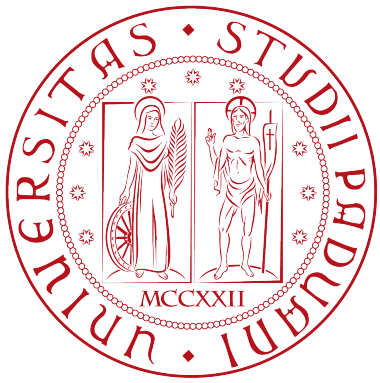
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Master's Degree in Mathematics



Nonnegative matrix factorization of phonocardiograms for heart rate detection

Supervisor:
Prof. Fabio Marcuzzi

Candidate:
Erik Chinellato
2021406

December 16, 2022

Academic Year 2021-2022

Contents

Abstract

Phonocardiograms (PCGs) are recordings of the sounds and murmurs made by the heart detected through specialized microphones placed on a patient's thorax. Alongside electrocardiograms (ECGs), they are a tool used in a medical environment to assess patients' conditions relative to their cardiac rhythm. Unlike the latter, in which, during each cardiac cycle, only one main peak can be detected within the voltage-over-time graph (the so called R wave), in PCGs two distinct peaks can be observed. These peaks are associated to the first and second heart sound (S1 and S2), generated by the closure of specific valves within the heart. In the following we shall refer to R waves and S1, S2 sounds as 'cardiac events'. In order to extrapolate the heart's activity from ECGs or PCGs, one needs to detect all cardiac events within the signal of choice. When it comes to ECGs, this process is relatively straightforward since only one R wave ought to be identified during each cycle. Moreover, such signals usually contain very low levels of noise, mainly caused by powerline interference, which can be easily removed using notch filters. On the other hand, event detection within PCGs is a quite challenging task. Indeed, not only do we need to detect two sounds each cycle, but also the signal itself is often severely contaminated by many different types of noise, such as the patient's movement, ambient sources, microphone movement or other body-related murmurs. As a consequence, the analysis of PCGs is often carried out with the aid of a synchronous ECG signal and requires a careful denoising of the audio file through digital filtering and signal envelope estimation.
The objective of the dissertation was to develop a method of detecting cardiac events within PCG signals that does not rely on the knowledge of ECGs. In particular, we achieved our goal by leveraging the modelling and learning

capabilities of Nonnegative Matrix Factorization (NMF) applied to the spectrogram of PCGs.

In Chapter 1, after providing a brief summary on the qualitative and quantitative properties of both ECGs and PCGs, we discuss how the techniques devised in the later chapters differ from the more traditional ones employed for cardiac event detection within PCGs.

In Chapter 2 we introduce the mathematical foundations of Nonnegative Matrix Factorization. In particular, we present $\beta-$divergences, analyze their main properties, and show how the most simple NMF optimization problem can be modified to induce certain regularizing features on the factors. Subsequently, we formulate algorithms to solve the NMF problem based on the majorization-minimization framework and focus our attention towards multiplicative update rules for the factors. Lastly, we provide some convergence results for the aforementioned algorithms.

In Chapter 3 we analyze the traditional ways of detecting cardiac events within synchronous pairs of ECGs and PCGs. This required the introduction of particular families of digital filters, namely notch, low-pass and zero-phase, as well as the general structure for a Window Search algorithm. We then start working out the preliminary details about our NMF-based approach. In particular, we start by fitting the parameters of the backbone algorithms to the PCG dataset. Lastly, we conduct a thorough study on the way the NMF constructs basis functions and how the latter are affected by the noise present in PCGs. This study led to a database-wide analysis of the time-frequency structure of heart sounds.

In Chapter 4 we start devising a NMF-based heart rate detection algorithm based on the information gathered in the previous chapter. In particular, we give emphasis to PCG denoising and develop multiple routines. Among the spectrogram-based routines, we devise the Norm Clipping (NC) and Weighted Time Compression (WTC) algorithms. Moreover, we discuss how a sparse variant of the NMF algorithm can help producing cleaner nonnegative factors. Among the signal-based routines, on the other hand, we devise the Localized Median Filter (LMF) and the NMF Adaptive Noise Canceler (NMF-ANC) algorithm. The NMF-ANC, in particular, is based on Adaptive Noise Cancellation, and is able to denoise the PCG by exploiting the information provided within the nonnegative factors. More precisely, from the latter the algorithm is able to generate a reference for the noise present within the PCG and thus uses it to eliminate all the correlated components.

In Chapter 5 we finalize the remaining details of the detection algorithm by describing how the previously developed routines can be used to iteratively denoise the PCG signals.

Chapter 1

Introduction

Continuous cardiac monitoring is a key tool employed to detect early symptoms of cardiovascular diseases as well as to assess the general condition of a patient relative to their cardiac rhythm. The most widespread technique used in a medical environment to ascertain a patient's heart rate is the ***electrocardiogram*** (ECG), a recording of the heart's electrical activity measured through electrodes that detect changes in the muscles' polarization following each cardiac cycle.

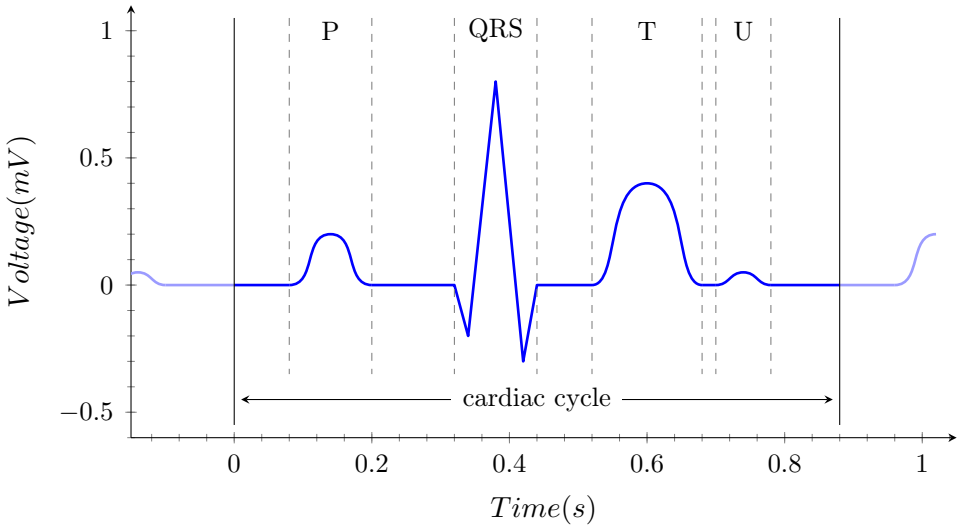


Figure 1.1: *Typical electrical signal recorded during a cardiac cycle.*

Without entering into the details and taking as reference Figure 1.1, during each cardiac cycle six distinct waves can be identified in an ECG:

- **P wave**: low amplitude wave that lasts around 80ms;
- **Q,R and S waves**: they form the QRS complex that lasts around 80-100ms and within it, the R wave usually has the largest amplitude out of all the others;
- **T wave**: medium amplitude wave that lasts around 160ms;
- **U wave**: low amplitude (or completely absent) wave.

Given the properties of these waves, the heart rate is usually extrapolated from an ECG by first identifying consecutive R waves and then measuring the time elapsed between each occurrence.
Perhaps a lesser-known technique that shares the same purposes as the ECG is the ***phonocardiogram*** (PCG), a recording of the sounds and murmurs made by the heart detected through a specialized device, the phonocardiograph.

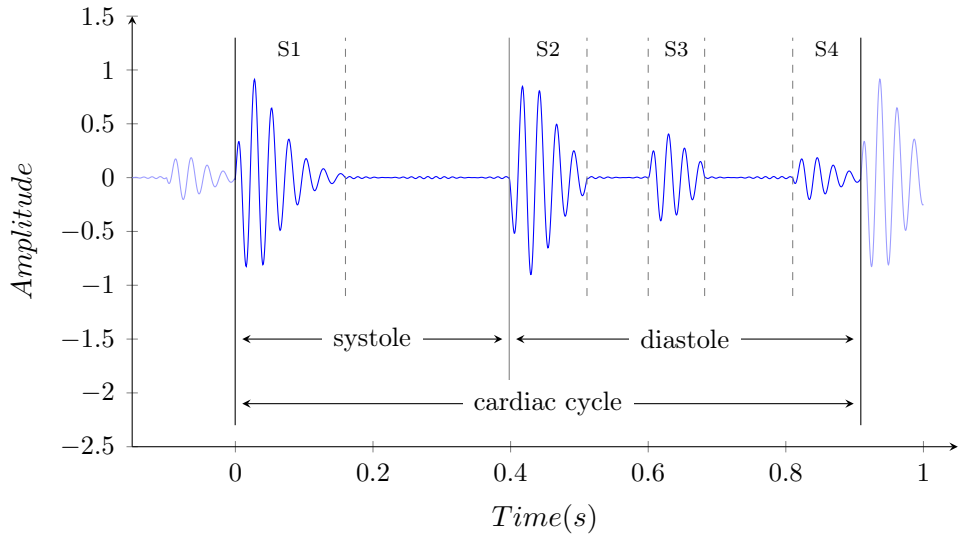


Figure 1.2: *Typical audio signal recorded by a phonocardiograph during a cardiac cycle.*

Taking as reference Figure 1.2, during each cardiac cycle four distinct heart sounds can be identified in a PCG:

- **First heart sound (S1)**: it is caused by the closure of the atrioventricular valves at the beginning of ventricular systole and lasts around 70-150ms with a frequency content between 50-150Hz;

- **Second heart sound (S2)**: it is caused by the closure of the semilunar valves at the beginning of ventricular diastole and lasts around 60-120ms with a frequency content between 50-200Hz;
- **Third and Fourth heart sound (S3 and S4)**: they are rarer sounds associated to both normal and pathologic states that last around 40-100ms with a frequency content between 50-90Hz.

As a consequence, the most common way to calculate the heart rate given a PCG is to correctly detect consecutive same-type sounds and measure the time elapsed between each occurrence.

From a mathematical and engineering point of view, both ECGs and PCGs have been thoroughly studied for decades and efficient algorithms that detect cardiac events have been developed. Most of these algorithms share the same backbone structure: a peak-detection routine is applied to a denoised, preprocessed input data (the latter being either an ECG voltage-over-time graph or a PCG audio file), as we will see in more details in Chapter 3. While the *purpose* of this dissertation is nothing new, namely detecting cardiac events from a PCG, the *means* we intend to employ in order to achieve the former are non-canonical. As a matter of fact we will present a Nonnegative Matrix Factorization (NMF) approach that does not rely on the knowledge of a synchronous ECG signal. In particular, we will focus our attention towards PCG denoising and develop multiple routines specialized in addressing different types of noise. Most notably, we will devise a NMF-based Adaptive Noise Canceler that fully takes advantage of the information provided within the two nonnegative factors to generate an estimate for the noise present in the data.
To cite a few related works, in [8] PCG signals are denoised using a NMF approach that relies on the presence of synchronous ECGs to detect the rows of the right factors that contain cardiac events' activation coefficients. In [15] the authors develop a NMF-based noise sensing procedure that requires an a priori construction of dictionaries for both the events to be detected and the noise present in the data. Similarly, in [4] the dictionaries for the events and noise are used to construct time-frequency weighting masks for the nonnegative input spectrogram matrix. Lastly, though with different premises, in [16] an adaptive noise canceler similar to the one we devise in this dissertation is used to remove ego noise from audio signals recorded by a rescue robot.

Chapter 2

Nonnegative Matrix Factorization

Introduced in the '90s by Paatero et al. [19] as an alternative to the established principal component analysis, nonnegative matrix factorization has since become known as an effective and flexible data analysis procedure. Indeed in the 1999 paper by Lee and Seung [17] which popularized this method, the authors highlight how the nonnegativity constraints allow the former to both learn a set of visible variables as well as to infer the values of hidden ones in much the same way that a neural network would: the nonnegativity of the considered quantities is strictly related to the fact that the firing rates of neurons cannot be negative.
In the following sections we will introduce the NMF formulation and provide a general algorithmic scheme used to construct the nonnegative factors. Subsequently we will present some specific formulations that will be employed in the later chapters as well as mention convergence results.

2.1 General formulation

Given a nonnegative matrix $X \in \mathcal{M}_{m\times n}(\mathbb{R}^+)$, a factorization rank r and an error measure $D(\cdot,\cdot)$ between two matrices, the problem of computing a NMF of X consists in solving the following constrained, nonlinear optimization problem:

$$\begin{aligned} \text{minimize} \quad & D(X, WH) \\ \text{subject to} \quad & W \in \mathcal{M}_{m\times r}(\mathbb{R}^+) \\ & H \in \mathcal{M}_{r\times n}(\mathbb{R}^+) \end{aligned} \tag{2.1}$$

In other words, we are interested in approximating X as the product of two nonnegative matrices with respects to the chosen distance $D(\cdot,\cdot)$. Indeed, a solution pair (W,H) of (2.1) is called *approximate* NMF of X since we do not require the product WH to exactly reconstruct the given matrix.

The choice of the error measure is of crucial importance when designing a NMF model since not only does it need to reflect known information about the data collected in X, but also its regularity plays a significant role when deriving optimality conditions. A similar argument can be made with regards to the choice of the factorization rank. As a matter of fact, if the factors have physical meaning then some information on r may be available prior to the execution of numerical algorithms, otherwise one may choose the former based on the results yielded by the latter.

2.1.1 β-divergences

As far as the error measure is concerned a noteworthy mention is the family of ***β-divergences***. They are a parametric family of distance measures d_β defined on nonnegative scalars commonly used in a NMF setting as part of the error measure:

$$d_\beta(z,y) = \begin{cases} \frac{z}{y} - \log\frac{z}{y} - 1 & \text{if } \beta = 0, \\ z\log\frac{z}{y} - z + y & \text{if } \beta = 1, \\ \frac{1}{\beta(\beta-1)}(z^\beta + (\beta-1)y^\beta - \beta z y^{\beta-1}) & \text{if } \beta \neq 0, 1. \end{cases}$$

$$D_\beta(\cdot,\cdot) = \sum_{i,j} d_\beta((\ \cdot\ )_{ij}, (\ \cdot\ )_{ij})$$

Some relevant properties of the β-divergences that can be inferred from Figure (2.1) below are the following:

- *Convexity.* The map $d_\beta(z,y)$ is convex in the second argument for $\beta \in [1,2]$. This implies that $D_\beta(X, WH)$ is convex in H for W fixed and vice versa, which makes alternating optimization strategies easier to implement.
- *Positive homogeneity.* All β-divergences are positively homogeneous of degree β, that is, they satisfy:

$$d_\beta(\gamma z, \gamma y) = \gamma^\beta d_\beta(z,y)$$

for any $\gamma > 0$ and $z, y \geq 0$. This implies that as β increases, the β-divergences become more and more sensitive to large input values.

- *Limit behaviour.* For $\beta \leq 1$ it holds $\lim_{y \to 0} d_\beta(z, y) = +\infty$ while for $y \leq z$ we notice that the β-divergences decrease as β increases. This implies that NMF formulations with β-divergences will tend to overapproximate (resp. underapproximate) the entries of X for $\beta \leq 1$ (resp. $\beta \geq 1$).

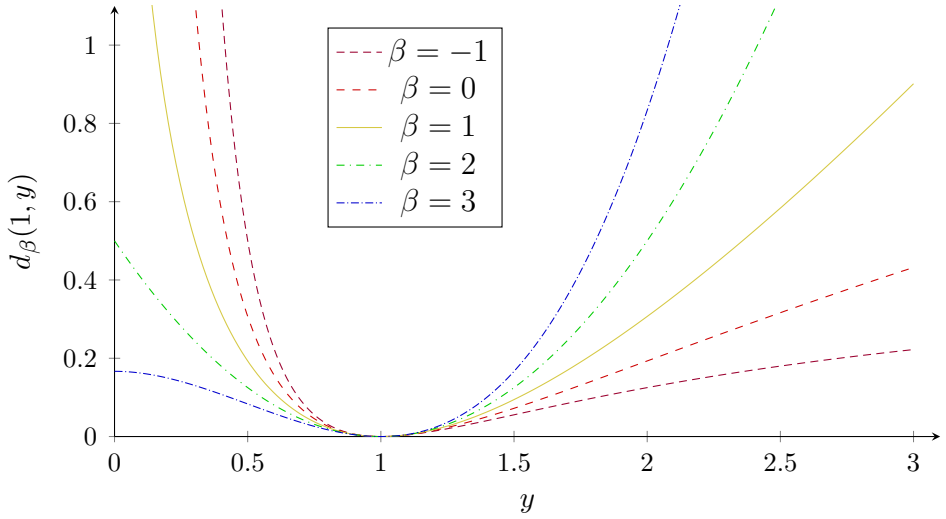


Figure 2.1: *Graphs of the β-divergences $d_\beta(1, \cdot)$ for $\beta \in \{-1, 0, 1, 2, 3\}$.*

Moreover, the following table provides the domains of $d_\beta(z, \cdot)$ and $d'_\beta(z, \cdot)$ depending on the values of z and β:

	$\beta \leq 0$	$\beta \in (0, 1]$	$\beta > 1$
$z = 0$	$\varnothing$	$\mathbb{R}^+$	$\mathbb{R}^+$
$z > 0$	$\mathbb{R}^{++}$	$\mathbb{R}^{++}$	$\mathbb{R}^+$

	$\beta \leq 0$	$\beta \in (0, 1)$	$\beta \in [1, 2)$	$\beta \geq 2$
$z = 0$	$\varnothing$	$\mathbb{R}^{++}$	$\mathbb{R}^+$	$\mathbb{R}^+$
$z > 0$	$\mathbb{R}^{++}$	$\mathbb{R}^{++}$	$\mathbb{R}^{++}$	$\mathbb{R}^+$

Table 2.1: *Domains of $d_\beta(z, \cdot)$ (top) and $d'_\beta(z, \cdot)$ (bottom) depending on z and β.*

From the latter we notice that the β-divergence NMF formulation for $\beta \leq 0$ should only be used when the input matrix X is strictly positive. Similarly, by choosing $\beta \leq 1$ the resulting NMF formulation cannot approximate strictly positive entries of X with 0, that is, if $X_{ij} > 0$ then $(WH)_{ij} > 0$.

Lastly, it may be useful to notice that in applications the choice of the specific value for the parameter β often depends on the assumed statistical nature of the data.
Indeed, given $\hat{W} \in \mathcal{M}_{m\times r}(\mathbb{R}^+)$, $\hat{H} \in \mathcal{M}_{r\times n}(\mathbb{R}^+)$ and assuming X to be an observation of a collection of random variables $\widetilde{X}_{ij}$ that depend on the deterministic parameters $(\hat{W}\hat{H})_{ij}$ through the relations $\mathbb{E}[\widetilde{X}_{ij}] = (\hat{W}\hat{H})_{ij}$, it is known that an estimate on $\hat{W}$, $\hat{H}$ can be obtained by considering the maximum likelihood estimator associated to the observation X. In this framework, it can be proven that such estimator is the NMF solution of problem (2.1) where $D(\cdot,\cdot)$ depends on the density distributions of $\widetilde{X}_{ij}$. More precisely if $\widetilde{X}_{ij}$ are i.i.d. and:

- $\widetilde{X}_{ij} \sim \text{Gamma}\left(k, \frac{(\hat{W}\hat{H})_{ij}}{k}\right)$: then $D = D_0$, also known as the Itakura-Saito divergence;
- $\widetilde{X}_{ij} \sim \text{Poisson}\left((\hat{W}\hat{H})_{ij}\right)$: then $D = D_1$, also known as the Kullback-Leibler divergence;
- $\widetilde{X}_{ij} \sim \text{Gaussian}\left((\hat{W}\hat{H})_{ij}, \sigma\right)$: then $D(\cdot,\cdot) = D_2(\cdot,\cdot) = \|\cdot - \cdot\|_2$;
- $\widetilde{X}_{ij} \sim \text{Laplace}\left((\hat{W}\hat{H})_{ij}, \sigma\right)$: then $D(\cdot,\cdot) = \|\cdot - \cdot\|_1$;
- $\widetilde{X}_{ij} \sim \text{Uniform}\left((\hat{W}\hat{H})_{ij} - \delta, (\hat{W}\hat{H})_{ij} + \delta\right)$: then $D(\cdot,\cdot) = \|\cdot - \cdot\|_\infty$.

2.1.2 Regularization

The general NMF formulation presented in problem (2.1) is often modified in order to impose specific ***regularizing properties*** on the factors W and H. The reason behind the need for such additional constraints is twofold. Firstly, the aforementioned formulation lacks identifiability: there may exist multiple optimal solution pairs (W, H) and additional limitations on the factors could eliminate some of them. Secondly, we may be interested in designing a NMF model in which the factors have distinct physical or analytical interpretations: in this case, the nonnegativity constraints might simply not suffice.

A broader class of NMF models that is often used to address these issues is the following:

$$\begin{array}{ll} \text{minimize} & D(X, WH) + \alpha_W f_W(W) + \alpha_H f_H(H) \\ \text{subject to} & W \in \mathcal{M}_{m\times r}(\mathbb{R}^+) \\ & H \in \mathcal{M}_{r\times n}(\mathbb{R}^+) \end{array} \tag{2.2}$$

where f_W, f_H are regularizers that promote solutions with the desired properties and α_W, $\alpha_H \geq 0$ are penalty parameters. Notice however that when only one of the two factors is regularized, i.e. $\alpha_W = 0$, it is good practice to use these models with some other kind of regularization on W and vice versa. Indeed, due to identifiability issues, if (W, H) is a solution pair then also $(\gamma W, \frac{1}{\gamma}H)$ is a solution pair for any $\gamma > 0$ and thus most regularizers will actually promote the corresponding factor to approach zero while the other will increase in norm (consider for example a regularizer f_H satisfying $f_H(\frac{1}{\gamma}H) = \mathcal{O}_\infty(\frac{1}{\gamma})$).
Let us now describe some of the major classes of regularizers:

- *Sparsity-promoting.* In order to promote sparse factors it is common practice to employ the sparsification properties of the 1-norm. Thus, a possible regularizer choice is given by:

$$\begin{aligned} f_W(W) &= \|W\|_1 \\ f_H(H) &= \|H\|_1 \end{aligned}$$

- *Smoothness-promoting.* It is not uncommon in applications for the columns of W or the rows of H to be discretizations of piecewise smooth functions. In these cases, in order to promote such structure, one may employ the following regularizers:

$$\begin{aligned} f_W(W) &= \sum_{k=1}^{r} \sum_{i=1}^{m-1} (W(i,k) - W(i+1,k))^2 \\ f_H(H) &= \sum_{k=1}^{r} \sum_{i=1}^{n-1} (H(k,i) - H(k,i+1))^2 \end{aligned}$$

- *Orthogonality-promoting.* To promote the columns of W or the rows of H to overlap as little as possible, orthogonality constraints may be enforced through the regularizers:

$$\begin{aligned} f_W(W) &= \|W^\top W - I_r\|_2^2 \\ f_H(H) &= \|HH^\top - I_r\|_2^2 \end{aligned}$$

- *Minimum volume-promoting.* By interpreting the columns of W as a dictionary with which we are interested in reconstructing the data in the matrix X through the convex combinations encoded in the columns of H, it makes sense to look for factors W as close as possible to the data points. This can be achieved by enforcing the columns of W to span small volumes. With this in mind, one may use the regularizer:

$$f_W(W) = \det(W^\top W)$$

2.2 Algorithmic approaches

The regularized NMF formulation presented in problem (2.2) in the last section is NP-hard and, as such, all algorithms designed to tackle it follow iterative, local optimization schemes that strive to converge to local minima or, more generally, to stationary points.
An important class of algorithms employed in this framework is that of ***two-block coordinate descent*** (2-BCD) methods, where the blocks of W and H variables are optimized alternately, fixing at each iteration one of the 2 blocks. Letting $F(X; W, H)$ be the objective function of problem (2.2), all 2-BCD methods follow the scheme presented in Algorithm (2.2) below:

Algorithm 2.2 Two-block coordinate descent scheme for problem (2.2)

Input: A matrix $X \in \mathcal{M}_{m\times n}(\mathbb{R}^+)$ and a factorization rank r.

Output: An approximate solution (W, H) to problem (2.2).

1: `Initialize two matrices` $W^{(0)} \in \mathcal{M}_{m\times r}(\mathbb{R}^+)$, $H^{(0)} \in \mathcal{M}_{r\times n}(\mathbb{R}^+)$.
2: **for** $t = 1, 2, \ldots$ **do:**
3: $W^{(t)} =$ `update`$_W(X, W^{(t-1)}, H^{(t-1)})$
4: $H^{(t)} =$ `update`$_H(X, W^{(t)}, H^{(t-1)})$
5: **end for**

---

where, in particular:

- `update`$_W(X, W^{(t-1)}, H^{(t-1)})$ is either an approximate or exact 1-step solution to an iterative subroutine employed to solve:

$$\begin{aligned} &\text{minimize} && F(X; W, H^{(t-1)}) \\ &\text{subject to} && W \in \mathcal{M}_{m\times r}(\mathbb{R}^+) \end{aligned} \tag{2.3}$$

starting from the initial estimate $W^{(t-1)}$;

- $\texttt{update}_H(X, W^{(t)}, H^{(t-1)})$ is either an approximate or exact 1-step solution to an iterative subroutine employed to solve:

$$\begin{aligned} &\text{minimize} && F(X; W^{(t)}, H) \\ &\text{subject to} && H \in \mathcal{M}_{r\times n}(\mathbb{R}^+) \end{aligned} \tag{2.4}$$

  starting from the initial estimate $H^{(t-1)}$.

Moreover, if at each iteration both $\texttt{update}_W$ and $\texttt{update}_H$ are initialized to the *exact* 1-step solution of their respective subroutines, the resulting algorithm is called *exact* 2-BCD method.
Recalling the structure of the objective function F, it is not unusual for the latter to satisfy some symmetric properties such as:

$$F(X; W, H) = F(X^\top; H^\top, W^\top) \tag{2.5}$$

This is the case, for example, when $F = D_\beta$ or $F = D_\beta + \alpha_W f_W + \alpha_H f_H$ with $\alpha_W = \alpha_H$ and $f_W(\cdot) = f_H(\,\cdot^\top)$. When condition (2.5) is satisfied, problems (2.3) and (2.4) become equivalent and therefore Algorithm (2.2) can be simplified significantly since any update rule for W induces an update rule for H and vice versa. Indeed, given $\texttt{update}_W$, step 4 of the algorithm can be replaced by:

$$H^{(t)} = \texttt{update}_W(X^\top, {H^{(t-1)}}^\top, {W^{(t)}}^\top)^\top$$

Ultimately, such symmetry in the objective function allows us to develop only one update rule for either problem (2.3) or (2.4) instead of two distinct update rules.

2.2.1 Majorization-minimization framework

Algorithm (2.2) relies on two iterative subroutines applied to problems (2.3) and (2.4) to generate update rules for each factor. In this regard, ***majorization - minimization*** (MM) schemes are a class of iterative algorithms that can be applied to the following, more general optimization problem:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in \Omega \end{aligned} \tag{2.6}$$

Denoting $x^{(k)} \in \Omega$ the current iterate, such schemes have the following two-step structure:

1. *Majorization step.* Construct a majorizer of f at $x^{(k)}$, that is, a function $g(x^{(k)}; \cdot)$ satisfying:

   (i) $g(x^{(k)}; x) \geq f(x)\ \forall x \in \Omega$;

   (ii) $g(x^{(k)}; x^{(k)}) = f(x^{(k)})$.

2. *Minimization step.* Define $x^{(k+1)} \in \operatorname*{argmin}_{x \in \Omega} g(x^{(k)}; x)$.

