EÖTVÖS LORÁND UNIVERSITY
FACULTY OF INFORMATICS
DEPARTMENT OF NUMERICAL ANALYSIS

Transformation methods in signal processing

PHD THESIS

Péter KOVÁCS

Supervisor: Professor Sándor FRIDLI

Budapest, 2016

Contents

Preface

This dissertation summarizes the results of the author's research in signal processing. These results are presented in a unified approach which allows to show the connection between them. The main topic of the thesis is the development of adaptive models for ECG and EEG signals to perform efficient compression and classification. The design and construction of such systems involve several problems including optimization of free parameters, which is of great importance for providing a compact representation of the signal. In the thesis we propose a number of numerical algorithms based on different optimization methods. The mathematical background of our results and methods is the theory of rational functions. We show that the various systems of rational functions provide the basis for adaptive signal representations which turn to be better than the existing state-of-the-art methods in several respects.

In general, signal processing has become an important field of applied sciences over the last decades. Several mathematical results, especially orthogonal transforms were applied for solving practical problems in data compression, classification, detection, etc. In connection with this we note that the transforms by means of rational function systems in ECG signal processing were introduced by *Ferenc Schipp* and *Sándor Fridli* [FriLóc12]. The PhD thesis utilizes and extends their results to solve different problems, especially in biomedical signal processing. The main material of the dissertation was published in 4 journal and 5 conference papers, and presented at 10 international conferences. The conference papers appeared in IEEE proceedings. We note that 2 conference and 1 journal papers were written in collaboration with the research group in signal processing of Tampere University of Technology (TUT, Finland), where the author was a visiting researcher for a five-month period.

The thesis is organized as follows. Chapter 1 contains a brief introduction to the theory of rational function systems. Also, we provide a summary of the relevant previously existed results in adaptive signal modeling in order to clarify our contribution to the topics. For instance, we describe the main problems of ECG and EEG signal processing along with conventional testing techniques and databases.

In Chapter 2 we deal with the construction of pole optimization methods. Namely, we developed the hyperbolic variant of the so-called particle swarm optimization (PSO) algorithm, which was integrated into a more general multi-dimensional framework. By the means of this method one can calculate the best

pole configurations of the representation (i.e., the optimal positions and multiplicities). It also provides a novel technique for pole identification. Furthermore, we perform detailed stability and error analysis of the above methods.

In Chapter 3, we propose an electrocardiogram (ECG) signal generator based on spline interpolation. It turns to be an efficient tool for testing and evaluating signal models, filtering techniques, etc. In this study, the synthesized heartbeats were used to test the diagnostic distortion of the rational representation. At the end of this chapter, we introduce a mathematical model for the multi-channel QRS complex.

The main topic of Chapter 4 is the construction of a new generalized time-frequency distribution of EEG signals based on different types of rational functions, which is a joint work with *Kaveh Samiee* (TUT). We emphasize that the rational function systems were not used in this field before. Their application in epileptic seizure detection is the main outcome of our collaboration. Note that the hyperbolic PSO algorithm introduced in Chapter 2 was successfully applied in this problem, as well.

In Chapter 5 numerical methods for computing rational series expansions of discrete signals are presented. The collection of these algorithms can be found in the RAIT MATLAB toolbox together with usual signal processing techniques adapted to rational functions. We note that the MATLAB implementations within this chapter have been used in the applications shown in previous Chapters 2–4. Finally, we provide a comparative study of different signal compression methods.

We note that the Appendices are integrated parts of the dissertation. It means that complicated algorithms and MATLAB examples are excluded from the text of the thesis in order to make it more readable. Namely, pseudocodes of the main algorithms are moved to Appendix A while MATLAB examples can be found in Appendix B. The latter includes graphical user interfaces (GUI), as well, which demonstrates the use of rational functions and the ECG signal generator of Section 5.1 and Section 3.1, respectively.

Acknowledgment

I am very grateful to my supervisor Professor Sándor Fridli. He has been my mentor during the past five years and has helped me to implement my ideas, to present my papers as well as to write this thesis. He has been coordinating and supporting my scientific work, so I have been able to participate in several national and international conferences. Thanks to him, I have taken part in various projects which also contributed to enhance my knowledge.

I am greatly indebted to Professor Emeritus Ferenc Schipp, who also supported my research. I was impressed by his approach solving and modeling practical problems. In addition, he channeled my thoughts in the right direction towards the solution. His manner of lecturing also inspired my work.

I thank the Department of Numerical Analysis for initiating me as an assistant teacher of Eötvös L. University. I benefited a lot from the lecturers of the Department and I will try to return as much as possible to my students in the future.

I would like to thank Levente Lócsi and Kaveh Samiee for our fruitful joint work, and for their contribution to this thesis. I also thank to ELTE Bolyai College for accepting me as a member in 2010. The scientific spirit represented by the College has had a great impact on my career.

I am grateful to my family for supporting my doctoral studies. In particular, I thank to my father, whose endurance and diligence has inspired me in the past years.

Chapter 1

Introduction

In Section 1.1 we start with defining the general framework of the dissertation and discussing the most important terms in the signal processing jargon. Furthermore, we describe a signal model, conventional testing techniques and databases used in ECG and EEG signal processing. Then, in Section 1.2 we continue with Blaschke functions including their basic properties. The chapter is closed by Section 1.3, where we review the construction of orthogonal rational function systems called Malmquist–Takenaka systems, which will be followed by studying biorthogonal systems and the problem of non-uniform discretization.

1.1 Signal modeling

The analysis of physiological signals by means of mathematical transforms has proved to be an effective method in various aspects. For instance, compression and filtering methods are based on dimensionality reduction of the original signal. In addition, the coefficients of these transformations can also be used as features in classification tasks. Generally, these transform-domain based techniques are closely related to approximation theory in Hilbert spaces. Namely, $L^2_w(\mathbb{R})$ is usually considered to be the signal space with a positive weight function w. Then, the signal f is an element of $L^2_w(\mathbb{R})$, which is modeled via approximation from a closed subspace $H \subset L^2_w(\mathbb{R})$. It is well-known that the best approximation $\widetilde{f} \in H$ uniquely exists: $\|f - \widetilde{f}\|_{L^2_w} = \inf\{\|f - g\|_{L^2_w} \ : \ g \in H\}$ and $f - \widetilde{f} \perp H$. In practical applications H is a finite $N \in \mathbb{N}_+$ dimensional closed subspace, which implies that $\widetilde{f} = \sum_{k=0}^{N-1} c_k \Theta_k$, where $c_k \in \mathbb{R}$ and $\{\Theta_k : 0 \leq k < N\}$ is a *basis* of H. It can be easily shown that the coefficients c_k can be determined as the solution of the system of linear equations $\mathbf{G}c = \mathbf{b}$, where $\mathbf{G}_{i,j} = \langle \Theta_i, \Theta_j \rangle$ denotes the Gram matrix, $\mathbf{b}_i = \langle f, \Theta_i \rangle$, and $\langle .,. \rangle$ is the usual scalar product in $L^2_w(\mathbb{R})$. Although $\mathbf{G}$ has good numerical properties, e.g., it is symmetric and positive definite, the conditioning of the problem strongly depends on the base functions. For instance, one can get a pathological example in $L^2_w(0,1)$ by choosing $\Theta_k = x^k$ for which $\mathbf{G}_{i,j} = \frac{1}{i+j+1}$. In this case, the Gram matrix becomes the so-called Hilbert ma-

trix which is known to be highly ill-conditioned, namely, the condition number $\kappa_2(\mathbf{G}) = \mathcal{O}\left(\frac{(\sqrt{2}+1)^{4N}}{\sqrt{N}}\right)$ (see e.g., [HämHof91; SülMay03]). For a detailed analysis of the degree of ill-conditioning for Gram matrices of general inner-product spaces, we refer to [Tay78]. It is worth mentioning that the Gram matrix of basic rational functions is ill-conditioned as well (see e.g., the definition of $r_{a,k}$ in Section 1.3), which also explains the need of orthogonal and biorthogonal rational functions. If $\{\Theta_k : k \in \mathbb{N}\}$ is a set of *orthonormal* functions, the best approximation of f can be expressed as follows

$$(1.1) \qquad P_H^\Theta f(t) = \sum_{k=0}^{N-1} \langle f, \Theta_k \rangle \, \Theta_k(t)\,, \quad S^\Theta f(t) = \sum_{k=0}^{\infty} \langle f, \Theta_k \rangle \, \Theta_k(t)\,.$$

Here P_H is an orthogonal projection while $S^\Theta f$ is called the Fourier series, and $c_k = \langle f, \Theta_k \rangle$ is the Fourier coefficients of f. In the latter case, it is an important question whether the series $S^\Theta f$ converges. The answer is yes, since the Bessel's inequality holds in Hilbert spaces

$$(1.2) \qquad \sum_{k=0}^{\infty} |c_k|^2 \leq \|f\|_{L_w^2}^2\,.$$

Finally, if $\{\Theta_k : k \in \mathbb{N}\}$ is orthonormal and *complete* set of functions, Eq. (1.2) holds with equality, which is the so-called Parseval's equality. A detailed review of these results in Fourier analysis along with practical applications in signal processing can be found in [GasWit99].

Hereinafter, we will refer to discrete-time series as sequences or piecewise constant functions. Analog *signals* are considered as periodic or non-periodic functions defined on the real line. By the term *system* we refer to a finite or a countable set of functions. Furthermore, if the system depends on a vector of free parameters $\mathbf{a} \in \mathbb{C}^n$ ($n \in \mathbb{N}_+$), it is denoted by $\{\Theta_k^\mathbf{a} : k \in \mathbb{N}\}$, and it is called as an *adaptive* system. In this context *adaptivity* means the number of free parameters n.

In practice, a specific system is chosen for each problem. For instance, the most appropriate basis for audio signals is trigonometric functions (see e.g., MP3 encoding). Furthermore, discrete cosine transforms and wavelets are also preferred in case of compressing images (see e.g., JPEG and JPEG2000 standards). These techniques utilize a system of *fixed base* functions. The main disadvantage of this concept is the lack of adaptivity. Namely, if the signal f is not correlated with the shape of the base functions, the distance of f and the subspace H (i.e., the error of the representation) will be high. In order to rectify this problem, several methods have been introduced which extend the basic models with free parameters. For instance, Sörnmo et al. extended the classical Hermite orthogonal polynomials with free parameters via affine transform in [SörBör81]. Then, Coifman et al. [CoiMey92] increased the adaptivity of wavelets by using wavelet packets followed by Burrus et al., who highlighted the importance of adaptive wavelets in [BurGop97]. The general framework of these methods is similar to Eq. (1.1).

Namely, for a given vector of free parameters $\mathbf{a} := (a_0, a_1, \ldots, a_{n-1})$ and $n \in \mathbb{N}_+$ let us consider the orthonormal function system $\{\Theta_k^{\mathbf{a}} : k \in \mathbb{N}\}$. Then

$$(1.3) \qquad P_H^{\Theta, \mathbf{a}} f(t) = \sum_{k=0}^{N-1} \langle f, \Theta_k^{\mathbf{a}} \rangle \, \Theta_k^{\mathbf{a}}(t), \quad S^{\Theta, \mathbf{a}} f(t) = \sum_{k=0}^{\infty} \langle f, \Theta_k^{\mathbf{a}} \rangle \, \Theta_k^{\mathbf{a}}(t).$$

This way, the accuracy of the projection $P_H^{\Theta, \mathbf{a}}$ can be improved by minimizing the error of the approximation. Thus the optimal parameter $\mathbf{a}_o$ is defined by

$$(1.4) \qquad \mathbf{a}_o = \arg \min_{\mathbf{a}} \left\| f - P_H^{\Theta, \mathbf{a}} f \right\|_{L_w^2},$$

where $\arg \min_x f$ is the argument of the minimum of the function f. This can be found by using numerical optimization techniques such as gradient methods, simplex algorithms, Monte-Carlo simulations, etc. For instance, the system $\{\Theta_k^{\mathbf{a}} : k \in \mathbb{N}\}$ can be orthogonal polynomials [SörBör81], B-splines [KarMon97] and wavelets [BurGop97] while $\mathbf{a}$ is associated to the dilation parameters, base knots, etc.

Most of the previous algorithms are used in biomedical signal processing as well. However, we draw the reader's attention to the fact that medical principles are usually not taken into account. For instance, wavelet transforms are applied in ECG analysis [Add05], partly because they became very popular in many fields of signal processing. On the other hand, their ability to represent ECG signals is limited (see e.g., Section 5.3). In addition, in these methods the leads of the ECG records are treated as if they were independent from each other. Another approach for modeling ECG signals is based on the electrical field generated by heart. Then the electric field is interpreted as a vector-valued complex function

$$(1.5) \qquad f(z) = (f_1(z), f_2(z)) \qquad (z \in \overline{\mathbb{D}}),$$

where $f_1(z), f_2(z) \in \mathbb{R}$. In this dissertation, we will consider ECG as a time-dependent function $f(e^{it})$ ($t \in \mathbb{R}$). In this model, a heartbeat detected on a particular lead can be regarded as the projection of the electrical field f to a certain direction $e_\phi := (\cos \phi, \sin \phi)$:

$$E_\phi(t) := \langle f(e^{it}), e_\phi \rangle = f_1(e^{it}) \cos \phi + f_2(e^{it}) \sin \phi \qquad (t \in \mathbb{R}, \, \phi \in \mathbb{R}).$$

We note that this coincides with the conventional medical interpretation (see e.g., Chapter 1 in [CliAzu06; MorBra08]). Moreover, the model satisfies the so-called Einthoven's law, i.e., $\mathrm{II} = \mathrm{I} + \mathrm{III}$, where $\mathrm{I}, \mathrm{II}, \mathrm{III}$ denotes the limb leads. These signals measure the potential difference between the left and right arm, left leg and right arm, left leg and left arm, respectively. In our case, this can be written in the following form

$$E_{\phi_2}(t) := E_{\phi_1}(t) + E_{\phi_3}(t) \qquad (t \in \mathbb{R}),$$

where $\phi_1 = 0$, $\phi_2 = -\pi/3$, $\phi_3 = -2\pi/3$. This concept was introduced by Fridli and Schipp et al., and it was later used to model ECGs by rational functions [FriLóc12; Sch14].

One of the main scopes of this dissertation is to examine the performance of rational functions in different signal processing problems including compression and classification. For this reason, we utilize and extend the previously defined model. More precisely, we associate the free parameters $\mathbf{a}$ to the vector of inverse poles and the functions $\Theta_k^{\mathbf{a}}$ with the elements of the corresponding orthogonal and biorthogonal rational function systems. Thus, we get back the Nth partial sums of the Fourier series of f in Eq. (1.3). In addition, the c_k Fourier coefficients can be computed via continuous and discrete scalar products. In the former case we will use the trapezoidal rule to approximate integrals. Finally, we note that the error of the approximation is usually measured in $\|.\|_{L^2_w}$, which is conventional in ECG signal processing (see e.g., Section 3.2). This is why the theory of Hilbert spaces is so important in this field.

Signal processing applications

Although there is a wide range of applications of rational functions, we are focusing mainly on the synthesis, compression and classification of biomedical signals. For instance, the performance assessment of signal models is a difficult task, especially when the signal-to-noise ratio is high and the reference data is not available. This is when the synthetic signals come into picture. The main idea behind that is to generate signals with prescribed characteristics. Thus, the original signal is known, which can help us to validate a signal model in the presence of noise. The main problem here is to define a set of parameters (e.g., coefficients and base knots) which can guarantee valid signal morphology. In Section 3.1 we give a detailed description of this topic.

Additionally, let us consider the signal f with $M \in \mathbb{N}_+$ samples. Then for a given $N \leq M$ dimensional subspace we have to find the vector of optimal parameters $\mathbf{a}_o \in \mathbb{D}^n$ ($n \in \mathbb{N}_+$) by solving Eq. (1.4). Then, the original signal can be represented using only N number of coefficients instead of M samples. Now, the problem of compressing discrete-time signals can be reduced to the problem of interpolation and least square approximation for $N = M$ and $N \ll M$, respectively. In this sense, an algorithm is better than the others if it can provide less ℓ_2 error by using the same number of coefficients. A comparative study of rational functions along with other state-of-the-art methods can be found in Section 5.3.

The classification of biomedical signals is usually associated with a complex system which helps experts to analyze medical records visually. In general, preprocessing and feature extraction steps are performed, followed by (un)supervised data mining methods. For instance, the coefficients of a compact representation e.g., compression can be used to reduce the dimension of the original problem. This procedure results in a few parameters which can characterize the segments of physiological signals. Then the so-called feature vectors can be used to train

a classifier. Performance of these algorithms are evaluated through the following statistical measures:

$$\text{Sens} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Spec} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad \text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

where TP, TN, FP, FN denote the number of true positive/negative and false positive/negative hits of the test. For binary classification tasks, *Sensitivity/Specificity* is the rate of correctly classified events/nonevents among all events/nonevents while *Accuracy* is the ratio of the number of correctly classified patterns to the total number of patterns. In other words, *Sensitivity/Specificity* gives the probability of correctness for the positive/negative tests. Our goal is to find the optimal parameters of the classifier in order to maximize these statistical measures. To this end, in Chapter 4 we provide an EEG seizure detection algorithm based on different rational representations.

Experimental data

PhysioNet database

In Chapter 3 and Section 5.3, we model and compress the ECG signals. In order to validate the model and to examine the diagnostic distortion of the reconstruction, we use real ECG records of the PhysioNet database [GolAma00]. It is a collection of various biomedical signals including photoplethysmogram (PPG), blood pressure (BP), electromyogram (EMG), respiration (RESP), ECG, EEG, etc. In our experiments, we used the MIT-BIH Arrhythmia Database. It contains 48 half-hour long ECG recordings, obtained from 47 subjects. Each record was digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. Additionally, two or more cardiologists independently annotated each record. In our experiments, we used these annotations in order to perform beat-by-beat processing of the signals.

Bonn database

In Chapter 4, we construct an EEG seizure detection algorithm based on rational functions. In order to test our method, we evaluated several experiments on real EEG signals. The database has been provided by the University of Bonn and it is freely available online [AndLeh01]. This database has been widely used for EEG feature extraction and classification in the literature [TzaTsi12]. The EEG database consists of five sets (A-E). Each set contains 100 single-channel EEG segments, each with a duration of 23.6 seconds. Sets A and B have been recorded using the standard international 10-20 system for surface EEG recording. Five healthy volunteers participated in these tests with eyes open (A) and eyes closed (B). For sets C, D and E five epileptic patients were selected for presurgical evaluation of epilepsy by using intracranial electrodes. Depth electrodes were implanted

symmetrically to record EEG from the epileptogenic zone (D) and from hippocampal formation of the opposite hemisphere of the brain (C). Segments of set E were taken from contacts of all electrodes. In sets C and D, segments contain interictal intervals while seizure activities occur in set E. Each epoch was sampled at 173.61 Hz resulting in a total of 4096 samples.

1.2 Blaschke functions

Blaschke functions are special types of Möbius transformations. They have several applications in signal processing, such as representing the transfer functions of filters in system identification [SouPap02; SouSch11b; SouBok13; Sch14] or constructing general orthogonal basis [HeuVan05; BulGon09]. Blaschke functions play an important role in our work as well. Namely, they are required to implement the hyperbolic operations in Chapter 2 and to solve the discretization problem in Section 5.2 via the so-called argument function.

Möbius transformations

Let us denote the set of complex numbers by $\mathbb{C}$, the open unit disc by $\mathbb{D} := \{z \in \mathbb{C} : |z| < 1\}$, and the unit circle (or torus) by $\mathbb{T} := \{z \in \mathbb{C} : |z| = 1\}$. Furthermore, let $\mathbb{N}_+$ stand for the set of non-zero natural numbers, and $\mathbf{GL(2)} := \{\mathbf{M} \in \mathbb{C}^{2\times 2} : \det(\mathbf{M}) \neq 0\}$ for the set of non-singular matrices of order two. Then, to every

$$\mathbf{GL(2)} \ni \mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

we associate the function

$$t_{\mathbf{M}} : z \to \frac{az + b}{cz + d} \qquad (z \in \mathbb{C}).$$

The set of functions $\mathfrak{M} := \{t_{\mathbf{M}} : \mathbf{M} \in \mathbf{GL(2)}\}$ is the set of the so-called Möbius transformations. It is defined at any complex point $z \in \overline{\mathbb{C}}$ if we use the following extensions

$$t_{\mathbf{M}}\left(-\frac{d}{c}\right) = \infty, \qquad t_{\mathbf{M}}(\infty) = \begin{cases} \dfrac{a}{c} & \text{if } c \neq 0 \\ \infty & \text{if } c = 0 \end{cases}.$$

It can be shown that for any $\mathbf{M} \in \mathbf{GL(2)}$, the Möbius transformation $t_{\mathbf{M}}$ is a bijective conformal map of $\overline{\mathbb{C}}$ onto itself. Additionally, the inverse transformation can be defined by inverting the matrix $\mathbf{M}$. Then the inverse function $t_{\mathbf{M}}^{-1} = t_{\mathbf{M}^{-1}}$ is also a Möbius transformation. Furthermore, the functions $t_{\mathbf{M}}$ form a group under the operation of composition. Moreover, if $\mathbf{M}, \mathbf{N} \in \mathbf{GL(2)}$ then $t_{\mathbf{M}} \circ t_{\mathbf{N}} = t_{\mathbf{MN}}$ (see e.g., theorem 5.2c in [Hen74]). Finally, we recall the standard form of Möbius transformations which can be obtained by applying Schur decomposition

on $\mathbf{M} \in \mathbf{GL(2)}$. Namely, $\mathbf{M} = \mathbf{UTU}^*$ where $\mathbf{U}, \mathbf{T} \in \mathbf{GL(2)}$ are unitary and upper triangular matrices, respectively. Thus, any Möbius transformation may be built as

$$(1.6) \qquad t_{\mathbf{M}} = t_{\mathbf{U}} \circ t_{\mathbf{T}} \circ t_{\mathbf{U}^*}.$$

Blaschke transformations

The Blaschke functions can be derived from the set of Möbius transformations. More precisely, let us consider the subset $\mathbf{SH(2)} \subset \mathbf{GL(2)}$ as follows

$$\mathbf{B} = \begin{bmatrix} p & -q \\ -\overline{q} & \overline{p} \end{bmatrix}, \qquad \det(\mathbf{B}) = |p|^2 - |q|^2 = 1 \qquad (p, q, \in \mathbb{C}).$$

Then $\mathbf{B} \to t_{\mathbf{B}}$ defines a homomorphism and maps the matrices $\mathbf{B} \in \mathbf{SH(2)}$ to Blaschke functions

$$t_{\mathbf{B}}(z) := \frac{pz - q}{\overline{p} - \overline{q}z} = \frac{p}{\overline{p}} \frac{z - q/p}{1 - z\overline{q}/\overline{p}} = \epsilon \frac{z - a}{1 - \overline{a}z} =: B_{\mathfrak{a}}(z) \quad (z \in \mathbb{C}),$$

where $a := q/p \in \mathbb{D}$ and $\epsilon := p/\overline{p} \in \mathbb{T}$. Let us denote the set of Blaschke functions by $\mathfrak{B} := \{ B_{\mathfrak{a}} : \mathfrak{a} = (a, \epsilon) \in \mathbb{B} \}$ where $\mathbb{B} := \mathbb{D} \times \mathbb{T}$. Then $\mathfrak{B}$ is a subgroup of $\mathfrak{M}$ under the composition operator. Indeed, if we take the functions $B_{\mathfrak{a}_j}$ for $j = 1, 2$ then $B_{\mathfrak{a}_1} \circ B_{\mathfrak{a}_2} = B_{\mathfrak{a}} \in \mathfrak{B}$ and the parameters of the composition $\mathfrak{a} = (a, \epsilon)$ can be expressed as follows

$$(1.7) \qquad a = \frac{a_1 \overline{\epsilon}_2 + a_2}{1 + a_1 \overline{a}_2 \overline{\epsilon}_2}, \qquad \epsilon = \epsilon_1 \epsilon_2 \frac{1 + a_1 \overline{a}_2 \overline{\epsilon}_2}{1 + \overline{a}_1 a_2 \epsilon_2}.$$

Moreover, $B_{\mathfrak{e}}$ is the unit element of $\mathfrak{B}$ with $\mathfrak{e} := (0, 1)$ while the inverse transformation can be expressed as $B_{\mathfrak{a}}^{-1} = B_{\mathfrak{a}^{-1}}$, where $\mathfrak{a}^{-1} := (-a\epsilon, \overline{\epsilon})$. One can classify the elements of $\mathfrak{B}$ by considering the bijections of $\mathbb{D}$ and $\mathbb{T}$. In this sense, two other subgroups can be derived from $\mathfrak{B}$:

$$\mathfrak{B}_{\mathbb{I}} := \{ B_{\mathfrak{a}} : \mathfrak{a} = (r, 1), \, r \in (-1, 1) \}, \qquad \mathfrak{B}_{\mathbb{T}} := \{ B_{\mathfrak{a}} : \mathfrak{a} = (0, \epsilon), \, \epsilon \in \mathbb{T} \}.$$

Now, similarly to Eq. (1.6), one can use these subgroups as the building blocks of the Blaschke group $\mathfrak{B} = \mathfrak{B}_{\mathbb{T}} \circ \mathfrak{B}_{\mathbb{I}} \circ \mathfrak{B}_{\mathbb{T}}$. More precisely,

$$(1.8) \qquad B_{\mathfrak{a}} = B_{(0, e^{i(\alpha + \vartheta)})} \circ B_{(r, 1)} \circ B_{(0, e^{-i\alpha})} \quad (\mathfrak{a} := (re^{i\alpha}, e^{i\vartheta}) \in \mathbb{B}).$$

Finally, the parameter space $\mathbb{B}$ of the Blaschke functions will be described. To this end, let us consider the following two subsets of $\mathbb{B}$:

$$\mathbb{B}_1 := \{ (r, 1) : r \in (-1, 1) \}, \qquad \mathbb{B}_2 := \{ (0, \epsilon) : \epsilon \in \mathbb{T} \}.$$

It can be shown that $(\mathbb{B}_1, \circ)$ and $(\mathbb{B}_2, \circ)$ form subgroups of $\mathbb{B}$. Moreover, Eq. (1.7) implies that

$$(r_1, 1) \circ (r_2, 1) = (r, 1), \qquad r = \frac{r_1 + r_2}{1 + r_1 r_2} \qquad (r_1, r_2 \in (-1, 1)),$$

$$(0, \epsilon_1) \circ (0, \epsilon_2) = (0, \epsilon), \qquad \epsilon = \epsilon_1 \epsilon_2 \qquad (\epsilon_1, \epsilon_2 \in \mathbb{T}).$$

One can see that $(\mathbb{B}_2, \circ)$ and $(\mathbb{T}, \cdot)$ are isomorphic groups. Additionally, one can define the isomorphism between $(\mathbb{B}_1, \circ)$ and $(\mathbb{R}, +)$ by using the map

$$(1.9) \qquad \text{th}(s_1 + s_2) = \frac{\text{th}\, s_1 + \text{th}\, s_2}{1 + \text{th}\, s_1 \, \text{th}\, s_2} = \frac{r_1 + r_2}{1 + r_1 r_2} = r = r_1 \circ r_2,$$

where $r_j = \text{th}\, s_j$, $(s_j \in \mathbb{R}, j = 1, 2)$. Now, the analogue of Eq. (1.8) can be written as

$$\mathfrak{a} = \mathfrak{e}_1 \circ \mathfrak{b} \circ \mathfrak{e}_2,$$

where $\mathfrak{a} \in \mathbb{B}$, $\mathfrak{e}_1, \mathfrak{e}_2 \in \mathbb{B}_2$ and $\mathfrak{b} \in \mathbb{B}_1$. Note that in Chapter 2, we will use these identities in order to construct a non-deterministic hyperbolic inverse pole optimization technique. A further description of the Blaschke transformation and its relation to the hyperbolic geometry can be found in [Sch14].

Argument function

The Blaschke transformation $B_{\mathfrak{a}} : \mathbb{T} \to \mathbb{T}$ $(\mathfrak{a} \in \mathbb{B})$ is a bijective mapping of $\mathbb{T}$ onto itself. It means that

$$B_{\mathfrak{a}}(e^{it}) = e^{i\beta_{\mathfrak{a}}(t)} \quad (t \in \mathbb{R}),$$

where $\beta_{\mathfrak{a}} : \mathbb{R} \to \mathbb{R}$ is the so-called argument function. In order to give the explicit form of $\beta_{\mathfrak{a}}$, let us consider the following functions:

$$\gamma_s(t) := 2 \arctan(s \tan(t/2)) \qquad (t \in (-\pi, \pi),\ s \in (0, \infty)).$$

In addition, one can extend γ_s to $\mathbb{R}$ for any $s \in (0, \infty)$ as

$$\gamma_s(-\pi) := -\pi, \quad \gamma_s(\pi) := \pi, \quad \gamma_s(t + 2\pi) := \gamma_s(t) + 2\pi \qquad (t \in \mathbb{R}).$$

It can be shown that γ_s is a monotonic, differentiable function and

$$(1.10) \qquad \gamma_s'(t) := \frac{1 - r^2}{1 - 2r \cos t + r^2} := P_r(t) \quad (t \in \mathbb{R}),$$

where P_r is the so-called Poisson kernel and

$$s = s(r) := \frac{1 + r}{1 - r}, \quad r \in [0, 1).$$

One can also calculate the explicit form of the argument function by simple algebraic manipulations. Namely, let us consider the Blaschke function $B_{\mathfrak{a}} \in \mathfrak{B}$

with the parameters $\mathfrak{a} = (a, \epsilon) \in \mathbb{B}$ where $a = re^{i\alpha}$ and $\epsilon = e^{i\vartheta}$ with $\vartheta, \alpha \in [-\pi, \pi)$, $r \in [0, 1)$. Then the corresponding argument function can be defined as follows

(1.11)
$$\beta_{\mathfrak{a}}(t) := \vartheta + \alpha + \gamma_s(t - \alpha) \quad (t \in \mathbb{R}).$$

Note that the optimization problems considered in this study operate with r and α only, while ϑ is an arbitrary parameter of the argument function. However, for discretization problems, we had better choose ϑ in such a way that $\beta_{\mathfrak{a}}(-\pi) = -\pi$. In this case, we will use the simplified notation β_a, which realizes a mapping $\beta_a : [-\pi, \pi) \rightarrow [-\pi, \pi)$. Additionally, the inverse of this function β_a^{-1} can be used to construct discrete orthogonal rational systems (see e.g., Section 1.3). As a consequence, a non-uniform sampling scheme of discrete-time series can be defined, which was introduced by Schipp and Soumelidis et al. [SouSch02]. Based on their paper, we give a numerical algorithm to determine the related discretization points in Section 5.2.

1.3   Rational functions

In this section, we give a brief introduction to the theory of rational functions. This topic establish a good foundation for signal modeling in Chapters 3–4 and for numerical algorithms in Chapter 5.

Approximation theory and Fourier analysis give a basis for several applications in signal processing. Namely, one can reduce the redundancy of a signal by decomposing it into independent components. Then, only the coefficients of the representation need to be stored and can later be used for classification or detection purposes. Orthogonal function systems such as trigonometric, Walsh [SchWad90], wavelets [Dau92a] are preferred due to their low computational costs. Classical orthogonal polynomials [Sze67] are also widely used in this field. These kinds of polynomials can be regarded as a special type of rational functions whose poles are all fixed at infinity. However, if the poles are taken in the extended complex plane, we will eventually arrive at the theory of orthogonal rational functions [BulGon99]. These generalized orthogonal bases (GOB) have a wide variety of signal processing applications [HeuVan05]. During the last ten years, the theory of rational function systems have been further extended by Adhemar Bultheel et al. [BulGon09], Ferenc Schipp [FriGil13a; Sch14], Sándor Fridli [FriSch11; FriLóc12; FriGil13b], Margit Papp [PapSch01; PapSch04] and Alexandros Soumelidis [SouSch02]. Although there is a wide range of applications of these systems, we are focusing on the main theorems only, which provide the building blocks of our work.

Malmquist–Takenaka system

We recall that the Hardy space $H^2(\mathbb{D})$ is the collection of analytic functions $f : \mathbb{D} \to \mathbb{C}$ which are square integrable on $\mathbb{D}$:

$$\|f\|_{H^2} := \sup_{0 \le r < 1} \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} \left| f(re^{it}) \right|^2 dt \right)^{\frac{1}{2}} < \infty \, .$$

The radial limit function $f(e^{it}) := \lim_{r \to 1-0} f(re^{it})$ also exists, which belongs to $L^2(\mathbb{T})$, and $\|f\|_{H^2} = \|f\|_{L^2(\mathbb{T})}$ (see e.g., [Zyg59; Mór13]). Therefore one can define a scalar product in $H^2(\mathbb{D})$ by

$$(1.12) \qquad \langle f, g \rangle := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(e^{it}) \overline{g}(e^{it}) \, dt \qquad (f, g \in H^2(\mathbb{D})) \, .$$

The scalar product induces the norm $\|.\|_{H^2}$ on $\mathbb{T}$. Furthermore, $H^2(\mathbb{D})$ is complete with respect to the norm $\|f\|_{H^2} = \sqrt{\langle f, f \rangle}$, i.e., in other words $H^2(\mathbb{D})$ is a Hilbert space. In this sense, the best approximation exists for all the closed subspaces of $H^2(\mathbb{D})$. In particular, if we have an orthogonal basis of $H^2(\mathbb{D})$, the closest element of any subspaces can be computed via orthogonal projections. In order to achieve this, let us first consider the set of rational functions $\mathfrak{R}$ that are analytic on $\overline{\mathbb{D}}$. This is a normed subspace of $H^2(\mathbb{D})$. If $\mathfrak{P}$ denotes the set of polynomials, and $\mathfrak{R}_0 = \{ r \in \mathfrak{R} : \lim_{z \to \infty} r(z) = 0 \}$ is the set of proper rational functions in $\mathfrak{R}$, then $\mathfrak{R}$ can be decomposed as

$$\mathfrak{R} = \operatorname{span} \{ \mathfrak{P} \cup \mathfrak{R}_0 \} \, .$$

It can be shown by partial fraction decomposition that $\mathfrak{R}_0$ is spanned by the system of basic rational functions (RF):

$$r_{a,k}(z) := \frac{1}{(1 - \overline{a}z)^k} \qquad (a \in \mathbb{D} \setminus \{0\}, \, z \in \overline{\mathbb{D}}, \, k \in \mathbb{N}_+) \, .$$

Parameter a is referred to as the *inverse pole* (because $1/\overline{a}$ is the pole of $r_{a,k}$ in the usual sense), while k is said to be the *order* of the basic function. Using a terminology similar to the trigonometric case, the value $k = 1$ corresponds to the *fundamental tone* and $k > 1$ to the *overtones*. It can be shown that $\mathfrak{R}_0 = \operatorname{span} \{ r_{a,k} : a \in \mathbb{D} \setminus \{0\}, \, k \in \mathbb{N}_+ \}$. In other words, any function $f \in \mathfrak{R}_0$ can be written as

$$(1.13) \qquad f = \sum_{j=0}^{n-1} c_j r_{a_j, k_j} \, ,$$

with appropriate inverse poles $a_j \in \mathbb{D} \setminus \{0\}$, multiplicities $k_j \in \mathbb{N}_+$ and complex coefficients $c_j \in \mathbb{C}$ $(0 \le j < n, \, n \in \mathbb{N}_+)$. In order to construct orthogonal and biorthogonal rational functions, one can introduce the modified basic rational functions (MRF):

$$\varphi_{a,k}(z) = \frac{z^{k-1}}{(1 - \overline{a}z)^k} \qquad (a \in \mathbb{D}, \, z \in \overline{\mathbb{D}}, \, k \in \mathbb{N}_+) \, .$$

It can be shown that the functions $r_{a,k}$ and $\varphi_{a,k}$ span the same subspaces of $\mathfrak{R}$ for $a \neq 0$, $k \in \mathbb{N}_+$. Now, let us consider the sequence of inverse poles and multiplicities:

$$(1.14) \qquad \mathbf{b} := (b_k \in \mathbb{D},\, k \in \mathbb{N}), \qquad \boldsymbol{\nu} := (\nu_k \in \mathbb{N}_+,\, k \in \mathbb{N}),$$

where ν_k counts the occurrences of b_k in the segment $b_0, \ldots, b_k$. Then, for a given $n \in \mathbb{N}_+$ the subspaces of $\mathfrak{R}$ generated by $\mathbf{b}$ can be defined as follows:

$$\mathfrak{R}_n^{\mathbf{b}} := \mathrm{span}\left\{ \varphi_{b_k, \nu_k}^{\mathbf{b}} : 0 \leq k < n \right\}, \qquad \mathfrak{R}^{\mathbf{b}} := \bigcup_{n=0}^{\infty} \mathfrak{R}_n^{\mathbf{b}} \subset \mathfrak{R},$$

Note that the sequence $\boldsymbol{\nu}$ and hence the subspaces $\mathfrak{R}_n^{\mathbf{b}}$ depend on the order of the inverse poles in $\mathbf{b}$. In addition, $\mathfrak{R}^{\mathbf{b}}$ is everywhere dense in the Hardy space $H^2(\mathbb{D})$, if and only if the so-called Blaschke condition is satisfied [HeuVan05]:

$$(1.15) \qquad \sum_{k=0}^{\infty} (1 - |b_k|) = \infty.$$

In this case, the system $\left\{ \varphi_{b_k, \nu_k}^{\mathbf{b}} : k \in \mathbb{N} \right\}$ is closed in $H^2(\mathbb{D})$, but it does not form an orthogonal set. Hence, it is difficult to compute the c_k coefficients in Eq. (1.13). On the other hand, we can easily solve this problem by applying Gram–Schmidt orthogonalization to the modified rational functions. The corresponding rational function system is the so-called Malmquist–Takenaka (MT) system. A handy property of the MT system is that the elements can be explicitly expressed by Blaschke products. Taking the sequence of inverse poles $\mathbf{b} = (b_k \in \mathbb{D},\, k \in \mathbb{N})$, the MT system can be written as:

$$\Phi_k^{\mathbf{b}}(z) = \frac{\sqrt{1 - |b_k|^2}}{1 - \overline{b_k} z} \prod_{j=0}^{k-1} B_{b_j}(z) \qquad (z \in \overline{\mathbb{D}},\, k \in \mathbb{N}).$$

Here, we used the simplified notation of the Blaschke function $B_{\mathfrak{b}}$ for $\mathfrak{b} = (b, 1)$ as follows

$$(1.16) \qquad B_b(z) := \frac{z - b}{1 - \overline{b} z} \qquad (z \in \overline{\mathbb{D}}).$$

If Eq. (1.15) holds, the MT system is a generalized orthogonal basis in $H^2(\mathbb{D})$. Indeed, one can have the trigonometric system on $\mathbb{T}$ by assuming $b_i = 0$ for all inverse poles, the Laguerre system if all the inverse poles are equal and real, while the Kautz system can be derived by periodically repeating a complex conjugated pair of inverse poles.

Finally, we note that the MT system is orthonormal with respect to the scalar product in Eq. (1.12) which is defined on the torus $\mathbb{T}$. Thus, in the dissertation,

we restrict the functions $f \in H^2(\mathbb{D})$ to the unit circle (cf. Eq. (1.5)) by using the following map

$$(1.17) \qquad [-\pi, \pi) \ni t \to \mathrm{Re}\left(f(e^{it})\right), \mathrm{Im}\left(f(e^{it})\right).$$

In this sense, f becomes a function of real variable $t \in [-\pi, \pi)$, which can be interpreted as the time-domain. It is also worth mentioning that there exist rational orthogonal systems on the plane with respect to the area measure on $\mathbb{D}$. The detailed analysis of the construction of these systems can be found in [FriGil13b].

