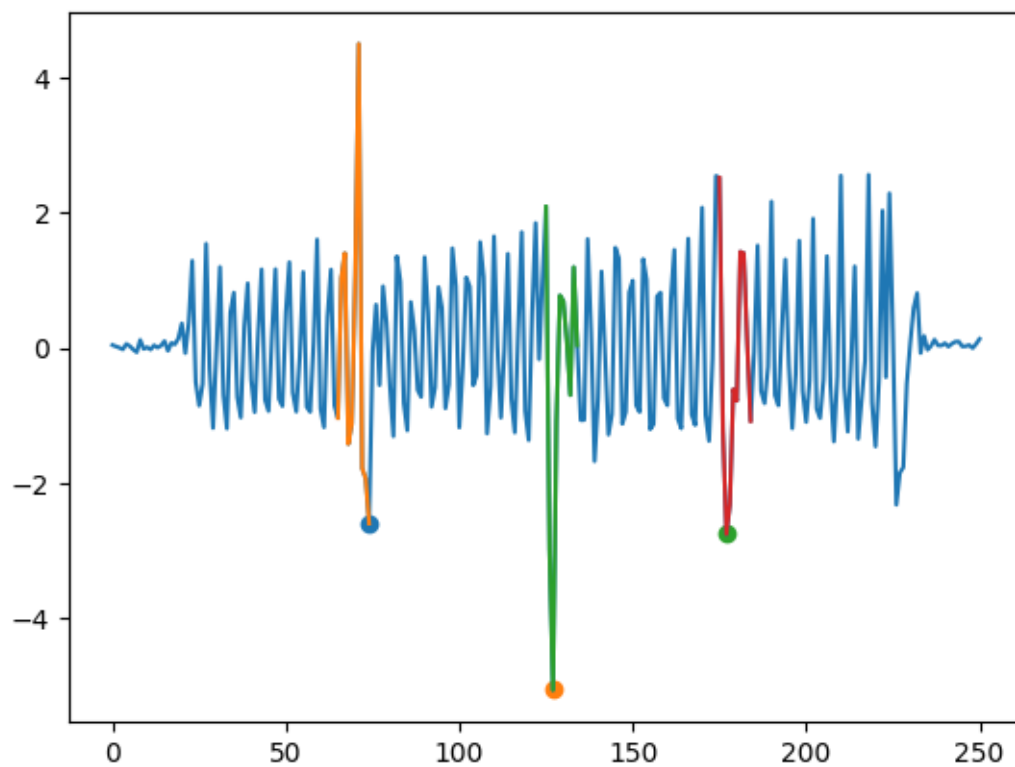
    pontos_mins.append(t_mins)
    index_experimento += 1

