Adaptive and Blind Array Processing Techniques for Extracellular Electrode Recordings

vorgelegt von

Diplom-Physiker
Michal Natora
aus Schwerzenbach, Schweiz

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
- Dr. rer. nat. -

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr. Reinhold Orglmeister
Berichter: Prof. Dr. Klaus Obermayer
Berichter: Prof. Dr. Aapo Hyvärinen

Tag der wissenschaftlichen Aussprache: 08. März 2011

Berlin, 2011

D 83

To my beautiful Małgosia

Acknowledgments

I would like to thank my supervisor Prof. Dr. Klaus Obermayer from the Neural Information Processing Group at the TU Berlin for enabling me to carry out the research in his group and for supporting the whole project. I also thank him for his scientific and organisational advice, as well as for the opportunity to present my work at various conferences.

I am grateful to Prof. Dr. Aapo Hyvärinen from the University of Helsinki. Although I had only few occasions to meet him in person, he gladly accepted to be a reviewer of my thesis and provided me with helpful comments.

I would like to thank all the people participating in the project, especially Felix Franke with whom I worked on some problems and their solutions, mainly concerning the spike sorting and positioning algorithm, and Sven Dähne and Philipp Meier who helped a lot with the technical part of the project. I thank also the project partners from the Max Planck Institute in Tübingen and the Thomas Recording GmbH for the fruitful collaboration resulting in a successful overall project.

The group of Prof. Gaute T. Einevoll (Norwegian University of Life Sciences) provided us data of simulated extracellular action potentials, which was crucial for evaluating the positioning algorithm. I sincerely thank him and the members of his group for this effort.

I am thankful to Prof. Clemens Boucsein (University of Freiburg) for giving us access to his simultaneous intra/extra-cellular recordings. These data helped much to evaluate the spike detection algorithm.

I would like to express my gratitude to Prof. Simon Broda (Universiteit van Amsterdam). Not only did he provide me with code for fast cdf evaluation, but he gave me valuable hints and comments as well.

A special thankyou goes to Steffen Grünewälder with whom I shared an office when I started work in the NI group. He taught me how things work and are done in the NI group, and provided me with valuable information regarding my project. Coffee breaks and other less scientific activities we did together will always stay as good

memories.

I thank all my colleagues from the NI group and the Bernstein Centre for Computational Neuroscience for creating a very pleasant work climate, that encouraged scientific and social interaction between all members.

A personal thank you is dedicated to my parents and Małgorzata M. Wójcik who showed great interest in the project and supported me throughout.

This research was supported by the Bundesministerium für Bildung und Forschung with the grant 01GQ0743.

Abstract

Electrophysiological recordings with electrodes, or more generally, with arrays of multi-electrodes, are key for recording neural activity data from the central nervous system. This technique delivers high temporal and spatial resolution, as well as enables neuron stimulation by current injection. The neuronal activity encoded by action potentials (simply called ”spikes”) of individual neurons, however, is not recorded directly; rather the measurement contains a mixture of spike trains from several neurons and additional noise. To determine the spiking times of a neuron and to determine a spike's originating neuron, spike detection and spike sorting algorithms are needed. The main focus of this thesis is the development of such algorithms.

The system consisting of neurons emitting spike trains, their mixture and corruption by noise, and of the process of recording these data with several electrodes channels, is modelled as a linear time-invariant multiple input, multiple output system. The problem of spike detection/sorting can then be regarded as a blind equalisation and source separation task. We use finite impulse response filters for equalisation and source separation throughout the thesis, and therefore, we first start with analysing some properties of these filters. Amongst others, their performance in terms of detection probability and false alarm probability is studied in the case when the spike waveform is perfectly known, and when it is estimated from the data themselves. The subsequently presented spike detection and sorting algorithms are two stage algorithms, consisting of a system identification phase and the following equalisation/separation. Common to them is that both stages can be performed with minimal human supervision although the spatial mixing and temporal distortion are unknown, and the ability to adapt to changing waveforms during the equalisation/separation stage. As such they can be termed as adaptive and blind array processing techniques. Finally, we also propose an unsupervised control algorithm for electrodes, which allows to move them to favourable recording sites. This closes the loop, as the system can now perform spike detection/sorting at any position and decides by itself whether to move the electrode to a more promising position or whether current quality of data is sufficient.

Zusammenfassung

Elektrophysiologische Ableitungen mit Elektroden, oder allgemeiner, mit einer ganzen Matrix von Multi-Elektroden, sind eine Schlüsseltechnik um neuronale Aktivitätsdaten aus dem zentralen Nervensystem aufzunehmen. Diese Technik liefert eine hohe zeitliche als auch räumliche Auflösung, und erlaubt sogar Neuronenstimulation mittels Injektion von elektrischem Strom. Die neuronale Aktivität, enkodiert durch Aktionspotentiale (auch genannt ”Spikes”), von einzelnen Neuronen wird jedoch nicht direkt aufgenommen; vielmehr enthält die Messung eine Mixtur von mehreren Spike Folgen verschiedener Neuronen und zusätzliches Rauschen. Um die einzelnen Spike Zeitpunkte eines Neurons und um das Herkunftsneuron eines Spikes zu bestimmen, sind Spike Detektions- und Spike Sortierungs-algorithmen notwendig.

Das System bestehend aus Spike Folgen generierenden Neuronen, deren Mixtur und die Korruption durch Rauschen, und aus dem Prozess des Messens dieser Daten mit mehreren Elektrodenkanälen, kann als ein lineares zeitinvariantes Multieingang/Multiausgang-System modelliert werden. Das Problem der Spike Detektion/Sortierung kann dann als ein blindes Entzerrungs- und Quellentrennungsproblem aufgefasst werden. Wir benutzen in dieser Arbeit immer endliche Impulsantwortsfilter für die Entzerrung und Quellentrennung, deshalb beginnen wir mit der Analyse einiger Eigenschaften dieser Filter. Unter anderem, analysieren wir deren Leistungsfähigkeit im Bezug auf die Detektionswahrscheinlichkeit und Falschalarmwahrscheinlichkeit wenn die Spike Funktion bekannt ist, aber auch wenn diese von den Daten geschätzt wird. Die nachfolgend präsentierten Spike Detektion und Sortierungsverfahren sind Zweistufenalgorithmen, bestehend aus einer Systemidentifikationsphase und einer darauffolgenden Entzerrung/Quellentrennung. Beide Verfahren sind sich insofern ähnlich, als dass beide Phasen nur minimalen menschlichen Eingriff verlangen obwohl die räumliche Mixtur und die zeitliche Verzerrung unbekannt sind, und dass beide Verfahren sich ändernden Spike Funktionen anpassen können. Deshalb können diese Verfahren allgemein als adaptive und blinde Matrixverarbeitungstechniken bezeichnet werden. Zuletzt, präsentieren wir auch einen unüberwachten Kontrolalgorithmus für Elektroden, welcher die Elektroden zu günstigen Aufnahmestellen bewegt. Das schliesst den Kreis, da nun das System an jeder beliebigen Position Spike Detektion/Sortierung ausführen kann und selbst entscheidet, ob die Elektrode zu einer vielversprechender Position zu bewegen ist, oder ob die momentane Signalqualität ausreichend ist.

List of Symbols and Abbreviations

Abbreviation	Description	Definition
$\boldsymbol{x}$	constant vector	page 12
$\boldsymbol{x}[t]$	time dependent vector	page 12
$\boldsymbol{x}_n = x_n = x(n)$	vector entry at dimension n	page 12
L_x	maximum index value, i.e. $\boldsymbol{x}_n$, $n = -L_x, \ldots, L_x$	page 12
T_x	dimension of vector $\boldsymbol{x}$, i.e. $T_x = 2L_x + 1$	page 12
$\boldsymbol{x} * \boldsymbol{y}$	convolution between $\boldsymbol{x}$ and $\boldsymbol{y}$	page 12
$\boldsymbol{x} \star \boldsymbol{y}$	cross correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$	page 12
$\|\boldsymbol{x}\|_p$	p-norm of vector $\boldsymbol{x}$	page 14
$\boldsymbol{C}$	noise covariance matrix	page 4
$\boldsymbol{D}_{m,n}$, $(\boldsymbol{D})_{m,n}$	matrix entry in m-th row and n-th column	page 12
$\boldsymbol{p}$	nominal steering vector	page 32
$\boldsymbol{q}$	actual steering vector	page 32
$\mathbb{E}[\cdot]$, $\langle\cdot\rangle$	expectation operator	page 11
M	number of sources/transmitters (e.g. neurons)	page 3
N	number of sensors/receivers (e.g. electrodes)	page 3
γ^i	threshold for filter output of filter $\boldsymbol{f}^i$	page 13
$Q(u)$	quality of data at position u	page 97
MPDR	minimum power distortionless response	page 22
MVDR	minimum variance distortionless response	page 22

Contents

Chapter 1

Introduction

The need for understanding the information processing mechanisms of the brain makes the availability of brain activity data essential in order to derive working principles or test existing theories about it. In contrast to many other fields, direct measurements of the activity of individual neurons or their connectivity structure is rarely possible. This is due the fact that invasive techniques are only of limited use in living organisms (and most often still do not allow for direct single cell recordings), whereas non-invasive techniques are not (yet) precise enough to resolve single neuron activity. This constraint of indirect measurement suggests that powerful algorithms must be at hand to allow for the reconstruction of the neural activity of individual cells.

One particular recording technique is the electrophyisological electrode. Starting with inserting single microelectrodes, this technique has evolved significantly in the recent years and is widely used for obtaining high quality data from living animals. One of the reasons of its popularity is the fact that this technique offers a very high temporal resolution as well as spatial resolution. This means that, given appropriate algorithms, individual action potentials of neurons are resolved allowing for studying firing statistics accurately, and, since activity from several neighbouring cells is recorded simultaneously, this gives the possibility to analyse the influence of cells on their neighbourhood (such as locking, burst propagation). Although many other techniques appeared for recording brain activity data, such as functional magnetic resonance imaging or electroencephalography, electrophyisological electrodes continue to be one the major recording tools. Amongst other, one of the reasons for that are the recent technological developments which add many further advantages. These include for example the development of multi-channel electrodes, such as tetrodes, octotodes, etc., see Fig. 1.1. The additional recording channels not only allow for recordings from more neurons simultaneously raising the information yield per experiment, but also improve the quality of the subsequent processing. Depending on the specific physical configuration of the individual channels, simultaneous recordings from more than a single tissue layer are possible as well, see Fig 1.1.

Nowadays, many laboratories use entire multi-channel electrode arrays (MEA). Again, such arrays can be used in order to record from several layers, or, on the other

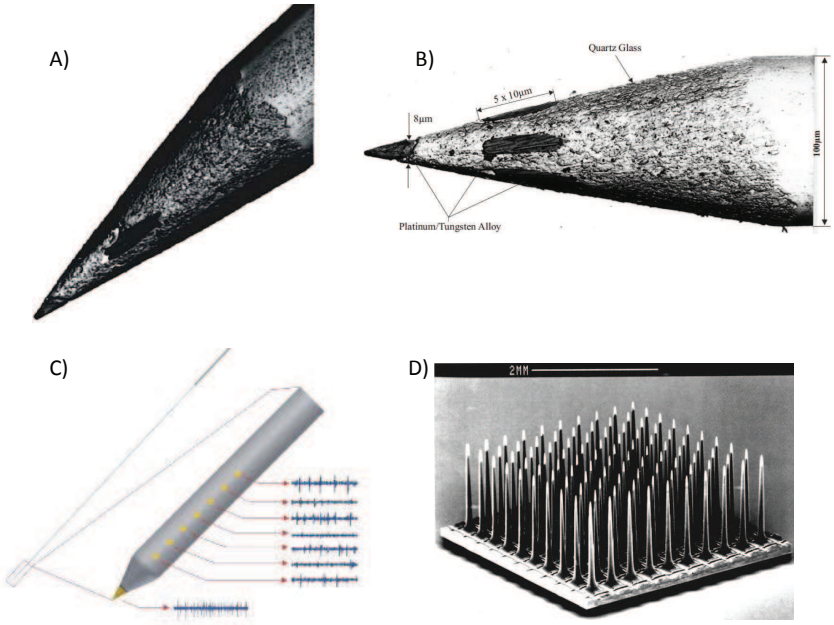

Figure 1.1: *A):* An electrode with 4 recording channels, called tetrode. *B):* A heptode, an electrode with 7 recording channels. *C):* An electrode with 8 recording channels for recording from several layers. All the pictures are from [205]. *D):* A multi-electrode array containing 100 single channel electrodes (from [235]).

hand, to acquire accurate information from many neurons belonging to the same subnetwork within a specific layer or region. Especially the latter possibility is becoming increasingly interesting as a means to verify network computing phenomena (ensemble coding) or to link network topological aspects of the brain structure to modern graph respectively complex network theories [197].

Even more promising are arrays which are directly implanted into the brain, a prominent example being the Utah array [133]. This allows for constant monitoring of specimens, and, combined with wireless transmission technology, for experiments outside the classical fixed laboratory setup. Such implantable electrodes will also certainly play a key role in the development of naturally controlled prosthesis and in next generation brain-computer interface (BCI) devices as well as in curing of brain diseases.

This latter aspect was enabled by the development of micro stimulation electrodes [205]. Such devices are not only able to record passively, but can actively induce well controlled electric currents into the brain tissue which stimulate the surrounding neurons. Once the neural code of a certain brain region is understood, such two-sided communication between the brain and an external devices would allow for fascinating applications.

In short, the electrophysiological electrodes technique is likely to stay and further evolve as one of the main recording tools. Hence, it is important to have algorithms at hand, which can handle the acquired data efficiently and extract the maximum possible information from them. Some major challenges are the following:

- How can the action potentials within continuous electrode recordings be optimally

detected? Errors in spike detection will propagate through all the subsequent analysis and may lead to wrong conclusions about the whole information processing principles of the brain. Moreover, most BCI devices can work reliably only if the precise firing times of the neurons are known.

- In order to make full use of the micro stimulation electrodes real-time action potential detection and classification algorithm must be available. Only then the neural information can be decoded online, and an appropriate micro stimulation sequence induced.

- In both, implanted electrode arrays (chronic recordings), as well as external MEA (acute recordings), the individual electrodes are more and more often controlled by electric motors which allow very precise electrode movements. Nevertheless, most experimenters rely on a manual procedure for placing the electrodes based on visual recording quality assessment. An automated procedure would possibly find not only more suitable recording positions in less setup-up time, but also allow for tracking neurons in the case of tissue drifts.

- Modern external arrays consist of up to 64 multi-channel electrodes, whereas implantable arrays even contain up to 100 electrodes. This makes it inevitable that all the algorithms operate in an unsupervised manner, as a manual processing would not only become infeasible due to time constraints, but would achieve an inferior performance as well.

In this thesis we will present novel algorithms which can deal with the mentioned problems. We develop techniques inspired and derived from more general array processing theories which are adaptive to changing recording conditions and operate to a large amount in an unsupervised manner.

1.1 Problem formulation and its characteristic

One of the earliest processing stages of the recorded data consists of extracting the individual action potentials, also called spikes, from the continuously sampled data stream. This process is denoted as ”spike detection”, whereas, combined with a further classification of the spikes, i.e. the assignment of every spike to a specific neuron, the overall procedure is denoted as ”spike sorting”. In order to develop well founded detection/sorting algorithms, it must be assumed that the measured data $x_{k,t}$ (on channel k at time t) can be represented by some specific signal model. Throughout this thesis we assume the following model (or a simplified form of it):

$$x_{k,t} = \sum_{i=1}^{M} \sum_{\tau} q^i_{k,t} s^i_{t-\tau} + n_{k,t} \qquad k = 1, \ldots, N \tag{1.1}$$

where M is the number of neurons, N is the number of recording channels (e.g. $N = 4$ for tetrodes, see also the ”List of Symbols and Abbreviations” on page v for used notation), and $q^i_{k,t}$ is the spatio-temporal waveform of neuron i. It is not always clear

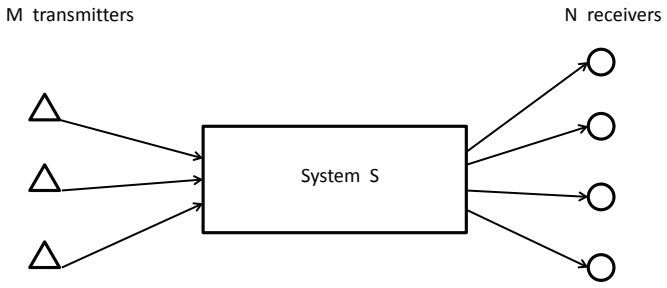

Figure 1.2: Sketch of a general system with M transmitters and N receivers, also called multiple-input multiple-output system (MIMO).

to distinguish between the signal s and noise n. Of course every physical system is subject to some thermal noise, but in the case of electrophysiological recordings this is not the main source of concern. In fact, what we call noise* is actually the neural activity from many "background" neurons. Then, we run into the problem of how to define a neuron which belongs to the signal or already to the background. Here, we omit this problematic by defining noise as all sources which follow a normal distribution, whereas signals as following a non-Gaussian statistics. In particular, we assume that the signals s^i follow a sparse Bernoulli distribution, the noise n is described by a zero mean, colored, multivariate Normal distribution, and the spike waveforms vary only slowly in time, i.e.

$$s^i \sim \text{Binomial}(1, p_i),\ p_i \ll 1\ \forall i \qquad \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}) \qquad \boldsymbol{q}^i[t] \approx \boldsymbol{q}^i. \tag{1.2}$$

The validity of these assumptions is not the topic of this thesis. It is sufficient to say that this model is widely accepted in the biosignal literature and has been validated by experiments, see e.g. [181, 168].

The goal of spike detection is to reconstruct the union of all the signals when only $x_{k,t}$ is observed, i.e.

$$\text{given } x_{k,t} \longrightarrow \cup_i s^i. \tag{1.3}$$

On the other hand, the task of spike sorting consists of reconstructing all the signals individually when only x is observed, i.e.

$$\text{given } x_{k,t} \longrightarrow s^i\ \forall i, \quad \text{up to permutation and scale.} \tag{1.4}$$

1.2 Relation to other fields

It is essential to notice that similar problems are also studied by other than the neuroscience community. We want to point out the similarities and differences between these different fields, as the transfer of methods developed in one field to a problem in another one often brings new insights. From a formal point of view, the M neurons are just a system of M transmitters, whereas a multi-channel electrode is a system of N receivers. The goal common to all fields is to infer the signal properties of the M transmitters given the

*The decision criteria what is signal and what is noise being our ability of visual discrimination between a spike waveform and the "rest".

measurements recorded by the N receivers, see Fig. 1.2. For example, such a problem is encountered in radar, sonar, seismic exploration [28], digital communication, wireless communication, image restoration [189], speech separation and enhancement [46], vibration analysis for fault detection [116], or mechanical signature analysis [169]. In the following subsections we explain some of them in more detail.

1.2.1 Radar and sonar

In radar or sonar systems, a similar problem as in electrophysiological recordings arises. Namely, M targets (such as planes, sub-marines or the like) are emitting[†] a particular signal signature which is recorded by an N-dimensional antenna array. The goal is to determine whether there are targets in the area, and if so, how many of them are present, and what their positions are. Up until very recently, the noise statistics (also called clutter statistics in this community) was assumed to be Gaussian[‡], and a time constant waveform (called steering vector) was assumed. On the other hand, the signal s is assumed to be continuous, and might be even normally distributed. One major difference is that most literature in this field deals with a parametrised steering vector, in the form of

$$\boldsymbol{q}^i(\theta_i) = \Big(1, \quad \exp(-j\Theta_i), \quad \ldots, \quad \exp(-j(N-1)\Theta_i)\Big)^\top \tag{1.5}$$

where $\Theta_i := (2\pi \cdot d \cdot \sin(\theta_i))/\lambda$, $j := \sqrt{-1}$, d being the distance between the sensors of the antenna array, and λ being the wavelength of the source signals [104]. This particular structure comes from the underlying physics, as the signal can be approximated by planar waves due to the open space and its geometry. Hence, in the far-field approximation a target can be represented by a single parameter, the direction-of-arrival θ_i. This is a significant reduction of complexity as the dimension of the steering vector is reduced from N to one.

Similar as for the spike detection and spike sorting problem, there is a large amount of literature concerned with just target detection and separately with multiple target classification, as well as with system calibration or blind beamforming where some of the blind source separation algorithms originate from. A good introduction into this field is given in [195, 210, 138]. In general, the most common approach to tackling these problems is the use of linear filters, called beamformers in this field. This class of filters will be discussed in more detail in Chap. 2.

1.2.2 Communications

In the community of communications (e.g. wireless communications, digital communications, etc.) the terminology SISO, MISO, and MIMO is used, denoting the configuration of the overall system, i.e. single-input single-output, multiple-input single-output, and multiple-input multiple-output. In contrast to the previously discussed field of radar, the waveform represents the channel fading rather than a target to be detected. Hence, the task is to reconstruct the original continuous signal s from a distorted measurement

[†] Active emission of signals or passive emission, i.e. reflections of a radar waves.

[‡] For more recent development, such as robust processing dealing with heavy tailed noise distributions see e.g. [113].

x. The amplitude of the "steering vector" most often follows a specific model, such as Rayleigh or Rician fading channels, and few approaches deal with general steering vectors, i.e. the channel is distributed according to a complex normal distribution

$$\boldsymbol{q}^i \sim \mathcal{CN}(\boldsymbol{a}, \boldsymbol{D}) \quad \begin{cases} \boldsymbol{a} = \boldsymbol{0} & \longrightarrow \text{Rayleigh} \\ \boldsymbol{a} \neq \boldsymbol{0} & \longrightarrow \text{Rician.} \end{cases} \tag{1.6}$$

In communications the task is seldom related to signal detection, as pilot signals are emitted for this purpose. The equivalent problem to spike sorting, i.e. multiuser interference elimination, was especially of concern in mobile communications regarding the code-division multiple access (CDMA) technique. However, since the recent introduction of a new technology, namely the orthogonal frequency-division multiple access (OFDMA), this problem seems to be have been solved. Introductions into this field can be found in [67, 186, 213].

The fields of radar and sonar and of communications can more generally be summarised under the term of spatio-temporal array processing techniques [128].

1.2.3 Blind source separation (BSS) and blind deconvolution

Perhaps the most fundamental research how to separate a mixture of M signals given N dimensional measurements was done in the field of blind source separation, primarily known from the various independent component analysis (ICA) algorithms. In the often illustrated application of separating audio sources[§], the problem is to separate M sound sources recorded with N microphones in a reverberation free environment, whereas it can be assumed that $M < N$ and the sources are pairwise statistically independent. Hence, it is assumed that the mixture is instantaneous, in the sense that only a spatial waveform over the different microphones enables a discrimination of the source, but there is no temporal structure. In contrast, the field of blind deconvolution or blind equalisation has in principle dealt with a SISO system, i.e. not spatial but only temporal correlations. Although the problem and the methods for its solving are quite similar in both fields, there seems to be only a limited literature pointing out the exact differences and similarities between the two fields [5, 6].

Later on, the field of ICA extended to cases where there is temporal correlation as well (convolutive ICA), or where there are more sources than sensors, $M > N$, (overcomplete/under determined ICA). One should note, however, that the convolutive case was solved most often by applying a Fourier transform, which leads to frequency ambiguities [160].

1.2.4 Terminology of the spike detection/sorting problem

Given the terminology of the different fields, the spike detection and the spike sorting problem, given the electrophysiological electrode recordings, can be described as a

[§]The popular term "cocktail problem" is ambiguous, as many regard it as separating just one source from the rest. Also note, that although blind source separation is often demonstrated on audio data, it originated from a problem in neuroscience [90, 106].

MISO (single electrode) or MIMO (multi-channel electrode), over-complete[¶], blind, and convolutive system. Although in theory the spike waveforms have a real physical interpretation in the sense of an electric current, and thus, could be parametrised by some generating quantities, in practice this is of no help due to the unknown neuron configuration. Hence, the waveforms cannot be reduced in their dimensionality and must be regarded as arbitrary, in contrast to the steering vectors in radar applications. On the other hand, the intrinsic signals s^i display an interesting property which makes them belong to a specific category. Namely, they are sparse, i.e. the waveforms occur only at few times in contrast to a continuous noise source, and binary (as a special class of discrete signals), i.e. only the two values 0 or 1 can be attained. This is in contrast to the continuous signals encountered in most of the communications technology.

1.3 Thesis summary

The goal of this thesis is to make explicit use of this particular system structure in order to develop powerful spike detection and spike sorting methods. The focus is on using ideas from the field of array processing and BSS for the proposed approaches. This is in noticeable contrast to most of the existing approaches to spike detection and spike sorting, which mainly rely on hypothesis testing, heuristics, and clustering. In particular, the spike sorting task is most often tackled by converting the data series into short vectors in which spikes are detected by hypothesis testing. Then, a feature extraction algorithm is used for further dimension reduction, and finally a clustering procedure is applied.

The literature in the field of array processing is very extensive and it is likely that in the future more and more BSS techniques will be applied or directly designed for the spike detection/sorting problem. In App. B.2 we point to some BSS literature which might be helpful for developing future algorithms handling the spike detection/sorting task. Existing approaches, which are based on similar underlying assumptions, are discussed in the corresponding chapters.

Next, we shortly summarise the results of subsequent chapters and point to published work based on parts of this thesis.

1.3.1 Thesis outline

In the remainder of Chap.1 a very short introduction to mathematical concepts which will be used later on is given, and the scientific contribution of this work is stated. The original research is presented in Chap. 2 to Chap. 6. Each chapter is self contained and includes an introduction to the discussed problem, a review of existing works, and a discussion. Due to this self contained character, some parts, especially concerning notation, problem formulation and introductory explanations, might slightly overlap between the individual chapters.

In Chap. 2 we start with the discussion of some aspects of linear filters. Linear filters will be used throughout the thesis as the primary tool for detecting and classifying

[¶]Except in the case when single, very high impedance electrodes are used, it is unrealistic to assume that activity from a number of neurons is recorded which is less than the number of channels.

spikes. In particular, we focus on the matched filter as it can be analytically expressed and achieves the best detection performance in most of the considered settings. We introduce a novel modification to the matched filter which improves the performance in other scenarios, such as real-time detection, and propose a measure how to assign a performance to detectors in the case of simultaneous detection and arrival times estimation.

In Chap. 3 we continue analysing the performance of filters when there is a mismatch between the true waveform of a neuron, and the waveform used for constructing the filter. We show that the results obtained from studying such a signal processing problem can be linked to findings in the econometrics theory. Based on this analysis we propose a filter adaptation scheme in the case when the neuron's waveform varies slowly over time.

In Chap. 4 an algorithm for spike sorting is proposed. Using standard spike detection and clustering techniques, initial waveforms are estimated and the corresponding filters calculated. The filter output is then processed by a technique called Deconfusion, similar to an un-mixing routine, leading to an improved classification performance. As a unique feature, our approach is suitable for online data processing, but is still capable of resolving overlapping spikes.

In Chap. 5 we address the problem of spike detection for which purpose an unsupervised and adaptive algorithm is formulated. The proposed algorithm is one of the very few approaches which uses techniques from BSS and blind deconvolution for spike detection. It offers superior detection performance, even when multiple neurons with distinct waveforms are present in the data, and adapts to changing waveforms.

In the last chapter, Chap. 6, an algorithm for unsupervised electrode placement is presented. Firstly, a quality measure is defined, which yields a higher score the better the signal-to-noise ratio and the separability of the neurons signals is. Then, the maximum of this quality measure is found by a stochastic optimization scheme, and the electrodes are moved to the corresponding position. To our knowledge, this is the first unsupervised positioning algorithm developed for multi-channel electrodes.

1.3.2 Contribution

- The work presented in Chap. 2 was partly published in

  ”Optimal convolutive filters for real-time detection and arrival time estimation of transient signals”, M. Natora, F. Franke, and K. Obermayer, *Proceedings of World Academy of Science, Engineering and Technology, Volume 55, pages 235-240, 2009*

- The work presented in Chap. 3 was partly published in

  ”Optimal steering vector adaptation for linear filters leading to robust beamforming”, M. Natora, F. Franke, S.A. Broda, and K. Obermayer, *Proceedings of the International Symposium on Communications, Control and Signal Processing, 2010*

- The work presented in Chap. 4 was partly published in

  ”Blind source separation of sparse overcomplete mixtures and application to neural recordings”, M. Natora, F. Franke, M. Munk, K. Obermayer, *Lecture Notes*

in Computer Science - Independent Component Analysis and Signal Separation, Volume 5441, pages 459-466, 2009,

and in

"An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes", F. Franke, M. Natora, C. Boucsein, M. Munk, K. Obermayer, *Journal of Computational Neuroscience, Volume 29, pages 127-148, 2010*

- The work presented in Chap. 5 was partly published in

  "Spike detection in extracellular recordings by hybrid blind beamforming", M. Natora, F. Franke, K. Obermayer, *Proceedings of 32nd Annual International Conference of the IEEE EMBS, pages 4636-4641, 2010*

  and in

  "An unsupervised and drift-adaptive spike detection algorithm based on hybrid blind beamforming", M. Natora, K. Obermayer, *EURASIP Journal on Advances in Signal Processing, Volume 2011, Article ID 696741, 13 pages*

- The work presented in Chap. 6 was partly published in

  "An automated online positioning system and simulation environment for multi-electrodes in extracellular recordings", M. Natora, F. Franke, P. Meier, E. Hagen, K. H. Pettersen, H. Linden, G. T. Einevoll, K. Obermayer, *Proceedings of 32nd Annual International Conference of the IEEE EMBS, pages 593-597, 2010*

- Code of several algorithms is available online from http://user.cs.tu-berlin.de/~natora/ for download.

1.4 Fundamental concepts

In this section we briefly present some notions and mathematical concepts which will be used in some of the subsequent chapters.

1.4.1 Digital signal processing

The voltage is recorded with an electrophysiological electrode digitally, usually with a sampling frequency between 8 – 40kHz. Hence, in the following we always assume discrete time series, i.e. $x(t) = x_t, t = 1, 2, \ldots$ unless otherwise stated.

A system S (as for example the one shown in Fig. 1.2) is called a LTI system ($x \rightarrow S \rightarrow y$), when it is linear and time-invariant [157]:

- time-invariant (stationary): If $x(t_1) \rightarrow S \rightarrow y(t_1)$ then $x(t-\tau) \rightarrow S \rightarrow y(t-\tau)\ \forall\tau$.
- linear: If $x_1(t) \rightarrow S \rightarrow y_1(t)$ and $x_2(t) \rightarrow S \rightarrow y_2(t)$, then $a_1 \cdot x_1(t) + a_2 \cdot x_2(t) \rightarrow S \rightarrow a_1 \cdot y_1(t) + a_2 \cdot y_2(t)$.
- stable: Additionally a system is called bounded-input bounded-output stable (BIBO stable) if $|x(t)| < \infty\ \forall t$, then $|y(t)| < \infty\ \forall t$ [190].

The impulse response h of a system S is the output when a Kronecker delta function is the input, i.e. $\delta(t) \rightarrow S \rightarrow h(t)$. The input output relation of stable LTI systems is described by its impulse response h by following relationship:

$$y_n = \sum_{m=-\infty}^{\infty} h_m x_{n-m} \tag{1.7}$$

whereas $||h|| < \infty$. The system is said to be causal if $h(t) = 0 \; \forall t < 0$ [105]. The z-transform of h is called the transfer function of the system S [157]. The transfer function $H(z)$ of a causal LTI system can be expressed as

$$H(z) = \frac{\sum_{j=0}^{J_b} b_j z^{-j}}{1 + \sum_{j=1}^{J_a} a_j z^{-j}}, \tag{1.8}$$

and hence the output y can be related to x by

$$y_n = \sum_{j=0}^{J_b} b_j x_{n-j} - \sum_{j=1}^{J_a} a_j y_{n-j}. \tag{1.9}$$

Depending on the choice of the coefficients a, b different filter classes are defined, in particular finite impulse response filters and infinite impulse response filters. The frequency response $H(\omega)$ of a filter is obtained by setting $z = e^{\sqrt{-1}\cdot\omega}$ in Eq. 1.8.

Finite duration Impulse Response (FIR) filters

The definition of FIR filters is given by $a_j = 0 \; \forall j$ in Eq. 1.8. This implies that the input-output relation is given by

$$y_n = \sum_{m=0}^{J_b} h_m x_{n-m}, \tag{1.10}$$

i.e. $h_j = b_j \; \forall j$, and the frequency response is simply $H(\omega) = \sum_{j=0}^{J_b} h_j \cdot e^{-\sqrt{-1}\cdot\omega}$. In this sense FIR filters perform a moving average operation. Some advantages of FIR filters compared to IIR (infinite (duration) impulse response) filters, which can perform an autoregressive moving average operation, are listed in [105]. Because FIR filters are applied in a non-recursive way, all FIR filters are stable, which implies that the filter output will always be finite, even if there is noise present in the system. On the other hand, IIR filters are applied in a recursive way which gives them more flexibility, but in general such filters are not stable.

1.4.2 Higher-order statistics

Higher-order statistics deals with properties of random variables which go beyond first and second order statistics such as mean and covariance. Here, we are mainly interested in cumulants and joint/cross-cumulants. The results summarised here can all be found in one of the introductory texts [139, 84, 199].

Let x_j, $j = 1, \ldots, N$, be N real random variables. The joint characteristic function is defined as

$$\phi(\boldsymbol{w}) := \mathbb{E}\left[e^{i\boldsymbol{w}^\top \cdot \boldsymbol{x}}\right]. \tag{1.11}$$

whereas $i := \sqrt{-1}$. Then, the joint cumulant is calculated by[‖]

$$\mathrm{Cum}(x_1 : n_1; x_2 : n_2; \ldots; x_N : n_N) = (-i)^m \frac{\partial^m \ln(\phi)}{\partial^{n_1} w_1 \ldots \partial^{n_N} w_N}_{\boldsymbol{w}=\boldsymbol{0}} \tag{1.12}$$

whereas, $m := \sum_j n_j$. If the joint characteristic function is not known, the cumulant can be directly computed from the joint moments. The general relationship is complicated, however, for the first couple of cumulants, following explicit relations are given:

$$\begin{aligned}
\mathrm{Cum}(x_1; x_2) &= \mathbb{E}[x_1 x_2] \\
\mathrm{Cum}(x_1; x_2; x_3) &= \mathbb{E}[x_1 x_2 x_3] \\
\mathrm{Cum}(x_1; x_2; x_3; x_4) &= \mathbb{E}[x_1 x_2 x_3 x_4] - \mathbb{E}[x_1 x_2]\,\mathbb{E}[x_3 x_4] \\
&\quad - \mathbb{E}[x_1 x_3]\,\mathbb{E}[x_2 x_4] - \mathbb{E}[x_1 x_4]\,\mathbb{E}[x_2 x_3]
\end{aligned} \tag{1.13}$$

whereas, if x_j is not zero mean, one has to replace x_j by $x_j - \mathbb{E}\left[x_j\right]\ \forall j$ on the right hand side of all equations in Eq. 1.13; and $\mathbb{E}[\cdot]$ denotes the expectation operator. From this it follows that

$$\begin{aligned}
\mathrm{Cum}(x) &= \mathbb{E}[x] \\
\mathrm{Cum}(x_1; x_2) &= \mathrm{Cov}(x_1; x_2) \\
\frac{\mathrm{Cum}(x : 3)}{\mathrm{Cum}(x : 2)^{3/2}} &= \mathrm{skewness}(x) \\
\frac{\mathrm{Cum}(x : 4)}{\mathrm{Cum}(x : 2)^2} &= \text{excess kurtosis}(x).
\end{aligned} \tag{1.14}$$

If x is normally distributed, i.e. $x \sim \mathcal{N}(\mu, \sigma)$, then it is $\mathrm{Cum}(x : 1) = \mu$, $\mathrm{Cum}(x : 2) = \sigma^2$, and $\mathrm{Cum}(x : m) = 0\ \forall m > 2$. The latter also holds for $\mathrm{Cum}(x_1 : n_1; \ldots; x_N : n_N)$ if the x_j are jointly Gaussian.

Some important properties of cumulants are stated in the following:

1. The joint cumulant of any permutation of x_j is again $\mathrm{Cum}(x_1 : n_1; \ldots; x_N : n_N)$.
2. For any constants c_j and $N > 1$, it is $\mathrm{Cum}(x_1 + c_1 : n_1; \ldots; x_N + c_N : n_N) = \mathrm{Cum}(x_1 : n_1; \ldots; x_N : n_N)$.
3. $\mathrm{Cum}(c_1 \cdot x_1 : n_1; \ldots; c_N \cdot x_N : n_N) = \prod_j c_j^{n_j} \cdot \mathrm{Cum}(x_1 : n_1; \ldots; x_N : n_N)$.
4. If the x_j are independent of y_j, then $\mathrm{Cum}(x_1 + y_1 : n_1; \ldots; x_N + y_N : n_N) = \mathrm{Cum}(x_1 : n_1; \ldots; x_N : n_N) + \mathrm{Cum}(y_1 : n_1; \ldots; y_N : n_N)$.
5. If any non-empty subset of the x_j is independent from the rest, then $\mathrm{Cum}(x_1 : n_1; \ldots; x_N : n_N) = 0$.

[‖]Alternatively, cumulants can be defined by the moment generating function instead.

1.5 Notation

In the literature it is most often assumed that both the signal s and the noise n are zero mean. In our setting, however, we cannot assume that both processes are zero mean, as the mean of s depends on the amount of spikes present. A zero mean signal implies that the calculation of filters (presented e.g. in Chap. 2 and Chap. 3) can be done on the basis of covariance matrices, i.e. for single channel data $\boldsymbol{R}_{t_1,t_2} = \mathbb{E}\left[x_{t_1} x_{t_2}\right] - \mathbb{E}\left[x\right]^2 = \mathrm{Cov}(x_{t_1}, x_{t_2})$. For non-zero mean signals, the correct filters are obtained by using matrices without mean subtraction, i.e. $\boldsymbol{R}_{t_1,t_2} = \mathbb{E}\left[x_{t_1} x_{t_2}\right]$. For simplicity, we still use the same notation $\mathbb{E}\left[x_{t_1} x_{t_2}\right] = \mathrm{Cov}(x_{t_1}, x_{t_2})$.

Usually, we use the notation of zero-centred vectors, i.e. a vector $\boldsymbol{y}$ is given by $\boldsymbol{y} = \left(y_{-L_y}, \ldots, y_{L_y}\right)^\top$, and thus the dimension is $T_y = 2L_y + 1$. The notation $\boldsymbol{y}_t$ refers to the vector entry at dimension t. This will be also denoted by $(\boldsymbol{y})_t$, $y(t)$ or simply y_t. For denoting a time varying vectorial quantity, the notation $\boldsymbol{y}[t]$ is used instead. Linear operators in the form of matrices are indexed in a similar way, i.e. the entry in the m-th row and n-th column is denoted as $\boldsymbol{D}_{m,n}$ or $(\boldsymbol{D})_{m,n}$, the indices being in the range $m, n = -L, \ldots, L$.

The discrete convolution between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted as $(\boldsymbol{x} * \boldsymbol{y})_t = \sum_\tau x_\tau y_{\tau - t}$. On the other hand, the discrete cross-correlation is defined by $(\boldsymbol{x} \star \boldsymbol{y})_t = \sum_\tau x_\tau y_{\tau + t}$. The length of the output vector is given by $T_x + T_y - 1$. In the case of multi-channel data, every channel is convoluted respectively cross-correlated individually, and the outputs are added up.

For simplicity, in most chapters we will deal with single channel data only, i.e. $N = 1$. This is only for notational convenience, and does not affect the generality of the proposed methods. In fact, the formalism in the case of multi-channel data is very similar to the single channel case, as the channels can be concatenated to a single vector again, see e.g. [222, 168].

Chapter 2

FIR filters and their performance

FIR filters have the advantage of being stable (see Sec. 1.4.1). The requirement of stability is of particular importance, since all data acquired by electrodes are noisy, and therefore we will analyse only FIR filters. Usually FIR filters are used for band-pass filtering which requires a design specified on the frequency response (which is $H(z)_{|z=e^{i\omega}}$ in Eq. 1.8) of the filter. In our case, however, we are rather interested in the filter response in the time domain. This is because of the specific signal model shown in Eq. 1.1, as every neuron exhibits a specific waveform to which the corresponding filter should respond. In this chapter we formulate a general optimisation problem for FIR filters based on the p-norm. As no universally optimal value for p seems to exist, we then focus on the case $p = 2$. We propose a new class of filters in this case, which is a generalisation of the existing ones. The issue how to evaluate the performance of a filter is discussed as well, in particular a novel measure for simultaneous detection and arrival time estimation is proposed.

2.1 p-norm filters

We assume that once the data are filtered with filter $\boldsymbol{f}^j$, a threshold γ^j is applied to the output $\boldsymbol{y}^i$, and presence of signal s^i is declared whenever $\boldsymbol{y}^j_t > \gamma^j$. Therefore, considering the spike sorting problem defined in Eq. 1.4 filter $\boldsymbol{f}^j$ should have a well defined output to its corresponding waveform $\boldsymbol{q}^j$, e.g. $\boldsymbol{f}^{j\top} \cdot \boldsymbol{q}^j = 1$, which then allows for an easy estimation of the source s^j, and a low response to the other waveforms $\boldsymbol{q}^i$ $i \neq j$ as well as to noise. Given the fact that $\mathrm{Var}\left(\boldsymbol{f}^j \star \boldsymbol{n}\right) = \boldsymbol{f}^{j\top} \cdot \boldsymbol{C} \cdot \boldsymbol{f}^j =: \sigma^2_{f^j}$ [†], these requirements can be formulated as the following optimisation problem:

$$\boldsymbol{f}^j = \underset{\boldsymbol{f}^j}{\operatorname{argmin}} \left\{ \left(\sum_{i=1}^{M} \beta_i^p \sum_{\tau} |\left(\boldsymbol{I}^{i,j}\right)_\tau|^p \right)^{1/p} + \alpha \boldsymbol{f}^{j\top} \boldsymbol{C} \boldsymbol{f}^j \right\} \quad \text{subject to } \boldsymbol{f}^{j\top} \cdot \boldsymbol{q}^j = 1 \tag{2.1}$$

[†]Using the definition of variance and the fact that $\boldsymbol{f} \star \boldsymbol{n}$ is zero mean (since we assumed $\boldsymbol{n}$ to be zero mean), one gets $\mathrm{Var}(\boldsymbol{f} \star \boldsymbol{n}) = \mathbb{E}\left[\sum_{t_1} f(t_1) n(t_1+\tau) \cdot \sum_{t_2} f(t_2) n(t_2+\tau)\right]$. By exchanging the expectation operator with the summation, and again using the fact that $\mathbb{E}[n(t)]^2 = 0$, one gets $\sum_{t_1,t_2} f(t_1) f(t_2) \mathbb{E}[n(t_1+\tau) n(t_2+\tau)] = \boldsymbol{f}^\top \cdot \boldsymbol{C} \cdot \boldsymbol{f}$.

where $\boldsymbol{l}^{i,j} := \boldsymbol{q}^i \star \boldsymbol{f}^j$, α controls the amount of template versus noise suppression, the β_i determines how much the i-th template should be suppressed, and p is an integer. The p-norm of a d-dimensional vector $\boldsymbol{x}$ is defined as

$$\|\boldsymbol{x}\|_p := \left(\sum_{i=1}^{d} |\,(\boldsymbol{x})_i\,|^p \right)^{1/p}, \tag{2.2}$$

which allows to re-write above minimisation problem more compactly as

$$\boldsymbol{f}^j = \underset{\boldsymbol{f}^j}{\operatorname{argmin}} \left\{ \left\| \hat{\boldsymbol{l}}^j \right\|_p + \alpha \cdot \sigma^2_{f^j} \right\} \qquad \text{subject to } \left(\boldsymbol{q}^j \star \boldsymbol{f}^j \right)_0 = 1 \tag{2.3}$$

where $\hat{\boldsymbol{l}}^j := \left(\beta_1 \boldsymbol{l}^{1,j}, \quad \ldots \quad, \beta_M \boldsymbol{l}^{M,j} \right)^\top$. In Eq. 2.3 all templates $\boldsymbol{q}^{i \neq j}$ as well as shifted versions of the corresponding template $\boldsymbol{q}^j$ are regarded as noise[‡], and the overall response to this "noise" vector $\hat{\boldsymbol{l}}^j$ is suppressed. If it is desired to individually suppress every template, the optimisation problem can be formulated as

$$\boldsymbol{f}^j = \underset{\boldsymbol{f}^j}{\operatorname{argmin}} \left\{ \sum_{i=1}^{M} \beta_i \left(\sum_{\tau} |\left(\boldsymbol{l}^{i,j} \right)_\tau|^p \right)^{1/p} + \alpha \boldsymbol{f}^{j\top} \boldsymbol{C} \boldsymbol{f}^j \right\} \quad \text{subject to } \boldsymbol{f}^{j\top} \cdot \boldsymbol{q}^j = 1 \tag{2.4}$$

or in a compact notation

$$\boldsymbol{f}^j = \underset{\boldsymbol{f}^j}{\operatorname{argmin}} \left\{ \sum_{i=1}^{M} \beta_i \left\| \boldsymbol{l}^{i,j} \right\|_p + \alpha \cdot \sigma^2_{f^j} \right\} \qquad \text{subject to } \left(\boldsymbol{q}^j \star \boldsymbol{f}^j \right)_0 = 1. \tag{2.5}$$

The two optimisation problem, Eq. 2.3 and Eq. 2.5, are related to each other by the triangle inequality (assuming that $\beta_i \geq 0 \; \forall i$)

$$\left\| \hat{\boldsymbol{l}} \right\|_p \leq \sum_i \left\| \beta_i \cdot \boldsymbol{l}^{i,j} \right\|_p = \sum_i \beta_i \left\| \boldsymbol{l}^{i,j} \right\|_p . \tag{2.6}$$

Also common to both optimisation problems is the fact that they are constrained convex minimisation problems [19]. In general, an explicit closed form solution for $\boldsymbol{f}^j$ is not obtainable, however powerful numerical methods exist. Explicitly, we use the `cvx` toolbox, see [72, 73], based on disciplined convex programming [74], to solve the mentioned optimisation problems numerically.

