In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

from os import path

import scipy.signal as signal
```

Variáveis da análise

In [ ]:
```
datasetFolder = "./dataset/picked"
ecg = "1003574.txt"
```

Carregando a base de dados

In [ ]:
```
header = [
    "amostra",
    "lead I",
    "lead II",
    "lead  II",
    "aVR",
    "aVL",
    "aVF",
    "V1",
    "V2",
    "V3",
    "V4",
    "V5",
    "V6"
]

dataset = pd.read_csv(
    path.join(datasetFolder, ecg)
)

dataset.columns = header

print(dataset.head())
print(dataset.shape)
```

```
   amostra  lead I  lead II  lead  II  aVR  aVL  aVF  V1   V2     V3   V4
\
0        1    -256     -154       102  205  -26 -179  70 -124 -16000  100
1        2    -252     -151       101  201  -25 -176  68 -120 -16000  110
2        3    -246     -148        98  197  -25 -172  87 -116 -16000  122
3        4    -236     -144        92  190  -26 -164  91 -114 -16000  126
4        5    -231     -140        91  185  -24 -161  70 -114 -16000  122

    V5  V6
0   99  64
1  104  72
2  118  83
3  114  78
4   99  67
(4999, 13)
```

Características do Dataset

In [ ]:
```
samplingFrequency = 500
samplingPeriod = 1 / samplingFrequency
nyquistFrequency = samplingFrequency / 2
```

Pré-processamento

In [ ]:
```
times = np.arange(
    0,
    dataset.shape[0] / samplingFrequency,
    samplingPeriod
)

frequencies = np.fft.fftfreq(dataset.shape[0], samplingPeriod)

dataset["Tempo"] = times
dataset["Frequencia"] = frequencies
```

Filtros

In [ ]:
```
def lowPassFilter(dataset, derivationName, cutoff):
    derivation = dataset[derivationName]
    order = 2

    normalCutoff = cutoff / nyquistFrequency

    b, a = signal.butter(order, normalCutoff, btype="low")

    derivationFiltred = signal.filtfilt(b, a, derivation)

    return derivationFiltred

def highPassFilter(dataset, derivationName, cutoff):
    derivation = dataset[derivationName]
    order = 2

    normalCutoff = cutoff / nyquistFrequency

    b, a = signal.butter(order, normalCutoff, btype="high")

    derivationFiltred = signal.filtfilt(b, a, derivation)

    return derivationFiltred


def notchFilter(dataset, derivationName, cutoff):
    derivation = dataset[derivationName]
    normalCutoff = cutoff / nyquistFrequency

    b, a = signal.iirnotch(normalCutoff, 60, samplingFrequency)

    derivationFiltred = signal.lfilter(b, a, derivation)

    return derivationFiltred
```

Derivadas

In [ ]:
```
def firstDerivate(dataset, derivationName):
    times = dataset["Tempo"]
```

```
    deltaTime = times[1] - times[0]

    derivation = dataset[derivationName]
    deltaDerivation = np.diff(derivation)

    return deltaDerivation / deltaTime

def secondDerivate(dataset, derivationName):
    times = dataset["Tempo"]
    deltaTime = times[1] - times[0]

    firstDerivative = firstDerivate(dataset, derivationName)
    deltaFirstDerivative = np.diff(firstDerivative)

    return deltaFirstDerivative / deltaTime
```

Subamostragem

In [ ]:
```
def subSampling(dataset, derivationName, factor):
    derivation = dataset[derivationName]
    times = dataset["Tempo"]

    return derivation[::factor], times[::factor]
```

Funções de Plot

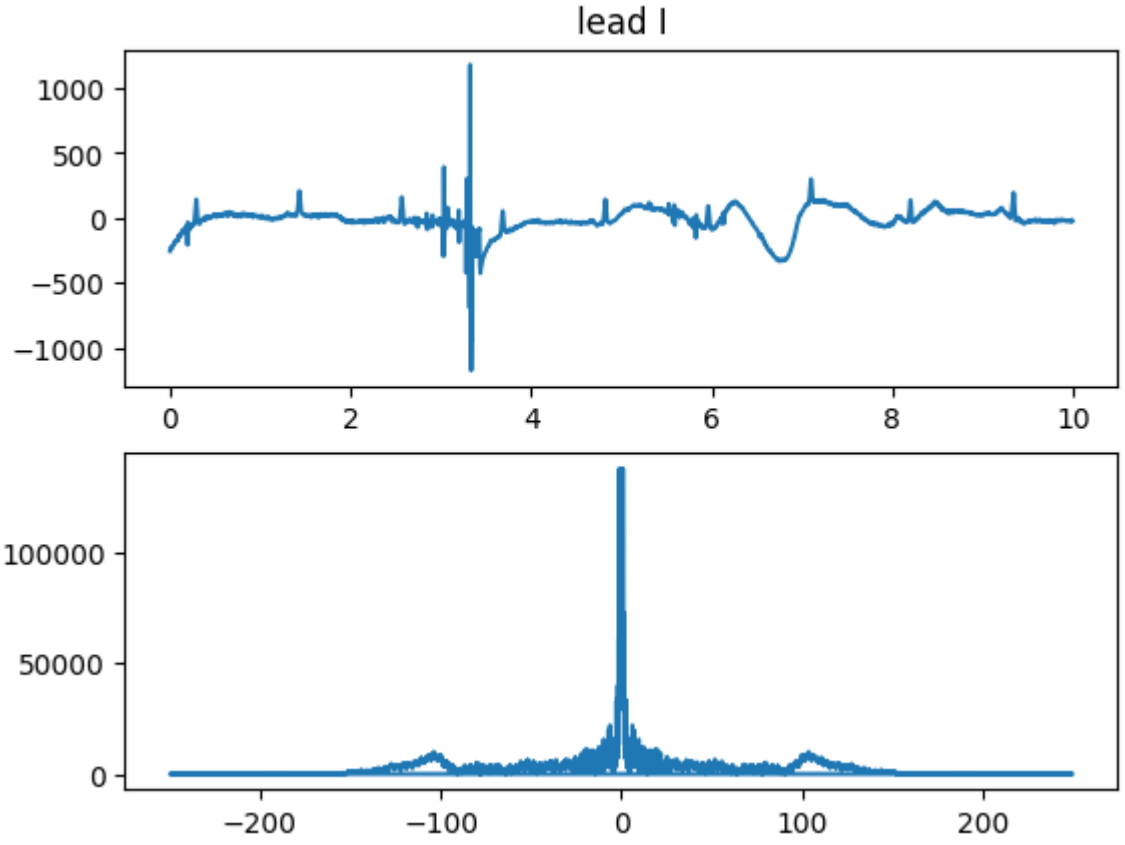
In [ ]:
```
def derivationFourierPlot(dataset, derivationName):
    times = dataset["Tempo"]
    frequencies = dataset["Frequencia"]
    derivation = dataset[derivationName]

    derivationFourier = np.fft.fft(derivation)
    derivationFourier = np.abs(derivationFourier)

    plt.figure()

    plt.subplot(2, 1, 1)
    plt.plot(times, derivation)

    plt.title(derivationName)

    plt.subplot(2, 1, 2)
    plt.plot(frequencies, derivationFourier)

def derivationLowPassPlot(dataset, derivationName, cutoff):
    times = dataset["Tempo"]
    derivation = dataset[derivationName]

    derivationFiltred = lowPassFilter(dataset, derivationName, cutoff)

    plt.figure()

    plt.subplot(2, 1, 1)
    plt.plot(times, derivation)

    plt.title(derivationName)

    plt.subplot(2, 1, 2)
    plt.plot(times, derivationFiltred)
```

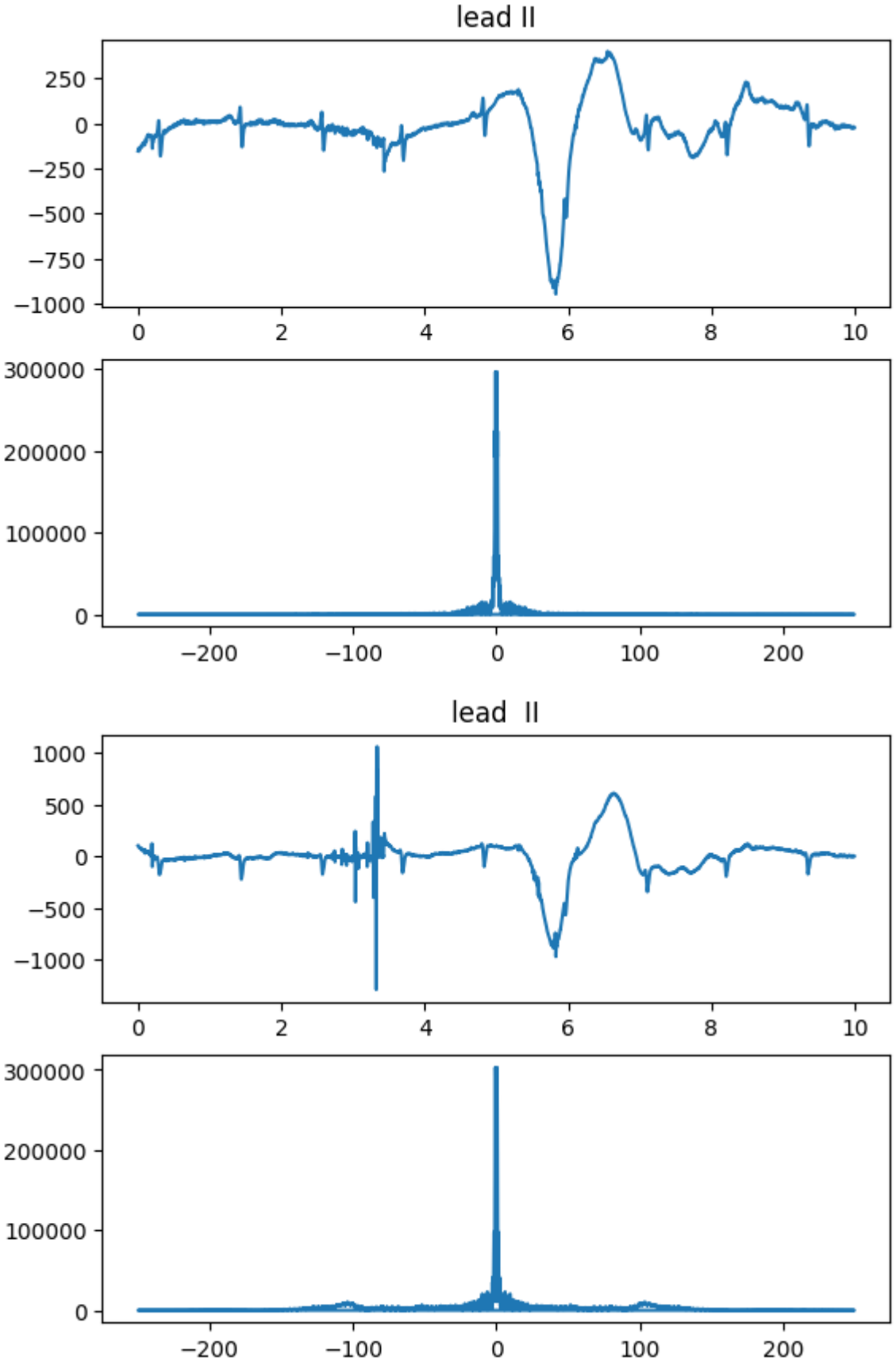
```
def derivationHighPassPlot(dataset, derivationName, cutoff):
    times = dataset["Tempo"]
    derivation = dataset[derivationName]

    derivationFiltred = highPassFilter(dataset, derivationName, cutoff)

    plt.figure()

    plt.subplot(2, 1, 1)
    plt.plot(times, derivation)

    plt.title(derivationName)

    plt.subplot(2, 1, 2)
    plt.plot(times, derivationFiltred)

def derivationNocthPlot(dataset, derivationName, cutoff):
    times = dataset["Tempo"]
    derivation = dataset[derivationName]
    derivationFiltred = notchFilter(dataset, derivationName, cutoff)

    plt.figure()

    plt.subplot(2, 1, 1)
    plt.plot(times, derivation)

    plt.title(derivationName)

    plt.subplot(2, 1, 2)
    plt.plot(times, derivationFiltred)

def derivationDerivatesPlot(dataset, derivationName):
    times = dataset["Tempo"]
    derivation = dataset[derivationName]

    firstDerivateDerivation = firstDerivate(dataset, derivationName)
    secondDerivateDerivation = secondDerivate(dataset, derivationName)

    _, axes = plt.subplots(3, 1, sharex = True)

    axes[0].set_title(derivationName)
    axes[0].plot(times, derivation)

    axes[1].set_title("First Derivate")
    axes[1].plot(times[:-1], firstDerivateDerivation)

    axes[2].set_title("Second Derivate")
    axes[2].plot(times[:-2], secondDerivateDerivation)

    plt.tight_layout()
    plt.show()


def subSamplingPlot(dataset, derivationName, factor):
    times = dataset["Tempo"]
    derivation = dataset[derivationName]

    [
        derivationSubSampling,
```

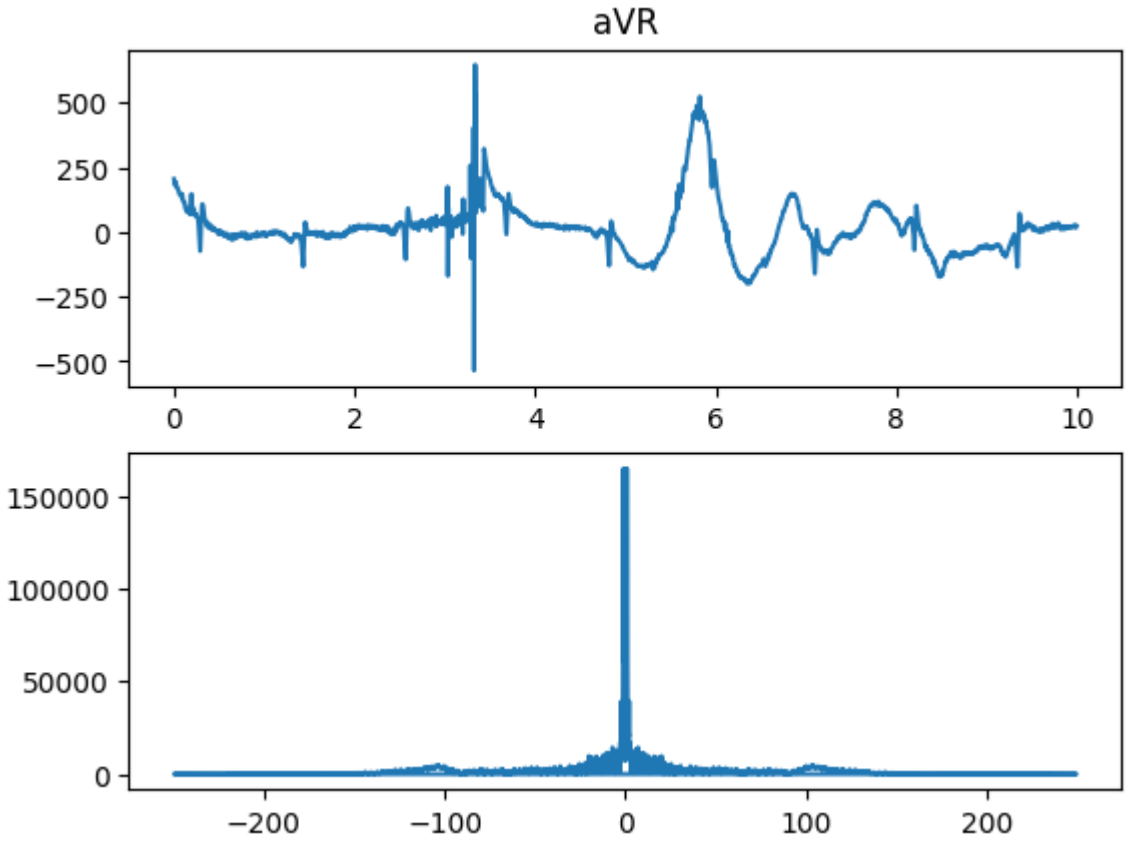
```
        timesSubSampling
    ] = subSampling(dataset, derivationName, factor)

    plt.figure()

    plt.subplot(2, 1, 1)
    plt.plot(times, derivation)

    plt.title(derivationName)

