M.Sc. Thesis

Multiway Component Analysis for the Removal of Far Ventricular Signal in Unipolar Epicardial Electrograms of Patients with Atrial Fibrillation

Jelimo Bettina Maswan B.Sc.

Abstract

Atrial fibrillation (AF) is one of the more common clinical arrhythmias with a high morbidity and mortality [1]. Despite this, the electrophysiological and pathological mechanisms associated with AF largely remain a mystery, encouraging the use of ever more sophisticated techniques to extract vital information for diagnostic and therapeutic purposes. Contamination by signals of ventricular origin is considered the main artifact present in high-resolution epicardial electrograms (EGMs) that hinders the accurate and efficient analysis of AF EGM datasets. Furthermore, the complexity and dynamism of AF signals calls for robust data analysis tools that can effectively reduce or remove ventricular activity (VA) while preserving the texture and morphology of atrial activity (AA). Multiway component analysis, specifically block term decomposition (BTD), proves useful for the decontamination of epicardial EGMs as demonstrated in this project by enabling the automatic estimation of VA on an electrode-by-electrode basis, which is thereafter temporally and/or power spectrally subtracted thus retaining AA at a relatively high accuracy. The performance of BTD compared to average beat subtraction (ABS) and the more restrictive canonical polyadic decomposition (CPD) is visually verified and numerically confirmed based on a set of key performance indices. Additionally, the technique is entirely data-driven i.e., does not depend on any statistical properties, but if/when available, can contribute to enhanced performance via the imposition of appropriate constraints in the tensor decomposition.

TUDelft

Multiway Component Analysis for the Removal of Far Ventricular Signal in Unipolar Epicardial Electrograms of Patients with Atrial Fibrillation

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Jelimo Bettina Maswan B.Sc.
born in Nairobi, Kenya

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Abstract

Atrial fibrillation (AF) is one of the more common clinical arrhythmias with a high morbidity and mortality [1]. Despite this, the electrophysiological and pathological mechanisms associated with AF largely remain a mystery, encouraging the use of ever more sophisticated techniques to extract vital information for diagnostic and therapeutic purposes. Contamination by signals of ventricular origin is considered the main artifact present in high-resolution epicardial electrograms (EGMs) that hinders the accurate and efficient analysis of AF EGM datasets. Furthermore, the complexity and dynamism of AF signals calls for robust data analysis tools that can effectively reduce or remove ventricular activity (VA) while preserving the texture and morphology of atrial activity (AA). Multiway component analysis, specifically block term decomposition (BTD), proves useful for the decontamination of epicardial EGMs as demonstrated in this project by enabling the automatic estimation of VA on an electrode-by-electrode basis, which is thereafter temporally and/or power spectrally subtracted thus retaining AA at a relatively high accuracy. The performance of BTD compared to average beat subtraction (ABS) and the more restrictive canonical polyadic decomposition (CPD) is visually verified and numerically confirmed based on a set of key performance indices. Additionally, the technique is entirely data-driven i.e., does not depend on any statistical properties, but if/when available, can contribute to enhanced performance via the imposition of appropriate constraints in the tensor decomposition.

Acknowledgments

I think of myself as a contradictory person; for one, although I like certainty, I often find myself drawn to spontaneity and a touch of disarray. It is with this same mindset that I began this research project, hoping to be challenged by the spontaneity and apparent disarray characteristic of epicardial electrogram signals in atrial fibrillation, while relying on the certainty of the wonderful support system I had at my disposal. Needless to say, I have not been disappointed in the least.

I would therefore like to express my sincerest gratitude to my supervisors, Prof. Alle-Jan van der Veen and Dr. Richard Hendriks, for their patience and guidance. It has been a pleasure to both work under and learn from them. I would also like to thank Dr. Natasja de Groot for her medical expertise and useful feedback, as well as Dr. Richard Heusdens for signing those all too many administrative forms. Many thanks to Bahareh Abdikivanani for bearing with my incessant questions, Minaskie Ramsoekh for her administrative prowess, the many friends I have made in the electrical engineering master program, and to the entire circuits and systems group at TU Delft for the interesting presentations and smiles shared along the corridors of EWI.

Lastly, I would like to thank my family for the emotional and financial support however far away they may be. You have always believed in me even when I doubted myself. In closing, I leave prospective readers of this thesis report with the wise words of Marilyn Monroe that embodied my research experience and guide my life on the whole: "You don't have to know where you're going to get exactly where you need to go." Thank you!

Jelimo Bettina Maswan B.Sc.
Delft, The Netherlands
26-09-2017

Contents

List of Figures

List of Tables

xiii

Introduction

<div style="text-align: right; font-size: 2em; font-weight: bold;">1</div>

Atrial fibrillation (AF) is the most common cardiovascular disease in clinical practice affecting up to 1% of the general population and up to 10% of the population over 70 years old with an incidence rate that is expected to double every decade [10]. Examples of major consequences linked to AF are an increased risk of mortality and strokes, and a poor quality of life [10]. Additionally, up to 60% of patients who undergo cardiothoracic surgery are vulnerable to AF-related complications which correspondingly increases the chances of morbidity and prolongs their stay at the hospital [11].

Although AF is one of the most commonly encountered arrhythmia in clinical practice, it is yet to be completely understood due to its complex pathophysiology with various triggers and substrates interacting in multiple ways, causing fibrillation waves that have various appearances ranging from narrow to broad, and disorganized to organized [10]. In order to better understand the pathophysiological mechanisms of AF, it is common to take invasive (epicardial or endocardial) atrial electrograms (EGMs). Epicaridal atrial electrograms are recorded on the surface of the atrium while endocardial atrial electrograms are recorded within the atrium.

AF is typically characterized by seemingly chaotic atrial activation with a cycle length of about 160 ms, and an irregular and frequently rapid ventricular response [10]. Consequently, the ventricular rate response varies in a rather unpredictable fashion, eventually resulting in a failure to restore and maintain sinus rhythm (SR), especially if it progresses from paroxysmal to persistent AF [10]. See Figure 1.1 for an illustration of atrial and ventricular activation in SR and in AF.

Clinically, the two key attributes currently used for characterizing AF are: 1) the dominant atrial frequency (DAF) i.e., the repetition rate of the fibrillatory waves, and 2) the organization of the AF i.e., the repetitiveness of the AF signal pattern. An estimation of the DAF is an indication of the atrial cycle length, and it has been demonstrated that AF recordings with a low DAF are more likely to terminate spontaneously and respond better to antiarrhythmic drugs or cardioversion, while a high DAF is more often associated with resistance to therapeutic interventions [12]. As for AF organization, the state of the arrhythmia can be inferred from the

<div style="text-align: center;">1</div>
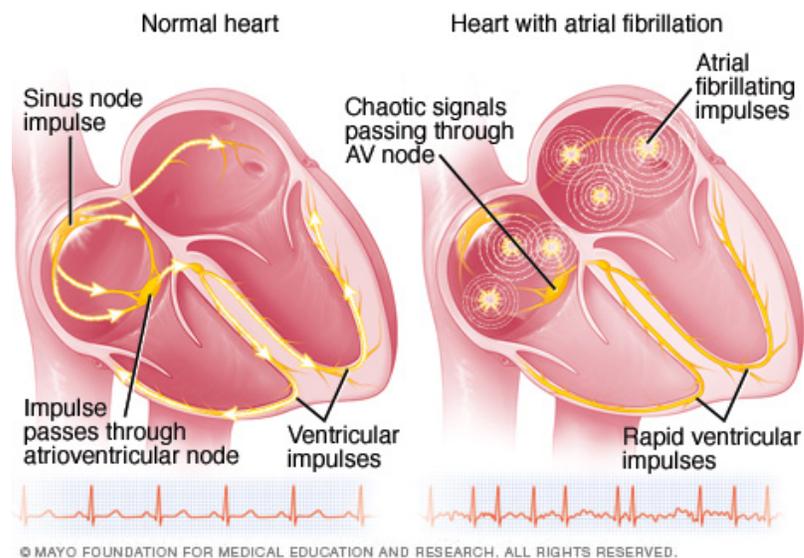
Figure 1.1: Illustration of atrial and ventricular activation rates in sinus rhythm and in atrial fibrillation [2].

repetitiveness of the AF signal pattern and may be used to determine appropriate AF treatment strategies such as catheter ablation and electrical cardioversion [13], [14]. Overall, the repetition rate of atrial waves is the most studied parameter with regard to clinical management, identification of pathomechanisms and evaluation of various treatment options [12].

Since the goal of recording surface ECG and/or invasive EGM is to understand the nature of AF, the presence of ventricular activity (VA) in the recordings is considered a contaminant signal that should be removed. However, due to the temporal, spectral and spatial overlap of atrial and ventricular activity, linear filtering techniques generally underperform and more advanced signal processing techniques such as nonlinear filtering and multidimensional signal processing are often necessary for the extraction of atrial signals during AF [14].

1.1 Thesis Motivation

Normally, a clinician attempts to visually identify and unmix atrial and ventricular sources using human reasoning. It would be a lot better if the process could be automated so as to [6]:

1. Unmix and isolate the atrial signal in atrial fibrillation into its constituent components

2. Provide information on the number of distinct components underlying the EGM measurements

3. Provide the spatial distribution of atrial activity besides the time series of the source itself

4. Track changes in the number, spatial distribution and morphology of atrial and/or ventricular activity over time

5. Determine the local activation time of atrial activity for the construction of activation time maps

The goal of this research project is to automatically extract the atrial activity from AF EGMs so that the characteristics of atrial electrograms can be studied without interference from ventricular activity. This separation is based on the fact that the two physiological signals originate from different, spatially isolated bioelectrical sources, and separation may exploit temporal redundancy among successive heartbeats as well as spatial redundancy when multichannel recordings are analyzed. See Figures 1.2 and 1.3 for examples of EGMs that are analyzed in this report, in SR and AF respectively.
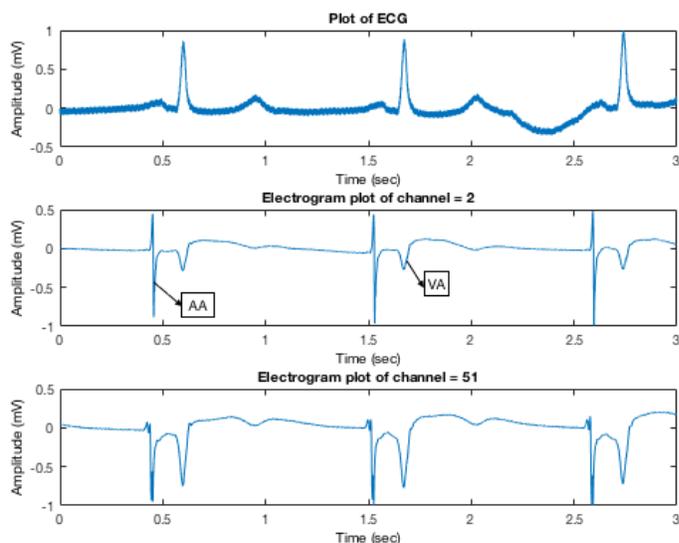
Figure 1.2: Epicardial electrograms in sinus rhythm.
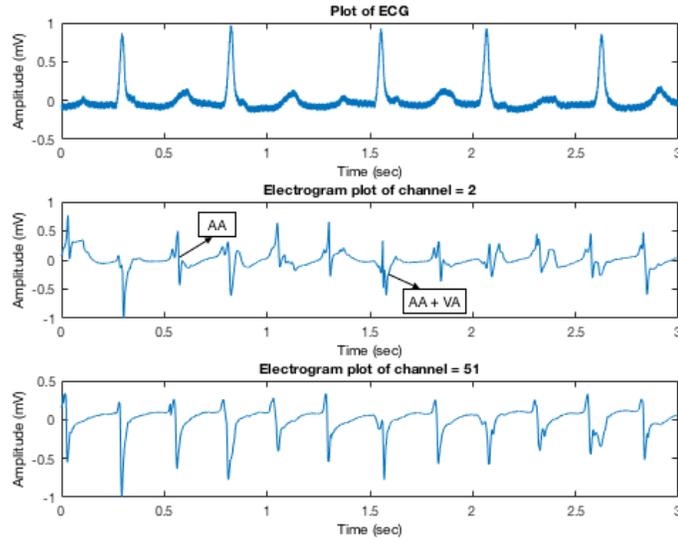
Figure 1.3: Epicardial electrograms in atrial fibrillation.

However, the inherent variability and complexity of the AF EGMs proves challenging, especially because an estimate of the AA is required on an electrode-by-electrode basis so as to capture the local atrial activation without VA at the various electrode array locations. See Figure 4.1 for the various locations at which the electrode array is placed to record EGMs during open chest surgery. Further details on the recording mechanism can be found in the chapter on "Experimental Methodology" under data acquisition.

Therefore, array processing techniques that exploit spatial diversity such as independent component analysis (ICA) prove unsuitable because ICA in its standard form can only extract as many sources as there are channels/electrodes i.e., to obtain 2 sources (AA and VA) using ICA, we would need at least 2 electrode recordings, such that one source would represent the AA in both electrodes and the other the VA. Therefore, to be more precise, ICA exploits spatial 'redundancy' across electrodes. While the number of signal recordings is not a problem in our case (we have plenty of data for each location), the very fact that the recorded AA varies from electrode to electrode implies that if we were to extract say 5 sources from 8 electrode recordings, then each source would be considered an 'average' of sources that are recorded by the 8 electrodes. Consequently, we would automatically lose part of the high-resolution data that was recorded. See Figures 1.5 and 1.6 to see an example of the variability present in AF EGM signals across 3 consecutive electrodes in 2 consecutive rows
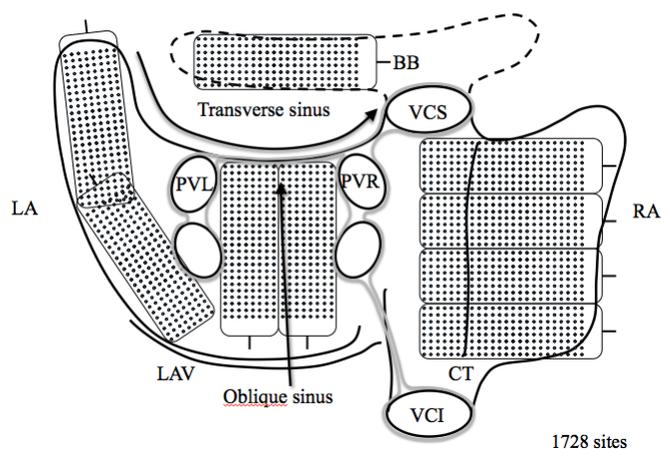
Figure 1.4: Image of the various EGM recording locations and the rectangular electrode array configuration.

recorded at the same electrode array location.
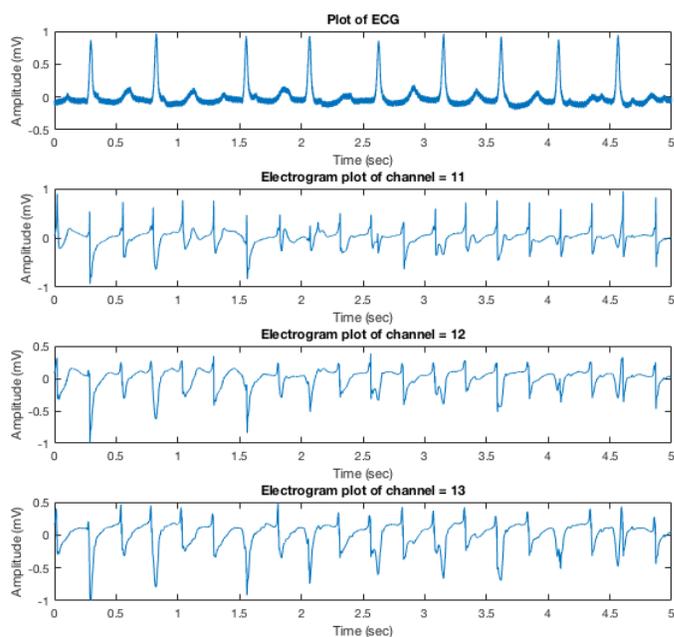
Figure 1.5: Variability of AF EGMs for electrodes 11, 12, 13 in row 2 at location BB0.

5
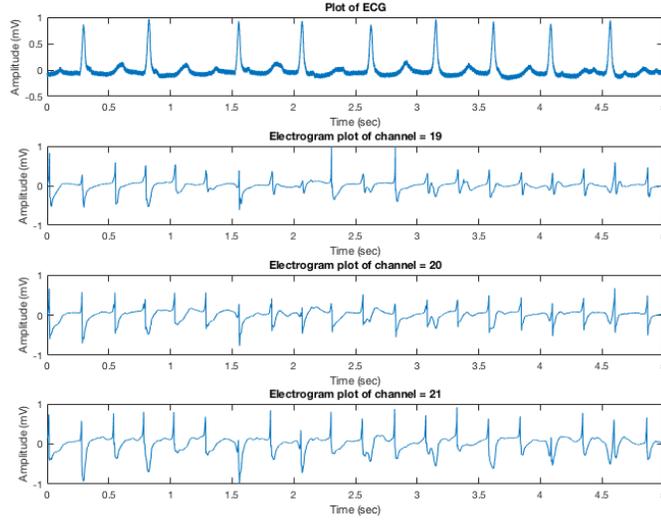
Figure 1.6: Variability of AF EGMs for electrodes 19, 20, 21 in row 3 at location BB0.

While the above mentioned restrictions are of concern, of even more significance is the fact that the dataset we are dealing with is new in the sense that there are no statistical studies that have been done on epicardial EGM signals to the best of our knowledge. This is probably in part due to the fact that every patient generates a unique set of AF signals, and though there may be a certain level of consistency across different patients such that it is possible to observe patterns and conclusively make statements about the statistical properties of AA and VA, this is yet to be published in the scientific literature. Therefore, while it is often assumed that during AF the AA and VA originate from independent sources and consequently have independent probability distributions such that standard approaches like ICA could work [15], [16], [10].

1.2 Mixing Model

As discussed in [17], it is known that the physiological mixing process that gives rise to the recorded AF EGM signals is quite complex. Not only are the observed recordings weighted and delayed, but each of the atrial source signals likely contributes to the sum with multiple delays corresponding to multiple paths by which the AF signals propagate from the sources to the recording location. Consequently, the filtered

sums of the source signals (AA and VA) results in a convolutive mixing process. The convolutive model is described by the following relation between the mth mixed signal, the original R source signals and additive sensor noise $v_m(t)$ [18]:

$$x_m(t) = \sum_{r=1}^{R} \sum_{l=0}^{L-1} a_{mrl} s_r(t-l) + v_m(t). \qquad (1.1)$$

The mixed signal is therefore a linear mixture of filtered versions of the source signals and a_{mrl} represents the mixing filter coefficients. While in reality the filter coefficients probably evolve in time, it is usually assumed that the mixing model is stationary. Additionally, we assume that the filters are of finite length i.e., $L < \infty$. The convolutive model can be written in matrix form as:

$$\mathbf{x}(t) = \sum_{l=0}^{L-1} \mathbf{A_l} \mathbf{s}(t-l) + \mathbf{v}(t). \qquad (1.2)$$

It is possible to simplify the convolutive mixing model by transforming the mixtures into the time-frequency (TF) domain where for small enough window frames, the signal can be assumed stationary, and the mixing model becomes instantaneous when the length of the discrete Fourier transform (DFT) $\gg$ window length. We also assume that the mixing process is noise-free such that we can drop the $\mathbf{v}(t)$ term.

The mixing model in each TF frame now becomes:

$$\mathbf{x}(k,t) = \mathbf{A}(k)\mathbf{s}(k,t), \qquad (1.3)$$

where k is a discrete frequency index, $\mathbf{A}(k)$ is a complex $M \times R$ matrix, $\mathbf{x}(k,t)$ is a complex $M \times 1$ vector and $\mathbf{s}(k,t)$ is a complex $R \times 1$ vector. Equation 1.3 describes an instantaneous mixing problem in each time-frequency bin.

The model just described is often assumed for blind source separation (BSS) of convolutive sources in the TF domain. However, we take a different spin on the instantaneous linear mixing model described by Equation 1.3 such that each TF bin is approximated by a linear combination of nonnegative rank-1 elementary spectrograms such that only one source is active in each TF bin. We stack the TF representations of each electrode recording on a row-by-row basis to form a three-way tensor, and jointly factorize the TF representations of the electrodes to extract

common latent components across the electrodes. The model is thus represented by:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{E_r} \circ \mathbf{c_r} = \sum_{r=1}^{R} \mathbf{a_r} \circ \mathbf{b_r} \circ \mathbf{c_r}, \tag{1.4}$$

where $\mathcal{X}$ is a mode-3 tensor of the TF representations of the EGM data, $\circ$ denotes the outer product, $\mathbf{E_r} = \mathbf{a_r} \circ \mathbf{b_r}$ is a rank-1 matrix, and vectors $\mathbf{a_r} \in \mathbb{R}^I$, $\mathbf{b_r} \in \mathbb{R}^J$ and $\mathbf{c_r} \in \mathbb{R}^M$. I is the length of the DFT, J is the number of window frames and M is the number of electrodes/recordings in a row of the electrode array.

Although it is sometimes the case that $\mathbf{E_r}$ is indeed a rank-1 matrix, this is fairly restrictive and we increase the assumed rank of the matrices $\mathbf{E_r}$ such that we perform a block term decomposition (BTD) in rank-$(L_r, L_r, 1)$ terms:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{E_r} \circ \mathbf{c_r}. \tag{1.5}$$

Our proposed approach is thus a factor analysis via BTD of the TF representation of each electrode across M electrodes in the same row to find an estimate of the VA. Our tensor model is illustrated in Figure 1.7 and this technique is often referred to as multiway linked component analysis.
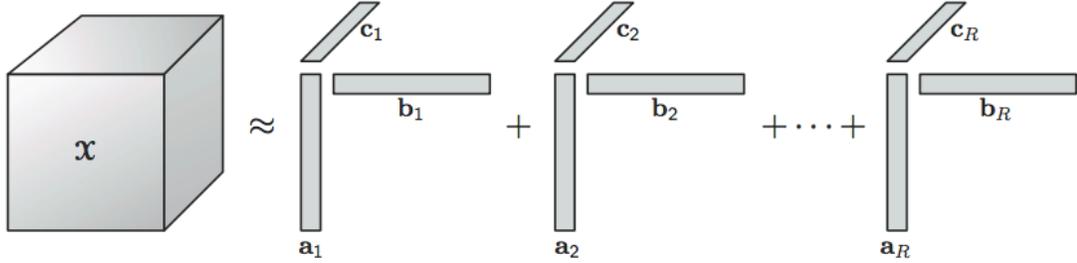
Figure 1.7: Illustration of the data model for factor analysis of AF EGMs[3].

1.3 Thesis Contributions

The bulk of this thesis report focuses on multilinear data analysis via tensor decompositions for the extraction of ventricular activity. The selected method, block term

decomposition, is inspired in part by the following key constraints: 1) complexity and variability of AF EGMs, 2) preservation of high-resolution data, and 3) lack of definitive AA and VA statistical properties.

In summary, we basically decompose the time-frequency tensorial representation of the AF EGM datasets while assuming that each source is characterized by specific temporal and spatial signatures, and originates from one direction. In this way, each of the the latent components represented by the *temporal × spectral* matrices are projected to a specific spatial subspace. This exploits both the spatial redundancy of the EGM recordings and the local variance in AA across electrodes in that the VA is assumed to originate from one direction across a row of the array while the AA originates from another, if not multiple, direction(s). By imposing a rank-1 constraint in the spatial mode of the tensor formed by the *time × frequency × channel* representation of the AF EGM datasets, we only allow one source from a particular direction to be active in each extracted component. Constraining the other modes to be rank-1 as well may work if there is only one source active in the time and/or frequency domain of the dataset. However, given the temporal and/or spectral overlap of AA and VA such that there is likely to be more than one active source in the TF representation, it is preferable to impose a low-rank constraint.

The main contributions of this thesis project are multilinear subspace techniques via tensor decompositions to obtain an estimate of the VA, temporal and/or power spectral subtraction to reduce or completely remove the VA, synthesis of pseudoreal AF EGM signals with which to evaluate the performance of the selected algorithm, and an extensive compilation of performance metrics to quantitatively measure the extent to which VA has been removed from the AF EGMs on a electrode-by-electrode basis. As an added plus, the technique is almost completely automatic, the 'almost completely' aspect due to the inherent variability in the EGM recordings requiring the user to initially manually determine the signal kurtosis range. Signal kurtosis is used to categorically identify the extracted components after tensorial decomposition (identifiability is problematic due to the permutation indeterminacy of factor analysis and blind source separation (FA-BSS) techniques). Besides this, certain inevitable implementational details like the FA-BSS scaling/counterscaling indeterminacy and unknown number of components necessitate a few algorithmic tricks as a workaround to still obtain good performance. Nevertheless, the results indicate sufficient (if not better) performance over other VA reduction and/or removal techniques in AF EGMs, as indicated by both visual inspection of the extracted AA and the evaluated performance indices.

1.4 Thesis Outline

This thesis report comprises seven chapters including the introduction. In chapter two, we give a brief overview of conventional VA reduction/removal techniques in ECG/EGM analysis as part of the background for this research project. Thereafter, in chapter three we cover multiway component analysis (MWCA) in detail, laying the groundwork for tensor decompositions. Chapter four goes over the experimental methodology i.e., the acquisition and preprocessing of the AF EGM data, as well as consolidating implementation details into a clear and concise step-by-step procedure. Chapter five discusses the performance metrics used to evaluate the capabilities of the selected algorithm, whose results are presented and analyzed in chapter six. Chapter seven then concludes the thesis report by reiterating the key aspects and significant results of this research project, and describes future research directions based on the results and insights obtained thus far.

While it has been our intention to minimize the number of mathematical definitions for ease of reading while still maintaining the required level of mathematical rigor, we may have unintentionally overlooked some mathematical intricacies that are of interest to some readers. For their satisfaction and those more interested in tensor decompositions, we have included an appendix containing other mathematical definitions related to this thesis project. The bibliography is also a good starting point to explore alternative applications of multilinear algebra.

Background

2

In this background section, factor analysis and blind source separation (FA-BSS) is introduced and traditional ventricular activity (VA) cancellation techniques, namely average beat subtraction (ABS) and adaptive filtering, as well as more recent matrix (two-way) FA-BSS techniques, specifically principal component analysis (PCA) and independent component analysis (ICA), are discussed. Multiway FA-BSS techniques aka tensor decompositions for the removal of VA, which underlie the main contributions of this thesis project, will be extensively explored in chapter three. However, before delving into previous work done on the cancellation of VA and the extraction of atrial and ventricular activity in surface ECG and invasive EGM recordings, we will first go over the mathematical notation that is consistently used throughout this thesis report.

2.1   Mathematical Notation

Scalars are denoted by lowercase italic letters $(a, b, ...)$, vectors by lowercase boldface letters $(\mathbf{a}, \mathbf{b}, ...)$, matrices by boldface capitals $(\mathbf{A}, \mathbf{B}, ...)$ and tensors by calligraphic letters $(\mathcal{A}, \mathcal{B}, ...)$. Italic capitals are used to denote index upper bounds $(i = 1, 2, ..., I)$. The entry with row index i and column index j in a matrix $\mathbf{A}$, i.e., $\mathbf{A}_{ij}$, is symbolized by a_{ij}. Similarly, we have $(\mathcal{A})_{i_1,i_2,...,i_N} = a_{i_1,i_2,...,i_N}$. The columns of $\mathbf{A}$ are denoted by $[\mathbf{a_1}, \mathbf{a_2}, ...]$.

For a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$, $\mathbf{A}_{:,:,i_3,...,i_N}$ represents the $I_1 \times I_2$ slice indexed by $i_3, ..., i_N$, and $\mathbf{A}_{i_1,...,i_{N-1},:,:}$ represents the $I_{N-2} \times I_N$ slice indexed by $i_1, ..., i_{N-2}$. The $(R \times R)$ diagonal matrix containing the values $a_1, a_2, ..., a_R$ is denoted by $diag(a_1, a_2, ..., a_R)$. The superscripts ^{T} and † denote the transpose and the Moore-Penrose pseudo-inverse respectively.