2.1.1 Single waveform

Let us consider the special case when only a single waveform is present in the data. Then, the filter optimisation problem is stated as

$$\boldsymbol{f}^j = \underset{\boldsymbol{f}^j}{\operatorname{argmin}} \left\{ \left\| \boldsymbol{M} \cdot \boldsymbol{l}^{j,j} \right\|_p + \alpha \boldsymbol{f}^{j\top} \boldsymbol{C} \boldsymbol{f}^j \right\} \qquad \text{subject to } \left(\boldsymbol{q}^j \star \boldsymbol{f}^j \right)_0 = 1, \tag{2.7}$$

where we introduced the diagonal suppression matrix $\boldsymbol{M}$ with $\boldsymbol{M}_{t,t} = 1$, if $\left(\boldsymbol{q}^j \star \boldsymbol{f}^j \right)_t$ should be suppressed, and $\boldsymbol{M}_{t,t} = 0$ otherwise[‖]. The formulation in Eq. 2.7 is still not

[‡] In order to exclude self-suppression one can set $\boldsymbol{l}^{j,j} = \boldsymbol{0}$.

[‖] Note that the matrix $\boldsymbol{M}$ is also defined symmetrically around zero, i.e. if $\left(\boldsymbol{q}^j \star \boldsymbol{f}^j \right)_t$ $t = -L, \ldots, L$, then also $\boldsymbol{M}_{t_1,t_2}$ $t_1 = -L, \ldots, L$, $t_2 = -L, \ldots, L$.

very practical, as it involves the α parameter. This makes it difficult to compare filters obtained from different p-norms, as one would have to calculate the filters for all α values and then choose the filter with the best performance. We alter the optimisation problem in the following way

$$\boldsymbol{f}^j = \underset{\boldsymbol{f}^j}{\operatorname{argmin}} \left\| \boldsymbol{M} \cdot \boldsymbol{I}^{j,j} \right\|_p \qquad \text{subject to} \begin{cases} \left(\boldsymbol{q}^j \star \boldsymbol{f}^j\right)_0 = 1 \\ \boldsymbol{f}^{j\top} \cdot \boldsymbol{C} \cdot \boldsymbol{f}^j \leq c \end{cases} . \tag{2.8}$$

In this formulation one seeks for a filter which has the optimal response to the waveform $\boldsymbol{q}^j$ under the constraint of an upper bound false alarm rate. Note that the optimisation problem is still convex, thus, can be solved with the same methods as described previously.

2.1.2 Performance criteria

The performance of a filter is assessed by its receiver operating characteristics (ROC) curves and the corresponding area under the curves (AUC) [140, 54]. Since, according to Eq. 2.8, all filters will have a bounded false alarm rate for noise, we therefore focus on the ability of the filters to suppress shifted versions of the waveform. If one allows a tolerance of $\pm\Delta$ samples in the arrival time estimation, the probability of detection is given by 1 minus the probability that the waveform is not detected within $\pm\Delta$, i.e.

$$P_\mathrm{D} = 1 - \prod_{\tau=-\Delta}^{\Delta} \mathrm{Prob}\left[\left(\boldsymbol{f}^j \star \boldsymbol{r}\right)_\tau < \gamma_j\right], \tag{2.9}$$

whereas $\boldsymbol{r}$ is a noisy data sample containing the waveform $\boldsymbol{q}^j$. Since we assumed Gaussian zero mean noise, this is expressed as

$$P_\mathrm{D} = 1 - \prod_{\tau=-\Delta}^{\Delta} \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\gamma_j - \left(\boldsymbol{f}^j \star \boldsymbol{q}^j\right)_\tau}{\sqrt{2 \cdot \sigma_{f^j}^2}}\right)\right), \tag{2.10}$$

where erf denotes the error function. Consequently, any detection of the waveform not within $\pm\Delta$ is classified as a false alarm (false positive detection) and the corresponding probability is given by

$$P_\mathrm{FA} = 1 - \prod_{\tau=-L}^{-\Delta+1} \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\gamma_j - \left(\boldsymbol{f}^j \star \boldsymbol{q}^j\right)_\tau}{\sqrt{2 \cdot \sigma_{f^j}^2}}\right)\right) \cdot \prod_{\tau=\Delta+1}^{L} \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\gamma_j - \left(\boldsymbol{f}^j \star \boldsymbol{q}^j\right)_\tau}{\sqrt{2 \cdot \sigma_{f^j}^2}}\right)\right), \tag{2.11}$$

where L denotes the length of $\boldsymbol{f}^j \star \boldsymbol{q}^j$. The AUC is then determined as the area under the curve in the P_FA-P_D plane by varying systematically the threshold γ_j.

2.1.3 Results

The two template shown in Fig. 2.1 were used as waveforms. Filters were calculated for all norms $p = 1, \ldots, 19$, and for the uniform/Chebyshev norm given by $p = \infty$. The noise

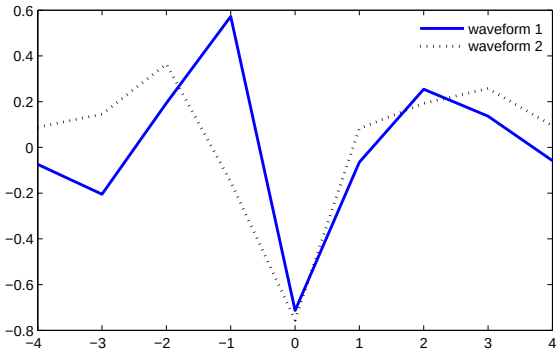

Figure 2.1: Waveform templates used in the filter optimisation problems.

covariance matrix was set to $\boldsymbol{C} = 0.5 \cdot \mathbf{1}$, and the noise constraint in Eq. 2.8 was set to $c = 0.5$ in the case of waveform 1, and $c = 0.75$ in the case of waveform 2. The threshold γ_j was varied systematically from -3 up to $+3$ in steps of 0.025. The optimisation was done with a suppression matrix $\boldsymbol{M}_{t,t} = 0$, $t = -\Delta, \ldots, +\Delta$, $\Delta = 0, 1, \ldots, 8$, and all filters had a length of 9 samples. The performance was evaluated by the criteria described in Sec. 2.1.2. In particular, the performance of a filter obtained from minimisation with a suppression matrix $\boldsymbol{M}_{t,t} = 0$, $t = -\Delta, \ldots, +\Delta$, was also evaluated with the same Δ for calculating the probability of detection and false alarm. The results in the case of waveform 1 are shown in Fig. 2.2, whereas the results in the case of waveform 2 are shown in Fig. 2.3.

From the two figures it is evident that no universal p exists. Rather the optimal norm to choose depends on the waveform shape, the noise constraint and the extent to which the waveform should be suppressed. The case when multiple waveform are present can be analysed in a very similar way. In fact, as the noise variance is already bounded in the optimisation problem Eq. 2.8, it is enough to add constraints on the suppression of non-corresponding templates in order to guarantee a sufficient discrimination performance. Explicitly, one would add the constraints $\left\|\boldsymbol{q}^i \star \boldsymbol{f}^j\right\|_\infty \leq c_i$, $\forall i \neq j$ in order to suppress the maximum false responses.

2.1.4 p-norm filters in literature

The design of filters based on convex optimisation criteria is not new. For example, in [107] convex optimisation is used to design robust beamformers. The focus, however, is only on the three cases $p = 1, 2, \infty$. A more general work, i.e. not restricted to antenna design requirements, is presented in [166], where convex optimisation is used for designing FIR filters which have a frequency response as close as possible to a given function (in the sense of the Chebyshev norm) under several constraints.

To our knowledge no work exists which systematically investigates the whole spectrum of p-norms in the *time domain*. In [184] optimisation again only involves $p = 1, 2, \infty$ with linear and quadratic constraints, which can be solved using second-order cone programming (this is a subclass of convex optimization). It is argued that

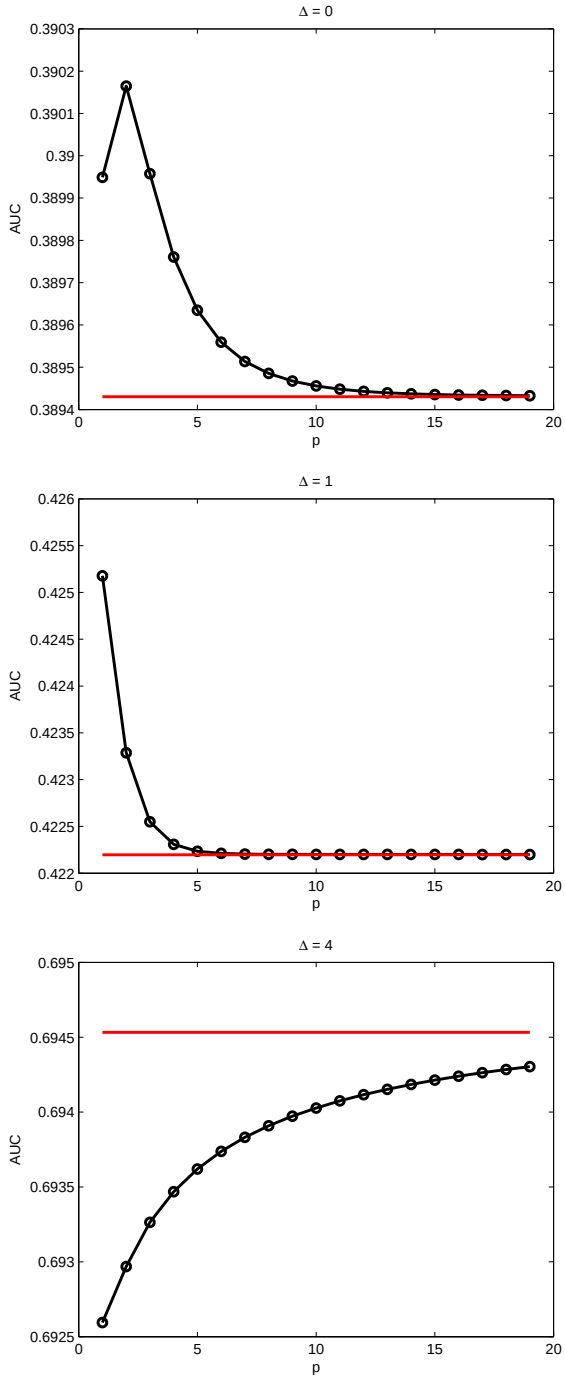

Figure 2.2: Area under the ROC curves for different p-norm filters and suppression matrices. The results were obtained in the case of waveform 1, see Fig. 2.1. The red line indicates the performance when $p = \infty$.

a general p-norm can be approximated by a combination of those three norms, which gives motivation to use these norms instead of a general p-norm. However, the problem formulation is done in the frequency domain of the FIR filter response.

In [118] an unconstrained minimisation problem in the frequency domain of a FIR filter is formulated. A large, even p-norm is used in order to approximate the Cheby-

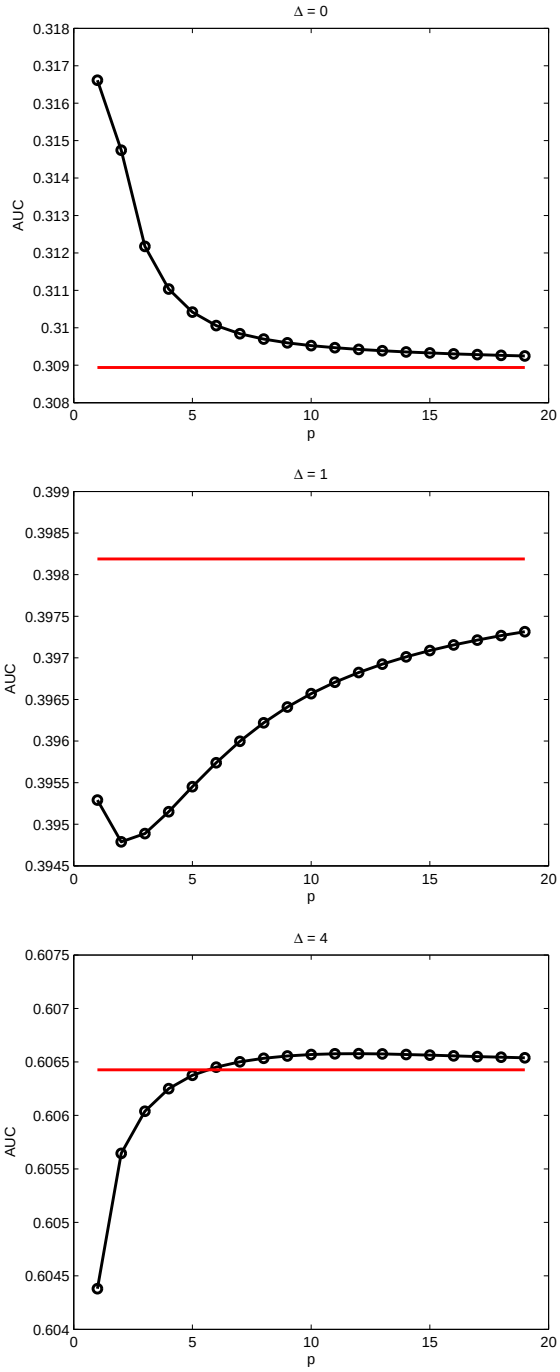

Figure 2.3: Area under the ROC curves for different p-norm filters and suppression matrices. The results were obtained in the case of waveform 2, see Fig. 2.1. The red line indicates the performance when $p = \infty$.

shev norm. The advantage of using an even p and not directly $p = \infty$ is that the cost function stays differentiable, so a gradient based approach can be used for solving the optimization problem.

In [192] also an unconstrained minimisation problem involving the frequency response of a FIR filter is formulated, but any p-norm is allowed.

In [151] a filter design framework is introduced, which allows to design FIR filters

in the frequency domain as well as time domain based on the Chebyshev norm. In particular, envelope-constrained filters can be obtained.

As a matter of fact, envelope-constrained filters, as introduced in [53], might be a better design choice than the time domain p-norm filters introduced in Sec. 2.1.1 when a specific time domain filter response is desired. The corresponding optimisation problem is given by

$$\boldsymbol{f}^j = \underset{\boldsymbol{f}^j}{\operatorname{argmin}}\, \boldsymbol{f}^{j\top} \boldsymbol{C} \boldsymbol{f}^j \qquad \text{subject to} \quad g(\tau) \leq \left(\boldsymbol{q}^j \star \boldsymbol{f}^j\right)_\tau \leq h(\tau) \quad \forall \tau, \tag{2.12}$$

where $g(\tau)$, $h(\tau)$ are arbitrary functions. As it is shown in [53], this is again a convex optimisation problem with linear constraints. Adaptive algorithms for this class of filters exist as well [221].

Matched filter ($p = 2$)

The case of $p = 2$ received particular attention in the literature due to two main reasons. Firstly, the optimisation problem given by Eq. 2.7 has an analytic solution, and, secondly, the resulting filter, most often called "matched filter", is optimal in the sense that it is the best possible linear transformation for *detecting* signal presence [98]. In the following section, thus, we focus on the case $p = 2$, but consider not sole detection, but simultaneous detection and arrival time estimation.

2.2 Convolutive filters for detection and arrival time estimation

For detection of signals in single data samples corrupted by Gaussian noise, linear filters, in particular the adaptive matched filter (AMF), have been proven to be powerful. Their performance is measured with respect to the probability of detection and of false alarm; see [94] for a performance analysis of the AMF and other filters. The AMF has been applied amongst others in radar and antenna systems [138]. In other applications, however, the incoming data stream does not consist of a few data samples, but of a continuous data stream, whereas the signal is present only in a few of the samples (transient signals). In this case, the signal must not only be detected, but also its arrival time must be estimated.

The research field of optimal simultaneous detection and estimation has been mainly initiated by the work presented in [141]. Based on this theory some detectors were developed [10, 58, 150], but most of these approaches rely on order statistics. In the work of [58], however, the authors mention, that especially in the case of long waveforms, linear convolutive filters* prove to be superior to order statistics. Moreover, linear convolutive filters are computationally much more efficient, and thus, more suitable for real-time applications than order statistics.

This raises the question of which detectors should be used for the mentioned task, and how their performance should be compared. In this chapter we focus in particular

*By convolutive filters we mean that the detection has to be made in continuous data, i.e. $\boldsymbol{f} \star \boldsymbol{x}$, and not just in single snapshots as in Chap. 3, i.e. $\boldsymbol{f}^\top \cdot \boldsymbol{x}$.

on the performance of linear filters, since they are easy to implement and are optimal in the class of linear transformations [210]. Although the performance of various detectors for transient signals was compared, see [61, 165, 226], these studies compared only the detection performance and linear convolutive filters were rarely used for comparison.

Linear convolutive filters, in the following abbreviated simply by the term linear filters, are a convenient approach for the task of simultaneous detection and arrival time estimation of transient signals, and, thanks to their computational efficiency, suitable for real-time applications. For example, they are used for extracting information from biomedical data [240, 204, 223], in speech processing (see [160] for a survey), in image restoration or in spatial beamforming; just to name a few fields of application.

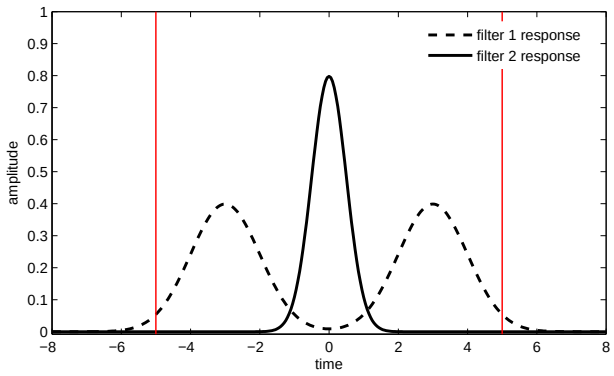

Figure 2.4: Illustration of the advantage of having a continuous performance measure. The black curves represent two possible filter responses to a particular waveform whose arrival time is at 0. Clearly, filter 2 predicts the arrival time more accurately than filter 1. However, just using the performance criterion presented Sec. 2.1.2 based on a tolerance zone (indicated by the red lines), both filters could achieve the same score.

However, to the knowledge of the author, no work exists to date which would propose a measure assigning a performance to detectors with respect to their ability of simultaneously detecting the presence as well as estimating the arrival time of transient signals. The method presented in Sec. 2.1.2 by introducing a tolerance zone is one possibility. There, any threshold-crossing inside this zone is regarded as signal detection and correct arrival time estimation, thus counted as a true positive detection. From a practical point of view, the use of a tolerance zone is reasonable. Depending on the desired accuracy, the user can chose an appropriate and task specific width of this zone. From a theoretic point of view, however, this approach is not fully satisfying. The filter response within the tolerance zone is not considered, and for more general problems, no well motivated zone width might be defined, which makes any particular choice of it artificial and arbitrary; see also Fig. 2.4. Instead, one would like to have a *continuous* measure indicating how well a filter performs in terms of simultaneous detection and arrival time estimation.

The remainder of this chapter is organised as follows: In Sec. 2.2.1 the general optimization problem in the case of $p = 2$ (see Sec. 2.1.1) is presented to which the linear filters are the solution. By modifying the optimisation criteria, a new class of linear filters is derived. In Sec. 2.2.1 a measure of performance of detectors with respect to simultaneous detection and arrival time estimation is presented. In Sec. 2.2.2 different

linear filters are compared with respect to this measure. The results from simulations in Sec. 2.2.2 agree with the theoretical findings and demonstrate the usefulness of these new filters and of the performance measure. The work is summarised and discussed in Sec. 2.2.4 and a brief outlook on further research directions is given.

2.2.1 Methods

Notation

For the representation of vectors, convolution or cross-correlation, we use the same notation as in Sec. 1.5. The symbol $\delta_y(x)$ denotes the usual Kronecker delta function, i.e. $\delta_y(x) = 1$, if $x = y$, and $\delta_y(x) = 0$ otherwise.

The notion of variance is slightly abused by attributing the variance to a probability density function (pdf) $f(x)$ rather than to a random variable X, i.e. for a discrete probability density

$$\mathrm{Var}_{f(x)} := \mathrm{Var}_{f(x)}(X) = \sum_x x^2 f(x) - \left(\sum_x x f(x) \right)^2 . \tag{2.13}$$

Linear convolutive filters

The measured data x_t is a continuously sampled data stream which is a linear mixture of a signal source and a noise source n_t. The signal is assumed to be sparse, i.e. consisting only of a short waveform $\boldsymbol{q}$ at specific times. Formally, the data generating process is written as *

$$x_t = \sum_\tau q_\tau s_{t-\tau} + n_t. \tag{2.14}$$

The point process s_t defines the times at which the waveform $\boldsymbol{q}$ is present, and can be modelled for example by a Bernoulli process. The noise n_t is assumed to be Gaussian, with zero mean and covariance matrix $\boldsymbol{C}$ (not necessarily white). It is assumed that the amplitude distribution of s_t as well as of $\boldsymbol{q}$ does not change in time, hence, only the presence of the waveform and its arrival time has to be detected, but not its amplitude scaling. Further, it is assumed that the signal waveform $\boldsymbol{q}$ and the noise covariance matrix $\boldsymbol{C}$ are known.

A perfect detector should retrieve the underlying point process s_t, as, in this case, all signals were detected and all arrival times estimated correctly. In the following, the focus will be on detectors in the class of linear filters which minimise the quadratic response to the data, combined with a pointwise thresholding of the filter output. This class of filters has the advantage of having an analytical expression, which allows for fast calculation (see Sec. 2.1 for other classes of linear filters). The optimisation problem for this kind of filters is stated as follows‖:

*For the sake of clarity and of simplicity, the analysis will be restricted to the case of single channel data. The entire method can be extended to multi-channel data in a straightforward manner.

‖Since the square root function is monotonic, we can drop it from the optimisation problem as compared to Eq. 2.7.

$$\boldsymbol{f} = \underset{\boldsymbol{f}}{\operatorname{argmin}} \left\{ |\boldsymbol{l}|^2 + \alpha \boldsymbol{f}^\top \boldsymbol{C} \boldsymbol{f} \right\} \qquad \text{subject to } \boldsymbol{f}^\top \cdot \boldsymbol{q} = 1 \tag{2.15}$$

where $\boldsymbol{l}$ is the filter response to the waveform $\boldsymbol{q}$, i.e. $\boldsymbol{l} := \boldsymbol{f} \star \boldsymbol{q}$. The optimisation criteria can be understood intuitively: The first term demands response of the filter to the waveform to be minimal, except for the correct arrival time, in which case the filter should respond with a well defined response of 1 (which is ensured by the optimisation constraint). The response of the filter to noise segments should be minimal as well. Since the noise was assumed to be Gaussian and zero mean, one has to minimise $\mathrm{Var}(\boldsymbol{f} \star \boldsymbol{n}) = \boldsymbol{f}^\top \boldsymbol{C} \boldsymbol{f}$ (see Sec. 2.1). The α parameter varies the ratio between minimisation of the filter response to the signal and to noise[‡].

The solution to the problem in Eq. 2.15 is given by

$$\boldsymbol{f} = \frac{\boldsymbol{H}^{-1}\boldsymbol{q}}{\boldsymbol{q}^\top \boldsymbol{H}^{-1}\boldsymbol{q}} \tag{2.16}$$

where the matrix $\boldsymbol{H}$ is given by $\boldsymbol{H} := \boldsymbol{\Xi} + \alpha \boldsymbol{C}$, and $(\boldsymbol{\Xi})_{k,l} := (\boldsymbol{q} \star \boldsymbol{q})_{k-l}$, see e.g. [222].

In the limit of $\alpha \to \infty$, the filter reduces to $\boldsymbol{f} = \boldsymbol{C}^{-1}\boldsymbol{q} / \left(\boldsymbol{q}^\top \boldsymbol{C}^{-1} \boldsymbol{q}\right)$, which is the minimum variance distortionless response (MVDR) beamformer, which is equivalent to the (adaptive) matched filter (AMF), see [177][†], also called Capone beamformer [210, 13]. This detector will be referred to as the "no suppression filter".

On the other hand, for a particular choice of α proportional to the occurrence frequency of the transient signal, the minimum power distortionless response (MPDR) beamformer is obtained [210]. Namely, in standard literature it is (using the convention of Sec. 1.5) $\mathrm{Cov}(\boldsymbol{x}) = \mathrm{Cov}(\boldsymbol{s} * \boldsymbol{q}) + \mathrm{Cov}(\boldsymbol{n}) = \left(\sigma_s^2 + \langle s \rangle^2\right) \cdot \boldsymbol{\Xi} + \boldsymbol{C}$, and comparing this to the definition of $\boldsymbol{H}$, this leads to $\alpha = 1/\left(\sigma_s^2 + \langle s \rangle^2\right)$, see e.g. [216], where $\langle \cdot \rangle$ denotes the mean. This detector will be referred to as the "full suppression filter".

The original optimisation problem in Eq. 2.15 will be generalised in two ways:

1) Variable suppression matrix: Instead of either full suppression of the signal or no suppression at all, one can demand to suppress only specific shifts t of the waveform. In this case $\boldsymbol{l}$ is replaced by $\boldsymbol{M} \cdot \boldsymbol{l}$, where the suppression matrix $\boldsymbol{M}$ is a diagonal matrix with $M_{t,t} = 1$ if the shift $(\boldsymbol{f} \star \boldsymbol{q})_t$ should be suppressed, and $M_{t,t} = 0$ otherwise.

2) Variable target function: In the original optimisation problem the response of the filter to the template had to be minimal, i.e. the least square distance to zero. Instead, one can minimise the distance to an arbitrary function $\boldsymbol{g}$, which is expressed by the substitution of $\boldsymbol{l}$ with $\boldsymbol{g} - \boldsymbol{l}$.

Combining both variations 1) and 2) this leads to a modified optimisation problem stated as

$$\boldsymbol{f} = \underset{\boldsymbol{f}}{\operatorname{argmin}} \left\{ |\boldsymbol{g} - \boldsymbol{M} \cdot \boldsymbol{l}|^2 + \alpha \boldsymbol{f}^\top \boldsymbol{C} \boldsymbol{f} \right\} \qquad \text{s.t. } \boldsymbol{f}^\top \cdot \boldsymbol{q} = 1. \tag{2.17}$$

The solution to this modified optimization problem can still be obtained analytically.

[‡]Note that a explicit weighting of the template suppression and noise suppression term is rarely used in standard literature, but was for example used in [222, 224].

[†]Note that in [177] the filters were obtained under the constraint $\boldsymbol{f}^\top \boldsymbol{C} \boldsymbol{f} = 1$ instead, however, in terms of detection performance the filters are equivalent. Also, we will still refer to this filter as the *adaptive* matched filter, even if the *exact* noise covariance matrix is known.

Proposition 2.1. *The solution to the optimisation problem stated in Eq. 2.17 is given by*

$$\boldsymbol{f} = \left(\boldsymbol{G}^{-1} - \frac{\boldsymbol{G}^{-1}\boldsymbol{q}\boldsymbol{q}^{\top}\boldsymbol{G}^{-1}}{\boldsymbol{q}^{\top}\boldsymbol{G}^{-1}\boldsymbol{q}}\right) \cdot (\boldsymbol{g} \star \boldsymbol{q})_{[-L_q, L_q]} + \frac{\boldsymbol{G}^{-1}\boldsymbol{q}}{\boldsymbol{q}^{\top}\boldsymbol{G}^{-1}\boldsymbol{q}} \tag{2.18}$$

where $\boldsymbol{G} := \tilde{\boldsymbol{\Xi}} + \alpha \boldsymbol{C}$*, and* $\tilde{\boldsymbol{\Xi}}_{k,l} := \sum_{\tau} (M_{\tau,\tau})^2 \, q_{k+\tau} q_{l+\tau}$*, and* $2L_q + 1$ *being the dimension of vector* $\boldsymbol{q}$ *(see Sec. 1.5).*

The proof is given in Sec. 2.2.5. If $\boldsymbol{g} = \boldsymbol{0}$ or $g_t = \delta_0(t)$ the first term in Eq. 2.18 disappears. Furthermore, if the suppression matrix $\boldsymbol{M}$ is the identity matrix, $\boldsymbol{M} = \boldsymbol{1}$, the original formula in Eq. 2.16 is obtained, whereas for $\boldsymbol{M}$ being the zero matrix, $\boldsymbol{M} = \boldsymbol{0}$, the no suppression filter is obtained; thus the filters in Eq. 2.18 constitute a generalisation of the existing filter design.

Performance measure

The processing flow of a detector consists of two consecutive steps: filtering, and an application of a threshold γ to the filter output. Hence, it is desired that after these two steps, the underlying point process s_t in Eq. 2.14 is obtained. If one achieves the correct estimation of this point process, the signal has been detected and the arrival times retrieved successfully.

Since a signal consisting of a unique waveform without amplitude variations was assumed, one can restrict itself to the analysis of detection and arrival time estimation of the waveform itself. Therefore, the output of a perfect detector D must always be $D(\boldsymbol{q} + \boldsymbol{n}) = \delta_0(t)$. As such, the perfect detector reconstructs the original point process s_t for *all* possible thresholds. Hence, one would like to have a measure which indicates the closeness of a detector output to the $\delta_0(t)$ function. In contrast, the classical performance measure, which is the probability of detection P_D (see e.g. [94]), only indicates whether the waveform was detected at all, but does not measure the closeness of the detection probability to the correct arrival time.

Based on these observations, the following measure of performance P_{DE} (for a fixed, but arbitrary threshold γ) for combined detection and arrival time estimation is proposed:

$$P_{DE} := \frac{\mathrm{Var}_{1/2(\bar{p}(t)+\delta_0(t))} - \mathrm{Var}_{1/2(p(t)+\delta_0(x))}}{\mathrm{Var}_{1/2(\bar{p}(t)+\delta_0(t))}} \tag{2.19}$$

where $\bar{p}(t)$ is a pdf for which $\mathrm{Var}_{1/2(p(t)+\delta_0(x))}$ is maximal, i.e.

$$\bar{p}(t) := \underset{p(t)}{\mathrm{argmax}} \left\{ \mathrm{Var}_{1/2(p(t)+\delta_0(t))} \right\}.$$

The function $p(t)$ is a detector dependent pdf which is at each point t in time proportional to the probability that the filter output is above the threshold γ, i.e.

$$p(t) := \frac{P_D(t)}{\sum_t P_D(t)} \tag{2.20}$$

where $P_D(t)$ is the classical probability of detection. In the case of linear filters, one has $P_D(t) = \mathrm{Prob}[(\boldsymbol{f} \star (\boldsymbol{q} + \boldsymbol{n}))_t \geq \gamma]$. The motivation for this definition of performance measure P_{DE} will be discussed in Sec. 2.2.3.

Two important properties of P_{DE} are stated in the following propositions.

Proposition 2.2. *In the case of a discrete pdf defined on the interval* $[-a, a]$*,* P_{DE} *is given by*

$$P_{DE} = 1 - \frac{2}{a^2} \operatorname{Var}_{1/2(p(t)+\delta_0(t))} . \tag{2.21}$$

The proof is given in Sec. 2.2.5. In contrast to Eq. 2.19, the expression in Eq. 2.21 no longer depends on the unknown quantity $\bar{p}(t)$, and thus, allows for calculation of the performance measure in real applications.

Proposition 2.3. P_{DE} *takes values in the interval* $[0, 1]$*. The maximum value of* 1 *is attained if and only if* $p(t) = \delta_0(t)$*.*

The proof is given in Sec. 2.2.5. This last proposition establishes bounds on the range in which the values of P_{DE} fall. A value close to 1 indicates a good performance, whereas a value close to 0 indicates a poor performance of the detector. Moreover, it states that only the perfect detector can achieve the best possible performance.

As in the calculation of the quantity $p(t)$ a normalisation is involved in order to obtain a pdf (see Eq. 2.20), even a single small value exceeding the threshold will be normalised to a pdf. If the threshold is increased towards infinity, the measure might indicate a better and better performance, although the real probability of detection will become arbitrarily small. Hence, in contrast to the classical measures, one has to restrict the range of possible thresholds. A reasonable choice is to set $\gamma_{\max} = \max_t \{(\boldsymbol{f} \star \boldsymbol{q})_t\}$, and $\gamma_{\min} = \mathbb{E}[\boldsymbol{f} \star \boldsymbol{n}]$. The upper threshold is justified by the fact that in the noise-free case, a threshold greater than the maximal value of the filter response to the waveform would lead to zero detections. The lower bound of the threshold is also justified, since a threshold below the average response to a noise segment would always lead to detection of the signal, except when the detector is meaningless.

2.2.2 Results

Numerical evaluation

The measure in Eq. 2.21 indicates the performance of a filter for *one* fixed (but arbitrary) threshold γ. In order to assign an *overall* performance to a detector, however, a total measure is needed. As such, slightly modified receiver operating characteristics (ROC) and the area under these ROC curves (AUC) were used. The x-axis of the ROC curve corresponded to the probability of false alarm P_{FA} [94], i.e. the probability that a data segment containing only noise will be incorrectly detected as signal. Instead of P_D, the y-axis corresponded to the proposed P_{DE} measure. According to the properties of P_{DE} in Sec. 2.2.1, a larger value of the AUC indicates a better performance of the corresponding filter.

In this evaluation setting, three different linear filters were compared, namely the no suppression filter, the full suppression filter and a particular case of the proposed filter class. The waveform of the signal had a length of $T_q = 7$, whereas the noise covariance matrix was set to $\boldsymbol{C} = 0.025 \cdot \mathbf{1}$, resulting in a SNR of 14.0 db.

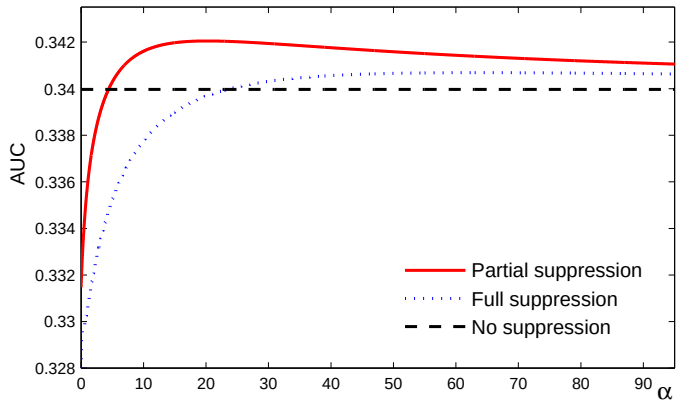

Figure 2.5: AUC of various filters for different α values. The partial suppression filter was calculated by Eq. 2.18 in which the target function was set to $g_t = \delta_0(t)$, $\forall t \in [-6, 6]$ and the diagonal matrix entries of $\boldsymbol{M}$ were set such that $M_{-1,-1} = M_{0,0} = M_{1,1} = 0$, and $M_{t,t} = 1\ \forall t \notin \{-1, 0, 1\}$. This proposed filter achieved the highest AUC score. The optimal performance for the partial suppression filter was achieved at $\alpha = 20.220$, for the full suppression filter at $\alpha = 63.685$, whereas, by construction, the no suppression filters had constant performance for all α values.

In the case of zero mean Gaussian noise the probability of detection is given by the expression

$$P_D(t) = 0.5 \cdot \left(1 - \mathrm{erf}\left(\frac{\gamma - l(t)}{\sqrt{2 \cdot \boldsymbol{f}^\top \boldsymbol{C} \boldsymbol{f}}}\right)\right),$$

where $\mathrm{erf}(x)$ denotes the standard error function, and $l(t) = (\boldsymbol{f} \star \boldsymbol{q})_t$. P_{FA} is obtained by $P_{FA} = P_D(l(t) = 0)$. P_{DE} was then calculated according to Eq. 2.21 with $a = 0.5 \cdot \left(T_{\boldsymbol{f} \star \boldsymbol{q}} - 1\right) = 6$.

For a linear filter, the average response to zero mean noise is zero, i.e. $\mathbb{E}[\boldsymbol{f} \star \boldsymbol{n}] = 0$. It turned out that for this particular evaluation setting one has $\max_t \{(\boldsymbol{f} \star \boldsymbol{q})_t\} = 1$ for all considered filters. Hence, the threshold γ was varied in the interval $[0, 1]$ (in steps of 0.002).

Recall, that the linear filters depended on the trade-off parameter α, see Eq. 2.16 and Eq. 2.18. The AUC was computed for all α values starting from $\alpha = 0$ in steps of 0.005 up to a value for which the performance started to converge to the performance of the no suppression filter; see Sec. 2.2.1 for explanation. The results are shown in Fig. 2.5.

Although the filters attain their best performance at different α values (see Fig. 2.5), the proposed filter, called partial suppression filter, achieved the highest AUC.

Simulations

The results from the previous section based on the proposed performance measure indicate that partial suppression filters are advantageous in comparison to the full and no suppression filters. To verify this result in a realistic setting, Monte Carlo simulations were performed. In particular, a single simulation consisted of a data stream containing 1000 signal segments and twice as many noise segments. The identical waveform and also the same noise statistics as the ones described in the previous section were used. The implementation was realised in MATLAB®.

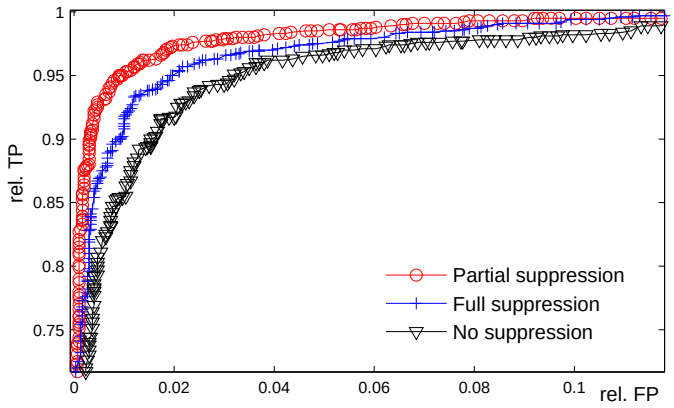

Figure 2.6: ROC curves for various filters based on a single simulation. The number of true positive detections was normalised by the total number of signal segments, whereas the number of false positive detections was normalised by the total number of data segments. Clearly, the proposed partial suppression filter outperforms the other filters.

For performance comparison the previously calculated filters were used, with the α parameter set at the specific values for which the respective filter achieved best performance (see Fig. 2.5).

As scope the area of real-time applications was chosen. In such a setting, at time t_0 only data x_t from precedent times $t \leq t_0$ are available. Nevertheless, the decision about signal presence has to be made already at time t_0. Consequently, every threshold crossing is immediately accounted for a signal presence, and every detection, which does not correspond to the exact signal arrival time, is counted as a false positive detection (FP). Accordingly, only successful detections at the exact arrival time of a signal are counted as true positive detections (TP). By varying the threshold (in steps of 0.0025) the corresponding ROC curves were obtained, see Fig. 2.6.

For the assessment of the overall performance, the AUC was computed and considered only up to the smallest (common for all filters) relative FP value for which rel. TP = 1, in order to avoid redundant computations. The AUCs of all filters averaged over 10 independent simulations are shown in Tab. 2.1, and the variance across the simulations was of the order of 10^{-7}.

The partial suppression filter achieved the best score, followed by the full suppression filter and lastly the no suppression filter. This is the same ranking as predicted in Sec. 2.2.2 by P_{DE}. In contrast, the classical performance measure P_D/P_{FA} would not have predicted the correct ranking: $P_D = \text{Prob}[\boldsymbol{f}^\top \cdot (\boldsymbol{q} + \boldsymbol{n}) \geq \gamma]$ is largest for the no suppression filter and smallest for the full suppression filter (and vice-versa for P_{FA}).

Table 2.1: Average AUC for various filters.

	Partial	Full	No
av. AUC	0.396487	0.395009	0.392696

2.2.3 Discussion

The trick was to re-normalise the detector output to a pdf, so that it then can be compared to the output of the perfect detector for all thresholds. The performance measure we chose for comparing the pdfs was motivated by $\mathrm{Var}_{1/2(p_1(x)+p_2(x))} - \mathrm{Var}_{p_2(x)}$, which indicates the difference in variances of the summed pdf and the desired pdf. Since, however, $p_2(x) = \delta_0(x)$ in our case, the second term vanishes. Note that $\mathrm{Var}_{p_1(x)} - \mathrm{Var}_{p_2(x)}$ would not be good definition, since $p_1(x)$ and $p_2(x)$ might have very similar variances, but their peaks might be very far apart, thus predicting the arrival time at very distinct time points.

Of course, there are already many well established measures to compare two probability distributions. In the following, we shortly discuss why most of them are not well suited for the problem we are interested in.

The symmetrised Kullback-Leibler and the Jensen-Shannon divergence, for example, can produce infinite values, as divisions by zero might occur. In our filter setting this might indeed happen when we assume a noise-free setting.

The Hellinger distance in the case of one distribution being a Kronecker delta reduces to

$$H^2 = 1/2 \sum_x \left(\sqrt{\delta_0(x)} - \sqrt{p_1(x)} \right)^2 = 1/2 \sum_x \left(p_1(x) + \delta_0(x) - 2\sqrt{\delta_0(x)}\sqrt{p_1(x)} \right)$$
$$= 1/2 \left(1 + 1 - 2\sqrt{p_1(0)} \right) = 1 - \sqrt{p_1(0)}.$$

Hence, only a single value, namely $p_1(0)$ is considered, which spoils the whole idea of comparing probability density functions. The same is true for the Bhattacharyya distance.