    plt.subplot(2, 1, 2)
    plt.plot(timesSubSampling, derivationSubSampling)
```

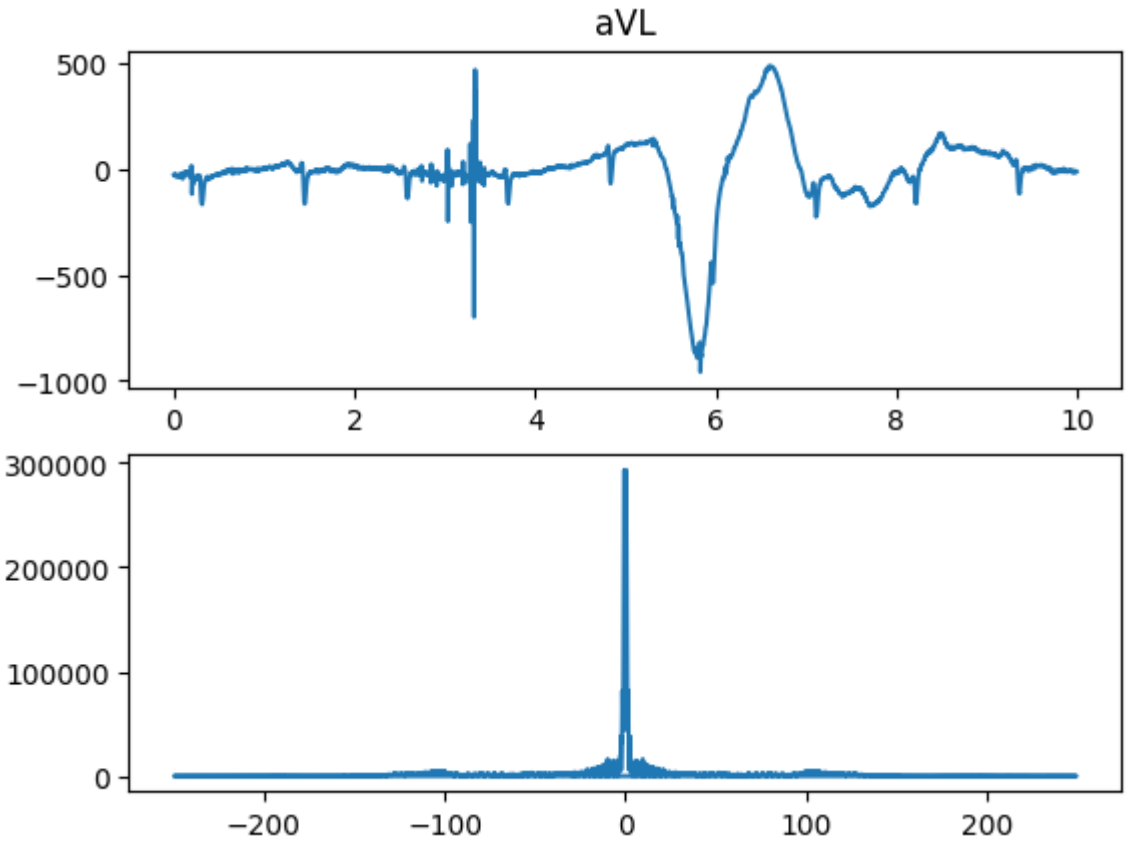
Derivações no domínio do Tempo e da Frequência

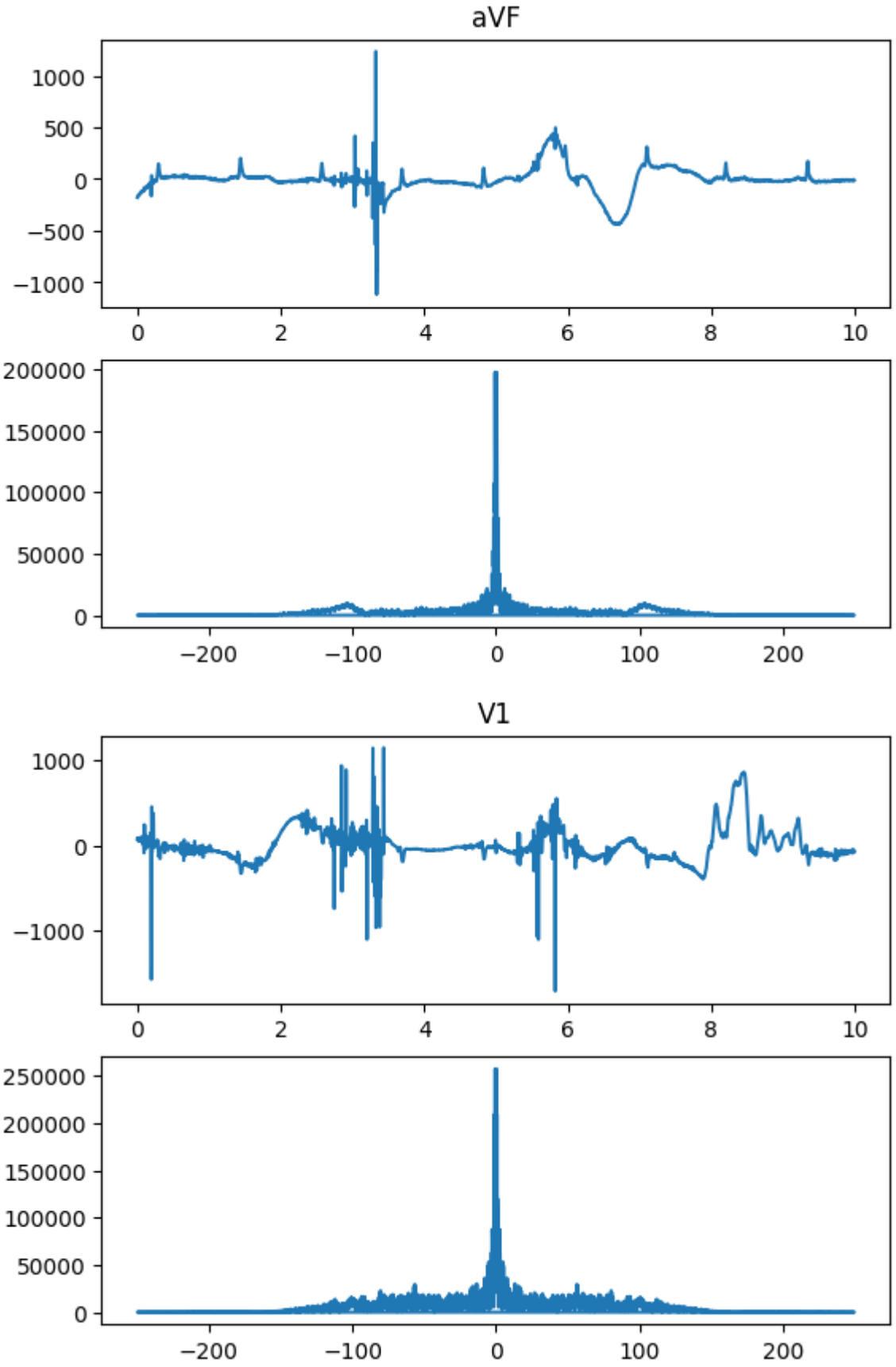
In [ ]:
```
for derivation in header[1::]:
    derivationFourierPlot(dataset, derivation)
```

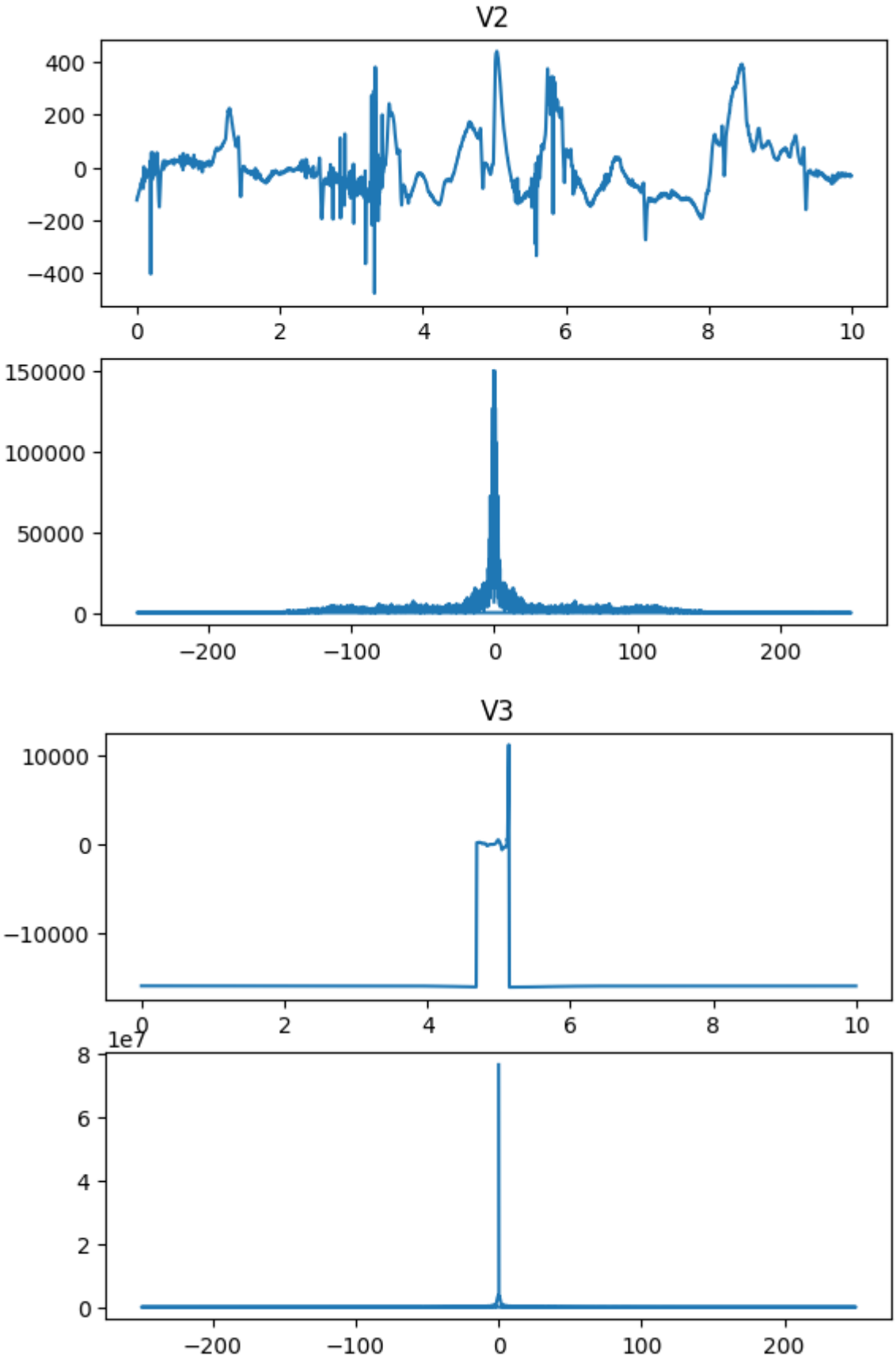


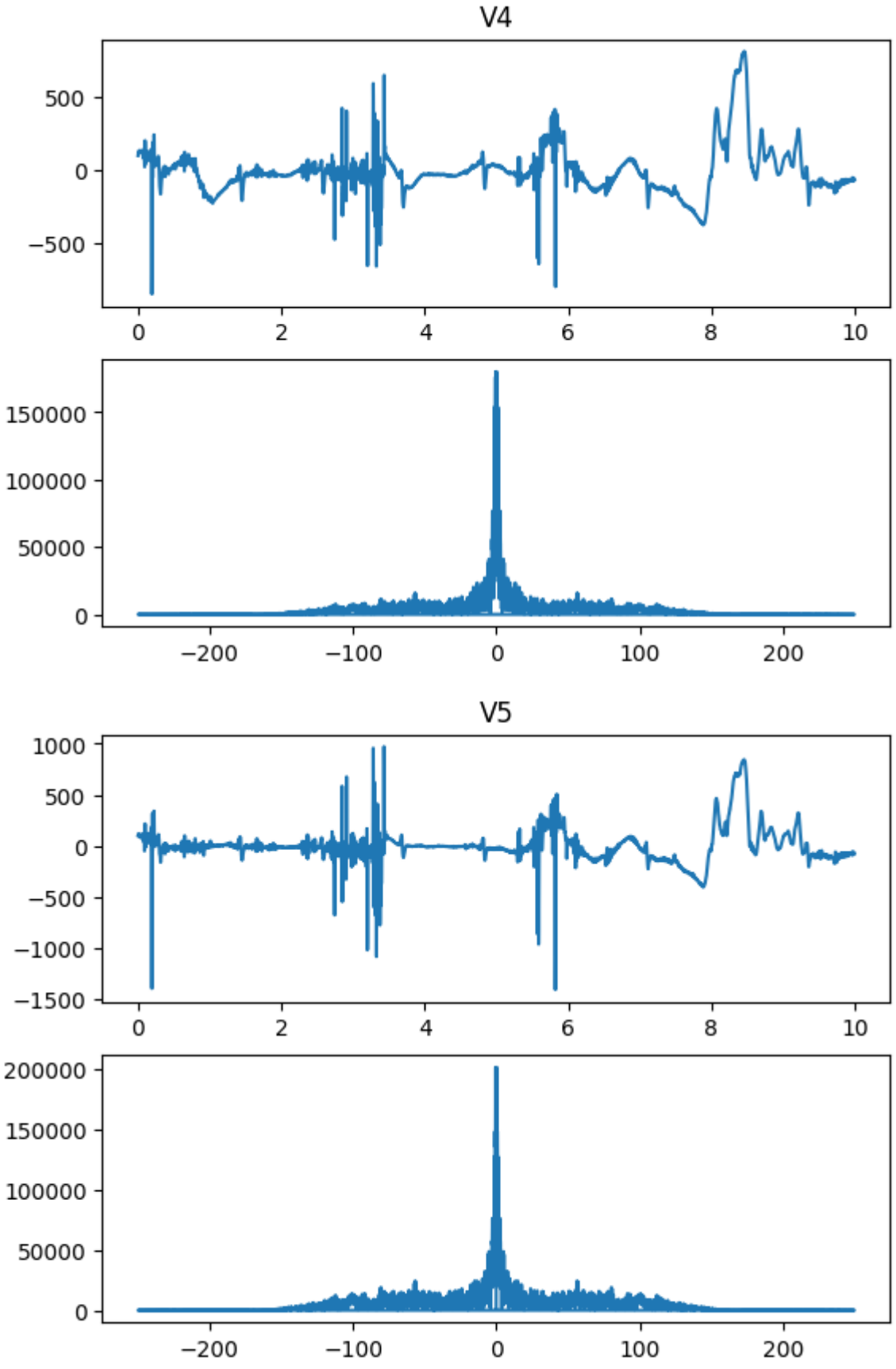

lead II
250
0
−250
−500
−750
−1000
0
2
4
6
8
10
300000
200000
100000
0
−200
−100
0
100
200
lead II
1000
500
0
−500
−1000
0
2
4
6
8
10
300000
200000
100000
0
−200
−100
0
100
200

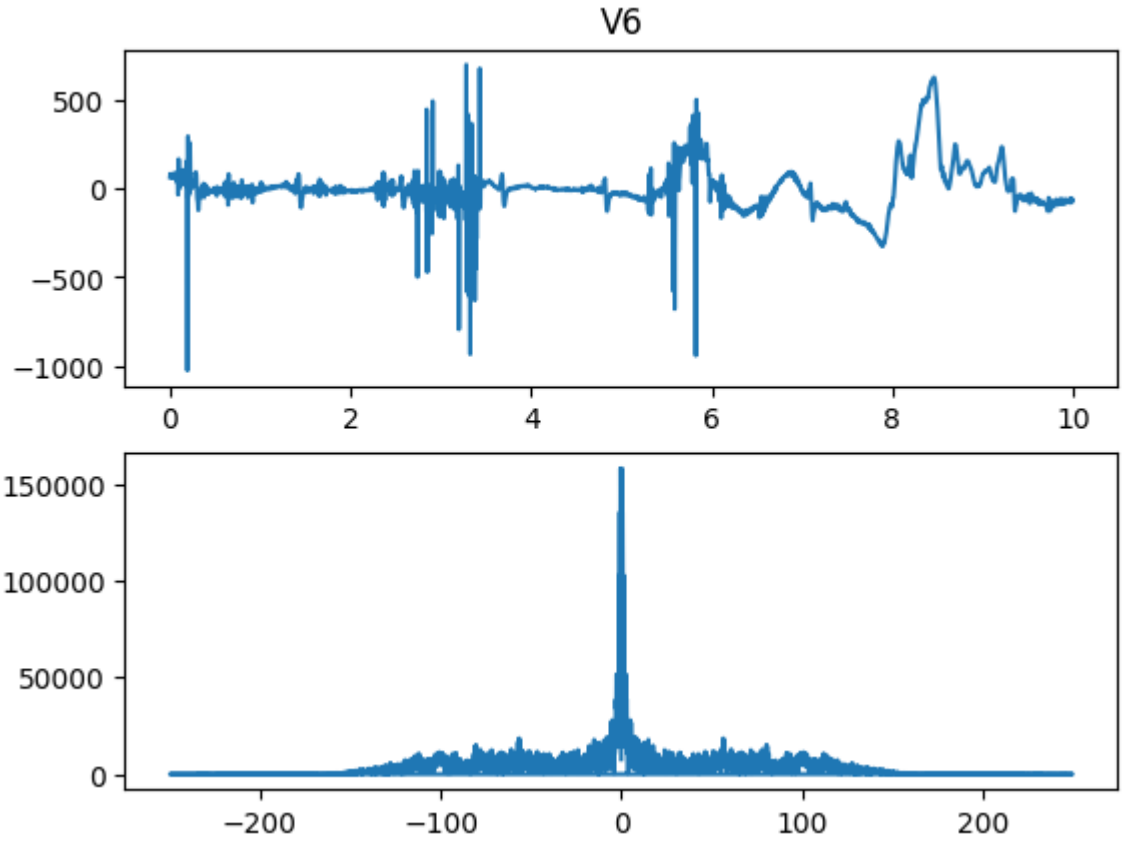

aVR
500
250
0
−250
−500
0
2
4
6
8
10
150000
100000
50000
0
−200
−100
0
100
200


aVL
500
0
−500
−1000
0
2
4
6
8
10
300000
200000
100000
0
−200
−100
0
100
200


aVF
1000
500
0
−500
−1000
0
2
4
6
8
10
200000
150000
100000
50000
0
−200
−100
0
100
200
V1
1000
0
−1000
0
2
4
6
8
10
250000
200000
150000
100000
50000
0
−200
−100
0
100
200

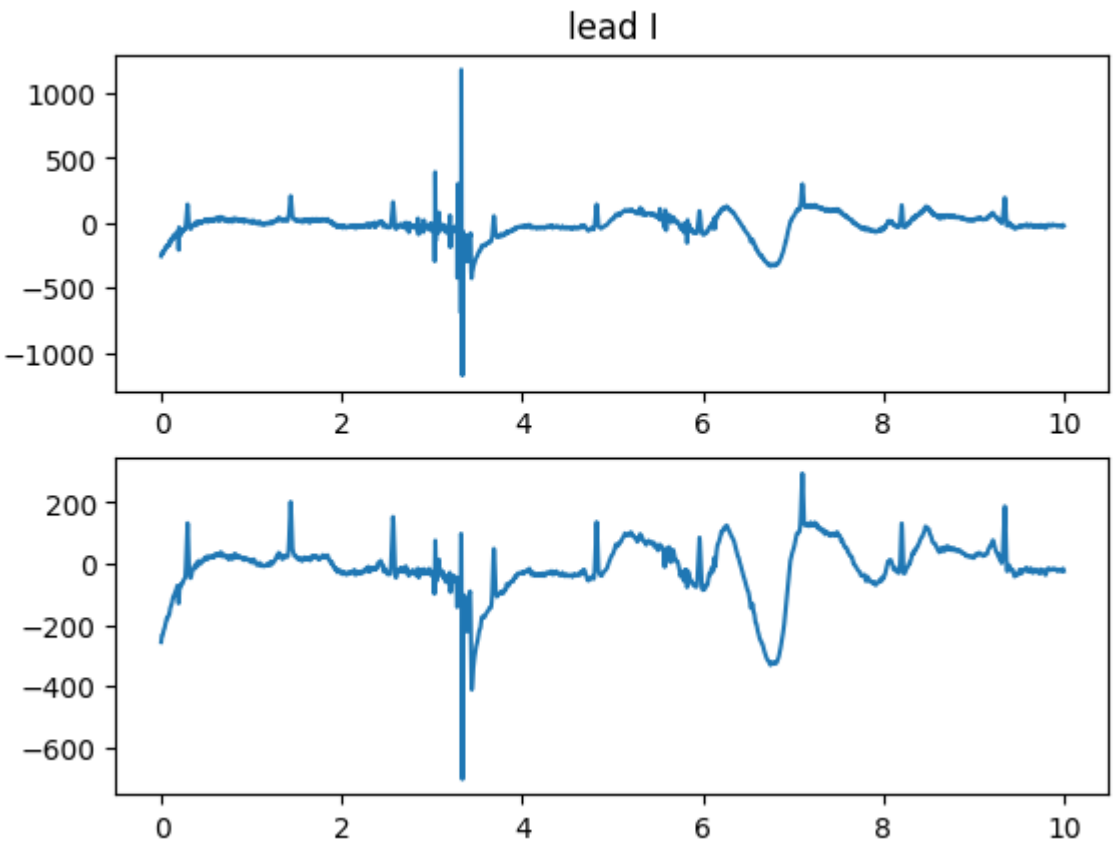

V2
400
200
0
−200
−400
0
2
4
6
8
10
150000
100000
50000
0
−200
−100
0
100
200
V3
10000
0
−10000
0
2
4
6
8
10
1e7
8
6
4
2
0
−200
−100
0
100
200

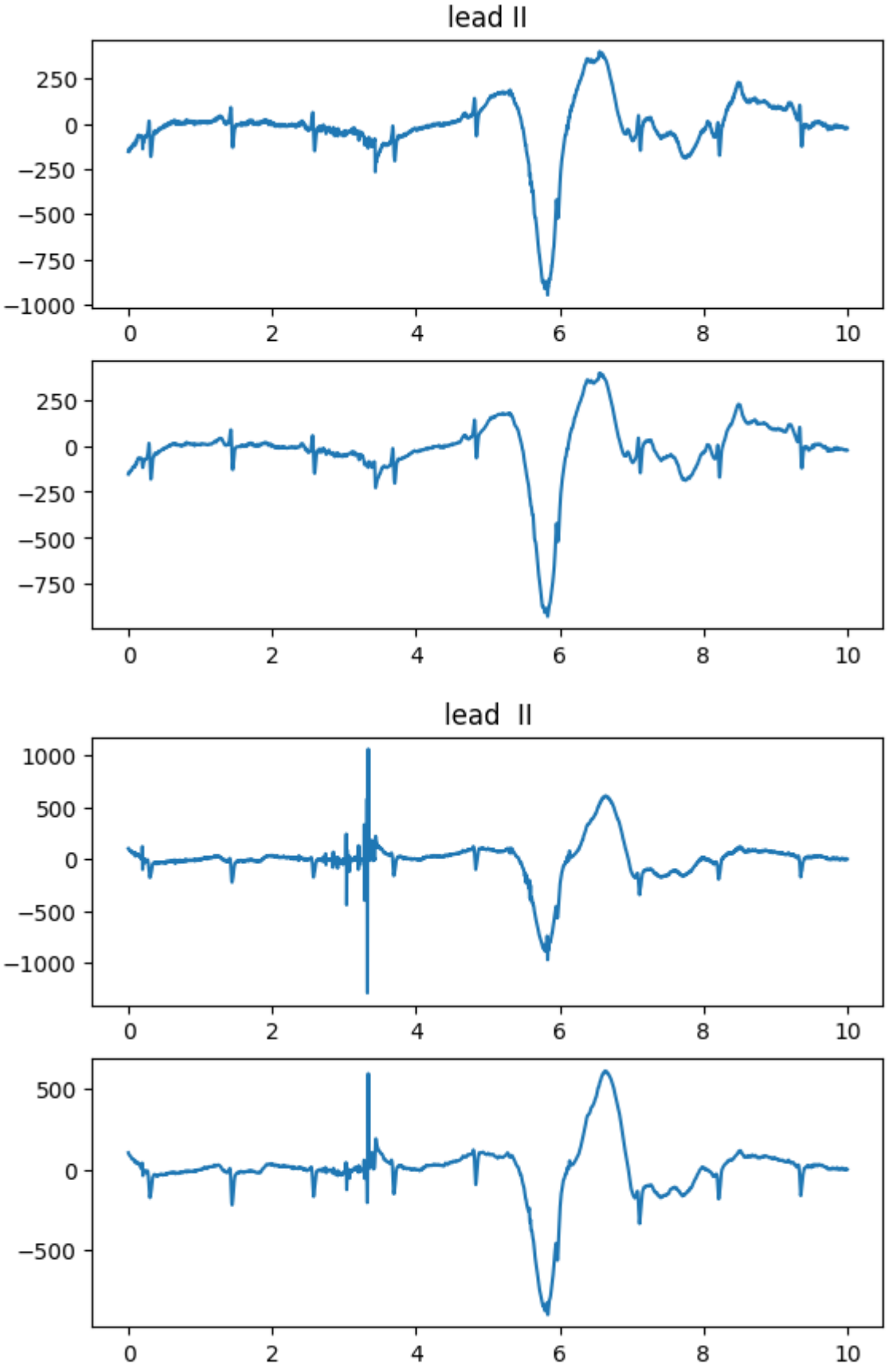

V4
500
0
−500
0
2
4
6
8
10
150000
100000
50000
0
−200
−100
0
100
200
V5
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10
200000
150000
100000
50000
0
−200
−100
0
100
200

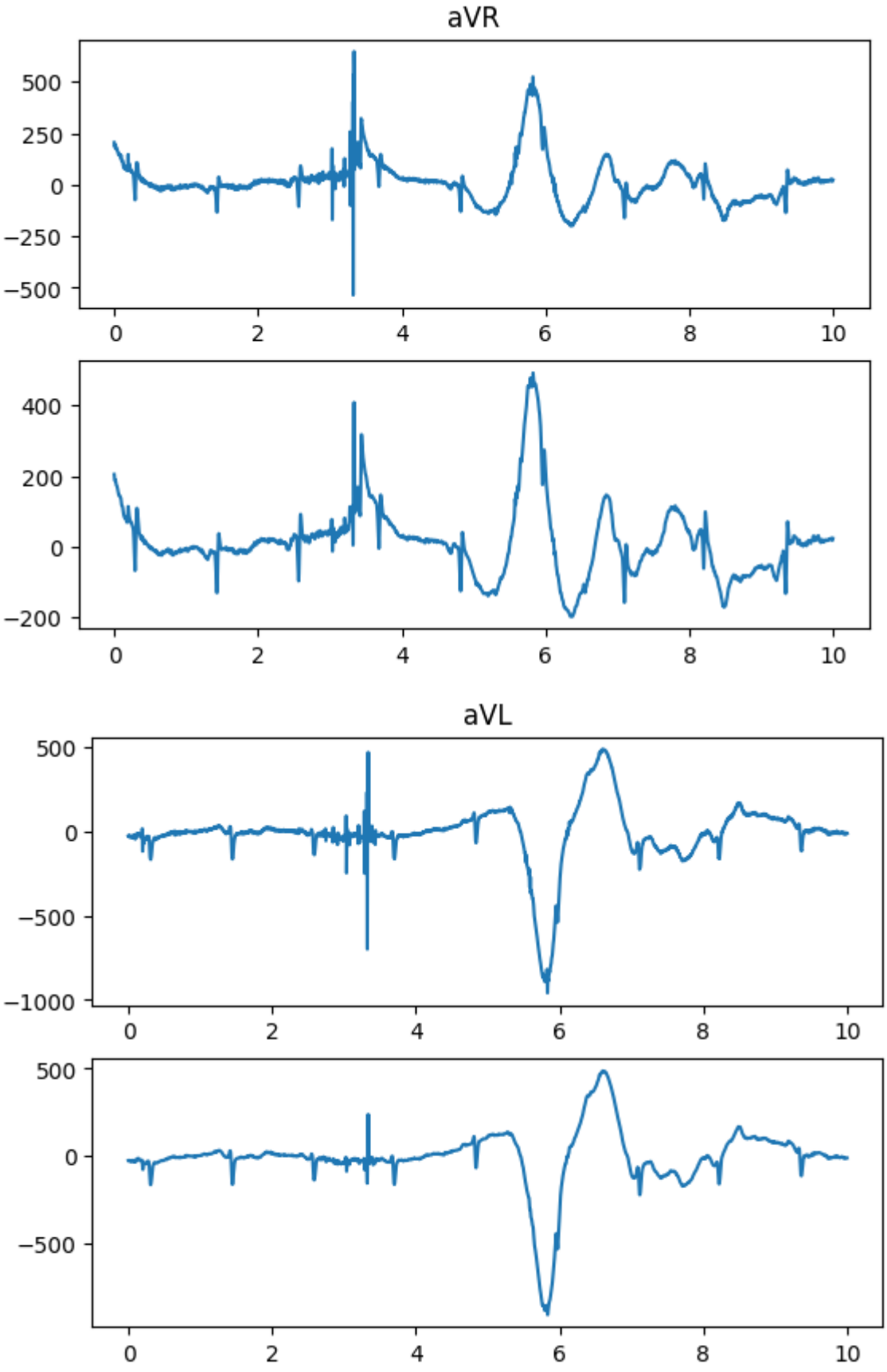


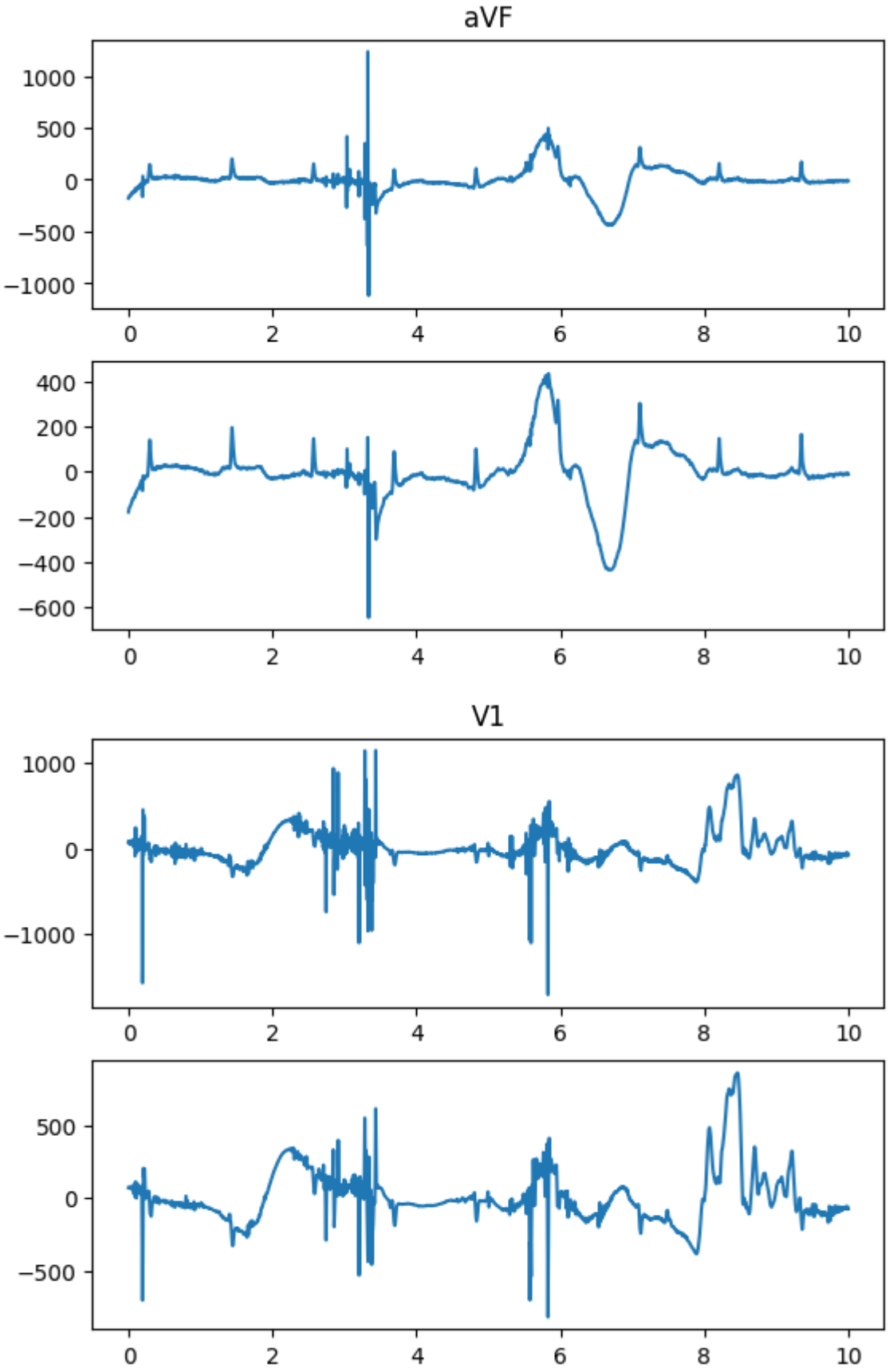
Derivações com Filtro Passa-Baixa com frequência de corte de 50hz