The simple conditions satisfied by the majorizer g are enough to guarantee a monotone decrease of the objective function. Indeed, it is trivial to check that:

$$f(x^{(k+1)}) \ \leq \ g(x^{(k)}; x^{(k+1)}) \ \leq \ g(x^{(k)}; x^{(k)}) \ = \ f(x^{(k)})$$

As a consequence, the only hurdle in applying these schemes is constructing a majorizer *simple enough* so that a global minimizer of $g(x^{(k)}; \cdot)$ can be computed in closed form at every iteration. One way of achieving this, for example, is choosing g in the form:

$$g(x^{(k)}; x) = \sum_i g_i(x^{(k)}; x_i)$$

where the g_i's are univariate functions. In this case computing a global minimizer of g is equivalent to working out global minimizers for every g_i.

Problem (2.6), although more general than (2.3) and (2.4), does not take into account that in our NMF setting the objective function depends also on external parameters (the $H^{(t-1)}$ variables for problem (2.3) and the $W^{(t)}$ variables for problem (2.4)). The majorization-minimization framework presented so far imposes no regularity on the majorizer with respects to those parameters. As a consequence, it will be useful to introduce a more encompassing framework which we will refer to as ***smooth majorization - minimization*** (SMM). Given a map $f : \mathcal{X} = \Omega_1 \times \cdots \times \Omega_n \to \mathbb{R}$ where $\Omega_i \subset \mathbb{R}^{n_i}$ and a current iterate $(x_1^{(k)}, \ldots, x_n^{(k)}) \in \mathcal{X}$, such schemes can be applied to optimization problems in the form:

$$\begin{aligned} &\text{minimize} && f(x_1, x_2^{(k)}, \ldots, x_n^{(k)}) \\ &\text{subject to} && x_1 \in \Omega_1 \end{aligned} \tag{2.7}$$

Much like in the MM framework, SMM schemes follow a similar two-step structure:

1. *Majorization step.* Construct a smooth, uniform majorizer of f, that is, a function $g : \Omega_1 \times \mathcal{X} \to \mathbb{R}$ satisfying:

   (i) $g(x, x_1, \ldots, x_n) \geq f(x, x_2, \ldots, x_n) \;\; \forall x \in \Omega_1, \;\; \forall (x_1, \ldots, x_n) \in \mathcal{X}$;

   (ii) $g(x_1, x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \;\; \forall (x_1, \ldots, x_n) \in \mathcal{X}$;

   (iii) $g \in \mathcal{C}(\Omega_1 \times \mathcal{X})$;

   (iv) $d \bullet (\nabla_x g(x_1, x_1, \ldots, x_n) - \nabla_{x_1} f(x_1, \ldots, x_n)) = 0 \;\; \forall (x_1, \ldots, x_n) \in \mathcal{X}$ and $\forall d \in \mathbb{R}^{n_1}$ such that $x_1 + d \in \Omega_1$.

2. *Minimization step.* Define $x_1^{(k+1)} \in \underset{x \in \Omega_1}{\operatorname{argmin}}\, g(x, x_1^{(k)}, \ldots, x_n^{(k)})$.

Simply put, SMM schemes require additional regularity on the majorizer. Indeed, at the current parameter iterate, the chosen map g must be a majorizer and its partial derivatives must coincide with those of the objective function. Moreover, these two properties must be satisfied uniformly for all such iterates.

2.2.2 Multiplicative updates

With respect to Algorithm (2.2), ***multiplicative updates*** (MU) are a noteworthy family of update rules for the factors that take form of componentwise matrix multiplications. More explicitly, under multiplicative update rules steps 3 and 4 could be rewritten as:

$$W^{(t)} = W^{(t-1)} \circ G_W(X, W^{(t-1)}, H^{(t-1)})$$
$$H^{(t)} = H^{(t-1)} \circ G_H(X, W^{(t)}, H^{(t-1)})$$

where G_W and G_H are nonnegative matrices when all their parameters are nonnegative as well.
Multiplicative updates are among the most popular update rules in a NMF framework. Indeed they are easily implemented, scale well with the input dimensions, and their first appearance was in [17], a paper that gained considerable popularity over the years. Despite these advantages, multiplicative updates suffer from a considerable drawback: the so called *zero locking phenomenon*. As a matter of fact, these update rules are unable to modify entries of the factors that have been set to zero. This seemingly small detail poses major concerns about the convergence of the resulting algorithm to

stationary points. As we will see in Section 2.2.3, first-order optimality conditions include some complementary slackness restrictions from which can be observed that setting a variable equal to zero (i.e. $W(i,j) = 0$) will prevent the algorithm from converging to a whole family of stationary points (i.e. satisfying $W(i,j) > 0$ and $[\nabla_W F(X; W, H)]_{ij} = 0$). In order to address this issue, it is good practice to initialize the first iterates $(W^{(0)}, H^{(0)})$ of Algorithm 2.2 to strictly positive matrices. Furthermore, most multiplicative updates algorithms will actually artificially maintain the entries of the new iterates above a fixed threshold for this very reason (see Theorem (2.4) for example).

2.2.3 Optimality conditions

In the previous pages we have presented the general structure of 2-BCD methods and provided an algorithm that can be employed to obtain a *numerical* solution to problem (2.2). In order to measure the quality of such a solution, we could check whether or not the latter satisfies some kind of optimality conditions relative to the problem at hand. Recalling that the ultimate goal of most iterative algorithms is to converge to stationary points of problem (2.2), assuming the objective function to be differentiable we may consider the set of optimality constraints given by the so called **Karush-Kuhn-Tucker** (KKT) conditions. Indeed if a couple (W, H) is a stationary point, it must satisfy the following:

$$\begin{aligned} &W \geq 0, \qquad \nabla_W F(X; W, H) \geq 0, \quad W \circ \nabla_W F(X; W, H) = 0 \\ &H \geq 0, \qquad \nabla_H F(X; W, H) \geq 0, \quad H \circ \nabla_H F(X; W, H) = 0 \end{aligned} \tag{2.8}$$

The above are first-order optimality conditions and as such they only provide a set of constraints that a stationary point must necessarily satisfy. The converse is, in general, not true: there may exist non-stationary points satisfying (2.8). As a consequence, the primary application of these conditions is gauging *how far* the numerical solutions are from an actual stationary point. Nevertheless, another possible use for the latter is developing heuristic update rules for the two factors. Indeed, by writing:

$$\nabla_W F(X; W, H) = \nabla_W^+ F(X; W, H) - \nabla_W^- F(X; W, H)$$

where $\nabla_W^+ F(X; W, H),\ \nabla_W^- F(X; W, H) > 0$, basic calculus results assert that:

- If $[\nabla_W^+ F(X; W, H)]_{ij} > [\nabla_W^- F(X; W, H)]_{ij}$: then a sufficiently small decrease of $W(i,j)$ will decrease the objective function;

- If $[\nabla^+_W F(X;W,H)]_{ij} < [\nabla^-_W F(X;W,H)]_{ij}$: then a sufficiently small increase of $W(i,j)$ will decrease the objective function.

Moreover, from the complementary slackness conditions we can observe that, at stationarity:

- If $W(i,j) > 0$: then it must hold $[\nabla^+_W F(X;W,H)]_{ij} = [\nabla^-_W F(X;W,H)]_{ij}$;
- If $W(i,j) = 0$: then it must hold $[\nabla^+_W F(X;W,H)]_{ij} > [\nabla^-_W F(X;W,H)]_{ij}$.

Analogous conclusions can be reached for the H variables. These remarks can therefore support the following heuristic, multiplicative update rules for the factors:

$$\begin{aligned} W^{(t)} &= W^{(t-1)} \circ \frac{[\nabla^-_W F(X;W^{(t-1)},H^{(t-1)})]}{[\nabla^+_W F(X;W^{(t-1)},H^{(t-1)})]} \\ H^{(t)} &= H^{(t-1)} \circ \frac{[\nabla^-_H F(X;W^{(t)},H^{(t-1)})]}{[\nabla^+_H F(X;W^{(t)},H^{(t-1)})]} \end{aligned} \tag{2.9}$$

While the approach used to derive updates (2.9) was indubitably heuristic, it can be proved (see [10, Section 8.2.2]) that the latter are actually the update rules generated by applying a rescaled gradient descent method to both problems (2.3) and (2.4).

2.3 Convergence results

In this section we will provide some convergence results for ***block coordinate descent*** (BCD) methods, a family of algorithms that can be applied when the variables of the considered objective function are divided into multiple blocks[1].
Given the following problem:

$$\begin{aligned} &\text{minimize} \quad f(x_1,\dots,x_n) \\ &\text{subject to} \quad (x_1,\dots,x_n) \in \mathcal{X} = \Omega_1 \times \cdots \times \Omega_n \end{aligned} \tag{2.10}$$

all (n-)BCD methods follow the scheme presented in Algorithm (2.3) below:

[1]For reference, Algorithm 2.2 is a BCD method where the variables are divided into only 2 blocks.

Algorithm 2.3 Block coordinate descent scheme for problem (2.10)

Output: An approximate solution $(x_1, \ldots, x_n)$ to problem (2.10).

1: `Initialize a point` $x^{(0)} = (x_1^{(0)}, \ldots, x_n^{(0)}) \in \mathcal{X}$.
2: **for** $t = 1,\ 2,\ \ldots$ **do:**
3: **for** $i = 1,\ 2,\ \ldots,\ n$ **do:**
4: $x_i^{(t)} =$ `update`$_i\,(\, x_1^{(t)},\ \ldots,\ x_{i-1}^{(t)},\ x_i^{(t-1)},\ \ldots,\ x_n^{(t-1)})$
5: **end for**
6: **end for**

where, in particular, `update`$_i\,(\, x_1^{(t)},\ \ldots,\ x_{i-1}^{(t)},\ x_i^{(t-1)},\ \ldots,\ x_n^{(t-1)})$ is either an approximate or exact 1-step solution to an iterative subroutine employed to solve:

$$\begin{aligned} &\text{minimize} \quad f(\, x_1^{(t)},\ \ldots,\ x_{i-1}^{(t)},\ x_i,\ x_{i+1}^{(t-1)},\ \ldots,\ x_n^{(t-1)}) \\ &\text{subject to} \quad x_i \in \Omega_i \end{aligned} \tag{2.11}$$

starting from the initial estimate $x_i^{(t-1)}$.

Again, if at each iteration all `update`$_i$'s are initialized to the *exact* 1-step solution of their respective subroutines, the resulting algorithm is called *exact* BCD method.
With respect to the above notation, we are now ready to state three convergence results:

Theorem 2.1. *[13, Corollary 2] The limit points of the iterates of an exact 2-BCD algorithm are stationary points of problem (2.10) provided that the following conditions hold:*

1. $f \in \mathcal{C}^1(\mathcal{X})$;
2. Ω_1 *and* Ω_2 *are closed convex sets.*

As far as Algorithm (2.2) is concerned, the second condition is always satisfied since the nonnegative orthant is a closed convex set. On the other hand, when working with β-divergences, the validity of the first condition must be cross-checked using Table (2.1).

Theorem 2.2. *[5, 6, Proposition 2.7.1] The limit points of the iterates of an exact BCD algorithm are stationary points of problem (2.10) provided that the following conditions hold:*

1. $f \in \mathcal{C}^1(\mathcal{X})$;
2. Ω_i *is a closed convex set* $\forall i = 1, \ldots, n$;
3. *Problem (2.11) admits a unique global solution independently of the variables* $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_n \in \Omega_1 \times \cdots \times \Omega_{i-1} \times \Omega_{i+1} \times \cdots \times \Omega_n$, $\forall i = 1, \ldots, n$;
4. `update`$_i$ *must return the exact global solution* $\forall i = 1, \ldots, n$.

Theorem 2.3. *[14, Theorem 2] The limit points of the iterates of an exact BCD algorithm are stationary points of problem (2.10) provided that the following conditions hold:*

1. $f \in \mathcal{C}^1(\mathcal{X})$;
2. Ω_i *is a closed convex set* $\forall i = 1, \ldots, n$;
3. *The subroutines employed for problems (2.11) follow a SMM framework and either one of the following is satisfied:*
   - *(i) the majorizers are quasi-convex in the first variables and their minimum is uniquely attained;*
   - *(ii) the level set* $\mathcal{X}_0 = \{x \in \mathcal{X} : f(x) \leq f(x^{(0)})\}$ *is compact and at least* $n - 1$ *of the majorizers have an uniquely attained minimum.*

In particular, an exact BCD algorithm that satisfies the hypothesis of Theorem (2.3) is part of the so called ***block successive upper-bound minimization*** (BSUM) framework.

We conclude this section by providing an application of the last result to an instance of Algorithm (2.2), in which $F = D_\beta$ and a particular family of majorizers is constructed to tackle problems (2.3), (2.4) using a SMM framework:

Theorem 2.4. *[10, Theorem 8.9] Let* $\epsilon > 0$ *and consider the multiplicative update rule:*

$$W^{(t)} = \max\left(\epsilon,\ W^{(t-1)} \circ \left(\frac{\left[\left(\left(W^{(t-1)}H^{(t-1)}\right)^{\circ(\beta-2)} \circ X\right) H^{(t-1)^\top}\right]}{\left[\left(W^{(t-1)}H^{(t-1)}\right)^{\circ(\beta-1)} H^{(t-1)^\top}\right]}\right)^{\circ\gamma(\beta)}\right)$$

$$H^{(t)} = \max\left(\epsilon,\ H^{(t-1)} \circ \left(\frac{\left[W^{(t)^\top}\left(\left(W^{(t)}H^{(t-1)}\right)^{\circ(\beta-2)} \circ X\right)\right]}{\left[W^{(t)^\top}\left(W^{(t)}H^{(t-1)}\right)^{\circ(\beta-1)}\right]}\right)^{\circ\gamma(\beta)}\right)$$

where:

$$\gamma(\beta) = \begin{cases} \dfrac{1}{2-\beta} & \text{if } \beta < 1, \\ 1 & \text{if } 1 \leq \beta \leq 2, \\ \dfrac{1}{\beta-1} & \text{if } \beta > 2. \end{cases}$$

Then, for any initial matrices $(W^{(0)}, H^{(0)})$, the limit points of the iterates are stationary points of problem (2.2)[2].

[2]Where the nonnegativity constraints are replaced by $W, H \geq \epsilon$ in order for the β-divergences and their derivatives to be defined everywhere.

Chapter 3

Data processing and analysis

In this chapter we will present the conventional techniques used to analyze both ECGs and PCGs. Moreover, we will introduce a NMF-based procedure to analyze PCG signals and identify the characterizing properties of its two main components.
All data used in the following sections and chapters was made publicly available by PhysioNet [1]. In particular, we will use two databases, which will be referenced in this dissertation as:

- *Database1* [12]. It contains 69 simultaneous ECG and PCG recordings, each with a duration of either 30 seconds (8 records) or 30 minutes (61 records), acquired synchronously from a three-lead ECG and a single PCG stethoscope. In some recordings, ambient noise picked up by the stethoscope is present;

- *Database2* [3]. It contains more than 3000 PCG recordings collected from four different auscultation locations on the volunteers' chests: each of the four is meant to record the sounds produced by a particular heart's valve. To compensate for the absence of synchronous ECG recordings, every PCG comes with a text file where the beginning and end instants of S1 and S2 sounds are annotated. Ambient noise picked up by the stethoscope is much less frequent and intrusive if compared to Database1.

As far as the NMF is concerned, we will use the code provided by Gillis in [11]. It is a repository containing all codes and examples present in [10], from which we will mainly use the implementation of the multiplicative update rule of Theorem 2.4. We shall refer to such algorithm as *MU-NMF(β)*.

3.1 Traditional ways of analyzing ECGs and PCGs

Let us now describe how the instantaneous heart rate can be reconstructed from a synchronous ECG-PCG signal pair. More precisely, the scope of this section is to present the more common techniques employed to process the data and correctly discern both R waves as well as S1 and S2 sounds.
Before diving into a step-by-step algorithm, which will be discussed in Section (3.1.3), it is necessary to observe that any procedure used to analyze ECGs and PCGs will be divided into two distinct routines:

- *Data filtering.* First and foremost the data needs to be filtered in order to remove as many non essential or disrupting components as possible. In ECGs for example, where powerline interference is often present, it is common practice to apply a filter to remove any 50-60Hz component. More generally, the data should be filtered to highlight the components of interest: in our case, we will be filtering out all components not directly related to R waves for ECGs or S1 and S2 for PCGs;
- *Peak detection.* Once the data has been filtered and normalized, peak detection routines are employed to identify the sought after characterizing events. In our framework, since we are dealing with one-dimensional signals, these routines will be in the form of adaptive window search algorithms.

As a consequence of the above distinction, it will be useful to analyze in more details both the concept of ***filter*** as well as the general scheme for an ***Adaptive Window Search*** (AWS) algorithm.

3.1.1 Notch, low-pass and zero-phase filters

In the realm of digital signal processing a filter is a procedure that eliminates from the input data unwanted (frequency) components. From a mathematical point of view, a filter can be described as a Discrete-time, Linear, Time-Invariant (DLTI) system, that is, an operator T transforming discrete-time input sequences $u(:)$ into discrete-time output sequences $y(:)$

$$y = T[u]$$

which is both linear and time-invariant[3]. For our purposes, we should recall that the output of a DLTI system can be expressed equivalently through either an Auto-Regressive Moving-Average (ARMA):

[3]That is, $T[u(:-j)](:) = T[u](:-j)\ \forall j \in \mathbb{Z}$.

$$\sum_{i=0}^{n_a} a_i y(k-i) = \sum_{j=0}^{n_b} b_j u(k-j) \qquad \forall k \in \mathbb{Z} \tag{3.1}$$

for some appropriate coefficients $\{a_i\}_{i=0,\dots,n_a}$ and $\{b_j\}_{j=0,\dots,n_b}$ (n_a is the order of the filter), or as a convolution sum:

$$y(:) = \sum_{j\in\mathbb{Z}} u(j)h(:-j) \tag{3.2}$$

where $h = T[\delta]$ is the discrete impulse response and $\delta(:)$ is the discrete impulse at time $k = 0$ (that is, $\delta(k) = \delta_{0,k}\ \forall k \in \mathbb{Z}$).
In particular, filters are uniquely identified by their discrete impulse response $h(:)$ and can be divided into two categories based on the support length of the same. Indeed, if the latter has a finite support length (this is the case when $n_a = 0$ and $n_b \neq 0$) the former is called Finite Impulse Response (FIR) filter. Conversely, if the latter has infinite support length (this is the case when $n_a \neq 0$ and $n_b \neq 0$) the former is called Infinite Impulse Response (IIR) filter.
In order to better understand how filters can be employed to eliminate unwanted frequency components from a given input $u(:)$, one should apply the Discrete Fourier Transform (DFT) to both sides of Equation (3.2). As a matter of fact, letting $\mathcal{Y}(:)$, $\mathcal{U}(:)$, $\mathcal{H}(:)$ be the DFTs of $y(:)$, $u(:)$, $h(:)$ respectively and recalling that convolution sums in the time domain transform into products in the frequency domain, we obtain:

$$\mathcal{Y}(:) = \mathcal{U}(:)\mathcal{H}(:)$$

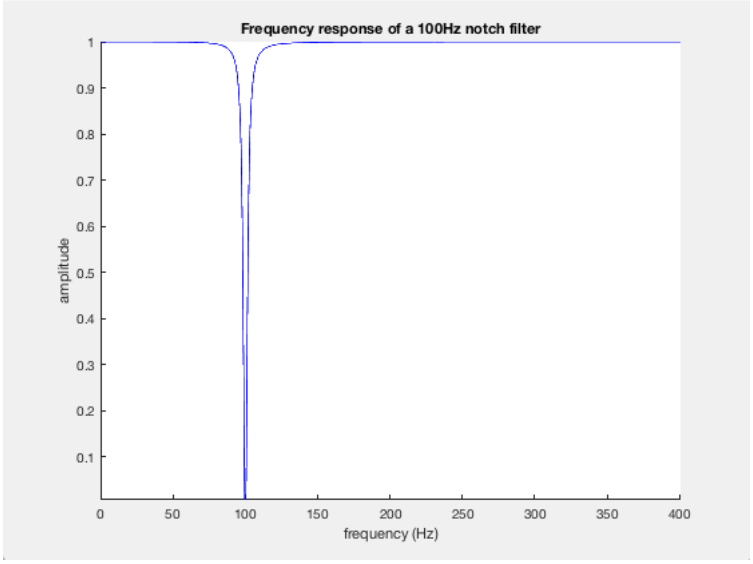
As a consequence, in the frequency domain the filter has the effect of multiplying the DFT of the input by the DFT of the discrete impulse response, the so called frequency response $\mathcal{H}(:)$ of the filter. This property can therefore be used to put to zero specific frequency components of any input, prior the construction of a filter with a suitable frequency response. In this brief discussion we will not enter into the details on how filters can be designed: Oppenheim et al. [18] and Antoniou [2] provide a more in-depth analysis.
It should also be noticed that, in general, filters introduce phase shifts in the output due to the fact that $\arg(\mathcal{H}(:)) \neq 0$. In some applications phase shifts may hinder the data analysis, thus one way of addressing this is to construct so called zero-phase filters. Zero-phase filters can be obtained by performing a forward-backward filtering, more precisely, the input $u(:)$ is filtered,

time-reversed, filtered again and finally time-reversed. Indeed, recalling that time-reversal transforms into a complex conjugation in the frequency domain, letting $\hat{y}(:)$ be the final output of such a process and $\widehat{\mathcal{Y}}(:)$ its DFT, we get:

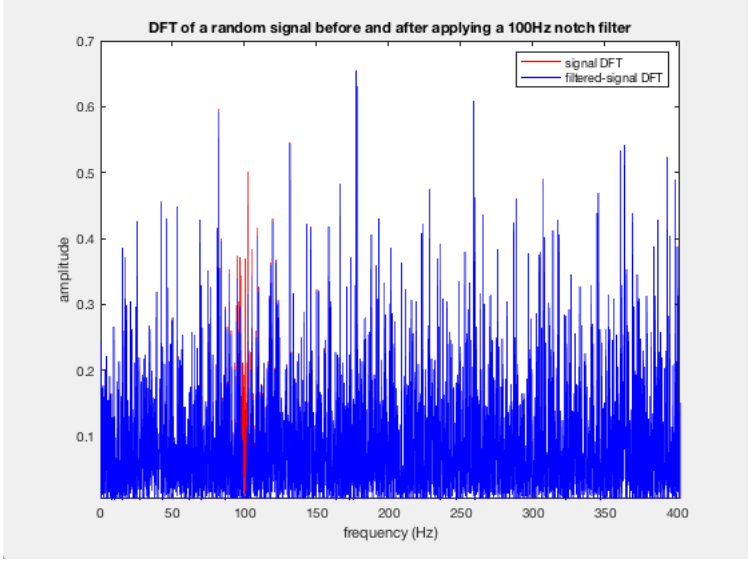
$$\widehat{\mathcal{Y}}(:) = \mathcal{U}(:)|\mathcal{H}(:)|^2$$

In other words, we obtain a new filter characterized by a purely real frequency response $|\mathcal{H}(:)|^2$ with similar filtering properties to the original one and no phase distortion.

As far as our data processing is concerned, we should introduce two distinct types of filters. The first are notch filters, whose purpose is attenuating a specific frequency component from the input signal. As we will see in the later sections, this is the type of filters that will be employed to remove powerline interference from ECGs.



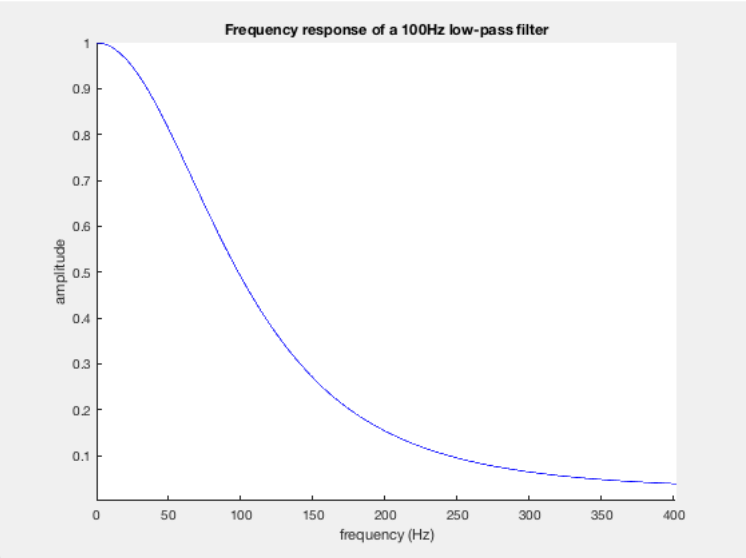
(a)



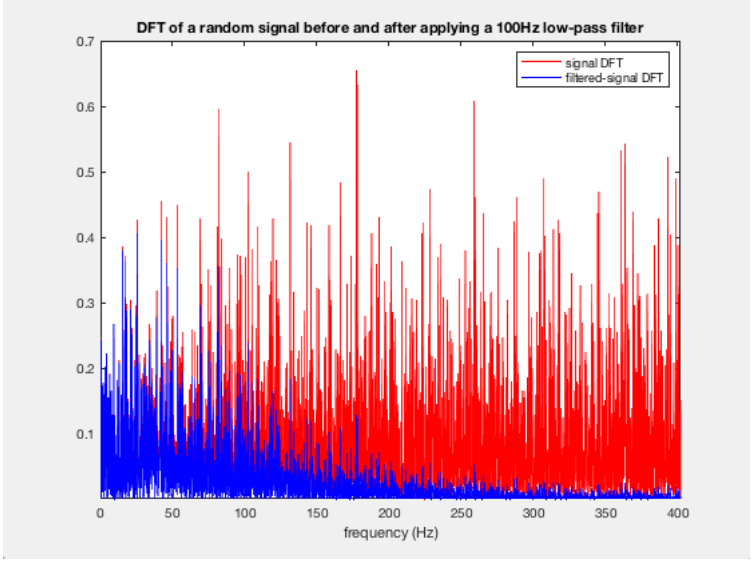
(b)

Figure 3.1: *Frequency response amplitude (a) and effect on the DFT of a random signal (b) of a second order zero-phase notch filter at the 100Hz frequency.*

The second are low-pass filters, which, as their name suggests, attenuate all frequency components above a certain threshold. Together with their high-pass counterpart, these filters will be used to highlight specific frequency bands corresponding to R waves and heart sounds in ECGs and PCGs respectively.



(a)



(b)

Figure 3.2: *Frequency response amplitude (a) and effect on the DFT of a random signal (b) of a second order zero-phase low-pass filter with a 100Hz frequency threshold.*

3.1.2 Adaptive window search algorithms

Window search algorithms are a family of schemes that rely on a shifting window to detect local peaks (maxima or minima) within a one-dimensional discrete signal. More precisely, such schemes iterate over each signal sample and construct a window centered at the latter. If the current sample value coincides with the peak computed over all samples spanned by the window, it is labeled as a local peak.
Clearly, the window radius plays a crucial role in the effectiveness of the algorithm: if it's too big, local peaks close together may not be labeled as such, conversely, if it's too small, the scheme may label as local peaks minor fluctuations in the signal. Moreover, the radii need not remain fixed for all samples. Indeed, each signal sample may be given a distinct window radius: the resulting scheme is known as an Adaptive Window Search, which we will include in Algorithm (3.1) below.

Recalling our main goal, in the next section we will see how, initially, all the radii will be fixed to a constant (Fixed Window Search) related to a rough estimate of the heart rate and then subsequently specialized to better detect R waves as well as S1 and S2 sounds.

Algorithm 3.1 Adaptive Window Search for local maxima

Input: A signal $x = \{x_k\}_{k=1,\dots,N}$ and window radii $r = \{r_k\}_{k=1,\dots,N}$.

Output: A set of local maxima indexes I.

```
Initialize I = ∅
for k = 1, ..., N  do:
    if k − r_k > 0 and k + r_k < N
        W = [k − r_k, ..., k + r_k]
    elseif k − r_k > 0
        W = [N − 2r_k, ..., N]
    else
        W = [1, ..., 2r_k]
    end if
    if max(x_W) == x_k
        I = I ∪ {k}
    end if
end for
```

3.1.3 Heart rate detection

We are now ready to describe the step-by-step procedure used to detect main heart events in a synchronous ECG-PCG pair. As a prerequisite, we assume that both ECG and PCG have been decimated to the same sampling frequency in order to be dealing with discrete signals of the same length and therefore have a time-correspondence between the indexes. Below, the steps labeled with "1.x" will deal with the ECG analysis while those labeled with "2.x" will deal with the PCG. For a more concrete outlook on the procedure, see the code provided in [12].