On the other hand, the Bregman divergence is very general, and it is not clear which specific convex function should be used in its definition.

Also, the used measures in communications, such as the maximum distortion or intersymbol-interference (see e.g. [189]), do not consider the shape of the distribution of the filter response, but only the difference between the largest filter output value and its energy.

One promising distance measure might be the Vasershtein metric, also called earth mover's distance. Recently, this metric has indeed been applied to measure the performance of filters [185], and it should be investigated if this could be applied for the detection/estimation problem as well.

2.2.4 Conclusion

To sum up, a measure was proposed which assigns a performance to a detector with respect to simultaneous detection and arrival time estimation of transient signals. Although the proposed measure is general and suitable for most detectors, the detector class of linear filters is of particular interest. We started to analyse filters as the solution of a general optimisation problem involving an arbitrary p-norm, but then focused on the case $p = 2$ which allows for an analytic solution. In this popular sub-class of minimal quadratic response filters, the existing filters were modified by introducing a suppression

matrix and a target function. The proposed filters have the advantage of still being analytically computable, but offer more flexibility than the existing filters. The widely used minimum variance distortionless response beamformer, the Capone filter and the minimum power distortionless response beamformer are all particular realisations within the proposed filter class.

In fact, the target function introduced in our filter class can be used for adjusting the smoothness of the filter response. This might be helpful in cases when the post processing consists not just of a pointwise thresholding, but of a more complex operation; e.g. when the data contains more than one signal source and a simultaneous detection and classification task has to be performed.

On the other hand, the suppression matrix allows for the selective suppression of specific filter responses. This can be useful for incorporating prior knowledge about the signal into the filter design, as for example a refractory period or dead time.

Using the proposed measure, two existing filters (AMF/MVDR and MPDR) were compared with a particular filter of the just proposed filter class. The measure indicated a favourable performance for the proposed filter, which was confirmed in simulations. In particular, the proposed filter was superior in a real-time detection and arrival time estimation task. This shows that the defined performance measure as well as the proposed filters are useful and advantageous.

In the performed evaluation the target function and the suppression matrix were set manually. As an outlook for further investigations one might think of an online adaption scheme: The filtering is started with the classical adaptive matched filter, while in the background an optimisation problem is solved, which aims at finding an optimal target function and suppression matrix. Once such a solution is found, the filter is adapted accordingly.

2.2.5 Proofs

Proof of proposition 2.1

The objective function of the optimisation problem in Eq. 2.17 is convex and since the optimisation constraint is linear, one can use the Lagrange multiplier method for solving it. The corresponding Lagrangian L is given by

$$L = |\boldsymbol{g} - \boldsymbol{M} \cdot \boldsymbol{l}|^2 + \alpha \boldsymbol{f}^\top \boldsymbol{C} \boldsymbol{f} + \lambda \left(\boldsymbol{f}^\top \boldsymbol{q} - 1 \right)$$

where λ is the Lagrange multiplier. The derivatives with respect to $\boldsymbol{f}$ and λ can be calculated as

$$\frac{\delta L}{\delta f_{t_i}} = 2 \left(\sum_\tau M_{\tau,\tau} q_{t_i+\tau} \sum_t M_{\tau,\tau} q_{t+\tau} f_t - g_\tau M_{\tau,\tau} q_{t_i+\tau} \right) + 2\alpha \sum_t \boldsymbol{C}_{t_i,t} f_t + \lambda q_{t_i}$$

$$\frac{\delta L}{\delta \lambda} = \left(\sum_t q_t f_t \right) - 1$$

The calculation of the second derivatives leads to

$$\frac{\delta L}{\delta f_{t_i} \delta f_{t_I}} = 2 \sum_{\tau} M^2_{\tau,\tau} q_{t_i+\tau} q_{t_I+\tau} + 2 \cdot \alpha \cdot \boldsymbol{C}_{t_i,t_I}$$

$$\frac{\delta L}{\delta \lambda \delta \lambda} = 0 \qquad\qquad \frac{\delta L}{\delta f_{t_i} \delta \lambda} = q_{t_i}$$

The second derivatives of L are independent of $\boldsymbol{f}$ and of λ. Therefore, the Taylor expansion of the first derivative of L around zero consists only of two terms and the solution can be obtained by solving

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\delta L}{\delta f_{t_i}} \\ \frac{\delta L}{\delta \lambda} \end{pmatrix} = \begin{pmatrix} \frac{\delta L}{\delta f_{t_i}} \\ \frac{\delta L}{\delta \lambda} \end{pmatrix}_{f_{t_i}=0,\lambda=0} + \begin{pmatrix} \sum_{t_I} \frac{\delta L}{\delta f_{t_i} \delta f_{t_I}}_{f_{t_i}=0,\lambda=0} \cdot f_{t_I} \\ \sum_{t_I} \frac{\delta L}{\delta \lambda \delta f_{t_I}}_{f_{t_i}=0,\lambda=0} \cdot f_{t_I} \end{pmatrix} + \begin{pmatrix} \frac{\delta L}{\delta f_{t_i} \delta \lambda}_{f_{t_i}=0,\lambda=0} \cdot \lambda \\ \frac{\delta L}{\delta \lambda \delta \lambda}_{f_{t_i}=0,\lambda=0} \cdot \lambda \end{pmatrix}.$$

In matrix notation, the above equation becomes

$$\boldsymbol{0} = \begin{pmatrix} -2\,(\boldsymbol{g} \star \boldsymbol{q})_{[-L_q,L_q]} \\ -1 \end{pmatrix} + \tilde{\boldsymbol{H}} \cdot \begin{pmatrix} \boldsymbol{f} \\ \lambda \end{pmatrix} \tag{2.22}$$

where one defined

$$(\boldsymbol{g} \star \boldsymbol{q})_{[-L_q,L_q]} := \left((\boldsymbol{g} \star \boldsymbol{q})_{-L_q}, \ldots, (\boldsymbol{g} \star \boldsymbol{q})_{L_q} \right)^\top$$

and

$$\tilde{\boldsymbol{H}} := \begin{pmatrix} 2\left(\tilde{\boldsymbol{\Xi}} + \alpha \cdot \boldsymbol{C}\right), & \boldsymbol{q} \\ \boldsymbol{q}^\top, & 0 \end{pmatrix}$$

and $\left(\tilde{\boldsymbol{\Xi}}\right)_{k,l} := \sum_\tau m^2_{\tau,\tau} q_{k+\tau} q_{l+\tau}$.

Define $\boldsymbol{G} := \tilde{\boldsymbol{\Xi}} + \alpha \boldsymbol{C}$. The inverse of $\tilde{\boldsymbol{H}}$ is then given by [122]

$$\tilde{\boldsymbol{H}}^{-1} = \begin{pmatrix} \frac{1}{2}\left(\boldsymbol{G}^{-1} - \frac{\boldsymbol{G}^{-1}\boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{G}^{-1}}{\boldsymbol{q}^\top\boldsymbol{G}^{-1}\boldsymbol{q}}\right) & \frac{\boldsymbol{G}^{-1}\boldsymbol{q}}{\boldsymbol{q}^\top\boldsymbol{G}^{-1}\boldsymbol{q}} \\ \frac{\boldsymbol{q}^\top\boldsymbol{G}^{-1}}{\boldsymbol{q}^\top\boldsymbol{G}^{-1}\boldsymbol{q}} & -\frac{2}{\boldsymbol{q}^\top\boldsymbol{G}^{-1}\boldsymbol{q}} \end{pmatrix}$$

The left multiplication of Eq. 2.22 with $\tilde{\boldsymbol{H}}^{-1}$ yields the solution for $\boldsymbol{f}$, which is given by

$$\boldsymbol{f} = \left(\boldsymbol{G}^{-1} - \frac{\boldsymbol{G}^{-1}\boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{G}^{-1}}{\boldsymbol{q}^\top\boldsymbol{G}^{-1}\boldsymbol{q}}\right)(\boldsymbol{g} \star \boldsymbol{q})_{[-L_q,L_q]} + \frac{\boldsymbol{G}^{-1}\boldsymbol{q}}{\boldsymbol{q}^\top\boldsymbol{G}^{-1}\boldsymbol{q}}$$

□

Proof of proposition 2.2

If one can show that $\mathrm{Var}_{1/2(\bar{p}(x)+\delta_0(x))} = a^2/2$, the proposition simply follows from Eq. 2.19. It is

$$\mathrm{Var}_{1/2(p(x)+\delta_0(x))} = \mathrm{Var}_{1/2p(x)} + \mathrm{Var}_{1/2\delta_0(x)} = \mathrm{Var}_{1/2p(x)}\,.$$

Strictly speaking $1/2p(x)$ is not a pdf, but $\mathrm{Var}_{1/2p(x)}$ is still defined as in Eq. 2.13.

It is

$$\mathrm{Var}_{1/2p(x)} = \sum_x x^2 \cdot 1/2p(x) - \left(\sum_x x \cdot 1/2p(x)\right)^2 \leq 1/2 \sum_x x^2 p(x) = 1/2\,\mathrm{Var}_{q(x)},$$

where q is a pdf of a discrete random variable with zero mean. The variance of *any* pdf $q(x)$ on the interval $[-a, a]$ is bounded by a^2 [3]. Hence, $\mathrm{Var}_{1/2p(x)} \leq a^2/2$.

Now, one can show that this upper bound is attained. Define $\bar{p}(x) = 1/2(\delta_{-a}(x) + \delta_a(x))$. Then, a straightforward calculation yields $\mathrm{Var}_{1/2(\bar{p}(x)+\delta_0(x))} = a^2/2$. □

Proof of proposition 2.3

It was already shown in the proof of proposition 2.2 that the lower bound is attained. It remains to show that the upper bound is attained, i.e. that $\mathrm{Var}_{1/2(p(x)+\delta_0(x))} = 0 \Leftrightarrow p(x) = \delta_0(x)$.

"⇐": $\mathrm{Var}_{\delta_0(x)} = 0^2 \cdot 1 - (0 \cdot 1)^2 = 0$.

"⇒": Let $q(x)$ be an arbitrary pdf. Without loss of generality, one can assume that $q(x)$ is zero mean. Hence, $0 = \mathrm{Var}_{q(x)} = \sum_x x^2 q(x)$, so $q(x) = 0\ \forall x \neq 0$. Since it must be that $\sum_x q(x) = 1$, it follows that $q(0) = 1$, i.e. $q(x) = \delta_0(x)$. By plugging in $q(x) = 1/2(p(x) + \delta_0(x)) = \delta_0(x)$, it follows that $p(x) = \delta_0(x)$. □

Chapter 3

Steering vector mismatch analysis and adaptation scheme

In many filter applications the exact steering vector is not know, and thus, robust beamforming methods have to be used. In this chapter, an algorithm which achieves robust beamforming via target tracking is proposed. In contrast to existing approaches, the algorithm works on sparse signals with arbitrary steering vector shapes, and the parameters of the algorithm are adapted in an optimal way. This is achieved by deriving and evaluating the probability of detection and false alarm for general steering vector mismatches. These probabilities are used to adjust the parameters, such that the number of false positive and false negative detections is minimal. Simulations confirm the theoretic results and show that the algorithm performs better than a generic approach.

3.1 Introduction and problem formulation

Let us assume a simplified model of Eq. 1.1 in the sense that only a single, spatial waveform is present in the data:

$$\boldsymbol{x}[t] = \bar{\boldsymbol{q}} \cdot s[t] + \boldsymbol{n}[t], \tag{3.1}$$

where $\bar{\boldsymbol{q}}$*is the steering vector (see Sec. 1.2.1), s is the source signal and $\boldsymbol{n}$ is a noise vector sampled from a stationary, zero mean Gaussian distribution with covariance matrix $\boldsymbol{C}$, i.e $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$. The notation using angular brackets, e.g. $\boldsymbol{x}[t]$, indicates a vectorial quantity at time t, in contrast to $\boldsymbol{x}(t)$ which denotes the vector entry at dimension t (as e.g. in Chap 2). The observation vector $\boldsymbol{x}[t]$ at time t, also called snapshot, has dimension $N \times 1$. We assume that used filters for detection are of the form

$$\boldsymbol{f} = \frac{\boldsymbol{H} \cdot \bar{\boldsymbol{p}}}{\bar{\boldsymbol{p}}^{\top} \cdot \boldsymbol{H} \cdot \bar{\boldsymbol{p}}}, \tag{3.2}$$

and a signal is declared as detected when the filter output exceeds a certain threshold γ, i.e. $\boldsymbol{f}^{\top} \cdot \boldsymbol{x}[t] \geq \gamma$.

*In this chapter we write $\bar{\boldsymbol{q}}$ to denote the deterministic waveform as $\boldsymbol{q}$ will indicate a random variable.

The expression in Eq. 3.2 covers the MVDR, MPDR* and general diagonal loading filters [48]. We refer to $\bar{\boldsymbol{q}}$ as the actual steering vector and to $\bar{\boldsymbol{p}}$ as the nominal steering vector. In order to maximise the detection performance it is desired that $\bar{\boldsymbol{p}} = \bar{\boldsymbol{q}}$, otherwise one has a, so called, steering vector mismatch. Since in general, however, the true steering vector is not known a priori, it has to be estimated from the data, which usually implies that $\bar{\boldsymbol{p}} \neq \bar{\boldsymbol{q}}$.

The field of robust beamforming emerged with the aim of designing filters which deliver acceptable detection performances even under steering vector mismatches [66]. Most of these approaches assume a model, either deterministic, or more recently, probabilistic, describing the steering vector mismatch and include this model into the optimization problem for the filter. This approach works well when the steering vector error is relatively small and stays constant over time. However, the actual steering vector might represent a target which is moving in space, as for example in the case of radar applications. Hence, it is $\bar{\boldsymbol{q}} = \bar{\boldsymbol{q}}[t]$ and, since the future trajectory of the target is not known, the mismatch between $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{q}}$ can become arbitrarily large if $\bar{\boldsymbol{p}}$ is not adapted.

Besides many other fields in which linear filters are used, such as digital communications systems or speech enhancement, they have been applied to biosignals. In particular, they are used for processing electrophysiological recordings from electrodes, either as a spike detection [102, 204] or a spike sorting technique (see literature in Chap. 4). In this case, the spatial model in Eq. 3.1 is adapted to a temporal model $x(t) = \sum_\tau \bar{q}(\tau)s(t-\tau) + n(t)$, $\bar{\boldsymbol{q}}$ representing the waveform of the action potential, and $s(t)$ being the neuronal firing sequence. Due to tissue relaxation the distance between the electrode and the neuron is changing over time, which leads to an altered observed waveform of the action potentials [21]. The above mentioned spike detection and sorting methods, all of which are relying on filters shown in Eq. 3.2, will suffer from performance degradation since no robust beamforming methods were used.

To our knowledge there are only few methods which try to achieve robust beamforming by tracking, i.e. by adapting the nominal steering vector according to the changes of the actual steering vector. In [2] an adaptation scheme for the nominal steering vector based on the generalized sidelobe canceller algorithm is proposed. However, the noise is assumed to be white and an optimal adaptation rate is said to exist only in the case when the actual steering vector is not arbitrary but depends just on a single parameter, namely, the direction of arrival. This makes the algorithm unsuitable for applications where the steering vector cannot be represented by some simple underlying model.

A similar scheme is presented in [63]. In that study, the noise can be colored, but the adaptation rates of the filter and the nominal steering vector are fixed. An algorithm for obtaining optimal adaptation, which would naturally depend on the system parameters, such as the target velocity, was not presented.

In [50] another adaptation scheme for updating the filter is proposed. However, only changes in the covariance of the noise are considered but not any changes in the steering vector. Moreover, the adaption scheme depends on several parameters which must be set manually.

In [207] a very high signal-to-noise ratio is assumed, and therefore a noise model

*See Sec. 2.2.1 for a derivation of these filters.

is not taken into account at all. Further, a specific physical model of the change of the actual steering is assumed, making the method unsuitable for more general changes.

In this contribution we consider measured data $\boldsymbol{x}$ in which the source signal s is present only at few specific times (sparse signal). This means that s is not a continuous process, but rather a sparse Bernoulli process. The former was assumed in [63] and implies that the steering vector is present in every measured snapshot. As a consequence of the sparseness, the nominal steering vector can only be adapted after a successful detection and not after every snapshot. To name a few examples, sparse signals are encountered in the aforementioned electrophysiological recordings or in geophysics [30].

In Sec. 3.2.1 the performance of a linear detector in the case of steering vector mismatch is derived. This result is used in Sec. 3.2.2 to propose an optimal adaptation scheme of the nominal steering vector which leads to robust beamforming via target tracking. Simulations in Sec. 3.3 show the effectiveness of this approach, and conclusive remarks are given in Sec. 3.5.

3.2 Method

3.2.1 Performance analysis under steering vector mismatch

In order to derive an *optimal* adaptation scheme, one has to understand how the performance of a detector depends on a steering vector mismatch. As performance measure we use the probability of detection P_D and false alarm P_{FA}. These two probabilities translate directly into the number of true positive and the number of false positive detections, which is a meaningful performance measure for detectors of sparse signals. P_D is defined by

$$P_D := \text{Prob}[\boldsymbol{f}^\top \cdot \boldsymbol{q} \geq \gamma] = \text{Prob}\left[\frac{\boldsymbol{p}^\top \boldsymbol{H}^\top \cdot \boldsymbol{q}}{\boldsymbol{p}^\top \boldsymbol{H}\boldsymbol{p}} \geq \gamma\right]. \tag{3.3}$$

The random variable $\boldsymbol{q}$ is a noisy observation of the actual steering vector and, according to the model in Eq. 3.1, distributed as $\boldsymbol{q} \sim \mathcal{N}(\bar{\boldsymbol{q}}, \boldsymbol{C})$. Since the nominal steering vector will be based on an estimation from noisy data, $\boldsymbol{p}$ is also a random variable and is assumed to be distributed as $\boldsymbol{p} \sim \mathcal{N}(\bar{\boldsymbol{p}}, \boldsymbol{D})$ (in general $\boldsymbol{D} \neq \boldsymbol{C}$).

In the case of the MVDR beamformer, we have $\boldsymbol{H} = \boldsymbol{C}^{-1}$, whereas in the case of the MPDR beamformer, $\boldsymbol{H} = (\text{Cov}(\boldsymbol{x}))^{-1}$. Also for diagonal loading filters, $\boldsymbol{H}$ is in general positive semi-definite or positive definite, and symmetric. We suppose that $\boldsymbol{H}$ can be estimated on the basis of a large amount of samples. Therefore, it is justified to assume that the estimate of $\boldsymbol{H}$ is so accurate that there is no mismatch between the estimate and the true covariance, and its distribution can be neglected. Summarising, the probability of detection is given by

$$P_D = 1 - \text{cdf}_\gamma\left(\frac{\boldsymbol{p}^\top \boldsymbol{H}\boldsymbol{q}}{\boldsymbol{p}^\top \boldsymbol{H}\boldsymbol{p}}\right), \tag{3.4}$$

where the notation $\text{cdf}_a(z) := \text{Prob}[z \leq a]$ is used. The probability of false alarm P_{FA} is obtained by setting $\bar{\boldsymbol{q}} = \boldsymbol{0}$ in Eq. 3.4. Hence, one has to obtain the cumulative distribution function of a ratio of quadratic forms of Gaussian random variables in order to analyse the performance of the desired detectors.

The expression in Eq. 3.4 can be simplified by defining $\boldsymbol{r} := (\boldsymbol{p}^\top, \boldsymbol{q}^\top)^\top$. Then, $\boldsymbol{r}$ is distributed as $\boldsymbol{r} \sim \mathcal{N}(\bar{\boldsymbol{r}}, \boldsymbol{G})$, where $\bar{\boldsymbol{r}} := (\bar{\boldsymbol{p}}^\top, \bar{\boldsymbol{q}}^\top)^\top$ and $\boldsymbol{G} := \begin{pmatrix} \boldsymbol{D} & \boldsymbol{E} \\ \boldsymbol{E} & \boldsymbol{C} \end{pmatrix}$, where $\boldsymbol{E}$ denotes the covariance between $\boldsymbol{p}$ and $\boldsymbol{q}$. This definition allows to rewrite the random variable in Eq. 3.4 as

$$\frac{\boldsymbol{p}^\top \boldsymbol{H} \boldsymbol{q}}{\boldsymbol{p}^\top \boldsymbol{H} \boldsymbol{p}} = \frac{\boldsymbol{r}^\top \begin{pmatrix} \boldsymbol{0} & \boldsymbol{H} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \boldsymbol{r}}{\boldsymbol{r}^\top \begin{pmatrix} \boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \boldsymbol{r}} = \frac{\tilde{\boldsymbol{r}}^\top \boldsymbol{A} \tilde{\boldsymbol{r}}}{\tilde{\boldsymbol{r}}^\top \boldsymbol{B} \tilde{\boldsymbol{r}}}, \tag{3.5}$$

where $\tilde{\boldsymbol{r}} \sim \mathcal{N}(\boldsymbol{G}^{-1/2} \cdot \bar{\boldsymbol{r}}, \boldsymbol{1})$, $\boldsymbol{A} := \boldsymbol{G}^{1/2^\top} \cdot \begin{pmatrix} \boldsymbol{0} & \boldsymbol{H} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \cdot \boldsymbol{G}^{1/2}$, and $\boldsymbol{B} := \boldsymbol{G}^{1/2^\top} \cdot \begin{pmatrix} \boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \cdot \boldsymbol{G}^{1/2}$, where $\boldsymbol{G}^{1/2}$ denotes the unique non-negative square root of $\boldsymbol{G}$.

The distribution of the ratio on the right hand side of Eq. 3.5 has been analysed for a long time, since it is of importance in econometrics and statistics [60][†]. The analysis is mainly focused on cases where $\boldsymbol{A}$ is symmetric and $\boldsymbol{B}$ is positive semi-definite or positive definite. In the definition above, $\boldsymbol{A}$ is not symmetric. However, we can replace $\boldsymbol{A}$ by the symmetrised version of it $\tilde{\boldsymbol{A}} := 1/2 \cdot (\boldsymbol{A} + \boldsymbol{A}^\top)$, since it is $\boldsymbol{z}^\top \boldsymbol{A} \boldsymbol{z} = \boldsymbol{z}^\top \boldsymbol{A}^\top \boldsymbol{z}$ for any square matrix $\boldsymbol{A}$ and any vector $\boldsymbol{z}$, thus $\boldsymbol{z}^\top \tilde{\boldsymbol{A}} \boldsymbol{z} = \boldsymbol{z}^\top \boldsymbol{A} \boldsymbol{z}$. Since $\boldsymbol{G}$ is positive definite (it is a covariance matrix) it follows that $\boldsymbol{B}$ is positive semi-definite if $\boldsymbol{H}$ is positive semi-definite. In the case of the beamformers considered herein, $\boldsymbol{H}$ is indeed positive semi-definite or positive definite as discussed before. Finally, it is

$$P_D = 1 - \mathrm{cdf}_\gamma \left(\frac{\tilde{\boldsymbol{r}}^\top \tilde{\boldsymbol{A}} \tilde{\boldsymbol{r}}}{\tilde{\boldsymbol{r}}^\top \boldsymbol{B} \tilde{\boldsymbol{r}}} \right) = 1 - \mathrm{cdf}_0 \left(\tilde{\boldsymbol{r}}^\top \left(\tilde{\boldsymbol{A}} - \gamma \boldsymbol{B} \right) \tilde{\boldsymbol{r}} \right). \tag{3.6}$$

There exists a closed form expression for the cdf in Eq. 3.6, however, it involves an infinite series of top order polynomials with a very slow convergence rate [60]. Techniques for fast evaluation by means of saddlepoint approximations were developed in [130, 25]. These approximations, which are based on asymptotic expansions of the corresponding inversion integrals, however, are not accurate enough for the problem at hand.

Instead, the basis of the presented algorithm[‡] is a result from [91], where it was shown that the inversion integral can be reduced to integrating a real function over an infinite range, namely

$$\mathrm{cdf}_0 \left(\tilde{\boldsymbol{r}}^\top \left(\tilde{\boldsymbol{A}} - \gamma \boldsymbol{B} \right) \tilde{\boldsymbol{r}} \right) = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty \frac{\sin\left(\beta(u)\right)}{u \cdot \rho\left(u\right)} \mathrm{d}u, \tag{3.7}$$

where β and ρ mainly depend on the eigenvalues λ_i and eigenvectors $\mathbf{e}_i$ of $\tilde{\boldsymbol{A}} - \gamma \boldsymbol{B}$, i.e. $\beta(u) := \frac{1}{2} \sum_i \arctan(a_i) + \frac{\theta_i a_i}{c_i}$, $\rho(u) := \exp\left\{ \frac{1}{2} \sum_i \frac{\theta_i b_i}{c_i} + \frac{1}{4} \ln(c_i) \right\}$, $a_i := \lambda_i u$, $b_i := {a_i}^2$, $c_i := 1 + b_i$, and $\theta_i := (\boldsymbol{e}_i^\top \boldsymbol{G}^{-1/2} \bar{\boldsymbol{r}})^2$.

This integral is evaluated by mapping it onto the finite range $[0, 1]$ via the substitution $u = (1-v)/v$ and replacing the integrand at $v = 0$ and $v = 1$ with its limits, which are zero

[†]For example this ratio arises when studying Gaussian auto-regressive models or in many test statistics, see also [117].

[‡]Developed by Prof. Simon Broda from the Department of Quantitative Economics, Universiteit van Amsterdam, The Netherlands.

and $1/2 \sum_i (1 + \theta_i)\lambda_i$ (see Sec. A.1 for a derivation), respectively. The resulting integral can then be straightforwardly evaluated using standard numeric integration routines.

3.2.2 Adaptation scheme

To account for the time varying actual steering vector $\boldsymbol{q}[t]$, the nominal steering vector is adapted after constant time steps of length T. Notably, after every period T, the nominal steering vector is estimated as the sample mean of the K last detections $\boldsymbol{x}^i$ ($\boldsymbol{x}^i := \boldsymbol{x}[t(i)]$ such that $\boldsymbol{f}^\top \cdot \boldsymbol{x}[t(i)] \geq \gamma$), i.e. $\bar{\boldsymbol{p}} = 1/K \cdot \sum_{i=K_{\max}-K+1}^{K_{\max}} \boldsymbol{x}^i$, where $K_{\max}$ denotes the total number of detections at time t.

The question arises, how many detections should be used for this estimate. If a large K is chosen, the estimate of the mean of $\boldsymbol{p}$ will be robust ($\boldsymbol{D}$ small), but the deviation from the true mean steering vector might be large ($\bar{\boldsymbol{p}} \neq \bar{\boldsymbol{q}}$). On the other hand, if a smaller K is chosen, then on average the estimated mean of the nominal steering vector will be closer to the mean of the actual steering vector ($\bar{\boldsymbol{p}} \approx \bar{\boldsymbol{q}}$), but the estimate will be more noisy ($\boldsymbol{D}$ large).

As an optimal trade-off, the value for K should be chosen such that the performance M of the detector is maximised. Usually, it is desirable that the amount of total relative error, which is the sum of false positive (FP) and false negative (FN) detections, is minimal. Therefore, we define our performance as

$$M := \beta_1 \cdot P_D + \beta_2 \cdot (1 - P_{FA}), \tag{3.8}$$

where P_D rsp. P_{FA} are given by the expression in Eq. 3.6, and β_1 rsp. β_2 are weighting parameters which determine their importance. Consequently, the optimal value for K is given by

$$K_{\text{opt}} = \underset{K}{\operatorname{argmax}} \{M(K)\}. \tag{3.9}$$

In order to use the expression of P_D in Eq. 3.6 for this adaptation scheme, the assumptions made in its derivation have to be verified. Firstly, it was assumed that $\boldsymbol{p}$ is Gauss distributed. This would be entirely correct only if all detections $\boldsymbol{x}^i$ were true positive, there were not any false negative detections, and the target was stationary ($\bar{\boldsymbol{q}}[t] = \text{const.}$). For reasonable threshold values γ, most detections will indeed be true positive and only few signal occurrences will be missed, and if the change of the actual steering vector is not too rapid, then $\boldsymbol{p}$ will be approximately Gauss distributed as $\boldsymbol{p} \sim \mathcal{N}(1/K \cdot \sum_{i=K_{\max}-K+1}^{K_{\max}} \boldsymbol{x}^i, \boldsymbol{D})$, where $\boldsymbol{D} = 1/K \cdot \boldsymbol{C}$. Secondly, the covariance matrix $\boldsymbol{H}$ was assumed to be known. The validity of this assumption depends on the amount of available data. In the case of continuously sampled data recordings, as in biomedical recordings or wireless communications, a large amount of observed samples $\boldsymbol{x}$ are indeed available, and hence, $\boldsymbol{H}$ can be estimated very reliably.

The evaluation of P_D requires the knowledge of the true actual steering vector at all time intervals, i.e. $\bar{\boldsymbol{q}}[k \cdot T]$, where k is an integer. Generally, this information is not available and the actual steering vector has to be estimated from the data themselves as well. For this estimation, again the sample mean of Q last detections is used, i.e. $\bar{\boldsymbol{q}} = 1/Q \cdot \sum_{i=K_{\max}-Q+1}^{K_{\max}} \boldsymbol{x}^i$. Assuming a linear change of the shape of the actual steering

Table 3.1: Maximum absolute error for different cdf evaluation techniques in the case of a F-distribution. SdpA1 and SdpA2 denote the first and second order saddlepoint approximation techniques.

Our method	SdpA1	SdpA2
$1.6 \cdot 10^{-8}$	0.0121	0.0089

vector * this will give an estimate of $\bar{\boldsymbol{q}}$ for a time t_e, $t_e < k \cdot T$. A large value of Q will lead to a more robust estimate, but also $t_e \ll kT$, which means that there is a large "lag" in the adaptation. A small value of Q might give a more noisy estimate in the specific setting (but still a correct estimate *on average*), as an advantage, however, the adaptation follows the change of the actual steering vector more rapidly. Despite this estimation, the covariance of q is still given by $\boldsymbol{C}$ (and not $1/Q \cdot \boldsymbol{C}$), hence, $\boldsymbol{q} \sim \mathcal{N}(1/Q \cdot \sum_{i=K_{\max}-Q+1}^{K_{\max}} \boldsymbol{x}^i, \boldsymbol{C})$.

We assumed complete blindness about the temporal structure of $\bar{\boldsymbol{q}}[t]$. Therefore, an optimal value of Q cannot be provided. In Sec. 3.3.2, however, we will show that the adaptation scheme works reliably for a wide range of Q values, thus the algorithm exhibits a robust behavior with respect to this parameter.

3.3 Results

3.3.1 Comparison of cdf evaluation techniques

In this section we briefly show the accuracy of the cdf evaluation technique presented at the end of Sec. 3.2.1 by applying it to two examples. The expression in Eq. 3.7 was evaluated via MATLAB® using the standard commands `eig` for finding eigenvalues and eigenvectors, and `quadl` for numerical integration based on an adaptive Lobatto rule. The integration error was set to 10^{-8}. For comparison, both examples were also evaluated by an first and second order saddlepoint approximation (see [25] and [24] respectively).

The first example is according to [60]. Namely, in the special case when $\tilde{\boldsymbol{A}} = \boldsymbol{l} \cdot \boldsymbol{l}^\top / (\boldsymbol{l}^\top \cdot \boldsymbol{l})$, where $\boldsymbol{l}$ is a N-dimensional vector of ones, $\boldsymbol{B} = (\mathbf{1}_{N\times N} - \boldsymbol{A})/(N-1)$, and $\tilde{\boldsymbol{r}} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}_{N\times N})$, then, the cdf in Eq. 3.6 is given by the cdf of the F-distribution $F(1, N-1)$.

As an example, the value $N = 10$ was chosen, and the threshold γ was varied in steps of 0.05 in the interval $[0, 11]$. The exact value of the cdf was assumed to be given by the MATLAB function `fcdf(`γ`,1,N-1)`. The three techniques were compared by means of the maximum absolute error between the exact cdf value and the value given by the corresponding technique for all thresholds. The results are shown in Table 3.1.

In the second example, we considered $\tilde{\boldsymbol{r}} = \left(\left(\boldsymbol{S}^{-1/2}\boldsymbol{r_1}\right)^\top, \boldsymbol{r_2}^\top\right)^\top$, where $\boldsymbol{r_1}$ is an arbitrary N_1-dimensional vector, $\boldsymbol{r_2}$ is the N_2-dimensional zero vector, and $\boldsymbol{S}$ is a diagonal matrix. Moreover, if we choose $\boldsymbol{A} = \begin{pmatrix} N_2 \cdot \mathbf{1}_{N_1\times N_1} & \mathbf{0}_{N_1\times N_2} \\ \mathbf{0}_{N_2\times N_1} & \mathbf{0}_{N_2\times N_2} \end{pmatrix}$ and $\boldsymbol{B} = \begin{pmatrix} \mathbf{0}_{N_1\times N_1} & \mathbf{0}_{N_1\times N_2} \\ \mathbf{0}_{N_2\times N_1} & N_1 \cdot \mathbf{1}_{N_2\times N_2} \end{pmatrix}$, then, the cdf in Eq. 3.6 is given by the cdf of the noncentral F-distribution $ncF(N_1, N_2, \lambda)$

*This is true for small time intervals considering a first order Taylor approximation.

Table 3.2: Maximum absolute error for different cdf evaluation techniques in case of a noncentral F-distribution. The same notation as in Table 3.1 is used.

Our method	SdpA1	SdpA2
$2.5 \cdot 10^{-6}$	0.0785	0.0785

with $\lambda = \sum_{i=1}^{N_1} (\boldsymbol{r_1}_i)^2 / \boldsymbol{S}_{i,i}$, which can be evaluated very accurately for example with the MATLAB function `ncfcdf(`γ`,`N_1`,`N_2`,`λ`)`.

We chose $\boldsymbol{r_1} = (1, 2, ..., 6)^\top$, $N_2 = 8$, $\boldsymbol{S} = \text{diag}(7, 6, ..., 2)$, and the threshold γ was varied in the interval $[1, 50]$ in steps of 0.05. The maximum absolute error to the exact value over all thresholds is reported in Table 3.2.

Clearly, in both examples the used method based on numerical integration is more accurate than both of the saddlepoint approximations.

3.3.2 Simulations

The proposed adaptation scheme was tested on data generated by Monte Carlo simulations. A single dataset consisted of 1400 snapshots, half of them containing the steering vector. White noise with a variance of 0.25, i.e. $\boldsymbol{C} = 0.25 \cdot \boldsymbol{1}_{N \times N}$, was used for the noise source. In the first 400 and last 400 snapshots a constant actual steering vector $\bar{\boldsymbol{q}}_1$ rsp. $\bar{\boldsymbol{q}}_2$ was simulated having the dimension $N = 7$. The signal-to-noise ratio (SNR) of both steering vectors was identical (6.0db), however, they were orthogonal to each other. In between, a normalised linear mixture was simulated. To sum up

$$\bar{\boldsymbol{q}}[t] = \begin{cases} \bar{\boldsymbol{q}}_1, \forall t \leq 400 \\ \alpha[t] \cdot \bar{\boldsymbol{q}}_3[t], \forall t \in [400, 1000] \\ \bar{\boldsymbol{q}}_2, \forall t \geq 1000 \end{cases} \tag{3.10}$$

where $\bar{\boldsymbol{q}}_3[t] := \frac{\bar{\boldsymbol{q}}_2 - \bar{\boldsymbol{q}}_1}{600} \cdot t + \frac{1000 \cdot \bar{\boldsymbol{q}}_1 - 400 \cdot \bar{\boldsymbol{q}}_2}{600}$. In this setting every snapshot corresponds to one time unit. The value of $\alpha[t]$ was set such that $\text{SNR}(\alpha[t] \cdot \bar{\boldsymbol{q}}_3) = 6.0\text{db}\ \forall t$. This guaranteed that any performance loss of the detector was caused by the change in shape of the actual steering vector, and not due to a simple decrease in SNR.

The first 400 snapshots served as initialisation, and were used to estimate the initial nominal steering vector. The adaptation scheme, hence, was applied on the snapshots $401 - 1400$. For this, the performance given by Eq. 3.8 was calculated for all K values (see Fig. 3.1), whereas the values for β_1 and β_2 were both set to 0.5. The expression in Eq. 3.7 was evaluated in the same way as described in Sec. 3.3.1. Finally, the K value for which M was the largest was chosen as K_{opt} (corresponds to Eq. 3.9), and the filter was re-calculated (using Eq. 3.2) based on the adapted nominal steering vector.

The MVDR beamformer was used as a filter, which implied $\boldsymbol{H} = \boldsymbol{C}^{-1}$. The steering vector was adapted after every 10th snapshot, i.e. $T = 10$, see Fig. 3.2. Three different values for Q were tested, namely $Q = \{10, 50, 150\}$. For simplicity, the covariance between $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{q}}$ was ignored, i.e. $\boldsymbol{E} = \boldsymbol{0}$ in the calculation of $\boldsymbol{G}$ (see Sec. 3.2.1), but similar results were obtained also without this simplification. In total, 10 datasets were simulated, over which the results were averaged. The proposed adaptation scheme was

compared to approaches in which a fixed, pre-defined number of detections was used for the computation of the nominal steering vector. The results are shown in Fig. 3.3.

3.3.3 Evaluation and comparison

From Fig. 3.1 one can observe that $M(K)$ exhibits a kind of plateau region, on which the optimal maximal value is located, meaning that the performance does not change a lot when K is varied in that region. This results in large variations of K_{opt} across the simulations (visible in Fig. 3.2), since the maximum might occur at different values of K due to fluctuations. Although the standard deviation of K_{opt} is large, this has small influence on the total performance (i.e. small standard deviations in Fig. 3.3), which is again consistent with the theoretic prediction in Fig. 3.1. Fig. 3.2 also demonstrates the tradeoff between minimisation of the mismatch between the means of the steering vectors and the minimisation of the variance of the nominal steering vector estimation. Notably, when the steering vector is close to stationary (at the beginning and the end of the simulations), the variance is minimised by taking a large number of detections for estimation, i.e. $K \gg Q$. On the other hand, when the actual steering vector varies rapidly, it is better to use a number closer to the number of detections used for the estimation of the actual steering vector, i.e. $K \approx Q$, in order to reduce the mismatch between the means of the steering vectors.

Fig. 3.3 illustrates that despite a rather rapid variation and extreme thresholds, both of which lead to a violation of the assumptions under which the adaptation scheme was derived, the proposed algorithm performs close to the theoretically optimal approach for a wide range of thresholds.

A non adaptive scheme, which assumes that the steering vector does change signif-

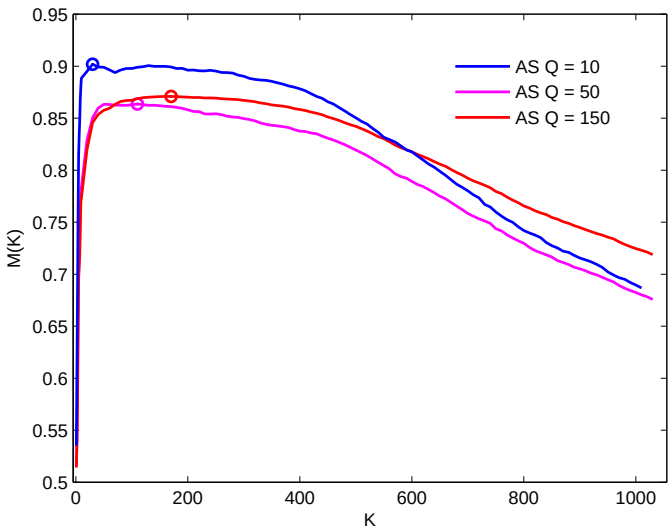

Figure 3.1: Exemplarily, the performance M (see Eq. 3.8) as a function of K ($K = 1, 2, ..., 9, 10, 20, 30, ..., K_{\text{max}}$, where K_{max} was the number of total detections at the corresponding time) is shown for $t = 700$ and for a threshold of $\gamma = 0.525$. The maximum value of M is marked with a circle, which determined K_{opt}. The different curves correspond to adaptation schemes (AS) with different Q values.

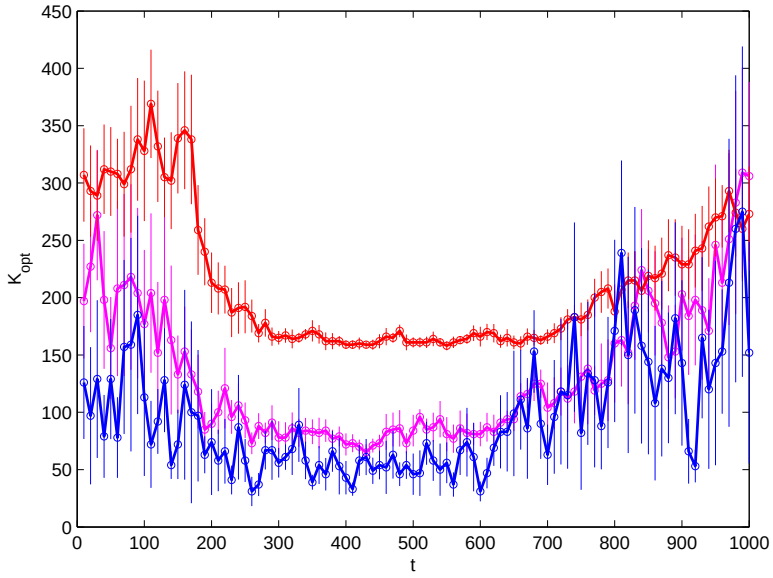

Figure 3.2: Average value of K_{opt} as a function of time. The same threshold and color coding as in Fig. 3.1 was used. The vertical lines indicate the standard deviation across the 10 simulations.

icantly, would use all detections for the steering vector estimation. Even if a change in the steering vector is detected, a fixed number of detections for estimation would be used in such a non-optimal adaptation scheme. Both approaches perform worse than the derived optimal adaptation scheme, see Table 3.3.

The simulations also revealed that the proposed adaptation scheme works well for a wide range of Q values. In particular for the case when the total relative error is minimised, i.e. the threshold is around 0.5, the performance of the three considered cases is very similar (Fig. 3.3).

Table 3.3: Average area under the curve (AUC) in descending order for various adaptation schemes. The AUC was calculated based on the results in Fig. 3.3 with a range of FP $\in [0.002, 0.73]$.

Opt.	$Q = 50$	$Q = 150$	$Q = 10$	$K = 400$	$K = K_{\max}$
0.6471	0.6291	0.6234	0.6142	0.6016	0.5169

3.4 Discussion and related literature

A large error in the (noise) covariance matrix estimation can be caused when only few data snapshots are available. However, in the case of continuously sampled recordings containing time-varying sparse signals, such as electropyhsiological recordings, other conditions apply than for radar applications. Namely, the covariance matrix can be estimated reliably due to the large amount of data, but the estimate of the steering vector is more problematic as the signal is sparse.