Additionally, some clinical letter sequences referring to fiducial points in ECG and EGM recordings are used in this report. The term QRS complex is often used to refer to ventricular activity. To better understand what is meant by these letter sequences, see Figure 2.1 for an illustration of these points on an ECG recording.
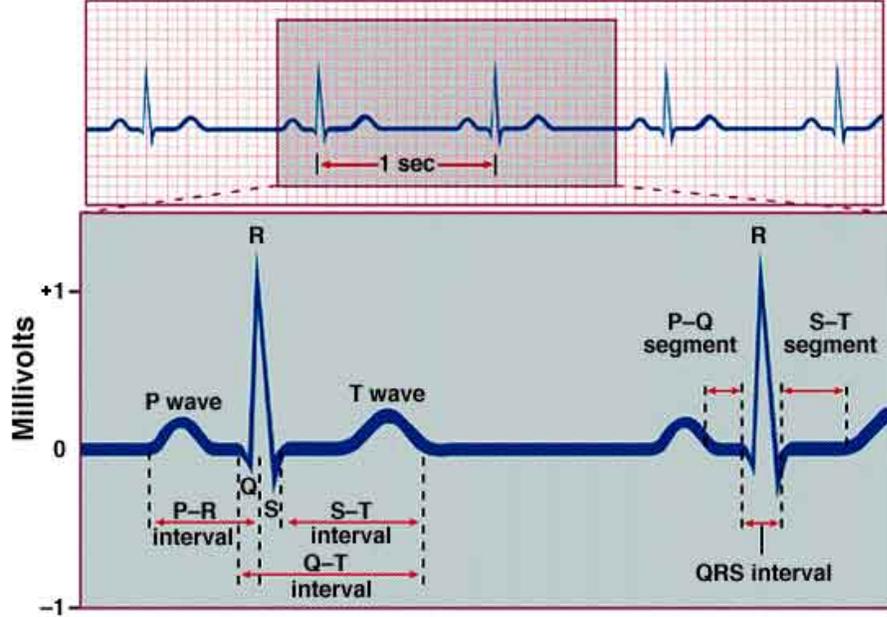
Figure 2.1: Fiducial points in an ECG recording [4].

2.2   Factor Analysis and Blind Source Separation

Factor analysis (FA), also known as component analysis, and blind source separation (BSS) are unsupervised learning methods with the goal of estimating R components represented in the matrix $\mathbf{S} \in \mathbb{R}^{T \times R}$ from the measurement matrix $\mathbf{X} \in \mathbb{R}^{I \times T}$ by decomposing $\mathbf{X}$ in R interpretable rank-1 components such that:

$$\mathbf{X} = \mathbf{A}\mathbf{S}^{\mathbf{T}} = \sum_{\mathbf{r}=1}^{\mathbf{R}} \mathbf{a_r}\mathbf{s_r^T} = \sum_{\mathbf{r}=1}^{\mathbf{R}} \mathbf{a_r} \circ \mathbf{s_r}, \tag{2.1}$$

where the outer product $\circ$ is defined in Table 2.1. $\mathbf{A} = [\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_R}] \in \mathbb{R}^{I \times R}$ is the unknown factor or mixing matrix (also sometimes called the basis matrix or dictionary depending on the application), $\mathbf{S} = [\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_R}] \in \mathbb{R}^{T \times R}$ is the matrix of loadings or sources (aka components or latent variables), and $1 \leq r \leq R$. In FA terminology, matrices $\mathbf{A}$ and $\mathbf{S}$ are the factor and loading matrices respectively, while in BSS terminology they refer to the mixing and source matrices respectively. Regardless of physical interpretation, the problem definition and proposed strategies to solve FA and/or BSS are essentially the same and from here onwards we shall

refer to them both as FA-BSS.

Although we explicitly assume that the columns of **S** are the loadings or source signals, the roles of **A** and **S** can be switched if we consider $\mathbf{X^T}$ in Equation 2.1. The term blind in BSS emphasizes the fact that nothing is known about the source signals or the mixing structure.

Without any constraint(s), the matrix factorization problem in Equation 2.1 is generally highly undetermined since it has an infinite number of solutions [9]. However, with some apriori information, FA-BSS enables the recovery of sources i.e.:

$$\mathbf{\hat{S}} = \mathbf{\Psi(X)} = \mathbf{S\Lambda P}, \tag{2.2}$$

where $\mathbf{\hat{S}}$ is an estimate of the loading/source matrix **S**, $\mathbf{\Psi}$ denotes a suitable FA-BSS algorithm, $\mathbf{\Lambda}$ is a diagonal scaling matrix, and **P** is a permutation matrix. Equation 2.2 clearly demonstrates the well-known unavoidable scaling ($\mathbf{\Lambda}$) and permutation (**P**) ambiguities of FA-BSS (nonuniqueness). In general, $\mathbf{\Psi}$ transforms the data matrix **X** into a representation that preserves as much of the information as possible while increasing the interpretability of the latent components. Examples of more popular FA-BSS techniques are Principal component Analysis (PCA) and Independent Component Analysis (ICA). In general, the type of information to be preserved determines the various FA-BSS criteria and methodologies [9]. For example, when using independent component analysis (ICA), a linear transformation of the matrix **X** is sought such that the latent components become statistically independent, representing an estimate of the source signals up to scaling and permutation indeterminacies. Alternatively, when using PCA, the transformation seeks components that are maximally uncorrelated.

More often than not, the data (epicardial EGMs in our case) is (or can be) organized as data matrices (or tensors), and described by linear (or multilinear) combination models such that FA-BSS boils down to decomposing the original data matrix (or tensor) into two (or more) factor matrices (the factor/mixing matrix **A** and loading/source matrix **S**). Although standard FA-BSS methods differ from each other because of the varying constraints imposed on the structure and properties of component matrices, there is no constraint on the sign of the elements in the factorized matrices and subtractive combinations (negative components) are generally allowed [19].

Table 2.1: Definitions of matrix and tensor products [9].

Matrix/Tensor Product	Definition
$\mathcal{C} = \mathcal{A} \times_n \mathbf{B}$	Mode-n product of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ and $\mathbf{B} \in \mathbb{R}^{J_n \times I_n}$ yields $\mathcal{C} \in \mathbb{R}^{I_1 \times ... \times I_{n-1} \times J_n \times I_{n+1} \times ... \times I_N}$
$\mathcal{C} = [\![\mathcal{A}; \mathbf{B^{(1)}}, \mathbf{B^{(2)}}, ..., \mathbf{B^{(N)}}]\!]$	Full multilinear product $\mathcal{C} = \mathcal{A} \times_1 \mathbf{B^{(1)}} \times_2 \mathbf{B^{(2)}} \ldots \times_N \mathbf{B^{(N)}}$
$\mathcal{C} = \mathcal{A} \circ \mathcal{B}$	Tensor or outer product of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times J_2 \times ... \times J_M}$ yields $\mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N \times J_1 \times J_2 \times ... \times J_M}$
$\mathcal{X} = \mathbf{a^1} \circ \mathbf{a^2} \circ \ldots \circ \mathbf{a^N}$	Tensor or outer product of vectors $\mathbf{a^{(n)}} \in \mathbb{R}^{I_n} (n = 1, ..., N)$ yields a rank-1 tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$
$\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$	Kronecker product of $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathbf{B} \in \mathbb{R}^{J_1 \times J_2}$ yields $\mathbf{C} \in \mathbb{R}^{I_1 J_1 \times I_2 J_2}$
$\mathbf{C} = \mathbf{A} \odot \mathbf{B}$	Khatri-Rao product of $\mathbf{A} = [\mathbf{a_1}, ..., \mathbf{a_R}] \in \mathbb{R}^{I \times R}$ and $\mathbf{B} = [\mathbf{b_1}, ..., \mathbf{b_R}]$ yields $\mathbf{C} \in \mathbb{R}^{IJ \times R}$ with columns $\mathbf{c_r} = \mathbf{a_r} \otimes \mathbf{b_r}$

2.3 Previous Work

2.3.1 Average Beat Subtraction

Average beat subtraction (ABS) was first proposed by Slocum et al. as a method of identifying P-waves (AA) during ventricular tachycharia [20]. It has since then become a standard technique for the removal of ventricular signal artifacts in both surface ECG and invasive EGM recordings [14]. The method exploits the fact that there is no fixed relationship between AA and VA, and that the QRS complex usually has a fairly consistent morphology across ECG leads or EGM electrodes [20], [14]. In ECG signal processing, an average beat that represents the ventricular cycle is obtained from each ECG lead and subtracted from the individual heartbeats. Ideally, the residual atrial signal only contains fibrillatory waveforms that can be further analyzed to better understand the nature of AF.

For the purposes of processing epicardial EGM data, the fiducial points from QRS complexes in the recorded ECG and EGM data are detected and aligned, and an average heartbeat is generated whose window length is determined by either the minimum or the mean of the R-R interval. The windows are then aligned with the fiducial points. In this manner, a template of average beats is constructed and subtracted from the original EGM signals in AF, resulting in only atrial activity. See Figure 2.2 for an example of the AA estimate (in green) obtained using ABS. It

should be noted that if the QRS complexes are not well aligned in time to each other, the resulting atrial signal usually contains residual ventricular activity. Temporal alignment is therefore crucial to any method that involves beat subtraction.
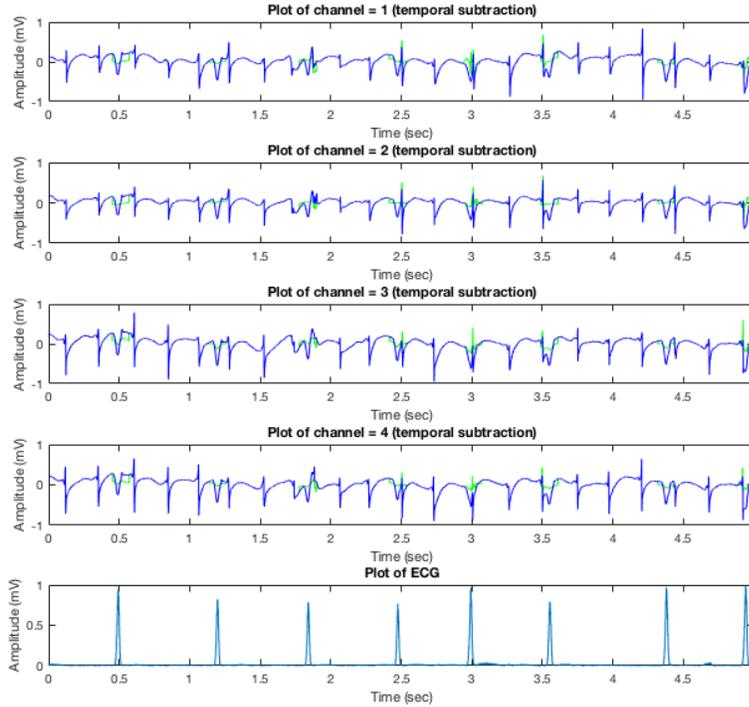
Figure 2.2: ABS for removal of VA in electrodes 1 to 4 of row 3 AF EGMs at recording location RA4.

Since ABS relies on the assumption that an average beat accurately represents an individual beat on a single lead/electrode, even minor changes due to variations in the heart's electrical axis in ECGs (primarily caused by respiratory activity) affect the QRS morphology leading to large QRS-related residuals [14]. Considering the variability of EGM signal recordings, it is easy to see that this is a formidable hindrance to effective VA reduction.

Finally, the most recent ABS method as of writing this thesis is based on adaptive singular value cancellation (ASVC) of VA [21]. The ASVC method detects all the R waves using the Pan and Tompkins method. The start and end points of each QRS complex are then detected and the complexes are subsequently aligned using

15

the R-peaks. Once all the beats are temporally aligned, their eigenvector sequence is obtained by singular value decomposition (SVD) such that the largest variance resulting from the SVD is assumed to uniquely represent VA, and therefore used as the primary cancellation template. The template is then adapted to individual QRS widths and heights, and is temporally aligned with each R peak in the EGM, before finally subtracting the customized template for each beat from every QRS complex. This SVD-based method generally provides a more accurate representation of VA morphed to each individual beat and consequently, a higher quality AA extraction [21], [14].

Whichever way, with subtraction techniques, there is the inevitable loss of atrial information since AA that coincides with VA is deleted from the EGM signal. The standard approach for coping with this loss of data is interpolation of the atrial signal in the QRS segments from the SQ intervals that enclose the QRS complex via sinusoidal interpolation [12]. The concatenated signal is then low-pass filtered to deal with any jumps that may occur at interval boundaries.

2.3.2 Adaptive Filtering

Due to the aforementioned shortcomings of ABS (see section 2.3.1), adaptive filtering has been proposed as a viable atrial activity estimation technique. Adaptive ventricular cancellation (AVC) is based on an adaptive filter that operates on the reference channel/electrode to produce an estimate of the interference, which is then subtracted or filtered from the channel under consideration. The channel under consideration contains both atrial and ventricular components, while the reference channel ideally contains pure VA that is precisely time-aligned with the QRS complex of the EGM. Again, since time alignment is an issue, if QRS complexes in the reference channel and the channel under consideration are not well aligned to each other, we end up with residual ventricular activity that makes further signal processing a challenge [12]. Additionally, we have to somehow obtain a reference signal, which is not always readily available.

See Figure 2.3 for a block diagram illustration of adaptive filtering of ECG signals. Naturally, the block diagram can be updated with more sophisticated techniques like LMS estimation based on a more thorough understanding of either ECG or EGM signals. Nevertheless, Figure 2.3 suffices to give a sense of what adaptive filtering entails. **Fs** indicates the signal sampling rate.
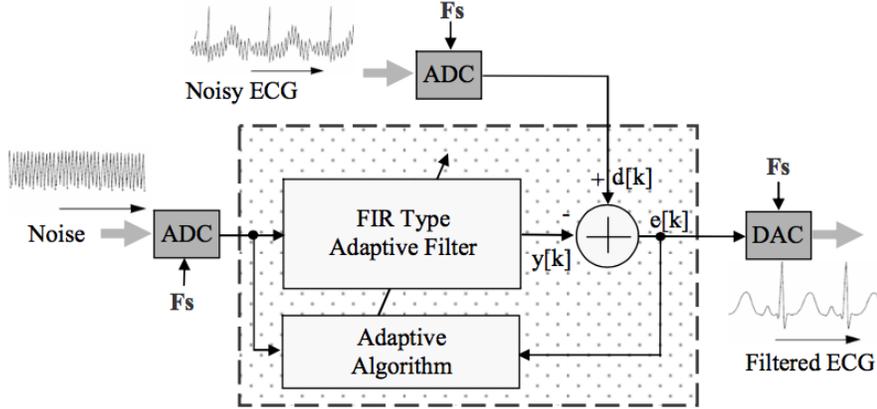
Figure 2.3: Standard adaptive filtering method for ECG signals (also applies to EGM; just replace ECG with EGM and noise with a VA estimate)[5].

2.3.3   Principal Component Analysis

Principal Component Analysis (PCA) performs an orthogonal linear transformation of the data such that it is projected onto a set of 'principal components' in the directions of maximum variance and minimal redundancy [12], [22]. The principal components are obtained as a linear combination of latent variables in the dataset, with weights chosen so that the principal components become mutually uncorrelated.

Let the signal segment of a beat (in sinus rhythm or otherwise) be represented by the column vector:

$$\mathbf{x} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}, \tag{2.3}$$

where N can either be the number of samples in a heart beat segment, or the entire length of the EGM. The segment is extracted from several successive beats or various EGM recordings, thus resulting in an $N \times M$ data matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1} & \mathbf{x_2} & ... & \mathbf{x_M} \end{bmatrix}. \tag{2.4}$$

The vectors $\mathbf{x_1}, ..., \mathbf{x_M}$ can be viewed as M observations of the random process $\mathbf{x}$. The derivation of principal components is based on the assumption that the signal $\mathbf{x}$ is a zero-mean random process characterized by the correlation $\mathbf{R_x} = \mathbf{E}[\mathbf{x}\mathbf{x^T}]$. The principal components of $\mathbf{x}$ result from applying an orthonormal linear transformation $\mathbf{\Psi} = [\psi_1, \psi_2, ..., \psi_N]$ to $\mathbf{x}$:

$$\mathbf{w} = \mathbf{\Psi^T}\mathbf{x}, \tag{2.5}$$

such that the elements of the principal component vector $\mathbf{w} = [w_1, w_2, ..., w_N]^T$ become mutually uncorrelated.

Since $\mathbf{R_x}$ is rarely known in practice, the $N \times N$ sample correlation matrix defined by:

$$\hat{\mathbf{R}}_\mathbf{x} = \frac{1}{M}\mathbf{X}\mathbf{X^T}, \tag{2.6}$$

replaces $\mathbf{R_x}$ when calculating the eigenvectors. PCA then amounts to computing the SVD of $\hat{\mathbf{R}}_\mathbf{x}$:

$$\hat{\mathbf{R}}_\mathbf{x} = \mathbf{A^{(1)}} \cdot diag(\sigma_1, ..., \sigma_R) \cdot \mathbf{A^{(2)T}} = \sum_{r=1}^{R} \sigma_r \mathbf{a_r^{(1)}}\mathbf{a_r^{(2)T}}, \tag{2.7}$$

where $\mathbf{A^{(1)}} \in \mathbb{R}^{I_1 \times R}$ and $\mathbf{A^{(2)}} \in \mathbb{R}^{I_2 \times R}$ are columnwise orthonormal and the singular values σ_r are positive, $1 \leq r \leq R$. If the singular values are distinct, then the dyads $\sigma_r \mathbf{a_r^{(1)}}\mathbf{a_r^{(2)T}}$ are unique for $1 \leq r \leq R$. The uniqueness of SVD comes from the orthogonality constraints on $\mathbf{A^{(1)}}$ and $\mathbf{A^{(2)}}$ [23].

In summary, the PCA algorithm involves computation of the covariance matrix from the ensemble of EGMs, eigenvalue and eigenvector decomposition of the covariance matrix, sorting eigenvectors in the descending order of eigenvalues, and finally projecting the original EGM data in the directions of sorted eigenvectors. The first few components represent most of the variability present in the data.

Since in AF the AA and VA overlap in time and frequency, this already invalidates a PCA decomposition either only in the time domain or in the frequency domain, and encourages the use of time-frequency decomposition techniques such as the short-time Fourier transform (STFT) and the wavelet transform. Additionally, since the PCs are only statistically uncorrelated, they are in general not directly related to actual independent physical or physiological sources [6], and PCA cannot irrevocably

recover independent source signals [6]. Furthermore, there is no reason to assume that bioelectrical sources of the heart are spatially orthogonal to one another unless the measurement electrodes are intentionally placed in an orthogonal configuration.

2.3.4 Independent Component Analysis

Independent Component Analysis (ICA) is a FA-BSS technique that attempts to extract mutually independent sources or latent components from a set of random variables, measurements or signals by linearly transforming the matrix of recordings $\mathbf{X}$ such that the source estimates are statistically independent [6]. The source components are assumed to be linearly or nonlinearly mixed together, and both the factor/mixing matrix $\mathbf{A}$ and loading/source components in matrix $\mathbf{S}$ are assumed unknown. Figure 2.4 below shows the ICA mixing and separation model.
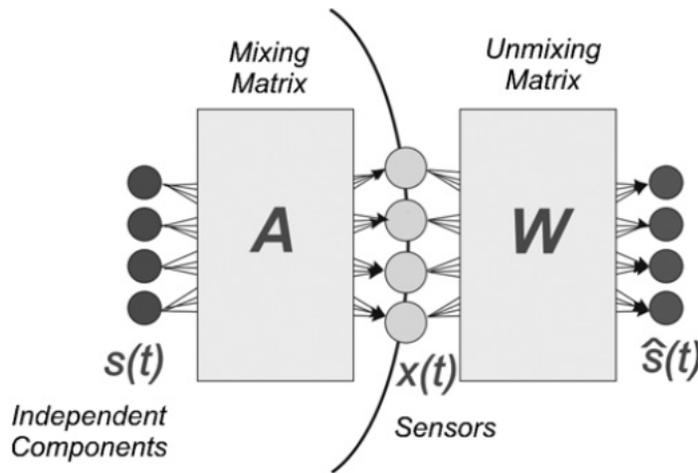
Figure 2.4: ICA mixing and separation model[6].

In contrast to PCA, $\mathbf{A}$ is not constrained to be orthogonal, but can take on any structure as long as $\mathbf{S}$ has full column rank [12]. The general ICA principles as defined by Hyvaerinen et al in [24] are:

- Nonlinear decorrelation: components are independent if both they and their appropriately chosen nonlinear transformations are uncorrelated

- Maximization of component nongaussianity: since the central limit theorem implies that summing nongaussian random signals generates more gaussian-like

signals, decomposing the sums of such signals by maximizing the nongaussianity of the original signals results in independent components (ICs)

ICA has been implemented through numerous approaches such as nongaussianity maximization, maximum likelihood estimation, nonlinear decorrelation, tensorial methods and nonlinear PCA [6]. Although second-order characteristics can be exploited to extract ICs since statistical independence implies statistical uncorrelation, most ICA methods take advantage of higher-order statistics to achieve better performance. Therefore, since Gaussian signals are only defined up to second-order statistics (mean and variance), it is impossible to separate more than one Gaussian signal from a mixture of signals using higher-order statistics ICA. Additionally, ICA typically finds at maximum as many ICs as there are ICA input signals, although workarounds to extract more ICs than input signals exist in literature [6].

Since biomedical signals are stochastic random signals by nature, fulfillment of the ICA assumptions, especially regarding nongaussianity, cannot be guaranteed [6]. However, even if ICA may fail with a specific set of input signals and using one algorithm with certain parameters, this does not necessarily imply that the dataset was unfit for ICA. Convergence can still be achieved by either changing the number of ICA input samples, i.e., the length of the EGM signal segment, and/or by changing the number of EGM electrodes, input signal bandwidth and sampling rate, or ICA parameters. Naturally, different ICA algorithms may yield different performances [6].

2.4   Conclusion

Standard AA and VA estimation techniques have been presented and briefly discussed in this chapter. The required time-alignment and inherent variability of AF EGM signals already implies that the performance of ABS and adaptive filtering is likely to fall short of our expectations. Furthermore, we really do not take full advantage of the spatial diversity of our AF EGM datasets with these two methods.

Matrix decompositions provide a more advanced alternative, but the imposition of constraints that are not necessarily valid may cause the algorithm to underperform. While incorporation of prior information via constraints is usually beneficial e.g., if the sources are uncorrelated then $\mathbf{S}$ is orthogonal, the data model becomes invalid when the priors do not correspond to properties of the available datasets leading to subpar performance.

Multiway component analysis in chapter three goes a step further to allow for a tensor

representation of the AF EGM datasets, whose decomposition based on multilinear algebra is unique under milder conditions, and only requires that the columns of the factor matrices be linearly independent.

Multiway Component Analysis **3**

Two-way (matrix) component analysis is a well established FA-BSS technique whose umbrella covers the aforementioned principal component analysis (PCA) and independent component analysis (ICA). However, given the ubiquity of modern homogeneous and/or heterogeneous sensor modalities, a lot of the data collected nowadays is of a multiway character that is better represented by multiway arrays (tensors). Tensor decompositions therefore allow the extension of matrix component analysis (two-way CA) to multiway component analysis (MWCA)

Datasets from a variety of fields such as audio and speech processing, image and video processing, biomedical engineering etc., can naturally be represented as high dimensional arrays, i.e., tensors. For example, in biomedical applications, an ECG/EGM recording can be formed from a $channel(electrodes) \times time$ matrix, and multiple disparate recordings (trials) form a third-order tensor of $channel \times time \times trial$. If we consider the time-frequency representation of each channel, we obtain a fourth-order tensor of $channel \times time \times frequency/scale \times trial$. See Figure 3.1 for the time-frequency representation of a multichannel signal recording as a three-way tensor. Via multiway FA-BSS we can extract spectral components, temporal components, and spatial components of ECG/EGM data simultaneously, together with the links between them that are represented by a core tensor [9].

While it is of course possible to unfold multiway data to form a large matrix and apply two-way FA-BSS methods, this often results in a loss of internal structure, and the flattened view along with the rigid assumptions inherent in two-way analysis are at times not suitable for multiway data [9], [8]. Through tensor decompositions, sophisticated models that capture multiple interactions and couplings can be developed and implemented, instead of standard dyadic interactions. To put it differently, we can only discover hidden components within multiway data if the analysis tools account for the intrinsic multidimensional patterns present [8].

Furthermore, tensor decompositions are not just matrix factorizations with additional subscripts, but rather are more structurally rich than linear algebra. For instance, basic notions such as rank have a more subtle meaning in multilinear algebra, and the uniqueness conditions for higher-order tensor decompositions are more
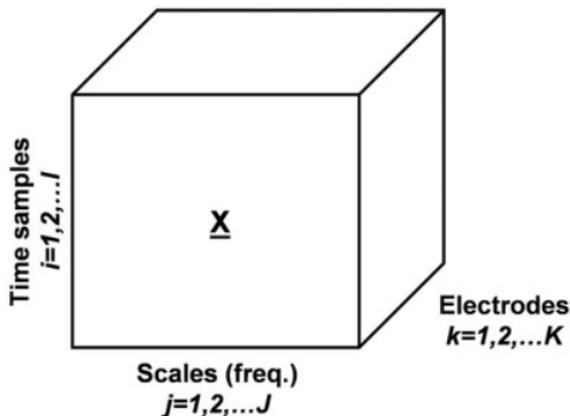
Figure 3.1: Time-frequency representation of a multichannel signal recording **X** as a tensor $\mathcal{X}$ [7].

natural and relaxed than those for matrices [25], [26], [8]. To quote the paper "Tensor Decompositions for Signal Processing Applications" [8]: "data analysis techniques using tensor decompositions are shown to have great flexibility in the choice of constraints which match data properties and extract more general latent components in the data than matrix-based methods."

As a natural extension of FA-BSS, multiway FA-BSS is an attractive and promising approach because it allows components to be simultaneously extracted from different domains (modes) of tensor data. Besides the well-known FA-BSS indeterminacies (arbitrary scaling and permutation of the rank-1 terms), it is quite often the case that the physical meaning of the factors is not directly apparent. If the model in Equation 2.1 (see the section "Factor Analysis and Blind Source Separation") is unconstrained, it admits infinitely many combinations of **A** and **S**. Standard matrix factorizations in linear algebra such as QR-factorization, eigenvalue decomposition (EVD) and SVD obtain their uniqueness from the restrictive constraints imposed such as triangularity and orthogonality. Additionally, other constraints like statistical independence, sparsity, nonnegativity and uncorrelatedness can permit the unique estimation of **A** and **S** [8].

In anticipation of any confusion as to the formal definition of tensors, we here define a tensor as a multiway array of numbers that are multilinear mappings over a set of vector spaces [27]. While it is possible to make sense of tensor decompositions from a linear algebra perspective, certain subtleties unique to higher-order arrays necessitate mathematical definitions specific to higher-order tensors. The interested

reader is referred to the appendix under the section "Mathematics of Tensor Decompositions". For the purposes of understanding the factorization models discussed in this document, the few but important mathematical definitions will be explained in the appropriate sections.

3.1 TUCKER Decomposition

Tucker decomposition, aka higher-order singular value decomposition (HOSVD), is a form of higher-order PCA. It was first introduced by Tucker in 1963 and has been referred to by a variety of names since then; more commonly N-mode PCA and HOSVD [3]. The Tucker model decomposes a tensor into a core tensor multiplied by a matrix along each mode. Thus, in the three-way case where $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we have [3]:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}(\mathbf{a_p} \circ \mathbf{b_q} \circ \mathbf{c_r}), \tag{3.1}$$

where $\mathbf{A} \in \mathbb{R}^{I \times P}, \mathbf{B} \in \mathbb{R}^{J \times Q}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$ are the factor matrices (usually orthogonal) and can be thought of as the principal components in each mode. The tensor $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ is called the core tensor and its entries show the level of interaction between the different components. Figure 3.2 shows the Tucker decomposition of a three-way tensor.

Elementwise, the Tucker decomposition in 3.1 is written as:

$$x_{ijk} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}(a_{ip} \circ b_{jq} \circ c_{kr}), \tag{3.2}$$

for $i = 1, ..., I, j = 1, ..., J$ and $k = 1, ..., K$. Here P, Q and R are the number of components (i.e., columns) in the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ respectively. If P, Q, R are smaller than I, J, K, the core tensor $\mathcal{G}$ is a compressed version of $\mathcal{X}$.

