Medical Informatics-Cardiogram
Assignment 3

Vasileios Nastos
MSc in department of Informatics
and Telecommunications
University of Ioannina
Arta, 47100, Greece
Id:137

May 17, 2023

1 Introduction

Electrocardiography (ECG) is a fundamental diagnostic technique used in the field of cardiology to assess the electrical activity of the heart. By measuring and recording the electrical signals produced by the heart during each heartbeat, ECG provides valuable insights into the heart's health and functioning. An electrocardiogram (ECG) is the graphical representation of the electrical activity recorded by an ECG machine. It displays a series of waves and intervals that correspond to specific events occurring during the cardiac cycle. These events include the depolarization and repolarization of the heart's chambers, which are crucial for maintaining a regular heartbeat. Correlation analysis, in the context of ECG, refers to the statistical method used to measure the relationship or association between different variables within the ECG signal. By examining the similarities or dissimilarities between various waves, intervals, or segments of the ECG, correlation analysis can provide valuable insights into the overall cardiac health and the presence of any abnormalities. Correlation analysis in ECG can be employed in multiple ways. Firstly, it can be used to compare an individual's ECG waveform with a standard or normal waveform, helping healthcare professionals identify any deviations or irregularities. Such comparisons can aid in the diagnosis of various cardiac conditions, including arrhythmias, ischemia, or myocardial infarction. Furthermore, correlation analysis allows for the detection of specific patterns or changes in the ECG signal that may indicate the presence of cardiac abnormalities. By examining the consistency, coherence, and timing of various ECG features, healthcare professionals can gain insights into the underlying cardiac conditions and make informed decisions regarding treatment and management. In conclusion, ECG is a vital diagnostic tool that provides valuable information about the electrical activity of the heart. The electrocardiogram serves as a graphical representation of this activity, allowing for visual analysis. Correlation analysis plays a crucial role in interpreting the ECG signal by assessing relationships between different variables within the waveform. By employing correlation techniques, healthcare professionals can detect abnormalities, diagnose cardiac conditions, and provide appropriate medical interventions for patients.

In this assignment the experiments will be performed in a small subset of an ecg signal. In table 1 some statistics are presented about the correspondent signal while in figure 1 the signal is visualized.

Statistic	Value
Samples	250.000
Mean	-0.222
Median	-0.311
Std	0.181
Iqr	0.035
Skewness	1.038
Kurtosis	-0.392