Nevertheless, most of the research regarding the analysis of performance in the case of a steering vector mismatch was done in the field of beamforming, i.e. radar systems. Probably the first study was presented in [99], in which it was assumed that only the

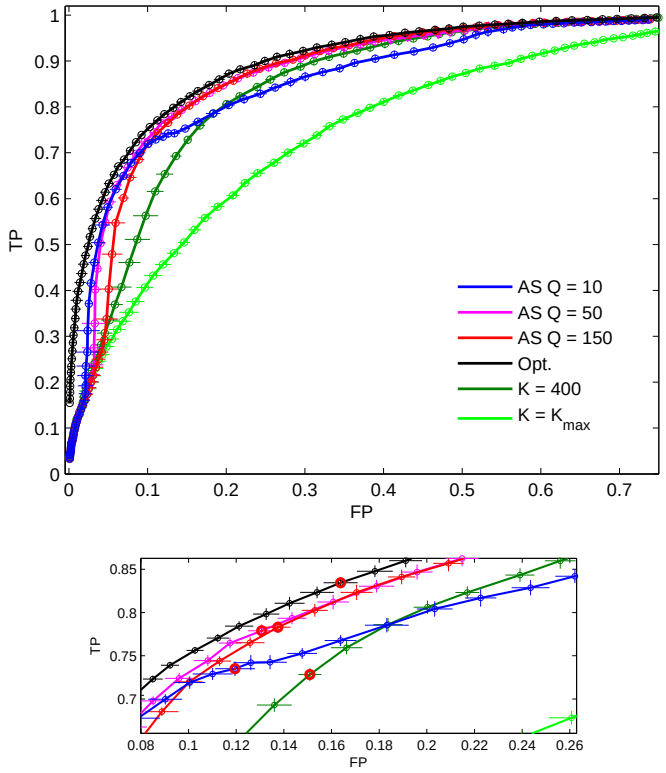

Figure 3.3: *Top:* Receiver operating characteristics curves: For every adaptation scheme the threshold γ was varied from -0.5 up to 1.5 in steps of 0.025, and the resulting average relative number of false positive (FP) and true positive (TP) were plotted together with their standard deviations. The black curve corresponds to the theoretically best possible adaptation scheme, in which the filter was calculated based on the true steering vector, i.e. $\bar{\boldsymbol{p}} = \bar{\boldsymbol{q}}[t]$, $t = k \cdot T$. The green lines resulted, when a non-optimal number of detections was used for adaptation, i.e. $K = 400$ (dark green) rsp. $K = K_{\max}$ (light green). The color coding of the other lines is the same as in Fig. 3.1. The corresponding areas under the curves are listed in Table 3.3. *Bottom:* Magnification of a part of the top figure. In the herein considered setting the optimal threshold which minimises the total error is $\gamma = 0.5$. The corresponding points on the ROC curves are marked with a red circle.

actual steering vector is a random variable, whereas the nominal steering vector is deterministic. On the other hand, it was assumed that the covariance matrix $\boldsymbol{H}$ is Wishart distributed. This study was later generalized to a broader class of detectors in [94]. In [175] it was furthermore assumed that the mean of the covariance matrix distribution does not necessarily correspond to the true mean. However, the mean mismatch was not completely arbitrary, but had to follow a so called generalized eigen-relation. This restriction was dropped in [135, 17], allowing for an arbitrary mismatch in the covariance matrix.

The performance of a beamformer was also analysed in [228], but not in terms of the probability of detection and false alarm, but in terms of a simpler quantity called signal-to-interference-plus-noise ratio (SINR). This analysis was later extended to the case when a steering vector mismatch is present, under the restriction that the mean of the nominal steering vector corresponds to the actual steering vector [229]. In [14] similar assumptions as in [229] were made, however, the covariance matrix was more general covering also so called diagonal loading filters. Note that the average SINR provides less information than the probability of detection and false alarm. P_D and

P_{FA} can be used to directly account for the number of true positive and true negative detections, as well as to evaluate more complex performance measures as in Chap 2.

In [1] only a deterministic mismatch was considered, and the covariance matrix was directly dependent on the nominal steering vector as well.

We used the novel performance analysis to derive an adaptation scheme, but also in the case of stationary data, as a way to compute the nominal steering vector. The initialisation of a beamformer, i.e. the determination of the nominal steering vector is called calibration in the beamforming literature. Such calibration algorithms rely either on algebraic techniques, e.g. in [215, 170] the steering vector is found as the intersection of three different sets, or on statistical properties of the data using blind source separation techniques, e.g. in [26] the blind source separation algorithm JADE is proposed, or using maximum likelihood estimations, e.g. in [12, 13, 51]. These algorithms may be used to find the nominal steering vector in the initialisation phase, when the actual steering vector is constant, but are unsuitable for temporally changing environments.

We connected the performance analysis of a beamformer to a distribution used in econometrics and statistics*. For this, we had to assume that the covariance matrix is deterministic, i.e. it can be estimated from an infinite number of data samples. If, however, only a finite number of samples is available and the noise is Gaussian, then the estimate of the covariance matrix will follow a Wishart distribution. Still, the problem could be reduced to the evaluation of a distribution similar to $\boldsymbol{r}^\top \boldsymbol{M}\boldsymbol{r}$ as in Eq. 3.6. Hence, in the following we give a short overview of some existing literature dealing with related problems.

The form $\boldsymbol{r}^\top \boldsymbol{M}\boldsymbol{r}$ can be seen as a trace of a matrix [70], as the product of two random variables [123, 68, 69, 52], or as a quadratic form [77]. The last interpretation seems to be the most common, and in the following we list some works which deal with this case when solely $\boldsymbol{r}$ is a random variable.

The author in [77] gave a formula for the distribution in terms of Laguerrian expansions. However, according to [60], either this expansions are not convergent everywhere, or they contain unsolved integrals.

In [7, 71] the pdf and cdf were derived for some special cases only, whereas in [125, 117, 156] only moments were computed.

The exact cdf was first derived in [59], for the central case (zero mean Gaussian distributed vector $\boldsymbol{r}$). This result was later extended to the non-central case, and also an expression for the pdf was given [60]. As already mentioned in Sec. 3.2.1 the exact formula is slow to evaluate. Hence, fast methods based on numerical integration have been developed. For example in [120] some advice is given how to choose a proper step size and truncation for evaluation the cdf via the integral presented in [91]. In [119, 23] numerical integration is done to evaluate the pdf.

3.5 Conclusion

In this chapter P_D and P_{FA} were derived in the case when both actual and nominal steering vector are random. It was shown that these probabilities can be linked to quantities

*See [243] for an overview of some fields where the distribution is used.

studied by the econometrics and mathematical statistics communities. A very accurate numerical method for their efficient evaluation was applied and compared to other techniques based on saddlepoint approximations.

Furthermore, this analysis allowed to propose an optimal adaptation scheme for the nominal steering vector. In this sense, an algorithm for robust beamforming via target tracking was proposed. In our algorithm the shape of the steering vector as well as its temporal evolution can be arbitrary. This makes the algorithm also suitable for applications beyond radar and antenna systems; amongst others, it can be used for digital communication systems or in biomedical signal processing[†].

To our knowledge the analysis of the beamformer performance when both steering vectors as well as the estimate of the covariance matrix deviate from the true quantities has not been carried out yet. This problem is left for further research.

[†]In particular, it will be applied in the spike detection algorithm presented in Chap. 5.

Chapter 4

Online spike sorting with instantaneous overlap resolution

The work in this chapter was done in equal proportion together with Felix Franke.

In this chapter we address the problem of spike sorting, as defined in Eq. 1.4. Many algorithms have been developed to this end, however, to date, none of them manages to fulfil a set of demanding requirements. In particular, it is desirable to have an algorithm that operates online, detects and classifies overlapping spikes in real time, and that adapts to non-stationary data. Here, we present a combined spike detection and classification algorithm, which explicitly addresses these issues. Our approach makes use of linear filters to find a new representation of the data and to optimally enhance the signal-to-noise ratio. We introduce a method called "Deconfusion" which de-correlates the filter outputs and provides source separation. Finally, a set of well-defined thresholds is applied and leads to simultaneous spike detection and spike classification. By incorporating a direct feedback, the algorithm adapts to non-stationary data and is, therefore, well suited for acute recordings. We evaluate our method on simulated and experimental data, including recordings from the prefrontal cortex of awake behaving macaques. We compare the results to existing spike sorting methods, and conclude that our algorithm meets all of the mentioned requirements and outperforms other methods under realistic signal-to-noise ratios and in the presence of overlapping spikes.

4.1 Introduction

In order to understand higher brain functions and the interactions between single neurons, an analysis of the simultaneous activity of a large number of individual neurons is essential. One common way to acquire the necessary amount of neuronal activity data is to use simultaneous extracellular recordings, either with single electrodes or, more recently, with multi-electrodes like tetrodes [154], see also Chap. 1. However, the recorded data does not directly provide the isolated activity of single neurons, but a mix-

ture of neuronal activity from many neurons additionally corrupted by noise. The task of so called "spike sorting" algorithms is to reconstruct the single neuron signals (i.e. spike trains) from these recordings. Many approaches for analysing the data after acquisition, i.e. *offline* spike sorting algorithms, have been developed in the last years; see for example [220, 42, 167, 100, 200, 191, 87, 110, 55]. Although more methods are available in this category, there are several reasons to favour methods which provide results already during the recordings, termed *real-time online* sorting algorithms. For example, real-time online spike sorting techniques are indispensable for conducting "closed-loop" experiments and for brain-machine interfaces [180, 152]. The few existing approaches to real-time online sorting [204, 180, 4] are *clustering based* and have at least one of the following drawbacks (see also Sec. 4.5 for further discussion): 1) They are not explicitly formulated for data acquired from multi-electrodes, 2) they do not resolve overlapping spikes, 3) they do not perform well on data with a low signal-to-noise ratio 4) they are not able to adapt to non-stationarities of the data as caused by tissue drifts. We discuss the reasons and importance of these issues in the following:

1) Multi-electrodes (e.g. tetrodes) provide significantly more information about the local neuronal population than single electrodes [79, 173]. Having several recording electrodes closely spaced instead of one, the same action potential is present on more than one recording channel. The so called stereo-effect - a neuron specific amplitude distribution among the recording channels - allows for a better discrimination between action potentials from different neurons [75]. This allows also for a more reliable resolution of overlapping spikes.

2) Tetrodes record from an increased number of neurons compared to high impedance single electrodes, thus, overlapping spikes are more likely to occur. Also, studies stress the relevance of ensemble coding, which translates into local synchronised firing and hence a raised occurrence frequency of overlapping spikes [182]. To identify such a code, the resolution of overlapping spikes is crucial and efforts have been made addressing this issue [44, 225, 245, 136, 33]. However, the cited approaches are all computationally very expensive, making a real-time online implementation difficult. One of the reasons for this computational complexity is the implementation of separate subroutines for the processing of overlapping spikes, which, additionally, are more complex than the processing steps for non-overlapping spikes.

3) Most of the spike sorting approaches use a stand-alone standard spike detection technique (see for example [38, 152, 173] and Chap. 5 for commonly used spike detection techniques), and a separate classification procedure. Neither the shape of the waveforms nor their change over time or their amplitude distribution across the recording channels is taken into account by the spike detection method. This leads to a poor detection performance, in particular when the signal-to-noise ratio (SNR) is low. Further, the spikes are cut and aligned on some feature (e.g., peak position) as a preprocessing to the classification algorithm. However, overlapping spikes, which severely alter the spike waveform, are not identified as such. This leads to wrong alignments and false classifications by the clustering procedure.

4) There are two general approaches to extracellular recording with electrodes, namely acute and chronic recording methods. In acute recordings, individual electrodes

are advanced into tissue at the beginning of each recording session anew, causing a compression of the tissue [29]. During the experiment the tissue relaxes and the distances between the electrodes and neurons change; an effect called tissue drift [21]. As a consequence, the shape of the measured waveforms and the characteristic of the background noise changes. Sorting algorithms which do not take into account such variations will perform poorly on data from acute recordings.

An approach based on *blind source separation (BSS) techniques* and addressing primarily problems 1) and 4) was presented in [202], in which independent component analysis (ICA) was applied to multi-channel data recorded by tetrodes. Later, the method was adopted to data recorded by dodecatrodes (12 channels) [201]. However, both approaches had to deal with several new problems: Amongst others, time delays between the channels were not considered, biologically meaningless independent components had to be discarded manually, and different neuronal signals with similar channel distributions could not be classified correctly. Furthermore, the methods can only be applied to data recorded with certain electrode types (i.e. tetrodes, dodecatrodes). The most severe problem, though, is the fact that the method cannot deal with data containing neuronal activity from a greater number of neurons than recording channels (overcompleteness).

In this chapter, we present a real-time online spike sorting method based on the BSS idea, which explicitly addresses the four issues 1)-4), but also avoids the drawbacks of the methods in [202] and [201]. In particular, the here proposed method works on data recorded with an arbitrary number of electrodes, and the number of neurons which can be extracted is not limited in any way by the number of electrodes. In sum, a spike sorting algorithm for multi-electrode data, which detects and resolves overlapping spikes with the same computational cost as non-overlapping spikes, is formulated. The method makes optimal use of an arbitrary number of simultaneously recorded channels and can even run on single channel data. Moreover, since spike detection, spike alignment, and spike classification are not separate parts, but are combined into a single algorithm, our method performs well on data with low SNR and containing many overlapping spikes. By incorporating a direct feedback, the algorithm adapts to varying spike shapes and to non-stationary noise characteristics. The algorithm is fully automatic and due to its linear and parallel computation steps it is ideally suited for real-time applications (see Fig. 4.3 for a summary of our method).

This chapter is organised as follows: In Sec. 4.2 we present our method step by step. First, we briefly introduce linear filters. These filters were used in e.g. radar applications [217], geophysics [178] as well as for spike detection [204, 223], but to our knowledge have not been applied in the presented way to spike sorting yet. Moreover, in contrast to those studies, we do not directly apply a threshold to the filter outputs, but consider them as a new representation of the data. In this representation the spike sorting task can be handled as a well defined BSS problem, which we solve with a un-mixing technique we will refer to as “Deconfusion”.

The evaluation of our method is done on a dataset from real recordings and also on simulated data. The experimental setup, used equipment and the characteristic of recorded data are described in Sec. 4.3. The advantages and abilities of the method are

demonstrated in Sec. 4.4. The noise robustness and the ability to successfully resolve overlapping spikes is evaluated systematically on synthetic data. Finally, the method is applied to data from extracellular recordings made in the prefrontal cortex of awake behaving macaques. This data set is particularly challenging, because the tetrodes are not implanted chronically, but inserted before every experiment anew, leading to tissue drifts. We conclude that our method adopts to non-stationarities and also successfully resolves overlapping spikes in real data. A summary and a discussion of further improvements is given in Sec. 4.6.

4.2 Methods

4.2.1 Generative model

The exactly same data model as introduced in Eq. 1.1 as well as the same assumptions as in Eq. 1.2 are used in this chapter. Namely, the measured data $x_{k,t}$ are a convolution of the waveforms $q^i_{k,t}$ with the corresponding intrinsic spike trains s^i corrupted by colored Gaussian noise $n_{k,t}$, i.e.

$$x_{k,t} = \sum_{i=1}^{M} \sum_{\tau} q^i_{k,\tau} s^i_{t-\tau} + n_{k,t} \qquad k = 1, \ldots, N. \tag{4.1}$$

4.2.2 Calculation of linear filters

Spike sorting is achieved when the intrinsic spike trains s^i are reconstructed from the measured data $\boldsymbol{X}$, where $(\boldsymbol{X})_{k,t} := x_{k,t}$. Since, according to the model assumptions, the data were generated by a convolution of intrinsic spike trains with fixed waveforms, the most straightforward procedure would be to apply a deconvolution on $\boldsymbol{X}$ in order to retrieve s^i. For an exact deconvolution a filter with an infinite impulse response is necessary. In general, such a filter is not stable and would amplify noise [178]. Nevertheless, a noise robust approximation for an exact deconvolution can be achieved with finite impulse response filters, to which we will refer as linear filter.

Let us briefly summarise the idea of these filters*: The goal is to construct a set of filters $\left\{\boldsymbol{f}^1, \ldots, \boldsymbol{f}^M\right\}$ such that each filter $\boldsymbol{f}^i$ has a well defined response of 1 to its matching template $\boldsymbol{q}^i$ at shift 0 (i.e. $\boldsymbol{q}^{i\top} \cdot \boldsymbol{f}^i = 1$), but minimal response to the rest of the data. This means that the spikes of neuron i are the signal for filter $\boldsymbol{f}^i$ to detect but will be treated as noise by filter $\boldsymbol{f}^{j\neq i}$.

Incorporating these conditions leads to a constrained optimisation problem

$$\boldsymbol{f}^i = \underset{\boldsymbol{f}^i}{\operatorname{argmin}} \operatorname{Var}\left(\boldsymbol{X} \star \boldsymbol{f}^i\right) \qquad \text{subject to } \boldsymbol{q}^{i\top} \cdot \boldsymbol{f}^i = 1 \tag{4.2}$$

to which the solution are the desired filters (see Sec. 4.7.1 for a more detailed derivation). A major advantage is the fact that the mentioned optimisation problem can be solved

*For more information about linear filters see Chap. 2. Note, however, that a slightly different derivation is presented in the following.

analytically. In particular, the filters are given by the following expression:

$$\boldsymbol{f}^i = \frac{\boldsymbol{R}^{-1}\boldsymbol{q}^i}{\boldsymbol{q}^{i\top}\boldsymbol{R}^{-1}\boldsymbol{q}^i} \qquad i = 1, ..., M \tag{4.3}$$

where $\boldsymbol{R}$ is the data covariance matrix[†]. Linear filters maximise the signal-to-noise ratio and minimise the sum of false negative and false positive detections, and are, therefore, optimal in this sense [138].

4.2.3 Filtering the data

Once the filters are calculated, they are cross-correlated with the measured data, i.e. $\sum_{k,\tau} x_{k,\tau+t} f^i_{k,\tau} =: y^i_t$. Note that we do not have to pre-process the data with a whitening filter, but the filters can be applied directly to $\boldsymbol{X}$. This is because the noise statistics is already captured in the matrix $\boldsymbol{R}$.

From a different point of view, the filtering just changes the representation of the templates. While in the original space the i-th template was represented by $\boldsymbol{q}^i$, its representation in the filter output space is given by the vectors $\boldsymbol{q}^i \star \boldsymbol{f}^j$, $j = 1, ..., M$, where $\left(\boldsymbol{q}^i \star \boldsymbol{f}^j\right)_t := \sum_{k,\tau} q^i_{k,t+\tau} f^j_{k,\tau}$, see also Fig. 4.1. This interpretation of filtering will be useful in the next section.

4.2.4 Deconfusion

The linear filters derived in Sec. 4.2.2 should suppress all signal components except their corresponding template with zero shift. Thus, the filter response to all templates (and their shifted variants) has to be minimal. This already leads to $\left(2T_f - 1\right)\cdot M$ minimisation constraints; a number which is normally greater than the number of free variables of a filter which is $T_f \cdot N$, where T_f is the dimension of the filter. In addition, if the SNR is low, the data covariance matrix $\boldsymbol{R}$ is similar to the noise covariance matrix $\boldsymbol{C}$, i.e. $\boldsymbol{R} \approx \boldsymbol{C}$. The lower the SNR, the less spikes from other neurons a filter will suppress. Thresholding of every filter output $\boldsymbol{y}^i$ individually will, thus, lead to many false positive detections. The idea is to de-correlated the filter output first in order to achieve an improved spike detection and classification.

We have seen in the previous section that each template $\boldsymbol{q}^i$ can be represented in the filter output by M vectors $\boldsymbol{q}^i \star \boldsymbol{f}^j$, $j = 1, ..., M$. Since the detection and classification of the spikes is based on the detection of high positive peak values in the filter output (by construction), all values below zero in the filter output are irrelevant, and thus, can be discarded. As a result, we ignore all values below zero by applying a half-wave rectification $I(x)$ to the filter output $\boldsymbol{Y}$, where

$$I(x) := \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}. \tag{4.4}$$

The next step is to consider $I(\boldsymbol{Y})$ as a linear mixture of different sources, where every source is the intrinsic spike train s^i of a neuron. Since there are as many filters as neurons,

[†]The obtained filters are called MPDR beamformers, see Sec. 2.2.1.

the dimension of the filter output space is equal to the number of neurons, and therefore, the detection and classification problem can be considered as a complete BSS problem. However, it is not guaranteed that the maximal response of filter $\boldsymbol{f}^i$ to spikes from neuron j will be at a shift of 0, i.e., when the filter and the template overlap entirely. This leads to the following model for the rectified filter output:

$$I(y_t^i) = \sum_j (\boldsymbol{A})_{i,j}\, s^j_{t+\tau_{i,j}} \tag{4.5}$$

with $\boldsymbol{A}$ being the mixture matrix, and $\tau_{i,j}$ being the shifts between the maximum response of filter $\boldsymbol{f}^j$ to template $\boldsymbol{q}^i$; i.e.,

$$\begin{aligned}(\boldsymbol{A})_{i,j} &= \max_\tau \left\{\left(\boldsymbol{q}^i \star \boldsymbol{f}^j\right)_\tau\right\} \\ \tau_{i,j} &= \operatorname*{argmax}_\tau \left\{\left(\boldsymbol{q}^i \star \boldsymbol{f}^j\right)_\tau\right\}\end{aligned} \tag{4.6}$$

where $(\boldsymbol{A})_{i,i} = 1$ and $\tau_{i,i} = 0\ \forall i$ by construction. We want to reconstruct the sources s^i by solving the corresponding inverse problem:

$$s_t^i \approx z_t^i = \sum_j (\boldsymbol{W})_{i,j}\, I(y^j_{t-\tau_{j,i}}) \tag{4.7}$$

with $\boldsymbol{W} = \boldsymbol{A^{-1}}$. Here, the relation to ICA becomes clear, since this is a similar inverse problem ICA solves. In contrast to ICA, we do not have to estimate $\boldsymbol{W}$ and $\tau_{i,j}$ from the data, but can calculate them directly from the responses (i.e. cross-correlation functions) of all filters to all templates, as illustrated in Fig. 4.1.

Once the matrix $\boldsymbol{W}$ is applied to $I(\boldsymbol{y})$, all values which were zero in $I(\boldsymbol{y})$ are set to zero in $\boldsymbol{z}$ as well. All steps of these procedure are summarised under the term "Deconfusion". After Deconfusion the false responses of the filters to non-matching templates are suppressed (see Fig. 4.2). In principle, it is possible that the inverse problem in Eq. 4.7 is not exactly solvable, if the shifts are not consistent. Consistent shifts have to satisfy the following equation

$$\tau_{j_1,k} - \tau_{j_1,i} = \tau_{j_2,k} - \tau_{j_2,i} \qquad \forall i, j_1, j_2, k. \tag{4.8}$$

A derivation is given in Sec. 4.7.2. For arbitrary templates and data covariance structures, Eq. 4.8 can in principle be violated. However, with templates from real experiments we did not observe this to be a problem.

4.2.5 Spike detection and classification

In the final step, thresholding is applied to every row i of $\boldsymbol{Z}$, where $\boldsymbol{Z}_{i,t} := z_t^i$. Again, by construction we have only to consider positive peaks. All local maxima after a threshold crossing are identified as spiking times of neuron i. In this sense, spike detection and spike classification is performed simultaneously.

The threshold is set for each row of $\boldsymbol{Z}$ individually such that the total relative error of false negative and false positive detections is minimal. Amongst others, the threshold depends on the variance of the noise, on the Deconfusion output, and on the firing frequencies of the neurons. A detailed derivation is given in Sec. 4.7.3.

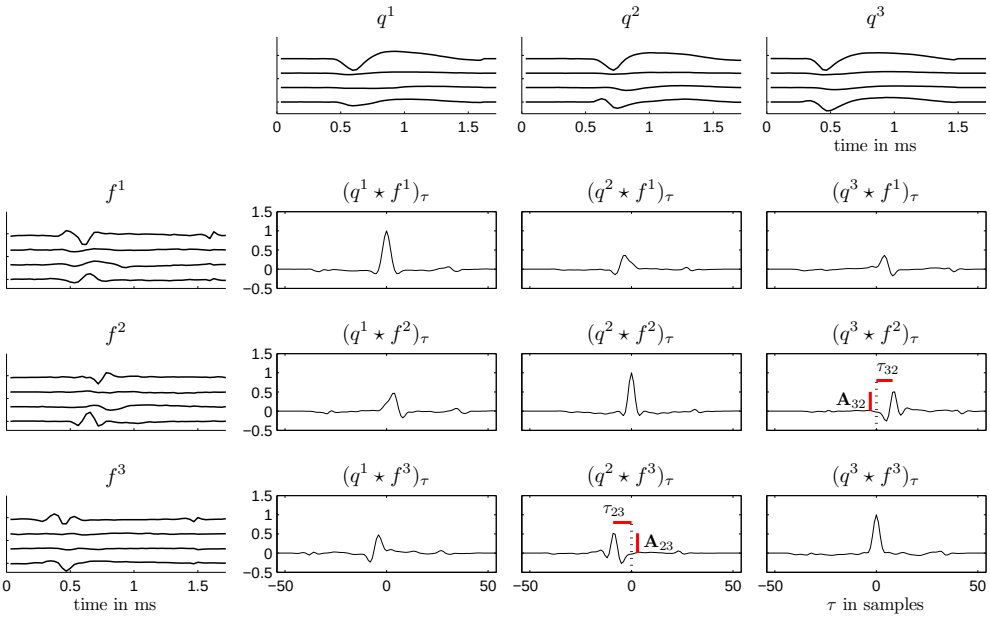

Figure 4.1: This figure illustrates the representation of the templates in the filter output space and the calculation of the Deconfusion parameters. In this example, three templates ($\boldsymbol{q}^1, \boldsymbol{q}^2, \boldsymbol{q}^3$, top row of the figure) originating from tetrode recordings are used. The corresponding linear filters are calculated by Eq. 4.3 and are shown on the left. The 9 plots show the responses of the linear filters to the templates, i.e. the cross-correlations $\boldsymbol{q}^i \star \boldsymbol{f}^j$, $i, j = 1, 2, 3$. The template $\boldsymbol{q}^i$ is now represented by the three vectors $\boldsymbol{q}^i \star \boldsymbol{f}^j$, $j = 1, 2, 3$. Although filter $\boldsymbol{f}^i$ has a maximum response of 1 to template $\boldsymbol{q}^i$, the filters do not provide an exact deconvolution, as the responses of filters $\boldsymbol{f}^{j \neq i}$ to template $\boldsymbol{q}^i$ are not equal to zero. However, since every template is represented on all filter output channels, the problem of extracting the signal from neuron i can be viewed as a source separation problem. The entry at position i, j of the mixing matrix $\boldsymbol{A}$ is given by the maximum peak value of $\boldsymbol{q}^i \star \boldsymbol{f}^j$; as an example $(\boldsymbol{A})_{2,3}$ and $(\boldsymbol{A})_{3,2}$ are shown. The shift indicates the position at which this maximum values occur; as an example the shifts $\tau_{2,3}$ and $\tau_{3,2}$ are shown.

4.2.6 Artifact detection

Artifacts were removed from our data in two ways. Firstly, all periods during which the animal had to perform a physical task (e.g., pressing a button) were not considered for further analysis. Secondly, for each period of length 10 ms the number of zero-crossings on each data channel was counted and summed up. All periods, in which this number was below 10% of the maximum number of possible zero crossings, were not considered for further analysis. This second type of heuristic removal aims at eliminating artifacts caused by oscillations of the electrode shaft inside the guiding tube (e.g., caused by movement of the animal).

4.2.7 Noise estimation

The noise covariance matrix $\boldsymbol{C}$ is determined by calculating the auto- and cross correlation functions of every channel. Only data points which were not part of any spike nor any artifact period, were used for the calculation. The noise covariance matrix is needed for the initialization phase, see Sec. 4.2.9, and for evaluation of the sorting result on real data, see Sec. 4.4.1.

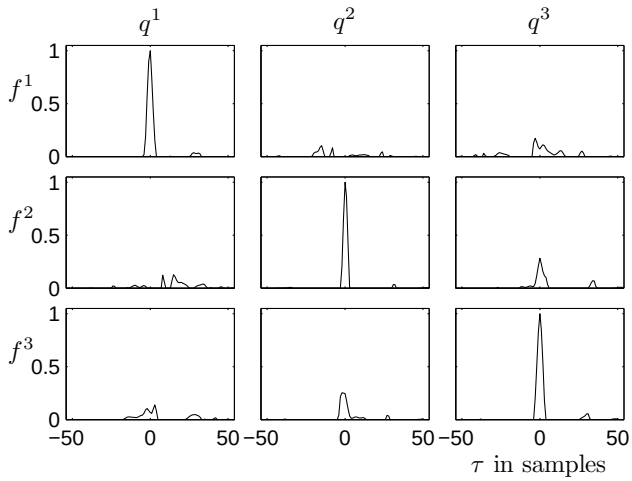

Figure 4.2: The figure shows the effect of Deconfusion on the filter outputs. The input for Deconfusion were the filter responses $\boldsymbol{q}^i \star \boldsymbol{f}^j$, $i, j = 1, 2, 3$ shown in Fig. 4.1. After Deconfusion the signal of neuron i is mainly present on the output channel i.

4.2.8 Adaptation

Due to tissue relaxations the measured waveforms change over time as the relative distance between the multi-electrode and the neurons change. In order to track these changes we re-estimate the templates as well as the data covariance matrix after every time period of length T. Each template $\boldsymbol{q}^i$ is re-estimated as the mean of the last 350 spikes detected from neuron i; whereas the spikes of neuron i are aligned on the maximal peak of the response of filter $\boldsymbol{f}^{i*}$. For the re-estimation only spikes which were classified by our method as non-overlapping spikes are used. The data covariance matrix is re-estimated from the last 30s of the recordings and the linear filters are re-calculated. Consequently, the Deconfusion and the thresholds are re-computed as well. In Sec. 4.4.1 we show that we can indeed track drifts with this approach.

Templates whose SNR decreases over time might be a concern. By constantly adapting the template, finally, there is a risk of getting a template which is very close to the noise signature, and the corresponding filter will detect pure noise. This can be prevented by removing filters at the appropriate moment. Consequently, we stop tracking templates whose SNR drops below 0.65. This value proved to be appropriate during simulations (see Sec. 4.4.1).

4.2.9 Initialisation phase

Most of the analysis done in the precedent sections was based on the assumption of known initial templates $\boldsymbol{q}^i$. Hence, before applying our method, one needs an initialisation phase during which the templates are found. In principle, any supervised or unsupervised learning method can be applied.

*The adaptation scheme proposed in Chap. 3 is not applied here, but rather a heuristic value for the number of spikes for averaging is used. This is due to the large number of filters required in tetrode recordings (see Sec. 4.4.1) and the resulting computational load. The adaptation scheme from Chap. 3, however, will be applied in the case of single electrode recordings in Chap. 5.

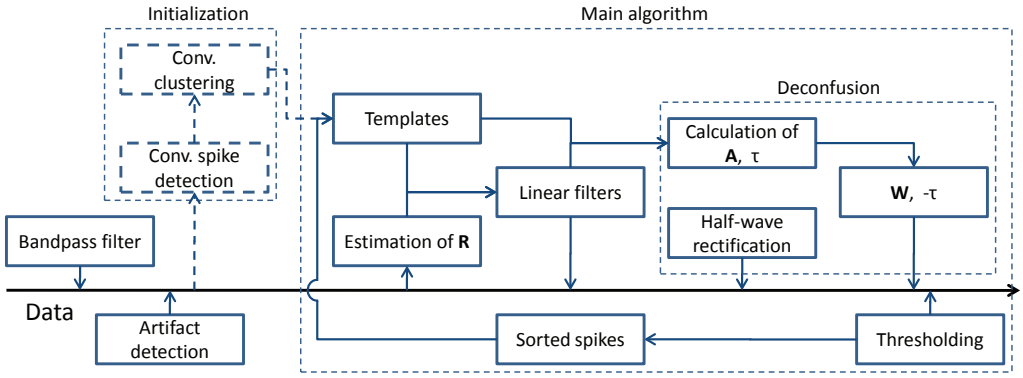

Figure 4.3: Schematic illustration of the way data are processed: The data are bandpass filtered and periods containing artifacts are excluded from further analysis (Sec. 4.2.6). During the initialisation phase a conventional spike detection and clustering method is used to determine initial templates (Sec. 4.2.9). The data covariance matrix $\boldsymbol{R}$ is estimated and for every template the corresponding linear filter is calculated as described in Sec. 4.2.2. The data are filtered and all values in the filter output below zero are set to zero (half-wave rectification). From all filter responses to all templates the un-mixing transformation is determined and applied to the processed data (Sec. 4.2.4). A threshold is applied to the Deconfusion output resulting in simultaneous spike detection and classification. The newly found spikes are used to re-estimated the templates. Also the covariance matrix of the data is re-calculated after regular time intervals (Sec. 4.2.8).

We want to emphasise that the initialisation phase is only necessary at the beginning of a recording session: Once the initial templates are estimated, the main algorithm runs online. Furthermore, because of the feedback described in Sec. 4.2.8, the initialisation does not have to be very accurate, as the templates are re-estimated after every period of length T. Usually we used an initialisation phase of about 30s in our real recordings (Sec. 4.3.3). This time window is short enough so that the templates change only very slightly in time and can, therefore, be clustered reliably, but long enough to acquire enough spikes to estimate robustly the mean waveforms.

Initial spike detection and initial spike alignment

During the initialization phase spike detection can be done with any conventional technique. We used an energy based approach, since it usually delivers a better performance than simple amplitude based approaches methods [144, 152]. In particular, we applied the MTEO detector (see [38] for definition) with k-values $[1, 3, 5]$ to each recording channel separately and set the threshold to 3.5 times the median of its output. Spike periods were defined as intervals of length 1.5ms, in which the output of the MTEO detector exceeded the threshold value at least once.

Correct spike alignment is crucial for a good clustering result. While in many studies an alignment based on the maximum and/or minimum peak value of a spike is used, again, methods based on the energy of a spike usually yield better results [55]. After cutting out all spikes around the peak of the detector, we used the following algorithm for alignment:

1. Calculate the average template over *all* spikes

2. Minimise the energy difference between every spike and the template by shifting the spikes

3. Repeat until convergence or a maximum number of iterations is reached

In our experiments described in Sec. 4.3.3 the average number of spikes in the first 30s of recordings is around 2500 and convergence is usually obtained after 15 to 20 iterations.

Initial clustering

Although a broad range of sophisticated clustering algorithms is available, we used a standard approach, since a very accurate initialization is not crucial for our method. The aligned spikes are whitened (e.g., see [168]) and projected into the space of the first 6 principle components. The clustering consists of a Gaussian mixture model in combination with the Expectation-Maximisation algorithm [236]. For every number of cluster means between 1 and 15 the clustering procedure is executed 3 times with random initial means. The covariance matrices are fixed to 2.5 times the identity matrix. The run and the number of means with the highest score according to the Bayesian inference criterion [236] are selected as initialisation for the main algorithm.

4.2.10 Signal-to-noise ratio (SNR)

The SNR is a scalar value which is an indicator for the difficulty of detecting a signal in noisy data. In this sense, the SNR definition should be dependent on the method used for signal detection. Several definitions of the SNR are used in the spike sorting literature. A very common one is to define the SNR by some maximal value, e.g., the maximum amplitude, the maximum difference in amplitudes (peak to peak distance), or the maximum of the absolute value of the amplitude, divided by the variance of noise σ^2, i.e.,

$$\mathrm{SNR}_p(\boldsymbol{q}) := \sqrt{\frac{\|\boldsymbol{q}\|_\infty^2}{\sigma^2}}$$

(e.g. see [38]). Another current definition for the SNR is based on the energy of a signal, i.e.,

$$\mathrm{SNR}_e(\boldsymbol{q}) := \sqrt{\frac{\|\boldsymbol{q}\|_2^2}{N \cdot T_f \cdot \sigma^2}}$$

(e.g. see [180]). We introduce a definition of SNR which is based on the Mahalanobis distance of a template $\boldsymbol{q}$ to zero:

$$\mathrm{SNR}_m(\boldsymbol{q}) := \sqrt{\frac{\boldsymbol{q}^\top \boldsymbol{C}^{-1} \boldsymbol{q}}{N \cdot T_f}}. \tag{4.9}$$

In the special case of single electrode data and of 1-dimensional templates ($T_f = 1$), all SNR definitions are equivalent. To show that SNR_m is an appropriate SNR definition for linear filters, while the other definitions are in contradiction with the meaning of signal-to-noise ratio, we simulated datasets containing a single neuron, which fired according to a Poisson statistic, and a noise covariance matrix $\boldsymbol{C}(\alpha) := (1-\alpha) \cdot \mathbf{1} + \alpha \cdot \frac{\boldsymbol{C}_{exp}}{\sigma^2}$, where $\mathbf{1}$ denotes the identity matrix, and $\boldsymbol{C}_{exp}$ is a noise covariance matrix from one of the experiments described in Sec. 4.3.1, with $\left(\boldsymbol{C}_{exp}\right)_{i,i} = \sigma^2$ for all i. The used template was

extracted from the same experiment. We simulated datasets for ten different α values between 0 and 1. The SNR_m decreased with increasing α, and consistently the detection performance of our method decreased, see Fig. 4.4. Note that $\mathrm{SNR}_p = \mathrm{SNR}_e = 1$ for all α values, which means that those definitions are inappropriate for the proposed method. Nevertheless, we always provide values for all three definitions of SNR in order to allow comparisons with other publications.

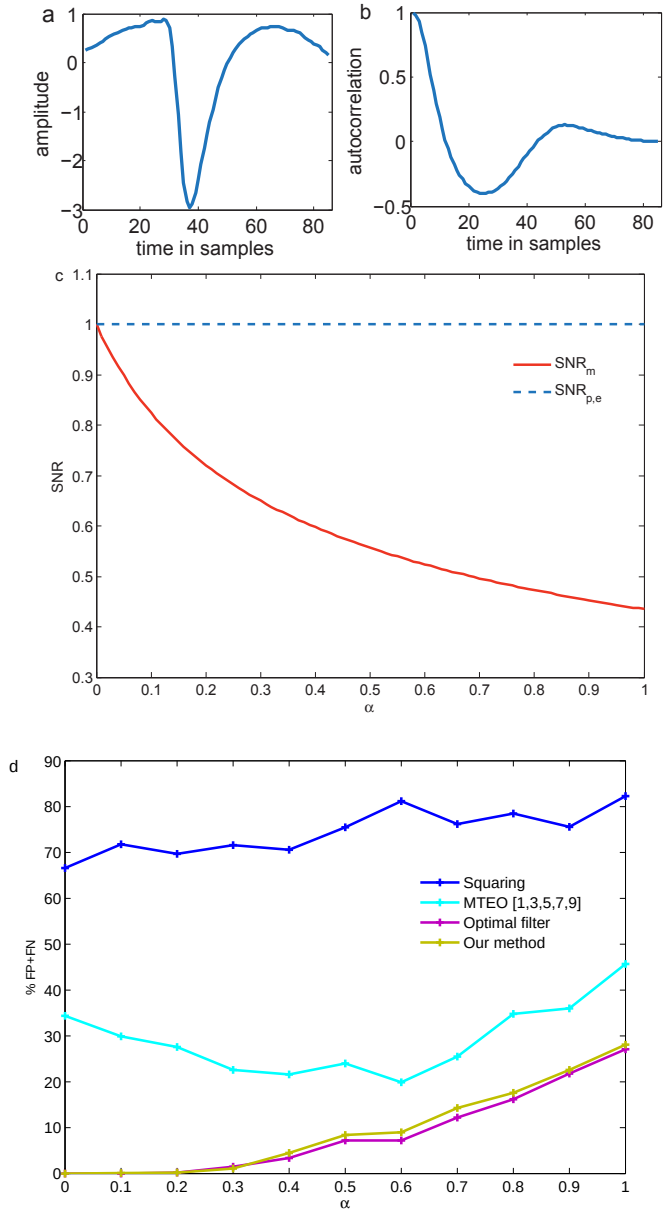

Figure 4.4: **(a)** Template $\boldsymbol{q}$ (in arbitrary units) used for the simulations. **(b)** Noise autocorrelation function of the same experiment from which the template was extracted. This autocorrelation was used to calculate $\boldsymbol{C}_{exp}$. **(c)** Plot of SNR_p ($\boldsymbol{q}$), of SNR_e ($\boldsymbol{q}$) and of SNR_m ($\boldsymbol{q}$) in dependence of α (see text for definition). **(d)** Average detection performance of different spike detection methods for different values of α. The method "squaring" consists of point wise squaring and thresholding, while "MTEO" is described in [38]. For each α value the average was done over 5 simulations, each with a noise covariance matrix $\boldsymbol{C}(\alpha)$ (see text for definition).

4.3 Experiments and datasets

For the performance evaluation of our method, three different datasets were used. All experiments were performed in accordance with German law for the protection of experimental animals, approved by the local authorities ("Regierungspräsidium"), and are in full compliance with the guidelines of the European Community (EUVD 86/609/EEC) for the care and use of laboratory animals.

4.3.1 Simultaneous intra/extra-cellular recordings

The experiments were done[†] in acute brain slices from Long Evans rats (P17 - P25). In every experiment a pyramidal cell from visual cortex, Layer 3 or 5 depending on the experiment, was simultaneously recorded intracellularly and extracellularly. Extracellular spike waveforms were recorded using a 4-core-Multifiber Electrode (Tetrode) from Thomas RECORDING GmbH, Germany. The cell was intracellularly stimulated by a current injection (varying from experiment to experiment between 80pA and 350pA). Extracellular recordings were sampled at 28kHz and filtered with a bandpass FIR filter (300Hz to 5000Hz).

The intracellularly recorded spikes were detected using a manually set threshold on the membrane potential. The threshold crossings in the membrane potential were used as triggers to cut out periods from the extracellular recordings (2ms before and 5ms after the trigger). In total, data were recorded from 6 different cells, which resulted in 9957 intracellularly detected spikes. For analysis only the recording channel with the highest SNR was considered. The SNR of the different experiments varied from $\mathrm{SNR}_m = 0.20$ ($\mathrm{SNR}_p = 0.79$, $\mathrm{SNR}_e = 0.39$) to $\mathrm{SNR}_m = 2.37$ ($\mathrm{SNR}_p = 7.09$, $\mathrm{SNR}_e = 3.64$).

4.3.2 Simulated data

Datasets with overlapping spikes

Dataset A1 The artificially generated data mimics a single channel recording of 15s length at a sample frequency of 32kHz containing activity from three neurons. Every simulation contained exactly 750 equidistantly distributed spikes of every neuron, which corresponds to a firing frequency of 50Hz. The three used templates were extracted from the recordings described in Sec. 4.3.1 and had a length of 2.1ms. The noise was generated by an ARMA model [81] approximating the noise characteristic shown in Fig. 4.4(b).

The relative number of overlapping spikes was systematically varied from 1% up to 50%. 75% of all overlapping spikes consist of overlaps between two templates (25% for each combination), and 25% of all overlapping spikes consist of overlaps between all three templates. The amount of overlap, i.e., how much the templates overlap, is distributed according to a uniform distribution on the interval $[1/3, 1]$. The SNR was kept constant for all overlapping ratios, namely, all three templates were scaled to an

[†]By Dr. Clemens Boucsein from the Institute for Biology III, Albert-Ludwigs-University, Freiburg, Germany.

equal SNR, which was $SNR_m = 1.2$. This corresponds to $SNR_p = 5.42$ and $SNR_e = 2.12$ (average values over the three templates).

Dataset B1 The second dataset contained activity from two neurons firing with a frequency of 10Hz each. The data were simulated with a sampling frequency of 25kHz and had a length of 50s. The noise was generated in the same way as for the dataset A1. The number of overlapping spikes was varied from 0% up to 60%, and the amount of overlap was drawn form a standard normal distribution. The two templates were scaled to equal height resulting in $SNR_p = 4.76$.

