Heart Rate Monitoring Using PPG Signals

Authors:

Ziar Khalik, 4098250
Ahmet Gerçekcioğlu, 4042700

Supervisors:
Richard Hendriks
Seyran Khademi

BSc graduation project,
Electrical Engineering,
TU Delft

Delft, June 2016

TECHNICAL UNIVERSITY OF DELFT

Abstract

Monitoring the heart rate using PPG signals that is worn on the wrist has the downside that it is susceptible to motion. The device on the wrist moves along with the motions of the arm, consequently creating artifacts in the measurement. There are many signal processing techniques that can remove these artifacts. This thesis discusses one of the possible ways to remove motional artifacts from PPG signals, which is the TROIKA framework. This framework consists of three parts: signal decomposition, sparse signal reconstruction and spectral peak tracking. The main subject in this thesis is removal of motional artifacts by signal decomposition. Two methods have been proposed to identify components that belong to motional artifacts. Both methods worked as intended for removing those artifacts. The results of the TROIKA framework that is put together were within expectations.

Contents

Abstract i

List of Figures iv

List of Tables v

Abbreviations vi

1 Introduction 1
- 1.1 Problem definition . . . 2
- 1.2 Outline . . . 3

2 Programme of Requirements 4
- 2.1 Functional requirements . . . 4
- 2.2 System requirements . . . 4
- 2.3 Development of manufacturing methodologies . . . 5
- 2.4 Business strategies, marketing and sales opportunities . . . 5

3 The TROIKA Framework 6
- 3.1 Signal Decomposition Algorithms . . . 6

4 Singular Spectrum Analysis 9
- 4.1 The algorithm . . . 9
- 4.2 Optimizing the parameters . . . 11
- 4.3 Results . . . 13

5 Removal of Motional Artifacts 16
- 5.1 Various scenarios . . . 16
- 5.2 Method 1: Removal of Components (RoC) . . . 18
  - 5.2.1 Identification of MA components . . . 19
  - 5.2.2 Optimizing the parameters . . . 20
  - 5.2.3 Results . . . 21
- 5.3 Method 2: Filtering of Components (FoC) . . . 22
  - 5.3.1 Criterion factor . . . 24
  - 5.3.2 Parameters . . . 24
  - 5.3.3 Spectral peak tracker . . . 25
  - 5.3.4 Results . . . 26
- 5.4 Temporal Difference . . . 28

6 Final Results **30**
6.1 Results from TROIKA 30
6.2 Results compared to other implementations 33

7 Conclusion **34**
7.1 Conclusion 34
7.2 Recommendations and future work 35

A Matlab code **36**

Bibliography **48**

List of Figures

1.1 ECG and PPG signals without MA 2
1.2 ECG and PPG without MA 3

3.1 The TROIKA framework ¸ 7

4.1 The eigenvalues of the matrix $\mathbf{XX^T}$ 12
4.2 Decomposition of an ECG signal 14
4.3 Decomposition of a PPG signal 15

5.1 The various situations that can be considered for MA and the HR. 18
5.2 Illustration of identifying the MA components. 20
5.3 The results from MA removal. 22
5.4 Components of dataset 1 window 58. 23
5.5 The result of the second method considered for the situations in Figure 5.1 27
5.6 Example of temporal difference. First plot shows the reconstructed signal, second plot shows the reconstructed signal after temporal difference. . . . 29

6.1 Results from the first method. 31
6.2 Results from the second method. 32

List of Tables

4.1 Reconstruction of the signal 13

6.1 Results from TROIKA. 33
6.2 Comparison of TROIKA with other implementations. 33
6.3 Execution times of different implementation 33

Abbreviations

BPM	**B**eats **P**er **M**inute
HR	**H**eart **R**ate
MA	**M**otional **A**rtifact(s)
PPG	**P**hoto**p**lethysmo**g**ram
ECG	**E**lectro**c**ardio**g**raphic
SSA	**S**ingular **S**pectrum **A**nalysis
EMD	**E**mpirical **M**ode **D**ecomposition
SVD	**S**ingular **V**alue **D**ecomposition
SSR	**S**parse **S**ignal **R**econstruction
SPT	**S**pectral **P**eak **T**racking
LED	**L**ight **E**mitting **D**iode

Chapter 1

Introduction

The examination and interpretation of arterial pulses for medical purposes was widely applied in the ancient times among the Indian, Chinese, Egyptian and Greek civilisations. The first device intended for the purpose of measuring the pulse rate was invented in the 17th century by Sanctorius and was called pulsilogy [1]. Nowadays using Electrocardiographic (ECG) signals is a standard fashion to measure the heart rate (HR) for the medical purposes, it measures the electrical activity in the heart. With the increasing interest for a healthier lifestyle people tend to be more attentive of their pulse rate and like to have a non-intrusive device for various activities, in order to measure the pulse rate. An alternative for ECG is Photoplethysmography (PPG). PPG is based on the illumination of the blood vessels using a LED and measuring the reflection of the light into a photo diode. An example of an ECG and a PPG signal can be seen in Figure 1.1.

One problem with using PPG sensors on the wrist is that they are susceptible to motion. When the watch moves, the measurements by the PPG sensors are distorted by so called motional artifacts (MA). There are many signal processing techniques to MA artifacts from PPG signals. The following techniques are going to be considered: Independent Component Analysis (ICA) [2], Adaptive Noise Cancellation (ANC) [3], the TROIKA framework [4] and Joint Sparse Spectrum Reconstruction [5]. According to [4] ICA has a shortcoming so that MA removal is not satisfactory. ICA assumes that that there is statistical independence in the signal, but this is not the case in PPG signals that contain MA [6]. ANC, TROIKA and JOSS are the state-of-the art at the moment, and the results in [4] and [5] look very promising. These three signal processing techniques will be studied in dept by the three subgroups to find the best solution for the problem.

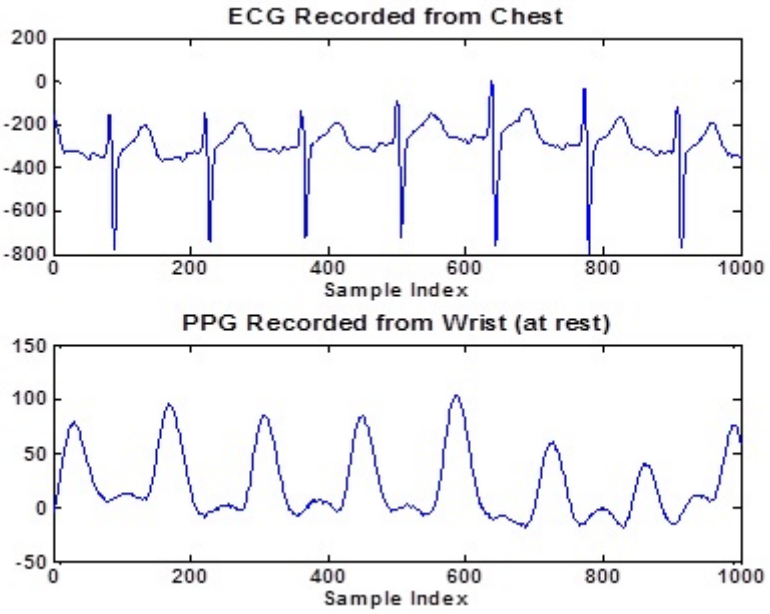


FIGURE 1.1: ECG and PPG signals without MA

1.1 Problem definition

There are various products in the market that use PPG signals for recording heart rates, however these watches are far from accurate [7]. The main problem with these devices is that they appear to be very sensitive to motional artifacts. Figure 1.2 shows an example of the periodograms of PPG signals respectively with and without motional artifacts. The artifacts are the result of the movements of the arms. These movements in the arm can influence the PPG signal values, because the watch moves along. The main goal for this project is to develop an algorithm that removes noise and motional artifacts from PPG signals.

The supervisor of the project has provided the group with twelve data sets. These data sets contain measurement results from a subject doing running exercises. Each data set consists of measurement from two PPG sensors, an ECG sensor and three accelerometer sensors that measure movements in the x, y and z directions. The PPG signals are sampled with a frequency of 125 Hz. The data is analysed in so called time windows. Each window consists of 1000 samples of a signal, which corresponds to 8 seconds. This window is shifted by 250 samples, or 2 seconds, to analyse the following piece of the signal.

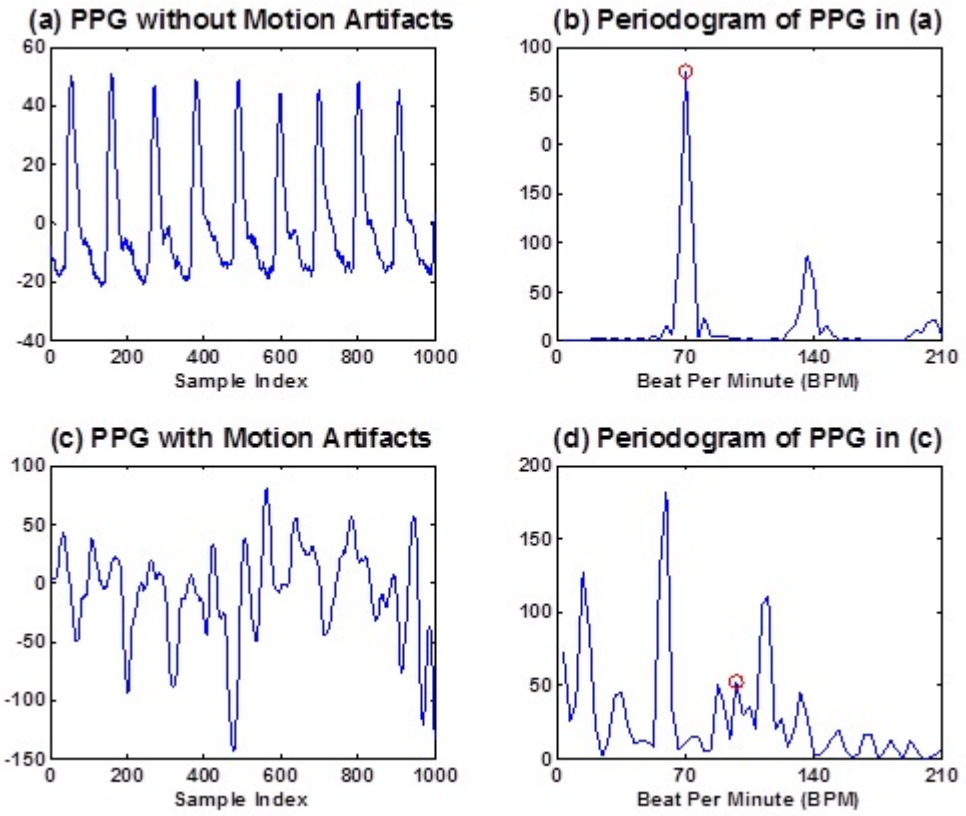


FIGURE 1.2: ECG and PPG without MA

1.2 Outline

This thesis will focus on the implementation of the TROIKA framework, which consists out of signal decomposition, sparse signal reconstruction and spectral peak tracking. Signal decomposition is the main subject in this thesis. The subgroups are divided as follows:

- Subgroup A1: Sparse signal reconstruction and spectral peak tracking
- Subgroup A2: Removal of MA using signal decomposition.
- Subgroup A3: Removal of MA using ANC.

The requirements for the system are given in chapter 2. Chapter 3 will give a general introduction to the TROIKA framework. Chapter 4 explains how singular spectrum decomposition is implemented. In chapter 5 the removal of motional artifacts from PPG signals is discussed and two solutions to remove these artifacts are presented. Finally the system is connected together and is evaluated and tested in the subsequent chapters.

Chapter 2

Programme of Requirements

Currently there are many wearable devices that can keep track of ones health. One of the features most of them have is the monitoring of the heart rate, which is typically done by using photoplethysmographic (PPG) signals that are measured with a watch on the wrist. The problem is that the measurements of the heart rate can be distorted by the movement of the watch when the wearer is doing physical exercise. This makes monitoring of the heart rate much harder and consequently less accurate. This is why there is need for an algorithm that can overcome these problems. This system will be marketed to businesses that produce wearable devices that have a heart monitoring function.

The requirements that are set for the system are based on given data sets. These data sets consist of a subject doing running exercises. The duration of the measurements of each data set is around five minutes. There are 12 data sets in total.

2.1 Functional requirements

[1.1] The system must give a heart rate in BPM of the wearer.

[1.2] The system must be able to be implemented in a wearable device that can be used during exercise.