Biorthogonal rational functions

By orthogonalization we obtain an orthonormal set of functions, but the time localization property of the basic rational form has been lost. Fortunately, biorthogonal rational functions (BRF) cure this problem. Let us consider a vector of inverse poles $\mathbf{b} \in \mathbb{D}^N$ ($N \in \mathbb{N}_+$). Suppose that $\mathbf{b}$ has n different entries which are

$$(1.18) \qquad \mathbf{a} := (a_0, \ldots, a_{n-1}) \in \mathbb{D}^n.$$

Set $\mathbf{m} := (m_0, \ldots, m_{n-1}) \in \mathbb{N}_+^n$, where m_k denotes the number of occurrences of a_k in the vector $\mathbf{b}$ (cf., Eq. (1.14)). Note that $N = m_0 + m_1 + \cdots + m_{n-1}$. We will take the system of modified rational functions $\{\varphi_{ki}^{\mathbf{a}} : k \in \mathbb{N}, 1 \leq i \leq m_k\}$, where we used the simplified notations $\varphi_{ki}^{\mathbf{a}} := \varphi_{a_k,i}^{\mathbf{a}}$. Then the corresponding biorthogonal systems can be defined by using the functions below:

$$\Omega_{\ell n-1}(z) := \frac{1}{(1 - \overline{a}_\ell z)^{m_\ell}} \prod_{i=0, i \neq \ell}^{n-1} B_{a_i}^{m_i}(z),$$

$$\omega_{\ell n-1}(z) := \frac{\Omega_{\ell n-1}(a_\ell)}{\Omega_{\ell n-1}(z)} \qquad (0 \leq \ell < n).$$

By Theorem 1 in [FriSch11] the following rational functions

$$\Psi_{\ell j}^{\mathbf{a}}(z) := \frac{\Omega_{\ell n-1}(z)(z - a_\ell)^{j-1}}{\Omega_{\ell n-1}(a_\ell)} \sum_{s=0}^{m_\ell - j} \frac{\omega_{\ell n-1}^{(s)}(a_\ell)}{s!} (z - a_\ell)^s,$$

$(0 \leq \ell < n, 1 \leq j \leq m_\ell)$ are biorthogonal to $\varphi_{ki}^{\mathbf{a}}$ with respect to the scalar product in Eq. (1.12). More precisely,

$$\left\langle \Psi_{\ell j}^{\mathbf{a}}, \varphi_{ki}^{\mathbf{a}} \right\rangle = \delta_{ij}\delta_{k\ell} \qquad (1 \leq i \leq m_k, 1 \leq j \leq m_\ell, 0 \leq k, \ell < n),$$

where $\delta_{\ell k}$ is the Kronecker delta symbol. Note that the previously defined rational function systems are complete in $H^2(\mathbb{D})$, if and only if Eq. (1.15) holds. Then a function $f \in H^2(\mathbb{D})$ can be approximated by the partial sums of its rational Fourier series. Namely, for a given vector of different inverse poles $\mathbf{a} \in \mathbb{D}^n$ and multiplicities $\mathbf{m} \in \mathbb{N}_+^n$ we have

$$(1.19) \qquad S_N^{\Psi, \mathbf{a}} f := \sum_{k=0}^{n-1} \sum_{i=1}^{m_k} \left\langle f, \Psi_{ki}^{\mathbf{a}} \right\rangle \varphi_{ki}^{\mathbf{a}}.$$

Hereinafter, for a vector of different inverse poles $\mathbf{a} \in \mathbb{D}^n$ with multiplicities $\mathbf{m}$ we will consider the MT system that corresponds to the vector

(1.20) $$\mathbf{b} := (\underbrace{a_0, \ldots, a_0}_{m_0}, \ldots, \underbrace{a_{n-1}, \ldots, a_{n-1}}_{m_{n-1}}) \in \mathbb{D}^N.$$

The corresponding MT expansion is as follows

(1.21) $$S_N^{\Phi,\mathbf{b}} f := \sum_{k=0}^{N-1} \left\langle f, \Phi_k^{\mathbf{b}} \right\rangle \Phi_k^{\mathbf{b}}.$$

Now, let us consider the N dimensional subspaces of $\mathfrak{R}$:

$$\mathfrak{R}_N^{\mathbf{b}} := \operatorname{span} \left\{ \Phi_k^{\mathbf{b}} : 0 \le k < N \right\} = \mathfrak{R}_N^{\mathbf{a}} := \operatorname{span}\{ \Psi_{ki}^{\mathbf{a}} : 0 \le k < n, \, 1 \le i \le m_k \}.$$

Then, the operators $S_N^{\Phi,\mathbf{b}} f : H^2(\mathbb{D}) \to \mathfrak{R}_N^{\mathbf{b}}$ and $S_N^{\Psi,\mathbf{a}} f : H^2(\mathbb{D}) \to \mathfrak{R}_N^{\mathbf{a}}$ have the following properties:

  i) $S_N^{\Phi,\mathbf{b}}$ and $S_N^{\Psi,\mathbf{a}}$ are projections to $\mathfrak{R}_N^{\mathbf{b}}$ and $\mathfrak{R}_N^{\mathbf{a}}$,

  ii) $\left\langle f - S_N^{\Phi,\mathbf{b}} f, g \right\rangle = \left\langle f - S_N^{\Psi,\mathbf{a}} f, h \right\rangle = 0 \quad (g \in \mathfrak{R}_N^{\mathbf{b}}, \, h \in \mathfrak{R}_N^{\mathbf{a}})$,

  iii) $S_N^{\Phi,\mathbf{b}} f$ and $S_N^{\Psi,\mathbf{a}} f$ are the best approximations in the subspaces $\mathfrak{R}_N^{\mathbf{b}}$ and $\mathfrak{R}_N^{\mathbf{a}}$.

It is clear that the vector of inverse poles $\mathbf{b}$ in Eq. (1.20) and thus the corresponding MT and BRF systems are defined if the vector of different inverse poles $\mathbf{a}$ and multiplicities $\mathbf{m}$ are given in Eq. (1.18). If there is no ambiguity, we will use the simplified notations S_N^{Φ}, S_N^{Ψ} and Φ_k, $\Psi_{k\ell}$.

Real-valued rational functions

It is known that the complex trigonometric system $\left\{ e^{int} : t \in \mathbb{R}, n \in \mathbb{Z} \right\}$ is a complete orthogonal set in $L^2(\mathbb{T})$. In addition, the statement also holds for the real $\cos(nt)$ and imaginary parts $\sin(nt)$ of this system. Following this analogy, we can have the same properties for rational functions as well. Namely, we restrict the elements of the RF system to $\mathbb{T}$ in the same way as in Eq. (1.17) by

$$[-\pi, \pi) \ni t \to \operatorname{Re}\left(r_{a,k}(e^{it}) \right), \operatorname{Im}\left(r_{a,k}(e^{it}) \right).$$

We will use these real valued rational functions in Section 3.3 to model the QRS complex of an ECG signal. In order to construct real valued orthogonal rational functions, let us consider a sequence of inverse poles $\mathbf{a} = (a_k \in \mathbb{D}, k \in \mathbb{N})$. Then the MT system can be extended as

$$\Phi_{-k}(z) := \overline{\Phi_k(z)} \qquad (k \in \mathbb{N}, z \in \mathbb{T}).$$

If $a_0 = 0$ and Eq. (1.15) holds, the extended MT system remains orthogonal and complete in $L^2(\mathbb{T})$:

(1.22) $$\langle \Phi_k, \Phi_\ell \rangle = \delta_{k,\ell} \qquad (k, \ell \in \mathbb{Z}).$$

Now, using the same constraints, the real valued MT system can be defined as follows:

$$U_0 := 1, \ V_0 := 0, \ U_k := \mathrm{Re}(\Phi_k), \ V_k := \mathrm{Im}(\Phi_k), \qquad (k \in \mathbb{N}_+).$$

Note that the system remained orthogonal, but $\langle U_k, U_k \rangle = \langle V_k, V_k \rangle = 1/2$ $(k \in \mathbb{N}_+)$. Then, for an $f \in H^2(\mathbb{T})$ the real valued partial sums of the MT–Fourier series can be written in the form

$$\mathrm{Re}\, S_n^\Phi f = \langle \mathrm{Re}\, f, U_0 \rangle\, U_0 + 2 \sum_{k=1}^{n-1} \left(\langle \mathrm{Re}\, f, U_k \rangle\, U_k + \langle \mathrm{Re}\, f, V_k \rangle\, V_k \right),$$

(1.23)

$$\mathrm{Im}\, S_n^\Phi f = \langle \mathrm{Im}\, f, U_0 \rangle\, U_0 + 2 \sum_{k=1}^{n-1} \left(\langle \mathrm{Im}\, f, U_k \rangle\, U_k + \langle \mathrm{Im}\, f, V_k \rangle\, V_k \right).$$

These types of the rational representations can be useful, especially when $f \in L^2(\mathbb{T})$ is a real valued function, e.g., f is an analog signal. In this case f has non-zero Fourier coefficients with negative indices, so it cannot be interpreted as the elements of $H^2(\mathbb{T})$. Nevertheless, we can handle this issue by using the analytic representation of real signals; that is to say we should compute the function $F = f + i\mathcal{H}f$, where $\mathcal{H}$ denotes the well-known Hilbert transformation. Then, Eq. (1.23) remains valid for $F \in H^2(\mathbb{T})$ and the approximation of f is equal to $\mathrm{Re}\, S_n^\Phi F$. The detailed implementation of the complex and real valued MT system will be further discussed in Section 5.1.

Discretization problem

Since our research interests include discrete time series, it would be evident to use discrete orthogonal function systems as well. In this sense, we obtain an interpolation procedure, i.e., the perfect reconstruction of the signal is possible at these points. For instance, in case of the trigonometric system one can achieve discrete orthogonality by using the uniform discretization of the torus $\mathbb{T}$. The points of uniform discretization can be received as the solutions of the following equation:

(1.24)
$$z^n = \zeta_0^n \qquad (\zeta_0 \in \mathbb{T}, \ n \in \mathbb{N}_+).$$

In particular, for $\zeta_0 = 1$ we get back the nth roots of unity:

$$\left\{ \zeta_k = e^{2\pi i k/n} : 0 \le k < n \right\}.$$

Once the discretization points are known, the Fourier coefficients can be efficiently calculated by using the fast Fourier transform (FFT) introduced by Cooley and Tukey in [CooTuk65], which can be generalized for constructing discrete orthogonal rational functions. More precisely, for a given vector of inverse poles $\mathbf{a} := (a_0, \dots, a_{n-1}) \in \mathbb{D}^n$, Eq. (1.24) can be generalized as follows

$$A_n(z) = A_n(\zeta_0) \qquad (\zeta_0 \in \mathbb{T}, \ n \in \mathbb{N}_+),$$

where A_n denotes the so-called Blaschke products

$$A_n(z) = \prod_{k=0}^{n-1} B_{a_k}(z) \qquad (z \in \overline{\mathbb{D}}).$$

As we know from Section 1.2, these Blaschke products can be expressed by the sum of their argument functions

$$A_n(e^{it}) = e^{i\theta_\mathbf{a}(t)} \qquad (t \in \mathbb{R}),$$

where

(1.25) $$\theta_\mathbf{a}(t) = \frac{1}{n} \sum_{k=0}^{n-1} \beta_{a_k}(t) \qquad (t \in \mathbb{R}).$$

Since the β_{a_k} functions are continuous and strictly monotonic, there is exactly one solution $\tau_k \in [\tau_0, \tau_0 + 2\pi)$ for every $k = 0, 1, \ldots, n-1$ which satisfies the equation

$$\theta_\mathbf{a}(\tau_k) = \theta_\mathbf{a}(\tau_0) + 2\pi \frac{k}{n},$$

where $\zeta_0 := e^{i\tau_0}$. Now, the set of discretization points $\mathbb{T}_n$ and the corresponding weight function ρ_n can be defined as follows

(1.26) $$\mathbb{T}_n := \left\{ e^{i\tau_k} : 0 \le k < n \right\}, \qquad \rho_n(e^{it}) = \frac{1}{\theta_\mathbf{a}'(t)}.$$

Finally, by Theorem 2 in [FriSch11] the corresponding MT system $\{ \Phi_k : 0 \le k < n \}$ forms a discrete orthogonal system on $\mathbb{T}_n$. Namely,

$$[\Phi_k, \Phi_\ell]_n = \delta_{k,\ell} \qquad (0 \le k, \ell < n),$$

where $[.\,,.]_n$ denotes the discrete scalar product

(1.27) $$[f, g]_n = \sum_{z \in \mathbb{T}_n} f(z)\overline{g}(z)\rho_n(z) \qquad (f, g \in \mathfrak{R}_n).$$

The same is true for a vector of different inverse poles $\mathbf{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{D}^n$ with multiplicities $\mathbf{m} = (m_0, \ldots, m_{n-1}) \in \mathbb{N}_+^n$:

$$[\Psi_{\ell j} \varphi_{ki}]_N = \delta_{ij}\delta_{k\ell} \qquad (1 \le i \le m_k, \ 1 \le j \le m_\ell, \ 0 \le k, \ell < n),$$

where $N = m_0 + m_1 + \cdots + m_{n-1}$. We note that in Section 5.2 we give a numerical method to compute the non-uniform discretization points $\mathbb{T}_n$, which enables us to use FFT type algorithms [Sch02] to determine the related MT–Fourier coefficients.

Related works

The adaptivity of a system is very important for signal processing. On the one hand, the error of the representation can be decreased by choosing optimal parameters for the system. On the other hand, they can be used as features in classification tasks. A high level of compactness can be achieved in different ways. That is why the well-known Hermite orthogonal polynomials were used by Sörnmo et al. [SörBör81]. As far as we know, it was one of the first attempts to model the QRS complexes of the ECG by using adaptive systems. The main concept of this method was to exploit the shape similarity between Hermite polynomials and QRS complexes. In order to increase the adaptivity, dilation parameters were introduced via argument transformations of the fixed Hermite basis. It was later used by Jané et al. [JanOlm93] to construct an ECG compression method. Then, a recent study improved this algorithm by using discrete Hermite functions [SanSab12]. Expert systems also applied these orthogonal bases in clustering QRS complexes [LagPet00] and classifying heartbeats [HarEde04]. Additionally, other classical orthogonal polynomials such as Chebyshev and Legendre polynomials were used in modeling the QRS complex [GeoVal12]. We note that the construction of our work with rational functions is quite similar to the evolution of Hermite based ECG processing methods. For instance, the MT system was first applied in order to construct ECG compression methods [FriLóc12; LócKov12]. Then, we modeled the QRS complex [FriKov12] by utilizing the shape similarities of the elements of the basic RF system. The potentials of rational functions in QRS detection was examined by Gilián et al. [GilKov14]. Furthermore, different types of rational functions were adopted in EEG classification tasks like epileptic seizure detection [KovSam14; SamKov14].

Although their evolutionary structures are similar, our approach is essentially different from the Hermite based ECG processing methods. For instance, the latter has only one degree of freedom (i.e., dilation) while the rational function systems have an infinite number of free parameters (i.e., inverse poles). Since we can have a high-dimensional search space, finding the optimal parameters is more difficult. It also raises the need for more sophisticated optimization techniques (see e.g., Chapter 2). Besides, various problems arise in case of rational functions. Namely, the classical polynomial spaces $\mathcal{P}_N$ become dense in H^2 as the dimension N tends to infinity. However, the same is true for the subspaces of rational functions $\mathfrak{R}_N^b$ only if Eq. (1.15) holds. Furthermore, constructing quadrature formulas is also difficult in case of rational functions. Indeed, in order to define Gaussian quadrature formulas on the real line or Szegő type quadrature formulas on the unit circle the zeros of rational functions should be simple and lie on $\mathbb{R}$ or $\mathbb{T}$, respectively. Unfortunately, it is not true for orthogonal rational functions. For instance, in case of the MT system, there can be multiple zeros which lie inside $\mathbb{D}$. Appropriate abscissas can be obtained via quasi- and para-orthogonal systems, i.e., orthogonality conditions are not required for constants. Bultheel et al. [BulGon09] introduced a unified approach via the so-called Cayley transform, which maps the

extended real line to $\mathbb{T}$ and the upper half plane to $\mathbb{D}$. They constructed rational quadrature formulas via quasi-orthogonal and para-orthogonal systems by showing that the zeros of these functions are simple and lie on $\mathbb{T}$. We note that there is another way to construct quadrature formulas for rational functions. Namely, one can define rational interpolation via the non-uniform discretization points of $\mathbb{T}_n$ (see e.g., Eq. (1.26)), which implies the construction of Gauss type quadrature formulas. Moreover, it can be proved that these formulas have a maximal domain of validity. For further details we refer to the paper [Sza04]. It is also worth mentioning that a more general concept was introduced in [BulGon09] in order to construct rational orthogonal bases. In this sense it is not only the system, but the corresponding weight function can be changed. Hence, the number of free parameters, i.e., adaptivity can be increased by using a sequence of varying complex inner products. Although these results are beyond the scope of this dissertation, it is in our interests to examine their efficiency in signal processing applications. Classical orthogonal polynomials can also be generalized by using rational weight functions [FisGol92]. These methods including different numerical algorithms, quadrature rules and approximations techniques are summarized in [Gau04]. Other transformations like Walsh functions [BerChi00], wavelets [Add05], B-splines [KarMon97], etc., have also been applied in signal processing algorithms. In Section 5.3 we give a detailed description and comparisons of these methods along with rational function based approaches. Besides, we enumerate our motives to use rational systems over the other transformation methods :

- flexibility in the sense that not only the coefficients but the system itself can also be varied, which means that the base functions can be adapted to the signal;

- the coefficients give a compressed representation of the signal, so they can be used as features in further processing steps;

- the elementary waves are localized in time and the basic functions can carry time-frequency information;

- this is a simple analytic representation of the original signal and thus the whole set of analytic tools can be applied on the representation;

- only a couple of arithmetic operations are required to recover the signal;

- fast Fourier algorithms can be constructed, which provide efficient implementations.

As it was mentioned in the previous sections, the MT and the BRF systems have orthogonal and biorthogonal properties in discrete sense as well. In this case, the non-uniform discretization of $\mathbb{T}$ is defined by the argument function of Blaschke products in Eq. (1.26). However, such interpolation processes can be constructed on the uniform grid of $\mathbb{T}$ as well by using the discrete analogue of the Cauchy integral formula. For further details, we refer to [PapSch04].

Chapter 2

Optimization

This chapter is about the optimization of the parameters of finite rational function systems. Since such systems are defined by the collection of inverse poles with multiplicities, the following parameters should be determined: the number, the positions and the multiplicities of the inverse poles. A typical choice for such optimization problems is Nelder–Mead simplex algorithm. In this specific case [Lóc09], a bijection was used between $\mathbb{R}^2$ and $\mathbb{D}$ for keeping the inverse poles inside the unit circle. This is, however, a very artificial constraint, which comes from the Euclidean geometry. We will show that the Poincaré model of the hyperbolic geometry outlined in Section 2.1 is a more natural choice in this respect. It follows from the hyperbolic model that the vertices of the simplex stay inside $\mathbb{D}$ without any additional constraints. We note that this algorithm will be used in Section 3.3 for localizing the optimal inverse poles of the QRS model.

We note that the hyperbolic variants of the geometric operators needed for dimensions higher than 2 are not trivial. We consider two approaches to address this issue. Namely, the required operators can be defined via complex analysis or analytic geometry. In this dissertation, we chose the former one because it can be adapted to other optimization techniques like particle swarm optimization (PSO) in Section 2.2. The corresponding hyperbolic Nelder–Mead algorithm was published in [FriKov12]. Then the construction was applied to PSO [KovKir13] for single and multi-pole systems as well. The approach based on analytic geometry [Lóc13] uses geometric interpretations of hyperbolic lines and planes which become circular arcs and spherical caps in $\mathbb{R}^3$. This way, the hyperbolic Nelder–Mead algorithm can be extended up to 3 dimensions.

In the following section of this chapter we present a new optimization method that applies to both the optimal positions and the number of the inverse poles. This is justified by the fact that in previous works the number of the inverse poles were fixed, and optimization algorithms were applied to the positions of the inverse poles only. It is an important generalization of the previous methods, which provides a more compact and adaptive representation of discrete signals. This way we can utilize the well-known fact that smooth functions have better approximation properties. In this context we refer to Jackson's theorems in [Nat52], and the fact

that the Fourier coefficients of such regular functions tend faster to zero. Hence, the same level of accuracy can be achieved by keeping fewer terms in the (MT) Fourier series. Then the problem is that we do not have a priori information about the smoothness of the signal. A key point is that no such information is necessary when we use the MDHPSO algorithm in Section 2.3. ECG signal compression is a typical example for this problem, where abnormal heartbeats have more complex waveforms in contrast with normal ones. Thus the latter ones can be represented by fewer parameters in order to achieve the same level of accuracy. The developed MDHPSO algorithm automatically adjusts these constraints.

In the final section, we perform the corresponding stability analysis with experiments. We consider higher dimensional problems and test the previously mentioned optimization techniques on real and synthesized data. For this purpose, the ECG signal generator in Section 3.1, artificial signals with random inverse poles and the PhysioNet ECG database are used. We note that the cost function should be evaluated only at some specific points of the simplex, and therefore the Nelder–Mead algorithm can be faster than the PSO. On the other hand, the chance of being trapped in a local extrema is higher in the first case. We remark that some partial results for low dimensions can be found in [Lóc09].

The material of this chapter was published in [FriKov12; KovKir13].

2.1 Hyperbolic Nelder–Mead algorithm (HNM)

The Nelder–Mead simplex algorithm [NelMea65] is a fast and widely used deterministic optimization search technique for multidimensional unconstrained minimization. Despite its age it is still a very popular method for practitioners. It is simple and does not contain differentiation. We note that there are only a few results about the convergence properties of the Nelder–Mead algorithm. They are under strict conditions and for dimensions ≤ 2 (see e.g., [LagRee98]). The method is based on concepts and transformations in the usual Euclidean geometry. Concerning the use of the original Nelder–Mead algorithm for finding the inverse poles of rational functions, we refer to [Lóc09]. In this section we take the Poincaré model on $\mathbb{D}$ of the hyperbolic geometry. Our reason for taking the hyperbolic model is that we need to keep the inverse poles within the unit circle. We note that the hyperbolic model turned out to be useful in system and control theories as well (see e.g., [BokSch09]). In the Poincaré model the arcs intersecting the unit circle perpendicularly and the diameters play the role of straight lines. They can be described by means of the Blaschke functions. Namely, every hyperbolic line can be given in a parametric form

$$(-1, 1) \ni t \to B_{\mathfrak{a}}(t) \in \mathfrak{B} \qquad (\mathfrak{a} \in \mathbb{B}).$$

It can be shown that for any pairs $w_1, w_2 \in \mathbb{D}$, $w_1 \neq w_2$ there exist a unique parameter $\mathfrak{a} := (a, \epsilon) \in \mathbb{B}$ and a number $p \in (0, 1)$ such that $B_{\mathfrak{a}}(0) = w_1$, and $B_{\mathfrak{a}}(p) = w_2$. Moreover, the parametrization of the hyperbolic line connecting

w_1 and w_2 is $B_{\mathfrak{a}}$, and $B_{\mathfrak{a}}$ maps the interval $[0, p]$ onto the hyperbolic segment connecting w_1, and w_2. These parameters can be calculated as follows

$$(2.1) \qquad p = |B_{w_1}(w_2)|, \quad \epsilon = \frac{B_{w_1}(w_2)}{|B_{w_1}(w_2)|}, \quad \mathfrak{a} = -\overline{\epsilon} w_1,$$

where we recall that $B_{w_1} = B_{(w_1, 1)}$ (see e.g., Eq. (1.16)). The so-called pseudo-hyperbolic metric on $\mathbb{D}$ is defined by

$$\rho_0(z_1, z_2) := \frac{|z_1 - z_2|}{|1 - \overline{z}_1 z_2|} = |B_{z_1}(z_2)| \qquad (z_1, z_2 \in \mathbb{D}).$$

Then $(\mathbb{D}, \rho_0)$ is a complete metric space which is invariant with respect to the Blaschke transforms, i.e.,

$$(2.2) \qquad \rho_0(B_{\mathfrak{a}}(z_1), B_{\mathfrak{a}}(z_2)) = \rho_0(z_1, z_2) \qquad (z_1, z_2 \in \mathbb{D}, \mathfrak{a} \in \mathbb{B}).$$

One can prove that the group of hyperbolic congruences can be identified with the collection of the transforms $\{B_{\mathfrak{a}} : \mathfrak{a} \in \mathbb{B}\}$. In particular, the geometric operations in the Nelder–Mead algorithm can all be expressed in the hyperbolic plain by means of the $B_{\mathfrak{a}}$ functions. We note that the interval $(-1, 1)$ itself is a hyperbolic line in which the distance between the points $-1 < q < p < 1$ is

$$\rho_0(p, q) = \frac{p - q}{1 - pq}.$$

We take the $\overline{0p}$ hyperbolic line segment as a special case. Let the hyperbolic middle point of it be denoted by p_F and let p_T be the reflection of 0 with respect to the point p. Then the following equations of second degree hold for them

$$p_F = \frac{p - p_F}{1 - pp_F}, \quad p = \frac{p_T - p}{1 - pp_T}.$$

It is easy to see that they both have a unique solution in $(0, 1)$. For any hyperbolic line segment $\overline{w_1 w_2}$ the middle point w_F and the reflection w_T of w_1 with respect to w_2 can be be given in the form

$$B_{\mathfrak{a}}(p_F) = w_F, \quad B_{\mathfrak{a}}(p_T) = w_T,$$

where $B_{\mathfrak{a}}$ is the parametrization of the hyperbolic line connecting w_1 and w_2. The hyperbolic version of the Nelder–Mead algorithm is designed to minimize the functions of type $F : \mathbb{D} \to \mathbb{R}$. In our problems related to the QRS complexes, we will need the following variant of it. In order to define the main step of the algorithm, let z_1, z_2, z_3 belong to $\mathbb{D}$ and be indexed according to the relation

$$F(z_3) \leq F(z_2) \leq F(z_1).$$

Furthermore, let the middle point of the hyperbolic line segment $\overline{z_2 z_3}$ be denoted by z_0, and the hyperbolic reflection of z_1 with respect to z_0 by z_e. Then a new
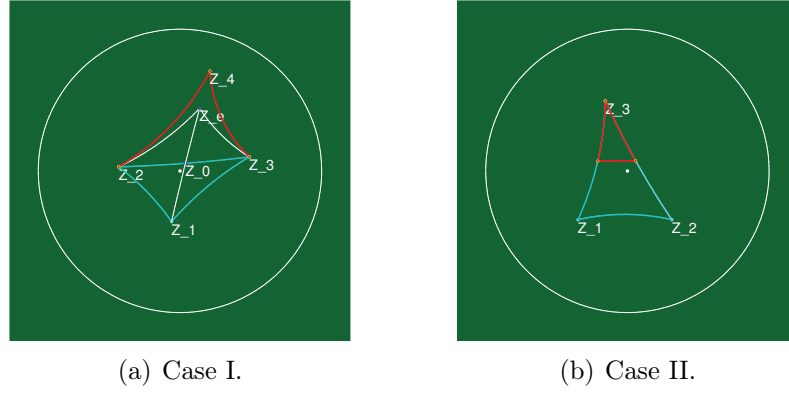
(a) Case I.   (b) Case II.

Figure 2.1: Hyperbolic operations of the HNM algorithm.

point z' that depends on the value $\mathsf{F}(z_e)$ will be defined. Finally, we replace z_1 by z' (*Case I.*), or modify the original triple (*Case II.*) in order to have the new triple of points. We note that in *Case I.* we define z' so that the condition

$$\max\{\mathsf{F}(z_2), \mathsf{F}(z_3), \mathsf{F}(z')\} < \max\{\mathsf{F}(z_1), \mathsf{F}(z_2), \mathsf{F}(z_3)\}$$

holds for it.

   Case I.:

a) If $\mathsf{F}(z_3) \leq \mathsf{F}(z_e) < \mathsf{F}(z_2)$, then let $z' = z_e$.

b) If $\mathsf{F}(z_e) < \mathsf{F}(z_3)$, then let z_4 be the hyperbolic reflection of z_0 with respect to z_e. This is illustrated in Fig. 2.1(a). Then by comparing the values $\mathsf{F}(z_4)$ and $\mathsf{F}(z_e)$, the point z' is defined as follows:
If $\mathsf{F}(z_4) < \mathsf{F}(z_e)$, then let $z' = z_4$; otherwise let $z' = z_e$.

c) If $\mathsf{F}(z_2) \leq \mathsf{F}(z_e) < \mathsf{F}(z_1)$, then let us take the middle point of the hyperbolic line segment $\overline{z_0 z_e}$ and denote it by z_4. Provided $\mathsf{F}(z_4) \leq \mathsf{F}(z_e)$, let $z' = z_4$; otherwise turn to *Case II.* below.

d) If $\mathsf{F}(z_1) \leq \mathsf{F}(z_e)$, then let z_4 be the middle point of the hyperbolic line segment $\overline{z_0 z_1}$. Provided $\mathsf{F}(z_4) < \mathsf{F}(z_1)$, let $z' = z_4$; otherwise turn to *Case II.* below.

   Case II.: Let z'_1 be the middle point of $\overline{z_1 z_3}$. Similarly, let z'_2 be the middle point of $\overline{z_2 z_3}$. Then the triangle $z_1 z_2 z_3$ will be replaced by $z'_1 z'_2 z_3$. This is illustrated in Fig. 2.1(b).

It is easy to check that by this construction the condition made above for z' is fulfilled. By repeating the steps, the triangle shrinks around the best vertex and the process can be stopped when the desired accuracy is reached.

   We note that generally the limit of the process may depend on the starting triangle. So it should be chosen according to the nature of the problem. For

instance, we have experienced that there are at least two local minima only when it is applied for the record of an entire heartbeat rather than for the QRS complex, and the process may converge to any of them. In that case we can ensure the proper convergence by taking the initial values close to the expected limit. On the other hand, our experiments showed that in case of QRS complexes and the linear subspace $\mathcal{L}_a$ defined in Section 3.3, the minimum is unique and the process converges to the point of minimum. Some of the theoretical results related to the original Nelder–Mead algorithm in [LagRee98] can be applied to the hyperbolic variant as well. For instance, non-degeneracy of the hyperbolic simplices and convergence properties for strictly convex functions were proved in [Lóc13].

2.2 Hyperbolic particle swarm optimization (HPSO)

Although HNM produced good inverse pole configurations, it turned out to be unstable. In other words, it was sensitive to the initial conditions (i.e., the starting points of the simplex). As a consequence, running the algorithm several times on the same problem may result in different inverse poles (see e.g., Section 2.4). This issue, which is well-known in optimization theory, is caused by being trapped in local maxima or minima. In this section, we show that the well-known particle swarm optimization (PSO) method is more appropriate for this problem.

Namely, we present an extension of the PSO algorithm based on the Poincaré model of the hyperbolic geometry. We apply this method on ECG signals to determine the optimal parameters of the rational function systems. For the sake of simplicity, we are considering the simple two dimensional case first, where we need to find only one inverse pole with predefined multiplicity. Thus, we need to find the position of this single inverse pole for a rational system which minimizes the ℓ^2 error in Eq. (1.4). Then, we extend the method to multi-pole problems. Finally, we apply the multi-dimensional PSO algorithm to determine the optimal number of inverse poles as well.

Basic PSO algorithm

The basic PSO algorithm was introduced by Eberhart and Kennedy [KenEbe95] as a population based stochastic optimization technique. The method was inspired by the social behavior of bird flocking or fish schooling. The algorithm works similarly to a swarm which is flying through the (problem) space while they are looking for an optimal point (e.g., food). Both the swarm and its particles have a memory, i.e., they can remember their own known global and personal best positions, respectively. During the search process the swarm is navigated by these optima. In case of n dimensional search spaces, the method is initialized with a random population $\{x_k \in \mathbb{R}^n : 1 \leq k \leq S\}$, where $S \in \mathbb{N}_+$ denotes the size of the swarm and each individual x_k is a potential solution for the given problem. In

this sense, it shows similarities to the Evolutionary Algorithms such as Genetic Algorithm. For instance, in our case each particle x_k represents an inverse pole configuration. Generally, the PSO can also be considered as a special case of the Monte-Carlo simulation where the initial random population is led by certain points at each iteration. During the search process each individual $x_k \in \mathbb{R}^n$ keeps track of its position in an n dimensional search space related to the personal $\widetilde{y}_k \in \mathbb{R}^n$ and global $\widehat{y} \in \mathbb{R}^n$ best solutions so far achieved. In each step, both the position and the velocity of the kth particle are updated as follows:

$$v_k = c_1 r_1 \cdot (\widetilde{y}_k - x_k) + c_2 r_2 \cdot (\widehat{y} - x_k) + w \cdot v_k\,,$$

(2.3)
$$x_k = x_k + v_k\,,$$

where the learning factors c_1, c_2 are predefined constants and $r_1, r_2 \in (0,1)$ are uniformly distributed random numbers. The inertia weight w was introduced later [ShiEbe98] in order to control the overall behavior of the swarm. For instance, one can favor exploration by increasing the value of w. Arbitrary large jumps are usually inhibited in the search space. For this reason, the velocities and the positions are restricted to a certain interval defined by the variables, $V_{max}, X_{min}, X_{max}$. Note that in further experiments, we use the algorithm by setting $c_1 := 1.5$, $c_2 := 2$ while w is linearly decreasing from 0.8 to 0.2. For other strategies of the parameter selection and convergence analysis, we refer to [Tre03; BerEng06]. The pseudocode of the basic PSO algorithm Alg. 4 can be found in Appendix A.

In this section, we first adopt the PSO method to single-pole problems in Eq. (1.4). For this reason, the particles contain only two coordinates related to the real and imaginary parts of the inverse pole. Namely, if the algorithm terminates in the ℓth optimal particle, then $a_0 := x_{\ell,1} + i x_{\ell,2}$. In this case, the optimal inverse pole and its multiplicity are $\mathbf{a}_o := a_0 \in \mathbb{D}$ and $\mathbf{m} := m_0 \in \mathbb{N}_+$, respectively. Then the corresponding MT system reduces to the following form

$$\Phi_k(e^{it}) = \frac{\sqrt{1 - r^2}}{1 - r e^{i(t-\alpha)}} \cdot e^{ik\beta_{a_0}(t)} \quad (t \in [-\pi, \pi))$$

with $(0 \le k < m_0)$, where $a_0 = r e^{i\alpha}$, $\alpha \in [-\pi, \pi)$, $r \in [0,1)$. Both the first factor of this product and the argument function β_{a_0} are translated in time with the angle α (cf., Eq. (1.11)). Furthermore, the parameter r can be associated to a kind of dilation operation. One can see an example of this property in Fig. 2.2, which displays the real part of Φ_0 related to inverse poles with various angles and absolute values. Generally, the dilation and the translation operations can be realized by the parameters $(r,1) \in \mathbb{B}_1$ and $(0, \epsilon) \in \mathbb{B}_2$. This concept was also applied to construct hyperbolic wavelets in [SouSch11a].

Hyperbolic PSO algorithm

As we know from Chapter 1, the inverse poles of the MT system should lie within the unit circle. Hence, in order to apply the basic PSO algorithm to the single-pole
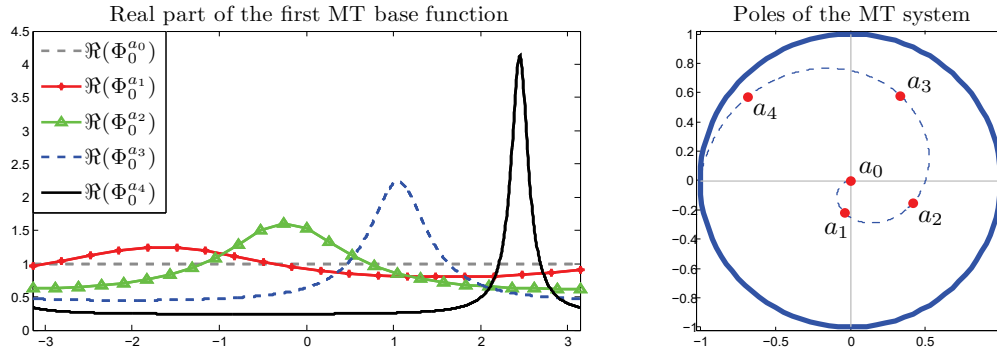
Figure 2.2: Translation and dilation properties of the MT system with respect to the position of the inverse pole.

optimization problem, we should restrict the search space to the open unit disc $\mathbb{D}$. This implies the idea to use the Poincaré model of the hyperbolic geometry. Namely, we will replace the operators $\cdot, +, -$ in Eq. (2.3) by hyperbolic multiplication $\odot$, addition $\oplus$ and subtraction $\ominus$.

Hyperbolic multiplication

Using the terminology of the Euclidean geometry, the vector scalar multiplication of the hyperbolic space can be defined in a similar way. Namely, it means the scaling of a hyperbolic vector by keeping its direction. In this case, the geodesics of this space are represented by arcs of circles that are orthogonal to the torus. Furthermore, as it was shown in Section 2.1, the hyperbolic segments can be defined via the Blaschke functions. Hence, the hyperbolic vector $\overrightarrow{w_1 w_2}$ can be interpreted as a directed line segment of Eq. (2.1) with $B_a(0) = w_1$ and $B_a(p) = w_2$. Let us consider the scaling of a hyperbolic vector $\overrightarrow{w_1 w_2}$ by the factor $\lambda \in \mathbb{R}$. In order to determine the coordinates of the new endpoint, we should recall the definition of the hyperbolic metric

$$\rho(z_1, z_2) := \operatorname{arth}(\rho_0(z_1, z_2)) \qquad (z_1, z_2 \in \mathbb{D}).$$

Then $(\mathbb{D}, \rho)$ is a complete metric space which is also invariant with respect to the Blaschke transforms. Now, we should calculate the new w_λ endpoint, which satisfies the following equation

$$\rho(w_1, w_\lambda) = \lambda \rho(w_1, w_2).$$

Using the identities of Eq. (2.1) for the segment $\overrightarrow{w_1 w_2}$ and adopting the invariance of ρ_0 in Eq. (2.2) for the hyperbolic metric ρ, the problem can be written as

$$\operatorname{arth}(s_\lambda) = \rho(0, s_\lambda) = \lambda \rho(0, p) = \lambda \operatorname{arth}(p),$$

where $w_\lambda = B_a(s_\lambda)$. Reordering this equation, we can get

(2.4) $$s_\lambda = \operatorname{th}(\lambda \operatorname{arth}(p)).$$

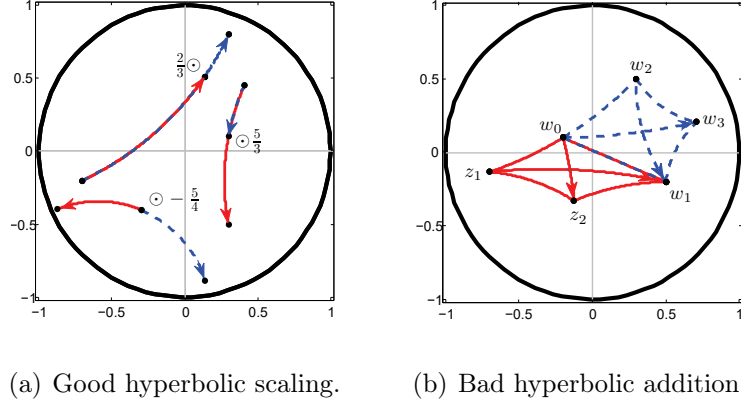(a) Good hyperbolic scaling.　　　　(b) Bad hyperbolic addition.

Figure 2.3: Hyperbolic operators.

In summary, the expression $\lambda \odot \overrightarrow{w_1 w_2} := \overrightarrow{w_1 w_\lambda}$ can be evaluated with Eqs. (2.1) - (2.4) as $w_\lambda = B_{\mathfrak{a}}(s_\lambda)$ for any $\lambda \in \mathbb{R}$. Fig. 2.3(a) shows an example where the dashed blue lines represent the original vectors while the scaled ones are marked by red color.