Datasets with SNR variation

Dataset A2 The SNR_m was systematically varied from 0.6 to 1.4 (which is equivalent to 2.71 to 6.32 average SNR_p and 1.06 to 2.48 average SNR_e). The amount of overlapping spikes was constant and set to 7%, which is approximately the overlap ratio resulting by chance under the assumption of independent spike trains. The three used templates and the way how noise was generated were the same as in dataset A1.

Dataset B2 Again, both templates had equal height, and the SNR_p was systematically varied from 2.78 up to 100. The amount of overlapping spikes was set to 1.6%, corresponding to chance probability.

The over-completeness, the equal SNR of all templates, and the presence of overlapping spikes make these datasets particularly challenging.

4.3.3 Acute recordings

Tetrodes were placed in ventral prefrontal cortex for individual recording sessions, sampling data from the same region across experiments. Recordings were performed[‡] simultaneously from up to 16 adjacent sites with an array of individually movable fiber micro-tetrodes [49]. Recording positions of individual tetrodes were manually chosen to maximise the recorded activity and the signal quality. Data were sampled at 32kHz and bandpass filtered between 0.5kHz and 10kHz.

Neuronal activity was recorded while 2 macaque monkeys performed a visual short-term memory task. The task required the monkeys to compare a test stimulus to a sample stimulus presented after a 3 second long delay and to decide by differential button press whether both stimuli were the same or not. Stimuli consisted of 20 different pictures of fruits and vegetables which were presented for 0.5s (test stimulus) or for 2s (sample stimulus). Correct responses were rewarded. Match and non-match trials were randomly presented with an equal probability. This experimental setup was presented in [234].

Approximately, the monkeys performed 2000 trials per session, which is equivalent to almost 4 hours of recording time. For the evaluation of our algorithm only the first 5 seconds of every trial were processed, as the remaining data might contain severe artifacts caused by the monkey's movement.

[‡]By Dr. Matthias Munk, Max Planck Institute for Biological Cybernetics, Tübingen, Germany.

Table 4.1: Average performance of the proposed method for non overlapping and overlapping spikes. Each column represents the true category of events detected as spikes (e.g. "N" meaning "noise", "AB" meaning an overlapping spike of template A and template B, etc.), while each row represents the category to which they were assigned by our algorithm. Each total number of classifications was divided by the number of corresponding spike events, resulting in a percentage value. The bold numbers represent the percentage of correct classifications. The table shows the average performance over 10 simulations with an overlap ratio of 40% (see Sec. 4.3.2). For a systematic evaluation over different overlap ratios the absolute numbers of the correct classifications were added and divided by the total number of inserted spikes; see Fig. 4.5

	N	A	B	C	AB	AC	BC	ABC
A	0.0	**96.0**	0.1	0.0	**91.7**	**93.5**	1.7	**92.0**
B	0.0	0.0	**98.2**	0.1	**87.4**	9.7	**92.8**	**87.2**
C	0.0	0.0	0.0	**97.8**	1.1	**92.0**	**92.1**	**88.7**

4.4 Results and discussion

The performance of a spike sorting method depends on its capability to detect spikes and to assign every spike to a putative neuron. As described in Sec. 4.2.5, our method achieves both simultaneously. We evaluated the performance of our approach thus as a combined detection and classification technique, and compare it against techniques commonly used.

4.4.1 Spike sorting performance

Resolution of overlapping spikes

We recall that the applied operations to the recorded data could be summarised in Eq. 4.7. The cross-correlation between the filters and the data is a linear operation. The following Deconfusion consists of a half-wave rectification, which is a non-linear operation, but affects only noise and not the action potentials (represented in the filter output), and the un-mixing, which is linear again. Hence, one can expect that if the superposition of spike waveforms is also linear, overlaps should be resolved successfully. We validated this assumption on the dataset A1 described in Sec. 4.3.2. The algorithm was executed in the same way as described in Sec 4.2. In order to allow the method to adapt (Sec. 4.2.8), the method was iterated 5 times on the same dataset. We also compared the performance of our method to those of two popular clustering based offline methods, one of them being the method described in Sec. 4.2.9, which will be abbreviated as "GMM". Since this is also the method which is used for initialisation of our algorithm, the comparison with GMM directly provides information about the improvements in sorting when our method is used.

The other algorithm, called "KlustaKwik", was explicitly developed for clustering neuronal data and was first introduced in [79]. The clustering parameters were set to their default values. Spike detection and alignment was done in the same way as described in Sec. 4.2.9. To provide an upper bound on the performance our approach could achieve, we included the evaluation with the optimal filters calculated directly from the real templates. Note that other existing, purely clustering-based sorting methods, either in the PCA space or in the original data space, would perform similarly to GMM and KlustaKwik.

Table 4.2: Same evaluation as in Tab. 4.1, but for the method "GMM" described in Sec. 4.2.9. The method sorts non overlapping spikes well, but has difficulties in resolving overlapping spikes

	N	A	B	C	AB	AC	BC	ABC
A	0	**81.0**	0.0	0.1	**27.8**	**27.3**	0.4	**21.5**
B	0.2	14.5	**100.0**	0.6	**68.0**	4.2	**45.0**	**42.7**
C	0.1	4.4	0.1	**99.4**	4.7	**69.0**	**53.2**	**41.7**

For the evaluation the relative number of TP was counted (Tab. 4.1, Tab. 4.2). The simulations show that our method indeed resolves overlapping spikes and outperforms the clustering based methods; see Fig. 4.5. Our method works even for datasets with a large amount of overlapping spikes, and the performance is close to the theoretical bound of this approach. On the other hand, the performance of the purely clustering based methods rapidly decreases with an increasing amount of overlapping spikes. Overlapping spikes are mostly detected as single events by conventional spike detection techniques, which leads to a high FN rate. Furthermore, since the waveforms of overlapping spikes are distorted, their distances to the corresponding cluster means are large, making it difficult to assign them to a neuron. This results in a low TP score for clustering based methods.

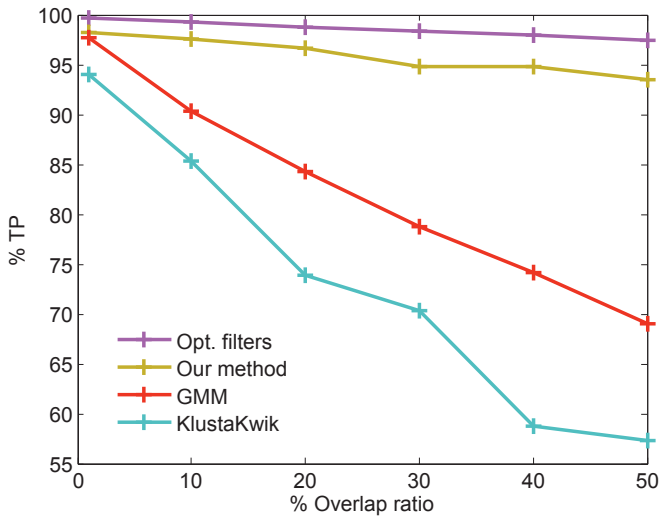

Figure 4.5: Average performance of the different spike sorting methods over 10 simulations. The x-axis indicates the overlap ratio, i.e. the relative number of overlapping spikes (see Sec. 4.3.2) while the y-axis represents the correct classifications in percentage (true positives divided by total number of spikes).

On the dataset B1 we compared our method with two other existing approaches, namely "OSort" and "WaveClus" presented in [180] and in [171] respectively. "OSort" is an online sorting algorithm based on Euclidean distance classification of pre-whitened spikes. Spikes are detected by thresholding the local energy of the data and pre-whitened using an estimated noise covariance matrix. The euclidean distance to each mean waveform form every cluster is calculated, and either the spike is assigned to one of the clusters or a new cluster is created. Since the mean waveforms are updated after every spike assignment, at the end, it is tested whether the distances between all the clusters

are still large enough, and merged if necessary.

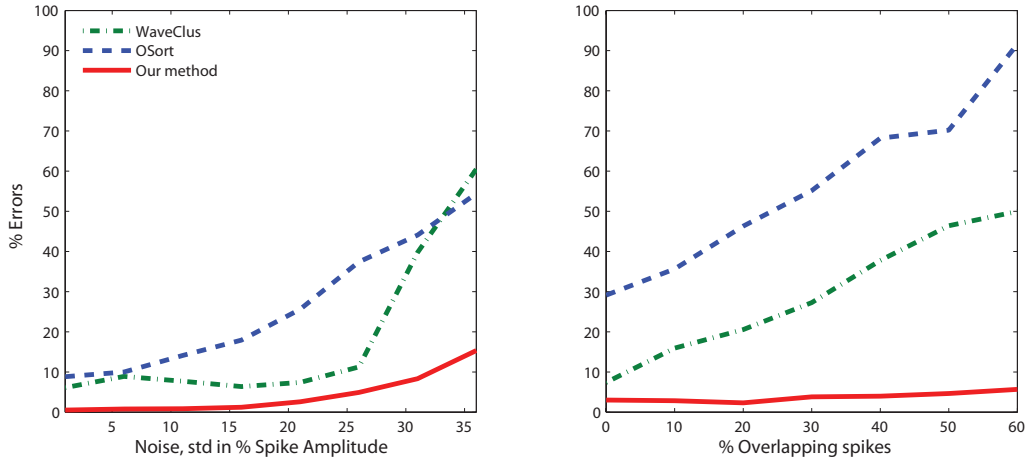

Figure 4.6: Average performances over 10 simulations of different spike sorting methods. The error is defined as the sum of false positive detections and false negative detections times 100, divided by the total number of inserted spikes. *Left:* Performance for different noise levels. The noise level is varied by changing the noise standard deviation with respect to the maximal height of the template. *Right:* Performance for different amounts of overlapping spikes; the noise level was set to 21%.

”WaveClus” is an offline method based on wavelet feature extraction and superparamagnetic clustering. Spikes are detected by positive thresholding and features are extracted by wavelet decomposition. In particular, the Haar wavelets are applied, and the 10 most discriminative coefficients are used for further processing, whereas discriminative power is determined by a statistical test for normality. Finally, the superparamagnetic clustering algorithm with an automatically selected temperature is used for clustering.

The parameters for the mentioned competing algorithm were set according to their reference. The sorting results are shown in Fig. 4.6, right, and confirm the findings on dataset A1.

Performance for various SNR

The evaluation on the dataset with a varying SNR (see Sec. 4.3.2) was done in the same way as in the previous section. The results for dataset A2 are shown in Fig. 4.7, whereas the results for dataset B2 are shown in Fig. 4.6, left. The performance of the clustering based methods is severely affected by a low SNR. The performance of the proposed method follows the one of the GMM algorithm, since it relies on its output for initialisation. Nevertheless, our method is always superior to it. Because of the rapid decrease in performance from a SNR level of 0.7 to an SNR level of 0.6, we stop the algorithm from detecting spikes from neurons with a SNR lower than 0.65 in real recordings by deleting the corresponding templates and filters. In contrast, the optimal filter method is only slightly affected by a low SNR level, indicating that a more elaborate initialisation would increase the performance of the proposed method on datasets with very low SNRs.

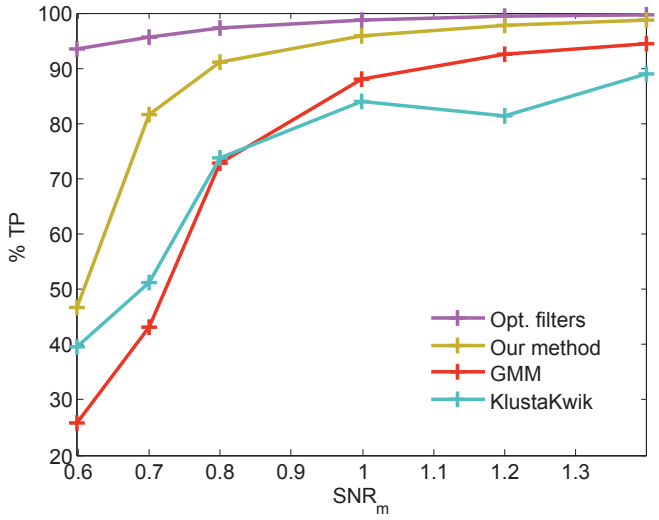

Figure 4.7: Average performance of the different spike sorting methods over 10 simulations with respect to various SNR levels. Note that the performance of the proposed method degrades with the performance of the GMM algorithm. This is because the output of the GMM is used as the initialisation for our method. However, our method is always superior to it. Low SNRs do not severely affect the performance of the optimal filter.

Performance on experimental data

We applied our method to data recorded in the prefrontal cortex of monkeys performing a short-term memory task as described in Sec. 4.3.3. For illustrative purposes, we show the results obtained by processing data from one tetrode, since the qualitative outcomes from processing other tetrodes and different recording sessions are similar.

For the initialisation phase we used the first 7 trials of the recording. The initial spike detection and clustering was done as described in Sec. 4.2.9, resulting in a total of 3219 detected spikes, which were assigned to 8 clusters. This basic clustering was used as an initialisation for the main algorithm, which was executed in the same way and with the same parameters as described in Sec. 4.2 (see also Fig. 4.3 for a summary). The 7 trials used for initialisation were also processed with the main method in order to improve the sorting quality.

The templates after the first 90 trials are shown in Fig. 4.8, and seem to be reasonable by visual inspection. In total, our method found almost 200000 spikes (57111, 18060, 50724, 51709, 3974, 7057, 444, 10915 for each template). Two well-established tests to quantitatively assess the sorting quality of a method performing on real data are the inter spike interval distribution and the projection test [180, 168]; the evaluation of our sorting with both tests is shown in Fig. 4.8. The relative number of spikes during the first 3ms is smaller than 1.5% for all neurons, implying that the refractory period is respected. On the other hand, the projection test verifies that the spikes of a single neuron have not been artificially split by the sorting algorithm into multiple clusters or that spikes from multiple neurons are assigned to the same cluster. The sorting of our method also passes the projection test since the cluster distributions do not overlap and are close to the theoretical prediction of a normal distribution with a variance of 1. In sum, the good results of these two tests imply that the found clusters are well separated and indeed

correspond to single neurons, as well as that the assumptions made in Sec. 4.2.1 are justified.

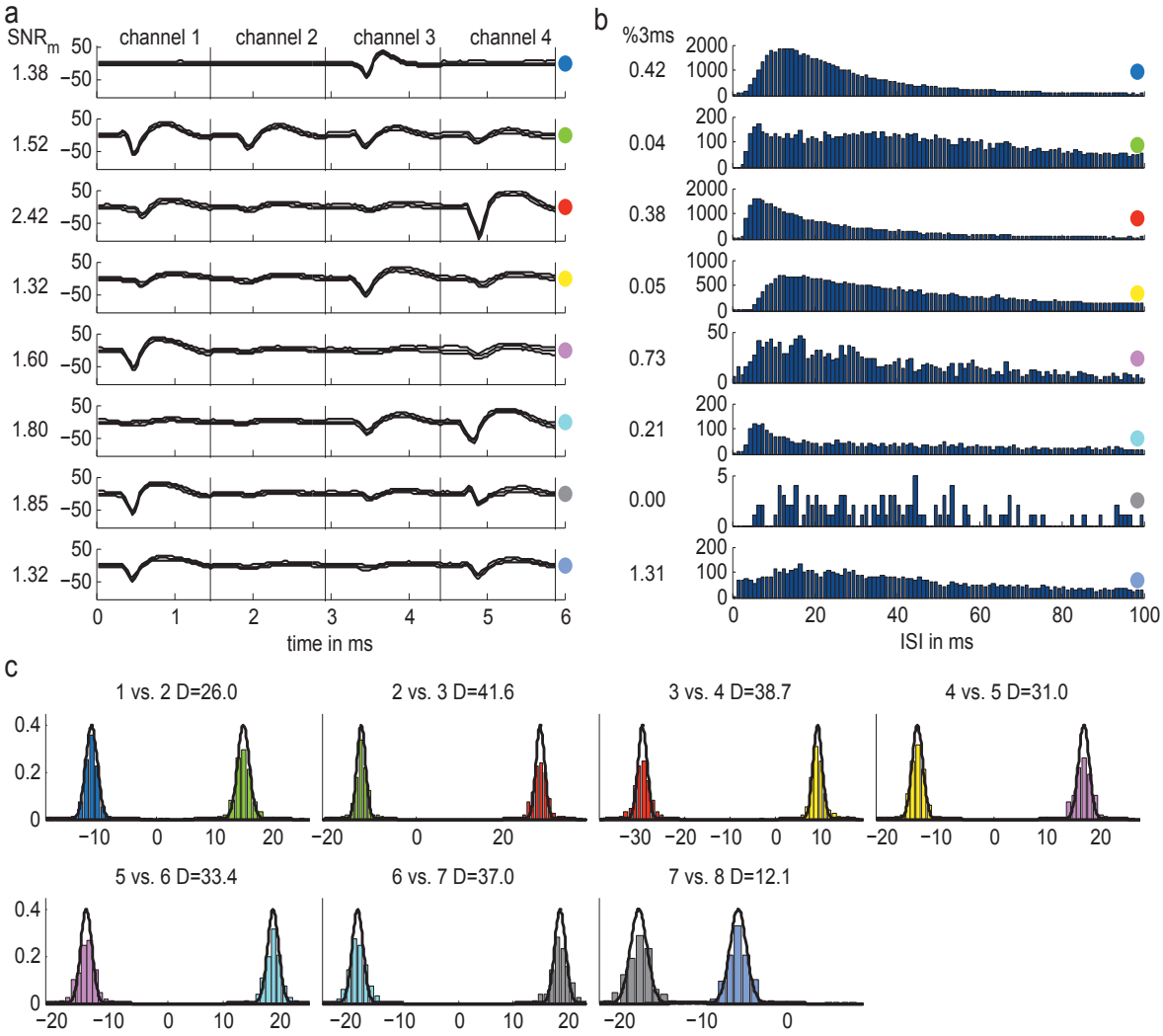

Figure 4.8: **(a)** Plot of the concatenated templates and their standard deviation. For the averaging all detected spikes from trial 50 to trial 90 were used. The vertical lines indicate the concatenation points of the individual tetrode channels, while the colored dots on the right serve as a label. On the left, the SNR_m value is shown, the channel dimension of the template being $T_f = 47$ and $N = 4$. The corresponding SNR_p values are $(10.06, 13.28, 21.82, 11.57, 13.12, 13.32, 14.27, 10.34)$, and the SNR_e values are $(1.84, 3.73, 4.22, 2.91, 2.90, 3.45, 2.99, 2.53)$, respectively. **(b)** Histograms of the inter-spike interval distributions with a bin size of 1ms. The numbers on the left indicate the percentage of spikes with an inter-spike interval of less than 3ms. **(c)** Projection test of the found clusters. The fit (solid line) represents a Gaussian distribution whose mean is the corresponding template and with variance 1. The D value indicates the distance in standard deviations between the means. Note that in the case of acute recordings, the waveforms change over time and thus the projection test is only meaningful for short time intervals. For the projection test the same spikes as in **(a)** were used.

Since we inserted the tetrodes before every experiment anew, our algorithm has to deal with the variability in the data caused by tissue drifts. The adaption procedure described in Sec. 4.2.8 was executed after every trial and adapted the algorithm correspondingly. The time period over which the templates were assumed to be constant was set to $T = 5\text{s}$ *. As a result, 2 neurons could be tracked from the beginning to the

*The value of T was set to 5s just for convenience of implementation, since the first 5s of each trial

very end of the experiment. The other templates were deleted earlier, since their SNR_m dropped below 0.65.

The disappearance of neurons from the recording volume is a common phenomenon in our recordings. However, the opposite, i.e., the appearance of new neurons during recordings, is rarely observed. This might be explained by the fact, that at the beginning of the experiments, the tetrodes are explicitly placed at a position where a lot of neuronal activity is measured. Therefore, it is more probable that during the tissue drifts the high activity population of neurons disappears than that new, highly active neurons appear. We discuss this problem also in Sec. 4.4.3.

The evaluation in Fig. 4.8 shows that the clustered spikes, although whitened, are not perfectly Gaussian distributed. This deviation is caused by overlapping spikes, but it is also due to an intrinsic waveform variability, as it is observed for example during bursts [56]. In this sense, the generative model assumed in Sec. 4.2.1 is not strictly valid anymore. Nevertheless, our method achieves a good performance, even for datasets containing bursting neurons identified by visual inspection. This can be explained by the fact that the scaling of the waveform during burst is close to linear [180]. Because of the linear character of our method (e.g. see Sec. 4.4.1), the response to a linearly scaled waveform will also only be scaled by the same factor. Hence, the algorithm classifies spikes from bursting neurons correctly as long as the amplitude degradation of the spikes is not too strong.

4.4.2 Limitations of our method

We have shown that our method is of great potential for spike detection and classification applications. However, there is a principle limitation: Since the filtering and the Deconfusion are linear operations, it is impossible to discriminate waveforms which are strictly linear dependent, i.e., when the spike waveform of one neuron is a multiple of the waveform of another neuron. A possible way to solve this problem is to sort the templates according to their SNR. Spikes with the highest SNR are detected first. Whenever a spike is found, the corresponding template is subtracted from the data and all other filter outputs are re-calculated for the affected period. This procedure is repeated for templates with a lower SNR. Further, if the sum of the waveforms of two different neurons with a certain shift is nearly identical to another neurons spike waveform, it is impossible to judge whether a spike is an overlap or not. Only probabilistic methods or soft clustering could give a hint at where the waveform came from.

4.4.3 Newly appearing neurons

We have not addressed the problem of neurons which are not detected during the initialization phase. As we observe spikes from neurons whose SNR decreases due to tissue drifts, and finally disappear completely from the recorded data, the opposite might also happen; i.e., neurons, previously undetected, slowly appear in the recording volume. A possible solution would be to run a conventional spike detection method in parallel to our method. All spikes detected by the conventional spike detection technique, but not by

were processed.

our method, could be collected, aligned and clustered. Respecting the newly found clusters, corresponding filters could be initialised and the Deconfusion procedure adapted accordingly.

4.4.4 Implementation and computational complexity

Especially for a real-time implementation the runtime of an algorithm is crucial. After the initialisation phase, the proposed method consists mainly of linear operations. The adaptation of the covariance matrix, of the templates and of the Deconfusion parameters need only to be computed every few seconds. Therefore, the computational burden lies in the application of the linear filters and the Deconfusion to a new sample of recorded (multi-channel) data.

Parallel computing

It is important to note that the cross-correlation for every filter - even for every channel of every filter - are independent of each other and can, thus, be computed in parallel as simple vector-matrix multiplications. For a so called vector processor such a multiplication would be one single operation only or could be implemented efficiently on a modern consumer computer-graphics hardware or on programmable digital signal processors.

4.5 Discussion and related literature

As the last survey paper [110] is already outdated, we present in this section a short summary of some existing approaches to spike sorting, and point out the differences to our method. In most cases, spike sorting is seen as a three stage process consisting of spike detection, feature extraction, and clustering. In this sense any combination of any detection, extraction and clustering algorithm is possible, which explains the extensive spike sorting literature. The main drawback of most approaches is, however, that there is no feedback incorporating the gained knowledge. This means that the clustering step is the final step, and the gained information about waveform shape and noise statistics is not used to improve the spike detection, feature extraction and finally again the clustering. In contrast, in our approach any existing spike sorting algorithm can be used as initialisation, and the obtained information is used to construct filters which not only offer improved detection performance, but also allow for drift adaptation and overlap resolution.

The focus of the selected literature in this chapter lies on feature extraction and clustering procedures, as spike detection methods will be discussed in Chap. 5. Most of the published spike sorting algorithms indeed apply the just mentioned three stage process and will be discussed in Sec. 4.5.1. There are only few spike sorting methods relying on blind source separation (BSS), probably due to the fact that BSS is not a simple approach for spike sorting, as only few methods deal with the over-complete convolutive case. Some approaches will be discussed in Sec. 4.5.2. Moreover, in App. B.2 we list some more general studies on BSS and blind deconvolution as a starting point for developing spike sorting algorithm based on their principles in future work.

4.5.1 Spike sorting based on clustering

We begin the discussion with studies which also rely on template extraction, and then move to algorithms relying on other principles.

In [198] a supervised neural network, supervised template matching and supervised classification based on principle components (PC) are compared. Template matching is done by calculating the squared distance. The data/noise covariance matrix is not taken into account. The neural network achieves best performance followed by template matching. For the PCA method only the first 2 PC were used.

On the other hand, in [8] the data are whitened first, and the euclidean distance to templates is used for classification. The templates are not learnt blindly, but given a priori (supervised method). The method is non-adaptive, and overlapping spikes are only resolved when not more than two waveforms superimpose.

Also in [36] it is assumed that the templates are already estimated. The method yields an improvement if the different neuron clusters have different variances. The variance of a cluster is estimated by the Levenberg-Marquart algorithm, which is basically an iteration procedure for minimising a function which is a sum of squared non-linear functions. Once the variance estimate is done, spikes are assigned to a cluster by a modified Euclidean distance (which is not well motivated).

In the work [78] the templates are assumed to be known, and spikes are sorted by computing the variance of the output signal after filtering with the templates (in order to reduce the computational complexity). The method is intended for low power devices, such as implantable prosthetics.

In [220] detected spikes are projected into PC space. In this space the density is computed, and templates are then extracted from point regions of high density. Next, spike sorting is performed by template matching using the Chebyshev norm. Overlapping spikes are not processed directly, but must be first recognised as such and then resolved by a more complex sub-routine.

The authors of [172] present a method designed for non-convex clustering, i.e. when the data distributions are non-convex. A relation between the entropy and the Cauchy-Schwartz divergence is established. As such, the entropy between clusters is maximised while the entropy within a cluster is minimised. The method requires a training phase, and the number of clusters must be given. The problem of classifying overlaps is not mentioned.

The main focus of [9] is handling non-stationarities in the data, such as changing templates and noise characteristics. The data are split into many frames, in which the data are assumed to be stationary. In every frame a clustering based on a Gaussian mixture model is applied (local clustering). In order to combine the local clustering into a global one the theory of types is used. The algorithm allows for splitting and merging of clusters. On the other hand, this approach is not capable of online processing and overlaps are also not resolved.

In [231, 232] another spike sorting method is presented which is capable of dealing with non-stationarities. The classification is based on a Gaussian mixture model. In each new frame the Expectation-Maximisation algorithm is initialised by the clustering result of the previous frame. Model selection is not done by the BIC, but a more accurate ap-

proximation based on Laplace's approximation is used. There is also a routine dealing with the case when the number of clusters is different in two subsequent frames. In contrast to [9] the algorithm processes one frame after another; there is no global clustering. Still, the clustering can only take place after the data of a full frame is acquired, and spike overlaps are not handled (probably overlaps are classified as background by the uniform mixture component).

The method presented in [82] uses a hidden Markov model to represent spikes. In this way the refractory period is incorporated. In order to estimate the model parameters, such as transition probability, template, etc., the Baum-Welch algorithm is employed. Once these parameters are learnt, spikes are found and classified by applying the Viterbi algorithm. In principle, this method can resolve overlapping spikes. However, the computational load is enormous, making a real-time implementation infeasible (the algorithm is not even capable of online processing though). Moreover, a lot of hand-tuning is required (see section 2.4 in the paper), which classifies this algorithm rather as semi-supervised.

The proposed method in [32] is similar to the one in [180], in the sense that clusters are built online, merged and split. However, this is all done on wavelet coefficients and not on the original data as in [180]. Also the spike alignment is more sophisticated.

A good comparison of several approaches is found in [203], and the code is available online. However, the paper does not address the problem of overlapping spikes. The raw data are filtered with a bandpass FIR filter, and two different FIR filters are compared for this task. Spike detection is done by simple amplitude thresholding. For feature extraction PCA and wavelet coefficients were compared. Finally, 2 clustering techniques and two estimation techniques were compared: mixture of Normal distributions, mixture of Student‘s t-distributions, and 2 estimation techniques, namely Normal expectation maximisation and robust variational Bayes. The number of clusters is estimated with minimum message length, as it is argued that this method performs better than Akaike information criterion and Bayes information criterion. The algorithms are tested on simulated and real data. Robust variational Bayes, with wavelet features and Student‘s t-distribution mixture model performs best.

In the work of [16] spikes are detected by amplitude thresholding and projected into PC space. A hierarchical classification algorithm is used for clustering, whereas overlapping spikes are not handled.

The focus of [100] is on feature extraction and clustering. Feature extraction is done by projection pursuit maximising the negentropy, which leads to better results than PCA. Clustering is done by a Gaussian mixture model, whereas the number of components is determined visually. Overlapping spikes are not considered.

The algorithm presented in [167] is computationally expensive, but makes use of almost all available information; not only is the waveform information used, but also the temporal aspects, such as the inter spike interval distribution and altered waveforms due to bursting are considered as well. Specific models for all statistics are assumed, whereby their parameters have to be estimated from the data. This is done by constructing corresponding Markov chains and Monte Carlo simulations of them. The number of clusters has to be user specified, thus, the method is semi-automatic. Overlapping spikes

are not considered. A slightly modified version of the algorithm was published in [42].

The authors of [191] argue that a Gaussian distribution is not appropriate for modelling the variability of a waveform and instead a t-distribution should be used. An expectation-maximisation algorithm adapted for t-distributions is presented and used for clustering. Overlapping spikes are treated as outliers or noise.

The paper [87] proposes to perform spike sorting in the wavelet coefficient space. However, the most discriminative wavelets are chosen by hand, and no method is presented how to do this automatically.

In [55] the found spikes are initially clustered, whereas the number of clusters is set to a value around 10 times larger than the number of estimated neurons. Then, an aggregation process merges the clusters based on the interface energy and inter spike interval times. No online scheme is presented nor are overlapping spikes considered.

It is argued in [241] that the derivative of the spike waveform in the frequency domain is better suited for distinguishing spikes form different neurons than the original waveform. The overall spike sorting algorithm is as follows: Spike detection with NEO detector (see [145]), pre-processing with a frequency shaping filter (in this case the derivative), feature extraction via PCA, and clustering with the mean shift clustering algorithm. Similar to [168] a measure is defined which indicates the sorting performance. This measure relies on the earth mover distance.

In [4] a method is presented which is based on spike trajectories (i.e. derivatives of the signal), i.e. a spike is assigned to the class to which it has minimum distance in phase space. The method does not resolve overlapping spikes, but it is automatic and online (however, it needs a learning phase).

A neural network approach is used in [33], which needs human supervision for learning. All overlaps must be also trained with the network, meaning that when many neurons are present and when not only overlaps between pairs of neurons are considered, this approach is computationally very expensive and contains a lot of human dependent parameters such as the number of hidden layer, appropriate summation function, etc.

4.5.2 Spike sorting based on source separation

An offline algorithm is presented in [200], which combines clustering with ICA. Firstly, all regular spikes are clustered with the *k*-means algorithm, whereas the number of clusters is around twice the number of neurons. Each cluster is then decomposed using the FastICA algorithm. The obtained independent basis vectors are compared between the clusters, and clusters with similar basis vector are merged (i.e. only the spatial waveform, not the temporal waveform is used). A similar procedure is applied to irregular spikes, which most probably represent overlapping spikes. However, some of the problems mentioned in Sec. 4.1 for the pure ICA approaches are also of concern in this method.

In [206] spikes are enhanced by filters which respond to discontinuities in the data. FastICA is then applied to the filter output in order to achieve spike sorting. However, the number of filters is not determined automatically. Also, spike detection methods based on transiency detection are not optimal as will be discussed in Chap. 5.

In the work of [86] a method based on array processing is presented, which is de-

rived from the field of radar applications (see Sec. 1.2.1). This methods works for multi-channel electrodes which are aligned parallel to an axon, as the algorithm relies on specific propagation velocities of spikes from individual neurons, and the resulting time delays when the spike is visible on a channel. The method is supervised, and does not consider blind beamforming approaches as proposed for example in [174, 39, 246].

The paper [121] claims to have solved the general convolutive MIMO source separation problem, assuming sparse sources. However, only a method in the case of two sensors is presented which works only in a high SNR setting.

As the neuroscience literature deals with algorithms based on blind source separation only to a limited extent, we investigate also some methods from the blind source separation literature, which, however, were not applied to neural data. In the blind source separation literature there exist approaches dealing with over-complete systems containing sparse and finite alphabet sources. However, they are often limited to the noise-free case (e.g. [65, 111]), deal only with instantaneous mixtures (e.g. [62, 34]), or work only for under-complete mixtures (e.g. [112]). In principle, a convolutive mixture can be transformed into an instantaneous one by applying the Fourier transform, however, this approach is not without problems [160] and will not be discussed further in this thesis. Nevertheless, as a pure BSS approach might improve the sorting quality (as it will for spike detection, see Chap. 5), it should be pursued in the future, hence in B.2 we give an overview over some existing BSS literature.

4.6 Conclusion and outlook

An automatic method for simultaneous spike detection and spike classification was presented, having several advantages which were demonstrated on various datasets. Explicitly, the method makes use of the additional information provided by multi-electrodes and has no constraints concerning the number of recording channels or the number of neurons present in the data. It resolves overlapping spikes instantaneously, performs well on datasets with a low SNR, and it adapts to non-stationarities present in the data. Moreover, the method operates online and is well suited for a real-time implementation.

In the first step of our algorithm, optimal linear filters were used to enhance the SNR. Linear filters account for the noise statistics as well as for the full, multi-channel template, and are, therefore, superior to other methods in detecting spikes of a specific neuron. Further, we used the output of the linear filters as a new representation of the data. The advantage of the filter output space is that its dimension is equal to the number of neurons, whereas this was not the case in the original data space. This allowed us to treat the spike sorting problem as a well defined source separation problem and solve it by Deconfusion.

In the final step, a channel specific threshold was applied providing simultaneous spike detection and classification. Unlike in many other methods, the thresholds need not to be set manually by a human supervisor but are determined automatically in an optimal way. The advantage of a combined spike detection and classification, in contrast to existing spike sorting methods, was demonstrated on simulated datasets. Especially in the presence of overlapping spike and low SNR, our method achieved better performances.

We showed that, in the case of linear filters, a proper definition of the signal-to-noise ratio is based on the Mahalanobis distance, whereas other commonly used definitions do not reflect the difficulty in detecting the signal.

By iteratively updating all quantities, namely the linear filters, the Deconfusion parameters, and the thresholds, the algorithm adopts to non-stationarities present in the data. As such, the method is also suitable for recordings made in acute experiments in which the multi-electrodes are inserted each time anew.

Two drawbacks of the proposed method were discussed, namely the incapability to detect newly appearing neurons and the problem of strictly linear dependent templates. However, for both problems a possible solution was sketched. The detailed study and realisation of these solutions is left for future studies.

By qualitative arguments, systematic runs on realistically simulated data and on real data from awake behaving macaques, we have shown that the algorithm is capable of resolving overlapping spikes; without additional computing time. However, for the acute recordings in awake behaving monkeys we cannot proof that the found solution is correct, since the ground truth is unknown. Only massive simultaneous intra- and extracellular recordings in vivo could be used to assess the quality of the sorting in real experiments. Due to technical limitations, such a dataset is currently not available.

The algorithm mainly consist of linear, independent operations, which can be executed in parallel and implemented in hardware. Therefore, the algorithm can be used for real-time implementations, making it an potential spike sorting method for brain-machine interfaces and for the execution of closed-loop experiments.

4.7 Derivations

4.7.1 Derivation of optimal linear filters

Filter $\boldsymbol{f}^i$ should respond with a peak to its matching template $\boldsymbol{q}^i$, but should have minimal response to the rest of the data. In particular, one demands that the response to the matching template is 1, i.e. ${\boldsymbol{q}^i}^\top \cdot \boldsymbol{f}^i = 1$. The response of the filter to the data is $\boldsymbol{X} \star \boldsymbol{f}^i$, where $\left(\boldsymbol{X} \star \boldsymbol{f}^i\right)_t = \sum_{k,\tau} x_{k,\tau+t} \cdot f^i_{k,\tau}$. Using the third assumption of Sec. 4.2.1 the response of a filter to $\boldsymbol{X}$ will be small (and therefore well distinguishable from the peak response of 1 to the matching template) if the variance of the filter output is small, i.e., one has to minimise $\mathrm{Var}\left(\boldsymbol{X} \star \boldsymbol{f}^i\right)$. In summary, the constrained minimisation problem is stated as

$$\boldsymbol{f}^i = \underset{\boldsymbol{f}^i}{\operatorname{argmin}} \left\{\mathrm{Var}\left(\boldsymbol{X} \star \boldsymbol{f}^i\right)\right\} \qquad \text{subject to } {\boldsymbol{q}^i}^\top \cdot \boldsymbol{f}^i = 1. \tag{4.10}$$

A short calculation (see Sec. 2.1) shows that

$$\mathrm{Var}\left(\boldsymbol{X} \star \boldsymbol{f}^i\right) = {\boldsymbol{f}^i}^\top \cdot \boldsymbol{R} \cdot \boldsymbol{f}^i. \tag{4.11}$$

Thus, the Lagrangian L of this minimisation problem is given by

$$L = {\boldsymbol{f}^i}^\top \cdot \boldsymbol{R} \cdot \boldsymbol{f}^i + \lambda\left({\boldsymbol{q}^i}^\top \cdot \boldsymbol{f}^i - 1\right) \tag{4.12}$$

where λ is the Lagrange multiplier. Since the objective function is convex in $\boldsymbol{f}^i$, there exists a single minimum, which can be found by solving $\nabla_{\boldsymbol{f}^i,\lambda} L = 0$. In fact, the minimum

is attained at

$$\boldsymbol{f^i} = \frac{\boldsymbol{R^{-1}q}^i}{\boldsymbol{q}^{i\top}\boldsymbol{R^{-1}q}^i}. \tag{4.13}$$

Often, linear filters are derived in the frequency domain instead, but linear filter defined in the time domain have several advantages, see [224].

4.7.2 Derivation of Deconfusion

$I(y_t^i)$ can be expressed as a linear combination of the sources s^j at shifts $\tau_{i,j}$:

$$I(y_t^i) = \sum_j (\boldsymbol{A})_{i,j}\, s^j_{t+\tau_{i,j}} \tag{4.14}$$

We show that

$$z_t^i := \sum_j (\boldsymbol{W})_{i,j}\, I(y^j_{t-\tau_{j,i}}) \tag{4.15}$$

with $\boldsymbol{W} = \boldsymbol{A}^{-1}$ is the corresponding inverse problem. By inserting the expression in Eq. 4.14 into Eq. 4.15 one obtains

$$\begin{aligned} z_t^i &= \sum_j (\boldsymbol{W})_{i,j} \sum_k (\boldsymbol{A})_{j,k}\, s^k_{t+\tau_{j,k}-\tau_{j,i}} \\ &= \sum_{j,k} (\boldsymbol{W})_{i,j}\, (\boldsymbol{A})_{j,k}\, s^k_{t+\tau_{j,k}-\tau_{j,i}} \\ &= \sum_j (\boldsymbol{W})_{i,j}\, (\boldsymbol{A})_{j,i}\, s^i_{t+\tau_{j,i}-\tau_{j,i}} + \sum_{j,k\neq i} (\boldsymbol{W})_{i,j}\, (\boldsymbol{A})_{j,k}\, s^k_{t+\tau_{j,k}-\tau_{j,i}} \\ &= s_t^i + \sum_{j,k\neq i} (\boldsymbol{W})_{i,j}\, (\boldsymbol{A})_{j,k}\, s^k_{t+\tau_{j,k}-\tau_{j,i}} \end{aligned} \tag{4.16}$$

Hence,

$$z_t^i = s_t^i \qquad \Longleftrightarrow \qquad \sum_{j,k\neq i} (\boldsymbol{W})_{i,j}\, (\boldsymbol{A})_{j,k}\, s^k_{t+\tau_{j,k}-\tau_{j,i}} = 0 \qquad \forall\, j,k,i \neq k \tag{4.17}$$

This is true, if

$$\tau_{j_1,k} - \tau_{j_1,i} = \tau_{j_2,k} - \tau_{j_2,i} \qquad \forall\, j_1, j_2, i, k. \tag{4.18}$$

Note that this condition is always satisfied for $k = i$.

4.7.3 Derivation of the optimal threshold

If we assume that the noise in the Deconfusion output is still a mixture of Gaussians (as an approximation for a mixture of truncated Gaussians), it follows for its variance

$$\begin{aligned}\sigma_k^2 &:= \mathrm{Var}\left(\boldsymbol{z}^k\right) \\ &= \mathrm{Var}\left(\sum_i w_{k,i} \cdot y^i_{\tau_{k,i}}\right) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M} \mathrm{Cov}\left(w_{k,i}y^i_{\tau_{k,i}}, w_{k,j}y^j_{\tau_{k,j}}\right) \\ &= \sum_{i=1}^{M} (w_{k,i})^2\,\mathrm{Var}\left(y^i_{\tau_{k,i}}\right) + 2\sum_{i=1}^{M}\sum_{j>i}^{M} w_{k,i}w_{k,j}\,\mathrm{Cov}\left(y^i_{\tau_{k,i}}, y^j_{\tau_{k,j}}\right) \\ &= \sum_{i=1}^{M} (w_{k,i})^2\,\boldsymbol{f}^{i\top}\boldsymbol{C}\boldsymbol{f}^i + 2\sum_{i=1}^{M}\sum_{j>i}^{M} w_{k,i}w_{k,j}\,\boldsymbol{f}^{i\top}\boldsymbol{C}_{|\tau_{k,j}-\tau_{k,i}|}\boldsymbol{f}^j\end{aligned} \tag{4.19}$$

where $\boldsymbol{C}_{|\tau_{k,j}-\tau_{k,i}|}$ are shifted covariance matrices, i.e. taking temporal correlations into account of order $T_f + |\tau_{k,j} - \tau_{k,i}|$.

The optimal threshold for the detection and classification of spikes from neuron k is chosen such that the overlap between the distribution of the spikes from neuron k and the distribution of the other spikes (from neurons j, $j = 1, ..., M$, $j \neq k$) is minimal. We assume the distributions to be Gaussian, with means $\mu_{k,j}$ and variance ${\sigma_j}^2$. The $\mu_{k,j}$ are given by the maximal response values of filter j to template k after Deconfusion, i.e.

$$\mu_{k,j} = \max_{\tau}\left\{\left(\boldsymbol{W}\cdot I\left(\left(\boldsymbol{q}^1 \star \boldsymbol{f}^j \quad \ldots \quad \boldsymbol{q}^M \star \boldsymbol{f}^j\right)^\top\right)\right)_{k,\tau}\right\} \tag{4.20}$$

whereas the variance is given by Eq. 4.19. One has only to consider the maximal false response and not the whole response, because the refractory period is in general longer than the length of the template. Thus the optimal threshold γ_k is given by

$$\gamma_k = \underset{\gamma_k}{\mathrm{argmin}}\left\{\beta_k\left(1 - \frac{1}{2}\,\mathrm{erfc}\left(\frac{\gamma_k - 1}{\sqrt{2}\sigma_k}\right)\right) + \sum_{j\neq k}\frac{\beta_j}{2}\,\mathrm{erfc}\left(\frac{\gamma_k - \mu_{k,j}}{\sqrt{2}\sigma_j}\right)\right\} \tag{4.21}$$

where $erfc$ denotes the complementary error function, and β_j is a normalised weight. Namely, if one wants to have an equally good detection performance as false alarm performance, then the weights should be chosen as $\beta_k = 0.5$, and the β_j $j \neq k$ proportional to the firing frequency of neuron j such that $\sum_{j\neq k}\beta_j = 0.5$. Note that the threshold must lie in the interval $[0, 1]$, hence this minimisation problem can be solved numerically with a line search algorithm, for example using the "fminbnd" command of MATLAB.