[1.3] The system must give accurate heart rate values during rest and exercise.

2.2 System requirements

[2.1] The average absolute error on each data set must be below 10 BPM.

[2.2] The average absolute error over all data sets must be below 5 BPM.

[2.3] The system must be able to calculate the heart rate within 2 seconds on a wearable device.

2.3 Development of manufacturing methodologies

[3.1] The algorithm is developed in MATLAB, which has a rich set of toolboxes and functions that allow for easy implementation of the system.

2.4 Business strategies, marketing and sales opportunities

[4.1] The system will be marketed in a business to business model, particularly to smart-watch manufacturers.

[4.2] Licensing will be used to make profit on the system.

Chapter 3

The TROIKA Framework

TROIKA is a framework that consists of three key parts: signal decomposition, Sparse Signal Reconstruction (SSR) and Spectral Peak tracking (SPT). TROIKA works with a single PPG signal and also uses acceleration data. Before the PPG signal goes into the signal decomposition state, it goes through a bandpass filter. This filter removes frequencies that cannot be associated with a humanly possible heart beats. The signal decomposition stage is used to denoise the PPG signal and remove motional artifacts. this is done by decomposing a single signal into multiple components. The components can then be analysed and the ones associated with noise and interference can be removed. After that the signal is reconstructed without those components. Before the signal goes through the SSR stage, it is temporally differentiated to remove MA components that correspond with aperiodic movements. The SSR stage gives a high resolution spectrum estimation of the signal, which makes the peak tracking easier. The SPT stage analyses the signal that is created by SSR and selects the spectral peaks that correspond with the heart beat. It makes use of the frequency harmonic relation of heart rates and that the heart rate can't make big changes in two successive time windows. A flowchart of the TROIKA framework can be seen in Figure 3.1.

3.1 Signal Decomposition Algorithms

Because the focus of this thesis lies on signal decomposition, multiple algorithms that can do this will be considered. A couple of these signal decomposition algorithms are Singular Spectrum Analysis (SSA), Single-channel Independent Component Analysis (SCICA) and Empirical Mode Decomposition (EMD).

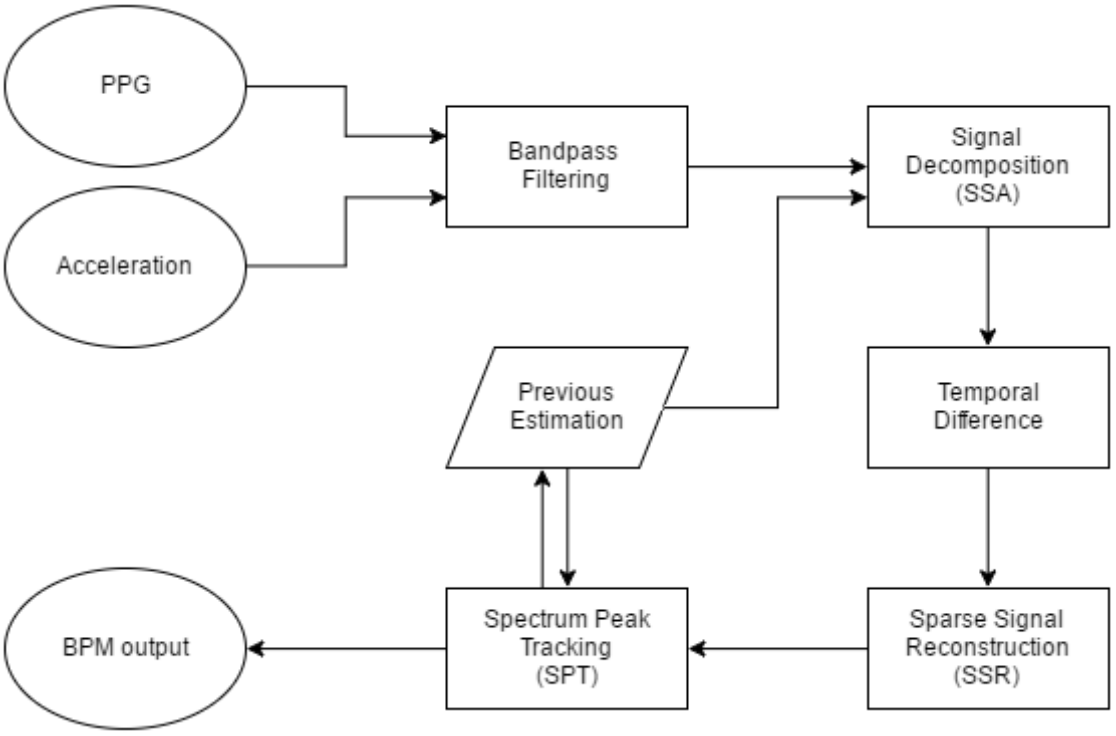


FIGURE 3.1: The TROIKA framework ¸

SSA [8] is a powerful technique of time series analysis. It can be used in with many applications and works with arbitrary statistical processes. This includes linear and non-linear signals, stationary and non-stationary signals and so on. SSA decomposes a time series into oscillatory components and noise. A disadvantage of the SSA algorithm is that it does not automatically remove MA. This is because there is not enough information to determine which components contain the heart rate signal. However there is a solution to this: accelerometer data can be used to recognize MA components in the SSA decomposition and they can be excluded from the reconstruction.

The SCICA algorithm [9] can be used to separate independent sources from a single time series. It uses an ICA algorithm to divide a single channel into multiple components. ICA is a technique that is used to separate linearly mixed sources. A disadvantage of the SCICA algorithm is that the independent source processes need to have disjoint spectral support [9]. This is not the case with ECG and PPG, so more information has to be known about the signal and it gets harder to separate the signal.

The EMD algorithm [10] is an empirical algorithm, which is more intuitive. It works well on non-stationary and non-periodic signals, which meets the requirements of our application. A disadvantage of the EMD algorithm is that it cannot identify IMF's whose frequencies are very close to each other. This can be a problem with determining

the heart rate using the PPG heart rate monitor around the wrist. If the frequency of the MA is close to the frequency of the heart rate, it will have an influence on the outcome. According to [11] this problem can be solved by using a slightly different version of the EMD algorithm, the EEMD (Ensemble EMD). This version adds white noise to the original signal each time before it goes through the EMD.

Looking at the disadvantages of each of the algorithms, it seems that the SSA algorithm is the most promising technique. The disadvantage can be solved easily, because the given data sets include accelerometer data.

Chapter 4

Singular Spectrum Analysis

In the previous chapter it is explained that one of the algorithms for signal decomposition to use in this case is the SSA algorithm. This chapter explains how the algorithm works and can be implemented in Matlab. The notation in this section is as follows:

- Bold uppercase stands for a matrix (e.g. $\mathbf{X}$).
- Normal uppercase stands for a vector (e.g. X).
- Normal lowercase stands for a scalar (e.g. x).
- Superscript T stands for transposed (e.g. $\mathbf{X}^T$).
- Subscript i, j stands for the element in a matrix in row i column j (e.g. $x_{i,j}$).

4.1 The algorithm

Implementing the SSA algorithm is fairly straightforward: the steps in the algorithm have to be followed. The basic SSA algorithm consists of two complementary stages: decomposition and reconstruction. The steps for the decomposition stage are [8][12]:

1. Computing the L-trajectory matrix $\mathbf{X}$. This transfers a one-dimensional time series $Y_N = (y_1, \ldots, y_N)$ into the multidimensional series $X_1, \ldots, X_K$ with vectors $X_i = (y_i, \ldots, y_L)^T$. L is the window length and is chosen such that $2 \leq L \leq N$. K is chosen such that $K = N - L + 1$. The result is the trajectory matrix $\mathbf{X} = [X_1, ..., X_K]$:

$$\mathbf{X} = \begin{pmatrix} \mathrm{y}_1 & \mathrm{y}_2 & \mathrm{y}_3 & \cdots & \mathrm{y}_K \\ \mathrm{y}_2 & \mathrm{y}_3 & \mathrm{y}_4 & \cdots & \mathrm{y}_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{y}_L & y_{L+1} & y_{L+2} & \cdots & \mathrm{y}_N \end{pmatrix} \tag{4.1}$$

2. Compute the matrix $\mathbf{S} = \mathbf{X}\mathbf{X}^T$. This is a positive definite and symmetric matrix which can be decomposed using eigenvalue decomposition. Also the eigenvalues of this matrix are real and positive.

3. Singular value decomposition (SVD) of the matrix $\mathbf{S}$: Compute the eigenvalues and eigenvectors of the matrix $\mathbf{S}$. If $\lambda_1, \ldots, \lambda_L$ are the eigenvalues of $\mathbf{S}$ in decreasing order and $U_1, \ldots, U_L$ are the corresponding eigenvectors of these eigenvalues, then $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i} (i = 1, \ldots, d)$. Here $d = max\{i, \text{such that } \lambda_i > 0\}$. The trajectory matrix $\mathbf{X}$ can now be written as:

$$\mathbf{X} = \mathbf{X}_1 + \ldots + \mathbf{X}_d, \tag{4.2}$$

where $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$. The collection ($\sqrt{\lambda_i}, U_i, V_i$) is called the eigentriple of the SVD.

The steps for the reconstruction stage are:

1. Grouping: The elementary matrices $\mathbf{X}_i$ are split into m disjoint subsets $I_1, \ldots, I_m$. Let $I = i_1, \ldots, i_p$, then the matrix $\mathbf{X}_I$ corresponding to the group I is defined as $\mathbf{X}_I = \mathbf{X}_I(1) + \ldots + \mathbf{X}_I(p)$. The trajectory matrix $\mathbf{X}$ can now be written as:

$$\mathbf{X} = \mathbf{X}_I(1) + \ldots + \mathbf{X}_I(m). \tag{4.3}$$

2. Diagonal averaging: Each matrix $\mathbf{X}_I$ is transformed into a new time series of length N. These are the components that the original time series is split into. Let $\mathbf{X}$ be an $L \times K$ matrix with elements $x_{i,j}$, $1 \leq \mathrm{i} \leq \mathrm{L}$, $1 \leq \mathrm{j} \leq \mathrm{K}$. Then $L^* = min(L, K), K^* = max(L, K)$ and $N = L + K - 1$. Let $x^*_{ij} = x_{ij}$ if L < K and $x^*_{ij} = x_{ji}$ otherwise. Then diagonal averaging transfers the matrix $\mathbf{X}$ to the series $g_0, \ldots, g_{N-1}$ by the following formula:

$$g_k = \begin{cases} \frac{1}{k+1} \sum_{m=1}^{k+1} x^*_{m,k-m+2} & \text{for } 0 \leq \mathrm{k} < L^*\text{- } 1, \\ \frac{1}{L^*} \sum_{m=1}^{L^*} x^*_{m,k-m+2} & \text{for } L^* \text{ - } 1 \leq \mathrm{k} < K^*, \\ \frac{1}{N-k} \sum_{m=k-K^*+2}^{N-K^*+1} x^*_{m,k-m+2} & \text{for } K^* \leq \mathrm{k} < \mathrm{N}. \end{cases} \tag{4.4}$$

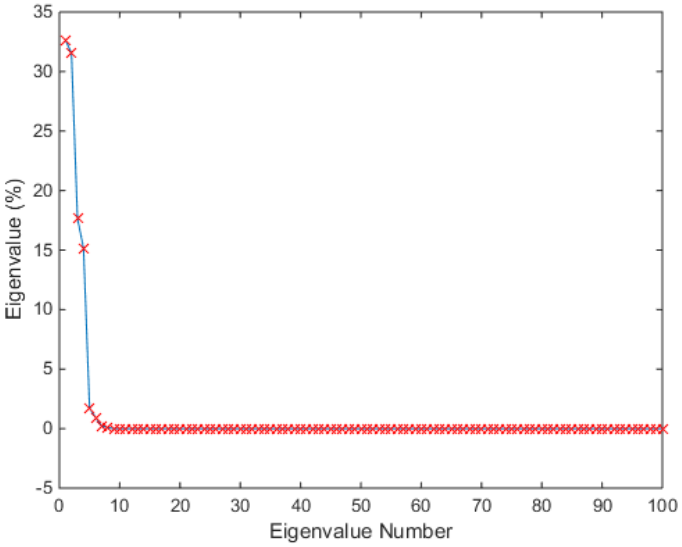
4.2 Optimizing the parameters

The SSA algorithm has two parameters that can be used to optimize it for the signal that is going to be decomposed. These are the window length L and the choice of the grouping. Generally for the window length L it is good to choose a value that is less than half the length of the signal that is going to be decomposed, so that the error can be balanced and there is the ability to resolve lower frequencies. Also if there is a periodicity in the signal, L can be chosen such that it's a multiple of the period of the signal [13]. The grouping is done such that eigenvalues that have amplitudes that are about the same size are grouped together.
When looking at the trajectory matrix in equation (4.1), it can be seen that the window length is the height of the trajectory matrix. The amount of information that the vectors X_i contain is dependent on the window length. The SVD is computed for the matrix $\mathbf{XX^T}$ with the dimensions of L $\times$ L, so the window length corresponds to the number of eigenvalues of the matrix. To make a choice for the size of the window length, the eigenvalues can be examined. Figure 4.1 shows three plots of eigenvalues of the matrix $\mathbf{XX^T}$ with various window lengths.