Hyperbolic addition

According to the previous section the natural approach to define a hyperbolic addition would be a proper interpretation of the parallelogram rule. Unfortunately, this method cannot be used since the difference of two vectors $\overrightarrow{w_0 w_1} - \overrightarrow{w_0 w_2}$ is usually not equal with the addition in the opposite direction $\overrightarrow{w_0 w_1} + ((-1) \odot \overrightarrow{w_0 w_2})$ in hyperbolic sense. One can see an example for this phenomenon in Fig. 2.3(b). Here, z_1 denotes the reflection of w_2 onto w_0, i.e., $\overrightarrow{w_0 z_1} = (-1) \odot \overrightarrow{w_0 w_2}$. Then the hyperbolic difference of $\overrightarrow{w_0 w_1} - \overrightarrow{w_0 w_2}$ was constructed as $\overrightarrow{w_0 z_2} = \overrightarrow{w_0 w_1} + \overrightarrow{w_0 z_1}$. As we mentioned, it is not equal to the translation of the vector $\overrightarrow{w_2 w_1}$. Moreover, their magnitudes are also different: $\rho(w_1, w_2) = 0.92$ and $\rho(w_0, z_2) = 0.46$.

Hence, the proper definition of hyperbolic addition relies on the composition of Blaschke functions. Namely, if $\mathfrak{a}_1 = (w_1, 1)$ and $\mathfrak{a}_2 = (w_2, 1)$, then by Eq. (1.7) $\mathfrak{a}_1 \circ \mathfrak{a}_2 = (w, \epsilon)$, where $w = \dfrac{w_1 + w_2}{1 + w_1 \overline{w_2}}$ and $\epsilon = \dfrac{1 + w_1 \overline{w_2}}{1 + \overline{w_1} w_2}$. This can be interpreted as a vector addition in the hyperbolic space for vectors with initial point at zero (see e.g., [Hen74]). Therefore, in this section, we will use the following operators

$$\overrightarrow{0 w_1} \oplus \overrightarrow{0 w_2} := \overrightarrow{0 w}, \qquad \text{where } w = \frac{w_1 + w_2}{1 + w_1 \overline{w_2}} \qquad (w_1, w_2 \in \mathbb{D}),$$

$$\overrightarrow{0 w_1} \ominus \overrightarrow{0 w_2} := \overrightarrow{0 w}, \qquad \text{where } w = \frac{w_1 - w_2}{1 - w_1 \overline{w_2}} \qquad (w_1, w_2 \in \mathbb{D}).$$

Multi-pole extension

Although, we considered only the case of single-pole optimization in the last sections, it is easy to extend the method into higher dimensions. Dealing with multi-pole problems, the optimal inverse poles $\mathbf{a_o} = (a_0, \ldots, a_{n-1}) \in \mathbb{D}^n$ should be determined for a signal f. Hence in Eq. (1.4) $\|.\|_{L^2_w} = \|.\|_{H^2}$, $\mathbf{a_o}$ is the vector of inverse poles and the functions $\Theta_k^{\mathbf{a_o}}$ are associated with the elements of the corresponding MT or BRF systems. However, describing rigid movements of the hyperbolic space is a difficult task in high $(3 <)$ dimensions. So, the geometrical approach cannot be applied directly on multi-pole problems. Thus, we perform the optimization separately on each inverse pole as

$$(2.5) \qquad \mathbf{a}_{o_i} = \arg \min_{\mathbf{a}_i} \left\| f - S_n^{\Theta, \mathbf{a}} f \right\|_{H^2} \quad (i = 0, \ldots, n-1).$$

In this sense, the original multi-pole problem is separated into single-pole optimizations by applying successively the two dimensional hyperbolic PSO (HPSO). However, Eq. (2.5) should be evaluated n times, it represents only one update step of a certain particle. Note that the swarm cannot leave the unit circle during the algorithm. It is a natural consequence of the hyperbolic model, which makes the search space boundaries X_{min}, X_{max} used in the original Euclidean algorithm unnecessary. Now, the positional updates of the kth particle in Eq. (2.3) can be modified as follows

$$
\begin{aligned}
v_k &= c_1 r_1 \odot (\widetilde{y}_k \ominus x_k) \oplus c_2 r_2 \odot (\widehat{y} \ominus x_k) \oplus w \odot v_k\,, \\
(2.6) \qquad x_k &= x_k \oplus v_k\,,
\end{aligned}
$$

A detailed review of the related hyperbolic operations and their applications can be found in [Sch14]. Furthermore, the proper implementation of this algorithm is based on [KovKir13]. In that article, we showed that the HPSO can outperform other optimization techniques in terms of reconstruction error and stability. The latter one proved to be important, especially when the MT system is used in classification problems [KovSam14].

2.3 Multi-dimensional HPSO (MDHPSO)

Although PSO based algorithms are generally related to static environments, many practical problems change dynamically, which highlights the adaptivity of the systems. The multi-dimensional (MD) PSO algorithm was introduced by Kiranyaz et al. [KirPul11] to remove the necessity of setting fixed dimensions a priori. In order to avoid this drawback, the native structure of the swarm was extended by dimensional parameters. Thus, the particles can seek both positional and dimensional optima. In order to adapt this concept to inverse pole optimization, we extended the original algorithm (see e.g., Alg. 5) via hyperbolic operators. More precisely, the update step of the kth particle is modified as follows:

Position updates:

$$v_k^{d_k} = c_1 r_1 \odot (\widetilde{y}_k^{d_k} \ominus x_k^{d_k}) \oplus c_2 r_2 \odot (\widehat{y}^{d_k} \ominus x_k^{d_k}) \oplus w \odot v_k^{d_k},$$
$$(2.7) \qquad x_k^{d_k} = x_k^{d_k} \oplus v_k^{d_k},$$

Dimension updates:

$$vd_k = [c_1 r_1 \cdot (\widetilde{d}_k - d_k) + c_2 r_2 \cdot (\widehat{d} - d_k) + vd_k],$$
$$d_k = d_k + vd_k,$$

where $[.]$ is the integer rounding operator. The main changes comparing to Eq. (2.6) are the dimensional indices $d_k, \widetilde{d}_k, \widehat{d} \in \mathcal{I}$, which denote the current, personal and global best dimensions, respectively. In this case, every particle has a certain position and velocity at each dimension. For instance, $x_k^{d_k}$ denotes the position of the kth particle at the dimension $d_k \in \mathcal{I}$. In this dissertation, the index set is $\mathcal{I} = \{1, \ldots, 30\}$, where the indices address the inverse pole configurations in Tab. 2.1. In order to provide valid indices, we round d_k to the closest integer number and map it into $\mathcal{I}$. Further details can be found in Appendix A. We note that the best solution can also be determined in an exhaustive manner by applying the basic PSO algorithm for each dimension. However, it highly increases computational costs, which is unacceptable, especially in real time applications. On the contrary, the MDPSO can be interpreted as only two consecutive PSOs executed on the positional and the dimensional search space, which significantly decreases the computational complexity. For further details, we refer to [KirInc14].

The MDPSO was originally applied to evolve Artificial Neural Networks (ANN) for supervised learning [KirInc09], where the weights and bias of the network should be determined in order to minimize the classification error. Furthermore, the complexity of the corresponding ANN is also important. Thus, the required number of layers and neurons, i.e., the dimensionality should be minimized as well. Using the same analogy, we can identify the number of different inverse poles of the rational systems as the layers of an ANN and the multiplicities as the number of neurons in a specific layer. More precisely, let us consider the vector of different inverse poles $\mathbf{a} = (a_0, \ldots, a_{n-1})$ and let $\mathbf{m} = (m_0, \ldots, m_{n-1})$ represent the vector of multiplicities. Additionally, the architecture space can be described as the multi-layer perceptron (MLP) configurations in [KirInc09]. One can see an example in Tab. 2.1, where each inverse pole configuration is assigned to a dimensional index of the architecture space. Finally, we define the cost function f_c^α of the optimization. Let us consider the case when Eq. (1.15) is satisfied. Then, $\lim_{N \to \infty} \left\| f - S_N^{\Theta, \mathbf{a}} f \right\|_{H^2} = 0$ for any $f \in H^2(\mathbb{T})$, where $N = m_0 + m_1 + \ldots m_{n-1}$. In other words, the approximation tends to be more accurate as the number of inverse poles are increased. Thus, the MDPSO algorithm will always terminate in the highest dimension (i.e., in the largest subspace). In order to avoid that, we compose

Table 2.1: Architecture space of the MDHPSO.

Dimension index	Configuration	Dimension index	Configuration
1	(8)	11	(4, 2, 2)
2	(4, 2)	12	(6, 2, 2)
3	(6, 2)	13	(8, 2, 2)
4	(8, 2)	14	(4, 4, 2)
5	(4, 4)	15	(6, 2, 2)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	(8, 8)	30	(14, 8, 8)

f_c^α as the linear combination of the approximation error and the compression ratio:

$$\mathrm{PRD} := \frac{\left\| f - S_N^{\Theta,\mathbf{a}} f \right\|_{H^2}}{\left\| f - \bar{f} \right\|_{H^2}} \times 100\,, \qquad \mathrm{CR} := \frac{2 \cdot (n + N)}{M} \times 100\,,$$

$$f_c^\alpha(\mathrm{PRD}, \mathrm{CR}) := \alpha \cdot \mathrm{PRD} + (1 - \alpha) \cdot \mathrm{CR} \qquad (\alpha \in [0, 1])\,,$$

where f is a discrete-time signal, $\bar{f}$ is the mean of f and M is the number of samples. The approximation error is computed as the usual percent root mean square difference (PRD). In addition, the compression ratio (CR) is proportional to the number of inverse poles and coefficients $n + N$, which is multiplied by two, counting both their real and imaginary parts.

In this sense, the approximation error is extended by using the CR as a penalty term in higher dimensions. It means that a particle should provide much better approximation if it wants to step into a subspace with higher dimension. In other words, updating dimensions is only allowed when a certain amount of improvement can be achieved. Furthermore, the PRD and the CR are inversely proportional to each other. Thus, we can adapt the method to different applications by weighting these measures via α.

Note that we extended the original algorithm by using the hyperbolic operators $\oplus$, $\ominus$, $\odot$ in Eq. (2.7). For this reason, the modified algorithm is called MDHPSO, where the cost function is adapted to inverse pole optimization problems. Fig. 2.4 shows an example where we apply the algorithm on BP, RESP, ECG and EEG signals. As it can be seen, the method automatically adjusts the number of coefficients and the inverse poles to the complexity of the signal. For this reason, only two inverse poles are used to represent the BP and RESP signals with 6 and 8 coefficients, respectively. In case of ECG and EEG signals the optimal dimension of the architecture space is increased. Hence, it results in three different inverse poles (i.e., 6 degree-of-freedom) with 10 and 18 coefficients.

(a) BP : $\mathbf{m} = (\,4,\,2\,)$ (b) RESP : $\mathbf{m} = (\,4,\,4\,)$

(c) ECG : $\mathbf{m} = (\,4,\,4,\,2\,)$ (d) EEG : $\mathbf{m} = (\,8,\,6,\,4\,)$
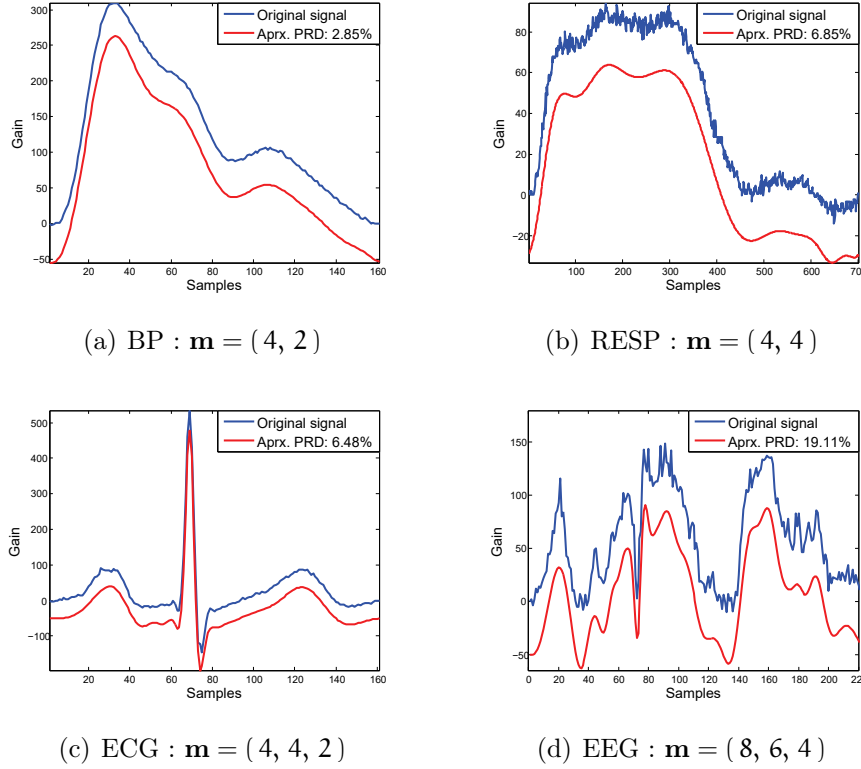
Figure 2.4: Approximation of different types of biomedical signals with $\alpha = 0.5$ by using the first four channels of the record *slp02a* from the MIT-BIH/slpdb database [GolAma00].

2.4 Pole stability analysis

The main goal of this subsection is to highlight the weaknesses of the original (Euclidean) Nelder–Mead (NM) algorithm and to justify the application of other optimization techniques including HPSO. For this reason, we use experiments to show that the NM method can hardly localize the original inverse poles of the MT system. On the contrary, HPSO has better pole identification properties.

NM algorithm

Compressing signals by using rational functions requires two phases. In phase one we have to find a proper system of rational functions. Phase two is the approximation in the corresponding linear subspace $\mathfrak{R}_N^a$. In phase one the original NM algorithm is an effective method for finding the inverse poles of the rational functions in question. However, the NM algorithm is not stable in the sense that it may result in different inverse poles for the same problem, depending on the initial values. Therefore, the NM algorithm for rational approximation of ECG signals needs to be calibrated. In Section 3.1, we propose an ECG signal generator
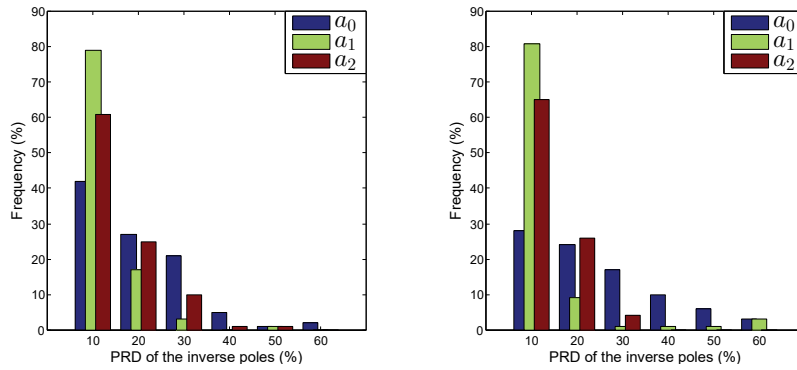
Figure 2.5: PRD of each inverse pole according to normal (left) and asymmetrical (right) tests.

that can be very useful for this purpose. It produces realistic ECG signals which are not affected by noise. We use this algorithm to test the stability of the NM algorithm for synthesized ECG signals. The testing process consists of four steps:

1. Generate normal ECG curves as rational functions.

2. Apply the NM algorithm to find the inverse poles.

3. Approximate the generated ECGs using the inverse poles of step 2.

4. Compare the inverse poles of the original and the approximated curves of step 1 and 3.

The problem with step one is how to directly create a realistic ECG curve by inverse pole vectors and coefficients. This is when the method of Section 3.1 comes into picture. Namely, we use our algorithm for generating a realistic ECG curve. Then we apply rational approximation to represent it as a rational function. It is clear that the way to control the ECG curve through poles and coefficients would be more difficult than to set these parameters explicitly utilizing our spline ECG model. Accordingly, a normal electrocardiogram was created by our polynomial model and then we applied steps 2–3 to determine an appropriate inverse pole-vector and coefficients for step 1. Now, we have a normal ECG curve that was generated by rational functions, so we can continue the test by repeating steps 2–4.

Here, we used the same synthetic ECG dataset as in Section 3.2. In addition, we use three inverse poles $\mathbf{a} = (a_0, a_1, a_2)$ with multiplicities $m_0 = m_2 = 1$ and $m_1 = 2$. The middle pole is repeated twice in order to emphasize its role in representing the QRS complex. Fig. 2.5 shows the PRD (see e.g., Eq. (3.2)) between the original and the approximated inverse poles. As it was expected, the inverse poles which correspond to significant lobes (QRS, T) of the heartbeat can be identified almost correctly. Namely, the error rate is less than 10% in
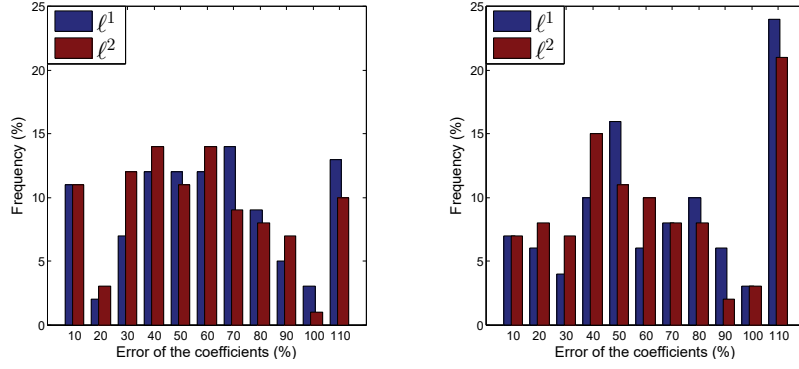
Figure 2.6: PRD of the coefficient vectors in ℓ^1 and ℓ^2 norm according to normal (left) and asymmetrical (right) tests.

most cases for a_1 and a_2. However, it is not true for a_0, which can have very large PRD. Consequently, the simplex algorithm can result in very different inverse poles with respect to the initial values. Note that a small displacement of the original inverse poles can greatly change the rational system and thus the corresponding coefficients of the representation. In order to visualize this problem, we computed the coefficients of the rational representation with respect to the original $\mathbf{a}$ and the optimized $\mathbf{a} + \Delta\mathbf{a}$ (find by NM) inverse pole vectors. Then, the PRD of these coefficient vectors are calculated in ℓ^1 and ℓ^2 norms. As it can be seen in Fig. 2.6, the coefficients are sensitive to the displacement $\Delta\mathbf{a}$ of the original inverse pole vector. Additionally, it causes higher error in case of asymmetric (i.e., more complex) waveforms. Thus, stabilization of the optimization is inevitable, especially for classification problems of Section 4.2.

Another definition of stability is based on the theory of numerical analysis; that is to say an algorithm is stable if it does not magnify numerical errors. Informally, a method cannot provide significantly different output for small changes of the input. Unfortunately, NM does not satisfy this requirement, because it is sensitive to the initial conditions. This phenomenon can be seen in Fig. 2.7. Here, we run the NM algorithm on the original data, which returns with an optimal set of inverse poles (green dots). Then, we slightly modify one of these inverse poles by adding 0.1 to its imaginary part (black star). This is followed by the reinitialization of the NM method using these modified inverse poles. The new set of predicted inverse poles (red circles) can be seen in Fig. 2.7(a). Although we made a small change in the initial conditions, the NM algorithm terminated in a very different optimum. Fig. 2.7(b) shows the corresponding approximations, where we applied a small vertical shift for better visualization. Note that the difference of the PRDs is less than 1%, which is irrelevant for compression problems. However, changes of the inverse poles remain a serious issue in case of classification tasks.
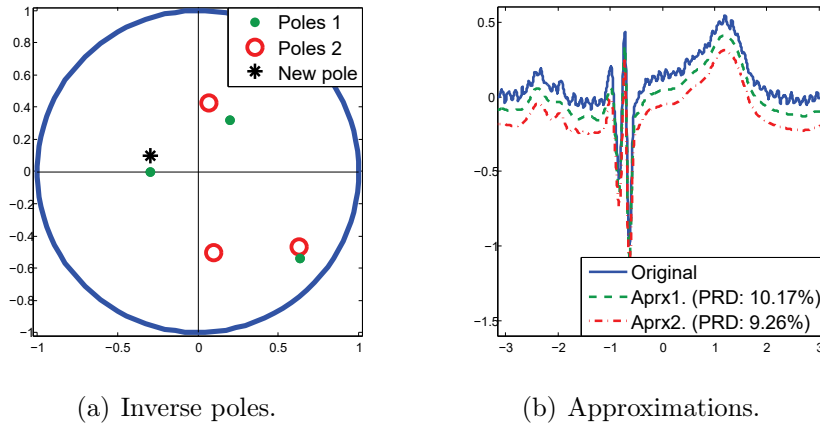
(a) Inverse poles.

(b) Approximations.

Figure 2.7: Inverse pole predictions by NM for different initial conditions.

HPSO algorithm

In this section, we perform some experiments on the HPSO algorithm to test its efficiency in various aspects. Our goal is to compare this method with the original (Euclidean) NM method. Recall that the inverse poles were strongly dependent on the initial conditions. For this reason, we test the stability of the inverse poles and the approximation error for the HPSO.

In the first experiment, we generate 400 signals with random inverse poles and coefficients. Then both algorithms are executed 100 times on each synthesized signal. One can see an example in Fig. 2.8(a) and Fig. 2.8(b), which show the resulted inverse poles on the same record. The original parameters are marked by black squares. Moreover, Fig. 2.8(c) displays the average error of 100 executions for all the 400 signals. One can see that the stability of the inverse poles is about twice as good as that of the HPSO algorithm.

In the second experiment we test the approximation error of both optimization methods. For this purpose, we use the MIT-BIH Arrhythmia Database from PhysioNet [GolAma00]. Namely, the first 3 minutes of each ECG record is segmented into heartbeats. Then these beats are approximated by using HPSO and NM algorithms (overall run > 9000). Relying on former research on ECGs [FriLóc12], we chose 3 inverse poles with multiplicities $m_0 = m_2 = 2, m_1 = 4$. Fig. 2.8(d) shows the average error of each ECG record, where the approximation error is measured in PRD.

Although the HPSO is slightly better than the NM algorithm, the difference is not significant. For this reason, we conclude that the inverse poles of HPSO are more stable while the approximation keeps its accuracy. Note that it is of great importance in case of classification tasks. Namely, it is a natural assumption to extract similar features (e.g., inverse poles and coefficients) for similar signals. However, it is not guaranteed with the NM algorithm. On the other hand, HPSO proved to be better in this sense, which makes it is possible to use the inverse poles and coefficients as features, for instance in ECG beat classification.

(a) Inverse poles of 100 runs of the NM algorithm.

(b) Inverse poles of 100 runs of the HPSO algorithm.

(c) Average PRDs of the resulted inverse poles for synthetic signals.
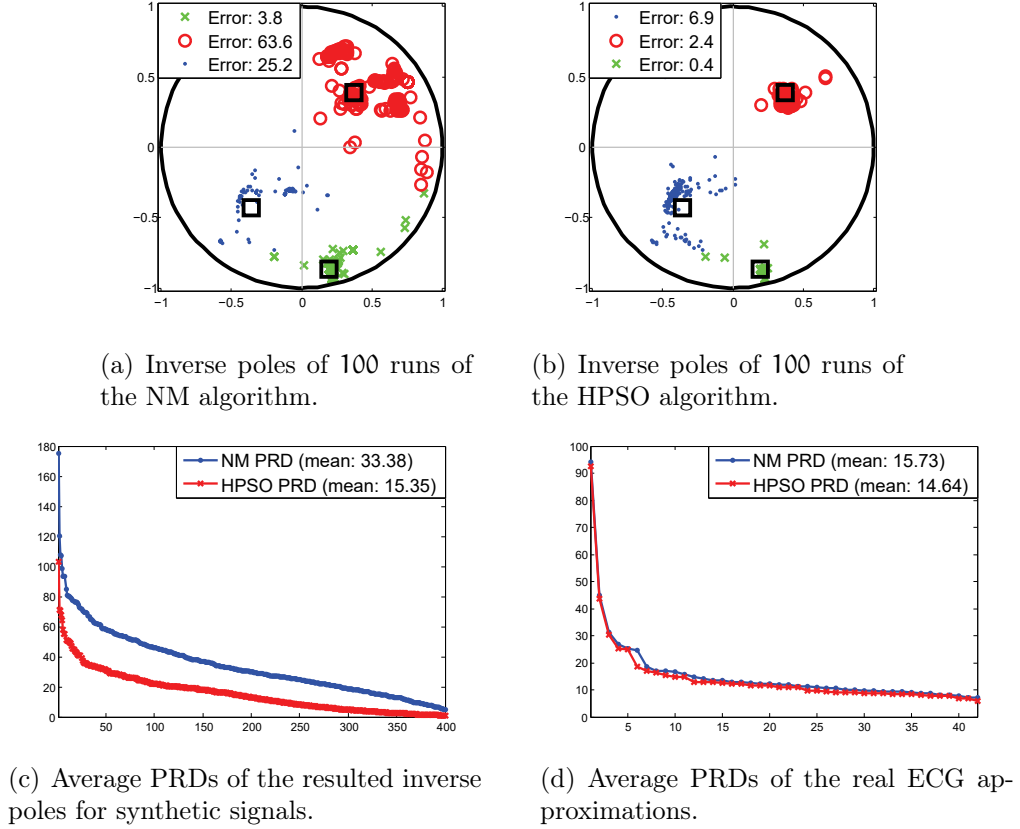
(d) Average PRDs of the real ECG approximations.

Figure 2.8: Experimental results on synthetic and real data.

It is worth mentioning that the performance of PSO type algorithms are dependent on the starting positions of the particles, which is crucial especially in higher dimensions. For instance in [RicVen04] the swarm was initiated by using centroidal Voronoi tessellation, which makes the starting positions more evenly distributed in the search space. It was also shown by experiments that the performance of the PSO algorithms can be improved by using this scheme in high-dimensional search spaces. Although this topic is beyond the scope of the dissertation, extending these results to hyperbolic variation of the Voronoi tessellation and HPSO is an interesting question.

Conclusions

We have extended the basic PSO algorithm to hyperbolic spaces by using the Poincaré disk model. The particles of this method move inside the unit circle, which does not require additional conditions like X_{min} and X_{max}. The algorithm is able to construct good approximations with better inverse pole stability. We will also use this property in Chapter 4 in order to extract features for EEG seizure detection problems. Furthermore, we showed by experiments that the HPSO outperforms the NM algorithm in every aspect (stability, approximation

error), except for the execution time, which depends on the size of the swarm. For instance, in case of n dimensional search spaces, the NM method evaluates the objective function at only $1, 2$ or $n + 2$ points when termination occurs in *Case I.* and *Case II.*, respectively (cf., Section 2.1). On the contrary, we used 20 number of particles in our HPSO experiments. Of course, this drawback of the algorithm can be improved by applying parallel implementations. Since there is an isomorphism between the Poincaré disk and the Beltrami–Klein model, Alg. 4 – 5 in Appendix A can be further extended to the latter model as well (see e.g., [Gre93]). In Section 2.3, the MD PSO scheme is also adopted for optimizing rational functions via hyperbolic geometry. In this sense, a good approximation can be achieved without using any types of a priori information. Moreover, it is possible to determine not just the positions of the optimal inverse poles, but the multiplicities and the best number of different inverse poles as well. We note that Section 5.3 gives a comparative analysis of different ECG compression methods. In these experiments, we used the original NM algorithm. Our expectation is that the results can be further improved by applying MDHPSO.

It is worth mentioning that the problem in Eq. (1.4) can be solved by using constrained optimization techniques as well. These methods can be classified into two categories, derivative-free methods, and methods involving derivatives of the objective function. In our case, signals are regarded as discrete-time series, consequently the derivatives are not available. In order to overcome this problem, one can use derivative-free algorithms such as sequential quadratic programming (SQP). The SQP methods optimize the quadratic model of the objective function (or its Lagrangian) at each iteration. The process requires the approximation of the gradient vector and the Hessian matrix, which can be computed via divided differences and quasi-Newton methods. Although there exist several variants of these algorithms, non of them turn to be the best in every respect. We refer to a recent study [RioSah13], in which 22 leading software implementations of derivative-free methods are analyzed. The motivation why we used MDHPSO instead of the available state-of-the-art methods was that it is able to determine also the optimal number of free parameters (i.e., inverse poles) while it avoids brute-force executions. We have no knowledge about any other algorithms by means of which that optimization is possible. Comparison analysis of the MDHPSO and the existing optimization methods will be a subject of a future work.

Chapter 3

ECG signal modeling

In this chapter we address two problems. First we develop an electrocardiogram (ECG) signal generator based on spline interpolation. Data simulation is an important task, especially when test data is not available. In addition, it is a useful tool for comparing signal processing algorithms using different constraints. For instance, in case of ECG compression methods preserving clinical features is crucial. Hence, these techniques are usually validated by medical experts in the literature, which is an expensive and time-consuming protocol. This problem can be avoided by using synthesized data with known characteristics. It is worth mentioning that there are other examples in the literature where artificial signals are used. Namely, McSharry et al. [McSCli03] proposed a dynamical model called ECGSYN to generate synthesized ECG signals. It is an open source program built in PhysioNet [GolAma00]. Another example is a former paper of Quiroga et al. [QuiNad04] where they simulated neuronal spike activity for clustering purposes.

To test our model developed in Section 3.1, we need to measure the reconstruction error. There are different ways to evaluate the distortion of the reconstructed signal. We will use the so-called PRD and WDD measures. PRD expresses the numerical error of the approximation and WDD imitates the diagnostic distortion. The disadvantage of WDD is that the medical features (wave amplitudes, widths, shapes, etc.) contained in it are hard to detect, especially in case of noisy signals. An important application of the signal generator is that by using it, the WDD distortion properties of the various approximation methods can be directly calculated and compared. Thus, our approach is more appropriate for comparing ECG compression algorithms via WDD than the previously mentioned methods like ECGSYN. As a result, we can characterize the diagnostic distortion of the rational ECG representation as well. It is one of the main contribution of this chapter since there were no such experiments before.

The second problem, which we consider in Section 3.3, is to construct a QRS model based on elementary rational functions. It is worth mentioning that there are other QRS models which utilize approximation theory in Hilbert spaces. For instance, Sörnmo et al. [SörBör81] applied Hermite functions to represent the QRS complex. Later, Georgiev et al. also used Legendre and Chebyshev polynomials in

[GeoVal12]. Then, Sandryhaila et al. [SanSab12] extended the basic Hermite model to discrete dilated orthogonal functions. Our concept is completely different, since we are modeling all the limb leads. Therefore, it is essential to reveal a relation between optimal inverse poles of different leads.

In this chapter the results of the publications [Kov12; FriKov12; LócKov12] are summarized.

3.1 ECG signal generator

Heart functioning is associated with electric activity of the heart over time. This causes electrical changes on the human skin, which is amplified and measured by an ECG device. For this purpose electrodes are placed over the surface of the human body according to a prescribed arrangement. Then the electric voltage between two of them is measured. Each pair is called lead and they detect the same electric heart activity but from a different angle. The 12-lead ECG, in which 12 different electrical signals are recorded, are the most widely used. The role of digital signal processing in ECG analysis is to help cardiologists to detect and diagnose diseases. Compressing, transferring and making automated diagnosis of multi-lead recordings by devices with narrow bandwidth is also important, especially for patients who require 24 hours of observation, for instance Holter monitoring.

Many algorithms have been constructed to analyze, measure and compress ECGs. Generally, there are two different ways to test these methods. The first one is to use large ECG databases such as PhysioNet to compare the algorithms. Unfortunately, the real ECGs are usually affected by several types of noise [FriJan90], which makes it difficult to validate the accuracy of the tested procedures. However, using synthetic realistic ECG signals can be advantageous, especially when we want to compare different signal processing techniques. In contrast to real ECG signals, we can apply and evaluate our methods either on realistic and noise-free, or on distorted signals.

In this section, we generate signals similar to those corresponding to lead II, which is the voltage between the right arm and the left leg. By taking advantage of separating diagnostic and geometrical properties, we may construct a model that is flexible and well parametrized by each heartbeat. Namely, in our interpretation ECG signals are curves with prescribed parameters, including base knots, derivatives, curvature which give a strictly mathematical control over the signal. We note that Clifford, McSharry and their coauthors [McSCli03; CliAzu06; SaySha10] carried out a similar program based on a dynamical model. This method controls the characteristics of the synthesized heartbeat via interwave (PQRST) timings, widths and amplitudes. On the other hand, it is not enough to model all the possible diagnostic distortions of a heartbeat. Our approach is essentially different. We use both diagnostic and geometrical features to generate synthetic ECGs.
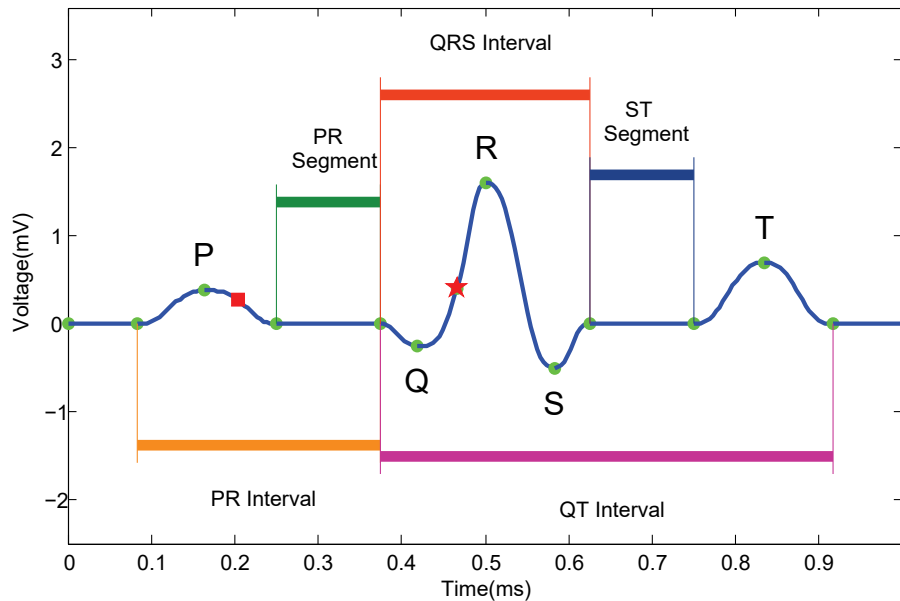
Figure 3.1: Main ECG characteristics of a heartbeat.

ECG morphology

An ECG signal can be segmented into heartbeats. Each heartbeat consists of five standard waves labeled with the letters P, Q, R, S, T. These waves indicate the depolarization and the repolarization phases of heart muscles. Besides, five more interwave timings called PR, PR segment, QRS, QT, ST segment are used. These intervals are indicated on Fig. 3.1. Now we give a brief introduction about the role of these clinical features:

P wave: P waves are usually a low-amplitude feature that represent the depolarization of the atria prior to atrial contraction. They are hard to detect, but important to distinguish various cardiac arrhythmias.

QRS complex: It reflects the depolarization of the ventricles. This is the most significant wave of the ECG due to the large muscle mass of the ventricles. It can be easily detected and it is often used to determine the heart rate.

T wave: The T wave represents the repolarization of the ventricles. It is a recovery phase of the cardiac muscle. The shape of this wave carries a lot of information about cardiac abnormalities, which is why it is important to analyze its geometrical properties such as symmetry, asymmetry and slope.

PR segment: The PR segment is the flat, usually isoelectric segment between the end of the P wave and the start of the QRS complex. Most of the delay in the PR segment occurs in the AV node.

PR interval: It is the time elapsing between the beginning of the P wave and the beginning of the next QRS complex. It reflects conduction through the AV node[1].

ST segment: It represents the period from the end of ventricular depolarization to the beginning of ventricular repolarization. ST level shifts are significant markers of cardiac abnormalities.

QT interval: It represents the time between the start of ventricular depolarization and the end of ventricular repolarization. The QT interval is inversely proportional to heart rate: shortens at faster heart rates and lengthens at slower heart rates.

There is one more feature, the so-called J point that will be used as a control point in our model. The J point has no standard definition. An informal definition in Chapter 10 of [MorBra08] states that "it is the point where the QRS ends and the ST segment begins". A more formal definition in [CarGam02], [Gol06] says that "it is the first point of the inflection on the upstroke of the S wave". In our model the J point is represented by the knot x_{11}, which can be chosen as an inflection point. In addition, we calculate the length of the ST segment from this point. Thus, our model can satisfy both of the aforementioned medical definitions.

Mathematical Model

One way to synthesize the electrocardiogram is to use dynamical models [McSCli03; SaySha10]. In this case the ECG curve represents a trajectory determined by a differential equation related to the appropriate dynamical system. This model has the advantage of well parametrization, robustness and filtering properties. It is also possible to generate long-term recordings such as 24-hour Holter ECGs by taking into account the heart rate variability. Namely, the heart rate may be significantly influenced by sympathetic and parasympathetic[2] activities. For instance, the heart rate increases and decreases during inhalation and exhalation, respectively. This phenomena is called respiratory sinus arrhythmia (RSA). In a dynamical model RSA can be treated and parametrized as well, making the synthesized electrocardiogram more realistic. For proper algorithms and further analysis see e.g., [McSCli03; SaySha10].

---

[1]There are two important nodes in the heart, the sinoatrial and the atrioventricular node. The former is the physiological pacemaker of the heart that is responsible for triggering the heartbeat. The latter is a part of the electrical control system of the heart which has to delay the cardiac pulses, so that the atria can contract completely before the ventricles begin contracting.

[2]Sympathetic and parasympathetic systems are the main parts of the autonomic nervous system. The first one is supposed to prepare the human body for stressful times, sudden loads, to generate energy and to pump more blood to the brain, to the muscles and to the lungs. The second one is responsible for storing energy and enhancing digestion.

Our approach is an interpolation method. More precisely, we employ piecewise polynomial interpolation based on 15 control points. This approach has the following benefits:

- Geometrical parameters of the curve can be adjusted from segment to segment.

- We are able to set diagnostic parameters of the ECG irrespectively of its geometrical parameters.

- It is possible to use error measures based on different ECG features like diagnostic intervals, amplitudes, etc.

Polynomial interpolation

We control one heartbeat from the electrocardiogram through 15 base points. Denoting these points by $x_1, \ldots, x_{15}$, we have to find the spline S that satisfies the following equation:

$$(3.1) \qquad S^{(i)}(x_k) = f^{(i)}(x_k), \quad (k = 1, \ldots, 15;\ i = 0, 1, 2),$$

where $f : \mathbb{R} \to \mathbb{R}$ is a time-varying function that represents the ECG curve. The problem can easily be solved by the classical Hermite interpolation method. The main question to be answered, is how to choose the base points? Of course, they must be correlated with the diagnostic parameters, because they determine the diagnostic intervals and the wave amplitudes of the ECG. For this reason, we use the standard points of the electrocardiogram characterized by special geometrical properties. Fig. 3.1 shows the end points of the segments of a heartbeat, the extrema and the inflection points.

It follows naturally from our model that the endpoints of the diagnostic subintervals and the points of local extrema must be base points. This way 13 control points are set. It turned out that additional base points should be taken if we want to preserve the geometrical properties, such as the curvature of our signal. Then the problem is twofold. On the one hand, we need to find sufficient proper control points, whereas on the other hand, we need to minimize their number in order to keep the model as simple as possible. Our experience showed that certain inflection points play a key role in this respect. In case of ECG signals a typical inflection point of this type appears in the QRS complex. That point is indicated by a star in Fig. 3.1. Yet another control point is necessary in order to describe the P wave. All the possible P wave shapes can be found in [ZigCoh00] and some of them are presented in Fig. 3.2. The corresponding new control point is marked by a square in Fig. 3.1.