Table 1: Signal stats

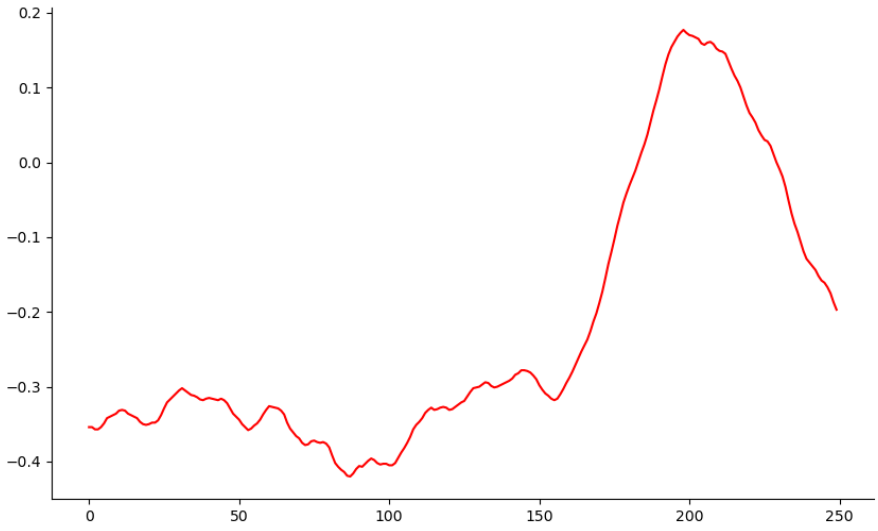


Figure 1: Label Visualization

2 Code

Listing 1: Full assingment code

```
import os,numpy as np,statistics
import matplotlib.pyplot as plt
from tabulate import tabulate
from scipy.stats import skew,kurtosis


def dft(signal):
    N=len(signal)

    RFx=np.zeros(N)
    IFx=np.zeros(N)

    for k in range(N):
        for n in range(N):
            RFx[k]+=signal[n] * np.cos(2 * np.pi * k * n/N)
            IFx[k]-=signal[n] * np.sin(2 * np.pi * k * n/N)
    return RFx,IFx

def idft(RFx,IFx):
    N=len(RFx)

    Ry=np.zeros(N)
```

```
    Iy=np.zeros(N)

    for n in range(N):
        for k in range(N):
            Ry[n]+=RFx[k] * np.cos(2 * np.pi * k * n / N)
            Iy[n]+=IFx[k] * np.sin(2 * np.pi * k * n /N)
    return Ry,Iy

def convolution_freq(x,y):
    N = max(len(x), len(y))
    x_padded = np.pad(x, (0, N - len(x)), mode='constant')
    h_padded = np.pad(y, (0, N - len(y)), mode='constant')

    X = np.fft.fft(x_padded)
    H = np.fft.fft(h_padded)

    y= X * H
    return np.real(np.fft.ifft(y))


def correlation(x, y):
    # Pad the signals with zeros to match their lengths
    N = max(len(x), len(y))
    x_padded = np.pad(x, (0, N - len(x)), mode='constant')
    y_padded = np.pad(y, (0, N - len(y)), mode='constant')

    # Calculate correlation using the convolution of x and reversed y
    z = np.flip(y_padded) # Reverse the second signal
    r = np.convolve(x_padded, z)

    return r


class ECG:
    def __init__(self) -> None:
        self.ecg_signal=np.loadtxt(os.path.join('','ecg.txt'))
        self.sliced_signal=self.ecg_signal[50:300]

    def plot_signal(self):
        plt.figure(figsize=(10,6))
        plt.plot(np.arange(len(self.sliced_signal)),self.sliced_signal,color='r')
        ax=plt.gca()
        ax.spines['right'].set_visible(False)
        ax.spines['top'].set_visible(False)
        plt.savefig(os.path.join('','figures','singal.png'))
        plt.show()

    def statistics(self):
        q1,q3,_=statistics.quantiles(data=self.sliced_signal,n=4)
        rows=[
            ['Samples',self.sliced_signal.shape[0]],
            ['Mean',statistics.mean(self.sliced_signal)],
            ['Median',statistics.median(self.sliced_signal)],
            ['Std',statistics.stdev(self.sliced_signal)],
            ['Iqr',q3-q1],
            ['Skewness',skew(self.sliced_signal)],
            ['Kurtosis',kurtosis(self.sliced_signal)]
        ]
        print(tabulate(tabular_data=rows,headers=['Statistic Meter','Value'],tablefmt='fancy_grid',
             floatfmt='.3f'))
        with open(os.path.join('stats.tex'),'w') as writer:
            writer.write(tabulate(tabular_data=rows,headers=['Statistic Meter','Value'],tablefmt='latex',
                 floatfmt='.3f'))


    def normalize_carediogram(self,shock_response):
        signal_fft=np.fft.fft(self.sliced_signal)
        shock_response_fft=np.fft.fft(shock_response,len(self.sliced_signal))
        smoothed_signal_fft=signal_fft * shock_response_fft
        return np.real(np.fft.ifft(smoothed_signal_fft))
```

```
    def congruence_in_the_frequency_domain(self,shock_response):
        signal_copy=self.sliced_signal
        if len(self.sliced_signal)<len(shock_response):
            signal_copy=np.pad(signal_copy,(0,len(shock_response)-len(signal_copy)),'constant')
        elif len(self.sliced_signal)>len(shock_response):
            shock_response=np.pad(shock_response,(0,len(self.sliced_signal)-len(shock_response)),'
                constant')

        N=len(signal_copy)
        signal_copy=np.concatenate((np.zeros(N),signal_copy))
        shock_response=np.concatenate((shock_response,np.zeros(N)))

        RFx,IFx=dft(signal_copy)
        RFsh,IFsh=dft(shock_response)

        RFy=RFx * RFsh - IFx * IFsh
        IFy= RFx * IFsh + IFx *RFsh

        Ry,Iy=idft(RFy,IFy)
        freq_signal=len(signal_copy) * Ry
        return freq_signal