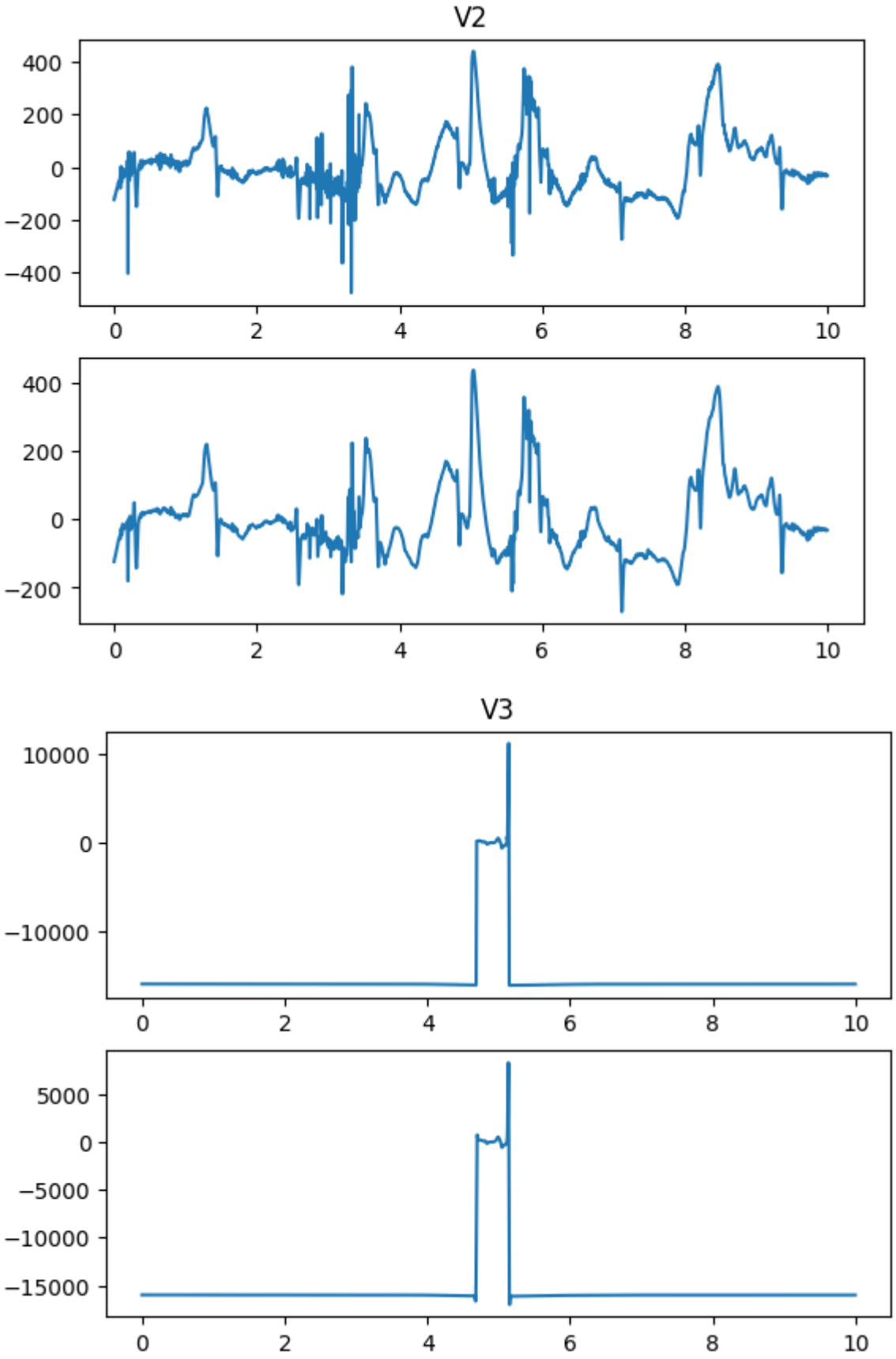
In [ ]:
```
for derivation in header[1::]:
    derivationLowPassPlot(dataset, derivation, 50)
```

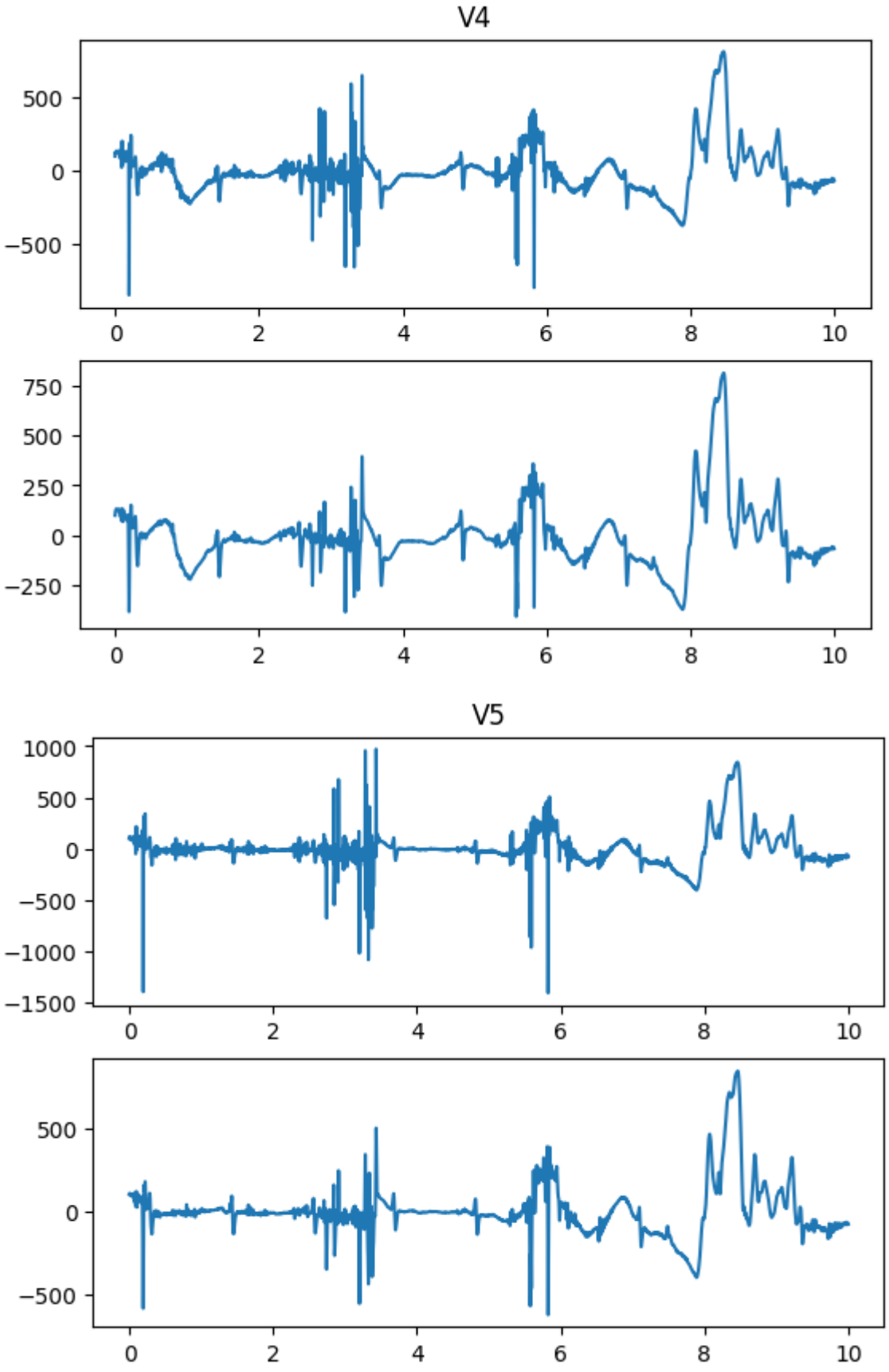


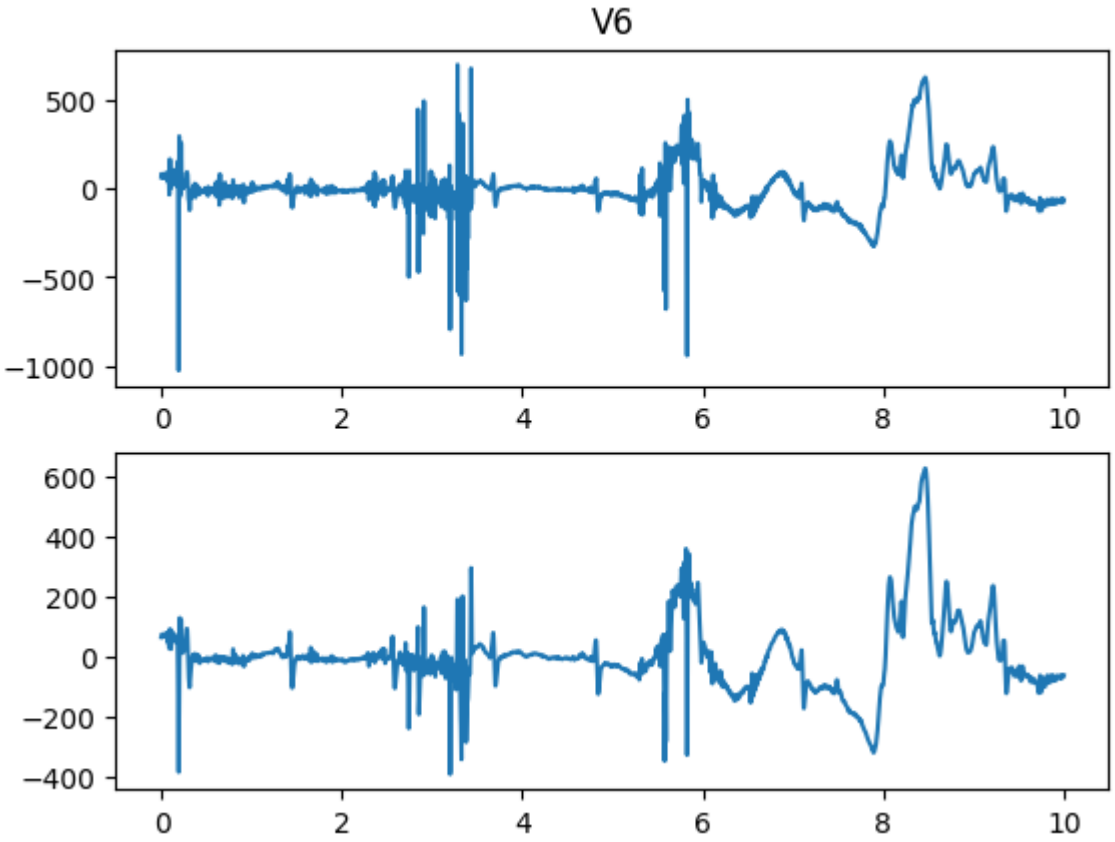

lead II
250
0
−250
−500
−750
−1000
0
2
4
6
8
10
250
0
−250
−500
−750
0
2
4
6
8
10
lead II
1000
500
0
−500
−1000
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10


aVR
500
250
0
−250
−500
0
2
4
6
8
10
400
200
0
−200
0
2
4
6
8
10
aVL
500
0
−500
−1000
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10


aVF
1000
500
0
−500
−1000
0
2
4
6
8
10
400
200
0
−200
−400
−600
0
2
4
6
8
10
V1
1000
0
−1000
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10

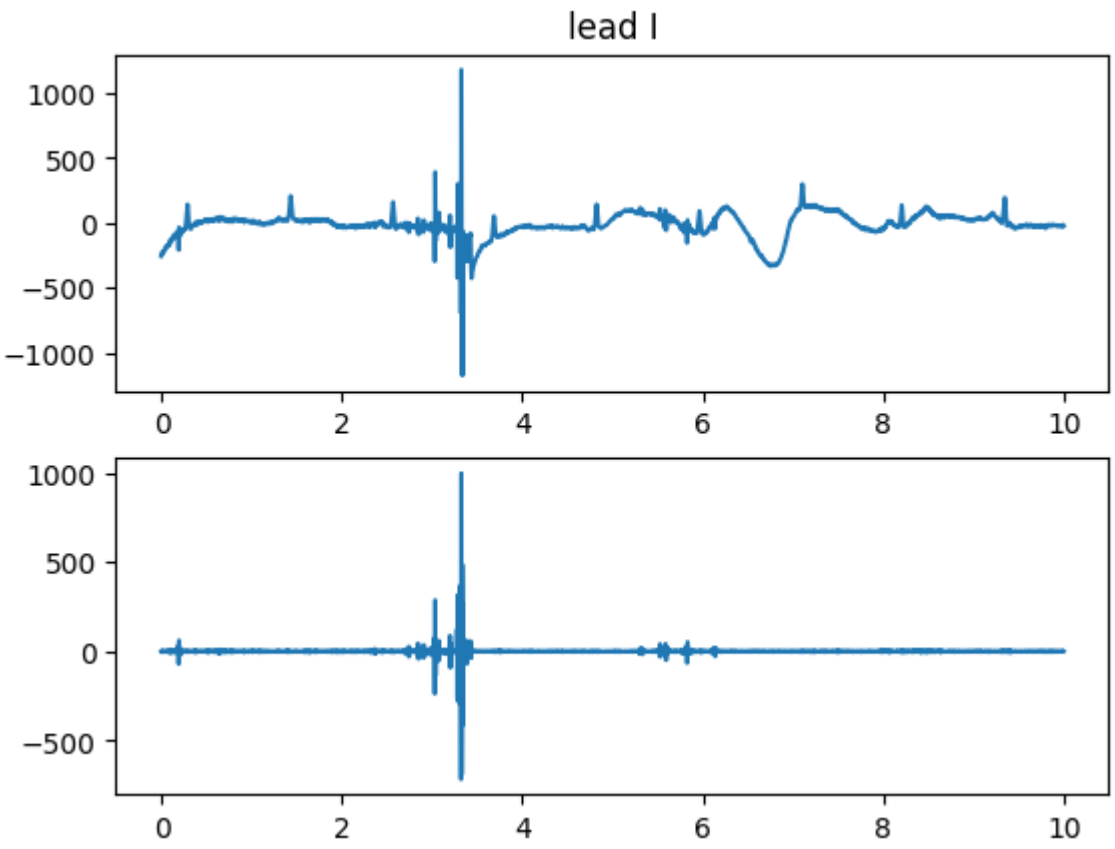

V2
400
200
0
−200
−400
0
2
4
6
8
10
400
200
0
−200
0
2
4
6
8
10
V3
10000
0
−10000
0
2
4
6
8
10
5000
0
−5000
−10000
−15000
0
2
4
6
8
10

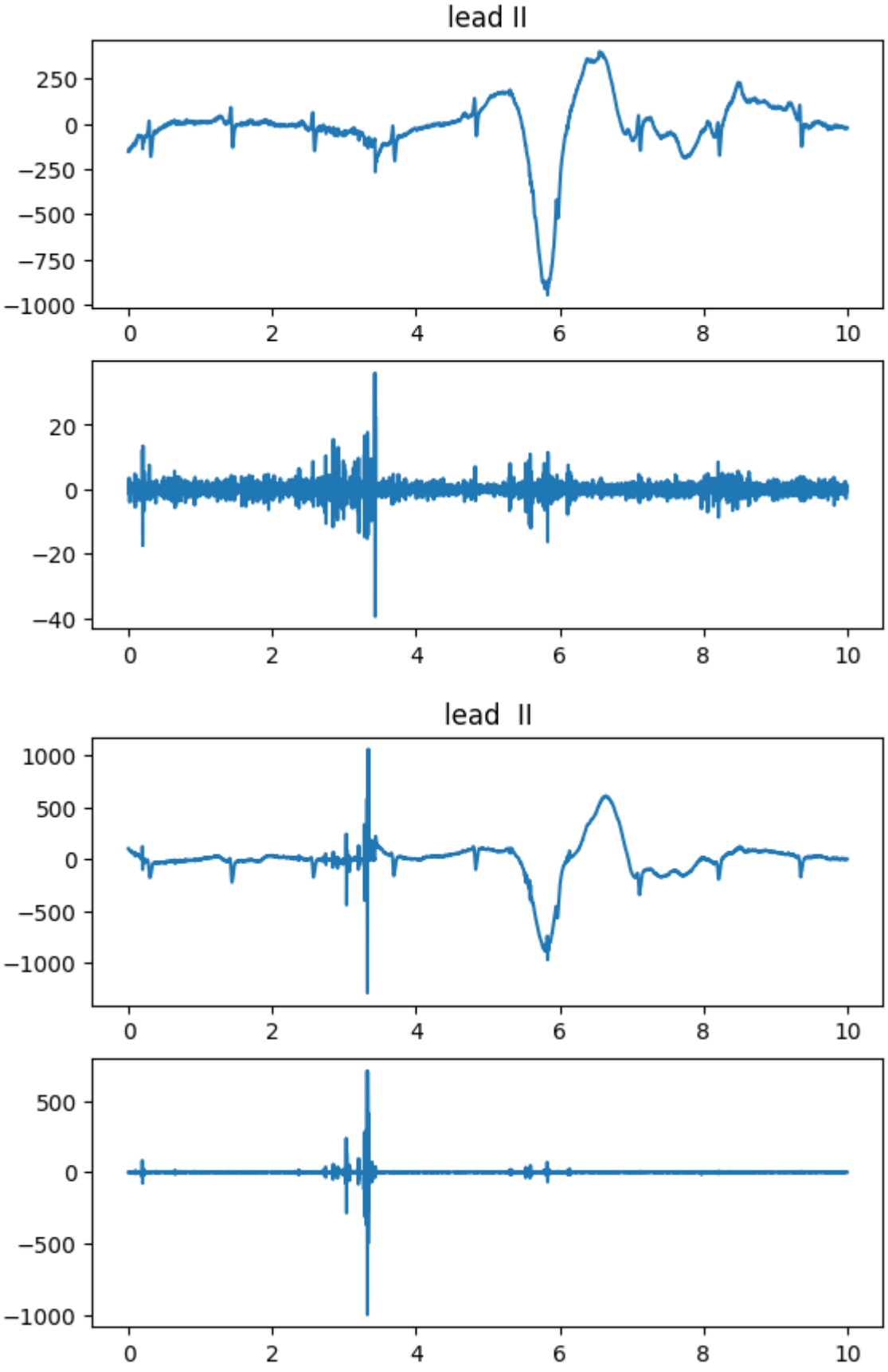

V4
500
0
−500
0
2
4
6
8
10
750
500
250
0
−250
0
2
4
6
8
10
V5
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10

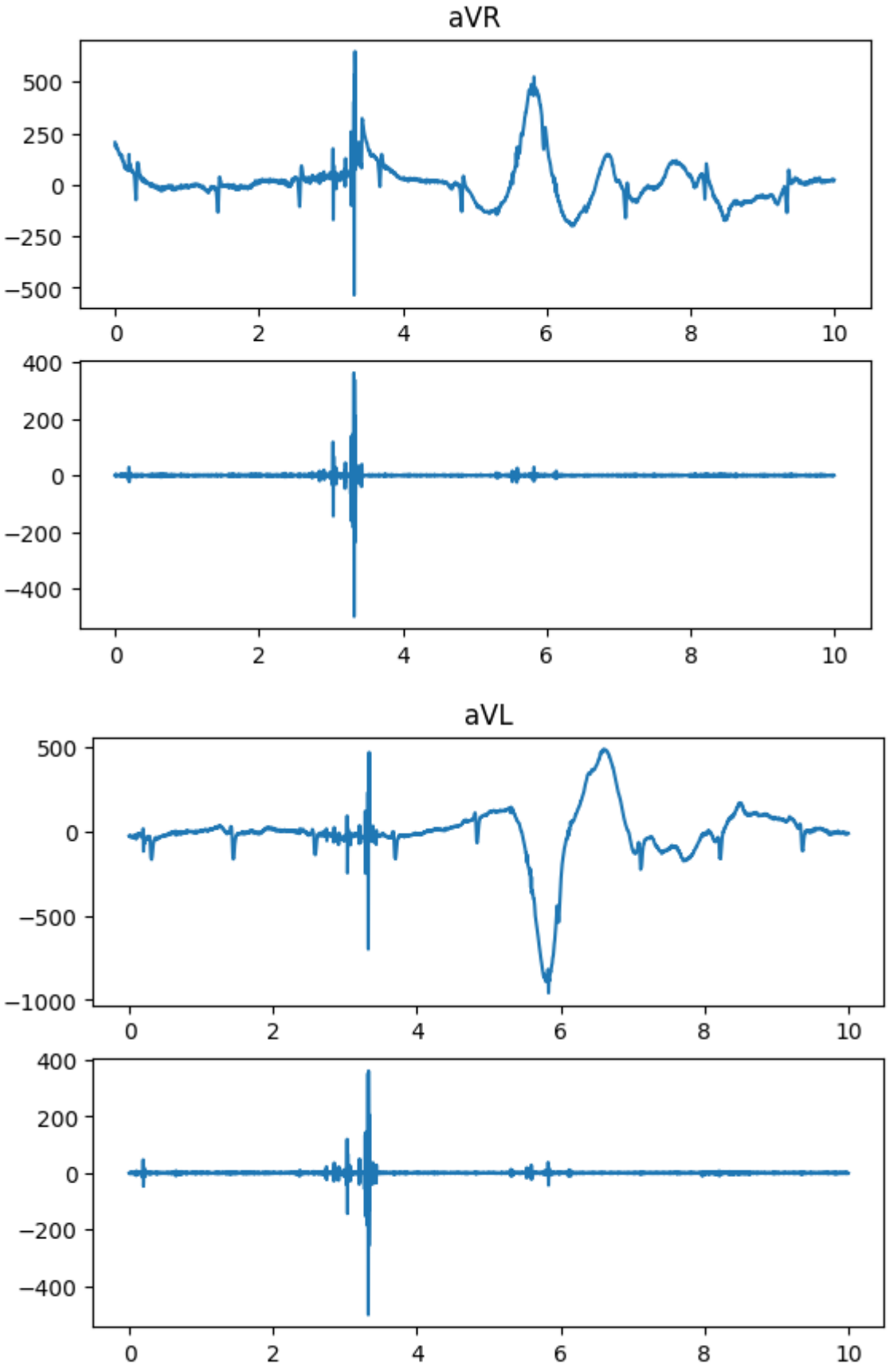


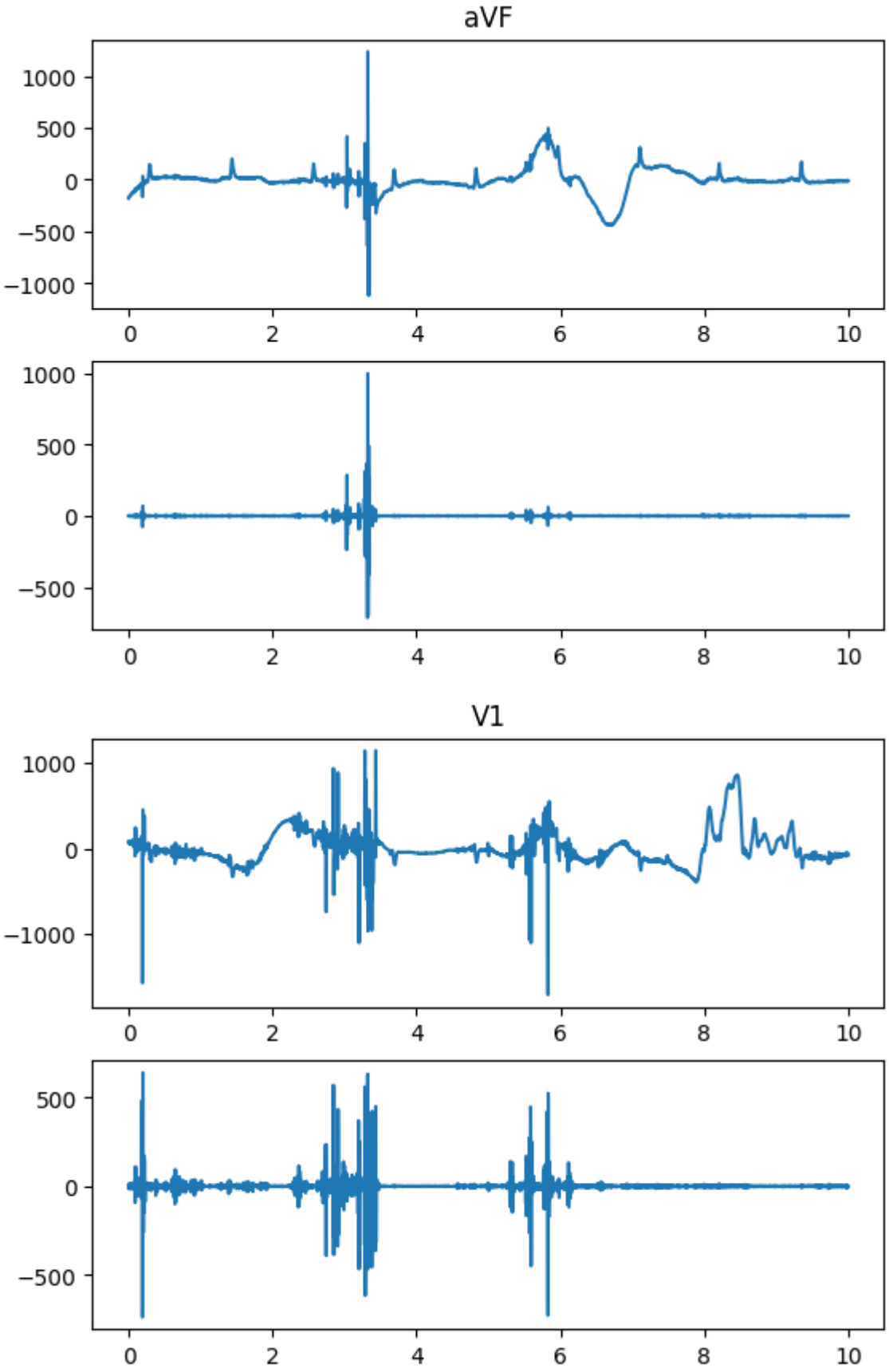
Derivações com Filtro Passa-Alta com frequência de corte de 60hz

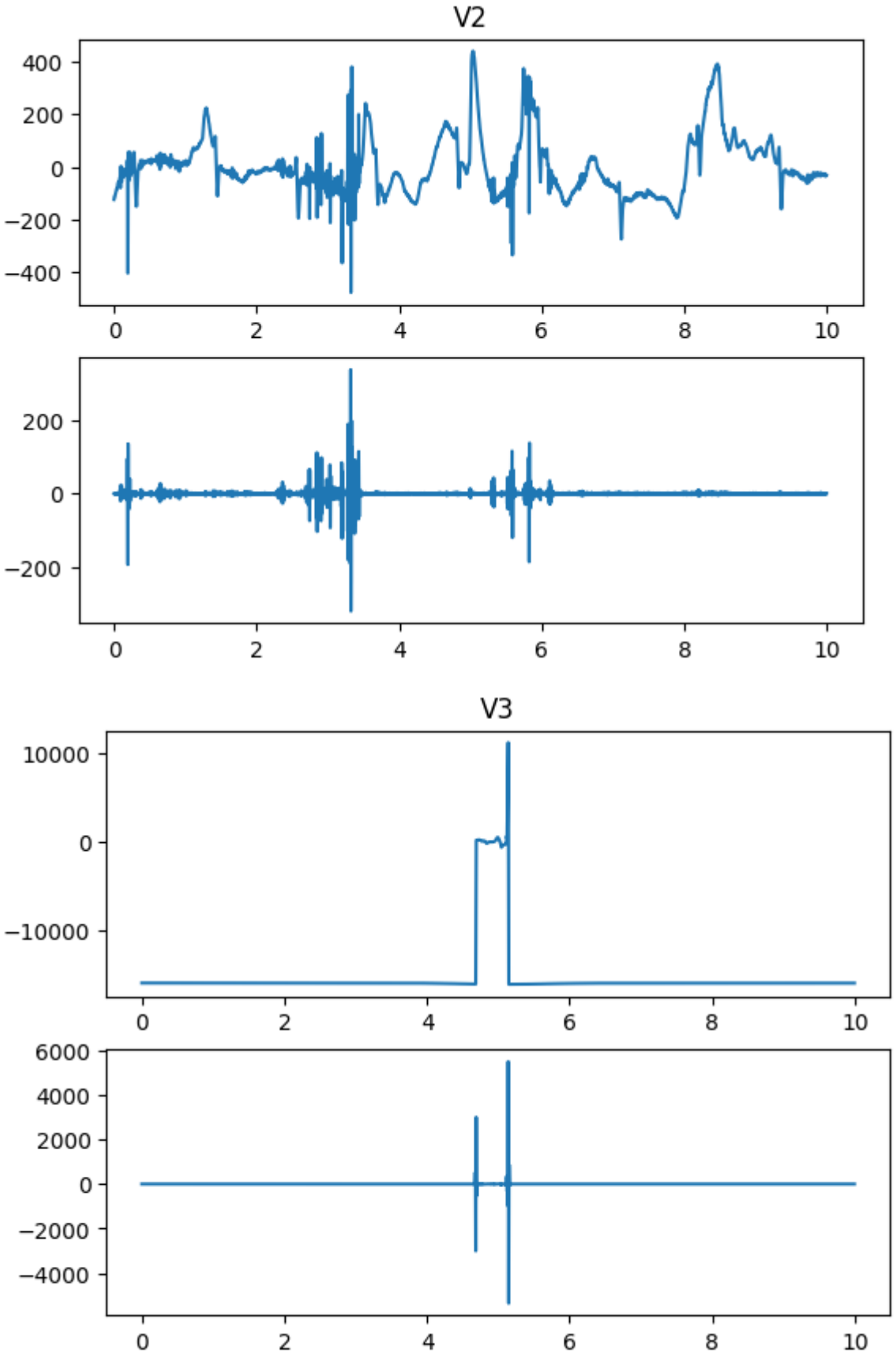
In [ ]:
```
for derivation in header[1::]:
    derivationHighPassPlot(dataset, derivation, 60)
```

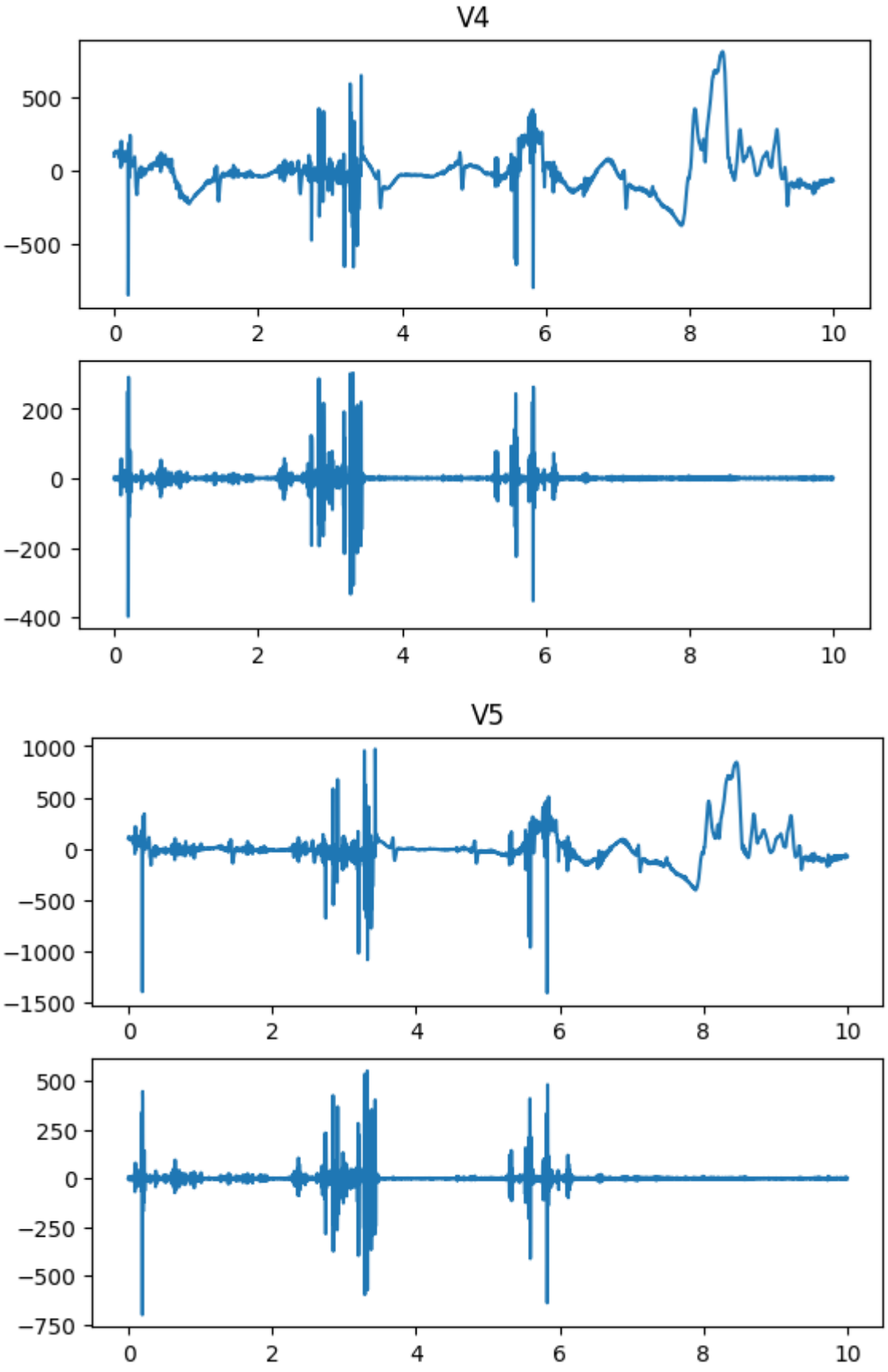


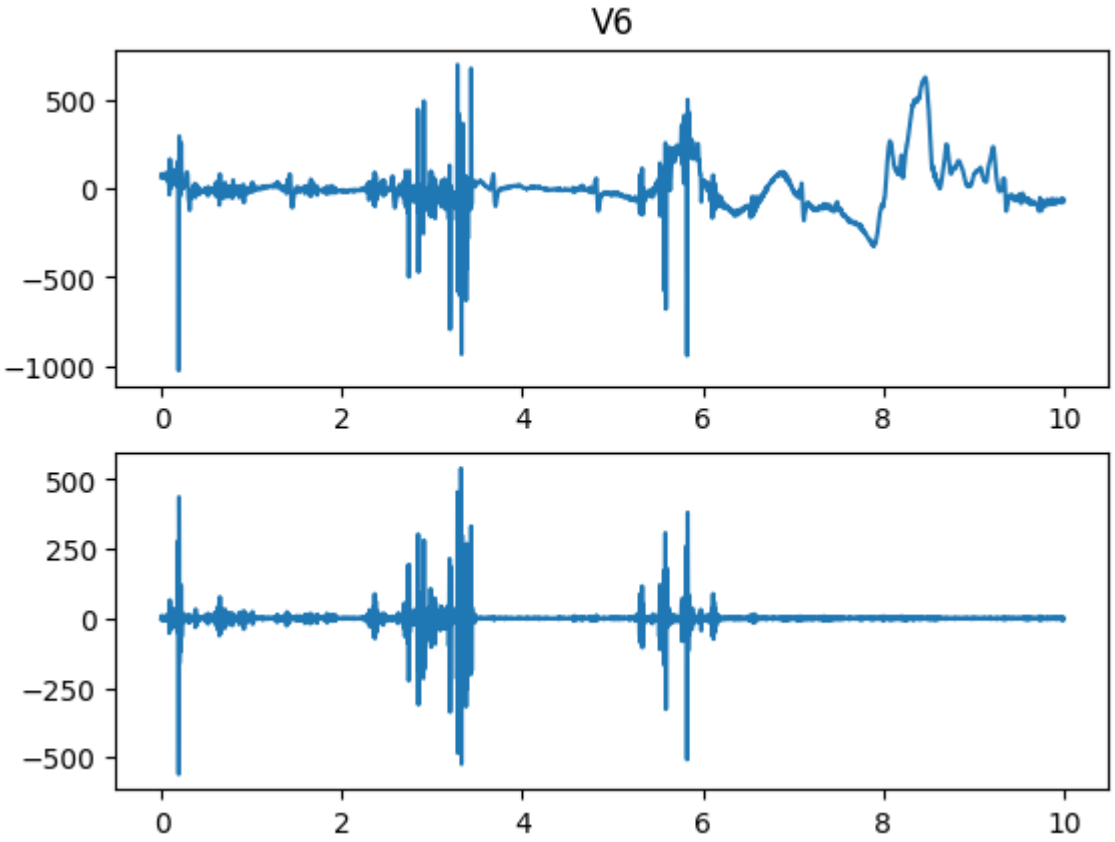

lead II
250
0
−250
−500
−750
−1000
0
2
4
6
8
10
20
0
−20
−40
0
2
4
6
8
10
lead II
1000
500
0
−500
−1000
0
2
4
6
8
10
500
0
−500
−1000