Additional constraints on the factor matrices and/or core tensor are necessary to obtain a unique and physically meaningful decomposition. For example, in order to get orthogonal factor matrices in Equation 3.1, truncated SVD is often applied to mode-n matricization $\mathbf{X_{(n)}}$ for $n = 1, 2, ..., N$ i.e., the orthonormal bases are obtained via the SVD of the mode-n matricized tensor $\mathbf{X_{(n)}} = \mathbf{U_n} \sum_{\mathbf{n}} \mathbf{V_n^T}$. As in the matrix
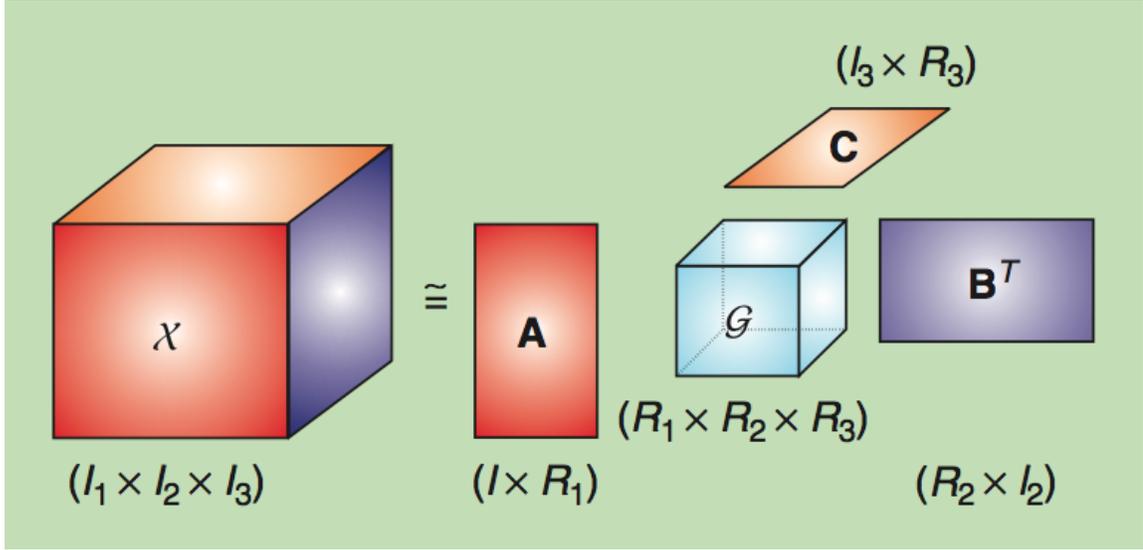
Figure 3.2: Tucker decomposition of a three-way tensor [8].

case, the multilinear singular values govern the multilinear rank, while the multilinear singular vectors allow, for each mode separately, an interpretation as in PCA making the Tucker decomposition a useful tool for compression and signal enhancement in multiway datasets [8]. This brief description explains why the Tucker decomposition is sometimes referred to as higher-order SVD.

Since the TUCKER decomposition is explained primarily for erudition purposes as a gentle introduction to canonical polyadic decompositions, the details of how it is computed in practice are not elaborated in this document and the interested reader is encouraged to consult one of the many resources mentioned in the references for information thereon.

3.2   Canonical Polyadic Decomposition

A polyadic decomposition (PD) represents an N-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ as a linear combination of rank-1 tensors in the form [8]:

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \mathbf{a_r^{(1)}} \circ \mathbf{a_r^{(2)}} \circ \ldots \circ \mathbf{a_r^{(N)}}. \tag{3.3}$$

26

Equivalently, $\mathcal{X}$ can be expressed as a multilinear product with a diagonal core:

$$\mathcal{X} = \mathcal{D} \times_{\mathbf{1}} \mathbf{A}^{(\mathbf{1})} \times_{\mathbf{2}} \mathbf{A}^{(\mathbf{2})} \ldots \times_{\mathbf{N}} \mathbf{A}^{(\mathbf{N})} \tag{3.4}$$

$$= [\![\mathcal{D}; \times_{\mathbf{1}} \mathbf{A}^{(\mathbf{1})}, \mathbf{A}^{(\mathbf{2})}, \ldots \mathbf{A}^{(\mathbf{N})}]\!], \tag{3.5}$$

where $\mathcal{D} = diag_N(\lambda_1, \lambda_2, ..., \lambda_R)$. The tensor rank is defined as the smallest value of R for which Equation 3.3 holds exactly; the minimum rank PD is called canonical PD (CPD) and is desired in signal separation. In fact, Canonical Polyadic Decompositions (CPDs) can be viewed as a special case of TUCKER decompositions where the core tensor is superdiagonal and $P = Q = R$ i.e., we restrict the core tensor $\mathcal{G}$ to be diagonal with $R_1 = R_2 = \ldots = R_N = R$ [3]. The CPD of a two-way tensor (matrix) $\mathbf{X}$ and a three-way tensor $\mathcal{X}$ is shown in Figure 3.3.
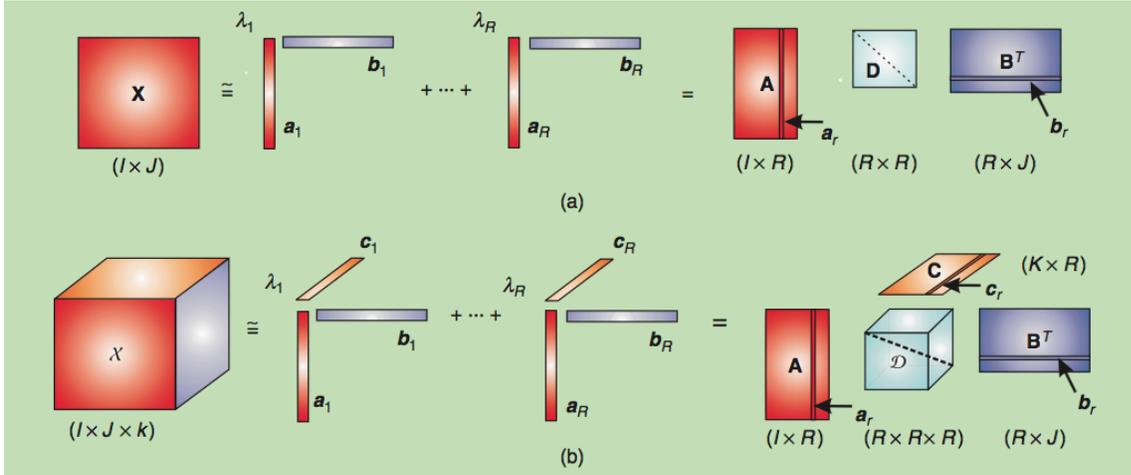
Figure 3.3: Canonical Polyadic Decomposition (CPD) of (a) a two-way tensor (matrix) (b) a three-way tensor [8].

Unlike the decomposition of a matrix in rank-1 terms where uniqueness is obtained by imposing orthonormality, the CPD of a tensor is unique up to scaling/counterscaling and permutation (FA-BSS indeterminacies) under mild conditions. Interestingly, the CPD can even be unique for values of R that are greater than the tensor modes [23]. From a more theoretical perspective, the following theorem defines the uniqueness of CPDs:

Theorem 1: Consider a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ that admits the CPD where $N = 3$. If (i) the columns of $\mathbf{A}^{(1)}$ are linearly independent, (ii) the columns of $\mathbf{A}^{(2)}$ are linearly

independent, and (iii) $\mathbf{A^{(3)}}$ does not have proportional columns, then the CPD is unique modulo permutation of the rank-1 terms and modulo scaling/counterscaling of factors within the same term [23].

Indeed, the uniqueness of a CPD is based entirely on multilinear algebra and has nothing to do with the physical attributes of the factors. Under the conditions defined in Theorem 1, the CPD can be computed via a generalized Eigenvalue Decomposition (EVD) [23].

Before moving on to the computation of CPDs in practice, it is useful to address the meaning of tensor rank in the context of CPD. The rank of a tensor $\mathcal{X}$, denoted $rank(\mathcal{X})$, is defined as the smallest number of rank-1 tensors that generate $\mathcal{X}$ as their sum i.e., the smallest number of components in an exact CP decomposition. While the definition of tensor rank is analogous to matrix rank, their properties are quite different. Furthermore, finding the rank of a given tensor is NP-hard and there is no straightforward algorithm to calculate it [28], [3]. Numerically, the rank of a tensor is determined by fitting various rank-R CPD models. Additionally, while for matrices, the maximum rank (defined as the largest attainable rank) and the typical rank (defined as any rank that occurs with probability greater than zero) are the same, for tensors the two may be different.

3.2.1 Computation of CPDs

Needless to say, there have been a lot of significant developments in the computation of CPDs with the goal of decreasing the computation time and storage space. In this regard, numerous linear and nonlinear algorithms exist, but in this report we focus on the alternating least squares (ALS) approach since the computation of the CPD is intrinsically a multilinear problem that can be solved by optimizing a sequence of linear subproblems [8]. Other more sophisticated algorithms will be briefly mentioned for expository purposes and more information thereon can easily be found in the literature.

When finding latent variables or source components, the CPD only permits an approximation of the data due to either the presence of noise or the inexactness of the model [23]. It is therefore standard to fit the decomposition in a least squares sense i.e., by minimizing the Frobenius norm of the difference between the data tensor and its CP approximation (if the noise is Laplacian, we fit the least absolute error). The working horse of CPD is therefore the Alternating Least Squares algorithm (ALS).

We now describe the ALS algorithm for a third-order tensor, but this can easily be

generalized to higher-order tensors. We want to find the CPD with R components for the tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ that best approximates it i.e.:

$$\min_{\hat{\mathcal{X}}} \quad ||\mathcal{X} - \hat{\mathcal{X}}|| \quad \text{with} \quad \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a_r} \circ \mathbf{b_r} \circ \mathbf{c_r} = [\![\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \qquad (3.6)$$

The ALS method works as follows: fix $\mathbf{B}$ and $\mathbf{C}$ to solve for $\mathbf{A}$, then fix $\mathbf{A}$ and $\mathbf{C}$ to solve for $\mathbf{B}$, then fix $\mathbf{A}$ and $\mathbf{B}$ to solve for $\mathbf{C}$; continue repeating the entire procedure until some convergence or stopping criterion is satisfied. Common convergence or stopping criteria are little or no improvement in the objective function, little or no change in the factor matrices, the objective value is at or near zero, and exceeding a predefined maximum number of iterations [3].

ALS is pretty straightfoward and often works fairly well [3], but it can also get very slow especially for ill-conditioned problems. Additionally, like most alternating algorithms, it is not guaranteed to converge to a stationary point and/or find a global minimum, and the update procedure breaks symmetry if the tensor to be factorized is symmetrical [23]. To rectify the first problem (nonconvergence), we can opt to only update the factor matrix whose cost function has decreased the most at any given step, but this comes with the penalty of an N-times increase in computational cost per iteration. For the second problem (breaking symmetry), we can do ALS with a line search after each major iteration such that all the component matrices are simultaneously updated based on the standard ALS search directions [3]. The line search scheme works because of the multilinearity of the CPD in that we can compute the optimal step along a line by polynomial rooting. Other optimization tricks to improve convergence and/or retain symmetry such as Tikhonov regularization are also possible [3].

Lastly, the question of rank i.e., the number of R components that the data should be factored into is of concern, since as already mentioned, there is no finite algorithm to determine the rank of a tensor [3]. The standard approach is to try different R until a "good" fit is found. Given the ambiguousness of the word "good", the fact that some tensors may have approximations of lower rank that are arbitrarily close in terms of fit, and that the data is often noisy invalidating a best fit approach to find the rank, researchers have come up with techniques to compare different fits (what value of R to use). The most common method is a consistency diagnostic called CORCONDIA, where, since we have already established that CPD is a special case of the Tucker Decomposition with a superdiagonal core tensor, the CORCONDIA carries out a modified Tucker Decomposition (the factor matrices are not orthogonal) and tries to make the core tensor maximally diagonal. Significant deviations from

a superdiagonal core or nonconvergence due to too many iterations either indicate that the decomposition rank is not appropriate, that the dataset it too noisy, or that the CPD model does not apply for the given dataset. In some cases applying proper constraints on the CPD such as nonnegativity, orthogonality or bounding the condition number of the factor matrices can improve the fit of the CPD. However, if meaningful constraints cannot be imposed and CORCONDIA still fails, then the CPD is not the right model for the data and an alternative decomposition should be considered [23].

Indeed performing the CORCONDIA (also known as rank test) on the time-frequency representation of EGM data in SR and AF is part of the motivation to consider Block Term Decomposition, where we relax the rank-1 constraint in CPD. This is because although the rank test works fine for a time-frequency representation of the recordings in SR, it fails for AF recordings implying that the CPD model is only valid for SR and does not hold for AF.

3.3 Block Term Decomposition

Block Term Decomposition (BTD), otherwise referred to as Block Component Analysis (BCA) is a generalization of Canonical Polyadic Decomposition (CPD), where instead of decomposing a matrix or tensor into rank-1 terms, we relax the rank-1 constraint and decompose the dataset into $rank - (L_r, L_r, 1)$ terms [29]. This is because the rank-1 constraint is a relatively strong assumption on the components that are obtained, and one may wonder if this is indeed satisfied for different kinds of datasets. In fact, it is likely possible (and often is), that a low multilinear rank is a better representation of reality [29], which opens up a whole new set of possibilities for multiway FA-BSS. It has been shown in [30] and [31] that BCA is related to Sparse Component Analysis (SCA) and Compressive Sensing (CS) respectively. While in SCA the factors are low-dimensional in that they are often zero, in BCA the factors have a low intrinsic dimension that is characterized by multilinear rank. In compressive sensing, low intrinsic dimensionality is used for compact signal representation while in BCA this trait serves as the basis for signal separation.

The BTD of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is a sum of rank $(L_r, L_r, 1)$ terms for $1 \leq r \leq R$, and is defined as:

$$\mathcal{X} = \sum_{r=1}^{R} (\mathbf{A_r} \cdot \mathbf{B_r^T}) \otimes \mathbf{c_r}, \tag{3.7}$$

where each of the matrices $\mathbf{A_r} \in \mathbb{R}^{I_1 \times L_r}$ and $\mathbf{B_r} \in \mathbb{R}^{I_1 \times L_r}$ has linearly independent columns and the vectors $\mathbf{c_r} \in \mathbb{R}^{I_3}$ are nonzero. We assume that R is minimal [29]. Figure 3.4 shows both the $(L_r, L_r, 1)$ and the generalized BTD of a three-way tensor.
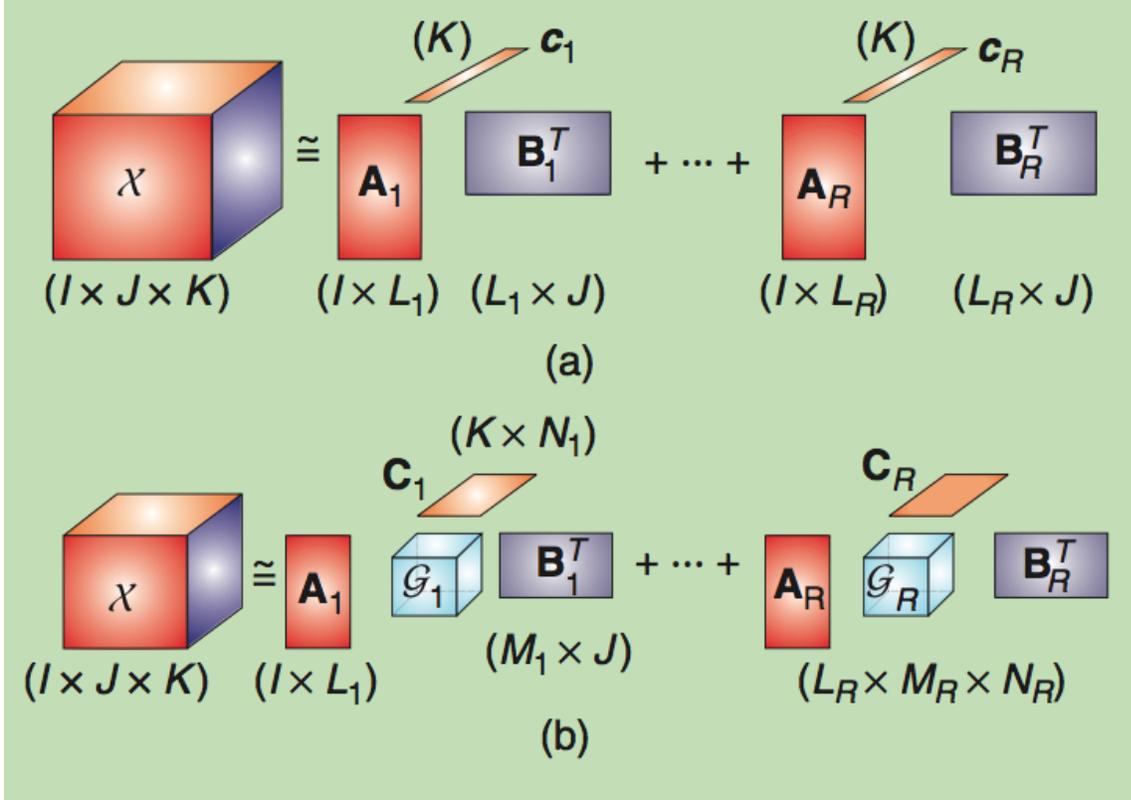
Figure 3.4: (a) $(L_r, L_r, 1)$ BTD of a three-way tensor (b) Generalized BTD of a three-way tensor [8].

The BTD is also unique up to FA-BSS indeterminacies i.e., post-multiplication of $\mathbf{A_r}$ by a square nonsingular matrix $\mathbf{W_r}$ and premultiplication of $\mathbf{B_r^T}$ by $\mathbf{W_r^{-1}}, (1 \leq r \leq R)$ also leads to a valid decomposition of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$.

It can be claimed that the success of CPD is probably due to the fact that the rank-1 terms capture the essence of components that are actually a lot more complex. However, in more complicated situations where the rank-1 constraint no longer suffices, it is logical to check if BCA provides more information. To put it differently, a rank-1 term can be called an atom since it is a constituent element that cannot be split into smaller parts. CPD splits a data tensor into atoms. For three-way

($temporal \times spectral \times spatial$) data, CPD results in latent components consisting of rank-1 temporal by rank-1 spectral by rank-1 spatial weighting factors. However, for biomedical signals or speech signals whose sources have a particular spectral content, the latent components correspond to molecules rather than atoms. BCA can thus be viewed as separation on the level of molecules, and provides a better signal model. Thus, for $temporal \times spectral \times spatial$ data, BTD results in components consisting of rank-L_r temporal by rank-L_r spectral by rank-1 spatial weighting factors, depending on how you order the modes of the tensor i.e., the components can also be represented by rank-L_r temporal by rank-L_r spatial by rank-1 spectral weighting factors [29].

3.3.1 Computation of BTDs

Similar to CPD, the block term decomposition in $rank - (L_r, L_r, 1)$ terms is trilinear in the component matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. Consequently, the solution to BTD can also be obtained via an alternating least squares (ALS) approach where we solve multiple linear least squares subproblems. The update rules and associated challenges with the ALS method for BTD are similar to those already described in the section on computing the CPD. Refer to the section "Computation of the CPD" for further details on ALS.

Using the BTD model defined in Equation 3.7, let $\mathbf{A} = [\mathbf{A_1}, ..., \mathbf{A_R}], \mathbf{B} = [\mathbf{B_1}, ..., \mathbf{B_R}]$ and $\mathbf{C} = [\mathbf{c_1}, ..., \mathbf{c_R}]$. We then rewrite the tensor $\mathcal{X}$ in its matrix representations as follows:

$$\mathbf{X_{IJ \times K}} = [\mathbf{A_1} \odot_{\mathbf{c}} \mathbf{B_1} \mathbf{1_{L_1}} \ldots (\mathbf{A_R} \odot_{\mathbf{c}} \mathbf{B_R}) \mathbf{1_{L_R}}] \cdot \mathbf{C^T}, \tag{3.8}$$

$$\mathbf{X_{JK \times I}} = (\mathbf{B} \odot \mathbf{C}) \cdot \mathbf{A^T}, \tag{3.9}$$

$$\mathbf{X_{KI \times J}} = (\mathbf{C} \odot \mathbf{A}) \cdot \mathbf{B^T}, \tag{3.10}$$

where $\odot$ denotes the Khatri-Rao product as defined in Table 2.1 and $\odot_{\mathbf{c}}$ denotes the columnwise Khatri-Rao product.

The ALS algorithm for the decomposition in $rank - (L_r, L_r, 1)$ terms as described in [32] is shown below.

procedure $BTD\_ALS(\mathcal{X}, L_r, L_r, 1)$
    Initialize $\mathbf{B}, \mathbf{C}$
    Iterate until convergence:
        Update $\mathbf{A}$:
            $\mathbf{A} \leftarrow \left[(\mathbf{B} \odot \mathbf{C})^{\dagger} \cdot \mathbf{X_{JK \times I}} \right]^{T}$
        Update $\mathbf{B}$:
            $\tilde{\mathbf{B}} = \left[(\mathbf{C} \odot \mathbf{A})^{\dagger} \cdot \mathbf{X_{KI \times J}} \right]^{T}$
        For $r = 1, ..., R$:
            QR-factorization: $\tilde{\mathbf{B}}_{\mathbf{r}} = \mathbf{QR}, \mathbf{B_r} \leftarrow \mathbf{Q}$
        Update $\mathbf{C}$:
            $\tilde{\mathbf{C}} = \{ [\mathbf{A_1} \odot_{\mathbf{c}} \mathbf{B_1} \mathbf{1_{L_1}} \dots (\mathbf{A_R} \odot_{\mathbf{c}} \mathbf{B_R}) \mathbf{1_{L_R}}]^{\dagger} \cdot \mathbf{X_{IJ \times K}} \}^{\mathbf{T}}$
        For $r = 1, ..., R$:
            $\mathbf{c_r} = \tilde{\mathbf{c}}_{\mathbf{r}} / ||\tilde{\mathbf{c}}_{\mathbf{r}}||$
end procedure

3.4   Conclusion

This chapter has covered the specifics of multiway component analysis in depth first by introducing the TUCKER decomposition, then constraining the core of the TUCKER decomposition to be diagonal thus deriving the canonical polyadic decomposition (CPD), and finally relaxing the rank-1 constraint in CPD resulting in the block term decomposition (BTD).

In the results section, we apply the CPD and BTD discussed in this section to the AF EGM datasets with the goal of factorizing the time-frequency (TF) representations of AF EGM datasets such that the components obtained represent physiologically meaningful signals. It turns out that BTD is the preferred MWCA tool to get an estimate of the VA, enabling the realization of the research objective set out at the beginning of this project: reduction or complete removal of VA from AF EGMs via temporal subtraction and/or power subtraction of the VA estimate from the AF EGMs.

Experimental Methodology 4

The epicardial EGMs were graciously provided by the Translational Electrophysiology group within the department of Cardiology at Erasmus MC Rotterdam. All data analysis is carried out on a 13-inch Mid 2010 2.4 GHz Intel Core2 Duo MacBook Pro laptop using MATLAB, a proprietary programming language developed by MathWorks Inc. While there are a number of tensor analysis toolboxes in existence which in theory would return the same results irrespective of the toolbox used, we exclusively use the Tensorlab toolbox developed by N. Vervliet, O. Debals, L. Sorber, M. Van Barel and L. De Lathauwer within the Group Science, Engineering and Technology at KU Leuven, in part because of the elegance of the Structured Data Fusion framework (SDF) that abstracts the minutiae of algorithmic implementation, while allowing the user access to the algorithms should he/she desire to modify any for his/her purposes.

4.1 Data Acquisition

The epicardial EGM data was recorded using a rectangular electrode array (dimensions 4.6×1.4 cm) that is placed at different locations on the surface of the heart during open chest surgery. The electrode array consists of a rectangular electrode matrix (32 rows by 8 columns) whose wires are made of silver (diameter 0.3 mm) and the interelectrode spacing is 2 mm. Figure 4.1 (repeated here from the Introduction for the reader's convenience) shows the various positions at which the electrode array is placed to record EGM signals. The configurations are more or less the same for SR and AF EGMs, human error notwithstanding.

The recorded EGMs (256 channels per location) are then amplified (gain 1000), filtered using a 50 Hz fixed filter to remove power line interference and a bandpass filter (0.5 - 400 Hz) to remove baseline trend and high frequency noise. They are then multiplexed at a sampling rate of 1 kHz and A/D converted at a resolution of 8 bits. Signal measurement lasts 5 seconds in SR and 10 seconds in AF.
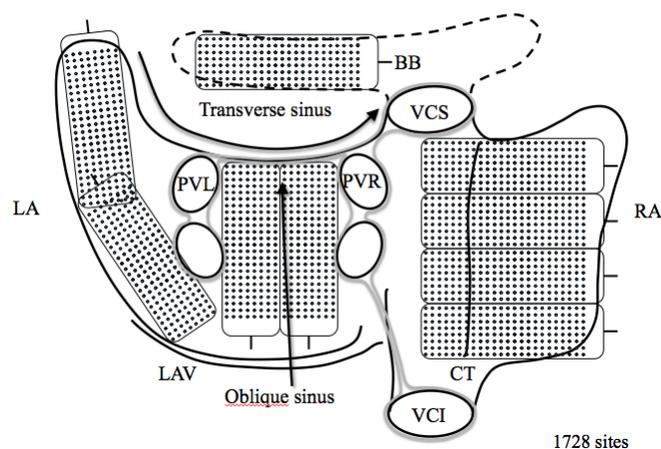
Figure 4.1: Image of the various EGM recording locations and the rectangular electrode array configuration.

4.2 Data Preprocessing

In light of the variability in amplitude and variance in the EGM recordings, some preprocessing of the dataset is necessary prior to the application of any algorithmic procedures. Fortunately, the data recordings are quite clean, and it is thus not necessary to further filter the signals to remove noise and/or other extra-cardiac contaminants. We therefore, on an electrode by electrode basis, simply remove the mean of the data and standardize it by scaling the amplitude (± 1) and variance (unit variance).

We also remove the reference electrode at channel 8 and the ECG recording at electrode 1, which is generally used to categorically identify the presence and location of VA since it is much easier to see in the ECG. As for the rest of the EGM data, we manually went over the dataset to identify faulty electrodes and removed them so as to only retain valid EGMs.

Due to the nonstationarity of the EGM signal and the convolutive rather than instantaneous mixing model assumed (due to the time delay between AA and VA and probably a more complex filter between the AA and VA generation sites and EGM recording sites), we utilize a time-frequency representation of the dataset. This is because for short enough window frames in the time domain, the EGM signal can be assumed stationary, and if the length of the short-time Fourier Transform (STFT) is much longer than the length of the framing window, convolutive mixing in the time domain becomes instantaneous mixing in the frequency domain[18].

For our particular application, we chose the simpler STFT to begin with, so as to demonstrate the applicability of a time-frequency analysis, before considering more sophisticated techniques like wavelet analysis. The drawback of this approach is that since the size of the window is fixed with the STFT, we still have to tradeoff resolution in the time domain for resolution in the frequency domain. While atrial and ventricular activity are temporally isolated in SR, they often overlap in AF, and so the bigger window (128 samples with an overlap of 50% i.e., 64 samples) when analyzing EGM signals in SR must be made smaller when analyzing EGM signals in AF (32 samples with an overlap of 16 samples). Moreover, while the spectral content (repetition rate) is the same for AA and VA in SR, it is different, although still overlapping in AF. In both cases the frames are windowed using a Hanning window to make the transition from frame to frame smoother and avoid ringing effects caused by the abruptness of a rectangular window. The inverse Fourier Transform (ISTFT) is always used to revert to the time domain post-analysis.

Furthermore, we only work with the magnitude (squared) of the data in the time-frequency representation (power spectral density), since at the moment it is not known how much information is contained in the phase of the signal, particularly during AF where the generation mechanism appears random. Thereafter, the time-frequency representation of each electrode in the same row is stacked to form a three-way tensor so as to exploit the spatial information as well. The three-mode tensor of the observed EGM signals is the input to the multiway algorithms discussed in this thesis.

Lastly, it is useful to know the rank of the tensor we are working with in order to get an accurate decomposition of the datasets. This is because even if it turns out that the dataset is better represented by lower-rank terms instead of rank-1 terms, the decomposition is valid for $1 \leq r \leq R$ where R is the rank of the tensor. Additionally, the number of components into which the data should be decomposed is always an input parameter, and generally hard to determine unless we conclusively know how many sources are present in the signal recordings. While we can safely assume two sources in SR (AA and VA), the complex generation mechanism of AA in AF (AA wave regeneration and AA wave splitting) results in what could be considered multiple sources, therefore making it impossible to state apriori the number of active sources present in an EGM at different locations. To make things even more complex, it could be possible that the number of sources recorded by one electrode is different from that recorded by another on the same electrode array at the same location for the duration of the EGM recording. As a simple solution to intelligently guess the rank of the tensor before applying any tensor decomposition to the EGM tensors, we determine the (low) rank of the tensor via a (low)rank test provided in the Tensorlab

toolbox.