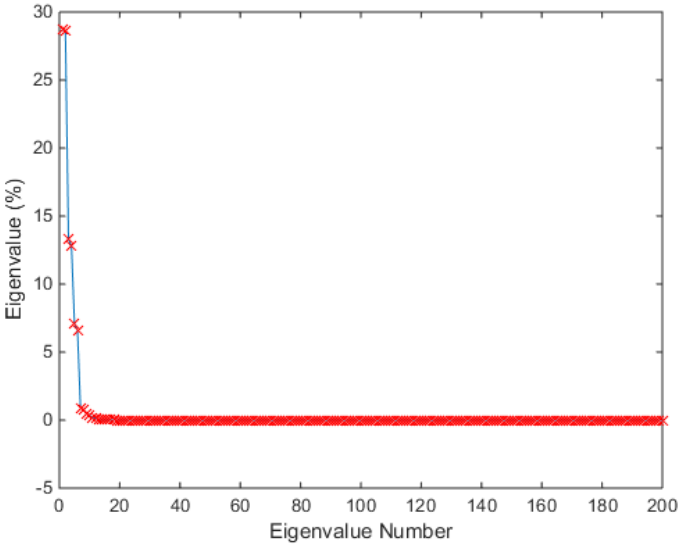
When examining the three plots in Figure 4.1 it can be seen that the size of the window length L indeed corresponds to the amount of eigenvalues of the matrix $\mathbf{XX^T}$. The sizes of the eigenvalues e relative to the sum of all eigenvalues in the y-axis are calculated as follows:

$$e_i(relative) = \frac{e_i}{\sum_{m=1}^{L} e_m} \times 100. \tag{4.5}$$

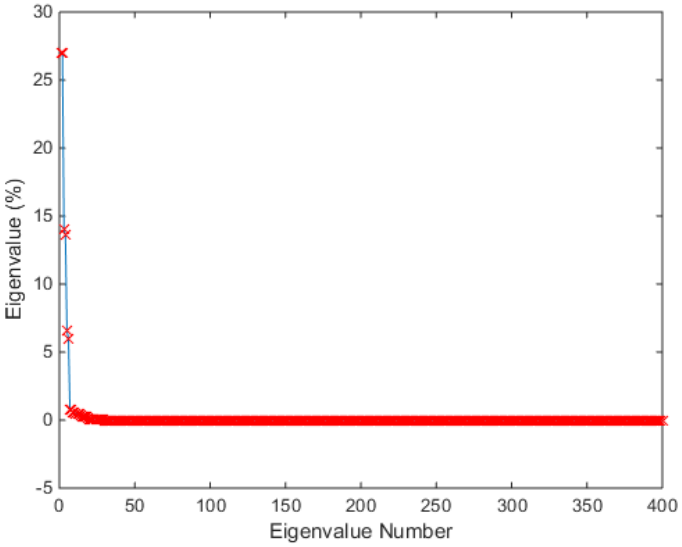
This is done to make it easier to see how much information the eigentriple that belongs to the eigenvalue contains about the original signal. When comparing Figure 4.1a to Figure 4.1b it can be noticed that in Figure 4.1a there are less eigenvalues that have a relative large size (i.e. larger than 5%). This means that some information is thrown away when L is chosen too small. Comparing Figure 4.1b to Figure 4.1c shows that the amount of large eigenvalues are the same but now there are more eigenvalues that are zero or close to zero. This leads to the conclusion that a larger window length doesn't contain much more information for the reconstruction of the signal. With all of this in mind, the effects on the reconstruction of the signal will be examined for $150 \leq L \leq 250$. In the tests the value L = 215 had given the best result in terms of computational load and accuracy of the reconstruction.
In the grouping stage the matrices that belong to the eigentriples where the eigenvalues are close to each other in size are grouped together. This is done because those matrices are basically equal to each other. Eigentriples that have an eigenvalue that is close to



(A) Eigenvalues for a window length of 100



(B) Eigenvalues for a window length of 200



(C) Eigenvalues for a window length of 400

FIGURE 4.1: The eigenvalues of the matrix $\mathbf{XX^T}$. The x-axis shows the eigenvalue, The y-axis shows the size of the eigenvalue relative to the sum of all eigenvalues in percentage.

zero, contain no significant information so they can be excluded from the reconstruction. This also reduces the computational load, because the corresponding matrices in the grouping stage are not computed anymore. When looking at the figures in Figure 4.1 it can be seen that after the tenth eigenvalue, the eigenvalues are getting close to zero. To determine the lowest possible number for the total eigentriples that are taken into consideration for the reconstruction, the cross correlation can be used. The cross correlation is a measure of similarity between two signals. Table 4.1 shows the results for one time window. It shows that when too much information is thrown away, the signal cannot be reconstructed properly. It can be concluded that considering only the first 9 eigentriples is sufficient to reconstruct the original signal without losing too much information.

TABLE 4.1: The results for the cross correlation between the original signal and the reconstructed signal when considering different amount of eigenvalues for reconstruction.

number of eigentriples	5	6	7	8	9	10	11	12
% of signal retained	90.6	90.6	97.0	97.0	99.2	99.2	99.9	99.9

4.3 Results

To check the performance of the decomposition by SSA, a decomposition of an ECG and a PPG signal can be done. All signals have gone through a bandpass filter before being decomposed. A normal human heart rate rarely goes below 35 BPM and doesn't get higher than 210 BPM. These frequencies are removed by the bandpass filter. The decomposition of the ECG signal can be seen in Figure 4.2. The figure shows the frequency plots of an ECG signal and the components which are created using SSA. In the first plot it can be seen that the maximum peak corresponds to the heart rate (red dot shows the true HR), which is around 77 BPM. Looking at the other plots, which are the components created by the decomposition, it can be seen that the components in the second and third plots correspond to the signal of the heart rate. The highest peak in those plots lies at the frequency of the heart beat, which is also the case in the original ECG signal, but it is a cleaner signal. For example, there are frequencies with significant amplitude in the original ECG signal around 150 BPM. However those frequencies are not present in the second and third plots.

The other plots show components that contain no to very little frequencies of the heart rate. These components consist mostly of noise that was in the original signal. It can also be seen that the maximum amplitude of the components lowers with each succeeding component. This is because of the same reason as explained in the last section: each succeeding component belongs to an eigenvalue, which were in descending order. So it shows that larger eigenvalues correspond to components that contribute

more to the original signal. This also confirms that the smaller eigenvalues contribute very little to the original signal, which in turns means that they are not significant in the reconstruction of the signal. The original signal can be reconstructed by adding all components back together.

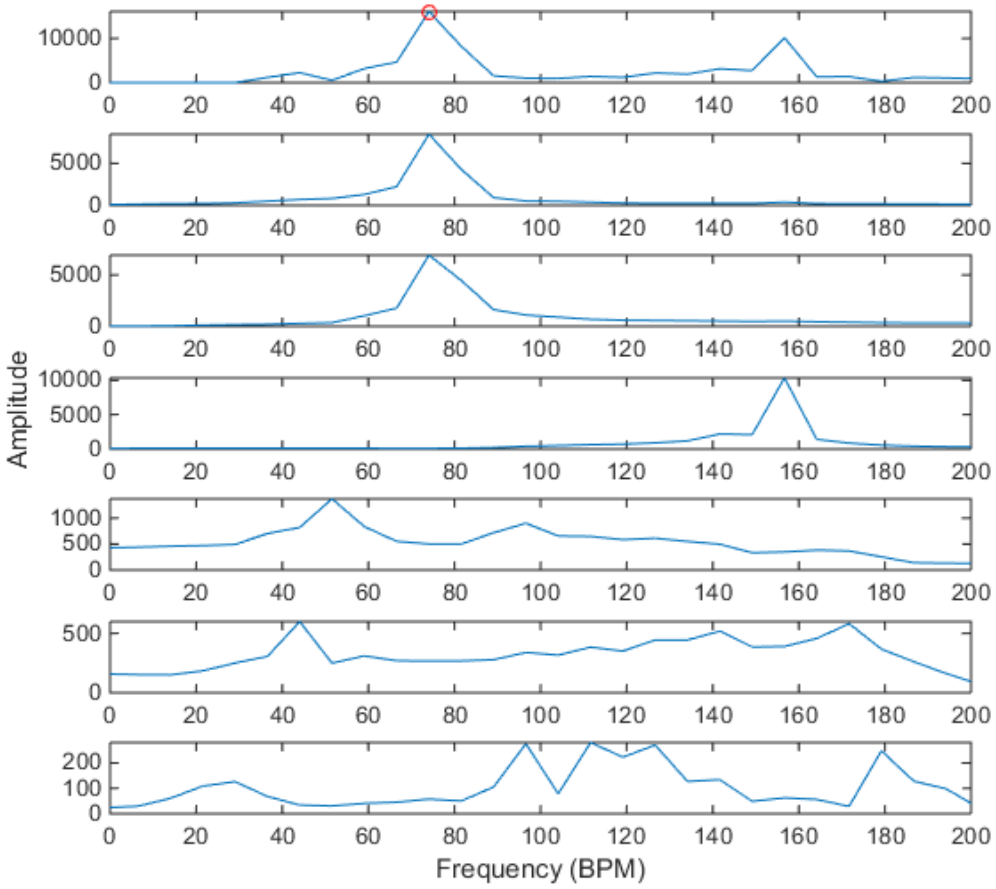


FIGURE 4.2: Decomposition of an ECG signal. Top plot shows the original ECG signal, the other plots show the corresponding components.

Next is the decomposition of the PPG signal that was recorded simultaneously with the ECG signal in Figure 4.2. Figure 4.3 shows the result. It can be noticed that there are obvious similarities between the decomposition of the ECG signal and the PPG signal. The first plot shows again that the maximum value corresponds to the true heart rate. It can also be seen that it contains more frequencies that do not belong to the frequency of the true HR, so the PPG contains more noise than the ECG signal. The second plot and third show the components that belong to the true HR, it only contains frequencies close to the true HR. The other plots show again components that contain no or little frequencies of the heart rate. Because the PPG signal has more noise

in it, these components have a higher amplitude than the corresponding components of the ECG signal.

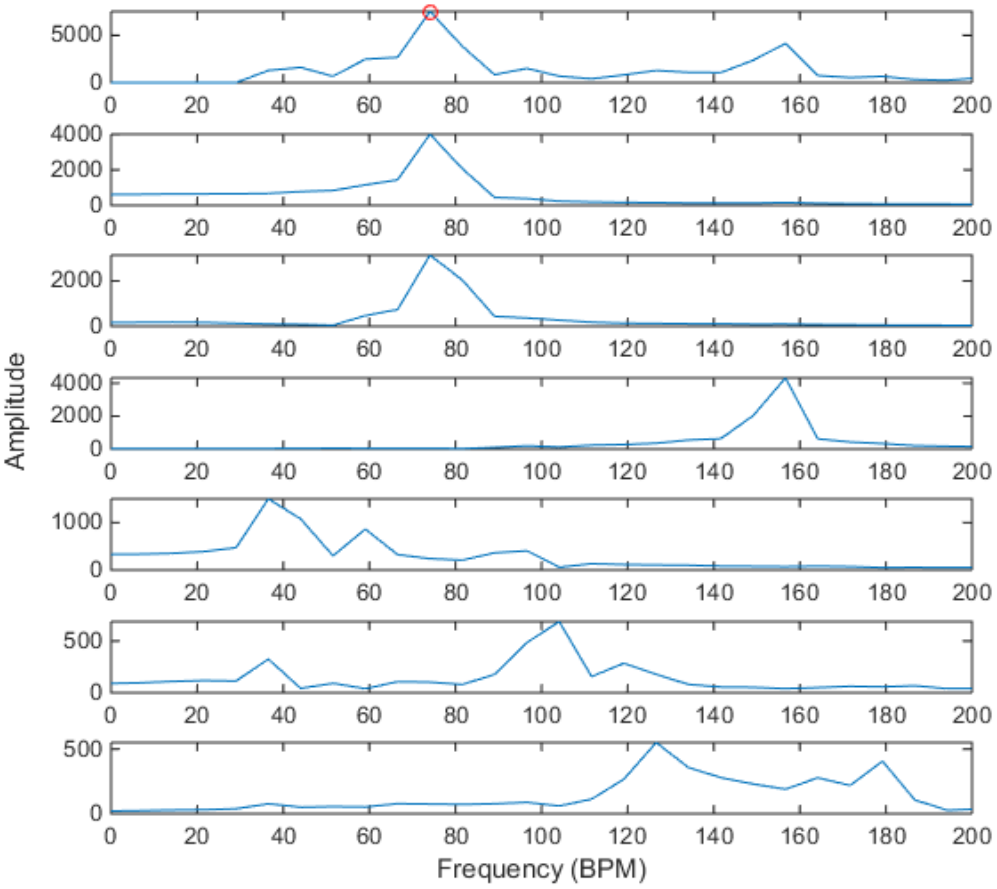


FIGURE 4.3: Decomposition of a PPG signal. Top plot shows the PPG signal, the other plots show the corresponding components.