    def plot_sliced_and_normalized(self):
        normalized_cardiogram=self.normalize_caredіogram(np.array([0.25,0.5,0.25]))
        plt.figure(figsize=(10,6))
        plt.plot(np.arange(len(self.sliced_signal)),self.sliced_signal,color='b')
        plt.plot(np.arange(len(normalized_cardiogram)),normalized_cardiogram,color='r')
        ax=plt.gca()
        ax.spines['right'].set_visible(False)
        ax.spines['top'].set_visible(False)
        plt.xticks(np.arange(0,300,20))
        plt.savefig(os.path.join('','figures','normalized_cardiogram.png'))
        plt.show()

    def plot_sliced_and_correlated(self):
        shifted_signal=self.congruence_in_the_frequency_domain(np.array([0.25,0.5,0.25]))
        plt.figure(figsize=(10,6))
        plt.plot(np.arange(len(self.sliced_signal)),self.sliced_signal,color='b')
        plt.plot(np.arange(len(shifted_signal)),shifted_signal,color='r')
        ax=plt.gca()
        ax.spines['right'].set_visible(False)
        ax.spines['top'].set_visible(False)
        # plt.xticks(np.arange(0,300,20))
        plt.savefig(os.path.join('','figures','shifted_cardiogram.png'))
        plt.show()
# Answer-Question 2
# When doubling the length of the two signals by multiplying by zero and padding with zeros, the signals
     need to be shifted to ensure correct convolution using the correlation theorem.
# By convention, when performing convolution in the time domain, the output signal is obtained by
     shifting one of the signals and flipping it. This is necessary to
# align the signals properly for the multiplication and summation operations.
# When doubling the length of the signals, one signal is incremented to the right by padding zeros at
     the beginning. This rightward shift ensures that when the signals are flipped during convolution,
     the proper alignment is maintained with the other signal.
# On the other hand, the other signal is incremented to the left by padding zeros at the end. This
     leftward shift ensures that after flipping, the signal is correctly aligned with the shifted signal
      during convolution.
# In summary, the leftward and rightward shifts of the signals when doubling their length with zeros
     ensure that the signals are properly aligned for correct convolution using the correlation theorem.

    def plot_sliced_convolution_freq(self):
        shock_response=np.array([0.25,0.5,0.25])
        y_freq=convolution_freq(self.sliced_signal,shock_response)
        y_corr=correlation(self.sliced_signal,shock_response)

        print('Output from convolution freq:')
        print(y_freq,end='\n\n')

        print('Output from convolution corr:')
        print(y_corr)
```

```
        plt.figure(figsize=(10,6))
        plt.plot(np.arange(len(y_freq)),y_freq,color='b')
        plt.plot(np.arange(len(y_corr)),y_corr,color='r')
        ax=plt.gca()
        ax.spines['right'].set_visible(False)
        ax.spines['top'].set_visible(False)
        plt.savefig(os.path.join('','figures','compare_correlation_in_sliced_cardiogram.png'))
        plt.show()
# Answer-Question 3
# It seems that both the frequency-domain convolution and correlation operations result in the same
    output, except for the first few elements which are zero in the correlation output.
# This is expected because convolution in the frequency domain is equivalent to correlation in the time
    domain, with the difference being the arrangement of the input signals.
# The convolution result is a sequence of values that represents the overlap between the two signals as
    one is slid over the other.
# In this case, it appears that the convolution output gradually decreases in magnitude.

if __name__=='__main__':
    ecg_signal=ECG()
    ecg_signal.plot_signal()
    ecg_signal.statistics()
    ecg_signal.plot_sliced_and_normalized()
    ecg_signal.plot_sliced_and_correlated()
    ecg_signal.plot_sliced_convolution_freq()
```