Chapter 5

Hybrid blind beamforming for spike detection

In this chapter we address the problem of spike detection, as defined in Eq. 1.3. We present a new spike detection algorithm which is based on methods from the field of blind equalisation and beamforming, and which is particularly adapted to the specific signal structure neuronal data exhibit. In contrast to existing approaches, our method blindly estimates several waveforms directly from the data, selects automatically an appropriate detection threshold, and is also able to track neurons by filter adaptation. The few parameters of the algorithm are biologically motivated, thus, easy to set. We compare our method with current state-of-the-art spike detection algorithms, and show that the proposed method achieves favourable results. Realistically simulated data, as well as data acquired from simultaneous intra/extra-cellular recording in rat slices are used as evaluation datasets.

5.1 Introduction

Extracellular recordings with electrodes constitute one of the main techniques for acquiring data from the central nervous system in order to study the neuronal code. One of the first processing stages of the recorded data, hence, consist of identifying the occurrence times of these spikes. To this end, various spike detection algorithms have been developed. To give a structured overview of the recent development in this field, we use a categorisation scheme based on the working principle of the methods. Note, that although the spike detection stage is one of the earliest, basically all algorithm require already some pre-processing. This includes a band pass filtering (usually between 0.5kHz and 10kHz), and a zero mean normalisation. In the following, we will still refer to this kind of pre-processed data as "raw" data, since all techniques rely on this initial step.

The first category of spike detection methods assumes that the spikes exhibit a larger amplitude than noise fluctuations. Hence, spikes can be detected as data segments which amplitude cross a certain threshold value. In [152] three different variations of this detection paradigm were described, including maximum, minimum and absolute value

thresholding. Other related approaches rely on the distance between the minimum and maximum value within a certain time frame [124], or temporally hierarchical maximum and minimum value thresholding [18].

The principle of the second category is based on the transient nature of a spike, thus, spikes can be detected by measuring some quantity describing the discontinuity of data. An example is the nonlinear energy operator which takes into account instantaneous energy and frequency, and which was used for spike detection in [145]. Further adaptations of this method to neural data have been proposed in [37, 38]. On the other hand, the approach in [158] considers only the instantaneous energy difference, while the proposed method in [144] calculates the derivative of a temporally accumulated energy. Also based on the first derivative of the data are methods presented in [4, 206].

The algorithms falling into the third category rely on the fact that spikes from a specific neuron exhibit a characteristic waveform. The similarity between a data segment and a specified waveform decides whether the considered data segment contains a spike. When the actual waveform in the data is unknown, a generic approach can be used. For example in [101, 147] a biorthogonal respectively a coiflet mother wavelet is used, since they exhibit a certain similarity in shape to waveforms found in some real recordings, and a spike is said to be detected when a specific function of wavelet coefficients exceeds a threshold value. In contrast, unsupervised estimation (also called blind estimation) of the waveform or blind equalisation has been performed in [41] by linear prediction, in [102] by automatic threshold setting, or in [187, 188] by using the cepstrum of bispectrum.

The choice which algorithm should be used in an application, surely depends on the two important aspects of computational complexity and detection performance. Limited power and computing resources, as encountered in implantable circuits [247], restrict applicable algorithm to have a very low computational load, hence mostly methods from the first category, and some few from the second one are used. When not limited by such constraints, it is favourable with respect to the detection performance to use algorithms belonging to the third category. This is motivated by the fact, that given the waveform and the noise covariance matrix, the matched filter, or equivalently the minimum variance distortionless response beamformer (MVDR)[†], is the optimal detector in case of Gaussian noise [210].

The aforementioned spike detection methods based on blind equalisation suffer from three main drawbacks. Firstly, they construct only a single filter. In many experimental situations, however, spikes from more than one neuron, having distinct waveforms, are present in the electrode recordings. The single filter either captures just one waveform, meaning that spikes from the other neurons will be detected poorly, or the filter is an average filter which will have a sub-optimal response to spikes from all the neurons. This problem aggravates the more neurons are present, and the more the waveforms are distinct, which is especially the case in multi-channel recording devices, such as tetrodes [75].

Secondly, few methods offer an automatic threshold selection mechanism, thus allowing for a truly unsupervised operation. The available approaches [227, 209, 31, 16] focus on the case when spike detection is done by amplitude thresholding (first cate-

[†]See Sec. 2.2.1 for more information about these and similar filters.

gory). For the above mentioned methods which rely on blind equalisation, none or only heuristic values are given regarding the choice of an appropriate threshold.

Thirdly, the mentioned methods are non-adaptive. Once a filter is calculated on a data segment in the time interval $[t, t+T]$, it is also applied to all subsequent data segments at times $\tau > t+T$. Particularly in acute recordings, the shape of the waveform will change over time [21], hence the performance of the filter will be sub-optimal if it is not adapted. One could re-estimate the template and re-calculate the filter after every time interval, however, this would increase the computational load significantly, and tracking of neurons would become difficult.

In this chapter, we propose a new spike detection algorithm which overcomes all those drawbacks. The algorithm is derived by considering the spike detection task as a blind equalisation problem in a multiple-input, single-output system. The algorithm consists of a two step procedure: In the first step, an iterative algorithm based on higher order statistics and deflation is used, which leads to an initial filter estimate. In the next step, the minimum variance distortionless responses (MVDR) beamformers are calculated, leading to an increased detection performance. This also allows to formulate a threshold selection algorithm as well as an effective adaptation scheme (see Fig.5.1 for a graphical representation of the whole algorithm). Because we use techniques from both fields, i.e. blind equalisation and classical beamforming, in the context of spike detection, we call our method hybrid blind beamforming for spike detection (*HBBSD*).

The rest of the chapter is organised as following: In Sec. 5.2 the algorithm and all its individual steps are described. The evaluation of its performance and comparison with existing spike detection methods are presented in Sec. 5.3. Conclusive remarks are given in Sec. 5.5.

5.2 Methods

5.2.1 Model of recorded data

In order to derive a well motivated algorithm avoiding heuristics as much as possible, the recorded data has to be described by some signal model. In the neuroscience community, it is widely accepted that the data x recorded at an electrode can often be represented as a linear sum of convolutions of the intrinsic spike trains s_i with constant waveforms $\boldsymbol{q}_i$ and colored Gaussian noise n (having a noise covariance matrix $\boldsymbol{C}$), see e.g. [181, 168]. Explicitly, it is

$$x(t) = \sum_{i=1}^{M} \sum_{\tau} q_i(\tau) s_i(t-\tau) + n(t), \tag{5.1}$$

where M is the number of neurons whose spikes are present in the recordings. For the sake of clarity, we restricted the model to single channel recordings, i.e. electrodes, but an extension to multi-channel data as provided by tetrodes is straightforward*.

Since the goal of spike detection is to recover the spike trains s_i from a linear time-invariant system without a priori knowledge about the shape of the waveforms $\boldsymbol{q}_i$, this

*Since $N = 1$, the index k in Eq. 1.1 can be omitted. For convenience the index i denoting the i-th source is now written as sub-script compared to Eq. 1.1 where it was a super-script.

can be viewed as a blind equalisation* problem. An overview about this topic and a survey of available methods dealing with such problems can be found in [35], for further literature see also Sec. 5.4 and App. B.2.

Most often, M, the number of sources, will be larger than the number of recording channels. In the model of a single electrode as described in Eq. 5.1, the number of recording channels is equal to one, in which case the generative system is referred to as multiple-input, single-output. In general, it is not possible to extract more sources than available recording channels [35]. In the following, we make explicit use of the unique properties of neural data, such as sparseness and binary alphabet, to overcome this restriction partially.

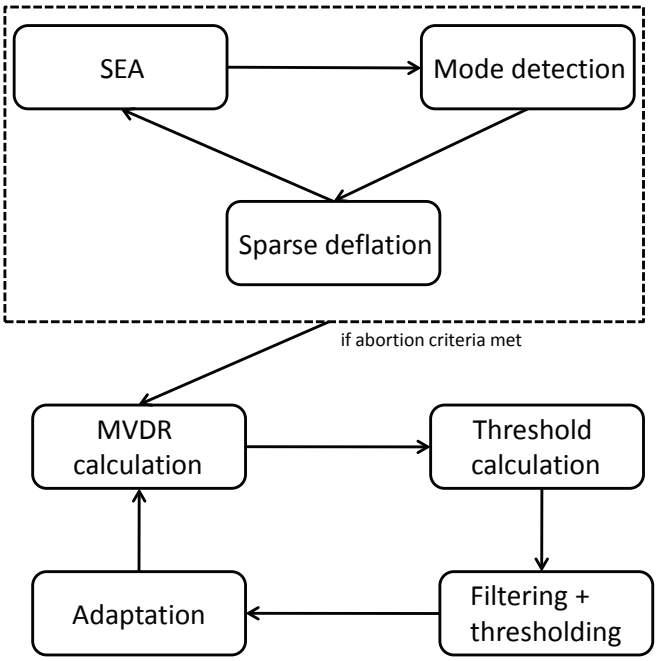

Figure 5.1: Schematic illustration of the proposed algorithm *HBBSD*. The algorithm starts with the super-exponential algorithm (SEA), and iterates between SEA, Mode detection and Sparse deflation repetitively, until certain abortion criteria described in Sec. 5.2.5 are met. Finally, the MVDR filters and the corresponding thresholds are calculated. Spike detection is done by thresholding the filter output and the newly detected spike are used to update the filters, allowing for neuron tracking.

5.2.2 Application of the super-exponential algorithm

The super-exponential algorithm (SEA) developed in [189] achieves blind equalisation via filter calculation by higher order cross cumulants. For real valued data, the filter $\boldsymbol{h}$ at iteration $k+1$ is computed as

$$\boldsymbol{h}^{(k+1)} = \frac{\boldsymbol{R}^{-1} \cdot \boldsymbol{d}^{(k)}}{\sqrt{\boldsymbol{d}^{(k)\top} \boldsymbol{R}^{-1} \cdot \boldsymbol{d}^{(k)}}} \tag{5.2}$$

*Often also called blind deconvolution, blind identification, or convolutive blind source separation.

where $\boldsymbol{R}$ is the data covariance matrix, $d^{(k)}(n)$ denotes the cross-cumulant[§] between p-times $y^{(k)}(t)$ and $x(t-n)$, and $y^{(k)}(t)$ is the filter output, i.e.:

$$\begin{aligned}(\boldsymbol{R})_{i,j} &= \mathrm{cov}(x(t-i), x(t-j)) \\ d^{(k)}(n) &= \mathrm{cum}(y^{(k)}(t) : p, x(t-n) : 1) \\ y^{(k)}(t) &= \sum_{\tau} h^{(k)}(\tau) x(t+\tau)\end{aligned} \tag{5.3}$$

The algorithm works when the signals s_i are non-Gaussian and when the $\boldsymbol{q}_i$ are stable[†]. In the context of neural recordings, both requirements are surely met. Firstly, the s_i represent the intrinsic spike trains, thus taking values of either 0 or 1, and whose probability density function follow most likely a sparse Bernoulli distribution, or their inter spike interval a Poisson distribution. Secondly, the waveforms $\boldsymbol{q}_i$ are finite impulse response filters, and hence are stable. The SEA algorithm is said to have reached convergence when the difference between two consecutive iterations is small enough (see also Sec. 5.3.3). For convenience, we call the filter obtained at the last iteration simply $\boldsymbol{h}$, instead of $\boldsymbol{h}^{(k_{last})}$.

The choice of the SEA instead of other blind equalisation algorithms was motivated by several of its features. It is shown that in the noise-free case, the algorithm converges independently of the initial condition to the globally optimal solution with a super-exponential convergence rate [189]. If one had access to an infinite amount of data, this property should also hold when Gaussian noise is present, as higher order cumulants are zero for Gaussian signals [189]. Moreover, the algorithm is not gradient based like Bussgang type algorithms, thus no step size selection is required, which reduces the amount of parameter settings for the user.

For neural data, we chose the order of the cumulant to be $p = 2$ or $p = 3$. In the former case, the vector $\boldsymbol{d}$ is proportional to a function of the skewness, a statistic which is well suited for asymmetric signals such as the s_i [155]. For $p = 3$, this makes the vector $\boldsymbol{d}$ proportional to a function of the kurtosis, which is a good statistics in case of sparse data following a model as in Eq. 5.1 [95, 88]. These findings were also confirmed in [116].

5.2.3 Mode detection in the SEA filter output

The SEA computes a single filter on the basis of a vector $\boldsymbol{d}$ which contains the statistics of all M waveforms. Nevertheless, as it is most likely that the characteristics of the neurons will be different with respect to signal-to-noise ratio, spiking frequency, or shape of waveform, it is expected that the filter will have various responses to the different neuronal waveforms. The idea is to identify spikes which belong to a single component and re-calculate the filter using only these spikes. The identification is done by a technique called mode finding [27].

Firstly, only the maximum values, denoted as m_i, of the filter output y within a certain range $2L_s + 1$ are extracted. Then, the probability density function pm of the m_i is

[§] See Sec. 1.4.2 for a short introduction to higher order statistics and used notation for cumulants.

[†] Stable in the sense of robust against noise, not in the sense of stationary in time.

estimated by a kernel density estimator, which in the assumed case of Gaussian noise is favourable to be a Gaussian kernel. The kernel bandwidth is chosen optimally depending on the amount of data [93]. The function *pm* will exhibit a high amplitude mode due to noise, and possibly several low amplitude modes caused by spikes [‡], see Fig.5.2. Hence, the second largest mode b_2 is the prominent spike mode, i.e. caused by spikes to which the filter responded the most, and which consequently should be extracted from the data first (see also Sec.5.2.3). All m_i which have a smaller distance to b_2 than to any other spike mode, and which are also larger than the first minimum separating the noise peak from the first spike mode, are considered to belong to b_2, see Fig.5.2. However, modes which are in the range of $\pm 2\hat{\sigma}_{n_h}$ around b_2 are not regarded as separate modes, whereas $\hat{\sigma}_{n_h}$ denotes the estimated standard deviation of the noise in the filter output (of filter $\boldsymbol{h}$) (see Sec.5.2.3).This is motivated by the fact that two Gaussian distributions with identical standard deviation do not exhibit two separate modes, unless their means are at least $2\sigma_{n_h}$ apart [179]. This merging of modes is necessary in order to minimise the number of spurious modes which do not represent an individual component but are mere artifacts caused by the kernel smoothing.

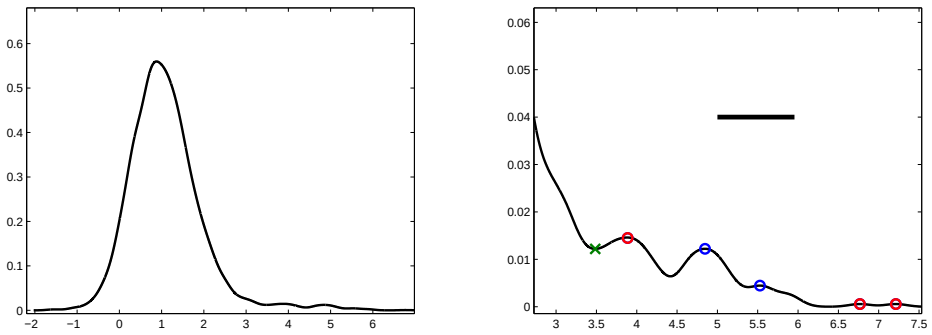

Figure 5.2: *Top:* Estimated probability density of the local maxima values m_i. The spareness of the data is clearly exhibited by the large noise peak (at around 1 on the x-axis) and some small spike peaks (at around 4 and 5 on the x-axis). *Bottom:* Zoom in on the spike modes. The circles indicate the local maxima of the modes that were found. The mode at around 3.9 was identified as largest (b_2), and the two modes indicated by blue circles are discarded, as they are within the range of $\pm 2\hat{\sigma}_{n_h}$. The estimated noise standard deviation $\hat{\sigma}_{n_h}$ is indicated by the thick bar. The green cross indicates the first local minimum, separating the noise peak from the spike modes.

Estimation of the filter output noise variance

To estimate σ_{n_h}, first the mean $\hat{\mu}_{n_h}$ of the filter output noise is estimated. If one can assume that n is zero mean, this step can be avoided, since then, it immediately follows that $\mu_{n_h} = 0$ as well. Otherwise, the probability density function of y is estimated by a Gaussian kernel density estimator as described in the previous section. Making again use of the sparseness of the data, the mean $\hat{\mu}_{n_h}$ is found as the global maximum of this probability density function.

[‡]Due to the large amount of noise samples, the kernel bandwidth will be relatively small, which guarantees that the modes caused by spikes will not be smoothed away.

As we expect that the response of filter $\boldsymbol{h}$ to spikes is larger than μ_{n_h}, we ignore all values of y which are above $\hat{\mu}_{n_h}$, since they are likely to contain spikes. Hence, $\hat{\sigma}_{n_h}$ is solely estimated on values of y which are smaller than $\hat{\mu}_{n_h}$.

Gaussianity of the modes

Strictly speaking, due to the maximum operation, the m_i do not follow a Gauss distribution anymore, but rather an extreme value distribution. Nevertheless, a Gaussian kernel is used for density estimation and the spike modes are assumed to be Gauss distributed as well. This is justified by the fact that the spike modes exhibit large amplitudes in the filter output, and thus their maxima values are still almost Gauss distributed even after a maximum operation.

Even when the noise in the original data is not perfectly Gauss distributed, after filtering it probably is due to the central limit theorem. Thus, it is justified to assume Gauss distributed noise in the filter output.

Largest spike mode finding

From the kernel density of the m_i, first a Gaussian distribution with mean $\hat{\mu}_{n_h}$ and standard deviation $\hat{\sigma}_{n_h}$ is subtracted (not shown in Fig.5.2). This removes the noise contribution to modes, and ensures that the largest spike mode b_2 is indeed the prominent one.

Note that in [102] also a mode detection procedure was applied. In contrast to our approach, it was done on a generic filter output consisting of squaring and lowpass filtering. Moreover, we merge modes based on their proximity in order to find all spikes belonging to the largest spike mode, whereas in [102] only the local minimum separating the noise mode from the spike mode is found and a single template is constructed.

5.2.4 Sparse deflation

In classical algorithms designed for multiple-input, multiple-output systems, sources are extracted one by one using a technique called deflation [92]. As such, one single waveform $\boldsymbol{q}_j$ is estimated via second order statistics, the source s_j is estimated via the convolution of the corresponding filter $\boldsymbol{h}_j$ with x, and the convolution between $\boldsymbol{q}_j$ and s_j is subtracted from the data x. This classical deflation procedure was developed by assuming that the sources are continuous signals, and that the waveforms have to be known only up to a scalar factor. In contrast, the signals representing the occurrences of spikes are discrete and sparse, and, as will be shown in Sec. 5.2.6, the waveforms need to be known without ambiguity.

Therefore, we propose an adapted deflation procedure which we call sparse deflation, as it relies on the sparseness of the data. At iteration j data segments $\boldsymbol{x}^{(j)}{}_i$ of length $2L_f + 1$ are cut out of x around the occurrence times $t_{m_i} + t_{\text{shift}}$ of the maxima m_i, $i = 1, ..., K$, which belong to mode b_2. The shift t_{shift} is determined so that the cut out data segments have maximum total energy. Without this step, extraction of different parts of the same waveform at several iterations would be possible, as the SEA filter does not

necessarily respond maximally at the middle of a waveform. Finally, the waveform is estimated as the median of all data segments[§], i.e.

$$\hat{q}_j(t) = \mathrm{med}\left\{x_1^{(j)}(t), \ldots, x_K^{(j)}(t)\right\} \qquad t = -L_f, ..., L_f, \tag{5.4}$$

where K is the total number of maxima m_i belonging to mode b_2. Instead of subtracting the estimated contribution of source s_j, the data segments $\boldsymbol{x}^{(j)}{}_i$ are simply removed from the data. The reduced data set $x \setminus \boldsymbol{x}^{(j)}{}_i$, $i = 1, ..., K$, is now used as the starting point for the next iteration of the algorithm. In particular, the steps described in Sec. 5.2.2 - 5.2.4 are repeated on the updated data $x \setminus \boldsymbol{x}^{(j)}{}_{i=1,...,K} =: x$.

5.2.5 Abortion criteria

The iteration loop is terminated if at least one of the following criteria is met:

- No spike mode can be identified in the filter output anymore, or the number of spikes belonging to the spike mode is below a relative threshold $minf$
- A maximum number of iterations is reached

If the loop abortion happens after the first iteration already, the filter obtained by Eq. 5.2 is used for further spike detection instead of the MVDR beamformers.

5.2.6 Calculation of the MVDR beamformers

Once the iteration loop described in the previous sections is completed, the final filters used for spike detection are calculated. Namely, we use the MVDR beamformer[¶] which is given by [210]

$$\boldsymbol{f}_i = \frac{\hat{\boldsymbol{C}}^{-1} \cdot \hat{\boldsymbol{q}}_i}{\hat{\boldsymbol{q}}_i^{\top} \cdot \hat{\boldsymbol{C}}^{-1} \cdot \hat{\boldsymbol{q}}_i}, \tag{5.5}$$

where $\hat{\boldsymbol{C}}$ is the estimate of the noise covariance matrix, and $\hat{\boldsymbol{q}}$ denotes the vectorial representation of the i-th estimated waveform, the individual entries being $\hat{\boldsymbol{q}}_{-T_f}, \ldots, \hat{\boldsymbol{q}}_{+T_f}$. The estimate of $\boldsymbol{C}$ is done after the last algorithm iteration, as the deflated data set $x \setminus \boldsymbol{x}^{(j=1,...,J)}{}_{i=1,...,K_J}$ contains far less spikes than the original data x allowing for a more accurate noise estimation.

5.2.7 Filtering and spike detection

After calculating the MVDR beamformers, the data are filtered with each of them, and a spike is declared as detected when the filter output z exceeds a certain threshold γ, i.e.

$$\begin{aligned} z_j(t) &= \sum_{\tau} f_j(\tau) x(t+\tau) \qquad \text{detection if } z_j(t) \geq \gamma_j \\ &=: (\boldsymbol{f} \star x)_t \,. \end{aligned} \tag{5.6}$$

[§]An even better performance could be achieved if the data segments were first upsampled, aligned, averaged and then downsampled [168].

[¶]Note that other filters could be used instead, e.g. adapted to a real-time detection task, as discussed in Chap. 2.

5.2.8 Threshold selection

The threshold for every filter is selected individually such that the probability of detection P_{D} is maximal (probability of a true positive detection), whereas the probability of false alarm P_{FA} (probability of a false positive detection) should be minimal. If one admits a certain tolerance Δ in the arrival time estimation, meaning that a spike is declared as correctly detected when the filter output exceeds the threshold somewhere in the interval $\left[t_{\mathrm{spike}} - \Delta, t_{\mathrm{spike}} + \Delta\right]$, the probability of detection for filter $\boldsymbol{f}_j$ given threshold γ_j is expressed as

$$P_{\mathrm{D}\,j}(\gamma_j) = 1 - \prod_{\tau=-\Delta}^{\Delta} P_{\mathrm{N}\,j}\left(\left(\boldsymbol{f}_j \star \bar{\boldsymbol{q}}_j\right)_\tau\right) \tag{5.7}$$

where $P_{\mathrm{N}\,j}(x) := 1/2 \cdot \left(1 + \mathrm{erf}\left(\frac{\gamma_j - x}{\sqrt{2}\sigma_j}\right)\right)$ with $\sigma_j = \sqrt{\boldsymbol{f}_j^\top \hat{\boldsymbol{C}} \boldsymbol{f}_j}$. Thus $P_{\mathrm{N}\,j}(x|_{x=(\boldsymbol{f}_j \star \bar{\boldsymbol{q}}_j)_\tau})$ is the probability that the spike is not detected at sample τ, whereas $\bar{\boldsymbol{q}}_j$ is defined in the next Sec. 5.2.9. Similarly, the probability that a noise segment of length $2\Delta + 1$ is falsely detected is given by

$$P_{\mathrm{FA}\,j}(\gamma_j) = 1 - \left(P_{\mathrm{N}\,j}(0)\right)^{2\Delta+1}. \tag{5.8}$$

An optimal detector would always achieve a perfect performance of $P_{\mathrm{D}} = 1$ and $P_{\mathrm{FA}} = 0$, thus any detector should have a performance as close as possible to the perfect performance[¶]. The optimal threshold, hence, is selected according to

$$\gamma_j = \underset{\gamma_j}{\mathrm{argmin}} \left\{ \left\| \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} P_{\mathrm{FA}\,j}(\gamma_j) \\ P_{\mathrm{D}\,j}(\gamma_j) \end{pmatrix} \right\|_2 \right\}. \tag{5.9}$$

This optimisation problem can be solved efficiently as it involves only a single parameter, namely the threshold γ_j, which should lie in the interval $[0, 1]$. In practice, we evaluate $P_{\mathrm{FA}\,j}$ and $P_{\mathrm{D}\,j}$ for all threshold values in $[0, 1]$ with a resolution of 0.0005, and select as optimal threshold the one which minimises Eq. 5.9[‡].

When the threshold is obtained by Eq. 5.9, it is assumed that detecting a spike is equally important as avoiding a false positive detection. However, with respect to subsequent analysis for understanding the working principles of the nervous system, it was shown, that not detecting a spike has more impact than declaring incorrectly a piece of noise as a spike [159]. This particular characteristic of neural data could be incorporated by introducing a weighting parameter in Eq. 5.9.

5.2.9 Adaptation to changing waveforms

In Eq. 5.1 we assumed that the waveforms $\boldsymbol{q}_i$ are constant in time, which is approximately true for short periods at the beginning of an experiment. Due to tissue relaxation, however, the distance between the electrode and the neurons changes, which leads to altered recorded waveforms [21]. In Chap. 3 we proposed an adaptation scheme for an

[¶]Note that in contrast to Sec. 2.1.2 we omit for simplicity the self-suppression term in the calculation of P_{FA}, and solely consider the false detections of noise samples.

[‡]Note, that the difference between Eq. 5.9 and the derivation in Sec. 4.7.3 are different norms. In Eq. 5.9 the $\|\cdot\|_p$, $p = 2$ norm was applied, whereas in Sec. 4.7.3 $p = 1$ was used.

estimated spatial waveform and the corresponding filter. This method was especially designed for sparse binary data such as neuronal data. Herein, we shortly summarise this method and extend it to multiple, temporal waveforms. In brief, after every time interval T, each waveform is updated as the mean of the K_{opt} last data chunks x of length $2L_f + 1$ which were detected as spikes, i.e.

$$\hat{\boldsymbol{q}}_j = 1/K_{\text{opt}\,j} \cdot \sum_{i=K_{\max j}-K_{\text{opt}\,j}+1}^{K_{\max}} \boldsymbol{r}_{j,i} \tag{5.10}$$

where $\boldsymbol{r}_{j,i} := \big(\boldsymbol{x}(t(i) - L_f), \quad \ldots, \quad \boldsymbol{x}(t(i) + L_f)\big)^\top$ such that $\boldsymbol{f}_j^\top \cdot \boldsymbol{r}_{j,i} \geq \gamma_j$, and $K_{\max j}$ denotes the maximum number of found spikes by filter $\boldsymbol{f}_j$. If two or more filters detect the same spike, the spike is assigned to one filter only, namely, to the one which had a response closest to 1. The optimal number of spikes for averaging is determined by

$$K_{\text{opt}\,j} = \underset{K}{\text{argmax}} \left\{ M_j(K) \right\} \tag{5.11}$$

where $M := P_{\text{D}\,j} + (1 - P_{\text{FA}\,j})$, and $\bar{\boldsymbol{q}}_j$ is estimated as the mean waveform of the Q last detections of filter $\boldsymbol{f}_j$.

5.2.10 Implementation

The higher order cross cumulants were calculated by the use of the HOSA toolbox [199]. The proposed algorithm was implemented in MATLAB® version 7.6, but not optimised for maximum computational speed yet. The code and a sample file will be made available at the website http://user.cs.tu-berlin.de/~natora/

Regarding computational complexity, the most expensive task is the computation of the cross cumulants during the SEA algorithm. This computation, however, can be done in parallel, in the sense that every time shift can be computed on a separate computing unit.

5.3 Performance Evaluation

5.3.1 Generation of artificial data

Artificial data were generated according to the model in Eq. 5.1. The waveforms were constructed from sorted spikes obtained from acute recordings in the prefrontal cortex of macaque monkeys and had a length of about 0.9ms, see Fig. 5.3. Detailed information about the sorting method and the experimental setup were described in Chap. 4. The spike arrival times were simulated as independent homogeneous Poisson processes with an enforced refractory period of 2ms. The noiseless data were simulated at a sampling frequency of 40kHz and then downsampled to 10kHz, in order to include the phenomenon of sampling jitter as encountered in real recordings. Gaussian noise with an autocorrelation structure measured in real recordings was simulated by an ARMA process and added to the spike trains (see Sec. 4.3.2 for more details). Three types of datasets were simulated, containing activity from one, two or three neurons.

Two data snapshots from the latter type are shown in Fig.5.4.

Figure 5.3: Waveform templates obtained from extracellular recordings in macaques and used for generation of artificial datasets.

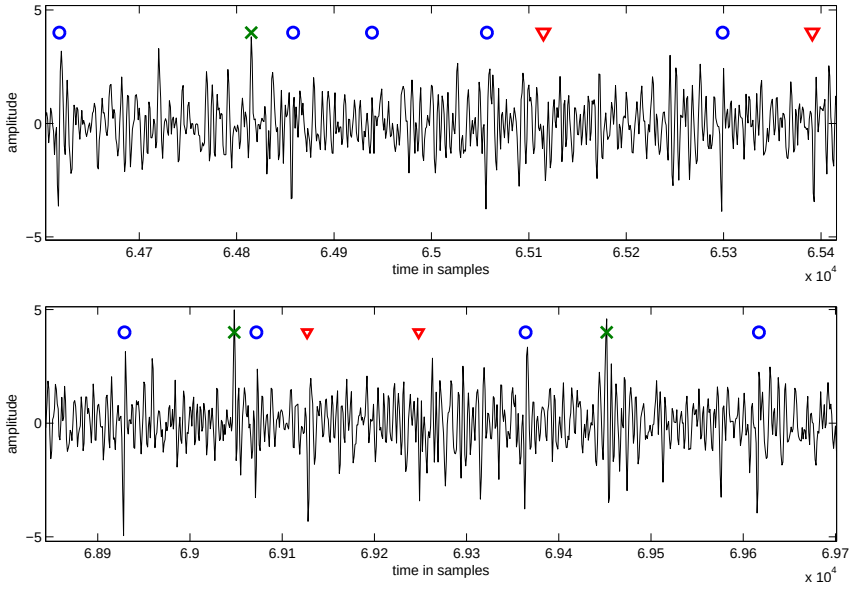

Figure 5.4: Data chunks of simulated data with different SNR values (in the top figure all waveforms had a SNR of 3.0, in the bottom figure the SNR was 4.0). The markers indicate the occurrence times of the inserted spikes, whereas the templates shown in Fig.5.3 were used.

5.3.2 Performance assessment

To allow for a better comparison, the most common definition of signal-to-noise ratio (SNR) utilised in the neuroscience community (see for example [147]), was used. Namely, the SNR of the i-th spike train is defined as the ratio between the norm of the corresponding waveform and the standard deviation of noise,

$$\mathrm{SNR}_i = \frac{\|\boldsymbol{q}_i\|_\infty}{\sigma_n}. \tag{5.12}$$

The detection performance of an algorithm was investigated by means of receiver operator characteristic (ROC) curves and the corresponding areas under the curves (AUC), similarly defined as in [102]. The ROC curves were calculated by evaluating the relative number of true positive (TP) and false positive detections (FP), given by

$$\begin{aligned} \mathrm{TP} &= \frac{\#\text{ of correct detections}}{\#\text{ of inserted spikes}}, \\ \mathrm{FP} &= \frac{\#\text{ of false detections}}{\text{maximum }\#\text{ of possible false detections}}. \end{aligned} \tag{5.13}$$

A detection was classified as correct, if the detectors response was within ±0.4ms of the true spiking time, which implied $\Delta = 2$, see Sec. 5.2.8, in the parameter setting of the *HBBSD* algorithm. Multiple detections within this time frame were ignored. Consequently, there is a maximum number of possible false positive detections a detector can produce in a dataset of finite length. By the definition in Eq. 5.13, both quantities TP and FP are bounded on the interval $[0, 1]$.

5.3.3 Parameter settings of HBBSD

In all subsequent simulations the following parameters were used in the *HBBSD* algorithm: The SEA algorithm was said to have reached convergence if $\|\boldsymbol{h}^{(k+1)} - \boldsymbol{h}^{(k)}\|_2 \leq 10^{-10}$. The SEA algorithm used higher order statistics with $p = 2$, but switched automatically to $p = 3$ if no convergence could be achieved in the former case. The SEA algorithm was initialised with a sine wave, but any other initialisation could be used. To ensure convergence following conditions were checked after every 150 iterations:

- If the skewness of the filter output was negative, the filter was changed to $\boldsymbol{h}^{(k)} \longrightarrow -\boldsymbol{h}^{(k)}$.
- If the last 10 $D(k)$ were not monotonically decreasing, where $D(k) := \|\boldsymbol{h}^{(k+1)} - \boldsymbol{h}^{(k)}\|_2$, the SEA algorithm was re-initialised with a random filter.

The minimum firing frequency $minf$ was set to 5Hz, the filter length was equal to 9 samples ($L_f = L_s = 4$), and a maximum number of 3 filters was allowed. Here we would like to point out that, unlike in some other methods, where the parameters are algorithm specific and thus their value setting is not an obvious task, the parameters of *HBBSP* are biologically motivated, allowing for a reasonable choice of their values. For example, since single channel data is analysed, it is sound to assume that action potentials from not more than 3 to 4 nearby neurons will be recorded, justifying a maximum filter amount of 3. The filter length can be chosen as the length of a spike, which is most often in the range of 0.4 to 1.0ms [147]. Besides, there exist methods to estimate the filter length even when no biologically motivated a priori knowledge is available [116, 176]. In Fig. 5.5 it is shown that the filter length L_f has only limited influence on the detection performance of the SEA algorithm. Finally, it is unlikely that neurons in a task relevant brain region will exhibit very low firing frequencies, but, as a matter of fact, the parameter $minf$ could be dropped entirely from the algorithm structure. The needed estimate of the waveform $\bar{\boldsymbol{q}}$ (see Sec.5.2.9) was obtained as the mean the $Q = 75$ last detections. As was demonstrated in Chap. 3, the choice of the value for Q is not critical.

Comparison of $p = 2$ vs. $p = 3$

The signals s_i are asymmetric as well as sparse, hence both statistics $p = 2$ and $p = 3$ should work in the SEA algorithm. In Fig. 5.6 the results are shown for both cases and also the performance of the corresponding MVDR filters (the maximum number filters set in HBBSD algorithm was one). The SEA algorithm achieved better performance in

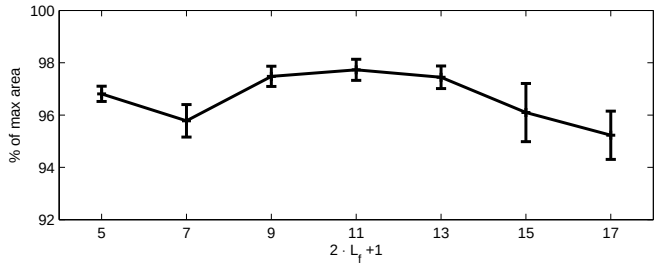

Figure 5.5: Same data and evaluation was used as in Sec. 5.3.5. The filter length L_f was varied and the AUC for the SEA algorithm was computed. The bars indicate the standard deviation over ten independent simulations.

the case of $p = 3$, and this performance advantage was propagated to the MVDR filters. The results were consistent for both datasets containing differently shaped waveforms. Nevertheless, since the performance difference between HBBSDkurt and HBBSDskew is quite small, the case $p = 2$ is used in the subsequent experiments. A lower order statistics has also the advantage of lower computational load and higher robustness to data outliers.

5.3.4 Competing algorithms

The algorithms chosen for comparison covered all three categories mentioned in Sec. 5.1. The focus, however, was on methods which make use of waveform information, since in general they achieve the best performance. In the case of amplitude crossing, the absolute value thresholding method was considered, hereinafter abbreviated as *ABS*. The non linear energy operator with a 5 point Bartlett window smoothing (*SNEO*) as described in [145] was chosen, representing a commonly used method based on the transient property of spikes. At last, 3 different methods relying on waveform information were compared. These included the wavelet method (*Wav*) presented in [147], the cepstrum of bispectrum method (*CoB*) from [187, 188], and the classical, single iteration, super-exponential method (*SEA*). The parameters for *Wav* and *CoB* were chosen according to their reference and adapted to the herein considered sampling frequency and spike length. Explicitly, for the *Wav* method, the wavelet family was set to "bior1.5", scales were set to $W_{\min} = 0.5$ms and $W_{\max} = 1.0$ms, number of scales was set to $J = 5$, acceptance mode was set to "liberal", and the sampling frequency was set to $f_s = 10$ kHz. In the case of *CoB*, the number of Fourier points was set to nfft = 80, the minimum spike interval was set to 2 ms, and sampling frequency was set to 10 kHz.

5.3.5 Performance on data with a single neuron

The first dataset contained spikes form a single neuron spiking at a frequency of 25Hz, whereas the waveform is shown on the left in Fig. 5.3. The ROC curves for every considered method are shown in Fig. 5.7.

In general, methods which estimate the waveform from the data outperform the generic approaches such as *ABS* and *SNEO*. *Wav* relies on an accordingly chosen mother wavelet by the user. However, if the shape information is not available, and thus the default mother wavelet is used, the performance of this method might be very poor, as

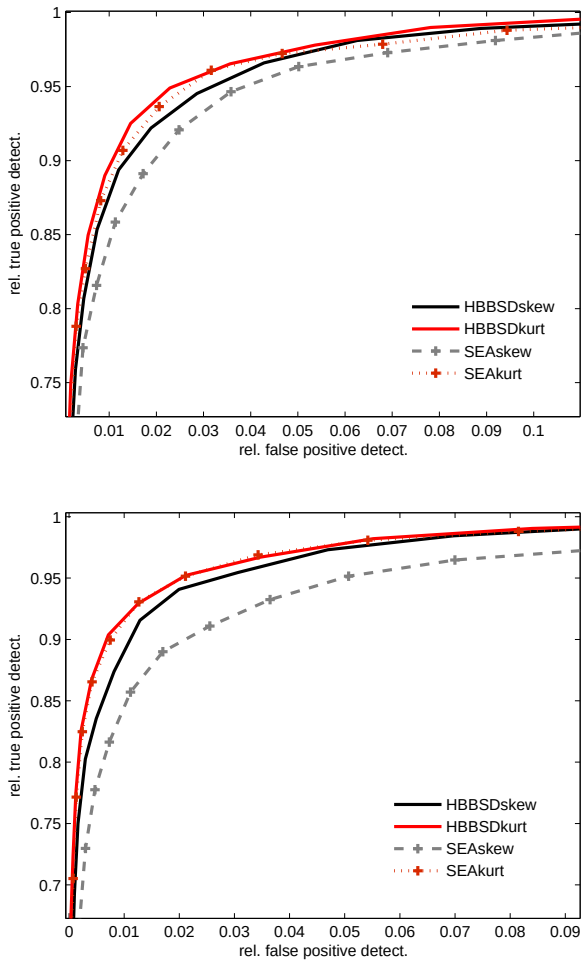

Figure 5.6: Average ROC curves for various spike detection methods. The shown results are an average over 10 independent simulations with a SNR = 3.0 and length of 6s. *Top:* The simulated data contained the left most waveform shown in Fig. 5.3. *Bottom:* The simulated data contained the middle waveform shown in Fig. 5.3.

indicated in Fig. 5.7. The decreasing number of true positive detections despite a decreasing threshold is explained by the fact, that *Wav* merges detected spike epochs if they are too close[‖]. The methods which estimate the filter from the data itself show good performance, whereas *HBBSD* achieves the highest score, followed by *SEA* and *CoB*. Based on these findings, we will focus on the comparison of *CoB*, *SEA* and *HBBSD* in the remaining sections.

5.3.6 Performance on data with two waveforms

Ten independent datasets, each of 6s in length, containing activity from two neurons with the first two waveforms shown in Fig.5.3 were simulated. The spiking frequencies were 15Hz and 25Hz respectively.

[‖]Of course *Wav* will achieve good results and outperform *ABS* and *SNEO* if the waveform is more similar to the used mother wavelet.

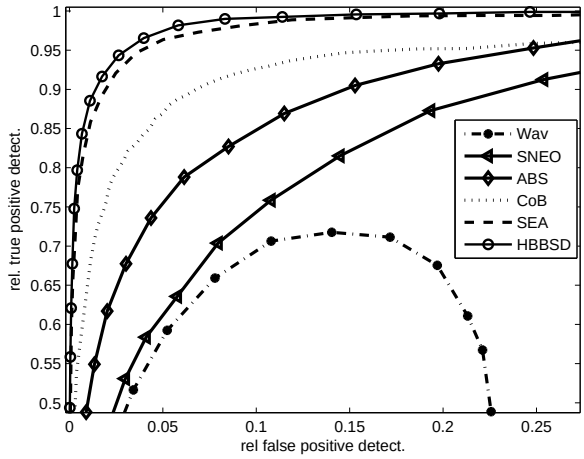

Figure 5.7: ROC curves for various spike detection methods. The shown results are an average over 10 independent simulations. Each simulation contained spikes from a single neuron, the signal-to-noise ratio being SNR = 3.0.

The SNR was varied from 3.0 to 4.25 in steps of 0.25 (all three spike trains always had equal SNR values), and again the ROC curves were computed for every method. To assess the overall performance for various SNR levels, the area under the ROC curves (AUC) was evaluated. In Fig. 5.8 the results for all compared methods are shown. *HBBSD* achieves a clearly better performance than the competing methods, since it calculates several filters. When the threshold is selected automatically, the performance of *HBBSD* often lies above the ROC curves (as e.g. in Fig. 5.8, bottom, or Fig. 5.9, top), since the threshold is selected for every filter individually, whereas for the ROC curves generation, the threshold is varied uniformly for all filters.

5.3.7 Performance on data with three waveforms

Five independents simulations, each of 10s in length, containing activity from three neurons with the three waveforms shown in Fig.5.3 were simulated. The spiking frequencies were 15Hz, 25Hz and 20Hz respectively. The SNR was varied from 3.0 to 4.25 in steps of 0.25 (all three spike trains always had equal SNR values), and again the ROC curves were computed for every method. To assess the overall performance for various SNR levels, the area under the ROC curves (AUC) was evaluated and is reported in Fig. 5.9. Again, *HBBSD* achieves the best performance throughout all SNR levels. The large standard deviation in the case of low SNR value (Fig. 5.9, bottom) is explained by the fact that sometimes only one or two MVDR filters were calculated, since, due to the high noise, no further modes in the SEA output could be identified.