It should be noted that although we work with the power spectral density of the datasets, the algorithms evaluated in this report can also utilize complex-valued data (magnitude and phase), as long as the appropriate constraints are applied (eg., nonnegativity is no longer a valid constraint with complex valued data).

4.3   Pseudoreal EGM Signals

It is of course imperative that we evaluate the performance of the selected algorithms using appropriate performance indices as defined in the following chapter on "Performance Metrics". However, for some of the metrics defined, it is necessary to have a pure atrial signal against which to compare the extracted atrial activity. Unfortunately, without a sophisticated biophysical model of the generation and propagation mechanism of atrial fibrillation, it is practically impossible to obtain a pure atrial and/or ventricular signal since the recorded signals are almost always contaminated by the far ventricular signal. Despite the absence of a pure atrial and/or ventricular signal, measuring and comparing the performance of any algorithm requires a number of quantitative indices against which we can claim that one algorithm is superior to another.

Therefore, as a workaround to this problem, we instead generated pseudoreal signals, following the technique developed for ECG's by Rieta et al. in the paper "Spatiotemporal Blind Source Separation Approach to Atrial Activity Estimation in Atrial Tachyarrhythmias" [10]. To synthesize pseuodreal signals, the VA is obtained from the SR recordings by segmenting out the VA at QRS peak locations. Since the duration of the VA is about 140 - 150 ms, we use a frame of 140 samples (sampling rate = 1 kHz) and window it with a Hamming window of the same length to avoid abrupt transitions between segments when we add the VA to the AA to generate pseudoreal AF.

For the atrial activity, we isolate the AA from T-Q intervals during AF episodes in the AF recordings and carefully extrapolate it between two adjacent T-Q segments to replace the regions of QRS activity that are segmented out of the AF EGMs (once again using the ECG signals to find the location of the QRS peaks). The extrapolation method is fairly simple, in that the AA prior to the QRS complex is replicated within the QRS interval, but linearly weighted such that the weights are one at the beginning of the interval and decrease down to zero at the end of the interval. Similarly, the AA following the QRS complex is replicated within the QRS interval and weighted from zero at the beginning of the interval rising up

to one at the end of the interval. Both segments are combined to build up the extrapolated AA wave(s) within the QRS interval. This is repeated for all electrodes in the AF EGMs. Although the reconstructed AA samples do not exactly correspond to the true AA signal masked by the QRS complex as seen in the deviations in Figure 4.2, which compares the real AF EGM signals to the generated pseudoreal AF signals, the model more or less preserves the all the key features of the recorded EGM signals. Table 4.1 summarizes the mean NMSE and standard deviation between the real EGMs and the pseudoreal EGMs, which indicates that the pseudoreal signals are a relatively good representation of the clinical EGM signals, while providing a synthesized version of pure AA and pure VA that is necessary for the evaluation of algorithmic performance.
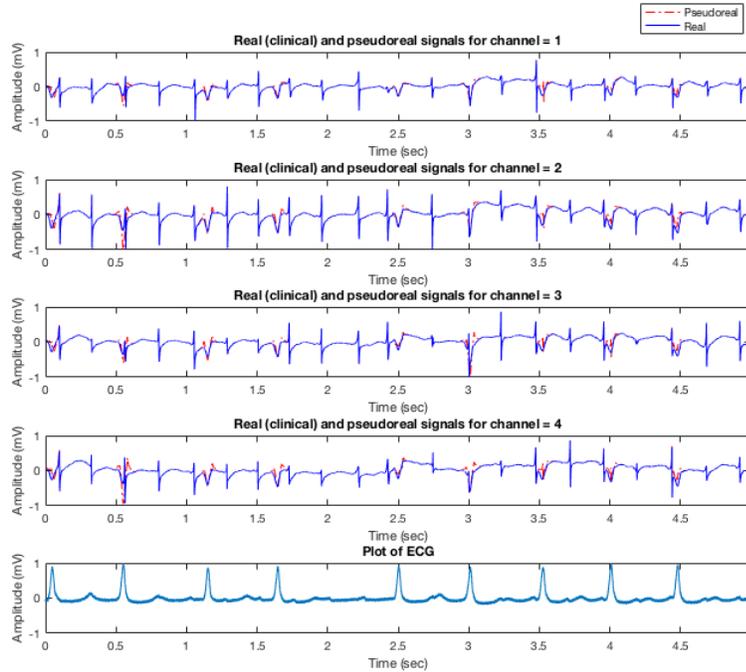
Figure 4.2: Pseudoreal and real EGM signals generated from real SR and AF EGMs of row 8 at location RA3.

In conclusion, the pseudoreal AF signals are generated by adding the extracted VA from SR recordings and the extracted AA from AF recordings. While not necessarily a realistic scenario with regard to actual AF generation and propagation mechanisms, it nevertheless gives us a way to objectively evaluate the performance of the selected
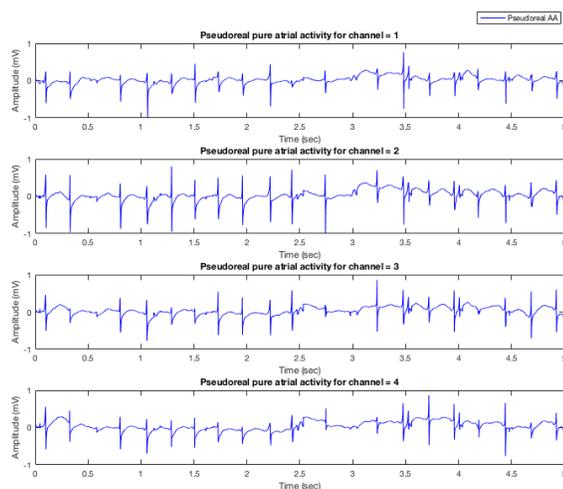
Figure 4.3: Pseudoreal AA signals generated from real AF EGMs of row 8 at location RA3.
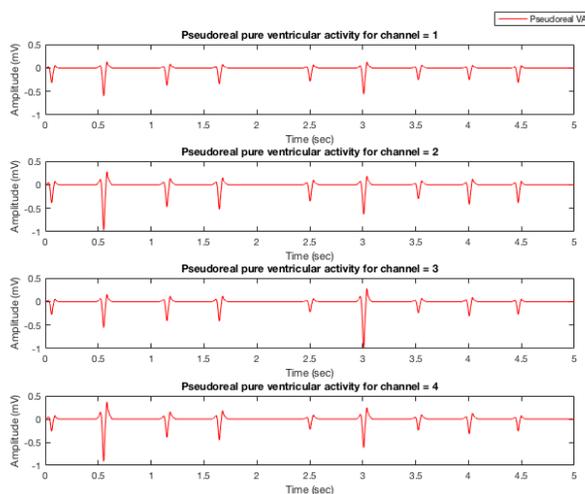
Figure 4.4: Pseudoreal VA signals generated from real SR EGMs of row 8 at location RA3.

algorithm in this report, whose results can of course also be confirmed by visual inspection of the estimated AA. See Figures 4.3, 4.4 and 4.2 for the estimates of the pure atrial activity, pure ventricular activity and the summation thereof, respectively, for electrodes 1 through 4 of row 3 in the EGMs of the electrode array placed at location RA1 of a patient in atrial fibrillation.

Table 4.1: Average and Standard Deviation of the Normalized Mean Square Error (NMSE) between generated pseudoreal and real clinical AF signals for the 10 different recording locations.

Recording Location	Average NMSE	Standard Deviation
BB0	0.1758	0.1300
LA2	1.0372	0.8509
LA3	0.6576	0.7932
PVL1	0.3333	0.1958
PVR1	0.1554	0.0614
RA1	0.0933	0.0504
RA2	0.1915	0.1013
RA3	0.1180	0.0465
RA4	0.1301	0.0599
RA5	0.1589	0.0696

4.4 Step-by-Step Algorithmic Procedure

1. Preprocessing: for each electrode, remove EGM signal mean and normalize to unit variance and unit amplitude.

2. If necessary, remove baseline wander (low-pass filter 0-1 Hz) and power-line interference (stopband filter at 50 Hz). Not necessary since our datasets are quite clean.

3. Tensorize the EGM data on a row-by-row basis by applying the STFT to each electrode recording to obtain a three-way tensor ($temporal \times frequency \times spatial$).

4. Apply the relevant FA-BSS algorithm i.e., CPD or BTD.

5. Find the ISTFT of the separated components.

6. Calculate signal kurtosis to identify the ventricular and non-ventricular source components.

7. Remove the VA by scaling the source component corresponding to VA accordingly (due to FA-BSS scaling ambiguity) and temporally or power spectrally subtract the VA estimate from the EGM data.

8. Measure the quality of ventricular activity reduction or removal using the defined performance metrics.

42

Performance Metrics

<div style="text-align: right; font-size: 3em;">5</div>

For the purposes of determining how well the ventricular artifacts have been either reduced or completely removed from epicardial EGM signals in atrial fibrillation, the following performance metrics have been selected: cross-correlation coefficient, dominant frequency, normalized mean square error (NMSE), spectral concentration and ventriculo-atrial sample ratio (VASR). They will each be briefly introduced in the following sections. Note that almost all of the performance indices can be computed on either the entire length of the signal recording or an a beat-by-beat basis. In general, the beat-by-beat approach, whenever applicable, provides more detailed information on ventricular activity (VA) reduction because it focuses on the beat segments where VA is typically expected. Nevertheless, to also account for the level of distortion that the algorithm may induce, performance indices that depend on the entire length of the EGM signals such as NMSE also prove to be of importance.

5.1   Cross-Correlation Coefficient

The cross-correlation coefficient ρ between two vectors $\mathbf{a}$ and $\mathbf{b}$ is a measure of the quality of a least squares fitting of the two vectors. It is also referred to as the normalized cross-covariance. Mathematically, this is expressed as:

$$\rho(\mathbf{a}, \mathbf{b}) = \frac{1}{N-1} \sum_{i=1}^{N} (\frac{a_i - \mu_a}{\sigma_a})(\frac{b_i - \mu_b}{\sigma_b}) \tag{5.1}$$

$$= \frac{\mathrm{cov}(\mathbf{a}, \mathbf{b})}{\sigma_a \sigma_b}, \tag{5.2}$$

where N is the signal length ($\mathbf{a}$ and $\mathbf{b}$ are of the same length), μ_a and σ_a are the mean and standard deviation of vector $\mathbf{a}$ respectively, and μ_b and σ_b are the mean and standard deviation of vector $\mathbf{b}$ respectively.

In our case, the two vectors $\mathbf{a}$ and $\mathbf{b}$ are the pure atrial signal and the extracted atrial signal respectively. The cross-correlation essentially measures the degree of

similarity between the two signals, which indicates how well the ventricular signal is removed from the EGMs, while preserving the atrial activity. Therefore, if the pure AA and extracted AA are perfectly correlated, the cross-correlation coefficient evaluates to 1 and if completely uncorrelated, it evaluates to 0.

5.2 Normalized Mean Square Error

The normalized mean square error (NMSE) is a measure of the difference between two vectors. Mathematically, for two vectors $\mathbf{a}$ and $\mathbf{b}$, it is defined as:

$$NMSE(\mathbf{a}, \mathbf{b}) = \frac{||\mathbf{a} - \mathbf{b}||^2}{||\mathbf{a} - \mu_{\mathbf{a}}||^2},$$
(5.3)

where μ_a is the mean of the vector $\mathbf{a}$. Once more, $\mathbf{a}$ and $\mathbf{b}$ represent the pure atrial signal and the extracted atrial signal respectively. Therefore, if the NMSE is low, then the two vectors have a lot in common, and if the NMSE is high, then there are a lot of deviations between the two vectors. A low NMSE is therefore preferred, but high NMSE values do not necessarily imply that the two vectors are quite different, but rather that there might be time shifts between the samples. Additionally, significant differences between peak values in the two vectors have a higher weight on the NMSE than other values [33]. This is why we also use the NMSE to measure the difference between the pseudoreal EGMs and the real EGMs.

5.3 Ventriculo-Atrial Sample Ratio

This performance metric is evaluated on a beat-by-beat basis. Based on the concept of signal-to-noise ratio, the ventriculo-atrial sample ratio (VASR) is defined as [34]:

$$VASR(dB) = 10log_{10}\left(\frac{E_{VA}}{E_{AA}}\right) = 10log_{10}\left(\frac{\sum_{i=1}^{M} s_{VA}^2(i)}{\sum_{i=1}^{M} s_{AA}^2(i)}\right),$$
(5.4)

where M is the length of the signal segment that would typically contain VA, s_{AA} is the segmented atrial signal and s_{VA} is the segmented original EGM recording. Essentially, we want to determine the ratio of the energy in the signal segments when contaminated with ventricular activity to that when VA is removed. Signal segments

44

with both atrial and ventricular activity are more energetic than those with only atrial activity. Consequently, a high VASR is undesirable. For the purpose of getting rid of ventricular activity, a VASR closer to zero is preferred. Negative values point to an error in the algorithm since that implies that there is more energy in the extracted atrial activity after VA reduction than there was prior to any processing.

5.4 Dominant Frequency

Estimating the dominant frequency (DF), although typically done to determine the cycle length of atrial waves, can also be used to measure the quality of ventricular reduction since the presence of ventricular activity has been shown to affect the dominant frequency of AA [1]. Consequently, if we calculate the dominant frequency of the original epicardial EGMs, that of the EGMs with reduced ventricular activity, and that of pure atrial activity, dominant frequencies close to that of pure atrial activity indicate a good reduction in ventricular activity.

The dominant frequency is defined as the largest spectral peak (maximum fundamental frequency) in the power spectrum (aka power spectral density) of a signal. To compute the power spectral density, we divide the signal into short overlapping segments that are appropriately windowed, calculate the FFT on the segments, square its magnitude and average the power spectra of the short segments (Welch's method). The length of the signal segments is important since it determines the estimation accuracy of the DF by restricting spectral resolution [14]. We choose the segment length to be about 1 second (1024 samples) with an overlap of 50% (512 samples) and an FFT length of 8192 samples.

5.5 Spectral Concentration

Based on the discussion in the paper "Spatiotemporal blind source separation approach to atrial activity estimation in atrial tachyarrhythmias" [10], where the spectral concentration of AF EGMs has been shown to be an indicator of effective reduction of ventricular activity, we define the spectral concentration (SC) around the main frequency peak f_p as:

$$SC = \frac{\sum_{0.82f_p}^{1.17f_p} P_{AA}(f_i)}{\sum_{0}^{f_s/2} P_{AA}(f_i)}, \tag{5.5}$$

where P_{AA} is the power spectrum of the AA signal computed using Welch's method with a 8192 point FFT, 1024 point Hamming window and 50% overlap, and f_s is the signal sampling rate (1 KHz). In general, a decrease in the SC corresponds to a better estimate of the atrial activity and thus a good reduction of the ventricular activity present in the EGM. In accordance with the results given in [10], a bandwidth of 2 Hz for a typical AF frequency of 4-6 Hz is chosen which they claim to be sufficient even for AF episodes that show a wideband spectrum with several peaks. If the bandwidth is wider, this parameter should be scaled accordingly. The spectral concentration is measured for the original EGM signals, the EGMs sans VA and the pure atrial EGMs when available (from pseudoreal EGMs).

6

Results

This chapter presents and discusses the results of the CPD and BTD algorithms for the estimation of ventricular activity in AF EGMs. We corroborate the claims made in chapter three that the CPD is valid for SR EGM signals, but not for AF EGM signals, and that BTD indeed provides a viable alternative for the estimation of VA. Interestingly, in spite of the number of gross assumptions made in selecting some of the key parameters in the BTD algorithm, the proposed solution works surprisingly well for most of the datasets. As is often the case when dealing with any set of datasets that are recorded at different times and/or locations, the variations in the recordings necessitate some algorithmic tweaking so as to optimize performance, but overall the methodology proves quite robust and should be able to handle other yet-to-be-recorded datasets estimation errors notwithstanding.

As a side note to facilitate the interpretation of figures present in this chapter, the multiple subplots in a figure depicting an obtained component represent the same component extracted across all electrodes in a particular row, which is why all are labelled with the same component number i.e., 1 or 2, even though there are 8 different subplots.

6.1 Clinical EGM Data

6.1.1 CPD

In this section, we demonstrate that the CPD indeed is a valid model for the decomposition of the EGM data in SR since the rank-1 constraint on each of the dimensions (*temporal* $\times$ *spectral* $\times$ *spatial*) actually holds. As previously mentioned in the explanation of CPDs and how they are computed, this is because the atrial (AA) and ventricular activity (VA) are temporally isolated, thus for the time-frequency bins in which there is signal energy, the energy is due to only one active source; AA or VA. This can be clearly seen in Figures 6.3, 6.4 and 6.5, where we decompose the tensor into 3 components: the depolarization of the AA, the VA, and the repolarization of the AA, which has a different spectral signature from the depolarization of the AA

since it is much slower. Figures 6.1 and 6.2 contain the original EGMs whose source components are shown in the aforementioned figures.
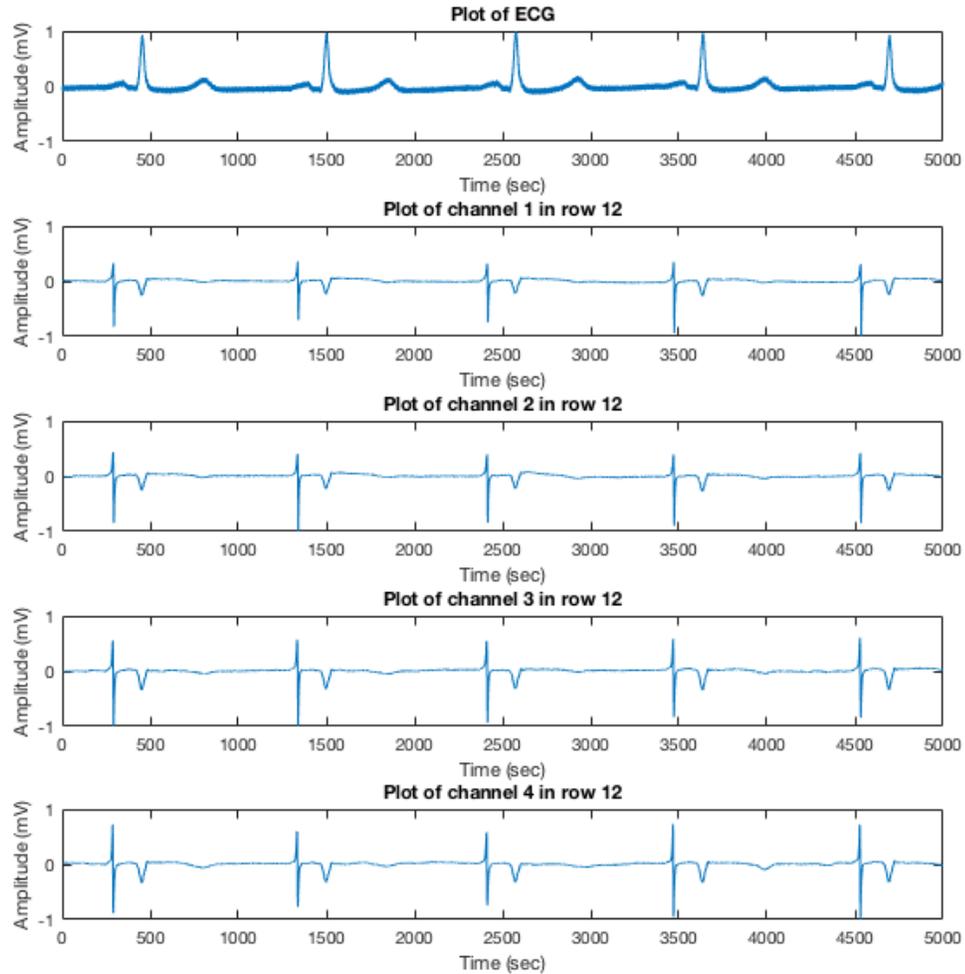
Figure 6.1: Electrode 1 to 4 EGMs in SR for row 12 at location RA2.

However, as shown in Figures 6.8, 6.9 and 6.10, the CPD is not a valid model for the AF EGM data. The original EGMs for these source components are shown in Figures 6.6 and 6.7. This is due to the temporal and/or spectral overlap, such that there can be more than one active source in each time-frequency bin. Therefore, the rank-1
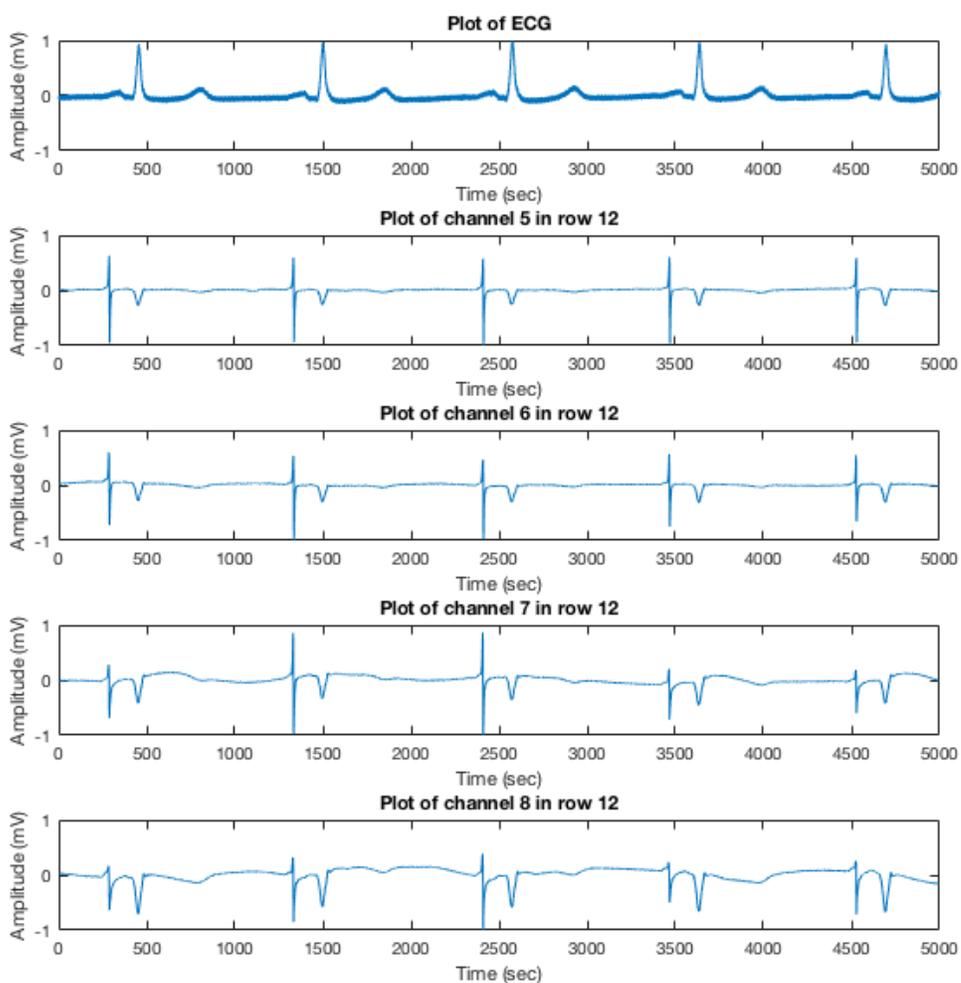
Figure 6.2: Electrode 4 to 8 EGMs in SR for row 12 at location RA2.

constraint is too restrictive since the multiple sources in each time-frequency bin belong to more than one subspace, in which case even though the rank-1 constraint in the spatial dimension may apply, the reconstruction of the source components identified by their (*temporal* × *spectral* × *spatial*) signatures will contain residuals due to the invalid rank-1 temporal and/or spectral components. The AA and VA are therefore not separable, unless the window frames are made small enough to include

49
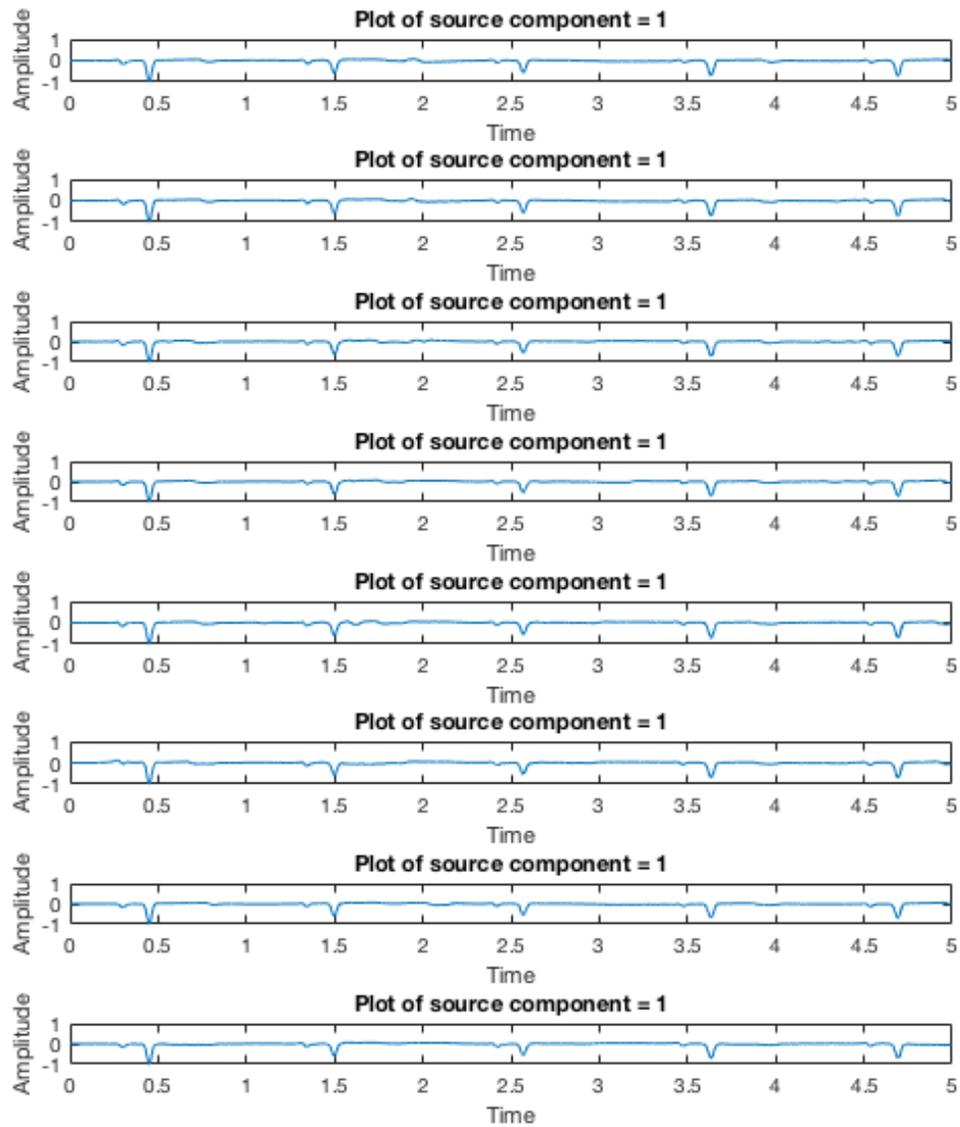
Figure 6.3: Extracted component 1 using CPD for row 12 at location RA2 SR EGMs.
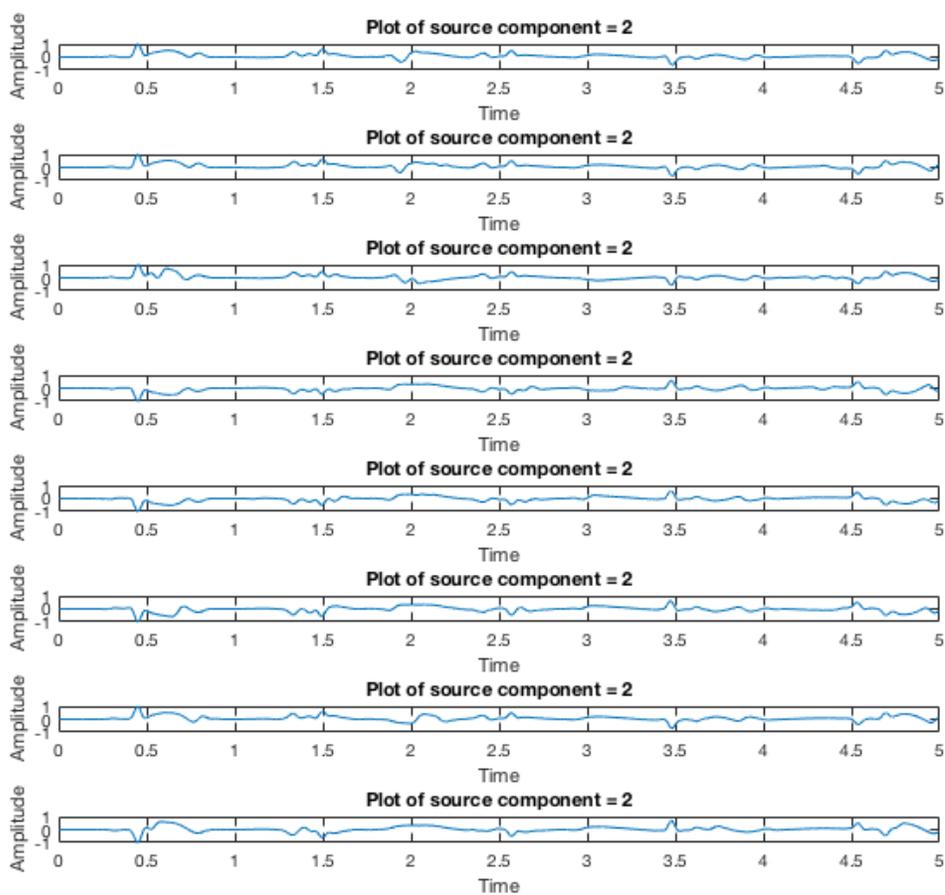
Figure 6.4: Extracted component 2 using CPD for row 12 at location RA2 SR EGMs.