- *Step 1.1:* ECG filtering. First of all, a 50Hz second order zero-phase notch filter is applied to the ECG in order to attenuate the influence of the powerline interference. Second of all, a second order zero-phase low-pass filter with a 10Hz frequency threshold is applied, followed by a second order zero-phase high-pass filter with a 40Hz frequency threshold. The purpose of these last two filters is to isolate the [10:40]Hz frequency band, which

characterizes the QRS complex.

- *Step 1.2:* ECG saturation. The filtered signal is then saturated above five-sigma using a rescaled sigmoid. This step normalizes the data and provides an upper bound for the peaks we are interested in detecting.

- *Step 1.3:* First round of R wave detection. Assuming a constant heart rate with a frequency $f_0 = 1.4$Hz (84bpm), a fixed window search algorithm with constant radii $r_0 = \lfloor \frac{1}{2f_0} \rfloor$ is applied. Subsequently, any (fake) peak detected within a distance r_0 from the preceding one is eliminated. Utilizing the local peaks index set I_0 returned by the above routine, a median heart rate frequency f_1 is computed. The latter constitutes a more accurate estimate of the heart frequency. Moreover, considering the median (instead of the mean) allows us to take into account some of the variability that might be present in the recording as well as to reduce the likelihood of skipping over local peaks during future window searches.

- *Step 1.4:* Second round of R wave detection. Initially, a fixed window search algorithm with constant radii $r_1 = \lfloor \frac{1}{2f_1} \rfloor$ is applied. From the resulting peaks index set I_1 a vector of instantaneous heart rate frequencies is computed. The latter is then normalized with a 3-element moving average filter as well as with a 3-element moving-median filter. Lastly, it is linearly interpolated over all discrete time samples to obtain a complete, average frequency vector $f_2 = \{f_2(k)\}_{k=1,\dots,N}$ (where N is the length of the ECG). Ultimately, a second adaptive window search algorithm with radii $r_2 = \{\lfloor \frac{4}{5f_2(k)} \rfloor\}_{k=1,\dots,N}$ is applied. The peaks set I_2 given by this last routine is then refined by eliminating all indexes associated to (fake) peaks with an amplitude below 33% of the median peak amplitude. Let us call $I_R = \{i_k\}_{k=1,\dots,M}$ this final set.

- *Step 2.1:* PCG filtering. A second order zero-phase low-pass filter with a 10Hz frequency threshold is applied, followed by a second order zero-phase high-pass filter with a 100Hz frequency threshold. As for the ECG, these filters isolate the [10:100]Hz frequency band in the PCG, which contains most of the frequency content of both S1 and S2.

- *Step 2.2:* PCG normalization. The PCG is normalized by applying a forward-backward 2-mean filter over a fixed number of samples. This procedure effectively transforms the PCG into a nonnegative signal approximating the envelope of the latter. As a consequence, both main heart sounds will be highlighted.

- *Step 2.3:* S1 and S2 sound detection. Initially, a fixed window search algorithm with sufficiently small and constant radii r_3 is applied. The output set I_3 will therefore contain the indexes associated to all main heart sounds peaks as well as, possibly, smaller amplitude peaks generated by minor fluctuations in the signal. At this point it is of key importance to observe that, in a heart cycle, S1 (and therefore S2) is slightly delayed if compared to the corresponding R wave in the synchronous ECG. This aspect can be exploited to localize both the S1 and S2 peak between two consecutive R wave peaks. More precisely, for each consecutive pair of indexes (i_k, i_{k+1}) in the set I_R (which we recall contains the ECG R wave indexes), the set I_3 is split into smaller subsets I_3^k each containing indexes falling between the corresponding pair, that is, $I_3 = \bigcup_{k=1}^{M-1} I_3^k$ and $i_k < j < i_{k+1}$ $\forall j \in I_3^k$. Assuming the peaks generated by S1 and S2 to have a much bigger amplitude than those associated to minor fluctuations, the indexes in every I_3^k are sorted by the amplitude of their peaks and only the first two are kept. In conclusion, the sets I_{S_1} and I_{S_2} containing the indexes for S1 and S2 peaks are obtained, respectively, by selecting in each I_3^k the first and second index sorted with respect to time.

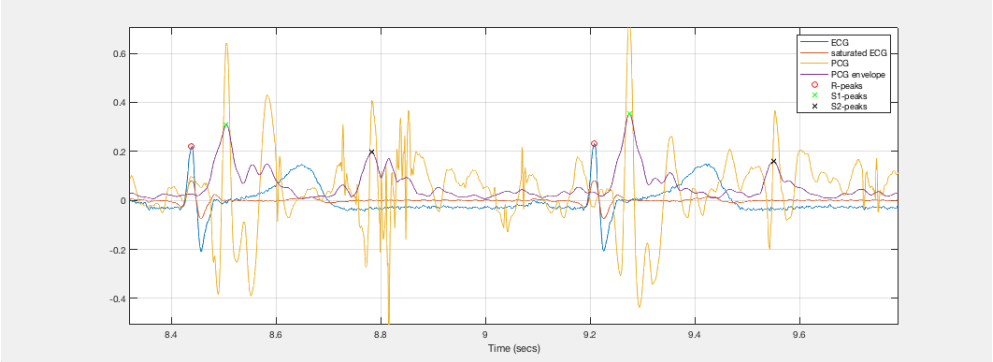


Figure 3.3: *Heart rate detection algorithm applied to the synchronous pair "ECG0005"-"PCG0005" from Database1.*

3.2 NMF approach

Unlike the previous section, in which the two main heart sounds in a PCG were exposed by peak-detection routines, we will now see how such a task can be performed by exploiting the learning capabilities of the NMF.
Given a discrete time-sampling x of a PCG signal, one way of producing a nonnegative matrix X to which apply the NMF is to consider the *elementwise*

squared-norm[4] of the discrete Short-Time Fourier Transform (STFT) matrix of x, which we will compactly denote with:

$$X = |STFT(x)|^2 \tag{3.3}$$

Such a matrix is more commonly known as the ***spectrogram*** of x. We recall that the columns of X are indexed by the time intervals corresponding to the (local) support of the window function used to compute the time-localized Discrete Fourier Transform (DFT) of x. The rows of X on the other hand are indexed by the discrete frequencies that can be detected by the DFT. As a whole, the element X_{ij} of the above matrix encodes the i-th discrete frequency content in the time-localized DFT of x computed on the j-th time interval. Simply put, X provides a time sensitive description of the frequency content in the PCG sampling x.
In particular, it should be noted that:

Remark 3.1. *Given the (quasi) time-periodic structure of a PCG, the above matrix will inherit columnwise-periodic properties.*

All time intervals containing sections of same-type heart sounds will correspond to columns of X with similar frequency contents. Similarly, time intervals inbetween heart sounds, in absence of (instrument) noise and other body related murmurs, will correspond to columns in which no frequencies are detected (or rather, in which the frequency content has negligible weight relative to the columns containing heart sounds). As a consequence, we wish to leverage this particular structure to facilitate the learning and modeling abilities of the NMF. To be more precise, if the two main heart sounds were to have sufficiently different frequency contents, we may expect the NMF to not only identify the former within the PCG signal, but also differentiate between the two.

Before diving into the analysis we need to have a rough understanding of what a NMF applied to X would generate. In other words, we have to give a mathematical and physical interpretation to the following three key components:

- *Factorization rank* r: it encodes the number of basis functions that will be used to approximate the matrix X. We may think of it as the degrees of freedom we expect the columns of X to have. Similarly, from a clustering

[4] Actually, we could also take $X = |STFT(x)|^n$ for $n > 0$, but in hindsight $n = 2$ produced the best results in our NMF framework.

standpoint, r might be seen as the number of different clusters that we need to take into account to correctly partition the columns of X based on their intrinsic properties. In our framework, a sensible choice for the factorization rank is $r = 2$ since we know the signals we are dealing with have two main components, namely the S1 and S2 heart sounds, which we can expect to behave differently in the frequency domain. Notice, however, that this is far from being the only possible choice: as we will see in Section 3.2.3 a higher number of basis functions might help produce better nonnegative factors;

- *Left factor* W: it contains the basis functions used to represent the matrix X. The choice (construction) of the basis functions is clearly dependent from the objective we intend to use. Nevertheless, when dealing with simple objectives like β-divergences, the algorithms will generate basis functions in order to better approximate columns of X that have a big weight with respect to the selected β-divergence. Again, in our framework, we expect the columns of X associated to heart sounds to have a much bigger weight than the others, hence it is sensible to assume the NMF will create basis functions directly related to the frequency contents of S1 and S2. Moreover, as highlighted earlier, if the latter are sufficiently different, the NMF might even generate two distinct basis functions, one for each sound;

- *Right factor* H: it encodes the coefficients used in the conical combinations of basis functions to approximate the matrix X. Every row of H contains the activation coefficients of the corresponding basis function. Similarly, every column contains the activation coefficients for each basis function at the corresponding time interval. As a consequence, if the NMF correctly encoded basis functions related to heart sounds, we may use H to identify when the latter appear during the PCG. Indeed, if a row of H corresponds to such a basis function, we may expect the activation coefficients within it to have a bigger norm in correspondence of the columns of H associated to time intervals containing the heart sound in question. This property can therefore be used to reconstruct the instantaneous heart rate.

We are now ready to start discussing more technical details and begin to analyze the concrete output obtained by applying the above procedure to real data.

3.2.1 STFT parameters

The first thing we need to address is the choice of the STFT parameters. In particular, the length of the support of the window function δT is of crucial

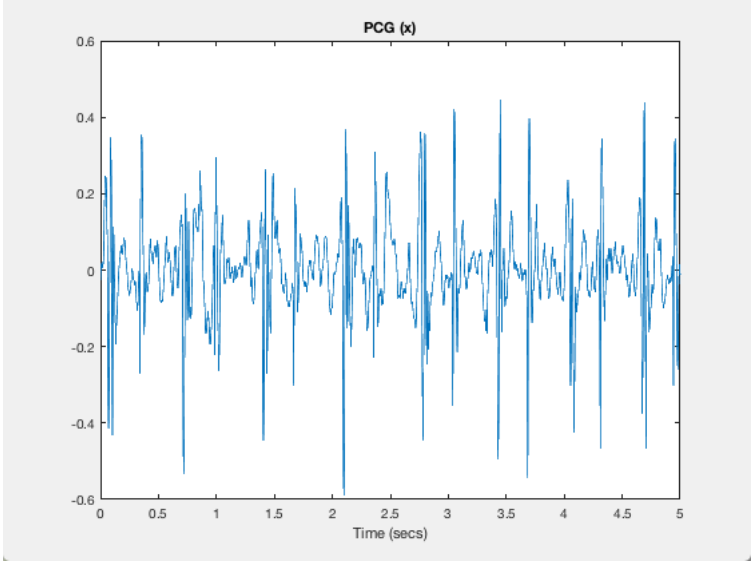
importance. Indeed we want the latter to be *small enough* to separate adjacent S1 and S2 sounds, but at the same time *big enough* to detect possible lower frequency contents. With this in mind and recalling that the two main heart sounds can have a duration as short as 50ms, a sensible choice would be to consider window functions with local supports of duration $\delta T = 25$ms. Unless otherwise specified, this is the value we will adopt.
The second parameter is the support overlap between consecutive window functions. Much like all the other parameters that we will take into account in this section, there are many different but equally valid choices, although one may prefer some specific values over others based on the task at hand. For example, if we were solely interested in detecting heart sounds and feeding the resulting matrix X directly to a NMF algorithm, a solid choice would be to construct the STFT with nonoverlapping local supports (that is, the support of the next window function begins right where the current one ended). On the other hand, if we were interested in producing a more detailed representation of the signal, we could choose to let the local supports overlap for a certain percentage p of their length (that is, the support of the next window function begins $(1-p)\delta T$ seconds after the current one began).
We should notice that these first two parameters influence greatly the structure of X. Indeed:

Remark 3.2. *The number of columns within the matrix X is a function of δT and p. Moreover, the value of p regulates how many columns will be allocated to each heart sound.*

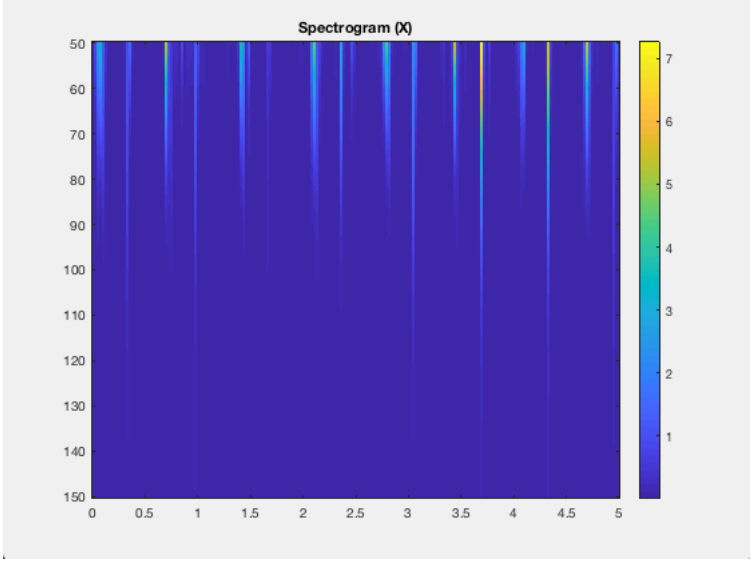
As a matter of fact, for $p = 0$ (nonoverlapping supports) we can expect, on average, each heart sound to fall within 2 to 4 columns of X, while for $p = 0.5$ we can expect 3 to 7 (assuming a sound duration between 50ms to 100ms).
Another important aspect is the frequency band we are interested in observing. We know that the main frequency content for both S1 and S2 should be located between 70Hz to 100Hz, but we could still decide to consider a broader band.
Last but not least is the choice of the window function. For simplicity of implementation, in this dissertation we will be using a Hamming window, unless otherwise specified. In its continuous form, the latter will therefore be given by:

$$w_H(t) = \tilde{\alpha} - (1-\tilde{\alpha})\cos\left(\frac{2\pi}{\delta T}t\right), \quad t \in [0, \delta T]$$

where $\tilde{\alpha}$ is usually chosen as a decimal truncation of $\alpha = \dfrac{25}{46}$.



(a)



(b)

Figure 3.4: *Phonocardiogram (a) and spectrogram (b) of signal "PCG0003" from Database1 on the time interval [10:15]s with overlap* $p = 0$ *and frequency band [50:150]Hz.*

In Figure (3.4) we can clearly observe the columnwise-periodic structure of the spectrogram. The columns corresponding to S1 and S2 have much bigger weights than the others thus making it possible to recognize each individual

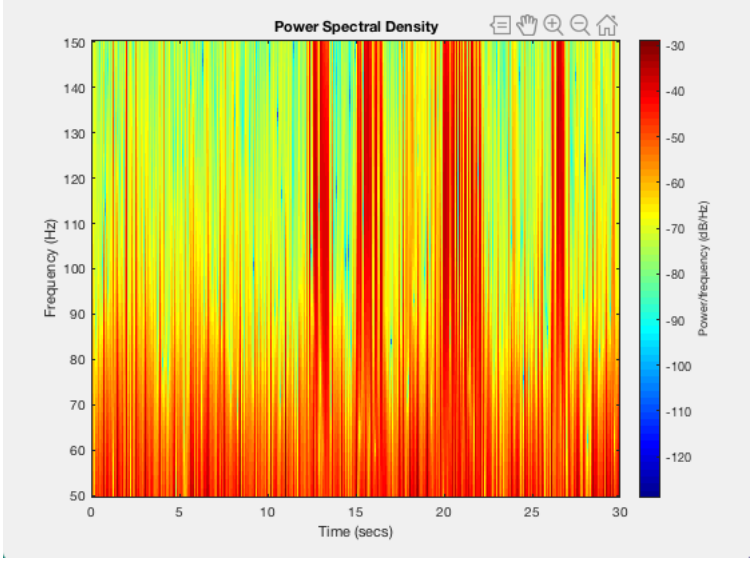
heartbeat. Arguably, extrapolating this information from the PCG alone would have been more difficult.

3.2.2 NMF parameters

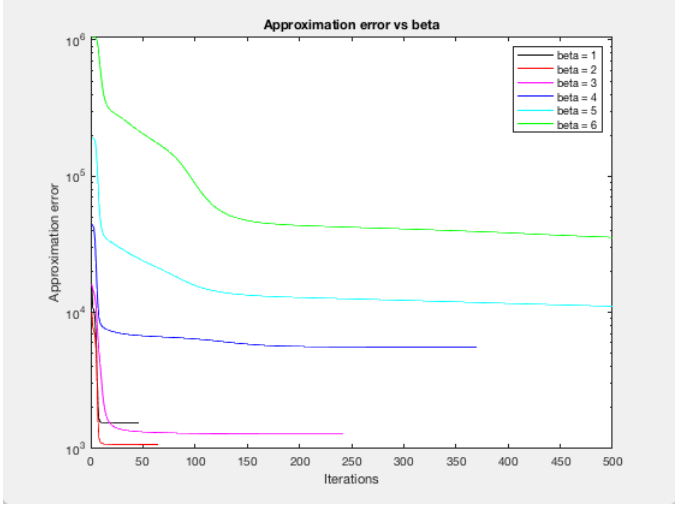
Let us now focus on the NMF parameters. As mentioned earlier, in this chapter we will limit ourselves to simpler algorithms, namely MU-NMF(β), which is based on Theorem (2.4). As a consequence, the only parameter we need to take into consideration is β itself.
First and foremost we should recall that due to the positive homogeneity property of β-divergences (see Section 2.1.1), as β increases, the objective function becomes more sensitive to larger values within the matrix X. Consequently, MU-NMF(β) will try to prioritize the approximation of columns containing such values. This is a double-edged sword. Indeed:

Remark 3.3. *If the PCG signal has relatively low-level noise, increasing β will in general guarantee a better overall approximation of X. If, on the other hand, the PCG contains some time-localized noise (e.g. ambient noise picked up by the stethoscope or other body related murmurs), increasing the value of the parameter will have the opposite result.*

The second thing we should stress is that regardless of the overall approximation, our main goal is to exploit the NMF to identify heart sounds; hence, it may be worth trading a bigger approximation error for a better, cleaner, representation of the frequency contents of S1 and S2.



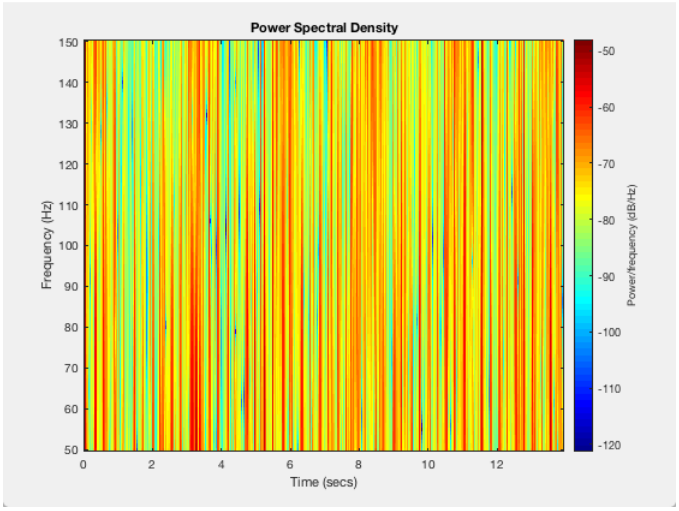
(a)



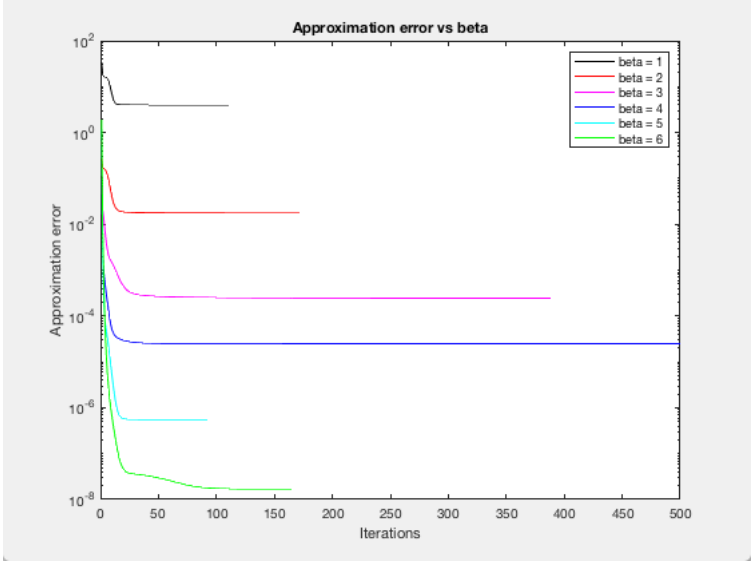
(b)

Figure 3.5: *Power Spectral Density (a) and MU-NMF(β) approximation error (b) for different values of $\beta \in \{1, 2, 3, 4, 5, 6\}$ of signal "PCG0004" from Database1 with overlap $p = 0$, frequency band [50:150]Hz and factorization rank $r = 2$.*

As evidenced by the power spectral density in Figure (3.5), the considered signal contains multiple time-localized noise bursts. In this case, the bigger values of β generate an overall worse approximation of the spectrogram if compared to the smaller ones. Moreover, even for $\beta = 2$ or $\beta = 3$, the small factorization rank does not allow the algorithm to generate enough basis functions to lower the approximation error.



(a)



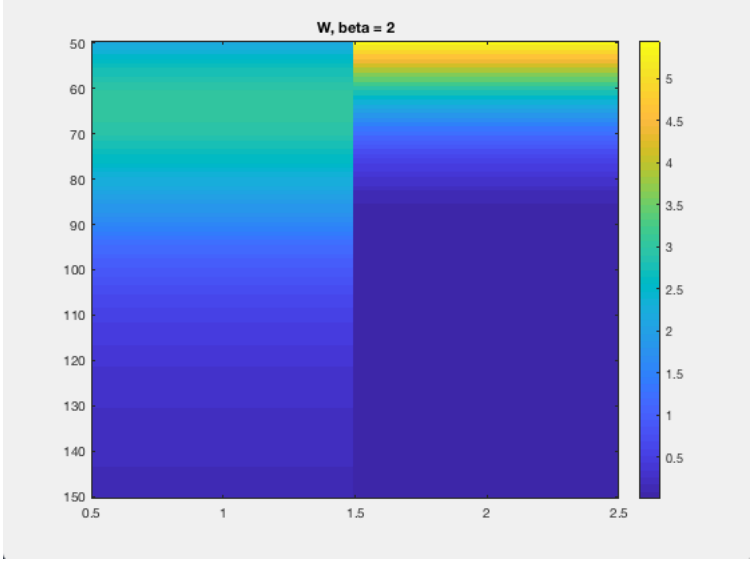
(b)

Figure 3.6: *Power Spectral Density (a) and MU-NMF(β) approximation error (b) for different values of $\beta \in \{1, 2, 3, 4, 5, 6\}$ of signal "PCG85338_PV" from Database2 on the time interval [0:14]s with overlap $p = 0$, frequency band [50:150]Hz and factorization rank $r = 2$.*

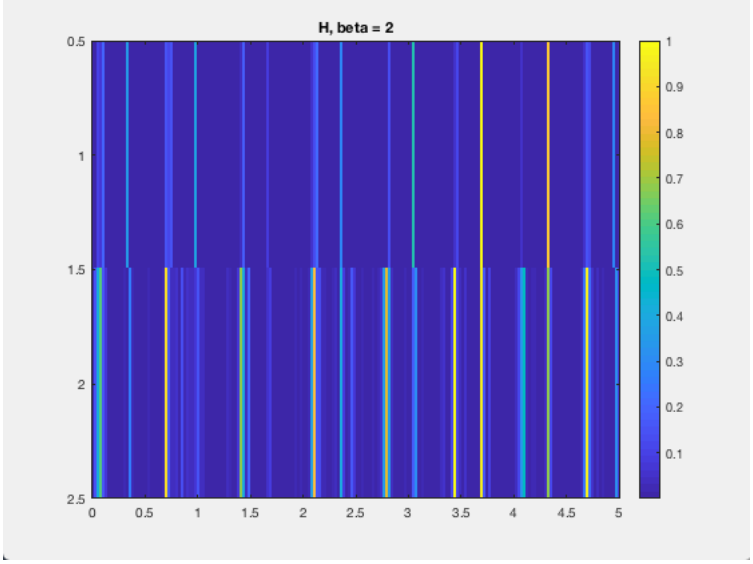
In Figure (3.6), on the other hand, we notice how a low-level noise allows the bigger values of β to better approximate the spectrogram of the signal, even with only two basis functions.
In light of these examples, we conclude that:

Remark 3.4. *Choosing an intermediate value of β, such as $\beta = 2$, is a sensible compromise to obtain a certain degree of noise robustness from the MU-NMF(β) algorithm.*

As a consequence, let us now fix $\beta = 2$ and see the actual nonnegative factors generated by MU-NMF(2) applied to the spectrogram in Figure (3.4):



(a)



(b)

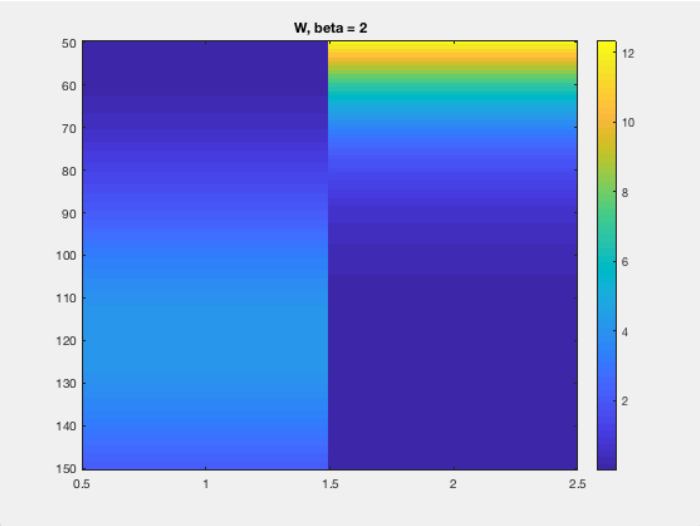
Figure 3.7: *Left factor W (a) and right factor H (b) generated by MU-NMF(2) applied to the spectrogram of signal "PCG0003" from Database1 on the time interval [10:15]s with overlap $p = 0$, frequency band [50:150]Hz and factorization rank $r = 2$.*

From the figure above we can conclude that the NMF was able to correctly detect both S1 and S2; indeed the activation coefficients in the right factor H have bigger norm in correspondence of the time intervals that contain these sounds. In particular, this means that both basis functions in the left

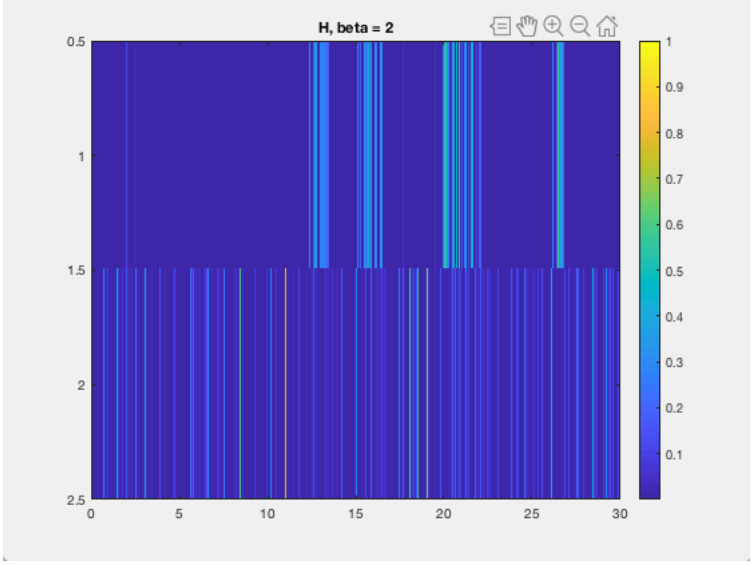
factor W are related to the frequency contents of S1 and S2. While the top row of H is able to precisely highlight the time intervals we are interested in detecting, the bottom one appears to be slightly less precise. This is due to the fact that some basis functions were activated to approximate columns of the spectrogram associated to time intervals that did not contain heart sounds, but had relevant weight nonetheless. Since these imperfections can be observed in proximity of actual heart sounds, a possible explanation is that the NMF is trying to describe the leftover murmur generated by either S1 or S2. These minor sounds have less relative weight if compared to the peak of their corresponding cardiac event, hence their activation coefficients have significantly smaller norms. That being said, we should notice that the algorithm was not able to differentiate between the two sounds, since both basis functions are activated to describe both S1 and S2. We can therefore assume that, in this case, the frequency contents of the two main heart sounds were not sufficiently different to incentivize the creation of two dedicated basis functions.