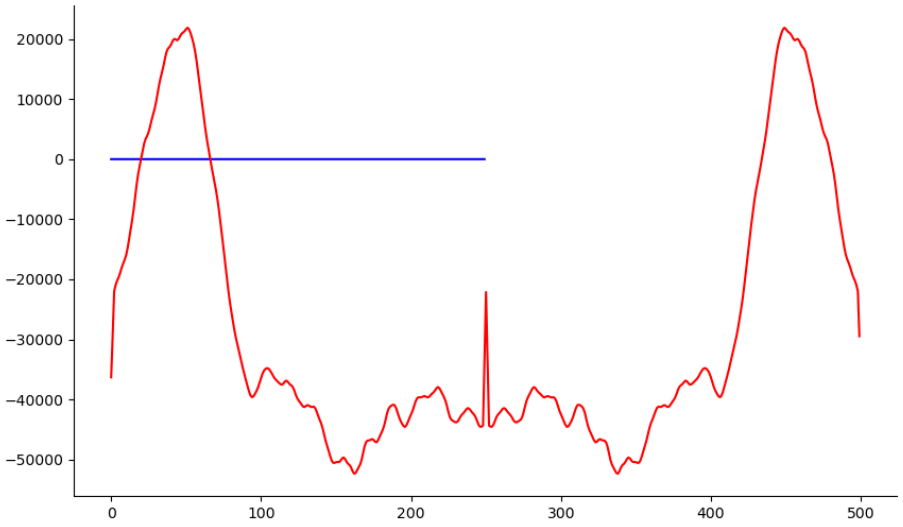
3 Results

3.1 Part 1

3.2 Part 2

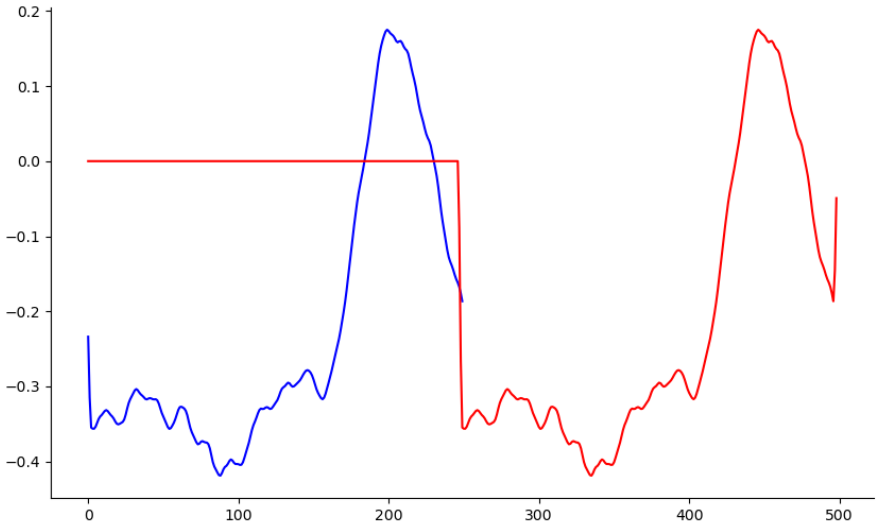
When doubling the length of the two signals by multiplying by zero and padding with zeros, the signals need to be shifted to ensure correct convolution using the correlation theorem. By convention, when performing convolution in the time domain, the output signal is obtained by shifting one of the signals and flipping it. This is necessary to align the signals properly for the multiplication and summation operations. When doubling the length of the signals, one signal is incremented to the right by padding zeros at the beginning. This rightward shift ensures that when the signals are flipped during convolution, the proper alignment is maintained with the other signal. On the other hand, the other signal is incremented to the left by padding zeros at the end. This leftward shift ensures that

after flipping, the signal is correctly aligned with the shifted signal during convolution. In summary, the leftward and rightward shifts of the signals when doubling their length with zeros ensure that the signals are properly aligned for correct convolution using the correlation theorem.