After having set the control points, we must specify the order of interpolation at these points. Based on geometric considerations given above, the values of the function, the first and the second derivatives will be described at the inner control points. So in this model we have 15 base points and 3 values associated
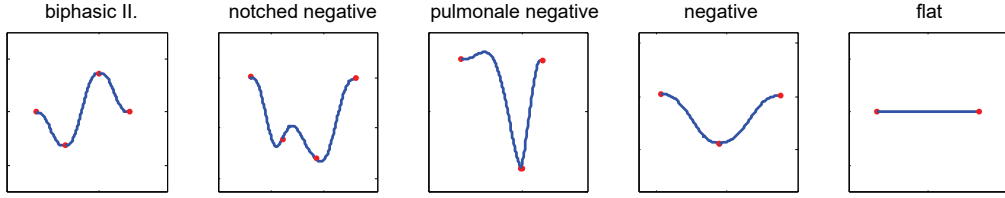
Figure 3.2: The first five shape models for the P wave.

to each of them. It means about 3×15 amplitudes and another 15 positions, which makes it complicated at first sight. However, these parameters cannot be chosen arbitrarily if we want to generate a realistic ECG signal. For instance, the ECG curves are not affected by noise in our model, so we might as well set the baseline to zero. It means that the values at 8 control points are equal to zero. Similarly, the tangents are zero at all the 5 extreme points. Moreover, we can set the range of the amplitudes and the interwave timings automatically. We took the data constraints from Tab. 3.1 in [CliAzu06]. This way the complexity of the problem can be reduced.

We will distinguish diagnostic and geometric parameters. Namely, the 15 diagnostic parameters can be divided into two classes. The first group contains the diagnostic intervals: PR interval, PR segment, QRS, QT, ST segment. Fig. 3.1 was generated by our program and it shows the onsets and the offsets of each interval related to the base points. The second class includes the positions and the amplitudes of the P, Q, R, S, T waves.

Tangents and curvatures are the geometric parameters. They can be set at some but not all of the base points. For instance, the curvature is set only at the extrema, but the tangents can be varied at all of the non-extreme points. Furthermore, at the endpoints, x_1, x_{15}, we demand only interpolation constraints. Thus, the variable parameters are summarized in the list below

$$
\begin{aligned}
S(x_k) &= f(x_k) & (k &= 1, 15), \\
S^{(i)}(x_k) &= f^{(i)}(x_k) & (k &= 2, 4, 5, 6, 8, 11, 12, 14; \ i = 0, 1), \\
S^{(i)}(x_k) &= f^{(i)}(x_k) & (k &= 3, 7, 9, 10, 13; \ i = 0, 2).
\end{aligned}
$$

ECG profiles

Qualifying ECG signal processing methods requires databases with a high amount of test signals. In order to generate such data, we need to vary our parameters. In addition, we have to keep the typical geometrical and diagnostic properties related to the actual ECG class. Therefore, we define the ECG profile by means of the upper and the lower bounds of the parameters. These bounds are typical of certain kinds of cardiac abnormalities. This way we can define profiles for several types of diseases using their medical statistics. Then we can generate different

Table 3.1: Parameters of the ECG model.

	P	Q	R	S	T
Amplitudes (mV)	(0.1,0.2)	0.20	(1,2)	0.25	(0.1,0.5)
Positions	(1/2,20)	(1/4,10)	(2/4,10)	(3/4,10)	(1/2,20)
Tangents	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
Curvatures	(-30,0)	(90,100)	(-100,0)	(-10,100)	(-70,0)
	PR int.	PR seg.	QRS	ST seg.	QT int
Intervals (secs)	(120,200)	(50,120)	(80,120)	(80,120)	(300,430)

realistic ECG signals of a certain type by randomizing the parameters between these bounds, whereas all of them carry the same diagnostic information.

In order to declare a new profile, we should give lower and upper bound pairs including the positions of the base points. The length of the diagnostic intervals and the amplitudes are given in milliseconds and in millivolts, respectively, except the amplitudes of the Q and S waves. Namely, these negative deflections are given in the percentage of the amplitude of the R wave. Tab. 3.1 shows the parameters of a normal ECG profile where the positions in the second row refer to the horizontal alignment of the wave peaks. These values are randomly generated from an I_k ($k = 1, \ldots, 5$) interval for each wave. The first element of the pair sets the center of each I_k, which is related to the right diagnostic interval that contains the wave. For instance, the first number of $(1/2, 20)$ at P means that the center of the interval I_1 has been set to the middle of the P wave. The second parameter sets the length of I_1 to 20 milliseconds. The parameters of the Q, R, S, T peaks are interpreted in a similar way. The constraints of x_4, x_8 (i.e., red star and square in Fig. 3.1) are calculated automatically via divided differences of the adjacent knots x_3, x_5 and x_7, x_9, respectively. We note that at the other control points the tangent and curvature are fixed to be zero in this profile. The correct amplitudes and interwave timings were adopted from Section 3.3 in [CliAzu06]. The proposed algorithm was implemented in MATLAB. A brief introduction of the program with examples can be found in the Appendix B.

3.2   ECG quality measures

Comparison analysis of ECG signal processing algorithms can be performed with respect to various definitions of distortion. There is a wide range of them which are classified into two groups, namely numerical and diagnostic distortions. The former one contains simple arithmetic operations, which provides a fast comparison of different algorithms in the sense of the reconstruction error. On the other hand, these methods are distributing the error equally among all portions of the ECG signal. It is a serious problem, since the ECG segments possess different diagnostic relevance. For instance, the same amount of numerical error in the QRS and in the P wave does not imply the same diagnostic distortion of the ECG. In order to

Table 3.2: Prediction ranges of WDD and PRD.

Measure	Quality groups			
	Very good	Good	Not good	Bad
WDD (%)	0-2.3	2.3-12	12-18	18-40
PRD (%)	0-2	2-9	9-19	19-60

cure this problem, clinical observations are integrated with numerical measures. A detailed review of these methods can be found in [AlF06].

The percent root mean square difference (PRD) is one of the most general error measure:

$$(3.2) \qquad \mathrm{PRD} = \frac{\left\| \mathsf{f} - S_{\mathsf{N}}^{\Theta, \mathbf{a}} \mathsf{f} \right\|_{\mathrm{L}_w^2}}{\left\| \mathsf{f} - \overline{\mathsf{f}} \right\|_{\mathrm{L}_w^2}} \times 100 \,,$$

where $\overline{\mathsf{f}}$ is the mean of the signal, f is the original signal with M samples, $S_{\mathsf{N}}^{\Theta, \mathbf{a}} \mathsf{f}$ is the reconstructed signal. Note that the results are independent from $\overline{\mathsf{f}}$ since it was eliminated from the denominator. Although it is easy to compute the PRD error, this formula neglects the diagnostic features of the electrocardiogram. However, we also have to measure the degree of distortion of the reconstructed signal in terms of the relevant diagnostic information. This is why the so-called Weighted Diagnostic Distortion (WDD) was introduced:

$$(3.3) \qquad \mathrm{WDD}(\beta, \widehat{\beta}) = \Delta \beta^\mathsf{T} \cdot \frac{\Lambda}{\mathrm{tr}[\Lambda]} \cdot \Delta \beta \times 100 \,,$$

where $\Delta \beta$ is the normalized difference between the original and the reconstructed feature vectors β, $\widehat{\beta}$ and Λ is a diagonal matrix of weights. Each feature vector consists of 18 elements which are composed of diagnostic intervals, wave amplitudes and wave shapes. In [ZigCoh00], a qualitative analysis of the WDD measure is also provided. More precisely, a large amount of real ECG data was evaluated by cardiologists and their mean opinion score (MOS) was compared to the WDD and PRD of the corresponding signal. Then, four quality groups were associated to different WDD and PRD ranges, which are summarized in Tab. 3.2. On the one hand, the WDD correlates very well with the MOS of the clinical experts. On the other hand, there is no standard code for practical applications. Besides, the measure is unstable due to the requirement of accurate classification for characteristic features of real ECG signals. In order to cure this problem, we decided to use synthetic ECG data in our experiments.

Experiments

In contrast to the dynamical model [McSCli03], our method can control all the diagnostic features that are required to compute the WDD. So we can test and

Table 3.3: Asymmetric parameters of the ECG model.

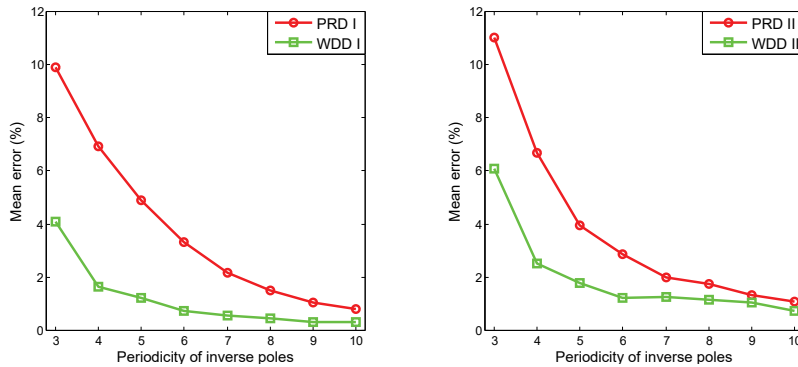|	P	Q	R	S	T
Positions	(3/4,10)	(2/8,10)	(6/8,10)	(7/8,10)	(1/4,10)
	PR	PR seg.	QRS	ST seg.	QT
Intervals (secs)	(130,170)	(40,40)	(80,120)	(80,120)	(300,430)

Figure 3.3: The average PRD and WDD error over 100–100 synthesized signals including normal (left) and asymmetric (right) ECGs.

compare the performance of ECG processing methods on one heartbeat by randomizing the relevant parameters. We performed two tests with different ECG parameters. First, we used our method to generate normal electrocardiograms and test the accuracy of the MT rational approximation. In the second case, we synthesized ECGs with asymmetric waveforms by changing the corresponding parameters of Tab. 3.1 to Tab. 3.3. Note that only 3 inverse poles were used to generate the MT system. However, inverse poles can be repeated periodically to make the approximation more accurate. Each test consists of 100 synthesized ECG signals sampled at 256 Hz and compressed under different periodicity conditions. Comparison of the original and the reconstructed signals in PRD and WDD can be seen in Fig. 3.3. As we expected the MT reconstruction is more sensitive to the asymmetries of the original signal. However, the PRD can be reduced to an acceptable level ($< 5\%$) by repeating the poles periodically at least 5 times. In this case, the compression ratio related to one beat is $36/256$ because we have 15 complex coefficients and 3 complex inverse poles. Moreover, according to Tab. 3.2, the MT reconstruction is of high quality since the WDD is less than 2%.

Conclusion

We have presented another method for generating electrocardiograms that are used to replicate many of the important features of the human ECG. We have shown how to synthesize ECG signals with the same diagnostic but different ge-

ometrical parameters and vice versa. In particular, we have demonstrated that the polynomial approximation can serve as an efficient method for ECG modeling. Depending on the positions of the base points, the procedure can be used to compare the accuracy of different signal processing methods via feature based measures like WDD.

Further research is required on simulating cardiac abnormalities which can serve as a tutorial for medical students. In this case, it is worth extending the model by additional base points and new ECG profiles to describe certain types of diseases. An additional question to be answered is that of synthesizing multilead ECGs.

3.3   QRS modeling

In this section, our aim is to provide a simple mathematical model for explaining and demonstrating the relation between different ECG records (i.e., leads). To this end, the electrode combinations I, II, III, aVR, AVL, aVF will be considered. For these leads the corresponding electrodes are coplanar and therefore we may use a two dimensional model for representing the relation between the records. The model we construct is based on elementary rational functions having a single inverse pole of second order. We show that the same inverse pole turns to be optimal for each of the electrode combinations. To find the optimal inverse pole, we have developed the hyperbolic version of the Nelder–Mead algorithm in Chapter 2. We also show that if we extend the function space by adding the elementary rational functions with the same inverse pole of order one, three and four, etc., then a good approximation of QRS complexes can be given for all records.

The mathematical model of the QRS complex

The QRS complex is of special diagnostic importance in the analysis of ECG signals. In our model they will be represented by means of basic rational functions. Namely, for an inverse pole $a \in \mathbb{D}$ we take the restriction of $r_{a,2}$ onto $\mathbb{T}$ and decompose it into real and imaginary parts

$$r_{a,2}(e^{it}) = U_{a,1}(t) + iU_{a,2}(t) \qquad (t \in \mathbb{R}, a \in \mathbb{D}).$$

Then the QRS complexes will be modeled as proper linear combinations of $U_{a,1}(t)$, $U_{a,2}(t)$, and the constant function $U_{a,0}(t) = 1$. In other words the QRS complexes will be modeled by the elements of the three dimensional linear subspace

$$\mathcal{L}_a := \mathrm{span}\{U_{a,k} : k = 0, 1, 2\}.$$

The role of the constant function is basically normalization. It has no effect on the shape of the curve. Therefore, it has no morphological importance. This means that our model is essentially two dimensional–in other words–a planar one. This modeling of course implies that also the QRS complexes should be interpreted

as real functions defined on an interval of length 2π. Thus, the preprocessing of the QRS signals is necessary, which will be described in the experiments. If the inverse pole a is fixed, i.e., the subspace $\mathcal{L}_a$ is given, then the approximation will be the Fourier projection of the QRS generated functions onto the subspace. To this order it is convenient to have an orthonormal basis in $\mathcal{L}_a$, which is quite easy to obtain in this case. Namely, if $a \in \mathbb{D}$ is given in Euler form $a = re^{i\alpha}$, then by

$$\frac{1}{(1 - \overline{a}z)^2} = \sum_{k=0}^{\infty} (k+1)(\overline{a}z)^k$$

$$= \sum_{k=0}^{\infty} (k+1)r^k(\cos(k(t-\alpha)) + i\sin(k(t-\alpha))) \qquad (z = e^{it})$$

we have

$$U_{a,1}(t) = \sum_{k=0}^{\infty} (k+1)r^k \cos(k(t-\alpha)), \quad U_{a,2}(t) = \sum_{k=0}^{\infty} (k+1)r^k \sin(k(t-\alpha)).$$

Hence it follows immediately that the functions $U_{a,0}$, $U_{a,1} - U_{a,0}$, $U_{a,2}$ are pairwise orthogonal with respect to the usual scalar product in $L^2(\mathbb{T})$. Consequently, this triple forms a basis in $\mathcal{L}_a$. Taking the norm induced by the scalar product in Eq. (1.12), we have

$$\|U_{a,1} - 1\|^2 = \|U_{a,2}\|^2 = \frac{1}{2} \sum_{k=1}^{\infty} (k+1)^2 r^{2k}.$$

We can express it in a closed form by considering

$$\frac{d}{dz} \frac{z}{(1-z)^2} = \frac{1+z}{(1-z)^3} = \sum_{k=1}^{\infty} k^2 z^{k-1}.$$

Namely, by substituting $z = r^2$, we obtain

$$\sum_{k=1}^{\infty} k^2 r^{2(k-1)} = \frac{1+r^2}{(1-r^2)^3}.$$

Hence,

$$N^2(r) := \|U_{a,1} - 1\|^2 = \|U_{a,2}\|^2 = \frac{1}{2}\left(\frac{1+r^2}{(1-r^2)^3} - 1\right).$$

Then the orthonormal basis we will use in $\mathcal{L}_a$ is

$$u_{a,0} := U_{a,0} = 1, \quad u_{a,1} := \frac{U_{a,1} - 1}{N(r)}, \quad u_{a,2} := \frac{U_{a,2}}{N(r)},$$

which can be written in an explicit form as follows

$$
u_{a,1}(t) = \sqrt{\frac{2}{\frac{1+r^2}{(1-r^2)^3} - 1}} \left(\frac{1 - 2r\cos(t-\alpha) + r^2\cos(2(t-\alpha))}{(1 - 2r\cos(t-\alpha) + r^2)^2} - 1 \right),
$$

$$
u_{a,2}(t) = \sqrt{\frac{2}{\frac{1+r^2}{(1-r^2)^3} - 1}} \frac{2r\sin(t-\alpha) - r^2\sin(2(t-\alpha))}{(1 - 2r\cos(t-\alpha) + r^2)^2}.
$$

Then by the Bessel formula we have that the best approximation of a preprocessed QRS complex in the Hilbert subspace $\mathcal{L}_a$ is

$$
d_f(a) = \|f\|^2 - \sum_{k=0}^{2} |\langle f, u_{a,k} \rangle|^2.
$$

This is the error for a fixed inverse pole a. Since we may choose a arbitrarily in the unit disc, the process goes on with minimizing the error function $d_f : \mathbb{D} \to \mathbb{R}$. If the point of minima is $a \in \mathbb{D}$, then the QRS complex will be represented by the corresponding Fourier-projection

$$
f_{QRS} := S_3^u f = \sum_{k=0}^{2} \langle f, u_{a,k} \rangle u_{a,k}.
$$

For this step of the process we have used the HNM algorithm (see e.g., Chapter 2).

Tests and Results

We have used signals of the PhysioNet PTB Diagnostic ECG Database to test our model. More precisely, 77 records of the 52 healthy subjects there, the first 10 heartbeats for each of them, altogether 770 heartbeats were taken.

In order to use our model, we had to transform the QRS complexes to 2π periodic functions. This preprocessing of the QRS complex went as follows. First we used the segmentation program called *ecgpuwave*, which is available as part of PhysioToolkit. For more information about the program, we refer to [GolAma00]. After the segmentation the result was a function restricted onto an interval. The values at the two endpoints were usually not equal. Therefore, we extended the function periodically by applying the program *periodize*, which is a part of the RAIT Matlab toolbox (see e.g., Section 5.1). We performed these steps for the six leads: I, II, III, aVR, aVL, aVF. Then we used the HNM algorithm for finding the best inverse pole for the rational approximation detailed above. During the algorithm the distance between the rational Fourier projection and the QRS function was calculated only for the interval on which the QRS complex was originally supported. This included about a hundred sample points.

In our first test we were interested in the dependence of the best inverse poles for the different leads. For this reason, we compared the 6 inverse poles received from the records of 6 leads of the same heartbeat. We found that the best inverse

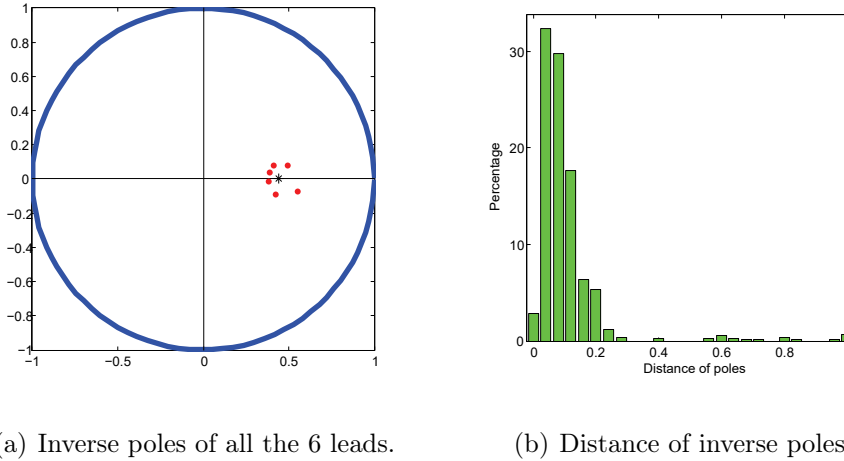(a) Inverse poles of all the 6 leads.   (b) Distance of inverse poles.

Figure 3.4: Statistics of the inverse poles.

poles for the different leads are very close to each other even for real records that contain noise. In Fig. 3.4(a) we show the positions of the inverse poles in the unit circle for the record *s0479* in the PTBDB database. The average of the six inverse poles is marked by $*$. In Fig. 3.4(b) we demonstrate the result of the test performed for all the 770 heartbeats. We calculated the maximum of the distances of the 6 inverse poles from their average. The test shows that the inverse poles are within a 0.1 radius circle for about 65% of the records and within a 0.2 radius circle for more than 95% of them. On the basis of the test, we may consider the inverse pole to be invariant with respect to the leads; therefore we may conclude that the inverse pole is characteristic for the heartbeat itself rather than for the different leads.

After having found the optimal inverse poles, one can calculate the Fourier projection of the QRS complexes onto the corresponding $\mathcal{L}_a$. Then we will receive a simple approximation for the QRS complexes in the 6 leads. We cannot of course expect a high accuracy from this approximation. All we want to demonstrate is the surprising result, i.e., even these simple rational functions show the basic characters of the records. In Fig. 3.5 we took the same records as in Fig. 3.4(a). The green curves correspond to the QRS complexes, and the black curves are the rational approximates. Besides the geometric similarity, we note that the PRD's calculated on the support of the QRS complexes fall between 8.7 and 13 for all leads.

Finally, we were interested in improving the accuracy of the approximation. For this reason, we kept the optimal inverse pole calculated but instead of just using the corresponding elementary rational function of degree 2, we added those of degree 1, 3, 4 and higher. In other words, we took the linear space spanned by the real and imaginary parts of the functions $r_{a,k}(e^{it})$ ($t \in \mathbb{R}$, $k = 0, \ldots, N$, $N \in \mathbb{N}$) where the multiplicity of the single pole is equal to N. Taking an orthonormal basis in this $2N + 1$ dimensional real subspace, the approximation is the Fourier projection of the functions corresponding to the QRS complexes onto the subspace. The degree
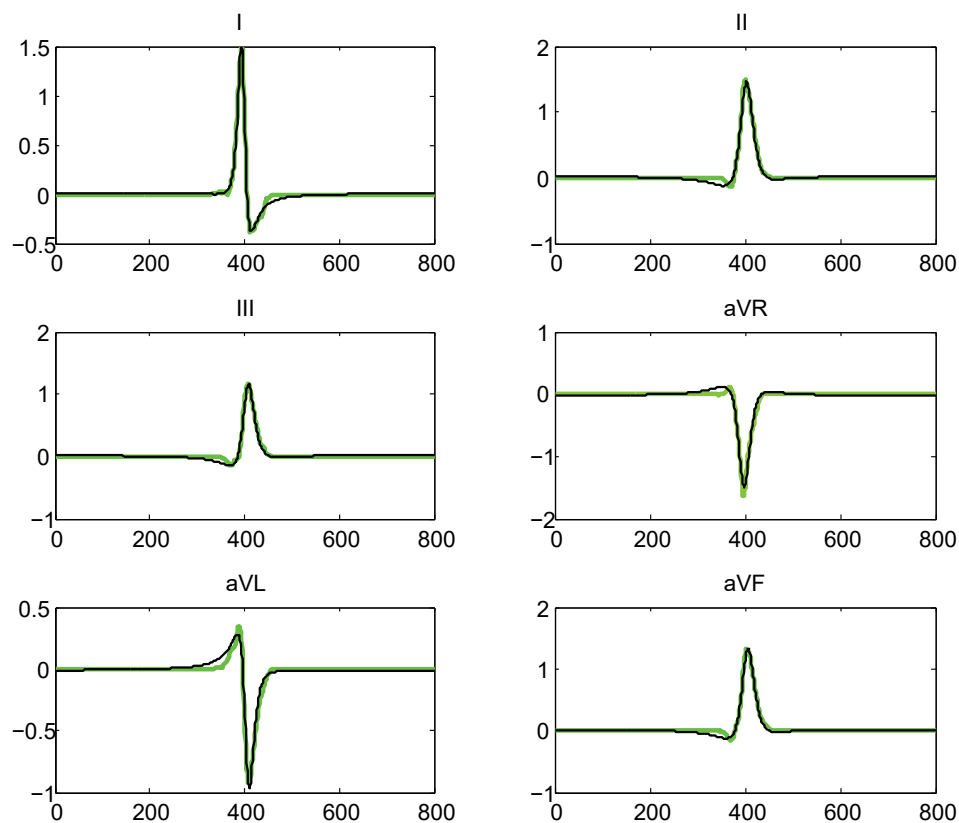
Figure 3.5: QRS approximations of all the 6 leads.

of approximation of the Fourier projection improves by increasing N. At the same time the compression ratio decreases since more and more coefficients should be stored. This relation is demonstrated in Fig. 3.6(a) by taking 5 records, (*s0306lre*, *s0301lre*, *s0324lre*, *s0479_re*, *s0552_re*) from the database. We performed the computation for $N = 12$ for the 770 heartbeats, and calculated the error in terms of PRD. The result is presented in Fig. 3.6(b). It shows that for more than 80% of the records the PRD is not greater than 5%.

We note that this method is related to the theory of discrete Laguerre systems, which are special Malmquist–Takenaka systems (see e.g., [HeuVan05]). Concerning the latest results in this respect, we refer to [FriSch11] and [SchSou11].

Conclusions

We have developed a mathematical model to represent the QRS complexes of an ECG signal. The efficiency of the method is confirmed by experiments. Although we used only one inverse pole, the approximation can be considerably accurate. In addition, the optimal inverse pole of each lead is highly correlated, which reveals the relationship of the 6 channels of an ECG record.
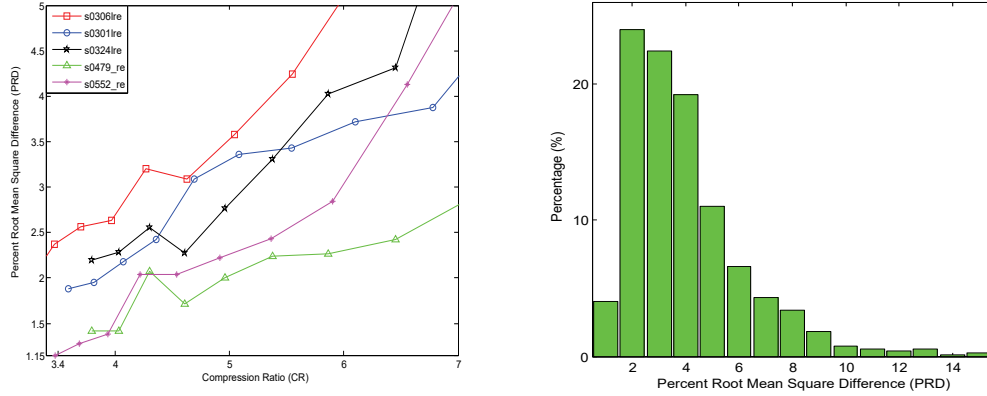
(a) Compression ratio.

(b) PRD of the approximates.

Figure 3.6: Statistics of the representation.

Finally, we would like to remark that the model and the results presented above naturally induce several questions that may initiate further investigations. Let us mention here two as examples.

a) Our model is a planar one, and therefore, can be applied for 6 leads only. Thus, it is worth developing a spatial model in order to demonstrate the relation between all the 12 leads.

b) As it was shown in this section, the best inverse poles for the different leads are characteristic for the heartbeat itself rather than for the leads. So far we have only used a database of healthy subjects. It remains an open question whether the position of the inverse pole could be an indicator for certain malfunctions of the heart.

Chapter 4

EEG signal modeling

In this chapter we examine the problem of epileptic seizure detection in electroencephalography (EEG). One of the challenges is to distinguish rhythmic discharges from non-stationary patterns occurring during seizures. The method developed in this chapter is based on an adaptive and localized time-frequency (t-f) distribution of EEG signals by means of rational functions. The corresponding rational Discrete Short Time Fourier Transform (DSTFT) is a novel feature extraction technique for epileptic EEG data. In order to separate seizure epochs from seizure-free epochs, a Multilayer Perceptron (MLP) classifier is fed by the coefficients of the rational DSTFT. The effectiveness of our method is compared with several state-of-art feature extraction algorithms used in off-line epileptic seizure detection. This comparative study shows that the proposed algorithm outperforms the competing techniques in terms of classification accuracy. What is more, it provides a compact representation of EEG time-series.

The mathematical foundation of the method is summarized in Section 4.1. Section 4.2 contains the construction and the analysis of the algorithm. We note that the HPSO process (see Section 2.2) plays an important role in the feature extraction. The results of the comparative study are provided in the final section.

A significant part of the material presented in this chapter is the result of a collaboration with Kaveh Samiee (Department of Signal Processing, TUT, Tampere, Finland) and was published in [KovSam14] and [SamKov14]. The contribution of this author includes the development of rational DSTFT for EEG signals and the corresponding error and frequency analysis of different t-f representations.

4.1 Discrete Short Time Fourier Transform

Classical STFT in feature extraction

The Fourier transform is a well-known tool for analyzing the frequency distribution of a signal. Unfortunately, the time information is lost in the transform. As a consequence, sudden changes of the signal cannot be localized in time. However,

there is a wide range of applications where both time and frequency information are required. Additionally, processing of the signal via FFT requires the values of the signal f at all sample points, which is impossible in real time applications. Truncating the signal was one of the first idea to solve this problem, which means that the signal is multiplied by a rectangular window $r_A = \chi_{[-A,A]}$. Unfortunately, it is equal to a convolution in the frequency domain by using the function $\mathrm{sinc}_A(t) = \frac{\sin 2\pi A t}{\pi t}$, which is not Lebesgue integrable. In order to replace r_A, a family of window functions was introduced like triangular, Hahn, Hanning, Gaussian, etc. Using this concept, the short time Fourier transform (STFT) of $f \in L^2(\mathbb{R})$ can be introduced as follows

$$\mathcal{F}_g^\epsilon f(t, \omega) = \int_{\mathbb{R}} f(\tau)\overline{g}(\tau - t)e^{-2\pi i \tau \omega}d\tau \quad (t, \omega \in \mathbb{R}),$$

where g is a window function with $g \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ and $\|g\|_{L^2} = 1$. Let us denote the trigonometric system by $\epsilon_\omega(\tau) = e^{2\pi i \tau \omega}$. Then, the STFT becomes a scalar product in $L^2(\mathbb{R})$:

$$\mathcal{F}_g^\epsilon f(t, \omega) = \langle f, g_{t,\omega} \rangle,$$
$$g_{t,\omega}(\tau) = g(\tau - t)\epsilon_\omega(\tau).$$

Using the inner product invariance and the transition property of the Fourier transform, we have the following expressions

$$\mathcal{F}_g^\epsilon f(t, \omega) = \langle \widehat{f}, \widehat{g}_{t,\omega} \rangle,$$
$$\widehat{g}_{t,\omega}(\xi) = \widehat{g}(\xi - \omega)\overline{\epsilon}_{\xi-\omega}(t).$$

Now, the STFT can be interpreted as the projections of f to *t-f* atoms $g_{t,\omega}$. Equations above also show that the spectra $\mathcal{F}_g^\epsilon f$ is localized both in time and frequency around the points (t, ω). For the fundamental results of this field, we refer to [Gab46]. In his paper, Dennis Gabor used a certain type of Gaussian window $g(\tau) = \pi^{-1/4}e^{-\tau^2/2}$ which is a kind of fixpoint of the Fourier transform. He proved two main properties of the STFT; namely, the inversion and the Parseval's formulas, which are analogous to those for the ordinary Fourier transform:

$$(4.1) \qquad f(x) = \iint_{\mathbb{R}^2} \mathcal{F}_g^\epsilon f(t, \omega)g_{t,\omega}(x)d\omega dt,$$

$$(4.2) \qquad \|f\|_{L^2} = \iint_{\mathbb{R}^2} \left|\mathcal{F}_g^\epsilon f(t, \omega)\right|^2 d\omega dt,$$

where Eq. (4.1) is understood in the sense of L^2 norm, i.e.,

$$\lim_{\delta \to \infty} \left\| \int_{-\delta}^{\delta} \int_{-\delta}^{\delta} \mathcal{F}_g^\epsilon f(t, \omega)g_{t,\omega}(x)d\omega dt - f(x) \right\|_{L^2} = 0.$$

Note that Eqs. (4.1)–(4.2) express the fact that f and $\mathcal{F}_g^\epsilon f$ contain the same amount of information in the sense of L^2. One can use this property to analyze the

signal. For instance, in case of epileptic seizure classification, certain frequencies of the signal should be detected first. Then, a proper feature vector should be extracted in order to characterize the seizure. A common feature extraction method is the $M \times N$ block decomposition of the *t-f* domain:

$$(4.3) \qquad F(i, j) = \iint_{\Omega_{ij}} \left| \mathcal{F}_g^\epsilon f(t, \omega) \right|^2 d\omega dt,$$

where $\Omega_{i,j} := [t_i, t_{i+1}] \times [\omega_j, \omega_{j+1}]$ for $0 \leq i < M$ and $0 \leq j < N$. Due to the energy conservation property of the STFT, the feature elements $F(i, j)$'s can be used to detect seizure activity. For instance, a high value of $F(i, j)$ indicates frequency components with high energy in the block $\Omega_{i,j}$. The reader is referred to the review [TzaTsi09], which applies this feature extraction technique to different *t-f* distributions. We also utilized this idea in order to analyze the performance of rational functions in seizure classification.

Rational DSTFT

In Chapter 3 we modeled the ECG signals, which are quasi-periodic waveforms. Thus, it naturally implies that one should choose the window function as $g = r_A$ in order to enclose an entire heartbeat. This can be carried out via segmentation of the signal. Unfortunately, EEGs are more chaotic waveforms and there is no such kind of quasi-periodic property as the ECG, which also legitimates the construction of rational STFT based methods. Let us denote the uniformly sampled $f(t)$ and $g(t)$ functions by $f[n]$ and $g[n]$, respectively. Then, the discrete (D)STFT over the compactly supported g window function can be written as

$$(4.4) \qquad \mathcal{F}_g^\epsilon f[n, k] = \sum_{m=0}^{M-1} f[m] \overline{g}[m - n] \overline{\epsilon}_k[m] \qquad (n \in \mathbb{N}),$$

where $\epsilon_k[m] := e^{2\pi i k \frac{m}{M}}$ $(0 \leq k < M)$, and M is the window size. In our case, M is also equal to the number of equally spaced frequency bins. This algorithm can be interpreted as a successive evaluation of Fourier transforms over short segments of the whole signal (see e.g., Chapter 5. in [Mit11]). Additionally, the frequencies can be visually represented by displaying the squared magnitude of the Fourier coefficients at each segment. The resulting diagram is called the spectrogram of the signal f.

Using the same terminology as in Eq. (4.4), we can define a similar representation of the signal by replacing the trigonometric basis ϵ_k with the elements of the GOB. More precisely, let us consider ℓ different inverse poles $\mathbf{a} = (a_0, \ldots, a_{\ell-1})$ with multiplicities $\mathbf{m} = (m_0, \ldots, m_{\ell-1})$ and the uniformly sampled functions f, g. In order to define a generalized *t-f* representation, we set M to be equal to the sum of the multiplicities. Then the rational DSTFT can be written as

$$(4.5) \qquad \mathcal{F}_g^\phi f[n, k] = \sum_{m=0}^{M-1} f[m] \overline{g}[m - n] \overline{\phi}_k[m] \qquad (n \in \mathbb{N}),$$

with $0 \leq k < M$, where $\phi_k[m] := \Phi_k(e^{2\pi i \frac{m}{M}})$. For a single inverse pole a_0 with multiplicity $m_0 = M$, we can also use the biorthogonal systems $\psi_k[m] := \Psi_{0,k+1}(e^{2\pi i \frac{m}{M}})$ or $\varphi_k[m] := \varphi_{0,k+1}(e^{2\pi i \frac{m}{M}})$. The corresponding inverse transform (see e.g., [Mit11]) can be written in a similar form

$$(4.6) \qquad f[m]\overline{g}[m-n] \approx \frac{1}{M} \sum_{k=0}^{M-1} \mathcal{F}_g^\phi f[n,k]\phi_k[m] \qquad (n \in \mathbb{N}),$$

where $0 \leq m < M$. This procedure can also be interpreted as a windowed Fourier transform, but now we are using different bases. In addition, if we set $a_k = 0$ for all k, then $\varphi_k[m] = \phi_k[m] = \epsilon_k[m]$, so we get back the classical DSTFT as a special case, which can be seen in Figs. 4.2–4.3. We note that the $\approx$ token was used in Eq. (4.6). The main reason for this is that there are no proper inversion formulas for these types of rational systems at the uniform discretization of the unit disk. On the other hand, the perfect reconstruction is possible if we resample the original signal f over an appropriate non-uniform discrete grid defined in Section 1.3.

In order to decrease the dimensionality of the feature vector, we will use only the first N coefficients of the whole spectra at each window. This can be interpreted as a linear dimensionality reduction step. Furthermore, we will apply the HPSO algorithm to determine the optimal inverse pole vector $\mathbf{a_o}$ in Eq. (1.4), which minimizes the reconstruction error for the first N rational Fourier coefficients. Note that it is not possible with the classical STFT since it has no free parameters. These coefficients and the expansions in Eqs. (1.19)–(1.21) will be calculated by using the `biort_coeffs` and `mt_coeffs` commands of the RAIT MATLAB toolbox, which are described in Section 5.1.

Let us consider some examples with the following test signals

$$f_1(t) = e^{-20(t-\frac{1}{5}\pi)^2} \cdot \sin(100\pi t),$$
$$f_2(t) = e^{-20(t-\frac{4}{5}\pi)^2} \cdot \sin(2000\pi t),$$
$$f_3(t) = \sin(2\pi 6^t),$$
$$f(t) = f_1(t) + f_2(t),$$
$$g(t) = \sin^2(\pi t),$$

where $t \in [0, \pi]$ and g is the so-called Hanning window. In our first experiment we apply the rational DSTFT to f which was uniformly sampled at the sampling frequency $f_s = 4000$ Hz. Additionally, we use only one pole a_0 choosing the first $N = 32$ coefficients by setting the window length $M = 128$. Hence, we compute 32 rational components at each windowed segment. For the sake of simplicity, we use the uniform distribution on the real interval $[-1, 0]$ as the poles of the biorthogonal rational function system. Then the absolute square values of the coefficients $\mathcal{F}_g^\psi f[n,k]$ are displayed in Figs. 4.2-4.3. We note that for $a_0 = 0$ we get back the classical DSTFT, where the two frequency components f_1 and f_2 are well separated.
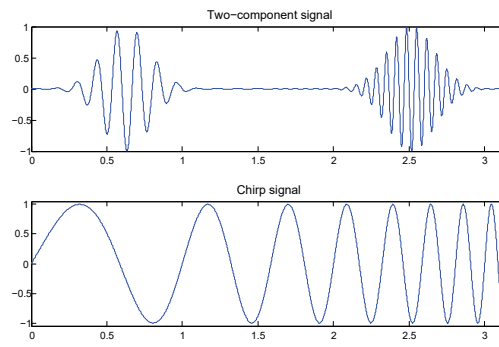
Figure 4.1: Schematics of the test signals $f_1 + f_2$ and f_3.
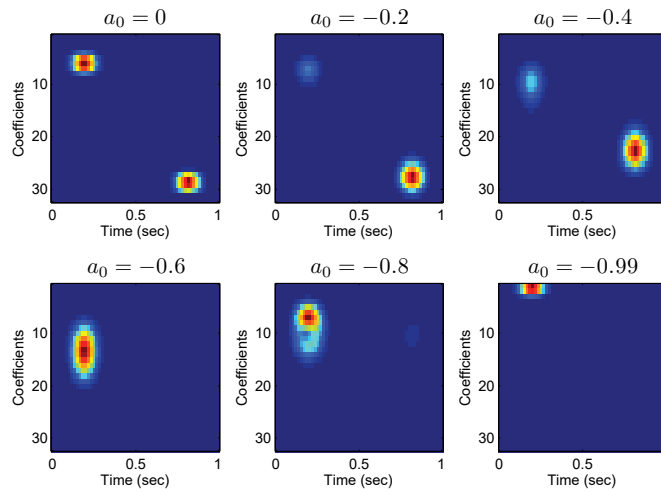
Figure 4.2: Rational DSTFT spectrograms of the function $f = f_1 + f_2$.
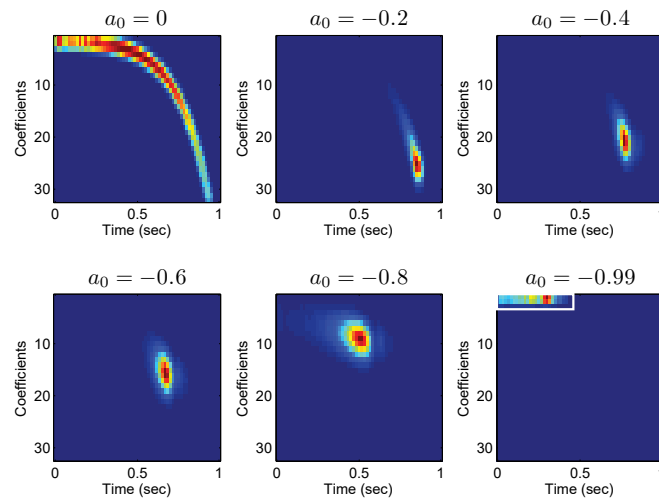
Figure 4.3: Rational DSTFT spectrograms of the chirp signal f_3.