Chapter 5

Removal of Motional Artifacts

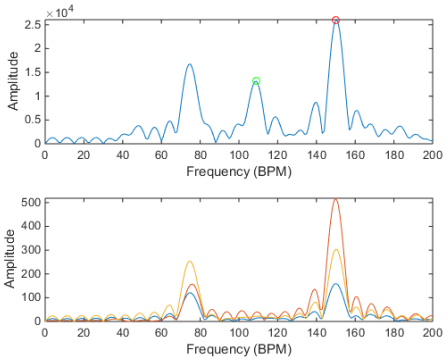
The biggest cause of distortions in the PPG signal are motional artifacts. Chapter 4 showed how a PPG signal can be decomposed into components and that there are one or more components that contain information about the HR. However the identification of those components is not always straightforward. It is not always the case that the first two components contain information about the HR. Any of the components can contain that information. This is why there is a need for a method to select the components that belong to the true HR. There are various situations to consider and some criteria has to be set to identify those components. This chapter explains two methods that can be used to remove MA.

5.1 Various scenarios

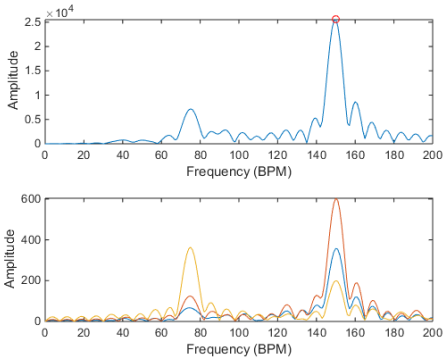
The ways MA appear in PPG signals are not always consistent. The most important cases to consider are:

1. The frequencies of the MA peaks are far from the true HR.
2. The frequency of the MA peak coincides with the true HR.
3. There are no clear MA peaks, so there is no information about the peaks in the PPG signal.
4. There is no clear peak at the true HR.
5. The frequency of the MA peak is close to the true HR.

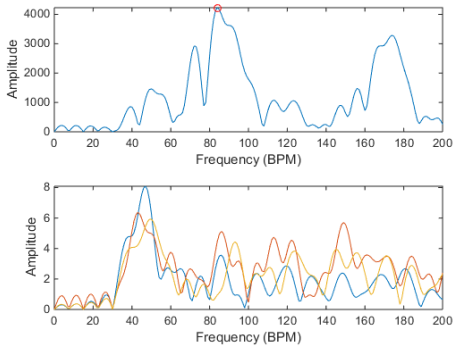
Figure 5.1 shows different plots to showcase these situations. The first plot shows the PPG signal and the second plot shows the three acceleration data. The red and green dot in the plots stand for respectively the maximum value and the true HR. If there is no green dot in the plot, it means that the highest peak corresponds with the HR. When thereThe PPG signals and the acceleration data have been converted to the frequency domain by doing the fast fourier transform (FFT) with a resolution of N = 7500, which corresponds to a resolution of approximately 1 BPM. This has been done to be able to distinguish the peaks more accurately. For example, in the case where the MA is close to the true HR, there would be only one peak instead of two in the frequency domain if the FFT would be taken with N = 1000 points. One other thing to notice is that the peaks in the acceleration data are also present in the PPG signal.



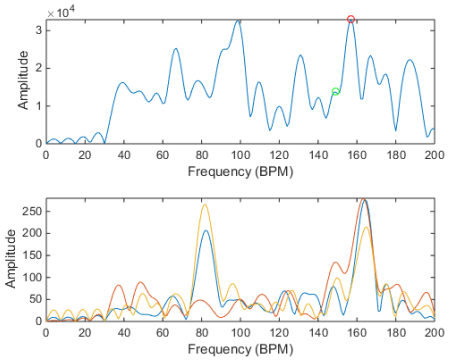
(A) Window where the MA are far from the true HR.



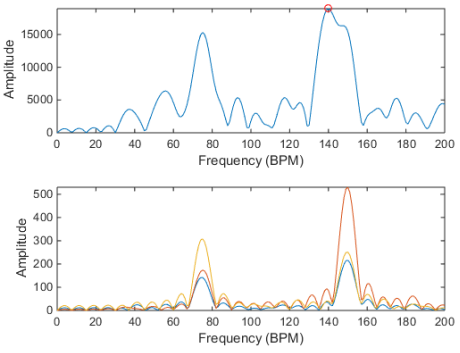
(B) Window where the frequency of the MA coincides with the true HR.



(C) Window where there is no MA.



(D) Window where there is no obvious peak at the true HR.



(E) Window where MA is close to the true HR.

FIGURE 5.1: The various situations that can be considered for MA and the HR. First plot shows the PPG signal, second plot shows the acceleration data of the x,y and z directions.

5.2 Method 1: Removal of Components (RoC)

The first method to remove MA from the PPG is to identify the components that correspond to MA and reconstruct the signal without those components. The main idea is that the components that have a peak at approximately the same frequency as the

peaks of the acceleration data (MA components) in the frequency domain are identified so that they can be excluded from the reconstruction. It is important to point out that the conversion to the frequency domain has been done by doing a FFT with a resolution of N = 7500. This has been done for the same reason as explained in the previous section, namely to better distinguish peaks that are close to each other.

5.2.1 Identification of MA components

There are two possible situations for the acceleration data: either there are peaks that correspond to the MA in the PPG signal or the signal is noisy. In the case that the signal is noisy the algorithm does not look for peaks in the acceleration data, because there is no information about MA. One way to recognize whether a signal is noisy or not is to use kurtosis. The kurtosis of a signal can tell how peaked the signal is. A low kurtosis means that the signal is flat, which means that it consists mainly out of noise. The next step is to locate the peaks of the acceleration data that are responsible for the MA peaks in the PPG signal. When the amplitude of the peak in the acceleration data is small, the corresponding peak in the PPG signal will be small too. For this reason only the peaks that have an amplitude larger than 50% of the maximum peak in the acceleration data are considered. The following step is to compare the locations of those peaks to the location of the maximum peak of each component of the PPG signal that is decomposed by SSA. Since the peaks of the acceleration data do not always exactly coincide with the MA peaks in the PPG signal, some tolerance (MA tolerance) has to be set. If maximum peak of a component lies between that tolerance, it is marked as a MA component.
In the cases that the MA is far from the true HR and where there is no MA present, this solution would suffice. But when the MA occurs at the same or close to the frequency as the true HR there is a problem: the components that correspond to the true HR will also be excluded from the reconstruction. A solution for this is to use information about the previous estimated value of the HR: components that have their maximum peak at around the same frequency as one of the peaks of the acceleration data, will not be removed if they are close to the frequency of the previous estimated HR. However the HR can change significantly between two time windows. This means that here too some kind of tolerance has to be set, the BPM tolerance. One other thing that can be done to enhance the reconstructed signal even more is to remove components that have a maximum peak very far from the previous estimated HR. This range should be large enough that even when the previous estimation of the HR had a large error, the component that corresponds with the HR should not be removed.
One disadvantage of this method is that it relies heavily on the fact that the estimation

of the HR is accurate in the previous time window. It wouldn't work if the previous estimation of the HR has a large error. However this problem can be solved in spectral peak tracking [14]. An illustration of all of this can be seen in Figure 5.2. In this example, the two components that have a maximum peak within the MA tolerance (red area) would be excluded from reconstruction. While the component that has a maximum peak within the BPM tolerance (the blue area) would not be excluded. This would be the component that contains the HR signal.

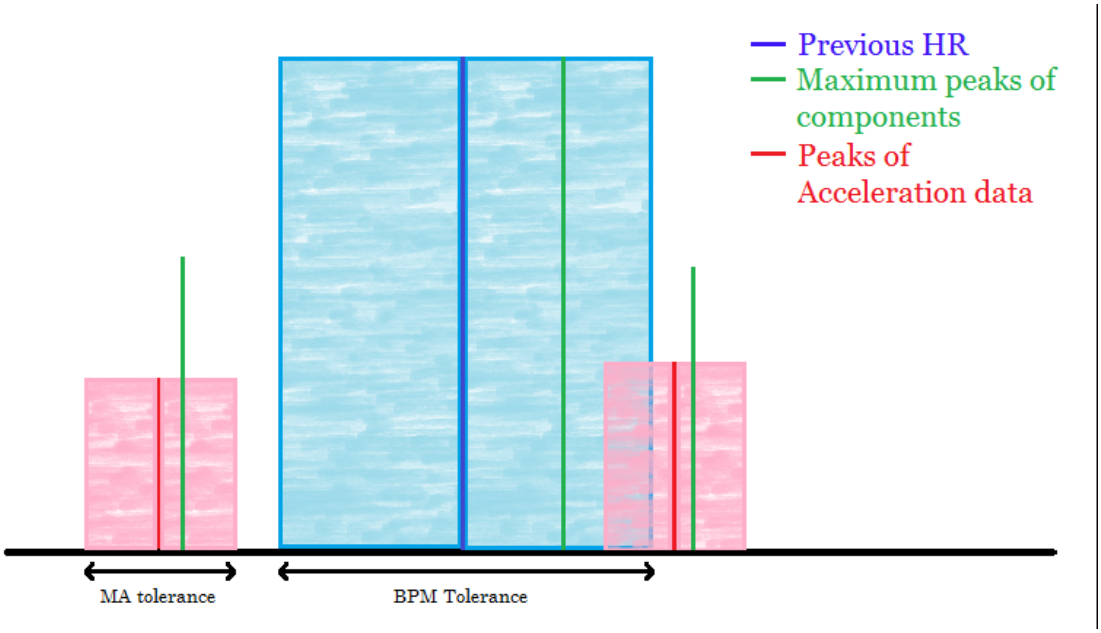


FIGURE 5.2: Illustration of identifying the MA components.

5.2.2 Optimizing the parameters

The parameters that can be optimized in this method are the criterion for the kurtosis, the MA tolerance and the BPM tolerance. To determine the correct criterion for the kurtosis, the acceleration data in the given data sets can be examined. It is found when there are clearly distinguishable peaks in the acceleration data, the kurtosis of the signal is above the value of 120.

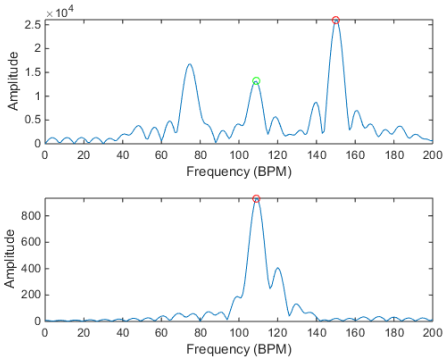
The next parameter to optimize is the MA tolerance. This tolerance has to be large enough that it includes the components that correspond with MA but small enough that it does not include the true HR when the MA is close to that frequency. When analysing the data sets, it is seen that the maximum difference between the locations of the peaks in the acceleration data and the corresponding peaks in the PPG signal is around 5 BPM. Thus a tolerance of ± 5 BPM should be sufficient to determine the components that belong to MA.