0
2
4
6
8
10


aVR
500
250
0
−250
−500
0
2
4
6
8
10
400
200
0
−200
−400
0
2
4
6
8
10
aVL
500
0
−500
−1000
0
2
4
6
8
10
400
200
0
−200
−400
0
2
4
6
8
10


aVF
1000
500
0
−500
−1000
0
2
4
6
8
10
1000
500
0
−500
0
2
4
6
8
10
V1
1000
0
−1000
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10

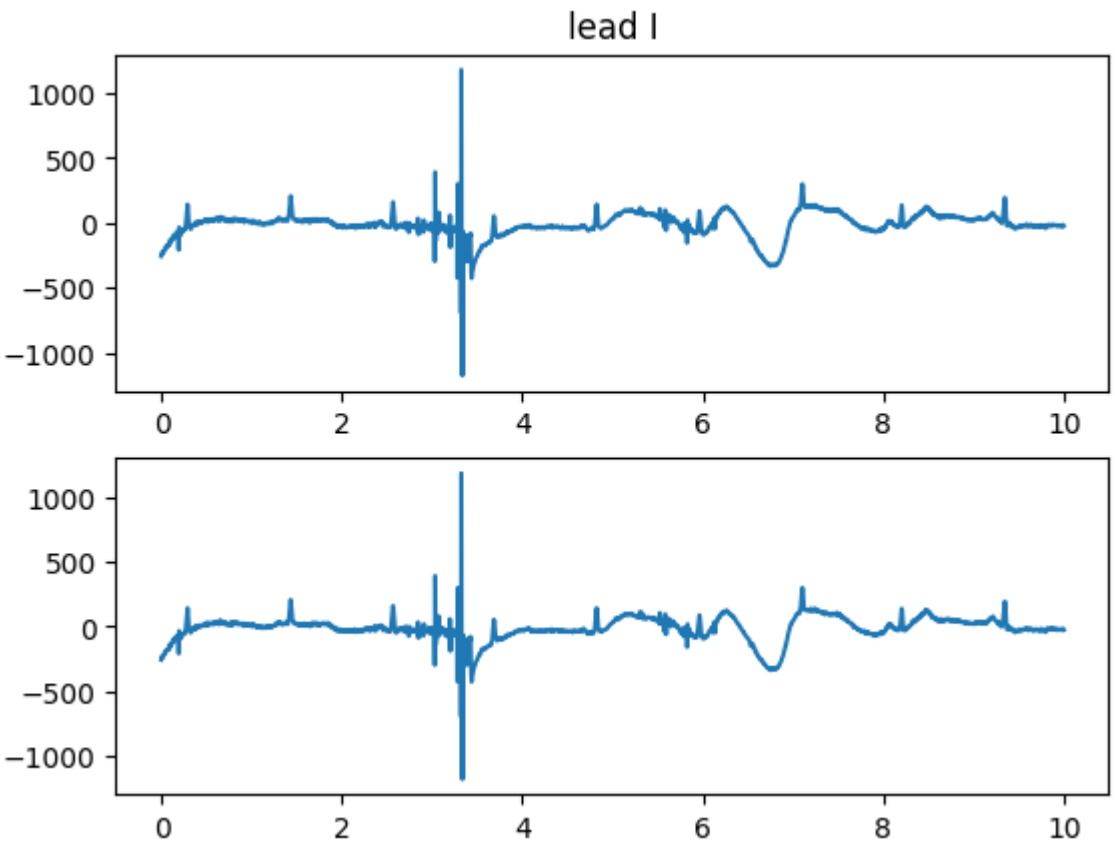

V2
400
200
0
−200
−400
0
2
4
6
8
10
200
0
−200
0
2
4
6
8
10
V3
10000
0
−10000
0
2
4
6
8
10
6000
4000
2000
0
−2000
−4000
0
2
4
6
8
10

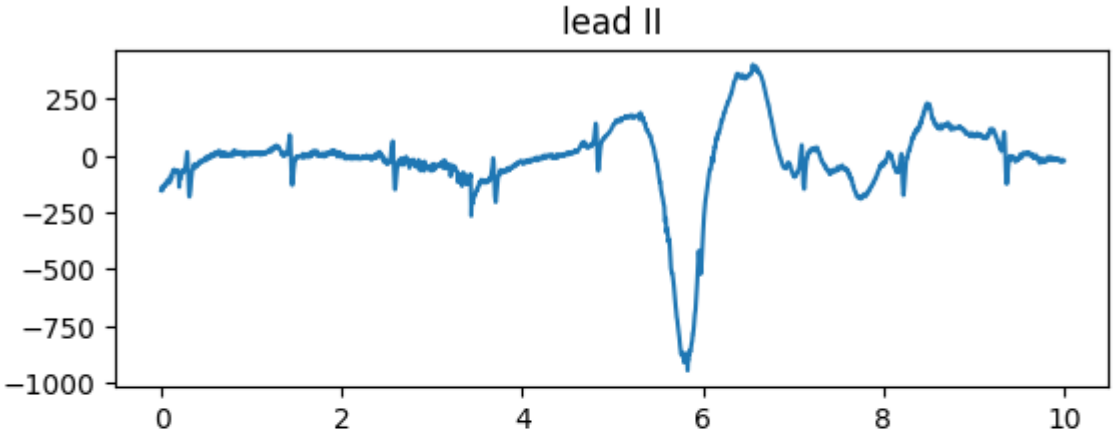

V4
500
0
−500
0
2
4
6
8
10
200
0
−200
−400
0
2
4
6
8
10
V5
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10
500
250
0
−250
−500
−750
0
2
4
6
8
10

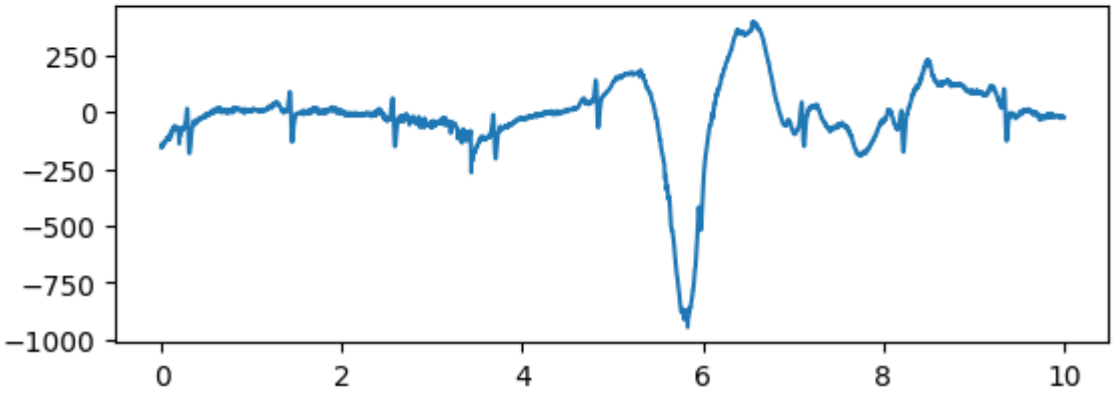


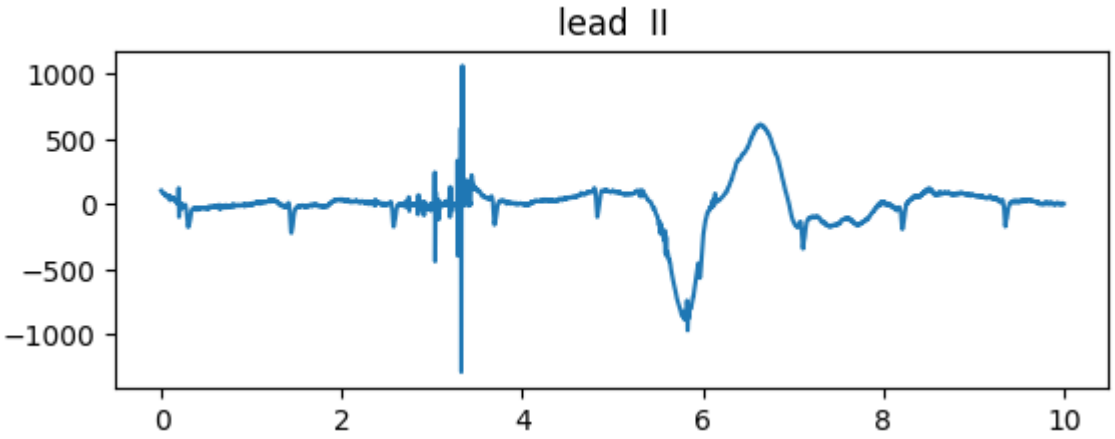
Derivações com Filtro Notch com banda de corte 50 e 60hz

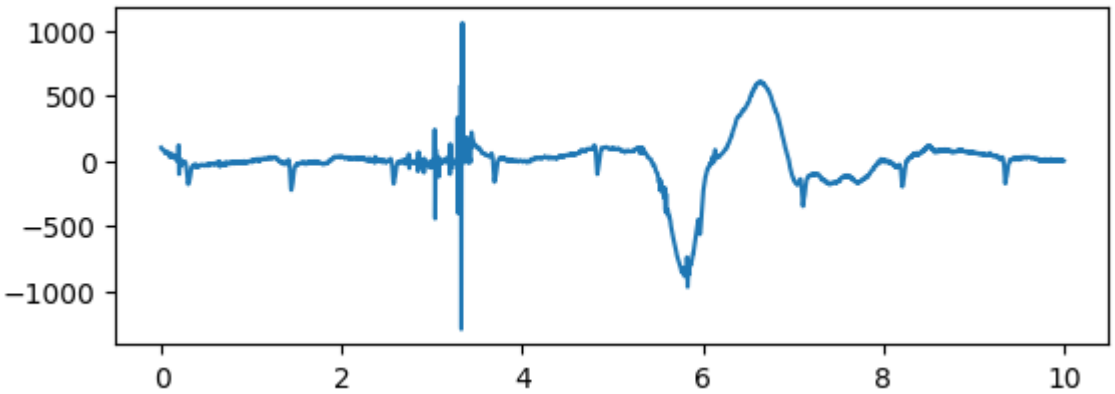
In [ ]:
```
for derivation in header[1::]:
    derivationNocthPlot(dataset, derivation, 50)
```

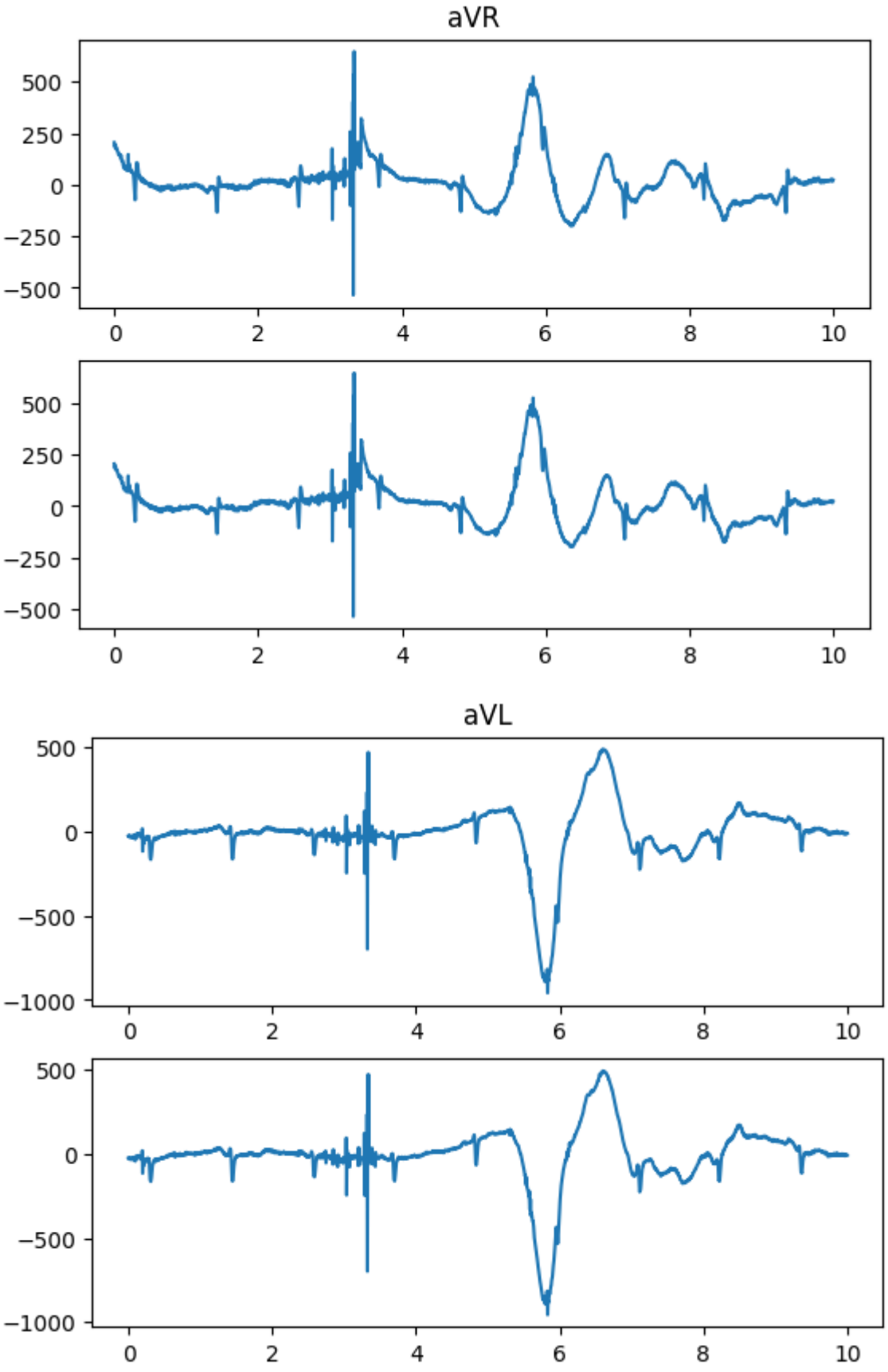


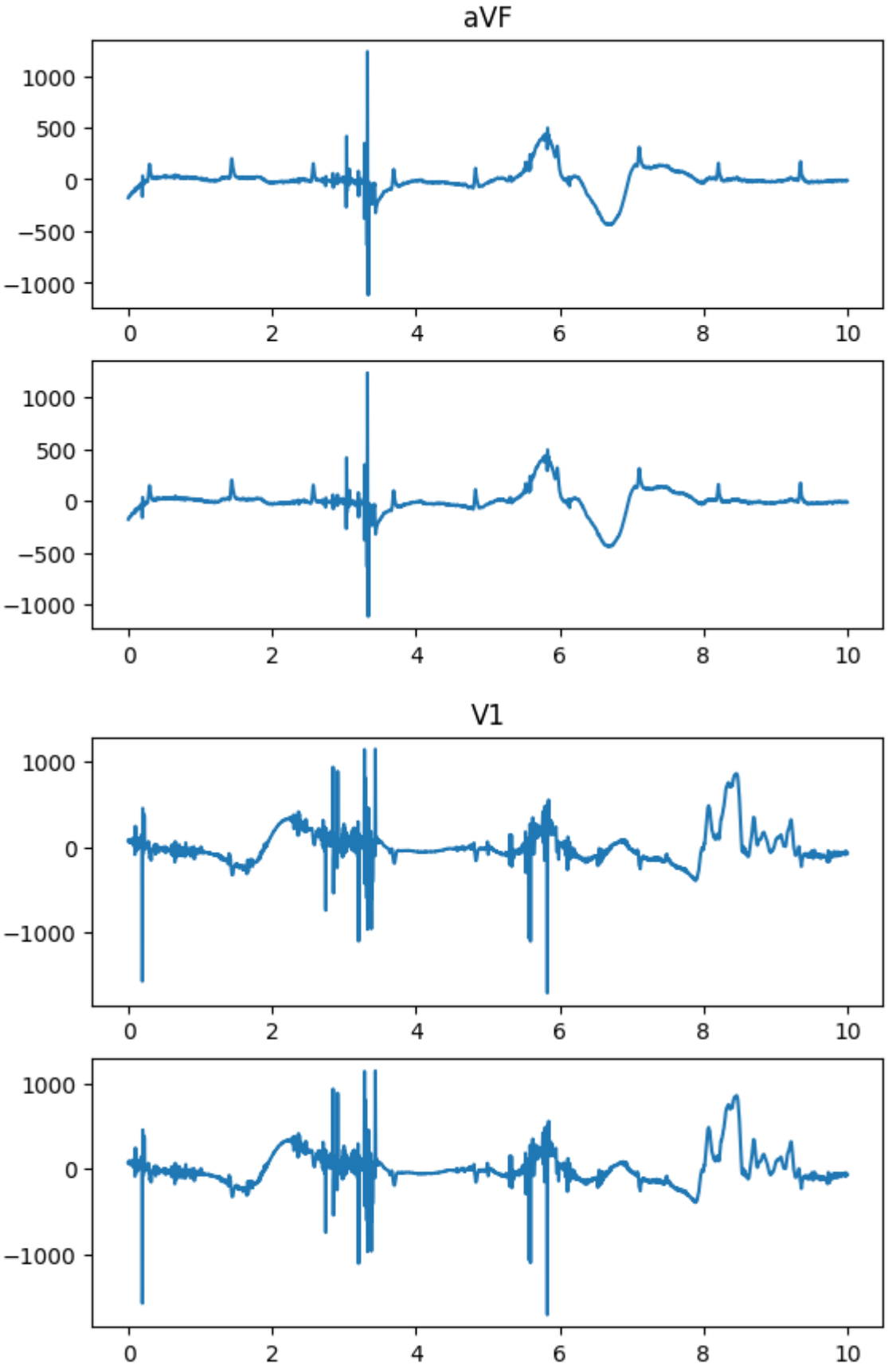

lead II
250
0
−250
−500
−750
−1000
0
2
4
6
8
10

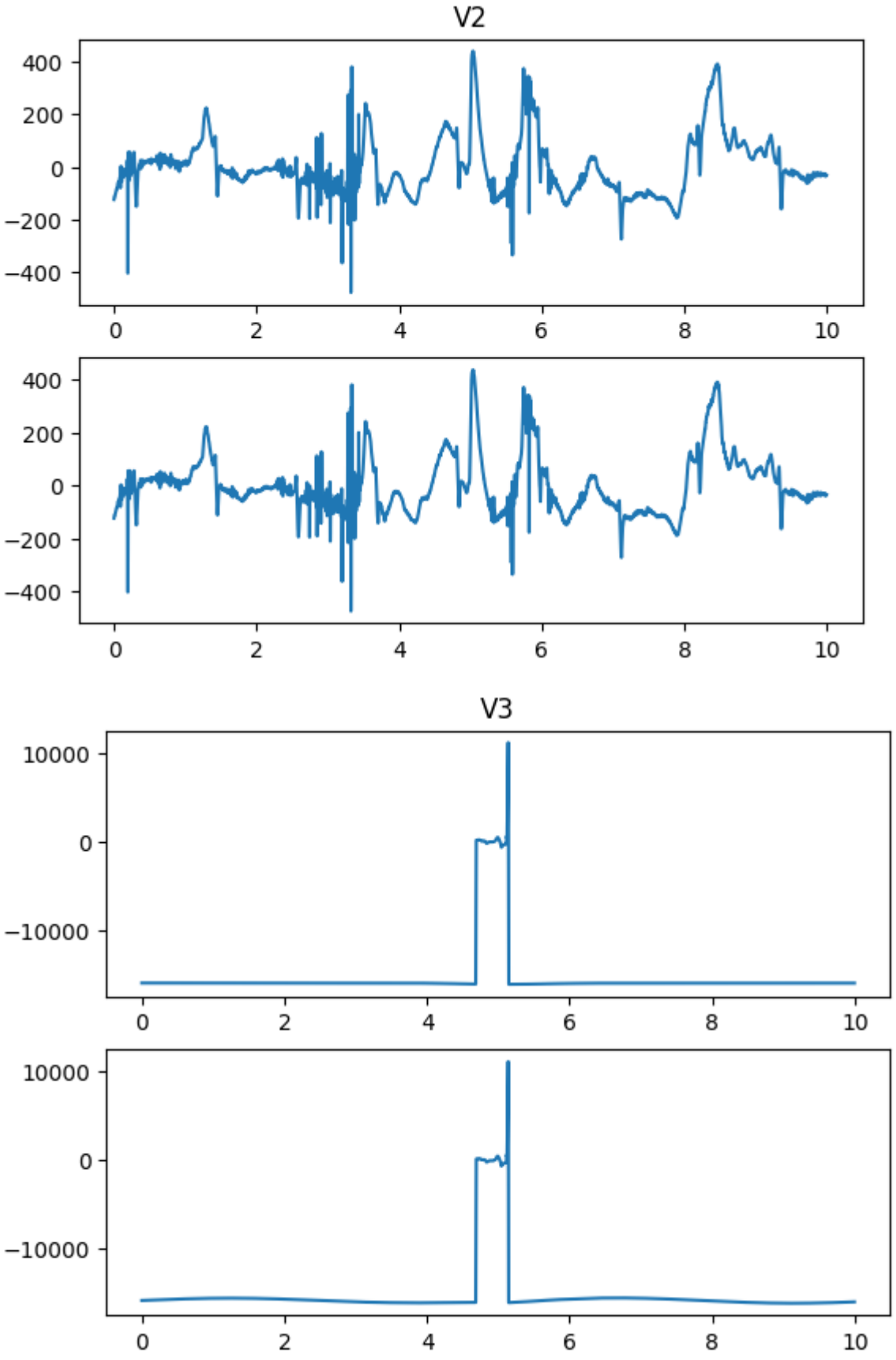

250
0
−250
−500
−750
−1000
0
2
4
6
8
10

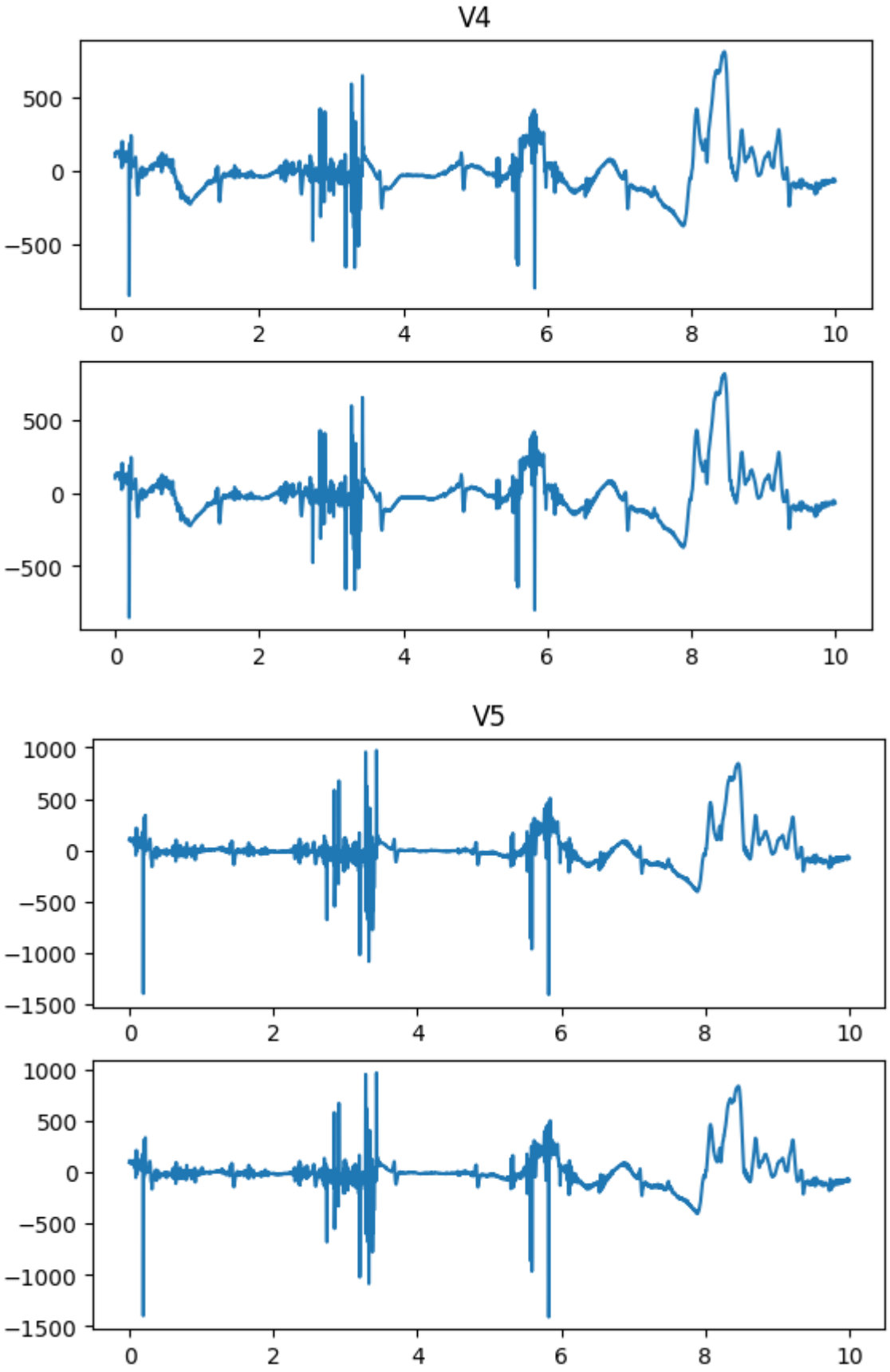

lead II
1000
500
0
−500
−1000
0
2
4
6
8
10

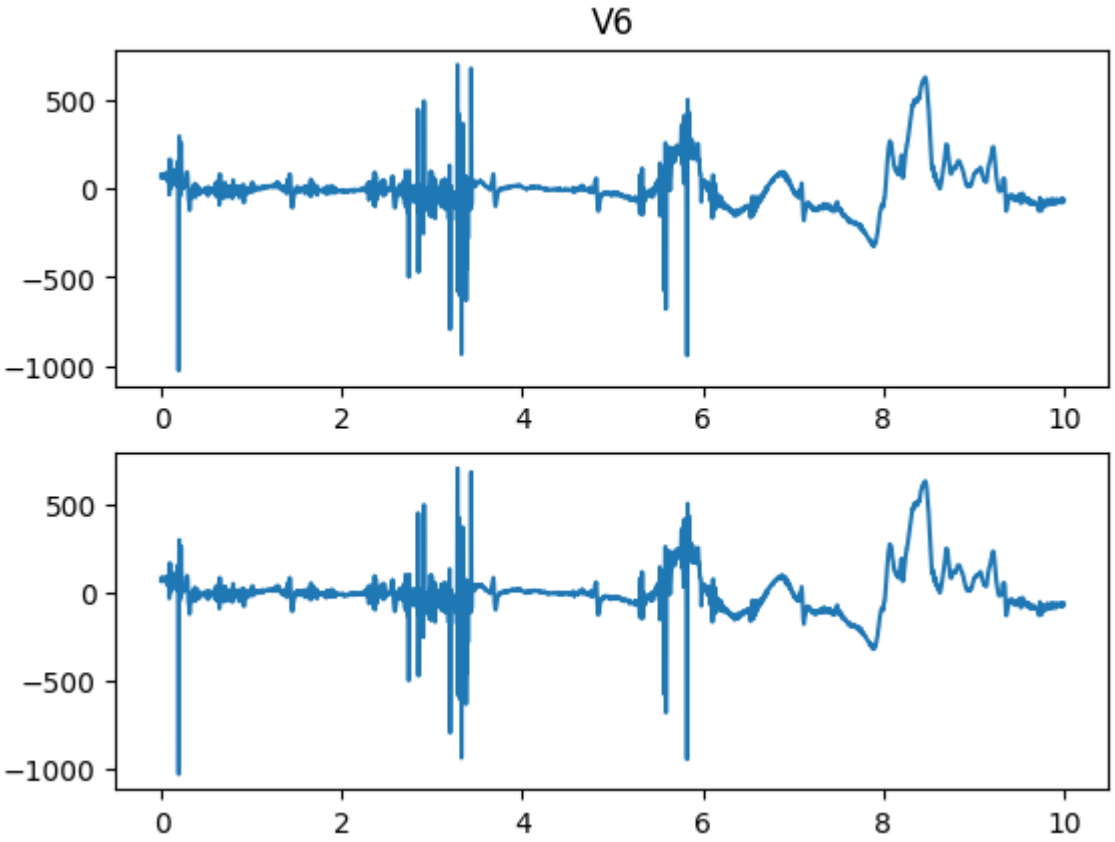

1000
500
0
−500
−1000
0
2
4
6
8
10


aVR
500
250
0
−250
−500
0
2
4
6
8
10
aVL
−1000

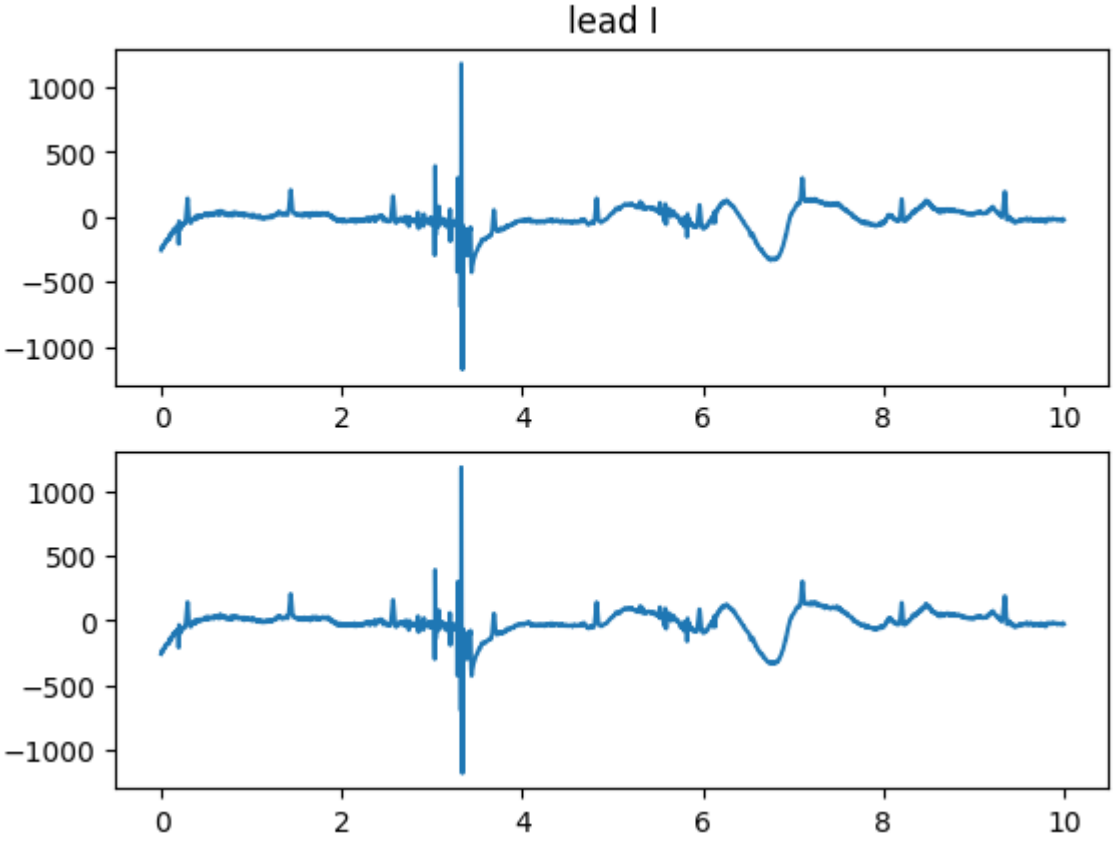

aVF
1000
500
0
−500
−1000
0
2
4
6
8
10
V1

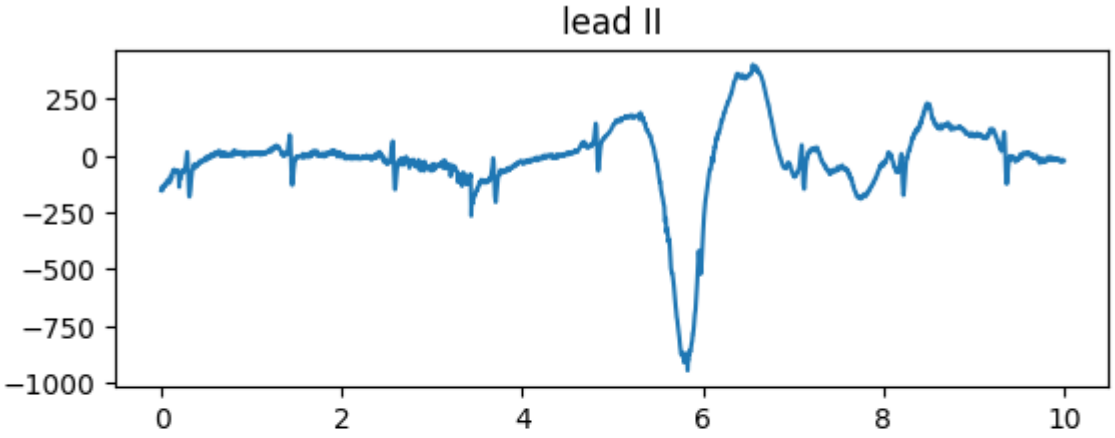

V2
400
200
0
−200