4.2 Seizure classification

EEG is one of the most common techniques used for monitoring brain activities. Generally, expert neurologists analyze the records visually, which is time-consuming. In particular, the noisy characteristics of the EEG recording makes it difficult to separate seizures from artifacts with similar time-frequency patterns. To address this problem, machine learning algorithms have been widely used for automatic detection or prediction of epileptic seizures in raw EEG signals. We note that the raw EEG data may be corrupted by major artifacts caused by muscle activities or eye blinking. Hence, pre-processing of the signal is necessary. Then, feature extraction steps are performed, followed by (un)supervised data mining methods. Lately, a wide range of general signal processing and machine learning algorithms have been adapted for seizure detection and EEG classification. The most important step is to extract discriminative features in the time and/or frequency domains.

For instance Polat and Güneş [PolGün07; PolGün08] used the fast Fourier transform (FFT) based feature extraction to detect rhythmic discharges in epileptic seizures, while Gabor et al. [GabLea96; Gab98] used FFT to calculate the spectrograms in the frequency band 0 to 32 Hz for each EEG channel. Namely, a two dimensional FFT spectrogram of size 8×32 was computed for 8 EEG channels. A Self-Organizing Map (SOM) was then used for classification [Koh89]. However, wavelet transforms are more appropriate for analyzing non-stationary EEG signals since they provide a simultaneous representation in the time-frequency domain. For instance, Adeli et al. [GhoAde07; AdeZho03] applied Daubechies wavelet transform to analyze the epileptic seizure pattern in five frequency sub-bands: delta (0–4 Hz), theta (4–7 Hz), alpha (8–12 Hz), beta (13–30 Hz), and gamma (30–60 Hz). Using these five sub-bands and chaos theory, non-linear dynamics of the EEG are simultaneously quantified by a classifier. Then, the correlation dimension (CD) is used to represent the complexity of the system, and the chaoticity is measured by the largest Lyapunov exponent (LLE) in the five sub-bands. Finally, each EEG signal is classified into three classes: healthy, interictal and ictal. It is also worth mentioning that the special STFT, namely the Gabor transform, has been found effective in epileptic seizure analysis. Chen et al. [CheZha10] described the EEG signal patterns during epilepsy periods by means of the so-called Frequency Band Relative Intensity Ratio (FBRIR). In this method, a filter bank consisting of a number of Gabor filters, with different parameters such as orientations and central frequencies, was used to detect seizure and seizure-free epochs. The advantage of this method is that it works simultaneously on both scalp EEG and ECG and so improves information extraction [NasPou11]. Although EEG and ECG are a one dimensional signals there are various techniques which convert them into two dimensional images. Then the usual 2D feature extraction methods can be applied as well. For instance, Chou et al. [ChoChe06] compressed ECG signals by utilizing JPEG2000 encoding. In case of EEG, Samiee et al. [SamKir15] extracted 2D features in order to classify epileptic seizures. They used the aforementioned 5
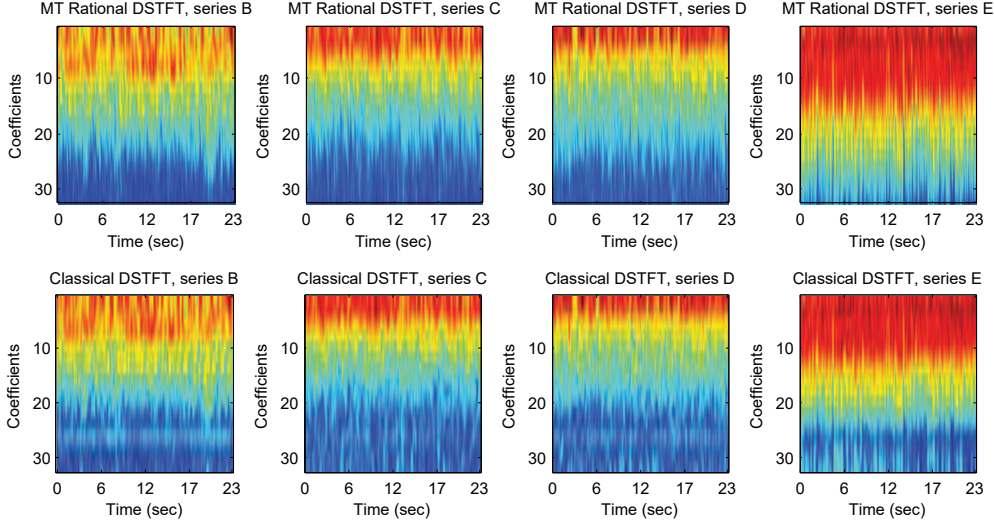
Figure 4.4: Spectrograms of sets B to E for different selection of poles. First row: rational DSTFT spectra for $a_0 = -0.1 + 0.1i$; Second row: classical DSTFT spectra.

sub-bands of 23 channels to convert the EEG into images, which was followed by a five-level Daubechies wavelet (db4) transform.

The proposed feature extraction algorithm is adaptive, scalable and the model is compact in both time and frequency domains. Moreover, we show that–in terms of robustness and efficiency–our method outperforms conventional *t-f* feature extraction techniques commonly used in epilepsy research.

Feature extraction using rational DSTFT

In the present work, we apply the generalized rational DSTFT to EEG time-series of the Bonn dataset [AndLeh01] (see e.g. Section 1.1). The seizure epochs in set E contain low frequency oscillations in the theta frequency sub-band along with high frequency rhythmic spikes and high amplitude changes in the alpha and beta frequency sub-bands. These characteristics of the seizure patterns can be used to distinguish them from seizure-free EEG segments. Thus, we want to achieve a better resolution by increasing the number of coefficients. In this work, we use one inverse pole a_0 with $m_0 = M$ multiplicity. Then we represent each EEG epoch of length M by the first N coefficients of the rational DSTFT. The optimal parameters a_0, M, N will be determined in Section 4.2. The absolute values of the coefficients and five statistical measures are regarded as features: 1. absolute mean value; 2. absolute median value; 3. absolute standard deviation; 4. absolute maximum value; 5. absolute minimum value of the coefficients. One can see an example in Fig. 4.5, where three segments of an EEG signal were approximated by using the first 16 terms of the MT (green line) and the trigonometric (red line) Fourier series. The absolute values of the corresponding coefficients are also displayed.
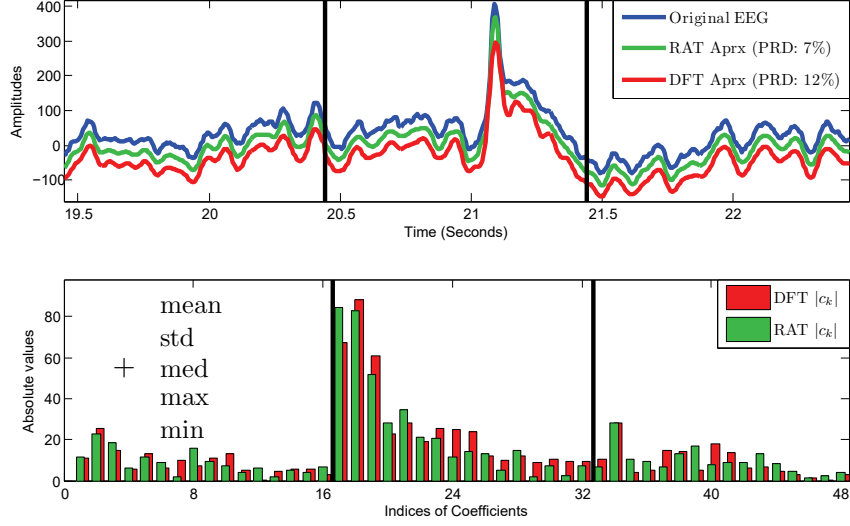
Figure 4.5: Feature elements for a three-seconds-long segment of the record *D003* from the Bonn EEG dataset.

Thanks to the Parseval's formula in Eq. (4.2), the coefficients can reflect neural activity. For instance, there is a spike in the middle segment, which significantly increases the absolute values of the coefficients.

The resulting feature vector contains $N+5$ feature elements for each M sample long EEG epoch. For instance, in Fig. 4.4, one can see the rational DSTFT spectra for all the five sets and the comparison with the classical DSTFT spectra. Here we used the MT system with 32 coefficients and a fixed window size of 0.5 second (i.e., $N = 32$, $M = 88$).

Classification

In order to evaluate the efficiency and the robustness of the proposed feature extraction method, we perform various binary classification tasks. We mainly investigate the detection of epileptic seizure epochs (Set E) in the presence of seizure-free epochs (Sets A, B, C and D). Thus, we define six classification tasks of set E

- (E–A): in the presence of set A;

- (E–B): in the presence of set B;

- (E–C): in the presence of set C;

- (E–D): in the presence of set D;

- (E–A,C): in the presence of sets A and C;

- (E–A,B,C,D): in the presence of sets A, B, C and D.

In all of these classification tasks the rational DSTFT features were tested by using various classifiers such as Naive Bayes, Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Multilayer Perceptron (MLP) architectures. After a series of numerical experiments, the feed forward MLP has been selected as the optimal classifier, which is trained by the back-propagation algorithm. More precisely, the MLP architecture consists of $N + 5$ neurons in the input layer, $(N + 6)/2$ neurons in the hidden layer and 2 neurons in the output layer representing seizure and seizure-free classes. It is worth mentioning that the MLP is one of the most common classifiers used for EEG analysis in the literature. Thus, our experiments show the effectiveness of the proposed feature extraction method utilizing a conventional classifier.

Impact of the inverse pole on the classification accuracy

Once the inverse poles and multiplicities are given, the orthogonal and biorthogonal rational systems are determined. Then the coefficients of the representation can be easily calculated by using the discretization of the scalar product in Eq. (1.12). Although the system is adaptive due to the free parameters provided by the inverse pole(s), it has an effect on the coefficients as well. Namely, the shape of the base functions varies along with the position of the inverse pole a_0. For this reason, the classification results can also change. One of the main interests of this section is to analyze the impact of the position of the inverse pole a_0 on the classification accuracy. More precisely, we compare the classification results of the simple (E–A) task for a set of inverse poles which are chosen from a 50×50 grid of the unit circle. Half of the dataset is used to train the classifier and the other half is used for testing. Additionally, for each one-second-long epoch $M = 176$, we extract the first $N = 16$ coefficients with respect to four different rational systems (MT, RF, MRF and BRF). Then we apply MLP architecture as the classifier, which was described in the previous section.

According to our experiments, the number of zero coefficients increases in the rational DSTFT as the absolute value of the inverse pole increases. This phenomenon is caused by the terms in the rational series expansions in Eq. (4.5) as they represent higher frequencies while the inverse poles tend closer to the torus. These zero coefficients can reduce the accuracy of the (A–E) classification task up to 6%. However, for the inverse poles which are close enough to the center, the classification accuracy is similar to the classical DSTFT, which is a special case of the proposed model for $a_0 = 0$. This can be seen in Fig. 4.6, where the classification results of the four rational systems have been displayed at each inverse pole. During this experiment we extracted different rational coefficients for all EEG records while we used the same inverse pole for every segment. Then the classification accuracy at each inverse pole is indicated by a color code. One can see that the results are similar near the zero inverse pole. Unfortunately, stepping away from zero can degrade the results by introducing more zero coefficients to the rational expansion. All the rational function systems defined in Section 1.3

(a) MT   (b) RF

(c) MRF   (d) BRF
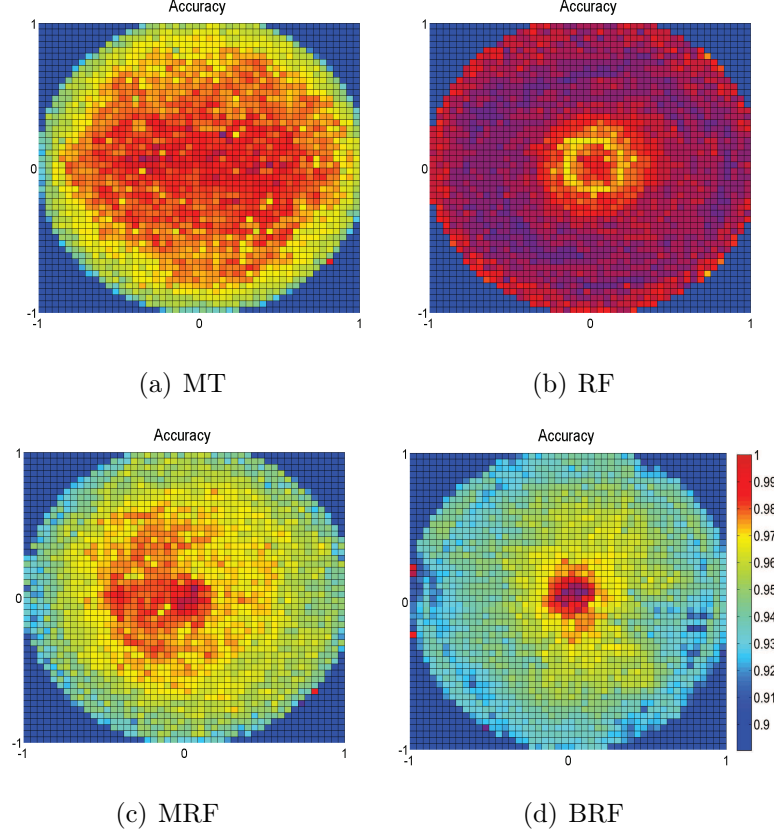
Figure 4.6: Classification results obtained by different rational coefficients for each inverse pole in the 50×50 grid.

share the same behavior because they span the same subspaces of $\mathfrak{R}$ for non-zero inverse poles. Tab. 4.1 shows the average and the highest classification accuracies for each rational system. As it can be seen, MT representation can outperform all the other rational series expansions. Therefore, we will consider only MT rational systems in our further experiments. Moreover, the proposed algorithm can outperform the classical DSTFT technique as well by fixing the inverse pole at $a_0 = -0.1 + 0.1i$. Although the difference is very small, it is just the simplest classification task. We conclude that the performance of the classical DSTFT based seizure detection methods can be improved via rational DSTFT, especially for more complex classification tasks. Furthermore, the results highlight the importance of choosing the right inverse pole as well. For this reason, we decided not to fix the inverse pole for the whole signal. More precisely, an optimal inverse pole was chosen by using the HPSO algorithm in order to minimize the mean square error (MSE) of the reconstructed signal. The purpose of the optimization procedure is to make a compact representation of each segment. Consequently, the coefficients can carry more information and they can be used as features. Flexibility is one of the main advantages of the rational DSTFT comparing to other STFT representations. Namely, the rational systems can be varied from segment

Table 4.1: Comparison of average and best classification accuracies obtained by different rational coefficients in the 50×50 grid.

Rational System	Avg Acc (%)	Best Acc (%) ($a_0 \neq 0$)
MT	**97.9**	**99.4**
RF	94.6	97.3
MRF	96.8	99.2
BRF	94.4	99.3
classical STFT	-	99.0

to segment in contrast with the uniform representations such as classical DSTFT or even the wavelet transform where the shapes of the base functions are fixed for all segments. For this reason, we expect an improvement in the classification accuracies comparing to STFT based features.

Impact of the time-frequency resolution

The classification accuracy of EEG epochs mostly depends on two primary regularizations of the time-frequency transform: 1) time resolution, which is equal to the size of each epoch M, in real applications, it is generally more than 1 second; 2) frequency resolution, which is related to the number of coefficients N (i.e., features). Hence, in case of multichannel EEG data, high-dimensional feature vectors should be avoided. For this reason, we investigate the trade-off between time and frequency resolutions to reduce computational complexity.

In order to find the best choice for the epoch size and the optimal number of coefficients, we consider all the possible combinations for $M = 173, 256, 512, 1024$ samples long segments with $N = 8, 16, 32$ number of MT coefficients. We evaluated the average classification accuracy of the six classification tasks for each combination. As it can be seen in Tab. 4.2, the best average classification accuracy using MT rational DSTFT is obtained for 32 coefficients with 256 sample size window, which is roughly equal to 1.5 seconds. Besides, the highest average Sensitivity (true-positive-ratio) and Specificity (true-negative-ratio) are also obtained for this combination of epoch size and number of coefficients. Moreover, by increasing the size of the epoch, the performance of the MT rational DSTFT is dropped. This phenomenon is due to the fact that only a limited number of base functions were used for the representation. On the contrary, the classical DSTFT operates with an equal number of bases and samples in the epoch (i.e., $N = M$). The former method is an approximation while the latter one is an interpolation procedure. For this reason, rational DSTFT may not be able to fit the signal in the presence of a long epoch and high frequency changes. Hence, we fix the window size to $M = 256$ samples and we extract $N = 32$ coefficients in our further experiments.

Table 4.2: Classification results of MT rational DSTFT for different epoch sizes and number of coefficients.

Classification task		Window size + Number of coefficients											
		173			256			512			1024		
		8	16	32	8	16	32	8	16	32	8	16	32
(E–A)	Sens%	99.5	99.0	99.0	99.9	100.0	99.9	99.5	100.0	100.0	97.9	96.9	100.0
	Spec%	99.1	99.9	99.9	98.1	99.4	99.6	98.4	98.4	99.5	96.6	99.0	97.6
	Acc%	99.3	99.5	99.5	99.0	99.7	99.8	99.0	99.3	99.8	97.3	98.0	98.8
(E–B)	Sens%	92.5	98.9	98.9	99.1	94.9	99.6	98.5	99.8	99.0	99.5	88.1	99.0
	Spec%	98.3	98.4	98.4	94.3	98.9	99.0	95.6	92.8	98.2	91.3	98.6	95.7
	Acc%	95.4	98.7	98.7	96.8	96.9	99.3	97.1	96.4	98.6	95.3	93.5	97.3
(E–C)	Sens%	96.9	96.8	96.8	95.5	97.3	99.3	87.9	99.5	98.3	96.4	94.8	99.5
	Spec%	96.7	97.3	97.3	97.1	99.0	97.7	95.3	93.5	98.4	78.7	91.3	90.8
	Acc%	96.8	97.0	97.0	96.3	98.1	98.5	91.5	96.6	98.4	87.3	93.0	95.0
(E–D)	Sens%	94.5	94.1	94.1	91.3	93.7	95.6	86.0	95.9	96.9	83.4	82.9	90.7
	Spec%	94.3	95.9	95.9	94.5	94.7	94.1	88.6	89.4	92.5	86.0	93.7	92.8
	Acc%	94.4	95.0	95.0	92.9	94.2	94.9	87.3	92.8	94.8	84.8	88.5	91.8
(E–A,C)	Sens%	98.8	98.7	98.7	98.2	99.1	99.2	94.6	99.3	97.8	94.0	97.7	100.0
	Spec%	95.0	96.6	96.6	96.7	97.6	98.9	94.7	97.7	97.5	91.6	92.6	94.1
	Acc%	97.5	98.0	98.0	97.7	98.6	99.1	94.7	98.8	97.7	93.2	96.0	98.0
(E–A,B,C,D)	Sens%	98.3	99.6	99.6	97.4	97.1	99.2	98.3	97.8	98.7	97.3	98.5	98.2
	Spec%	91.6	87.6	87.6	93.7	94.3	93.8	78.7	91.1	91.6	66.8	81.3	88.3
	Acc%	96.9	97.2	97.2	97.4	96.5	98.1	94.4	96.5	97.3	90.8	94.8	96.1
Average	Sens%	96.8	97.9	97.9	96.9	97.0	**98.8**	94.1	98.7	98.5	94.8	93.2	97.9
	Spec%	95.8	96.0	96.0	95.7	**97.3**	**97.2**	91.9	93.8	96.3	85.2	92.8	93.2
	Acc%	96.7	97.6	97.6	96.7	97.3	**98.3**	94.0	96.7	97.8	91.5	94.0	96.2

Table 4.3: Comparison of classification results of MT rational DSTFT coefficients with classical DSTFT coefficients.

Classification task		Feature extraction	
		Classical DSTFT	MT rational DSTFT
(E–A)	Sens%	98.2	99.9
	Spec%	100.0	99.6
	Acc%	99.1	**99.8**
(E–B)	Sens%	98.3	99.6
	Spec%	93.3	99.0
	Acc%	95.8	**99.3**
(E–C)	Sens%	94.0	99.3
	Spec%	98.1	97.7
	Acc%	96.1	**98.5**
(E–D)	Sens%	94.2	95.6
	Spec%	94.5	94.1
	Acc%	94.3	**94.9**
(E–A,C)	Sens%	98.9	99.2
	Spec%	98.9	98.9
	Acc%	97.7	**99.1**
(E–A,B,C,D)	Sens%	97.7	99.2
	Spec%	93.1	93.8
	Acc%	96.8	**98.1**

4.3   Experimental results

In this section, we compare the performance of the MT rational DSTFT with the classical DSTFT in terms of classification accuracy. Based on the results of Section 4.2, we decided to use the optimal inverse pole at each segment by setting $N = 32$ and $M = 256$. Furthermore, we applied the classical DSTFT, which is embedded in MATLAB. For all of the six classification tasks defined in Section 4.2, we consider 50% of the data, chosen randomly, to be the training set and the remaining data to be the test set. More precisely, for the MT rational DSTFT, we extract feature vectors containing 37 elements (32 coefficients + 5 statistical values). For the classical DSTFT, we choose 32 coefficients with the largest absolute values from each 256 sample long epoch. Tab. 4.3 shows the classification results for the six classification tasks. As it can be seen, MT rational DSTFT yields the highest classification accuracy. It means that the proposed method is able to achieve higher sensitivity while keeping the same specificity of the classical DSTFT. Remember that sets A, B and sets C, D and E are recorded using scalp and intracranial electrodes, respectively. However, it has no influence on the performance of the proposed method as it can be seen in Tab. 4.3. Therefore, the classical DSTFT based EEG feature extraction algorithms can be improved by using the proposed rational t-f distribution.

(a) Set A. (b) Set B. (c) Set C.

(d) Set D. (e) Set E.
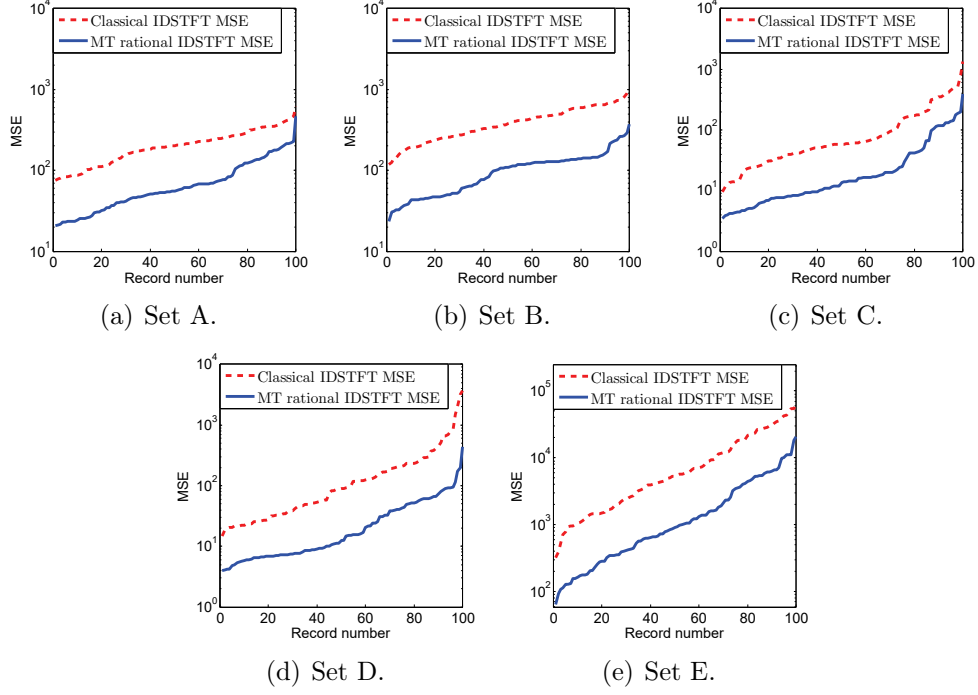
Figure 4.7: Mean square errors of the Bonn Database.

Although we achieved a small amount of enhancement in the sense of classification accuracy, the reconstruction error has been significantly decreased compared to the classical DSTFT. Furthermore, the representation is more compact thanks to the adaptive nature of the rational function systems which can be seen in Fig. 4.7. The inverse discrete (ID) STFT of the signal was computed by using the optimal parameters $N = 32$ and $M = 256$. Then the average mean square error was evaluated for all EEG records of the Bonn dataset. The overall MSEs are as follows: 2.52×10^3 (classical) and 5.5×10^2 (rational). In this sense, the accuracy of our representation is about 4.5 times better. As a consequence, the proposed representation is more robust than the classical DSTFT, i.e., one can achieve a better classification result with our model using a smaller set of coefficients or features. In other words, the proposed method provides a sparse representation of the signal while the components remain orthogonal.

Time-frequency analysis

In quantum mechanics, the well-known Heisenberg uncertainty principle states that the position and momentum of a particle cannot be measured at the same time with arbitrarily high precision. In signal processing, this theorem is referred to as Gabor limit, which can be written as follows

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi}, \tag{4.7}$$

where

$$\sigma_f^2 = \int_{\mathbb{R}} x^2 \, |f(x)|^2 \, dx \,, \qquad\qquad \Delta t = \frac{\sigma_f}{\|f\|_{L^2}} \,,$$

$$\sigma_{\widehat{f}}^2 = \int_{\mathbb{R}} \xi^2 |\widehat{f}(\xi)|^2 d\xi \,, \qquad\qquad \Delta f = \frac{\sigma_{\widehat{f}}}{\|\widehat{f}\|_{L^2}} \,.$$

As a consequence, the higher the level is for frequency resolution, the lower it is for time localization, and vice versa (see e.g., Chapter 1. in [Mal08]).

Although several procedures have been constructed to cure this problem, there is no absolute best among them. Unfortunately, all *t-f* distributions suffer from this problem to some extent. In the previous sections, we have already compared the proposed method with the classical DSTFT, where a particular window g was used for time localization. Here the length of the function g controls the trade-off between the *t-f* resolutions. The Wigner–Ville transform (WVT) was another solution, which does not spread the time or frequency support of sinusoids (see Theorem 4.7. in [Mal08]). Although it gives a very good *t-f* resolution, it can hardly be used in the case of multi-component signals because of the so-called cross terms. However, there is a family of *t-f* representations, which may solve this problem. The Cohen's class of distributions [Coh89] is a generalization of the WVT where most of the cross terms are eliminated using different kernels. The related power spectra can be written in the form

$$\mathcal{C}_g f(t, \omega) = \int_{\mathbb{R}} \mathcal{K}_g f(t, \tau) e^{-2\pi i \tau \omega} d\tau,$$

where $\mathcal{K}_g f(t, \tau)$ denotes the convolution of the autocorrelation and the kernel function $g(u, \tau)$ defined as

$$\mathcal{K}_g f(t, \tau) = \int_{\mathbb{R}} g(u - t, \tau) \overline{f}(u - \frac{1}{2}\tau) f(u + \frac{1}{2}\tau) du \,,$$

$$g(u, \tau) = \int_{\mathbb{R}} \phi(\theta, \tau) e^{2\pi i \theta u} d\theta \,.$$

These distributions have a wide range of applications. Additionally, a recent study [TzaTsi09] has already introduced a feature synthesis for epileptic seizure detection in EEG based on these *t-f* distributions. For the sake of completeness, we also tested the proposed algorithm against 13 Cohen's class distributions with different kernels. In order to extract features from each *t-f* representation, we follow the same procedure as suggested in [TzaTsi09]. More precisely, a grid is constructed for each one-second-long EEG epoch to divide its *t-f* plane into regions. Each region points out a specific frequency sub-band in a certain time window. The frequency sub-bands are: 0–2.5 Hz, 2.5–5.5 Hz, 5.5–10.5 Hz, 10.5–21.5 Hz and 21.5–43.5 Hz. Moreover, we divide the time axis into three equal windows resulting in a total 15 *t-f* regions. Then the energy of the signal is computed in each region using Eq. (4.3) in order to extract feature vectors. Furthermore, the energy

Table 4.4: Comparison of classification results of different *t-f* distributions.

| Distribution | Classification task | | | | | | | | |
| | (E–A) | | | (E–B) | | | (E–C) | | |
	Sens%	Spec%	Acc%	Sens%	Spec%	Acc%	Sens%	Spec%	Acc%
Born-Jordan	71.5	100.0	85.8	71.7	100.0	85.9	71.4	100.0	85.8
Butterworth	72.0	100.0	86.1	72.1	100.0	86.1	71.9	100.0	86.0
Choi-Williams	72.5	100.0	86.3	72.7	100.0	86.4	72.4	100.0	86.3
Generalized	72.2	100.0	86.2	73.0	100.0	86.6	72.0	100.0	86.1
Margenau-Hill (MH)	70.1	100.0	85.1	70.0	100.0	85.1	70.0	100.0	85.1
Pseudo MH	100.0	0.0	49.8	100.0	0.0	49.8	100.0	0.0	49.8
Wigner-Ville (WV)	66.5	100.0	83.3	66.7	100.0	83.4	66.3	100.0	83.3
Pseudo WV	71.5	100.0	85.8	71.6	100.0	85.9	71.5	100.0	85.8
Zhao-Atlas-Mark	66.0	100.0	83.1	66.0	100.0	83.1	66.0	100.0	83.1
Rihaczek	70.1	100.0	85.1	70.0	100.0	85.1	70.0	100.0	85.1
Page	70.1	100.0	85.1	70.0	100.0	85.1	70.0	100.0	85.1
Pseudo Page	98.1	100.0	99.1	97.2	96.1	96.7	97.6	96.6	97.1
Pseudo interference	71.2	100.0	85.7	71.6	100.0	85.9	67.7	100.0	89.1
MT (N=16, M=173)	99.0	99.9	**99.5**	98.9	98.4	**98.7**	96.8	97.3	**97.0**
MT (N=32, M=256)	99.9	99.6	**99.8**	99.6	99.0	**99.3**	99.3	97.7	**98.5**

| Distribution | Classification task | | | | | | | | |
| | (E–D) | | | (E–A,C) | | | (E–A,B,C,D) | | |
	Sens%	Spec%	Acc%	Sens%	Spec%	Acc%	Sens%	Spec%	Acc%
Born-Jordan	67.2	98.6	83.0	66.7	100.0	88.8	74.4	100.0	94.8
Butterworth	68.0	98.6	83.4	66.8	100.0	88.8	68.7	99.4	93.2
Choi-Williams	68.1	98.6	83.4	66.8	100.0	88.8	68.6	99.4	93.2
Generalized	67.3	98.5	83.0	67.7	88.8	89.1	67.8	99.3	93.0
Margenau-Hill (MH)	69.2	98.0	83.7	72.5	100.0	90.8	71.0	99.2	93.5
Pseudo MH	69.0	95.6	82.4	70.4	100.0	90.0	69.6	98.8	92.9
Wigner-Ville (WV)	66.1	99.3	82.8	69.3	100.0	89.7	74.5	98.9	94.0
Pseudo WV	66.7	98.4	82.6	68.6	100.0	89.4	69.7	99.3	93.3
Zhao-Atlas-Mark	60.3	99.1	79.8	67.9	100.0	89.2	71.4	99.5	93.8
Rihaczek	69.2	98.0	83.7	72.5	100.0	90.8	71.0	99.2	93.5
Page	69.2	98.0	83.7	72.5	100.0	90.8	71.0	99.2	93.5
Pseudo Page	71.6	95.4	83.6	70.4	100.0	90.0	73.1	98.7	93.6
Pseudo interference	67.0	98.5	82.8	67.7	100.0	89.1	68.1	99.4	93.1
MT (N=16, M=173)	94.1	95.9	**95.0**	98.7	96.6	**98.0**	99.6	87.6	**97.2**
MT (N=32, M=256)	95.6	94.1	**94.9**	99.2	98.9	**99.1**	99.2	93.8	**98.1**

of the signal over the entire segment is also computed as an additional feature resulting in a feature vector with 16 elements. Finally, the feature vectors are fed into the MLP classifier, which was trained in the same way by using 50% of the data. Tab. 4.4 shows the results of the six classification tasks using different *t-f* distributions and the comparison with two sets of MT rational DSTFT features, one with similar epoch size $M = 173$ and $N = 16$ coefficients and the other one with the optimal setting ($M = 256$ samples and $N = 32$ coefficients) determined in Section 4.2. The classification results of the Cohen's class *t-f* distributions do not differ significantly from each other, except for the Pseudo Page distribution. Furthermore, we could achieve the best accuracy in all the binary classification tasks, and the difference is more than **8.3%** and **3.3%** for the last two classification tasks, respectively. Moreover, the proposed method shows a significant improvement in terms of sensitivity and accuracy for the first four classification tasks. We note that the database is very small compared to long term recordings used in practical applications [GolAma00]. Additionally, our algorithm uses only a few coefficients of the rational decomposition in contrast to *t-f* features in [TzaTsi09], where the sums of several coefficients were used from certain partitions of the power spectra.

Wavelet based power spectra (WT) have been also developed to maintain the trade-off between the *t-f* domains. In this case, the frequency domain is logarithmically scaled by dynamically changing the window size. For this reason, the lower frequencies have fine frequency distribution while the higher ones are well localized in time. It has proved to be a very useful property in many practical applications [Add02]. Although there exist various types of wavelet functions, the decomposition procedure itself generally works with a fixed orthogonal basis. In case of rational functions, both the adaptive representation [FriLóc12] and the non-uniformly spaced frequency scale [SouSch02] are available.

Feature analysis

Since our extracted features are coefficients of the *t-f* distribution, finding the most discriminative elements in the feature vector indicates the frequency range where epileptic seizure patterns are very active. In order to determine the most significant components in the feature vector, we sort the feature elements in a supervised manner via SVM. Using the whole feature vector of the training set, the elements can be ranked by using a recursive leave-one-out procedure according to the weights of the support vectors [GuyWes02]. One can see the top five most significant elements for each classification task in Tab. 4.5. For five classification tasks, the median value of the coefficients is the most discriminative component of the feature vector. In addition, the most common elements in the top five ranking list are the 10th and 11th coefficients. Thus, it is expected that the seizure patterns of set E should be more active in a specific frequency sub-band. In order to illustrate this phenomenon the classical STFT spectrograms of the 10th and the 20th rational components were displayed for sets E and D in Fig. 4.8. As it can be seen, the spectra of set E possesses a higher magnitude for the theta and alpha frequency sub-bands in contrast with the spectra of set D. Furthermore, the power spectra of the 10th rational component $c_{10}\Phi_{10}$ shows a noticeable visual difference in the theta sub-band. Namely, seizure patterns of set E are mostly active in a wider frequency range comparing to set D. The comparison of the power spectra of the 20th rational component $c_{20}\Phi_{20}$ of sets D and E shows that the dominant seizure activity of set E is concentrated in the alpha frequency sub-band. This explains why the 20th coefficient is ranked as the most significant feature element in the discrimination of sets E and D.

Comparison with state-of-the-art methods

As a final performance assessment, we compare our method with other state-of-the-art feature extraction techniques proposed for seizure classification in the literature. These methods include: 1) Wavelet analysis which decomposes an EEG signal into five frequency sub-bands by using second-order Daubechies (db2) wavelet transform. The feature vector consists of the maximum, minimum, average and standard deviation of the wavelet coefficients of each sub-band [GülÜbe05;

Table 4.5: Top five most significant feature elements for each task.

				Classification task		
	(E–A)	(E–B)	(E–C)	(E–D)	(E–A,C)	(E–A,B,C,D)
	36	36	36	20	36	36
# top 5	12	5	18	19	10	10
	10	8	11	21	37	11
	11	7	16	18	12	32
	5	11	19	31	16	9

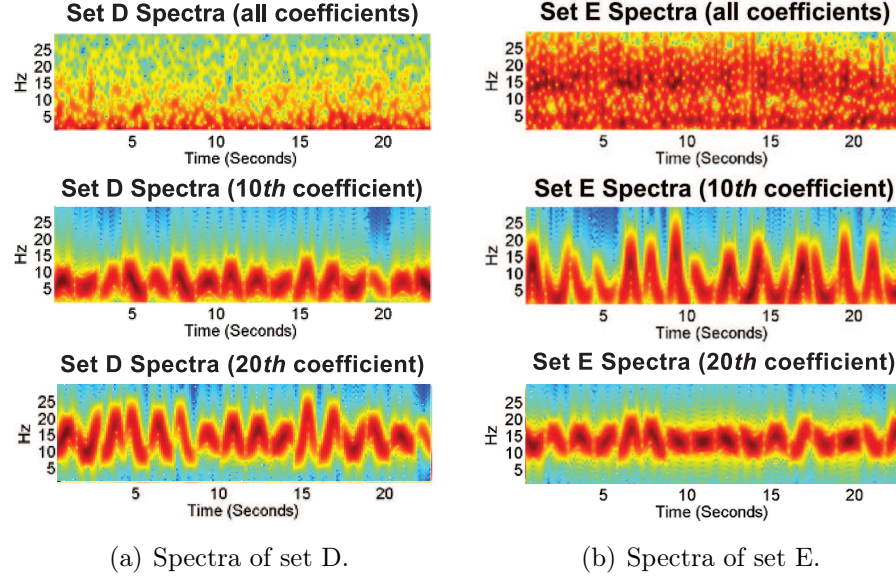(a) Spectra of set D.   (b) Spectra of set E.

Figure 4.8: Spectra of sets D, E by using different number of rational components.

KulPer11]; 2) Entropy based feature extraction using Shannon entropy (SEntr), log-energy entropy and sample entropy [AydSar09]; 3) Spectral analysis applying Welch FFT Power Spectral Density (PSD) [PolGün07]; 4) Approximate entropy (AEntr) based feature extraction [SriEsw07]. For all of these methods, the same MLP classifier was used by feeding 50% of the data in the learning phase. As it can be seen in Tab. 4.6, our method is mostly ranked as the first or the second best approach in terms of classification accuracy. It can outperform both the entropy and the approximate entropy based feature extraction methods in all classification tasks. Besides, due to consideration of all coefficients in Welch PSD based method, a large dimensional feature vector is obtained in contrast to our method, which uses only $32 + 5$ features. This is why it is faster to train the classifier by the proposed compact features. Our approach is ranked first together with the wavelet based method on average. The high accuracy level obtained by the proposed method in all the six classification tasks demonstrates its ability to discriminate between epileptic seizure and seizure-free epochs in the presence of imbalanced EEG data.

Table 4.6: Classification results obtained by different feature extraction methods using MLP classifier.

