SENSING ECG SIGNALS WITH VARIABLE PULSE WIDTH FINITE RATE OF INNOVATION

Gilles Baechler

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer Science and Communication Systems
Laboratory for Audiovisual Communications

Supervisors: Prof. Martin Vetterli
Dr. Ali Hormati

in collaboration with

QUALCOMM INC., SAN DIEGO
Office of the Chief Scientist
Think Tank Group

Thesis reviewer: Dr. Ronald Crochiere

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Communication Systems

March 2012

Acknowledgements

First and foremost, I would like to thank my Professor Martin Vetterli for his guidance, advice and suggestions and most importantly to have provided me the opportunity of doing my Master thesis at Qualcomm Inc. in San Diego. Thanks to my supervisor and friend Ali, whose great help, patience and daily advices made this thesis possible. My thanks also go to Yann, who introduced me to the FRI world and helped me to refresh my signal processing skills at the early stage of the project.

On Qualcomm side, thanks to my managers Chong and John, my supervisor Jim and all my team: Babak, Frank, Jess, Onur and Ron. In particular, I would like to thank Jim, Ron and Frank for the time they dedicated to me and the interesting off-line discussions and conversations I shared with them. I had a wonderful experience at Qualcomm and I really enjoyed my internship in this company. I would also like to thank Thierry Blu and Pina Marziliano for their suggestions and remarks during our biweekly calls.

Last but not least, I thank my parents, Jean-Paul and Thérèse and my siblings, Martial, Benjamin and Magalie for their continuous support throughout my studies. Finally, my thanks go to Valentine for her care, love and unconditional support.

San Diego, March 16, 2012 G. B.

Abstract

Mobile health is gradually taking more importance in our society and the need of new power efficient devices acquiring biosignals for long periods of time is becoming substantial.

In this thesis, we study the power reduction we could achieve on ECG sensing devices. Emphasis is made on reducing the number of samples both during the sensing phase and the compression phase. To that end, a new scheme called variable pulse width finite rate of innovation (VPW-FRI) is investigated. This new technique relies on the classical finite rate of innovation (FRI) theory and enables the use of a sum of asymmetric Cauchy-based pulses to model ECG signals. Research is done in order to implement VPW in practice and its performance are carefully analysed. Among others, we consider the potential instability of the method, we study its compression effectiveness and compare it with compression schemes widespread in the literature. We also evaluate the spectrum extrapolation performance of VPW when fed with signals sampled at sub-Nyquist rates and propose a modification that improves it. Furthermore, we introduce a method based on the similarities between different heart beats that reduces the computational costs of VPW. The parametric nature of VPW finally allows us to use it as a noise reduction algorithm. In parallel, we review and test a non-uniform sensing technique that adapts the sampling rate to the slope of the signal.

The material related to FRI and VPW is patented and proprietary of Qualcomm Inc.

Keywords: variable pulse width, finite rate of innovation, sampling, compressed sensing, electrocardiogram, mobile health, ECG lossy compression

Glossary

Acronyms

ADC	analog-to-digital converter
AR	auto-regression
AV	atrioventicular
AWGN	Additive white Gaussian noise
AZTEC	amplitude zone time epoch coding
CORTES	coordinate reduction time encoding system
CT	cosine transform
DC	direct-current
DCT	discrete cosine transform
DFT	discrete Fourier transform
DPCM	differential pulse code modulation
ECG	electrocardiogram
ESPRIT	estimation of signal parameters via rotational invariance techniques
FAN	fan-out oriented
FRI	finite rate of innovation
FT	Fourier transform
HMM	hidden Markov models
KLT	Karhunen-Loeve transform
LS	least squares
LVQ	learning vector quantization
MAP	maximum a posteriori
RBFN	radial basis function neural networks
SA	sinoatrial
SAPA	scan-along polygonal approximation
SNR	signal-to-noise ratio

SRR	signal to residual ratio
SVD	singular value decomposition
TEO	Teager energy operator
TLS	total least squares
TP	turning point
VPW	variable pulse width
VPW-FRI	variable pulse width finite rate of innovation

List of symbols

$[.]$	discrete indexing
$(.)$	continuous indexing
$a, b, c, \ldots$	scalars
$A, B, C, \ldots$	constant scalars
$\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$	vectors
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$	matrices
δ	Dirac delta function
$\mathbb{Z}, \mathbb{R}, \mathbb{C}$	integer, real and complex numbers
$\mathrm{sign}(x)$	signum function (equal to 1 if $x > 0$, -1 if $x < 0$ and 0 if $x = 0$)
i	imaginary unit defined such that $i^2 = -1$
$\Re\{.\}$	real part
$\Im\{.\}$	imaginary part
$.^*$	complex conjugate
$\angle$	phase or argument of a complex number
$\vert . \vert$	absolute value or magnitude of a complex number
$\lfloor . \rfloor$	floor function
$\Vert . \Vert$	L^2-norm
$.^T$	transpose of a matrix or a vector
$.^H$	Hermitian transpose or conjugate transpose of a matrix
$.^{-1}$	inverse of a matrix
pinv	pseudo-inverse of a matrix
$*$	convolution product
$\mathrm{diag}(\boldsymbol{X}, y)$	vector containing the diagonal values of the matrix $\boldsymbol{X}$ located y positions above the main diagonal (below if $y < 0$)
$\mathrm{toeplitz}(\mathbf{x}, \mathbf{y})$	Toeplitz matrix with first column defined as the vector $\mathbf{x}$ and first row as $\mathbf{y}$

Contents

1 Introduction

1.1 Context

Mobile health, also called mHealth is a recent term describing the use of mobile communication technologies and networks to deliver healthcare. In the past decades, the field of communications and mobile technologies have undergone tremendous progresses: the technology is evolving fast, devices are getting smaller, more popular and more affordable while their computational power and functionalities keep increasing. At the same time, people are getting more sensitive to their own health and the population is progressively more receptive to the use of mobile devices to record their health and daily lives: mobile equipment can now be used to record our heart beats while running, more and more sensors are now embedded in mobile phones, and a huge amount of applications is available to take advantage of them.
All these factors point towards the same direction: mHealth is progressively becoming the next big thing!

In parallel, cardiovascular diseases remain the biggest cause of death in the world. According to the World Health Organization[1], about 17.3 million people died from a cardiovascular disease in 2008: that represents 30% of the total number of deaths and it is expected to increase to 26.3 million by 2030! These numbers justify the need to further develop and provide efficient, low-power and low-cost methods to continuously monitor and carefully analyse the heart beats of sensitive persons.

[1] http://www.who.int

1.1.1 Global project and thesis outline

This thesis is part of a large project, launched at Qualcomm Inc. during the Spring 2011. The aim of this project is to investigate the feasibility and development of low-power end-to-end portable systems for acquisition of biosignals. In a first phase, the efforts are concentrated on the design of a device to sense ECG signals, as displayed in Figure 1.1. The system is made of a lightweight patch and a mobile unit, with more power capabilities than the patch. On the patch, analog sensors acquire the electric signal coming from the heart. The acquisition is followed by an analog-to-digital converter (ADC) (possibly preceded with some analog filtering). The data is processed on the patch and sent via a radio channel to a mobile device (cell phone or other independent device). With more power available, the device can perform operations that were not possible on the patch, and if necessary stores the signals or forwards them to the cloud.

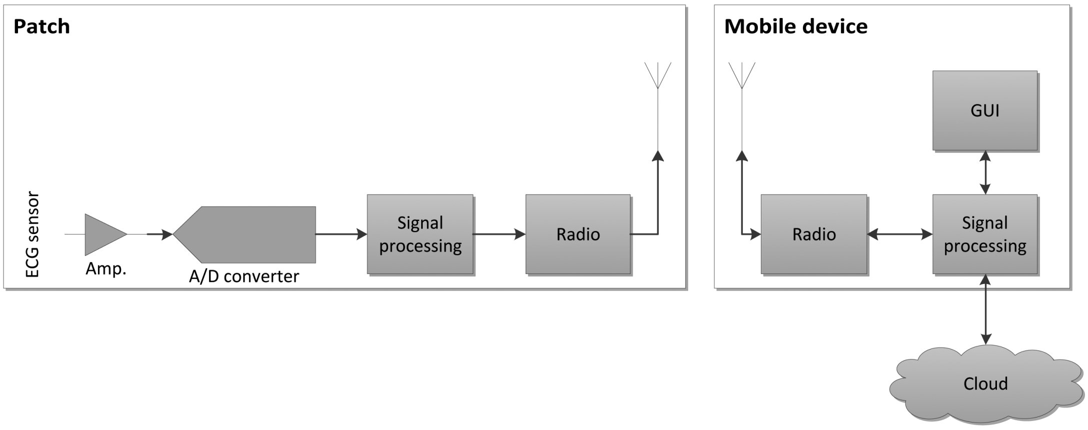


Figure 1.1: End to end system designed to acquire ECG signals

The core of this thesis is focussed on the signal processing units (on the patch and on the mobile device) and the main objective is to investigate new and innovative ways to sample and compress ECG data. In particular, research is done in order to minimize the amount of information sampled by the patch and transmitted between the patch and the mobile device but also between the mobile device and the cloud (or some storage unit).

The thesis is organized as follows. The first chapter introduces the context, a medical background about ECGs and the mathematical tools necessary for a good understanding of the topics covered in the next chapters, and in particular the theory of finite rate of innovation (FRI). Chapter 2 explores the state of the art techniques in ECG compression and introduces a new scheme: the variable pulse width finite rate of innovation (VPW-FRI). An extensive analysis of this very recent method is proposed in Section 4.3. Chapter 3 covers power efficient sampling, and some methods relying on non-uniform sampling are reviewed. We also try to lower the sampling rate using some of the methods exposed in Chapter 2. We describe an improvement that enhances the performance of the variable pulse width (VPW) method at sub-Nyquist sampling rates in Section 3.2.2. Practical implementation issues and in depth analysis of the algorithms introduced in Chapters 2 and 3 are described in Chapter 4. The algorithms are then compared and their performance is evaluated in Chapter 5. Potential future work on the topic and the conclusion are finally exposed in Chapter 6.

Throughout the thesis, emphasis is made on FRI theory and particularly VPW. We first elaborate on the mathematical model of VPW and investigate its properties. Next, new applications and practical uses are proposed and analysed. A full-fledged routine to process long ECG records is described and two main improvements to VPW-FRI are exposed: one of them enhances its frequency spectrum interpolation accuracy and the other one reduces its running-time.

1.2 ECG signals: physiological overview

This section (based on [24, 55]) provides the necessary background for a basic understanding of the heart activity and its interpretation as electrical signals.

First let us have a look at the heart anatomy (cf. Figure 1.2). The heart is composed of two main parts, separated by the septum. These two parts are in turn subdivided into two chambers: the atrium (upper part), and the ventricle (lower part). The role of the left part (displayed on the right in Figure 1.2) is to pump oxygenated blood from the lungs and send it to the body through the aorta. On its way back, the deoxygenated blood re-enters the heart by flowing into the right atria and is sent back to the lungs by the right ventricle through the pulmonary arteries.

The right atrium also contains two nodes, namely the sinoatrial (SA) node located on top of it and the atrioventicular (AV) node, situated at the bottom. The SA node controls the atria contractions while the AV node controls the ventricles and is connected to them with *branches* via the septum.

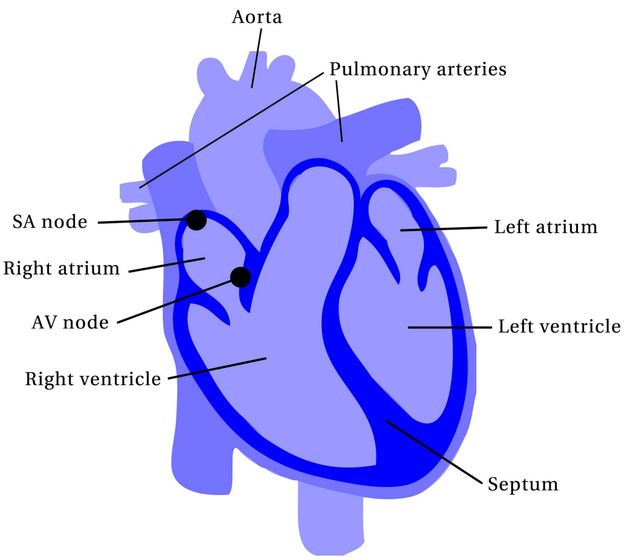


Figure 1.2: Anatomy of the heart

The nodes are responsible to trigger and coordinate the contraction of the different chambers by propagating an electric signal through the cells. A vector can be assigned to each group of cells depending on the change of polarization. A *dominant vector* characterizes the *global* depolarization wave and is defined as the sum of all the vectors of the individual cells and can be sensed at the surface of the skin using electrodes. A positive voltage is sensed for a dominant vector moving towards the location of the electrode and a negative voltage is recorded for a vector moving away from the electrode. Therefore the shape of the captured voltage varies depending on the locations of the electrodes on the skin.

There are 10 standard locations for the electrodes: 2 on the arms (labelled *RA* and *LA*), 2 on the legs (*RL* and *LL*) and 6 on the chest (V_1 to V_6). Using these 10 locations, we can generate 12 standard resulting signals named *leads*. A lead can be composed of the difference of two electrodes (for example lead I is defined as the difference of voltage between the right and left arm, $LA - RA$). Taking the difference amplifies the signal related to the heart activity and minimizes the noise and is necessary for the electrodes that are located away from the heart (on the limbs). For the electrodes positioned close to the heart, a lead simply consists of the signal recorded on a single electrode. One can see the different leads as different angles of view on the dominant vector.

Every cardiac cycle consists of two phases for both the atria and the ventricles: the *activation* and the *recovery*, also known as *repolarization* and *depolarization*. A full cardiac cycle with the polarization of the cells and the direction of the resulting vector is illustrated in Figure 1.3. The ECG displayed in the middle is lead II[2] from the left leg to the right arm ($LL - RA$), meaning that a dominant vector pointing towards the left leg is recorded with a positive voltage and a vector pointing towards the left arms results in a negative voltage. At the beginning of the cycle, all the cells are at rest, which is translated to the baseline of the ECG. The start is initiated by the SA node, which plays the role of a pacemaker. The electrical signal polarizes the atria and its dominant vector points towards the AV node, which results into a positive voltage, called the *P wave*. Its size is relatively small since the atria muscles have a small mass compared to the ventricles muscles. The depolarization phase results in the contraction of the left and right atria, which has the effect of sending blood to the ventricles. At the end of the P wave, when all the atria cells are depolarized, the resulting ECG signal is back to its baseline state.

Then the AV node starts depolarizing, along with its branches. The electric wave propagates through the septum and reaches first the right ventricle: being smaller and having thinner walls than the left one, it depolarizes faster. As a consequence, the dominant vector first points towards the right ventricle and gradually moves towards the left ventricle, which results in a positive voltage that becomes negative when the dominant vector moves away from the direction of the left leg.
The depolarization of the ventricles results in the ECG in what is called the *QRS complex*, composed of the *Q wave*, the *R wave* (the large peak) and the *S wave*, each of them being defined by the state of the dominant vector. During this phase, the blood present in the ventricles is ejected and sent through the body. Observe that in Figure 1.3, the Q wave is almost absent: that is due to the fact that the dominant vector is perpendicular to the electrodes axis. The shape of the QRS relies on the position of the electrodes but also on the *electrical axis* [24], which can be seen as the main direction of the QRS complex and varies from person to person.
It should be noted that the repolarization of the atria occurs at the same time as the depolarization of the ventricles, and since their muscular mass is much smaller, its effect is completely masked by the QRS complex.
When both ventricles are completely depolarized, the heart is in a resting state for a short period of time and the resulting vector is null before the repolarization of the

[2] To be exact, it is a *modified* lead II, with electrodes placed at equivalent positions relative to the heart but much closer to it than the limbs

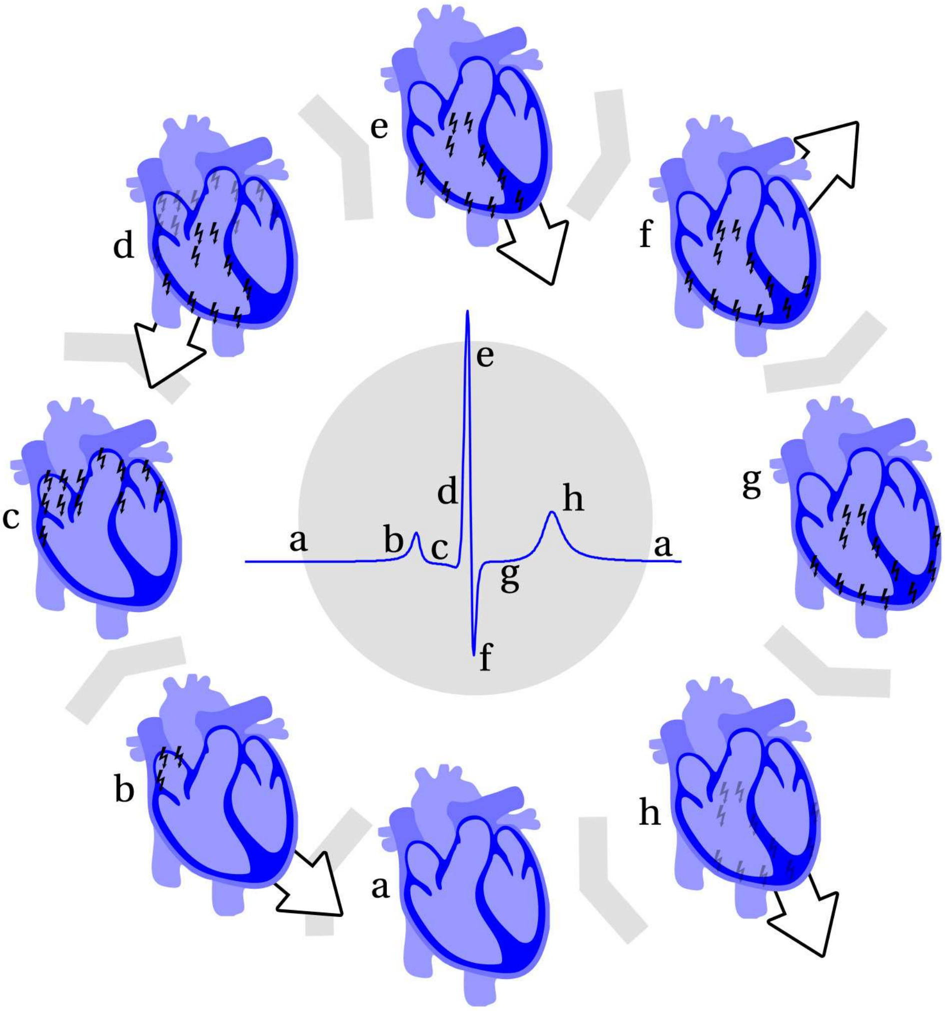


Figure 1.3: One cardiac cycle. Each state of the heart is mapped to its corresponding wave on the lead II ECG signal using the letters a to h. The direction of the dominant vector is represented by the white arrows and the electrodes axis goes from the top left of the heart (negative) to the bottom right (positive). The depolarized cells are illustrated with black ↯ and the repolarizing cells are illustrated with gray ↯.

ventricles occurs. The repolarization can be seen on the ECG record as a positive wave, also named the *T wave*. Once the T wave ends, all of the heart cells are polarized again and a new cycle can start. Figure 1.4 illustrates the different pulses in a period of an ECG and their corresponding letter.

We do not discuss the diagnostic of the different pathologies and potential malfunctions of the heart and their interpretation here. The reader interested in more information about it should refer to [24, 19].

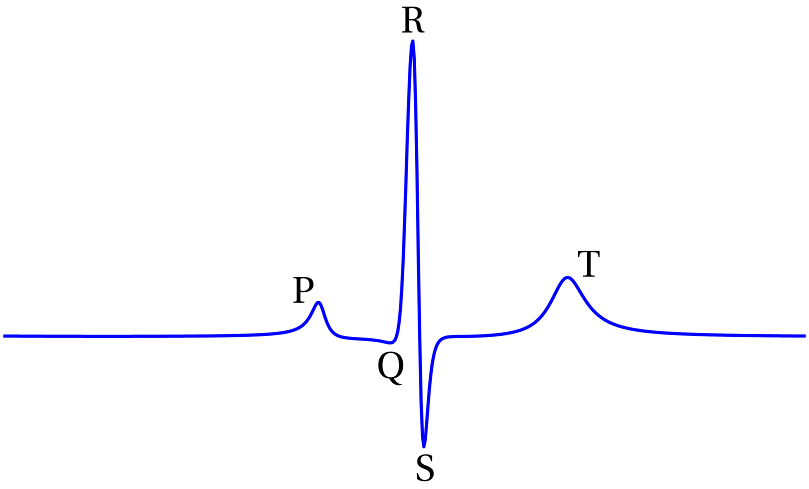


Figure 1.4: Name of the different waves composing the period of an ECG.

1.3 Signals with finite rate of innovation

This section provides a first contact with the theory of FRI. The classical sampling theory pioneered by Kotel'nikov [36], Nyquist [48], Shannon [54] and Whittaker [61] relies on the assumption that the signals to be acquired are band-limited to a maximum frequency: the *Nyquist frequency*. Even if this hypothesis does not hold, the signals can simply be low-pass filtered before being sampled at a rate at least twice the Nyquist frequency.

FRI theory is based on a different assumption: the signals acquired must have *finite number of parameters per unit of time*. These parameters are seen as *innovations*, hence the name FRI. FRI sampling works by using non-linear techniques to recover these degrees of freedom. Unlike the conventional Shannon-Nyquist sampling theory, signals sampled with FRI are not restricted to a finite support in the frequency domain. A periodic stream of Dirac pulses is a good example of such a signal. Another advantage of FRI is that it reuses two of the main building blocks of the conventional sampling theory: low-pass filtering and uniform sampling.

We present here an overview of the theory behind FRI. It is based on [10, 28, 43, 58] and introduces, through a simple example, the concepts required to understand some of the methods presented in this report. The reader already familiar with FRI could skip this Section. For more detailed information about FRI, please refer to [10, 58].

1.3.1 Rate of innovation of a signal

Before diving into the mathematics and the algorithms, let us introduce the class of signals used in the FRI framework. As stated above, FRI requires signals to have a *finite degrees of freedom per unit of time*. This concept is formalized in what follows.

Let $C_x(t_1, t_2)$ be a function that counts the number of degrees of freedom of a signal $x(t)$ in the interval $[t_1, t_2]$. The *rate of innovation* ρ is defined as [58]

$$\rho = \lim_{\tau \to \infty} \frac{1}{\tau} C_x\left(-\frac{\tau}{2}, \frac{\tau}{2}\right). \tag{1.1}$$

For most practical applications, another notion is introduced, the *local rate of innovation*, defined over an interval of length τ as follow:

$$\rho_\tau(t) = \frac{1}{\tau} C_x\left(t - \frac{\tau}{2}, t + \frac{\tau}{2}\right). \tag{1.2}$$

Example: periodic stream of Diracs FRI is not a generic method and its algorithms need to be adapted for every different class of input signal. The best way to introduce FRI is probably through an example. The typical signal to which FRI is applied is a τ-periodic stream of K Diracs sampled at a rate of $f_s = \frac{N}{\tau}$ $(N > 2K)$:

$$x(t) = \sum_{n \in \mathbb{Z}} \sum_{k=0}^{K-1} c_k \delta(t - t_k - n\tau), \tag{1.3}$$

where $\{t_k\}_{k=0}^{K-1}$ represent the locations of the pulses and $\{c_k\}_{k=0}^{K-1}$ their amplitude. An example of a stream of Diracs is illustrated in Figure 1.5. Since Dirac pulses have an infinite support in the Fourier domain, it is impossible to sample them and perfectly reconstruct them using the classical Shannon-Nyquist theorem. However, as they have a few number of parameters per unit of time (low rate of innovation), they make a very good candidate for FRI.

The goal of FRI is to recover the different parameters (t_ks and c_ks) in order to perfectly reconstruct the original signal by using a finite number of Fourier series coefficients.

As there are $2K$ free parameters (K positions and K amplitudes) per period, the rate of innovation of $x(t)$ is

$$\rho = \frac{2K}{\tau}. \tag{1.4}$$

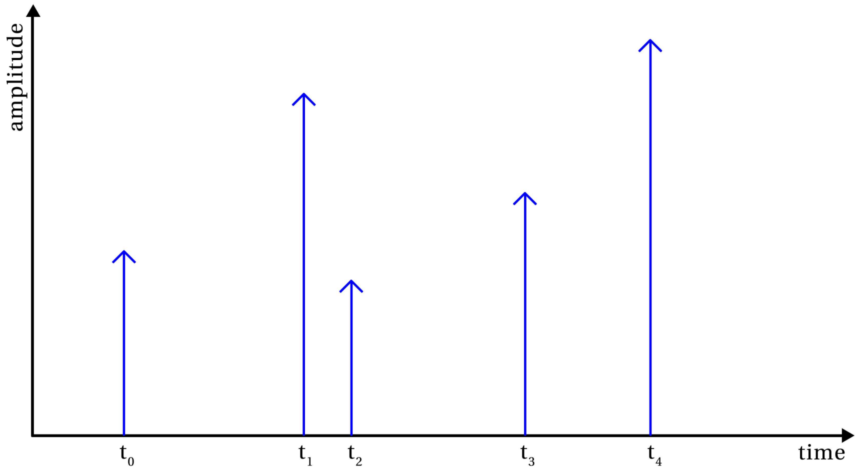


Figure 1.5: Example of a stream of 5 Diracs.

We can compute its Fourier Series Coefficients as

$$\begin{aligned} X[m] &= \frac{1}{\tau}\int_0^{\tau} x(t)e^{-i2\pi tm/\tau}dt \\ &= \frac{1}{\tau}\int_0^{\tau}\sum_{k=0}^{K-1} c_k\delta(t-t_k-n\tau)e^{-i2\pi tm/\tau}dt \\ &= \frac{1}{\tau}\sum_{k=0}^{K-1}\int_0^{\tau} c_k\delta(t-t_k)e^{-i2\pi tm/\tau}dt \\ &= \frac{1}{\tau}\sum_{k=0}^{K-1} c_k e^{-i2\pi t_k m/\tau}, \end{aligned} \tag{1.5}$$

where $m \in \mathbb{Z}$. The FRI reconstruction algorithm requires to know the Fourier series coefficients for at least $2K$ consecutive values. The reason why we need $2K$ or more consecutive values becomes clear later (cf. equation 1.11), but intuitively it makes sense that if we want to recover $2K$ parameters, we need at least $2K$ Fourier samples. Since the Fourier Series coefficients are separated by a distance of $1/\tau$ (due to the τ-periodicity in time of the signal), the minimal bandwidth B has to be larger than $\frac{2K}{\tau}$. As in classical sampling theory, we can apply a low pass filter before sampling $x(t)$ at a bandwidth sufficient to capture enough coefficients.

Annihilating filter and pulse location recovery An *annihilating filter* (as defined in [58]) is a filter which, convolved with a signal, produces a resulting signal of zero. In other words, if $S[m]$ is a signal and $A[m]$ is the corresponding annihilating filter, $(S * A)[m] = 0, \forall m \in \mathbb{N}$.