−400
0
2
4
6
8
10
400
200
0
−200
−400
0
2
4
6
8
10
V3
10000
0
−10000
0
2
4
6
8
10
10000
0
−10000
0
2
4
6
8
10

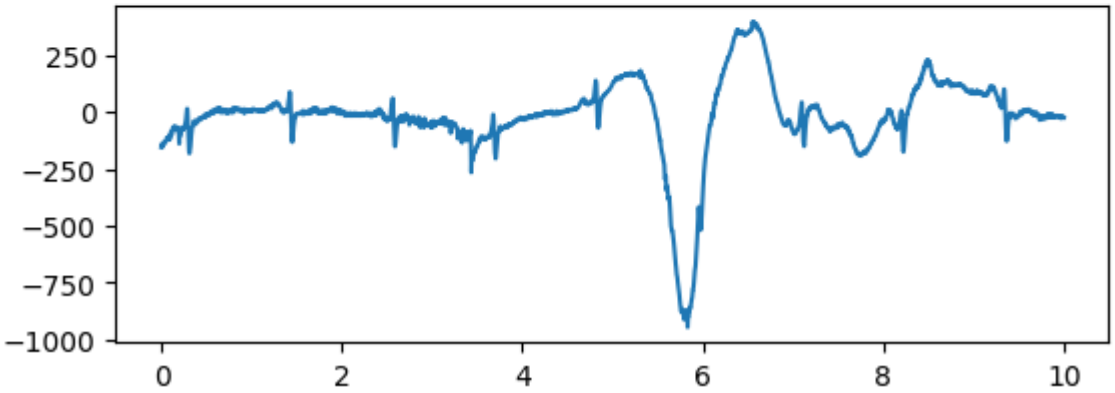

V4
500
0
−500
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10
V5
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10

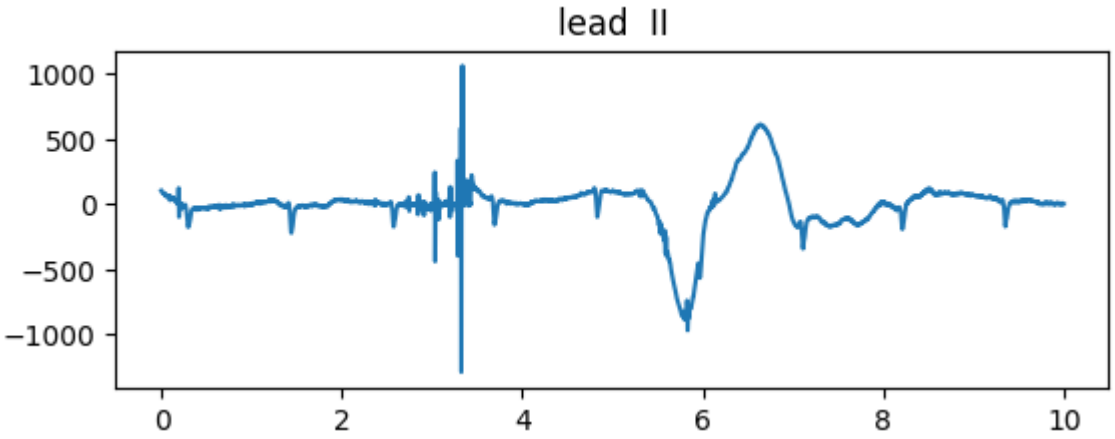


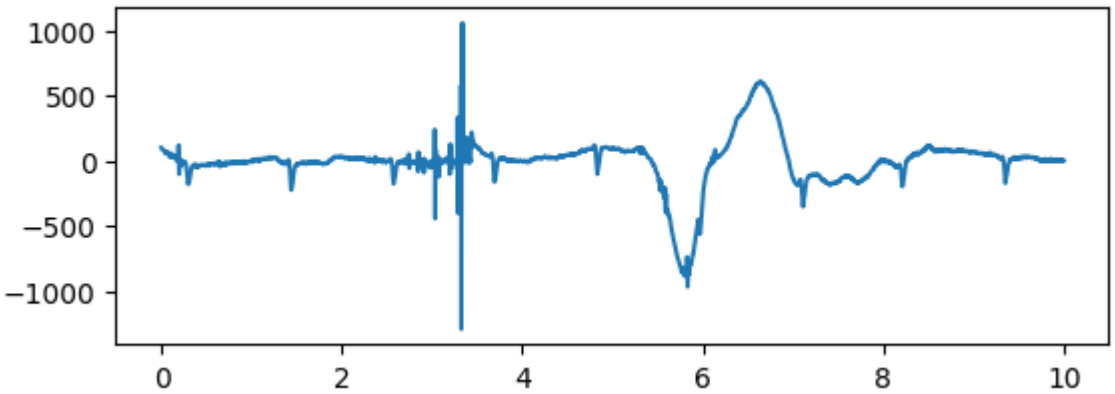
In [ ]:
```
for derivation in header[1::]:
    derivationNocthPlot(dataset, derivation, 60)
```

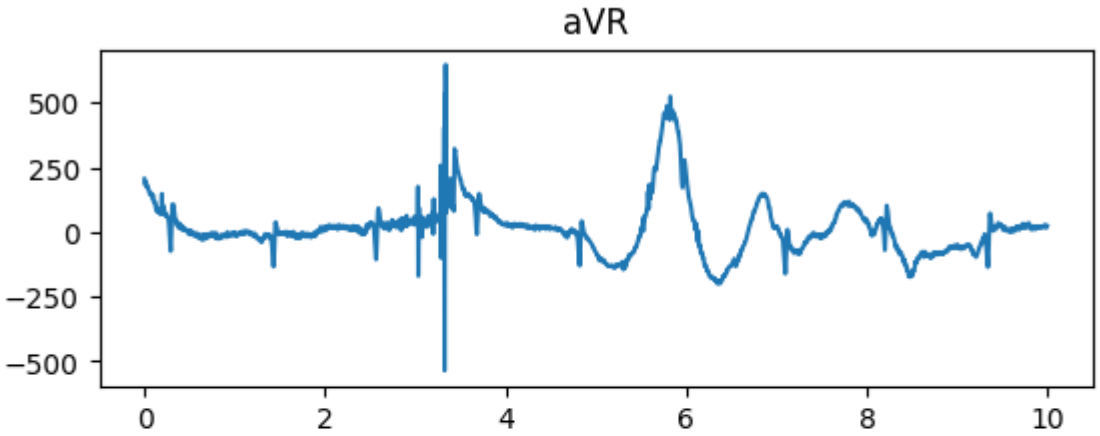


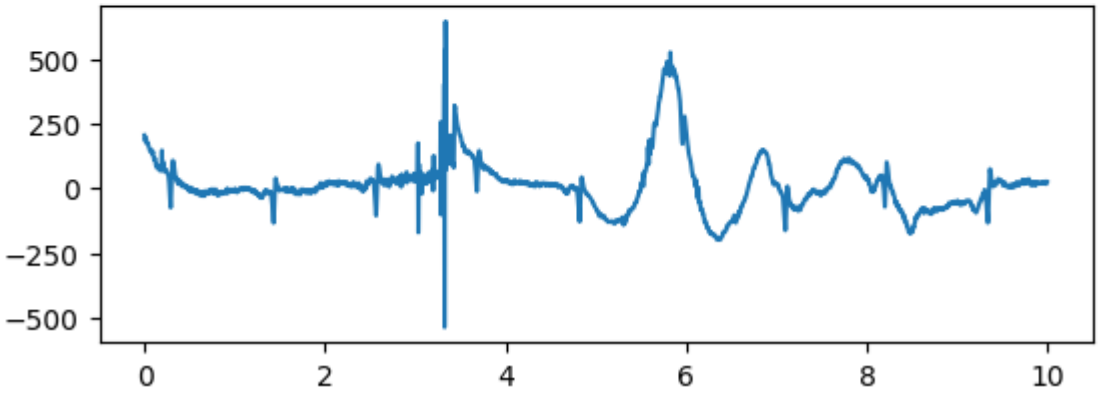

lead II
250
0
−250
−500
−750
−1000
0
2
4
6
8
10

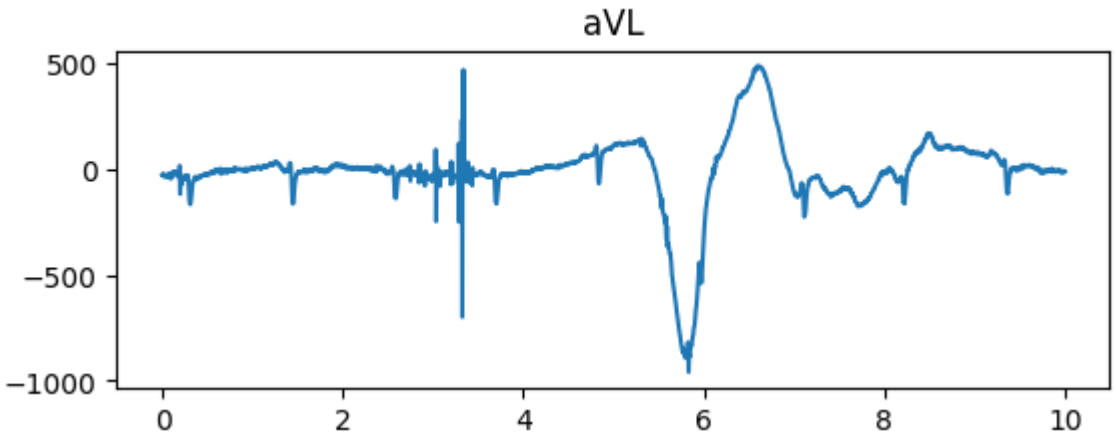

250
0
−250
−500
−750
−1000
0
2
4
6
8
10

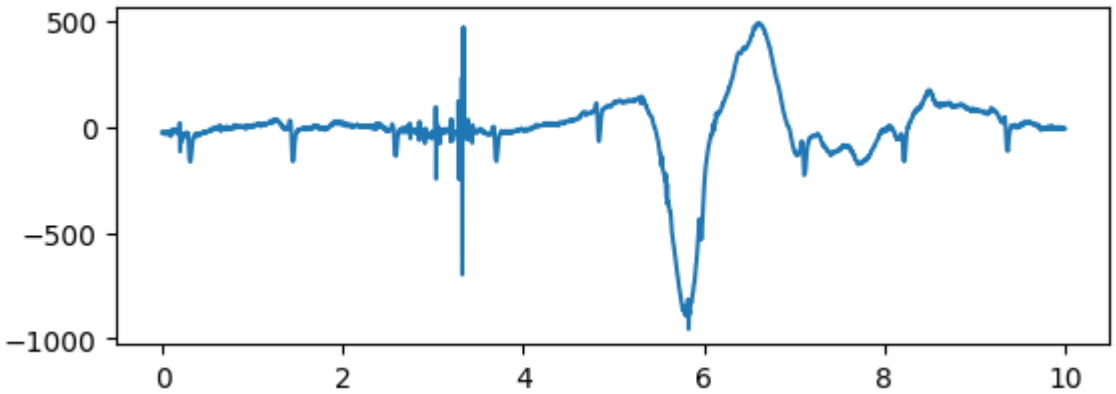

lead II
1000
500
0
−500
−1000
0
2
4
6
8
10

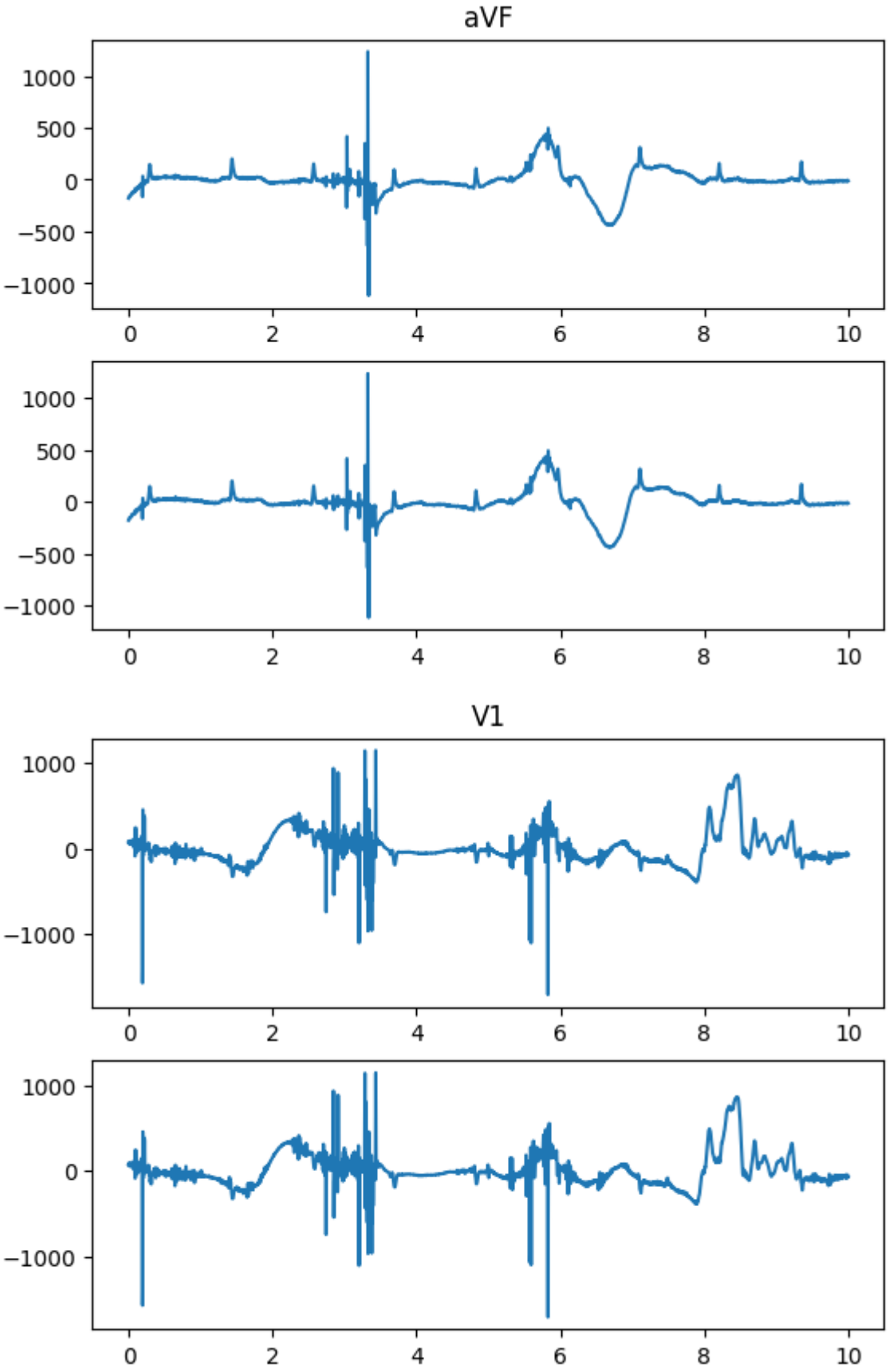

1000
500
0
−500
−1000
0
2
4
6
8
10

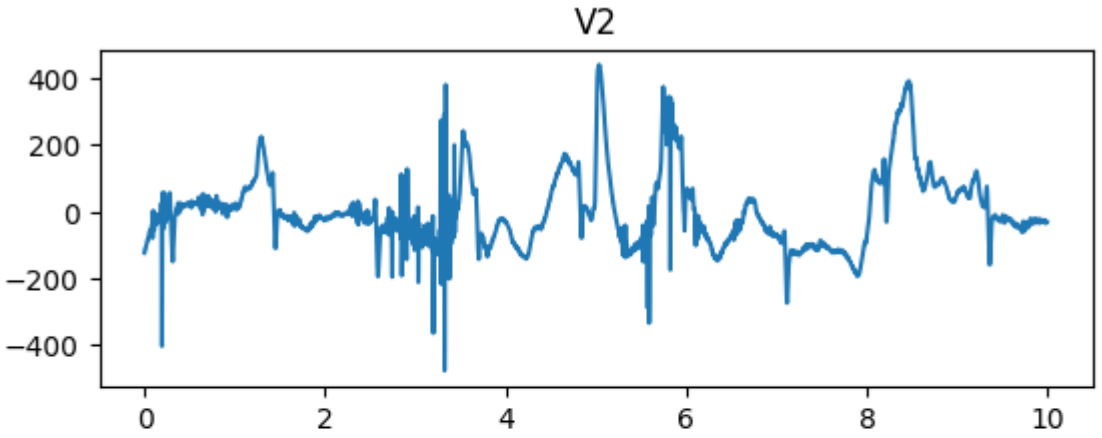

aVR
500
250
0
−250
−500
0
2
4
6
8
10

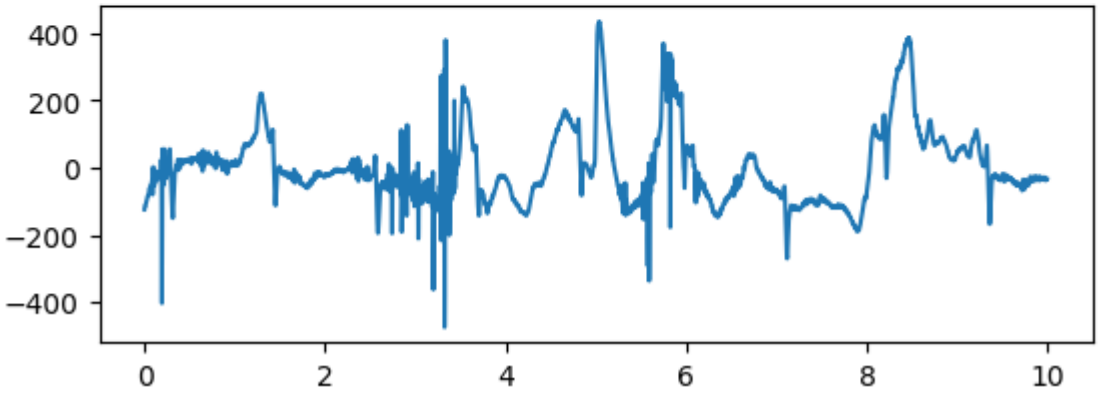

500
250
0
−250
−500
0
2
4
6
8
10

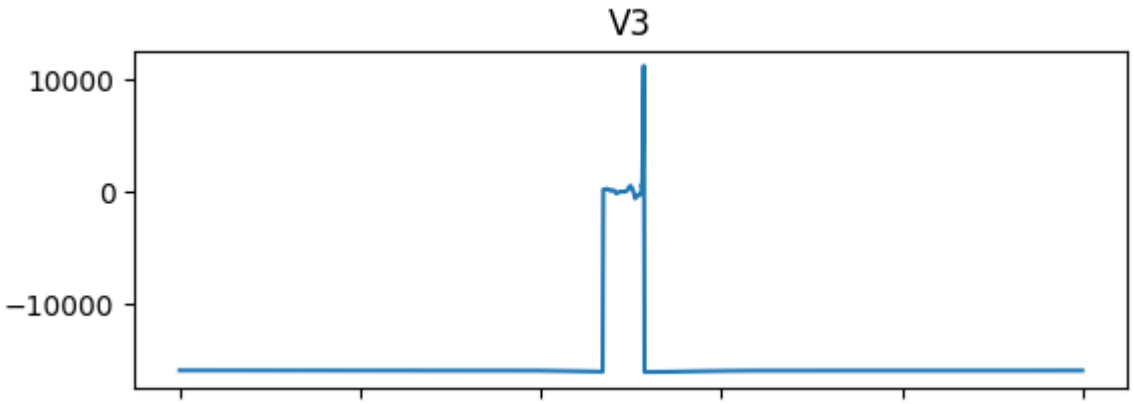

aVL
500
0
−500
−1000
0
2
4
6
8
10

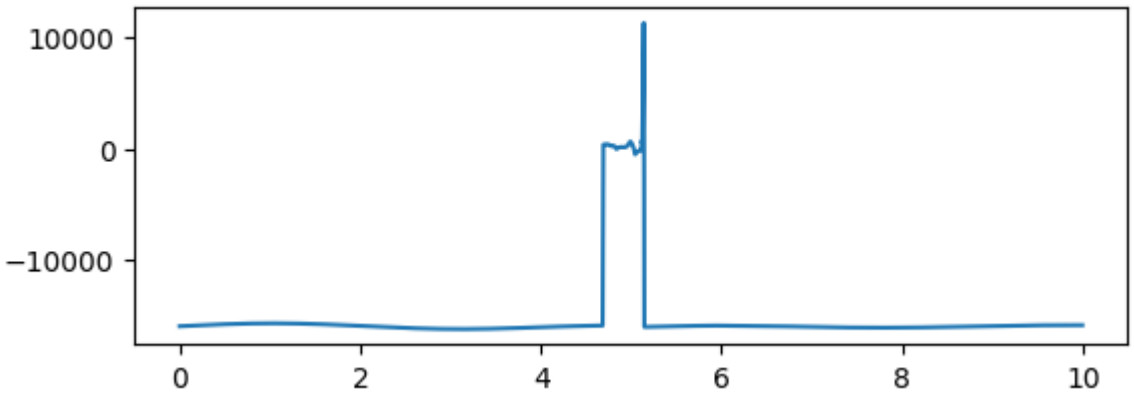

500
0
−500
−1000
0
2
4
6
8
10

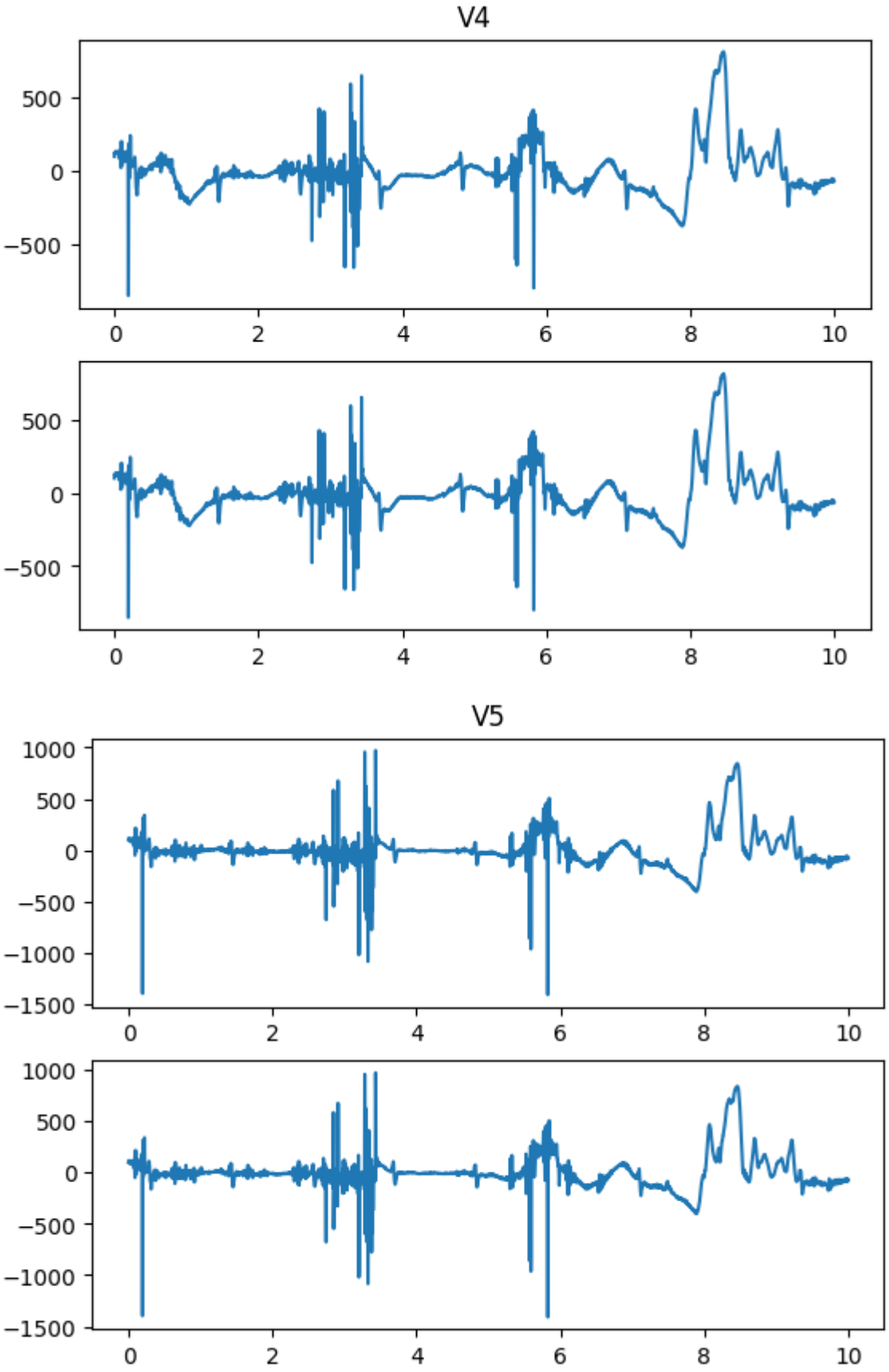

aVF
1000
500
0
−500
−1000
0
2
4
6
8
10
V1

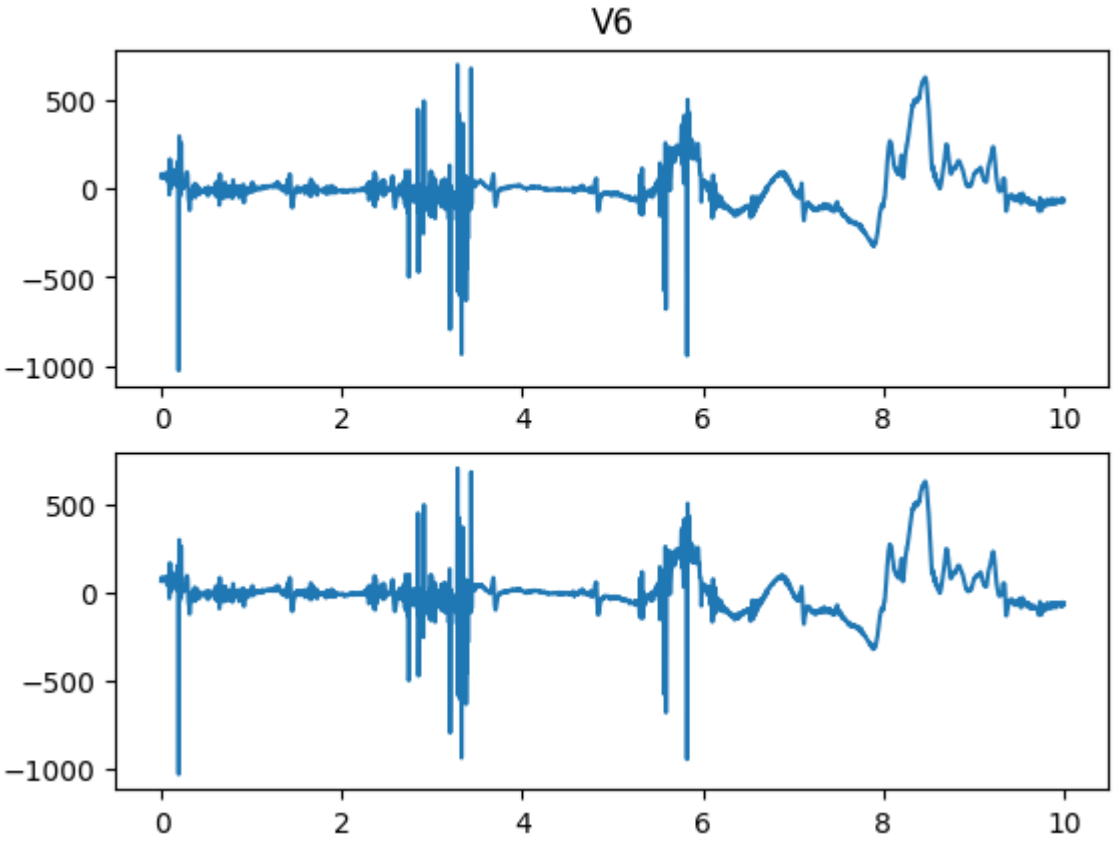

V2
400
200
0
−200
−400
0
2
4
6
8
10


400
200
0
−200
−400
0
2
4
6
8
10


V3
10000
0
−10000
0
2
4
6
8
10

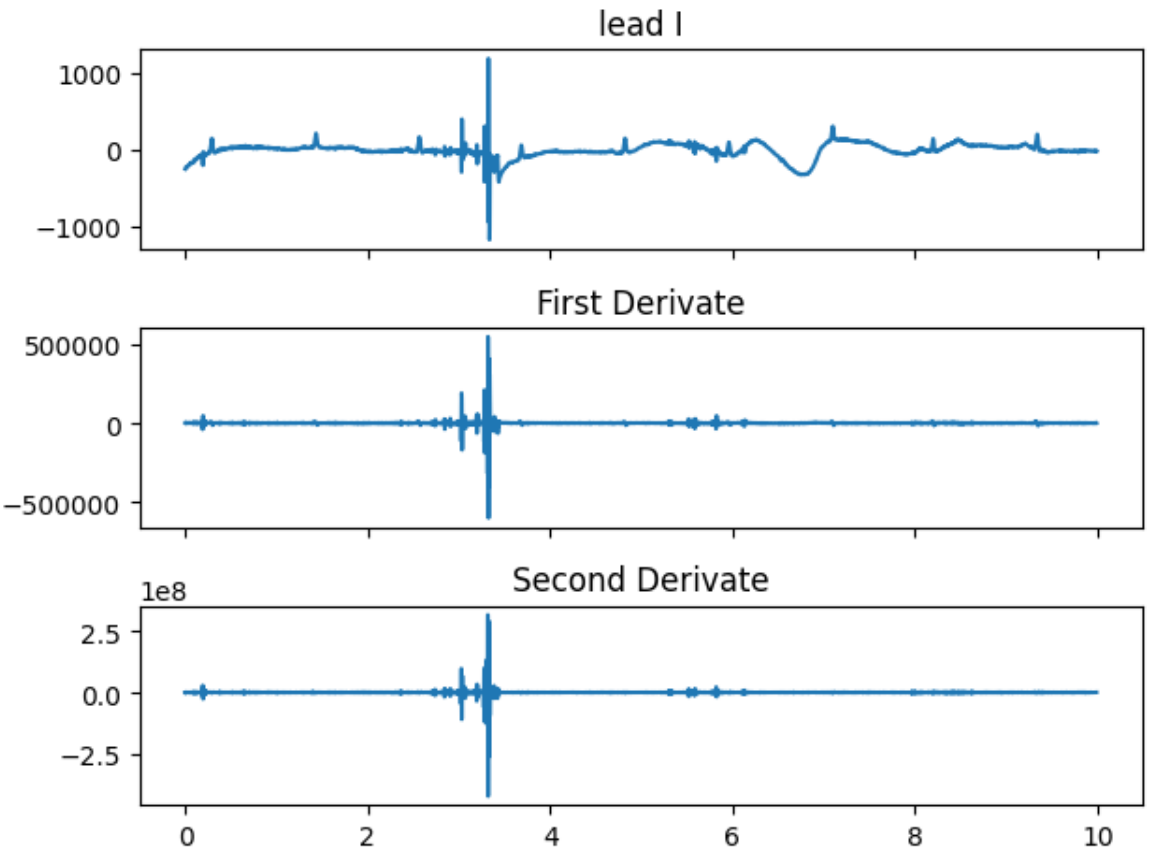

10000
0
−10000
0
2
4
6
8
10

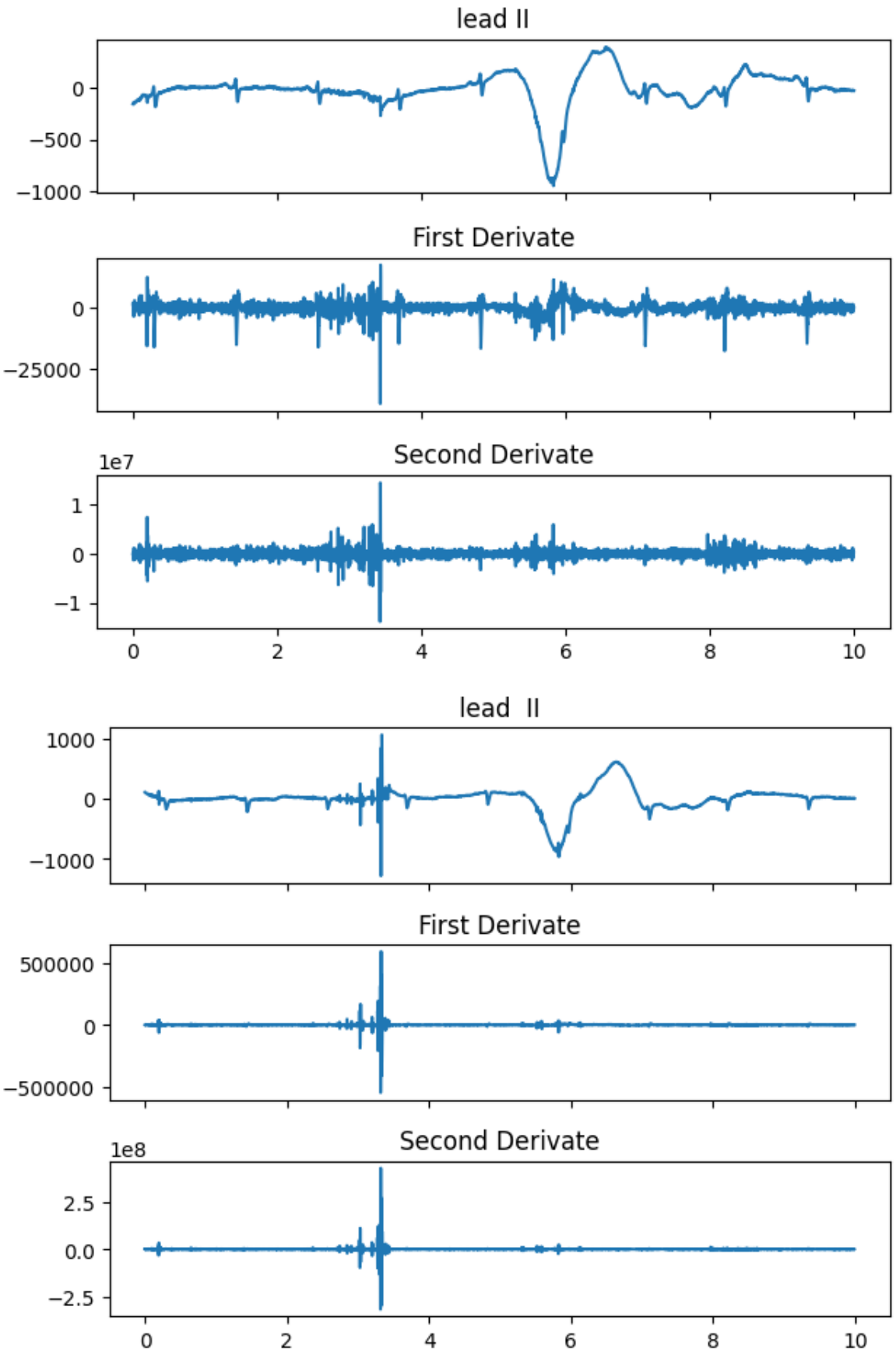

V4
500
0
−500