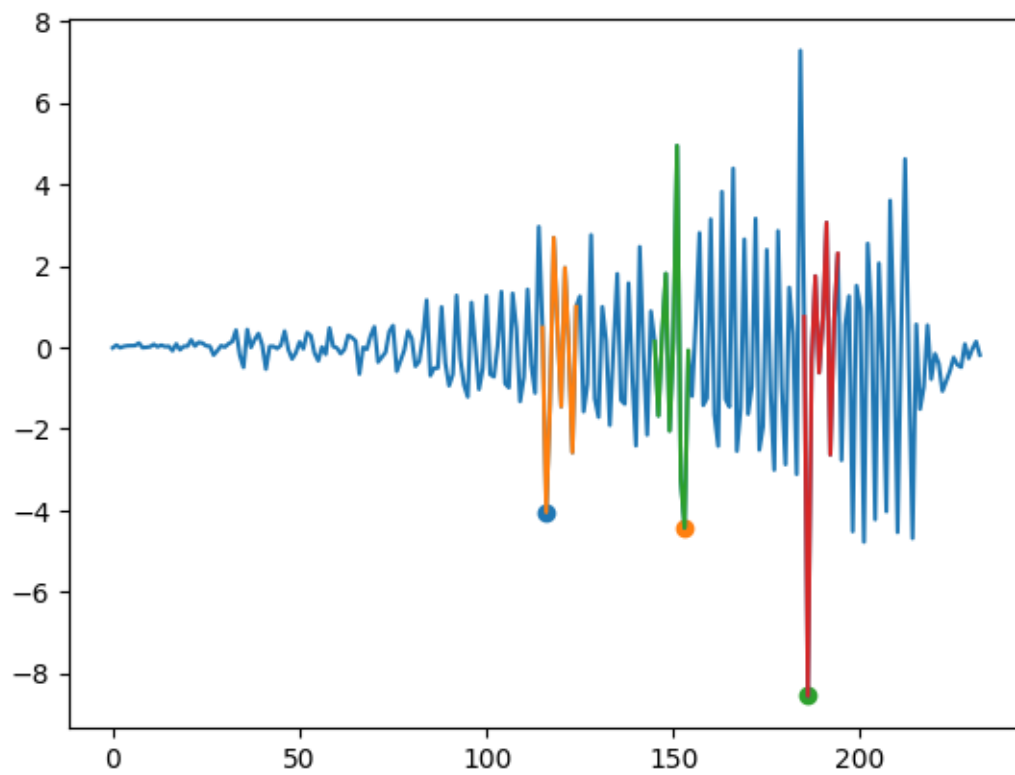
```

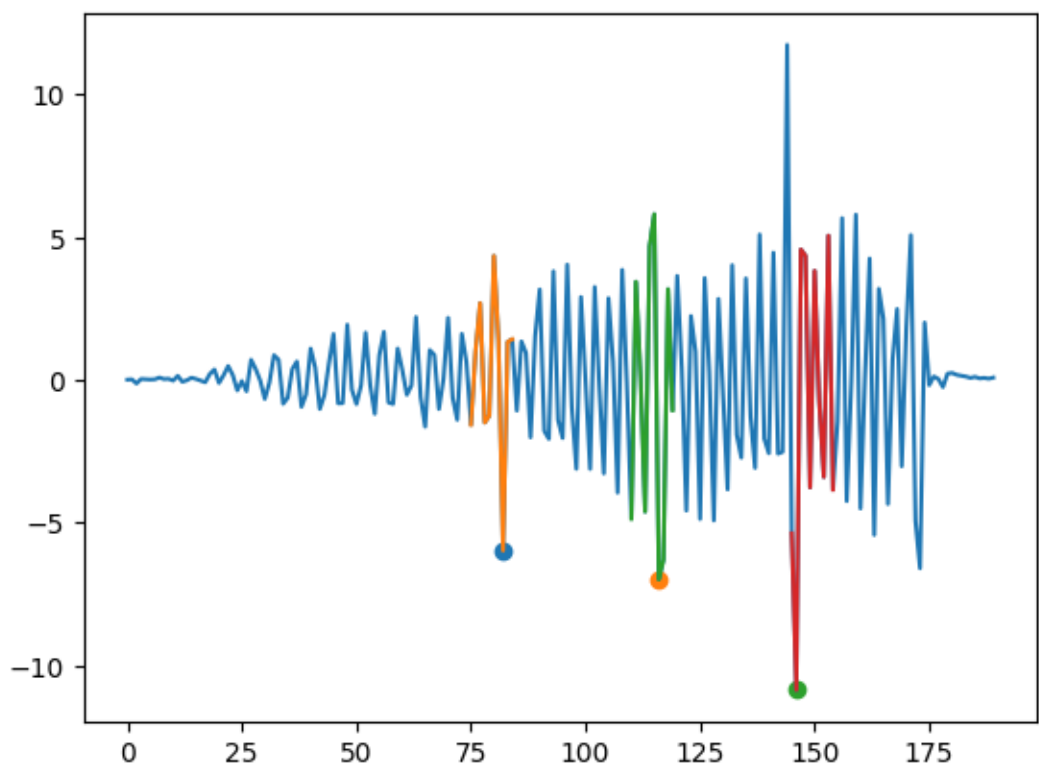
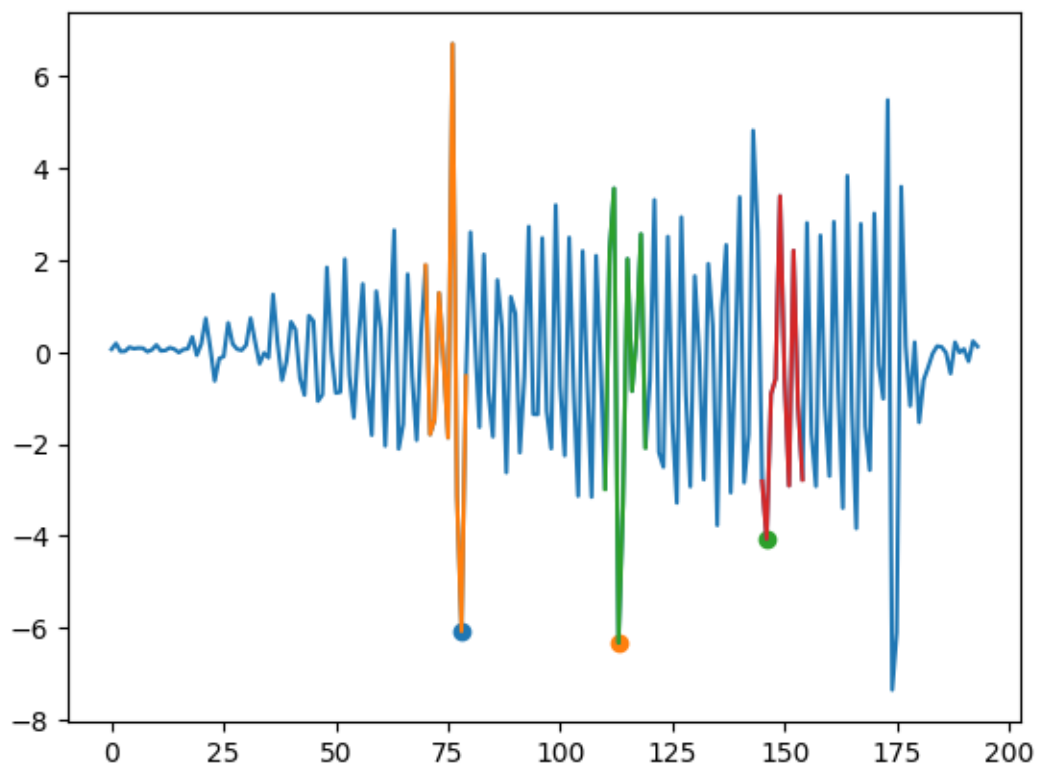
Tempos

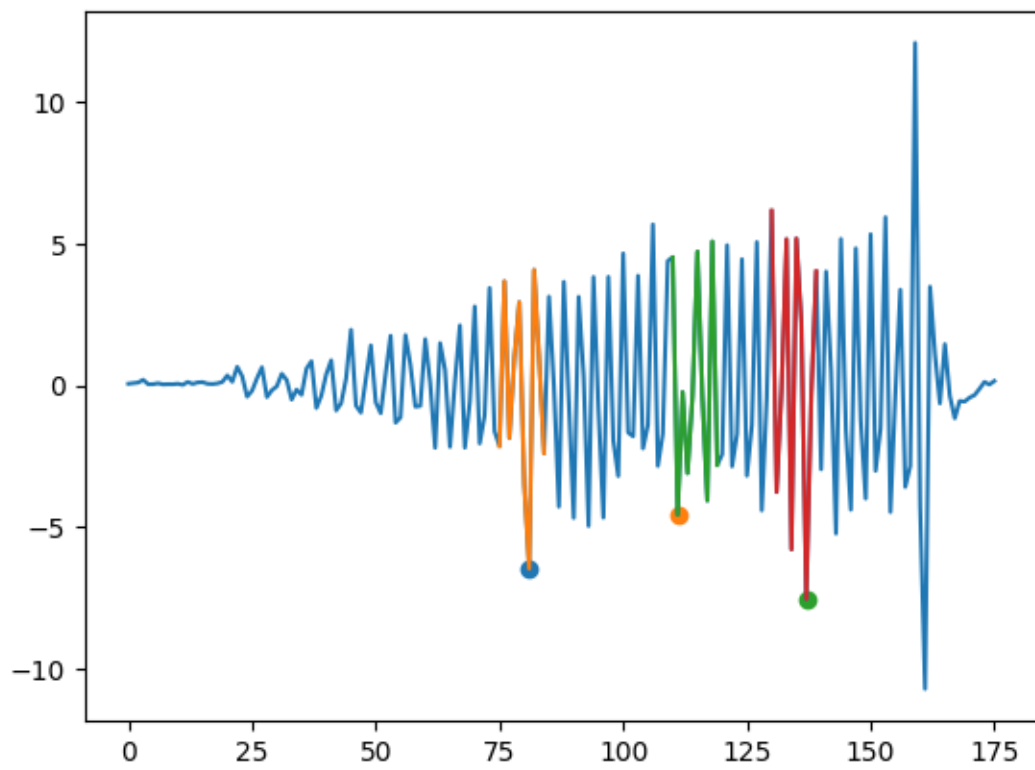