To estimate a correct value for the BPM tolerance, the maximum change in the HR

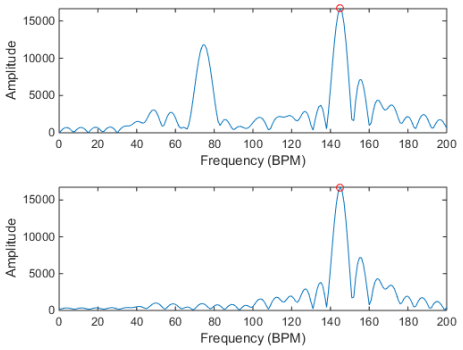
between two successive windows can be determined. From this it is given that the HR can change by a maximum of 10 BPM.. To make sure that the component that belongs to the HR will not be removed, a BPM tolerance of ±11 BPM has been chosen. Components that have a maximum peak at a frequency difference of ±4 × BPM tolerance from the previous estimated HR are also excluded from reconstruction.

5.2.3 Results

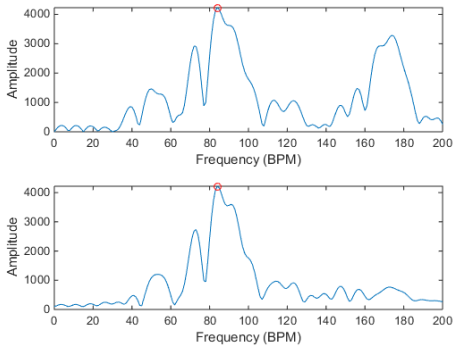
To check whether the RoC method functions correctly, it has been performed on the same time windows as in Figure 5.1. The results can be found in Figure 5.3. This time the top and bottom plots display respectively the original signal and the reconstructed signal after MA is removed. The red dot in the plots stand for the maximum amplitude and the green dot shows where the true HR is. It can be observed that MA removal works as expected in the cases 1,2 and 5, see Figures 5.3a, 5.3b and 5.3e. In those plots there is one large peak and it belongs to the true HR. In the case that there is no information about the MA, it is hard to determine which peak belongs to the HR. This is why the result Figure 5.3c is not ideal. In the case of when the peak belonging to the HR is not present and the MA is near the HR, shown in Figure 5.3d, it is also hard to find the peak belonging to the HR. But even in those cases the result is much better than before removing the MA, because there is much noise that is in the frequencies far from the HR which is removed. In some cases when MA is close to the HR there are still two dominant peaks after removal of MA. To make it easier for the SPT to identify which peak belongs to MA and which one to the HR in those cases, the locations of the acceleration peaks are sent to the SPT.



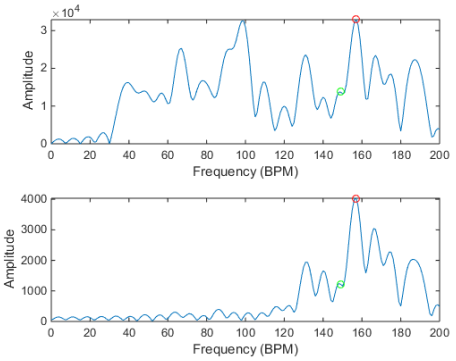
(A) Window where the MA are far from the true HR.



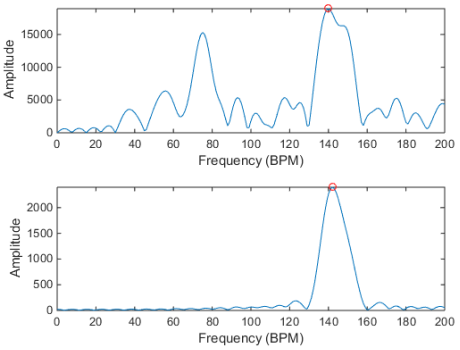
(B) Window where the frequency of the MA coincides with the true HR.



(C) Window where there is no MA.



(D) Window where there is no obvious peak at the true HR.



(E) Window where MA is close to the true HR.

FIGURE 5.3: The results from MA removal. The first plot shows the original signal, the second plot shows the reconstructed signal.

5.3 Method 2: Filtering of Components (FoC)

Removing a complete component can be a waste of useful data. Considering that a component can have one or more artifact peaks and a HR peak that can be distinguished very well from one and another in the frequency domain. As it is seen in Figure

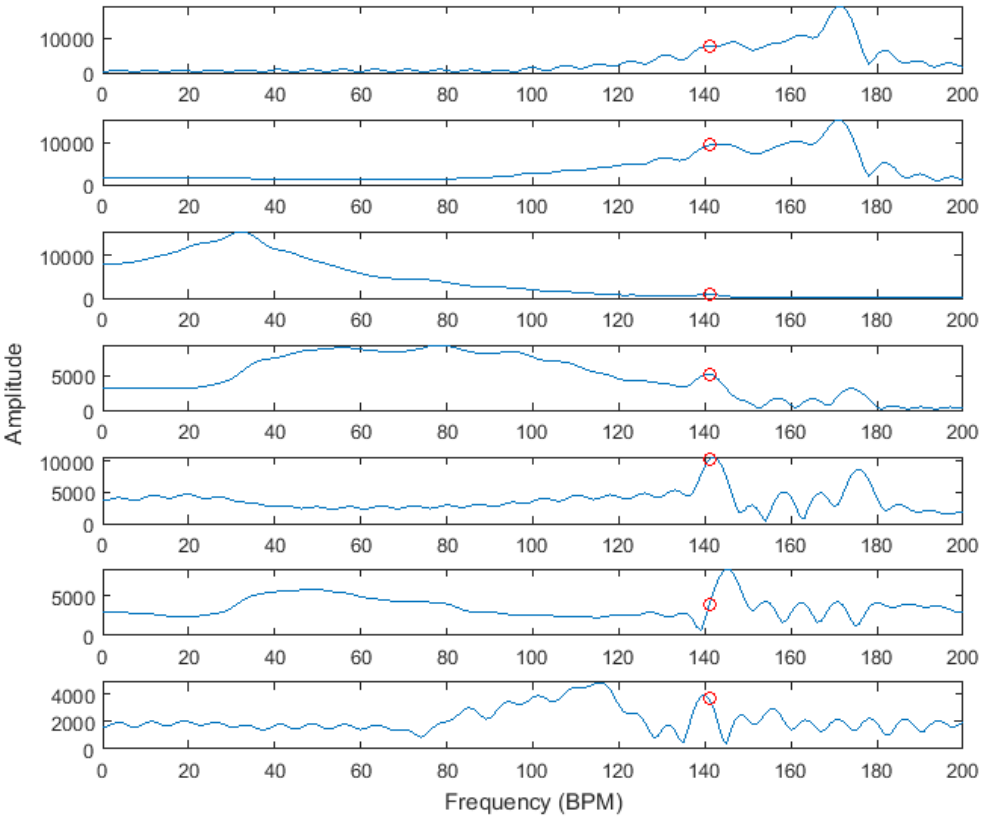


FIGURE 5.4: Components of dataset 1 window 58.

5.4 for the first seven components for a data set window. The red dot in Figure 5.4 is where the real BPM lies. Components one, two, four and seven contain HR peaks that could be otherwise removed in the Removal of Components method because the HR peaks are not the highest peak. Thus another method has been proposed for MA removal. In order to denoise and remove motional artifacts there is a different method with the same SSA based algorithm. The difference between the RoC method and this method is the fact that the components are not being removed after a certain criterion matching but rather filtered individually with a band-stop filter and added together in the frequency domain. This method will be called Filtering of Components. However there are disadvantages to this kind of noise and artifact filtering. There is always the probability that the HR peak and artifact peak lie in the same interval thereby removing the HR peak. This is the reason why a good criterion factor whether to remove or not to remove a peak is very essential. Also there are extra parameters in contrast to the RoC method. The RoC method only looks at the highest peak of a component. The FoC method has to look at more peaks in the component in order to filter peaks from the component when necessary.

5.3.1 Criterion factor

The criterion factor is the factor that determines when a peak of a component should be filtered out. Firstly the peaks of the acceleration data are chosen for the criterion factor. If the acceleration peak is within a certain interval with the component peaks the peak of the signal will get filtered out. This is interval will be called the MA tolerance interval. Second criterion is that the peaks of the signal should be outside the BPM tolerance interval with regards to the previous calculated BPM. In the default state is only the first criterion considered. For the non-default state both of the criterion's are considered. The last criterion is the kurtosis criterion, if the kurtosis of the acceleration window is higher than 130 the peaks will be filtered out otherwise nothing happens. This kurtosis criterion holds for both states. Results show that having a kurtosis criterion gives better results. When the kurtosis is low it means that there is too much noise to recognize MA peaks. The signal may look flat but there are many small peaks that go into the criterion factor thereby filtering out too much of the signal data.

5.3.2 Parameters

The parameter that determines which peaks of the acceleration data will be taken into consideration is referred to as σ. If acceleration peak is σ times larger than or equal to the maximum acceleration peak, it will go into the criterion factor. σ has to be taken such that the HR peak or a peak within respectable limits of the HR will be identified as a dominant peak for the peak tracker. It is important to point out that the sparse signal reconstruction algorithm is unnecessary for the FoC method. Because the SPT for FoC always considers only the maximum peaks. Hence for the SPT of FoC all peaks are considered as dominant peaks. If you take σ too high it will leave too many artifact peaks, if you take it too low it may filter out or lower the HR peak. The maximum HR variability for two seconds tend to be 10 BPM. However taking the BPM tolerance interval as 10 BPM is not a wise decision. For the calculation of a single window it seems to be working well. But on the longer term the BPM tolerance should be bigger. The BPM tolerance interval represents the same BPM tolerance interval as for the RoC method. A peak tracker can calculate a single heart peak wrong. Thus In the queue of window calculations a domino effect of a shifted BPM tolerance from the true HR is sometimes inevitable. This effect will be called "derailing" from the real track. Hence the BPM tolerance interval should be bigger. The remaining parameters that are mentioned in the RoC method keep the same value. MA tolerance interval and kurtosis value criterion are kept at the same value as for the RoC method due to practical reasons. Changing these parameters does not obtain better results. σ has two

discrete values, depending on the state of the SPT it will make a decision between the two. For the optimal average error σ is taken as 0.8 for the default state and 0.2 for the non-default state. The SPT for the FoC method works with two different states. For optimal results BPM tolerance interval is set to 13, MA tolerance interval is set to 5 and kurtosis value criterion is set to 130.

5.3.3 Spectral peak tracker

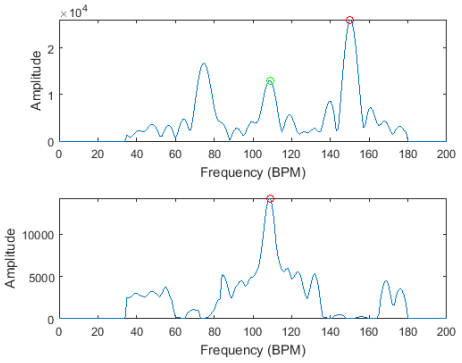
The peak tracking for RoC method has a different way of functioning compared to the SPT of the FoC method. The reason for the different SPT methods relies on the fact that the SPT for the RoC method doesn't seem to be working effectively for the FoC method. For the SPT of the RoC method you can look to [14]. When filtering the components contrary to removing, most of the signal data outside the BPM tolerance interval seem to be filtered out. So all the dominant peaks are within the BPM tolerance interval for the FoC method. SPT of RoC also considers the peaks outside the BPM-interval, in some cases. If the SPT of the RoC can not find a dominant peak three times in a row within the BPM tolerance interval, it will search the whole spectrum for the maximum peak. This is the retracking method for the SPT of RoC. Hence the problem is that for the FoC method the SPT will think that the trace is always on the right BPM track, because all the dominant peaks are within the BPM tolerance interval. This will lead to derailing from the true BPM track without knowing that it has derailed. This is the reason a different SPT for the FoC method has been built. Derailing is a phenomenon that is sometimes inevitable due to that the window signal is simply very bad. A peak at the HR is absent when the window data is bad. So it will take a peak that is too far from the HR. The main problem however is when the SPT is unable to "retrack" to the real BPM track.