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10
V5
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10

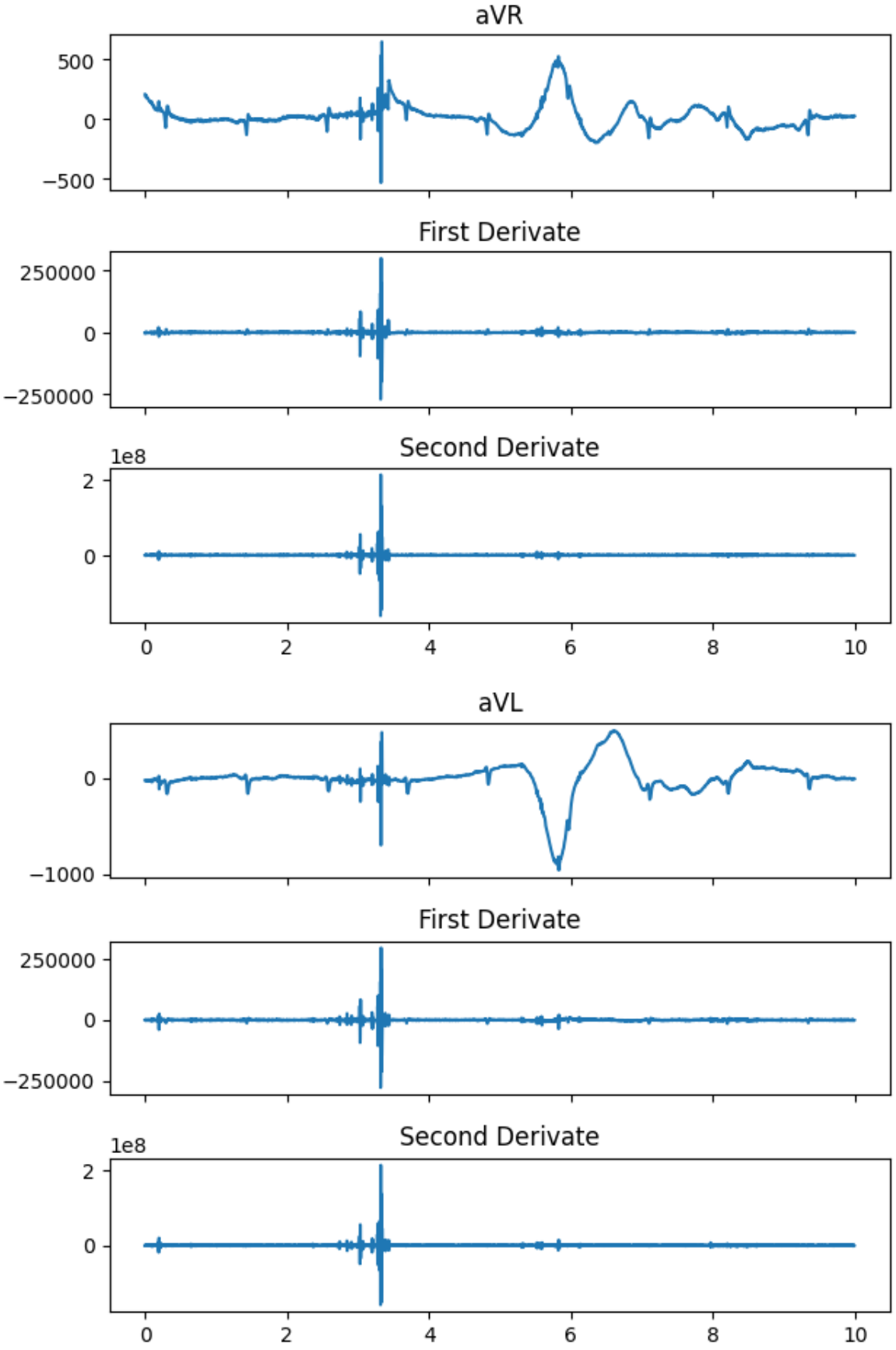


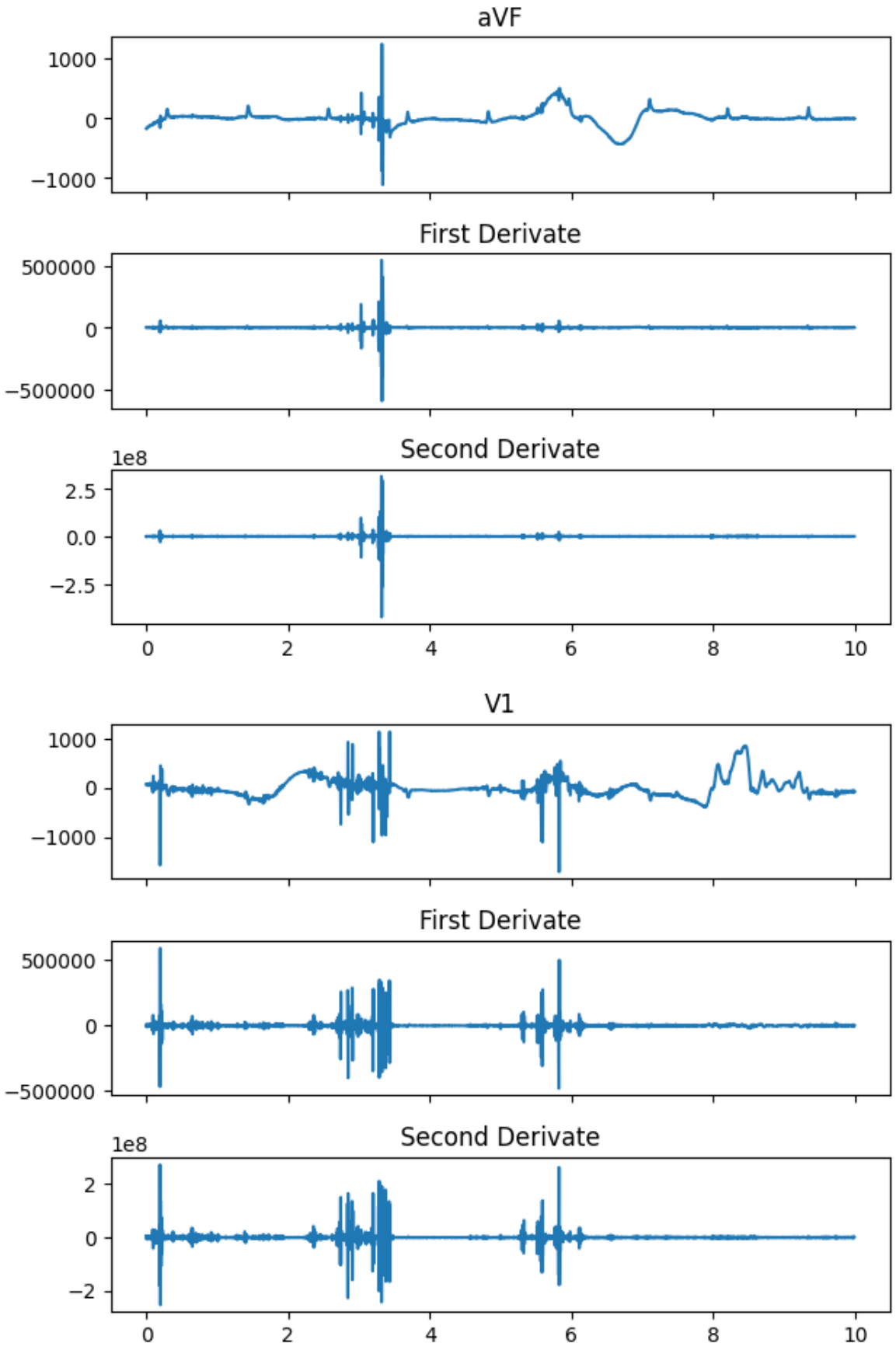
Derivadas das derivações

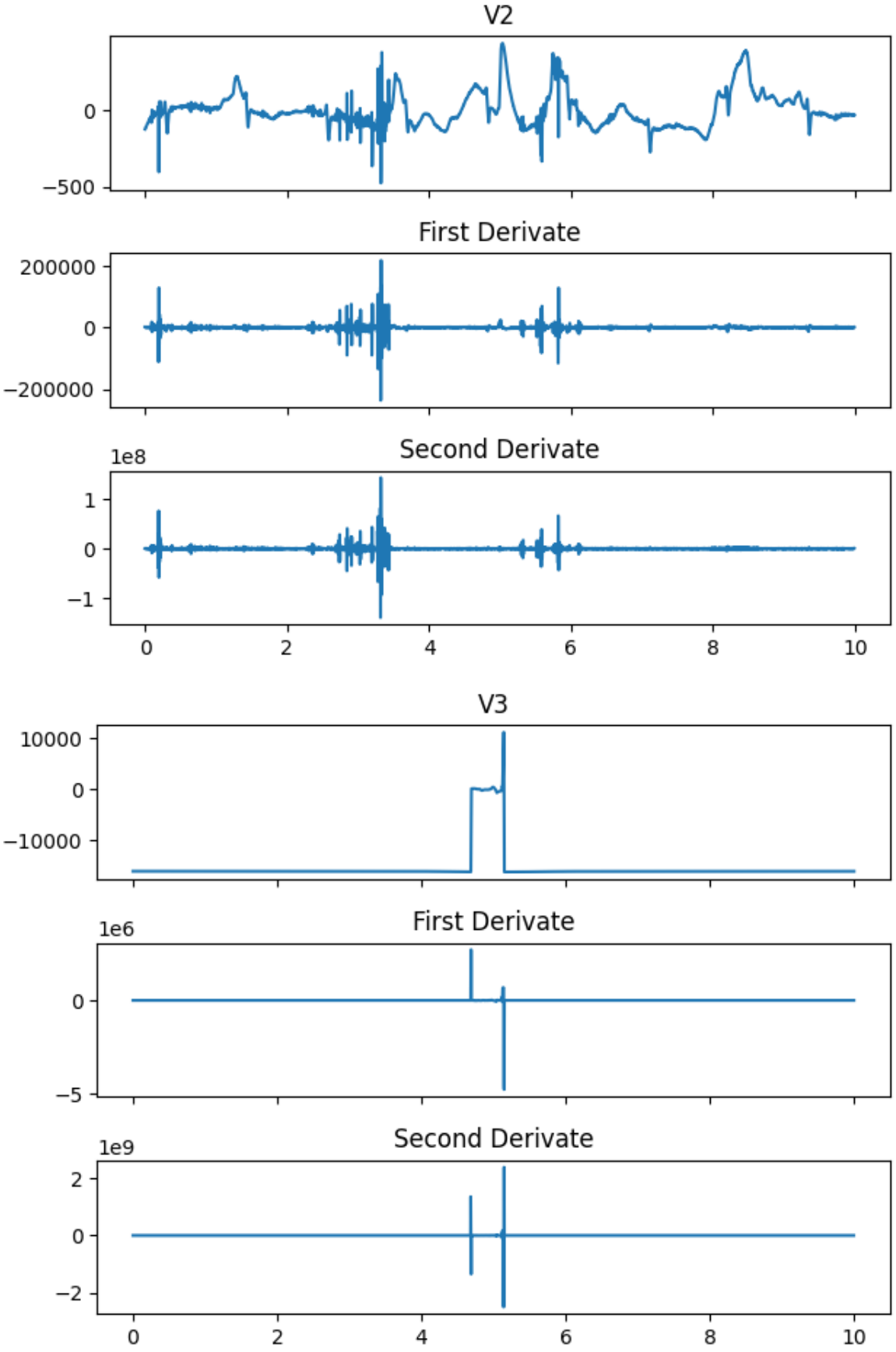
In [ ]:
```
for derivation in header[1::]:
    derivationDerivatesPlot(dataset, derivation)
```

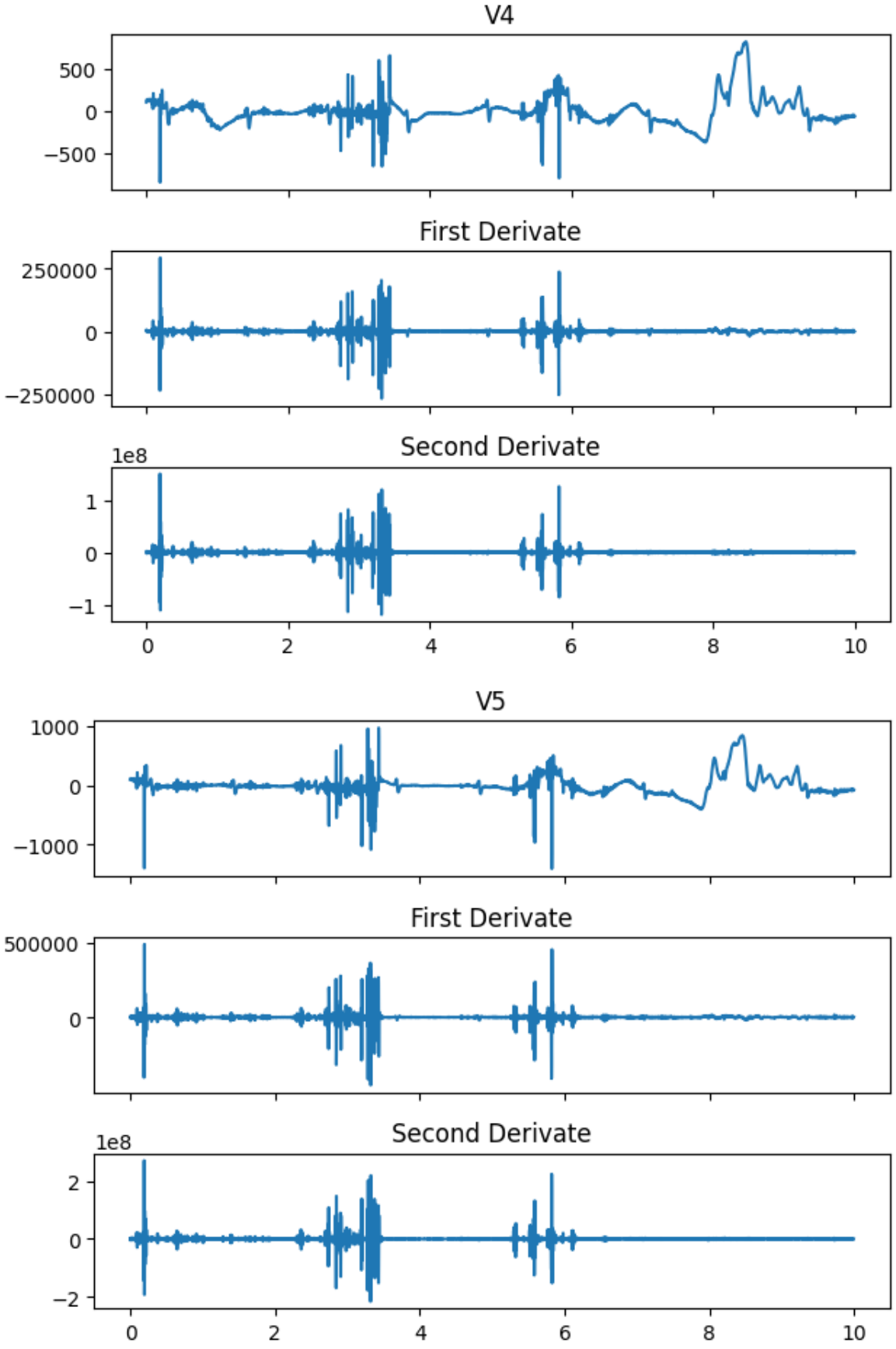


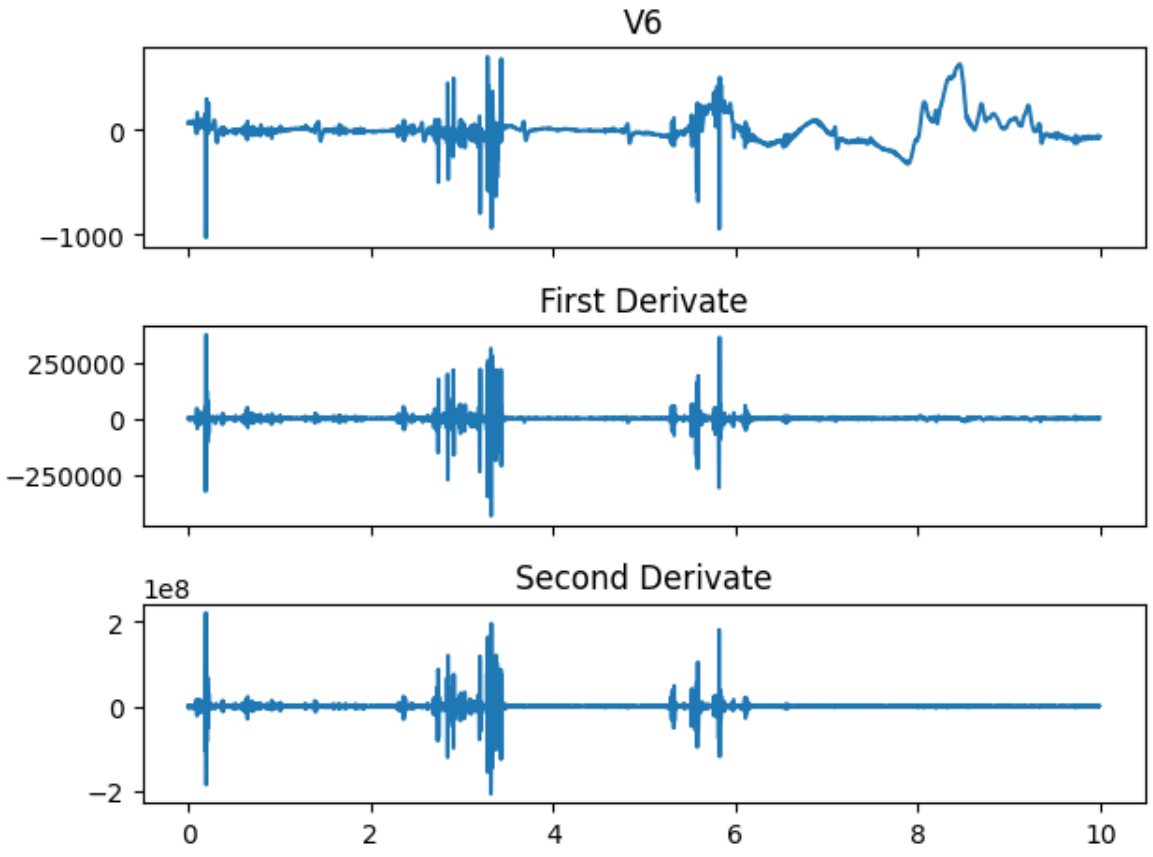

lead II
0
−500
−1000
First Derivate
0
−25000
Second Derivate
1e7
1
0
−1
0
2
4
6
8
10
lead II
1000
0
−1000
First Derivate
500000
0
−500000
Second Derivate
1e8
2.5
0.0
−2.5
0
2
4
6
8
10


aVR
500
0
−500
First Derivate
250000
0
−250000
Second Derivate
1e8
2
0
0
2
4
6
8
10
aVL
0
−1000
First Derivate
250000
0
−250000
Second Derivate
1e8
2
0
0
2
4
6
8
10


aVF
1000
0
−1000
First Derivate
500000
0
−500000
Second Derivate
1e8
2.5
0.0
−2.5
0
2
4
6
8
10
V1
1000
0
−1000
First Derivate
500000
0
−500000
Second Derivate
1e8
2
0
−2
0
2
4
6
8
10

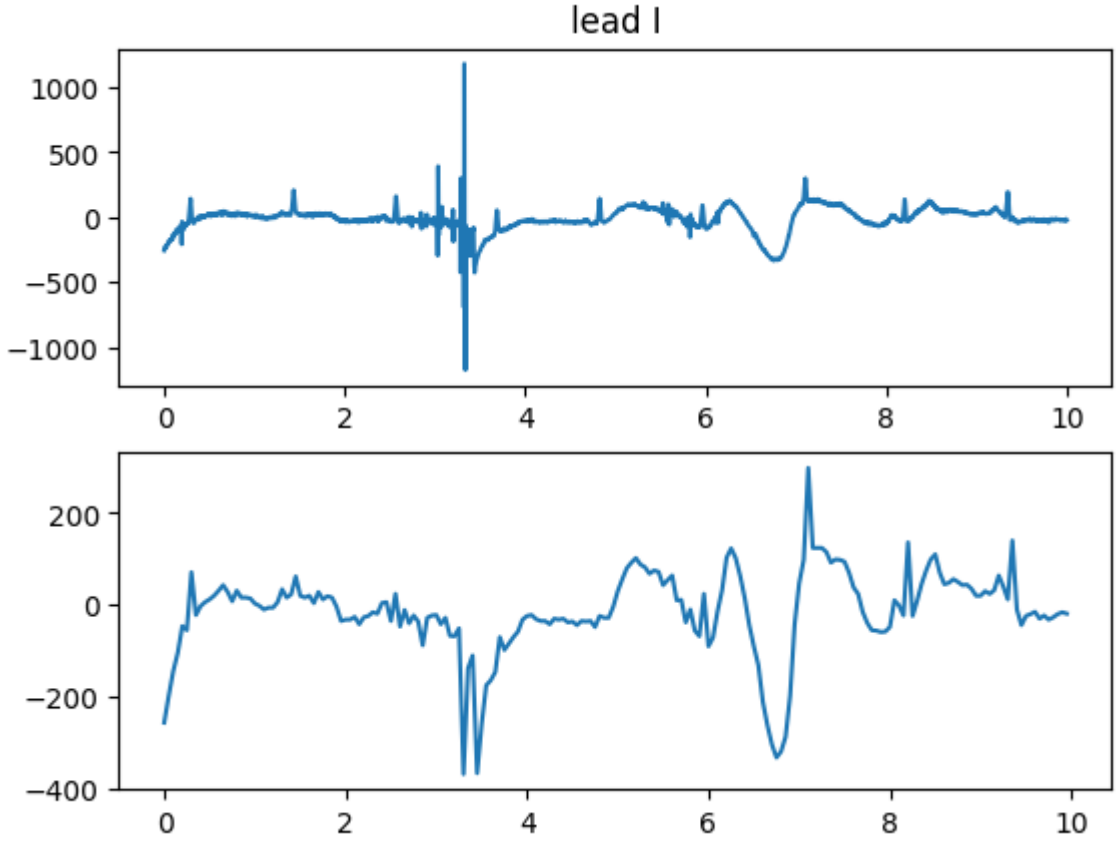

V2
0
−500
First Derivate
200000
0
−200000
Second Derivate
1e8
1
0
−1
0
2
4
6
8
10
V3
10000
0
−10000
First Derivate
1e6
0
−5
Second Derivate
1e9
2
0
−2
0
2
4
6
8
10

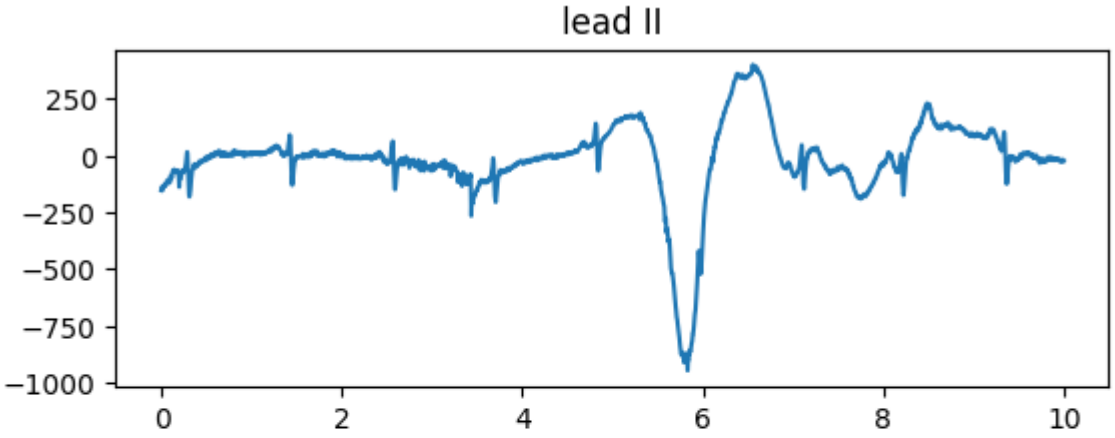

V4
500
0
−500
First Derivate
250000
0
−250000
Second Derivate
1e8
1
0
−1
0
2
4
6
8
10
V5
1000
0
−1000
First Derivate
500000
0
Second Derivate
1e8
2
0
−2
0
2
4
6
8
10

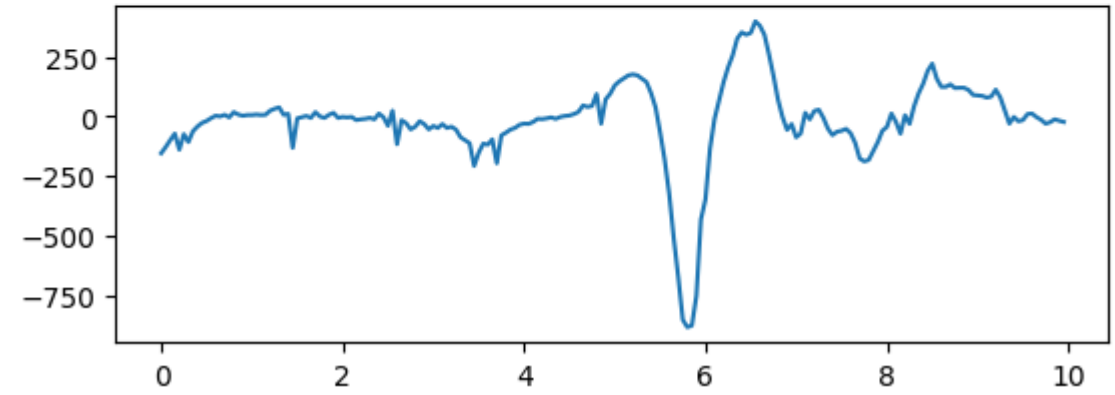


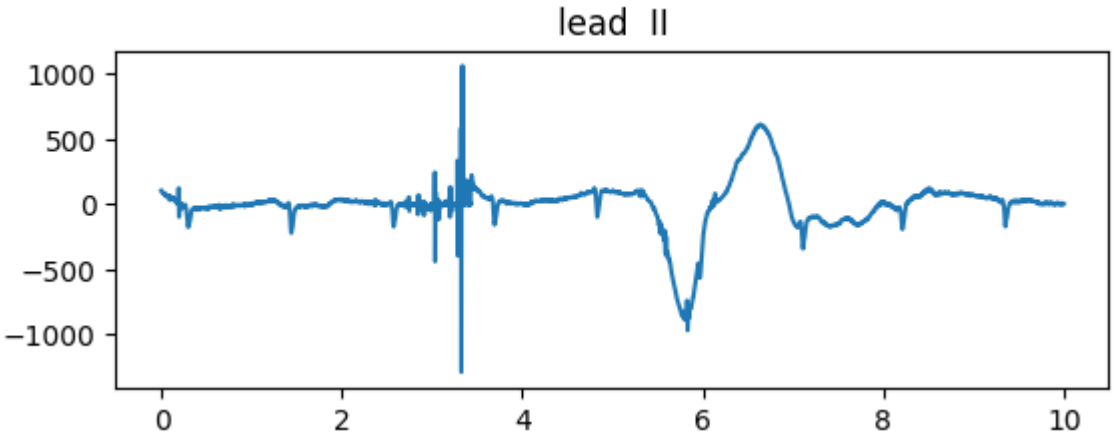
SubAmostragem das derivações com fator 25