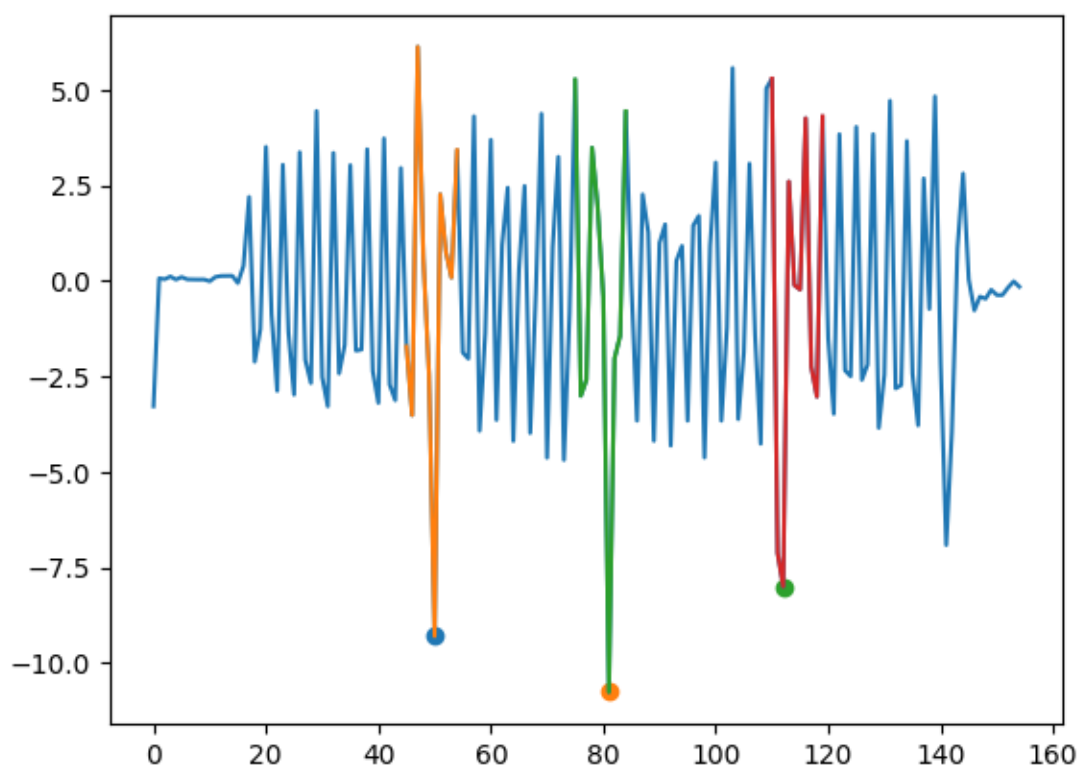
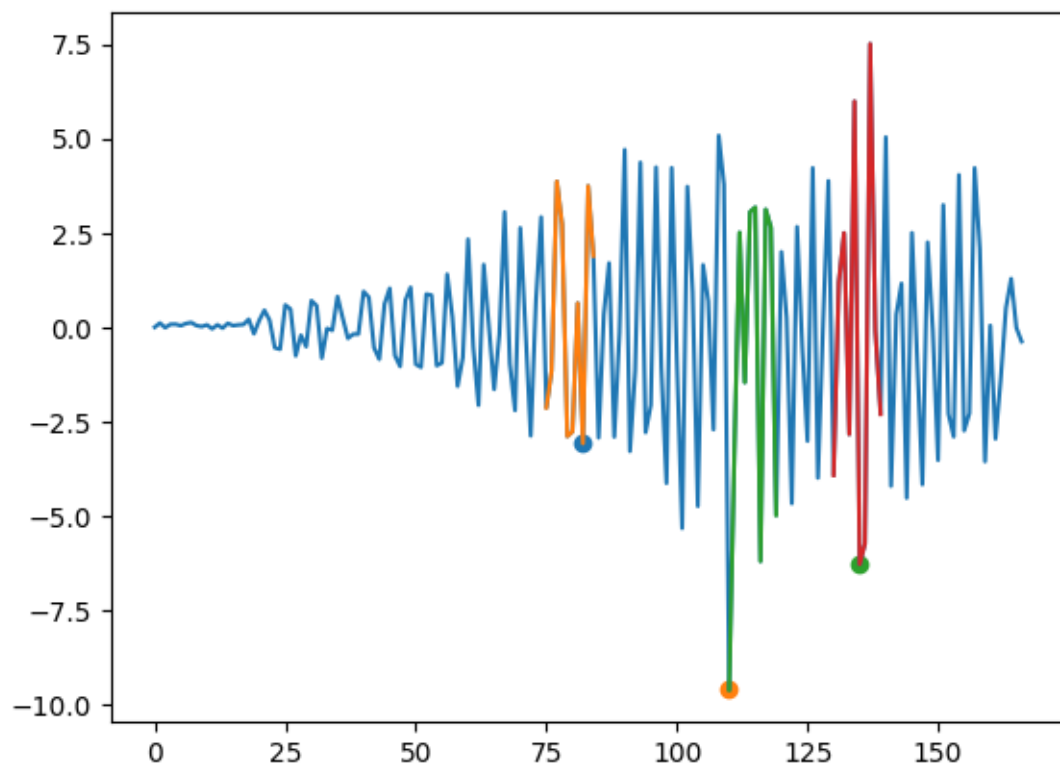


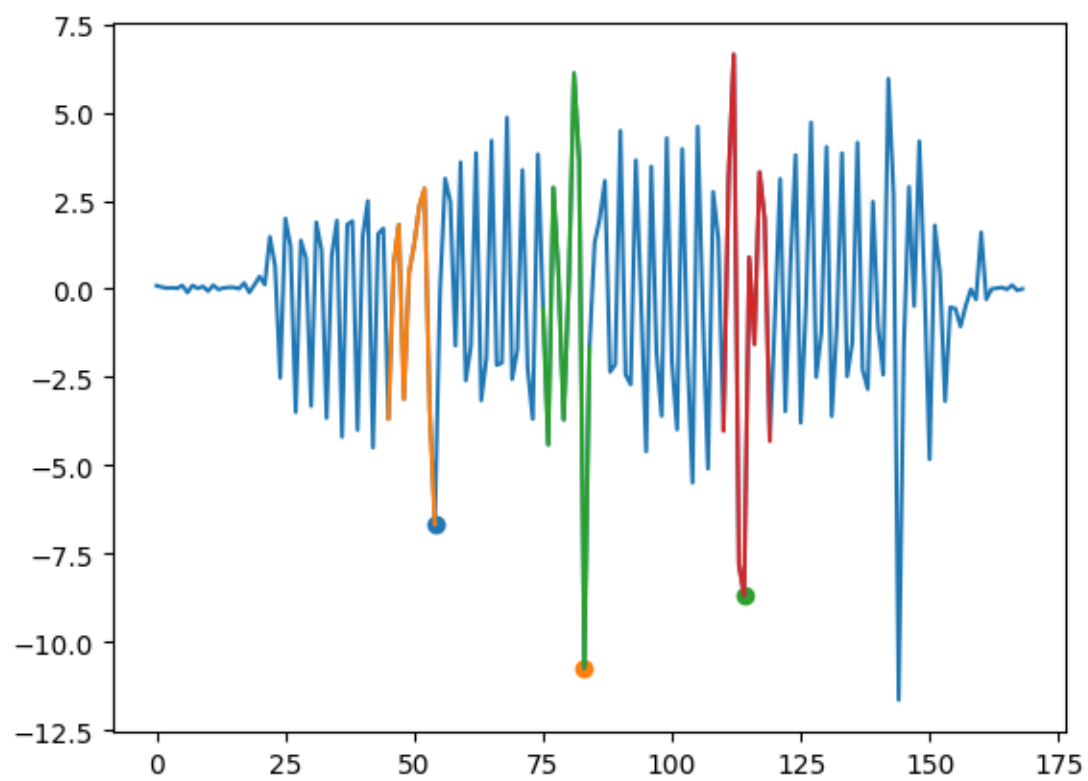


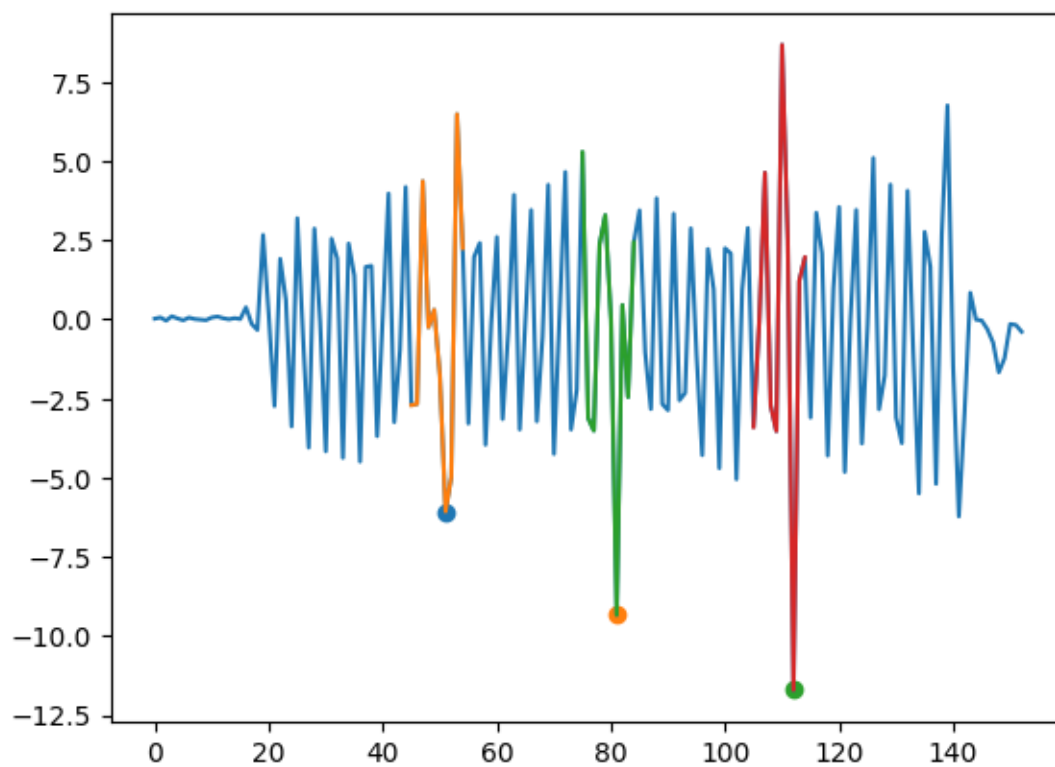