The SPT for FoC works as follows: The first two windows operate in the default state. In the default state the σ is equal to 0.8 and prevBPM = 0, which means the whole spectrum is being searched and the maximum peak is taken as the calculated BPM. After the calculation of the first two windows the difference between the outcome is compared, if the difference is bigger than a certain number, which is taken as 8, is the next state the default state again. When this happens the third window is checked again in the default state , if the difference is smaller than eight with the second outcome, there is a certainty that the peak tracker is on the right track. σ changes then to 0.2 and the previous calculated outcome is taken as the prevBPM (as input for the filtering part). In the non-default state the calculated BPM is equal to the highest resulting peak after filtering outside the BPM tolerance interval, the maximum peak within the BPM-interval is taken. After every j'th window the peak tracker gets set to default state and

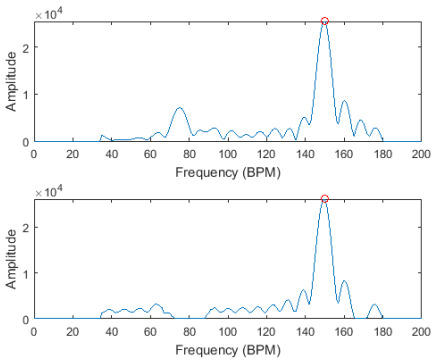
searches again for the maximum peak in order to retrack when BPM trace gets derailed of the real trace. However naturally the opposite can also occur, jumping out of the BPM trace when on the right track is sometimes inevitable.A good indicator of whether the SPT is on the right track or not has not been found yet, only assumptions can be made. 'j' can be seen as another variable parameter that is played with to check how the total average error differs. Also when the difference between the previous calculated BPM and the current calculated BPM is higher than 8 BPM the next state will be the default state. When difference is higher than 8 BPM there might be a chance that the trace is derailed from the track. This the reason that the next state is default state. At all the other times the SPT is in the non-default state.

5.3.4 Results

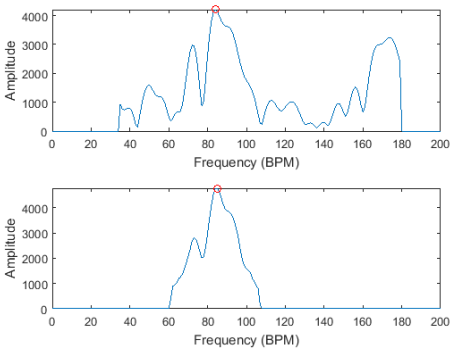
As an illustration are the results for the windows in Figure 5.1 shown in Figure 5.5



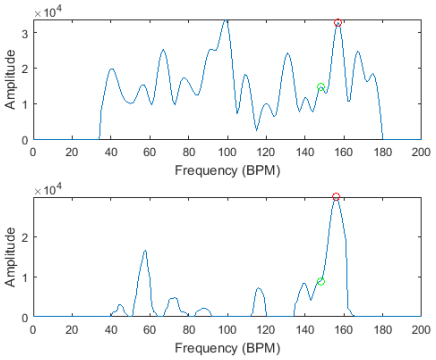
(A) Window where the MA are far from the true HR.



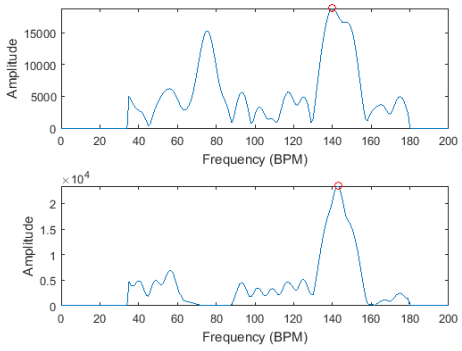
(B) Window where the frequency of the MA coincides with the true HR.



(C) Window where there is no MA.



(D) Window where there is no obvious peak at the true HR.



(E) Window where MA is close to the true HR.

FIGURE 5.5: The result of the second method considered for the situations in Figure 5.1

These results are based on the fact that the previous calculated BPMs are true and that the SPT function is in the non-default state with σ equal to 0.2. The red dots are the highest peaks and the green dots are where the real HR lies. As it is expected in Figure 5.5 the first four cases seem to work fine. These are the cases when MA peaks are far from the heart peaks, when MA coincides with the heart peak, when there is not a visible MA peak and when MA peak is close to the HR peak. These are the cases where

the artifact peaks can be filtered out easily.The peaks that correspond with the HR are the highest peaks. When there is not a visible peak at the HR problems may occur with determining HR. However the peaks that are close to the HR can still be identified as the calculated HR peak by the SPT. When the previous window is accurately calculated.

5.4 Temporal Difference

In the previous sections it can be noticed that both methods do not always remove all artifacts in the signal. This can be improved by using temporal difference. Temporal difference is the name given to the operation that returns the difference between sequential values of a signal. If $y = [h(1), h(2), \ldots, h(M)]$, then the first order difference is defined as $y' = [h(2) - h(1), h(3) - h(2), \ldots, h(M) - h(M-1)]$. The second order difference for y is the first order difference of y'. As long as k is not large, the k-th order difference maintains the fundamental and harmonic frequencies of the signal. This means that non-periodic frequencies will be removed. Because the PPG component that is associated with the HR is approximately periodic in a short time window and MA is generally aperiodic, the temporal difference is applicable here. Figure 5.6 shows an example of where the maximum peak in the frequency domain does not belong to the HR. The true HR here is 103 BPM. When the temporal difference operation is performed on the signal, that peak is weakened. The peak that belongs to the HR is now the maximum.

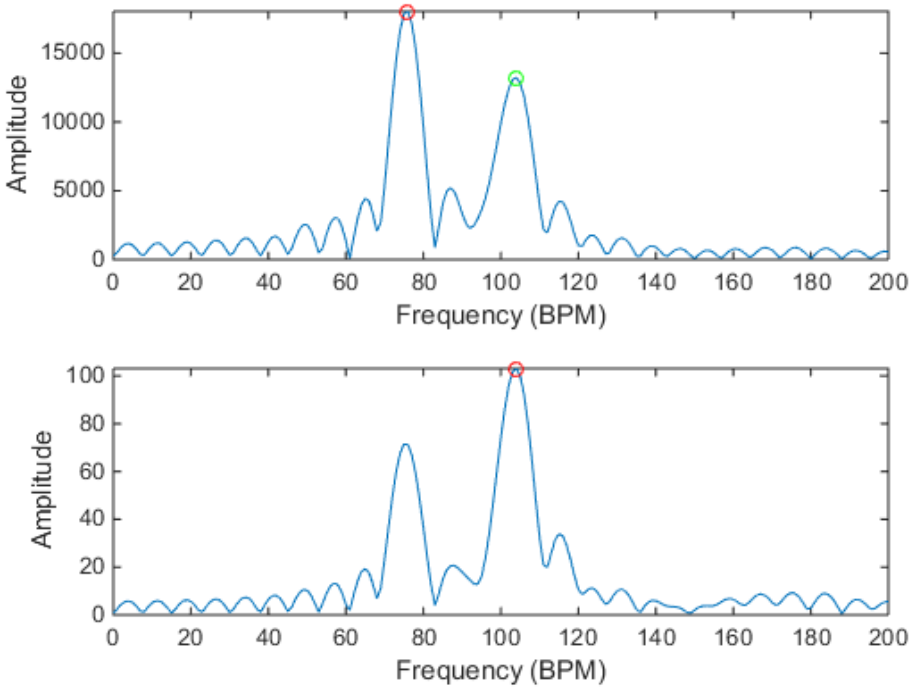


FIGURE 5.6: Example of temporal difference. First plot shows the reconstructed signal, second plot shows the reconstructed signal after temporal difference.

Chapter 6

Final Results

To check the performance of the whole system, SSR and SPT are connected with SSA. The 12 data sets will be ran through our implementation of TROIKA and will be compared to the acquired values from TROIKA [4]. The average error of one data set will be calculated according to [15] in the following way:

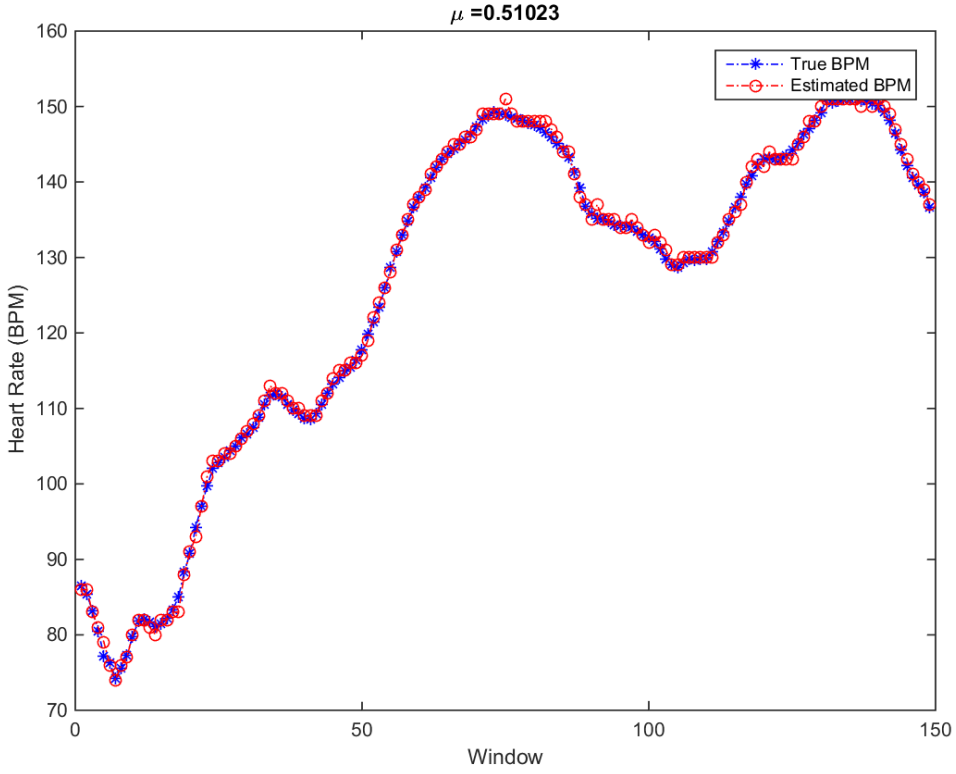
$$\mu = \frac{1}{N} \sum_{t=1}^{N} |BPM_{est}(i) - BPM_{true}(i)| \tag{6.1}$$

where,

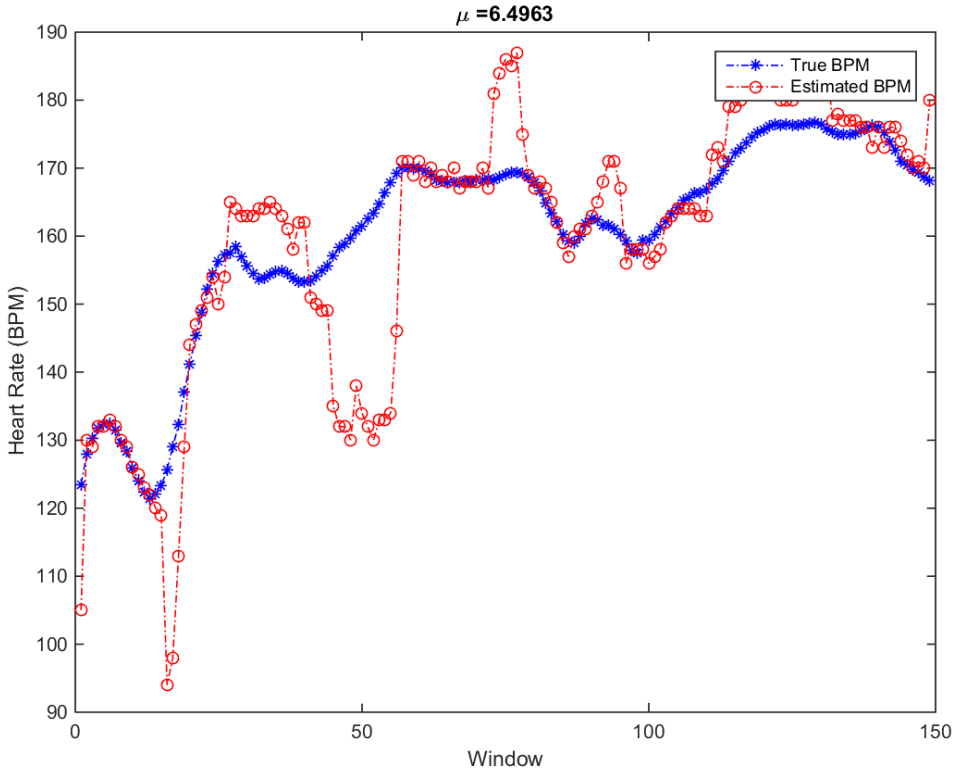
1. N is the total number of estimates, which is equal to the number of time windows.
2. $BPM_{true}(i)$ denotes the ground-truth of the heart rate value in the i-th time window in BPM, which was included in the data sets.
3. $BPM_{est}(i)$ denotes the corresponding estimate in BPM.