3.2.3 Presence of noise in the PCG

In the previous section we briefly analyzed the effect of noise on the performance of MU-NMF(β). We will now further explore this subject by shifting our attention to the nonnegative factors and, in particular, to the coefficients' matrix H. In the figure below we include the two factors generated by MU-NMF(2) applied to the spectrogram associated to the power spectral density of Figure (3.5):



(a)



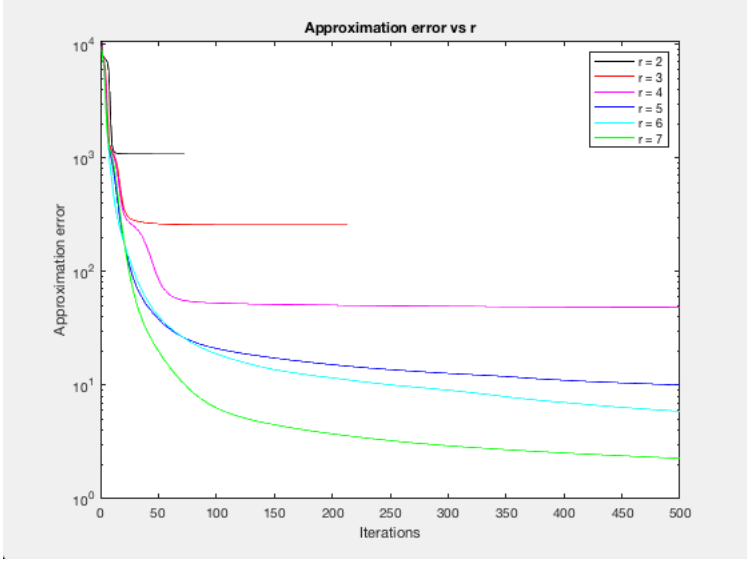
(b)

Figure 3.8: *Left factor W (a) and right factor H (b) generated by MU-NMF(2) applied to the spectrogram of signal "PCG0004" from Database1 with overlap $p = 0$, frequency band [50:150]Hz and factorization rank $r = 2$.*

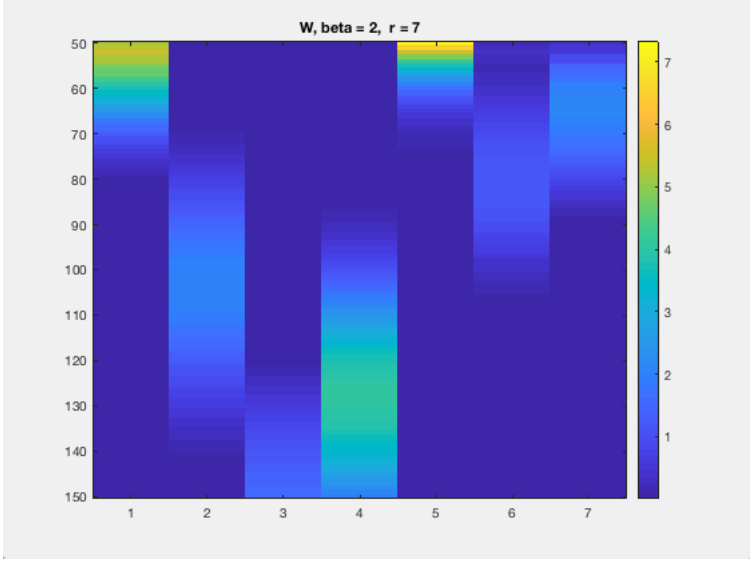
As we can clearly see the noise has compromised the first of the two basis functions. Indeed the latter has a peak frequency content around 120Hz, which cannot be tied back to neither S1 nor S2 since their peak frequencies rarely reach over 100Hz (see Section 3.2.5 for the complete time-frequency analysis). Supporting this interpretation is the fact that the first basis function is activated in correspondence of time intervals containing noise, as we can conclude by comparing the first row of H with the power spectral density of Figure (3.5). As far as the second basis function is concerned, we can reasonably assume it is related to the frequency contents of heart sounds, since at times preceding the noise we can discern a somewhat time-periodic activation of the latter from the second row of H, although the quality of the representation is arguably inferior to that of the low-level noise case presented in Figure (3.7). This loss of precision is due to the fact that the NMF has tried to represent heart sounds, which may have considerable variance in frequency contents, using only one basis function, as the first one was entirely allocated to describe noise. In this example, the presence of noise has almost completely destroyed the learning and heart sound detection capabilities of the NMF.

A possible solution to the above problem is increasing the factorization rank

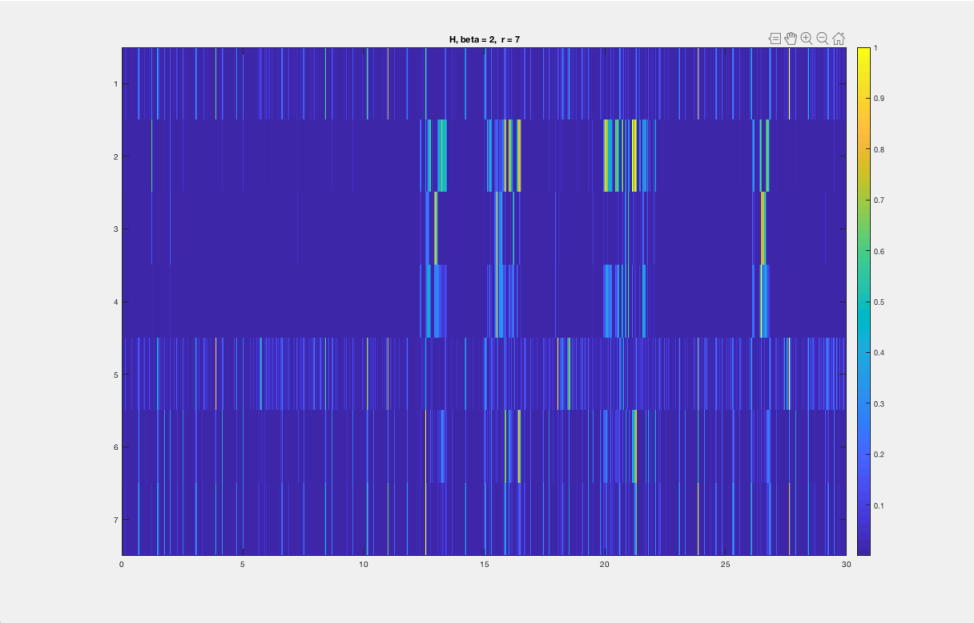
r, giving the NMF more basis functions to approximate the data. Recalling the clustering interpretation provided in the previous sections, the noise (based on its frequency contents) can be seen as extra clusters that need to be taken into account during the modelling process. We will now analyze how the factorization rank affects the basis functions when applying MU-NMF(2) to the same signal.



(a)



(b)



(c)

Figure 3.9: *MU-NMF(2) approximation error (a) for different values of* $r \in \{2, 3, 4, 5, 6, 7\}$*, left factor* W *(b) and right factor* H *(c) generated by MU-NMF(2) applied to the spectrogram of signal "PCG0004" from Database1 with overlap* $p = 0$*, frequency band [50:150]Hz and factorization rank* $r = 7$*.*

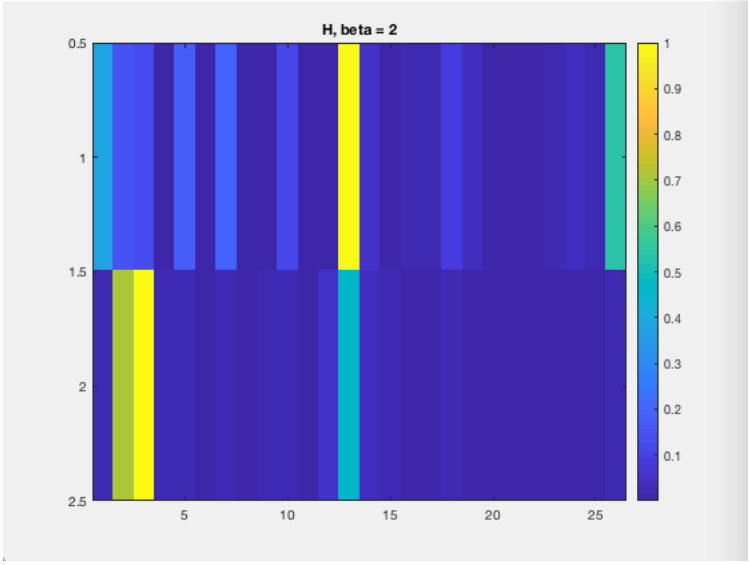
As expected, Figure (3.9) shows how the approximation error decreases as the factorization rank r increases. As for the factors, we choose to show the results for $r = 7$: just like in the previous case, the algorithm allocated some columns of W for the description of noise (columns 2, 3 and 4), while the others are used to model heart sounds and possibly some related murmurs (columns 1, 5, 6 and 7). When compared to the one in Figure (3.8), some of the rows of H associated to heart sounds show some significant improvement in quality. Indeed, in rows 6 and 7 we can clearly recognize the time periodicity related to S1 and S2, although in correspondence to time intervals containing noise, some imperfections can be observed. Amongst the noise, the NMF activated most of the basis functions at its disposal in order to better approximate it and therefore reduce the error: the trade-off is a loss of accuracy when modelling the heart sounds. This behavior suggests, in particular, that the noise at hand and heart sounds share a significant portion of their frequency contents.
Summing up what we deduced in this section:

Remark 3.5. *If in presence of noise, increasing the factorization rank* r

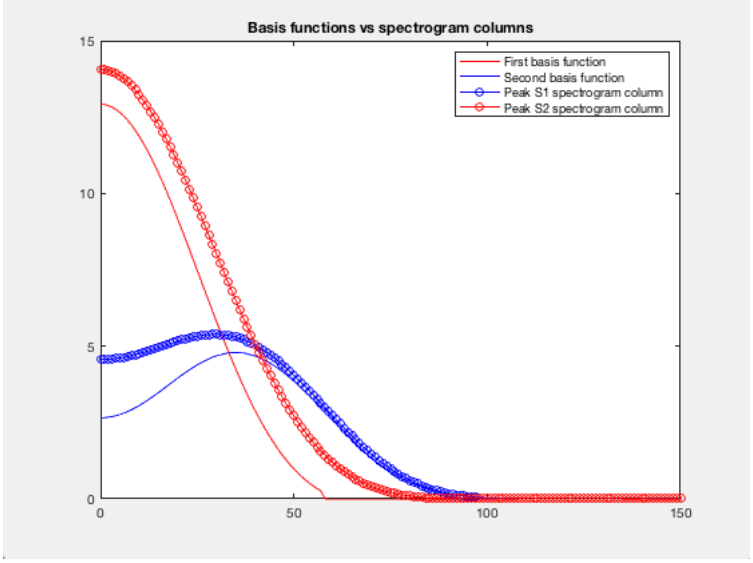
allows the MU-NMF(2) algorithm to lower the approximation error and generate cleaner factors. Nevertheless, if the noise and heart sounds share a portion of their frequency contents, the basis functions associated to the latter may be activated to describe the former, leading to a loss of accuracy.

3.2.4 Basis function analysis

Let us now analyze the choice of the basis functions made by the NMF in absence of ambient noise. As we noticed in Figure (3.7), for example, the algorithm was not separating S1 and S2, but was rather creating two basis functions and activating both in correspondence of both sounds. In order to study in more detail the relation between basis functions and spectrogram columns, we shall apply the NMF to single hand-picked cardiac cycles spectrograms. More precisely, the cardiac cycles will begin with an S1 sound and end right before the next; this way we will be able to observe the features of both S1 and S2. Furthermore, we will consider the broader frequency band [0:150]Hz in order to provide a more complete description of the frequency contents of the heart sounds.

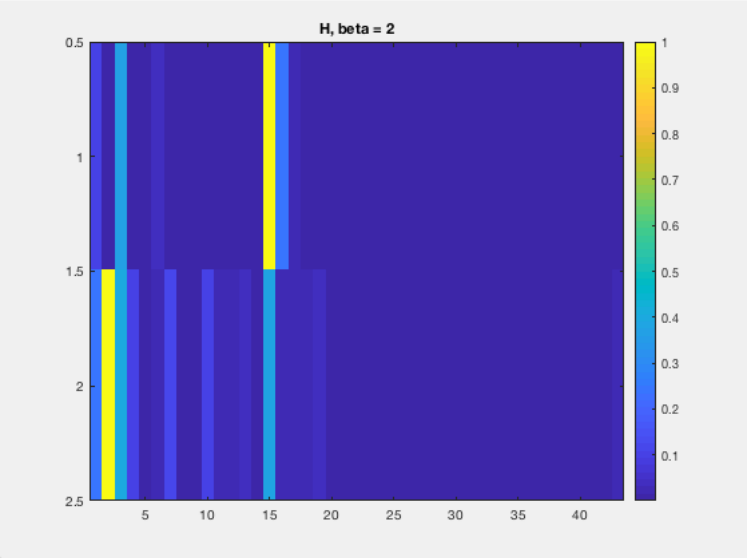


(a)

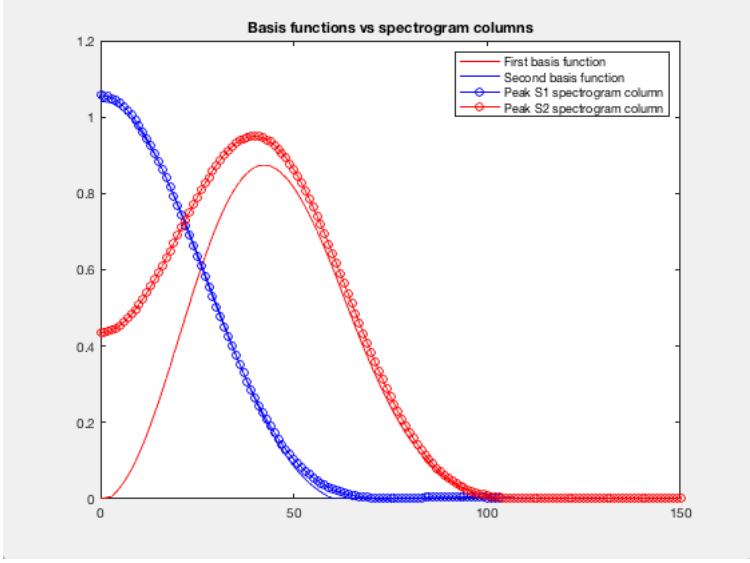


(b)

Figure 3.10: *Right factor H (a) and comparison between basis functions and heart sounds (b) generated by MU-NMF(2) applied to a cardiac cycle spectrogram of signal "PCG0003" from Database1 with overlap* $p = 0$*, frequency band [0:150]Hz and factorization rank* $r = 2$*.*



(a)



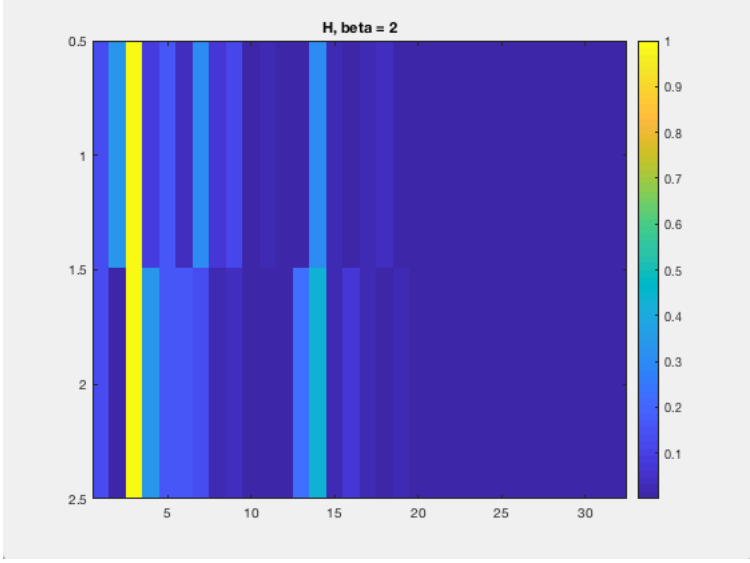
(b)

Figure 3.11: *Right factor H (a) and comparison between basis functions and heart sounds (b) generated by MU-NMF(2) applied to a cardiac cycle spectrogram of signal "PCG0010" from Database1 with overlap $p = 0$, frequency band [0:150]Hz and factorization rank $r = 2$.*

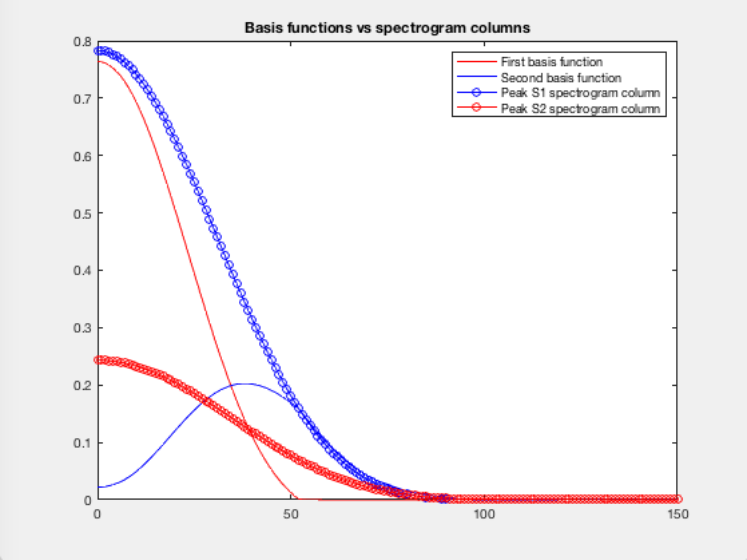
In Figure (3.10) we see how focusing on a single heart cycle allowed the NMF to differentiate between S1 and S2: the first basis function was used to describe the latter while the second one to describe the former. As a matter of fact the two sounds appear to have very distinct frequency contents. While S1 mainly activates frequencies in the band [0:100]Hz with a peak around 40Hz, S2 activates frequencies in the band [0:80]Hz with a peak of much bigger weight than the former. Similarly, in Figure (3.11) the NMF was also able to differentiate between the two, but the frequency contents now appear to be swapped: S1 activates frequencies in the lower band [0:60]Hz while S2 in the upper band [0:100]Hz and the peaks have comparable weights.
We should stress that these heart cycles were selected from different signals. In general:

Remark 3.6. *Different PCG signals can be characterized by heart sounds of quite contrasting frequency contents, although within a single fixed PCG most heart sounds show comparable behavior.*

That being said, even in the latter case, some variance can be observed: we will explore this matter in the next section.



(a)



(b)

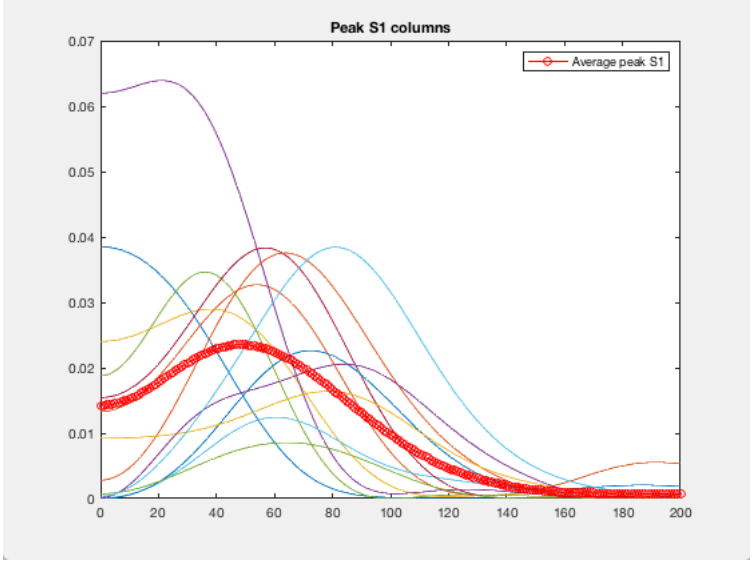
Figure 3.12: *Right factor H (a) and comparison between basis functions and heart sounds (b) generated by MU-NMF(2) applied to a cardiac cycle spectrogram of signal "PCG0012" from Database1 with overlap $p = 0$, frequency band [0:150]Hz and factorization rank $r = 2$.*

Lastly, in Figure (3.12) above we included a case in which the frequency contents of the two main heart sounds were not sufficiently different for the NMF to create dedicated basis functions. Indeed we can notice that the

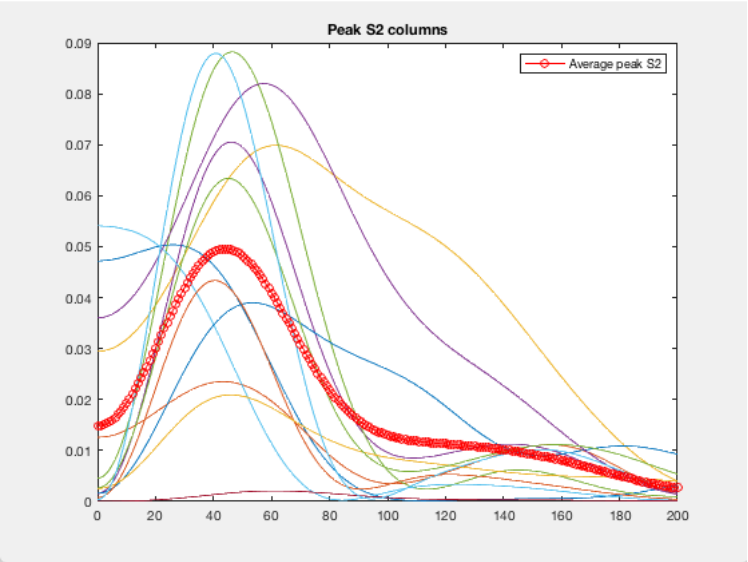
columns of the spectrogram associated to the peaks of S1 and S2 differ from one another by a positive factor. The algorithm thus generated two basis functions based on columns of the spectrogram with significantly different frequency contents, and activated both to describe both heart sounds.

3.2.5 Time-frequency analysis of S1 and S2 sounds

In this section we intend to carry out an in depth time-frequency analysis of both main heart sounds S1 and S2. The need for such an investigation follows from the quite substantial variance that can be observed among peak heart sounds spectrogram columns, even within the same PCG signal. In Figure (3.13) below we provide an example of such behavior. This study, in particular, will help us better understand the limitations of the MU-NMF(β) algorithm and it will lay the foundations upon which to refine the latter.



(a)



(b)

Figure 3.13: *Peak S1 spectrogram columns (a) and peak S2 spectrogram columns (b) of signal "PCG50699_AV" from Database2 with overlap* $p = 0$ *and frequency band [0:200]Hz.*

There are mainly two possible ways to tackle the analysis of such complex signals:

- *Model-driven analysis.* From an anatomical point of view, it is well known that the heart sounds S1 and S2 are generated by the closure of particular heart valves. Such movement, combined with complex hemodynamics of the blood on both sides of the valve, gives rise to vibrations which propagate as sounds waves through the chest cavity. These sound waves, when recorded with external instruments such as an electronic stethoscope, are what we call heart sounds, and based on the valve that initiated the latter, we may distinguish between S1 and S2. As a consequence, in order to analyze heart sounds, one may construct a first physical-mathematical model to describe the vibrations, and then a second one to characterize how the latter propagate within the thorax. With such models at hand, it would then be possible to train them on real data and use the resulting model-sounds to facilitate the learning process of the NMF. The difficulty in this approach is directly correlated to the complexity of the phenomena we are dealing with. As a matter of fact, modelling the vibrations requires a detailed description of both the heart's anatomy (namely the structure of the valves and surrounding tissues, arteries' elastic properties and volume capacity, etc.) as well as a precise approximation of the hemodynamics that

take place around its valves (which in turn requires to take into consideration the forces applied to both sides of the valves, potential turbulent flow of the blood within the heart's cavities, etc.). Similarly, producing a model that describes how the resulting sound waves propagate, must take into account the different densities and resonance frequencies of the materials that compose the chest walls (bones, muscles, fat and skin).

- *Data-driven analysis.* Without a detailed underlying model of the heart sounds, in order to analyze the time-frequency properties of the latter, one may employ a more statistical, data-driven approach. Indeed, given a sufficiently large and varied database, the intrinsic properties of these signals can be inferred from the data itself. The knowledge of such properties, like average frequency-content-over-time or peak-frequency-content, which we briefly mentioned in the previous chapters, may provide useful insights on the hidden learning patterns of the NMF algorithms we employ. Moreover, having at our disposal a mean behavior for these sounds, can potentially offer a way of normalizing our data by excluding signal samples with too big of a deviation; for example, if we knew the average peak-frequency-content for both sounds, we would be able to recognize the presence of noise in the data by looking at any unusual peak frequency outside the established band.

Given our interest in exploiting the learning and generalization capabilities of the NMF, the latter approach is the one we choose to employ.