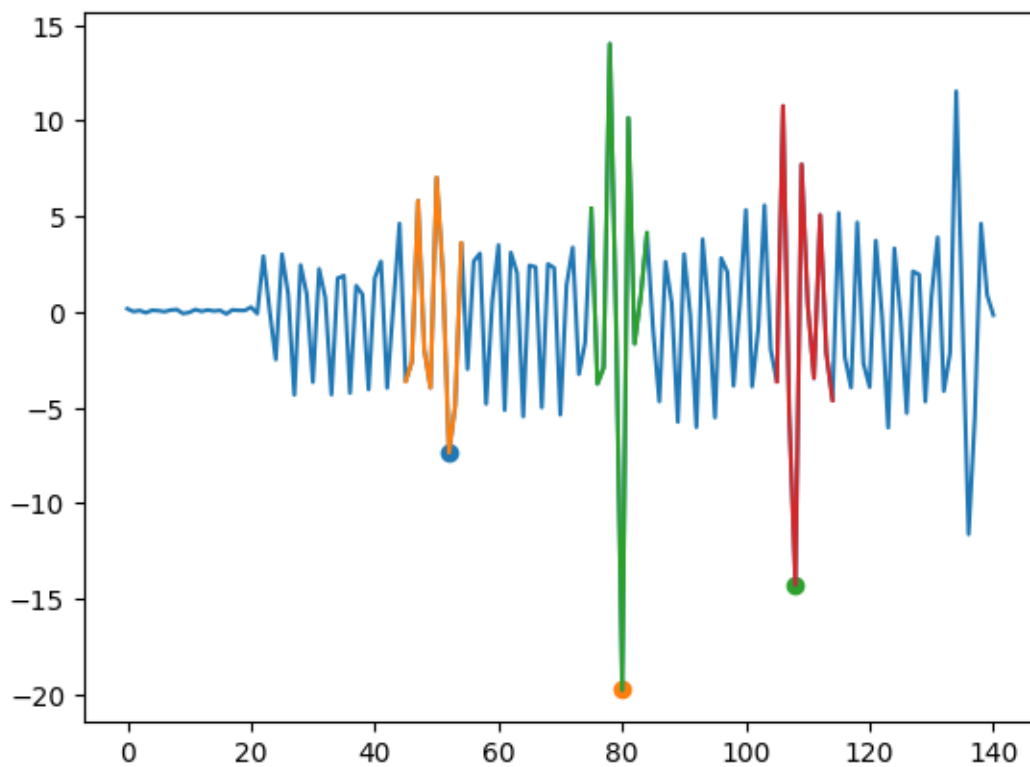


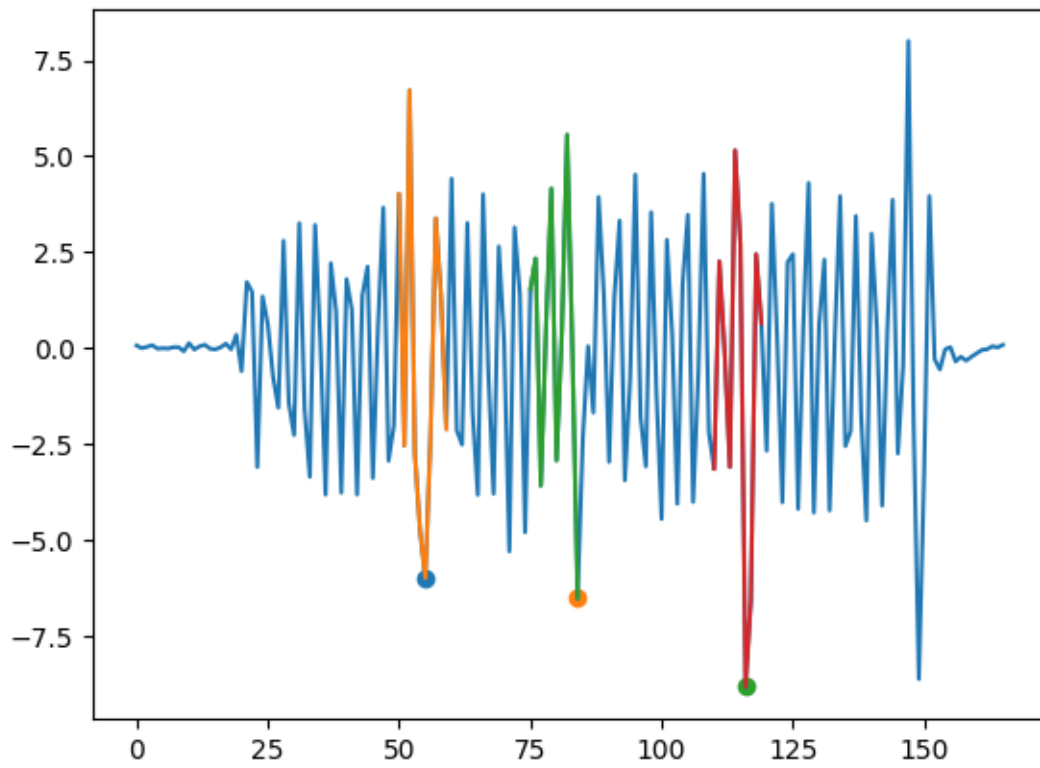


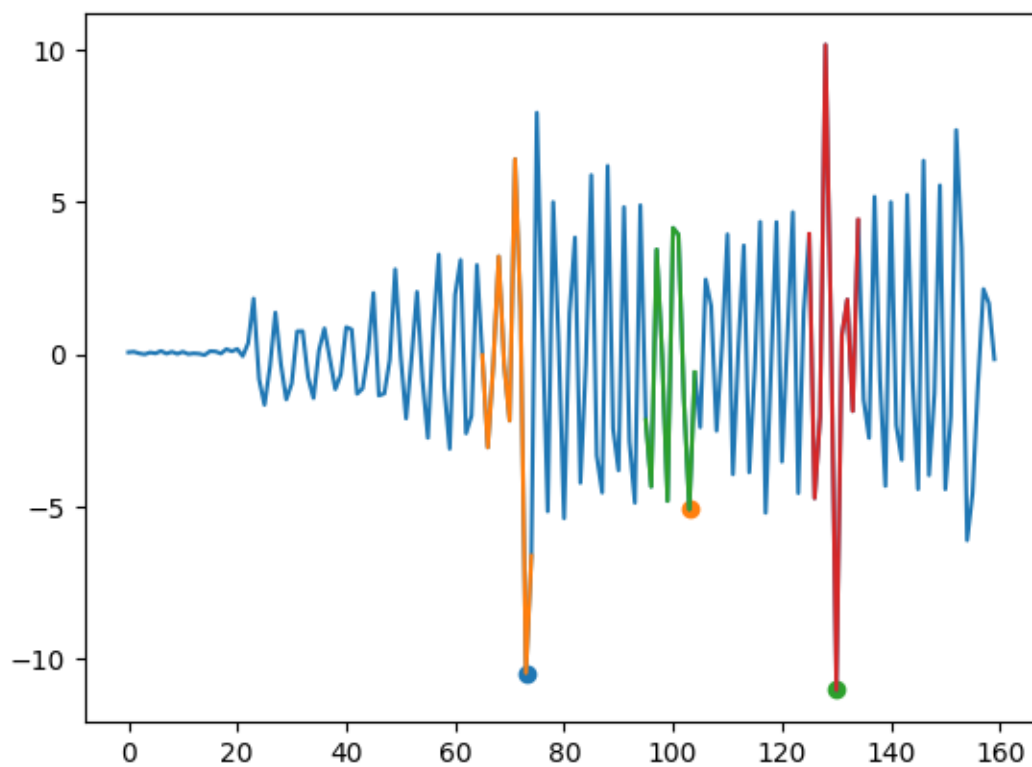