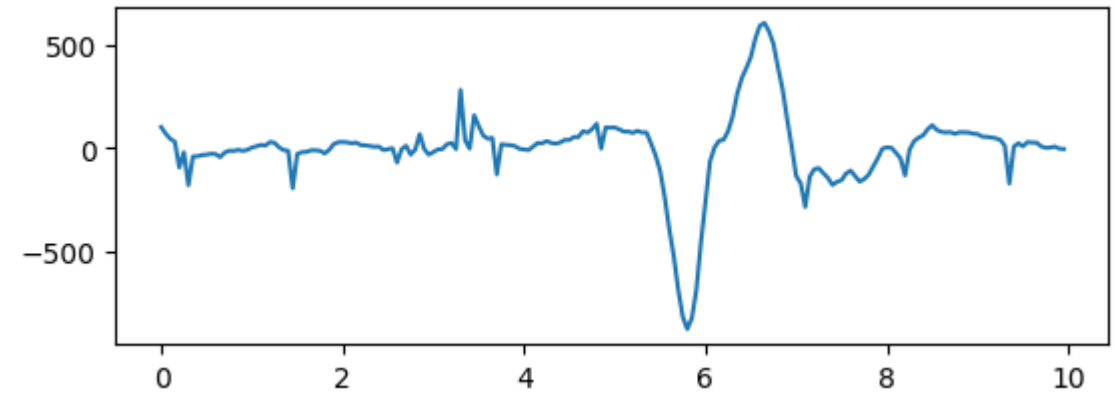
In [ ]:
```
for derivation in header[1::]:
    subSamplingPlot(dataset, derivation, 25)
```

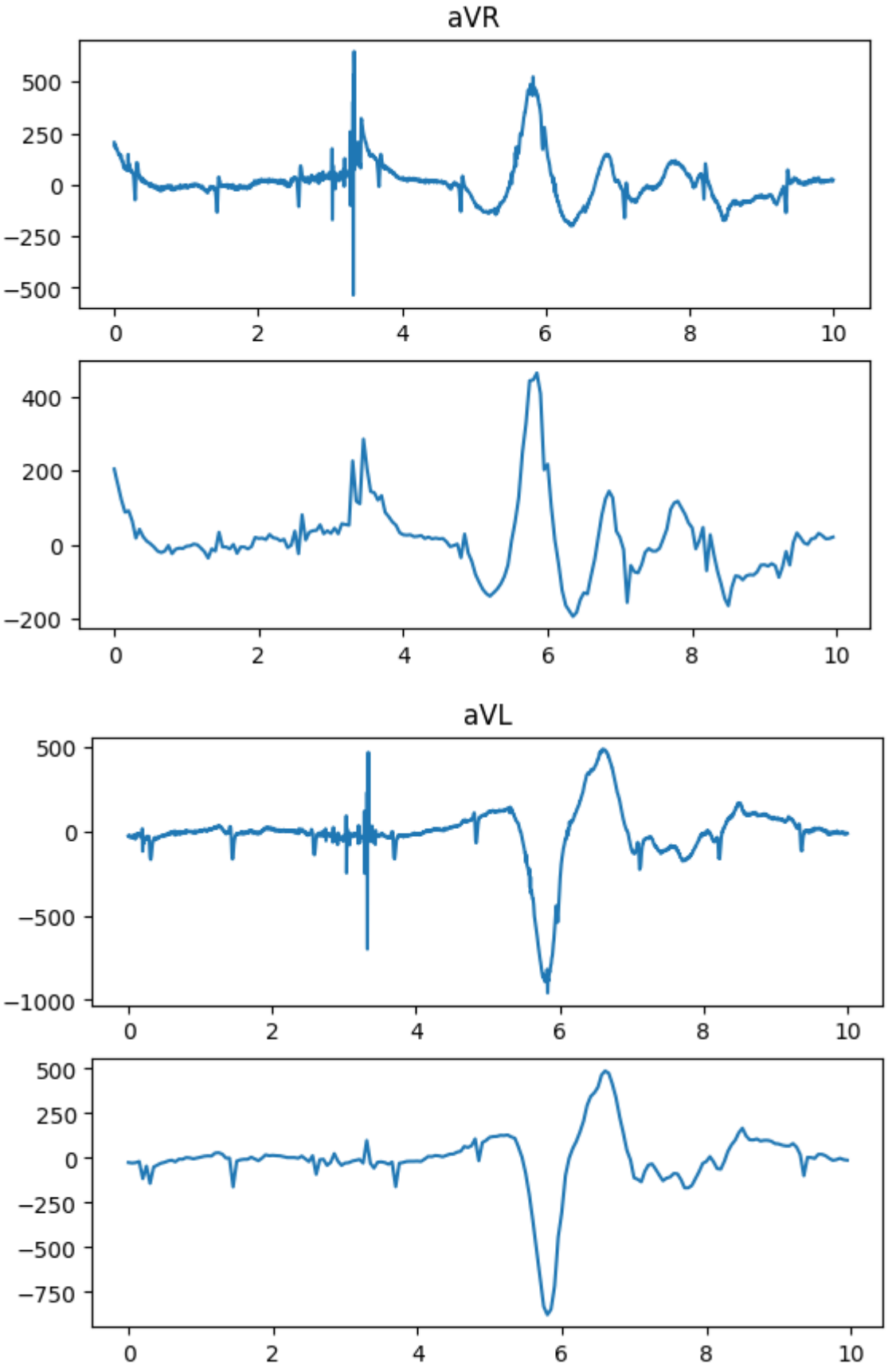


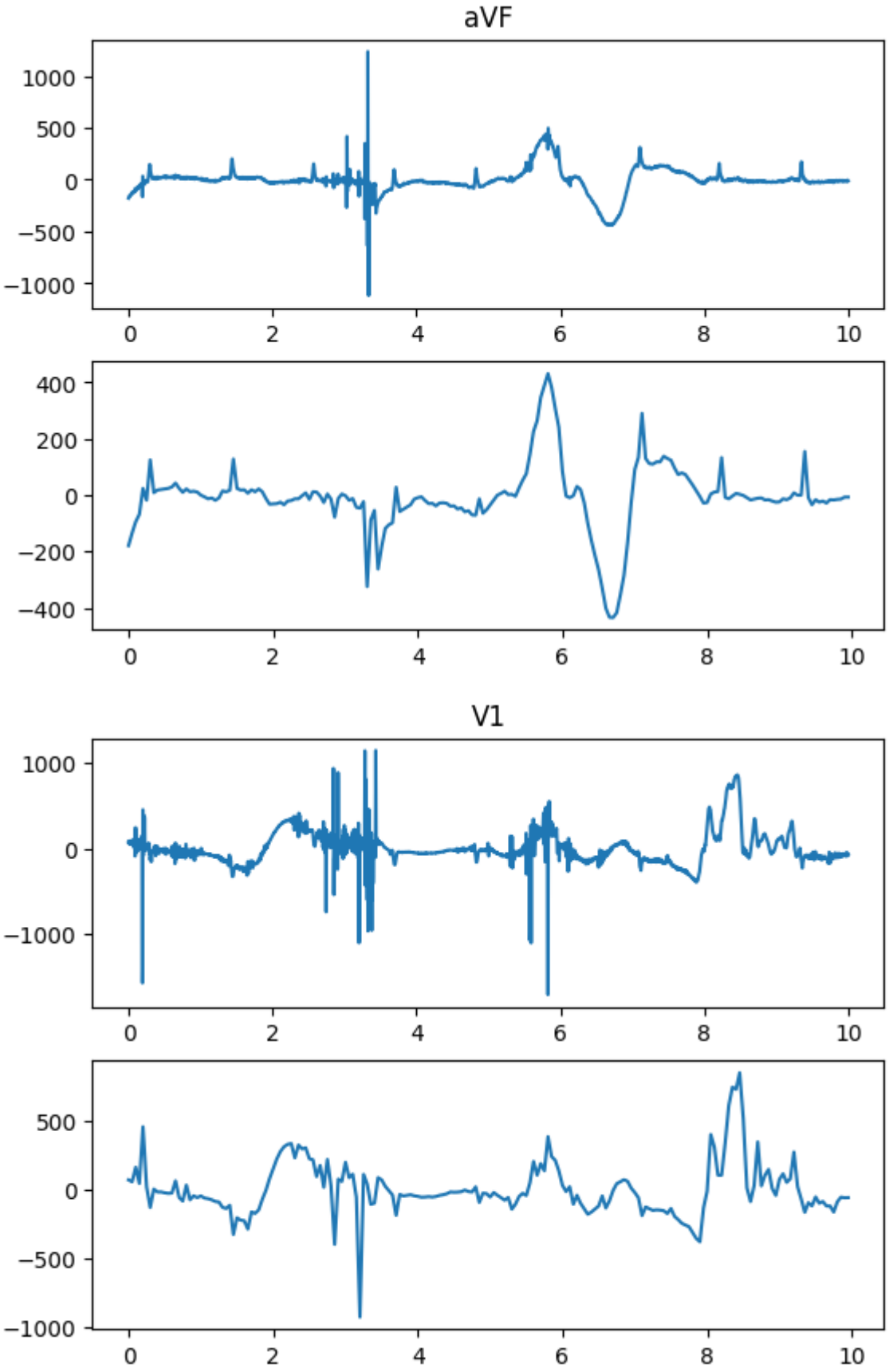

lead II
250
0
−250
−500
−750
−1000
0
2
4
6
8
10

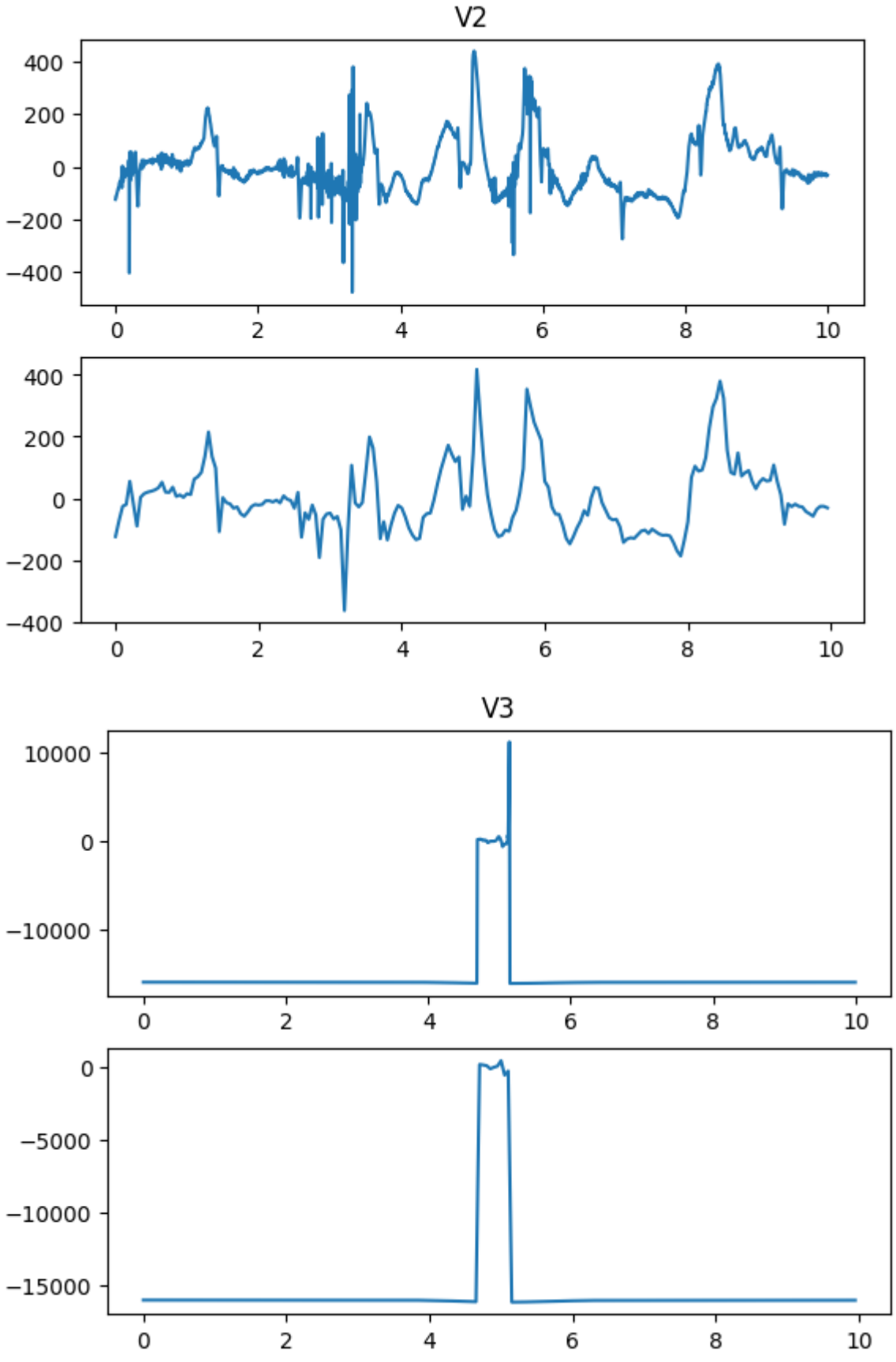

250
0
−250
−500
−750
0
2
4
6
8
10

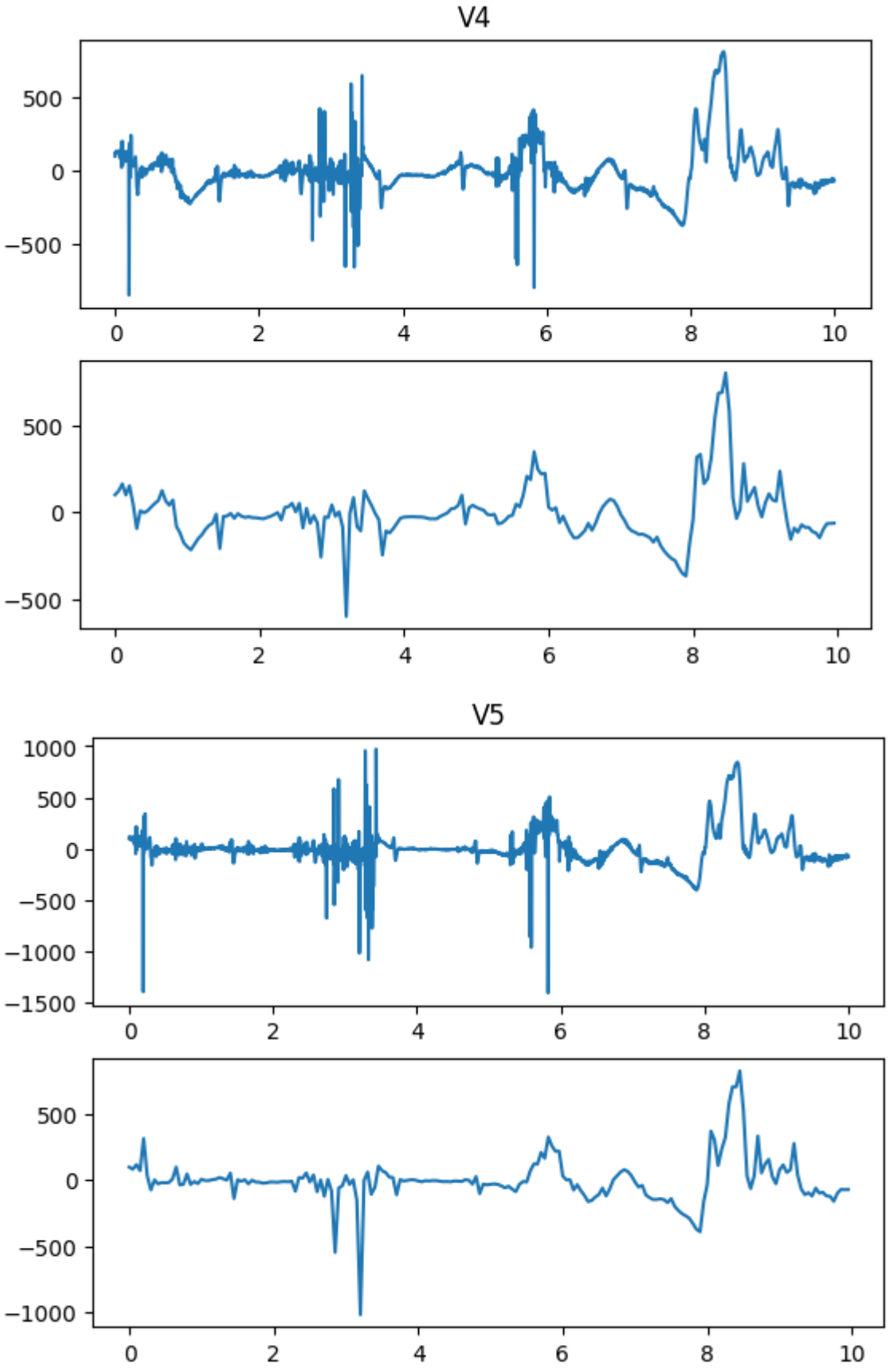

lead II
1000
500
0
−500
−1000
0
2
4
6
8
10

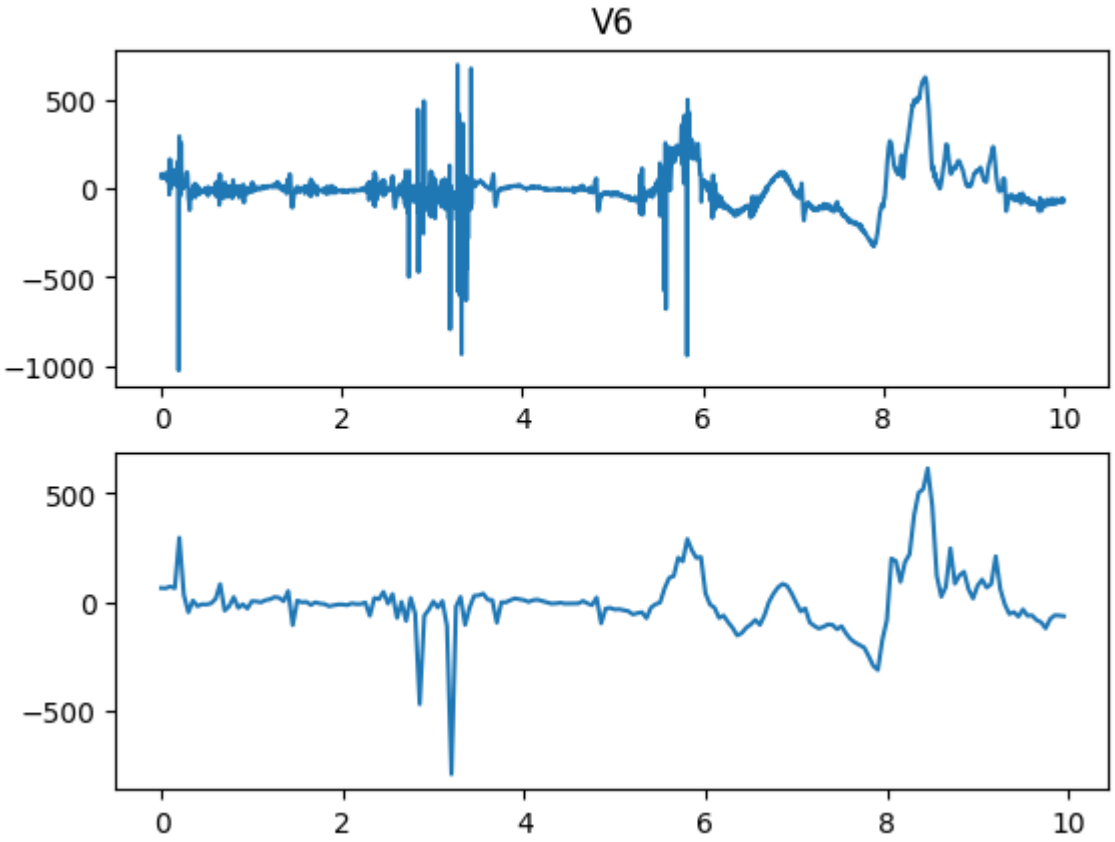

500
0
−500
0
2
4
6
8
10


aVR
500
250
0
−250
−500
0
2
4
6
8
10
400
200
0
−200
0
2
4
6
8
10
aVL
500
0
−500
−1000
0
2
4
6
8
10
500
250
0
−250
−500
−750
0
2
4
6
8
10


aVF
1000
500
0
−500
−1000
0
2
4
6
8
10
400
200
0
−200
−400
0
2
4
6
8
10
V1
1000
0
−1000
0
2
4
6
8
10
500
0
−500
−1000
0
2
4
6
8
10


V2
400
200
0
−200
−400
0
2
4
6
8
10
400
200
0
−200
−400
0
2
4
6
8
10
V3
10000
0
−10000
0
2
4
6
8
10
0
−5000
−10000
−15000
0
2
4
6
8
10


V4
500
0
−500
0
2
4
6
8
10
500
0
−500
0
2
4
6
8
10
V5
1000
500
0
−500
−1000
−1500
0
2
4
6
8
10
500
0
−500
−1000
0
2
4
6
8
10



Variâncias das derivações

In [ ]:
```
dataset[header[1::]].var()
```

Out[ ]:
```
lead I      9.075951e+03
lead II     3.110597e+04
lead  II    3.665359e+04
aVR         1.089760e+04
aVL         3.160792e+04
aVF         1.509681e+04
V1          4.039725e+04
V2          1.228149e+04
V3          1.161060e+07
V4          2.650075e+04
V5          2.981561e+04
V6          1.759974e+04
dtype: float64
```

Matriz de Correlação

In [ ]:
```
correlationMatrix = dataset[header[1::]].corr()