Proposition 1. *The signal* $S[m] = \sum_{k=0}^{K-1} c_k u_k^m$, *where* $c_k \in \mathbb{R}$, $u_k \in \mathbb{C}$, *is annihilated by the filter (assuming that* $A[0] = 1$*):*

$$\begin{aligned} A(z) &= \sum_{k=0}^{K} A[k] z^{-k} \\ &= \prod_{k=1}^{K} (1 - u_k z^{-1}) \end{aligned} \tag{1.6}$$

Proof. Taken from [58]. We have:

$$\begin{aligned} A[m] * S[m] &= \frac{1}{\tau} \sum_{k=0}^{K} A[k] S[m-k] \\ &= \frac{1}{\tau} \sum_{k=0}^{K} A[k] \sum_{l=0}^{K-1} c_l u_l^{m-k} \\ &= \frac{1}{\tau} \sum_{l=0}^{K-1} c_l u_l^m \sum_{k=0}^{K} A[k] u_l^{-k}. \end{aligned}$$

Using equation (1.6), we know that $\sum_{k=0}^{K} A[k] u_l^{-k} = A(u_l) = 0$. Therefore

$$A[m] * S[m] = 0. \tag{1.7}$$

□

From equation (1.6), we see that the roots of the annihilating filter for the Fourier coefficients of the stream of Diracs $X[m]$ are

$$u_k = e^{-i2\pi t_k/\tau}. \tag{1.8}$$

Therefore, u_k allows us to compute the parameters t_k as

$$t_k = -\frac{\tau \angle u_k}{2\pi}, \tag{1.9}$$

where $\angle u_k$ is the phase of u_k. It remains to show how to find the annihilating filter coefficients. To that end, equation (1.7) can be rewritten as (with $A[0] = 1$)

$$-X[m] = A[1]X[m-1] + A[2]X[m-2] + \cdots + A[K]X[m-K]. \tag{1.10}$$

Hence the annihilating filter can be used to predict the next Fourier coefficient of the signal based on the K preceding coefficients. Thus for every entry, we can write equation (1.10) into the following matrix form:

$$\begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ X[-1] & X[-2] & \cdots & X[-K] \\ X[0] & X[-1] & \cdots & X[-K+1] \\ \vdots & \vdots & \ddots & \vdots \\ X[K-1] & X[K-2] & \cdots & X[0] \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \begin{bmatrix} A[1] \\ A[2] \\ \vdots \\ A[K] \end{bmatrix} = - \begin{bmatrix} \vdots \\ X[0] \\ X[1] \\ \vdots \\ X[K] \\ \vdots \end{bmatrix}.$$

The infinite matrix above has a *Toeplitz form*: every diagonal from left to right is made of the same coefficients. Observe that by knowing at least $2K$ coefficients, we can fully recover the annihilating filter. For example with $X[m]$ for $m = -K+1, \dots, K$, we find the filter A by solving the linear system of equations below:

$$\begin{bmatrix} X[0] & X[-1] & \cdots & X[-K+1] \\ X[1] & X[0] & \cdots & X[-K+2] \\ \vdots & \vdots & \ddots & \vdots \\ X[K-1] & X[K-2] & \cdots & X[0] \end{bmatrix} \begin{bmatrix} A[1] \\ A[2] \\ \vdots \\ A[K] \end{bmatrix} = - \begin{bmatrix} X[1] \\ X[2] \\ \vdots \\ X[K] \end{bmatrix}. \tag{1.11}$$

Of course if more than $2K$ coefficients are available, we can make the above matrix taller and add stability to the system. It is particularly recommended to do so when the input signal contains a high amount of noise.

Pulse amplitude recovery Now that we have the locations of the pulses, it remains for us to find their amplitudes. Taking K consecutive Fourier coefficients (or more) and letting

$$\boldsymbol{V} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ u_0 & u_1 & \cdots & u_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_0^{K-1} & u_1^{K-1} & \cdots & u_{K-1}^{K-1} \end{bmatrix}, \tag{1.12}$$

$$\boldsymbol{x} = [X[0], X[1], \dots, X[K-1]]^T \tag{1.13}$$

and

$$\boldsymbol{c} = [c_0, c_1, \ldots, c_{K-1}]^T,$$

we can represent equation (1.5) with the following set of linear equations:

$$\frac{1}{\tau}\boldsymbol{V}\boldsymbol{c} = \boldsymbol{x}. \tag{1.14}$$

The *Vandermonde matrix* $\boldsymbol{V}$ is non-singular as $u_i \neq u_j$ for any $i \neq j$. Therefore, the amplitudes $\boldsymbol{c} = \{c_k\}_{k=0}^{K-1}$ can be recovered by

$$\boldsymbol{c} = \tau \boldsymbol{V}^{-1}\boldsymbol{x}. \tag{1.15}$$

Note that the set of coefficients $\boldsymbol{x}$ we choose for the analysis does not necessarily need to be located around the DC component. In fact, any set of consecutive coefficients will work, but the most natural set of the Fourier coefficients is usually $X[m]$ for $m = -K-1, \ldots, K$, as it can minimize the frequency at which the signal is sampled.

Noisy case

In the presence of noise, it is important to first *denoise* the matrix of equation 1.11. Some denoising methods are described in Section 1.4.

1.3.2 Periodic nonuniform spline

We see here how the above method can be generalized to periodic nonuniform splines (with period τ).
A signal is a periodic nonuniform spline of degree R with knots at $\{t_k\}_{k=0}^{K-1} \in [0, \tau]$ if and only if its $(R+1)$th derivative is a periodic stream of K Diracs, that is

$$x^{R+1}(t) = \sum_{n\in\mathbb{Z}} \sum_{k=0}^{K-1} c_k \delta(t - t_k - n\tau).$$

Figure 1.6 illustrates an example a part of a non-uniform spline of degree 1. As we can see, its second derivative is a stream of Diracs. Since we are back to a stream of Diracs like in equation (1.3), we can recover its degrees of freedom using the same approach as in the previous Section.

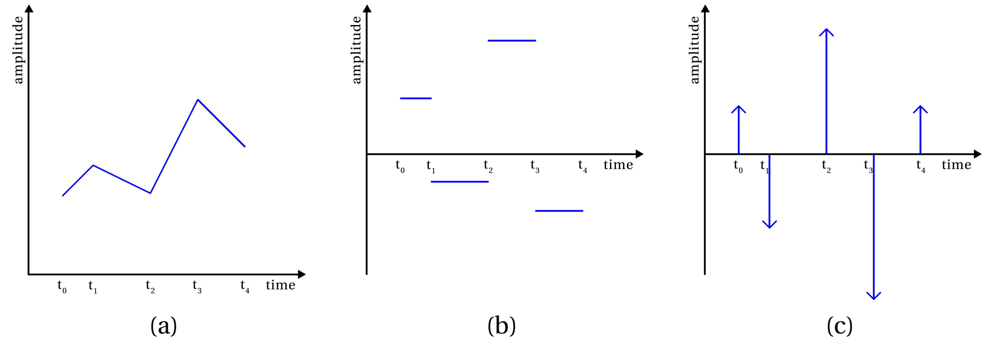


Figure 1.6: (a) Part of a non-uniform spline of degree 1, (b) its first and (c) second derivatives.

1.4 Denoising methods for ECG

Most of the time, the signals used in practice do not perfectly match the underlying model assumed by FRI. Indeed, they usually contain a certain amount of noise and there might be some model mismatch as well. Some of the most common sources of noise in ECGs include (but are not limited to) power-line interferences, electromagnetic interferences, muscle and motion artefacts or respiration.

Noisy signals are not likely to be modelled with a finite number of parameters and that is the reason why the signal needs to be preprocessed before. This section introduces some of the techniques used in the FRI framework to reduce the presence of noise.
The methods presented below also address the problem of model mismatch: indeed, real data almost never matches perfectly the pulse shapes expected by FRI.
One can see the denoising block as the analogue to the low-pass filtering in the case of the Shannon-Nyquist sampling theory: a preprocessing stage in order to fit the data into the underlying model.

1.4.1 Cadzow denoising

Cadzow denoising [13] is the most commonly used method for FRI denoising and is also reviewed, among others, in [10, 28]. Cadzow's method is an iterative algorithm that directly works on the Fourier coefficients $X[m]$ used by FRI to find the annihilating filter. Let $\hat{\boldsymbol{F}}$ be the Toeplitz-shaped matrix of measurements (noisy coefficients) defined as

$$\hat{\boldsymbol{F}} = \begin{bmatrix} \hat{X}[L] & \hat{X}[L-1] & \cdots & \hat{X}[0] \\ \hat{X}[L+1] & \hat{X}[L] & \cdots & \hat{X}[1] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{X}[M] & \hat{X}[M-1] & \cdots & \hat{X}[M-L] \end{bmatrix},$$

where $\hat{X}[m]$ ($m = 0, \ldots, M-1$ and $M \geq 2K$) are the Fourier coefficients of the input signal used for analysis and $L = \lfloor \frac{M}{2} \rfloor$. Note that when $M > 2K$, this matrix is always square (when M is even) or almost square (when M is odd), unlike the extension of the matrix on equation 1.11, which becomes taller. Also observe that here the coefficients are chosen only for positive frequencies, but the technique is easily generalized to any set of coefficients.

Similarly, let us define F as the matrix of noise-free coefficients:

$$\boldsymbol{F} = \begin{bmatrix} X[L] & X[L-1] & \cdots & X[0] \\ X[L+1] & X[L] & \cdots & X[1] \\ \vdots & \vdots & \ddots & \vdots \\ X[M] & X[M-1] & \cdots & X[M-L] \end{bmatrix},$$

where $X[m] = \sum_{k=0}^{K-1} c_k u_k^m$ (which follows the model assumed by FRI). In the case of the stream of Diracs, $u_k = e^{i2\pi t_k/\tau}$ but, as we see later, u_k can take different values and this analysis can be generalized to different classes of signals.

$\boldsymbol{F}$ can be decomposed as follows [28]:

$$\boldsymbol{F} = \underbrace{\begin{bmatrix} u_0^L & u_1^L & \cdots & u_{K-1}^L \\ u_0^{L+1} & u_1^{L+1} & \cdots & u_{K-1}^{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_0^M & u_0^M & \cdots & u_{K-1}^M \end{bmatrix}}_{\boldsymbol{C}} \underbrace{\begin{bmatrix} c_0 & 0 & \cdots & 0 \\ 0 & c_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{K-1} \end{bmatrix}}_{\boldsymbol{D}} \underbrace{\begin{bmatrix} 1 & u_0^{-1} & \cdots & u_0^{-L} \\ 1 & u_1^{-1} & \cdots & u_1^{-L} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_{K-1}^{-1} & \cdots & u_{K-1}^{-L} \end{bmatrix}}_{\boldsymbol{E}}, \quad (1.16)$$

where $\boldsymbol{C}$ and $\boldsymbol{E}$ have Vandermonde structures and $\boldsymbol{C}$ is a diagonal matrix. Since the u_ks are all distinct (recall that we assume that we have K *different* pulses), the rank of $\boldsymbol{C}$ and $\boldsymbol{E}$ is K, which implies that $\boldsymbol{F}$ is also of rank K.

However, the rank of the noisy matrix $\hat{\boldsymbol{F}}$ is greater than or equal to K. Cadzow denoising alternatively ensures that (a) the rank of $\hat{\boldsymbol{F}}$ is K and (b) $\hat{\boldsymbol{F}}$ is a Toeplitz matrix as follows:

(a) The rank of the matrix is reduced by computing the singular value decomposition (SVD) of $\hat{\boldsymbol{F}}$ and keeping only its K largest singular values. In other words

$$\hat{\boldsymbol{F}}_{\boldsymbol{K}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{K}}\hat{\boldsymbol{V}}^H, \tag{1.17}$$

such that $\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^H$ is the SVD of $\hat{\boldsymbol{F}}$ and $\hat{\Sigma}_K(i,i) = \hat{\Sigma}(i,i)$ for $i = 0,\dots K-1$ and 0 otherwise (assuming that the singular values are sorted in descending order). It can be shown that $\hat{\boldsymbol{F}}_{\boldsymbol{K}}$ minimizes the Frobenius norm of the difference between $\hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{F}}_{\boldsymbol{K}}$ under the constraint that $\operatorname{rank}(\hat{\boldsymbol{F}}_{\boldsymbol{K}}) = K$. This result is known as the *Eckart-Young theorem* and a proof of it can be found in [20].

(b) The Toeplitz form of $\hat{\boldsymbol{F}}_{\boldsymbol{K}}$ is ensured by averaging the entries of each diagonal as follows:

$$\hat{\boldsymbol{F}}_{\boldsymbol{T}} = \begin{bmatrix} \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},0)) & \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},1)) & \cdots & \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}}^{(a)},L)) \\ \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},-1)) & \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},0)) & \cdots & \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},L-1)) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},L-M)) & \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},L-M+1)) & \cdots & \operatorname{mean}(\operatorname{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{K}},0)) \end{bmatrix},$$

where mean($\boldsymbol{x}$) computes the average of the vector $\boldsymbol{x}$ and diag($\boldsymbol{X}, y$) extracts the diagonal values of the matrix $\boldsymbol{X}$ located y positions above the main diagonal (below if $y < 0$). We can see that $\hat{\boldsymbol{F}}_{\boldsymbol{T}}$ also minimizes the Frobenius norm of the difference between $\hat{\boldsymbol{F}}_{\boldsymbol{K}}$ and $\hat{\boldsymbol{F}}_{\boldsymbol{T}}$ under the constraint that $\hat{\boldsymbol{F}}_{\boldsymbol{T}}$ is Toeplitz. Indeed, minimizing the Frobenius norm is the same as minimizing the sum of the squares of all entries of the difference-matrix. The constraint fixes a single value for each diagonal. Therefore, we can treat every diagonal separately, and the value that minimizes the sum of the squares of each diagonal is simply the mean of the diagonal entries.

Stopping criterion and convergence The algorithm usually stops after the ratio of the Kth and the $(K+1)$th singular value falls below some fixed threshold. It has been shown that the procedure described above always converges to a matrix satisfying the conditions stated above [13]. The pseudo-code of this method can be found in Algorithm 1.

Extensive research has already been made in order to make Cadzow faster and

Algorithm 1 Cadzow denoising

```
Input: a vector of measurements $\hat{\boldsymbol{x}}$ of size $M$, a threshold $t$ used as stopping criterion.
Output: a vector $\boldsymbol{x}$ such that Cadzow conditions are verified.
  $r = 1$
  $M = \text{length}(\hat{\boldsymbol{x}})$
  $L = \lfloor \frac{M}{2} \rfloor$
  $\hat{\boldsymbol{x}}_{\boldsymbol{r}} = [\hat{x}[L], \hat{x}[L-1], \dots, \hat{x}[1]]$
  $\hat{\boldsymbol{x}}_{\boldsymbol{c}} = [\hat{x}[L], \hat{x}[L+1], \dots, \hat{x}[M]]$
  $\boldsymbol{F} = \text{toeplitz}(\hat{\boldsymbol{x}}_{\boldsymbol{c}}, \hat{\boldsymbol{x}}_{\boldsymbol{r}})$
  while $r > t$ do
      $\{\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}\} = \text{svd}(\boldsymbol{F})$
      $r = S[K+1, K+1] / S[K, K]$
      $S[K+1 : \text{end}, K+1 : \text{end}] = 0$
      $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^H$
      for $i = 1 \to L$ do
          $\hat{x}_r[i] = \text{mean}(\text{diag}(\boldsymbol{F}, i))$
      end for
      for $i = 1 \to M - L$ do
          $\hat{x}_c[i] = \text{mean}(\text{diag}(\boldsymbol{F}, -i))$
      end for
      $\boldsymbol{F} = \text{toeplitz}(\hat{\boldsymbol{x}}_{\boldsymbol{c}}, \hat{\boldsymbol{x}}_{\boldsymbol{r}})$
  end while
  return $\boldsymbol{x} = \left[\boldsymbol{F}[1, \text{end} : -1 : 2]^T ; \boldsymbol{F}[:, 1]\right]$
```

more efficient. Among others, we can mention [6].

1.4.2 ESPRIT algorithm

Estimation of signal parameters via rotational invariance techniques (ESPRIT) is another denoising algorithm that is based on some of the same assumptions as Cadzow. ESPRIT directly outputs the roots of the annihilating filter and works as follows.

Like in Cadzow, $\hat{\boldsymbol{F}}$ is reduced to $\hat{\boldsymbol{F}}_{\boldsymbol{K}}$ of rank K using equation (1.17). In the noiseless case, we have

$$\begin{aligned}\boldsymbol{F} &= \boldsymbol{C}\boldsymbol{D}\boldsymbol{E} \\ &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H,\end{aligned}$$

where matrices $\boldsymbol{C}$, $\boldsymbol{D}$ and$\boldsymbol{E}$ are defined in equation (1.16) and $\boldsymbol{U\Sigma V}^H$ is the SVD of $\boldsymbol{F}$. First let us show that the subspace spanned by the matrix $\boldsymbol{F}$ is the same as the one spanned by the first K columns of $\boldsymbol{U}$, denoted $\boldsymbol{U_K}$. We can prove this as follows:

$$\begin{aligned}\text{colspan}\,(\boldsymbol{F}) &= \{\boldsymbol{F\alpha}, \forall \boldsymbol{\alpha} \in \mathbb{C}^L\}\\ &= \{\boldsymbol{U\Sigma V}^H\boldsymbol{\alpha}, \forall \boldsymbol{\alpha} \in \mathbb{C}^L\}\\ &\overset{(a)}{=} \{\boldsymbol{U'_K\beta}, \forall \boldsymbol{\beta} \in \mathbb{C}^K\}\\ &= \text{colspan}\,(\boldsymbol{U_K}),\end{aligned}$$

where $\boldsymbol{U'_K} = \boldsymbol{U\Sigma}$ is a linear combination of the columns of $\boldsymbol{U_K}$. (a) is due to the fact that since $\boldsymbol{V}^H$ is orthonormal and of rank K, $\forall \boldsymbol{\beta} \in \mathbb{C}^K, \exists \boldsymbol{\alpha} \in \mathbb{C}^L$ such that $\boldsymbol{\beta} = \boldsymbol{V}^H\boldsymbol{\alpha}$. Similarly

$$\begin{aligned}\text{colspan}\,(\boldsymbol{F}) &= \{\boldsymbol{F\alpha}, \forall \boldsymbol{\alpha} \in \mathbb{C}^L\}\\ &= \{\boldsymbol{CDE\alpha}, \forall \boldsymbol{\alpha} \in \mathbb{C}^L\}\\ &\overset{(b)}{=} \{\boldsymbol{C\beta}, \forall \boldsymbol{\beta} \in \mathbb{C}^K\}\\ &= \text{colspan}\,(\boldsymbol{C}).\end{aligned}$$

Let $\boldsymbol{E'} = \boldsymbol{DE}$ be a linear combination of the rows of $\boldsymbol{E}$. (b) comes from the fact that since $\boldsymbol{E'}$ is full rank (K), $\forall \boldsymbol{\beta} \in \mathbb{C}^K, \exists \boldsymbol{\alpha} \in \mathbb{C}^L$ such that $\boldsymbol{\beta} = \boldsymbol{C'\alpha}$. Therefore we can write

$$\boldsymbol{C} = \boldsymbol{U_K\Psi}, \tag{1.18}$$

where $\boldsymbol{\Psi}$ is a $K \times K$ nonsingular matrix.

Moreover, define $\boldsymbol{C^{(1)}}$ as the matrix formed by all but the first row of $\boldsymbol{C}$ and $\boldsymbol{C^{(2)}}$ as the matrix formed by all but the last row of $\boldsymbol{C}$. $\boldsymbol{U_K^{(1)}}$ and $\boldsymbol{U_K^{(2)}}$ are defined similarly relative to $\boldsymbol{U_K}$. Furthermore, observe that

$$\boldsymbol{C^{(1)}} = \boldsymbol{C^{(2)}\Phi}, \tag{1.19}$$

where $\boldsymbol{\Phi}$ is a $K \times K$ diagonal matrix such that $\Phi(i,i) = u_i$. Therefore, from equations (1.18) and (1.19), we have:

$$\begin{aligned} \boldsymbol{U_K^{(1)}} &= \boldsymbol{C^{(1)}}\boldsymbol{\Psi}^{-1} \\ &= \boldsymbol{C^{(2)}}\boldsymbol{\Phi}\boldsymbol{\Psi}^{-1} \\ &= \boldsymbol{U_K^{(2)}}\boldsymbol{\Psi}\boldsymbol{\Phi}\boldsymbol{\Psi}^{-1} \\ &= \boldsymbol{U_K^{(2)}}\boldsymbol{H}, \end{aligned} \tag{1.20}$$

where $\boldsymbol{H} = \boldsymbol{\Psi}\boldsymbol{\Phi}\boldsymbol{\Psi}^{-1}$. The matrices $\boldsymbol{H}$ and $\boldsymbol{\Phi}$ are similar and therefore share the same eigenvalues, namely the roots $\{u_k\}_{k=0}^{K-1}$ of the annihilating filter.

Back to the noisy case, equation (1.20) becomes

$$\boldsymbol{\hat{U}_K^{(1)}} = \boldsymbol{\hat{U}_K^{(2)}}\boldsymbol{H}. \tag{1.21}$$

ESPRIT solves equation (1.21) either using a least squares (LS) or a total least squares (TLS) approach [26]. Based on our experiments, both methods were leading to similar results. The pseudo-code for ESPRIT with LS can be found on Algorithm 2.

Algorithm 2 ESPRIT denoising and roots finding

```
Input: a vector of noisy measurements $\boldsymbol{\hat{x}}$ of size $M$.
Output: the roots $u_k$ of the vector $\boldsymbol{a}$ such that $\boldsymbol{a} * \boldsymbol{\hat{x}} = \boldsymbol{0}$.
  $M = \text{length}(\boldsymbol{\hat{x}})$
  $L = \lfloor \frac{M}{2} \rfloor$
  $\boldsymbol{\hat{x}_r} = [\hat{x}[L], \hat{x}[L-1], \dots, \hat{x}[0]]$
  $\boldsymbol{\hat{x}_c} = [\hat{x}[L], \hat{x}[L+1], \dots, \hat{x}[M]]$
  $\boldsymbol{F} = \text{toeplitz}(\boldsymbol{\hat{x}_c}, \boldsymbol{\hat{x}_r})$
  $\{\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}\} = \text{svd}(\boldsymbol{F})$
  $\boldsymbol{U} = \boldsymbol{U}[:, 1:K]$
  $\boldsymbol{H} = \boldsymbol{U}[2:\text{end}, :]\,\text{pinv}(\boldsymbol{U}[1:\text{end}-1, :])$
  return eigenvalues$(\boldsymbol{H})$
```

The main advantage of ESPRIT over Cadzow is that it is a non-iterative algorithm[3], which makes its execution time significantly faster (depending on how many iterations are needed for Cadzow).

There is a third denoising approach that can be used and that is briefly mentioned

[3] Note that SVD algorithms are iterative, what is meant here is that the number of SVDs is always fixed, unlike in Cadzow.

here. It has been invented by B. De Moor [18] and its investigation in the FRI framework has been suggested by Frank Quick[4]. It finds the vector of denoised coefficients $\boldsymbol{x}$ and the corresponding annihilating filter $\boldsymbol{a} = \{A[m]\}_{k=0}^{K-1}$ from the vector of noisy measurements $\hat{\boldsymbol{x}}$ by solving the following least squares problem:

$$\{\boldsymbol{x}, \boldsymbol{a}\} = \underset{\boldsymbol{x},\boldsymbol{a}}{\operatorname{argmin}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 \text{ under the constraints } \boldsymbol{T}_{\hat{\boldsymbol{x}}}\boldsymbol{a} = \boldsymbol{0} \text{ and } \|\boldsymbol{a}\| = 1,$$

where $\boldsymbol{T}_{\hat{\boldsymbol{x}}}$ is the Toeplitz matrix built from the Fourier coefficients $\hat{\boldsymbol{x}}$ as in equation 1.11. De Moor uses Lagrange multipliers to resolve this constrained least squares problem. Its advantage is that, unlike ESPRIT and Cadzow, it gives the optimal L^2 solution. However, its convergence is not guaranteed and its speed is extremely slow.

[4] Researcher at Qualcomm Inc.

2 ECG Compression Techniques

In this chapter, we describe different ECG compression approaches and also introduce a new model. The focus is on their mathematical and algorithmic description, details regarding their a practical implementation can be found in Chapter 4 and a more precise analysis of the performance of some selected methods is provided in Chapter 5.

2.1 Review of some ECG compression methods

ECG compression has been widely studied in the literature in the past decades and several approaches have been proposed to reduce the bit rate of ECG signals. There are of course a lot of generic data compression (both lossless and lossy) methods which have been developed so far. To name a few, polynomial predictors [4] and interpolators [35], data compression by differential pulse code modulation (DPCM) coding [50], and entropy coding (for example the well known Huffman coding [29]). However, since this thesis is dedicated to ECG signals, the focus is on techniques specifically targeted at ECG signals. We briefly review here the most popular lossy compression methods (lossless compression is in not in the scope of the thesis). A more complete overview of most of the ECG compression techniques (on which this section is based) can be found in [31]. Other details regarding some of the most used algorithms are in [55]. At the end of the chapter, we investigate some uses of FRI for ECG compression and also present a new FRI-based technique.

Compression methods ECG lossy compression methods are usually classified in two main categories:

– **Direct methods:** These methods are directly applied in the time domain. amplitude zone time epoch coding (AZTEC), turning point (TP), fan-out oriented (FAN) and coordinate reduction time encoding system (CORTES) algorithms mentioned below fall into this category.

– **Transform based methods:** Transform based techniques try to find a sparse representation in a different basis and compression is achieved by discarding the zero or small coefficients.

Note that all the compression techniques presented here assume a complete input signal (sampled at the Nyquist rate or more). In Chapter 3, we relax this assumption.