3.3 Part 3

It seems that both the frequency-domain convolution and correlation operations result in the same output, except for the first few elements which are zero in the correlation output. This is expected because convolution in the frequency domain is equivalent to correlation in the time domain, with the difference being the arrangement of the input signals. The convolution result is a sequence of values that represents the overlap between the two signals as one is slid over the other. In this case, it appears that the convolution output gradually decreases in magnitude.



```
Output from convolution freq:
[-0.23375 -0.31475 -0.35475 -0.35625 -0.35625 -0.3535  -0.3485  -0.34325
 -0.34    -0.338   -0.3355  -0.33275 -0.3315  -0.33275 -0.3355  -0.338
 -0.34    -0.34275 -0.3465  -0.3495  -0.3505  -0.34975 -0.3485  -0.34725
 -0.344   -0.3375  -0.32925 -0.322   -0.317   -0.313   -0.309   -0.30525
 -0.3035  -0.305   -0.308   -0.3105  -0.31225 -0.31425 -0.3165  -0.31725
 -0.31625 -0.3155  -0.316   -0.317   -0.31725 -0.317   -0.3185  -0.32275
 -0.329   -0.33525 -0.34    -0.3445  -0.3495  -0.354   -0.3565  -0.3555
 -0.35225 -0.3485  -0.3435  -0.33725 -0.33125 -0.3275  -0.327   -0.328
 -0.3295  -0.3325  -0.3385  -0.34725 -0.35525 -0.361   -0.3655  -0.36975
 -0.37425 -0.377   -0.37625 -0.37375 -0.37275 -0.37375 -0.3745  -0.37475
 -0.37675 -0.3825  -0.39175 -0.40075 -0.40675 -0.41075 -0.4145  -0.418
 -0.41875 -0.4155  -0.4105  -0.40725 -0.40575 -0.403   -0.39925 -0.39725
 -0.3985  -0.4015  -0.40325 -0.40325 -0.4035  -0.4045  -0.40425 -0.401
 -0.395   -0.38825 -0.38175 -0.37475 -0.3665  -0.358   -0.3515  -0.34675
 -0.3415  -0.33575 -0.33125 -0.3295  -0.33    -0.32975 -0.32825 -0.3275
 -0.3285  -0.33    -0.3295  -0.327   -0.324   -0.32125 -0.318   -0.313
 -0.30725 -0.303   -0.301   -0.2995  -0.297   -0.295   -0.29575 -0.2985
 -0.30025 -0.29975 -0.298   -0.296   -0.294   -0.29175 -0.2885  -0.28475
 -0.2815  -0.279   -0.27825 -0.27925 -0.2815  -0.28525 -0.29075 -0.2975
 -0.30375 -0.3085  -0.31225 -0.3155  -0.317   -0.315   -0.30975 -0.30275
 -0.29525 -0.28775 -0.27975 -0.271   -0.262   -0.25325 -0.245   -0.23625
 -0.2255  -0.2135  -0.20125 -0.18775 -0.17225 -0.15475 -0.13675 -0.11975
 -0.10275 -0.08575 -0.06975 -0.055   -0.04225 -0.03125 -0.021   -0.0105
  0.001    0.01275  0.0245   0.03775  0.053    0.0685   0.08325  0.0985
  0.11475  0.13025  0.14325  0.15325  0.161    0.1675   0.17275  0.175
  0.17325  0.1705   0.16875  0.167    0.164    0.16     0.15825  0.1595
  0.16     0.15725  0.15275  0.1495   0.1475   0.14325  0.135    0.12525
  0.1165   0.1085   0.09925  0.088    0.0765   0.067    0.05975  0.05225