5.3.8 Performance on simultaneous intra/extra-cellular recordings

The same data as described in Sec. 4.3.1 were used, however, only single channel data were considered, and the data were downsampled to 10kHz for faster processing. Two cells from Long Evans rats (P17-P25) were stimulated by a current injection and simul-

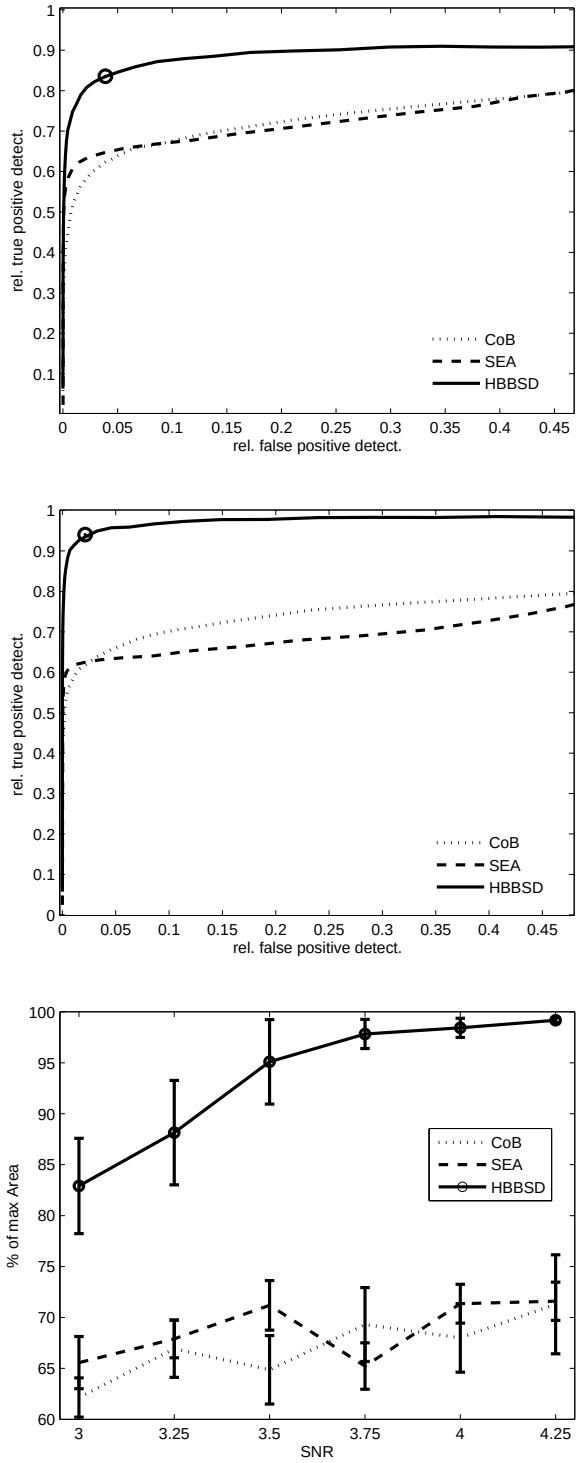

Figure 5.8: Average ROC curves and AUC for various spike detection methods. The shown results are an average over 10 independent simulations. The top figure shows the ROC curves the case of SNR = 3.25, the next figure in the case of SNR = 3.75. The circle indicates the performance of the *HBBSD* algorithm when the threshold is selected automatically according to Sec.5.2.8. The bottom figure shows the average relative AUC and the corresponding standard deviations for several SNR levels.

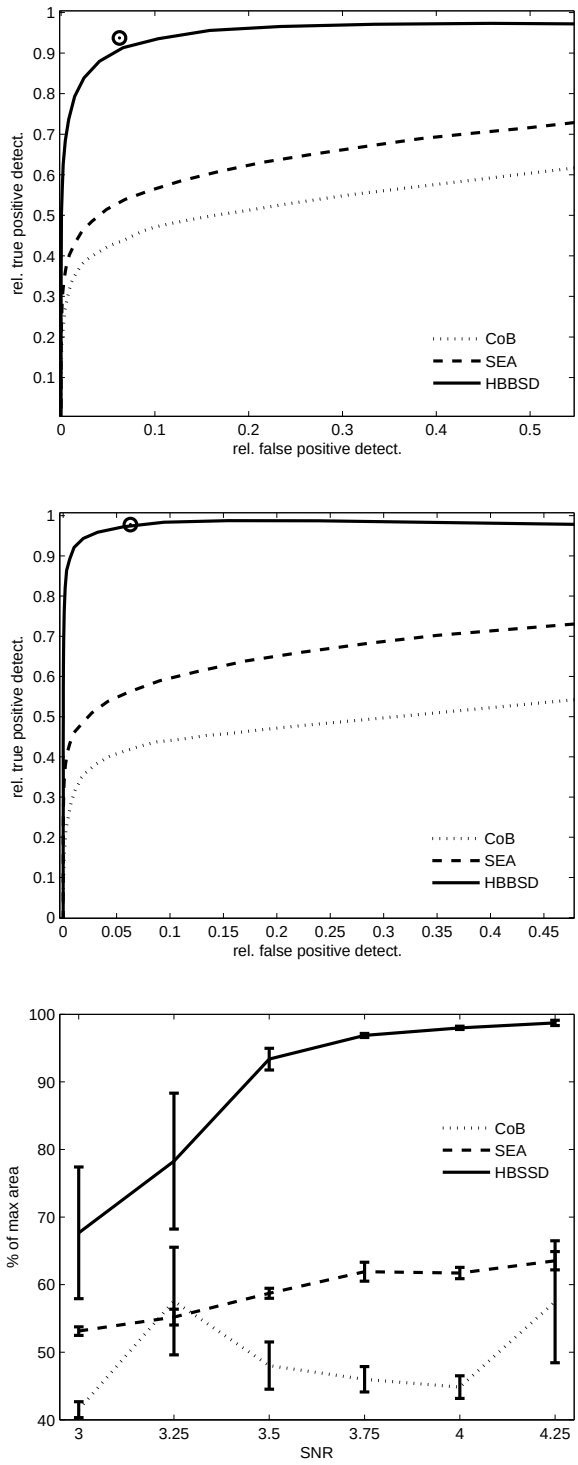

Figure 5.9: Average ROC curves for various spike detection methods. The shown results are an average over 5 independent simulations. The upper two plots show the performance for SNR values of 3.5 and 4.0 respectively. The circle indicates the performance of *HBBSD* when the threshold is selected automatically. The bottom plot shows the relative area under the ROC curves and the corresponding standard deviations for several SNR levels.

taneously the extracellular potential was recorded. In one of the experiments, the total number of spikes was 244, and the SNR was empirically determined as 3.050. Since the ground truth was known, the spikes were removed from the data, and higher order statistics were calculated on the remaining noise samples indicating a skewness of -0.053 and an excess kurtosis of -0.161. In the second experiment, a total of 103 spikes were found, the SNR being 3.008, the skewness being -0.012, and the excess kurtosis being -0.295. All the algorithms were applied to these real data with the same parameter settings as in

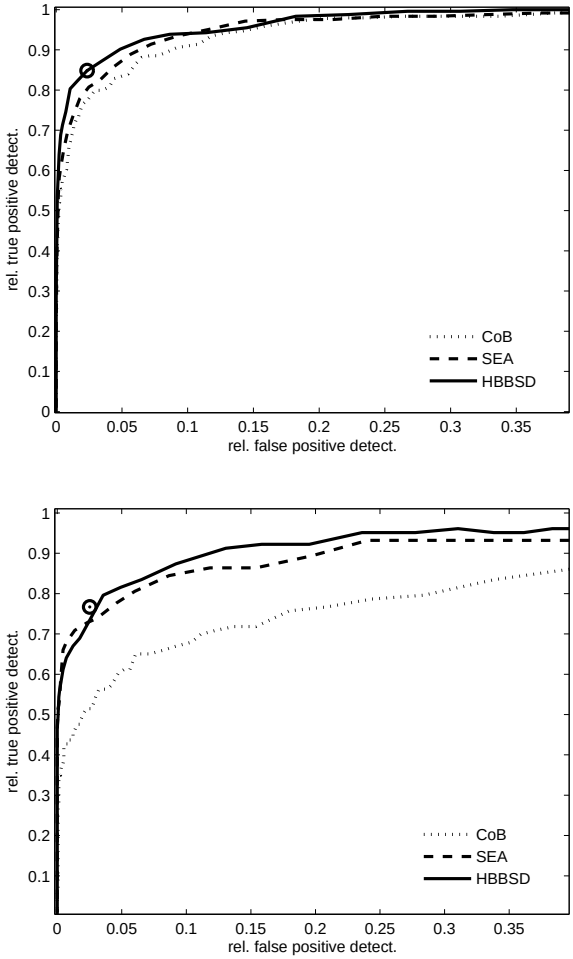

Figure 5.10: ROC curves for various spike detection methods on two datasets from simultaneous intra- and extracellular recordings of cells in rat slices. The circle indicates the performance of *HBBSD* when the threshold is selected automatically. *Top:* Performance on a dataset with an empirical SNR value of 3.050 containing 244 spikes. *Bottom:* Performance on a dataset with an empirical SNR value of 3.008 containing 103 spikes.

the case of artificial data. The results are shown in Fig. 5.10. As the data contained activity from only one cell, the performance gain of *HBBSD* compared to the other methods is not that pronounced as on datasets containing several distinct waveforms. The results show also, that *HBBSD* is robust to violations of the assumptions made in the data model Eq. 5.1. Neither the skewness nor the excess kurtosis of the noise were equal to zero, nevertheless, the algorithm still achieved favourable results.

5.3.9 Performance on non-stationary data

Data with temporally changing waveforms were generated in the following manner: The first 8s contained temporally constant waveforms and served as initialisation data for the spike detection algorithms. Afterwards, the waveforms started to change for the next 2.5min according to a normalised linear mixture (drift data), and finally in the last 50s, again constant waveforms were present (end data). To sum up, the waveforms followed the model**

$$\boldsymbol{q}[t] = \begin{cases} \boldsymbol{q}_{i_1}, \forall t \leq 8s \\ \alpha_{i_3}[t] \cdot \boldsymbol{q}_{i_3}[t], \forall t \in [8s, 158s] \\ \boldsymbol{q}_{i_2}, \forall t \geq 158s \end{cases} \tag{5.14}$$

where $\boldsymbol{q}_{i_3}[t] := \frac{\boldsymbol{q}_{i_2} - \boldsymbol{q}_{i_1}}{150s} \cdot t + \frac{158 \cdot \boldsymbol{q}_{i_1} - 8 \cdot \boldsymbol{q}_{i_2}}{150}$. The value of $\alpha_{i_3}[t]$ is set so that the SNR value stays constant all the time. Two different scenarios were simulated. In the first one, the data contained a 25Hz firing neuron, whose waveform had a SNR of 3.5 and changed from the second to the first waveform shown in Fig. 5.3. In the second scenario, data containing two neurons firing at 15Hz and 25Hz respectively were simulated. The waveform of one neuron changed from the second to the first waveform, whereas the waveform of the second neuron changed from the first to the third waveform shown in Fig. 5.3.

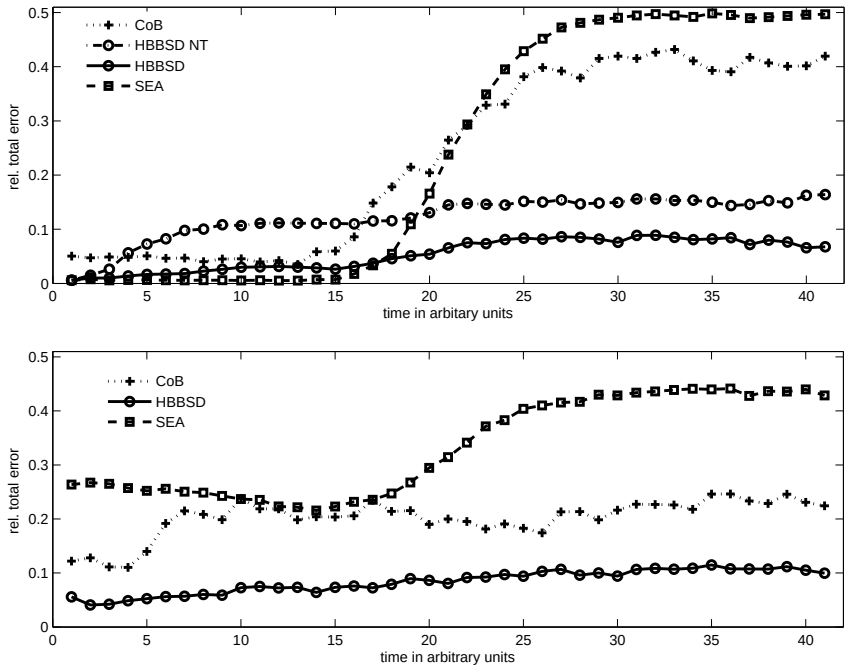

Figure 5.11: Average relative total error of various spike detection method in the case of non-stationary waveform templates. The shown results are an average over 10 independent simulations. *Top:* Data containing a single, temporally changing waveform. *Bottom:* Data containing two, temporally changing waveforms.

The filters of the *HBBSD* method were adapted as described in Sec. 5.2.9 after every $T = 5$s, whereas the thresholds as described in Sec. 5.2.8. For comparison to non-

**In order to distinguish the time dependent waveforms from the notation in previous section where the time index referred to a vector entry, the notation $\boldsymbol{q}[t]$ is used here.

adaptive methods, the MVDR filter from the SEA algorithm applied on the initialisation data was calculated and used for spike detection on the drift and end data. The threshold was also kept constant to the value obtained on the initialisation data by the method described in Sec. 5.2.8 (this method is still denoted as *SEA* in Fig. 5.11, since it relies on a single filter). Similarly the filter computed by the *CoB* method on the initialisation data, was used for spike detection on all subsequent data segments. The threshold was set to the default value of $0.04 \cdot k_i$, where k_i denotes the maximum of the filter output on the i-th data segment [187]. The performance of the algorithm was evaluated with respect to the relative total error TE which is defined as

$$\mathrm{TE} = \frac{\mathrm{FP} + (1 - \mathrm{TP})}{2} \tag{5.15}$$

where FP and TP are given by Eq. 5.13. The worst possible detector would have a score of TE = 1, the score for any reasonable detector, however, should not exceed TE = 0.5, as it either detects all spikes but also generates a lot of false positive detections or vice versa.

The results for both scenarios are shown in Fig. 5.11. The *HBBSD* algorithm was run in one of the scenarios without adapting the threshold, which is denoted as *HBBSD NT*. Clearly, the adaptive algorithms achieve much better performance than the static methods, whereas the fully adaptive *HBBSD* scores best. *CoB* achieves in general a better performance then *SEA*, because the threshold is data driven (i.e. relative value of maximum filter output amplitude), while on the other hand a fixed absolute value for *SEA* was used.

5.4 Discussion and related literature

The reason why *HBBSD* achieves better performance than the spike detection algorithms belonging to the first two categories mentioned in Sec. 5.1 is clear: Taking into account the full waveform shape for detecting a spike is always more advantageous than just considering its amplitude or energy. In this section we want to discuss some of the differences to other blind equalisation methods, but also point to some relevant literature, which might be useful for further development of spike detection algorithms.

The question remains why *HBBSD* outperforms *CoB*, even though *CoB* also uses higher-order statistics. The difference is that *CoB* intendeds to construct an inverse filter, whereas our method constructs a matched filter, resp. the MVDR filter. An exact inverse filter, denoted as $\boldsymbol{f}^{-1}$, achieves perfect channel equalisation, meaning that $\boldsymbol{f}^{-1} \star \boldsymbol{q} = \delta_0$. In general, $\boldsymbol{f}^{-1}$ will be an IIR filter, and only in some exceptional cases it reduces to a FIR filter. Then again, chances are high that the exact inverse filter is not stable, which is an essential requirement in the case of spike detection, as all recordings are noisy. The problem of stability is avoided in [188] by computing an approximate inverse filter which is a FIR filter. To further suppress the noise, a wavelet based denoising algorithm is applied on the filter output. However, if the waveform and noise statistics are estimated well enough, the resulting matched filter outperforms in terms of detection performance any two stage method combing inverse filtering with denoising. Another advantage of *HBBSD* is that the actual waveforms are estimated, and thus can be displayed to the

user. This allows for a semi-supervised operation mode, in which the algorithm finds the waveforms, and the user decides whether they are used for further spike detection or not.

Wavelet based methods achieve in general a good performance when the mother wavelet is similar to the actual waveform, and new methods still continue to appear. For example in [244] a good literature list of existing approaches to spike detection via wavelet transform is given. The proposed method uses biorthogonal mother wavelet, but instead of thresholding on individual wavelet coefficients (as in [147]), the output at different scales is multiplied, i.e. multiscale correlation coefficient are considered. A spike is detected when this coefficient is larger than the output at a single scale.

On the other hand, in [11] the continuous wavelet transform is used as feature extraction. Spike detection is then done by hypothesis testing in the wavelet space. First, it is tested if there are spikes at all. If there is only noise, the hypothesis is that the wavelet coefficients are Gauss distributed, whereas if there are spikes present, the hypothesis is that the distribution is a mixture of a Gauss and an uniform distribution. Which hypothesis is true is decided by the Bayesian information criterion. If the second hypothesis is true, spikes are detected via maximum posteriori probability.

The biggest advantage of wavelet methods is that they do not need an initialisation/learning phase, but can be directly applied to the recordings. This is especially important in acute recordings, as due to tissue drifts, new neurons might appear which were not present during the initialisation phase. As all spikes are generated by neurons, the waveforms cannot be completely arbitrary but follow a mono-, bi-, or tri-phasic shape. Therefore, an approach relying on a single mother wavelet family might deliver poor performance, but a future attempt which combines multiple wavelets families could overcome the problem.

Another issue with blind deconvolution methods relying on higher order statics might be their susceptibility to data outliers. This problem, however, might be mitigated by using robust higher order statistics [230].

In Sec. 5.2.3 we used a rather heuristic approach to identify the modes in the filter output. For example we did not employ any tests to decide whether a found maximum corresponds to a mode resulted from a true underlying probability distribution, or whether it is just an artifact/outlier. Nevertheless, the problem of identifying modes is a known problem and several test has been proposed. They rely for example on the amount of excess mass [146, 57], critical bandwidth selection [194, 127], or graphical mode tree representations [142, 143]. Such a test was not implemented in our algorithm to avoid further computational complexity. It is also important to notice that we do not attempt to identify true components in the filter output, as this would correspond to the task of spike sorting. The found mode can, e.g. in the case of similar waveform of two neurons, still be a mixture of two or more components. In this scenario, however, the MVDR filter will also be calculated from an average template, and spike from both neurons will be detected well. The *HBBSD* algorithm solely aims at deliver favourable performance, when this is not the case, i.e. when highly various waveforms are present in the data.

5.5 Conclusion

To our knowledge, blind equalisation algorithms relying on higher order statistics have rarely been applied to the task of neural spike detection. In this work, the super-exponential algorithm has been used for initial filter estimation. Furthermore, a mode detection and a sparse deflation procedure have been proposed in order to extract multiple spike waveforms, which have then been used for constructing MVDR filters.

To sum up, a novel method for unsupervised spike detection has been presented, which relies on the inherent characteristics of data from neural recordings, such as sparseness and binary sources. For instance, the sparseness of the neuronal signal was exploited for mode finding in the filter output and for proposing a sparse deflation procedure which reduces error propagation. On the other hand, the binary source property allowed for an appropriate choice of the statistics for the SEA algorithm as well as for an easy estimation of the waveforms and construction of the MVDR filters.

In contrast to existing blind devonvolution methods which assume a finite alphabet or binary sources such as [242, 112, 114, 43], we also made use of the spareness property and formulated a statistical algorithm (as opposed to deterministic/algebraic ones) which does not rely on extensive optimisation of some cost functions. On the other hand, existing approaches dealing with sparse signals often assume instantaneous mixtures or apply a corresponding transformation into the frequency domain [153], or use clustering techniques together with further assumptions about the data (like high SNR) [121]. In this contribution, we operated always in the time domain whereby no further assumptions had to be made about the data. Moreover, we focused on the task of spike detection, thus, the complete separation of all sources is not required as it is in the existing approaches. The special structure induced by spareness and convolutive filters is currently still being investigated and only first attempts have been made to fully incorporate it into algorithm design [131, 237].

The main advantage of our method, namely that several data driven filters are calculated, resulted in a superior performance of *HBBSD* compared to wavelet methods or other existing blind equalisation algorithms. Furthermore, since the waveforms are estimated, this could be used as an initialisation for a spike sorting algorithm, for example using the idea of [103]. On the basis of waveform estimation, we also proposed a procedure for optimal threshold selection and drift adaptation. Especially the latter one again relies on the distinct properties of neural data.

The whole algorithm was tested on various datasets and compared to current state-of-the-art spike detection techniques. The used data covered simulated datasets containing one, two, or three distinct waveforms, but also experimental data containing a single waveform. In all these different conditions the proposed algorithm worked well and delivered better performance than the competing methods.

Chapter 6

Unsupervised (multi-channel) electrode positioning

Although nowadays most multi-electrode arrays are equipped with an electric motor drive unit which could be controlled automatically by a machine, this unit is still operated manually in most laboratories. Manual control, however, is not only a tedious procedure which consumes a lot of potential experiment time and is infeasible for large arrays containing up to 64 electrodes, but can also be very inaccurate due to continuous tissue drifts. We propose a quality measure which indicates the difficulty to detect spikes at a given electrode position, as well as the difficulty to classify spikes correctly. Furthermore, a positioning algorithm based on stochastic approximation is developed, which finds an optimal recording position with respect to this quality measure. The algorithm does not only position the electrode in an unsupervised manner, but also monitors continuously the quality and corrects for tissue drifts. The method is demonstrated on realistically simulated data, and it is shown that it is indeed able to find favourable recording positions even in drifting environments.

6.1 Introduction

The use of large arrays of multi-electrodes (AME) is a popular recording technique, since it combines two favourable aspects with respect to data analysis[‡]. Namely, the temporal resolution is high enough so that the activity of single neurons is available, and at the same time the activity from a large number of neurons from the same subnetworks is recorded, allowing for a spatial resolution high enough to study neural interaction phenomena. While more and more methods are published about the problems how to process, sort and analyse such large amounts of data obtained from AME recordings (see e.g. [16, 203], and Chap. 4, 5), only few contributions deal with the task of properly positioning the individual multi-electrodes. When considering acute recording experiments, in which often arrays of 16 or up to 64 tetrodes are used, it is evident that

[‡]see also Chap. 1 for more information about electrophysiological multi-electrodes.

positioning every tetrode manually is a time consuming part of the experiment. This is of particular concern when carrying out experiments with primates, as maximum experiment duration is often limited by national animal protection laws. Hence, there is a need for unsupervised multi-electrode positioning algorithms which would place individual electrodes not only faster and more reliably than a human, but possibly also several electrodes simultaneously, considerably reducing the setup time.

Two other important factors motivate the use of such an unsupervised positioning system. First of all, a manual placement of the electrodes introduces a certain bias in the subsequent recording. Most often, the experimenter will try to place the electrode at positions where neurons with high firing frequency and high signal-to-noise ratio (SNR) are present. The quality assessment of these two criteria is done by visual inspection, thus, the final electrode position depends on personal judgement of a human and might be far away form the actual optimal recording position. Moreover, in brain regions in which computing is done in a distributed way (ensemble coding), criteria other than high firing frequency might be more appropriate.

Secondly, even when the experimenter succeeds to place all electrodes at favourable positions at the beginning of the recording session, this effort may have only limited benefit. In fact, due to the insertion of electrodes during the setup period, the brain tissue is compressed, while during the experiment, the tissue relaxes again, which leads to a displacement between the electrodes and the surrounding neurons [21]. Consequently, an experimenter would have to monitor the recording quality of each electrode constantly, and adapt its position in order to maintain acceptable recording performance.

In [148, 21] an autonomous electrode positioning algorithm was proposed, which was designed to position *single channel* electrodes such that *single unit* recordings are achieved. In combination with a micro drive unit described in [29, 22] this algorithm was used to autonomously control electrodes in implantable electrode devices. The positioning algorithm was then slightly improved, mainly the clustering part, in the work of [231, 232]. Because basically all the fundamental research on which the later publications rely was done in [148], we will always refer to this paper for comparison.

The hereinafter proposed method differs from the existing approach in several ways. In brief, the use of tetrodes (or other multi-channel electrodes) allows for a superior discrimination performance of the recorded spikes, simplifying spike classification on such data as compared to data from single channel electrodes [75]. This is due to the fact, that a spike waveform is recorded simultaneously on several recording channels (”stereo-effect”), thus, the discrimination task is facilitated thanks to the higher dimensional space in which the spikes are represented. Keeping this advantage in mind, it should be preferred to record from several neurons in order to maximise the information yield about the local neural population. Hence, we propose a quality measure which favours electrode positions where it is most likely that several well discriminable neurons are present, and not just a single cell. Moreover, in contrast to the work presented in [148, 231] the proposed quality measure does not rely on error-prone results of spike sorting. Since our algorithm is especially designed for multi-channel electrodes, and in particular it will be tested with 4-channel electrodes, the term ”tetrode” will be often used in the following sections, but the method applies to any k-channel electrode in fact.

In general, a positioning algorithm relies on a processing flow as shown in Fig. 6.1. After detecting spikes (first step), features are extracted from them and eventually a clustering procedure is applied (second step). This is then used to compute a quality measure which indicates the goodness of the current electrode position (third step). Based on this measure a positioning/control algorithm decides where to move the electrode in the next time step (fourth step). The differences between the steps in our approach and the one presented in [148] will be discussed in the corresponding subsequent sections.

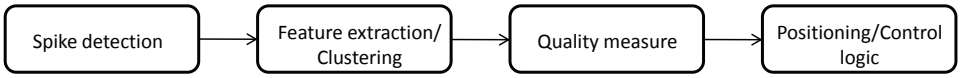

Figure 6.1: Processing stages needed for an unsupervised electrode positioning algorithm.

To evaluate the proposed automatic positioning algorithm, realistically simulated extracellular potential recordings were utilised. Namely, compartmental membrane currents of a spiking, reconstructed L5 pyramidal cell from [126] were simulated by the group of Prof. Gaute T. Einevoll (Norwegian University of Life Sciences) using the simulation tool NEURON [149, 83], and used to calculate extracellular potential-traces using the line-source method presented in [85]. The field potentials were then used in a simulation environment which allows the simulation of virtual tetrode movements in a volume containing several neurons and realistic noise.

The remainder of this chapter is organised as follows. The simulator used for evaluation of the system is outlined in Sec. 6.2. In Sec. 6.3 all the processing stages of the algorithm are presented, including the new quality measure and the positioning algorithm. The results of the evaluation can be found in Sec. 6.4, and conclusive remarks are given in Sec. 6.5.

6.2 Extracellular action potential simulation

The simulation of extracellular action potentials is a research field of its own. In this section a very brief description is given of how the data provided by the group of Prof. Gaute T. Einevoll were simulated. More detailed explanations and discussions about the problematics in calculating extracellular action potential can be found in [161, 162, 164, 163].

6.2.1 Calculation of extracellular field potentials

Extracellular field potentials around a reconstructed layer 5 pyramidal neuron were calculated using a forward electrostatic scheme similar to the line-source method described in [85]. The reconstructed neuron was a cat L5 pyramidal neuron published in [126]. The membrane currents for each of the 1094 compartments of the reconstructed neuron were calculated using the simulation tool NEURON [149] with the Python interpreter [83], using a somatic action potential (AP) trace as a forced boundary condition in the single compartment representing soma.

For the compartmental neuron simulation, purely passive membrane properties were assumed, with an intracellular, axial resistivity of $R_\mathrm{a} = 150\Omega\mathrm{cm}$, membrane resistivity $r_\mathrm{m} = 30000\Omega\mathrm{cm}^2$, membrane capacitance $c_\mathrm{m} = 1.0\mu\mathrm{F/cm}^2$, and an initial crossmembrane potential of $v_\mathrm{init} = -65\mathrm{V}$. The simulated membrane currents and the corresponding coordinates of these sources were used to estimate the extracellular potential (EP) at each time-step using the line-source method [85], with a homogenous extracellular conductivity of $\sigma_\mathrm{e} = 0.3\mathrm{S/m}$.

The soma, with mid-point position $\boldsymbol{r}_\mathrm{soma} = (0,0,0)^\top$, was treated as a point source, and the contribution to the EP from the somatic membrane current $I_\mathrm{soma}(t)$ in coordinate $\boldsymbol{r}$ is in the quasistatic approximation to Maxwell's equations given by $\Phi(\boldsymbol{r},t) = \frac{1}{4\pi\sigma_\mathrm{e}}\frac{I_\mathrm{soma}(t)}{|\boldsymbol{r}-\boldsymbol{r}_\mathrm{soma}|}$. The analytical solution to the linearly super-positioned potential from n segments, where $I_k(t)$ is the membrane current of segment k, is given by [85];

$$\Phi(\vec{r},t) = \sum_{k=1}^{n} \frac{I_k(t)}{4\pi\sigma_\mathrm{e}\Delta s_k} \log\left|\frac{\sqrt{h_k^2+\rho_k^2}-h_k}{\sqrt{l_k^2+\rho_k^2}-l_k}\right| , \tag{6.1}$$

where Δs_k is the segment length, ρ_k the distance perpendicular to the axis of the line-source, h_k the longitudinal distance to the end-point of the segment, and $l_k = \Delta s_k + h_k$ the longitudinal distance from the start-point of the segment [162, 85]. The calculations of EPs were done during the same simulations as the NEURON simulations, still using the Python interpreter.

In order to avoid singularities in the EP when the distance to individual segments was small, the minimum allowable distance to each line source was set to be the same as the diameter of each segment. This also ensured that the potential is not calculated within the intracellular space of the chosen morphology. The calculation of the EP was performed over the coordinates of 3D cubic grids spanning [-200, 200]μm and [-100, 100]μm, with spatial resolutions of 5 and 10 μm respectively, sampling the extracellular signature of the AP in the volume surrounding the somatic compartment and basal dendrites. The calculation of potentials at larger distances was not deemed necessary due to the low resulting extracellular amplitudes compared to the noise added at a later point. The resulting potential traces and corresponding coordinates were written to file on the HDF5-format, and then used by the extracellular recording simulator.

6.2.2 3-dimensional extracellular recording simulator

The simulator was mainly developed by Philip Meier, and by Felix Franke, and is publicly available [137]. It allows to compose a scene containing an arbitrary number of neurons of the type described in Sec. 6.2.1, each neuron having a unique orientation and firing rate. A multi-channel electrode recording is then simulated by generating a spike train according to a Poisson process with the corresponding firing frequency and the pre-calculated waveforms inserted*. An arbitrary fixed spatial channel configuration can be defined, but in this chapter we solely used a configuration imitating a tetrode. In

*The pre-simulated waveforms are only available at the resolution of 5μm, see Sec. 6.2.1. In order to allow electrode movements on finer resolutions, the waveforms were linearly interpolated inbetween using trilinear interpolation.

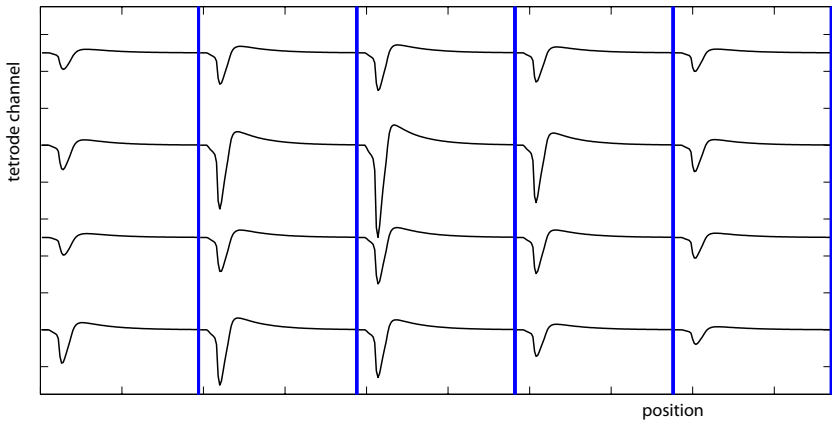

Figure 6.2: A single neuron was placed at the position $(10, 15, 150)^\top \cdot \mu$m. The five sections (separated by the blue vertical lines) show the waveforms recorded with a tetrode whose tip was at the position $(10, 15, z)^\top \cdot \mu$m, with z being 125, 137.5, 150, 162.5, and 175.

detail, the 3 rear channels were 10μm apart from one another, and the distance from the tip channel to each of the rear channels was 10.4μm.

Finally, Gaussian noise with a covariance structure fitted to real recording noise (see Sec. 4.3.2 for more details) is added to the spike trains. The capabilities of this simulator are illustrated in Fig. 6.2.

6.3 Processing stages of the positioning algorithm

In the following subsection all the processing stages needed to achieve an unsupervised multi-channel electrode positioning (see Fig. 6.1) are described.

6.3.1 Spike detection

The positioning algorithm should work fast, i.e. find a good recording position in an amount of time which is considerably shorter than a human operator would need. The used spike detection must, hence, detect spikes reliably given only few seconds of data. For this reason we suggest to use an algorithm based on amplitude or transiency detection, rather than relying on waveform information (see Sec. 5.1 for an introduction to spike detection). The proposed spike detection algorithm in Chap. 5 needs a learning phase in order to estimate the waveforms blindly, whereas a method based on wavelets (as e.g. in [147] and used in [148]) might perform poorly when the mother wavelet does not match the actual waveforms. A spike detection method based on amplitude or transiency seems, therefore, to be a good tradeoff between detection performance and processing time needed.

6.3.2 Feature extraction, clustering

In the approach presented in [148] the found spikes are aligned and projected into a wavelet space. Only the largest wavelet coefficients are then used for the subsequent

clustering, which relies on a Gaussian mixture model. The clustering is done for each electrode position individually, whereas in [232] the clustering is more sophisticated in the sense that information from the previous clustering at step $k-1$ is incorporated in step k.

Nevertheless, an unsupervised clustering on short data chunks will always be prone to errors. The biggest issue is the reliable estimate of the number of clusters, which corresponds to the number of neurons. Since the quality measure will depend on the number of clusters, a suddenly changed number will greatly influence the score of the quality and the following positioning logic. Moreover, clustering is always a time consuming task, thus, prolonging the positioning time in total. We, therefore, omit the clustering step completely and define a quality measure solely relying on unsorted spike information. Of course such a measure might not predict the true optimal position in all cases, but it will be shown that the chosen approach works well.

6.3.3 Quality measure

The goal of the positioning system presented in [148, 21] was to isolate a *single* neuron. Accordingly, the quality measure was designed in such a way that the better a single neuron is isolated from the "rest", the higher the quality. In [20] several quality measures were proposed and compared, including SNR of dominant cluster, projection t-statistics, L-ratio, isolation distance, silhouette ratio and symmetric Kullback-Leibler divergence. Finally, however, the simplest measure was chosen, i.e. the SNR of the dominant cluster whereas the SNR was defined as the peak-to-peak amplitude of the waveform. This was due to the sensitivity of the other measures to clustering errors.

On the other hand, the objective of the proposed method is to find a position where activity from many, well separable neurons is recorded. Therefore, we need a quality measure which does not indicate how well the dominant cluster is separated, but how well all present clusters are separable. In the following, again the data model presented in Eq. 1.1 and Eq. 1.2 is assumed, where $\boldsymbol{q}^i$ represents the waveform of neuron i. Further, let us assume that all neurons have equal firing rates, and that the noise is white[§]. Then, using linear decision boundaries, the data is easier to cluster the more the waveforms are apart [80], i.e. the larger Q_D is, where

$$Q_D := \frac{2}{M \cdot (M-1)} \sum_{i=1}^{M} \sum_{j>i}^{M} \left\| \boldsymbol{q}^i - \boldsymbol{q}^j \right\|, \tag{6.2}$$

and M is the number of neurons. The idea is to find a similar measure for un-clustered data. Let ς_k^i denote the empirical SNR[**] of the i-th detected spike on channel k. Then, we define Q_{stereo} as

$$Q_{\text{stereo}} := \frac{1}{N} \sum_{k=1}^{N} \frac{1}{W} \sum_{i}^{W} \left(\langle \varsigma \rangle_k - \varsigma_k^i \right)^2 =: \langle \text{Var}\left(\varsigma_k\right) \rangle_N , \tag{6.3}$$

[§]This last assumption is not necessary. Instead once the cluster centres are determined, a whitening transformation can be applied, see e.g. [168, 180].

[**]We chose the SNR for simplicity, but in principle, any other feature of a spike could have been chosen, including the waveform itself. A more advanced feature diminishes the risk that two neurons will have quite distinct waveforms, but still cancel each other out in the quality calculation.

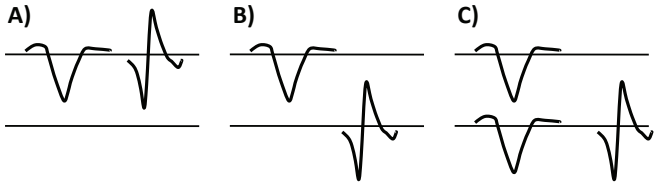

Figure 6.3: Toy examples illustrating Q_{stereo}. Assume that both neurons/waveforms have a SNR of 1 or 0, depending on the recording channel (in this toy example a multi-channel electrode with 2 channels is assumed). Then, the configuration in A) yields $Q_{\text{stereo}} = 0$, whereas for B) one has $Q_{\text{stereo}} = 0.5$, and $Q_{\text{stereo}} = 0.25$ for C). This means, that the configuration in A) does not allow a separation of the two neurons (at least not based on the SNR values), while the configuration B) is more favourable than C), since there is not any overlap in the SNRs of the waveforms.

where $\langle\varsigma\rangle_k := 1/W \sum_{i=1}^{W} \varsigma_k^i$, W denotes the total number of detected spikes, and N indicates the number of recording channels. The motivation behind this definition is the following: When two neurons have different SNR values on a channel, this implies that their spikes can be easily separated, as one can take the SNR value as a discriminative feature. Hence, the discirminability between neurons becomes better, the larger the spread of SNR values. The spread is expressed as the sample variance of the found spikes, and the mean over all channels is taken for a consistent normalisation in the case of various multi-channel recording devices. The usefulness of this measure is also illustrated in Fig. 6.3.

On the other hand, we define Q_{SNR} as

$$Q_{\text{SNR}} := \langle\langle\varsigma\rangle_k\rangle_N \,. \tag{6.4}$$

This measure simply indicates the average SNR value of the recording, hence, how easily spikes can be detected at the current recording position. The corresponding measure, if clustering were done, would be

$$Q_S := \frac{1}{M} \sum_{i=1}^{M} \left\| \boldsymbol{q}^i \right\| . \tag{6.5}$$

In order to check the usefulness of these two quality measures Q_{SNR} and Q_{stereo}, some prototypical neuron configurations were constructed and a virtual tetrode track simulated using the simulator described in Sec. 6.2.2, see Fig. 6.4. The tetrode was moved in steps of 0.5μm. No spike detection was used[†], instead the noise free waveforms simulated at each position were directly used for computing both of the quality measures Q_{SNR} and Q_{stereo}, see Fig. 6.5. As can be seen from this figure, the proposed measures exhibit very similar properties as the true ones Q_S and Q_D. Note, that these results were obtained in the case of equal fire rates of neurons belonging to the same cluster (namely 10Hz). As is turns out, however, simulations show that the proposed measures are quite robust, and will approximately display local maxima near the true local maxima even if the fire rates are not equal. Exemplarily, two cases are shown in Fig. 6.6. All simulations and quality measure calculations were also done in the case of a rotated tetrode configuration. The results show that a simple rotation of the tetrode

[†]Spike detection errors and noise might introduce some errors complicating the evaluation of the properties of a quality measure.

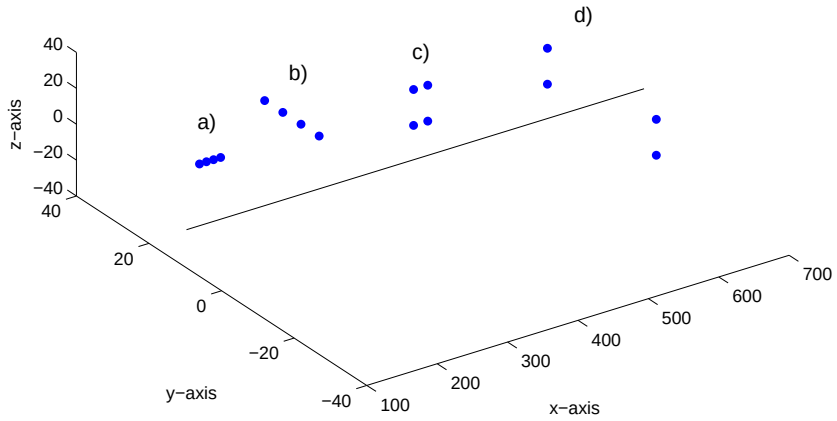

Figure 6.4: The blue dots symbolise neurons, whereas the black line indicates the track of the virtual tetrode. In this plot, the x-axis was shortened for illustration purposes, when in fact during the simulation each cluster a)-d) was at least 200μm apart from the next one, so that there is no interference between them. All neurons were identical and oriented identically, namely along the x-axis. Cluster a) consists of neurons in a line parallel to the tetrode track (inter neuronal distance being 10μm). Cluster b) consists of 4 neurons in a line perpendicular to the tetrode track (inter neuronal distance being 5μm). Neurons in cluster c) are arranged in the corners of a rectangle parallel to the track (inter neuronal distance being 20μm), whereas the rectangle in cluster d) is perpendicular to the track, i.e. the track runs trough its central point (inter neuronal distance being 20/30μm).

could significantly improve the recording quality, since the quality maxima are still at the same position. This suggests the development of motor drive units which allow for one-dimensional translational as well as for one-dimensional rotational movements.

6.3.4 Positioning and control logic

The objective of the positioning logic is to find a suitable position for the tetrode, where the quality measure exhibits a local maximum. This task is usually required at the beginning of an acute recording session. Once such a position is found, the quality of this position will most likely decrease after a while due to tissue relaxation. Consequently, the algorithm should detect such a decrease and re-position the tetrode until an acceptable recording quality is found again.

Since in Sec. 6.3.3 we defined two quality measures, the question arises whether they should be combined into a single quality or optimised separately. Q_{SNR} indicates only the difficulty to detect spikes, but contains no information about their discriminability. If one simply adds the two qualities, Q_{SNR} and Q_{stereo}, there is a high chance of losing important information and, thus, moving the tetrode to a position where the discriminability is sub-optimal*. Hence, we propose the following optimisation scheme: The tetrode is advanced with a constant step size, until Q_{SNR} exceeds a certain threshold. Once this happens, the positioning logic described in this section is activated, which attempts to find an optimal position with respect to Q_{stereo}. The advantage of this scheme is that

*See for example the first row in Fig. 6.5, in which Q_{SNR} exhibits a local maximum at a position where Q_{stereo} exhibits a local minimum.