2.1.1 AZTEC

AZTEC [15] has been the method of choice for many applications due to its simplicity, low complexity and relatively good compression ratio. AZTEC converts ECG signals into a sequence of *plateaus* and *slopes* as follows.
Let $x[n]$ for $n = 0, \ldots, N-1$ be the ECG signal we want to compress. Out of the sequence $x[n]$, AZTEC only picks a certain number of samples denoted by $x[n_k]$ ($k = 0, \ldots, K-1$). n_0 is usually chosen as 0 and $n_{K-1} = N-1$. To find the remaining values, let us define the following variables:

$$\begin{aligned} x_{k,min}[n] &= \min\{x[n_{k-1}], \ldots, x[n]\} \\ x_{k,max}[n] &= \max\{x[n_{k-1}], \ldots, x[n]\}. \end{aligned}$$

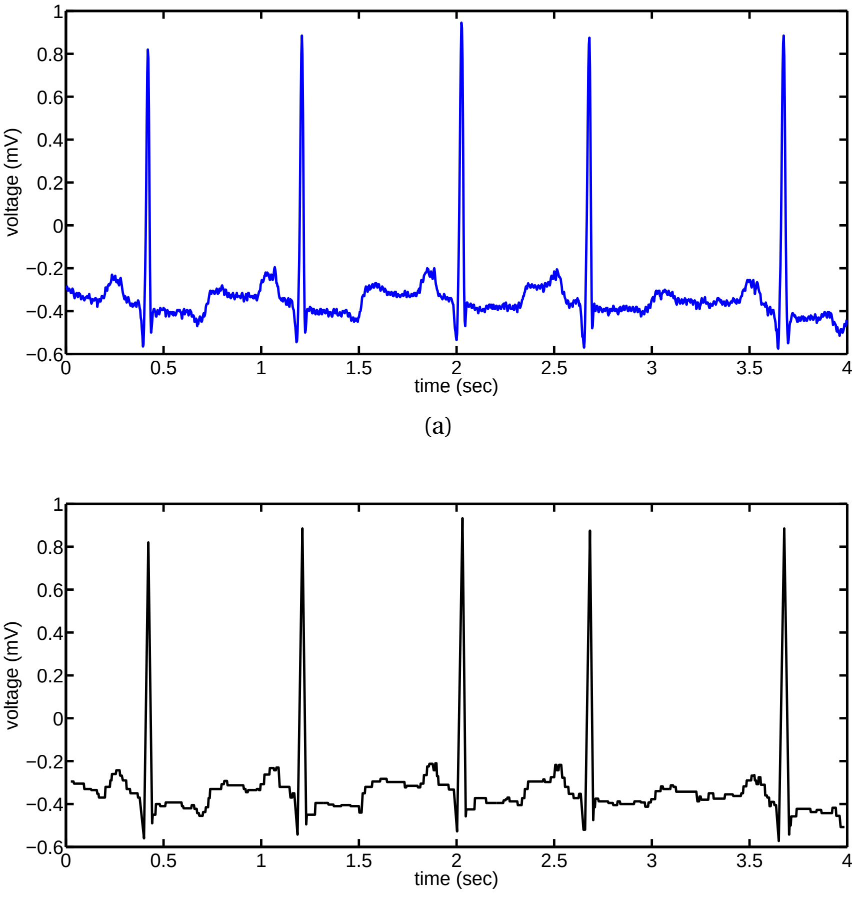
n_k is defined as the first value n such that the inequality below does not hold any more:

$$x_{k,max}[n] - x_{k,min}[n] \le \epsilon,$$

where ϵ is the error tolerance. A plateau is constructed by a straight line going from $x[n_{k-1}]$ to $x[n_k]$ and its amplitude is $\frac{x_{k,max}[n_{k-1}]+x_{k,min}[n_k]}{2}$. When the slope of the signal is too steep (or in other words, when $n_k - n_{k-1}$ falls below some fixed threshold), a slope is defined in place of a plateau and in this case the value $x[n_k]$ is kept instead of the average of the two extreme points of the sequence.

The number of samples is controlled by ϵ: a larger ϵ leads to smaller sampling rate.

Starting with an input signal sampled at 500 Hz (and quantized at 12 bits per sample), a compression ratio of 10 to 1 is claimed [15]. One of the drawbacks of AZTEC is that the compressed signal presents very visible discontinuities between the plateaus, as it can be seen in Figure 2.1, which illustrates AZTEC compression on a signal taken from the MIT-BIH arrhythmia database[1]. Moreover, since the compressed samples are not uniformly spaced any more, one needs to keep track of their time of occurrence (either by storing the differential time between each sample or their absolute time). Finally, note that there is no need to differentiate the plateaus from the slopes as it can be deducted from the distance with the previous sample.



(a)

(b)

Figure 2.1: (a) Fragment of record 100 from the MIT-BIH database sampled at 360 Hz and (b) its AZTEC compression with a ratio of 5 : 1.

Several modifications (for example [22] involving adaptive thresholds) have been proposed to the original AZTEC, mainly to minimize the steep steps or discontinuities

[1] http://www.physionet.org/physiobank/database/mitdb/

present in the P and T waves.
There are some other ways to get rid of the discontinuities, for example processing the output of the AZTEC with a smoothing filter as in [16](the price to pay is that the QRS complex is also smoothed in the process).

Another interesting technique similar to the AZTEC method is the scan-along polygonal approximation (SAPA) [30] (see also the FAN [21] algorithm, proven to be equivalent to SAPA [7]). Unlike AZTEC, the output of SAPA does not exhibit any discontinuity (even though its slope is not continuous). In fact, in SAPA all portions of the input signal are modelled with slopes. For more details regarding this algorithm, please refer to [55, 31].

2.1.2 Turning Point method

The TP method [47] has a fixed compression ratio of 2 to 1. Out of 3 consecutive points, it always discards the middle one, except when it is representing a *turning point*, in which case it discards the third one.
As for AZTEC, the TP algorithm does not have uniformity of its samples. Another disadvantage is its fixed and relatively low compression ration.
Some variations of this method also exist, for instance the CORTES [1], which is a hybrid of the TP and the AZTEC method.

2.1.3 Transform based methods

In the case of transform based methods, the compression is performed in a different domain that provides a sparser representation of the input signal. The main challenge resides in finding such a sparse representation in an efficient manner. Having a sparse representation means that most of the coefficients are zero or close to zero. That allows us to discard them without harming too much the original signal. A simple example that illustrates sparsity is the continuous signal $x(t) = cos(at)$, that would require an infinite number of samples for a representation in time. However, taking the Fourier transform (FT) of $x(t)$ leads to $X(\omega) = \sqrt{2\pi} \cdot \frac{\delta(\omega-a)+\delta(\omega+a)}{2}$, a spectrum that is zero everywhere except for two single values.

Some of the most popular and efficient methods include the FT, cosine transform (CT) or Karhunen-Loeve transform (KLT) [33, 40]. The latter is the most optimal one but is also the most computationally expensive one since new basis vectors needs to be computed for every different input signal. Moreover, the computation of the KLT basis vectors is very costly as it involves the eigendecomposition of large matrices. Therefore, other suboptimal transforms such as FT and CT are often preferred to it in practice.

Unlike most direct methods, transform based methods process signals by blocks. For ECG signals, the most natural way is to defined a block as a single heart beat, therefore QRS detection must precede every transform based method (cf. appendix A.1 for more details about QRS detection).

We recommend the reader looking for a more complete summary of the aforementioned techniques as well as other state-of-the art methods to refer to [31].

2.1.4 QRS spline modelling

To the best of our knowledge, this method is the first application of the FRI theory to ECG compression. The method is due to P. Marziliano et al. [27], based on their previous work on shot noise [44]. The main idea behind it is to decompose each heart beat into two different components as in Figure 2.2:

- A low frequency part, which has no frequency higher than 10 Hz.
- A high frequency part, which is almost only present in the QRS complex.

Sampling ECG signals at the Nyquist frequency would require a fair amount of coefficients, since the bandwidth of ECG signals usually ranges up to 80-100 Hz or in the extreme cases even 150 Hz.
The essence of this method is to parametrize the high frequency part using FRI and to sample the residual low frequency part at a lower frequency as follows.

Assuming that only the QRS complex is present in frequencies higher than $\frac{B}{\tau}$ Hz ($\frac{B}{\tau}$ is usually around 10 Hertz in practice), we can rewrite the Fourier series coefficients of the ECG signal as

$$X[m] = \begin{cases} X_L[m] + X_H[m] & \text{if } m \in [-B, B] \\ X_H[m] & \text{if } m \notin [-B, B] \end{cases},$$

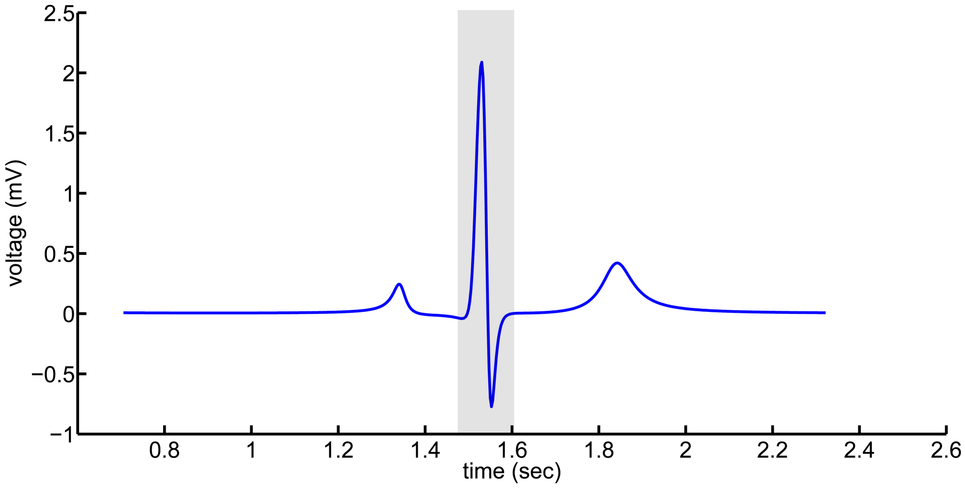


Figure 2.2: One period of an ECG signal separated into the lower frequency part (white are) and the higher frequency part (gray area)

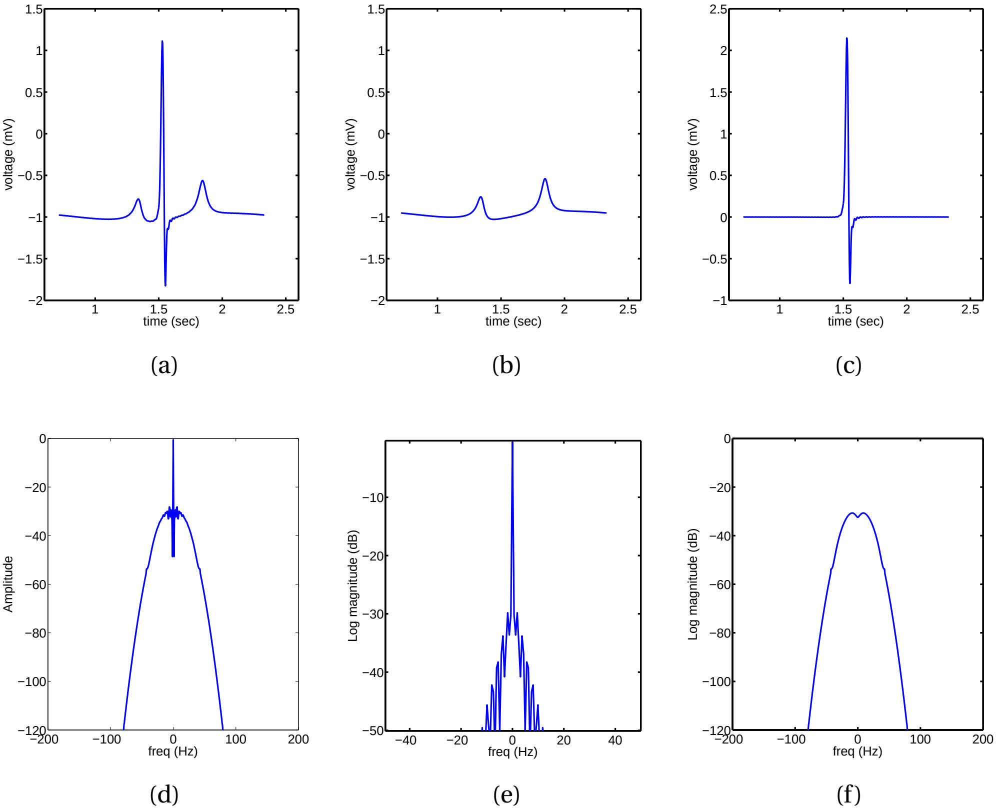
where X_L is the low frequency part, X_H is the spectrum that contains only the QRS complex and τ is the period of the signal. Figure 2.3 illustrates the decomposition of the ECG signal in both the time and the frequency domain.

In this model, the QRS complex $X_H[m]$ is approximated as a non-uniform spline of order 1. Recall that a spline is a piecewise polynomial function, and that its order corresponds to the highest order of its polynomials. As we have seen in Section 1.3.2, the second derivative of $X_H[m]$, expressed as $(\frac{i2\pi m}{\tau})^2 X_H[m]$ in the frequency domain, can be modelled with a stream of Diracs. The number of parameters for each Dirac is 2 (time location and amplitude) and, since we model the QRS complex using 4 'straight lines', the corresponding number of Diracs is $K = 5$. Hence the local rate of innovation in each heart beat is given by

$$\rho_\tau(t) = \frac{2K}{\tau}.$$

By picking Fourier coefficients from $m = B + 1$ to $B + 2K$ (containing only frequencies representing the QRS complex), we are able to efficiently parametrize the QRS complex.

Once the locations $\{t_k\}$ and amplitudes $\{c_k\}$ of the stream of Diracs have been recovered, we can reconstruct an estimated (parametrized) version of the QRS complex



(a) (b) (c)

(d) (e) (f)

Figure 2.3: (a) Synthesized ECG signal decomposed into the sum of (b) a low frequency signal containing the P and T waves and (c) a higher frequency signal containing the QRS complex. Their respective spectrum is displayed on (d), (e) & (f). Some high frequencies are still present out of the QRS complex, but they drop very fast.

by integrating twice the parametrized stream of Diracs:

$$\hat{X}_H[m] = \left(\frac{\tau}{i2\pi m}\right)^2 \frac{1}{\tau} \sum_{k=0}^{K-1} c_k e^{-i2\pi t_k m/\tau}.$$

The residual low frequency part is then calculated as

$$\hat{X}_L[m] = X[m] - \hat{X}_H[m].$$

As it now contains almost no high frequency any more, $\hat{X}_L[m]$ is close to zero for frequencies larger than around 10-15 Hz. Hence only a small number of Fourier coefficients need to be kept. For each heart beat, the compressed signal consists of $\hat{X}_L[m]$ and the 10 parameters (5 amplitudes and 5 time locations) used to model the QRS as a spline.

The reconstruction of the signal is quite straightforward. Given the Diracs parameters, we can calculate the Fourier coefficients of the Diracs up to the sampling frequency of our choice and integrate it twice to obtain the coefficients of the spline. Finally, we just need to replace the low frequencies of the spline with the residual signal and take the inverse Fourier transform to obtain a time domain representation of the signal.

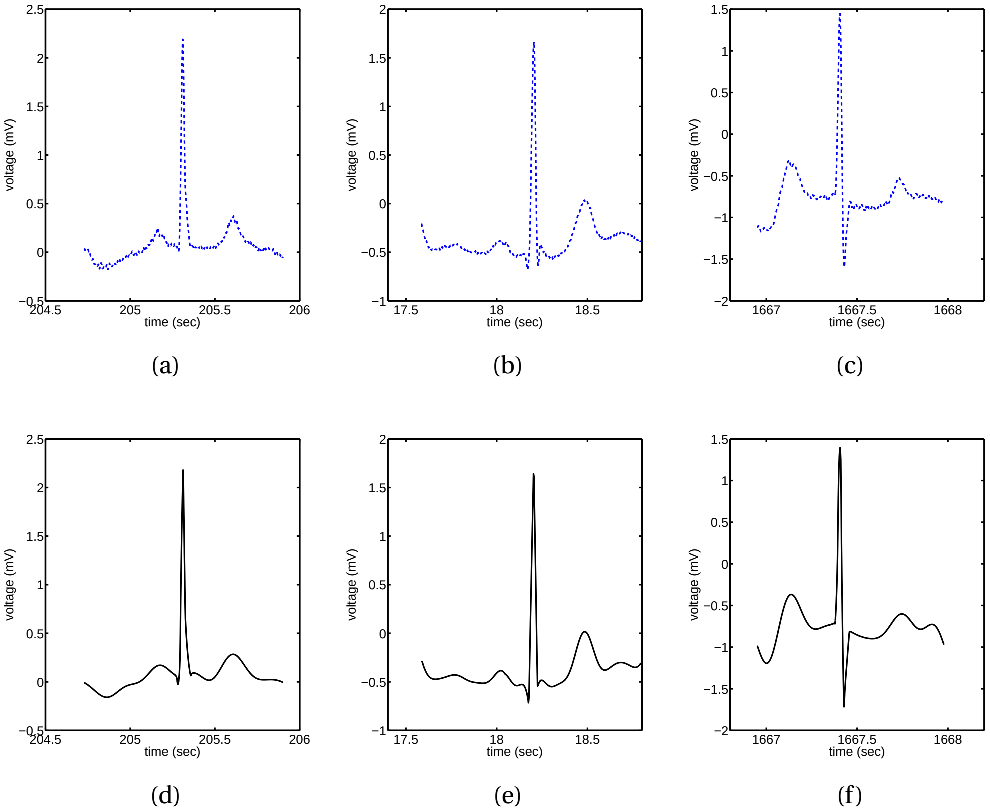


Figure 2.4: original signals from the MIT BIH database: (a), record 101 (b) record 103 & (c) record 123 and their respective reconstructions (d), (e) & (f) with parameters $K = 5$ and $L = 7$ (total of 24 samples per heart beat. Assuming that the period of one beat is about 1 second, this is equivalent to a compression ratio of 15 to 1).

Our own implementation of this method gave us satisfactory results and very high compression ratios on *some selected signals*, as illustrated in Figure 2.4: the QRS complex is particularly sharp and accurate in these examples. However, the results were unstable and we did not manage to consistently achieve a good reconstruction. Unfortunately, we could not access to the code from the authors to compare our implementation to theirs.

To improve the robustness of the algorithm (which was crucial for a practical utilization of it), we add an extra step in our implementation: the extraction of the QRS complex by windowing it in time before being fed to the algorithm. This simple modification forces the Diracs locations to always be within the window, which was not always enforced with the previous implementation.

Based on our experiments and simulations, a Tukey window centered on the R peak[2] and with a width of 0.1 second (approximately the average width of a QRS complex) was leading to reasonably good and more consistent results.

Even though it makes the algorithm more polyvalent and improves the quality of the QRS modelling, the use of a window in the time domain also comes with some drawbacks, some of which are discussed in Section 3.2.1 and in Chapter 5.

2.2 Variable Pulse Width FRI

This section introduces a new ECG compression scheme: the VPW model. This new method was developed at Qualcomm Inc. during the Summer 2011 and its discovery is due to Frank Quick and Ronald Crochiere [51]. It can be seen as an extension of FRI for streams of Diracs (cf. Section 1.3.1 for the details) in the sense that it adds two other free parameters (damping factor and asymmetry) to the Dirac pulse model.

2.2.1 A new model

A brief look at any ECG signal shows that a heart beat can be interpreted as a sum of 5 pulses (one pulse for each of the P, Q, R, S and T waves). To stay consistent with the FRI notation introduced in Section 1.3, we denote the number of pulses by K.

The idea of VPW is to parametrize each of these pulses with a determined shape. In the case of ECGs, a pulse following a Cauchy-Lorentz distribution turns out to be a very suitable shape to start with (the choice of the pulse shape becomes clear in the next paragraphs, when the recovery of the parameters is described). With this model, we can characterize every pulse $p_k(t)$ by

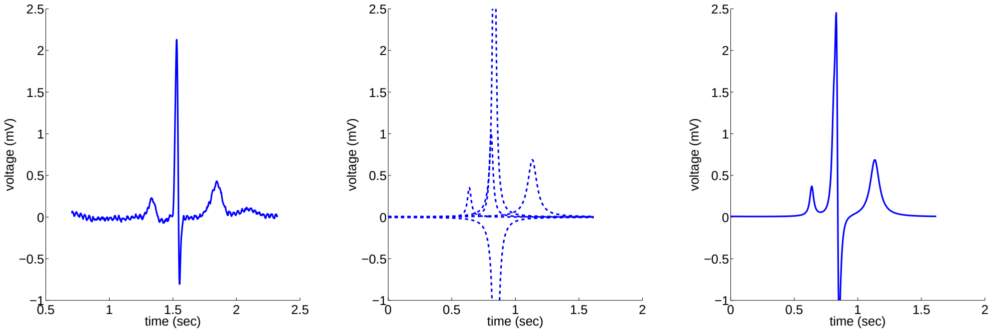
$$p_k(t) = \frac{c_k}{\pi} \frac{a_k}{a_k^2 + (t - t_k)^2}, \tag{2.1}$$

[2] The location of the R peak can be found using some QRS detection methods (cf. appendix A.1).

where k is the index of the pulse, c_k its amplitude, a_k its width and t_k its location in time. An example of a Cauchy-Lorentz pulse can be seen in Figure 2.6(a). Every heart beat $y^{(1)}(t)$ is therefore modelled as the sum of K such pulses:

$$\begin{aligned} y^{(1)}(t) &= \sum_{k=0}^{K-1} p_k(t) \\ &= \sum_{k=0}^{K-1} \frac{c_k}{\pi} \frac{a_k}{a_k^2 + (t - t_k)^2}. \end{aligned}$$

Figure 2.5 shows an example of an ECG heart beat with a possible decomposition in 5 different Cauchy-Lorentz pulses. The sum of the different pulses is represented in Figure 2.5(c). Despite the fact that most of the pulses in Figure 2.5(c) are slightly higher than the original ones, we can see that this model already offers a solid approximation of the input signal. The good news is that we can do even better!



(a) (b) (c)

Figure 2.5: One period of a symmetric VPW pulse from the MIT-BIH arrhythmia database (sample 123) and its decomposition into 5 Cauchy-Lorentz pulses: (a) original signal, (b) its decomposition into 5 symmetric pulses (c) and the sum of the 5 pulses.

The main disadvantage of the model proposed so far is that all the pulses are perfectly symmetric, which is obviously not the case on real ECG data. To add some asymmetry to our model[3], let us introduce a second pulse shape $q_k(t)$, which is simply the Hilbert transform of $p_k(t)$ (again, the choice of this pulse shape becomes clear

[3] The idea of introducing an asymmetric part is due to John Hong, from Qualcomm Inc.

later):

$$q_k(t) = \frac{d_k}{\pi} \frac{t - t_k}{a_k^2 + (t - t_k)^2}. \tag{2.2}$$

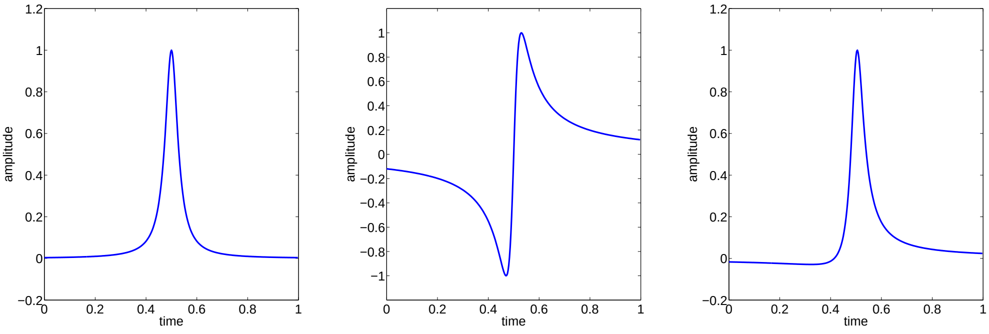
Observe that all the parameters are the same as in $p_k(t)$, except for the amplitude, denoted now as d_k. Such an asymmetric pulse is illustrated in Figure 2.6(b).
We can also define $y^{(2)}(t)$ as the sum of K asymmetric pulses:

$$\begin{aligned} y^{(2)}(t) &= \sum_{k=0}^{K-1} q_k(t) \\ &= \sum_{k=0}^{K-1} \frac{d_k}{\pi} \frac{t - t_k}{a_k^2 + (t - t_k)^2}. \end{aligned}$$

The pulse shape model $r_k(t)$ used in VPW consists of the sum of the symmetric and asymmetric pulses defined in equations (2.1) and (2.2):

$$\begin{aligned} r_k(t) &= p_k(t) + q_k(t) \\ &= \frac{1}{\pi} \frac{c_k a_k + d_k(t - t_k)}{a_k^2 + (t - t_k)^2}. \end{aligned}$$

An example of a VPW pulse is pictured in Figure 2.6c. The complete VPW model $y(t)$



(a) (b) (c)

Figure 2.6: Example of a VPW pulse: (a) symmetric pulse (Cauchy-Lorentz distribution), (b) asymmetric pulse and (c) VPW pulse: sum of the Cauchy-Lorentz and the asymmetric pulse.

for every heart beat is hence given by

$$\begin{aligned} y(t) &= \sum_{k=0}^{K-1} r_k(t) \\ &= y^{(1)}(t) + y^{(2)}(t) \\ &= \sum_{k=0}^{K-1} \frac{1}{\pi} \frac{c_k a_k + d_k(t-t_k)}{a_k^2 + (t-t_k)^2}. \end{aligned} \tag{2.3}$$

This model offers a broader range of shapes than the simpler symmetric model. As depicted in Figure 2.7, the sum of 5 VPW pulses in Figure 2.7(c) matches very closely the input waveform in Figure 2.7(a).

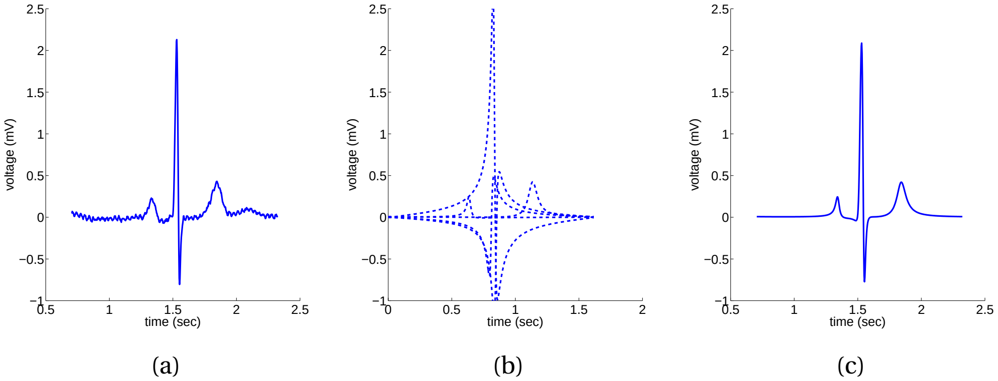


Figure 2.7: One period of a pulse from the MIT-BIH arrhythmia database (sample 123): (a) original signal, (b) its decomposition into 5 symmetric pulses and the sum of the 5 pulses.

At this point we have an interesting model, but it remains to show how to extract the parameters from the waveform $y(t)$, and that is where FRI theory comes to the rescue!

Like in Section 1.3.1, FRI analysis is performed in the discrete frequency domain. Since the Fourier domain is discrete, we need to consider a periodized version of the signal $y(t)$, denoted by $x(t)$

$$\begin{aligned} x(t) &= \sum_{n\in\mathbb{Z}} x(t-n\tau) \\ &= \sum_{n\in\mathbb{Z}} \sum_{k=0}^{K-1} \frac{1}{\pi} \frac{c_k a_k + d_k(t-t_k-n\tau)}{a_k^2 + (t-t_k-n\tau)^2}, \end{aligned} \tag{2.4}$$

where τ is the period of the signal. Contrary to the classical settings of FRI with a stream of Diracs, the tails of neighboring pulses directly contribute to the value of each period of the signal. Thus characterizing such a signal in the time domain with very wide pulses would be equivalent to computing an infinite sum. Fortunately, as we see in Section 2.2.3, there exists a closed form for $x(t)$, that eliminates the infinite sum.