Let us start by providing an explanation for the variation that can be observed between peak heart sounds spectrogram columns. It is of key importance to recall that, so far, all spectrograms have been computed with nonoverlapping time intervals ($p = 0$). By combining this parametrization choice with the fact that both S1 and S2 may have a *time-dependent frequency content*, we conclude that:

Remark 3.7. *The aforementioned variation derives from the relative shift of these sounds with respect to our fixed time intervals.*

Consider for example an heart sound of duration 50ms: since the length of the window function's local support used to compute the STFT is $\delta T = 25$ms, the former may fall within our time sampling in an intermediate position between the following two extremes:

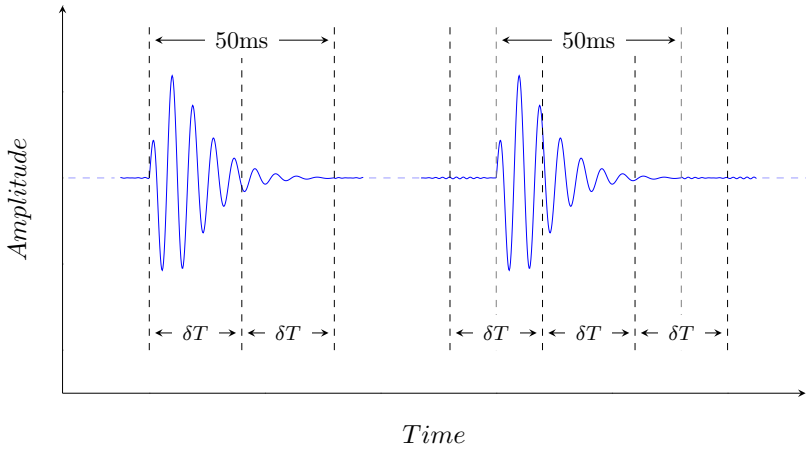
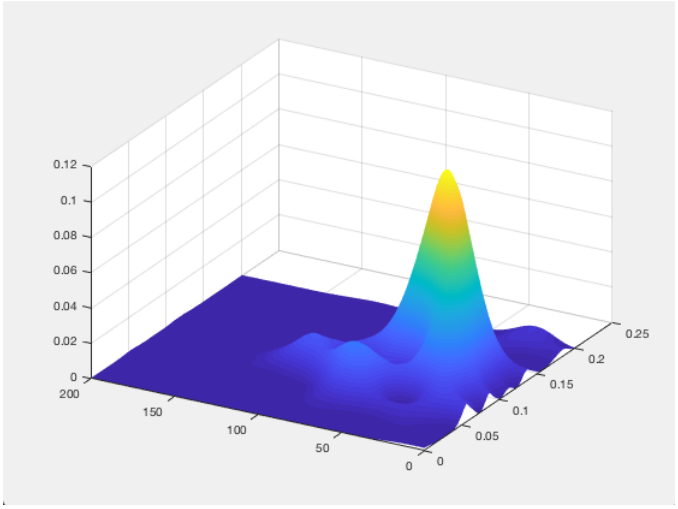
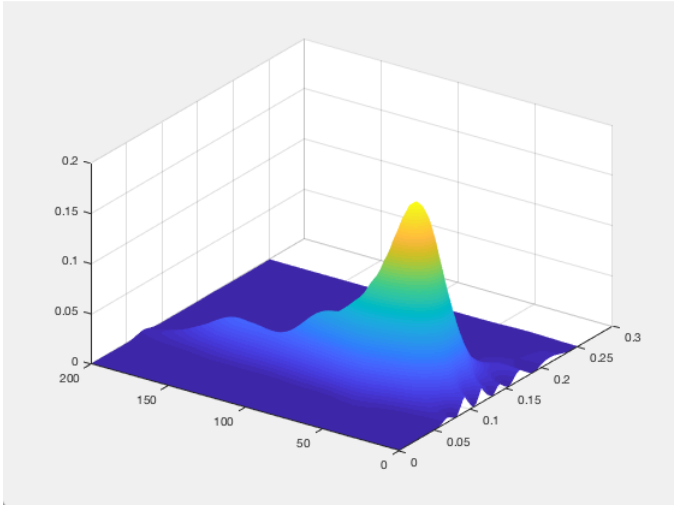


Figure 3.14: *The two possible extremes in relative shift between an heart sound of duration 50ms and STFT local supports.*

While in the first case the sound will be entirely contained in only 2 spectrogram columns, in the second case 3 columns will be necessary. Moreover, due to the time-dependent frequency content, all 5 of these columns will be different from one another.
As a consequence, in order to obtain a more encompassing description in the time-frequency domain, it will be useful to consider spectrograms computed with overlapping supports. The bigger the overlap percentage p, the better we will be able to analyze how the frequency content of these sounds changes over time. In the following figure we include the spectrograms, computed with overlap $p = 0.95$, of an S1 and S2 sound:



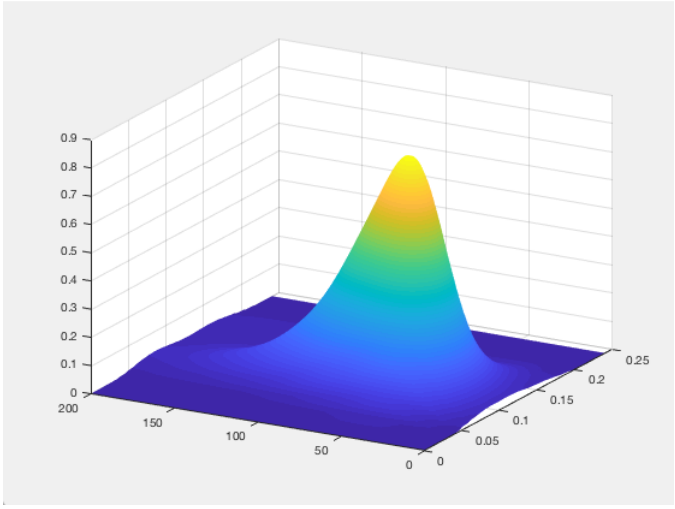
(a)



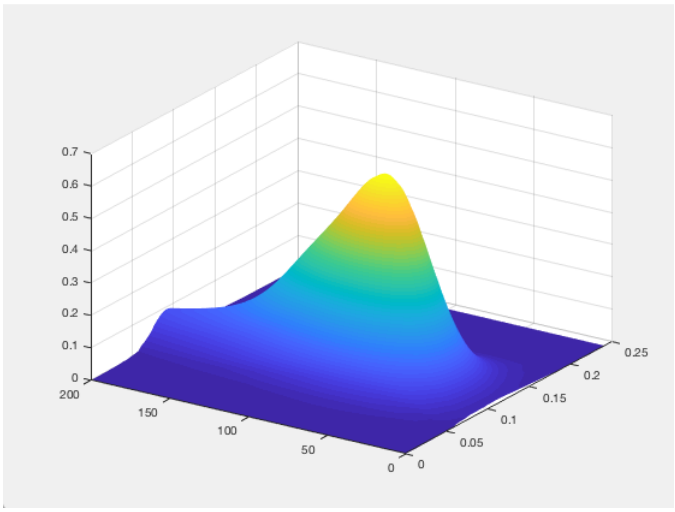
(b)

Figure 3.15: *Spectrogram of an S1 sound (a) and spectrogram of an S2 sound (b) of signal "PCG50699_AV" from Database2 with overlap* $p = 0.95$ *and frequency band [0:200]Hz.*

As we can observe, the frequency contents of both sounds change quite significantly during their evolution, although a distinct 'bell-shape' containing the characterizing peak frequency can be identified in both spectrograms. To reiterate, the variance present in Figure (3.13), which contained columns of a spectrogram computed with $p = 0$, derives from selecting different time sections from the above spectrogram computed with $p = 0.95$. In other words, we can think of the latter as an (over) complete dictionary that is able to fully encode the frequency content of the corresponding heart sound.
Moreover, from Figure (3.15) we can also notice some major differences between S1 and S2. Indeed, while S1 activates frequencies in the band [0:120]Hz with a peak around 40Hz, S2 activates also higher frequencies, namely those in the band [0:200]Hz, and has a slightly higher peak at around 50Hz.
These two properties, that is, S2 activating higher frequencies and being characterized by a higher frequency peak when compared to S1, are actually intrinsic to the heart sounds in question. A possible way to verify this assertion is to exploit the large number of data contained in Database2 and compute an average spectrogram for both S1 and S2 by adding together all respective spectrograms for all the PCGs in the database.



(a)



(b)

Figure 3.16: *Average spectrogram of an S1 sound (a) and average spectrogram of an S2 sound (b) from Database2 with overlap* $p = 0.95$ *and frequency band [0:200]Hz.*

Not only does Figure (3.16) support the previous assertion, but it also shows how the variance around the bell-shaped curve present in the particular example of Figure (3.15) vanishes. We can therefore conclude that:

Remark 3.8. *Apart from minor local variations, the average frequency content of an heart sound can be described in the time-frequency domain as a bell-shaped curve whose peak frequency and band differ between S1 and S2. In particular, these differences can be used to identify one from the other.*

Chapter 4

NMF-based heart rate detection

The study conducted in the previous chapter on the properties and potential obstacles of phonocardiograms, as well as on the required computational tools, will now be brought to fruition. Indeed we shall contextualize the use of STFT and NMF with the purpose of identifying heart sounds within a more complete algorithmic scheme. This will be done in Section 4.1. In particular, a substantial amount of emphasis will be given to the process of denoising PCGs since, as seen in Section 3.2.3, the presence of noise often hinders the identification of heart sounds by the NMF. In Section 4.2 we will therefore explore some denoising options and develop a NMF noise canceler based on adaptive noise cancellation.

4.1 General scheme

Let us now discuss in more details how the study carried out in the previous chapter will allow us to design a NMF-based heart rate detection algorithm. Starting from a PCG signal, we saw how to correctly choose the STFT parameters in order to obtain a complete time-dependent representation of the heart sounds' frequency contents. Subsequently, by considering the spectrogram, we generated a nonnegative matrix to feed into MU-NMF(β) and argued about the optimal choice of the β parameter. The NMF then produced two factors W and H, the former containing *some* basis functions related to heart sounds and the latter containing the activation coefficients for each time interval. In order to finally detect time intervals containing heart sounds, we can apply a Window Search routine to one of the rows of H associated to the latter. Now, if the considered PCG is not affected by noise,

the results obtained in the previous chapter indicate that the Window Search routine should indeed perform as expected and only detect heart sounds. On the other hand, if the PCG contains noise, the rows of H become less clean and the aforementioned routine may fail its task.

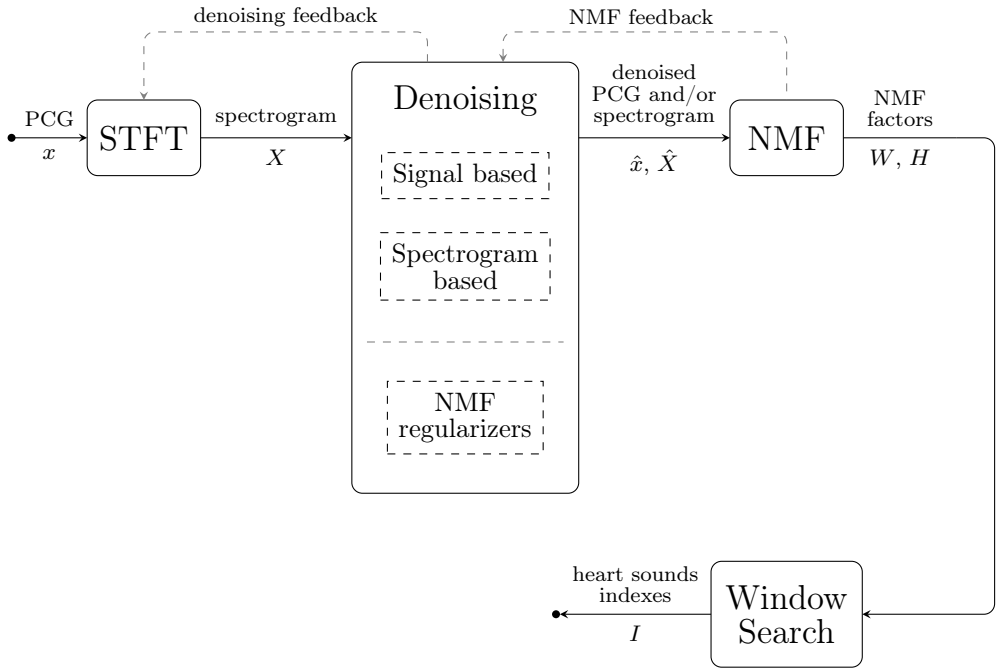


Figure 4.1: *State machine representation for the NMF-based heart rate detection algorithm. All operations involving dashed lines are optional.*

As a consequence, taking as reference Figure (4.1), in the following procedure we shall focus our attention on a pivotal aspect of the detection algorithm, namely PCG denoising:

- *Step 1:* STFT. Given a PCG signal x, we calculate its STFT with parameters $\delta T = 25$ms, $p = 0.95$ and frequency band [0:200]Hz. Indeed in Section 3.2.5 we saw how these choices led to a more accurate representation of S1 and S2 sounds. Subsequently, we compute the spectrogram $X = |STFT(x)|^2$;
- *Step 2:* Denoising. Based on the quality of the gathered data, we can choose to apply some denoising routines. More precisely, we may distinguish between three categories of denoising:
  (a) Signal based. These routines are applied directly to the PCG x and

produce a denoised signal $\hat{x}$. It is therefore necessary recompute the STFT and the associated spectrogram $\hat{X}$ (*denoising feedback*);

(b) Spectrogram based. The signal is denoised through the manipulation of its spectrogram. The output of such routines is a denoised spectrogram matrix $\hat{X}$;

(c) NMF regularizers. In our framework, an alternative way of denoising the data is by choosing some regularizers with which to modify the standard algorithm MU-NMF(β) at the next step. The idea being that certain families of regularizers may aid in producing cleaner nonnegative factors.

- *Step 3:* NMF. The algorithm MU-NMF(2) (or some variant) is applied to the resulting spectrogram of the previous step, producing two nonnegative factors W and H. Indeed in Section 3.2.2 we saw how choosing $\beta = 2$ is a good compromise between clean and noisy PCGs when it comes to reducing the approximation error. As far as the factorization rank r is concerned, again we concluded that the latter should be increased if in presence of noise. The choice is therefore case-dependent. Moreover, as we will see in the next section, the NMF factors can be exploited to design denoising routines. As a consequence, we may feed the NMF output back to the denoising stage (*NMF feedback*);

- *Step 4:* Window Search. After having selected a row of H associated to heart sounds, we apply to it a Window Search algorithm. The output will be a set of indexes I corresponding to the detected heart sounds.

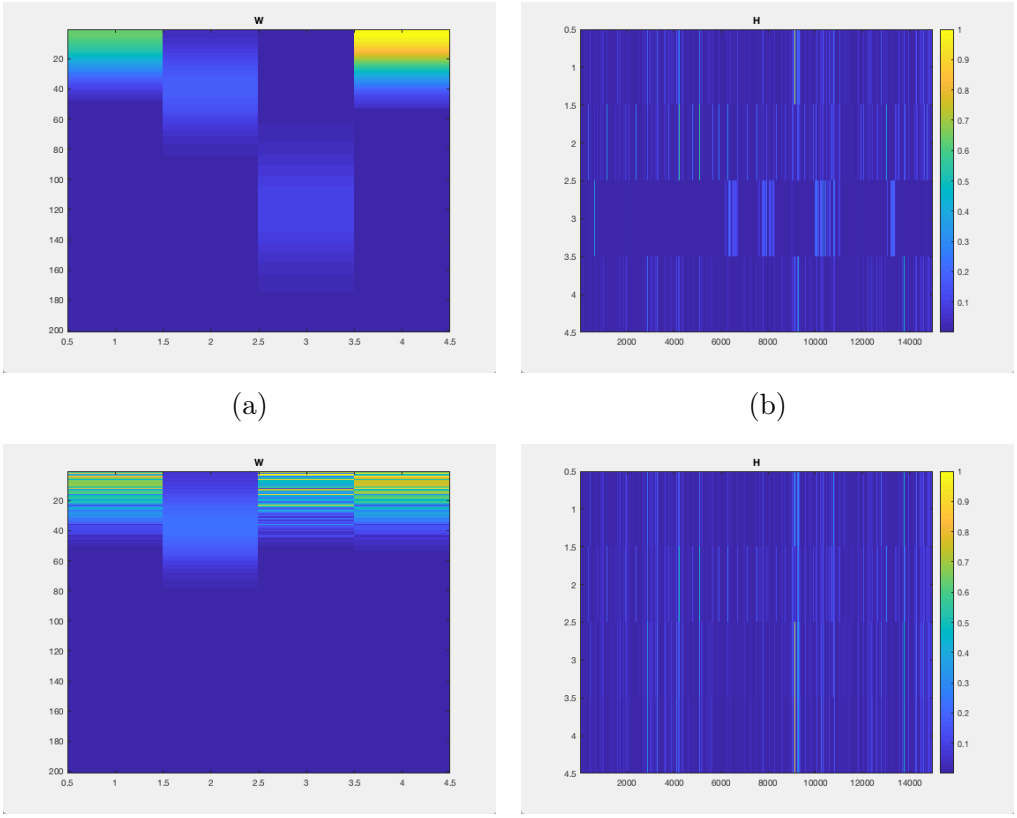
4.2 PCG denoising

Before presenting the denoising options we developed, it should be emphasised how the latter differ from the one employed in the traditional heart rate detection algorithm of Section 3.1.3. Indeed, in such algorithm, the first step involving PCG signals requires the application of a band-pass filter to isolate the [10:100]Hz frequency interval (see *Step 2.1*). Since we now intend to apply NMF to the spectrogram matrix of the signal, such a process would be counterproductive, *even if we were to know the exact frequency band activated by heart sounds in the specific PCG at hand.* The reason for this is at least twofold. First of all, band-passing the PCG reduces the modelling options at the disposal of the NMF, since all basis functions would then be restricted to a shorter frequency interval. Secondly, in the presence of noise with a frequency content overlapping that of heart sounds, the NMF

may not have enough information to distinguish between clean spectrogram columns containing cardiac events and corrupted columns containing noise, hence activating the same basis function for both. See Figure (4.2) for an exemplification of this issue. As a consequence, in our framework:

Remark 4.1. *Band-passing the PCG signal limits our options when it comes to noise recognition and the subsequent denoising process.*

As we will see in Section 4.2.5, all information regarding the frequency content of noise extrapolated from the NMF factors will be of key importance for the development of a NMF-based noise canceler.



(a) (b)

(c) (d)

Figure 4.2: *NMF factors generated by MU-NMF(2) applied to the spectrogram of signal "PCG0004" from Database 1. For the top factors (a) and (b) the signal was not band passed and we can see the different frequency contents of heart sounds (second basis function) as well as noise (first, third and fourth). For the bottom factors (c) and (d) the signal was first band-passed on the interval [0:100]Hz, resulting in a loss of information about the noise and an overall worse representation of both basis functions and detected cardiac events (second row of H).*

4.2.1 Norm Clipping

In a NMF spectrogram-based learning, high-energy noise impulses are among the most disruptive types of noise that can affect the chosen dataset. Indeed, in the time-frequency domain, such impulses are associated to spectrogram columns with norm that overshadows that of the clean features we are interested in detecting. As a consequence, in the case of PCGs, any NMF algorithm will prioritize the creation of basis functions to describe these corrupted spectrogram columns instead of the ones containing cardiac events. A possible way of addressing this concern, is to manipulate the spectrogram matrix and remove (clip) all such problematic columns. More precisely, given a column-wise description of a spectrogram matrix $X = \{X_j\}_{j=1,\dots,N}$ and a set of time indexes $J \subset \{1, \dots, N\}$ associated to noisy columns, a Norm Clipping (NC) denoising routine produces a matrix $\hat{X} = \{\hat{X}_j\}_{j=1,\dots,N}$ where:

$$\hat{X}_j = \begin{cases} 0 & \text{if } j \in J, \\ X_j & \text{if } j \notin J. \end{cases}$$

In particular, assuming high-energy impulses to be characterized by large 2-norm columns within the STFT matrix of the signal, the index set J can be obtained by thresholding the 1-norm of the spectrogram columns. As a consequence, the complete NC denoising routine can be summarized as in the following Algorithm (4.2.1):

Algorithm 4.2.1 Norm Clipping

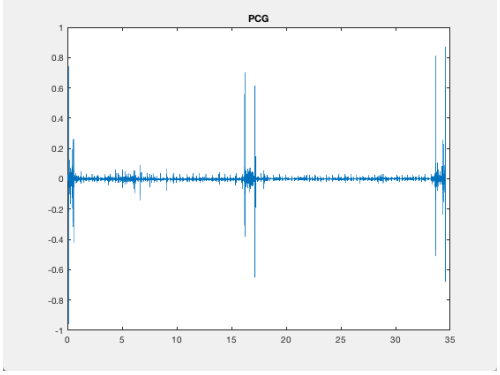
Input: A spectrogram $X = \{X_j\}_{j=1,\dots,N}$ and threshold M.

Output: A denoised spectrogram $\hat{X} = \{\hat{X}_j\}_{j=1,\dots,N}$.

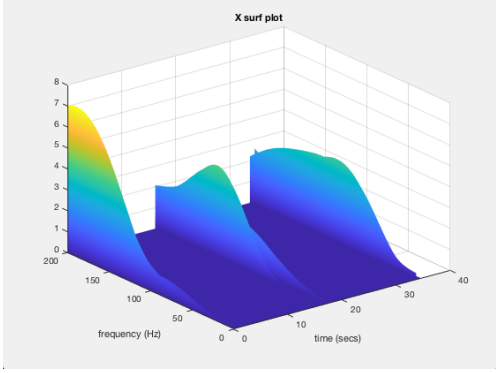
```
Initialize X̂ = X
for j = 1, ..., N  do:
    if ‖X_j‖_1 > M
        X̂_j = 0
    end if
end for
```

As highlighted in Figure (4.3), after the application of the NC routine the columns of the spectrogram affected by impulsive noise have been removed,

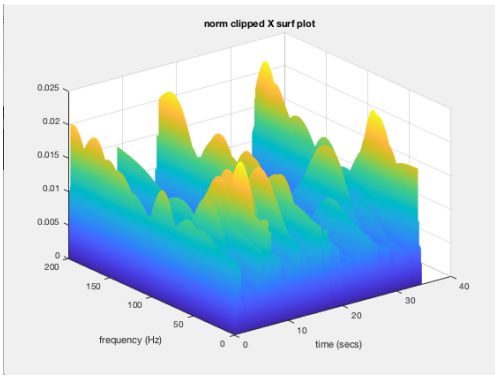
thus uncovering the clean frequency contents of cardiac events, characterized by smaller norms. It should be noted, however, that some higher frequencies correlated to the noise are still present.



(a)



(b)



(c)

Figure 4.3: *"PCG49577_MV" from Database2 (a). Surf plot for the unclipped (b) and Norm Clipped (c) spectrogram of the signal using a threshold* $M = 1.35$.

4.2.2 Weighted Time Compression

Cardiac events are often followed or preceded by low-energy murmurs with which they share most of their frequency content. These minor disturbances cause NMF algorithms to activate clean basis functions during time intervals not containing the main events we are interested in detecting, thus producing contaminated right nonnegative factors. In the following pages we will present a spectrogram-based denoising technique that aims at reducing the influence of such murmurs and, at the same time, highlighting the columns

containing cardiac events.
In Section 3.2.5 we remarked how the spectrograms of S1 and S2 sounds appear as bell-shaped curves and, in particular, how taking the average over multiple sounds produced spectrograms without any local fluctuation. This study supports the idea that by applying a Weighted Time Compression (WTC) to the columns of a spectrogram we would be able to both normalize the latter, aiding the creation of appropriate basis functions in the NMF phase of the detection algorithm, as well as remove any unwanted minor fluctuation, hence creating cleaner right nonnegative factors. More concretely, given a column-wise description of a spectrogram matrix $X = \{X_j\}_{j=1,\dots,N}$ and a compression factor of c columns, the output of the WTC is a spectrogram $\hat{X} = \{\hat{X}_j\}_{j=1,\dots,\lfloor \frac{N}{c} \rfloor}$ given by:

$$\hat{X}_j = \frac{1}{w_j} \sum_{i \in I_j} X_i$$

$$w_j = \begin{cases} |E_j| & \text{if } E_j \neq \varnothing, \\ c & \text{if } E_j = \varnothing. \end{cases}$$

where $E_j \subset I_j = \{(j-1)c+1, \dots, jc\}$ is a set of the time indexes associated to columns containing cardiac events within the j-th interval of compression I_j. Clearly E_j cannot be known a priori, so in order to develop an algorithm for the WTC, we need a way to estimate how many indexes in I_j are actually related to cardiac events. One possible approach, recalling that our NMF algorithm of choice will be MU-NMF(2), is to threshold the 2-norm-squared of the columns to be compressed by a value M. In particular, this value should be chosen in such a way that only columns associated to the peaks of S1 and S2 sounds will pass the threshold.
With this in mind, let us consider a clean spectrogram $X^{cyc} = \{X^{cyc}_j\}_{j=1,\dots,n}$ of a single cardiac cycle, computed, as usual, with an overlap of $p = 0.95$ and define M_1, M_2 as the 2-norm-squared of peak S1, S2 sound columns respectively. From Figure (3.16) we know that, on average, S2 sounds are characterized by a bell-shaped curve with a slightly lower peak than S1 sounds. As a consequence, our threshold M should be fixed to the 2-norm-squared of peak S2 sound columns, that is:

$$M = M_2 \tag{4.1}$$

Furthermore, from the same Figure, we can observe that the ratio r between S1 and S2 peaks can be approximated, on average, as $r = \frac{9}{7}$. In particular, this allows us to estimate the 2-norm-squared of peak S1 sound columns:

$$M_1 = r^2 M_2 \tag{4.2}$$

Now, since peak cardiac sounds both have a duration of around 5% that of the cycle, letting $q = \frac{1}{20}$ we can expect qn columns of X^{cyc} to be allocated to each main sound, while the remaining ones will have negligible 2-norm. By defining the 2-norm-squared mean of the cycle:

$$m^{cyc} = \frac{1}{n} \sum_{j=1}^{n} \|X_j^{cyc}\|_2^2 \tag{4.3}$$

we can approximate the same quantity as:

$$m^{cyc} \approx \frac{1}{n}(qnM_1 + qnM_2) \tag{4.4}$$

Expanding (4.4) using (4.1) and (4.2) yields:

$$m^{cyc} \approx q(r^2 + 1)M = \alpha M \tag{4.5}$$

where $\alpha = \frac{13}{98}$. Equivalently, given m^{cyc}, we get the threshold:

$$M \approx \frac{1}{\alpha} m^{cyc} \tag{4.6}$$

In conclusion, since we can reasonably assume the 2-norm-squared mean to be independent from the number of considered cardiac cycles by the quasi-periodic nature of a PCG's complete spectrogram $X = \{X_j\}_{j=1,\dots,N}$, we may obtain the sought after threshold through the following relations:

$$\begin{aligned} M &\approx \frac{1}{\alpha} m \\ m &= \frac{1}{N} \sum_{j=1}^{N} \|X_j\|_2^2 \end{aligned} \tag{4.7}$$

The estimated set of time indexes $\hat{E}_j$ will therefore be defined as:

$$\hat{E}_j = \{i \in I_j : \|X_i\|_2^2 > M\} \tag{4.8}$$

The following Algorithm (4.2.2) contains a schematic representation of the WTC denoising routine:

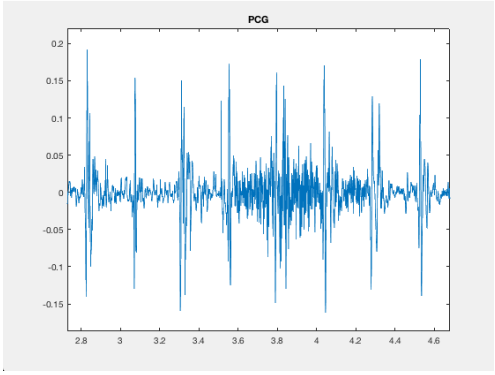
Algorithm 4.2.2 Weighted Time Compression

Input: A spectrogram $X = \{X_j\}_{j=1,\dots,N}$ and compression factor c.

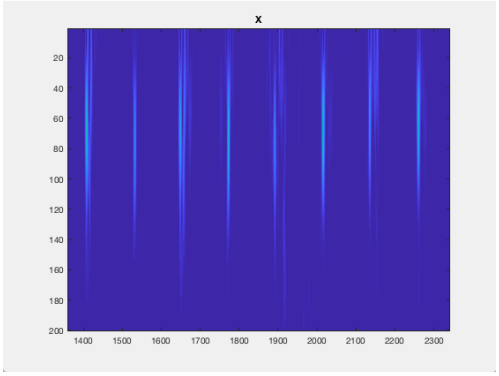
Output: A denoised spectrogram $\hat{X} = \{\hat{X}_j\}_{j=1,\dots,\lfloor \frac{N}{c} \rfloor}$.

$M = \frac{98}{13N} \sum_{j=1}^{N} \|X_j\|_2^2$
for $j = 1, \dots, \lfloor \frac{N}{c} \rfloor$ **do:**
    $X^{temp} = 0$
    $w^{temp} = 0$
    check = `false`
    for $i = 1, \dots, c$ **do:**
        $X^{curr} = X_{(j-1)c+i}$
        $X^{temp} = X^{temp} + X^{curr}$
        if $\|X^{curr}\|_2^2 > M$
            $w^{temp} = w^{temp} + 1$
            check = `true`
        end if
    end for
    if check
        $w = w^{temp}$
    else
        $w = c$
    end if
    $\hat{X}_j = \frac{1}{w} X^{temp}$
end for

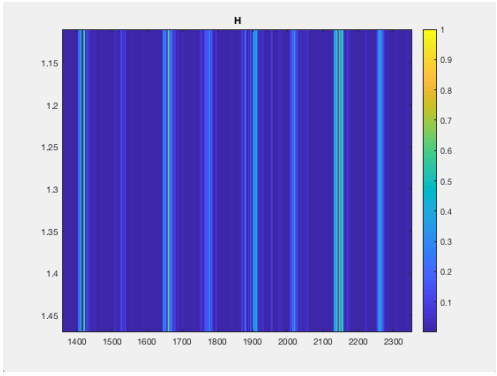
The effect of the WTC routine on a PCG containing low-energy murmurs can be appreciated in Figure (4.4). The murmurs affecting the second and third heart beat within the provided zoom-in of Figure (4.4a) have been successfully attenuated, as evidenced by the absence of imperfections within the activation coefficients' matrix of Figure (4.4e).