only one active source, which we then in turn have to trade off for spectral resolution since the size of the short-time frequency transform (STFT) window is fixed. Since it is possible to simply relax the rank-1 constraint by using the $(L_r, L_r, 1)$ BTD model instead, we opt to do that instead, and this indeed provides better results for the AF EGM data as seen in the BTD results section.
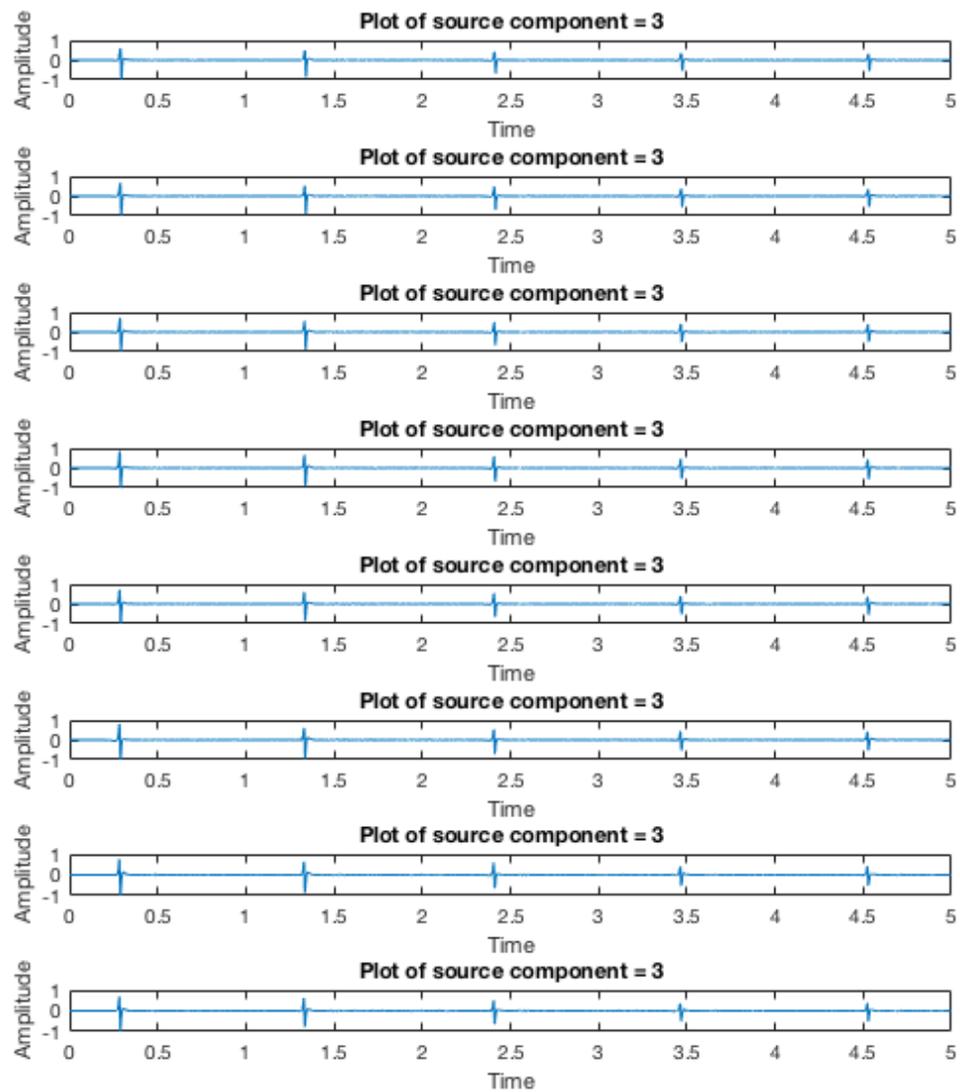
Figure 6.5: Extracted component 3 using CPD for row 12 at location RA2 SR EGMs.
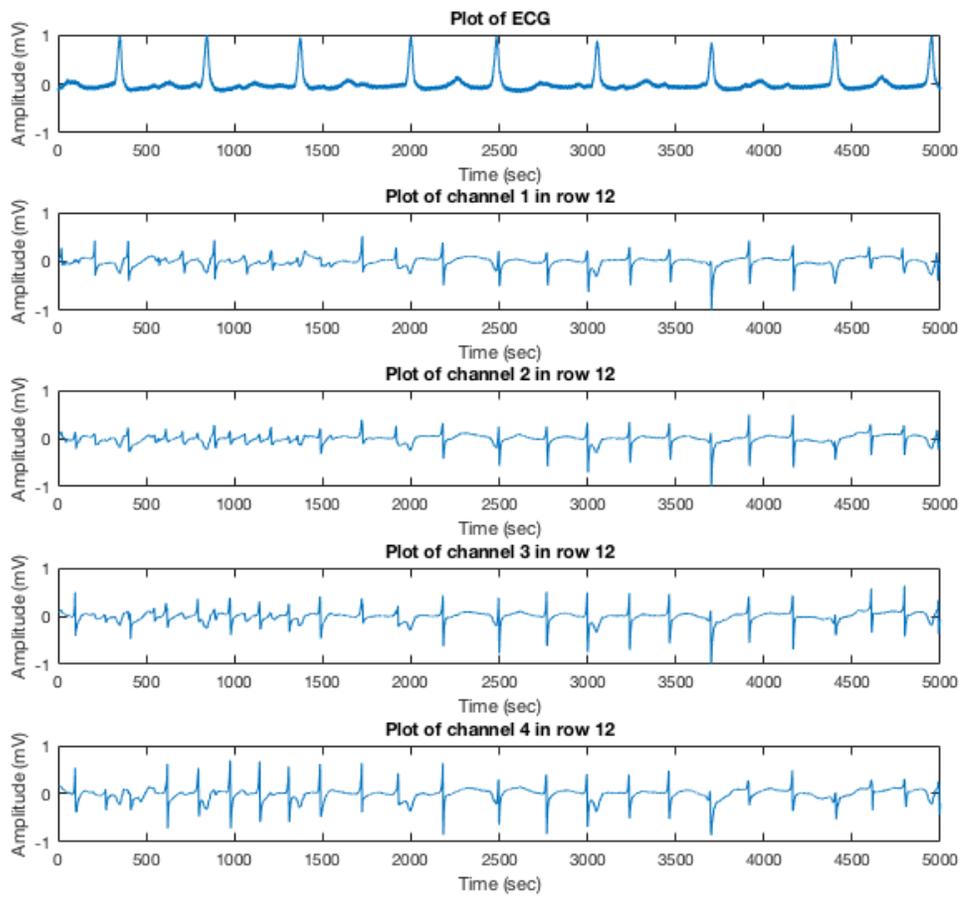
Figure 6.6: Electrode 1 to 4 EGMs in AF for row 12 at location RA2.
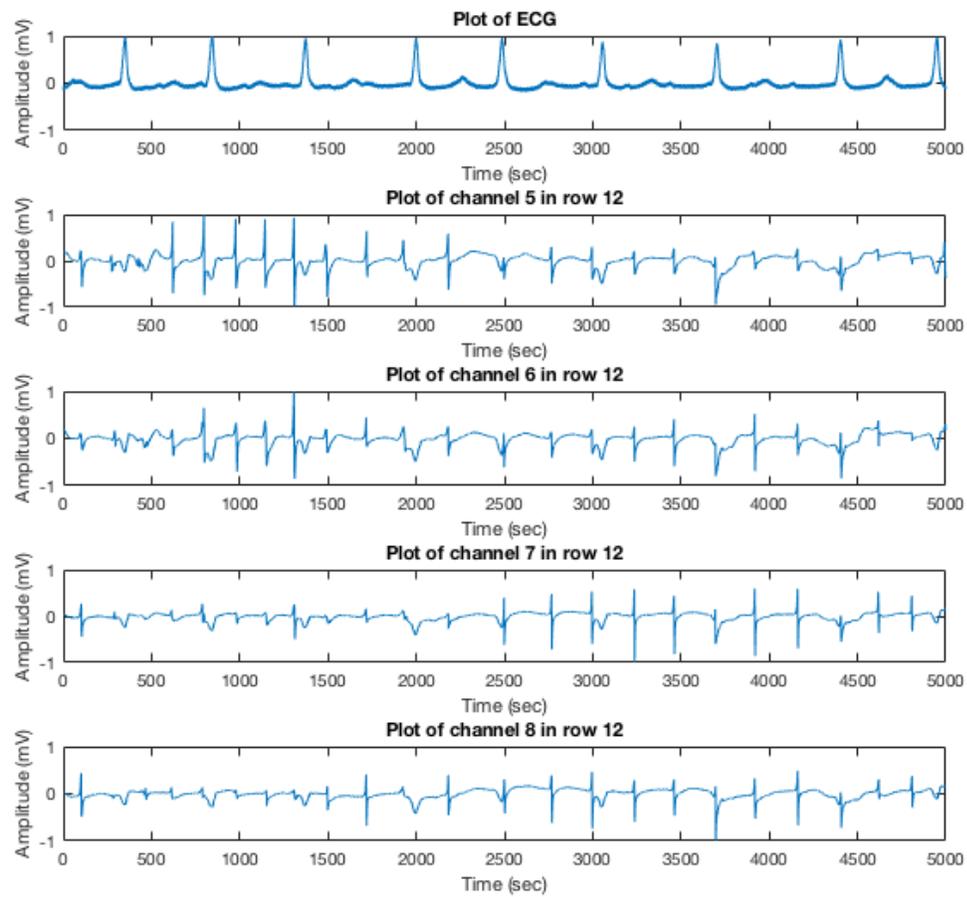
Figure 6.7: Electrode 4 to 8 EGMs in AF for row 12 at location RA2.
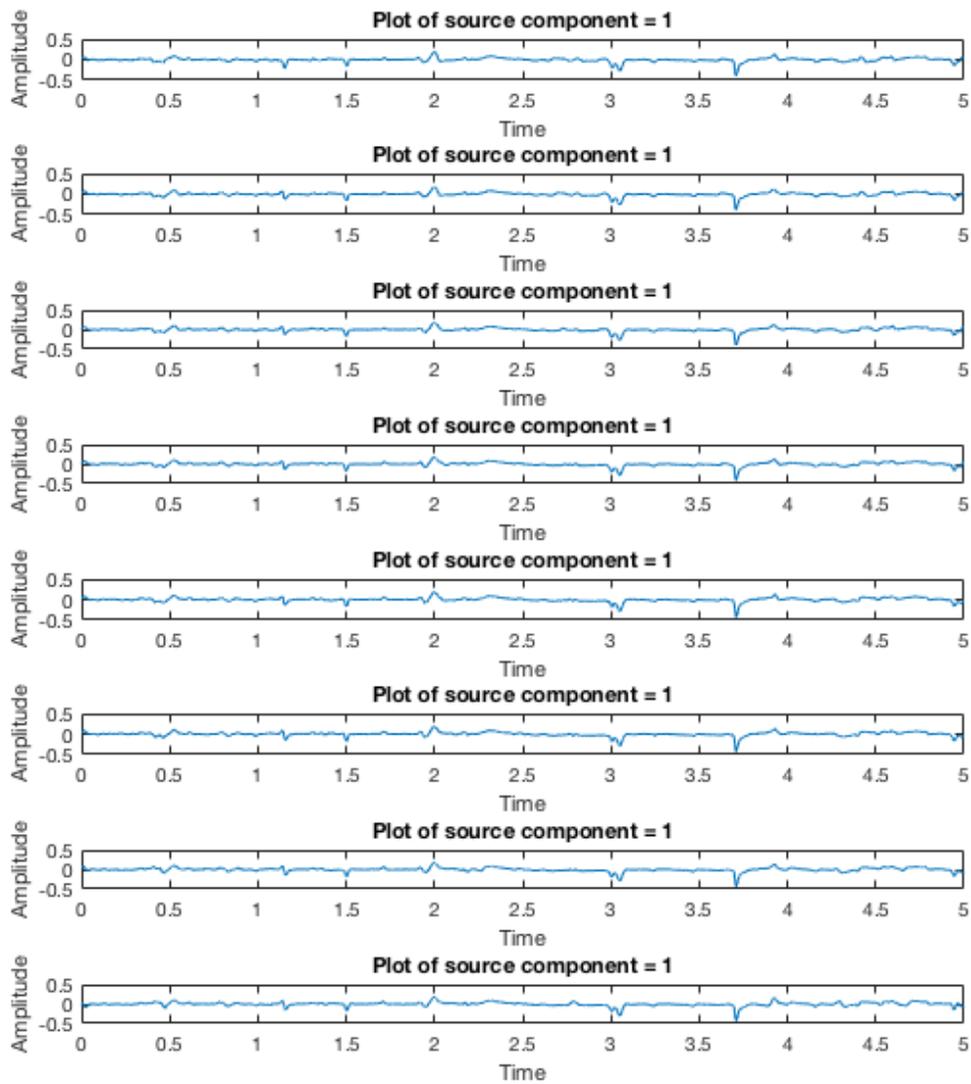
Figure 6.8: Extracted component 1 using CPD for row 12 at location RA2 AF EGMs.
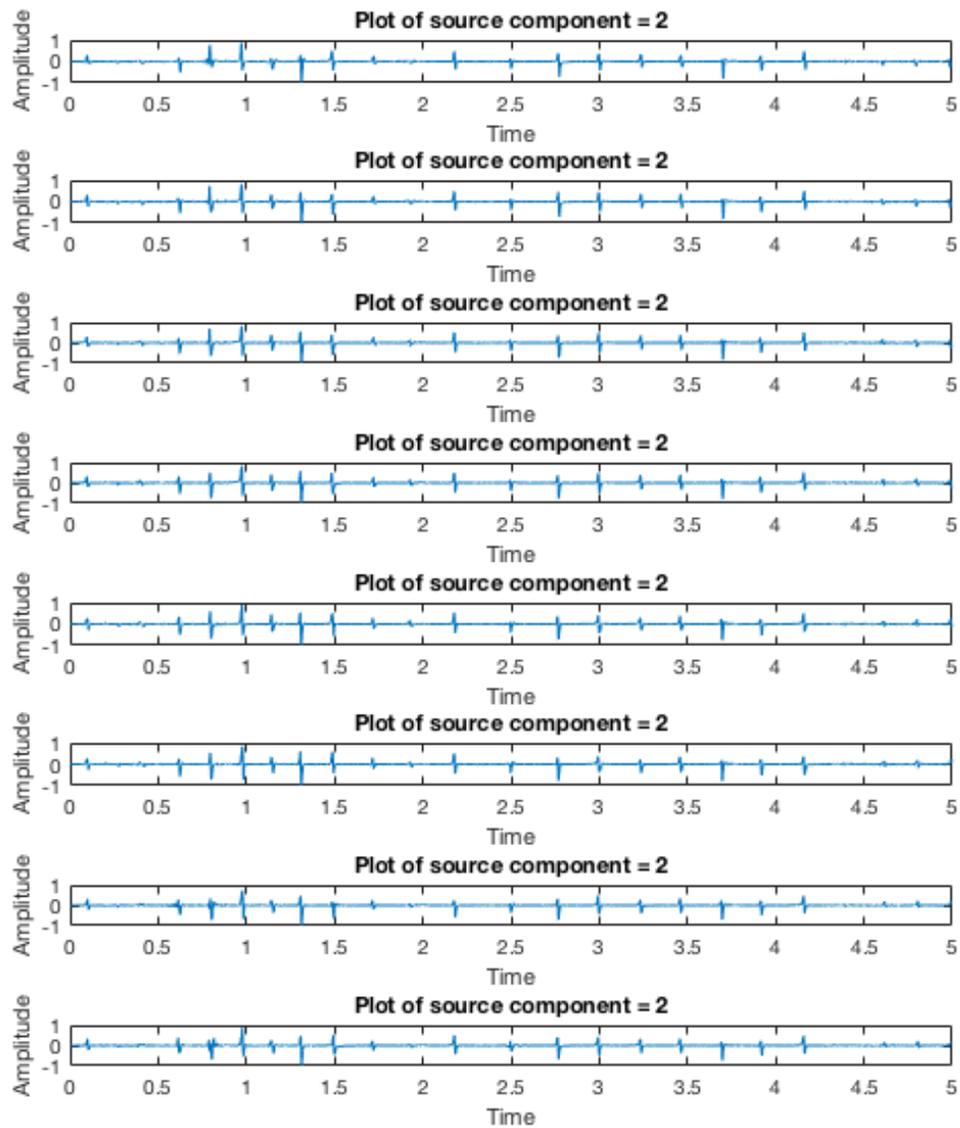
Figure 6.9: Extracted component 2 using CPD for row 12 at location RA2 AF EGMs.
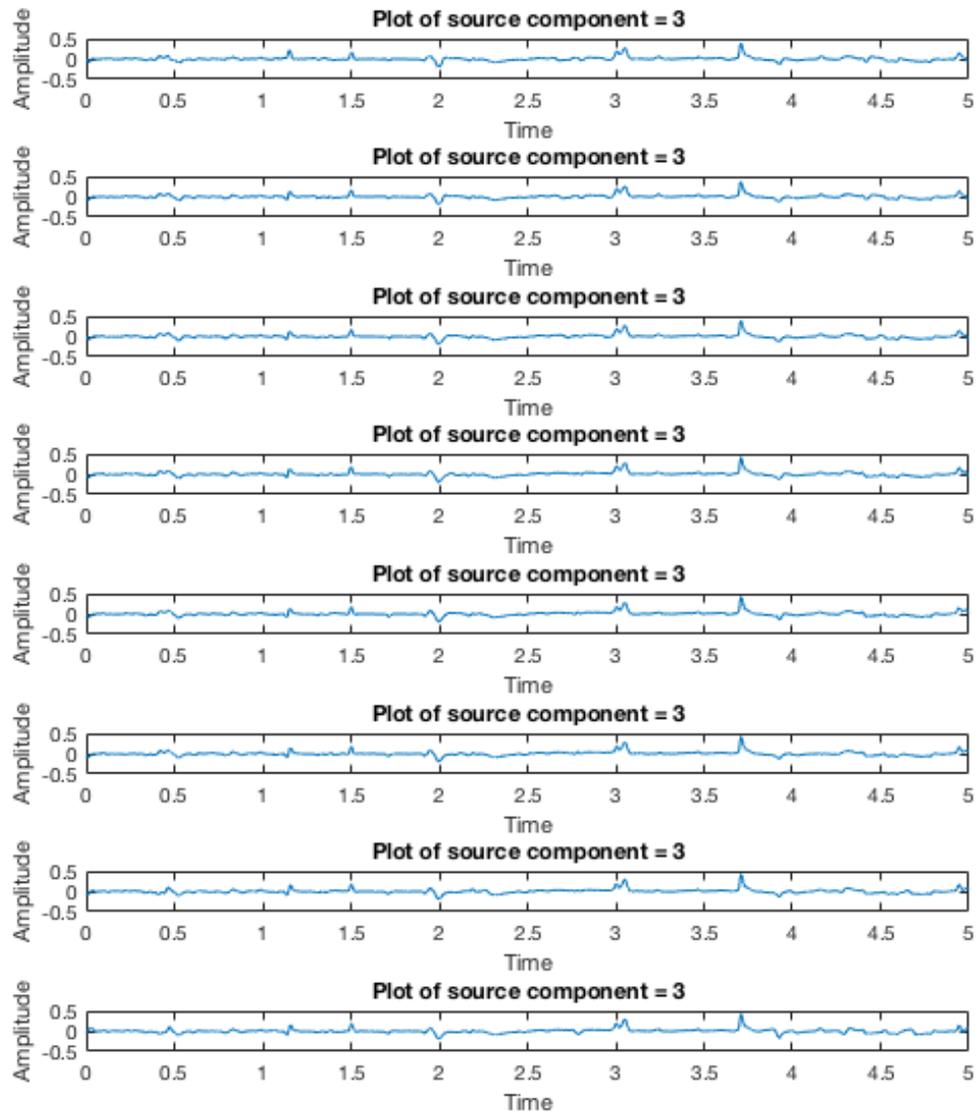
Figure 6.10: Extracted component 3 using CPD for row 12 at location RA2 AF EGMs.

6.1.2 BTD

We now consider the results of the BTD algorithm at length. The original EGMs whose extracted components are depicted in this section are shown in Figures 6.11 and 6.12 for reference. As already demonstrable in Figures 6.13 and 6.14 for row 5 at location RA1, the BTD is indeed capable of extracting the VA simply by relaxing the rank-1 constraint in the temporal and spectral modes, but retaining the rank-1 constraint in the spatial mode. The relaxation basically allows for more than one source to be active in each time-frequency bin, while the rank-1 constraint in the spatial mode confines the linearly independent temporal and spectral columns to unique spatial subspaces for different components.

Visually, it is already possible to conclude from Figure 6.15 that the VA estimate obtained is quite good and minimally modifies the AA when temporally subtracted, and even when VA and AA temporally overlap, it is still possible to preserve the AA to a considerable extent. This is again proven to be true when observing the results of the pseudoreal AF EGM signals in the section "Pseudoreal EGM Data".

The performance of the BTD algorithm is also evaluated using a set of quantitative measures, but given the immense amount of data to be analyzed (7 datasets each with 136 electrodes), we only show tabulated averages and standard deviations of the performance measures per recording location. Since it is important to know how much our model deviates from the actual tensorized datasets, we also tabulate the relative Frobenius norms, especially since we make somewhat brazen and imprecise assumptions like decomposing the tensors into only two components when it is very possible that there are more than two latent sources.

Table 6.1 compares the results of temporal subtraction of the VA estimate obtained using the BTD algorithm to that obtained using average beat subtraction (ABS). Table 6.2 shows the relative Frobenius norm of the tensor data to its BTD.

6.1.3 Discussion

It is difficult to tell with absolute certainty that the reduction in ventricular activity (VA) for all the electrodes at a particular recording location is good based on the performance metrics since only the averages and standard deviations are tabulated in Table 6.1. We can however on average claim to have a substantial reduction in VA for most of the recording locations as indicated by low VASR values and an overall decrease in the spectral concentration (SC). Additionally, changes in the dominant frequency (DF) between the original EGMs and the AA estimates correspond to
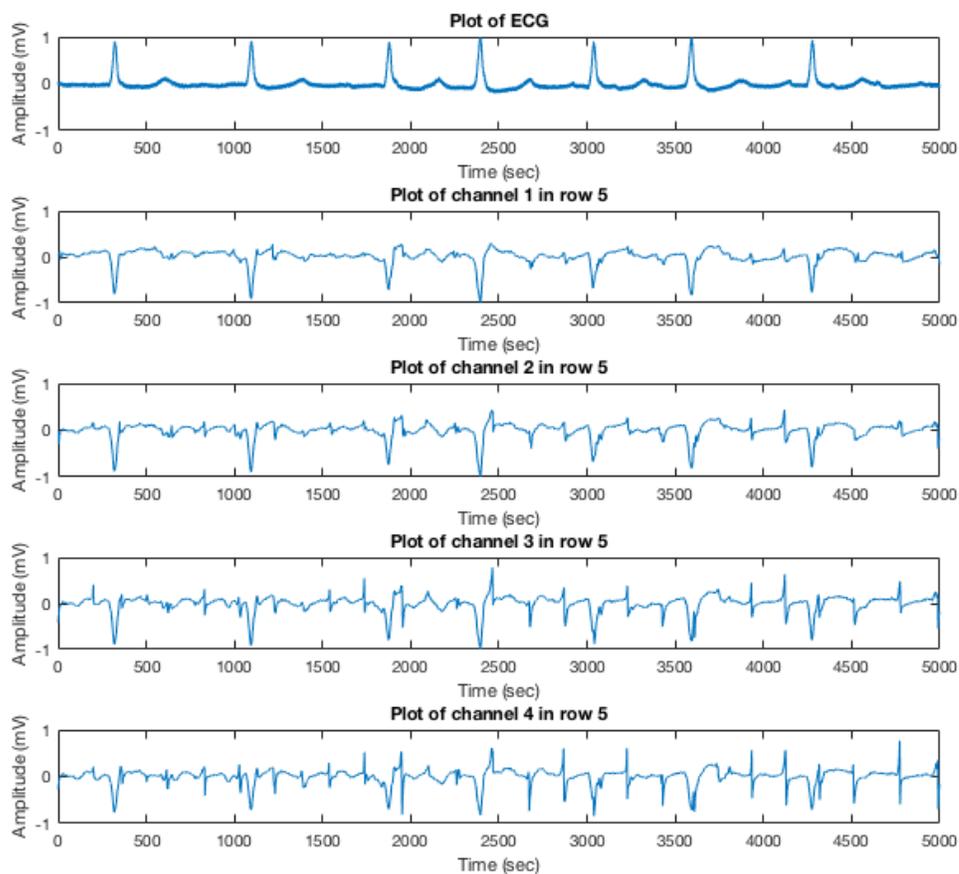
Figure 6.11: Electrode 1 to 4 EGMs in AF for row 5 at location RA1.

reduction in VA since the presence of VA modifies the dominant atrial frequency due to spectral overlap of AA and VA. However, this metric should not be used on its own to qualify the goodness of VA reduction because no change in the DF is observable when there is no spectral overlap.