Classification task		Feature extraction				
		Wavelet	SEntr	Welch	AEntr	MT RDSTFT
(E–A)	Sens%	100.0	92.1	98.7	87.3	99.9
	Spec%	100.0	99.9	100.0	89.6	99.6
	Acc%	**100.0**	96.0	99.4	88.4	99.8
(E–B)	Sens%	99.2	90.5	98.2	91.1	99.6
	Spec%	99.1	99.4	100.0	94.0	99.0
	Acc%	99.2	94.9	99.1	92.6	**99.3**
(E–C)	Sens%	98.7	89.6	92.3	0.0	99.3
	Spec%	97.9	97.6	99.9	100.0	97.7
	Acc%	98.3	93.6	96.1	50.3	**98.5**
(E–D)	Sens%	95.6	85.2	86.7	2.8	95.6
	Spec%	97.4	94.4	99.6	99.4	94.1
	Acc%	**96.5**	89.8	93.2	51.3	94.9
(E–A,C)	Sens%	97.5	93.6	98.3	0.0	99.2
	Spec%	99.7	98.0	99.7	100.0	98.9
	Acc%	99.0	96.5	**99.3**	66.3	99.1
(E–A,B,C,D)	Sens%	96.0	90.3	95.0	0.0	99.2
	Spec%	98.7	96.7	98.7	100.0	93.8
	Acc%	**98.1**	95.5	97.9	79.8	**98.1**

Finally, we compare the proposed algorithm with the results of some state-of-the-art epilepsy detection methods reported in the literature using the same dataset. We consider the classification accuracies of the same classification tasks, which are summarized in Tab. 4.7. Compared to the others, our method achieved the best results with a significant improvement for the classification tasks (E–B) and (E–C). It is ranked second best for the classification task (E–D), where the difference is about 0.6%. In addition, the proposed method is close to the best scores in terms of the highest classification accuracy, where the difference is 0.2% and 2% for the classification tasks (E–A) and (E–A,B,C,D), respectively. The deviation is caused by the high variance between the inverse pole(s) and the coefficients. More precisely, in each segment the optimal inverse pole is selected via the hyperbolic modification of the basic PSO algorithm (see e.g., Section 2.2). Furthermore, altering the inverse pole(s) changes the base functions as well. This procedure can affect the variance of the extracted coefficients, which can cause variations in the classification results. In order to address this problem, a more sophisticated classification approach such as ensemble learning with bagging techniques could be considered. Such techniques may reduce the variance between the inverse poles and coefficients by using, e.g., bootstrap replicates.

Table 4.7: Comparison of the proposed algorithm with state-of-the-art methods for the same classification tasks performed on the Bonn EEG dataset.

Author(s)	Classification	Accuracy (%)
Srinivasan et al. [SriEsw05]	(E–A)	99.6
Subasi [Sub07]	(E–A)	95
Parvin Kumar et al. [PraSri10]	(E–A)	96 – 99.8
Guo et al. [GuoRiv09]	(E–A)	95.2
Nicoletta and Georgiou [NicGeo12]	(E–A)	93.5
Xie and Krishnan [XieSha08]	(E–A)	100
Kaya et al. [KayUya14]	(E–A)	99.5
Proposed method	(E–A)	99.8
Nicoletta and Georgiou [NicGeo12]	(E–B)	82.9
Proposed method	(E–B)	99.3
Nicoletta and Georgiou [NicGeo12]	(E–C)	88.0
Proposed method	(E–C)	98.5
Nicoletta and Georgiou [NicGeo12]	(E–D)	79.9
Kaya et al. [KayUya14]	(E–D)	95.5
Proposed method	(E–D)	94.9
Tzallas et al. [TzaTsi07]	(E–A,B,C,D)	97.7
Ocak [Oca08]	(E–A,B,C,D)	96.2
Sheng-Fu Liang et al. [SheCha11]	(E–A,B,C,D)	97.8 – 98.5
Guo et al. [GuoRiv10]	(E–A,B,C,D)	98.3
Xie and Krishnan [XieKri13]	(E–A,B,C,D)	100
Proposed method	(E–A,B,C,D)	98.1

Conclusions

The main contribution of this section is to provide an effective and novel feature extraction algorithm for epileptic seizure detection in EEG signals. We proposed the rational DSTFT, which is an adaptive generalization of the classical DSTFT. Additionally, we described the epileptic seizure patterns in the *t-f* domain by using different types of rational functions. In order to increase the efficiency, we investigated both the optimal number of coefficients and the best window size. Moreover, we analyzed the proposed features in order to determine the most significant ones. The performance of different rational systems was compared with the classical DSTFT in terms of classification accuracy. Furthermore, we increased the adaptivity of our method by applying the HPSO algorithm to determine the optimal position of the inverse pole for each EEG epoch. For this reason, the proposed method gives a compact *t-f* representation of the signal. We also showed that the inverse rational DSTFT has a smaller mean square error compared to the classical DSTFT for the same number of non-zero coefficients. Finally, we compared the performance of our feature extraction procedure with 13 Cohen's class *t-f* distributions. Several state-of-the-art feature extraction methods were taken into account, which are commonly used in epileptic seizure detection and

classification. The comparative study shows that the proposed rational DSTFT can be used as a robust feature extraction method in off-line epilepsy analysis and detection. In addition, the proposed algorithm utilizes results from the recently developed theory of signal models for rational orthogonal systems. Experiments confirmed that the MT rational DSTFT could be successfully applied to perform epileptic seizure pattern detection in EEGs.

We note that the adaptability of the system could be further improved by using multi-dimensional HPSO to determine the optimum number of unique inverse poles. MDHPSO can explore multi-dimensional search spaces while it will converge to the best inverse pole configuration in the optimal dimension.

Taking advantage of the adaptive behavior of rational representations, we expect that our method will perform well on longer EEG recordings. Hence, future work will focus on the use of rational DSTFT in multichannel long-term epileptic seizure detection in order to study the proposed method in the presence of high frequency EEG activities (i.e., high gamma 50 Hz) as it was suggested in [WorPar04]. For this purpose, we shall test our future algorithms on larger datasets such as Physionet [GolAma00], which contains more variability between seizure and seizure-free epochs. Furthermore, using long-term EEG recordings, we can evaluate the performance of future methods for real-time seizure detection by measuring the delay of detected seizures onset.

Chapter 5

Numerical methods

In this chapter we develop numerical methods for computing rational series expansions of discrete signals. They are related to and applied in the algorithms presented in the previous chapters. Section 5.1 provides subroutines for evaluating Blaschke functions, MT and biorthogonal rational systems along with continuous and discrete implementations. The complex and real valued interpretations of Section 1.3 are included as well. We also made interactive graphical user interfaces (GUI) for visual demonstration of these algorithms, which can help users to understand the contribution of certain parameters such as poles, multiplicities, etc. Implementing GUIs requires only technical competence; therefore, we give only a brief introduction to their functionality and a few examples in Appendix B. We note that the so-called RAIT toolbox was implemented in collaboration with Levente Lócsi. Here, we present only our contribution to this project, which includes the implementations of the real valued MT systems, of the continuous and of the discrete biorthognal systems, and signal processing applications with experiments on ECG recordings. In this section we summarize the results of the following publications [KovLóc12a; KovLóc12b].

In Section 5.2, we introduce a fast algorithm to compute the non-uniform discretization points for discrete rational orthogonal and biorthogonal systems. In order to do that, we need new concepts for constructing an effective numerical solution. Namely, a good estimation process is developed for the sampling points based on the monotonic behavior of the argument function. Then, a sequence of fixed-point iteration is executed starting with appropriate initial values. Furthermore, we perform tests for the convergence of some root-finding algorithms in order to achieve the best accuracy. These methods are compared in terms of evaluation time and step number. Finally, we present a parallel implementation in CUDA which can further improve the execution time. The main material in this section is taken from [KovVad13]. The contribution of this author is the proposed algorithm with experiments while the parallel implementation was made in collaboration with Viktor Vad.

In the final section, we give a short review of different ECG compression methods, and provide a comparative study for these methods. We show that the ratio-

nal representations outperform B-splines, discrete wavelet transforms and wavelet packets in several respects.

5.1   RAIT MATLAB toolbox

In this section, we present a general MATLAB library for rational function systems and their applications. The collection of these algorithms can be found in the rational approximation and interpolation toolbox (RAIT) [KovLóc12a] together with usual signal processing techniques adopted to rational functions. The procedures of this library can be divided into 6 classes, which are listed in Tab. 5.1. We have different types of rational functions, namely the elementary rational functions (RF), the biorthogonal (BRF), and the MT systems. Three routines were implemented for each system to compute the values, the coefficients and the projection onto $\mathfrak{R}_N^{\mathbf{a}}$. The names of the functions abbreviate the type of the system and the related operation. For instance, if we want to calculate the coefficients of a function with respect to the discrete complex biorthogonal system, then the abbreviation `biortdc_coeffs` should be used. We introduce the parametrization of these procedures via the continuous BRF system. All functions of the first three classes can be called in the same way.

- `biort_system(M,ipoles)`. Calculates the values of the BRF system at M number of uniformly distributed points on $\mathbb{T}$. The vector of different inverse poles $\mathbf{a} := (a_0, \ldots, a_{n-1})$ and multiplicities $\mathbf{m} := (m_0, \ldots, m_{n-1})$ are determined from `ipoles`, automatically. The program returns with a matrix where the kth row contains the values of the kth member of the BRF system. Namely, let us denote the largest index by $j(k)$ for which $N_{j(k)} := m_0 + \ldots + m_{j(k)-1} < k$. Then the matrix $\mathbf{\Psi} \in \mathbb{C}^{N \times M}$ can be defined as follows:

$$\mathbf{\Psi} = \left[\Psi_{j(k),\ell(k)}(e^{it_m}) \right]_{1 \le k \le N,\ 0 \le m < M},$$

  where $t_m = 2\pi m/M$, $\ell(k) := k - N_{j(k)}$ and $N = m_0 + \ldots + m_{n-1}$.

- `biort_coeffs(s,ipoles)`. Computes the BRF–Fourier coefficients of the discrete signal `s` with respect to the inverse poles `ipoles`. The implementation is based on the simple matrix-vector product form of orthogonal systems. The program returns with the coefficient vector $\mathbf{c} \in \mathbb{C}^{N \times 1}$ and the ℓ^2 error e of the approximation:

$$\mathbf{c} = \frac{1}{M}\mathbf{\Psi}\mathbf{s}, \qquad e = \|\mathbf{s} - \widetilde{\mathbf{s}}\|_2,$$

  where $\widetilde{\mathbf{s}} \in \mathbb{C}^{M \times 1}$ denotes the reconstructed signal. In order to get the right coefficients, $\mathbf{s}$ must be an element of $H^2(\mathbb{D})$ (see e.g., the function `addimag` on page 77).

- `biort_generate(M,ipoles,c)`. Computes the linear combination of the corresponding BRF system by using the coefficient vector `c`. The program returns with the related element of $\mathfrak{R}_N^a$ where `N` is the number of inverse poles in `ipoles`:

$$\widetilde{\mathbf{s}} = \boldsymbol{\Psi}^* \mathbf{c},$$

where $\boldsymbol{\Psi}^*$ is the conjugate transpose of $\boldsymbol{\Psi}$. Note that the values of the reconstructed signal $\widetilde{\mathbf{s}}$ are computed at the same grid $t_m = 2\pi m/M$ ($0 \leq m < M$).

The Blaschke class includes the procedures that implement the derivatives and the inverse of the argument function β_a. Additionally, the visualization of the Blaschke function also belongs to this library:

- `blaschkes_img(I,a)`. Transforms an image `I` of size $N \times N$, applying the Blaschke function defined by the inverse pole `a`. To this end, we first associate each pixel to a complex number $z \in \mathbb{D}$. Then the real and imaginary parts of $B_a(z)$ determine the new coordinates of the pixels (see e.g., Alg. 1). The program returns with the Blachke transformed image B, the absolute values $|B_a(z)|$ and the arguments $\angle B_a(z)$ of the projection. One can see an example in Fig. B.1, which depicts the Blaschke transformed Lena image.

The problem of calculating the projection onto $\mathfrak{R}_N^a$ is twofold. On the one hand, we need to find the best poles for the selected rational system to minimize Eq. (1.4). On the other hand, we need to calculate the projection itself. The latter can be managed by using functions of the first three classes, whereas the former can be solved by applying different types of optimization methods. To this end, the (Euclidean) Nelder–Mead simplex algorithm was applied for rational functions in [Lóc09]. Later, we adopted this method to the real valued MT system and to the complex continuous BRF functions as well.

- `simplex_biort(s,mult,period,init,show,eps)`. Gives the inverse poles of the continuous BRF system that best fits the approximation of the signal `s`. The parameters `mult` and `period` determine the multiplicities and the periodicity of the inverse pole vector. Initial values are defined by the `init` variable. Furthermore, the predicted inverse poles can be displayed at each step due to the `show` parameter. We can also control the accuracy of the process through the `eps` variable. All the simplex algorithms are called consistently by using the right abbreviation of the rational system, such as `mtdc`, `mtdr`, `biort`, etc.

There are many procedures that perform conversations between systems, visualize rational functions, etc. We only present the most important methods below, but additional examples can be found in Appendix B.

- `addimag(f)`. All the rational–Fourier expansions operate on functions that are analytic on the unit circle. Unfortunately, most of the real valued signals

do not satisfy this constraint. To this end, we eliminate the Fourier coefficients with negative indices via $F = f + i\mathcal{H}f$ (cf., the `hilbert` command in MATLAB). In this case, $F \in H^2(\mathbb{T})$ and the BRF approximation of the signal f is equal to $\mathrm{Re}\, S_N^\Psi F$. Note that every procedure of the first three class in Tab. 5.1 can be used in the same way.

- `coeff_conv(M,ipoles,coeffs,base1,base2)`. It converts the coefficients `coeffs` between the continuous systems `base1` and `base2`. Parameters `M` and `ipoles` define the number of samples and the inverse poles, respectively. From Section 1.3, we know that the rational functions MT, MRF and BRF span the same subspaces of $\mathfrak{R}$ for a given vector of inverse poles $\mathbf{a}$ and multiplicities $\mathbf{m}$. As a consequence, the change of basis can be easily calculated via Gram matrix transforms. For instance, if we have the MT coefficient vector $\mathbf{c} \in \mathbb{C}^{N \times 1}$, then the coefficients $\mathbf{c}' \in \mathbb{C}^{N \times 1}$ of the corresponding BRF system are as follows

$$\mathbf{c}' = \frac{1}{M}\mathbf{\Psi}\mathbf{\Phi}^*\mathbf{c} \qquad (\mathbf{\Phi},\ \mathbf{\Psi} \in \mathbb{C}^{N \times M}).$$

A similar function called `coeffd_conv` is also available to perform conversations between discrete orthogonal rational systems.

- `discretize_dr(ipoles,eps)`. Computes the non-equidistant discretization for the real valued MT system. The inverse poles are given in `ipoles` and the accuracy of the method can be set due to the variable `eps`. Recall that from Section 1.3 the first element of `ipoles` should be equal to zero. Here, the set $\mathbb{T}_n^R$ should be determined (cf., $\mathbb{T}_n$ in Eq. (1.26)). Using the kernel function of the real MT system, one can show that the discretization points can be computed by inverting the following expression

$$\eta_{\mathbf{a}}(t) = \theta_{\mathbf{a}}(t) - \frac{t}{2n} \qquad (t \in \mathbb{R}).$$

If $a_0 = 0$, the monotonic property of $\theta_{\mathbf{a}}$ still holds for $\eta_{\mathbf{a}}$ as well. Thus, there is only one $\tau_k^R \in [\tau_0^R, \tau_0^R + 2\pi)$ for every $k = 0, 1, \dots, 2n - 2$ which satisfies the equation

$$\eta_{\mathbf{a}}(\tau_k^R) = \eta_{\mathbf{a}}(\tau_0^R) + \pi\frac{k}{n}.$$

Then, the discretization points of the real valued MT system can be written as

$$\mathbb{T}_n^R := \left\{ e^{i\tau_k^R} : 0 \leq k \leq 2n - 2 \right\}.$$

Note that the program automatically insert a zero inverse pole into `ipoles`, so the first condition in Eq. (1.22) is satisfied. This implementation utilizes the bisection method based on the enhanced order of the discretization points (see e.g., [Lóc11]). Section 5.2 provides another numerical solution to this problem.

Table 5.1: References of all functions in the toolbox. Bold names indicate the procedures implemented by this author.

Biort_sys	Rat_sys	MT_sys
biort_system	lf_system	mt_system
biort_coeffs	lf_generate	mt_coeffs
biort_generate	**mlf_system**	mt_generate
biortdc_system	**mlf_coeffs**	**mtdc_system**
biortdc_coeffs	**mlf_generate**	**mtdc_coeffs**
biortdc_generate	**mlfdc_system**	**mtdc_generate**
	mlfdc_coeffs	**mtdr_system**
	mlfdc_generate	**mtdr_coeffs**
		mtdr_generate

Blaschke	Simplex	Other
arg_der	simplex_mt	addimag
arg_fun	simplex_mtdc	bisection_order
arg_inv	**simplex_mtdr**	**coeff_conv**
arg_inv_anim	**simplex_biort**	**coeffd_conv**
argdr_fun	**simplex_biortdc**	discretize_dc
argdr_inv	coords2params	**discretize_dr**
blaschkes	coords2params_all	**dotdc**
blaschkes_img	**multiply_poles**	dotdr
	periodize_poles	**kernel**
		subsample
		periodize
		norm_sig

In order to improve the accuracy of the approximation, we have to transform the signals to 2π periodic functions, i.e., the values at the end points should be equal. We give two solutions to this problem.

- `periodize(s,alpha)`. We treat the discontinuity near the endpoints by using the extension of the original signal `s`. The preprocessing steps are carried out as follows. First, we apply the well-known Savitzky–Golay FIR smoothing filter applying second order polynomials width frame size equal to 11 samples. It is followed by the prediction of the signs of the slopes in the smoothed data. Then, we extend the signal at the two endpoints by keeping the slopes. Then a Tukey window is applied to generate a 2π periodic signal. The parameter `alpha` controls the length of extension.

(a) Projection onto $\mathfrak{R}_N^a$.
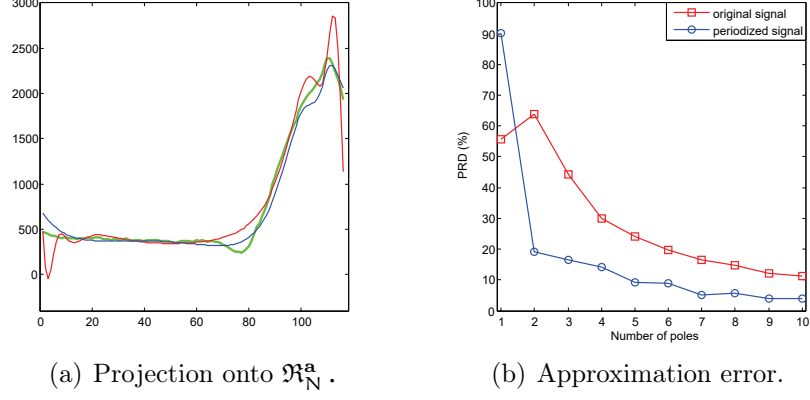
(b) Approximation error.

Figure 5.1: Approximating real signals using the BRF system by taking the third lead of the record *s0301lre* from the PhysioNet/PTBDB database [GolAma00].

- `norm_sig(s)`. It is a simple procedure which removes the linear approximation from the discrete signal of length M:

$$\widetilde{\mathbf{s}}[m] := \mathbf{s}[m] - \left(\mathbf{s}[0] + m \frac{\mathbf{s}[M-1] - \mathbf{s}[0]}{M-1} \right) \qquad (0 \leq m < M).$$

Note that it is important to take care of such discontinuities of the signal; otherwise the reconstruction error will be high. For instance, Fig. 5.1 shows the original signal (green) and its approximations with (blue) and without (red) periodization. The effect of jump discontinuity can be seen as the red colored function starts to oscillate near the ends of the interval.

Conclusion

A new MATLAB toolbox has been introduced together with signal processing methods which can be useful in a wide range of applications. For instance, these systems are capable of representing different types of discrete-time series. Both equidistant and non-uniform discretizations can be used. Four types of signal representations are available due to different classes of rational function systems. Moreover, two MATLAB GUIs were implemented for educational purposes.

5.2   Discretization problem

In this section, we provide a fast algorithm to compute the non-uniform grid in Eq. (1.26). Let us consider a vector of different inverse poles $\mathbf{a} := (a_0, a_2 \ldots, a_{n-1}) \in \mathbb{D}^n$, multiplicities $\mathbf{m} := (m_0, \ldots, m_{n-1}) \in \mathbb{N}_+^n$ and $N = m_0 + m_1 + \cdots + m_{n-1}$. Furthermore, we define the inverse pole vector $\mathbf{b} \in \mathbb{D}^N$ following Eq. (1.20). Then, a non-uniform discretization can be defined via the inverse of the corresponding argument function:

$$(5.1) \qquad D_{\mathbf{b}}^M = \left\{ \tau_k \in [-\pi, \pi) : \tau_k = \theta_{\mathbf{b}}^{-1}(t_k), \ t_k \in D_{\mathbf{0}}^M \right\},$$
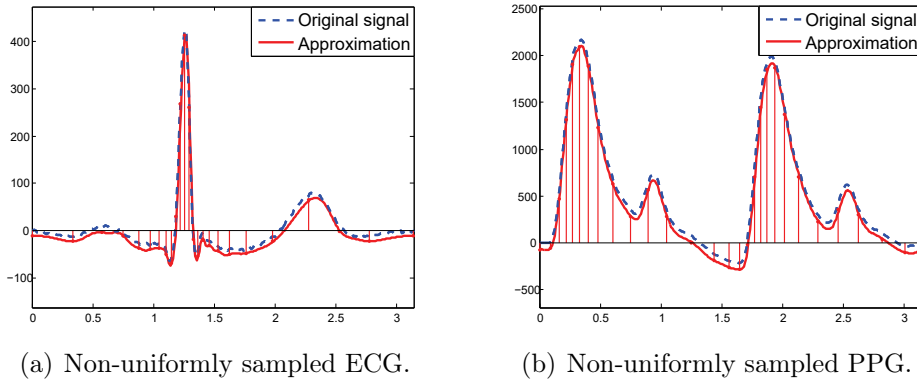
(a) Non-uniformly sampled ECG.

(b) Non-uniformly sampled PPG.

Figure 5.2: Rational function approximation of real signals.

where

$$D_0^M = \left\{ -\pi + 2\pi \frac{k}{M} : 0 \le k < M \right\}.$$

In addition, these irregular sampling points can be identified by $\mathbb{T}_N$ in Eq. (1.26) for $M = N$.

We note that the non-uniform distribution of the unit circle was first introduced by Schipp et al. [SchBok98; SouSch02]. Although there exist a few numerical methods [Lóc11; KovLóc12a] to solve this problem, the efficiency of these algorithms is inadequate for large M. Let us consider an example in Fig. 5.2 where we approximated a real ECG and a PPG signal by using $n = 3$ and $n = 4$ different inverse poles, respectively. These signals have been resampled according to the related non-uniform grid. The PRD is less than 5% for both signals in Fig. 5.2, but we applied a small vertical shift for better visualization of the approximations. If we choose the appropriate inverse poles for the system, then the sampling points will be dense near sudden changes of the signal. So we can achieve better resolution for high frequency components. The problem of discretization is addressed in the present work. We note that this is especially important if the inverse poles and thus the discretization points are always adjusted to individual segments of the signal (see e.g., [KovLóc12a; KovKir13]).

Related works

There are several numerical algorithms available for solving the non-linear equations in Eq. (5.1). The bisection method is a very simple one which is able to determine a root of an $f : \mathbb{R} \to \mathbb{R}$ function. This algorithm is robust, which means that continuity is a necessary and sufficient condition for being terminated in a root (if it exists). Non-existent derivatives or the possibility of multiple roots do not cause any problems during the execution.

Unfortunately, the price of robustness is paid in the speed of the convergence, which is linear. This property was improved in [Lóc11], by taking advantage of the monotonic behavior of the argument function. Namely, the initial intervals

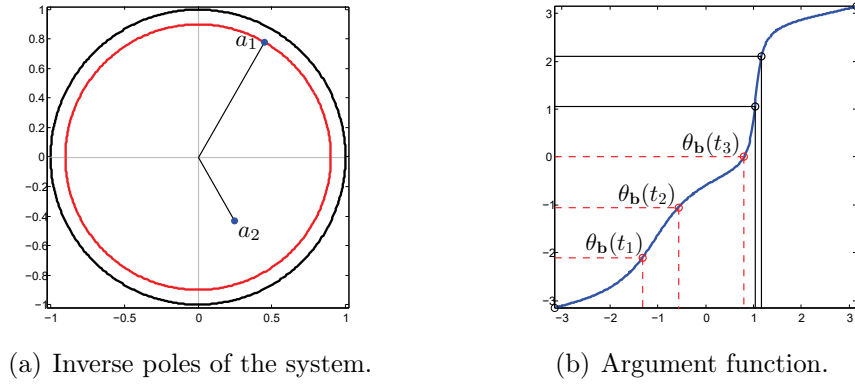(a) Inverse poles of the system.   (b) Argument function.

Figure 5.3: Rational function system defined by two poles.

can be set dynamically according to the previously calculated results. For this reason, let us consider a simple example with two inverse poles $a_0 = 0.9 \cdot e^{\frac{\pi}{3}i}$ and $a_1 = 0.5 \cdot e^{-\frac{\pi}{3}i}$ with $m_0 = m_1 = 1$. Fig. 5.3(b) shows the related argument function $\theta_b(t)$ for which $b = a = (a_0, a_1)$. Let $\theta_b(t_1) < \theta_b(t_2) < \theta_b(t_3)$. If the preimages of the points $\theta_b(t_1), \theta_b(t_3)$ are already computed, then monotonicity implies that t_2 lays between t_1 and t_3. As a consequence, the next bisection method for t_2 can be initiated from the interval $[t_1, t_3]$. Although this trick saves a lot of time, this is not enough when we have a large number of inverse poles, especially in pathological cases for $|a_i| \approx 1$. A detailed information about the execution time can be found in [Lóc11].

The parallel implementation of the bisection method can further improve the execution time. Let us consider M to be an integer power of two. By taking advantage of the previous section, we can evaluate the middle point $t_{M/2}$ of the distribution first. Then, in the second iteration two bisection methods can be performed on the intervals $[t_0, t_{M/2}]$ and $[t_{M/2}, t_{M-1}]$. Generally, in the kth step we have 2^{k-1} parallel executions of the bisection method. Unfortunately, this is not an optimal implementation for GPU computing [SanKan10]. In CUDA-based GPU architecture [Far11] a parallel job is represented by a kernel. A kernel consists of threads, which represent a unit of tasks executed in a parallel Single Instruction Multiple Data (SIMD) fashion. Threads are managed in a hierarchically organized way and the individual threads are handled together in the smallest hierarchical entity, the warp. Threads in the same warp share their memory addresses and they execute the same instruction at the same time. Furthermore, a definite number of threads create a block which can share a specific amount of memory, and they can communicate via synchronization. Blocks form the upmost hierarchical level, the grid. During the execution, a thread can identify itself by *id* numbers. A thread has a unique *id* in the block, and a block also has a unique *id* in the grid. It means that the maximal performance can only be reached if we compute all the points of D_b^M at the same time. In this case, one should use the original interval $[-\pi, \pi]$ at each run of the bisection method.

Fixed-point iterations

Another way to suppress the limitations of the previous algorithm is to swap the bisection for Householder's method [Hou70]. This is a general fixed-point iteration which can be written in the following form

$$(5.2) \qquad \tau_{k,l+1} = \tau_{k,l} + d \cdot \frac{(1/f)^{(d-1)}(\tau_{k,l})}{(1/f)^{(d)}(\tau_{k,l})}, \quad (d \in \mathbb{N}_+),$$

where $\tau_{k,l}$ is the lth approximation of the point $\tau_k \in D_\mathbf{b}^M$. The convergence rate of this method is equal to $d+1$ if the multiplicity of the root is equal to one, $f \in C^{(d+1)}$ and $\tau_{k,0}$ is close enough to the solution. Using Eq. (1.10), the first derivative of the average argument function can be written in the form

$$\theta_\mathbf{b}'(t) = \frac{1}{N} \sum_{j=0}^{N-1} \beta_{b_j}'(t) = \frac{1}{N} \sum_{j=0}^{N-1} P_{r_j}(t - \alpha_j),$$

where the inverse poles are given as $b_j = r_j \cdot e^{i\alpha_j}$, $(r_j \in [0,1),\ \alpha_j \in [-\pi, \pi))$. Since the Poisson kernel $P_r(t)$ has infinitely many derivatives, it is possible to use high order Householder methods. On the other hand, in this study we perform tests only for $d = 1, 2$ (Newton/Halley) because of the complexity of higher derivatives. In further sections, we will show that the difference between these methods is not significant.

However, we can only achieve faster convergence if the initial approximation of the solution is reasonably accurate. In order to cure this problem, we apply the inverse interpolation technique. Additionally, we utilize the benefits of the monotonic property of $\theta_\mathbf{b}$ via monotone piecewise cubic Hermite interpolation. In this case, the approximation of the inverse function has the following parameters:

- $x_i = -\pi + i \cdot \frac{2\pi}{L-1}$: values;

- $f_i := \theta_\mathbf{b}(x_i)$: base points;

- $d_i := 1/\theta_\mathbf{b}'(x_i)$: derivatives;

with $i = 0, 1, \ldots, L-1$ where L is the number of different base points. Now, we can fit a monotone piecewise cubic Hermite interpolation polynomial H which satisfies the following constraints:

$$H(f_i) = x_i, \quad H'(f_i) = d_i \quad (0 \le i < L).$$

In order to get a monotone interpolation of the argument function, we applied the algorithm in [FriCar80]. Then, the initial approximation of the solution is computed by $\tau_k \approx \tau_{k,0} = H(t_k)$ where $t_k \in D_0^M$. Note that both the number of base knots L and their positions have significant effects on the overall performance. More precisely, the former one increases the number of polynomial segments while the latter one influences the accuracy of the initial approximation of the solution. In the following two sections, we will provide an optimal setup of these parameters.

(a) Argument function for $n = 40$.

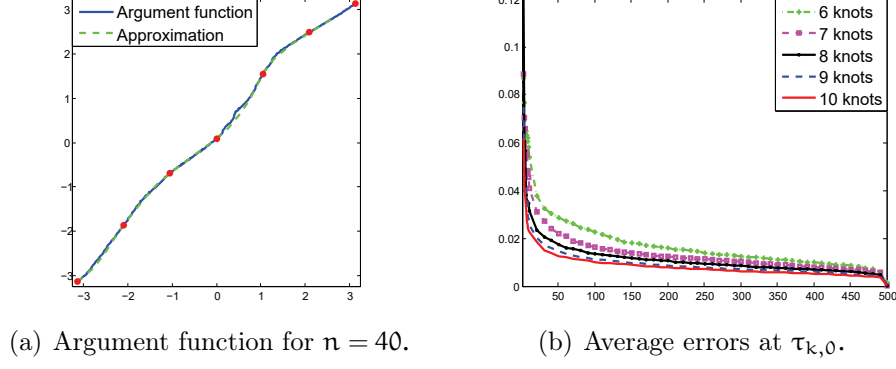(b) Average errors at $\tau_{k,0}$.

Figure 5.4: Experiments on large number of poles using uniform base knots.

Number of base knots

In our first experiment, we test the approximation error of the argument function against the number of uniformly sampled base knots. For this reason, the vector of inverse poles $\mathbf{a} \in \mathbb{D}^n$ is generated randomly with the multiplicities $m_0 = m_1 = \ldots = m_{n-1} = 1$. Then, we compute the ℓ^2 error between the monotone piecewise Hermite interpolation polynomial H and the original function $\theta_{\mathbf{b}}$. More precisely, we perform the following steps:

1. generate different inverse poles $a_i \in \mathbb{D}$ $(i = 0, \ldots, 500)$ and let $\mathbf{a} = a_0$, $n = 1$;

2. approximate the set $D_{\mathbf{b}}^M$ $(\mathbf{b} = \mathbf{a})$ via inverse interpolation using L knots;

3. compute the average error at $\tau_{k,0}$ for $0 \leq k < M$;

4. set $n = n + 1$ and $\mathbf{a} := (a_0, \ldots, a_{n-1})$ then repeat $2 - 4$ as long as $n < 500$.

This process was repeated ten times for different number of base knots. Fig. 5.4 shows the results for $L = 6, \ldots, 10$. Experiments for $L \leq 5$ are not displayed because of their high approximation error.

In Fig. 5.4(a) the argument function is similar to the identity function. It is caused by both the large number of poles and the fact that every pole is drawn from the standard uniform distribution of $\mathbb{D}$. Furthermore, increasing the number of base knots does not decrease significantly the approximation error, especially for large n. One can see this phenomenon in Fig. 5.4(b), where the average ℓ^2 error is less than 0.03 for $n > 50$. Furthermore, Fig. 5.4(b) shows that only 7 base points are enough to estimate the initial values $\tau_{k,0}$ with 1-digit precision in most cases. Note that the error is much higher for $n \leq 10$, which is investigated in the next subsection.

Rate of convergence

The rate of convergence of the aforementioned Householder's method is $d + 1$. We examine this statement for $n \leq 10$ number of inverse poles with high multiplicities.

One of the main interests in this experiment is to use inverse poles which are close to $\mathbb{T}$. Therefore, a random number of inverse poles are generated within the track limited by $\mathbb{T}$ and the inner circle in Fig. 5.3(a). This can be written as

$$(5.3) \qquad a_j = r_j \cdot e^{i\alpha_j}, \qquad (\, r_j \in [0.9, 1), \; \alpha_j \in [-\pi, \pi) \,).$$

for $0 \le j < n$. The test is organized as follows:

1. generate n different inverse poles and multiplicities;

2. construct the vectors $\mathbf{a} \in \mathbb{D}^n$, $\mathbf{b} \in \mathbb{D}^N \in$ and $\mathbf{m} \in \mathbb{N}_+^n$;

3. choose base knots for the approximation of θ_b;

4. compute the average error of the initial values $\tau_{k,0}$;

5. repeat from 2 by using the same poles $\mathbf{a}$ changing $\mathbf{b}$ and $m_j := m_j + 1$ where j is a random integer drawn from the set $\{1, \ldots, n\}$.

Note that our test cases satisfy the condition in Eq. (5.3) for at least one inverse pole. As a result, we have fewer poles but high multiplicities. It is in contrast with the first experiment, where we had a lot of inverse poles with single multiplicities. The main difference between the two experiments is the approximation error which is much higher in this case. It is an undesirable effect of the inverse poles which have great absolute values ($r_j \ge 0.9$). One can see this phenomenon in Fig. 5.3(b), where the argument function has high first derivatives near $\alpha_0 = \pi/3$ caused by the inverse pole a_0. To overcome this problem, we extend the uniformly sampled 7 base knots with additional points where the argument function has high curvatures. We assume that this means a major improvement of the initial values. Now, we go through the technical details about how we can choose the right base points.

First of all, we investigate the relationship between the absolute value of a single inverse pole and the shape of the curvature of the related argument function. Hence, in Fig. 5.5 we plot the first derivative and the curvature for a single inverse pole $a = 0.9 \cdot e^{i\frac{\pi}{3}}$, which can be computed as

$$\kappa(t) = \frac{|\beta_a''(t)|}{(1 + \beta_a'^2(t))^{\frac{3}{2}}}, \qquad (t \in [-\pi, \pi]).$$

As we know from Eq. (1.10), the first derivative of the argument function (i.e., $P_r(t - \alpha)$) attains its maximum value at $t = \alpha$ which is equal to $\dfrac{1 + r}{1 - r}$. This quotient increases as r tends closer to 1, which also determines the deviation from the identity function. At the same time, the width of the main lobe of the Poisson kernel decreases, which is marked by a red arrow in Fig. 5.5(a). Our aim is to predict this behavior by using the global extrema of the curvature. To this end, we choose additional knots for making the initial approximation more accurate. Let us consider Fig. 5.5(b), where we plot the width of the main lobe of β_a' against r.

(a) Curvature and first derivate of the argument function.

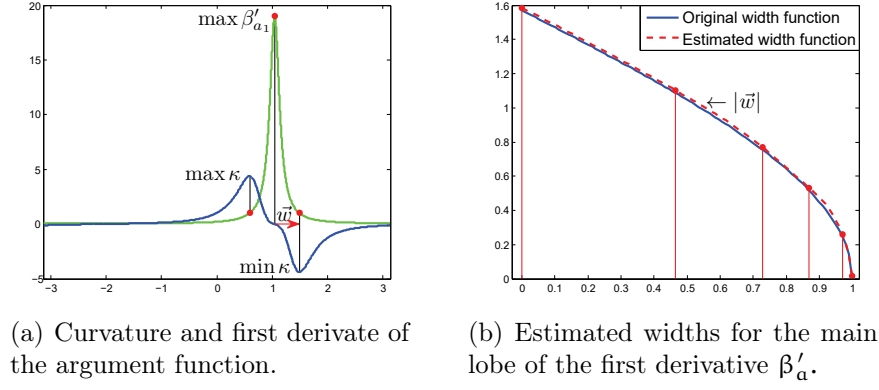(b) Estimated widths for the main lobe of the first derivative β_a'.

Figure 5.5: Estimated parameters of the argument function.

The graph of this figure is associated to the width function $\vec{w}(r)$ for $r \in [0, 1)$. We approximated this function by using piecewise cubic spline interpolation in Fig. 5.5(b). The estimation is marked by red dashed lines and the base knots (r_k, w_k) are denoted by the red points. Furthermore, the number and the positions of the base points have been localized by using a B-spline optimization method [KarMon97]. Although it is a slow procedure, we apply it only once in order to give an optimal estimation of $\vec{w}$. In further executions, we use only the precomputed approximation of the width function to choose new base points for estimating $\tau_{k,0}$ (see e.g., Alg. 3).

In Fig. 5.6(a) we plot the approximations of the argument function from Fig. 5.3(b). On the one hand, the ℓ^2 error over the equidistant knots is significant because the critical points are missing. On the other hand, the second approximation is much better due to the new (red) base points which are determined by the lines 9–11 in Alg. 3. We repeated this experiment 500 times with $n = 2 \dots 10$ random inverse poles for $M = N$. Finally, the Newton's method is applied with 10 iterations to refine the initial values of the estimation. During the tests, we evaluated about $127,000$ points in D_b^M. As a final step, we counted the points where the error was greater than 10^{-6}. Hence, the total number of divergent points can be seen in Fig. 5.6(b). Although the same number of base knots were used for both the uniform and the non-uniform approximations, the difference between them is significant. In the former case the average number of divergent points is more than 8%, whereas in the latter case it is reduced to 1% only. We note that it is also possible to switch back to the bisection method in a divergent point where the previously computed neighbors are chosen as the initial interval.

Parallel implementation in CUDA

Fortunately, not only algorithmic optimization can be achieved in this case. If we take a closer look at the solution of D_b^M using Newton/Halley iteration, we can notice that the solution processes of τ_k's are independent from each other, but the algorithmic instructions are completely the same. This finding gave us

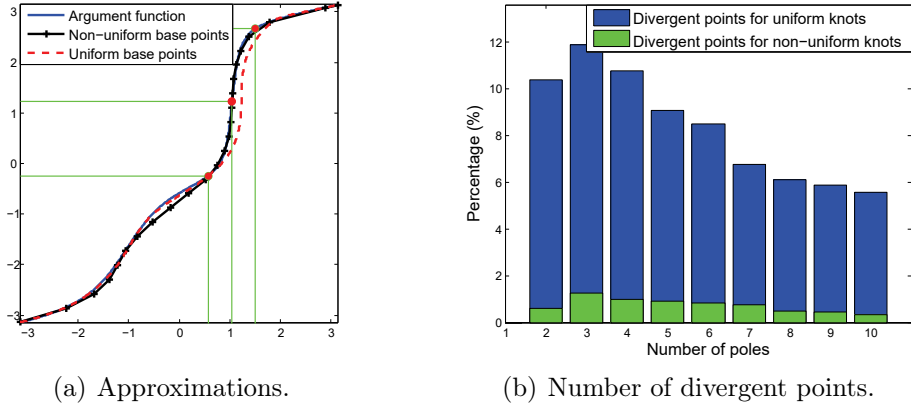(a) Approximations.   (b) Number of divergent points.

Figure 5.6: Estimated parameters of the argument function.

the opportunity to use massively parallel SIMD optimization. Our parallelization is relatively simple on principle. We dedicated a thread to the solution process of τ_k, where k could be determined by the *id* of the thread. At the moment, the monotone piecewise cubic Hermite interpolation polynomial fitting part is executed on the CPU. In this form one block represents one inverse pole configuration. On the one hand, it means there is a technical boundary for M, although we feel it is still enough for practical situations. On the other hand, it will be straightforward to generalize the implementation in order to solve $D_{\mathbf{b}}^M$ for many $\mathbf{b}$ with the same kernel execution.