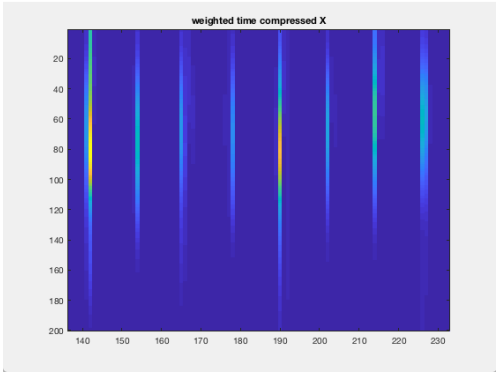
(a)



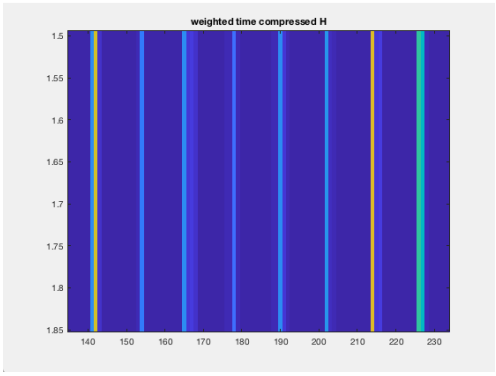
(b)



(c)



(d)



(e)

Figure 4.4: *Zoom-in on low-energy murmurs within "PCG85315_PV" from Database2 (a). Zoom-in on the affected columns of the uncompressed (b) and Weighted Time Compressed (d) spectrogram of the signal using a compression factor* $c = 10$*. Zoom-in on the affected coefficients of the right factors generated by MU-NMF(2) applied to the uncompressed (c) and Weighted Time Compressed (e) spectrogram.*

Before moving on to the following denoising routine, it should be emphasized that the choice of the compression factor c should be done taking into account the overlap percentage p with which the spectrogram is computed. Indeed, as we remarked in Section 3.2.1, the latter controls how many columns of the spectrogram will be allocated to each cardiac sound, hence the role of the former is reducing such quantity. Furthermore, since our end goal is detecting cardiac events from the rows of H, in that regard, applying a time-wise compression to the spectrogram corresponds to a loss of precision. As a consequence, should the desired results require a high-level of precision, the compression factor must be chosen as small as possible to attenuate most murmurs and at the same time not compromise the outcome.

4.2.3 Sparse NMF

As we will now discuss, a possible way of dealing with the influence of low-energy murmurs on the quality of the right nonnegative factors generated by the NMF, is to modify the algorithm MU-NMF(2) itself. As a matter of fact, such disturbances appear within the matrix H as minor activation coefficients corresponding to time intervals not containing any cardiac event. The difference in relative weight between these coefficients and the ones actually identifying cardiac events is often quite noticeable, and it suggests that a sparse version of the algorithm applied so far could yield decent results. To clarify, with *sparse version* we intend an algorithm derived from the modified objective:

$$F(X; W, H) = D_2(X, WH) + \lambda \|H\|_1 \tag{4.9}$$

which adds a 1-norm regularizer on the right factor H through a penalty (see Section 2.1.2 on NMF regularization). By an appropriate choice of the penalty parameter $\lambda > 0$, we would then be able to incentivize the algorithm to *not* activate any basis function during time intervals containing murmurs, since putting to zero such coefficients produces a bigger decrease of the objective if compared to the decrease followed by better approximating the corresponding columns of the spectrogram X.
As shown in the work by Févotte et al. [9], from (4.9) one may derive a closed-form multiplicative update rule, similar to the one presented in Theorem (2.4), for the values $\beta \notin (1, 2)$. Omitting the careful thresholding required to maintain the following quantities strictly positive, the general update rule for $\beta \geq 2$ is given by:

$$W^{(t)} = W^{(t-1)} \circ \left(\frac{\left[\left(\left(W^{(t-1)} H^{(t-1)} \right)^{\circ(\beta-2)} \circ X \right) H^{(t-1)^\top} \right]}{\left[\left(W^{(t-1)} H^{(t-1)} \right)^{\circ(\beta-1)} H^{(t-1)^\top} \right]} \right)^{\circ\gamma(\beta)}$$
$$H^{(t)} = H^{(t-1)} \circ \left(\frac{\left[W^{(t)^\top} \left(\left(W^{(t)} H^{(t-1)} \right)^{\circ(\beta-2)} \circ X \right) - \lambda O \right]}{\left[W^{(t)^\top} \left(W^{(t)} H^{(t-1)} \right)^{\circ(\beta-1)} \right]} \right)^{\circ\gamma(\beta)} \tag{4.10}$$

where $\gamma(\beta)$ is defined as in Theorem (2.4) and $O \in \mathcal{M}_{r\times n}(\mathbb{R}^+)$, $O(i,j) \equiv 1$. In particular, as remarked in Section 2.1.2, since the objective was modified by adding a regularizer for H, to prevent the convergence to degenerate stationary points the above update rule must be implemented alongside a normalization on the columns of W at each iteration.
Let us now analyze an ideal case to better understand the relation between the penalty parameter λ and the sparsification property of the 1-norm regularizer in the case $\beta = 2$. More precisely, let's consider a column-wise description of a spectrogram $X = \{X_j\}_{j=1,\dots,N}$ and, in particular, focus on one of its columns X_j. To simplify the computations, suppose the dictionary $W = \{W_k\}_{k=1,\dots,r}$ to be fixed and, furthermore, its first column to be precisely X_j, hence $W_1 = X_j$. In this framework, when it comes to approximating the j-th column of X, we can restrict ourselves to the following objective:

$$f(X_j; W_1, H(1,j)) = D_2(X_j, W_1 H(1,j)) + \lambda H(1,j) \tag{4.11}$$

since activating any other coefficient $H(k,j)$ for $k \neq 1$ would result in a worse approximation. Omitting the dependence of f from all quantities except $H(1,j) = x$ and recalling that $W_1 = X_j$, we get the following cost function:

$$f(x) = \frac{1}{2}\|X_j - X_j x\|_2^2 + \lambda x = \frac{1}{2}\|X_j\|_2^2 x^2 + (\lambda - \|X_j\|_2^2)x + \frac{1}{2}\|X_j\|_2^2 \tag{4.12}$$

The global minimum is therefore attained at:

$$\bar{x}_\lambda = 1 - \frac{\lambda}{\|X_j\|_2^2} \tag{4.13}$$

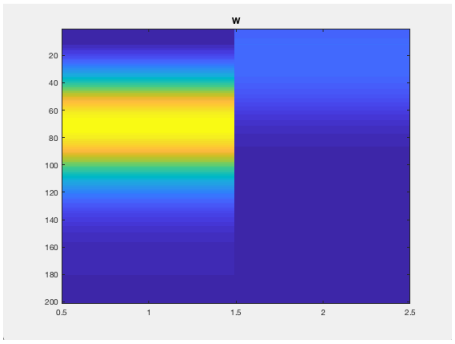
From this relation we notice, unsurprisingly, that the unsparsified algorithm ($\lambda = 0$) will activate the first basis function with a coefficient of precisely

$\bar{x}_0 = 1$. On the other hand, by adding the penalty term, such coefficient will be set to smaller and smaller values as λ increases. In particular, for all values $\lambda \geq \|X_j\|_2^2$, the activation coefficient will be $\bar{x}_\lambda = 0$ (recall the positive thresholding[5]) and the maximal sparsity on the j-th column of H is reached. In this ideal case, if we assume X_j to be a column containing some murmur we want to attenuate, we can fix a small $\delta > 0$ and set λ to any value in the interval:

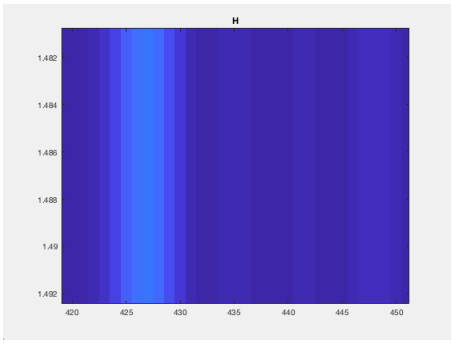
$$\left[\ \|X_j\|_2^2(1-\delta), \ \|X_j\|_2^2 \ \right]$$

in order to obtain an activation coefficient $\bar{x}_\lambda$ no greater than δ itself. To conclude, we should remark that all the above computations can also be derived, even in more generality, from the update rule for the H factor in Equation (4.10).

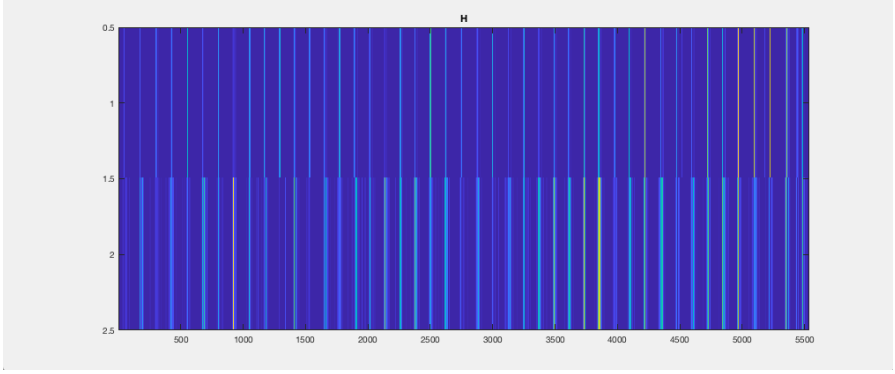
Going back to the denoising implications of the sparsified NMF algorithm, which we shall hereon define as MU-SNMF (Multiplicative Update Sparse NMF), in the following Figure (4.5) we can compare the factors generated by MU-NMF(2) and MU-SNMF(2) applied to the same spectrogram as the previous section, which, as already seen, contains some low-energy murmurs.



(a)

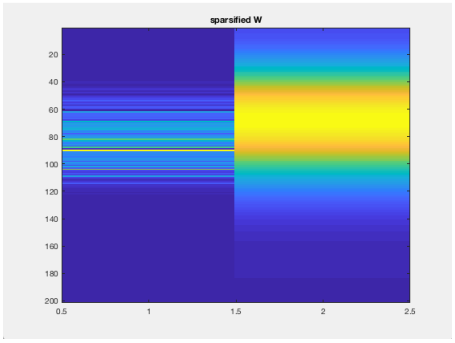


(b)

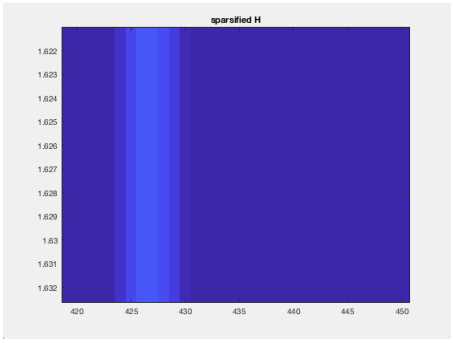


(c)

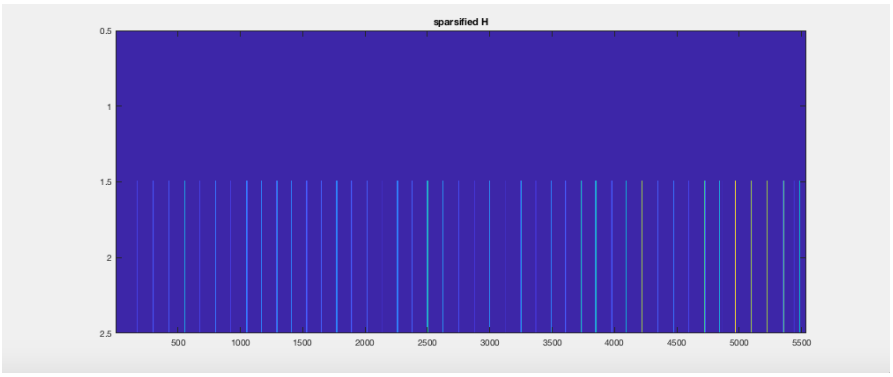
[5]To be more precise, it would be set to $\bar{x}_\lambda = \epsilon$ for a sufficiently small value of $\epsilon > 0$.



(d)



(e)



(f)

Figure 4.5: *Nonnegative factors generated by MU-NMF(2) (a)-(c) and MU-SNMF(2) (d)-(f) with penalty* $\lambda = 10$ *applied to the spectrogram of "PCG85315_PV" from Database2. Zoom-in on cardiac sound activation coefficients within the unsparsified (b) and sparsified (e) left factor.*

Interestingly enough, the choice of the penalty parameter in this particular example allowed MU-SNMF(2) to converge to a stationary point in which one of the two basis functions is never activated. We can therefore conclude that activating any basis function different from the second one contained in Figure (4.5d) would result in an increase of the objective. Moreover, as we can see from Figure (4.5b) and Figure (4.5e), the activation coefficients within the right factors appear sparser and more concentrated around the peak of the corresponding cardiac event when the sparse variant of the algorithm is involved.

4.2.4 Localized Median Filter

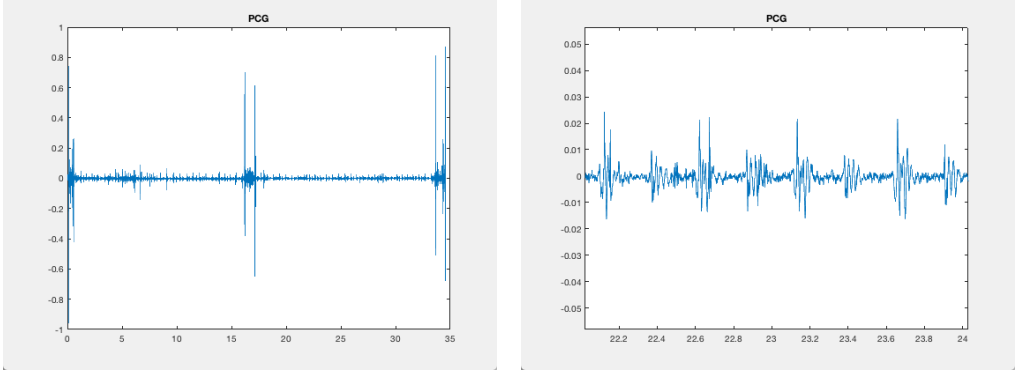
Similar to the aforementioned Norm Clipping technique, another signal-based denoising routine that can be used to attenuate high-energy noise impulses

relies on the application of a Median Filter. The latter is more commonly employed in a digital image processing environment to reconstruct the value of isolated corrupted pixels from the clean, neighboring ones. In its most simple form, the Median Filter applies a moving window to the signal, computing the median over all samples spanned by the former. More precisely, given a one-dimensional discrete signal $x = \{x_k\}_{k=1,\dots,N}$ and a window length of L samples, the filter produces a signal $\hat{x} = \{\hat{x}_k\}_{k=1,\dots,N-L+1}$ where:

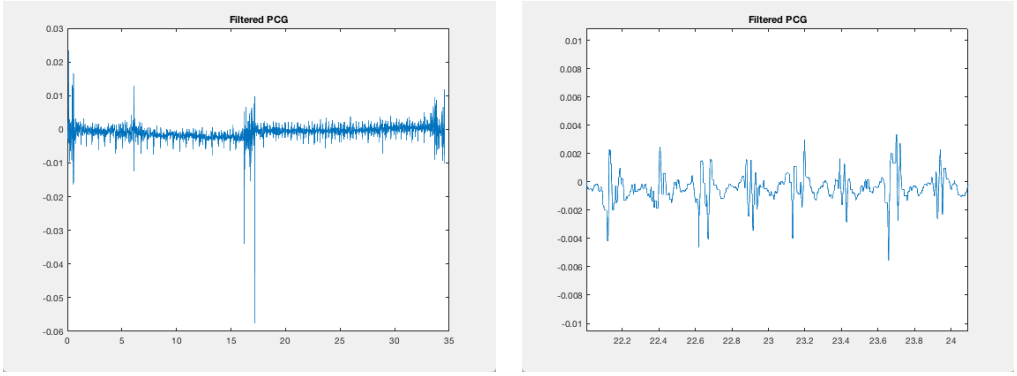
$$\hat{x}_k = \text{median}(x_k, \dots, x_{k+L-1})$$

As a consequence, as long as the corrupted samples are sufficiently isolated or time-localized, the moving median window will also span clean samples, thus attenuating the former's influence.

We should remark that applying a Median Filter to audio signals, such as PCGs, is usually not as straightforward as one might think, especially if we are interested in preserving the key features within the clean signal samples. As for our dataset, in which the PCGs are down-sampled to a frequency $f_s = 1\text{kHz}$, even short noise impulses with a duration in the tenths of seconds will corrupt hundreds of consecutive samples. Under these circumstances, we would then be required to fix the window length L to the same order of magnitude in order to effectively denoise the signal; on the other hand, such a wide window would also have undesired effects on the clean sections of the PCG. In Figure (4.6) below we notice how the high-energy impulses were correctly attenuated by the filter using a window of $L = 31$ samples, although upon closer inspection, the clean sections containing exclusively heart sounds have been severely modified. This is particularly problematic in our framework, since maintaining the clean frequency contents of PCGs unaltered is of crucial importance for a NMF spectrogram-based learning.



(a) (b)



(c) (d)

Figure 4.6: *Unfiltered (a) and median filtered (c) "PCG49577_MV" from Database2 using a* $L = 31$ *sample window. Zoom-in on four clean heart beats within the unfiltered (b) and median filtered (d) signal. The window length was chosen based on the duration of the high-energy impulses that can be observed at times* $t = 0s$*,* $t = 16.2s$*,* $t = 17.2s$*,* $t = 33.7s$ *and* $t = 34.5s$*.*

In order to address these issues, the method we propose, which we shall hereon refer to as Localized Median Filter (LMF), applies the filter exclusively to the windows centered at samples corrupted by high-energy impulses. In other words, given a set of indexes $I \subset \{1, \dots, N\}$ associated to corrupted samples and a radius r, the LMF produces a signal $\hat{x} = \{\hat{x}_k\}_{k=1,\dots,N}$ where:

$$\hat{x}_k = \begin{cases} \text{median}(x_{k-r}, \dots, x_{k+r}) & \text{if } k \in I, \\ x_k & \text{if } k \notin I. \end{cases}$$

Moreover, much like in the Norm Clipping routine, assuming the high-energy impulses to be characterized by high amplitudes within the signal, the index set I can be obtained by thresholding the absolute value of the PCG samples. We can therefore summarize the LMF denoising routine in the following Algorithm (4.2.4):

Algorithm 4.2.4 Localized Median Filter

Input: A signal $x = \{x_k\}_{k=1,\dots,N}$, window radius r and threshold M.

Output: A denoised signal $\hat{x} = \{\hat{x}_k\}_{k=1,\dots,N}$.

`1:` `Initialize` $\hat{x} = x$

2: **for** $k = 1, \ldots, N$ **do:**

3: **if** $|x_k| > M$

4: $l = \max(k - r, 1)$

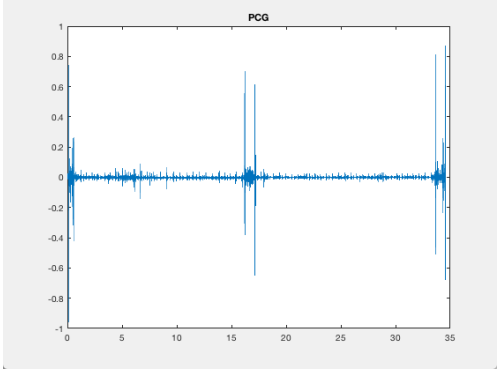
5: $u = \min(k + r, N)$

6: $\hat{x}_k = \text{median}(x_l, \ldots, x_u)$

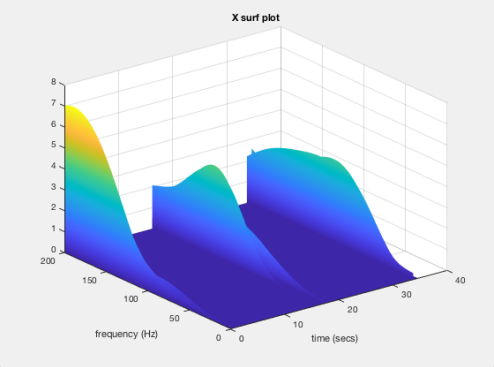
7: **end if**

8: **end for**

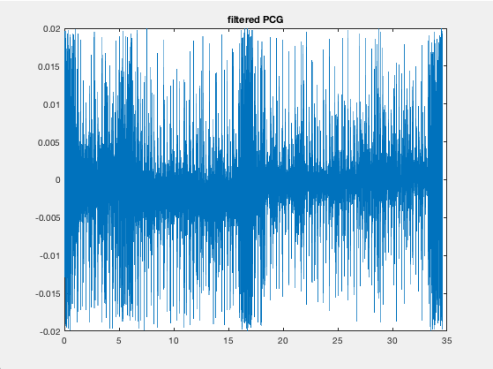
In Figure (4.7) we chose to apply the above denoising routine to the same PCG signal as before with radius $r = 15$ and threshold $M = 0.02$. In particular, the value for M was chosen based on the amplitude of the clean signal sections. As we can observe, the high-energy impulses were removed and in the spectrogram of the filtered signal the columns associated to cardiac events now have significant weight. This last remark is of pivotal importance if we recall the analysis carried out is Section 3.2.3 regarding the presence of noise in PCGs and the construction of basis functions by the NMF. Indeed, if we were to apply MU-NMF(2) to the noisy spectrogram, the nonnegative factors would be modelled to describe the noise rather than cardiac events, thus hindering the foundations of our NMF-based learning. Moreover, comparing the spectrograms of Figure (4.7d) and Figure (4.3c) produced by NC, we notice how the LMF was also able to attenuate the high-frequency content of the impulsive noise.



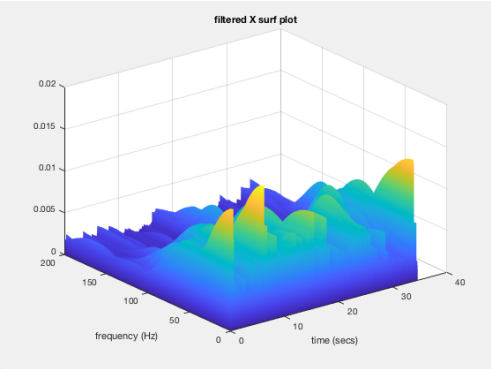
(a)



(b)



(c)



(d)

Figure 4.7: *Unfiltered (a) and Localized Median Filtered (c) "PCG49577_MV" from Database2 using a radius* $r = 15$ *and threshold* $M = 0.02$*. Spectrogram surf plot for the unfiltered (b) and Localized Median Filtered (d) signal.*

4.2.5 NMF Adaptive Noise Canceler

In the previous sections we have developed several denoising techniques whose purpose was dealing with signals affected by either high-energy noise impulses or low-energy murmurs. We will now shift our attention towards *noise bursts*, that is, medium-to-high-energy disturbances characterized by a longer duration (see, for example, Figure (3.5a)). Due to the prolonged duration of this type of noise, the aforementioned techniques are often ineffective at providing a sufficiently clean signal or spectrogram for the subsequent NMF application. In the following pages we will therefore devise a NMF-based Noise Canceler that heavily relies on the information provided by the nonnegative factors to produce cleaner PCGs.
As we will now briefly introduce, the mathematical foundations for this kind of denoiser lie in the theory of Adaptive Filtering [7]. In the particular case of Adaptive Noise Cancellation, let us then consider a noisy random signal $y = \{y(t)\}_{t \in T}$ decomposed additively as the sum of a clean signal $s = \{s(t)\}_{t \in T}$ and noise $n = \{n(t)\}_{t \in T}$:

$$y(t) = s(t) + n(t) \tag{4.14}$$

In particular, in our notation we are assuming t to be a discrete time index. The fundamental premise for the arguments to follow is that we suppose to be known a *reference noise signal* $r = \{r(t)\}_{t \in T}$ correlated to the actual noise n present in y. In this framework, we model the noise n by assuming the

latter to depend linearly from the reference through some unknown (time-dependent) coefficients. More precisely, we consider the model:

$$n(t) = \sum_{i=0}^{N_h-1} h_i(t) r(t-i) = \mathbf{h}(t)^T \mathbf{r}(t) \tag{4.15}$$

where:

- $N_h \in \mathbb{N}^*$ is the model order;
- $\mathbf{h} = \{\mathbf{h}(t)\}_{t \in T}$, $\mathbf{h}(t) = [h_0(t), \dots, h_{N_h-1}(t)]^T$ are the coefficients to be estimated[6];
- $\mathbf{r}(t) = [r(t), \dots, r(t-N_h+1)]^T$ are the components of the reference we assume to be correlated to $n(t)$.

If we then measure the quality of the approximated noise, at each time instant, through the error:

$$e(t) = y(t) - \mathbf{h}(t)^T \mathbf{r}(t) \tag{4.16}$$

we may consider the following time-dependent objective to be minimized:

$$F(t; \mathbf{h}(t)) = \mathbb{E}\left[e(t)^2\right] \tag{4.17}$$

where t is to be considered fixed. In particular, since $\nabla_{\mathbf{h}(t)} e(t) = -\mathbf{r}(t)$, the gradient with respects to the $\mathbf{h}(t)$ variables of the former is given by:

$$\nabla_{\mathbf{h}(t)} F(t; \mathbf{h}(t)) = -2\mathbb{E}\left[y(t)\mathbf{r}(t)\right] + 2\mathbb{E}\left[\mathbf{r}(t)\mathbf{r}(t)^T\right] \mathbf{h}(t) \tag{4.18}$$

More compactly, we shall write:

$$\nabla_{\mathbf{h}(t)} F(t; \mathbf{h}(t)) = -2\mathbf{R}_{yr}(t) + 2\mathrm{R}_{rr}(t)\mathbf{h}(t) \tag{4.19}$$

As a consequence, by applying a first order method to find stationary points of (4.17), from the above computations it follows that the desired estimated coefficients $\hat{\mathbf{h}}(t)$ will satisfy the linear system:

$$\mathrm{R}_{rr}(t)\hat{\mathbf{h}}(t) = \mathbf{R}_{yr}(t) \tag{4.20}$$

[6]Unlike y, s, n and r, the coefficients $\mathbf{h}$ are *not* random signals, despite the time dependence.