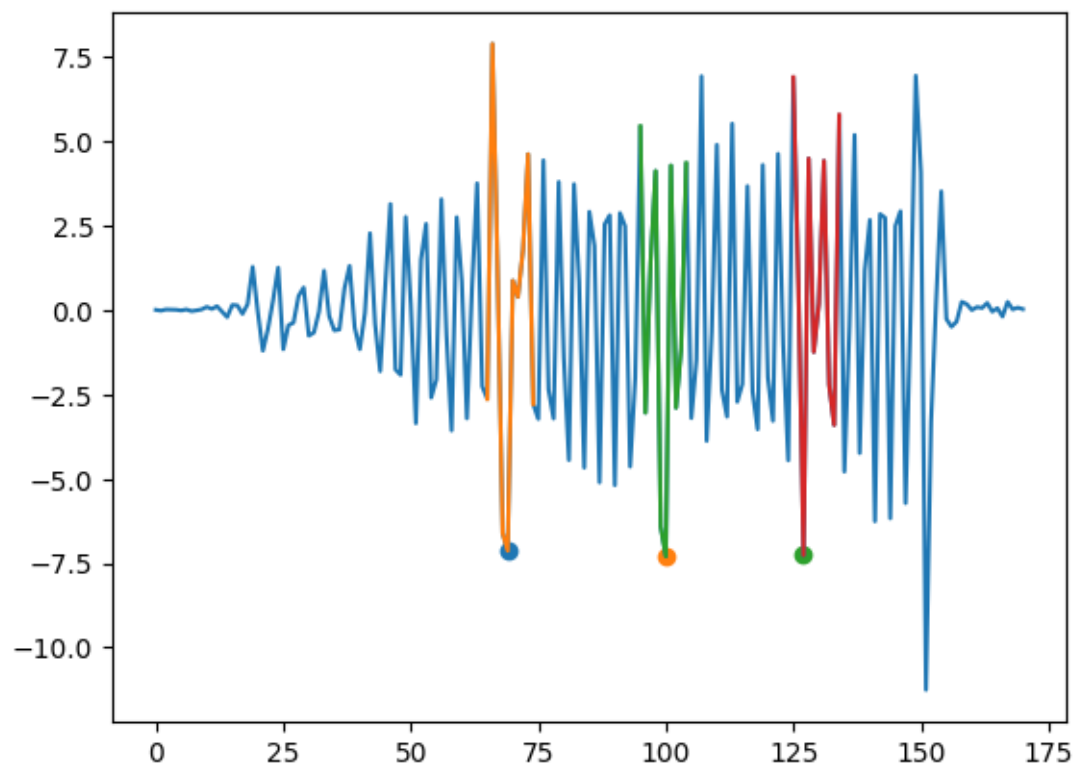


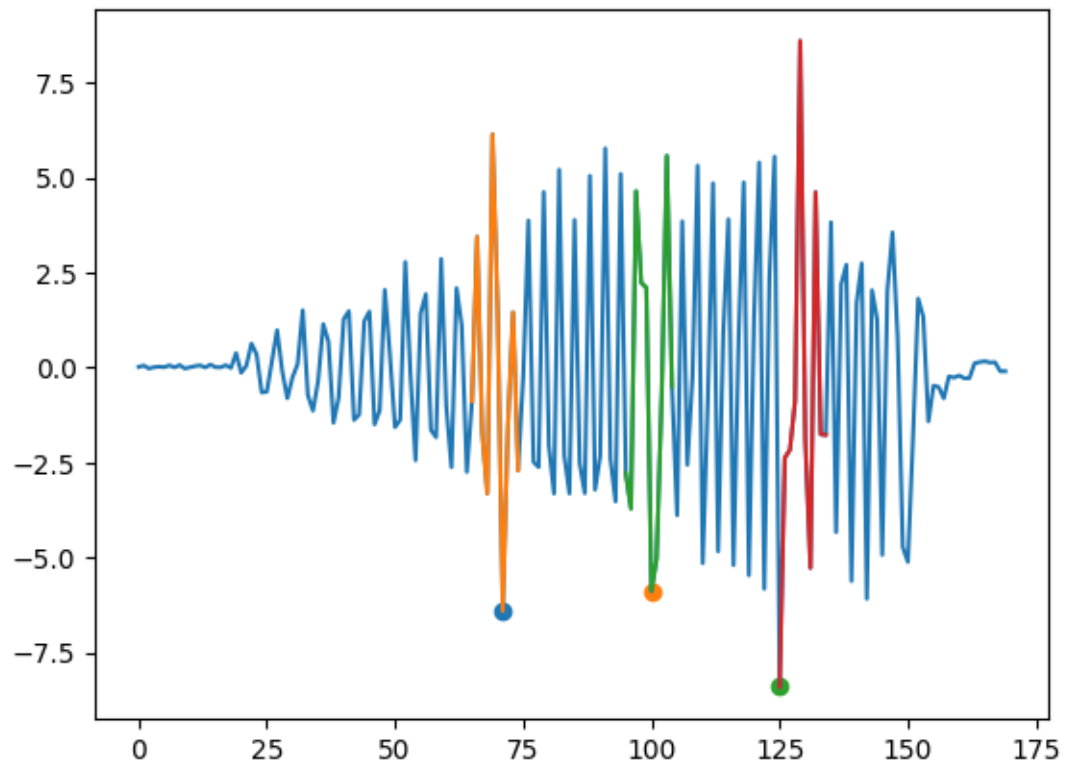


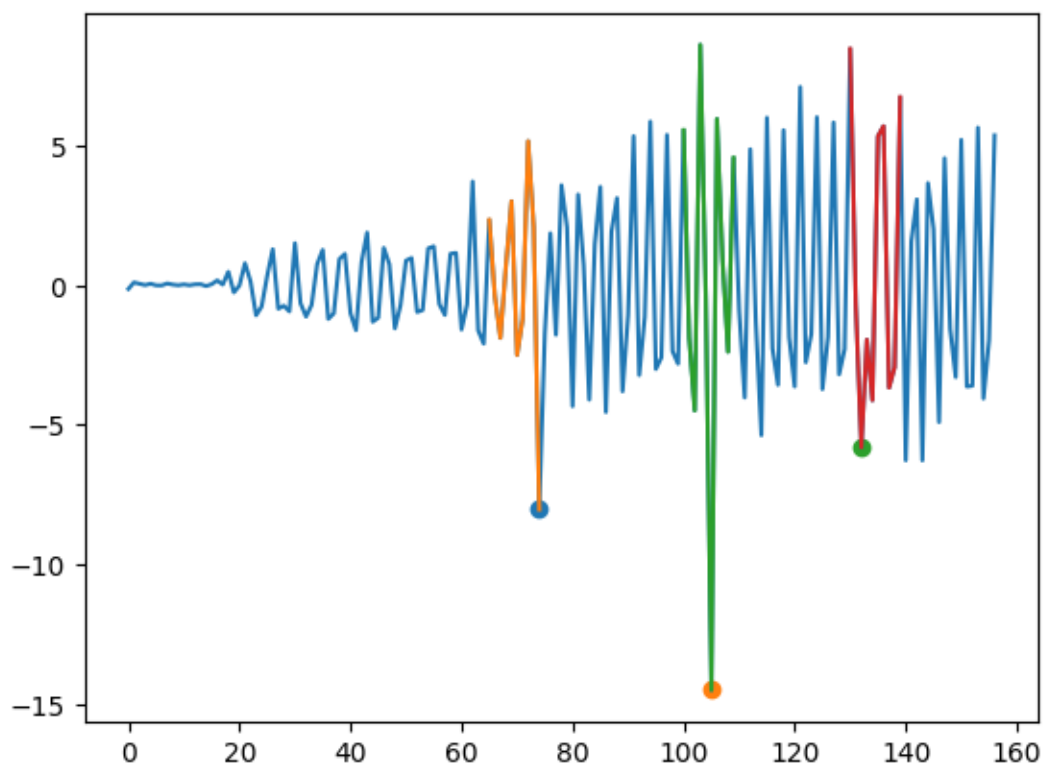
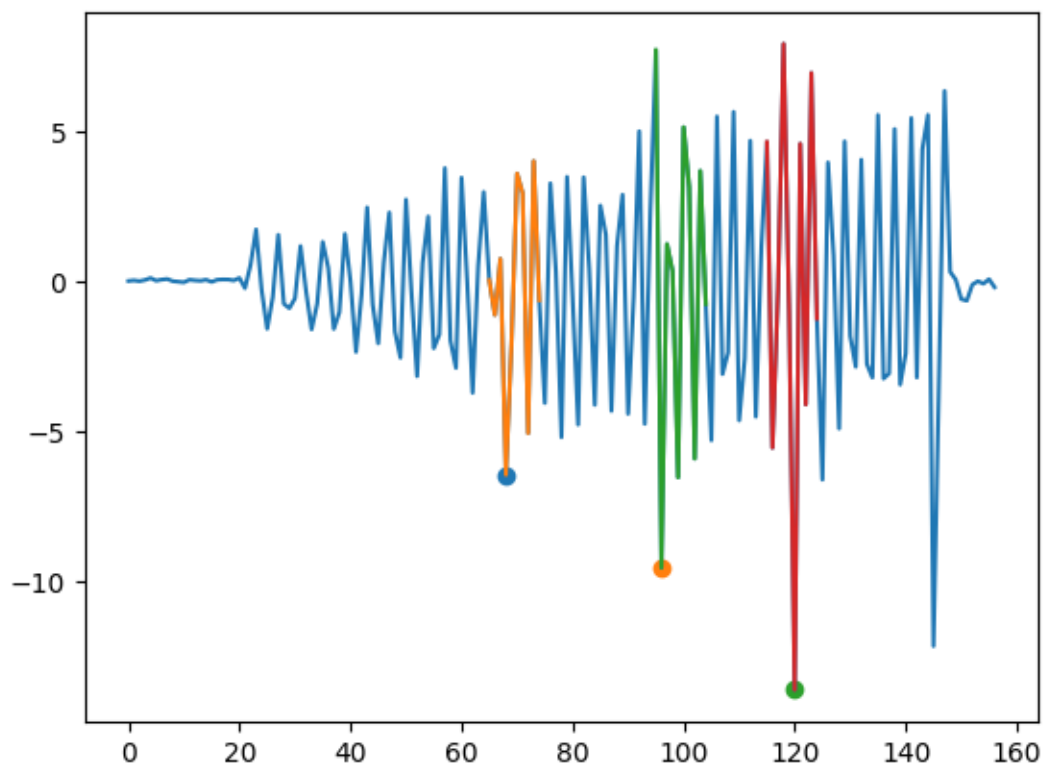