Computing the Fourier series coefficients of $x(t)$ leads to the following result:

$$\begin{aligned} X[m] &= X^{(1)}[m] + X^{(2)}[m] \\ &= \frac{1}{\tau}\sum_{k=0}^{K-1}\left(c_k - i d_k \operatorname{sign}(m)\right) e^{-2\pi/\tau(a_k|m| + i t_k m)}, \end{aligned} \tag{2.5}$$

for $m \in \mathbb{Z}$. Since the period of the signal is τ, the Fourier samples are taken every $1/\tau$ Hz. $X^{(1)}[m] = \frac{1}{\tau}\sum_{k=0}^{K-1} c_k e^{-2\pi/\tau(a_k|m|+it_k m)}$ denotes the Fourier series of the Cauchy-Lorentz pulses and $X^{(2)}[m] = -\frac{1}{\tau}\sum_{k=0}^{K-1} i d_k \operatorname{sign}(m) e^{-2\pi/\tau(a_k|m|+it_k m)}$ denotes the contribution from the asymmetric pulses. Observe that for strictly positive indices, $X^{(1)}[m]$ and $X^{(2)}[m]$ only differ in their respective amplitudes, one being purely real while the other one being purely complex (and negative).

The reader familiar with FRI theory will notice similarities between equation (2.5) and the Fourier series coefficients of the stream of Diracs described on equation (1.5). In fact, for positive frequencies, we can get from equation (2.5) to equation (1.5) simply by setting $a_k = d_k = 0$. The same can be done in the time domain, by setting $d_k = 0$ and taking the limit when a_k goes to zero in equation (2.3):

$$\begin{aligned} \lim_{a_k \to 0} y(t)|_{d_k=0} &= \lim_{a_k \to 0} \sum_{k=0}^{K-1} \frac{c_k}{\pi} \frac{a_k}{a_k^2 + (t - t_k)^2} \\ &= \sum_{k=0}^{K-1} c_k \delta(t - t_k). \end{aligned}$$

Therefore, the VPW model is truly a generalization of the FRI model for the stream of Diracs! Its similarities with the classical FRI algorithm enable us to reuse and adapt the techniques developed for the parameters recovery.

2.2.2 Parameters recovery

The parameters recovery is very similar to the recovery for the periodic stream of Dirac pulses presented in Section 1.3.1, therefore we emphasize here on the differences.

First, note that because of the absolute value $|m|$ in the Fourier series coefficients, we cannot choose a set of coefficients both in the positive and the negative part of the Fourier spectrum. For example the set chosen in equation 1.11 would not work. The most straightforward set to chose is the coefficients $X[m]$ for $m = 0,\dots,M-1$ $(M > 2K)$, like illustrated in Figure 2.8 [51]. The size of this set is the theoretical minimum required to recover the parameters, but of course a bigger set leads to more stable and accurate results, especially when working with real ECG signals and in the presence of noise.

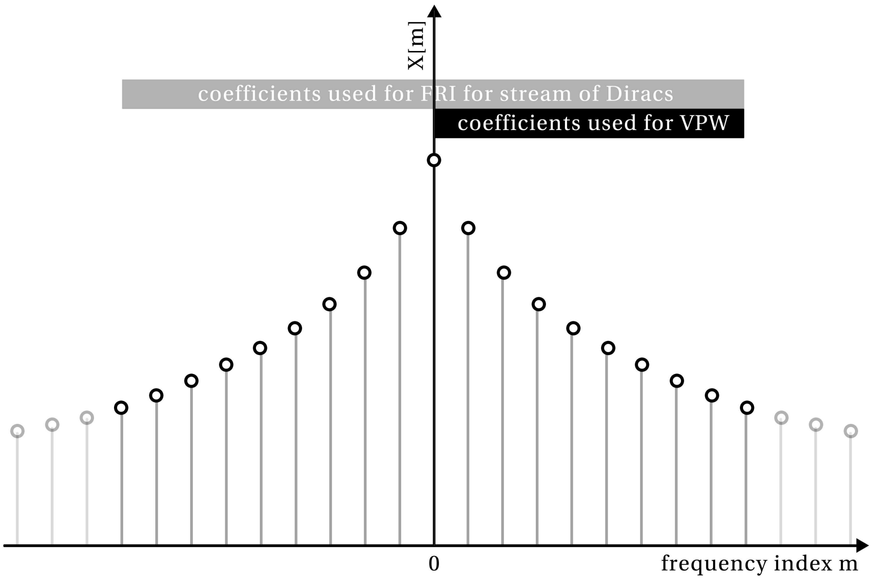


Figure 2.8: Simulation of the Fourier spectrum of a VPW pulse with the (usual) selection of the coefficients in the case of a stream of Dirac (gray) and in the case of VPW pulses (black).

From Proposition 1, the roots $\{u_k\}_{k=0}^{K-1}$ of the annihilating filter become $u_k = e^{-2\pi/\tau(a_k+it_k)}$. Observe that now the roots are no more located on the unit circle but can rather be anywhere in the complex plane. Of course if stability of the system is required, they must be located inside or on the unit circle (in other words, a_k must be positive).

The locations of the pulses t_k is recovered exactly as in equation (1.9), that is $t_k = -\frac{\tau\angle u_k}{2\pi}$ and the width a_k is computed as $a_k = -\frac{\tau log|u_k|}{2\pi}$, where $|u_k|$ is the magnitude of u_k. The amplitudes $\{ck\}$ and $\{dk\}$ are found the same way as for the stream of Diracs, by solving equation 1.14. But this time the solution gives complex values $\{e_k\}_{k=0}^{K-1}$. The amplitude of the symmetric part is then given by the real part of it and the amplitude of the asymmetric is given by the imaginary part. That is to say,

$c_k = \Re\{e_k\}$ and $d_k = \Im\{e_k\}$.

2.2.3 Waveform reconstruction

Once the set of parameters has been retrieved, it remains to reconstruct the original signal. This can be done in both the *frequency domain* and the *time domain*.

The frequency domain reconstruction is straightforward, as we simply inject the parameters we found in equation (2.5) and follow it with an inverse Fourier transform to get a time domain representation of the signal. In the original FRI, this method does not work as well as in VPW: since the frequency spectrum is not damped, a Fourier spectrum limited in length will automatically generate some Gibbs phenomenon in the time domain.

For the reconstruction in the time domain, unlike with the stream of Diracs where the introduction of the periodicity is not an issue, the tails get folded back on the signal up to infinity in VPW. There are different ways to cope with this problem:

- Inject the recovered parameters in equation (2.4) and restricting the sum to a few neighboring terms. This is a simple approach, although it is not very elegant and far from exact when the pulses are very wide.
- Find a closed form for equation (2.4). This can be done by taking the inverse Fourier transform of equation (2.5) (the details of the following computation are provided in appendix A.2):

$$\begin{aligned} x(t) &= \sum_{k=0}^{K-1} \sum_{m=-\infty}^{\infty} X[m] e^{i2\pi m t/\tau} \\ &= \frac{1}{\tau} \sum_{k=0}^{K-1} \frac{c_k \left(1 - |z_k(t)|^2\right) + 2 d_k \Im\{z_t\}}{(1 - z_k(t))\,(1 - z_k(t)^*)}, \end{aligned} \tag{2.6}$$

  where $z_k(t) = e^{2\pi/\tau(-a_k + i(t-t_k))}$ and $\Im\{z_t\}$ is the imaginary part of z_t. We can verify that when τ goes to infinity, we indeed get back to equation (2.3) (cf. appendix A.2 for more details):

$$\begin{aligned} \lim_{\tau\to\infty} x(t) &= \sum_{k=0}^{K-1} \frac{1}{\pi} \frac{c_k a_k + d_k (t - t_k)}{a_k^2 + (t - t_k)^2} \\ &= y(t). \end{aligned}$$

The importance of accounting for the periodicity of the signal is illustrated in Section 4.3.3.

Reconstruction in the time domain is obviously faster than reconstruction in the frequency domain as it avoids an inverse Fourier transform, It is also more flexible in the sense that is provides infinite resolution of the reconstructed signal, since we can compute it for any t. On the other hand, frequency domain reconstruction allows us the introduction of hybrid methods (cf. Chapter 3) and permits adjustments in the frequency domain (cf. Chapters 3 and 4).

2.2.4 Applications

There are several applications that could make use of VPW-FRI. The first that comes to mind is compression: with only 20 parameters (for $K = 5$) per heart beat, VPW could achieve very high compression rates provided that VPW signals are a good approximation of ECG signals. Moreover, since FRI-based algorithm only need a relatively small number of coefficients, we can think of using VPW as a compressed sensing method that could extrapolate the spectrum of highly downsampled signals (cf. Section 3.2.2 for more details). Finally, due to the very smooth nature of VPW pulses, one could think of VPW-FRI as a *noise removal* technique. A typical example is the 60-Hz interference noise that is present in most of real ECG signals: since VPW preserves only the global shape of the pulses, it could potentially completely get rid of such artefacts (more details can be found in Section 5.3.2). An in-depth analysis of the aforementioned applications is provided in Chapter 5. Other applications such as the use of VPW for automated ECG diagnosis or as an ECG signal synthesizer are suggested as future work in Chapter 6.

There are still a few questions that remain to be addressed for a reliable and practical implementation of VPW-FRI. Among others, we need to ensure the stability of the annihilating filter (it can happen that some of its roots are outside of the unit circle, which causes the exponentials to amplify instead of decaying at higher frequencies). The non-periodicity in time of the input signals also needs to be studied and corrected. A more in-depth analysis concerning these issues and other interrogations is exposed in Section 4.3.

3 Power Efficient Sampling

Like compression, the goal of power efficient sensing is to retain the maximum information from the input signal with the smallest number of bits. The difference is that compression is directly achieved at the sampling stage and the compression scheme never has access to the complete data.

3.1 Sampling is power hungry!

It is quite intuitive that a higher sampling rate requires more power when converting a continuous-time signal to digital: among other examples, a faster and more precise clock is needed. Furthermore, if samples are later transmitted (which is the case in the framework presented in Section 1.1.1), a low-sampling rate is also beneficial in reducing the transmission power required. Typically in the settings described in 1.1.1, the patch is very low power, and almost no or very little compression can be performed on it before the transmission of the samples.

On the other hand, the sampling rate needs to be sufficiently high so that the signal can be well approximated even in the worst conditions: for instance a noisy signal with a very high heart beat rate.

We discuss in this chapter the trade-offs and the limitations when working at sub-Nyquist sampling rates. We study the ability of the FRI-based algorithms introduced in Chapter 2 to extrapolate the spectrum that has been lost in the low-pass filtering prior to sampling at a low rate. We also review some methods based on non-uniform sampling, that could be used to lower down the average sampling rate of ECG signals while preserving the quality and sharpness of the QRS peaks.

3.2 FRI-based techniques

In Chapter 2, no assumption is made on the sampling rate of the input signals of the FRI-based techniques. In theory and in the noiseless setting, we should be able to fully recover the signal (if it precisely follows the underlying model) with a sampling rate bigger or equal to the rate of innovation. In practice however (in the presence of noise and model mismatch), our experiments show that sampling at the rate of innovation considerably degrades the recovered signal. In this section, we investigate the limits of the FRI-based algorithms when the input signals are sampled at rates below the Nyquist frequency (estimated to be between 50 and 120 Hz for the ECG signals used in this thesis). By discarding the high frequency part of the signal in the sampling process, we assess and study the extrapolating capabilities of the algorithms. Last, we present a way to ‘beat’ sinc interpolation with a modified version of VPW-FRI on signals sampled at sub-Nyquist rates.

3.2.1 QRS spline modelling

At first glance, the QRS spline modelling method seems an interesting candidate for recovery of low sampling rate signals: with a rate of innovation of only 10, the extraction of the parameters does not require too many Fourier samples so in theory, we could sample the signal at very low rates. However, as stated in Section 2.1.4, the algorithm needs to be assisted in order to correctly recover the QRS. The fix proposed in Section 2.1.4 is to window the signal in time before passing it to FRI. Unfortunately, this operation is equivalent to a convolution in the frequency domain and the higher frequency coefficients become ‘mixed’ with the lower frequency coefficients. When sampling the signal at a low rate, the information contained in the high frequency components is lost. In other words, using the low coefficients of a low-passed signal windowed in time is not equivalent to using the low coefficients of a high frequency rate signal windowed in time. Our experiments confirmed this issue and we did not manage to get relevant results at reasonably low sampling rates.

3.2.2 VPW-FRI

With VPW, the global shape of the spectrum is no longer flat. It is due to the fact that the roots of the annihilating filter are not located on the unit circle of the complex plane but rather inside it, which makes the frequency spectrum roll off.

One could raise the question if VPW could be used the same way as FRI on low-pass filtered signals. In other words, can we acquire ECG signals at a very low sampling rate and use VPW to reconstruct the higher frequency part of the spectrum that has been destroyed in the process?
Assuming that it is the case, we could use VPW as an intermediate step between a low sampling rate sensing and the final compressed signal: using the framework introduced in Section 1.1.1, ECG signals would be sampled on the patch at a very low rate (around 50 to 60 Hertz), and the samples would be transmitted to the mobile device, which, having more computational power (VPW-FRI in its current state of development still consumes a relatively large amount of computations) could reconstruct the signal by using VPW and at the same time achieve a higher compression ratio by extracting the parameters of the pulses from the signal.

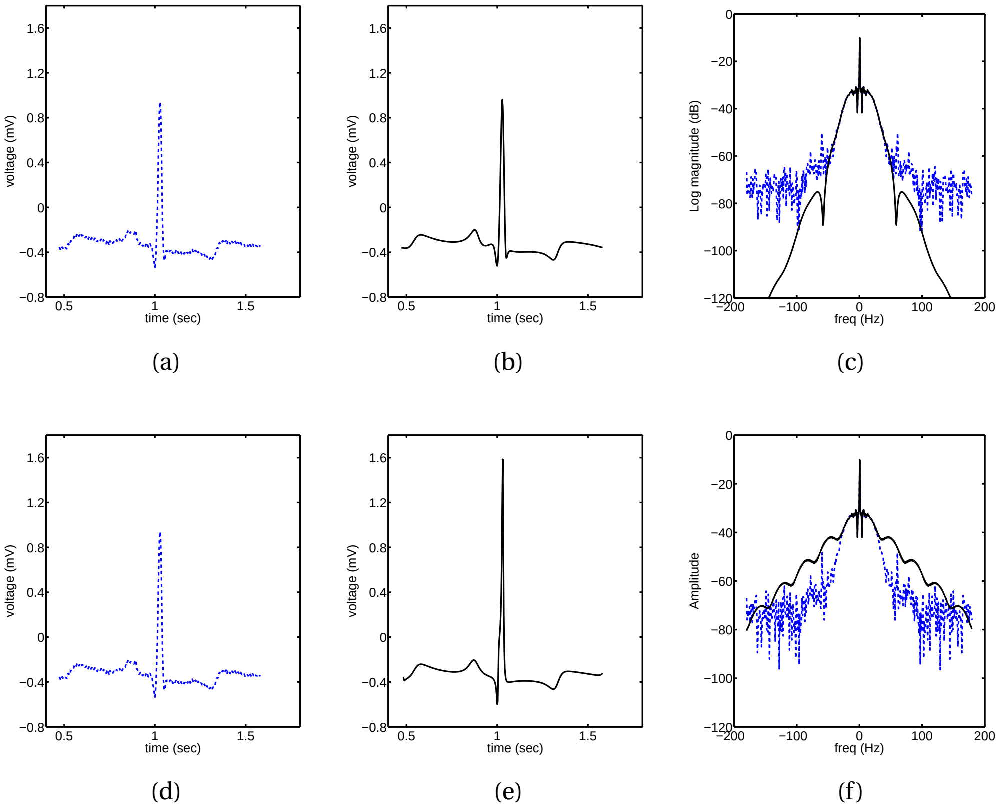
Unfortunately our experiments (cf. Section 5.3.1 for more details) show that, while VPW performs reasonably well in the frequency band containing the coefficients it uses for the analysis, the accuracy of the spectrum extrapolation out of the band is fairly limited. The poor performance of VPW is most probably due to the presence of noise in the input signal and to the model mismatch between the input and VPW-type pulses. As depicted in Figure 3.1, the behavior of the spectrum out of the frequency band cannot be controlled. Indeed, Figure 3.1(f) (representing an ECG signal downsampled at 50 Hz and processed with VPW) exhibits some lobes and the VPW approximation does not follow the original spectrum except between 0 and 25 Hz. In the time domain, that translates mostly into strong overshooting of the R peak and a complete absence of the small S peak.

Enhancing VPW[1]

As mentioned in the previous section, VPW is not a good frequency extrapolating method for ECG signals. We propose here a fix to mitigate this issue and improve the reconstruction of signals sampled at low rates.
First let us have a closer look at the spectrum of a beat of a typical ECG signal represented in Figure 3.2. Its shape in the frequency domain is almost always the same and consists of three main parts:

– A rather flat and ‘oscillating’ part around the DC term.

[1] Joint work with Ali Hormati, post-doctoral researcher at EPFL.



(a) (b) (c)

(d) (e) (f)

Figure 3.1: (a) & (d) Original ECG signal, signal reconstructed with FRI in (b) time & (c) frequency using frequency coefficients from 0 to 180 Hz and (e) time & (f) frequency reconstruction using only the coefficients from 0 to 25 Hz. The dashed line is the original signal.

- A transition part, rolling off and ranging from around 10-15 Hz to 60-70 Hz.
- The noise level (located at−65 dB and below in Figure 3.2(b))

Note that the length and parameters of each part might change from signal to signal. Based on many experiments, we noticed that it is crucial that VPW approximates well the shapes of the first two parts if we want to have a faithful representation of the signal. Unfortunately, when the input signal is downsampled to very low rates, only the beginning of the transition part is used for analysis, and there is no guarantee that the end of it will be well approximated. Figure 3.1(f) illustrates the issues we might encounter using plain VPW-FRI on a 60 Hz signal. The spectrum mismatch is obvious for frequencies higher than 30 Hz (which corresponds to the sampling frequency of 60 Hz). In the time domain, that translates into a strong overshooting for the R peak and a complete miss of the S peak.

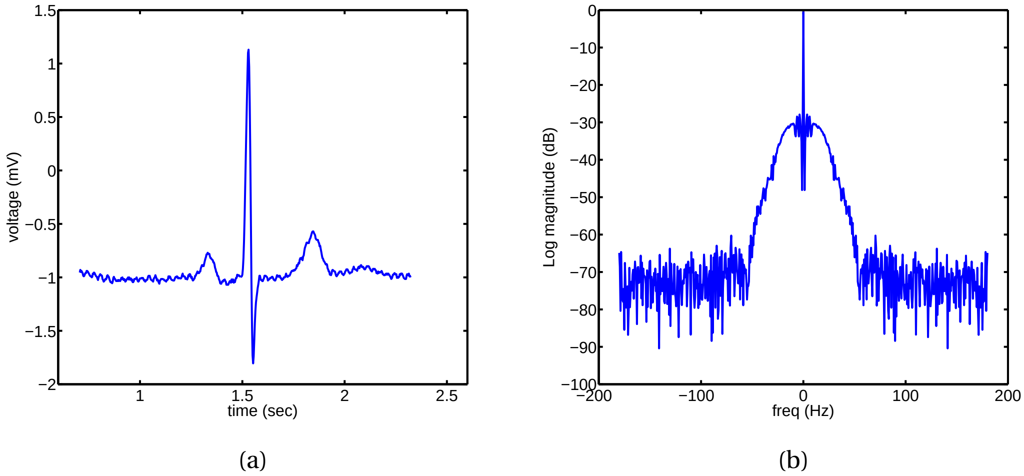


Figure 3.2: One period of a VPW pulse from the MIT-BIH arrhythmia database (sample 123) (a) in the time domain (b) and in the frequency domain.

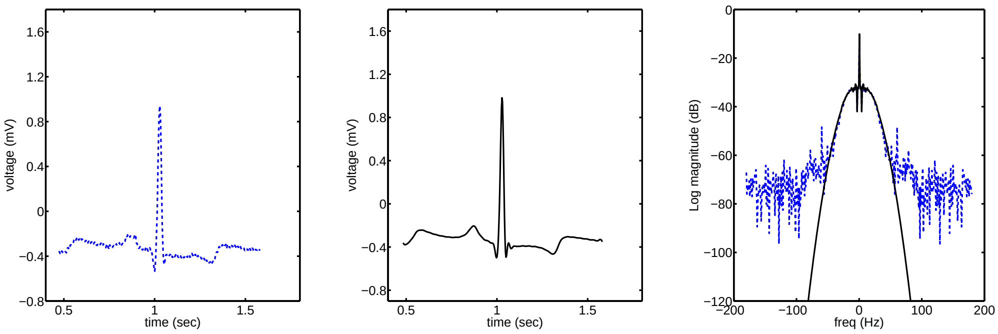
In a nutshell, the idea behind *enhanced VPW* is to adjust the magnitude of the frequency coefficients that VPW tries to extrapolate. We observed that the transition part could be reasonably well approximated with a second order polynomial on a logarithmic scale. The coefficients of the polynomial are found by minimizing the following least square problem:

$$\{\alpha,\beta\} = \underset{\{\alpha,\beta\}}{\operatorname{argmin}} \left\| \log X[m] - \alpha m^2 - \beta \right\|_2,$$

where $X[m]$ represents the Fourier coefficients of the downsampled signal selected between the beginning of the transition part F_t (around 15 Hz) and F_s Hz, F_s being the sampling frequency of the input signal. Hence for the minimization, the indices m range between $\lfloor F_t \cdot \tau \rfloor$ and $\lfloor F_s \cdot \tau \rfloor$, where τ is the period of the input signal. Notice the absence of a first order coefficient, since the shape of the transition part is always symmetric around zero. The second order polynomial is then used to adjust and update the magnitude of the coefficients $\hat{X}[m]$ output by VPW as follows:

$$\hat{X}_{updated}[m] = \frac{\hat{X}[m]}{\left|\hat{X}[m]\right|} e^{\alpha m^2 + \beta},$$

for $m = \lfloor F_s \cdot \tau \rfloor + 1, \lfloor F_s \cdot \tau \rfloor + 2 \ldots, \lfloor F_r \cdot \tau \rfloor$ where F_r is the frequency at which we want to reconstruct the signal. The time domain signal is then simply found by taking an inverse Fourier transform on the coefficients $\hat{X}[m]$. An example of the effect of the enhanced method is illustrated in Figure 3.3. The input signal is the same as in Figure 3.1 and is also downsampled to 60 Hz. Nonetheless, we notice a serious improvement in the reconstructed signal, both in the time and frequency domain (the mismatch in the frequencies higher than 100 Hz has almost no influence on the result) when it is enhanced: the most noticeable effect is that it solves the R peak overshooting problem and models accurately the S peak.



(a) (b) (c)

Figure 3.3: Enhanced VPW. One period of a VPW pulse from the MIT-BIH arrhythmia database (sample 123) (a) original signal at 360 Hz, (b) reconstruction in the time domain (b) and in the frequency domain. The input signal is downsampled to 60 Hz and enhancing is applied to the output of VPW. The dashed line represents the original signal.

A more extensive analysis of the performance of the enhanced technique is proposed in Chapter 5.

3.3 Non-uniform sampling

All the FRI-based compression methods introduced so far are based on uniform sampling. However, we could reduce the number of samples acquired and transmitted by sampling the signal in a non-uniform fashion, similar to what AZTEC does. Indeed, ECG signals can present a lot of variations and some portions like the QRS complex might require a relatively high sampling rate, while the transition parts containing the P and T waves can tolerate much lower sampling frequencies and still be recon-

structed accurately. Moreover, an increase in the heart beat rate could also lead to an augmentation of the required sampling rate.
These observations justify a deeper investigation of non-uniform sampling techniques in order to sense ECG signals. In this section, we describe the *locally adaptive sampling scheme.*

3.3.1 Locally-adaptive sampling

There are several ways to sample a signal in a non-uniform fashion, but the basic idea behind non-uniform sampling is often the same: a *feature* is extracted from the continuous-time signal and whenever it reaches a threshold, a sample is generated. An example is to have a sampling rate proportional to the area between some fixed value and the signal by integrating it, like in [38, 39][2].
The technique discussed below is based and influenced from the work on *level-crossing sampling* of [59] and has been developed and proposed by Ali Hormati as part of the project presented in Section 1.1.1.

The approach is comparable to AZTEC (cf. Section 2.1.1). Let $x(t)$ be a continuous-time signal and $x[t_n]$, $n \in \mathbb{Z}$, be the nth sample. The scheme works as follows. Having a sample at t_{n-1}, as long as $x(t)$ for $t > t_{n-1}$ remains within the interval $\{x[t_{n-1}] - \delta, x[t_{n-1}] + \delta\}$, no sample is generated. The next sample $x[t_n]$ is defined by $x(t)$ as soon as the continuous-time signal crosses the boundaries. The boundaries are always adapted to the previous sample.
In other words, when the signal exhibits a lot of variations (for example like in the QRS complex), more samples are created and when the signal remains flat, no sample is output. The sampling process is illustrated in Figure 3.4.

This sampling technique can be improved by borrowing some ideas in [37]. The value δ needs to remain relatively small in order to capture the details at moderate slopes. Still, it has been noticed that when the slope is steeper and with a small δ, we have more samples than needed. We can reduce the sampling rate in steep slope regions by adjusting the value of δ and the height of the interval according to the slope. For instance, when the slope of the signal reaches a fixed threshold, δ is multiplied by an integer constant (usually 2,3 or 4), that increases the height of the *sampling interval.* In our current practical implementation, the adjustments of the δ are done

[2] This choice of features is obviously not a good idea for ECG signals, as it is not guaranteed that the sampling rate will be higher in the QRS complex, depending on its orientation

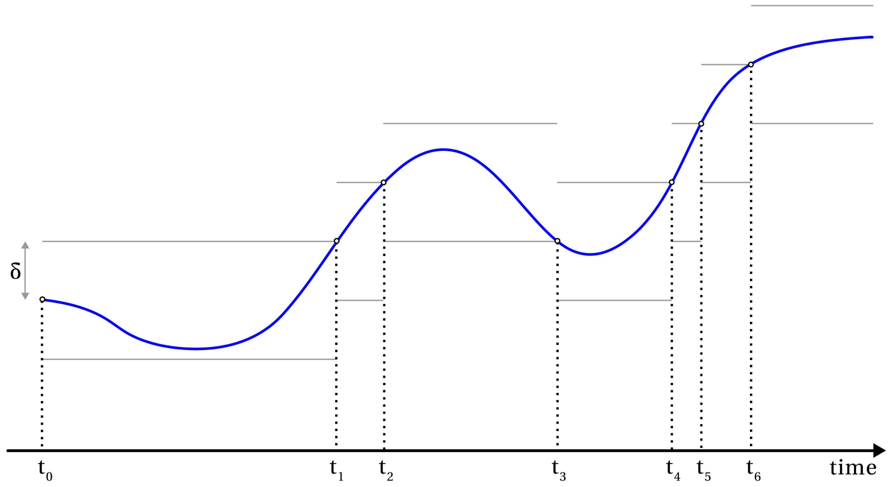


Figure 3.4: Level-crossing sampling: whenever the continuous-time signal crosses the interval, a new sample is generated and the interval is adjusted accordingly. The little circles represent the samples and their time locations are displayed on the x axis.

empirically. An example illustrating this method on a real ECG signal is shown in Figure 3.5.