6.1 Results from TROIKA

Figures 6.1 and 6.2 show the best and worst obtained results using respectively the first and second method. The results of the average errors for all data sets for different situations are shown in Table 6.1.

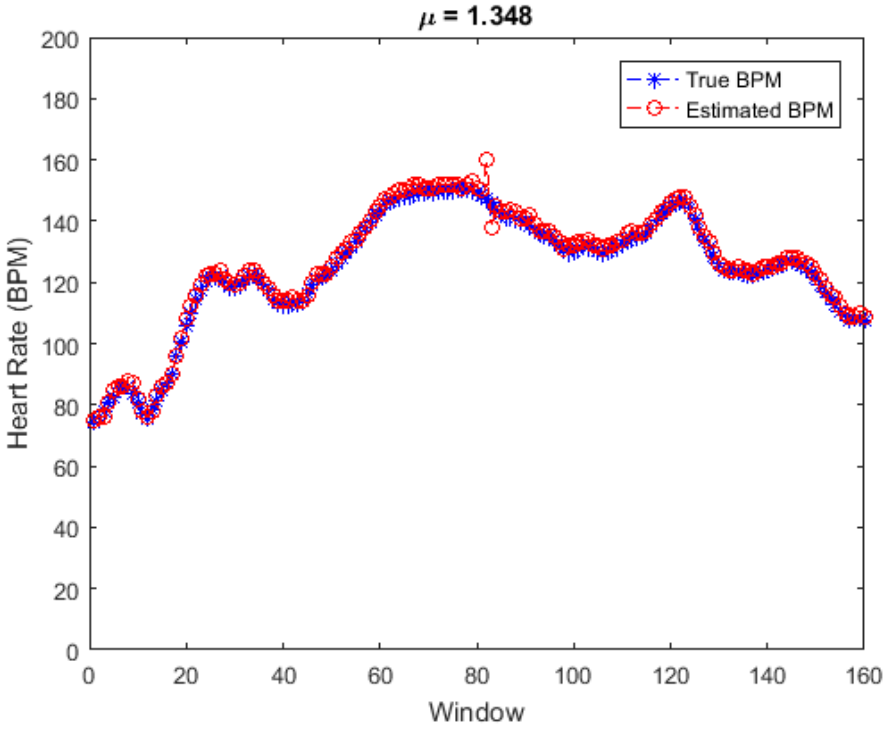


(A) Data set 9.

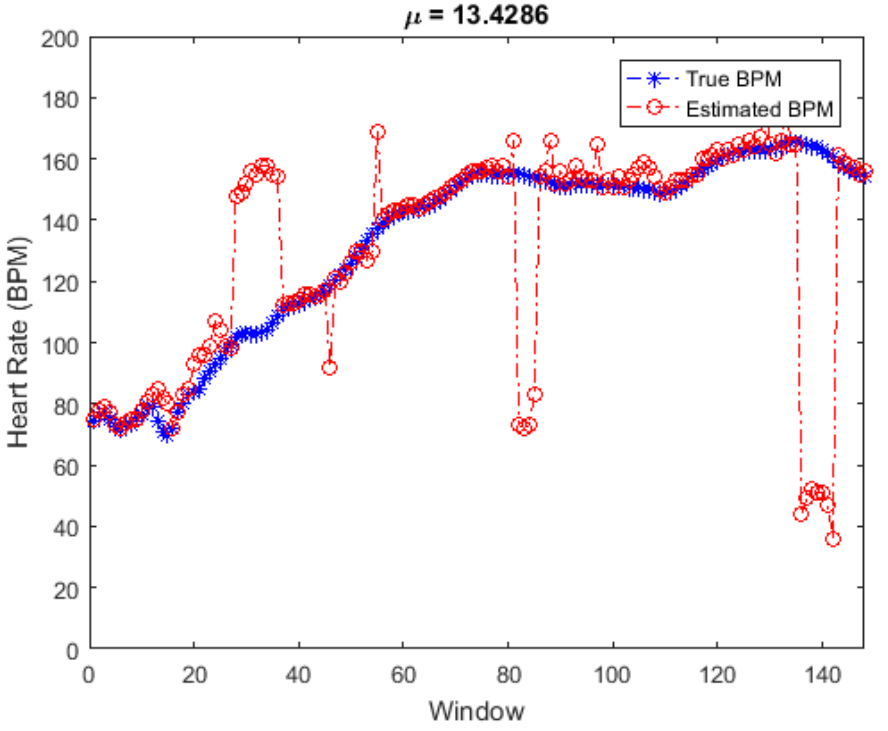


(B) Data set 10.

FIGURE 6.1: Predictions made using the RoC method compared to the ground-truth of the heart rate value for the best and worst data sets.



(A) Dataset 8



(B) Dataset 1

FIGURE 6.2: Predictions made using the FoC method compared to the ground-truth of the heart rate value for the best and worst data sets.

TABLE 6.1: The results of the implementation of TROIKA versus the results of the TROIKA implementation in [4].

Data set	1	2	3	4	5	6	7	8	9	10	11	12	**Average**
without SSA	6.31	29.12	2.82	1.75	0.92	2.68	0.90	0.92	0.78	23.80	1.94	9.89	**6.82**
using RoC	3.01	2.50	2.22	1.48	0.68	1.13	0.66	0.73	0.51	6.50	1.13	1.77	**1.86**
using FoC	13.53	6.85	2.29	2.01	1.78	4.06	3.61	1.34	1.36	11.81	2.88	2.98	**4.54**
TROIKA [4]	2.29	2.19	2.00	2.15	2.01	2.76	1.67	1.93	1.86	4.70	1.72	2.84	**2.34**

6.2 Results compared to other implementations

To further measure the performance of our implementation of TROIKA, the results will be compared to some other algorithms that were implemented by the other subgroups. These algorithms are JOSS and SMART [14]. Also the execution time for each implementation is given, which were calculated by running the code on a desktop computer. These results can be seen in Tables 6.2 and 6.3.

TABLE 6.2: The average error of our implementation of TROIKA versus some other implementations made by the other subgroups.

Data set	1	2	3	4	5	6	7	8	9	10	11	12	**Average**
ANC with SPT	1.79	1.09	0.66	1.09	0.84	1.14	0.95	0.62	0.44	3.92	5.87	1.46	**1.66**
TROIKA	3.01	2.50	2.22	1.48	0.68	1.13	0.66	0.73	0.51	6.50	1.13	1.77	**1.86**
JOSS	2.26	2.65	1.03	1.21	0.59	1.18	0.67	0.55	0.57	3.67	0.78	1.48	**1.39**
SMART	2.59	2.52	0.74	1.06	0.73	1.19	0.61	0.47	0.46	4.09	0.88	1.68	**1.42**

TABLE 6.3: Comparison between the execution times of each implementation. The execution times are the times that it had taken the algorithm to go trough all 12 data sets.

	Execution time (s)
ANC	42
TROIKA	183
JOSS	186
SMART	37

Chapter 7

Conclusion

7.1 Conclusion

This thesis discussed a signal processing technique, TROIKA, that removes motion artifacts from PPG signals. TROIKA is a framework that consists of three parts: signal decomposition, sparse signal reconstruction and spectral peak tracking. Removing motional artifacts by signal decomposition has been the main highlight of this thesis. The first step in the progress was to decompose the PPG signal using SSA. For this to work efficiently the parameters in the algorithm had to be optimized. These were the window length and the number of eigentriples to consider in the reconstruction of the signal. After optimizing those parameters, SSA proved to be a robust algorithm to decompose PPG signals fast and efficiently. However it was not obvious which component in the decomposition of the signal by SSA belonged to the heart rate. Two methods had been proposed to overcome this problem. The first method was the RoC method, which removes components that have a peak at the same frequency as the peaks in the acceleration data. The second method, FoC, used a slightly different approach. It removed peaks from the components that corresponded to peaks in the acceleration data by filtering and used a different SPT. Both methods showed that they were able to remove most of the motional artifacts in various situations. However the results from TROIKA when using the RoC method were better in terms of the average error, which was 1.86 BPM over all 12 data sets. These results were even better than the TROIKA implementation in [4].

The results from the implemented TROIKA were not as good as that from JOSS. Despite the fact that the execution times of TROIKA and JOSS were about the same, JOSS had a lower final average. This is why JOSS was chosen to be the basis of SMART, which proved to be the best implementation to remove motional artifacts accurately and

fast. The implementation of TROIKA satisfies each requirement in the programme of requirements. Because SMART gives even better results, it also satisfies every requirement.

7.2 Recommendations and future work

As it is seen in the results the RoC method shows better results than FoC method. Thus it can be said that the FoC method needs some improvement in order to show the same or even better results than RoC. The part of the SPT for FoC can be improved. When looking at the resulting plots of the datasets for FoC we can see that the plot derails in some rechecking cases from the true trace. With an improved "on track" checker can the SPT function more accurately. With rechecking is meant that the whole spectrum is searched for the heart rate instead of the BPM tolerance interval. The current checker rechecks every 'j' seconds the whole spectrum again. If there is a better way to check the track instead of rechecking every given timeslot the error rate could be reduced. There could be various ways to do this, the SPT of RoC looks whether there are dominant peaks within the BPM tolerance. However this is not reliable for the SPT of FoC because all the resulting dominant peaks are within the BPM tolerance. So there should be looked for other alternatives. For future work an improvement of the SPT is necessary for the FoC method.
The data sets that were used in this research only consisted of running exercises. For future research on this subject, more sport activities can be looked upon. Sports like boxing or rowing have different characteristics when it comes to motions of the arm. This can affect the measurements of the PPG signals in a different way than when doing running exercises.
One other thing that can be looked at in future research is to lower the computational load of the TROIKA framework. When comparing TROIKA to SMART and ANC, the execution time of TROIKA is around four times higher. This means that TROIKA would use more power for the same calculation.

Appendix A

Matlab code

```
function [ y ] = BP( x, fs, BPM )
%BP Summary of this function goes here
%   Detailed explanation goes here
    x = x ;
    N = length(x);
    dt = 1/fs;
    t = dt*(0:(N-1));
    count1 = timeseries(x,t);

    %countmean = mean(count1);

    if BPM == 0
        interval = [0.6 3.5];

    else
        fbpm = BPM/60;
        interval = [fbpm - 0.3, fbpm + 0.3];
    end

    filteredSig = idealfilter(count1,interval, pass );
    y =  filteredSig.data;
    y=y ;
end
```

Listing A.1: Matlab code the bandpass filter

```
function [yrecon, locs_MA] = SSA(x1, Fs, prevBPM, s)
% This function decomposes a signal using SSA.
%   INPUTS:
%       x1 = input signal (including acceleration data)
%       Fs = sample rate
%       prevBPM = previous estimated HR
%       s = number of PPG signal
%   OUTPUTS:
%       yrecon = reconstructed signal
%       locs_MA = locations of the maximum peaks of the acceleration data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialization
L = 215; %window length
N = length(x1(1,:)); %length of signal
K = N-L+1;
X = zeros(L,K);
I = 9; % Amount of eigenvalues taken
X_group = cell(I,L,K);

% Put the PPG signal in x
x = x1(s,:);

% Step1 : Build trayectory matrix
for i=1:L
    X(i,1:K) = x(1,i:i+K-1); % trajectory matrix
end

% Step 2: SVD
   S = X*X';
   [U,autoval] = eig(S);
   [d,i] = sort(-diag(autoval)); % sort eigenvalues to descending order
   d = -d;
   U = U(:,i); % sort left-singular vectors according to the eigenvalues
   V = (X')*U; % calculate right-singular vectors

% Step 3: Grouping
   Vt = V';
   i = 1;
   groupnumber = 1;
   while(i <= I)
      % if two eigenvalues are close to each other, group them together
      if abs((d(i)-d(i+1))/sum_eig) < 0.02
          X_group{groupnumber} = U(:,i)*Vt(i,:) + U(:,i+1)*Vt(i+1,:);
          i = i+1;
      else
          X_group{groupnumber} = U(:,i)*Vt(i,:);
```

```
      end
      groupnumber = groupnumber+1;
      i = i+1;
   end

% Step 4: Reconstruction
   Lp = min(L,K);
   Kp = max(L,K);

   % diagonal averaging
   y = zeros((groupnumber-1), N);
   for i=1:(groupnumber-1)
      y(i,:) = hankelize(X_group{i}, Lp, Kp, N);
   end

% Remove MA
   % put acceleration data into a new matrix
   acc = x1(s+2:s+4, :);

   % determine components that have to be excluded from reconstruction
   [deleteComponents, locs_MA] = RoC(acc, y, Fs, prevBPM);
   yrecon = zeros(1,N);

   % Recontruct the signal without the MA components
   if(length(deleteComponents) == (groupnumber-1))
       deleteComponents = 0;
   end
    for i=1:(groupnumber-1)
      if(~any(i == deleteComponents))
        yrecon = yrecon + y(i,:);
      end
    end

%Temporal Difference

    % don t do temporal difference if the signal is still the original
    if(length(deleteComponents) >= (groupnumber-3))
        yrecon=diff(yrecon);
        yrecon=diff(yrecon);
    end
```