```

[8]: # Plot aceleracao por experimento (MU)

index_experimento = 0
for experimento in dados_experimentos[0:4]:

    t = experimento['Time (s)']
    z = experimento['z']

    inicio = 0
    intervalo = 1
    novo_t = [0]
    novo_v = [0]

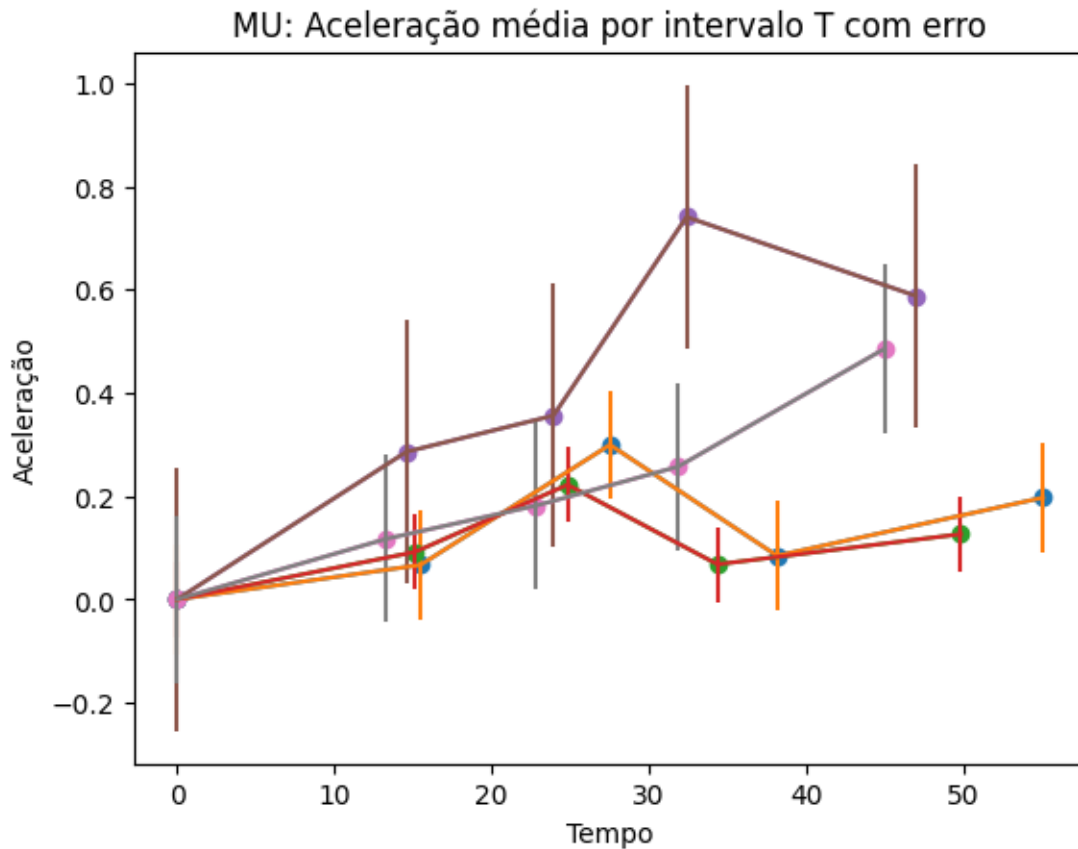
    aMedia = np.mean(z)

    for ponto in pontos_mins[index_experimento]:
        data_p = np.array(z[inicio : ponto])
        t_p = np.array(t[inicio : ponto])
        novo_t.append(t[ponto])
        novo_v.append(np.mean(data_p))
        inicio = ponto
        intervalo += 1
    ponto = t.size
    data_p = np.array(z[inicio : ponto])
    t_p = np.array(t[inicio : ponto])
    novo_t.append(t[ponto-1])
    novo_v.append(np.mean(data_p))
    inicio = ponto

    plt.title("MU: Aceleração média por intervalo T com erro")
    plt.plot(novo_t, novo_v, marker="o", linestyle='-')
    plt.errorbar(novo_t, novo_v, yerr=np.std(novo_v))
    plt.ylabel("Aceleração")
    plt.xlabel("Tempo")

    index_experimento += 1
    color_index = 0

```

```
[9]: # Plot aceleracao por experimento (MUA)

index_experimento = 5
for experimento in dados_experimentos[5:9]:

    t = experimento['Time (s)']
    z = experimento['z']

    inicio = 0
    intervalo = 1

    novo_t = [0]
    novo_v = [0]

    aMedia = np.mean(z)

    for ponto in pontos_mins[index_experimento]:
        data_p = np.array(z[inicio : ponto])
        t_p = np.array(t[inicio : ponto])
        novo_t.append(t[ponto])
```

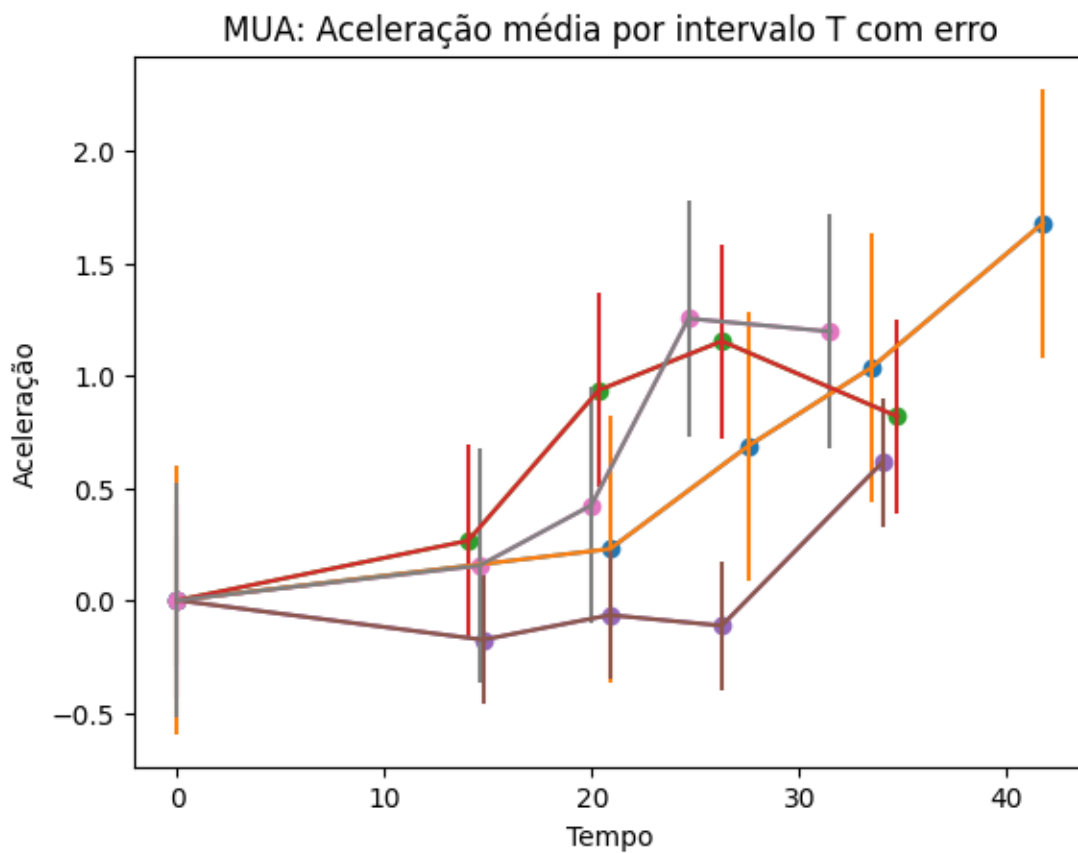
```

    novo_v.append(np.mean(data_p))
    inicio = ponto
    intervalo += 1
ponto = t.size
data_p = np.array(z[inicio : ponto])
t_p = np.array(t[inicio : ponto])
novo_t.append(t[ponto-1])
novo_v.append(np.mean(data_p))
inicio = ponto

plt.title("MUA: Aceleração média por intervalo T com erro")
plt.plot(novo_t, novo_v, marker="o", linestyle='-')
plt.errorbar(novo_t, novo_v, yerr=np.std(novo_v))
plt.ylabel("Aceleração")
plt.xlabel("Tempo")

index_experimento += 1
color_index = 0

```



```

[10]: # Plot velocidade por experimento (MU)

index_experimento = 0
for experimento in dados_experimentos[0:4]:

    t = experimento['Time (s)']
    z = experimento['z']
    v = vetor_velocidade (z, t)

    inicio = 0
    intervalo = 1
    novo_t = [0]
    novo_v = [0]

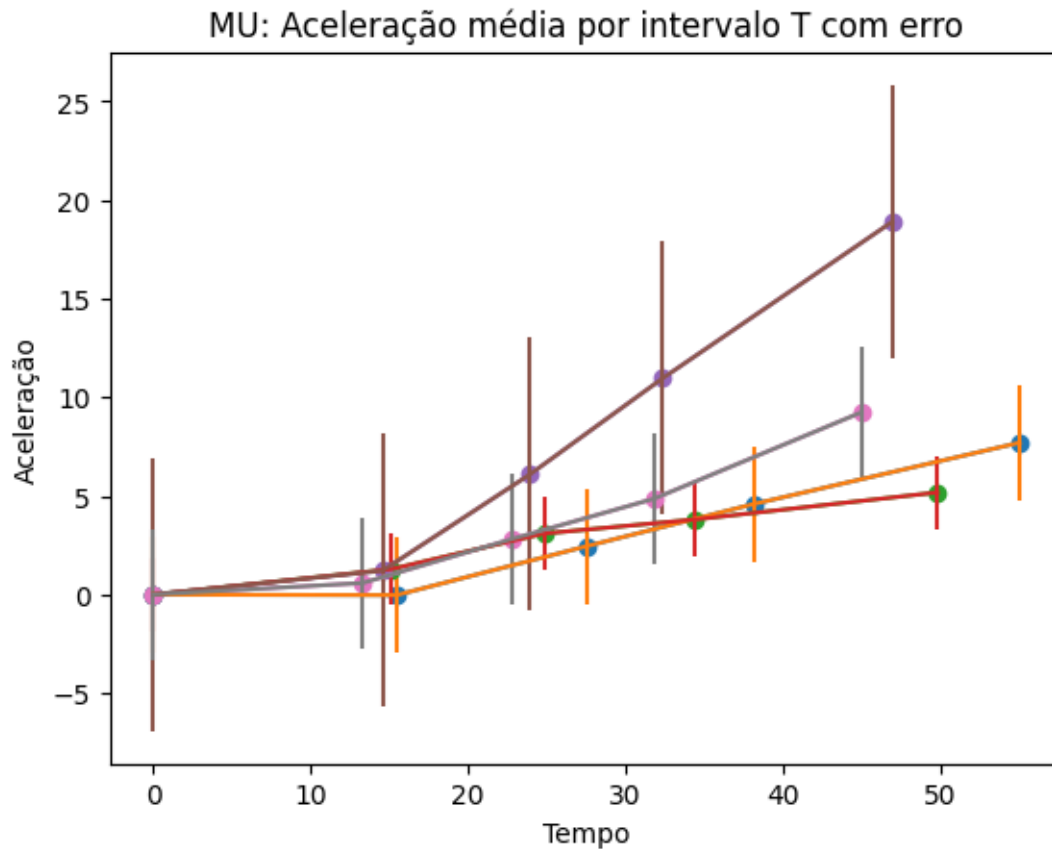
    vMedia = np.mean(v)