More specifically, comparison with the results of ABS indeed points to a reduction in VA since ABS is known to work well in practice, but this should be considered with a grain of salt since ABS has its own shortcomings despite being the go-to method for VA removal in AF EGMs. It is also apparent from some relatively higher VASR values that the BTD algorithm does not perform as well for certain
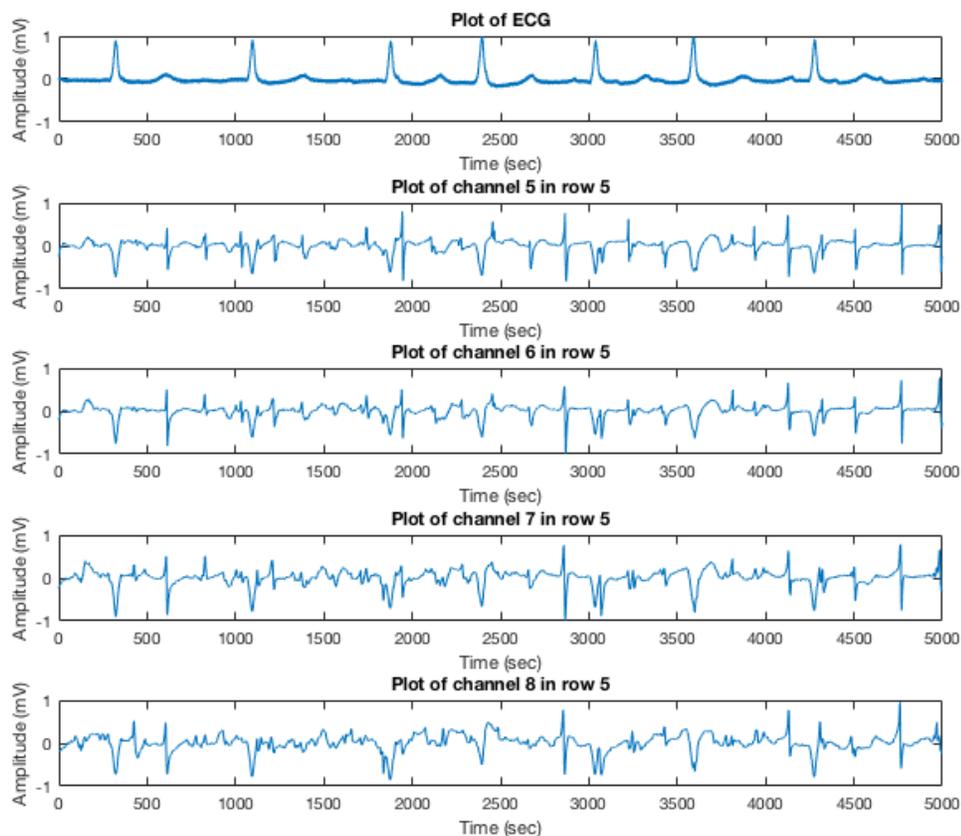
Figure 6.12: Electrode 4 to 8 EGMs in AF for row 5 at location RA1.

recording locations, particularly location LA3 and RA1. This is likely due to the organized nature of the AF EGMs recorded at these locations whose AA and VA overlap temporally, spectrally and/or spatially such that the extracted components contain residuals making it difficult to identify the VA component solely based on signal kurtosis. The estimation errors are most probably due to selection of the wrong component for subtraction.

Since the quantitative values shown here are averages, it is a good idea to compare the results of the metrics on an electrode-by-electrode basis to identify the erroneous estimates, especially if the AF EGM data sans VA is to be used for mapping purposes. We opted not to do this here due to the sheer amount of space that would be
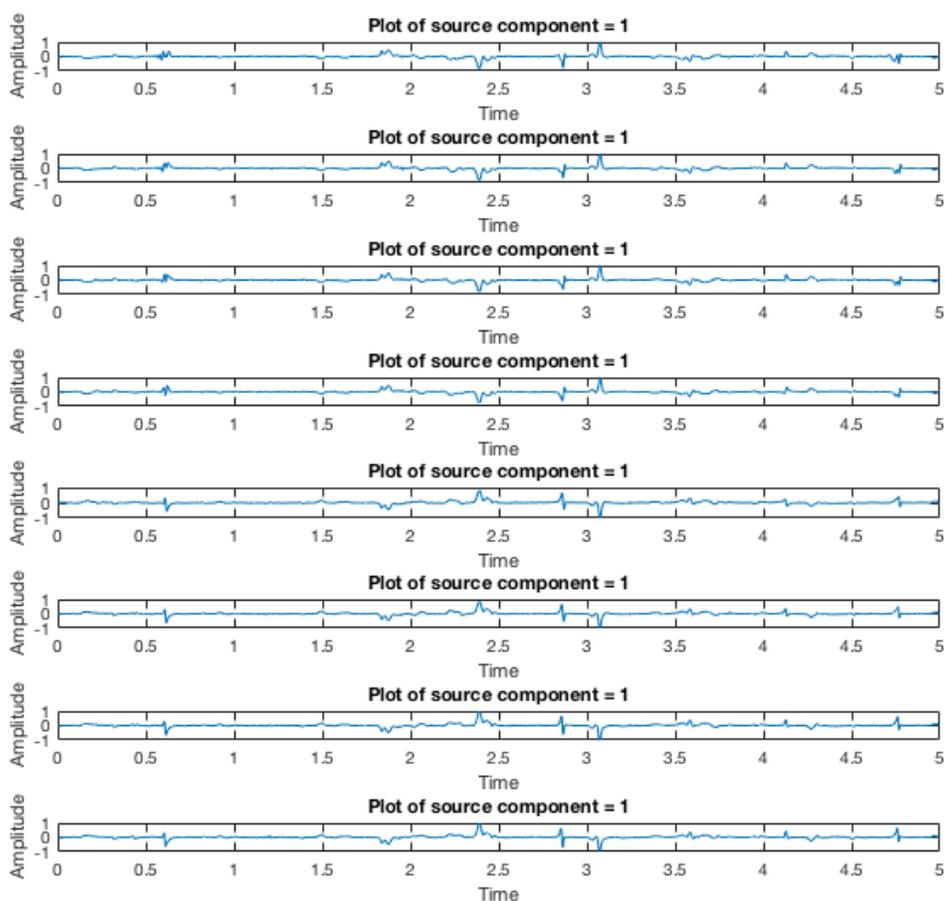
Figure 6.13: Non-VA component using $L_r, L_r, 1$ BTD for row 5 real AF EGMs at RA1.

required to display all the values, without adding much insight on the usefulness of the BTD. Visually analyzing the AA signal estimates if in doubt as to the accuracy of the algorithm is also recommended, possibly with the aid of the associated ECG signal when necessary and available. Furthermore, if we look at the VASR values, a lower VASR does not categorically indicate better VA reduction since we lack a benchmark value below which we can assert that VA is significantly reduced or completely removed. In theory, one would have to determine such a value for each dataset and visually verify its verity by looking at the AA estimate, before using it to
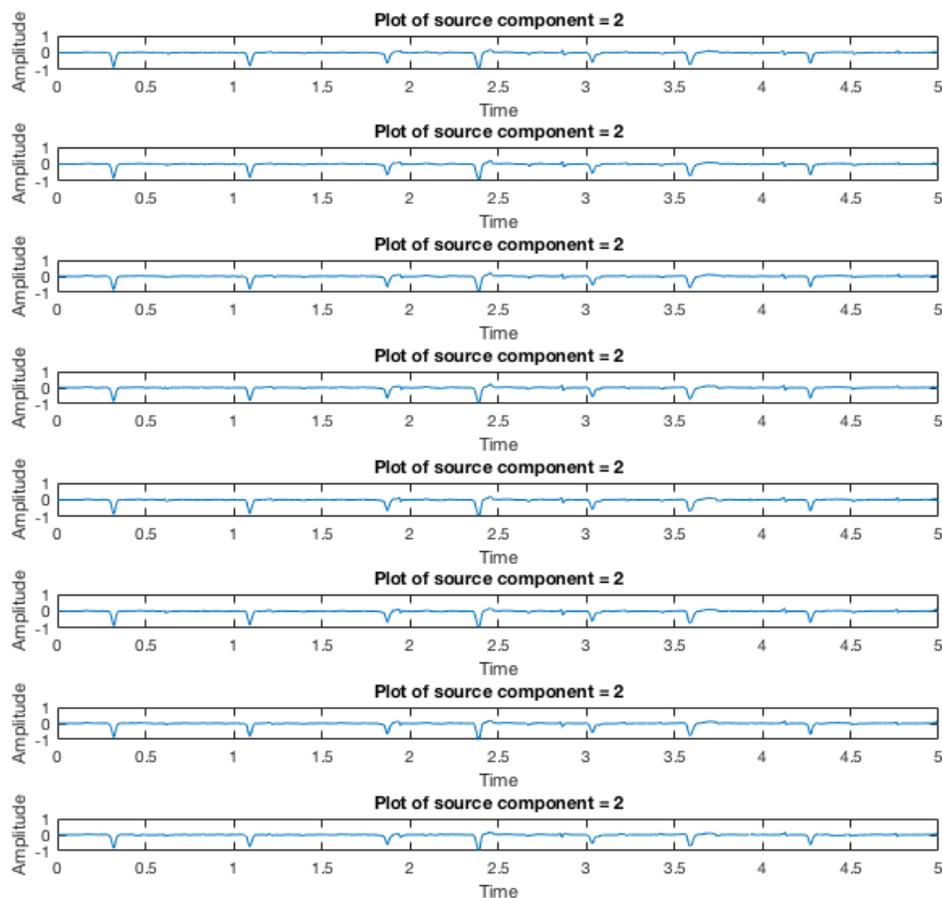
Figure 6.14: Ventricular activity component using $L_r, L_r, 1$ BTD for row 5 real AF EGMs at RA1.

qualify our algorithmic performance. This is not done here, although it is relatively straightforward if one chooses the right electrode as the reference standard; the tricky part is deciding which electrode to use.

On the whole, without pure AA signals against which to measure the effectiveness of VA reduction, caution should be taken when making definitive statements on algorithmic performance using only real AF EGMs since the performance metrics
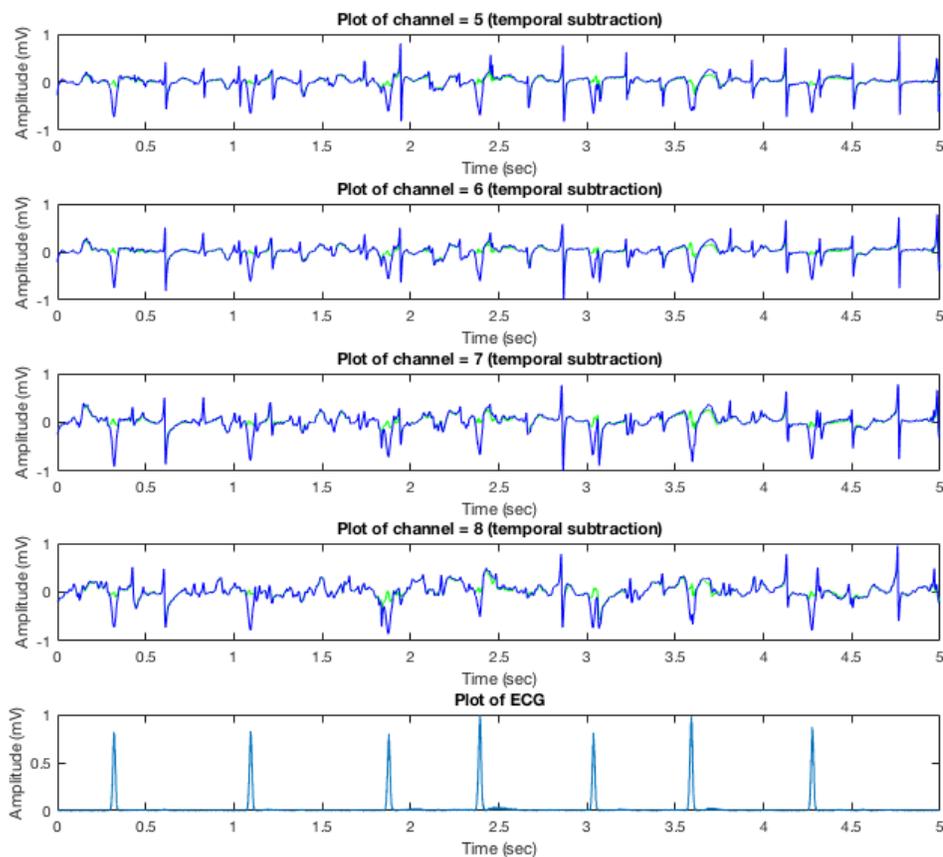
Figure 6.15: Obtained atrial activity after temporal subtraction for electrodes 4 to 8 in row 5 real AF EGMs at RA1.

themselves need to be investigated to determine appropriate thresholds for comparison purposes. This is one of the many reasons why we opted to generate pseudoreal AF EGM signals; while an imprecise representation of reality, the availability of pseudoreal pure AA signals allows us to more meaningfully analyze the performance metrics. However, it should be noted that the inexactness of the pseudoreal AF EGM synthesis procedure may also at times contribute to either decreased performance or over-optimistic results.

Table 6.1: Comparison of ABS to BTD for real EGM data as measured by the mean of performance metrics across all electrodes at a recording location (the standard deviation is shown in parentheses).

Recording Location	Performance Metric	ABS AA Estimate	BTD AA Estimate	Original EGM
BB0	VASR	10.98 (5.26)	2.16 (1.62)	-
	DF	3.91 (0)	3.9 (0.02)	3.91 (0)
	SC	0.28 (0.13)	0.3 (0.14)	0.29 (0.07)
LA3	VASR	6.28 (2.86)	4.35 (2.3)	-
	DF	4.51 (0.26)	4.57 (0.28)	4.51 (0.04)
	SC	0.1 (0.03)	0.1 (0.02)	0.13 (0.03)
RA1	VASR	7.63 (3.37)	4.53 (2.42)	-
	DF	4.74 (1.02)	4.78 (1.03)	4.53 (1.03)
	SC	0.2 (0.06)	0.18 (0.06)	0.2 (0.04)
RA2	VASR	7.22 (1.17)	3.52 (1.95)	-
	DF	4.21 (0.71)	4.12 (0.88)	4.03 (0.49)
	SC	0.13 (0.05)	0.12 (0.06)	0.16 (0.05)
RA3	VASR	6.28 (1.44)	3.49 (2.27)	-
	DF	4.13 (0.28)	4.1 (0.63)	4.08 (0.18)
	SC	0.16 (0.05)	0.14 (0.05)	0.18 (0.04)
RA4	VASR	7.07 (1.56)	3.9 (2.08)	-
	DF	4.06 (0.15)	3.97 (0.31)	3.95 (0.1)
	SC	0.15 (0.05)	0.14 (0.04)	0.18 (0.04)
RA5	VASR	8.63 (1.75)	3.79 (2.11)	-
	DF	5.09 (0.6)	4.75 (0.68)	4.42 (0.47)
	SC	0.2 (0.04)	0.19 (0.05)	0.2 (0.05)

Table 6.2: Mean and Standard Deviation (std) of the Frobenius Norm between the Actual Dataset and the Decomposed Dataset.

Recording Location	Mean Frobenius Norm (std)
BB0	0.19 (0.09)
LA3	0.2 (0.06)
RA1	0.29 (0.1)
RA2	0.29 (0.07)
RA3	0.35 (0.03)
RA4	0.31 (0.06)
RA5	0.21 (0.05)

6.2 Pseudoreal EGM Data

6.2.1 BTD

Here we show the results of the BTD algorithm using pseudoreal AF EGM data. As seen in Table 6.3, the mean cross-correlation for all electrodes in the various datasets is quite high, while the mean NMSE is quite low, indicating good performance. The few datasets for which the performance is not as satisfactory shall be dealt with in the discussion section, as well as suggesting ways to enhance the performance for those particular datasets.

The VASR for the AA estimate and the pure AA are also comparable, as shown in Table 6.3, also pointing to a good reduction in the VA. Even more important is the comparison of the dominant frequencies and spectral concentrations between the 'pure' AA and the AA estimates for the various datasets, shown in Table 6.3, whose closeness reveals the significant reduction in VA across the electrodes, outliers and algorithmic errors notwithstanding. The Frobenius norm between the pseudoreal data and the BTD model is once again displayed to illustrate the level of inaccuracy of the chosen model.

6.2.2 Discussion

The quantitative results for pseudoreal AF EGM data also confirm the reduction/removal of VA that can be observed visually. Once more, the exact value of the dominant frequency does not tell us much with regard to VA reduction, but the fact that is changes relative to that of the original EGM signals implies that VA has been affected somehow. Besides, the closeness between the DF of the AA estimates and the pseudoreal pure AA verifies the reduction in VA compared to the mixed pseudoreal AF EGMs. With regard to the spectral concentration metric, it is impossible to determine if the values observable in Table 6.3 indicate a reduction in VA. The uninterpretability is probably caused by the way in which the pseudoreal AF EGM signals are generated since they are not an accurate representation of the generation mechanism in the heart.

As for the remaining performance indices, the availability of 'pure' AA signals enables the calculation of cross-correlation coefficients and NMSEs. We are able to observe good VA reduction based on relatively high cross-correlation coefficients and low NMSE values for most of the recording locations. The similarity between VASR for the 'pure' AA and the AA estimates also implies that the VA indeed has been either

Table 6.3: Comparison of ABS to BTD for pseudoreal EGM data as measured by the mean of performance metrics across all electrodes at a recording location (the standard deviation is shown in parentheses).

Recording Location	Performance Metric	ABS AA Estimate	BTD AA Estimate	Pure AA	Original EGM
BB0	CC Coeff	0.91 (0.03)	0.9 (0.06)	-	-
	NMSE	0.45 (0.2)	0.39 (0.13)	-	-
	VASR	10.87 (3.33)	2.91 (1.98)	3.33 (2.19)	-
	DF	3.91 (0.02)	3.9 (0.02)	3.9 (0.02)	3.9 (0.03)
	SC	0.29 (0.11)	0.27 (0.09)	0.29 (0.1)	0.25 (0.05)
LA3	CC Coeff	0.84 (0.18)	0.72 (0.24)	-	-
	NMSE	0.35 (0.4)	0.68 (0.67)	-	-
	VASR	8.04 (4.04)	4.33 (2.85)	8.47 (2.33)	-
	DF	4.57 (0.38)	4.49 (0.35)	4.54 (0.12)	4.06 (0.56)
	SC	0.09 (0.03)	0.1 (0.03)	0.1 (0.04)	0.12 (0.03)
RA1	CC Coeff	0.89 (0.07)	0.81 (0.17)	-	-
	NMSE	0.26 (0.18)	0.49 (0.52)	-	-
	VASR	6.21 (2.53)	3.58 (1.84)	4.18 (1.92)	-
	DF	4.64 (1.1)	4.9 (1.03)	4.98 (1.07)	4.84 (1.06)
	SC	0.18 (0.06)	0.17 (0.05)	0.21 (0.06)	0.19 (0.05)
RA2	CC Coeff	0.91 (0.03)	0.82 (0.1)	-	-
	NMSE	0.22 (0.07)	0.42 (0.25)	-	-
	VASR	7.91 (1.25)	4.3 (2.29)	4.87 (1.33)	-
	DF	4.39 (0.88)	4.16 (0.81)	4.09 (0.45)	3.95 (0.33)
	SC	0.13 (0.05)	0.12 (0.05)	0.14 (0.05)	0.16 (0.04)
RA3	CC Coeff	0.93 (0.03)	0.77 (0.1)	-	-
	NMSE	0.15 (0.06)	0.52 (0.21)	-	-
	VASR	5.21 (1.23)	3.14 (1.98)	4.32 (1.28)	-
	DF	4.08 (0.24)	3.98 (0.51)	4.07 (0.29)	4.05 (0.2)
	SC	0.18 (0.06)	0.15 (0.05)	0.18 (0.07)	0.17 (0.04)
RA4	CC Coeff	0.93 (0.02)	0.69 (0.17)	-	-
	NMSE	0.15 (0.06)	0.66 (0.36)	-	-
	VASR	6.31 (2.04)	2.84 (1.91)	4.2 (1.24)	-
	DF	4.03 (0.29)	4.11 (0.69)	4.06 (0.22)	3.84 (0.21)
	SC	0.14 (0.04)	0.12 (0.04)	0.14 (0.04)	0.15 (0.04)
RA5	CC Coeff	0.92 (0.02)	0.79 (0.11)	-	-
	NMSE	0.28 (0.1)	0.47 (0.19)	-	-
	VASR	7.1 (1.92)	1.8 (1.49)	2.44 (1.34)	-
	DF	5.27 (0.62)	5.18 (0.74)	5.33 (0.63)	5.24 (0.58)
	SC	0.21 (0.04)	0.2 (0.05)	0.23 (0.04)	0.22 (0.04)

substantially reduced or completely removed. Again, the performance at certain locations is subpar compared to ABS due to significant overlaps of AA and VA in the temporal, spectral and/or spatial modes. In such cases, while the ABS does not provide any analytical insight into the AF EGM data, it still manages to remove most of the VA by pinpointing the window frames with VA using the ECG's QRS complex and subtracting the average VA template.

We would however like to once more emphasize we are examining averages and the

Table 6.4: Mean and Standard Deviation (std) of the Frobenius Norm between the Pseudoreal Dataset and the Decomposed Dataset.

Recording Location	Mean Frobenius Norm (std)
BB0	0.2 (0.09)
LA3	0.15 (0.03)
RA1	0.31 (0.09)
RA2	0.21 (0.03)
RA3	0.36 (0.04)
RA4	0.35 (0.07)
RA5	0.28 (0.05)

performance for each electrode might vary, necessitating a more thorough evaluation of the metrics on an electrode-by-electrode basis if the AF EGM data is to be used in applications that require high-resolution AF EGMs. In addition, while we now sort of have a threshold via the 'pure' AA signal against which to compare our calculated performance metrics for the AA estimates, the inaccuracies associated with the generation procedure for pseudoreal EGMs may influence the values of the performance metrics. Nevertheless, given the lack of an alternative method of acquiring pure AA signals, our methodology will have to suffice for now.

It is apparent that despite the albeit somewhat inaccurate assumptions made for purposes of simplicity, for example the number of components into which the tensor is decomposed and the heuristic approach to determining the low multilinear rank of the tensors, we are still able to obtain a fairly good estimate of the VA, which is thereafter temporally or power spectrally subtracted from the original AF EGM recordings. While temporal subtraction usually suffices for most of the datasets, there are more challenging datasets like those recorded at locations LA2, PVR1 and PVL1, where the VA's morphology significantly varies for electrodes in the same row, not only with regard to shape, but also with regard to sign. As shown in Figure 6.16, the VA appears as a positive sweep for beats 6, 8 and 9, and as a negative sweep for beats 2 and 5. This becomes problematic when correcting for the sign and scaling ambiguities of FA-BSS techniques, and temporal subtraction often results in oversubtraction. As a workaround to this, power spectral subtraction proves useful, with the minimum capped at zero so that you cannot subtract more energy than is actually present in a particular TF bin, since it is the energy of the signal (squared DFT coefficients for each TF bin), and not the morphology of the signal in the time domain, that determines what is left over after subtraction.

Most of the outliers and/or erroneous results in each dataset are due to the fairly

67
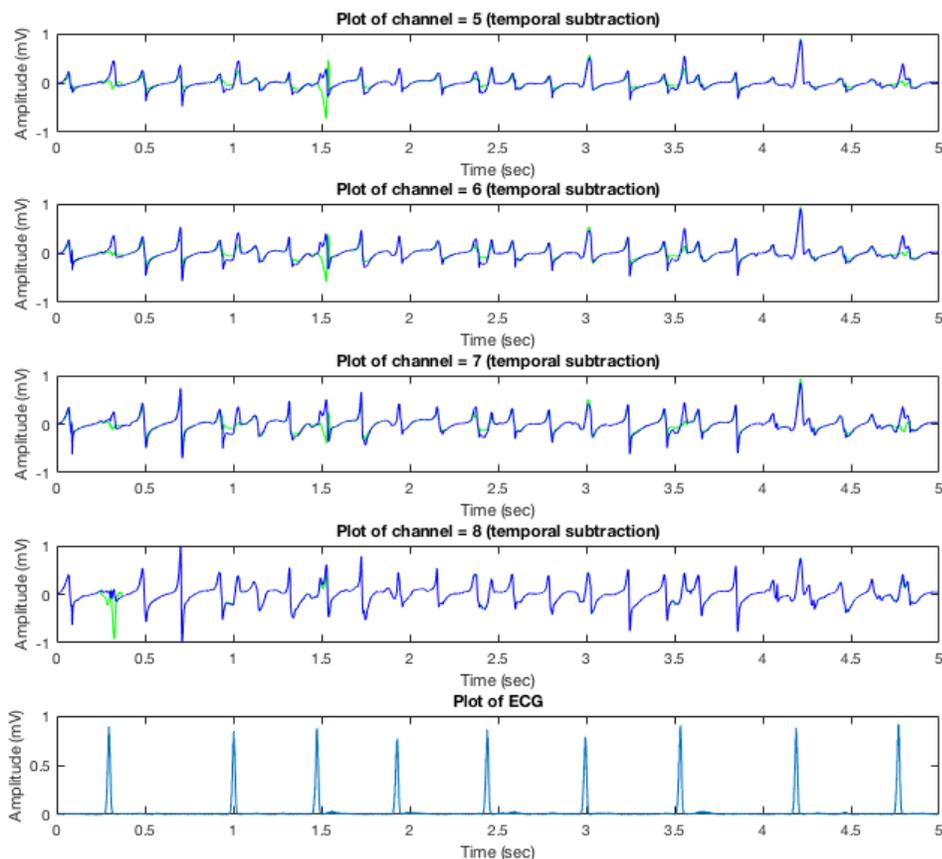
Figure 6.16: Illustration of the appearance of VA as both a positive and negative sweep in row 15 PVL1 AF EGMs.

simplistic component selection process wherein the signal kurtosis is used to decide which source component is the VA and which is not. We use the standard deviation of the kurtosis of the VA extracted from different electrodes in the same row as a deciding factor i.e., a Euclidean distance measure of the electrode signal kurtoses from the mean VA signal kurtosis, since the VA is consistent across the electrodes while the local AA varies from electrode to electrode such that the non-VA component standard deviation will be higher. This of course fails when you have highly organized AA, or if the AA closely resembles the VA, such that AA residuals are present in the VA estimate possibly making the standard deviation of the VA component larger

68

than that of the non-VA component.

Furthermore, the idiosyncrasies unique to each recording location necessitate the alteration of input parameters. We however use the same BTD parameters for all analyzed datasets to keep things simple, but it is likely that properly adjusting the various input parameters for each dataset based on a more thorough understanding of its properties will improve the algorithmic performance. Irrespective of the pro's and con's of the BTD approach, the fact that the algorithm is more or less fully automatic is a big plus. Computation time still poses a bit of a challenge, since tensor decompositions are computationally taxing, but improvements in computational efficiency as well as increased computational power should soon alleviate this problem.

6.3 Conclusion

The results of the various algorithms evaluated in this thesis report are presented in this chapter, and their suitability for MWCA of epicardial EGM data, both in SR and AF, is discussed. It has been verified that the CPD algorithm works for SR data since the rank-1 constraint in the signal model actually holds, but that CPD fails for AF EGM data. BTD is shown to provide superior results as measured by the quality of the VA estimated that is obtained. The VA estimate is either temporally or power spectrally subtracted, depending on the nature of the recordings in the dataset, but may also be used for other more sophisticated methods like nonlinear Wiener filtering and semi-blind nonnegative matrix factorization.

The algorithmic performance is evaluated both visually and quantitatively, and the dissatisfactory as well as erroneous results are explained in the discussion section of this chapter. This concludes the bulk of the research done on MWCA for the removal of far ventricular signal in AF EGMs. In the succeeding chapter we wrap up the report and make recommendations for future work in line with the objectives of this master's thesis project and the overall goals of research on atrial fibrillation within the circuits and systems group at TU Delft in collaboration with Erasmus MC Rotterdam.

Conclusions and Future Work **7**

In this chapter we round up the thesis project by stating conclusions on tensorial analysis of AF EGM datasets and making recommendations for future research work. Although we have been able to automatically remove the VA via multiway component analysis to a fairly high accuracy on most datasets, and thus have achieved the research objective set out at the beginning of the thesis project, there is definitely more to be done. This is primarily with regard to refining the quality of the results so as to obtain even better performance. More specifically, we still need to eliminate erroneous estimates, properly handle the idiosyncrasies of AF EGM data recorded at different locations, and further exploit multiway component analysis techniques for the characterization of AA. This chapter covers these aspects to some level of detail, and the reader is encouraged to use his/her imagination to conceive further applications of MWCA for the analysis of AF EGM signals.

7.1   Conclusions

Clearly, the AF EGM datasets prove quite challenging when it comes to the automatic extraction and identification of source components using FA-BSS algorithms because the components not only overlap temporally and spectrally, but also spatially at times. Ideally, source separation/factor analysis works excellently if one can find a domain or combination of domains in which the sources/factors are separable and identifiable based on statistical and/or other signal properties. However, if this is not possible, we are usually compelled to either settle for subpar performance or devise a way to improve the data model based on a thorough understanding of the signals being dealt with. Unfortunately, at the time of writing this thesis report, statistical information on AA and VA was still not at our disposal, and so we resorted to multilinear data analysis techniques that map the datasets to unique subspaces (linearly independent bases) in the temporal, spectral and spatial domains, whose connections are maintained through a core tensor. In this way, it is presumed that the extracted latent components i.e., distinct temporal, spectral and spatial factors/signatures, represent physiologically interpretable sources.