Listing A.2: Matlab code for SSA

```
function y = hankelize(X_group, Lp, Kp, N )
% This function calculates a time signal from a matrix
%   INPUTS:
%       X_group = grouped matrices
%       Lp = minimum of L and K
%       Kp = maximum of L and K
%       N = length of signal
%   OUTPUTS:
%       y = reconstructed time signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   y=zeros(N,1);
   for k=0:Lp-2
     for m=1:k+1;
      y(k+1)=y(k+1)+(1/(k+1))*X_group(m,k-m+2);
     end
   end

   for k=Lp-1:Kp-1
     for m=1:Lp;
      y(k+1)=y(k+1)+(1/(Lp))*X_group(m,k-m+2);
     end
   end

   for k=Kp:N
      for m=k-Kp+2:N-Kp+1;
       y(k+1)=y(k+1)+(1/(N-k))*X_group(m,k-m+2);
     end
   end

end
```

Listing A.3: Matlab code for diagonal averaging

```
function [deleteComponents, locs_MA] = RoC(acc_data, y, Fs ,...
    prevBPM)
% This function identifies the components that belong to MA.
%   INPUTS:
%       acc_data = the 3 acceleration data (x, y and z)
%       y = components of a signal
%       Fs = sample rate
%       prevBPM = previous estimated HR
%   OUTPUTS:
%       deleteComponents = a list containing the numbers of the components
%                          that have to be excluded from reconstruction
%       locs_MA = locations of the maximum peaks of the acceleration data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialization
locs_MA = [];
index_MA = [];
locs_y = zeros(1,length(y(:,1)));
deleteComponents = [];
N = 60*Fs;
MA_tol= 5;
BPM_tol = 11;

f = (0:Fs/(N-1):Fs)*60; % define frequency axis

% Find MA peaks
for i=1:length(acc_data(:,1))
    X(1,:) = abs(fft(acc_data(i,:),N)); % calculate FFT of acceleration
    X(1,:) = X(1,:)/max(X(1,:)); % normalize the signal

    H = Hfilter(Fs,N); % create filter

    X = X.*H; % remove frequencies outside humanly possible heart rates

    K = kurtosis(X(1,:));
    % if signal is not noisy, find 3 peaks that are larger than
    % 50% of the maximum peak.
    if(K>130)
        [pks,index_MA] = findpeaks(X,  sortStr ,  descend );
        for m=1:3
            if pks(m) >= 0.5
                % put the locations of the peaks in a vector
                locs_MA = [locs_MA f(index_MA(m))];
            end
        end
    end
end
```

```
locs_MA = unique(locs_MA); % remove duplicates

emptiness = isempty(locs_MA); % check if there are peaks
k = 1;
max_HR = 210;

% Determine the components that have to be removed
for i=1:length(y(:,1))
    Y(1,:) = abs(fft(y(i,:), N)); % calculate FFT of the component
    [~,index_y] = max(Y(1,1:max_HR)); % find the biggest peak
    locs_y(i) = f(1,index_y); % find the frequency of the biggest peak

   % if there is no previous estimate, remove all components that have a
   % peak at the frequencies of the MA peaks
   if(prevBPM<=0)
       if(emptiness == 0)
           for j=1:length(locs_MA)
               if(locs_y(i) > locs_MA(j) - MA_tol && locs_y(i) < ...
                       locs_MA(j) + MA_tol)
                   deleteComponents(k)=0;
                   k=k+1;
               end
           end
       end
   % else remove the components that have a peak at the frequencies of the
   % MA peaks, except when they re near the frequency of the previous
   % estimate of the HR
   else
       if(locs_y(i) < (prevBPM-(BPM_tol*4)) || locs_y(i) > ...
               (prevBPM+(BPM_tol*4)))
         deleteComponents(k)=i;
         k=k+1;
       else
         for j=1:length(locs_MA)
           if(locs_y(i) > locs_MA(j) - MA_tol && locs_y(i) < locs_MA(j)...
                   + MA_tol && (locs_y(i) < prevBPM -(BPM_tol) || ...
                   locs_y(i) > prevBPM + (BPM_tol)))
               deleteComponents(k)=i;
               k=k+1;
           end
         end
       end
   end

end
deleteComponents = unique(deleteComponents); % remove duplicates
```

LISTING A.4: Matlab code for the RoC method

```
function [ H ] = Hfilter( Fs, N )
% Simple bandpass filter that removes every frequency outside 0.6 and 3 Hz.
%   INPUTS:
%       Fs = sample frequency
%       N = length of signal
%   OUTPUTS:
%       H = filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    fmin = 0.6;
    Nmin = floor((fmin/Fs)*N);

    fmax = 3;
    Nmax = ceil((fmax/Fs)*N);
    range1 = [Nmin, Nmax];

    H = zeros(1,N);
    for j = range1(1):range1(2) %Make filter
        H(j) = 1;
    end

end
```

Listing A.5: Matlab code for Hfilter

```
function[y_new] =FoC(acc_data,y,Fs ,prevBPM ,sigma,MA_tol,BPM_tol)
% This function identifies the peaks that belong to MA.
%   INPUTS:
%       acc_data = the 3 acceleration data (x, y and z)
%       y = components of a signal
%       Fs = sample rate
%       prevBPM = previous estimated HR
%       sigma=ratio for which acceleration data will be considered
%       MA_tol= the interval where the acceleration data and the PPG-signal
%       data must coincide (MA tolerance)
%       BPM_tol= the interval where the PPG signal peaks do not get
%       filtered out (BPM tolerance)
%   OUTPUT:
%       y_new=the filtered PPG signal.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialization

locs_MA = [];
index_MA = [];
locs_y = [];
PeakFilter=[];
N = 60*Fs;
BPM_tol_min=BPM_tol;
BPM_tol_plus=BPM_tol;
y_new=[];


f = (0:Fs/(N-1):Fs)*60; % define frequency axis

for i=1:length(acc_data(:,1))
    X(1,:) = abs(fft(acc_data(i,:),N)); % calculate FFT of acceleration


% Make filter to remove frequencies outside human heart rate interval
    fmin = 0.6;
    Nmin = floor((fmin/Fs)*N);

    fmax = 3;
    Nmax = ceil((fmax/Fs)*N);
    range1 = [Nmin Nmax];

    H = zeros(1,N);
    for j = range1(1):range1(2)

        H(j) = 1;
    end
```

```
    X = X.*H;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Find peaks of acceleration data and apply sigma criterion and kurtosis
% criterion
   [indV,indF] = findpeaks(X(1,:), SortStr , descend ); %find peaks
    index_MA = indF(1:length(indF));
    value_MA=indV(1:length(indV));
         for criterion=1:length(value_MA)
              if (value_MA(criterion) <= max(sigma*value_MA))
                  index_MA(criterion) = 0;
              end
         end

    index_MA=index_MA(index_MA~=0);




    K = kurtosis(X(1,:));

    if(K>130)
          locs_MA = [locs_MA f(index_MA)];
   end
end



locs_MA = unique(locs_MA); % Remove duplicates
emptiness = isempty(locs_MA);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%find peaks of components and filter them out after determined which peaks
%have to be filtered out

for i=1:length(y(:,1))
    Y(1,:) = abs(fft(y(i,:), N));
    Y(1,:) = Y.*H;
    [~,index_y]=findpeaks(Y(1,1:250), SortStr , descend );

    PeakFilter=ones(1,length(Y(1,:)));

    locs_y(1,1:length(index_y))=f(1,index_y);
```

```
for l=1:length(f(1,index_y))

    %%initial situation also known as the deafult state
if(prevBPM==0)
       if(emptiness == 0)
           for j=1:length(locs_MA)

               if(locs_y(1,l)>locs_MA(j)-MA_tol && locs_y(1,l)< ...
                   locs_MA(j)+MA_tol)
                     Filter = FilterPeak(Y,locs_y(1,l),prevBPM);
                     PeakFilter=Filter.*PeakFilter;
               end

           end

       end



else

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%non-default state
for j=1:length(locs_MA)

               if(((locs_y(1,l)>locs_MA(j)-MA_tol && locs_y(1,l)< ...
                   locs_MA(j)+MA_tol)) && ((locs_y(1,l)< ...
                   prevBPM-BPM_tol_min || ...
                   locs_y(1,l)>prevBPM+BPM_tol_plus)))

                     Filter = FilterPeak(Y,locs_y(1,l),prevBPM);
                     PeakFilter=Filter.*PeakFilter;
               end
 end
 end
end
 Y=PeakFilter.*Y;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Convert new PPG signal back to time domain

y_gefilterd = (ifft(Y));

if isempty(y_new)==1
    y_new=y_gefilterd;
```

```
else
    y_new=(y_new+y_gefilterd);
end


end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Listing A.6: Matlab code for the FoC method

```
function [ Sig_filtered ] = PeakBandpassFilter( Sig,loc )
% This function is written in order to filter indexwise the peaks out
%   INPUTS:
%      Sig:Signal in the frequency domain
%      Loc:location of the peak
%   OUTPUTS:
%      Sig_filtered: indices that need to be zero d
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Z=zeros(1,length(Sig));
rechterSchuiver=loc;
while(Sig(rechterSchuiver) >Sig(rechterSchuiver+1) ...
    && rechterSchuiver <= length(Sig))
    rechterSchuiver=rechterSchuiver+1;

end

linkerSchuiver = loc;
while(Sig(linkerSchuiver) >Sig(linkerSchuiver+1) && rechterSchuiver >= 0)
        Z(linkerSchuiver)=1;
        linkerSchuiver=linkerSchuiver-1;
end
Sig=Z.*Sig;


Sig_filtered=Sig;




end
```

Listing A.7: Matlab code for the peak filtering for FoC method

Bibliography

[1] A. Maziar Zafari Nima Ghasemzadeh. A brief journey into the history of the arterial pulse. *SAGE-Hindawi Access to Research*, 2011(1):1–15, 2011.

[2] Byung S. Kim and Sun K. Yoo. Motion artifact reduction in photoplethysmography using independent component analysis. 2006.

[3] Sarah Ostadabbas Rasoul Yousefi, Mehrdad Nourani and Issa Panahi. A motion-tolerant adaptive algorithm for wearable photoplethysmographic biosensors. 2014.

[4] Zhilin Zhang. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. 2015.

[5] Zhilin Zhang. Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction. 2015.

[6] Jianchu Yao and Steve Warren. A short study to assess the potential of independent component analysis for motion artifact separation in wearable pulse oximeter signals. 2005.

[7] Carli Velocci. Your fitbit might not be as accurate as you would think, may 2016. URL http://gizmodo.com/your-fitbit-might-not-be-as-accurate-as-you-might-think-1778055574.

[8] Hossein Hassani. Singular spectrum analysis: Methodology and comparison. 2007.

[9] C.J. James M.E. Daviesa. Source separation using single channel ica. 2007.

[10] Patrick Flandrin Gabriel Rilling and Paulo Gon¸calves. On empirical mode decomposition and its algorithms. 2003.

[11] Sapna Prabhu Mitali R. Ambekar. A novel algorithm to obtain respiratory rate from the ppg signal. 2015.

[12] Hossein Hassani. A brief introduction to singular spectrum analysis.

[13] Wayne. Where's the magic? (emd and ssa), 2013. URL http://www.r-bloggers.com/wheres-the-magic-emd-and-ssa-in-r.

[14] Julio Cesar Ballesteros Borrero and Tiamur Ali Khan. Heart rate monitoring using ppg signals. 2016.

[15] Zhilin Zhang. 2015 ieee signal processing cup, 2015. URL http://www.signalprocessingsociety.org/spcup2015/.