  0.04375  0.03625  0.031    0.027    0.02075  0.011    0.0005  -0.00925
 -0.02    -0.034   -0.05075 -0.06725 -0.08125 -0.0935  -0.106   -0.11825
 -0.12775 -0.134   -0.139   -0.14475 -0.1515  -0.15725 -0.16175 -0.1675
 -0.176   -0.1865 ]

Output from convolution corr:
[ 0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.       0.
  0.       0.       0.       0.       0.       0.       0.      -0.0885
 -0.2655  -0.35475 -0.35625 -0.35625 -0.3535  -0.3485  -0.34325 -0.34
 -0.338   -0.3355  -0.33275 -0.3315  -0.33275 -0.3355  -0.338   -0.34
 -0.34275 -0.3465  -0.3495  -0.3505  -0.34975 -0.3485  -0.34725 -0.344
 -0.3375  -0.32925 -0.322   -0.317   -0.313   -0.309   -0.30525 -0.3035
 -0.305   -0.308   -0.3105  -0.31225 -0.31425 -0.3165  -0.31725 -0.31625
 -0.3155  -0.316   -0.317   -0.31725 -0.317   -0.3185  -0.32275 -0.329
 -0.33525 -0.34    -0.3445  -0.3495  -0.354   -0.3565  -0.3555  -0.35225
 -0.3485  -0.3435  -0.33725 -0.33125 -0.3275  -0.327   -0.328   -0.3295
 -0.3325  -0.3385  -0.34725 -0.35525 -0.361   -0.3655  -0.36975 -0.37425
 -0.377   -0.37625 -0.37375 -0.37275 -0.37375 -0.3745  -0.37475 -0.37675
 -0.3825  -0.39175 -0.40075 -0.40675 -0.41075 -0.4145  -0.418   -0.41875
 -0.4155  -0.4105  -0.40725 -0.40575 -0.403   -0.39925 -0.39725 -0.3985
 -0.4015  -0.40325 -0.40325 -0.4035  -0.4045  -0.40425 -0.401   -0.395
 -0.38825 -0.38175 -0.37475 -0.3665  -0.358   -0.3515  -0.34675 -0.3415
 -0.33575 -0.33125 -0.3295  -0.33    -0.32975 -0.32825 -0.3275  -0.3285
 -0.33    -0.3295  -0.327   -0.324   -0.32125 -0.318   -0.313   -0.30725
 -0.303   -0.301   -0.2995  -0.297   -0.295   -0.29575 -0.2985  -0.30025
 -0.29975 -0.298   -0.296   -0.294   -0.29175 -0.2885  -0.28475 -0.2815
 -0.279   -0.27825 -0.27925 -0.2815  -0.28525 -0.29075 -0.2975  -0.30375
 -0.3085  -0.31225 -0.3155  -0.317   -0.315   -0.30975 -0.30275 -0.29525
 -0.28775 -0.27975 -0.271   -0.262   -0.25325 -0.245   -0.23625 -0.2255
 -0.2135  -0.20125 -0.18775 -0.17225 -0.15475 -0.13675 -0.11975 -0.10275
 -0.08575 -0.06975 -0.055   -0.04225 -0.03125 -0.021   -0.0105   0.001
  0.01275  0.0245   0.03775  0.053    0.0685   0.08325  0.0985   0.11475
  0.13025  0.14325  0.15325  0.161    0.1675   0.17275  0.175    0.17325
  0.1705   0.16875  0.167    0.164    0.16     0.15825  0.1595   0.16
  0.15725  0.15275  0.1495   0.1475   0.14325  0.135    0.12525  0.1165
  0.1085   0.09925  0.088    0.0765   0.067    0.05975  0.05225  0.04375
  0.03625  0.031    0.027    0.02075  0.011    0.0005  -0.00925 -0.02
 -0.034   -0.05075 -0.06725 -0.08125 -0.0935  -0.106   -0.11825 -0.12775
 -0.134   -0.139   -0.14475 -0.1515  -0.15725 -0.16175 -0.1675  -0.176
 -0.1865  -0.14525 -0.04925]
```