    for ponto in pontos_mins[index_experimento]:
        data_p = np.array(v[inicio : ponto])
        t_p = np.array(t[inicio : ponto])
        novo_t.append(t[ponto])
        novo_v.append(np.mean(data_p))
        inicio = ponto
        intervalo += 1
    ponto = t.size
    data_p = np.array(v[inicio : ponto])
    t_p = np.array(t[inicio : ponto])
    novo_t.append(t[ponto-1])
    novo_v.append(np.mean(data_p))
    inicio = ponto

    plt.title("MU: Aceleração média por intervalo T com erro")
    plt.plot(novo_t, novo_v, marker="o", linestyle='-')
    plt.errorbar(novo_t, novo_v, yerr=np.std(novo_v))
    plt.ylabel("Aceleração")
    plt.xlabel("Tempo")

    index_experimento += 1
    color_index = 0

```



```
[11]: # Plot velocidade por experimento (MUA)

index_experimento = 5
for experimento in dados_experimentos[5:9]:

    t = experimento['Time (s)']
    z = experimento['z']
    v = vetor_velocidade (z, t)

    inicio = 0
    intervalo = 1
    novo_t = [0]
    novo_v = [0]

    vMedia = np.mean(v)

    for ponto in pontos_mins[index_experimento]:
        data_p = np.array(v[inicio : ponto])
        t_p = np.array(t[inicio : ponto])
        novo_t.append(t[ponto])
```

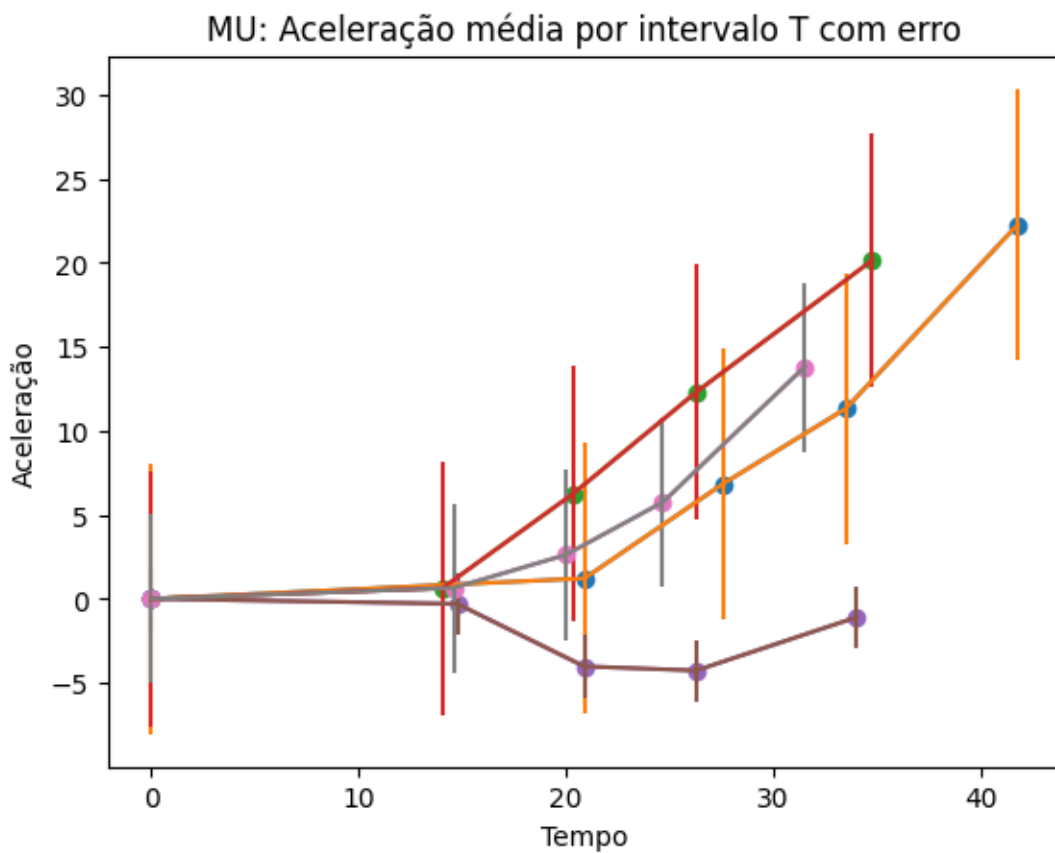
```

    novo_v.append(np.mean(data_p))
    inicio = ponto
    intervalo += 1
ponto = t.size
data_p = np.array(v[inicio : ponto])
t_p = np.array(t[inicio : ponto])
novo_t.append(t[ponto-1])
novo_v.append(np.mean(data_p))
inicio = ponto

plt.title("MU: Aceleração média por intervalo T com erro")
plt.plot(novo_t, novo_v, marker="o", linestyle='-')
plt.errorbar(novo_t, novo_v, yerr=np.std(novo_v))
plt.ylabel("Aceleração")
plt.xlabel("Tempo")