print(correlationMatrix)

plt.imshow(correlationMatrix, cmap='hot', interpolation='nearest')
```

	lead I	lead II	lead II	aVR	aVL	aVF
lead I	1.000000	0.104996	-0.400884	-0.543960	-0.164336	0.699694
lead II	0.104996	1.000000	0.868974	-0.891582	0.963697	-0.637026
lead II	-0.400884	0.868974	1.000000	-0.550664	0.969552	-0.935015
aVR	-0.543960	-0.891582	-0.550664	1.000000	-0.738306	0.218883
aVL	-0.164336	0.963697	0.969552	-0.738306	1.000000	-0.819710
aVF	0.699694	-0.637026	-0.935015	0.218883	-0.819710	1.000000
V1	-0.010549	-0.046913	-0.037968	0.044355	-0.043704	0.025375
V2	-0.013972	-0.099203	-0.084435	0.090593	-0.094822	0.059848
V3	0.057647	0.133928	0.094691	-0.139110	0.117402	-0.051663
V4	0.023367	-0.051607	-0.059169	0.033175	-0.057612	0.054848
V5	0.020459	-0.055649	-0.061446	0.037935	-0.060842	0.055482
V6	0.011424	-0.117435	-0.113868	0.094225	-0.119722	0.092815

	V1	V2	V3	V4	V5	V6
lead I	-0.010549	-0.013972	0.057647	0.023367	0.020459	0.011424
lead II	-0.046913	-0.099203	0.133928	-0.051607	-0.055649	-0.117435
lead II	-0.037968	-0.084435	0.094691	-0.059169	-0.061446	-0.113868
aVR	0.044355	0.090593	-0.139110	0.033175	0.037935	0.094225
aVL	-0.043704	-0.094822	0.117402	-0.057612	-0.060842	-0.119722
aVF	0.025375	0.059848	-0.051663	0.054848	0.055482	0.092815
V1	1.000000	0.564386	-0.032743	0.859754	0.872864	0.832745
V2	0.564386	1.000000	0.259094	0.705962	0.700882	0.729846
V3	-0.032743	0.259094	1.000000	0.039717	-0.033124	-0.048946
V4	0.859754	0.705962	0.039717	1.000000	0.945876	0.925868
V5	0.872864	0.700882	-0.033124	0.945876	1.000000	0.980664
V6	0.832745	0.729846	-0.048946	0.925868	0.980664	1.000000

Out[ ]: <matplotlib.image.AxesImage at 0x727da222b590>