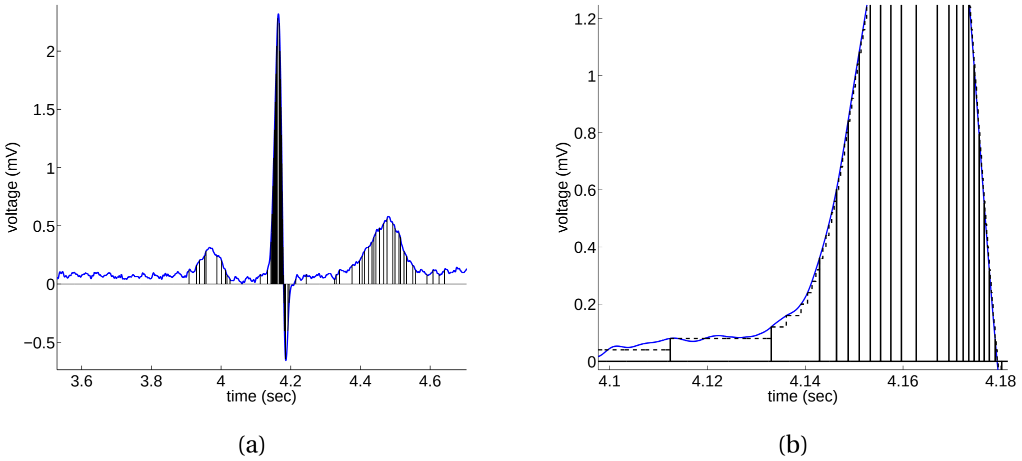


Figure 3.5: Locally adaptive sampling on record 123 from the MIT-BIH database with an average sampling rate of 57 Hz. (b) is a close-up of (a) of the QRS complex that illustrates the augmentation of the interval size when the slope increases. The dashed step function represents the value of δ.

Like in AZTEC, both the locations and the amplitudes of the samples need to be recorded and/or transmitted. However, since the amplitude is always a multiple of δ,

the number of bits required to represent its value can be significantly smaller than the number of bits used to represent the locations. We do not discuss the quantization process and the circuit-level implementation further since they are not in the scope of the thesis.
In Section 3.3.2, we discuss the use of FRI for the analog transmission of the samples.

The big advantage of this technique (and most of the non-uniform sampling techniques) is that the narrow peaks of the QRS complex are well reconstructed since their local sampling rate is much higher than the rest of the signal.
The code implementing this new sampling scheme used for our analysis has been kindly provided by Ali Hormati. It contains a few improvements that are not described above. For instance, it generates a sample when the value of the slopes varies quickly, mimicking AZTEC's behavior that generates a sample when the signal reaches a local minimum or maximum. This improvement contributes to a better representation of the QRS peaks.

3.3.2 Use of FRI for transmission of the samples

Within the framework defined in Figure 1.1, there are two ways we could transmit the data from the patch to the mobile device. The first and most intuitive one consists of quantizing the values and the time locations of the samples and sending all the data digitally.

The second possibility[3] is to directly transmit analog 'samples' as Dirac pulses (or any other pulse shape) over the channel. It is well known that under certain conditions (the most notable example is an independent and identically distributed Gaussian source sent over an additive white Gaussian noise channel), analog transmission is optimal in the sense that no coding strategy can perform better (cf. [23] for an in-depth study on the topic). The analog signal could then be sampled on the mobile device (at a rate high enough) and the classical FRI algorithm (for streams of Diracs) would be used to recover the parameters of the samples, namely their amplitudes and their locations.
One of the advantages of such a sampling scheme is that it does not require any quantization nor any clock. Moreover, it could be implemented using pretty simple and power efficient components. Its biggest drawback is that analog transmission

[3] Suggested by Ali Hormati

does not offer the same flexibility as digital transmission: it is for example impossible to embed error correcting codes or to encrypt the data that is sent over the channel. Finally, note that the number of samples should be estimated and the sampling rate on the mobile device has to be adjusted accordingly.

Hybrid method We can also combine a non-uniform sampling method and an FRI-based method to get the best of both worlds. For example, we tried to combine locally-adaptive sampling with VPW-FRI reconstruction, in order to achieve a low and power efficient sampling rate (while still getting enough information about the QRS complex) on the front end and to benefit from the denoising and smoothing properties along with the high compression rate offered by VPW.

The implementation of this hybrid method is straightforward: the signal is simply sensed in a non-uniform fashion using the algorithm described in Section 3.3.1, extrapolated at a higher and uniform sampling rate and finally passed to VPW-FRI.

4 Implementation details

We study here the details that we need to take care of for a proper practical implementation of the FRI-based algorithms described in Chapters 2 and 3.
For an efficient evaluation of the optimal settings, we introduce the signal to residual ratio (SRR), defined as

$$SRR = 20\log\left(\frac{\sum_{n=0}^{N-1} x^2[n]}{\sum_{n=0}^{N-1} (x[n]-\hat{x}[n])^2}\right). \tag{4.1}$$

This value represents the squared error between the original signal $x[n]$ and its reconstructed version $\hat{x}[n]$. Moreover, we need some test signals in order to confront the algorithms to real data. The signals used throughout this chapter are taken from the MIT-BIH arrhythmia database (cf. Section 5.2 for more details about it).

4.1 Batch processing and stitching

To implement the QRS spline modelling method and VPW-FRI in practice (as well as any transform-based method), we need to segment the input signal before going to the Fourier domain. For both methods, the most straightforward approach is to process each heart beat separately, although for VPW-FRI, more or less any other kind of segmentation would be suitable, as long as the number of pulses inside each segment can be approximated.
In order to extract each beat separately, some QRS detection methods are needed. A couple of these methods are reviewed in appendix A.1.

Once the QRS locations have been found, every beat is extracted as follows. Let

$x[n]$ be the recorded signal sampled at frequency F_s ($n = 0, 1, \ldots, N-1$, where N is the total length of the signal) and let b_i the beat at index i that we want to process. Moreover, let t_{i-1}, t_i, t_{i+1} be the time of occurrence of respectively the R peak just preceding b_i, the R peak of b_i and the R peak just following b_i. To have proper stitching, we need to have some overlapping between every consecutive heart beat. We denote the overlapping time duration as OT (for our practical implementation, $OT = 0.15$ sec[1]). $b_i[n]$ is therefore extracted from $x[n]$ for indices n in the range $\left\lfloor \left(\frac{t_{i-1}+t_i-OT}{2}\right) F_s \right\rfloor, \ldots, \left\lfloor \left(\frac{t_i+t_{i+1}+OT}{2}\right) F_s \right\rfloor$.
The extracted heart beat is then fed to the algorithm (in our case either the QRS spline modelling or VPW-FRI), which outputs the reconstructed beat $\hat{b}_i[n]$.

It remains to stitch the reconstructed beats together. To that end, let us introduce the sequence w of size $L = \lfloor OT \cdot F_r \rfloor$, where F_r is the sampling frequency of the recovered signal. w is used to create a window for the stitching process. Note that F_r is not equal to F_s in the compressed sensing settings (F_s being pushed to its lower limits and therefore F_r being higher).

$$w[n] = \frac{\cos\left(\pi + \frac{n\pi}{L}\right) + 1}{2}, \tag{4.2}$$

for $n = 0, 1, \ldots, L-1$. Here, w simply consists of half of a period of a cosine normalized between 0 and 1, but some other transition shapes could be used. The only constraint on the choice of the window is that the total weight on two consecutive samples is always equal to 1 for a given location. In other words

$$w[m] = 1 - w[L-1-m], \qquad \forall m \in \{0, 1, \cdots, L-1\}.$$

Finally, $\hat{b}_i[n]$ is stitched to the other reconstructed beats using a Tukey window (or other window shapes built from a w satisfying equation (4.2)):

$$x[n+n_L] = \begin{cases} (1 - w[n]) \cdot b_i[n] & n \in \{0, 1, \ldots, L-1\} \\ b_i[n] & n \in \{L, L+1 \ldots, n + n_\Delta - L\} \\ w[n - n_\Delta + L - 1] \cdot b_i[n] & n \in \{n + n_\Delta - L + 1, \ldots, n + n_\Delta\} \end{cases},$$

where $n_L = \left\lfloor \left(\frac{t_{i-1}+t_i-OT}{2}\right) F_r \right\rfloor$, $n_H = \left\lfloor \left(\frac{t_i+t_{i+1}+OT}{2}\right) F_r \right\rfloor$ and $n_\Delta = n_H - n_L$.

[1] One could make this value adaptive to the heart rate to get optimal results, but in our implementation we chose to fix it. The reconstruction does not suffer when this overlapping time is applied to heart rates up to 180 Hz.

This stitching process is easily adapted to live ECG processing (allowing a delay of one heart beat for the analysis of the beat) and can provide precious information about the overall behavior of the algorithms as it can be easily tested on a much larger amount of data.

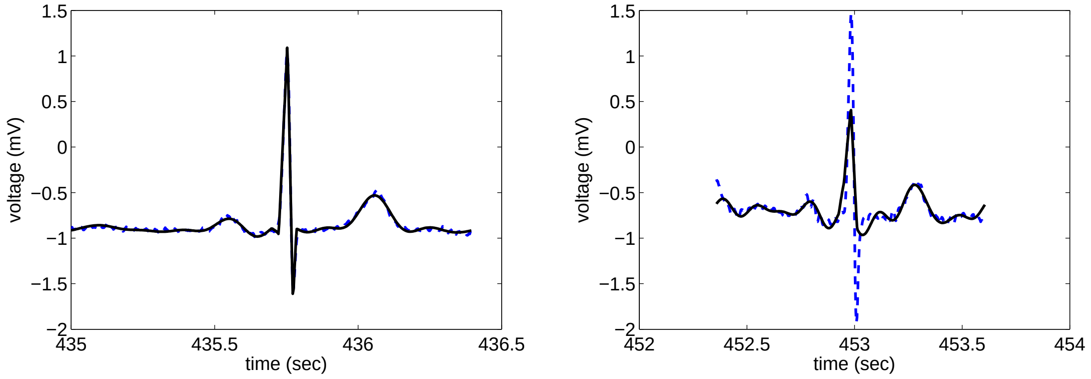
The attentive reader will notice that each heart beat can have a different length. For a practical and low-power implementation, this should be fixed in order to have a fixed-length discrete Fourier transform (DFT). It is also worth noting that no slope analysis is performed as to where the signal should be segmented. In the technique mentioned above, the boundaries of each heart beat are simply chosen to be exactly between the two nearest QRS peaks. In the case of VPW-FRI, if this value appears to be right in the middle of the P wave or T wave, it could be advisable to move it off the wave to avoid duplicated pulses on two consecutive heart beats and therefore possibly achieve a higher compression ratio.

4.2 QRS spline modelling: implementation details and analysis

Even though QRS spline modelling leads to great compression ratios for a very high SRR in some cases, the biggest issue we encountered was its lack of consistency. Even with the QRS windowing proposed in 2.1.4, small variations in the input sometimes generated tremendous differences in the output as it can be observed in Figure 4.1.

Another problem is that the QRS detection needs to be flawless and the QRS width estimation needs to be pretty accurate, which is hard to guarantee in practice. Any variation in either one of these quantities significantly impairs the quality of the reconstructed signal. Last, this method is restricted to a small class of signals whose QRS complex can be modelled with a non-uniform spline, which again is not always verified on practical signals. Indeed, some ECG signals simply do not have a QRS that is shaped like straight lines and that leads to the instability of the algorithm. Examples of such signals are records 102, 104 or 118 from the the MIT-BIH arrhythmia database. Therefore, we do not discuss this technique further in the analysis.

Nonetheless, the idea of parametrizing only the QRS complex still remains of great interest, but using a first order non-uniform spline to approximate it is probably not the best model one could use. The use of 3 overlapping VPW pulses could be part of a



(a)

(b)

Figure 4.1: 2 different samples from record 123 from the MIT-BIH arrhythmia database: (a) a very accurate reconstruction and (b) an imprecise reconstruction. The dashed line is the original signal and the compression parameters are as follow: 20 Fourier coefficients are kept for the low frequency part and 10 parameters are used to model the QRS complex (total of 50 real-valued coefficients per beat).

solution to the problem, but different parametric models could be proposed as well.

4.3 VPW-FRI: implementation details and analysis

This section describes some details for a practical implementation of VPW. As it is a new technique, we provide a more precise and complete analysis.

4.3.1 Periodicity of the input signal

Since the spectrum used by VPW is discrete in frequency, the corresponding input signal is assumed to be periodic in time. We are processing the signal beat by beat, so in most of the cases, we can expect that the input signal is *pseudo-periodic*. However, it is not always verified. One of the factors that could result in the non-periodicity of the signal is the respiratory rhythm, introducing baseline wander in the input signal. In order to improve the accuracy and reliability of the algorithm, we present here a few techniques that can be used to deal with the non-periodicity of the input signals:

1. **Use an extra VPW pulse to model the discontinuity:** if we simply input a non-periodic signal to VPW, the algorithm considers that there is a discontinuity at the borders of the extracted beat. Therefore a pulse is used to model this

discontinuity. This method is very simple in the sense that it does not modify the input signal. The only requirement is to increase the total number of pulses by 1.

2. **Correct the input signal**[2]**:** another technique is to modify the extracted beat so that the amplitude of the starting point and ending point is the same. One way to do this is to add the sequence $c[n]$ to the extracted beat, which consists of half of a period of a cosine:

$$c[n] = \begin{cases} (x[N-1]-x[0])\,\frac{\cos(\frac{n\pi}{N-1})+1}{2} & x[N] \geq x[1] \\ (x[0]-x[N-1])\,\frac{\cos(\frac{n\pi}{N-1}+\pi)+1}{2} & x[N] < x[1] \end{cases},$$

   for $n = 0, \ldots, N-1$, where N is the length of the extracted signal and F_s is the sampling frequency. The adjusted signal is processed with VPW-FRI and $c[n]$ is later subtracted from the reconstructed signal. While this technique spares one pulse, it introduces some skewness in the signal that is proportional to the difference between the edges of the signal. Unfortunately it does not provide any guarantee regarding the continuity of the slope (or some higher order derivative) at the boundaries.

3. **Smooth the boundaries of the beats:** a third possibility is to smooth the edges of the signal *before* VPW processing. This could be done by multiplying the signal in time with a Tukey window, so that the borders of the extracted beat go to zero, ensuring the continuity of the signal at its boundaries. By carefully choosing the Tukey window and by correctly extracting the individual beats (with overlapping) such that total weight of the sum of two consecutive beats is always 1, we are able to reconstruct a smooth and continuous flow of beats.
   Like the first method described here, this approach also requires one or two extra pulses to model the adjustments of the borders. Its advantage is that the slope is now continuous at the boundaries.

4. **Use the discrete cosine transform (DCT)**[3]**:** This approach proposes to replace the DFT by the DCT. Recall that the DCT is a DFT applied to a signal exhibiting an even symmetry. For non-symmetric input signals, it is necessary to first double the length of the signal and fill the second half with its symmetric part. This ensures that the DFT coefficients are real, but more importantly for us,

[2] Suggested by Ronald Crochiere from Qualcomm Inc.

[3] Suggested by Chong Lee from Qualcomm Inc.

it makes the edges of the input signal at the same amplitude (even though it does not guarantee the continuity of its derivatives). This approach seems interesting, however it requires subsequent changes in the FRI algorithm, now that the Fourier coefficients are real.

Unfortunately, the two last methods have not been implemented in practice and are not investigated further. A comparison concerning the first and second method is exhibited in Table 4.1. The two methods have been tested on the first 2 minutes of all the signals (48) of the MIT-BIH arrhythmia database.

Method	Average SRR		
	$F_s = 60$ Hz	$F_s = 120$ Hz	$F_s = 360$ Hz
Extra pulse (do nothing)	16.58 dB	22.06 dB	21.99 dB
Correction of the input	16.54 dB	23.03 dB	23.33 dB

Table 4.1: Comparison of two methods to make the signal input to VPW-FRI pseudo-periodic. The test has been run at three different sampling frequencies: 60, 120 and 360 Hz.

At low sampling frequencies, it seems that the two approaches lead to very similar results. At higher frequencies, the correction of the input with half a period of a cosine has the edge.

4.3.2 Stability and poles outside the unit circle

While the stability of the annihilating filter is not an issue in the original version of FRI for the stream of Diracs[4], it is not guaranteed in the VPW model, and it can happen that some of the roots $u_k = e^{-2\pi/\tau(a_k+it_k)}$ of the annihilating filter are located outside the unit circle. This effect is much more pronounced and frequent when the (noisy) input signal has been previously downsampled, or equivalently when only the low-frequency coefficients are used for the analysis. Here we investigate three ways to deal with poles outside the unit circle.

- **Invert the sign of a_k:** this approach simply inverts the sign of a negative a_k:

$$a'_k = -a_k \qquad \forall a_k < 0.$$

[4] The magnitude of the poles of the annihilating filter is not use in that case, and it is always assumed to be equal to 1.

That way, the new magnitude of the root is proportional to its previous distance to the unit circle: the magnitude of a pole located slightly outside the unit circle does not change too much while a pole located at an infinite distance of the unit circle is mapped to 0 in the complex plane.

- **Bring the pole back on the unit circle:** another way to deal with instability is to simply bring back the pole *on* the unit circle, regardless of its distance to it as

$$a'_k = 0 \qquad \forall a_k < 0. \tag{4.3}$$

- **Discard the pole:** the last thing we can do is simply discard the poles laying outside the unit circle.

Notice that setting $a_k = 0$ in the second approach results in Dirac-shaped pulses in the corresponding parametric signal whenever some poles are outside the unit circle. However, no real ECG signal looks like a Dirac[5]. Therefore, we can refine this technique by placing the poles marginally inside the unit circle at a distance e^R, where R defines an arbitrary threshold for a_k. Incorporating the threshold in equation (4.3), we get

$$a'_k = R \qquad \forall a_k < R.$$

$R > 0$ should be a relatively small value: based on our experiments, $R = \frac{1}{200}$ turns out to be suitable. A similar modification can be applied to the second approach, in which case we discard any pole such that $a_k < R$.

The techniques described above have been tested on the 48 records of the MIT-BIH arrythmia database and the results are displayed in Table 4.2. The input signals have been downsampled to 60 Hz in order to increase the likelihood of getting unstable filters. Note that since a_k is not always strictly positive, the time reconstruction formulas derived in Appendix A.2 do not hold any more. Therefore the reconstruction has to be performed in the frequency domain for the two first rows of the Table[6]. Even in this case, the reconstructed signal is not exact: the spectrum does not roll off

[5] Except when the patient is equipped with a pacemaker, in which case the signal contains very sharp spikes.

[6] In fact, the time domain reconstruction formula for $a_k \leq 0$ gives better results (in terms of SRR) than the frequency domain formula in practice, as the corresponding pulses still have damped spectrum, the only effect of the time domain reconstruction is that it inverts the amplitude sign of the pulses for $a_k < 0$.

any more and we are losing all the frequencies higher than the sampling rate of the reconstructed signal.

Method	SRR
Plain (no correction)	< -10 dB
Sign inversion	13.65 dB
Magnitude correction ($a_k < 0$)	11.77 dB
Magnitude correction ($a_k < 1/200$)	16.44 dB
Discard all poles s.t. $a_k < 0$	11.77 dB
Discard all poles s.t. $a_k < 1/200$	11.99 dB

Table 4.2: Test of different methods to make the annihilating filter stable. The SRR is the average of the SRR obtained on the 48 records (only their first 30 seconds) of the complete MIT-BIH arrhythmia database.

The first thing we observe is that it is really necessary to correct the poles that are outside the unit circle as the outliers tremendously decrease the global SRR.
The best results are achieved when the outside poles are brought back inside at a fixed distance to the unit circle. The role of the proposed adjustment is twofold: as mentioned, it makes the annihilating filter stable by bringing all the poles inside the unit circle, but it also slightly 'smooths' all the Dirac-shaped pulses of the corresponding parametric signal. We also notice that poles on the unit circle should be avoided as they lead to very poor and unsatisfactory results. That can be explained by the shape of the spectrum, which is not decaying any more.

4.3.3 Robustness to DC offset

In this section, we investigate the behaviour and robustness of VPW-FRI with respect to the direct-current (DC) component of the heart beat. Recall that the DC term is represented by the Fourier coefficient $X[0]$. We observed that a single pole is usually used to adjust the DC level of the input signal. The larger the DC value of the signal, the closer the pole to zero. In other words, the width a_k of this corresponding pulse becomes very large and its tails are no more negligible. This situation justifies the use of the periodic formula (2.6) in place of the non-periodic formula (2.3) for the reconstruction in time and Figures 4.2 and 4.3 illustrate well this phenomenon. The signal in Figure 4.2 has a DC component close to 0, hence the pulses are relatively narrow and the choice of the reconstruction formula does not make a big difference. However, when a big artificial DC offset (100 mV in this case) is applied to the same

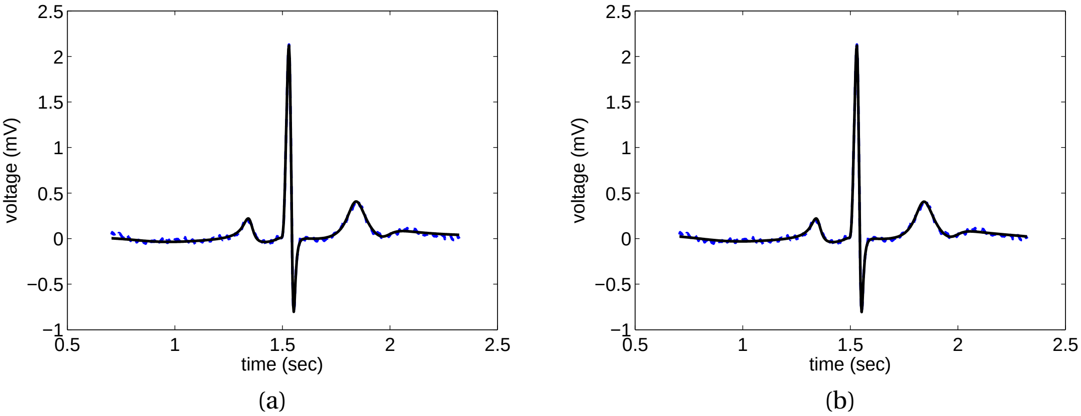


Figure 4.2: ECG pulse with a DC term around 0 mV (a) non-periodic time reconstruction formula and (b) periodic time reconstruction formula. The dashed line is the original signal.

input signal as in Figure 4.3, the use of equation (2.3) gives significantly worse results compared to equation (2.6).

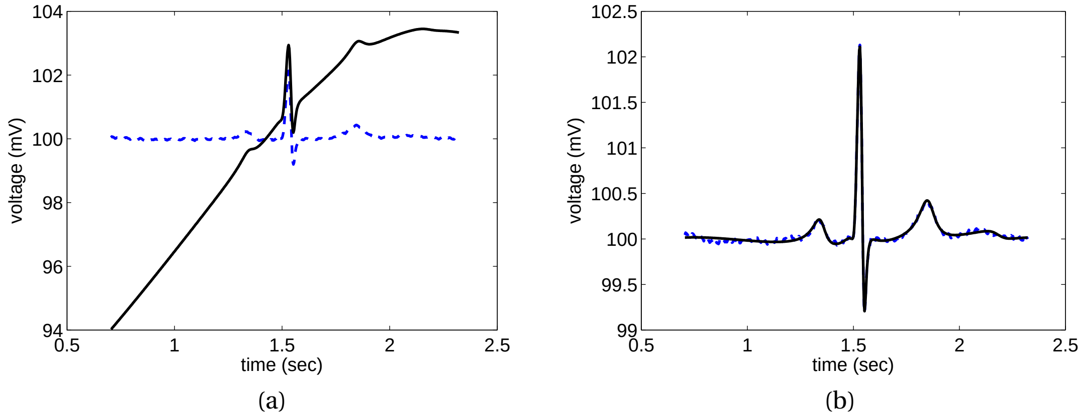


Figure 4.3: ECG pulse with an artificial DC term of ~ 100 mV: (a) non-periodic time reconstruction formula and (b) periodic time reconstruction formula. The dashed line is the original signal.

4.3.4 Running-time improvement: Beat prediction[7]

This section intends to improve the total running-time of the VPW algorithm, in particular the root finding phase. The idea[8] presented here has been borrowed from pitch prediction techniques [52] in speech processing and aims at diminishing the power consumption and fasten the running-time of the VPW analysis phase. To that end, it takes advantage of the similarities between two consecutive heart beats.

The computation of the annihilating filter and the denoising phase being the most power consuming parts of the algorithm, the essence of this approach is that, instead of optimizing the algorithms, we could also reduce the number of times the filter is computed. Indeed, when two consecutive beats are similar, we can assume that their annihilating filter is also looking the same. To check that this is the case, we simply convolve the annihilating filter we found for the previous beat with the Fourier coefficients of the current beat. If the norm of the result is smaller than some fixed threshold κ, we can legitimately assume that the filter is adequate and reuse the coefficients t_ks and a_ks from the previous beat. As for the c_ks and d_ks, we simply compute them using equation (1.15) with the previous roots u_ks and the new set of coefficients. This process is described in Algorithm 3.

Algorithm 3 Extracting the parameters of VPW pulses using beat prediction

```
Input: the vector $\boldsymbol{u} = [u_0, u_1, \dots, u_{K-1}]$ containing the roots of the annihilating filter
  of the previous beat, the set of Fourier coefficients $\boldsymbol{x} = [X[0], X[1], \dots, X[M-1]]$ of
  the current beat (with period $\tau$) and a threshold $\kappa$.
Output: the corresponding pulses parameters $\boldsymbol{t} = [t_0, t_1, \dots, t_{K-1}]$, $\boldsymbol{a} =$
  $[a_0, a_1, \dots, a_{K-1}]$, $\boldsymbol{c} = [c_0, c_1, \dots, c_{K-1}]$ and $\boldsymbol{d} = [d_0, d_1, \dots, d_{K-1}]$.
  $\boldsymbol{h}$ = polynomial($\boldsymbol{u}$)
  if $\|\boldsymbol{x} * \boldsymbol{h}\|_2 > \kappa$ then
      $\boldsymbol{u}$ = compute annihilating roots for $\boldsymbol{x}$
  end if
  $\boldsymbol{t} = -\frac{\tau \angle \boldsymbol{u}}{2\pi}$
  $\boldsymbol{a} = -\frac{\tau log|\boldsymbol{u}|}{2\pi}$
  find $\boldsymbol{c}$ and $\boldsymbol{d}$ from $\boldsymbol{t}$ and $\boldsymbol{x}$
  return $\boldsymbol{t}, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{d}$
```

Of course this approach trades some accuracy in the recovery in exchange for a faster running-time. We tried this method on record 123 (30 min of data) from

[7] Joint work with Ali Hormati.