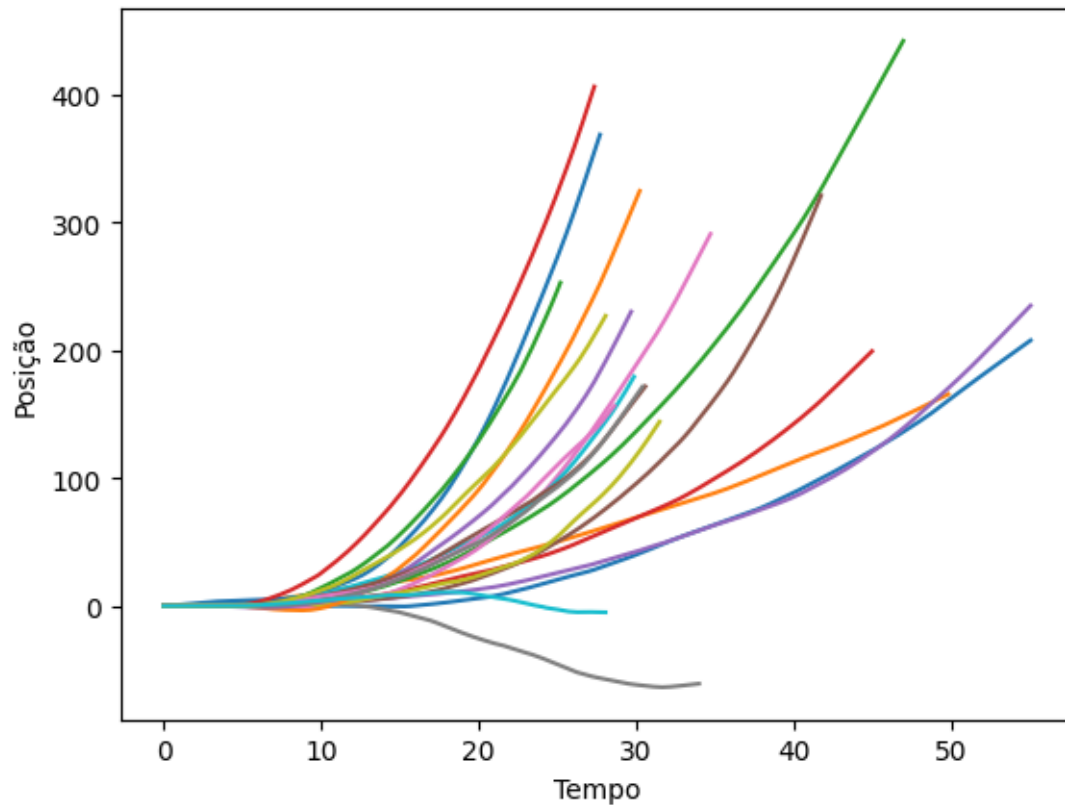
index_experimento += 1
color_index = 0

```



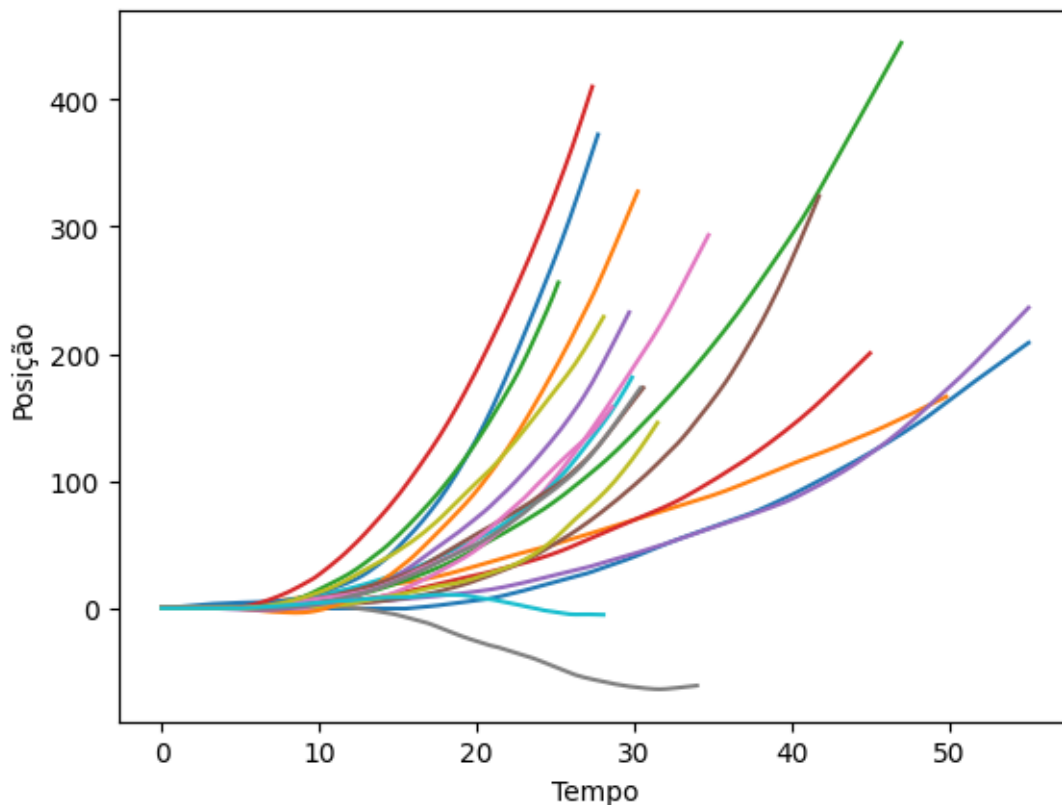
[19]: *# Gráfico do tempo pela posição determinada pelo Método de Euler*

```
for experimento in dados_experimentos:
    t=experimento['Time (s)']
    z = experimento['z']
    posicao = []
    v = vetor_velocidade(z,t)
    posicao = vetor_posicao(v,t)
    plt.plot(t,posicao)
    plt.ylabel("Posição")
    plt.xlabel("Tempo")
    print()
```



```
[20]: # Gráfico do tempo pela posição determinada de forma analítica

for experimento in dados_experimentos:
    t= experimento['Time (s)']
    z = experimento['z']
    v=[]
    v.append(0)
    for i in range(1,len(z)):
        v.append(velocidade_analitica(v[i-1], z[i-1], t[i]-t[i-1]))
    s=[]
    s.append(0)
    for i in range(1,len(z)):
        s.append(posicao_analitica(s[i-1], v[i-1], z[i-1], t[i]-t[i-1]))
    plt.plot(t,s)
    plt.ylabel("Posição")
    plt.xlabel("Tempo")
```



[16]: *# Imprime o notebook*

```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!pip install py pandoc
!cp "drive/My Drive/Colab Notebooks/EP3_MAC0209.ipynb" ./
!jupyter nbconvert --to PDF "EP3_MAC0209.ipynb"
!cp "EP3_MAC0209.pdf" "drive/My Drive/Colab Notebooks/"
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
pandoc is already the newest version (2.9.2.1-3ubuntu2).
texlive is already the newest version (2021.20220204-1).
texlive-latex-extra is already the newest version (2021.20220204-1).
texlive-xetex is already the newest version (2021.20220204-1).
0 upgraded, 0 newly installed, 0 to remove and 45 not upgraded.
Requirement already satisfied: py pandoc in /usr/local/lib/python3.10/dist-
packages (1.13)
[NbConvertApp] Converting notebook EP3_MAC0209.ipynb to PDF
[NbConvertApp] Support files will be in EP3_MAC0209_files/
[NbConvertApp] Making directory ./EP3_MAC0209_files
```


[illegible]

```
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Making directory ./EP3_MAC0209_files
[NbConvertApp] Writing 97122 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 2570471 bytes to EP3_MAC0209.pdf
```

```
[17]: !ls
```

drive EP3_MAC0209.ipynb EP3_MAC0209.pdf sample_data

Planilha de dados obtidos experimentalmente usando o acelerômetro

Tabela de estatísticas (média) de tempos de deslocamentos das realizações

Caso	t0	t1	t2	t3	t4	Comprimento	Velocidade K	Aceleração K
MU 1	3,49	15,62	25,92	35,38	44,96	32	0,79	-
MU 2	3,28	9,43	14,72	20,23	25,60	32	1,44	-
MUA 1	3,72	15,70	21,71	27,00	31,71	32	1,31	0,04
MUA 2	3,42	12,85	18,13	22,89	27,14	32	1,48	0,04

Tabela de estatísticas (desvios padrão) de tempos de deslocamentos das realizações

Caso	t0	t1	t2	t3	t4	Comprimento	Velocidade K	Aceleração K
MU 1	0,27	2,38	3,05	3,69	4,49	32	0,08	-
MU 2	0,47	0,37	0,30	0,53	0,86	32	0,06	-
MUA 1	0,98	2,91	3,29	3,74	4,00	32	0,13	0,01
MUA 2	0,34	0,52	0,56	0,77	0,67	32	0,04	0,00

Tabela de realizações dos experimentos

	MU	MUA	Total
Trajeto 1 Reto	5	5	10
Trajeto 2 ZZ	5	5	10
Total	10	10	20

Tabela de tempos de deslocamentos das realizações

Amostra	t0	t1	t2	t3	t4	Comprimento	Velocidade K	Aceleração K	Movimento	Trajeto
r1	3,42	15,48	27,54	37,8	48,96	32	0,71	-	MU	1
r2	3,78	15,12	24,84	34,37	43,73	32	0,81	-	MU	1
r3	3,6	14,58	23,94	32,4	41,04	32	0,86	-	MU	1
r4	3,6	13,32	22,86	31,83	40,68	32	0,86	-	MU	1
r5	3,06	19,62	30,42	40,5	50,39	32	0,71	-	MU	1
r6	5,4	20,88	27,54	33,48	38,52	32	1,16	0,03	MUV	1
r7	3,06	14,04	20,34	26,28	31,32	32	1,23	0,03	MUV	1
r8	3,1	14,76	20,88	26,27	31,13	32	1,28	0,03	MUV	1
r9	3,78	14,58	19,98	24,66	28,98	32	1,45	0,04	MUV	1

	r10	3,24	14,22	19,8	24,3	28,62	32	1,45	0,04	MUV	1
	r11	2,7	9	14,58	20,16	25,38	32	1,42	-	MU	2
	r12	3,6	9,72	14,94	20,52	25,92	32	1,44	-	MU	2
	r13	3,78	9,36	14,4	19,44	24,48	32	1,55	-	MU	2
	r14	2,88	9,18	14,58	20,16	25,38	32	1,43	-	MU	2
	r15	3,42	9,9	15,12	20,88	26,82	32	1,38	-	MU	2
	r16	3,06	12,42	18	22,86	27,18	32	1,45	0,04	MUV	2
	r17	3,78	13,14	18,54	23,4	27,72	32	1,46	0,04	MUV	2
	r18	3,24	12,96	18,1	23	27	32	1,50	0,05	MUV	2
	r19	3,78	12,24	17,28	21,6	26,09	32	1,54	0,04	MUV	2
	r20	3,24	13,5	18,72	23,57	27,71	32	1,47	0,05	MUV	2