We also exploited a few specialties of the GPU architecture. The evaluation of argument function is the bottleneck of the iterative part. If we examine this part, a high amount of redundancy can be noticed. Namely, each inverse pole uses a specific temporary variable which depends only on the value of the inverse pole. It is a natural assumption that these temporary variables could be precomputed. However, they need to be stored in a way that every thread can reach them simultaneously.

For this reason, the CUDA architecture has a special memory, called texture memory. Although the concept of texture memory came from the field of Computer Graphics, it is useful for general computing as well. It is due to the fact that the simultaneous reading is realized via cache memory. For this reason, we stored the inverse pole dependent temporary values in texture memory. As it was mentioned before, threads in the same warp execute the same instructions. In our case, it means that the efficiency of a warp depends on the thread, whose iteration has the smallest rate of convergence. Since in CUDA the branching is achieved by instruction masking, the naive implementation would execute instructions ineffectively on threads that reached the desired accuracy. Therefore, we used voting functions, which guarantees that a warp finishes the execution when the desired efficiency is reached by all threads in the warp. It does not make the execution much faster, but the solution will be more accurate without increasing execution time.
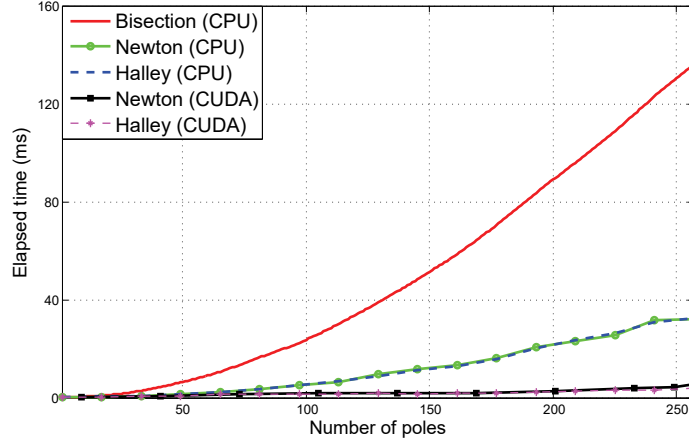
Figure 5.7: Comparing the performance of each method.

Our CPU code was executed on Intel Xeon W3530 2.80 Ghz, and the GPU was a NVIDIA Quadro 600. The experiments were performed on an increasing number of inverse poles, which were generated randomly. The time needed for execution can be observed in Fig. 5.7. We note that our algorithm can be used to initialize both the bisection and the Newton's method. For this reason, the execution time of this step is not included in this experiment. In order to express the effect of the algorithmic and the technical optimization, we run our method on both CPU and GPU architectures. Although we did not use parallel implementation, there is a significant difference between the original bisection method [Lóc11] and the single core execution of our procedure. The proposed solution needs much less time than the former algorithm. In addition the CUDA-based parallelization improves efficiency with another magnitude. To compare the asymptotic behavior of these procedures, we fit quadratic polynomials on the curves in Fig. 5.7. By using the `polyfit` MATLAB command we have the following results

$$y_{\text{bisect}} = 0.46x^2 + 0.45x - 2.3,$$
$$y_{\text{CPU}} = 0.13x^2 - 0.18x - 0.06,$$
$$y_{\text{CUDA}} = 0.01x^2 + 0.02x + 0.48.$$

Comparing the coefficients, we can see that the magnitude of the quadratic term decreased from 10^{-1} to 10^{-2}, i.e., asymptotically it is closer to a linear polynomial. Notice that there is no remarkable improvement for the Halley method, which is a natural consequence of the use of higher derivatives. Their complex form makes the evaluation of Eq. (5.2) more difficult.

Conclusion

We proposed an algorithm to solve the discretization problem. New concepts have been revealed to improve both the execution time and the accuracy. We introduced

not only a theoretical but a practical modification as well. Namely, we developed a parallel implementation for CUDA-based GPU architecture.

On the one hand, the high number of inverse poles or multiplicities can increase the execution time. On the other hand, if we model dynamic environments, it is also necessary to optimize the inverse poles. In this case, optimization techniques like PSO, NM, etc., require the computation of several irregular discretization set at each step. In the future, we want to apply our method in the aforementioned procedures to localize the best inverse pole(s) of discrete orthogonal rational systems.

5.3   Compression techniques

In this section, we want to investigate the efficiency of rational functions in compressing different types of ECG signals. For this reason, we perform experiments to compare the compression ratio and the accuracy of different methods in biomedical signal processing. A wide range of algorithms have been proposed in this field which are classified into the following categories [JalHut90]:

1. Parameter extraction techniques: algorithms are utilizing mostly polynomial prediction and interpolation [NavCoh93], entropy and differential coding [HamTom91]. These methods are usually combined with beat codebook, vector quantization [WanYua97; SunTai05], etc.

2. Direct time-domain techniques: the main concept is based on the reduction of the sampling frequency by decomposing the signal into plateaus, slopes, turning points and peaks [CoxNol68; Mue78; BohBar88; IshShi83; FirGor08]. There exist hybrid methods [AbeTom82] as well which utilize the advantages of these algorithms.

3. Transform-domain techniques: these methods can be interpreted as projections of the signal to low dimensional subspaces of a certain domain. Namely, the original data are represented by linear combinations of sinusoids [Lee-Buc06], wavelets [Add05], Walsh functions [BerChi00], orthogonal polynomials [JanOlm93], principal components [CasLagne], etc.

In this section, we are focusing on transform-domain techniques. To this end, we used the review [Add05] as a reference of these ECG signal compressing methods. In order to assess their quality, we perform our tests on two main classes of these algorithms. Namely, we compare the performance of rational functions with wavelet and polynomial approximation based methods.

Furthermore, we want to use an objective measure of the level of compression and distortion rate. For this reason, synthesized ECG signals in Section 3.1 are also used to test these methods. In this case, we can generate electrocardiograms with different diagnostic and geometrical features while the analytic forms of the original signals are also provided. So, both compression and distortion rates can

be compared by using different types of error measures, such as PRD and WDD (see e.g., Section 3.2).

B-spline approximation

It is an efficient and adaptive transform method for signal compression. In this case, we consider the optimization of the system of B-splines $B_{k,\ell}^{\mathbf{a}}$ which can be defined as follows:

$$B_{k,0}^{\mathbf{a}}(t) = \begin{cases} 1 & \text{if } a_i \leq t < a_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$B_{k,\ell}^{\mathbf{a}}(t) = \frac{a_{k+\ell+1} - t}{a_{k+\ell+1} - a_{k+1}} B_{k+1,\ell-1}^{\mathbf{a}}(t) + \frac{t - a_k}{a_{k+\ell} - a_k} B_{k,\ell-1}^{\mathbf{a}}(t).$$

The base functions are characterized by the knot vector $\mathbf{a} = (a_0, a_1, \ldots, a_{n+\ell})$ and their optimal locations should be determined. For a fixed ℓ order of B-splines, the corresponding parameters of our framework in Eq. (1.3) are the knots $\mathbf{a}$, the system $\Theta_k^{\mathbf{a}} = B_{k,\ell}^{\mathbf{a}}$ and the coefficients c_k. Here, the optimization process contains an iterative method which removes the most insignificant knot at each step. Although the system is highly adaptive, the coefficients can hardly be computed due the non-orthogonal property of the B-spline base functions. In order to fix this problem, an efficient knot ranking algorithm was proposed in [KarMon97].

Wavelet decomposition

The theory of discrete wavelet transform (DWT) has several practical applications in signal processing [Mal08]. Generally, it starts with a system $\{\phi(t-j)\}_{j\in\mathbb{Z}}$ which is usually an orthonormal basis of an initial subspace $V_0 \subset L^2(\mathbb{R})$, where $\phi(t)$ is the so-called scaling function. Then, the original signal is decomposed into the subspaces V_i and W_i, which are orthogonal complements of each other with respect to V_{i+1}. The embedded subspaces $V_i \subset V_{i+1}$ ($i \in \mathbb{Z}$) are referred to as multiresolution analysis (MRA), which can be used to define an orthonormal basis of V_i and W_i as follows:

$$\phi_{i,j}(t) = 2^{i/2}\phi(2^i t - j) \quad (i, j \in \mathbb{Z}),$$
$$\psi_{i,j}(t) = 2^{i/2}\psi(2^i t - j) \quad (i, j \in \mathbb{Z}).$$

The scaling function ϕ and the so-called mother wavelet ψ are completely characterized by a compactly supported low-pass filter h as follows:

$$\phi(t) = \sum_{j=0}^{\ell-1} h_j\phi(2t - j), \quad \psi(t) = \sum_{j=0}^{\ell-1} g_j\phi(2t - j),$$

where $g_j = (-1)^{1-j} h_{\ell-1-j}$ is a high-pass filter and ℓ is the filter length. Although the constraints of orthogonality restrict the values of the filter's coefficients, we still have $\frac{\ell}{2} - 1$ degree-of-freedom to choose h_j [BurGop97]. In this section, we are using the so-called Daubechies wavelets [Dau92b] which have $\ell/2$ vanishing moments, i.e., they can suppress at least $\frac{\ell}{2} - 1$ order of polynomials. Now we can insert the DWT into our framework in Eq. (1.3) by defining the vector index $\mathbf{k} = (i, j)$, which runs through the dilation i and the translation j indices of the mother wavelet ψ. Then, for a fixed ℓ length of low-pass filter $\Theta_{\mathbf{k}}^{\mathbf{a}} = \psi_{\mathbf{k}}$. Note that the Daubechies wavelets have no free parameters, i.e., the system is not adaptive. In order to cure this problem, the wavelet packets transform (WPT) was introduced by Coifman et al. [CoiMey92], which utilizes different tilings of the time-frequency plane. Namely, both the approximation and the detailed coefficients are further decomposed at each level of the MRA. Then all coefficients that have high entropy are removed. The remaining terms are the 'best' optimal selection in the sense of Shannon entropy. A detailed description of the implementation can be found in [Add02]. Although it is possible to construct other orthogonal wavelets with $\frac{\ell}{2} - 1$ degree-of-freedom, these algorithms are beyond the scope of this dissertation. For further details of this topic, we refer to the papers [BreLuc07; BlaCru07; AboAlA13].

Experiments on synthetic data

In this experiment, we generated 100 different synthetic ECG signals with normal diagnostic parameters defined in Section 3.1. According to the review [Add02] we use Daubechies wavelets with five vanishing moments (db5). The embedded MATLAB commands `wdencmp` and `wpdencmp` were used to perform tests on DWT and WPT. Since the applied ECG model [Kov12] is based on splines, a B-spline based compression algorithm would result in a perfect reconstruction. Thus, during the first experiment, we skip this method in order to compare the results in a fair, objective way.

Now, we apply the aforementioned algorithms to process 100 synthetic ECG signals. During the compression, the thresholds are decreased at each step until the PRD exceeds 5%. Fig. 5.8(a) shows the required number of coefficients. One can see that the average number of MT coefficients is fewer than the DWT. However, the WPT is better, but it is still comparable with the performance of rational functions. Fig. 5.8(b) expresses the cases where the WDDs of the reconstructed signals are between $0 - 10\%$, $10 - 20\%$, etc. Note that the WDD of the MT approximation is the best among all the tested methods.

Experiments on real data

In our next experiment, we are applying the algorithms to real ECG data. Our comparisons include the rational representation, DWT, WPT and a B-spline based adaptive method [KarMon97]. The test contains the first 3 minutes of 14 real ECG signals of the MIT-BIH database from PhysioNet [GolAma00]. Some preprocessing
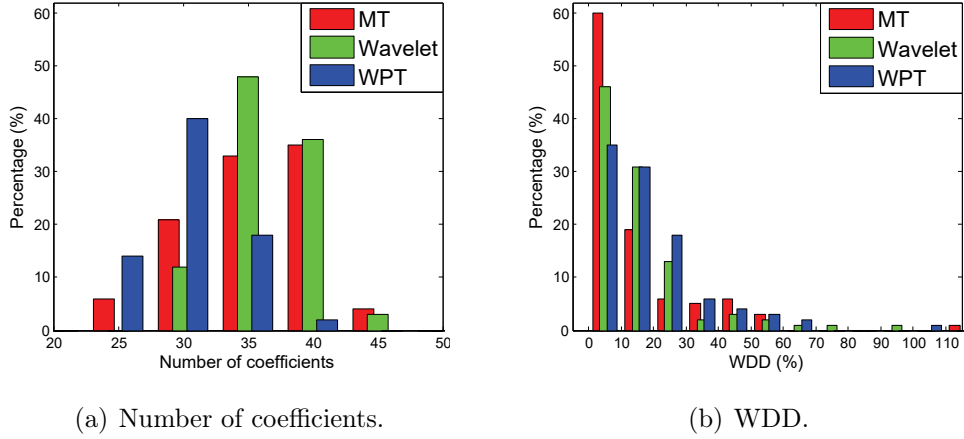
(a) Number of coefficients.
(b) WDD.

Figure 5.8: Statistics of the tested methods using 100 synthetic beats.

steps are necessary to deal with noise, but these algorithms are beyond the scope of this dissertation, this is why we just enumerate the applied techniques. First of all, the signals are segmented into beats, which is followed by a simple baseline subtraction algorithm [Cline] to deal with baseline wandering. Additionally, the well-known Karhunen–Loéve transformation (KLT) is also applied to filter out high frequency noises. Namely, the series of beats in a signal are aligned along with their R peaks. Then, only the first four KLT bases were computed. The KLT was implemented via SVD decomposition of 20 successive beats. In order to improve the speed of this preprocessing step, we utilized the running SVD algorithm in [CheAxt10].

In contrast with our previous experiment, now we fixed the parameters of the systems to 30. It means that the DWT and the WPT can have 30 coefficients, since they have no free parameters. The MT representation possesses three inverse poles with multiplicities $\mathbf{m} = (3, 6, 3)$ i.e., $30 = 2 \cdot (12 + 3)$. Finally, it means 15 base knots and 15 coefficients for the B-splines. In this case, the compression ratio is the same for all these procedures. Thus, the algorithm with less approximation error is better. The final results can be seen in Fig. 5.9, where the average PRD is assigned to all of the related methods and records. As we expected, the B-spline approximation cannot keep up with the other three methods. Although the difference is not significant in case of MT, DWT and WPT, the rational representation performed slightly better in most cases.

Note that our primary goal was to demonstrate the feasibility and applicability of rational functions in ECG compression. The potential application area can be extended to other types of data which are not necessarily biomedical or physiological signals. Furthermore, for every algorithm (Fig. 5.9) we counted only the number of coefficients required to achieve a certain level of accuracy. Other parameters such as quantization level or the indices of the thresholded wavelet coefficient vector were not considered, although the latter one can notably increase
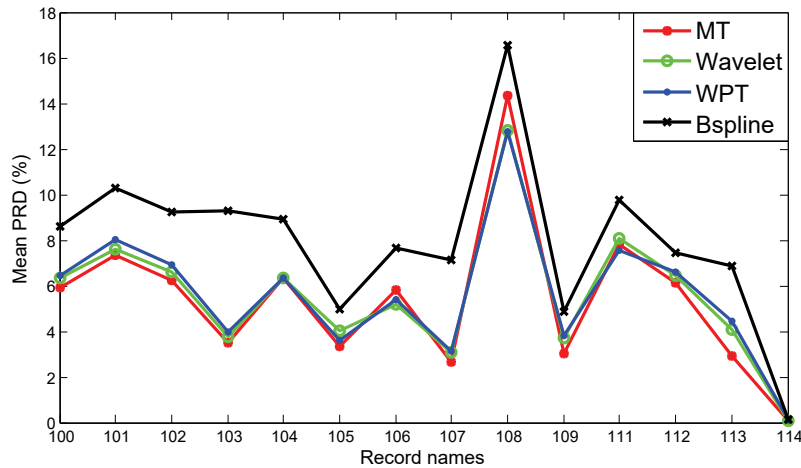
Figure 5.9: Statistics of 14 compressed real ECG signals.

the amount of the required data. To this end, efficient data structures such as embedded zerotree wavelet (EZW) [Sha93] or set partitioning in hierarchical trees (SPIHT) are available in [SaiPea96]. It is also worth mentioning that the WPT algorithm has outperformed the simple wavelet thresholding method in some cases (Fig. 5.9). We note that it did not turn to be useful in practice due to the additional header information needed to identify the best basis. This phenomenon was described, for instance, by Hilton in [Hil97]. Finally, we conclude that the rational representations can achieve the same level of accuracy as the DWT and WPT. On the other hand, rational methods do not require additional information for the reconstruction, which is not true for wavelet based algorithms. For instance, in the latter case, the indices of non-zero coefficients should be also retained. It is also worth mentioning that we counted the number of parameters only. However, the results are dependent on the number of bits used for storing these parameters.

Conclusion

The main disadvantage of B-splines is the lack of orthogonality, i.e., the coefficients can be computed via the least square method. On the contrary, DWT and WPT grant low computational complexity. Unfortunately, the price of speed is paid in the adaptivity of the systems. Representations of real signals by means of optimized rational functions attempted to fix this problem. Namely, we are keeping the orthogonality while increasing the number of free parameters. As confirmed by experiments, rational functions have a great potential in compressing signals. Only one question–that of the filtering property–remains open: can we use these expansions for denoising ECG?

Chapter 6

Notations

$\mathbb{N}$	is the set of natural numbers with zero $\mathbb{N} := \{0, 1, 2, 3, \dots\}$
$\mathbb{N}_+$	is the set of natural numbers without zero $\mathbb{N}_+ := \{1, 2, 3, \dots\}$
$\mathbb{Z}$	is the set of integers
$\mathbb{R}$	is the set of rational numbers
$\overline{\overline{\mathbb{R}}}$	is the extended real line given by $\mathbb{R} \cup \{\infty\}$
$\mathbb{C}$	is the set of complex numbers
$\overline{\mathbb{C}}$	is the extended complex plane given by $\mathbb{C} \cup \{\infty\}$
$\mathbb{D}$	is the open unit disc defined as $\{z \in \mathbb{C} : |z| < 1\}$
$\overline{\mathbb{D}}$	is the closed unit disc defined as $\{z \in \mathbb{C} : |z| \leq 1\}$
$\mathbb{T}$	is the unit circle or torus defined as $\{z \in \mathbb{C} : |z| = 1\}$
$\mathbb{B}$	is the Cartesian product of $\mathbb{D}$ and $\mathbb{T}$
$\mathbb{I}$	is the open real interval $(-1, 1)$
$\arg\min$	is the argument of the minimum $\arg\min_x f := \{x : f(x) \leq f(y) \ (y \in D_f)\}$
$\mathfrak{R}$	is the set of rational functions that are analytic on $\overline{\mathbb{D}}$
$H^2(\mathbb{D})$	is the Hardy space of square integrable functions that are analytic on $\mathbb{D}$
$L^2(\mathbb{T})$	is the space of square integrable functions on $\mathbb{T}$
$L^2_w(\mathbb{R})$	is the w-weighted space of square integrable functions on $\mathbb{R}$
$\overline{a}$	is the complex conjugate of $a \in \mathbb{C}$
$\widehat{f}$	is the Fourier transform of the function $f \in L^2(\mathbb{T})$
$\mathcal{H}f$	is the Hilbert transform of the function $f \in L^2(\mathbb{T})$
$\mathcal{F}^\epsilon_g f(t, \omega)$	is the trigonometric short time Fourier transform of f with a window g
$\mathcal{C}_g f(t, \omega)$	is the Cohen's class distribution of f with a specified window g

Appendix A

Algorithms

Algorithm 1 Blaschke transformation of an image $I \in \mathbb{C}^{N \times N}$

1: **function** BLASCHKES_IMG(I,a)

2:     **for** $k \leftarrow 1, N$ **do**

3:         **for** $\ell \leftarrow 1, N$ **do**

4:             $x := 2\ell/N - 1$

5:             $y := 2k/N - 1$

6:             $z := x + iy$

7:             **if** $|z| < 1$ **then**

8:                 $K := \lfloor (\mathrm{Im}\, B_a(z) + 1) \cdot N/2 \rfloor$

9:                 $L := \lfloor (\mathrm{Re}\, B_a(z) + 1) \cdot N/2 \rfloor$

10:                 $B(k, \ell) := I(K, L)$

11:                 $B_{abs}(k, \ell) := |B_a(z)|$

12:                 $B_{arg}(k, \ell) := \angle B_a(z)$

13:             **else**

14:                 $B(k, \ell) := 0$

15:                 $B_{abs}(k, \ell) := 0$

16:                 $B_{arg}(k, \ell) := 0$

17:             **end if**

18:         **end for**

19:     **end for**

20:     **return** B, B_{abs}, B_{arg}

21: **end function**

Algorithm 2 hyperbolic multiplication $\odot$

1: **function** HYP_MUL(λ, w_1, w_2)
2:      **if** $w_1 = w_2$ **then**
3:          $s_\lambda := 0$
4:          $w_\lambda := w_1$
5:      **else**
6:          $p := |B_{w_1}(w_2)|$
7:          $\epsilon := B_{w_1}(w_2)/|B_{w_1}(w_2)|$
8:          $a := -\bar{\epsilon}w_1$
9:          $s_\lambda := \text{th}(\lambda \,\text{arth}(p))$
10:         $w_\lambda := \epsilon B_a(s_\lambda)$
11:      **end if**
12:      **return** s_λ, w_λ
13: **end function**

Algorithm 3 Choosing base points with high curvature.

1: **function** CHOOSE_KNOTS($\mathbf{a}$)
2:      load('bspline_parameters', r_k, w_k) $\triangleright$ base points (r_k, w_k)
3:      knots $:= [-\pi, \pi]$
4:      **for** $i \leftarrow 1, n$ **do**
5:          $r := |a_i|$
6:          $\alpha := \angle a_i$
7:          **if** $0.9 \le r$ **then**
8:              width $:= \text{spline}(r_k, w_k, r)$ $\triangleright$ cubic spline interpolation
9:              knots $:= [\text{knots}, \alpha]$
10:             knots $:= [\text{knots}, \alpha + \text{width}]$
11:             knots $:= [\text{knots}, \alpha - \text{width}]$
12:          **end if**
13:      **end for**
14:      **return** knots
15: **end function**

Algorithm 4 Basic Particle Swarm Optimization

1: **function** $\mathrm{BPSO}(f, S, N, V_{max}, c_{1,2})$
2:      $w_0 := 0.2$ $\triangleright$ Initializing weights, particles, etc.
3:      $w_N := 0.8$
4:      **for all** $k \leftarrow 1, S$ **do**
5:          Randomize x_k, v_k
6:          Initialize $\widetilde{y}_k, \widehat{y} := x_k$
7:      **end for**
8:      **for** $\ell \leftarrow 1, N$ **do**
9:          **for** $k \leftarrow 1, S$ **do**
10:              **if** $f(\widetilde{y}_k) > f(x_k)$ **then** $\triangleright$ Updating the local best positions.
11:                  $\widetilde{y}_k := x_k$
12:              **end if**
13:          **end for**
14:          **if** $f(\widehat{y}) > \min\limits_{1 \le i \le S} f(\widetilde{y}_i)$ **then** $\triangleright$ Updating the global best position.
15:              $\mathrm{gbest} := \arg \min\limits_{1 \le i \le S} f(\widetilde{y}_i)$
16:              $\widehat{y} := x_{gbest}$
17:          **end if**
18:          $w := w_N - \ell(w_N - w_0)/N$ $\triangleright$ Updating the inertia weight.
19:          **for** $k \leftarrow 1, S$ **do** $\triangleright$ Updating v_k and x_k.
20:              Randomize $r_1, r_2 \in [0, 1]$
21:              $v_k := c_1 r_1 \cdot (\widetilde{y}_k - x_k) + c_2 r_2 \cdot (\widehat{y} - x_k) + w \cdot v_k$
22:              **if** $|v_k| > V_{max}$ **then**
23:                  $v_k := \mathrm{sgn}\, v_k \cdot V_{max}$
24:              **end if**
25:              $x_k := x_k + v_k$
26:          **end for**
27:      **end for**
28:      **return** x_{gbest}
29: **end function**

Algorithm 5 Multi-Dimensional Particle Swarm Optimization

1: **function** MDPSO(f, S, N, D_{min}, D_{max}, VD_{max}, V_{max}, $c_{1,2}$)

2:      $w_0 := 0.2$ ▷ Initializing weights, particles, etc.

3:      $w_N := 0.8$

4:      **for all** $k \leftarrow 1, S$ **do**

5:          **for all** $d \leftarrow D_{min}, D_{max}$ **do**

6:              Randomize x_k^d, v_k^d

7:              Initialize $\widetilde{y}_k^d$, $\widehat{y}^d := x_k^d$

8:          **end for**

9:          Randomize d_k, vd_k

10:         Initialize $\widetilde{d}_k$, $\widehat{d}$

11:      **end for**

12:      **for** $\ell \leftarrow 1, N$ **do**

13:          $w := w_N - \ell(w_N - w_0)/N$ ▷ Updating the inertia weight.

14:          **for** $k \leftarrow 1, S$ **do**

15:              **if** $f(x_k^{d_k}) < f(\widetilde{y}_k^{d_k})$ **then**

16:                  **if** $f(x_k^{d_k}) < f(\widetilde{y}_k^{\widetilde{d}_k})$ **then**

17:                      $\widetilde{d}_k := d_k$ ▷ Updating the local best dim.

18:                  **end if**

19:                  $\widetilde{y}_k^{d_k} := x_k^{d_k}$ ▷ Updating the local best positions.

20:              **end if**

21:              **if** $f(x_k^{d_k}) < \min\left\{f(\widehat{y}^{d_k}), \min\limits_{1 \leq i \leq S, i \neq k} f(x_i^{d_k})\right\}$ **then**

22:                  **if** $f(x_k^{d_k}) < f(\widehat{y}^{\widehat{d}})$ **then**

23:                      $\widehat{d} := d_k$ ▷ Updating the global best dim.

24:                  **end if**

25:                  $gbest^{d_k} := k$ ▷ Updating the global best positions.

26:                  $\widehat{y}^{d_k} := x_{gbest^{d_k}}^{d_k}$

27:              **end if**   ▷ Updating d_k, vd_k, $x_k^{d_k}$ and $v_k^{d_k}$.

28:              UPDATE_POS($x_k^{d_k}$, $v_k^{d_k}$, $\widetilde{y}_k^{d_k}$, $\widehat{y}^{d_k}$, V_{max}, $c_{1,2}$, w)

29:              UPDATE_DIM(d_k, vd_k, $\widetilde{d}_k$, $\widehat{d}$, D_{min}, D_{max}, VD_{max}, $c_{1,2}$)

30:          **end for**

31:      **end for**

32:      **return** $x_{gbest^{\widehat{d}}}^{\widehat{d}}$

33: **end function**

Algorithm 6 Applying dimensional updates.

1: **function** UPDATE_DIM(d_k, vd_k, $\widetilde{d}_k$, $\widehat{d}$, D_{min}, D_{max}, VD_{max}, $c_{1,2}$)
2:     Randomize $r_1, r_2 \in [0, 1]$
3:     $vd_k = [vd_k + c_1 r_1 \cdot (\widetilde{d}_k - d_k) + c_2 r_2 \cdot (\widehat{d} - d_k)]$
4:     **if** $|vd_k| > VD_{max}$ **then**   ▷ Clamping vd_k.
5:         $vd_k = \operatorname{sgn} vd_k \cdot VD_{max}$
6:     **end if**
7:     $d_k = d_k + vd_k$
8:     **if** $d_k > D_{max}$ **then**   ▷ Clamping d_k.
9:         $d_k = D_{max}$
10:     **else if** $d_k < D_{min}$ **then**
11:         $d_k = D_{min}$
12:     **end if**
13:     **return** d_k, vd_k
14: **end function**

Algorithm 7 Applying positional updates.

1: **function** UPDATE_POS($x_k^{d_k}$, $v_k^{d_k}$, $\widetilde{y}_k^{d_k}$, $\widehat{y}^{d_k}$, V_{max}, $c_{1,2}$, w)
2:     Randomize $r_1, r_2 \in [0, 1]$
3:     $v_k^{d_k} = c_1 r_1 \cdot (\widetilde{y}_k^{d_k} - x_k^{d_k}) + c_2 r_2 \cdot (\widehat{y}^{d_k} - x_k^{d_k}) + w \cdot v_k^{d_k}$
4:     **if** $|v_k^{d_k}| > V_{max}$ **then**
5:         $v_k^{d_k} = \operatorname{sgn} v_k^{d_k} \cdot V_{max}$ ▷ Clamping $v_k^{d_k}$.
6:     **end if**
7:     $x_k^{d_k} = x_k^{d_k} + v_k^{d_k}$
8:     **return** $x_k^{d_k}$, $v_k^{d_k}$
9: **end function**

Appendix B

MATLAB examples

MATLAB GUIs

In this section, we want to demonstrate the use of the ECG signal generator and the rational functions of Sections 3.1–5.1. For this reason, three interactive MATLAB GUIs are implemented.

The first GUI is called `blaschke_tool`. It can be used to visualize the connection between the position of the inverse pole and the values of the argument function. It is also possible to display grayscale images that are projected onto the unit disc via Blaschke functions (see e.g., Fig. B.1(a) and Fig. B.1(b)). Furthermore, one can see the absolute values and the arguments of the Blaschke function in Fig. B.1(c) and Fig. B.1(d).

In order to demonstrate the properties of the MT systems, we build up the GUI called `malmquist_tool`. Here, the user can change the positions, the number and the multiplicities of inverse poles, interactively. The argument function is also displayed on the unit disc. Furthermore, all the members of the MT system can be visualized according to the selected inverse poles. It is not only the complex case, but the real valued MT–Fourier expansions are implemented as well. Moreover, both the real and the complex discretizations are available for interpolation purposes. All of these functions can be animated by moving the inverse poles in the unit circle. Fig. B.2 shows some illustrations generated by the toolbox. In Fig. B.2(a) one can see the inverse poles and the discretization of the real valued MT system while Fig. B.2(b) displays the corresponding argument function. It also illustrates that the discretization points tend to be dense near the inverse pole as it gets closer to the torus. This is due to the fact that the set $\mathbb{T}_n$ in Eq. (1.26) can be interpreted as a solution of an electrostatic equilibrium problem [PapSch01].
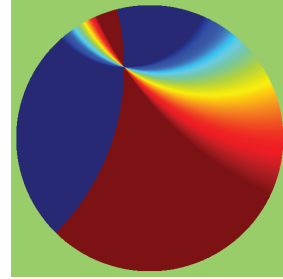
(a) Original image.

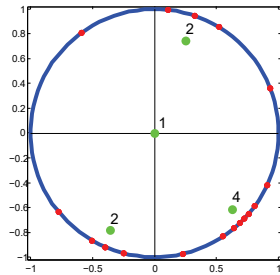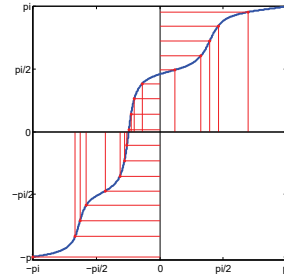
(b) Transformed image.
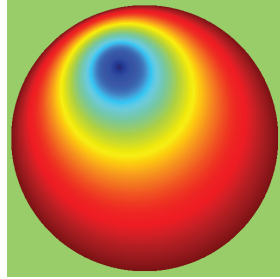
(c) Absolute values of a Blaschke function.

(d) Arguments of a Blaschke function.

Figure B.1: Properties of a Blaschke function.

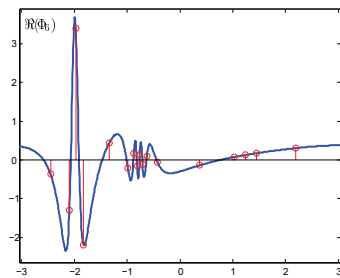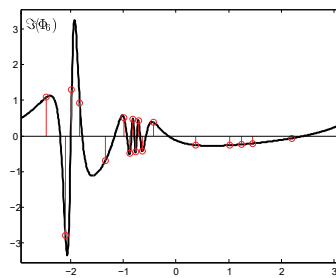(a) Inverse poles, multiplicities and $\mathbb{T}_n$.

(b) Argument function.

(c) Real part of Φ_6.

(d) Imaginary part of Φ_6.

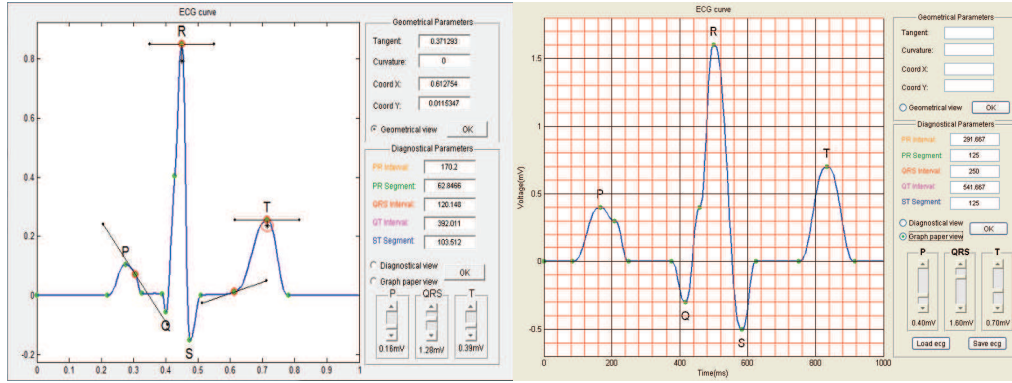Figure B.2: Visualization of the MT systems defined by four inverse poles.

Figure B.3: Geometrical view (left) and graph paper view (right) of an ECG curve.

In order to control the ECG generation process, a MATLAB Toolbox called ECGGEN was also created with an interactive GUI. It consists of three view modes, each with a different functionality:

Geometrical view: It was designed to show the geometrical parameters of the ECG curves. Therefore, only tangents and curvatures can be seen at each base point. The positions of these points are expressed with their coordinates.

Diagnostical view: It is a conversation between geometrical and diagnostic properties. Users can change the duration of five diagnostic intervals and the amplitudes of the waves. All parameters are given in milliseconds (ms) and millivolts (mV). This view mode can be seen in Fig. 3.1.

Graph paper view: It is a standard medical viewpoint. Each ECG recorder prints the output signals to a graph paper which has a regular grid. As usual, the time is represented on the x-axis and the voltage is represented on the y-axis. In addition, each small square of the grid is 1 mm in length which represents 40 ms time and 0.1 mV amplitude.

The parameters are converted proportionally between view modes, so we can examine the same ECG signal from different viewpoints.

Examples

In this section we present some MATLAB instructions of the RAIT toolbox. First, we start with a real ECG curve from the PTBDB database of Physionet [GolAma00]. Second, we perform the simplex algorithm to find the optimal inverse poles for the continuous MT system. Finally, we compute the coefficients and the corresponding MT expansion. Note that the original signal is Hilbert transformed using the command `addimag`. As we shall see later, this step can be skipped if we use the discrete real valued MT system.
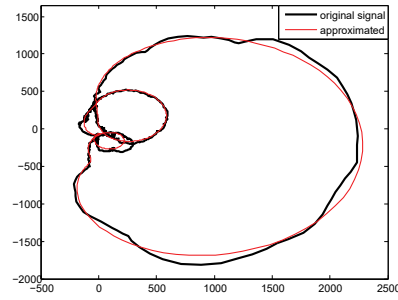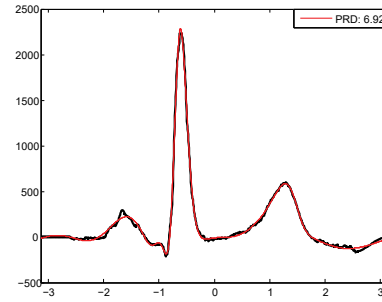
(a) The analytic signal.
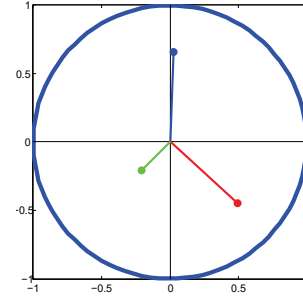
(b) Optimal inverse poles.
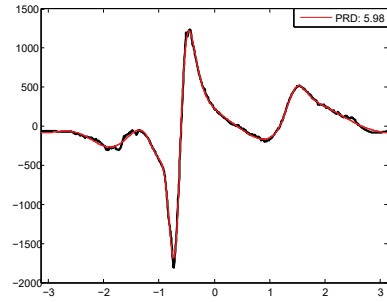
(c) Imaginary part of the projection.

(d) Real part of the projection.

Figure B.4:   MT approximation of the record *s0306lrem* from the Physi-oNet/PTBDB database.

```
>> s=periodize(ecg,0.1,0);
>> s_im=addimag(s);
>> m=[1 2 1];
>> pr=2;
>> init=zeros(1,6);
>> p=simplex_mt(s,m,pr,init,0,1e-6);
>> mp=periodize_poles(multiply_poles(p,m),pr);
>> mtco=mt_coeffs(s_im,mp);
>> s_mt=mt_generate(length(s),mp,mtco);
```

Now, we can get the reconstructed signal by taking the real part of the variable `s_mt`. In Fig. B.4, one can see the optimal inverse poles of the MT system and the approximation. Note that the 6.92% PRD is achieved by using only 8 coefficients and 3 inverse poles, which means 11 complex (or 22 real) parameters. In contrast, the original signal contains about 900 samples per heartbeat.

If we convert the coefficients into the basic rational function system (RF), the main diagnostic waves of the ECG curve can be separated. This method can be used to analyze ECG curves, because the corresponding waves express real diagnostic features. Fig. B.5(a) shows the lack of this property in case of the MT system, which does not reflect any diagnostic information. On the other hand, the basic rational representation preserves the individual waveforms of the ECG.

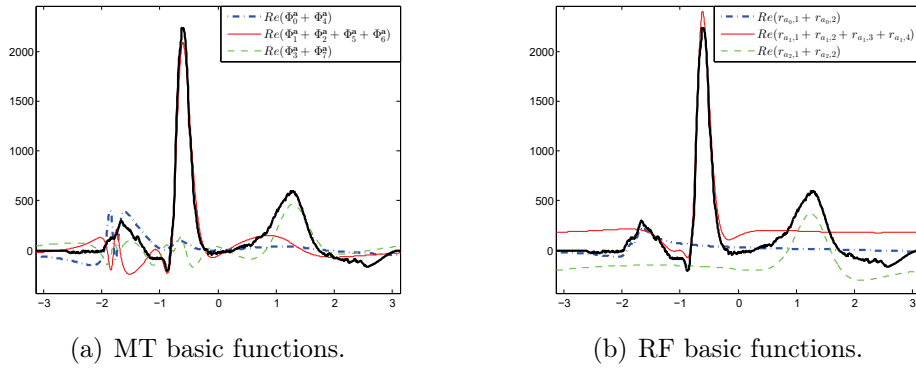(a) MT basic functions.   (b) RF basic functions.

Figure B.5: Basic functions of MT and RF systems.