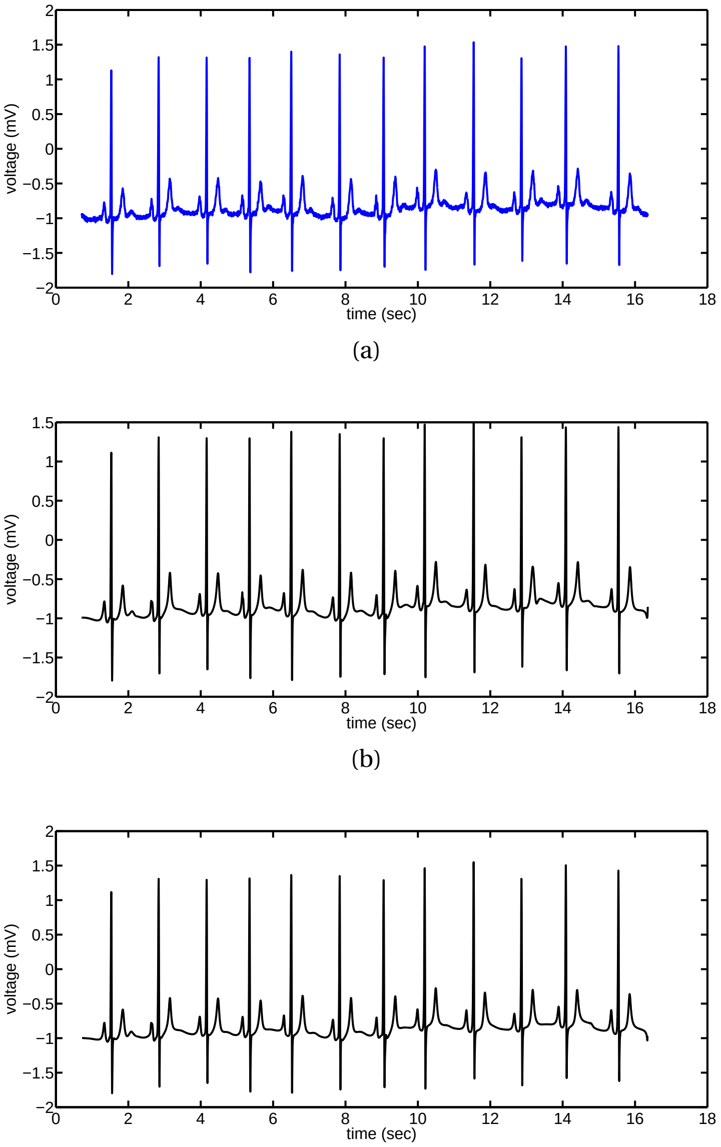
[8] Suggested by Chong Lee and Ronald Crochiere from Qualcomm Inc.

the MIT-BIH arrhythmia database and the results we obtained are summarized in Table 4.3. In 57% of the cases, the annihilating filter is not computed, that makes it around 29% faster. The SRR dropped by around 3 dB but the visual aspect of the signal can still be considered as acceptable as it can be seen in Figure 4.4.

Method	Global SRR	Running-time
raw VPW-FRI	22.44 dB	27.52 sec
with beat prediction	19.33 dB	19.60 sec

Table 4.3: Comparison of plain VPW with and without beat prediction on record 123 from the MIT-BIH database. The input sampling rate is 360 Hz, $K = 7$ and the threshold (for beat prediction) κ is 0.06. Note that the running-time includes some overhead for the QRS detection and the stitching process for example.

Note that to maximize the chances of having similar consecutive beats, it is important to align them according to their pulses positions in time. We chose in this report to align them relatively to their R peaks.



(a)

(b)

(c)

Figure 4.4: Beat prediction: (a) extract of record 123 from the MIT-BIH database at 360 Hz, (b) waveform reconstructed signal using plain VPW & (c) waveform reconstructed signal using beat prediction.

5 Performance Evaluation and Results

In this chapter, we first provide some individual results for VPW-FRI. Next, we evaluate the compression performance of the FRI-based algorithms and compare them to the AZTEC compression algorithm, which is the one we chose to implement as it is very popular in the literature and widely used in practice. Finally, we measure the compressed sensing performance of non-uniform sampling compared to uniform sampling methods (including FRI-based techniques).

5.1 Metrics

We use two main measures to evaluate the performance and compare the algorithms introduced in Chapters 2 and 3. Most of the literature utilizes the *compression ratio* P_{CR} [55], defined as

$$P_{CR} = \frac{\#\text{bits to represent } x(n)}{\#\text{bits to represent } \hat{x}(n)}. \tag{5.1}$$

To compute it, we would need to quantize the compressed and the input samples, which unfortunately does not fall into the scope of this thesis. Instead, we simply assume that the quantization of the input values and the compressed samples are the same. Therefore, it allows us to simplify equation (5.1) as follows:

$$\begin{aligned} P_{CR} &= \frac{\text{\#bits to represent } x(n)}{\text{\#bits to represent } \hat{x}(n)} \\ &\approx \frac{\text{\#bits/sample} \cdot \text{\#samples to represent} x(n)}{\text{\#bits/sample} \cdot \text{\#samples to represent } \hat{x}(n)} \\ &= \frac{\text{\#samples to represent } x(n)}{\text{\#samples to represent } \hat{x}(n)}. \end{aligned}$$

It should be noted that the compression rate is proportional to the sampling rate of the input signal: for instance, by upsampling it by a factor of 10, we can also increase the compression rate by a factor of 10, which does not lead to a fair comparison. Therefore throughout this chapter, we prefer to simply use the number of samples (or the number of samples per second) needed for the reconstruction. We also use the SRR introduced in equation (4.1) to evaluate the quality of the reconstructed signal relative to the original. Although this metric gives a good idea of the accuracy of a given algorithm, this measure is not always the best indicator of the performance of the algorithms. Indeed, since most of the time the original signal already contains noise, it is not always likely to perfectly approximate perfectly the input signal as it is probably better to get rid of some of the noise.

5.2 Test signals

Several different types of signals have been used to compare and analyse the performance of the algorithms previously described. Below is a list of the main sorts of signals experimented in this chapter.

- **The MIT-BIH database:** The MIT-BIH database [25] is the main source of data for our analysis. It is a free, very complete database and it contains close to 4 TB of recorded ECG signals. The MIT-BIH database has been used to assess the performance of algorithms in countless publications and a complete description of it can be found in [46]. We principally worked with the arrhythmia database, a subset of the main database that is probably the most popular in the literature. The arrhythmia database contains 48 different records sampled at 360 Hz.

- **MIT ECG synthesizer:** The MIT ECG synthesizer is open source Matlab software implemented by P. McSharry and G. Clifford [45], which simulates noise-free

ECG signals using Gaussian shaped pulses and models respiratory rhythm as a low-frequency oscillating sine wave.

- **Self-recorded signals:** We also worked on signals we recorded ourselves by using the Zephyr BioHarness™ recording device [1]. These test signals offered us the possibility to evaluate the algorithms under particular conditions, for instance very high beat rates and noisier signals.

An extract of the three aforementioned signals is pictured in Figure 5.1. It should be noted that the signals used for comparison (with the exception of the ECG synthesizer) already contain noise.

5.3 VPW: individual results

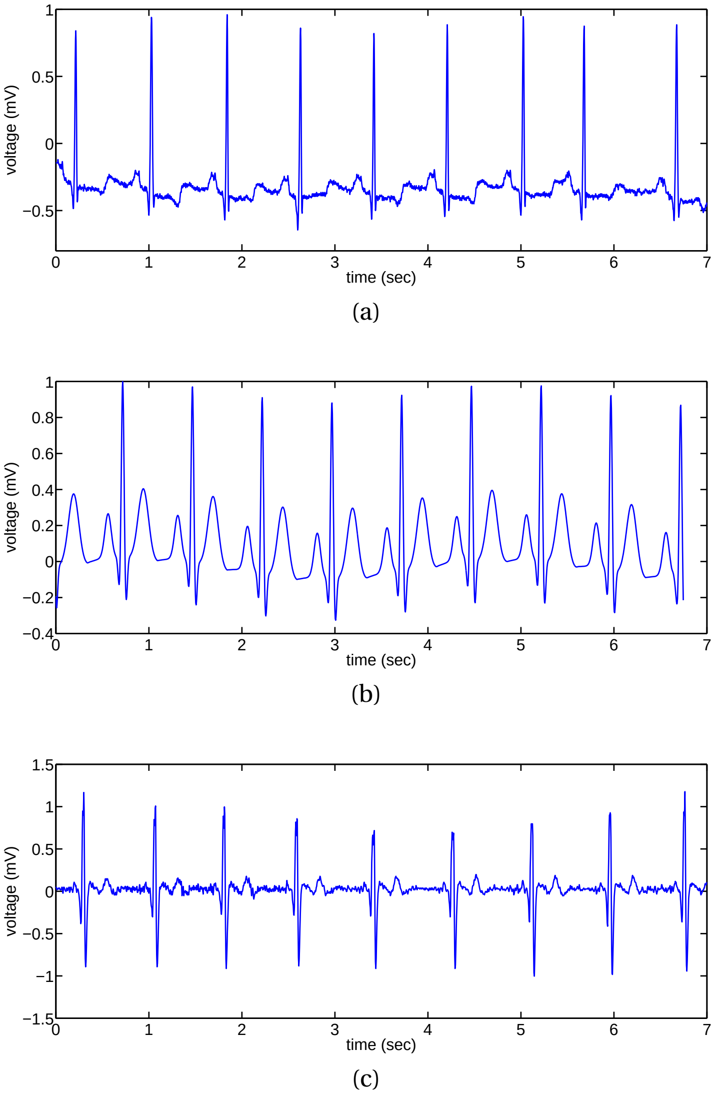
This section provides an evaluation of the performance of VPW. The main aspects studied are the number of coefficients used for analysis and the noise removal capabilities of VPW.

5.3.1 Sampling rate and number of coefficients used

In the noiseless settings and with no model mismatch, we know that the sampling rate needed for an accurate VPW analysis has to be greater than or equal to the rate of innovation of the signal. However, in the presence of noise and model mismatch, sampling a signal at a frequency close to the rate of innovation is usually not sufficient for a faithful reconstruction. Here we study the limitations of VPW when it is applied to real ECG signals. To that end, we test it on the MIT-BIH arrhythmia database (48 records, 1 minute each) at different sampling rates, ranging from 50 Hz to 360 Hz (the native sampling rate) and we record the resulting SRR. The number of pulses for each beat in the VPW model has been set to 7, meaning that a minimum of 15 Fourier coefficients are necessary. Recall that the number of coefficients available N is determined by the sampling frequency F_s and the length (period) of the beat τ (two consecutive coefficients are $\frac{1}{\tau}$ Hz far from each other) as

$$N = \frac{F_s}{2}\tau. \tag{5.2}$$

[1] http://www.zephyr-technology.com



(a)

(b)

(c)

Figure 5.1: Different types of signals used: (a) ECG record from the MIT-BIH arrhythmia database, (b) ECG modelled with Gaussian pulses generated from the MIT ECG synthesizer & (c) self-recorded signal using the Zephyr BioHarness™ recording device.

For instance, 15 Fourier coefficients require a signal of period 1 second to be sampled at 30 Hz at least. The results of the aforementioned experiment are illustrated in Figure 5.2. As mentioned in Section 3.2.2, at low sampling frequencies the quality of the reconstructed signal is significantly worse than at higher sampling rates. We see in Section 5.6 how enhanced VPW (exposed in Section 3.2.2) can improve the reconstruction in the low sampling rate regime.

We also observe that there is clearly a plateau starting at around 120-130 Hz, meaning

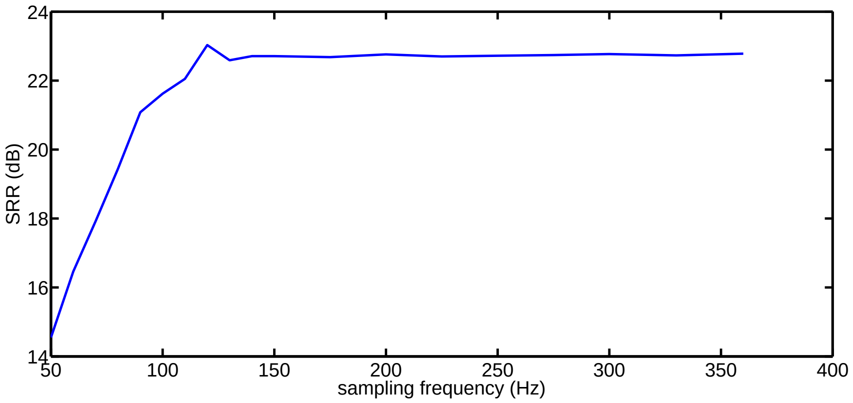
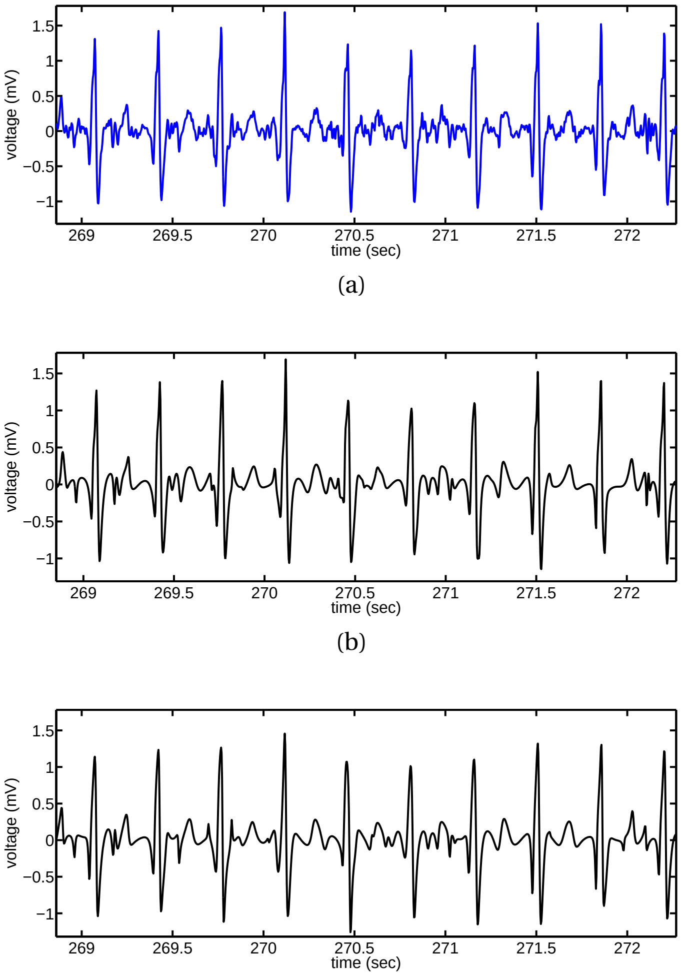


Figure 5.2: Average SRR of the first minute of all records of the MIT-BIH arrhythmia database vs. input signal sampling rate.

that a higher sampling rate does not lead to improved results and simply implies heavier computation costs and therefore slower running times.
It is also interesting to note the peak at 120 Hz, which is probably due to the 60-Hz interference noise we can find in most of the signals: its presence in the analysis slightly harms the performance of the algorithm. A fix that could possibly solve this problem is the use of a notch filter at 60 Hz for higher sampling rates.

Concerning the number of coefficients used, there is one more thing we should take care of: from equation (5.2), we see that the number of coefficients available is proportional to the period τ of the beat. Therefore a shorter beat means less Fourier coefficients for a fixed sampling rate. We have to make sure that VPW is also reliable when it is used with signals having a very high heart rate. Since we have not found any high heart rate signal in the MIT-BIH database, we have used our own records containing a maximal heart rate of 180 beats per minute to investigate this issue[2]. The algorithm is tested with signals sampled at 250 Hz (the native sampling rate of the recording device) and with the same signals after downsampling to 120 Hz. On a record of 8 minutes of data, we obtained a global SRR of respectively 16.01 dB and 14.98 dB, meaning that the downsampled signal performs almost as well as the original version. The SRRs measured appear to be relatively low, but it is mainly due to the fact that the original record is quite noisy, as we can see in Figure 5.3, representing a fragment of the signal and its reconstructions. Note finally that due to the presence of noise, K has been fixed to 9 for the analysis.

[2] I myself did this experiment and I am still alive!



(a)

(b)

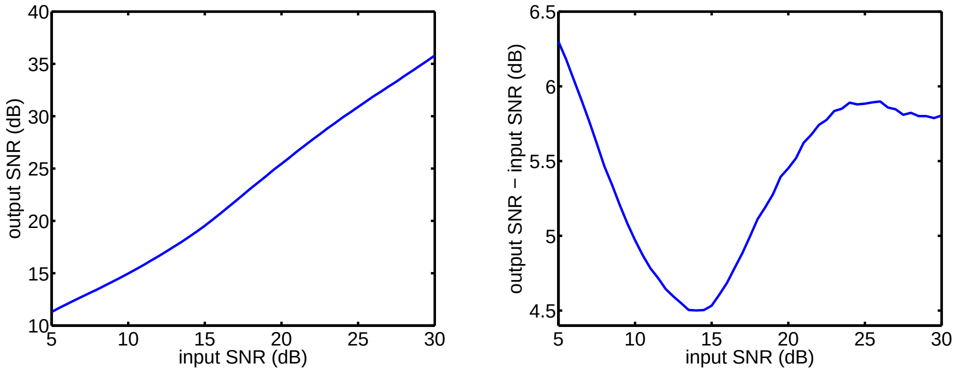
(c)

Figure 5.3: (a) Extract of a high heart rate signal sampled at 250 Hz, VPW reconstruction of (b) the same signal & (c) a version of the signal downsampled to 120 Hz.

5.3.2 Noise removal with VPW

We investigate here the use of VPW as a noise reduction algorithm. Unfortunately, we do not have access to a pure noise-free real ECG signal to analyse the noise removal performance of VPW, so we consider here three different input signals: an ECG signal generated from VPW pulses (hence no model mismatch), an ECG signal generated from the MIT synthesizer (model mismatch this time) and a somewhat clean real ECG signal from the MIT-BIH arrhythmia database. Additive white Gaussian noise (AWGN) has been added to these signals and the output signal-to-noise ratio (SNR) has been measured and compared with the input SNR.

For VPW-based signals with added noise, the experiments have been run with different noise levels and the results are illustrated in Figure 5.4. We see that VPW-FRI improves the SNR by at least 4.5 dB for this type of signal.



(a)

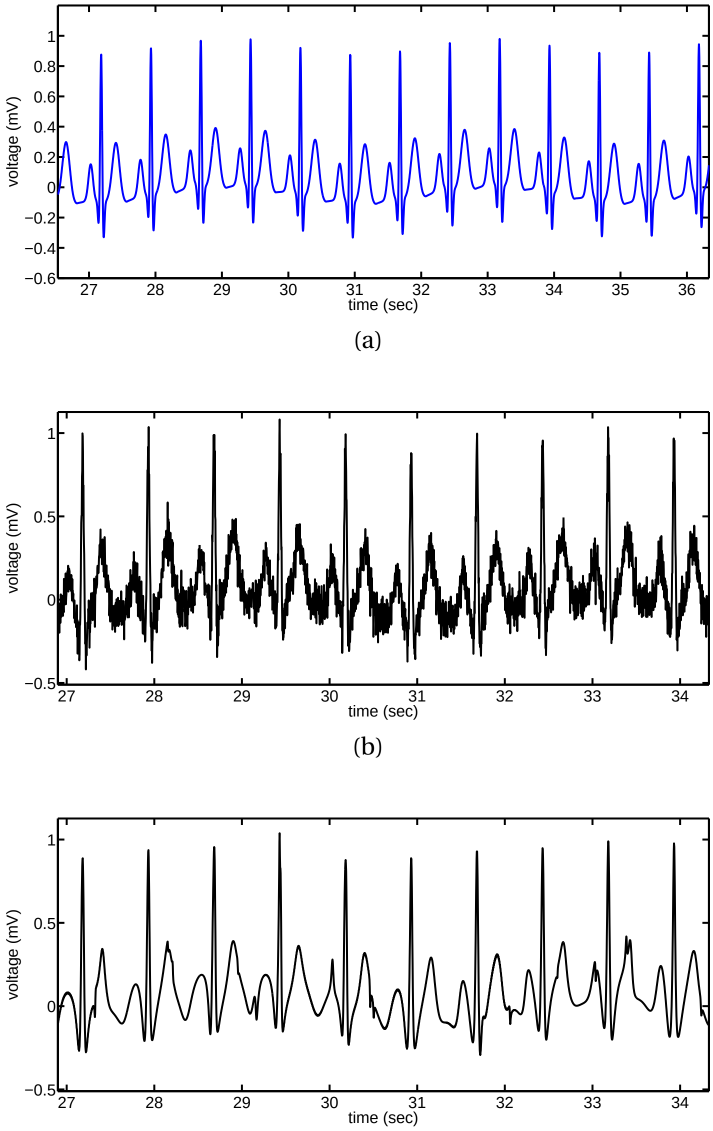
(b)

Figure 5.4: Noise reduction of VPW algorithm using a VPW-based signal as input: (a) measured output SNR & (b) improvement of the SNR (output SSNRNR - input SNR).

Additionally, we have used the MIT synthesizer to generate noise-free Gaussian-shaped pulses and added some noise corresponding to an input SNR of 10.0 dB. Applying VPW on 30 minutes of such a signal, the resulting SNR went up to 15.0 dB. An extract of this experiment is displayed in Figure 5.5. Similarly, we picked a relatively clean signal from the MIT-BIH arrhythmia database and artificially added some AWGN with an SNR of 15.0 dB to the signal. The output SNR of the reconstructed signal jumped to 19.4 dB in this case! Figure 5.6 illustrates the signals used. As we can see from Figures 5.5 and 5.6, VPW-FRI preserves the relevant shapes of the signal while removing most of the noise.
Another advantage of VPW over other conventional methods is that they usually rely on a threshold that needs to be adapted to every input signal and the signal goes through a normalization process. In the case of VPW, the sensitivity of the algorithm is directly controlled by the value K (which corresponds to the number of pulses) and no normalization is needed.

5.4 Performance comparison: Cadzow and ESPRIT

This section is devoted to an analysis of the denoising preprocessing stage used in FRI presented in Section 1.4. Cadzow and ESPRIT have been tested on the first minute



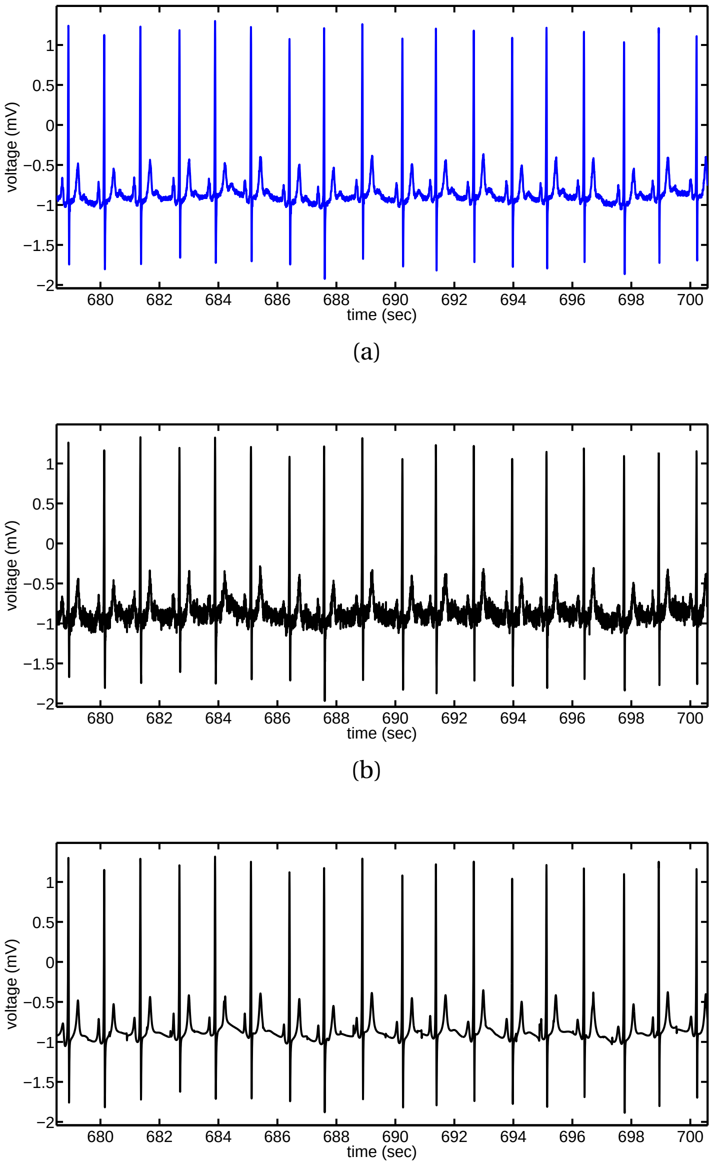
(a)

(b)

(c)

Figure 5.5: Denoising using VPW-FRI: (a) original signal with Gaussian pulses from the MIT synthesizer, (b) original signal with 10-dB AWGN noise in it (b) and the output of VPW-FRI fed with the noisy signal ($K = 9$). Apart from the noise, the reconstruction also suffers from model mismatch (Gaussian vs. VPW pulses).

of every file from the MIT-BIH arrhythmia database (48 files in total, sampled at 360 Hz) and the preprocessed signals have been sent to VPW with the parameter K set to 7. For the evaluation, we consider the total running-time of the algorithms and the reconstruction accuracy. The results of this experiment are summarized in Table 5.1. It should be noted that all the Fourier coefficients are used for VPW analysis (i.e. we



(a)

(b)

(c)

Figure 5.6: Denoising using VPW-FRI (a) original signal (record 123 from the MIT-BIH arrhythmia database), (b) original signal with 15-dB AWGN noise in it (b) and the output of VPW-FRI fed with the noisy signal ($K = 7$).

use frequencies up to 180 Hz (corresponding to a sampling rate of 360 Hz), while 60-65 Hz (sampling rate of 120-130 Hz) is sufficient as we have seen in Section 5.3.1), which slows down the computations and exaggerates the differences (recall that the size of the matrices increase linearly with the number of samples).

First we observe that the denoising process is necessary in practice as it greatly

Method	Running-time	SRR	Number of iterations/beat (if applicable)
No denoising	17.28 sec	12.11 dB	-
Cadzow	83.40 sec	15.40 dB	2
Cadzow	181.10 sec	16.82 dB	5
Cadzow	342.05 sec	17.98 dB	10
Cadzow	679.31 sec	19.90 dB	20
Cadzow	1114.60 sec	21.41 dB	35
ESPRIT TLS	45.46 sec	21.98 dB	-
ESPRIT LS	45.21 sec	22.14 dB	-

Table 5.1: Comparison the denoising methods

improves the quality of the reconstructed signals. Additionally, it accounts for a significant part of the overall timing: at least 60% when using ESPRIT and a lot more when Cadzow is used.

ESPRIT is clearly the method performing the best, both in terms of SRR and running-time. Cadzow can also lead to good results but the price to pay is a large number of iterations (around 25 or 30 for each beat) if we want to reach the same SRR as with ESPRIT, which makes the total running-time much slower. For a better control on the number of Cadzow iterations, we decided to fix it instead of relying on the threshold t. For the types of signals we studied, an average number of 35 iterations corresponds to $t = 10^{-5}$.