However, the uncertainties (unknown number of source components, unknown low multilinear rank, arbitrary permutation and scaling) associated with FA-BSS pose a difficulty, warranting the use of heuristics to extract meaningful latent components. In spite of the unknown low multilinear rank of the different source components and the unknown overall number of components into which the tensors are to be decomposed, by simply factorizing the tensors into two components while constraining the spatial dimension to be rank-1, we are able to get an estimate of VA. This is because for each constructed tensor, the far VA more or less presents itself to the electrode array as a signal originating from one direction. However, due to the overlapping factors i.e., the bases associated with AA and VA are not entirely different depending on the dataset, AA residuals are sometimes present in the extracted VA component. In such cases, a better approximation of the low multilinear rank may be able to separate the components due to the increased or decreased number of linearly independent subspaces i.e., a better data model. A more suitable guess of the overall number of components, also corresponding to a better data model, may be able to extract the truly different latent components present in the tensorized dataset. Note that determining the best low multilinear rank of a tensor is an NP-hard problem and is thus typically solved numerically using trial-and-error and/or best-fit approaches [35].

The adequate algorithmic performance of block term decomposition (BTD) is substantiated both visually and quantitatively as can be seen in the results section, and serves as a good starting point to advance the study of AF EGMs from a $(temporal \times spectral \times spatial)$ approach, as opposed to matrix decompositions that not only limits analysis to two modes, but additionally imposes strict constraints to guarantee uniqueness that are not necessarily applicable to AF EGM datasets. This last remark is quite relevant to nonstationary, dynamically complex physiological data that is at times difficult to model; unlike signals, for example in telecommunications, that are generated with the explicit intent of facilitating interpretability, in biomedical applications we attempt to demystify the surprisingly efficient functionality of the human body. While undoubtedly an exigent undertaking, the benefits thereof suffice to motivate the constant pursuit of a deeper understanding of, in our case, the generation and propagation of anomalous atrial signals in patients suffering from atrial fibrillation.

Using the selected algorithmic procedure, we are able to substantially reduce the presence of VA in AF EGMs, which is the first step in achieving the long-term goal of the research project on atrial fibrillation at TU Delft in collaboration with Erasmus MC; namely, the development of age- and gender-based bioelectrical diagnostics to characterize the invasive and noninvasive AF fingerprint thus enabling optimal AF

treatment and improving the therapeutic outcome of patients with AF.

7.2 Future Work

In general, FA-BSS usually presents a hurdle when it comes to interpretation because the mathematics works, but what do the extracted components actually mean? Even more so when dealing with 'abnormal' physiological signals whose variability often preempts the 'Aha! I see it!' solution to the problem. Furthermore, we had no way of telling apriori the number of components to extract or the low multilinear rank of the tensors, and thus relied on a trial-and-error approach, which of course is not robust to the natural variance in data recordings due to noise and/or other contaminants. Therefore, there is indeed much more to be done, not only to improve on the removal of VA, but to extend the usability of multiway data analysis techniques to the characterization of AA. For instance, is it possible to approximate the number of sources active in each time-frequency bin based on the multilinear rank of the $(time \times frequency \times electrodes)$ tensor? Or maybe extract temporal and/or spectral AA signatures from an AF patient based on a decomposition of the $(temporal \times spectral \times electrode \times trial)$ tensor to study the evolution of AF? Indeed, multiway component analysis shows a lot of promise in terms of allowing the joint analysis of multivariate or disparate datasets with the goal of acquiring ever more pertinent diagnostic and therapeutic details on AF.

Future possibilities notwithstanding, there are a few recommendations to be made within the scope of this project. For starters, the number of components into which the tensor is decomposed. For purposes of simplicity, we made the assumption that one component, the VA, originated from one directional subspace, and that 'everything else' should be mapped to the other component's subspace. This of course only works 100% if the VA is entirely separable from 'everything else' in one (or all) of the three domains i.e., it occupies a unique subspace, otherwise residuals are to be expected. Moreover, while this allows us to use the VA estimate to temporally or power spectrally subtract it from AF EGMs, we have, in a sense, not gained much information on the temporal, spectral and/or spatial signatures of AA. We therefore solve the given task i.e., the reduction/removal of VA, but much is still left to do on the subject of modeling AA.

Additionally, we do not take advantage of any apriori information on the generation and propagation mechanisms of AF. This is partly due to the fact that the scientific community is still unclear on the precise details thereof, but should this information be at our disposal, then it is certainly going to aid in the identification of the latent

components and improved removal of VA in AF EGMs. This goes hand in hand with knowledge of statistical properties. For example, if AA and VA were spatially mutually uncorrelated, then we could easily apply orthogonality as a constraint in the spatial mode of the tensor enhancing the quality of the decomposition.

FA-BSS indeterminacies (scaling and permutation) are admittedly dealt with in a fairly crude manner in this project. While it suffices to demonstrate the usefulness of MWCA, more advanced techniques like back-projection [36], [37] would definitely be preferable so as to obtain the correct sign, permutation and scaling of the extracted components.

Last but not least, the phase information of the AF EGM signals was ignored in the analysis since we are currently not aware its relevance. Perfunctory exploration of the phase component of the data indicated a level of randomness possibly correlated to the unpredictability of AF, and so we opted not to use it. However, it is conceivable that we were indeed in the wrong and phase information should be included in the proposed approach. Fortunately, all of the decomposition techniques presented in this report also allow for complex data as input, but a reinterpretation of the extracted factors/components would be inevitable.

All in all, AF analysis via tensor decompositions turns out to be a formidable but stimulating task that we believe holds a lot of promise, not only to effectively remove VA and other artifacts if present, but also to better characterize the nature of AF. Consequently, more suitable diagnostic and therapeutic tools to combat AF can be developed so as to alleviate its associated dangers and risks.

Appendix: Mathematics of Tensor Decompositions

<div align="right">

A

</div>
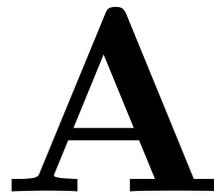
The following mathematical definitions apply to all higher-order arrays, although it is simpler to visualize and work with third-order arrays in some of the examples given for clarification purposes. A lot of the tensor definitions have been borrowed from the seminal works on multilinear algebra by DeLathauwer and Moor [27], and Kolda and Bader [3] respectively: "From Matrix to Tensor: Multilinear Algebra and Signal Processing" and "Tensor Decompositions and Applications".

Definiton 1: (Outerproduct) The outer product $\mathcal{A} \circ \mathcal{B}$ of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_P}$ and a tensor $\mathcal{B} \in \mathbb{R}^{J_1 \times J_2 \times ... \times J_Q}$ is defined by:

$$(\mathcal{A} \circ \mathcal{B})_{i_1, i_2, ..., i_P, j_1, j_2, ..., j_Q} \quad \overset{\text{def}}{=} \quad a_{i_1, i_2, ..., i_P} b_{j_1, j_2, ..., j_Q}, \tag{A.1}$$

for all values of the indices. The outer product generalizes expressions of the type $\mathbf{a}\mathbf{b}^{\mathbf{T}}$ where $\mathbf{a}$ and $\mathbf{b}$ are vectors.

Definition 2: (Scalar product) The scalar product $< \mathcal{A}, \mathcal{B} >$ of tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is defined as:

$$< \mathcal{A}, \mathcal{B} > \quad \overset{\text{def}}{=} \quad \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} b_{i_1 i_2 ... i_N} a_{i_1 . i_2 ... i_N} \tag{A.2}$$

Definition 3: (Orthogonality) Tensors whose scalar product equals 0 are mutually orthogonal.

Definition 4: (Frobenius-norm) The Frobenius norm of a tensor $\mathcal{A}$ is defined as:

$$||\mathcal{A}|| = \sqrt{< \mathcal{A}, \mathcal{A} >} \tag{A.3}$$

Analogous to the matrix Frobenius norm denoted as $||\mathbf{A}||$ for a matrix $\mathbf{A}$, the norm of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is the square root of the sum of the squares of all its

elements, i.e.,

$$||\mathcal{X}|| = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N}^2}. \tag{A.4}$$

The inner product of two same-sized tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is the sum of the products of their entries, i.e.,

$$< \mathcal{X}, \mathcal{Y} > = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N} y_{i_1 i_2 \ldots i_N}}. \tag{A.5}$$

Therefore, $< \mathcal{X}, \mathcal{X} > = ||\mathcal{X}||^2$.

Definition 5: The mode-n product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n \mathbf{U}$ is an $(I_1 \times I_2 \times \ldots \times I_{n-1} J_n \times I_{n+1} \ldots \times I_N)$-tensor defined by

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \ldots j_n \ldots i_N} = \sum_{i_n} a_{i_1 i_2 \ldots i_n \ldots i_N} u_{j_n i_n}, \tag{A.6}$$

for all index values.

Using this notation, the matrix product $\mathbf{A} = \mathbf{U}^{(1)} \mathbf{B} \mathbf{U}^{(2)\mathbf{T}}$ takes the form of the symmetric expression $\mathbf{A} = \mathbf{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}$, reflecting the fact that $\mathbf{U}^{(2)}$ acts in exactly the same way on the columns of $\mathbf{B}$ as $\mathbf{U}^{(1)}$ does for the rows. The mode-n product allows us to express the effect of a basis transformation in $\mathbb{R}^{I_n}$ on the tensor $\mathcal{A}$.

For distinct modes in a series of multiplications, the order of the multiplication is irrelevant, i.e.,

$$\mathcal{X} \times_m \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_m \mathbf{B} \times_n \mathbf{A} \quad (m \neq n). \tag{A.7}$$

If the modes are the same, then

$$\mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n \mathbf{BA}. \tag{A.8}$$

The n-mode (vector) product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{I_n}$ is denoted by $\mathcal{X} \bar{\times}_n \mathbf{v}$. The result is of order $N-1$, i.e., the size is $I_1 \times \ldots \times I_{n-1} \times$

$I_{n+1} \times \ldots \times I_N$ i.e. I compute the inner product of each mode-n fiber with the vector **v**. Elementwise,

$$\mathcal{X} \bar{\times}_n \quad v_{i_1 \ldots i_{n-1} i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} v_{i_n}. \tag{A.9}$$

For mode-n vector multiplication, precedence matters because the order of the intermediate results changes i.e.:

$$\mathcal{X} \bar{\times}_m \mathbf{a} \bar{\times}_n \mathbf{b} = (\mathcal{X} \bar{\times}_m \mathbf{a}) \bar{\times}_{n-1} \mathbf{b} = (\mathcal{X} \bar{\times}_n \mathbf{b}) \bar{\times}_m \mathbf{a}, \tag{A.10}$$

for $m < n$.

Definition 6: (Mode-n rank) The mode-n vectors of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ are the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying the index i_n and keeping the other indices fixed. The mode-n rank of $\mathcal{A}$, denoted by $R_n = rank_n(\mathcal{A})$, is the dimension of the vector space generated by the mode-n vectors.

Definition 7: (Rank-1 tensor) An Nth-roder tensor $\mathcal{A}$ has rank 1 when it equals the outer product of N vectors $U^{(1)}$, $U^{(2)}$, ..., $U^{(N)}$:

$$\mathcal{A} = U^{(1)} \circ U^{(2)} \circ \ldots \circ U^{(N)} \tag{A.11}$$

Definition 8: (Rank) The rank or an arbitrary Nth-order tensor $\mathcal{A}$, denoted by $R = rank(A)$, is the minimal number of rank-1 tensors that yield $\mathcal{A}$ in a linear combination.

Definition 9: (Symmetry) A tensor is cubical if every mode is the same size, i.e. $\mathcal{X} \in \mathbb{R}^{I \times I \times I \times \ldots \times I}$. A cubical tensor is called supersymmetric if its elements remain constant under any permutation of the indices. For instance, a three-mode tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times I}$ is supersymmetric if

$$x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji}, \tag{A.12}$$

for all $i, j, k = 1, ..., I$.

Tensors can be (partially) symmetric in two or more modes as well. For example, a three-mode tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$ is symmetric in modes one and two if all its frontal

slices are symmetric, i.e.:

$$\mathbf{X_k} = \mathbf{X_k}^T \tag{A.13}$$

for all $k = 1, ..., K$.

Definition 10: (Diagonal tensors) A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is diagonal if $x_{i_1 i_2 ... i_N} \neq 0$ only if $i_1 = i_2 = \ldots = i_N$. Figure A.1 shows a cubical tensor with ones along the superdiagonal.
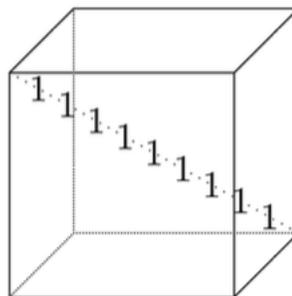
Figure A.1: Cubical 3-way tensor with ones along the superdiagonal[3].

Definition 11: (Kronecker, Khatri-Rao and Hadamard Products) The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times J}$ is denoted by $A \otimes B$. This results in a matrix of size $(IK) \times (JL)$ and defined by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \ldots a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ldots a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{12}\mathbf{B} & \ldots a_{IJ}\mathbf{B} \end{bmatrix} = [\mathbf{a_1} \otimes \mathbf{b_1} \mathbf{a_1} \otimes \mathbf{b_2} \mathbf{a_1} \otimes \mathbf{b_3} \ldots \mathbf{a_J} \otimes \mathbf{b_{L-1}} \mathbf{a_J} \otimes \mathbf{b_L}].$$

$$\tag{A.14}$$

The Khatri-Rao product is the "matching columnwise" Kronecker product. Given matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \cdot \mathbf{B}$. The result is a matrix of size $(IJ) \times K$ defined by:

$$\mathbf{A} \cdot \mathbf{B} = [\mathbf{a_1} \otimes \mathbf{b_1} \quad \mathbf{a_2} \otimes \mathbf{b_2} \quad \ldots \quad \mathbf{a_K} \otimes \mathbf{b_K}]. \tag{A.15}$$

If $\mathbf{a}$ and $\mathbf{b}$ are vectors, the the Khatri-Rao and Kronecker products are identical, i.e. $\mathbf{a} \otimes \mathbf{b} = \mathbf{a} \cdot \mathbf{b}$.

The Hadamard product is the elementwise matrix product. Given matrices $\mathbf{A}$ and $\mathbf{B}$, both of size $I \times J$, their Hadamard product is denoted by $\mathbf{A} * \mathbf{B}$. The result is also of size $I \times J$ and defined by:

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \dots a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \dots a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \dots a_{IJ}b_{IJ} \end{bmatrix}. \tag{A.16}$$

These matrix products have the following properties:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD},$$
$$(\mathbf{A} \otimes \mathbf{B}) = \mathbf{A}^\dagger \otimes \mathbf{B}^\dagger,$$
$$(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) = (\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C}))),$$
$$(\mathbf{A} \cdot \mathbf{B})^{\mathbf{T}}(\mathbf{A} \cdot \mathbf{B}) = \mathbf{A}^{\mathbf{T}}\mathbf{A} * \mathbf{B}^{\mathbf{T}}\mathbf{B},$$
$$(\mathbf{A} \cdot \mathbf{B})^\dagger = ((\mathbf{A}^{\mathbf{T}}\mathbf{A}) * (\mathbf{B}^{\mathbf{T}}\mathbf{B}))^\dagger(\mathbf{A} \cdot \mathbf{B})^{\mathbf{T}}.$$

$\mathbf{A}^\dagger$ denotes the Moore-Penrose pseudoinverse of $\mathbf{A}$.

Definition 12: (Matricization and vectorization) Matricization, also known as unfolding or flattening, is the process of reordering the elements of an N-way array into a matrix. For instance, a $2 \times 3 \times 4$ tensor can be arranged as a 6×4 matrix or a 3×8 matrix. It is also possible to vectorize a tensor, but this will not be further discussed since it is quite straightforward. The ordering is not important as long as it is consistent.

The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers to be the columns of the resulting matrix. Tensor element $(i_1, i_2, ..., i_N)$ maps to matrix element (i_n, j), where:

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1)J_k, \tag{A.17}$$

with

$$J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m.$$
(A.18)

Definition 13: (Tensorization) Tensorization is the formation of a tensor from lower-dimensional data. This can be accomplished by:

- Rearrangement of lower-dimensional data structures. For example, a one-way exponential signal $x(k) = az^k$ can be rearranged into a rank-1 Hankel matrix or a Hankel tensor [36][8]:

$$\mathbf{H} = \begin{pmatrix} x(0) & x(1) & x(2) \dots \\ x(1) & x(2) & x(3) \dots \\ x(2) & x(3) & x(4) \dots \\ \vdots & \vdots & \vdots \end{pmatrix} = a\mathbf{a} \circ \mathbf{b},$$
(A.19)

  where $\mathbf{b} = [1, z, z^2, ...]^T$.

- Mathematical construction. For example, a $(channel \times time)$ data matrix can be transformed into a $(channel \times time \times frequency)$ or $(channel \times time \times scale)$ tensor via time-frequency or wavelet representations.

Definition 14: (Super-symmetry) Moments and cumulants are symmetric in their arguments i.e.:

$$(\mathcal{M}_X^N)_{i_1 i_2 \dots i_N} = (\mathcal{M}_X^N)_{P(i_1 i_2 \dots i_N)}$$
(A.20)

$$(\mathcal{C}_X^N)_{i_1 i_2 \dots i_N} = (\mathcal{C}_X^N)_{P(i_1 i_2 \dots i_N)}$$
(A.21)

in which P is an arbitrary permutation of the indices.

Appendix: MATLAB Code

B

The MATLAB code used to analyze the epicardial electrogram datasets is presented below. MATLAB software is proprietary but can be easily acquired from the Mathworks Inc. website with the proper licenses. The Tensorlab toolbox is necessary to implement the tensor decompositions discussed in this thesis report, and can be downloaded for free from the Tensorlab website [38], whose url can be found in the bibliography.

B.1 Data Preprocessing

```matlab
1  function [rowsEGM,ecg,val] = loadEGM_noFigs(x)
   % Loads the given electrogram (EGM) datasets without plotting the ECG
       or
   % any channel data.
   % x is the kind of EGM dataset i.e. in sinus rhythm (SR) or in atrial
   % fibrillation (AF).
6
   % Load the EGM datasets
   egm_datasets;
   egm_datasets_AF;

11 % Load the requested EGM dataset
   if strcmp(x,'EGM_SR_BB0')
           EGM = EGM_SR_BB0;
           val = 1;
   elseif strcmp(x,'EGM_SR_LA2')
16         EGM = EGM_SR_LA2;
           val = 1;
   elseif strcmp(x,'EGM_SR_LA3')
           EGM = EGM_SR_LA3;
           val = 1;
21 elseif strcmp(x,'EGM_SR_PVL1')
           EGM = EGM_SR_PVL1;
           val = 1;
   elseif strcmp(x,'EGM_SR_PVR1')
```

```
                EGM = EGM_SR_PVR1 ;
26              val  = 1;
    elseif strcmp(x,'EGM_SR_RA1')
                EGM = EGM_SR_RA1 ;
                val  = 1;
    elseif strcmp(x,'EGM_SR_RA2')
31              EGM = EGM_SR_RA2 ;
                val  = 1;
    elseif strcmp(x,'EGM_SR_RA3')
                EGM = EGM_SR_RA3 ;
                val  = 1;
36  elseif strcmp(x,'EGM_SR_RA4')
                EGM = EGM_SR_RA4 ;
                val  = 1;
    elseif strcmp(x,'EGM_SR_RA5')
                EGM = EGM_SR_RA5 ;
41              val  = 1;
    elseif strcmp(x,'EGM_AF_BB0')
                EGM = EGM_AF_BB0 ;
                val  = 1;
    elseif strcmp(x,'EGM_AF_LA2')
46              EGM = EGM_AF_LA2 ;
                val  = 1;
    elseif strcmp(x,'EGM_AF_LA3')
                EGM = EGM_AF_LA3 ;
                val  = 1;
51  elseif strcmp(x,'EGM_AF_PVL1')
                EGM = EGM_AF_PVL1 ;
                val  = 1;
    elseif strcmp(x,'EGM_AF_PVR1')
                EGM = EGM_AF_PVR1 ;
56              val  = 1;
    elseif strcmp(x,'EGM_AF_RA1')
                EGM = EGM_AF_RA1 ;
                val  = 1;
    elseif strcmp(x,'EGM_AF_RA2')
61              EGM = EGM_AF_RA2 ;
                val  = 1;
    elseif strcmp(x,'EGM_AF_RA3')
                EGM = EGM_AF_RA3 ;
                val  = 1;
66  elseif strcmp(x,'EGM_AF_RA4')
                EGM = EGM_AF_RA4 ;
                val  = 1;
    elseif strcmp(x,'EGM_AF_RA5')
                EGM = EGM_AF_RA5 ;
```

82

```matlab
71              val = 1;
    else
            disp('Pseudoreal EGM Data')
            rowsEGM = 0;
            val = 0;
76          ecg = 0;
            return;
    end

    if EGM ~= 0
81      EGM(8,:) = []; % Remove the reference signal from the dataset
        EGM(148,:) = [];
        EGM(183,:) = [];
        EGM(184,:) = [];
        EGM(188:end,:) = []; % Remove the empty channels from the dataset
86

        % Basic preprocessing of the EGM signals
        % Remove the mean of the signal
        EGM = EGM - repmat(mean(EGM,2),1,size(EGM,2));
        % Variance normalization
91      for i = 1:size(EGM,1)
            EGM(i,:) = sqrt(size(EGM(i,:),2))*EGM(i,:)/norm(EGM(i,:));
        end
        % Amplitude normalization
        EGM = repmat(1./(max(abs(EGM),[],2)),1,size(EGM,2)).*EGM;
96      %
    end

    % Construct the EGM dataset on a row-by-row basis
    nsamples = size(EGM,2);
101 ecg = EGM(1,:);
    rowsEGM = zeros(23,8,nsamples);

    row1 = EGM(2:7,:);
    row1 = vertcat(row1,zeros(2,nsamples));
106 rowsEGM(1,:,:) = row1;

    for i = 2:18
        rowsEGM(i,:,:) = EGM(8*(i-1):8*(i-1)+7,:);
    end
111

    row18 = EGM(144:150,:);
    row18 = vertcat(row18,zeros(1,nsamples));
    rowsEGM(18,:,:) = row18;

116 for i = 20:23
```

```
        srt = 8*(i-1)-1;
        rowsEGM(i,:,:) = EGM(srt:srt+7,:);
      end

121  row23 = EGM(183:end,:);
     row23 = vertcat(row23,zeros(3,nsamples));
     rowsEGM(23,:,:) = row23;
     end
```

B.2   Pseudoreal EGM Signals

```
 1  % Generate artificial pure AA
    function pureAA = artificial_pureAA(dataSR,dataAF,row)
    % MATLAB function to create an pure atrial activity (AA) signal from
        signal in atrial
    % fibrillation (AF).

 6  [~,ecgSR,~] = loadEGM_noFigs(dataSR);
    [EGM,ecg,~] = loadEGM_noFigs(dataAF);
    EGM = squeeze(EGM(row,:,1:size(ecgSR,2)));
    ecg = ecg(1:size(ecgSR,2));

11  nchan = size(EGM,1);
    % nsamples = size(EGM,2);
    fs = 1000; % Sampling frequency is 1KHz
    time = 0:1/fs:(size(EGM,2)/fs)-(1/fs);

16  ecg = abs(ecg).^2;
    [~,locs] = findpeaks(ecg,time,'MinPeakHeight',0.4,'MinPeakDistance'
        ,0.150);

    % Isolate pure atrial signal from the EGM recordings in AF
    % Algorithm: 1) Find the peak locations of the QRST complex in the ECG
21  % signal 2) Isolate segments of ~ 90 seconds by taking 45 samples to
        the
    % left of the peak and 44 samples to the right of the peak for each
        channel
    % 3) For the segments that have been left over, decreasingly weigh the
        AA
    % segments to the left of the VA and increasingly weigh the segments to
        the
    % right of the VA and sum them up to create pseudoreal pure AA
        recordings
26  % 4) Concatenate the segments 5) The resulting signals are pure AA
    % recordings
```

```matlab
     % AA activity lasts approx. 0.09 sec but include delay at the AV node
         of
     % 0.07 sec giving a total of 0.16 sec = 160 samples
31   % VA activity lasts approx. 0.09 sec = 90 samples
     pureAA = EGM;
     AA_preBgn = zeros(1,60);
     AA_postBgn = zeros(1,60);
     AA_end = zeros(1,60);
36   for i = 1:nchan
         AA_preBgn(1:30) = EGM(i,(floor(fs*locs(1))-(40)):(floor(fs*locs(1))
             -(11)));
         AA_preBgn(31:60) = AA_preBgn(1:30);
         AA_postBgn(1:30) = EGM(i,(floor(fs*locs(1))+(21)):(floor(fs*locs(1)
             )+(50)));
         AA_postBgn(31:60) = AA_postBgn(1:30);
41       AA_preBgn = linspace(1,1/length(AA_preBgn),length(AA_preBgn)).*
             AA_preBgn; % Linearly weight the pre-AA signal
         AA_postBgn = linspace(1/length(AA_postBgn),1,length(AA_postBgn)).*
             AA_postBgn; % Linearly weight the post-AA signal
         AA_bgn = padarray((AA_preBgn+AA_postBgn),[0 (60-length(AA_postBgn))
             /2]);
         AA_bgn = hamming(length(AA_bgn))'.*AA_bgn;
         pureAA(i,(floor(fs*locs(1))-30:floor(fs*locs(1))+29)) = zeros(1,60)
             ;
46       pureAA(i,(floor(fs*locs(1))-30:floor(fs*locs(1))+29)) = pureAA(i,(
             floor(fs*locs(1))-30:floor(fs*locs(1))+29))+AA_bgn;
         for j = 2:length(locs)-1
             AA_pre = EGM(i,(floor(fs*locs(j))-(80)):(floor(fs*locs(j))-(21)
                 ));
             AA_post = EGM(i,(floor(fs*locs(j))+(21)):(floor(fs*locs(j))
                 +(80)));
             AA_pre = linspace(1,1/length(AA_pre),length(AA_pre)).*AA_pre; %
                  Linearly weight the pre-AA signal
51           AA_post = linspace(1/length(AA_post),1,length(AA_post)).*
                 AA_post; % Linearly weight the post-AA signal
             AA_inter = padarray((AA_pre+AA_post),[0 (60-length(AA_post))
                 /2]);
             AA_inter = hamming(length(AA_inter))'.*AA_inter;
             pureAA(i,(floor(fs*locs(j))-30:floor(fs*locs(j))+29)) = zeros
                 (1,60);
             pureAA(i,(floor(fs*locs(j))-30:floor(fs*locs(j))+29)) = pureAA(
                 i,(floor(fs*locs(j))-30:floor(fs*locs(j))+29))+AA_inter;
56       end
         AA_end(1:30) = EGM(i,(floor(fs*locs(j+1))+(21)):(floor(fs*locs(j+1)
             )+(50)));
```

```matlab
        AA_end(31:60) = AA_end(1:30);
        AA_end = padarray(AA_end,[0 (60-length(AA_end))/2]);
        AA_end = hamming(length(AA_end))'.*AA_end;
61      pureAA(i,(floor(fs*locs(j+1))-30:floor(fs*locs(j+1))+29)) = zeros
            (1,60);
        pureAA(i,(floor(fs*locs(j+1))-30:floor(fs*locs(j+1))+29)) = pureAA(
            i,(floor(fs*locs(j+1))-30:floor(fs*locs(j+1))+29))+AA_end;
    end

    % Generate artificial pure VA
66  function pureVA = artificial_pureVA(dataSR,dataAF,row)
    % MATLAB function to create an pure ventricular activity (AA) signal
        from signal in
    % sinus rhythm.

    % Isolate pure ventricular signal from the EGM recordings in NSR and/or
        the
71  % signal model developed by Bahareh and/or in the BSS_Hendriks_3 paper
    % Algorithm: 1) Find the peaks of the QRS complex in the ECG signal 2)
        Do the exact same procedure as for
    % creating the AA activity except use the NSR signals to isolate pure
        VA
    % and interpolate the data by concatenating zeros before and after the
        VA
    % and using a Hanning window 3) Ensure that the signal lengths are the
        same
76  % for both VA and AA.

    [EGM_SR,ecgSR,~] = loadEGM_noFigs(dataSR);
    [EGM_AF,ecgAF,~] = loadEGM_noFigs(dataAF);
    EGM_AF = squeeze(EGM_AF(row,:,1:size(EGM_SR,3)));
81  EGM_SR = squeeze(EGM_SR(row,:,1:size(EGM_SR,3)));
    ecgAF = ecgAF(1:size(EGM_SR,2));

    nchan = size(EGM_AF,1);
    nsamples = size(EGM_AF,2);
86  fs = 1000; % Sampling frequency is 1KHz
    time = 0:1/fs:(size(EGM_SR,2)/fs)-(1/fs);

    % Find the peak locations in NSR
    ecgSR = abs(ecgSR).^2;
91  [~,locsSR] = findpeaks(ecgSR,time,'MinPeakHeight',0.3,'MinPeakDistance'
        ,0.150);
    % Find the peak locations in AF
    ecgAF = abs(ecgAF).^2;
    [~,locsAF] = findpeaks(ecgAF,time,'MinPeakHeight',0.3,'MinPeakDistance'
```

```matlab
                    ,0.150);

 96  pureVA = zeros(nchan,nsamples);
     for i = 1:nchan
         maxAmp = zeros(1,length(locsAF));
         maxAmp(1) = min(EGM_AF(i,floor(fs*locsAF(1))-(20):floor(fs*locsAF
             (1))+(19)));
         VA = EGM_SR(i,floor(fs*locsSR(4))-(60):floor(fs*locsSR(4))+(59));
101      VA = abs(maxAmp(1))*(1/max(abs(VA)))*VA;
         VA = padarray(VA,[0  (120-length(VA))/2]);
         VA = hamming(length(VA))'.*VA;
         pureVA(i,floor(fs*locsAF(1))-(45):floor(fs*locsAF(1))+(74)) =
             pureVA(i,floor(fs*locsAF(1))-(45):floor(fs*locsAF(1))+(74)) + VA
             ;
         for j = 2:length(locsAF)-1
106          maxAmp(j) = min(EGM_AF(i,floor(fs*locsAF(j))-(20):floor(fs*
                 locsAF(j))+(19)));
             VA = EGM_SR(i,floor(fs*locsSR(4))-(65):floor(fs*locsSR(4))+(74)
                 );
             VA = abs(maxAmp(j))*(1/max(abs(VA)))*VA;
             VA = padarray(VA,[0  (140-length(VA))/2]);
             VA = hamming(length(VA))'.*VA;
111          pureVA(i,floor(fs*locsAF(j))-(65):floor(fs*locsAF(j))+(74)) =
                 pureVA(i,floor(fs*locsAF(j))-(65):floor(fs*locsAF(j))+(74))
                 + VA;
         end
         maxAmp(end) = min(EGM_AF(i,floor(fs*locsAF(end))-(20):floor(fs*
             locsAF(end))+(19)));
         VA = EGM_SR(i,floor(fs*locsSR(4))-(60):floor(fs*locsSR(4))+(59));
         VA = abs(maxAmp(1))*(1/max(abs(VA)))*VA;
116      VA = padarray(VA,[0  (120-length(VA))/2]);
         VA = hamming(length(VA))'.*VA;
         pureVA(i,floor(fs*locsAF(end))-(69):floor(fs*locsAF(end))+(50)) =
             pureVA(i,floor(fs*locsAF(end))-(69):floor(fs*locsAF(end))+(50))
             + VA;
     end
     end
121
     % Generate pseudoreal EGM signals
     function [mixedAF,pureAA,pureVA] = artificial_superimposed_EGM(SR_Data,
         AF_Data,row)
     % MATLAB script to generate artificial (not synthetic) EGM signals.
     % This is done by isolating a pure atrial activity (AA) signal and
126  % superimposing a pure ventricular activity (VA) signal to create the
     % atrial fibrillation (AF) signal contaminated by far ventricular
         signal.
```

```matlab
    % This is useful for determining the performance of our algorithm since
        we
    % can measure how well the ventricular artifacts have been removed from
        the
131 % recorded unipolar epicardial AF signals.

    % Pseudoreal signals in AF
    pureAA = artificial_pureAA(SR_Data,AF_Data,row);
    pureAA = pureAA(:,1:4999);
136 pureVA = artificial_pureVA(SR_Data,AF_Data,row);

    % Load EGM data in atrial fibrillation
    [EGM,ecg,~] = loadEGM_noFigs(AF_Data);
    % [EGM,~,~] = loadEGM_noFigs(AF_Data);
141 EGM = squeeze(EGM(row,:,1:size(pureVA,2)));
    ecg = ecg(1:size(pureVA,2));
    fs = 1000; % Sampling frequency is 1KHz
    time = 0:1/fs:(size(EGM,2)/fs)-(1/fs);

146 % Superimpose the atrial and ventricular signals to create an
        artificial
    % signal in AF using the appropriate mixing model.
    mixedAF = pureAA + pureVA;
```