Fig. B.5(b) is generated by the following code :

```
>> len=length(s);
>> lfco=coeff_conv(len,mp,mtco,'mt','lf');
>> lfs=lf_system(len,mp);
>> P=real(lfco([1,5])*lfs([1,5],:));
>> QRS=real(lfco([2,3,6,7])*lfs([2,3,6,7],:));
>> T=real(lfco([4,8])*lfs([4,8],:));
```

Now, let us consider the discrete analogue of the previous examples. For a change, we use a segment of a central venous pressure (CVP) signal. First, the optimal inverse poles are determined. Then, the discrete coefficients are calculated, directly. Note that there is no need to apply the Hilbert transformation here. In Fig. B.6(a) and Fig. B.6(b) we display the reconstructed signal using only three different inverse poles. The latter figure was obtained by repeating the number of the inverse poles periodically. Hence, the same instructions were executed by setting the variable pr to 2. For instance, Fig. B.6 was obtained as follows:

```
>> s=periodize(cvp,0.1,0);
>> len=length(s);
>> m=[1 3 2];
>> pr=1;
>> init=[-0.6,-0.6,0.2,0.7,-0.4,-0.6];
>> p=simplex_mtdr(s,m,pr,init,0,1e-6);
>> mp=periodize_poles(multiply_poles(p,m),pr);
>> [cUk,cVk]=mtdr_coeffs(s,mp,1e-6);
>> s_mtdr=mtdr_generate(len,mp,cUk,cVk);
```
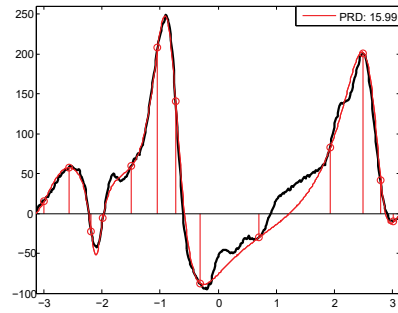
(a) Approximation of the signal by using 6 poles.
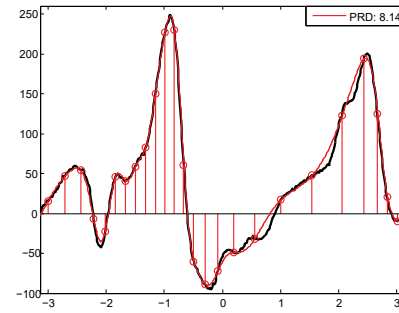
(b) Approximation of the signal by using 12 poles.

Figure B.6: Real valued MT approximation of the record *mgh001* from the PhysioNet/MGHDB database.

Publications of the Author

[FriKov12]   S. Fridli, P. Kovács, L. Lócsi, and F. Schipp. "Rational modeling of multi-lead QRS complexes in ECG signals". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 37 (2012), pp. 145–155.

[Kov12]   P. Kovács. "ECG signal generator based on geometrical features". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 37 (2012), pp. 247–260.

[KovKir13]   P. Kovács, S. Kiranyaz, and M. Gabbouj. "Hyperbolic Particle Swarm Optimization with Application in Rational Identification". In: *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*. Piscataway, USA: IEEE press, 2013, pp. 1–5.

[KovLóc12a]   P. Kovács and L. Lócsi. "RAIT: the rational approximation and interpolation toolbox for MATLAB". In: *Proceedings of the 35th International Conference on Telecommunications and Signal Processing (TSP)*. Piscataway, USA: IEEE press, 2012, pp. 671–677.

[KovLóc12b]   P. Kovács and L. Lócsi. "RAIT: The Rational Approximation and Interpolation Toolbox for MATLAB with Experiments on ECG Signals". In: *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems (IJATES)* 1.2–3 (2012), pp. 60–68.

[KovSam14]   P. Kovács, K. Samiee, and M. Gabbouj. "On application of rational discrete short time Fourier transform in epileptic seizure classification". In: *Proceedings of the 39th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Piscataway, USA: IEEE press, 2014, pp. 5839–5843.

[KovVad13]   P. Kovács and V. Vad. "Fast Computing of Non-Uniform Sampling Positions for Real Signals". In: *Proceedings of the 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. Piscataway, USA: IEEE Computer Society press, 2013, pp. 146–150.

[LócKov12]   L. Lócsi and P. Kovács. "Processing ECG signals using rational function systems". In: *Proceedings of the 7th IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. Piscataway, USA: IEEE press, 2012, pp. 123–127.

[SamKov14]   K. Samiee, P. Kovács, and M. Gabbouj. "Epileptic seizure classifi-
             cation of EEG time-series using rational discrete short time Fourier
             transform". In: *IEEE Transactions on Biomedical Engineering* 62.2
             (2014). (impact factor: 2.23), pp. 541–552.

References

[AbeTom82] J. P. Abenstein and W. J. Tompkins. "New data-reduction algorithm for real-time ECG analysis". In: *IEEE Transactions on Biomedical Engineering* BME-29.1 (1982), pp. 43–48.

[AboAlA13] M. Abo-Zahhad, A. F. Al-Ajlouni, S. M. Ahmed, and R.J. Schilling. "A new algorithm for the compression of ECG signals based on mother wavelet parameterization and best-threshold level selection". In: *Digital Signal Processing* 23 (2013), pp. 1002–1011.

[Add02] P. S. Addison. *The Illustrated Wavelet Transform Handbook – Introductory Theory and Applications in Science, Engineering, Medicine and Finance.* Bristol, UK: Institute of Physics Publishing, 2002.

[Add05] P. S. Addison. "Wavelet transforms and the ECG: a review". In: *Physiological Measurement* 26.5 (2005), pp. 155–199.

[AdeZho03] H. Adeli, Z. Zhou, and N. Dadmehr. "Analysis of EEG records in an epileptic patient using wavelet transform". In: *Journal of Neuroscience Methods* 123.1 (2003), pp. 69–87.

[AlF06] A. S. Al-Fahoum. "Quality assessment of ECG compression techniques using a wavelet-based diagnostic measure". In: *IEEE Transactions on Information Technology in Biomedicine* 10.1 (2006), pp. 182–191.

[AndLeh01] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger. "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state". In: *Physical Review E* 64.6, 1 (2001), 1–8.

[AydSar09] S. Aydin, H. M. Saraoğlu, and S. Kara. "Log Energy Entropy-Based EEG Classification with Multilayer Neural Networks in Seizure". In: *Annals of Biomedical Engineering* 37.12 (2009), 2626–2630.

[BerChi00] E. Berti, F. Chiaraluce, N. E. Evans, and J. J. McKee. "Reduction of Walsh-Transformed Electrocardiograms by Double Logarithmic Coding". In: *IEEE Transactions on Biomedical Engineering* 47.11 (2000), pp. 1543–1547.

[BerEng06]   F. van den Bergh and A. P. Engelbrecht. "A study of particle swarm optimization particle trajectories". In: *Information Processing Letters* 176.8 (2006), pp. 937–971.

[BlaCru07]   M. Blanco-Velasco, F. Cruz-Roländ, J. I. Godino-Llorente, and E. K. Barner. "Wavelet packets feasibility study for the design of an ECG compressior". In: *IEEE Transactions on Biomedical Engineering* 54.4 (2007), pp. 766–769.

[BohBar88]   L. N. Bohs and R. C. Barr. "Prototype for real-time adaptive sampling using the fan algorithm". In: *Medical and Biological Engineering and Computing* 26.6 (1988), pp. 574–583.

[BokSch09]   J. Bokor, F. Schipp, and A. Soumelidis. "Applying hyperbolic wavelet construction in the identification of signals and systems". In: *Proceedings of the 15th IFAC Symposium on System Identification (SYSID)*. 2009.

[BreLuc07]   L. Brechet, M. F. Lucas, C. Doncarli, and D. Farina. "Compression of Biomedical Signals with Mother Wavelet Optimization and Best-Basis Wavelet Packet Selection". In: *IEEE Transactions on Biomedical Engineering* 54.12 (2007), pp. 2186–2192.

[BulGon09]   A. Bultheel, P. Gonzalez-Vera, E. Hendriksen, and O. Njåstad. *Orthogonal rational functions*. Cambridge, UK: Cambridge University Press, 2009.

[BulGon99]   A. Bultheel, P. Gonzalez-Vera, E. Hendriksen, and O. Njåstad. "Elements of the theory of orthogonal functions". In: *Revista de la Academia Canaria de Ciencias* 9.1-2 (1999), pp. 127–152.

[BurGop97]   C. S. Burrus, A. R. Gopinath, and H. Guo. "Introduction to Wavelets and Wavelet Transforms: A primer". In: 1st. New Jersey, USA: Prentice Hall, 1997. Chap. 5.

[CarGam02]   S. D. Carley, R. Gamon, P. A. Driscoll, G. Brown, and P. Wallman. "What's the point of ST elevation?" In: *Emergency Medicine Journal* 19.2 (2002), pp. 126–128.

[CasLagne]   F. Castells, P. Laguna, L. Sörnmo, A. Bollmann, and J. M. Roig. "Principal component analysis in ECG signal processing". In: *EURASIP Journal on Advances in Signal Processing* 2007 (2007, [Online]), pp. 1–21.

[CheAxt10]   D. Chetverikov and A. Axt. "Approximation-free running SVD and its application to motion detection". In: *Pattern Recognition Letters* 31.9 (2010), pp. 891–897.

[CheZha10] L. Chen, E. Zhao, D. Wang, Z. Han, S. Zhang, and C. Xu. "Feature extraction of EEG signals from epilepsy patients based on Gabor Transform and EMD Decomposition". In: *Proceedings of the 6th International Conference on Natural Computation (ICNC)*. Vol. 3. 2010, pp. 1243–1247.

[ChoChe06] H. H. Chou, Y. J. Chen, Y. C. Shiau, and T. S. Kuo. "An effective and efficient compression algorithm for ECG signals with irregular periods". In: *IEEE Transactions on Biomedical Engineering* 53.6 (2006), pp. 1198–1205.

[CliAzu06] G. D. Clifford, F. Azuaje, and P. E. McSharry. *Advanced Methods And Tools for ECG Data Analysis*. Massachusetts, USA: Artech House, 2006.

[Cline] G. D. Clifford. *Collection of ECG processing algorithms: Filters*. Available: `http://www.mit.edu/~gari/CODE/FILTERS`. 2007, [Online].

[Coh89] L. Cohen. "Time frequency distributions–a review". In: *Proceedings of the IEEE* 77.7 (1989), pp. 941–981.

[CoiMey92] R. R. Coifman, Y. Meyer, and M. V Wickerhauser. "Wavelet Analysis and Signal Processing". In: *Wavelets and Their Applications*. Ed. by M. B. Ruskai et al. Boston, USA: Jones and Bartlett, 1992, pp. 153–178.

[CooTuk65] J. W. Cooley and J. W. Tukey. "An algorithm for the machine calculation of complex Fourier series". In: *Mathematics of Computation* 19.2 (1965), pp. 297–301.

[CoxNol68] J. R. Cox, F. M. Nolle, H. A. Fozzard, and G. C. Oliver. "AZTEC: preprocessing program for real-time ECG rhythm analysis". In: *IEEE Transactions on Biomedical Engineering* BME-15.2 (1968), pp. 128–129.

[Dau92a] I. Daubechies. *Ten Lectures on Wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.

[Dau92b] I. Daubechies. *Ten lectures on wavelets*. 1st. Philadelphia, USA: Society for Industrial and Applied Mathematics (SIAM), 1992.

[Far11] R. Farber. *CUDA application design and development*. 1st. San Francisco, USA: Morgan Kaufmann Publishers Inc., 2011.

[FirGor08] C. M. Fira and L. Goras. "An ECG signals compression method and its validation using NNs". In: *IEEE Transactions on Biomedical Engineering* 55.4 (2008), pp. 1319–1326.

[FisGol92] B. Fischer and G. H. Golub. "How to generate unknown orthogonal polynomials out of known orthogonal polynomials". In: *Journal of Computational and Applied Mathematics* 43 (1992), pp. 99–115.

[FriCar80]   F. N. Fritsch and R. E. Carlson. "Monotone Piecewise Cubic Interpolation". In: *SIAM Journal on Numerical Analysis* 17.2 (1980), pp. 238–246.

[FriGil13a] S. Fridli, Z. Gilián, and F. Schipp. "Rational Hermite-Fejér Interpolation". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 40 (2013), pp. 233–244.

[FriGil13b] S. Fridli, Z. Gilián, and F. Schipp. "Rational orthogonal systems on the plane". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 39 (2013), pp. 63–77.

[FriJan90]   G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle. "A Comparison of the Noise Sensitivity of Nine QRS Detection Algorithms". In: *IEEE Transactions on Biomedical Engineering* 37.1 (1990), pp. 85–98.

[FriLóc12]   S. Fridli, L. Lócsi, and F. Schipp. "Rational function system in ECG processing". In: *Computer Aided Systems Theory–EUROCAST 2011: Part I.* Ed. by R. Moreno-Díaz et al. Vol. 6927. LNCS. Heidelberg, Germany: Springer-Verlag Berlin, 2012, pp. 88–95.

[FriSch11]   S. Fridli and F. Schipp. "Biorthogonal systems to rational functions". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 35 (2011), pp. 95–105.

[Gab46]   D. Gabor. "Theory of communication. Part I: The analysis of information". In: *Journal of the Institution of Electrical Engineers – Part III: Radio and Communication Engineering* 93.26 (1946), pp. 429–457.

[Gab98]   A. J. Gabor. "Seizure detection using a self-organizing neural network: validation and comparison with other detection strategies". In: *Electroencephalography and Clinical Neurophysiology* 107.1 (1998), pp. 27–32.

[GabLea96]   A. J. Gabor, R. R. Leach, and F. U. Dowla. "Automated seizure detection using a self-organizing neural network". In: *Electroencephalography and Clinical Neurophysiology* 99.3 (1996), pp. 257–266.

[GasWit99]   C. Gasquet and P. Witomski. *Fourier Analysis and Applications: Filtering, Numerical Computation, Wavelets.* New York, USA: Springer-Verlag, 1999.

[Gau04]   W. Gautschi. *Orthogonal Polynomials, Computation and Approximation.* Numerical Mathematics and Scientific Computation. Oxford, UK: Oxford University Press, 2004.

[GeoVal12]   G. Georgiev, I. Valova, N. Gueorguieva, and L. Lei. "QRS Complex Detector Implementing Orthonormal Functions". In: *Procedia Computer Science* 12 (2012), pp. 426–431.

[GhoAde07]   S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr. "Mixed-Band Wavelet-Chaos-Neural Network Methodology for Epilepsy and Epileptic Seizure Detection". In: *IEEE Transactions on Biomedical Engineering* 54.9 (2007), pp. 1545–1551.

[GilKov14]   Z. Gilián, P. Kovács, and K. Samiee. "Rhythm-based Accuracy Improvement of Heart Beat Detection Algorithms". In: *Proceedings of the 41st Annual Conference on Computing in Cardiology (CinC)*. 2014, pp. 269–272.

[Gol06]   A. L. Goldberger. *Clinical Electrocardiography: a Simplified Approach*. 7th. Philadelphia, USA: Mosby Elsevier, 2006.

[GolAma00]   A. L. Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals". In: *Circulation* 101.23 (2000), pp. 215–220.

[Gre93]   M. J. Greenberg. "Euclidean and Non–Euclidean Geometries: Development and History". In: 3rd. New York, USA: W. H. Freeman and Company, 1993. Chap. 7.

[GülÜbe05]   İ. Güler and E. D. Übeyli. "Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients". In: *Journal of Neuroscience Methods* 148.2 (2005), pp. 113–121.

[GuoRiv09]   L. Guo, D. Rivero, J. A. Seoane, and A. Pazos. "Classification of EEG Signals Using Relative Wavelet Energy and Artificial Neural Networks". In: *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. GEC '09. New York, USA: ACM, 2009, pp. 177–184.

[GuoRiv10]   L. Guo, D. Rivero, and A. Pazos. "Epileptic seizure detection using multiwavelet transform based approximate entropy and artificial neural networks". In: *Journal of Neuroscience Methods* 193.1 (2010), pp. 156–163.

[GuyWes02]   I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. "Gene Selection for Cancer Classification using Support Vector Machines". In: *Machine Learning* 46.1–3 (2002), pp. 389–422.

[HämHof91]   G. Hämmerlin and K. H. Hoffmann. *Numerical Mathematics*. New York, USA: Springer-Verlag, 1991.

[HamTom91]   P. S. Hamilton and W. J. Tompkins. "Compression of the ambulatory ECG by average beat subtraction and residual differencing". In: *IEEE Transactions on Biomedical Engineering* 38.3 (1991), pp. 253–259.

[HarEde04]   H. Haraldsson, L. Edenbrandt, and M. Ohlsson. "Detecting acute myocardial infarction in the 12-lead ECG using Hermite expansions and neural networks". In: *Artificial Intelligence in Medicine* 32 (2004), pp. 127–136.

[Hen74]   P. Henrici. *Applied and Computational Complex Analysis. Power Series–Integration–Conformal Mapping–Location of Zeros.* Vol. 1. New York, USA: John Wiley & Sons, 1974.

[HeuVan05]   P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlberg. *Modelling and Identification with Rational Orthogonal Basis Functions.* London, UK: Springer-Verlag, 2005.

[Hil97]   M. L. Hilton. "Wavelet and wavelet packet compression of electrocardiograms". In: *IEEE Transactions on Biomedical Engineering* 44.5 (1997), pp. 402–444.

[Hou70]   A. S. Householder. *The Numerical Treatment of a Single Nonlinear Equation.* New York, USA: McGraw-Hill, 1970.

[IshShi83]   M. Ishijima, S. B. Shin, G. H. Hostetter, and J. Sklansky. "Scan-along polygonal approximation for data compression of electrocardiograms". In: *IEEE Transactions on Biomedical Engineering* BME-30.11 (1983), pp. 723–729.

[JalHut90]   S. M. S. Jalaleddine, C. G. Hutchens, R. D. Strattan, and W. A. Coberly. "ECG data compression techniques – a unified approach". In: *IEEE Transactions on Biomedical Engineering* 37.4 (1990), pp. 329–343.

[JanOlm93]   R. Jané, S. Olmos, P. Laguna, and P. Caminal. "Adaptive Hermite Models for ECG Data Compression: Performance and Evaluation with Automatic Wave Detection". In: *Proceedings of the International Conference on Computers in Cardiology.* 1993, pp. 389–392.

[KarMon97]   M. Karczewicz and G. Moncef. "ECG data compression by spline approximation". In: *Signal Processing* 59 (1997), pp. 43–59.

[KayUya14]   Y. Kaya, M. Uyar, R. Tekin, and S. Yıldırım. "1D-local binary pattern based feature extraction for classification of epileptic EEG signals". In: *Applied Mathematics and Computation* 243 (2014), pp. 209–219.

[KenEbe95]   J. Kennedy and R. C. Eberhart. "Particle swarm optimization". In: *Proceedings of IEEE International Conference on Neural Networks.* Vol. 4. 1995, pp. 1942–1948.

[KirInc09]   S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj. "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization". In: *Neural Networks* 22.10 (2009), pp. 1448–1462.

[KirInc14]   S. Kiranyaz, T. Ince, and M. Gabbouj. *Multidimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition.* Vol. 15. Adaptation, Learning, and Optimization. Heidelberg, Germany: Springer-Verlag Berlin, 2014.

[KirPul11]   S. Kiranyaz, J. Pulkkinen, A. Yildirim, and M. Gabbouj. "Multi-dimensional particle swarm optimization in dynamic environments". In: *Expert Systems with Applications* 38.3 (2011), pp. 2212–2223.

[Koh89]   T. Kohonen. *Self-organization and associative memory*. 3rd. New York, USA: Springer-Verlag, 1989.

[KulPer11]   K. A. H. Kulasuriya and M. U. S. Perera. "Forecasting epileptic seizures using EEG signals, wavelet transform and artificial neural networks". In: *Proceedings of the International Symposium on IT in Medicine and Education (ITME)*. Vol. 1. 2011, pp. 557–562.

[LagPet00]   M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandth, and L. Sörnmo. "Clustering ECG Complexes Using Hermite Functions and Self-Organizing Maps". In: *IEEE Transactions on Biomedical Engineering* 47.7 (2000), pp. 838–717.

[LagRee98]   J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. "Convergence properties of the Nelder–Mead algorithm in low dimensions". In: *SIAM Journal of Optimization* 9.1 (1998), pp. 112–147.

[LeeBuc06]   H. Lee and K. M. Buckley. "ECG data compression using cut and align beats approach and 2-D transform". In: *IEEE Transactions on Biomedical Engineering* 46.5 (2006), pp. 1198–1205.

[Lóc09]   L. Lócsi. "Approximating poles of complex rational functions". In: *Acta Univ. Sapientiae, Mathematica* 1.2 (2009), pp. 169–182.

[Lóc11]   L. Lócsi. "Calculating Non-Equidistant Discretizations Generated by Blaschke Products". In: *Acta Cybernetica* 20 (2011), pp. 111–123.

[Lóc13]   L. Lócsi. "A hyperbolic variant of the Nelder–Mead simplex method in low dimensions". In: *Acta Univ. Sapientiae, Mathematica* 5.2 (2013), pp. 169–183.

[Mal08]   S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd. Burlington, USA: Academic Press, 2008.

[McSCli03]   P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith. "A Dynamical Model for Generating Synthetic Electrocardiogram Signals". In: *IEEE Transactions on Biomedical Engineering* 50.3 (2003), pp. 289–294.

[Mit11]   S. K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. 4th. New York, USA: McGraw-Hill, 2011.

[Mór13]   F. Móricz. *Harmónikus analízis a komplex egységkörön*. Szeged, HU: Polygon, 2013.

[MorBra08]   F. Morris, J. W. Brady, and J. Camm. *ABC of Clinical Electrocardiography*. 2nd. Oxford, UK: Blackwell Publishing, 2008.

[Mue78]   W. C. Mueller. "Arrhythmia detection program for an ambulatory ECG monitor". In: *Biomedical sciences instrumentation* 14 (1978), pp. 81–85.

[NasPou11] S. Nasehi and H. Pourghassem. "Real-Time Seizure Detection Based on EEG and ECG Fused Features Using Gabor Functions". In: *Proceedings of the International Conference on Intelligent Computation and Bio-Medical Instrumentation (ICBMI)*. 2011, pp. 204–207.

[Nat52]   I. P. Natanszon. *Konstruktív függvénytan*. Ed. by B. Szőkefalvi-Nagy. Trans. by A. Rényi. Budapest, Hungary: Akadémiai Kiadó, 1952.

[NavCoh93] G. Nave and A. Cohen. "ECG compression using long-term prediction". In: *IEEE Transactions on Biomedical Engineering* 40.9 (1993), pp. 877–885.

[NelMea65] J. A. Nelder and R. Mead. "A simplex method for function minimization". In: *Computer Journal* 7.4 (1965), pp. 308–313.

[NicGeo12] N. Nicolaou and J. Georgiou. "Detection of epileptic electroencephalogram based on Permutation Entropy and Support Vector Machines". In: *Expert Systems with Applications* 39.1 (2012), pp. 202–209. (Visited on 07/30/2014).

[Oca08]   H. Ocak. "Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm". In: *Signal Processing* 88.7 (2008), pp. 1858–1867. (Visited on 03/14/2014).

[PapSch01] M. Pap and F. Schipp. "Malmquist–Takenaka systems and equilibrium conditions". In: *Mathematica Pannonica* 12.1 (2001), pp. 185–194.

[PapSch04] M. Pap and F. Schipp. "Interpolation by rational functions". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 24 (2004), pp. 223–237.

[PolGün07] K. Polat and S. Güneş. "Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform". In: *Applied Mathematics and Computation* 187.2 (2007), pp. 1017–1026.

[PolGün08] K. Polat and S. Güneş. "Artificial immune recognition system with fuzzy resource allocation mechanism classifier, principal component analysis and FFT method based new hybrid automated identification system for classification of EEG signals". In: *Expert Systems with Applications* 34.3 (2008), pp. 2039–2048.

[PraSri10] K. S. Pravin, N. Sriraam, P. G. Benakop, and B. C. Jinaga. "Entropies based detection of epileptic seizures with artificial neural network classifiers". In: *Expert Systems with Applications* 37.4 (2010), pp. 3284–3291.

[QuiNad04]   R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul. "Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering". In: *Neural Computation* 16.8 (2004), pp. 1661–1687.

[RicVen04]   M. Richards and D. Ventura. "Choosing a starting configuration for particle swarm optimization". In: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*. 2004, pp. 2309–2312.

[RioSah13]   L. M. Rios and N. V. Sahinidis. "Derivative-free optimization: a review of algorithms and comparison of software implementations". In: *Journal of Global Optimization* 56.3 (2013), pp. 1247–1293.

[SaiPea96]   A. Said and W. A. Pearlman. "A new, fast, and efficient image codec based on set partitioning in hierarchical trees". In: *IEEE Transactions on Circuits and Systems for Video Technology* 6.3 (1996), pp. 243–250.

[SamKir15]   K. Samiee, S. Kiranyaz, M. Gabbouj, and T. Saramäki. "Long-term epileptic EEG classification via 2D mapping and textural features". In: *Expert Systems with Applications* 42.20 (2015), pp. 7175–7185.

[SanKan10]   J. Sanders and E. Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. 1st. New York, USA: Addison-Wesley Professional, 2010.

[SanSab12]   A. Sandryhaila, S. Saba, M. Püschel, and J. Kovacevic. "Efficient Compression of QRS Complexes Using Hermite Expansion". In: *IEEE Transactions on Signal Processing* 60.2 (2012), pp. 947–955.

[SaySha10]   O. Sayadi, M. B. Shamsollahi, and G. D. Clifford. "Synthetic ECG Generation and Bayesian Filtering Using a Gaussian Wave-Based Dynamical Model". In: *Physiological Measurement* 31.10 (2010), pp. 1309–1329.

[Sch02]   F. Schipp. "Fast Fourier transform for rational systems". In: *Mathematica Pannonica* 12.2 (2002), pp. 265–275.

[Sch14]   F. Schipp. "Hyperbolic Wavelets". In: *Topics in Mathematical Analysis and Applications*. Ed. by T. Rassias and L. Tóth. Vol. 94. Optimization and Its Applications. Heidelberg, Germany: Springer-Verlag Berlin, 2014, pp. 633–657.

[SchBok98]   F. Schipp and J. Bokor. "Approximate Identification in Laguerre and Kautz Bases". In: *Automatica* 34.4 (1998), pp. 463–468.

[SchSou11]   F. Schipp and A. Soumelides. "On the Fourier coefficients with respect to the discrete Laguerre system". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 34 (2011), pp. 223–233.

[SchWad90]   F. Schipp, W. R. Wade, and P. Simon. *Walsh Series, An Introduction to Dyadic Harmonic Analysis*. Budapest, HU: Akadémiai Kiadó, 1990.

[Sha93]   J. M. Shapiro. "Embedded image coding using zerotrees of wavelet coefficients". In: *IEEE Transactions on Signal Processing* 41.12 (1993), pp. 3445–3462.

[SheCha11] C.-P. Shen et al. "Epileptic Seizure Detection for Multichannel EEG Signals with Support Vector Machines". In: *Proceedings of the 11th IEEE International Conference on Bioinformatics and Bioengineering.* 2011, pp. 39–43.

[ShiEbe98] Y. Shi and R. Eberhart. "A modified particle swarm optimizer". In: *Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (WCCI).* 1998, pp. 69–73.

[SörBör81] L. Sörnmo, P. L. Börjesson, M. E. Nygårds, and O. Pahlm. "A Method for Evaluation of QRS Shape Features Using a Mathematical Model for the ECG". In: *IEEE Transactions on Biomedical Engineering* 28 (1981), pp. 713–717.

[SouBok13] A. Soumelidis, J. Bokor, and F. Schipp. "An iterative identification of pole-stucture in dynamic systems based on hyperbolic metrics and Malmquist-Takenaka representation". In: *Proceedings of the 52nd IEEE Annual Conference on Decision and Control (CDC).* 2013, pp. 5180–5185.

[SouPap02] A. Soumelidis, M. Pap, F. Schipp, and J. Bokor. "Frequency domain identification of partial fraction models". In: *Proceedings of the 15th IFAC World Congress.* Vol. 15. 2002, pp. 465–471.

[SouSch02] A. Soumelidis, F. Schipp, and J. Bokor. "Frequency Domain Representation of Signals in Rational Orthogonal Bases". In: *Proceedings of the 10th IEEE Mediterranean Conference on Control and Automation (MED).* 2002.

[SouSch11a] A. Soumelidis, F. Schipp, and J. Bokor. "On hyperbolic wavelets". In: *Proceedings of the 18th IFAC World Congress.* Vol. 18. 2011, pp. 2309–2314.

[SouSch11b] A. Soumelidis, F. Schipp, and J. Bokor. "Pole Structure Estimation from Laguerre Representations Using Hyperbolic Metrics on the Unit Disc". In: *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC).* 2011, pp. 2136–2141.

[SriEsw05] V. Srinivasan, C. Eswaran, and N. Sriraam. "Artificial Neural Network Based Epileptic Detection Using Time-Domain and Frequency-Domain Features". In: *Journal of Medical Systems* 29.6 (2005), pp. 647–660.

[SriEsw07]   V. Srinivasan, C. Eswaran, and N. Sriraam. "Approximate Entropy-Based Epileptic EEG Detection Using Artificial Neural Network". In: *IEEE Transactions on Information Technology in Biomedicine* 11.3 (2007), pp. 288–295.

[Sub07]   A. Subasi. "EEG signal classification using wavelet feature extraction and a mixture of expert model". In: *Expert Systems with Applications* 32.4 (2007), pp. 1084–1093.

[SülMay03]   E. Süli and D. Mayers. *An Introduction to Numerical Analysis.* Cambridge, UK: Cambridge University Press, 2003.

[SunTai05]   C. C. Sun and S. C. Tai. "Beat-Based ECG compression using gain-shape vector quantization". In: *IEEE Transactions on Biomedical Engineering* 52.11 (2005), pp. 1882–1888.

[Sza04]   Z. Szabó. "Interpolation and quadrature formulae for rational systems on the unit circle". In: *Annales Univ. Sci. Budapest., Sect. Comp.* 21 (2004), pp. 41–56.

[Sze67]   G. Szegő. *Orthogonal polynomials.* 3rd. New York, USA: AMS Colloquium Publications, 1967.

[Tay78]   J. M. Taylor. "The condition of gram matrices and related problems". In: *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 80.1-2 (1978), pp. 45–56.

[Tre03]   I. C. Trelea. "The particle swarm optimization algorithm: convergence analysis and parameter selection". In: *Information Processing Letters* 85.6 (2003), pp. 317–325.

[TzaTsi07]   A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis. "Automatic Seizure Detection Based on Time-Frequency Analysis and Artificial Neural Networks". In: *Computational Intelligence and Neuroscience* 2007 (2007), pp. 1–13.

[TzaTsi09]   A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis. "Epileptic Seizure Detection in EEGs Using Time–Frequency Analysis". In: *IEEE Transactions on Information Technology in Biomedicine* 13.5 (2009), pp. 703–710.

[TzaTsi12]   A. T. Tzallas et al. "Automated Epileptic Seizure Detection Methods: A Review Study". In: *Epilepsy - Histological, Electroencephalographic and Psychological Aspects.* Ed. by D. Stevanovic. Rijeka, Croatia: InTech, 2012, pp. 75–98.

[WanYua97]   B. Wang and G. Yuan. "Compression of ECG data by vector quantization". In: *IEEE Transactions on Biomedical Engineering* 40.7 (1997), pp. 23–26.

[WorPar04]   G. A. Worrell, L. Parish, S. D. Cranstoun, R. Jonas, G. Baltuch, and B. Litt. "High-frequency oscillations and seizure generation in neocortical epilepsy". In: *Brain* 127.7 (2004), pp. 1496–1506.

[XieKri13]   S. Xie and S. Krishnan. "Wavelet-based sparse functional linear model with applications to EEGs seizure detection and epilepsy diagnosis". In: *Medical & Biological Engineering & Computing* 51.1–2 (2013), pp. 49–60.

[XieSha08]   S. Xie, S. Shan, X. Chen, and W. Gao. "V-LGBP: Volume based local Gabor binary patterns for face representation and recognition". In: *Proceedings of the 19th International Conference on Pattern Recognition (ICPR)*. 2008, pp. 1–4.

[ZigCoh00]   Y. Zigel, A. Cohen, and A. Katz. "The weighted diagnostic distortion (WDD) measure for ECG signal compression". In: *IEEE Transactions on Biomedical Engineering* 47.11 (2000), pp. 1424–1430.

[Zyg59]   A. Zygmund. *Trigonometric series I. - II.* Cambridge, UK: Cambridge University Press, 1959.

Transformation methods in signal processing

Péter Kovács

SUMMARY OF THE PHD DISSERTATION

This dissertation is concerned with the application of the theory of rational functions in signal processing. The PhD thesis summarizes the corresponding results of the author's research.

Since the systems of rational functions are defined by the collection of inverse poles with multiplicities, the following parameters should be determined: the number, the positions and the multiplicities of the inverse poles. Therefore, we develop the hyperbolic variant of the so-called Nelder–Mead and the particle swarm optimization algorithm. In addition, the latter one is integrated into a more general multi-dimensional framework. Furthermore, we perform a detailed stability and error analysis of these methods.

We propose an electrocardiogram signal generator based on spline interpolation. It turns to be an efficient tool for testing and evaluating signal models, filtering techniques, etc. In this thesis, the synthesized heartbeats are used to test the diagnostic distortion of the rational representation. In addition, we introduce a mathematical model for the multi-channel QRS complex.

We construct a new, generalized time-frequency distribution of signals based on different types of rational functions. This method is applied for the problem of epileptic seizure detection in electroencephalography. The effectiveness of our algorithm is compared to several state-of-the-art feature extraction methods.

Numerical methods for computing rational series expansions of signals are also presented in a MATLAB toolbox. We provide a fast algorithm to construct discrete rational orthogonal functions. Then, these systems are used to represent discrete-time series which are compared with usual signal compression techniques.

The main material of the dissertation has been published in 4 journal and 5 conference papers, and presented at 10 international conferences. The conference papers have appeared in IEEE proceedings.

Budapest, 2016

Transzformációs módszerek a jelfeldolgozásban

Kovács Péter

DOKTORI ÉRTEKEZÉS ÖSSZEFOGLALÁSA

Az értekezés témája a racionális függvényrendszerek vizsgálata és alkalmazása a jelfeldolgozás területén. A disszertációban ismertetem az egyes problémaköröket, illetve bemutatom az ezek megoldására kidolgozott módszereket.

A racionális függvények esetén a rendszert az inverz pólusok sorozata egyértelműen meghatározza. Így az adaptív racionális reprezentációk konstrukciójához a következő szabad paraméterek optimalizálására van szükség: inverz pólusok száma, értéke és multiplicitása. A probléma megoldására a dolgozatban a Nelder–Mead szimplex és az ún. Particle Swarm Optimization algoritmusok hiperbolikus változatát használtam. Továbbá kidolgoztam egy olyan többdimenziós módszert, mellyel az inverz pólusok értékei mellett azok optimális száma és multiplicitása is meghatározható. Az értekezésben részletesen elemzem az említett módszerek stabilitását és hibáját.

Az értekezésben kidolgoztam egy spline interpoláción alapuló elektrokardiogram (EKG) generátort. Ennek segítségével megvizsgáltam különböző EKG modellek numerikus és diagnosztikai hibáját, melyeket összehasonlítottam a racionális rendszerekkel. A disszertációban többcsatornás felvételek QRS komplexusát is modelleztem, melyhez elemi racionális függvényeket használtam.

Az értekezésben a diszkrét rövid idejű Fourier-transzformáció analógiáját követve kidolgoztam egy adaptív idő-frekvencia felbontást. Ennek segítségével elektroenkefalogram jelekben lehet epilepsziás rohamokat detektálni. A módszert több state-of-the-art algoritmussal is összehasonlítom.

A disszertációban tárgyalt racionális függvényekhez és a kapcsolódó transzformációkhoz egy önálló MATLAB eszköztárat készítettem. Ennek segítségével megvalósítottam egy racionális EKG tömörítő eljárást, melyet az irodalomban jól ismert technikákkal együtt teszteltem. Továbbá egy új, párhuzamosított módszert adtam a diszkrét racionális ortogonális rendszerek szerkesztésére.

Az értekezésben saját, illetve társszerzőkkel írt publikációimat összegeztem. Ebből 4 folyóirat, 5 pedig konferencia közlemény, melyek IEEE proceedingsben jelentek meg. Az eredményeket 10 nemzetközi konferencián prezentáltam.

Budapest, 2016

[1]ADATLAP
a doktori értekezés nyilvánosságra hozatalához

I. A doktori értekezés adatai
A szerző neve: Kovács Péter ....
MTMT-azonosító: 10035108 ....
A doktori értekezés címe és alcíme: Transformation methods in signal processing.................
DOI-azonosító[2]:10.15476/ELTE.2015.187 A doktori iskola neve: ELTE IK, Informatika doktori iskola...
A doktori iskolán belüli doktori program neve: Numerikus, szimbolikus számítások..............
A témavezető neve és tudományos fokozata: Dr. Fridli Sándor, tanszékvezető egyetemi tanár, az MTA doktora...
A témavezető munkahelye: ELTE IK, Numerikus analízis tanszék....................................

II. Nyilatkozatok
1. A doktori értekezés szerzőjeként[3]

a) hozzájárulok, hogy a doktori fokozat megszerzését követően a doktori értekezésem és a tézisek nyilvánosságra kerüljenek az ELTE Digitális Intézményi Tudástárban. Felhatalmazom az ELTE IK, Informatika Doktori Iskola hivatalának ügyintézőjét Ríz-Herczeg Krisztinát, hogy az értekezést és a téziseket feltöltse az ELTE Digitális Intézményi Tudástárba, és ennek során kitöltse a feltöltéshez szükséges nyilatkozatokat.

b) kérem, hogy a mellékelt kérelemben részletezett szabadalmi, illetőleg oltalmi bejelentés közzétételéig a doktori értekezést ne bocsássák nyilvánosságra az Egyetemi Könyvtárban és az ELTE Digitális Intézményi Tudástárban;[4]

c) kérem, hogy a nemzetbiztonsági okból minősített adatot tartalmazó doktori értekezést a minősítés (*dátum*)-ig tartó időtartama alatt ne bocsássák nyilvánosságra az Egyetemi Könyvtárban és az ELTE Digitális Intézményi Tudástárban;[5]

d) kérem, hogy a mű kiadására vonatkozó mellékelt kiadó szerződésre tekintettel a doktori értekezést a könyv megjelenéséig ne bocsássák nyilvánosságra az Egyetemi Könyvtárban, és az ELTE Digitális Intézményi Tudástárban csak a könyv bibliográfiai adatait tegyék közzé. Ha a könyv a fokozatszerzést követőn egy évig nem jelenik meg, hozzájárulok, hogy a doktori értekezésem és a tézisek nyilvánosságra kerüljenek az Egyetemi Könyvtárban és az ELTE Digitális Intézményi Tudástárban.[6]
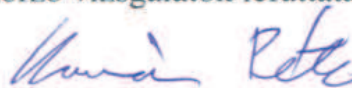
2. A doktori értekezés szerzőjeként kijelentem, hogy

a) az ELTE Digitális Intézményi Tudástárba feltöltendő doktori értekezés és a tézisek saját eredeti, önálló szellemi munkám és legjobb tudomásom szerint nem sértem vele senki szerzői jogait;

b) a doktori értekezés és a tézisek nyomtatott változatai és az elektronikus adathordozón benyújtott tartalmak (szöveg és ábrák) mindenben megegyeznek.

3. A doktori értekezés szerzőjeként hozzájárulok a doktori értekezés és a tézisek szövegének plágiumkereső adatbázisba helyezéséhez és plágiumellenőrző vizsgálatok lefuttatásához.

Kelt: 2015-11-16

a doktori értekezés szerzőjének aláírása

---

[1] Beiktatta az Egyetemi Doktori Szabályzat módosításáról szóló CXXXIX/2014. (VI. 30.) Szen. sz. határozat. Hatályos: 2014. VII.1. napjától.
[2] A kari hivatal ügyintézője tölti ki.
[3] A megfelelő szöveg aláhúzandó.
[4] A doktori értekezés benyújtásával egyidejűleg be kell adni a tudományági doktori tanácshoz a szabadalmi, illetőleg oltalmi bejelentést tanúsító okiratot és a nyilvánosságra hozatal elhalasztása iránti kérelmet.
[5] A doktori értekezés benyújtásával egyidejűleg be kell nyújtani a minősített adatra vonatkozó közokiratot.
[6] A doktori értekezés benyújtásával egyidejűleg be kell nyújtani a mű kiadásáról szóló kiadói szerződést.