Letting $\hat{n}(t) = \hat{\mathbf{h}}(t)^T \mathbf{r}(t)$ be the estimated noise at time t, we can then reconstruct the clean, denoised signal:

$$\hat{s}(t) = y(t) - \hat{n}(t) \tag{4.21}$$

Evidently, from a computational point of view, we cannot work with the random signals y and r, since, in most cases, they are unknown. As a matter of fact, in practical applications we only have at our disposal some *realizations* $\bar{y} = \{\bar{y}_k\}_{k=1,\dots,N}$ and $\bar{r} = \{\bar{r}_k\}_{k=1,\dots,N}$ of the signals, recorded for a finite number of discrete time samples. As a consequence, we will now briefly discuss how the above mathematical foundations for an Adaptive Noise Canceler (ANC) can be modified to accomodate our limited knowledge of the signals at hand. Let us distinguish between two cases:

Stationarity. In the case where the random signals y and r can be considered (jointly) stationary, Equation (4.20) loses its time dependence. Indeed, the coefficients of the system matrix $\mathrm{R}_{rr}(t)$ and right-hand-side $\mathbf{R}_{yr}(t)$ are expected values in the form $\mathbb{E}\left[y(t)r(t-k)\right]$ or $\mathbb{E}\left[r(t-k_1)r(t-k_2)\right]$ for some appropriate nonnegative time shifts $k, k_1, k_2 \geq 0$. Under stationarity hypothesis, such quantities depend exclusively on the relative shifts k and $|k_1 - k_2|$, hence $\mathrm{R}_{rr}(t) = \mathrm{R}_{rr}$, $\mathbf{R}_{yr}(t) = \mathbf{R}_{yr}$ and therefore the coefficients $\hat{\mathbf{h}}(t) = \hat{\mathbf{h}}$ now become independent of time as well. As a consequence, in order to obtain $\hat{\mathbf{h}}$, it suffices to construct an estimate for the system matrix and right-hand-side using the realizations $\bar{y}$ and $\bar{r}$ at our disposal. In particular, this can be done by recalling that given two (ergodic and stationary) random signals $a = \{a(t)\}_{t\,\in\,T}$, $b = \{b(t)\}_{t\,\in\,T}$ and two realizations $\bar{a} = \{\bar{a}_k\}_{k=1,\dots,N}$, $\bar{b} = \{\bar{b}_k\}_{k=1,\dots,N}$, an estimate $\hat{R}_{ab}(k)$ for the quantity $R_{ab}(k) = \mathbb{E}\left[a(t)b(t-k)\right]$ is given by:

$$\hat{R}_{ab}(k) = \frac{1}{N-k-1} \sum_{t=k+1}^{N} \bar{a}_t \bar{b}_{t-k} \tag{4.22}$$

The following Algorithm (4.2.5) summarizes how the Stationary Adaptive Noise Canceler (SANC) can be implemented:

Algorithm 4.2.5 Stationary Adaptive Noise Canceler

Input: A signal $y = \{y_k\}_{k=1,\dots,N}$, noise reference $r = \{r_k\}_{k=1,\dots,N}$ and model order N_h.

Output: A denoised signal $\hat{y} = \{\hat{y}_k\}_{k=1,\dots,N}$.

for $i = 0, \ldots, N_h - 1$ **do:**
  $\mathbf{R}_{yr}(i) = \hat{R}_{yr}(i)$
  for $j = 0, \ldots, N_h - 1$ **do:**
    $\mathrm{R}_{rr}(i, j) = \hat{R}_{rr}(|i - j|)$
  end for
end for
$\hat{\mathbf{h}} = \mathrm{R}_{rr}^{-1}\mathbf{R}_{yr}$
for $k = 1, \ldots, N$ **do:**
  $\mathbf{r}_k = [r_k, \ldots, r_{k-N_h+1}]^T$
  $\hat{n}_k = \hat{\mathbf{h}}^T\mathbf{r}_k$
  $\hat{y}_k = y_k - \hat{n}_k$
end for

Nonstationarity. If the random signals y and r cannot be considered stationary, the model's coefficients $\hat{\mathbf{h}}(t)$ are actually time dependent, but unlike the previous case, estimating the system matrix and right-hand-side from a single realization $\bar{y}$ and $\bar{r}$ would no longer produce accurate results. As a consequence, following the work of Candy [7], one possible approach to obtain the estimates $\hat{\mathbf{h}}(t)$ is to apply a gradient descent method to the objective (4.17) and approximate its gradient with its instantaneous version[7]:

$$\nabla_{\mathbf{h}(t)} F(t; \mathbf{h}(t)) \approx \hat{\nabla}_{\mathbf{h}(t)} F(t; \mathbf{h}(t)) = -2e(t)\mathbf{r}(t) \tag{4.23}$$

Fixing a descent step $\frac{\delta_i}{2}$ we can therefore consider the updates:

$$\hat{\mathbf{h}}^{(i+1)}(t) = \hat{\mathbf{h}}^{(i)}(t) + \delta_i e^{(i)}(t)\mathbf{r}(t) \tag{4.24}$$

where $e^{(i)}(t) = y(t) - \hat{\mathbf{h}}^{(i)}(t)^T\mathbf{r}(t)$. We should remark that, in this framework, the coefficients actually become random variables. In particular, (4.24) provides an iterative scheme that depends on the time index t and iteration counter i, but there is little correlation between the coefficients' estimate at time t and time $t + 1$. As a consequence, the two indexes are merged into a single time index, thus yielding the following update rule:

[7]The instantaneous gradient is obtained by dropping the expected value operator.

$$\hat{\mathbf{h}}(t+1) = \hat{\mathbf{h}}(t) + \delta_t e(t)\mathbf{r}(t) \tag{4.25}$$

Moreover, as remarked by the same author, choosing δ_t as:

$$\delta_t = \frac{\alpha}{\|\mathbf{r}(t)\|_2^2 + \beta} \tag{4.26}$$

for some fixed coefficients $\alpha \in (0, 2)$ and $\beta > 0$, is enough to guarantee the convergence of the scheme in a mean-squared sense.
In conclusion, since (4.25) does not contain any expected values, we can obtain an estimate of the coefficients by replacing the random variables y, r with their respective realizations $\bar{y}$, $\bar{r}$. The following Algorithm (4.2.6) describes how the Nonstationary Adaptive Noise Canceler (NSANC) can be implemented. For clarity, we should remark that the latter approximates the value $\|\mathbf{r}(t)\|_2^2$, present in the descent step, using a forgetting-factor approach on the estimated autocorrelation sequence $\{\hat{V}_k\}_{k=1,\dots,N}$ given by $\hat{V}_{k+1} = \gamma\hat{V}_k + (1-\gamma)\bar{r}_k^2$ for $\gamma \in (0, 1)$.

Algorithm 4.2.6 Nonstationary Adaptive Noise Canceler

Input: A signal $y = \{y_k\}_{k=1,\dots,N}$, noise reference $r = \{r_k\}_{k=1,\dots,N}$ and model order N_h.

Output: A denoised signal $\hat{y} = \{\hat{y}_k\}_{k=1,\dots,N}$.

Initialize $\hat{\mathbf{h}}(0)$, $\hat{V}_0 > 0$, $\alpha \in (0, 2)$, $\beta > 0$, $\gamma \in (0, 1)$.
for $k = 1, \dots, N$ **do:**
  $\mathbf{r}_k = [r_k, \dots, r_{k-N_h+1}]^T$
  $\hat{n}_k = \hat{\mathbf{h}}(k-1)^T\mathbf{r}_k$
  $\hat{y}_k = y_k - \hat{n}_k$
  $\hat{V}_k = \gamma\hat{V}_{k-1} + (1-\gamma)r_k^2$
  $\delta_k = \frac{\alpha}{\hat{V}_k+\beta}$
  $\hat{\mathbf{h}}(k) = \hat{\mathbf{h}}(k-1) + \delta_k\hat{y}_k\mathbf{r}_k$
end for

Let us now go back to the main objective for this section, namely, developing a NMF-based Noise Canceler.

Given a signal $x = \{x_k\}_{k=1,\dots,N}$ affected by noise bursts, we would like to apply the aforementioned Adaptive Noise Cancellation routines to denoise the former. Nevertheless, in order to do so, a noise reference signal $r = \{r_k\}_{k=1,\dots,N}$ must be available. What we will now show, is how to exploit the modelling capabilities of the NMF to obtain such reference.
From the analysis carried out in Section 3.2.3 we know that MU-NMF(2), if applied to the spectrogram X of the signal, will allocate one (or more) of the basis functions at its disposal to model the noise. As a consequence, while one column of the left factor W will be used to describe the prominent *frequency content* of the noise, the corresponding row of H will dictate *when* such frequencies are detected. In short, the method we propose extrapolates the reference directly from the signal x by windowing a band-passed version of itself. Let us now describe each step of the procedure in more detail:

Step 0: Column and row selection. Initially, we compute the spectrogram X of the signal and apply MU-NMF(2) (or one of its variants) to generate the two nonnegative factors[8] $W = \{W_k\}_{k=1,\dots,r}$ and $H = \{H^k\}_{k=1,\dots,r}$. Subsequently, we identify the column of W whose frequency content is (most) correlated to the noise and thus select the corresponding row of H as well. In the case where the noise can be distinctly recognized and time-localized from the signal x itself, this process can be completed by selecting the row of H (and consequently, the corresponding column of W) whose activation coefficients are most time-wise correlated to the former. Without loss of generality, let's assume of having selected the first column-row pair (W_1, H^1).

Step 1: Band-pass filter application. From W_1 we extract a frequency band $[f_l, f_h]$ containing the peak that characterizes the noise. This process can be either carried out by hand in a supervised framework, or automated by detecting the peak and constructing both a (adaptable) left and right radius. We then band-pass x on the extracted frequency interval (using, for example, low-pass and high-pass zero-phase filters with cutoff frequencies f_h and f_l respectively, see Section 3.1.1) to obtain a filtered signal $x^{bp} = \{x_k^{bp}\}_{k=1,\dots,N}$. So far we have frequency-localized the noise; the next step is to time-localize it.

Step 2: Weight decompression. Let us now focus on H^1, which we shall hereon refer to as a 'window'. Denoting $H^1 = \mathcal{W} = \{w_j\}_{j=1,\dots,M}$ the 'weights' of the window, the first thing we need to observe is that $\mathcal{W}$ cannot be immediately used to window x^{bp} since $M < N$. Indeed, if the signal x is sampled at a frequency f_s and the STFT was computed using parameters δT and p,

[8]Here W_k and H^k are, respectively, the k-th column of W and row of H.

one can check that:

$$M = \left\lfloor \frac{N - L_w}{\delta_w} \right\rfloor + 1$$

where $L_w = \lfloor f_s \delta T \rfloor$ and $\delta_w = L_w - \lfloor L_w p \rfloor$. This is due to the fact that each time frame of the STFT matrix is computed using exactly L_w samples of x. As a consequence, we need to devise a way of decompressing $\mathcal{W}$ into a signal of length N. By noticing that the j-th compressed time index of the STFT depends exclusively on the uncompressed time index interval $[1 + (j-1)\delta_w, L_w + (j-1)\delta_w]$ of x, the following Algorithm (4.2.7) accomplishes our goal. Let us denote $\hat{\mathcal{W}} = \{\hat{w}_k\}_{k=1,\dots,N}$ such decompressed weights window.

Algorithm 4.2.7 Weight Decompresser

Input: A compressed signal $\mathcal{W} = \{w_j\}_{j=1,\dots,M}$ related to a length N signal in a STFT-like manner with parameters f_s, δT and p.

Output: A decompressed signal $\hat{\mathcal{W}} = \{\hat{w}_k\}_{k=1,\dots,N}$

`Initialize` $\hat{\mathcal{W}} = \{\hat{w}_k\}_{k=1,\dots,N} = 0$, $\Gamma = \{\gamma_k\}_{k=1,\dots,N} = 0$
$L_w = \lfloor f_s \delta T \rfloor$
$\delta_w = L_w - \lfloor L_w p \rfloor$
for $j = 1, \dots, M$ **do:**
    for $k = 1 + (j-1)\delta_w, \dots, L_w + (j-1)\delta_w$ **do:**
        $\hat{w}_k = \hat{w}_k + w_j$
        $\gamma_k = \gamma_k + 1$
    end for
end for
for $k = L_w + (M-1)\delta_w + 1, \dots, N$ **do:**
    $\gamma_k = 1$
end for
for $k = 1, \dots, N$ **do:**
    $\hat{w}_k = \frac{\hat{w}_k}{\gamma_k}$
end for

Step 3: Reference creation. Now that we have both frequency and time-localized the noise, the last step is to create the reference $r = \{r_k\}_{k=1,\dots,N}$. In this regard, we propose to define the latter as the bandpassed signal x^{bp} windowed with the decompressed weights $\hat{\mathcal{W}}$, that is:

$$r_k = x_k^{bp}\hat{w}_k \qquad \forall k = 1, \dots, N \tag{4.27}$$

With respect to Figure (4.1), which contains the state machine representation of the NMF-based heart rate detection algorithm, we should remark that *Step 0* of the above procedure can be considered to be part of the *NMF feedback*, hence an a priori step preceding the actual denoising routine. With this in mind, the following Algorithm (4.2.8) summarizes all the steps described so far within the general scheme for what we shall hereon refer to as NMF Adaptive Noise Canceler (NMF-ANC):

Algorithm 4.2.8 NMF Adaptive Noise Canceler

Input: A signal $x = \{x_k\}_{k=1,\dots,N}$, a column-row pair (W_k, H^k) extracted from the NMF of its spectrogram X and a model order N_h.

Output: A denoised signal $\hat{x} = \{\hat{x}_k\}_{k=1,\dots,N}$

1: `Extract from` W^k `a frequency band` $[f_l, f_h]$

2: `Compute` $x^{bp} = \{x_k^{bp}\}_{k=1,\dots,N}$ `by band-passing` x `on the frequency interval` $[f_l, f_h]$

3: `Compute` $\hat{\mathcal{W}} = \{\hat{w}_k\}_{k=1,\dots,N}$ `from` H^k `by calling` Algorithm (4.2.7)

4: `Define` $r = \{x_k^{bp}\hat{w}_k\}_{k=1,\dots,N}$

5: `Compute` $\hat{x} = \{\hat{x}_k\}_{k=1,\dots,N}$ `from` x`,` r `and` N_h `by calling` Algorithm (4.2.5) `or` Algorithm (4.2.6)

Before providing the results yielded from applying the above denoising routine to concrete PCG signals, there are some key aspects of the latter that need to be discussed.
First and foremost we should highlight some possible complications regarding the extraction of the noisy frequency band $[f_l, f_h]$. As a matter of fact, while this process can be completely automated, one must keep in mind that:

Remark 4.2. *In the case where the noise and the clean signal features share a significant portion of their frequency content, it may be worthwhile reducing the diameter of the interval to keep the clean frequencies out of the noise reference.*

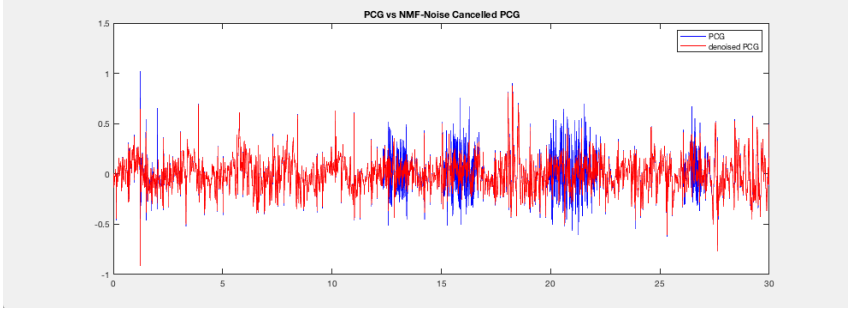
Of course, this aspect becomes more and more noteworthy as the time-correlation between the noise and the clean signal features increases.
Secondly, the choice of the model order N_h should be made based on the type and duration of the noise at hand:

Remark 4.3. *For noise bursts of prolonged duration, opting for a higher value of the order may aid the modelling of a more fitting noise estimate. Similarly, if the noise is more localized, reducing such parameter could help preserving the quality of the clean signal sections.*

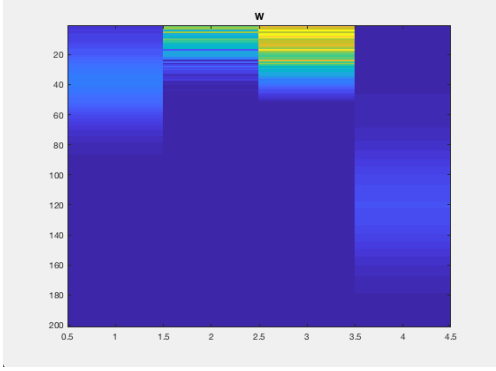
Lastly, one must decide whether to utilize the Stationary or Nonstationary version of the Adaptive Noise Canceler based on the intrinsic properties of the noise. For example:

Remark 4.4. *If the noise at hand appears to be characterized by a smaller frequency window and is present throughout the signal, a sensible conclusion would be to opt for the stationary version of the algorithm. On the other hand, if the noise only affects isolated portions of the signal, the nonstationary variant could potentially offer a more accurate noise estimation.*

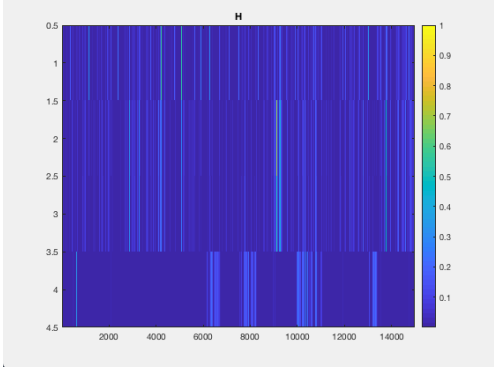
To conclude this section, in Figure (4.8) below we report the effects of the NMF-ANC on a PCG containing four distinct noise bursts.



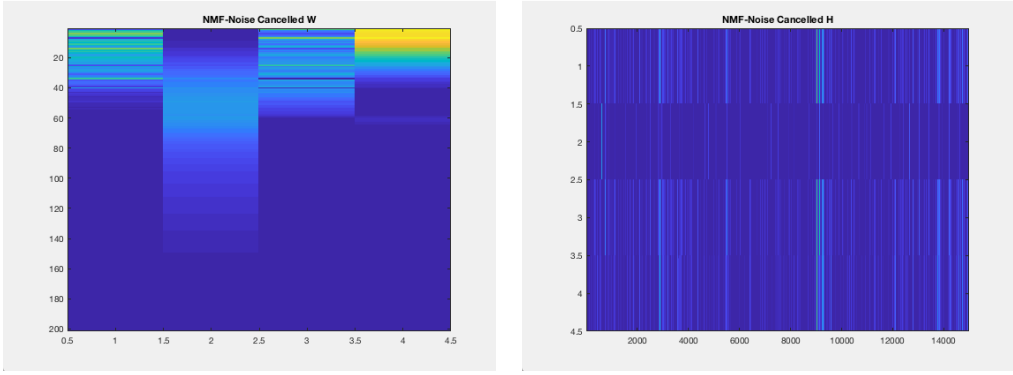
(a)



(b)



(c)



(d) (e)

Figure 4.8: *Comparison (a) between noisy (blue) and NMF Adaptive Noise Cancelled (red) "PCG0004" from Database1 with noise reference constructed from the fourth column-row pair of (b) and (c), frequency band [60:180]Hz and model order* $N_h = 1000$ *for the NSANC. Nonegative factors generated by MU-NMF(2) applied to the spectrogram of the noisy (b)-(c) and NMF Adaptive Noise Cancelled (d)-(e) signal.*

The noise reference was created from the fourth column-row pair of Figure (4.8b) and Figure (4.8c). In particular, we selected the frequency band $[60, 180]$Hz and chose the Nonstationary ANC routine with an order $N_h = 1000$ based on the duration and relative time-isolation of the noise. As we can observe, the noise bursts affecting the top nonnegative factors are completely absent in the bottom ones. When it comes to the detection of cardiac events, in Figure (4.8c) the dedicated row of H is the first, while in Figure (4.8e) it is the second. In particular, from the latter essentially all cardiac events can be easily and clearly identified. The same cannot be said for the former, since most cardiac events time-correlated to the noise bursts have been corrupted and are thus difficult to spot.
Remarkably, we also tested the effectiveness of the Stationary ANC routine on the same signal and reference, but in contrast with the Nonstationary one, multiple denoising applications were needed to obtain comparable results. The choice of using NSANC based on the noise properties was therefore well founded.

Chapter 5

Detection algorithm

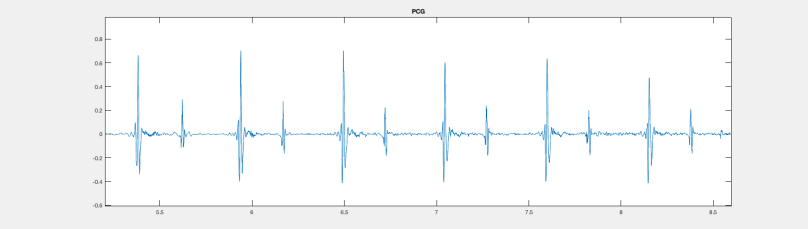
The denoising routines developed in the previous chapter were the final building blocks required to complete our proposed NMF-based heart rate detection algorithm. With respect to Figure (4.1), the last major aspect that needs to be discussed is the interaction between these routines and the signal's quality assessment within the denoising step of the procedure. More precisely, in the following sections we will expand on the state machine representation of the algorithm by describing in detail when each denoiser should be applied based on the signal's condition and other available data.

5.1 PCG quality assessment

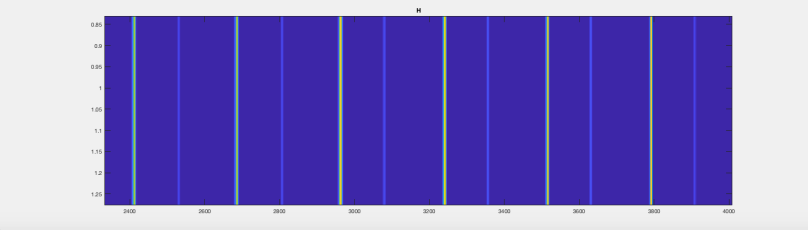
Before providing a state machine denoising flow for the signals in our dataset, we need a way to categorize PCGs based on their quality, that is, the type of noise they contain as well as on the effort required to identify cardiac events within them. In particular, this data quality assessment should be carried out by taking advantage of all available information, namely, the (current) signal x itself, its spectrogram X and the nonnegative factors W, H. Recall, as a matter of fact, that the data present within W and H can be utilized in the denoising process through the *NMF feedback* component of the detection algorithm. Let us point out that this categorization of PCGs should *not* be interpreted as a medically accurate way of differentiating these signals based on their physiological features. Rather, the purpose of the latter is to establish a more precise, albeit intuitive, procedure to justify the denoising choices we will present.
Based on the analysis of the databases at our disposal, we therefore propose the following four categories, which we shall denote in short as 'Q-' (as

in quality) followed by 'A'-'D', listed in increasing order of disruptiveness associated to the noise contained in the data:

Q-A: *Clean or mostly clean.* In this category fall PCG signals characterized by either low-level or no noise at all. Cardiac events should be easily identified by examining the signal x or its spectrogram X. Alternatively, at least one column of W should be unequivocally related to the frequency content of heart sounds, and the corresponding row of H must provide a clear time localization of cardiac events with little to no detectable interference.



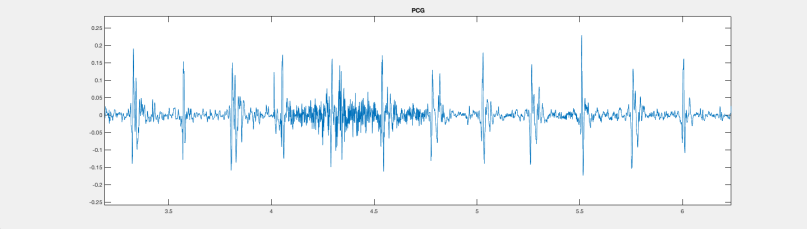
(a)



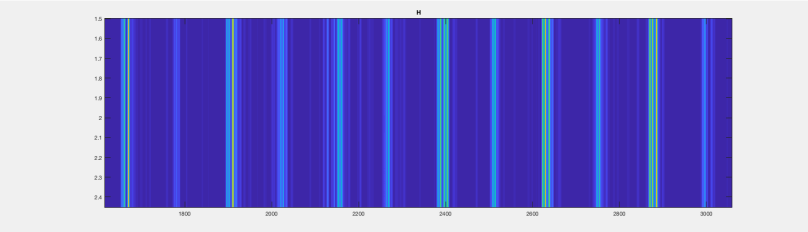
(b)

Figure 5.1: *Example of a PCG belonging to Q-A. Zoom-in on "PCG50115_PV" from Database2 (a) and zoom-in on one of the rows of the right factor H (b).*

Q-B: *Minor murmurs.* The PCGs in this category are affected by low-energy murmurs that do not compromise the clean signal features. Most cardiac events should be recognizable both in x, X and within at least one column-row pair of W, H. In the worst case, it may be difficult to differentiate between murmurs and particularly low-energy cardiac events.



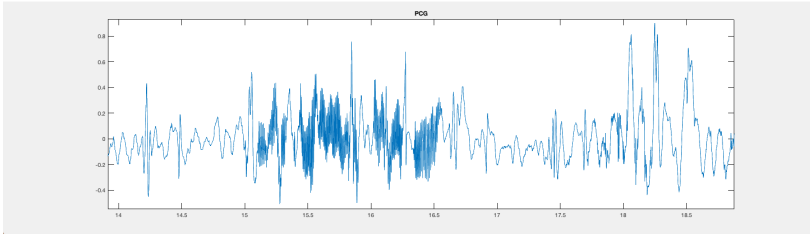
(a)



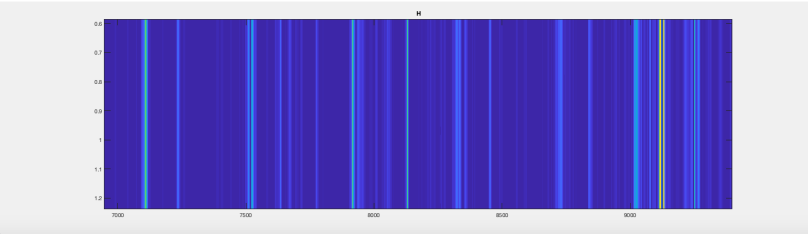
(b)

Figure 5.2: *Example of a PCG belonging to Q-B. Zoom-in on "PCG85315_PV" from Database2 (a) and zoom-in on one of the rows of the right factor H (b).*

Q-C: *Burst or widespread noise.* In this category, the PCGs are affected by medium-to-high-energy noise bursts characterized by a prolonged duration. Cardiac events time-correlated to such bursts should be difficult to be identified. In the worst case, no section of the signal is left uncompromised by the noise, but some cardiac event-related pattern can still be spotted within the rows of H. Alternatively, at least one column of W should be unequivocally related to the frequency content of the noise, and the corresponding row of H must provide a general time localization of the latter.



(a)

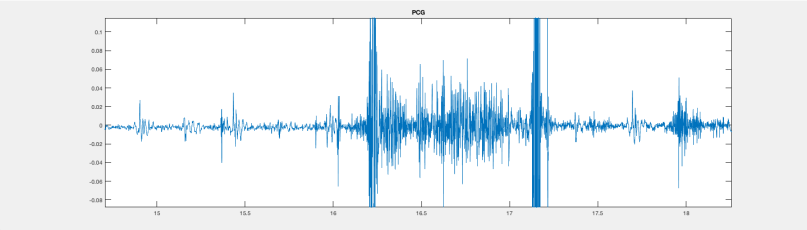


(b)

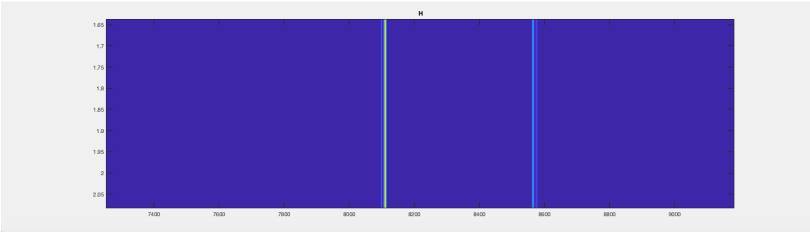
Figure 5.3: *Example of a PCG belonging to Q-C. Zoom-in on "ECGPCG0004" from Database1 (a) and zoom-in on one of the rows of the right factor H (b).*

Q-D: *Impulsive noise.* To this category belong PCGs containing disruptive high-energy noise impulses. Although greatly time-localized, thus leaving

some unaltered, clean sections, the noise should be clearly recognizable from both x and X. Moreover, in the worst case, the latter corrupted most if not all basis functions, hence rendering the nonnegative factors unusable for both cardiac event detection and denoising.



(a)



(b)

Figure 5.4: *Example of a PCG belonging to Q-D. Zoom-in on "PCG49577_MV" from Database2 (a) and zoom-in on one of the rows of the right factor H (b).*

We should remark that these categories are by no means disjoint: some PCGs in the considered dataset may be affected by multiple types of noise, hence belonging to more than one category. In particular, this aspect will reflect in the following state machine denoising flow, since a single PCG signal can undergo the application of multiple denoising routines.