Finally, we see that finding the roots using total least squares or simply least squares in ESPRIT does not change much (in fact the simple least squares method has the edge over the total least squares in our analysis).

5.5 Performance comparison: compression

VPW-FRI has been extensively discussed in Section 4.3. Now it is compared with other compression algorithms such as AZTEC or a simple downsampling of the input signal.

As a preliminary remark and as mentioned in Section 5.1, we do not address the problem of quantization in this thesis, therefore we simply focus on the *number of samples per second* as a compression measure. Recall that for every AZTEC-sample, we need to store the amplitude and the location, so the corresponding number of samples is 2 for each AZTEC-sample. The number of samples can be made variable by playing with the parameters of the different compression algorithms: for AZTEC,

we can increase it by lowering the error tolerance ϵ and for VPW, we can augment the number of VPW pulses for each heart beat.
Again, a test is performed using the MIT-BIH arrhythmia database (all 48 samples) and for each selection of the number of samples, the resulting signal is reconstructed at 360 Hz (the original frequency of the input signal) and its SRR is calculated. The results of this experiment can be seen in Figure 5.7. For the sake of comparison, we also include in the Figure the SRR obtained by simply downsampling the signal (low-pass filtering followed by decimation) to a given rate corresponding to the number of samples per second[3].

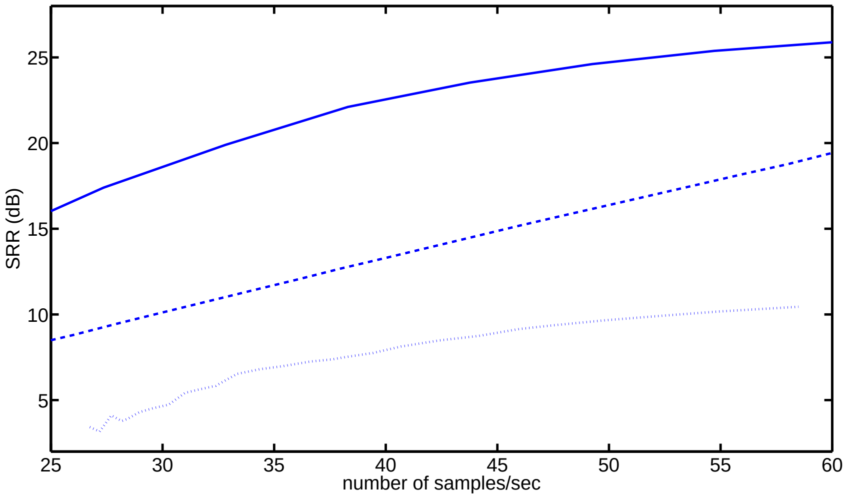


Figure 5.7: SRR corresponding to the number of samples per second for VPW (continuous line), sinc interpolation (dashed line) & AZTEC (dotted line).

VPW-FRI is the clear winner of this experiment: its beats AZTEC by almost 15 dB and sinc interpolation by around 8 dB for a fixed compression rate. Moreover, it is very good at preserving the morphological information of the signal, which makes it a very interesting candidate for compression.

The low performance of AZTEC can be partly explained by the shape of the reconstructed signal: it exhibits a lot of steps and discontinuities that highly decrease the value of the SRR. It could probably be greatly improved by using different interpolation methods for the reconstruction of the waveform.

[3] The resulting SRR is computed using an upsampled version (back to the original rate) of this signal.

Observe that QRS spline modelling has not been included in the analysis since the inconsistency problems described in Section 4.2 were leading to non-relevant results.

5.6 Performance comparison: compressed sensing

This section is dedicated to the analysis of the compressed sensing view of the algorithms. The main criterion for evaluation is to get the best possible reconstruction with the minimum number of inputs, sampled[4] from a continuous-time signal. At first, we focus on uniform sampling and then study the spectrum extrapolation accuracy of VPW and its enhanced version on low sampling rate input signals. The algorithms are also compared to sinc interpolation. Later, we investigate the compression performance of non-uniform sampling with locally-adaptive sampling.

5.6.1 Uniform sampling

VPW-FRI As we have seen in Section 5.3.1, plain VPW is not optimal for low-frequency signals. The main problem is that it misses the very narrow peaks whose high frequencies are destroyed in the downsampling process (see Figure 3.1). We can compare the spectrum extrapolation performance of VPW with a simple sinc interpolation in time. The average SRR for different input sampling rates is depicted in Figure 5.8. VPW-FRI is clearly not performing as well as sinc interpolation: although it allows much higher compression ratios, it still requires high sampling rates for its input signals. And of course at high sampling rates, since the samples contain almost all the energy of the signal, sinc interpolation is very efficient (even though it does not achieve any compression or denoising).

Enhanced VPW As suggested in Section 3.2.2, we can improve the SRR of the reconstructed signal by correcting the magnitude of the extrapolated part of its spectrum. A comparison of enhanced VPW with plain VPW and sinc interpolation is shown in Figure 5.9. The three extrapolation techniques have been tested on the beginning of the 48 samples of the MIT-BIH arrhythmia database. As we can see, enhanced VPW certainly improves the performance of plain VPW. It even beats sinc interpolation at low sampling rates (that is, between 50 & 60 Hz). The difference between them does not seem that significant, but recall that the reconstructed signals are compared

[4] Since all the experiments are performed on digital signals, there is not real sampling phase.

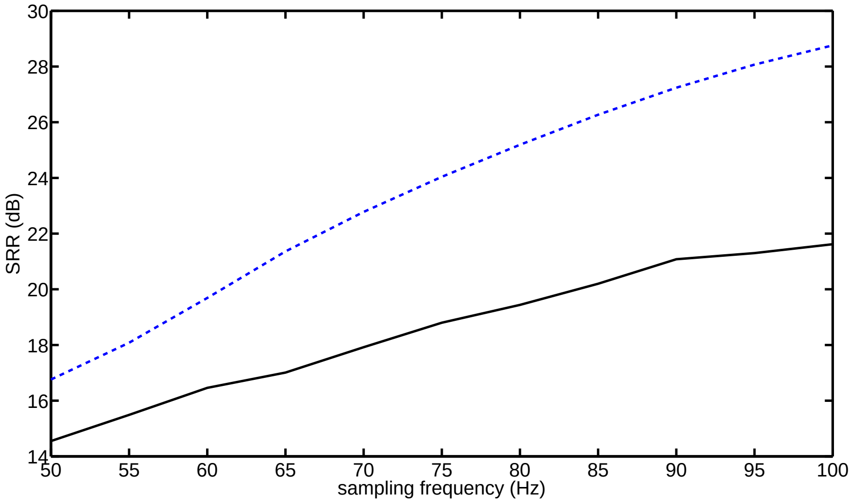
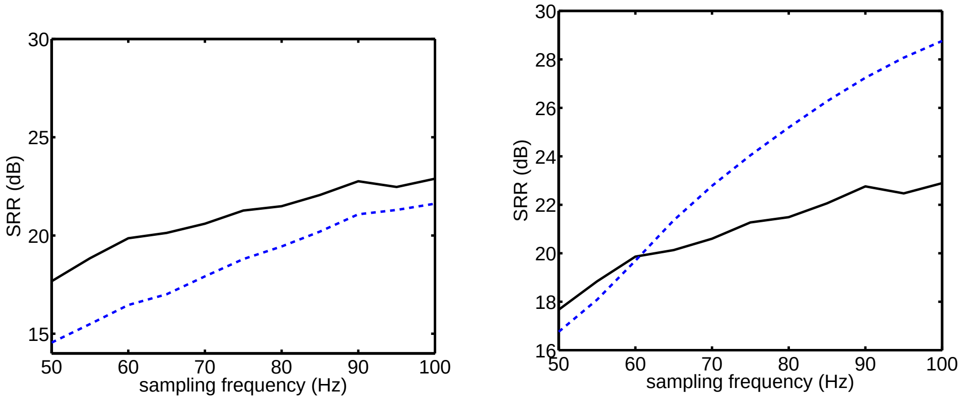


Figure 5.8: Continuous-line: SRR of the reconstructed signal using plain VPW-FRI for different sampling rates. Dashed line: SRR of the reconstructed signal using sinc interpolation.

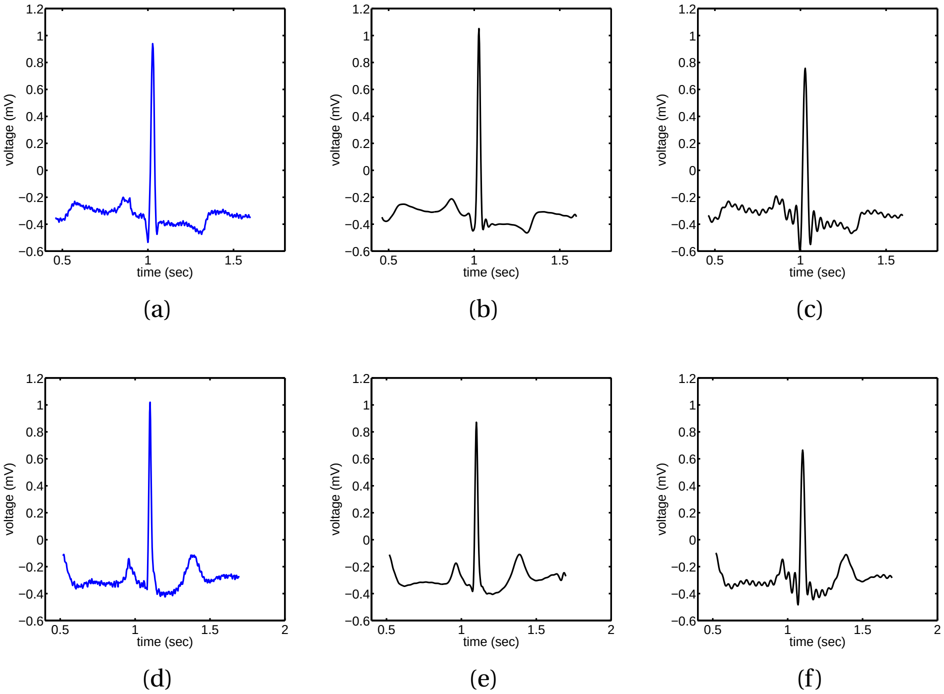


(a)

(b)

Figure 5.9: (a) Continuous-line: SRR of the reconstructed signal using enhanced VPW-FRI for different sampling rates. Dashed line: SRR of the reconstructed signal using plain VPW-FRI. (b) Continuous-line: SRR of the reconstructed signal using enhanced VPW-FRI. Dashed line: SRR of the reconstructed signal using sinc interpolation.

with *noisy* input signals (typically, the removal of the 60-Hz interference noise by VPW affects the resulting SRR). Moreover, the SRR has been computed on the *average* of all the records in the database: some of the signals almost have no more energy above 25-30 Hz. By picking signals with a spectrum containing more energy in the high frequencies, the advantage of enhanced VPW becomes more convincing. For instance, Figure 5.10 represents an extract of records 100 and 101 from the database, containing sharper peaks and therefore higher frequencies. Visually, we clearly see the advantage of enhanced VPW over sinc interpolation as it exhibits much less ringing effect and the peaks amplitudes are better approximated. We can also look at the SRR for different sampling rates, and this time enhanced VPW indubitably wins over sinc interpolation as it can be seen in Figure 5.11. Finally, remember that sinc interpolation requires all Fourier coefficients or time samples to be accurate, while VPW uses always only 28 samples (for $K = 7$) per block for its reconstruction.

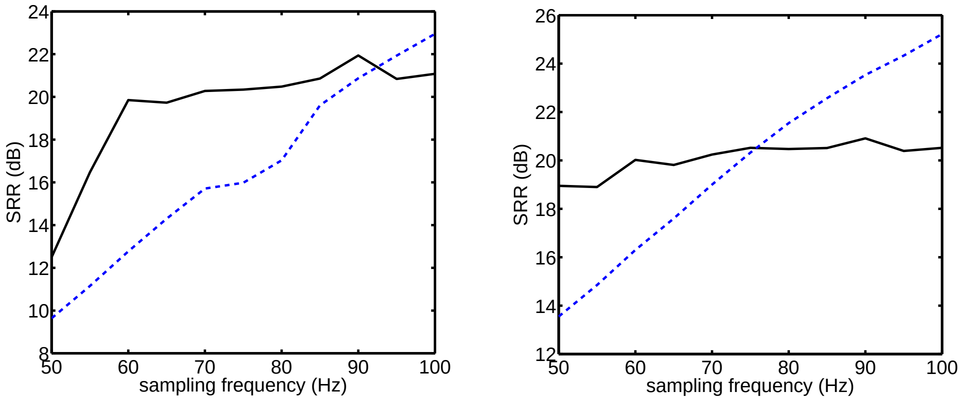


(a) (b) (c)

(d) (e) (f)

Figure 5.10: Extract of records (a) 100 & (d) 103 from the MIT-BIH database at 360 Hz, (b) & (e): reconstructed signal from enhanced VPW & (c) & (f): reconstructed signal using sinc interpolation. The input signal to the algorithms is always downsampled to 50 Hz.

Assuming that VPW has also access to the same coefficients as sinc interpolation (no compression needed), we can even push the performance a bit further by reusing the part of the spectrum of the input signal and use only VPW to extrapolate the unknown coefficients. Based on our experiments, this approach allows us to gain 1-2 dB for the low sampling frequencies when applied to the complete MIT-BIH arrhythmia database.



(a) (b)

Figure 5.11: Continuous-line: SRR of the reconstructed signal using enhanced VPW-FRI, dashed line: SRR of the reconstructed signal using sinc interpolation for records (a) 100 & (b) 101 from the MIT-BIH database.

5.6.2 Non-uniform sampling: locally-adaptive sensing

We now turn to non-uniform sampling and study the performance of locally-adaptive sampling as described in Section 3.3.1. The analysis presented here is very similar to the one exposed in Section 5.6.1. We run the algorithm for several values of δ and we compute the SRR corresponding to every different value. Recall that δ controls the average sampling rate of the algorithm: the smaller δ, the higher the average sampling rate. Since we are in a non-uniform sampling scheme, both the location and the amplitude need to be recorded for a single sample. However, observe that the amplitude is always represented by integer multiples of δ, which greatly restricts the range of values needed. Assuming that the locations are quantized with 16 bits, our practical implementations showed that 6 to 8 bits are enough for the amplitude. Consequently, we multiply the average sampling rate by 1.5 to get an estimation of the average number of values to be recorded and/or transmitted[5]. This experiment is summarized in Figure 5.12. The shape of the curve is similar to the results obtained with enhanced VPW in Figure 5.9(b).

Last thing we have tried with locally-adaptive sampling is to branch its output to the VPW algorithm for better and smoother recovery, as described in Section 3.3.2. The results are only marginally better in terms of SRR, but the visual appearance is

[5] A more thorough investigation on the quantization process would probably lead to more accurate results.

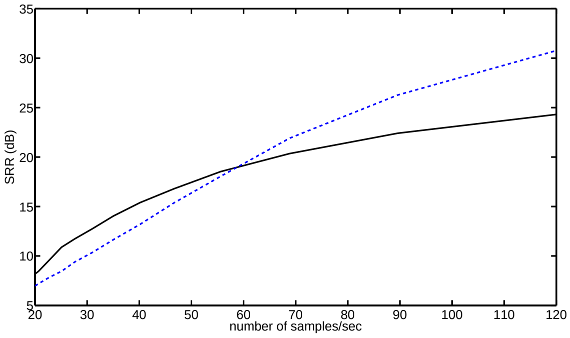
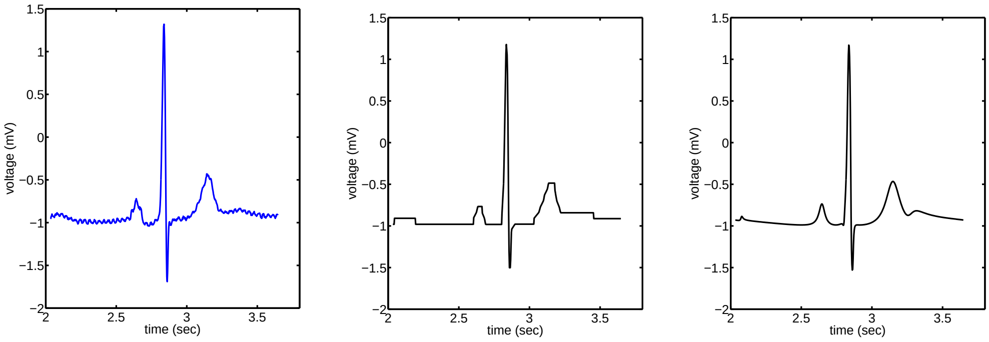


Figure 5.12: Continuous-line: SRR of the reconstructed signal using locally adaptive sampling for different sampling rates. Dashed line: SRR of the reconstructed signal using sinc interpolation.

more pleasant, as it is illustrated in Figure 5.13.



(a)

(b)

(c)

Figure 5.13: (a) Extract of the original signal (extract of record 123 from the MIT-BIH arrhythmia database), (b) signal sensed at an average rate of 30 Hz with locally-adaptive sampling & (c) output of the low sampling rate signal by VPW-FRI.

5.7 Chapter Summary

Since this chapter presents several results concerning the methods studied in the thesis and in particular VPW-FRI, we summarize here its key points main results obtained:

- **Noise removal:** first of all, VPW-FRI has proven to be a very powerful noise removal algorithm as it preserves well the shapes of ECG pulses while increasing the SNR by around 5 dB on average.

- **Cadzow vs. ESPRIT:** as a preprocessing stage for VPW-FRI, ESPRIT TLS (or LS) algorithm is preferred to Cadzow, mainly due to its running-time that is clearly faster.
- **Compression:** concerning the compression performance, signals reconstructed with VPW-FRI beat AZTEC compression by 15 dB in terms of SRR.
- **Compressed sensing:** as for compressed sensing, VPW-FRI (with the correction of its spectrum) has the edge over sinc interpolation *on average* and is significantly better at extrapolating the spectrum of ECG signals with high energy in the higher frequencies. Locally-adaptive sampling is also an interesting alternative over uniform sampling, but recall that it needs a different front-end architecture than the classical uniform sampling ADCs.

6 Conclusion and future work

We started by briefly reviewing the background necessary to study ECG signals and the state of the art in ECG compression methods. Then we introduced VPW as a new parametric approach based on FRI. We presented an in-depth analysis of VPW and provided several details regarding the practical implementation of this compression scheme. In particular, we described a routine to process long duration ECG signals as well as methods to extract single beats and detect the R peaks in ECGs. We investigated approaches to deal with the non-periodicity of the input signals: the best result obtained being to simply add half a period of a cosine before the VPW analysis (cf. Section 4.3.1. Moreover, we analysed the potential instability of the annihilating filter that was more frequent when a reduced number of Fourier coefficients was used. We proposed several ways to avoid poles outside of the unit circle of the complex plane.

We also evaluated the performance of two denoising methods used to 'massage' the data before VPW analysis: Cadzow and ESPRIT. Both were comparable in terms of accuracy of the reconstructed signal[1] but ESPRIT has proven to be significantly faster than Cadzow.

Next we compared VPW with AZTEC, an ECG compression technique broadly referenced in the literature and widespread in practical implementations. VPW turned out to have notably better reconstruction accuracy for an equivalent compression rate, achieving 20 dB of matching for only 30 samples per second, surpassing AZTEC by almost 15 dB. Moreover, part of the mismatch was due to the removal of the 60-Hz interference present in the original data.

Further, as the annihilator roots-finding is a costly process in terms of power and the bottleneck of the algorithm (along with the preprocessing phase for denoising),

[1] At the costs of a sufficient number of iterations for Cadzow.

we suggested beat prediction to reduce the number of times the roots are computed and the denoising process is done. The essence of beat prediction is to use a single annihilating filter for similar heart beats. Our early results decreased the running-time by almost 30% while keeping a reasonable reconstruction.
We also tested the use of VPW for other purposes such as cleaning up noisy signals: its parametric nature generates smooth signals and allows to preserve the morphology while getting rid of most of the interferences present in the input data. We managed to have an average of more than 5 dB improvement compared with noisy input signals.

We also investigated compressed sensing and non-uniform sampling in this thesis. We reviewed locally-adaptive sampling, a scheme that adapts the sampling rate to the slope of the signal by generating more samples when the slope gets steeper and no sample when the voltage remains flat. We studied its performance compared to uniform sampling paired with sinc interpolation and the results obtained were slightly in favor of locally-adaptive sampling for low sampling rates (around 50-60 Hz). Similar research has also be done regarding VPW for different input sampling rates and in particular low rates, where its performance were not satisfactory in the presence of noise and model mismatch. The spectrum extrapolation by VPW was not leading to better results than a simple sinc interpolation in terms of SRR. With enhanced VPW, we proposed a way to correct the magnitude of the reconstructed spectrum, which achieves much better results than plain VPW and finally beats sinc interpolation at low sampling rates.

In conclusion, VPW has proven to be a very powerful compression scheme as well as an interesting noise removal technique. Even though the extraction of the parameters is relatively power costly, the reconstruction of the waveforms is straightforward and has the advantage of providing a continuous-time representation.
Furthermore, the performance of VPW at low sampling rates could enable the development of low-power sensors and decrease the transmission costs of ECG data. The parameters and pulse shapes extracted with VPW could also lead to various new techniques in automated ECG diagnosis.

We wish that the work done in this thesis and principally on VPW will contribute to the success and the popularity of the FRI sampling theory. We also hope that it will lower the power consumption barriers in the state-of-the-art ECG recording devices and reduce the storage and transmission information needed while preserving an accurate representation of ECG signals. Finally, we wish that the research conducted at Qualcomm Inc. will encourage the experimentations, enable the use of VPW on a wider range of signals and open up new areas of research!

6.1 Future work

We expose here the work that remains to be done regarding the current state of the project. We also suggest some potential new areas that could be investigated and propose a new usage and a different outlook on the VPW algorithm.

The approaches described here are simply suggestions and are based on quick reflections and discussions. They should not be taken as facts as none of them has been tested in practice or proven to be correct.

6.1.1 Estimate the number of pulses in VPW

The number of pulses (the value K) in VPW determines the compression ratio of the record: the smaller the number, the higher the compression rate. However, a smaller number of VPW pulses increases the risk of missing some important features in the signals. In the presence of high-power noise, a small K can even lead the algorithm to completely miss the P wave or the T wave. Hence finding the correct value K is always a matter of trade-offs between high compression ratios and faithful reconstruction, the ultimate goal being to always find the minimal K such that all the features are preserved. One way to do it would be to run the algorithm for different values of K and to look at the phase of the annihilator roots (corresponding to the locations of the pulses). If decreasing K does not dramatically change the locations, it is probably safe to assume that we did not miss some important features. On the contrary, if a smaller K completely changes the roots positions in the complex plane, the value K is probably getting too small.

6.1.2 Multi-lead processing

In this thesis, even though we experimented and evaluated VPW on various leads, we never processed two or more leads in parallel. By doing so, apart from having a more robust QRS detection, we could also take advantage of the fact that the pulse location in a given heart beat does not change from lead to lead. Therefore, we could possibly compare the t_ks and if needed correct them across different leads.

6.1.3 Quantization and A/D conversion

Even though different sampling methods have been proposed and mentioned in this thesis, it should be noted that none of them have been implemented in practice in the current state of the project. One of the challenges remaining is to produce an ADC and possibly consider different converters both with uniform sampling and non-uniform sampling architectures. Similarly, the thesis does not address the quantization question, which is a crucial part of the project and truly allows a more precise and complete analysis of the compression rates we can achieve. Indeed, the behavior and performance of VPW still need to be evaluated with quantized samples.

6.1.4 Diagnosis using VPW-FRI

From a different point of view, VPW-FRI can possibly enable new diagnosis techniques and improve or simplify machine-learning-based ECG interpretation. There are already some known algorithms that can automatically interpret ECG and diagnose potential malfunctions or heart diseases. Among others, we can mention the VERITAS™ software[2] from Mortara®.

The parameters extracted by VPW directly characterize the pulse shapes and could be used as precious and useful features for an algorithm trained to detect ECG failures[3]. In particular, the characteristics of every single wave could be directly derived from the VPW parameters. This section provides a few thoughts and ideas about the implementation of such a method.
First, even though it leads to a less accurate analysis, symmetric VPW pulses might be preferred to asymmetric ones, as they provide a decomposition in pulses that is easier to read and interpret. Figure 2.5 indeed exhibits more distinguishable pulses than Figure 2.7, where the pulses are much more correlated with each other. We can obtain symmetric pulses by forcing the amplitudes $\{c_k\}_{k=0}^{K-1}$s to be real, which is done

[2] http://www.mortara.com/products/clinical-research/veritas-algorithms/.

[3] This idea has been suggested by John Hong from Qualcomm Inc.

by making the matrix defined in equation (1.12) real and by doubling its height as

$$\boldsymbol{V} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \Re\{u_0\} & \Re\{u_1\} & \cdots & \Re\{u_{K-1}\} \\ \Im\{u_0\} & \Im\{u_1\} & \cdots & \Im\{u_{K-1}\} \\ \vdots & \vdots & \ddots & \vdots \\ \Re\{u_0^{K-1}\} & \Re\{u_1^{K-1}\} & \cdots & \Re\{u_{K-1}^{K-1}\} \\ \Im\{u_0^{K-1}\} & \Im\{u_1^{K-1}\} & \cdots & \Im\{u_{K-1}^{K-1}\} \end{bmatrix}.$$

Similarly, we make the vector of equation (1.13) real as

$$\mathbf{x} = [X[0], \Re\{X[1]\}, \Im\{X[1]\} \ldots, \Re\{X[K-1]\}, \Im\{X[K-1]\}]^T.$$

where $\Re\{x\}$ and $\Im\{x\}$ are respectively the real and imaginary parts of x. Computing $\boldsymbol{c}$ as in equation (1.15) but using the matrix and vector defined above leads to results with no complex part.

Another thing one should pay attention to is that a different DC component can completely change the positions of the pulses. Therefore it would be wise to set it to a fixed value in order to have more uniformity between the different samples. The idea of using VPW as a feature extractor for beats classification seems very promising but again, its performance and usability remain to be proven.

6.1.5 Use of VPW to synthesize ECG signals

There are a wide range of ECG synthesizers that can be found for free on the Internet, but unfortunately, the shape of the pulses does not look very realistic in most of them. Since VPW has proven to be an interesting and suitable model for the analysis of ECG signals, we could also think of using it as a tool to generate ECG waveforms. Compared to the MIT ECG synthesizer, it could for example provide asymmetry in the pulses, a feature that is not available in this widespread simulator.

We also think that the study of an ECG generator will lead to a better understanding of the different parameters of VPW pulses and could probably be also beneficial for diagnosis purposes.

A Appendix

A.1 QRS detection

Most of the compression methods presented in the thesis are transform based, hence they can only process each heart beat separately. In order to chunk down the ECG signal into several pieces, we need to be able to detect the QRS complex with accuracy. There are plenty of ways to detect the QRS complex, as this problem has been widely studied in the literature over the past decades. Since QRS detection is not the main focus of this thesis, we quickly review here the most common detection methods and we give more details on the ones used for our implementation. More details and more sophisticated algorithms can be found on this topic in [34] and [55].