B.3 Average Beat Subtraction

```matlab
 1 function [EGM,AA_estimate] = EGM_ABS(AF_Data,ecg,nRow)
    % MATLAB script to implement average beat subtraction (ABS) as a
        benchmark
    % against which to measure the performance of the BTD algorithm.

    % The ABS algorithm
 6 % 1) Load the AF EGM datasets on a row-by-row basis
    % 2) Align the peaks of the VA to the QRS peaks in the ECG recording
    % 3) Slice out the VA in the AF EGM recordings
    % 4) Find an average template (either using SVD or find the mean)
    % 5) Subtract the template from the AF EGM recordings to remove the VA
11 % 6) Evaluate the performance of the algorithm using the performance
        metrics

    % For the normal operation of the function
    [EGM,~,val] = loadEGM_noFigs(AF_Data);
    row = nRow;
16
```

```matlab
    if val == 1
        EGM = squeeze(EGM(row,:,:));
    elseif val == 0
        EGM = squeeze(AF_Data(row,:,:));
21  end

    nchan = size(EGM,1);
    nsamples = size(EGM,2);

26  fs = 1000; % Sampling frequency is 1KHz
    time = 0:1/fs:(size(EGM,2)/fs)-(1/fs);
    % Align the data sets based on the peaks of the QRS complex in the ECG
    % Find the peak locations in sinus rhythm (SR)
    ecg = abs(ecg).^2;
31  [~,locs_QRS] = findpeaks(ecg(1:size(EGM,2)),time,'MinPeakHeight',0.4,'
        MinPeakDistance',0.150);


    % Aign and segment out the beats
    nbeats = length(locs_QRS);
36  locs_VA = zeros(size(EGM,1),nbeats);
    len_nbeat = 120;
    template_VA = zeros(size(EGM,1),nbeats,len_nbeat);
    for i = 1:nchan
        for j = 1
41          peak_VA = min(EGM(i,floor(fs*locs_QRS(j))-20:floor(fs*locs_QRS(
                j))+20));
            peakVA_loc = find(EGM(i,floor(fs*locs_QRS(j))-20:floor(fs*
                locs_QRS(j))+20) == peak_VA);
            locs_VA(i,j) = floor(fs*locs_QRS(j))-20 + peakVA_loc(1);
            template_VA(i,j,:) = EGM(i,locs_VA(i,j)-40:locs_VA(i,j)+79);
        end
46      for j = 2:nbeats-1
            peak_VA = min(EGM(i,floor(fs*locs_QRS(j))-20:floor(fs*locs_QRS(
                j))+20));
            peakVA_loc = find(EGM(i,floor(fs*locs_QRS(j))-20:floor(fs*
                locs_QRS(j))+20) == peak_VA);
            locs_VA(i,j) = floor(fs*locs_QRS(j))-20 + peakVA_loc(1);
            template_VA(i,j,:) = EGM(i,locs_VA(i,j)-59:locs_VA(i,j)+60);
51      end
        for j = nbeats
            peak_VA = min(EGM(i,floor(fs*locs_QRS(j))-20:floor(fs*locs_QRS(
                j))+20));
            peakVA_loc = find(EGM(i,floor(fs*locs_QRS(j))-20:floor(fs*
                locs_QRS(j))+20) == peak_VA);
            locs_VA(i,j) = floor(fs*locs_QRS(j))-20 + peakVA_loc(1);
```

```
56              template_VA(i,j,:) = EGM(i,locs_VA(i,j)-69:locs_VA(i,j)+50);
            end
        end

        avgTemplate = zeros(nbeats,len_nbeat);
61      for i = 1:nbeats
            avgTemplate(i,:) = mean(squeeze(template_VA(:,i,:)),1);
        end

        avg2Subtract = zeros(nchan,nsamples);
66
        % Subtract the average beat template from the AF EGM
        AA_estimate = zeros(nchan,nsamples);
        for i = 1:nchan
            for j = 1
71              avg2Subtract(i,locs_VA(i,j)-40:locs_VA(i,j)+79) = avg2Subtract(
                    i,locs_VA(i,j)-40:locs_VA(i,j)+79)+avgTemplate(j,:);
            end
            for j = 2:nbeats-1
                avg2Subtract(i,locs_VA(i,j)-59:locs_VA(i,j)+60) = avg2Subtract(
                    i,locs_VA(i,j)-59:locs_VA(i,j)+60)+avgTemplate(j,:);
            end
76          for j = nbeats
                avg2Subtract(i,locs_VA(i,j)-69:locs_VA(i,j)+50) = avg2Subtract(
                    i,locs_VA(i,j)-69:locs_VA(i,j)+50)+avgTemplate(j,:);
            end
            AA_estimate(i,:) = EGM(i,:) - avg2Subtract(i,:);
        end
```

B.4   Block Term Decomposition

```
    function [EGM,X_atrTime,X_vtrTime,relerr]= EGM_MLRD_AF_Temporal(AF_Data
        ,ecg,nRow,sigPick)
    % MATLAB function to compute the low multilinear rank decomposition (
        BTD) of the EGM
    % datasets in atrial using the Tensorlab framework, and to obtain the
    % atrial signal estimates via temporal subtraction.
 5
    % Load AF EGM datasets
    [EGM,~,val] = loadEGM_noFigs(AF_Data);
    row = nRow;

10  if val == 1
        EGM = squeeze(EGM(row,:,:));
    elseif val == 0
```

```
        EGM = squeeze(AF_Data(row,:,:));
    end
15
    nchan = size(EGM,1);
    fs = 1000; % Sampling frequency is 1KHz
    time = 0:1/fs:(size(EGM,2)/fs)-(1/fs);
    % Align the data sets based on the peaks of the QRS complex in the ECG
20  % Find the peak locations in sinus rhythm (NSR)
    ecg = abs(ecg).^2;
    [~,locs] = findpeaks(ecg(1:size(EGM,2)),time,'MinPeakHeight',0.4,'
        MinPeakDistance',0.150);

    % Calculate the time-frequency representation using STFT
25  fs = 1000;
    nfft = 256;
    nhop = 16;
    nwin = 32;

30  % Calculate the STFT of the entire EGM dataset
    [egm_stft_mag,egm_stft_phi] = EGM_STFT(EGM,fs,nfft,nhop,nwin);

    % Absolute value of STFTs (squared)
    egm_stft_mag = egm_stft_mag(1:(nfft/2+1),:,:).^2;
35
    stft_permute = permute(egm_stft_mag,[1 2 3]);

    % Calculate the multilinear low rank representation using the BTD
    % formulation
40
    % Begin of data reduction using LMRA
    options.Display = true;                  % Show progress on the command
        line.
    options.Initialization = @lmlra_rnd;     % Select pseudorandom
        initialization.
    options.Algorithm = @lmlra_minf;         % Select NLS as the main
        algorithm.
45  options.AlgorithmOptions.TolFun = 1e-5; % Set stop criteria.
    options.AlgorithmOptions.TolX   = 1e-5;
    [U_MLRA,S_MLRA] = lmlra(stft_permute,[5,10,2],options);
    stft_permute = lmlragen(U_MLRA,S_MLRA);
    % End of data reduction using LMRA
50
    L = [5 5];
    disp(L)
    U0 = ll1_rnd(size(stft_permute),L,'OutputFormat','btd'); % Generate
        random tensor to initialize the decomposition
```

```matlab
    R = length(L); % Number of tensors
55
    [Uhat,~] = ll1_nls(stft_permute,U0,L);

    relerr = frob(ll1res(stft_permute,Uhat))/frob(stft_permute);% Uhat in
        BTD format

60  T = zeros(size(stft_permute,1),size(stft_permute,2),size(stft_permute
        ,3),R);
    for i = 1:R
        T(:,:,:,i) = btdgen(Uhat(i));
    end

65  sigKurt = zeros(R,nchan,1);
    for i = 1:R
    % Reconstruct each basis as a separate source
    for j = 1:nchan
        C = squeeze(T(:,:,:,i));
70      XmagHat = squeeze(C(:,:,j));
        % Create upper half of frequency before ISTFT
        XmagHat = [XmagHat; conj(XmagHat(end-1:-1:2,:))];
        XmagHatR(i,j,:,:) = XmagHat;
        % Multiply with phase
75      XHat = XmagHat.*exp(1i*egm_stft_phi(:,:,j));
        xhatR(:,i,j) = real(invmyspectrogram(XHat,nhop))';
        xhat(:,j) = real(invmyspectrogram(XHat,nhop))';
        sigKurt(i,j,:) = kurtosis(squeeze(xhat(:,j)));
    end
80
    end


    maxKurt = 170*ones(size(sigKurt));
85  minKurt = 5*ones(size(sigKurt));

    cMid = 0;
    cmptStd = zeros(R,1);
    for i = 1:R
90      if  sum((sigKurt(i,:) > minKurt(i,:)) & (sigKurt(i,:) < maxKurt(i
            ,:))) > nchan-1
            cmptStd(i,:) = std(sigKurt(i,:));
            cMid = cMid+1;
        else
            cmptStd(i,:) = 1000;
95      end
    end
```

92

```matlab
    if cMid > 1
        if strcmp(sigPick,'min')
100         cmptIdx = find(cmptStd == min(cmptStd));
        else
            cmptIdx = find(cmptStd == max(cmptStd));
        end
    else
105     cmptIdx = find(cmptStd == min(cmptStd));
    end

    X_NewTime = zeros(size(EGM,2),nchan);
    X_vtrTime = zeros(size(EGM,2),nchan);
110
    % Subtract the ventricular activity in the temporal domain
    for j = 1:nchan
        for i = cmptIdx
            if  sum((sigKurt(i,:) > minKurt(i,:)) & (sigKurt(i,:) < maxKurt
                (i,:))) > nchan-1
115             scaling = max(abs(EGM(j,:)));
                X_Time = scaling*(1/max(abs(squeeze(xhatR(:,i,j))))).*
                    squeeze(xhatR(:,i,j));
                for k = 1
                    if max(X_Time(floor(fs*locs(k))-10:floor(fs*locs(k))
                        +10)) >  0.05
                        X_Time(floor(fs*locs(k))-45:floor(fs*locs(k))+74) =
                            -1*X_Time(floor(fs*locs(k))-45:floor(fs*locs(k)
                            )+74);
120             end
                end
                for k = 1
                    top = min(EGM(j,floor(fs*locs(k))-10:floor(fs*locs(k))
                        +10));
                    btm = min(X_Time(floor(fs*locs(k))-10:floor(fs*locs(k))
                        +10));
125             X_Time(floor(fs*locs(k))-45:floor(fs*locs(k))+74) = (
                        top/btm)*X_Time(floor(fs*locs(k))-45:floor(fs*locs(k
                        ))+74);
                end
                for k = 2:length(locs)-1
                    if max(X_Time(floor(fs*locs(k))-10:floor(fs*locs(k))
                        +10)) >  0.1
                        X_Time(floor(fs*locs(k))-70:floor(fs*locs(k))+69) =
                            -1*X_Time(floor(fs*locs(k))-70:floor(fs*locs(k)
                            )+69);
130             end
```

```
                    end
                    for k = 2:length(locs)−1
                        top = min(EGM(j,floor(fs*locs(k))−10:floor(fs*locs(k))
                            +10));
                        btm = min(X_Time(floor(fs*locs(k))−10:floor(fs*locs(k))
                            +10));
135                     X_Time(floor(fs*locs(k))−70:floor(fs*locs(k))+69) = (
                            top/btm)*X_Time(floor(fs*locs(k))−70:floor(fs*locs(k
                            ))+69);
                    end
                    for k = length(locs)
                        if max(X_Time(floor(fs*locs(k))−10:floor(fs*locs(k))
                            +10)) > 0.1
                            X_Time(floor(fs*locs(k))−69:floor(fs*locs(k))+50) =
                                −1*X_Time(floor(fs*locs(k))−69:floor(fs*locs(k)
                                )+50);
140                     end
                    end
                    for k = length(locs)
                        top = min(EGM(j,floor(fs*locs(k))−10:floor(fs*locs(k))
                            +10));
                        btm = min(X_Time(floor(fs*locs(k))−10:floor(fs*locs(k))
                            +10));
145                     X_Time(floor(fs*locs(k))−69:floor(fs*locs(k))+50) = (
                            top/btm)*X_Time(floor(fs*locs(k))−69:floor(fs*locs(k
                            ))+50);
                    end
                    X_Time = (1/max(abs(X_Time))).*X_Time;
                    X_vtrTime(:,j) = X_Time(1:length(EGM(j,:)));
                    X_NewTime(:,j) = EGM(j,:)' − X_vtrTime(:,j);
150         end
        end
    end

    X_NewTime = X_NewTime';
155 X_vtrTime = X_vtrTime';

    % Basic postprocessing of the extracted atrial signals
    % Remove the mean of the signal
    X_NewTime = X_NewTime − repmat(mean(X_NewTime,2),1,size(X_NewTime,2));
160 X_vtrTime = X_vtrTime − repmat(mean(X_vtrTime,2),1,size(X_vtrTime,2));
    % Variance normalization
    for i = 1:size(EGM,1)
        X_NewTime(i,:) = sqrt(size(X_NewTime(i,:),2))*X_NewTime(i,:)/norm(
            X_NewTime(i,:));
        X_vtrTime(i,:) = sqrt(size(X_vtrTime(i,:),2))*X_vtrTime(i,:)/norm(
```

94

```matlab
            X_vtrTime(i,:));
165  end
     % Amplitude normalization
     X_NewTime = repmat(1./(max(abs(X_NewTime),[],2)),1,size(X_NewTime,2)).*
         X_NewTime;
     X_vtrTime = repmat(1./(max(abs(X_vtrTime),[],2)),1,size(X_vtrTime,2)).*
         X_vtrTime;
     %
170  X_atrTime = X_NewTime;
```

B.5 Results Compilation

```matlab
     % MATLAB script to evaluate the performance of the BTD algorithm on
         real EGM signals
     % based on the performance measures.

 4   % Recorded EGM datasets
     SR_Data = {'EGM_SR_BB0';'EGM_SR_LA2';'EGM_SR_LA3';'EGM_SR_PVL1';'
         EGM_SR_PVR1';'EGM_SR_RA1';'EGM_SR_RA2';'EGM_SR_RA3';'EGM_SR_RA4';'
         EGM_SR_RA5'};
     AF_Data = {'EGM_AF_BB0';'EGM_AF_LA2';'EGM_AF_LA3';'EGM_AF_PVL1';'
         EGM_AF_PVR1';'EGM_AF_RA1';'EGM_AF_RA2';'EGM_AF_RA3';'EGM_AF_RA4';'
         EGM_AF_RA5'};


 9   nRows = 23; % Number of rows with recordings in the given datasets
     nCols = 8; % Number of columns with recordings in the given datasets
     nDsets = 10; % The total number of datasets in AF

     Excess_All = zeros(nDsets,nRows,nCols);
14   avgCh_VASR_All = zeros(nDsets,nRows,nCols);
     stdCh_VASR_All = zeros(nDsets,nRows,nCols);
     avgCh_VDR_All = zeros(nDsets,nRows,nCols);
     stdCh_VDR_All = zeros(nDsets,nRows,nCols);
     mixedDF_All = zeros(nDsets,nRows,nCols);
19   demixedDF_All = zeros(nDsets,nRows,nCols);
     mixedSpConc_All = zeros(nDsets,nRows,nCols);
     demixedSpConc_All = zeros(nDsets,nRows,nCols);
     relErr = zeros(nDsets,nRows);

24   % Evaluate the algorithm for all given datasets
     % for i = 1:length(AF_Data)
     for i = 1
         % Load the EGM dataset
         [~,ecgAF,~] = loadEGM_noFigs(AF_Data{i});
```

```
29
        % Evaluate the algorithm for all rows (can also change to do this
            for columns if appropriate)
        for j = 2:17
            row = j;
            % Perform BTD on EGM recordings in atrial fibrillation (AF)
34          [EGM,atr_Time,~,relerr] = EGM_MLRD_AF_Temporal(AF_Data{i},ecgAF
                ,row,'min');

            % Relative error between original tensor and decomposed tensor
            relErr(i,j) = relerr;

39          % VASR metric (done on the original EGM recording and the
                demixed signal)
            VASR_time = VASR(atr_Time,EGM,ecgAF);
            avgCh_VASR_All(i,j,:) = mean(VASR_time,2);
            stdCh_VASR_All(i,j,:) = std(VASR_time,[],2);

44
            % VDR metric (done on the original EGM recording and the
                demixed signal)
            VDR_time = VDR(atr_Time,EGM,ecgAF);
            avgCh_VDR_All(i,j,:) = mean(VDR_time,2);
            stdCh_VDR_All(i,j,:) = std(VDR_time,[],2);
49
            % Dominant frequency metric
            dFreqMixed = dominantFreq(EGM);
            dFreqTime = dominantFreq(atr_Time);
            demixedDF_All(i,j,:) = dFreqTime;
54          mixedDF_All(i,j,:) = dFreqMixed;

            % Spectral concentration metric
            spConcMixed = spectConc(EGM);
            spConcTime = spectConc(atr_Time);
59          demixedSpConc_All(i,j,:) = spConcTime;
            mixedSpConc_All(i,j,:) = spConcMixed;
        end
    end

64  dataOut = [squeeze(Excess_All(i,:,:))';squeeze(avgCh_VASR_All(i,:,:))';
        squeeze(stdCh_VASR_All(i,:,:))';squeeze(avgCh_VDR_All(i,:,:))';...
        squeeze(stdCh_VDR_All(i,:,:))';squeeze(mixedDF_All(i,:,:))';squeeze
            (demixedDF_All(i,:,:))';squeeze(mixedSpConc_All(i,:,:))';...
        squeeze(demixedSpConc_All(i,:,:))'];
```

96

Bibliography

[1] W. Li *et al.*, "Several insights into the preprocessing of electrograms in atrial fibrillation for dominant frequency analysis," *BioMedical Engineering OnLine*, vol. 15, April 2016.

[2] "Atrial fibrillation." http://www.mayoclinic.org/diseases-conditions/atrial-fibrillation/multimedia/img-20096412. Accessed: 2017-08-16.

[3] T. Kolda and B. Bader, "Tensor decompositions and applications," *Society for Industrial and Applied Mathematics*, vol. 51, pp. 455–500, 2009.

[4] "Review of data analysis methods for denoising and characterizing ecg." https://archive.cnx.org/contents/b72ca4e8-ecee-4a99-bb2c-6162d8880e08@1/review-of-data-analysis-methods-for-denoising-and-characterizing-ecg. Accessed: 2017-09-05.

[5] I. Tudosa, N. I. Adochiei, and R. Ciobotariu, "New aspects in ecg signal processing using adaptive filters," *Advanced Topics in Electrical Engineering*, July 2011.

[6] C. J. James and C. W. Hesse, "Independent component analysis for biomedical signals," *Physiological Measurement*, vol. 26, pp. R15–R39, December 2004.

[7] E. Acar *et al.*, "Multiway analysis of epilepsy tensors," *Bioinformatics*, vol. 23, pp. i10–i18, 2007.

[8] A. Cichoki *et al.*, "Tensor decompositions for signal processing applications," *IEEE Signal Processing Magazine*, vol. 15, pp. 145–163, February 2015.

[9] B. G. Zhou, Q. Zhao, and Y. Zhang, "Linked component analysis from matrices to high-order tensors: Applications to biomedical data," *Proceedings of the IEEE*, vol. 104, February 2016.

[10] F. Castells *et al.*, "Spatiotemporal blind source separation approach to atrial activity estimation in atrial tachyarrhythmias," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 2, pp. 258–267, 2005.

[11] J. J. Rieta and F. Hornero, "Comparative study of methods for ventricular activity cancellation in atrial electrograms of atrial fibrillation," *Physiological measurement*, vol. 28, no. 8, pp. 925–936, 2007.

[12] L. Sornmo *et al.*, "Analysis of atrial fibrillation: from electrocardiogram signal processing to clinical management," *Philosophical Transactions of the Royal Society*, vol. 367, pp. 235–253, 2008.

[13] H. Sih, "Measures of organization during atrial fibrillation," *Annali dell Istituto superiore di sanita*, vol. 37, pp. 361–369, January 2001.

[14] J. J. Rieta and R. Alcaraz, "Applications of signal analysis to atrial fibrillation," *InTech*, 2013.

[15] J. J. Rieta *et al.*, "Atrial activity extraction based on blind source separation as an alternative to qrst cancellation for atrial fibrillation analysis," *IEEE Computers in Cardiology*, vol. 27, pp. 69–72, 2000.

[16] J. J. Rieta *et al.*, "Atrial activity extraction for atrial fibrillation analysis using blind source separation," *IEEE Transaction on Biomedical Engineering*, vol. 51, pp. 1176–1186, July 2004.

[17] D. H. Brooks and R. S. MacLeod, "Electrical imaging of the heart," *IEEE Signal Processing Magazine*, pp. 24–42, January 1997.

[18] M. S. Pedersen *et al.*, *Convolutive Blind Source Separation Methods*, pp. 1065–1094. Springer Berlin Heidelberg, 2008.

[19] Y. J. Z. Y. X. Wang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[20] J. Slocum, E. Byrom, L. McCarthy, A. Sahakian, and S. Swiryn, "Computer detection of atrioventricular dissociation from surface electrocardiograms during wide qrs complex tachycardias," *Circulation*, vol. 72, pp. 1028–1036, 1985.

[21] R. Alcaraz and J. J. Rieta, "Adaptive singular value cancellation of ventricular activity in single-lead atrial fibrillation electrocardiograms," *Physiological Measurement*, vol. 29, pp. 1351–1369, October 2008.

[22] F. Castells *et al.*, "Principal component analysis in ecg signal processing," *Eurasip Journal on Advances in Signal Processing*, vol. 2007, 2007.

[23] L. D. Lathauwer, "A short introduction to tensor-based methods for factor analysis and blind source separation," *International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 558–563, September 2015.

[24] A. Hyvarinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, 2000.

[25] J. Kruskal, "Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Applications*, vol. 18, no. 2, pp. 95–138, 1977.

[26] I. Domanov and L. D. Lathauwer, "On the uniqueness of the canonical polyadic decomposition of third-order tensor - part i: Basic results and uniqueness of one factor matrix and part ii: Uniqueness of the overall decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 3, pp. 855–903, 2013.

[27] L. D. Lathauwer and B. D. Moor, "From matrix to tensor: Multilinear algebra and signal processing," *Mathematics in Signal Processing IV*, 1997.

[28] J. Hastad, "Tensor rank is np-complete," *J. Algorithms*, vol. 11, pp. 644–654, 1990.

[29] L. D. Lathauwer, "Block component analysis, a new concept for blind source separation," *Springer-Verlag Berlin Heidelberg*, pp. 1–8, 2012.

[30] A. M. Bruckstein *et al.*, "Afrom sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

[31] E. J. Candes *et al.*, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[32] L. D. Lathauwer, "Decompositions of a higher-order tensor in block terms - part iii: Alternating least squares algorithms," *Society for Industrial and Applied Mathematics*, vol. 30, no. 3, pp. 1067–1083, 2008.

[33] "Nmse." <https://rem.jrc.ec.europa.eu/RemWeb/atmes2/20b.htm>. Accessed: 2017-08-25.

[34] C. Schilling, "Analysis of atrial electrograms," *Karlsruhe Transactions on Biomedical Engineering*, vol. 17, May 2012.

[35] J. H. Goulart and P. Comon, "A novel non-iterative algorithm for lowmultilinear-rank tensor approximation," *Proceedings of the 25th European Signal Conference*, August 2017.

[36] N. Murata *et al.*, "An approach to blind source separation based on temporal structure of speech signals," *Neurocomputing*, vol. 41, no. 1-4, pp. 1–24, 2001.

[37] K. Matsuoka and S. Nakashima, "Minimal distortion principle for blind source separation," *Proc. International Conference on Independent Component Analysis and Blind Signal Separation*, pp. 722–727, 2001.

[38] L. Vervliet, N. and Debals, O. and Sorber, L. and Van Barel, M. and De Lathauwer, "Tensorlab 3.0," *Available online.*, 2016.