Remark 5.1. *In the remainder of this chapter, given a PCG signal x belonging to a category Q-X, we shall write $x \in$ Q-X.*

5.2 State machine denoising flow

Let us now describe the denoising flow in our proposed algorithm. As mentioned earlier, since the four quality categories are not disjoint, the former is characterized by an iterative structure: during each iteration a denoising routine is *proposed* based on the current signal quality assessment and result

requirements, such as detection precision. In a supervised environment, one can decide not to apply the proposed routine. This is the case, for example, when one of rows of the right factor has already reached a satisfactory level of cardiac event detection precision despite the persistence of noise. As a consequence, from an algorithmic point of view, if none of the proposed routines are accepted we simply stop the denoising process. This will be modelled by the boolean variable `none_accepted`. Moreover, since some denoising routines require the choice of thresholds or penalty parameters, we allow a 'roll-back' of the last operation in case a faulty assignment during the latter compromised the signal quality (e.g. fixing the penalty parameter for the MU-SNMF variant to an exceedingly large value). The choice of undoing the last operation will we modelled using the boolean variable `undo_last`. As far as the denoising *order* is concerned, the proposed routines should first address the more disruptive types of noise in order to open up the subsequent iterations to more denoising options. Notice, for example, that in most cases a PCG belonging to Q-D cannot be denoised using the NMF-ANC or the WTC.
In conclusion, the following Algorithm (5.2) provides a state machine representation of the denoising flow:

Algorithm 5.2 Denoising flow

Input: A PCG signal x.

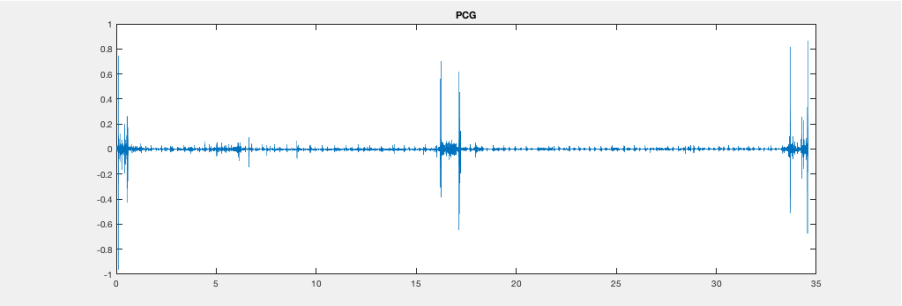
Output: A denoised PCG signal $\hat{x}$

Initialize $\hat{x}^{(0)} = x$
for $k = 0,\ 1,\ \ldots$ **do:**
    Assess the quality of $\hat{x}^{(k)}$ as in Section 5.1
    if undo_last
        $\hat{x}^{(k+1)} = \hat{x}^{(k-1)}$
    elseif $\hat{x}^{(k)} \in$ Q-A
        $\hat{x} = \hat{x}^{(k)}$
        break
    elseif $\hat{x}^{(k)} \in$ Q-D
        if signal_thresholding_possible
            propose computing $\hat{x}^{(k+1)}$ from $\hat{x}^{(k)}$ by applying the LMF algorithm (Section 4.2.4)

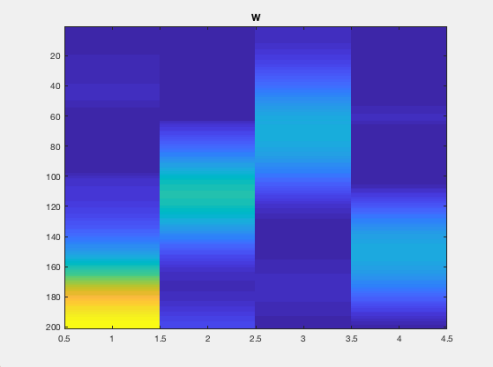
```
        else
            propose computing x̂^(k+1) from x̂^(k) by applying the NC algorithm (Section 4.2.1)
        end if
    elseif x̂^(k) ∈ Q-C
        propose computing x̂^(k+1) from x̂^(k) by applying the NMF-ANC algorithm (Section 4.2.5)
    elseif x̂^(k) ∈ Q-B
        if detection_precision_required
            propose computing x̂^(k+1) from x̂^(k) by applying the MU-SNMF variant (Section 4.2.3)
        else
            propose computing x̂^(k+1) from x̂^(k) by applying the WTC algorithm (Section 4.2.2)
        end if
    end if
    if none_accepted
        x̂ = x̂_k
        break
    end if
end for
```

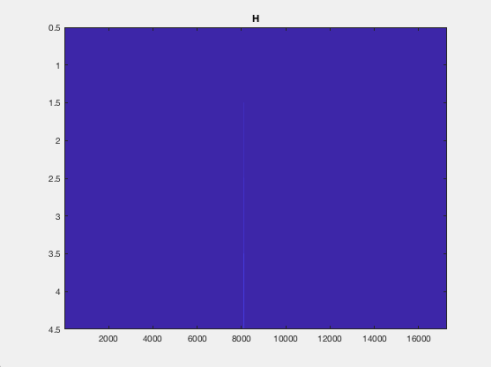
Before closing this section, we provide an application of the above denoising process to the PCG signal "PCG49577_MV" from Database2.

First iteration. As we can see from Figure (5.5) below, the considered PCG is affected by multiple disruptive noise impulses, as evidenced by the two nonnegative factors. As a matter of fact, no basis function is associated to the frequency contents of heart sounds, thus the right factor cannot be used for cardiac event detection. Moreover, in the time instants preceding and following the impulses, some slightly more prolonged disturbances can be observed. Recalling our earlier PCG categorization, during this first iteration the signal is assessed as belonging to both Q-D and Q-C. At this time, although reasonable, we cannot conclude that the PCG contains murmurs.



(a)



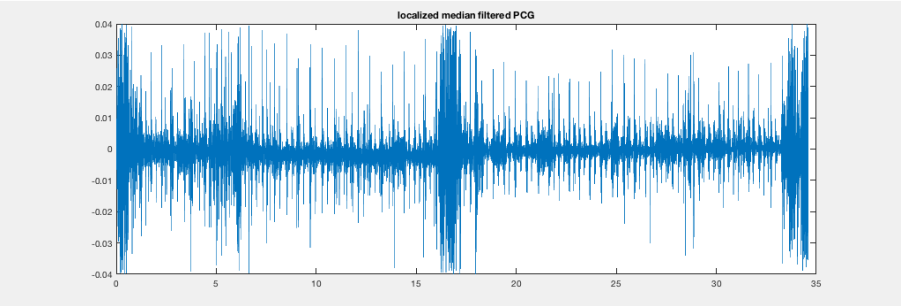


(b) (c)

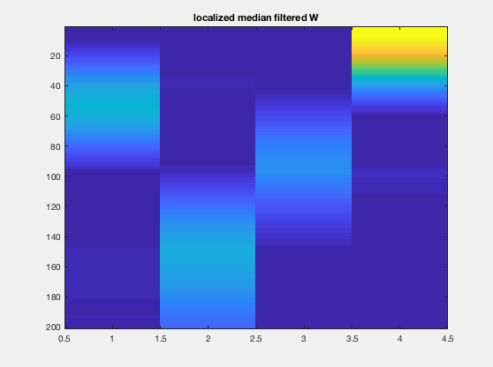
Figure 5.5: *"PCG49577_MV" from Database2 (a). Left nonnegative factor W (b) and right nonnegative factor H.*

Since the noise impulses can be easily identified from the PCG signal and separated from the clean sections, we choose to apply the LMF algorithm with a threshold $M = 0.04$. Such value allows us to mitigate the effects of noise on the NMF factors while leaving unaltered the remaining clean segments.

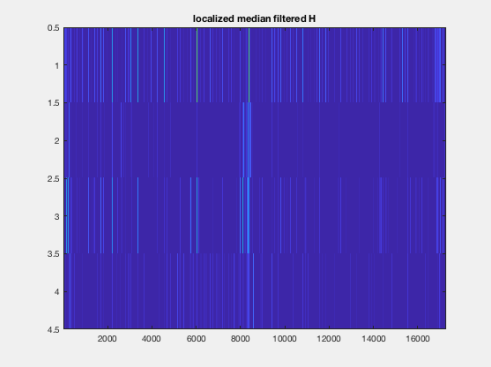
Second iteration. Judging by the NMF factors in Figure (5.6), the previous iteration was successful in addressing the noise impulses. As we can clearly see, now both the first and third basis functions are used to describe cardiac events. The remaining second and fourth are activated to approximate the noise bursts we saw in concomitance of the impulses during the last iteration. In particular, while the second one activates higher frequencies in the interval $[90, 200]$Hz, the fourth activates lower frequencies, around $[0, 60]$Hz. Additionally, by zooming in on the cardiac event-related rows of the right factor, we can also notice the presence of murmurs. The signal is therefore assessed as belonging to both Q-C and Q-B.



(a)



(b)

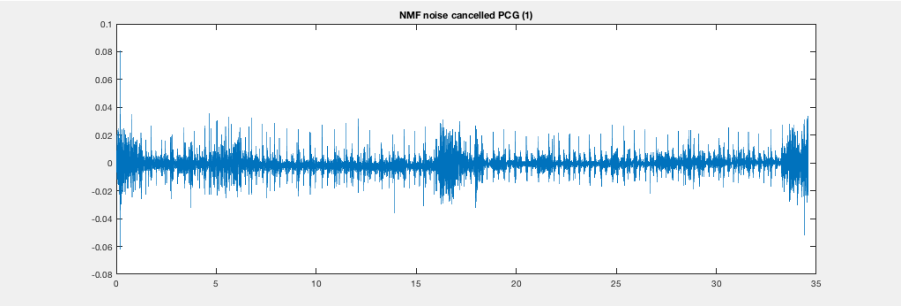


(c)

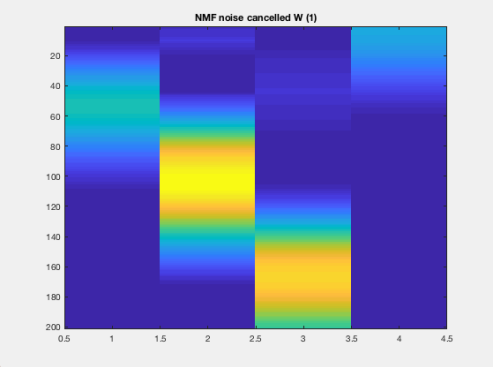
Figure 5.6: *Localized Median Filtered signal from Figure (5.5a) (a) with a threshold* $M = 0.04$. *Left nonnegative factor* W *(b) and right nonnegative factor* H.

Given the presence of a column-row pair associated to noise bursts, we apply the NMF-ANC algorithm in its nonstationary form using the second pair, frequency interval $[90, 200]$Hz and model order $N_h = 500$. The latter was chosen based on the noise duration, while the nonstationarity follows from the relative time-isolation of the main noise bursts.

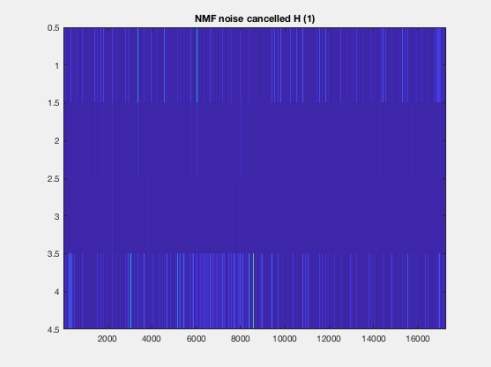
Third iteration. From Figure (5.7) we notice that the high-frequency noise burst has been correctly attenuated. While the first basis function is now allocated to describe cardiac events, the second and third one appear to be describing some localized, leftover higher-frequency component partially related to heart sounds. The remaining fourth basis function plays the same role it did in the previous iteration, namely, approximating the low-frequencies of the noise burst. Given the permanence of minor murmurs, we conclude that the signal still belongs to Q-C and Q-B.



(a)



(b)

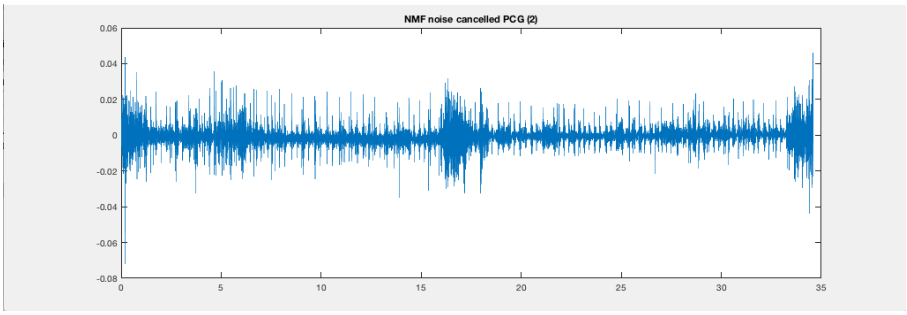


(c)

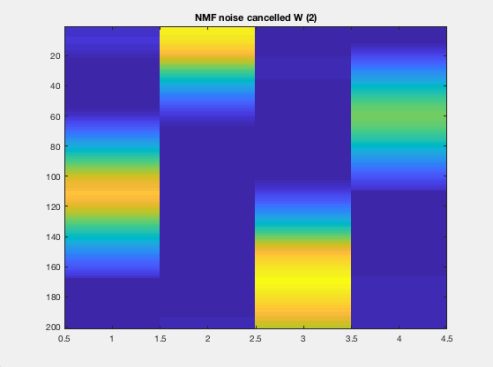
Figure 5.7: *NMF Adaptive Noise Cancelled signal from Figure (5.6a) (a) with noise reference constructed from the second column-row pair of (5.6b), (5.6c), frequency interval* $[90, 200]$*Hz and model order* $N_h = 500$ *for the NSANC. Left nonnegative factor* W *(b) and right nonnegative factor* H*.*

The presence of the fourth column-row pair associated to noise allows us to apply NMF-ANC a second time. In particular, we select the frequency interval $[0, 60]$Hz and model order $N_h = 800$, the latter being higher than in the previous iteration since the noise appears to be more widespread.

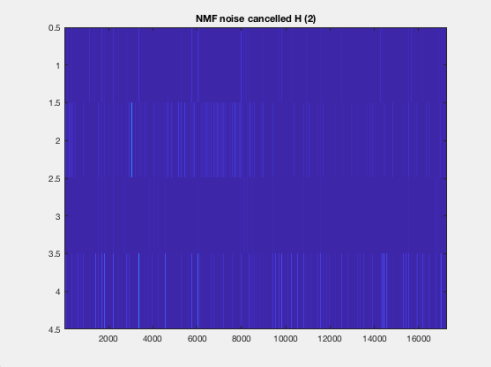
Fourth iteration. Similarly to the previous iteration, the NMF-ANC has successfully attenuated the lower-frequency component of the noise burst. Figure (5.8) shows that the second and fourth basis functions are associated to cardiac events. The first and third, on the other hand, are again activated in correspondence of higher-frequency components partially related to heart sounds given the time-periodic activation of the relative coefficients. Since the cardiac event-related rows are still affected by the presence of minor murmurs, the signal belongs to Q-C and Q-B.



(a)



(b)

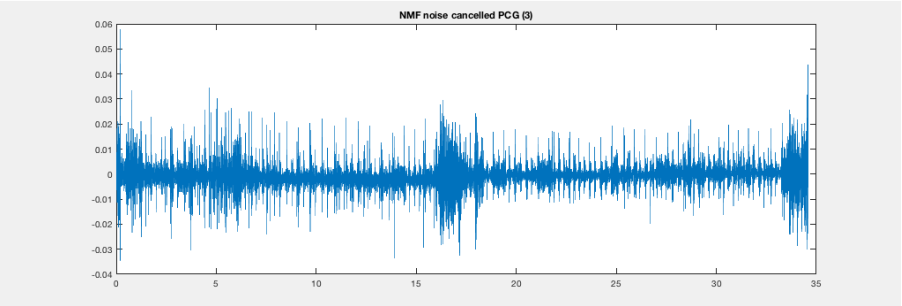


(c)

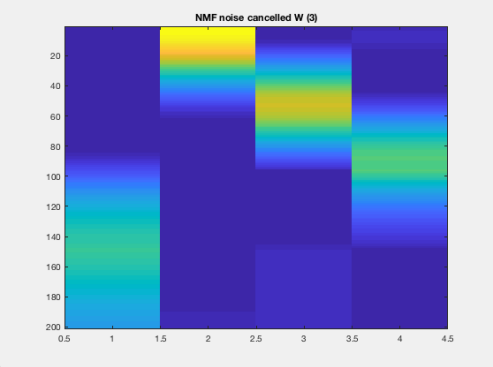
Figure 5.8: *NMF Adaptive Noise Cancelled signal from Figure (5.7a) (a) with noise reference constructed from the fourth column-row pair of (5.7b), (5.7c), frequency interval* $[0, 60]Hz$ *and model order* $N_h = 800$ *for the NSANC. Left nonnegative factor* W *(b) and right nonnegative factor* H*.*

In this iteration we address the higher-frequency disturbances by applying NMF-ANC one last time. In particular, we choose the third column-row pair, frequency interval $[100, 200]$Hz and model order $N_h = 300$.

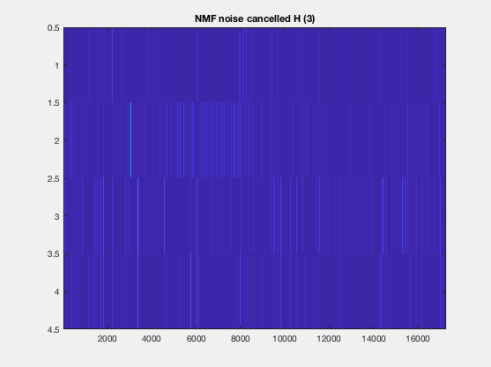
Fifth iteration. The three previous NMF-ANC applications have removed quite a considerable amount of burst noise and other disturbances affecting the original signal. During this process, the rows of the right factor associated to heart sounds got increasingly cleaner and most cardiac events can now be easily detected. Apart from some minor imperfections caused by noise sharing a significant portion of the frequency content characterizing heart sounds, from Figure (5.8) we can observe that the last thing we need to address in order to further improve the detection quality of such rows, are murmurs. The current signal therefore belongs to Q-B.



(a)



(b)

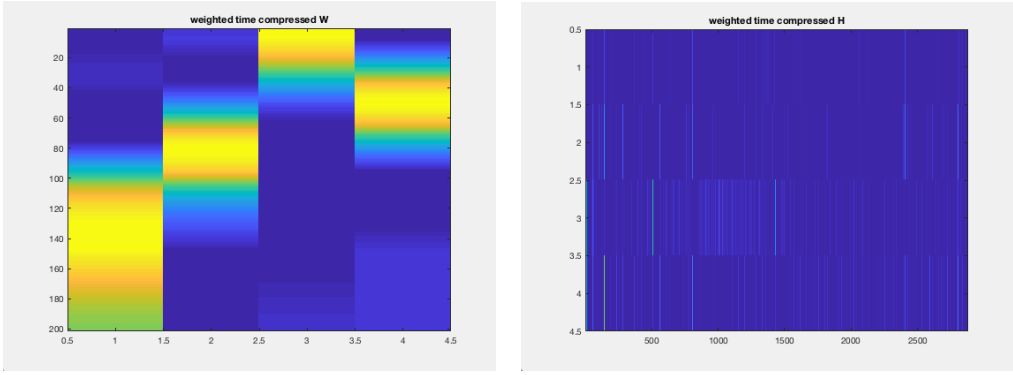


(c)

Figure 5.9: *NMF Adaptive Noise Cancelled signal from Figure (5.8a) (a) with noise reference constructed from the third column-row pair of (5.8b), (5.8c), frequency interval* $[100, 200]Hz$ *and model order* $N_h = 300$ *for the NSANC. Left nonnegative factor* W *(b) and right nonnegative factor* H*.*

As the conclusive step in the denoising process we choose to apply a WTC algorithm with a compression factor of $c = 6$.

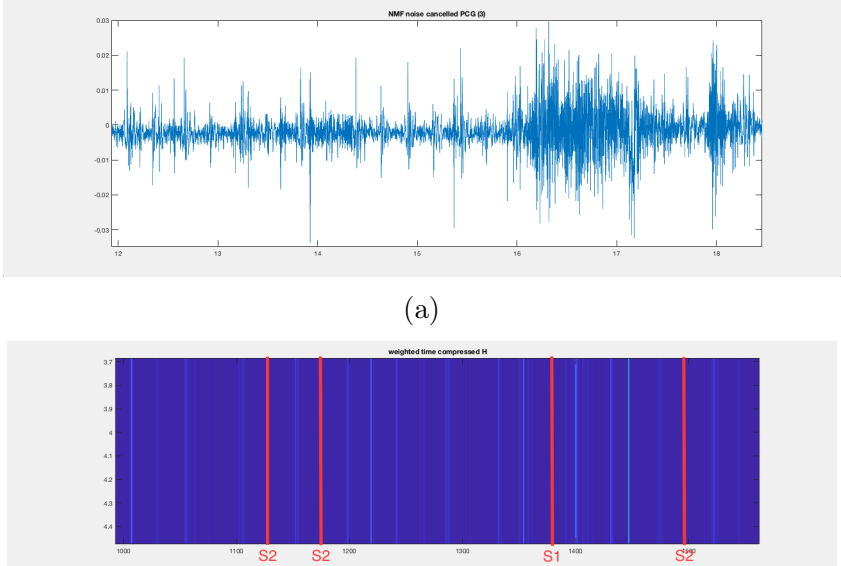
Sixth iteration. The effects on the two nonnegative factors by the previous iteration can be appreciated in Figure (5.10) below. In particular, some of the murmurs have been correctly attenuated in exchange for a loss of detection precision. The second and fourth basis functions are used to describe cardiac events while the remaining ones are activated in correspondence of some leftover noise. Again, due to the presence of minor imperfections, the signal cannot be placed in Q-A. Nevertheless, we decide to stop here the denoising process since a sufficiently accurate representation of cardiac events can be extrapolated from the fourth row of the right factor.



(a) (b)

Figure 5.10: *Weighted Time Compression applied to the spectrogram of signal shown in Figure (5.9a) with a compression factor* $c = 6$. *Left nonnegative factor* W *(b) and right nonnegative factor* H.

From a total of 61 heart beats, only one S1 sound and four S2 sounds cannot be detected from the aforementioned row. It should be noted, however, that such undetected S1 as well as three of the four S2 sounds were contained in the section of the original signal most affected by both noise impulses and noise bursts. The following Figure (5.11) shows the missing activation coefficients as well as the corresponding section of the denoised PCG.



(a)

(b)

Figure 5.11: *Zoom-in on the denoised PCG (a) and fourth row of the right factor (b) containing the undetected cardiac events.*

5.3 Conclusion and further work

As mentioned in the earlier chapters, event detection within PCG signals is a quite challenging task, especially if no additional information is supposed to be given. The wide variety of noises that can affect these audio signals as well as the great variance in contents, often originates the need to develop multiple different techniques to somehow regularize the data before extracting the desired information. While the first aspect was addressed by developing various denoising routines, some more intricate than others, the latter was taken care of by the modelling abilities of the NMF.
In conclusion, the goal we set for ourselves at the very beginning can be considered met, although there are a few facets that still need attention, especially with regard to the proposed NMF-ANC algorithm. The latter, as a matter of fact, expands on the denoising possibilities of the NMF by exploiting the Adaptive Noise Cancelling algorithm in a quite unconventional way. Indeed, in traditional applications the noise reference fed to the ANC is generated *independently* of the noisy signal. This is the case, for example, for most noise cancelling earbuds, where the noise reference is recorded from the surrounding ambient noise. In our work, on the other hand, we extrapolate the reference directly from the noisy signal. To our knowledge, this procedure, which in real applications would require the installation of just one microphone instead of multiple ones, hence cutting hardware costs, is still mostly unexplored. As remarked in previous sections, such denoising process, though particularly interesting from both a mathematical and signal processing point of view, has its weaknesses. The most evident being how to distinguish between noise and clean signal in the case when the two share similar properties. This observation is related to Remark (4.2). Other aspects that will need further investigating are touched by Remark (4.3) and Remark (4.4), namely, how to optimally choose both the ANC model order and stationary/nonstationary form with respect to the noise features. Lastly, it should be noted that the NMF-ANC algorithm can be applied to a much broader class of signals than that of PCGs. As a consequence, ironing out these technical details is of pivotal importance for developing a more general and encompassing denoising algorithm.

References

[1] Amaral L., Glass L., Goldberger A., Hausdorff J., Ivanov P. C., Mark R., Stanley H. E., ... - *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals*, Circulation [Online] 101(23), 215–220 (2000)

[2] Antoniou A. - *Digital filters: analysis, design, and signal processing applications*, McGraw-Hill Education (2018)

[3] Bahrami Rad A., Clifford G., Coimbra M., Costa P., Oliveira J., Elola A., Ferreira C., Jorge A., Nogueira M., Oliveira A. C., Renna F., Reyna M., Sameni R. - *The CirCor DigiScope Phonocardiogram Dataset (version 1.0.3)*, PhysioNet: `https://doi.org/10.13026/tshs-mw03` (2022)

[4] Benetos E., Feng Z., Zhou Q. - *Adaptive Noise Reduction for Sound Event Detection Using Subband-Weighted NMF*, Sensors, 19(14):3206 (2019)

[5] Bertsekas D. - *Nonlinear Programming: Second Edition*, Athena Scientific (1999)

[6] Bertsekas D. - *Corrections for the Book Nonlinear Programming: Second Edition*, `http://www.athenasc.com/nlperrata.pdf` (1999)

[7] Candy J. - *Model-based signal processing*, Wiley-IEEE Press (2005)

[8] Dia N., Duong-Hung P., Fontecave-Jallon J., Meignen S., Rivet B. - *Phonocardiogram Signal Denoising Based on Non-negative Matrix Factorization and Adaptive Contour Representation Computation*, IEEE Signal Processing Letters 25(10), 1475-1479 (2018)

[9] Févotte C., Idier J. - *Algorithms for nonnegative matrix factorization with the beta-divergence*, Neural Computation 23(9), 2421–2456 (2011)

[10] Gillis N. - *Nonnegative Matrix Factorization*, Society for Industrial and Applied Mathematics (2020)

[11] Gillis N. - *Codes and examples from the book Nonnegative Matrix Factorization*, `https://gitlab.com/ngillis/nmfbook/` (2020)

[12] Gordany P., Kazemnejad A., Sameni, R. - *EPHNOGRAM: A Simultaneous Electrocardiogram and Phonocardiogram Database (version 1.0.0)*, PhysioNet: `https://doi.org/10.13026/tjtq-5911` (2021)

[13] Grippo L., Sciandrone M. - *On the convergence of the block nonlinear Gauss–Seidel method under convex constraints*, Operations Research Letters 26(3), 127–136 (2000)

[14] Hong M., Luo Z.-Q., Razaviyayn M. - *A unified convergence analysis of block successive minimization methods for nonsmooth optimization*, SIAM Journal on Optimization 23(2), 1126–1153 (2013)

[15] Hong Kook K., Kwang Myung J., Sung Joo L., Yun Keun L. - *Nonnegative Matrix Factorization Based Adaptive Noise Sensing over Wireless Sensor Networks*, International Journal of Distributed Sensor Networks, 2014(2):1-9 (2014)

[16] Ishimura M., Kitamura D., Mae N., Makino S., Yamada T. - *Ego noise reduction for hose-shaped rescue robot combining independent low-rank matrix analysis and noise cancellation*, Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2016:1-6 (2016)

[17] Lee D. D., Seung H. S. - *Learning the parts of objects by non-negative matrix factorization*, Nature 401(6755): 788–791 (1999)

[18] Oppenheim A., Schafer R. - *Digital Signal Processing*, Pearson (1975)

[19] Paatero P., Tapper U. - *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics 5(2): 111–126 (1994)