QRS detection methods In general, the detection of the QRS complex is made of two phases:

- *preprocessing stage*, where features are extracted. This is usually further decomposed into a linear filtering stage and a nonlinear transformation phase.
- *decision stage*, which consists of comparing the processed data against a threshold.

The algorithms developed for QRS detection can be classified (relative to their preprocessing stage) into the following categories:

- **Signal derivative and digital filters:** these techniques usually use bandpass filter to extract the band of interest of the QRS complex (see for instance [3]).

They also use the slope information, as the QRS is the part that exhibits the most variations in the signal. A more detailed example is provided in Section A.1.1.

- **Wavelets and filter banks-based QRS Detection:** the basic idea behind such methods is to detect singularities using local maxima of the wavelet coefficients. See [2] for a complete example.
- **Neural networks:** the most used artificial neural networks for ECG detection are the multilayer perceptron [9], radial basis function neural networks (RBFN) [12] and learning vector quantization (LVQ) [9]. A neural network always requires a training phase, which is either supervised or non-supervised (as in LVQ). The latter category appears to be quite costly in terms of computational power. An example of a use of neural networks for QRS detection can be found in [8].
- **Additional approaches:** other approaches have been proposed. Among others, we can mention hidden Markov models (HMM) [14], Hilbert transform-based QRS detection [53], maximum a posteriori (MAP) estimation [11],...

The next sections present two examples of QRS detections methods which have been investigated more thoroughly in this report and implemented in practice.

A.1.1 Derivative-based approach

The first method implemented for the segmentation process is based on [34]. The first step is to bandpass the signal in the frequency of interest of the QRS complex (10 to 25 Hz as in [34]). A feature vector $z(n)$ is then created as a linear combination of the first and second derivatives of the bandpassed signal. The first and second derivatives are estimated as [3]:

$$y_1[n] = x[n+1] - x[n-1],$$
$$y_2[n] = x[n+2] - 2x[n] + x[n-2].$$

In our practical implementation, the feature is defined as [3]

$$z[n] = \tilde{y}_1[n] + \left|y_2[n]\right|, \tag{A.1}$$

where $\tilde{y}_1[n] = \{0.25, 0.5, 0.25\} * \left|y_1[n]\right|$ is a smoothed version of $\left|y_1[n]\right|$. In our settings, it turned out that slightly smoothing the vector $z[n]$ was giving better results by avoiding local maxima. Figure A.1(b) represents the feature vector $z[n]$ applied to an ECG signal.

For the detection phase, the feature vector is simply compared against a threshold denoted by Θ_x. Note that the threshold is dependent on the signal as different signals may lead to different results. For example [34] suggests a threshold of $0.3\max(x)$ or $0.4\max(x)$ when the feature vector is simply the first derivative. Based on our experiences, a threshold of $0.5\max(x)$ is suitable for the feature vector defined in Equation (A.1).

A.1.2 Second approach

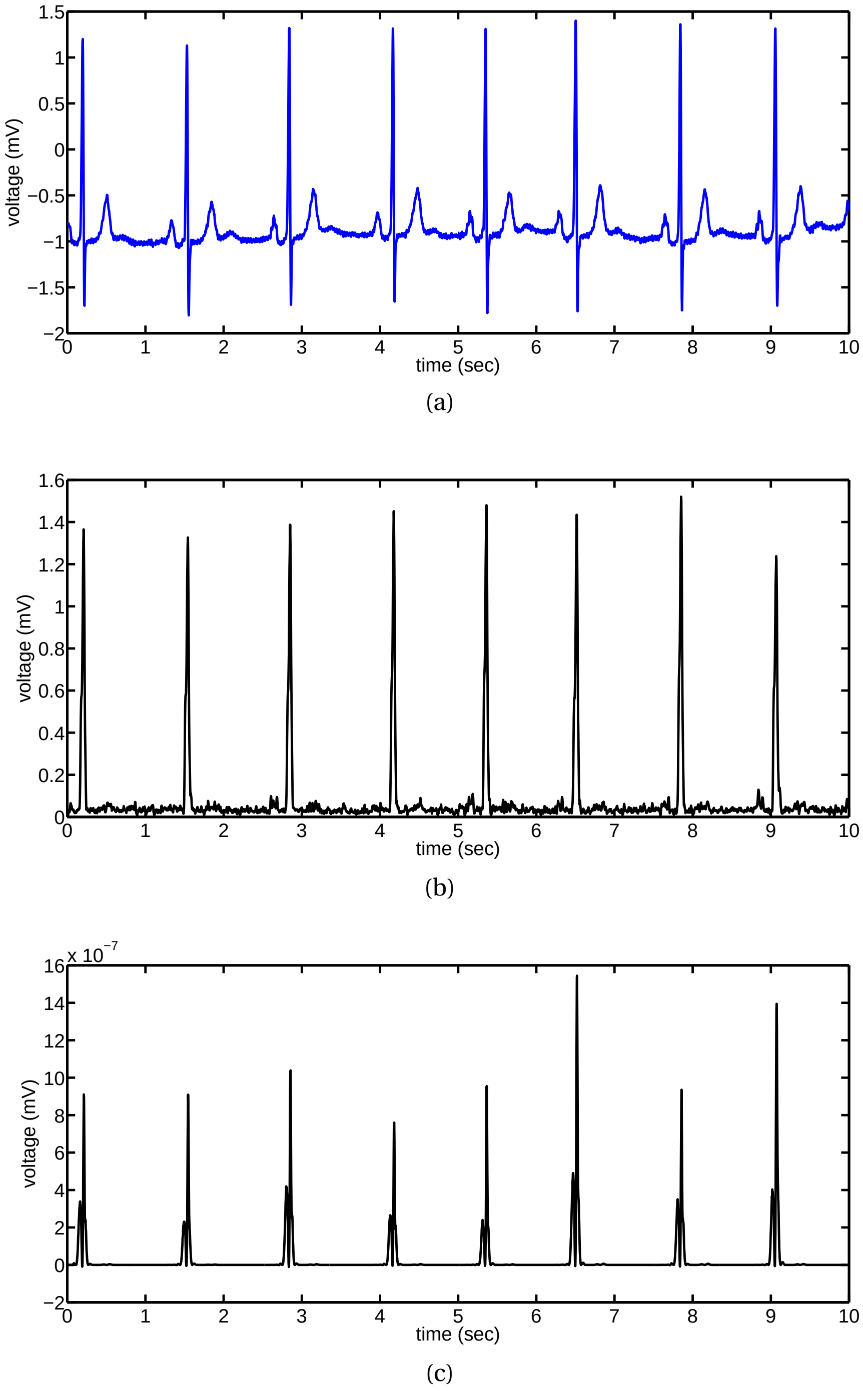
Another method which has been implemented and experimented on the current project is a real-time algorithm due to J. Pan and W. J. Tompkins [49]. It provides fairly better results than the simpler method presented in Section A.1.1 although its preprocessing stage is quite similar to it. The biggest difference resides in the thresholding phase, in which the threshold is constantly adapted to the feature vector. The preprocessing stage works as follows:

- First the signal is bandpass filtered between 5 and 15 Hz.
- Then the square of the derivative of the signal is computed in order to extract the slope information.
- Finally, the feature signal is integrated over a window of a size corresponding to the widest possible QRS width (150 ms in [49] and 120 ms according to [5]).

One of the strengths of this algorithm is that its threshold is adaptive. The way it works is briefly described here. In a nutshell, the algorithm uses two estimates: the signal peak SP and the noise peak NP. The two peaks are constantly updated for each new peak encountered. The threshold T_1 is defined as a combination of the two peaks:

$$T_1 = NP + 0.25(SP - NP).$$

A peak is classified as a SP whenever it is higher or equal to T_1 and as NP when it is lower. Finally, a second threshold $T_2 = 0.5T_1$ is defined and used to search back the



(a)

(b)

(c)

Figure A.1: Example of an ECG signal (a) and the QRS features described in (b) Equation (A.1) and (c) in Section A.1.2.

QRS peak when the algorithm fails to find it in a given period of time.[1]

[1] Note that the original paper uses in fact two feature vectors in parallel, the second one simply being the bandpassed signal. We chose to use only the other feature vector in our implementation.

We also experimented with the Teager energy operator (TEO) Ψ [32]:

$$\Psi\left[x[n]\right] = x[n]^2 - x[n+1]x[n-1],$$

which is a simple operation that approximates the energy of the signal $x[n]$. It has already been successfully used for QRS detection [63] and its simplicity makes it easy to be implemented in practice. The feature vector corresponding to the method described above is illustrated in Figure A.1(c).

A.2 Periodic continuous-time formulae for VPW-FRI

This section shows the details of the computation of the periodic continuous time domain formula of VPW-FRI using its Fourier series coefficients $X[m]$. It also demonstrates its link with the Cauchy-Lorentz distribution.

First recall that the Fourier series coefficients of the symmetric part of VPW-FRI are $X^{(1)}[m] = \frac{1}{\tau}e^{-a_k 2\pi|m|/\tau}e^{-i2\pi m t_k/\tau}$ for $m \in \mathbb{Z}$. Therefore, the time-domain representation of this signal can be computed as the inverse Fourier transform:

$$\begin{aligned}
x^{(1)}(t) &= \sum_{m=-\infty}^{\infty} X^{(1)}[m]e^{i2\pi m t/\tau} \\
&= \frac{1}{\tau}\sum_{m=-\infty}^{\infty} e^{-a_k 2\pi|m|/\tau}e^{-i2\pi m t_k/\tau}e^{i2\pi m t/\tau} \\
&= \frac{1}{\tau}\left(\sum_{m=0}^{\infty} e^{2\pi/\tau(-a_k+i(t-t_k))m} + \sum_{m=0}^{\infty} e^{2\pi/\tau(-a_k-i(t-t_k))m} - 1\right) \\
&\overset{(a)}{=} \frac{1}{\tau}\left(\frac{1}{1-e^{2\pi/\tau(-a_k+i(t-t_k))}} + \frac{1}{1-e^{2\pi/\tau(-a_k-i(t-t_k))}} - 1\right).
\end{aligned}$$

Note that (a) assumes that $a_k > 0$, which only holds when the annihilating filter is *stable*. By setting $z_t = e^{2\pi/\tau(-a_k+i(t-t_k))}$ we get:

$$\begin{aligned}
x^{(1)}(t) &= \frac{1}{\tau}\left(\frac{1}{1-z_t} + \frac{1}{1-z_t^*} - 1\right) \\
&= \frac{1}{\tau}\frac{1-|z_t|^2}{(1-z_t)(1-z_t^*)}, \qquad \text{(A.2)}
\end{aligned}$$

where $|z_t|$ is the magnitude of z_t.

Since the frequency domain is discrete, the above signal is periodic in time with period τ. To get a formula for a non-periodic continuous-time signal, it suffices to take the limit of equation (A.2) for $\tau \to \infty$. For simplicity, let us define $\beta_k(t) = -a_k + i(t - t_k)$.

$$\begin{aligned}
x^{(1)}_{non-per}(t) &= \lim_{\tau\to\infty} \frac{1}{\tau} \frac{1 - |z_t|^2}{(1 - z_t)(1 - z_t^*)} \\
&= \lim_{\tau\to\infty} \frac{1}{\tau} \frac{1 - e^{-4\pi a_k/\tau}}{1 - e^{2\pi\beta_k(t)/\tau} - e^{2\pi\beta_k^*(t)/\tau} + e^{-4\pi a_k/\tau}} \\
&\overset{(b)}{=} \lim_{\tau\to\infty} \frac{-1 - \frac{4\pi a_k}{\tau} e^{-4\pi a_k/\tau} + e^{-4\pi a_k/\tau}}{2\pi\beta_k(t) e^{2\pi\beta_k(t)/\tau} + 2\pi\beta_k^*(t) e^{2\pi\beta_k^*(t)/\tau} + 4\pi a_k e^{-4\pi a_k/\tau}} \\
&\overset{(c)}{=} \lim_{\tau\to\infty} \frac{2a_k e^{-4\pi a_k/\tau} - \frac{4\pi a_k^2}{\tau} e^{-4\pi a_k/\tau}}{-\pi\beta_k(t)^2 e^{2\pi\beta_k(t)/\tau} - \pi(\beta_k^*(t))^2 e^{2\pi\beta_k^*(t)/\tau} + 4\pi a_k^2 e^{-4\pi a_k/\tau}} \\
&= \frac{2a_k}{-\pi\beta_k(t)^2 - \pi(\beta_k^*(t))^2 + 4\pi a_k^2} \\
&= \frac{a_k}{\pi\left(a_k^2 + (t - t_k)^2\right)}, \qquad \text{(A.3)}
\end{aligned}$$

where (b) and (c) are derived from l'Hôpital's rule. Observe that (A.3) is simply the Cauchy-Lorentz distribution!

We can do the same reasoning for the asymmetric component of the signal, $X^{(2)}[m] = -\frac{1}{\tau} i e^{-a_k 2\pi|m|/\tau} e^{-i2\pi m t_k/\tau} \operatorname{sign}(m)$. We have

$$\begin{aligned}
x^{(2)}(t) &= \sum_{m=-\infty}^{\infty} X^{(2)}[m] e^{i2\pi m t/\tau} \\
&= -\frac{1}{\tau} \left(\sum_{m=0}^{\infty} i e^{2\pi/\tau(-a_k + i(t-t_k))m} + \sum_{m=0}^{\infty} i e^{2\pi/\tau(-a_k - i(t-t_k))m} \right) \\
&\overset{(d)}{=} -\frac{1}{\tau} \left(\frac{i}{1 - e^{2\pi/\tau(-a_k + i(t-t_k))}} + \frac{i}{1 - e^{2\pi/\tau(-a_k - i(t-t_k))}} \right).
\end{aligned}$$

Again, (d) only holds for $a_k > 0$. Setting $z_t = e^{2\pi/\tau(-a_k + i(t-t_k))}$ we get

$$\begin{aligned}
x^{(2)}(t) &= -\frac{1}{\tau} \left(\frac{i}{1 - z_t} + \frac{i}{1 - z_t^*} \right) \\
&= \frac{1}{\tau} \frac{2\Im\{z_t\}}{(1 - z_t)(1 - z_t^*)}. \qquad \text{(A.4)}
\end{aligned}$$

Taking the limit of equation (A.4) for $\tau \to \infty$ and with $\beta_k(t) = -a_k + i(t - t_k)$, we have

$$
\begin{aligned}
x^{(2)}_{non-per}(t) &= \lim_{\tau\to\infty} \frac{1}{\tau} \frac{-i(z_t - z_t^*)}{(1-z_t)(1-z_t^*)} \\
&\overset{(e)}{=} \lim_{\tau\to\infty} \frac{i e^{2\pi\beta_k(t)/\tau}\left(1+\frac{2\pi\beta_k(t)}{\tau}\right)}{2\pi\beta_k(t)e^{2\pi/\tau 2\pi\beta_k(t)} + \beta_k^*(t)e^{2\pi\beta_k^*(t)/\tau} + 4\pi a_k e^{-4\pi a_k/\tau}} \\
&\quad - \frac{i e^{2\pi\beta_k^*(t)/\tau}\left(1+\frac{2\pi\beta_k^*(t)}{\tau}\right)}{2\pi\beta_k(t)e^{2\pi\beta_k(t)/\tau} + 2\pi\beta_k^*(t)e^{2\pi\beta_k^*(t)/\tau} + 4\pi a_k e^{-4\pi a_k/\tau}} \\
&\overset{(f)}{=} \lim_{\tau\to\infty} \frac{-i\beta_k(t) e^{2\pi\beta_k(t)/\tau}\left(2+\frac{2\pi\beta_k(t)}{\tau}\right)}{-2\pi\beta_k(t)^2 e^{2\pi\beta_k(t)/\tau} - 2\pi(\beta_k^*(t))^2 e^{2\pi\beta_k^*(t)/\tau} + 8\pi a_k^2 e^{-4\pi a_k/\tau}} \\
&\quad + \frac{i\beta_k^*(t) e^{2\pi\beta_k(t)/\tau}\left(2+\frac{2\pi\beta_k(t)}{\tau}\right)}{-2\pi\beta_k(t)^2 e^{2\pi\beta_k(t)/\tau} - 2\pi(\beta_k^*(t))^2 e^{2\pi\beta_k^*(t)/\tau} + 8\pi a_k^2 e^{-4\pi a_k/\tau}} \\
&= \frac{-2i\beta_k(t) + 2i\beta_k^*(t)}{-2\pi\beta_k(t)^2 - 2\pi(\beta_k^*(t))^2 + 8\pi a_k^2} \\
&= \frac{t - t_k}{\pi\left(a_k^2 + (t-t_k)^2\right)},
\end{aligned}
\tag{A.5}
$$

where (e) and (f) are obtained using l'Hôpital's rule. Note that equation (A.5) can be computed faster by using the fact that $x^{(2)}_{non-per}(t))$ is simply the Hilbert transform of $x^{(1)}_{non-per}(t)$ and the following property:

$$
\frac{1}{t^2+1} \overset{\text{Hilbert transf.}}{\Longrightarrow} \frac{t}{t^2+1}. \tag{A.6}
$$

A.3 Matlab code

The Matlab® code implementing the methods experimented in this thesis is available on http://lcav.epfl.ch/.

Bibliography

[1] J.P. Abenstein and W.J. Tompkins. A new data-reduction algorithm for real-time ECG analysis. *Biomedical Engineering, IEEE Transactions on*, BME-29(1):43 –48, jan. 1982.

[2] V.X. Afonso, W.J. Tompkins, T.Q. Nguyen, and Shen L. ECG beat detection using filter banks. *Biomedical Engineering, IEEE Transactions on*, 46(2):192 –202, feb. 1999.

[3] M.L. Ahlstrom and W.J. Tompkins. Automated high-speed analysis of holter tapes with microcomputers. *Biomedical Engineering, IEEE Transactions on*, BME-30(10):651 –657, oct. 1983.

[4] C.A. Andrews, J.M. Davies, and G.R. Schwarz. Adaptive data compression. *Proceedings of the IEEE*, 55(3):267 – 277, march 1967.

[5] ANSI/AAMI-EC13. *American National Standards for Cardiac Monitors, Heart Rate Meters and Alarms.* Arlington, Va. : Association for the Advancement of Medical Instrumentation.

[6] Y. Barbotin. Finite Rate of Innovation sampling techniques for embedded UWB devices. 2009.

[7] Roger C. Barr, S.M. Blanchard, and Deborah A. Dipersio. Sapa-2 is the fan. *IEEE Transactions on Biomedical Engineering*, BME-32:337–337, 1985.

[8] S. Barro, M. Fernandez-Delgado, J.A. Vila-Sobrino, C.V. Regueiro, and E. Sanchez. Classifying multichannel ecg patterns with an adaptive neural network. *Engineering in Medicine and Biology Magazine, IEEE*, 17(1):45 –55, jan/feb 1998.

[9] C.M. Bishop. *Neural networks for pattern recognition.* Clarendon Press, 1995.

[10] T. Blu, P.-L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot. Sparse sampling of signal innovations. *Signal Processing Magazine, IEEE*, 25(2):31 –40, march 2008.

[11] P.O Borjesson, O. Pahlm, L. Sornmo, and M.-E. Nygards. Adaptive QRS detection based on maximum a posteriori estimation. *Biomedical Engineering, IEEE Transactions on*, BME-29(5):341 –351, may 1982.

[12] H.H. Bothe. *Neuro-Fuzzy-Methoden: Einführung in Theorie und Anwendungen.* Springer, 1998.

[13] J.A. Cadzow. Signal enhancement-a composite property mapping algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(1):49 –62, jan 1988.

[14] D.A. Coast, R.M. Stern, G.G. Cano, and S.A. Briller. An approach to cardiac arrhythmia analysis using hidden markov models. *Biomedical Engineering, IEEE Transactions on*, 37(9):826 –836, sept. 1990.

[15] J.R. Cox, F.M. Nolle, H.A. Fozzard, and G.C. Oliver. Aztec, a preprocessing program for real-time ECG rhythm analysis. *Biomedical Engineering, IEEE Transactions on*, BME-15(2):128 –129, april 1968.

[16] Jr. Cox, J.R., F.M. Nolle, and R.M. Arthur. Digital analysis of the electroencephalogram, the blood pressure wave, and the electrocardiogram. *Proceedings of the IEEE*, 60(10):1137 – 1164, oct. 1972.

[17] R. Crochiere, R. Cox, and J. Johnston. Real-time speech coding. *Communications, IEEE Transactions on*, 30(4):621 – 634, apr 1982.

[18] B. De Moor. Total least squares for affinely structured matrices and the noisy realization problem. *Signal Processing, IEEE Transactions on*, 42(11):3104 –3113, nov 1994.

[19] D. Dubin. *Rapid Interpretation of EKG's, Sixth Edition.* Cover Publishing Company, October 2000.

[20] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936.

[21] H. Fujiwara and T. Shimono. On the acceleration of test generation algorithms. *Computers, IEEE Transactions on*, C-32(12):1137 –1144, dec. 1983.

[22] B. Furht and A. Perez. An adaptive real-time ECG compression algorithm with variable threshold. *Biomedical Engineering, IEEE Transactions on*, 35(6):489 –494, june 1988.

[23] M. Gastpar and M. Vetterli. *To code or not to code.* PhD thesis, Lausanne, 2002. Prix EPFL de doctorats - EPFL doctorate award (2002).

[24] A.L. Goldberger. *Clinical electrocardiography: a simplified approach.* Mosby Elsevier, 2006.

[25] A.L. Goldberger et al. Physiobank, physiotoolkit, and physionet : Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[26] G.H. Golub and C.F.V. Loan. *Matrix computations.* Johns Hopkins studies in the mathematical sciences. Johns Hopkins University Press, 1996.

[27] Y. Hao, P. Marziliano, M. Vetterli, and T. Blu. Compression of ecg as a signal with finite rate of innovation. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 7564 –7567, jan. 2005.

[28] A. Hormati and M. Vetterli. *Sensing and Recovery under Sparsity Constraints: Theory and Applications.* PhD thesis, EPFL, 2010.

[29] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098 –1101, sept. 1952.

[30] M. Ishijima, S.-B. Shin, G.H. Hostetter, and J. Sklansky. Scan-along polygonal approximation for data compression of electrocardiograms. *Biomedical Engineering, IEEE Transactions on*, BME-30(11):723 –729, nov. 1983.

[31] S.M.S. Jalaleddine, C.G. Hutchens, R.D. Strattan, and W.A. Coberly. Ecg data compression techniques-a unified approach. *Biomedical Engineering, IEEE Transactions on*, 37(4):329 –343, april 1990.

[32] J.F. Kaiser. On a simple algorithm to calculate the ‘energy’ of a signal. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 381 –384 vol.1, apr 1990.

[33] K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Suomalaisen Tiedeakatemian toimituksia: Mathematica-physica. Universitat Helsinki, 1947.

[34] B.-U. Kohler, C. Hennig, and R. Orglmeister. The principles of software QRS detection. *Engineering in Medicine and Biology Magazine, IEEE*, 21(1):42 –57, jan.-feb. 2002.

[35] C.M. Kortman. Redundancy reduction - a practical method of data compression. *Proceedings of the IEEE*, 55(3):253 – 263, march 1967.

[36] V.A. Kotel'nikov. On the transmission capacity of 'ether' and wire in electric communications. *Izd. Red. Upr. Svyazzi, RKKA (Moscow)*, 1933.

[37] M. Kurchuk and Y. Tsividis. Signal-dependent variable-resolution clockless a/d conversion with application to continuous-time digital signal processing. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 57(5):982 –991, may 2010.

[38] A.A. Lazar and L.T. Toth. Time encoding and perfect recovery of bandlimited signals. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 6, pages VI – 709–12 vol.6, april 2003.

[39] A.A. Lazar and L.T. Toth. Perfect recovery and sensitivity analysis of time encoded bandlimited signals. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 51(10):2060 – 2073, oct. 2004.

[40] M. Loève. *Probability theory*. Number v. 2 in Graduate texts in mathematics. Springer-Verlag, 1978.

[41] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561 – 580, april 1975.

[42] J. Makhoul. Stable and efficient lattice methods for linear prediction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 25(5):423 – 428, oct 1977.

[43] P. Marziliano. *Sampling innovations*. PhD thesis, Lausanne, 2001.

[44] P. Marziliano, M. Vetterli, and T. Blu. Sampling and exact reconstruction of bandlimited signals with additive shot noise. *Information Theory, IEEE Transactions on*, 52(5):2230 – 2233, may 2006.

[45] P.E. McSharry, G.D. Clifford, L. Tarassenko, and L.A. Smith. A dynamical model for generating synthetic electrocardiogram signals. *Biomedical Engineering, IEEE Transactions on*, 50(3):289 –294, march 2003.

[46] G.B. Moody and R.G. Mark. The impact of the MIT-BIH arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45 –50, may-june 2001.

[47] W.C. Mueller. Arrhythmia detection program for an ambulatory ECG monitor. *Biomedical Sciences Instrumentation*, 14:81–85.

[48] H. Nyquist. Certain topics in telegraph transmission theory. *Trans. AIEE*, 47:617–644, Apr 1928.

[49] J. Pan and W.J. Tompkins. A real-time QRS detection algorithm. *Biomedical Engineering, IEEE Transactions on*, BME-32(3):230 –236, march 1985.

[50] K.C. Pohlmann. *Principles of digital audio*. McGraw-Hill, 2005.

[51] R.F. Quick, R.E. Crochiere, J.H. Hong, H. Hormati, and G. Baechler. Application of FRI to modelling electrocardiogram signals. *Submitted to EMBC*, 2012.

[52] Schafer R.W. Rabiner, L.R. *Digital Processing of Speech Signals*. (Prentice-Hall signal processing series). Pearson Education, 1979.

[53] Zhou S.-K., Wang J.-T., and Xu J.-R. The real-time detection of QRS-complex using the envelope of ECG. In *Engineering in Medicine and Biology Society, 1988. Proceedings of the Annual International Conference of the IEEE*, page 38 vol.1, nov. 1988.

[54] C.E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10 – 21, jan. 1949.

[55] L. Sörnmo and P. Laguna. *Bioelectrical signal processing in cardiac and neurological applications*. Academic Press, 2005.

[56] P. Stoica and R.L. Moses. *Spectral analysis of signals*. Pearson Prentice Hall, 2005.

[57] J. Szczupak, S. Mitra, and E. Jury. Some new results on discrete system stability. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 25(1):101 – 102, feb 1977.

[58] M. Vetterli, P. Marziliano, and T. Blu. Sampling signals with finite rate of innovation. *Signal Processing, IEEE Transactions on*, 50(6):1417 –1428, jun 2002.

[59] C. Vezyrtzis and Y. Tsividis. Processing of signals using level-crossing sampling. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 2293 –2296, may 2009.

[60] G. Walker. On periodicity in series of related terms. *Proceedings of The Royal Society of London. Series A, Containing Papers of A Mathematical and Physical Character (1905-1934)*, 131:518–532, 1931.

[61] J.M. Whittaker. The fourier theory of cardinal functions. 1929.

[62] G.U. Yule. On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers. *Royal Society of London Philosophical Transactions Series A*, 226:267–298, 1927.

[63] A. Zhang, L. Chai, and H. Dong. QRS complex detection of ECG signal by using teager energy operator. In *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, pages 2095 –2098, may 2008.