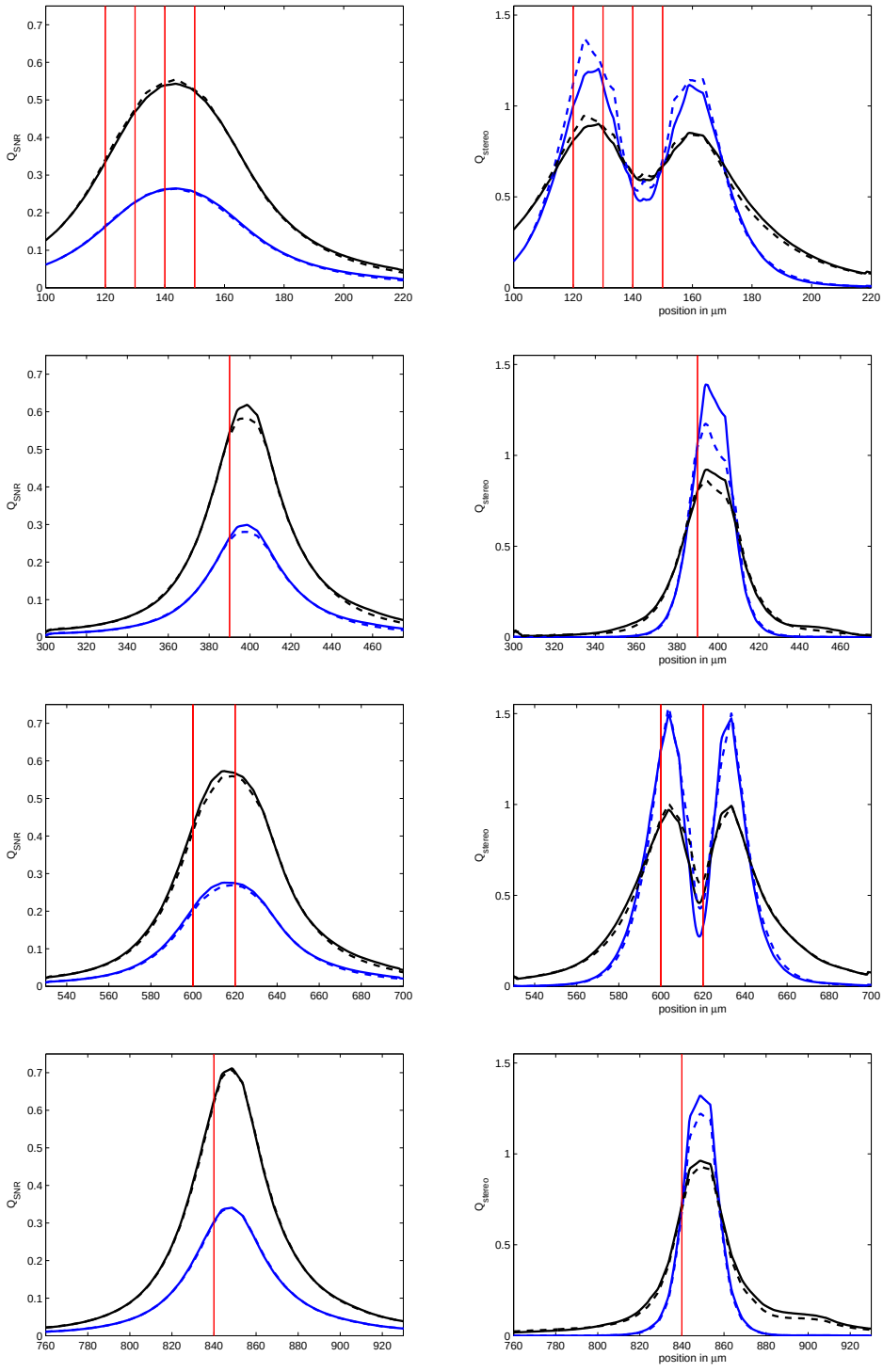

Figure 6.5: Q_{SNR}, left column, and Q_{stereo}, right column, are plotted in blue for various neuron configurations, i.e. row one corresponds to configuration a) in Fig. 6.4, row two corresponds to configuration b), and so on. In black are plotted Q_S, left, and Q_D, right. The dotted lines show the corresponding quality profile when the neurons are rotated by 45 degree around the axis defined by the tetrode track in Fig. 6.4. The red lines indicate the x-axis positions of the neurons.

it finds the position where the neurons are discriminated best, under the condition that

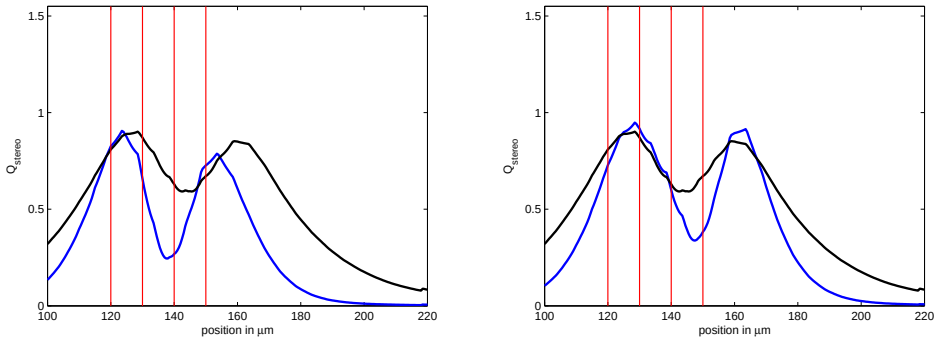

Figure 6.6: Shown in blue are Q_{stereo} for configuration a) in Fig. 6.4 when the individual neurons have different firing rates; and in black again is Q_D is plotted. On the left hand side, the neurons had firing rates of 30, 10, 10, 1 Hz, (the order corresponds to the order on the x-axis), while on the right hand side the firing rates were 10, 30, 1,10 Hz.

they can be detected sufficiently well. Note that in [148, 231] only a single quality was defined, hence, such two step optimisation scheme was not necessary.

The decision logic of the positioning algorithm was realised by implementing a finite state machine consisting of 4 states, see Fig. 6.7. The electrode stays at each position for a certain amount of time for gathering sufficient data to reliably estimate the quality measure. Depending on this value, the algorithm decides to which subsequent state the system should transit. In the following subsections each state as well as the transition criteria are described in detail. The term ”quality” or simply ”Q” will always refer to Q_{stereo}.

Search

This is the initial state and as long as the quality of the signal is below a certain threshold $Q_{\min}$, the electrode is simply advanced in the direction D (D is either -1 or 1, since electrodes can be moved only in either of two directions, namely back or forth) by a constant step size S_s. If three consecutive quality estimates yield a value larger than $Q_{\min}$, the algorithm changes to the ”optimize” state[†].

Optimize

The goal of this state is to determine the position at which the quality function exhibits a local maximum, thus, where the tetrode should be moved to, i.e. to find a tetrode position u^* such that

$$u^* = \underset{u \in U}{\operatorname{argmax}}\, Q(u) = \{u^* \in U \mid Q(u^*) \geq Q(u)\ \forall u \in U\} \tag{6.6}$$

The optimal position can be found by applying some optimisation techniques, however, some aspects must be considered: The function $Q(u)$ is not given, but only noisy measurements of it are available. Therefore, the gradient $g(u) := \nabla Q(u)$ is not available

[†]We require three consecutive quality estimations to be above threshold in order not to trigger the ”optimize” state by an outlier.

directly either. In this sense, one is dealing with a subclass of stochastic optimisation problems, namely stochastic approximation [64]. A common way to solve this kind of problems is to approximate the gradient by a finite difference. One could, for example, use the two-sided finite difference approximation, as described in [196]. Although this technique is widely applied, it is inappropriate for use in our setting. Namely, a realisation of a two-sided finite difference would imply that, in order to estimate the derivative, the tetrode would have to be moved forward and backward at every position (dithering). This might damage the brain tissue and also evoke further drifts. Similarly, any random search algorithm is inappropriate as well. Two feasible approaches are presented in the following.

Optimisation 1: Steepest ascent In [15] methods, called pseudo-gradient schemes, which avoid the direct estimation of the gradient are listed. In general, one attempts to transfer the idea of the steepest ascent algorithm to stochastic optimisation, i.e. $u_{k+1} = u_k + a_k \cdot g(u_k)$. One of them, presented in [219, 218], relies only on the sign estimation of the gradient. However, in this case the information about the quality at previous positions cannot be used directly, but a two step update rules has to be applied. For simplicity let us define the following basic update rule approximating the gradient by the difference between the last two consecutive steps, i.e.

$$u_{k+1} = u_k + a_k \cdot \frac{Q(u_k) - Q(u_{k-1})}{u_k - u_{k-1}}. \tag{6.7}$$

This update scheme can be related to the one-sided finite difference algorithm in [196] on p. 157, by the relation $c_k = u_k - u_{k-1}$. In [196] it is shown that sufficient conditions for convergence of the finite difference algorithm include, amongst other, all of the

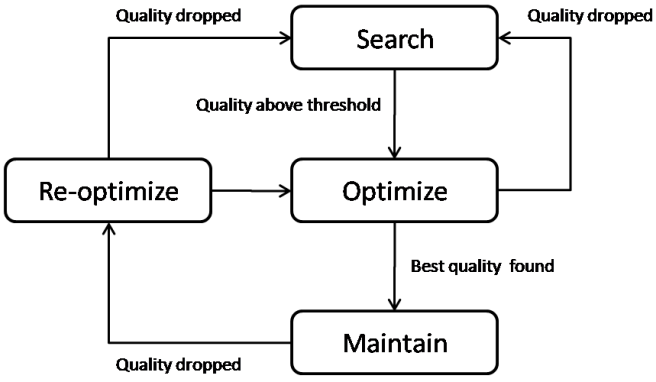

Figure 6.7: Illustration of the decision logic of the proposed positioning algorithm implemented as a finite state machine. The initial state is the ”search” state, in which the tetrode is advanced in the same direction until a reasonable quality is detected. After the transition to the ”optimze” state, the algorithm tries to find an optimal position such that the quality is maximum. Once such a position is reached, the algorithm switches to the ”maintain” state, and the quality is monitored. If the quality drops below a threshold value, the ”re-optimize” state is triggered, which decides in which direction the tetrode should be moved in order to find the maximum quality again.

following:

$$\begin{array}{lll} a_k > 0 & c_k > 0 & \forall k \\ a_k \to 0 & c_k \to 0 & \text{if } k \to \infty \\ \sum_{k=0}^{\infty} a_k = \infty & \sum_{k=0}^{\infty} \frac{{a_k}^2}{{c_k}^2} < \infty & \left(\sum_{k=0}^{\infty} a_k c_k < \infty \right) \end{array} \tag{6.8}$$

The last inequality in parentheses can be avoided when some conditions regarding the third derivative of Q are fulfilled (see p. 160, [196] for a detailed list). From these conditions it is evident that the convergence of such a simple update rule as the one in Eq. 6.7 is hard to guarantee, since the c_k cannot be chosen arbitrarily, but depend on the positions themselves. Therefore, we propose a second optimisation procedure.

Optimisation 2: Newton-Raphson A slightly different approach is to try to transfer the Newton-Raphson method to stochastic optimisation, having in mind that the Newton-Raphson method has a faster convergence in the near of the solution than the steepest ascent algorithm. The Newton-Raphson is for finding a root x of a function, i.e. $f(x) = 0$. Adapting this algorithm to find a local minimum/maximum of a function instead, leads to the update rule $u_{k+1} = u_k - a_k \cdot g(u_k)/g'(u_k)$ (in the original algorithm $a_k = 1\ \forall k$, but a different choice might increase stability, which is particularly important in the case of stochastic optimisation, see [196], chapter 1). In order to guarantee that the algorithm converges to a local maximum and not minimum when $g' > 0$, one can combine it with the steepest ascent algorithm, i.e.

$$u_{k+1} = u_k + a_k \cdot \frac{g(u_k)}{|g'(u_k)|}. \tag{6.9}$$

The problem of direct estimation of the gradient and the second derivative g' can be avoided by introducing an interpolation function. In [148] a single polynomial is fitted through all available quality measurements. The order of the polynomial is determined by maximising the model posterior probability, and has to be re-estimated at every new electrode position. At each position the quality is measured several times, although it is not specified how often exactly. The polynomial is fitted in such a way that the distance to all measured data is minimised in the least square sense. The Newton-Raphson scheme in Eq. 6.9 is then applied, and the optimal position u^* is said to be found at iteration k when $a_k \cdot g(u_k)/|g'(u_k)| < \epsilon$.

In order to simplify this procedure and to omit the error-prone task of order estimation [20], we use piecewise interpolation instead. As interpolation functions cubic Hermite polynomials are a reasonable choice, as, for example, they have no overshoots and oscillations in contrast to splines [45]. This avoids the task of order estimation and there is no risk of oscillations at the ends of the fitted data. Furthermore, in order to avoid a convergence which might oscillate around the optimal position as in [148], which would imply a passing of the tetrode through the same tissue several times resulting again in tissue damage and drift evocation, the following rule is applied: If $Q(u_{k+1}) < Q(u_k)$, then a second order polynomial $P_2(u)$ is fitted through the last three qualities $Q(u_{k+1}), Q(u_k), Q(u_{k-1})$. The final position is then determined as the maximum

Table 6.1: Default parameter values for the proposed positioning algorithm.

Parameter	Value	Reference
S_s	10μm	page 101
a_k	0.25	Eq. 6.9
T_{mm}	0.8	page 104
S_r	2μm	page 104
δ_{min}, δ_{max}	2μm , 15μm	page 105

of this polynomial, i.e. $u^+ = \text{argmax}_u P_2(u)$. Of course, it is probable that u^+ will not necessarily correspond to the optimal u^*. However, due to the fact that for small deviations any function is locally well approximated by a second order polynomial, u^+ and u^* will for the problem at hand lie close enough to each other, while the more important issue of tissue damaging is avoided. Once the optimal electrode position is reached, the algorithm switches to the ”maintain” state.

Maintain

The electrode stays at the best found position and the current quality is monitored in regular time intervals, until the quality drops under a certain value. Explicitly, if the quality Q drops below a certain absolute value, i.e. $Q < Q_{mm}$, or below a certain relative value, i.e. $Q < T_{mm} \cdot Q^*$, where Q^* denotes the highest quality which was measured at position u^+, the ”re-optimize” state is triggered. On the other hand, if there is a sudden dramatic quality drop, $Q < Q_{min}$, the algorithm returns to the ”search” state.

Re-optimze

Once the quality is not sufficient anymore, the algorithm has to find out in which direction the tetrode should be moved in order to find higher qualities again. Therefore, in this state the algorithm moves the electrode in an arbitrary direction[‡] by a constant step size of S_r. If the quality is even lower than at the previous position, the direction is inverted and the algorithm switches back to the ”optimize” state.

6.3.5 Exception handling

In a practical application of the positioning algorithm some additional constraints should be introduced in order to deal with unexpected events and technical limitations encountered in real recordings. Some of them are listed below:

- The transitions between states were mainly defined by some conditions on Q_{stereo}. In an ideal experiment, the algorithm should transit from the ”search” to the ”optimize” state, and then, depending on the tissue drift, iterate between the three states ”maintain”, ”re-optimize” and ”optimize”. If, however, for some reason there is

[‡]In fact, only when the ”re-optimize” state is reached for the first time, the direction is random. For subsequent decisions, the direction from the previous visit of the ”re-optimize” state is used first.

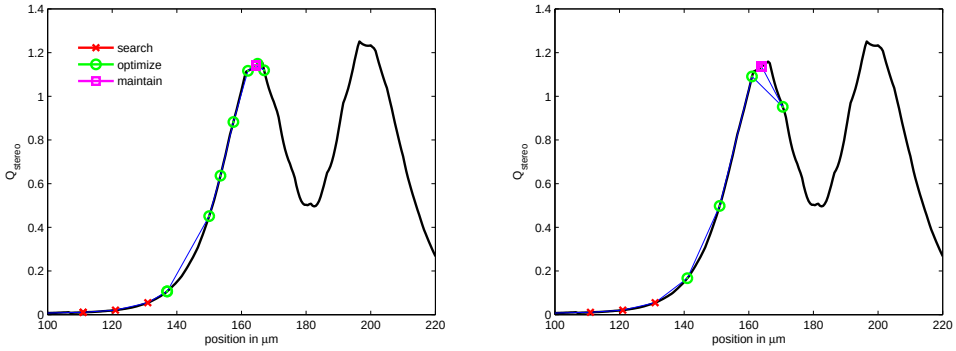

Figure 6.8: Visualisation of the positioning algorithm in the case of a static and noise-free quality profile. *Left:* Result in the case of $a_k = 0.15$. *Right:* Result in the case of $a_k = 0.25$. The other parameters are identical and given in Tabl. 6.1.

suddenly a huge quality loss, i.e. $Q_{\text{stereo}} < Q_{\text{min}}$ or $Q_{\text{SNR}} < Q_{\text{SNRmin}}$, the position logic immediately returns back to the search state.

- The position update scheme in Eq. 6.7 and Eq. 6.9 compute the position to which the tetrode should be moved. However, due to some errors in measurement the returned values might be either below the precision of the used motor drive unit, or very large, making it risky to drive such long distances without checking the quality in between. Thus, it makes sense to restrict the predicted position updates to some interval, i.e. $\delta_{min} < |u_{k+1} - u_k| < \delta_{max}$.

- The region U (see Eq. 6.6), in which an optimal position should be found, must be set in order to prevent the advancement of the tetrode in tissue layers not under investigation. Whenever the algorithm reaches a boundary of this region, the tetrode changes direction.

- There is always a chance that a neuron lies directly on the track of the tetrode. Since the SNR will be higher the smaller the distance between the tetrode and the neuron, it is likely that the tetrode will be advanced until it penetrates the neuron and possibly damages it. This could be avoided by introducing a maximum allowed Q_{SNR} value, which once measured causes the tetrode to stop at the current position.

- A maximum number of iterations should be defined in which the algorithm remains in the ”optimize” state. Due to fluctuations (such as very irregular fire frequencies) the algorithm might not be able to find the optimal position in reasonable time. In such a cases, the algorithm should switch to the ”search” state and try to find a different region with acceptable quality.

6.4 Results

In this section we present some results regarding the capabilities of the positioning algorithm described in Sec. 6.3.4. The used data was again generated with the simulator

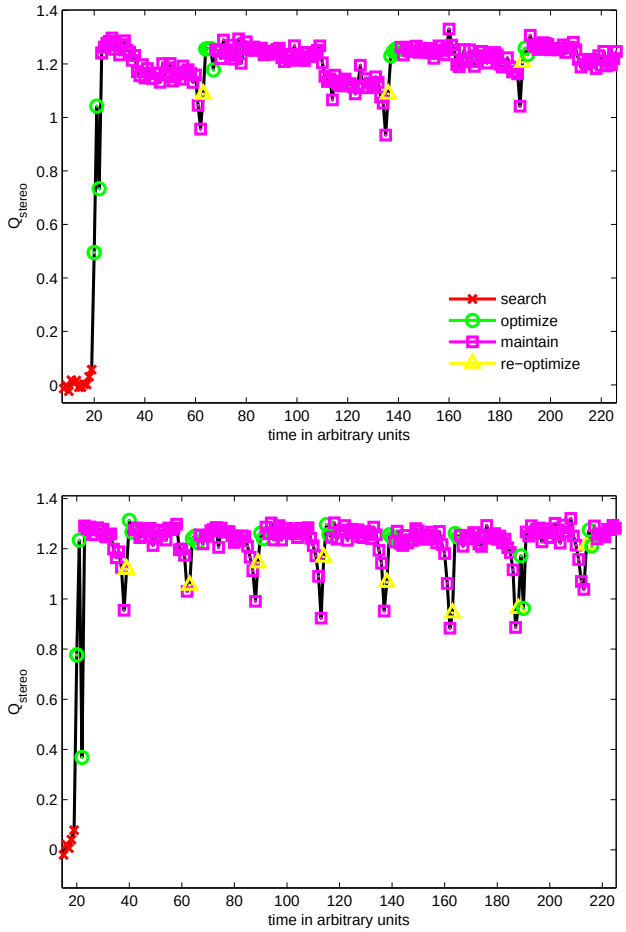

Figure 6.9: Visualisation of the positioning algorithm in the case of a periodically drifting (in time) and noisy quality profile. The parameters are set as in Tabl. 6.1, whereas the minimum quality Q_{mm}, see Sec. 6.3.4, was set to 0.5. In the top figure the drift velocity was set to $v_d = 0.75 \frac{\mu m}{\text{time step}}$, whereas in the bottom figure it was set to $v_d = 1.5 \frac{\mu m}{\text{time step}}$.

described in Sec. 6.2.2. If not stated otherwise, the default parameters listed in Tabl. 6.1 are used for the positioning algorithm. These values were obtained by testing the algorithm in various scenarios. As it turns out, they are quite similar to the values obtained in [148].

6.4.1 Static environment

This section serves as a demonstration and visualisation of the positioning algorithm. For this, the noise-free, flipped quality profile Q_{stereo} already shown in the first row of Fig. 6.5 was used. The positioning algorithm was executed with the parameter values summarised in Tabl. 6.1, and the results are shown in Fig. 6.8. As can be seen from this figure, the algorithm successfully finds a position close to the local maximum. Ideally, the algorithm would find the second local maximum, which is even the global maximum. This, however, would be only possible when some exploratory mechanism would be added, which, on the other hand, would cause additional electrode movement and, thus,

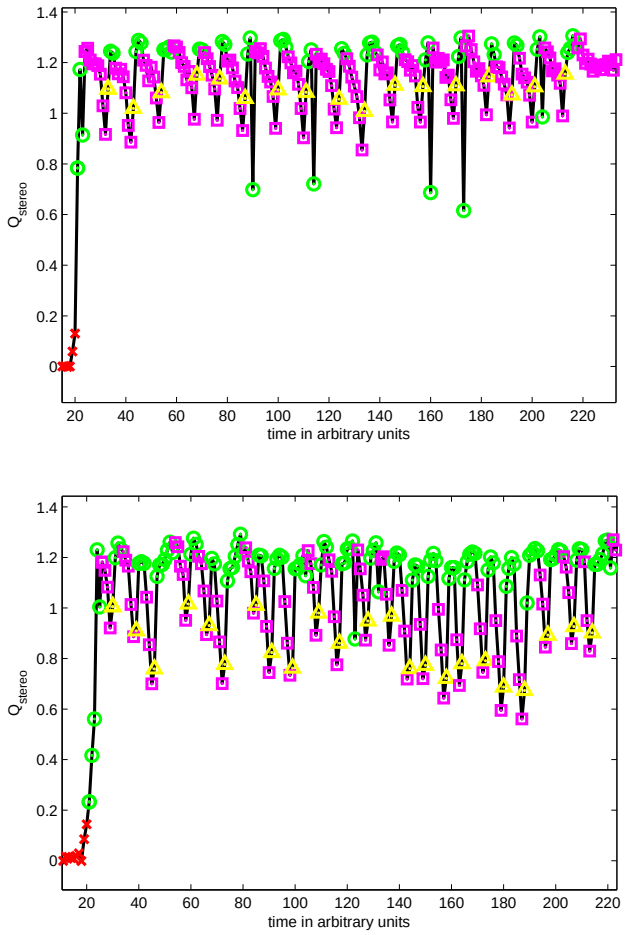

Figure 6.10: Visualization of the positioning algorithm in the case of a continuously drifting and noisy quality profile. Same notation and paramters as in Fig. 6.9 are used. In the top figure the drift velocity was set to $v_d = 0.75 \frac{\mu m}{\text{time step}}$, whereas in the bottom figure it was $v_d = 1.5 \frac{\mu m}{\text{time step}}$.

more tissue damage.

6.4.2 Drifting environment

The positioning system was also tested in a noisy and drifting environment. In particular, the quality profile shown in the last row of Fig. 6.5 was used, but the sampling resolution was increased to 0.1μm. At every time instance Gaussian zero mean noise with a standard deviation of 0.02 was added in order to mimic a real recording and the resulting uncertainty in the computed quality measure. Two scenarios were considered: In the first one a periodically occurring neuron drift was assumed. After an initial constant period of 30 time steps, a drift with a velocity of $v_d \frac{\mu m}{\text{time step}}$ and duration of 5 time steps followed by a constant period of 20 time steps occurred periodically. In the second scenario, after a constant period of 20 time steps, a continuous drift with a velocity of $v_d \frac{\mu m}{\text{time step}}$ occurred. A time step was defined as one cycle of the entire algorithm, i.e. as one iteration through all the stages shown in Fig. 6.1. A drift was mimicked by shifting the whole quality profile by v_d per time step. This corresponds to a tissue drift such that

only the distance on the x-axis between the tetrode and the neurons changes, while the distances between all neurons are not changed (i.e. 1-dimensional translation of a rigid configuration).

The results are shown in Fig. 6.9 and Fig. 6.10. For both configurations the proposed algorithm is able to follow the drift and to retain an acceptable recording quality. The maximum drift velocity the algorithm is able to handle depends on its parameter values and on the quality profile itself. For example, if the drift per time step is larger than the search step size S_s, the algorithm will never be able to find a good recording position. Even if there is a constant phase first, so that the algorithm finds a good initial position, a following drift with a high velocity might cause such a sudden quality drop, that immediately the "search" state is triggered, and the algorithm will not be able to track the drift.

6.5 Conclusion

We defined two quality measures for extracellular recordings, which indicate the detectability of spikes and their separability. These two measures did not require any spike clustering, but can be directly computed on unclassified spikes. Nevertheless, the proposed measures correspond well to the true qualities obtained if ground truth information were available.

Furthermore, we proposed a positioning algorithm whose goal is to find an optimal recording position. In particular, based on a stochastic approximation scheme, the quality measure is optimised until an optimal position is found. The positioning algorithm operates in an unsupervised manner, and its parameters can be determined from simulations. Their values agree well with the ones obtained in [148]. Furthermore, the numerous differences of our approach compared to the existing works with respect to the proposed quality measures and positioning algorithm were discussed at the corresponding places in the preceding sections. A summary of the key differences can be found in Tab. 6.2. In short, to our knowledge the proposed quality measure and positioning logic are the first ones especially designed for multi-channel electrodes in order to achieve recording positions with favourable information yield, in contrast to existing approaches

Table 6.2: Main differences between the existing approach presented in [148] and our proposed approach for unsupervised electrode positioning.

	Existing approach	Our approach
Developed for	obtaining single cell recordings for single channel electrodes	obtaining good neuron discriminability for multi-channel electrodes
Quality measure	SNR of dominant neuron (spike clustering required)	SNR of all neurons, and variance of SNR distribution (spike clustering *not* required)
Stochastic approx.	order estimation and global polynomial fitting	piecewise Hermite polynomials and final second order polynomial fit

which aim for single cell measurements.

The proposed positioning algorithm was run on realistically simulated data. It was able to find good recording positions in static environments as well as to retain sufficient quality in drifting environments. An application to real experiments should be the next step to take.

Appendices

Appendix A

Appendix to Chap. 3

A.1 Limits of integrand

Since $u = (1 - v)/v$, the two boundaries of $v = 0$ and $v = 1$ correspond to $u = \infty$ and $u = 0$. For the case of $u = \infty$ it is easy to see that (with the definition $\rho(u) = e^{r(u)}$)

$$\lim_{u\to\infty} \frac{\sin(\beta(u))}{u \cdot e^{r(u)}} = 0 \tag{A.1}$$

since the enumerator is bounded $\sin(\beta(u)) \leq 1$, and $\lim_{u\to\infty} r(u) = \infty$ (because $\theta_i \geq 0$ $\forall i$).

To prove the second limit, we use l'Hospital's rule, namely

$$\lim_{u\to 0} \frac{\sin(\beta(u))}{u \cdot e^{r(u)}} = \lim_{u\to 0} \frac{\frac{\partial}{\partial u}\sin(\beta(u))}{\frac{\partial}{\partial u}(u \cdot e^{r(u)})} = \lim_{u\to 0} \frac{\beta'(u) \cdot \cos(\beta(u))}{e^{r(u)} + u \cdot r'(u) \cdot e^{r(u)}}. \tag{A.2}$$

It is

$$\begin{aligned} \lim_{u\to 0} \beta(u) &= 0 \\ \lim_{u\to 0} \beta'(u) &= 1/2 \sum_i \lambda_i + \theta_i \lambda_i \\ \lim_{u\to 0} r(u) &= 0 \\ \lim_{u\to 0} r'(u) &= 0 \end{aligned} \tag{A.3}$$

hence

$$\lim_{u\to 0} \frac{\sin(\beta(u))}{u \cdot e^{r(u)}} = 1/2 \sum_i \lambda_i + \theta_i \lambda_i. \tag{A.4}$$

□

Appendix **B**

Appendix to Chap. 4

B.1 Threshold calculation with truncated Gaussians

The distribution of the rectified filter output $I(y^j)$ of filter $\boldsymbol{f}^j$ is proportional to

$$I(y^j) \sim \sum_i \sum_\tau \mathrm{tG}_\tau^{q^i, f^j} \tag{B.1}$$

where $\mathrm{tG}_\tau^{q^i, f^j} \sim \mathrm{tG}\left(\left(\boldsymbol{q}^i \star \boldsymbol{f}^j\right)_\tau, \sqrt{\boldsymbol{f}^{j\top} \boldsymbol{C} \boldsymbol{f}^j}, 0, \infty\right)$ is a truncated Normal distribution*. After applying the Deconfusion matrix, the threshold is obtained by minimising the false negative and the false positive probability, hence

$$\gamma_j = \underset{\gamma_j}{\mathrm{argmin}} \left\{ \beta_j \cdot \int_{-\infty}^{\gamma_j} \sum_i w_{j,i} \sum_{\tau=-\Delta}^{\Delta} \mathrm{tG}_\tau^{q^j, f^i} + \int_{\gamma_j}^{\infty} \sum_i w_{j,i} \sum_{k \neq j} \sum_\tau \beta_k \cdot \mathrm{tG}_\tau^{q^k, f^i} \right\} \tag{B.2}$$

where $\beta_j = 0.5$ and the β_k, $k \neq j$, are proportional to the firing frequency of neuron k such that $\sum_{k \neq j} \beta_j = 0.5$. The Δ is the tolerance zone in which a spike is still classified correctly (see Sec. 2.1.2). Note that in Eq. B.2 we neglected to include the region of the filter response outside the delta zone in the right expression, as the contribution of this region is usually quite small. The optimisation problem can be solved by a line search algorithm.

B.2 Literature overview

This section serves as a pointer to some of the existing literature in the fields of blind source separation and blind deconvolution.

B.2.1 Blind source separation

In [153] a survey of methods for blind source separation, especially when data is sparse, is given. Instantaneous, anechoic and echoic (convolutive) cases and as well as the over-

*The first argument being the mean, the second one being the standard deviation, followed by the lower and upper truncation bounds.

complete case are considered. Algorithms are divided into two classes: staged estimation of mixing parameters and estimation of sources, and joint estimation. However, the deflation approach is not considered.

The paper [160] also provides a survey of methods for blind source separation, considering especially non-instantaneous mixtures. Algorithms are grouped in two categories: higher-order statistics, and second order statistics (+ additional conditions). The advantage of second order methods is that in general they are less sensitive to noise and outliers, and are often computationally more efficient. The paper discusses well the problems of BSS in the frequency domain.

The survey paper [129] considers instantaneous and convolutive mixtures. As far as instantaneous mixtures are concerned, methods are discussed which are based on moments, contrast functions, deflation, or whitening plus rotation. In the case of convolutive mixtures, higher-order statistics, frequency approach, and second order statistics are considered.

The paper [40] makes a distinction between blind signal extraction and blind source separation. In the later case, all sources are extracted at the same time. In the former case, one aims only at extracting a certain number of sources. The paper proposes a unifying framework for both cases.

In [89] an algorithm for separating instantaneous sources from a linear mixture is presented. The mutual information is formulated in terms of Negentropy, i.e. the mutual information is minimised (independent sources) when the Negentropy is maximised. In order to allow for fast computation, the negentropy is approximated by an expression depending on a function G, for which almost any arbitrary non-quadratic function is allowed. A criterion how to choose this function optimally with regards to the signal distribution is presented. Finally, a newton method to find the maximum of the negentropy is presented. This leads to the following update rule:

$$\begin{aligned} \boldsymbol{w}_1 &= \boldsymbol{R}^{-1} \cdot \mathbb{E}\left[\boldsymbol{x} \cdot G'(\boldsymbol{w}^\top \boldsymbol{x})\right] - \mathbb{E}\left[G''(\boldsymbol{w}^\top \boldsymbol{x})\right] \cdot \boldsymbol{w} \\ \boldsymbol{w}_2 &= \frac{\boldsymbol{w}_1}{\sqrt{\boldsymbol{w}_1{}^\top \cdot \boldsymbol{R} \cdot \boldsymbol{w}_1}} \end{aligned} \tag{B.3}$$

which is quite similar to the SEA (see Sec. B.2.2, Eq. B.4) update rule.

The paper [237] solves the MIMO problem in the time domain, when there are more sensors than sources. The algorithm makes explicit use of the Toeplitz structure of the matrices. The FIR filters are estimated by minimizing a novel cost function. The cost function can be minimised efficiently by introducing a new variable, consequently the function is not quartic but only quadratic, and, hence, can be optimised efficiently. In comparison to other approaches, this is a one stage algorithm.

The work in [46] extends the FastICA (from [89]) algorithm to under-complete convolutive blind source separation, and works in the spatio-temporal domain. Two versions of the algorithm are proposed, one in which the sources are extracted sequentially (deflation) and one for parallel source extraction. In the deflation approach the error accumulates at each separation stage, therefore, parallel extraction should be preferred. Since the method is not gradient based, no step size has to be adapted, and the algorithm works well for a wide range of initial conditions. However, the method assumes spatially and temporally white signals. The filters are learnt with the FastICA algorithm together

with a heuristics. This ensures that the filters are paraunitary, which is an extension of the orthogonality constraints in FastICA.

In [28] the sources do not necessarily have to be independent and identical, but no noise in the model is assumed. A contrast function with a reference signal is used. An iterative approach is developed such that the reference signal is updated and does not have to be defined by the user. This approach has the advantage that the contrast function has to be optimised only in the non-reference variables, which allows for faster calculation. The filters correspond to the minimum power distortionless response beamformers, and the sources are extracted sequentially[†].

B.2.2 Blind channel estimation/identification/equalisation

A good overview is offered in [35] which treats the SISO as well as MIMO case. The paper divides the algorithms into two classes: implicit and explicit methods, i.e. whether they use higher-order statistics implicitly or explicitly. For example, Bussgang type algorithms (including the Sato and the constant modulus algorithm (CMA)) belong to the former category, whereas the super-exponential algorithm (SEA) to the latter. The IFC algorithm (gradient based), the SEA and the CMA algorithm are presented in more detail. Their relation to one another as well as their dis/advantages are pointed out. Several improvements for these algorithms are proposed: Hybrid algorithm of SEA and IFC, prewhitening of data, and smart initial condition (via simplified SEA algorithm).

Another survey paper is [208], in which, however, mostly only the SIMO case is considered.

A survey over some instantaneous blind source separation methods and some blind deconvolution methods is also given in [214], also comparing their respective performances.

In [211] the constant modulus algorithm is developed. Constant modulus means that the transmitted signal is of the form $|s(t)| = 1 \; \forall t$, whereas $s(t)$ is a complex valued signal. Hence, the output after equalisation must also have constant modulus which is directly incorporated into the algorithm. No further assumptions about the signal statistics are necessary.

In [189] an algorithm called super-exponential algorithm (SEA) is presented, which achieves blind equalisation by higher-order statistics computations, namely

$$\begin{aligned} \boldsymbol{w}_1 &= \boldsymbol{R}^{-1} \cdot \boldsymbol{d} \\ \boldsymbol{w}_2 &= \frac{\boldsymbol{w}_1}{\sqrt{\boldsymbol{w}_1{}^{\top} \cdot \boldsymbol{R} \cdot \boldsymbol{w}_1}} \end{aligned} \tag{B.4}$$

where $\boldsymbol{d}$ is a cross-cumulant between the filter output and the original data. The algorithm on page 49 in [222] is a special case of the super-exponential algorithm. In particular, p (from [189]) = d (from [222]), and q = 1.

In [193] blind filtering is done by two different approaches: Gradient based or Hessian based[‡]. However, the paper also cites literature which states that the cost function of the constant modulus algorithm (gradient based) is the same as the optimisation criterion

[†]The deflation approach is often also called hierarchical or multistage approach.

[‡]Which are just different terms for implicit and explicit methods.

for the super-exponential algorithm (Hessian based). The later converges much faster in the case of stationary sources, but also requires more computational cost. In the case of non-stationary sources, however, the gradient based method might be faster. The paper considers a convex combination of both approaches.

In [155] it is argued that when the signal is spiky and non-symmetric, it is better to learn filters based on maximisation of skewness instead of kurtosis. By better, the paper means that less iterations are needed to achieve convergence to the correct deconvolution filter. The filer is calculated via steepest ascent iterations, and resembles the MPDR filter. All algorithms are developed for SISO systems. Two different algorithms are presented: One in which after every step the filter must be normalised, and one in which this is approximately ensured by the adaptation step already.

In [132] an analysis of the convergence properties of a wide family of Bussgang blind deconvolution algorithms is conduced. Kurtosis and Skewness maximisation can be viewed as special cases of this more general approach.

In [183] another Bussgang type algorithm is presented, however, spikes are modelled as the sum of two Gaussian distributions.

The authors of [109] develop an algorithm for retrieving the waveform/channel response when only a single measurement is available, i.e. the signal appears only once (no repetition). This is a deterministic method using the z-transform and its greatest common divisor via eigenvalues calculation. The same authors presented a very similar method in [108], only the greatest common divisor is achieved in a different way.

Another deterministic algorithm for impulsive, i.e. non-repetitive, sources in the case of a SIMO system is presented in [169]. A version of the algorithm is developed when the channel response is sparse (not the source).

In [116] the super-exponential algorithm in the case of skewness and of kurtosis maximisation in a SISO system is compared with each other. It is concluded that skewness is better when the signal has an asymmetric distribution, whereas kurtosis is better when the signal changes abruptly. The paper also proposes a heuristics how to choose the filter length.

The super-exponential algorithm is modified in [97] in the sense that higher-order cumulants are used to estimate the template as well as the multiplication matrix (instead of second order statistics). This makes the algorithm less sensitive to Gaussian noise, but decreases a little the convergence speed and requires more computation time.

A different modification of the super-exponential algorithm is proposed in [233]. A SISO system and white Gaussian noise are assumed. The noise variance is first estimated using minimum description length. This estimate is then used to modify the Hessian matrix of the adaptive matched filter. The algorithm is computationally more efficient than [97], converges faster, and has better performance in noisy settings.

A deflation algorithm for MIMO system is presented in [96]. Sources can be temporally correlated, but have to be spatially uncorrelated. Furthermore, no noise is assumed in the model. The algorithm is very similar to the super-exponential algorithm, but is only exponential.

The authors of [239] come form the field of CDMA. In contrast to most other works, not the AMF but the MPDR is obtained. This is done by jointly optimising a cost func-

tion (basically the variance of the filter output) with respect to the filter and the steering vector/template, respecting some constraints. Several gradient based algorithms are proposed, and global convergence is shown. The methods aim at extracting a single source from a mixture of several sources and noise.

The minimum mean square error receiver (MMSE) solution is the MVDR solution in the case when the steering vector is perfectly known. Since, however, the steering vector is not known, all blind methods will be inferior. The authors of [212] show how the steering vector can be estimated blindly. In particular, the steering vector should be chosen such that the variance of the filter output is maximum under the unity constraint of the steering vector. The vector is then given by the smallest eigenvector of the inverse of the data covariance matrix.

The paper [238] modifies the method presented in [212]. The cost function is very similar, the only difference is that the data matrix is now to the power of m. The optimal detector for this problem would be the maximum likelihood sequence estimator, but it requires exponential computational load. A linear solution, like the proposed one, is sub-optimal, but computationally feasible. The optimal linear solution is called the minimum mean square error receiver (MMSE). The MOE (proposed in [212]) which is similar to the MVDR beamformer, provides an approximated solution that approaches the MMSE. The proposed method allows also for a noise estimation without using minimum description length techniques.

In [92] the SEA is extended to the MIMO case, but no noise is assumed. The paper proposes a two stage algorithm. First, SEA together with deflation is applied. This gives first estimates of the channel responses. From this, filters are constructed which are then used as initial condition for the second SEA algorithm. This second SEA algorithm is a straightforward extension of the SISO SEA algorithm. The advantage of deflation is its global convergence, however, a disadvantage is the error propagation. The second SEA algorithm has no error propagation, but does not exhibit global convergence. Thus, a combination of both seems to be a good choice.

In [76] discrete sources are assumed, but the method is designed for under-complete, instantaneous mixtures. The method is based on maximum a posteriori estimation. The sources and the mixing matrix are estimated simultaneously. A gradient based approach has to be applied in order to solve the optimisation problem. Multiple sources are extracted by deflation.

The authors of [112] consider the under-complete MIMO case, and when the source signals are finite alphabet random variables. It is assumed that the finite alphabet is known. The paper presents a cost function which must be minimised by gradient approaches in order to obtain the filters. The method even works when the sources are correlated. This is the main advantage provided by the finite alphabet information. In the case of uncorrelated sources, the sources can be extracted one by one (deflation) and the paper gives references to the literature concerned with how to estimate noise.

A method which aims at recovering binary sources from an over-complete MIMO system is presented in [114]. This algorithm, however, is deterministic and resembles more a clustering procedure, and the waveforms cannot be completely arbitrary. Multiple sources are extracted by a deflation procedure.

In [115] an algorithm for blind deconvolution for a MIMO system, when there are more sources M than senors N, is proposed. The algorithm can extract N sources which are true sources. The algorithm also works for analog sources, not only binary/discrete sources. It is a two stage algorithm, whereas in the first step a single source is extracted based on a contrast function involving fourth order cross moments. It is argued that a contrast function involving only fourth order moments like kurtosis (no cross moments) is unsuitable in the case of over complete systems. After one source is extracted, this source is deconvolved using a FIR filter, which is obtained by optimising a different contrast function.

The approach in [43] is clustering based and assumes a MISO system with binary sources. The technique is similar to the one presented in [242]. Although the paper deals with noisy systems, the method does not seem to be very robust, as it is formulated on the original raw data.

In [134] or [30], the waveform/template is known. The underlying process is assumed to be a combination of sparse Bernoulli and Gauss distribution, and must be estimated.

The authors of [47] analyse the relation between blind deconvolution (SISO) and blind source separation. It is shown that in the case of circulant matrices the algorithms for blind deconvolution and BSS are the same. In the limit of very large dimensions circulant matrices approach Toepiltz matrices.

Similarities between blind deconvolution and blind source separation are also pointed out in [5, 6]. However, the discussion is more from a point of view of understanding, i.e. qualitative rather than quantitative.

Bibliography

[1] E. Aboutanios and B. Mulgrew. Assessment of the single data set detection algorithms under template mismatch. In *Proc. Fifth IEEE International Symposium on Signal Processing and Information Technology*, pages 269–274, 2005. [cited at p. 41]

[2] S. Affes, S. Gazor, and Y. Grenier. Robust adaptive beamforming via LMS-like target tracking. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-94*, volume 4, pages 269–272, 1994. [cited at p. 32]

[3] M.I. Ageel. Variance upper bounds and a probability inequality for discrete alpha-unimodality. *Aplicationes Mathematicae*, 27:403 – 410, 2000. [cited at p. 30]

[4] T. Aksenova, O. Chibirova, O. Dryga, I. Tetko, A.L. Benabid, and A. Villa. An unsupervised automatic method for sorting neuronal spike waveforms in awake and freely moving animals. *Methods*, 30(2):178–187, Jun 2003. [cited at p. 44, 65, 71]

[5] R. Attux, A. Neves, L.T. Duarte, R. Suyama, C. Junqueira, L. Rangel, T.M. Dias, and J.M.T. Romano. On the relationships between blind equalization and blind source separation part i: Foundations. *Journal of Communication and Information Systems*, 22(1):41–52, 2007. [cited at p. 6, 117]

[6] R. Attux, A. Neves, L.T. Duarte, R. Suyama, C. Junqueira, L. Rangel, T.M. Dias, and J.M.T. Romano. On the relationships between blind equalization and blind source separation part ii: Relationships. *Journal of Communication and Information Systems*, 22(1):53–61, 2007. [cited at p. 6, 117]

[7] B. Baldessari. The distribution of a quadratic form of normal random variables. *The Annals of Mathematical Statistics*, 38(6):1700–1704, 1967. [cited at p. 41]

[8] I.N. Bankman, K.O. Johnson, and W. Schneider. Optimal detection, classification, and superposition resolution in neural waveform recordings. *IEEE Trans Biomed Eng*, 40(8):836–841, Aug 1993. [cited at p. 63]

[9] A. Bar-Hillel, A. Spiro, and E. Stark. Spike sorting: Bayesian clustering of non-stationary data. *J Neurosci Methods*, 157(2):303–316, Oct 2006. [cited at p. 63, 64]

[10] B. Baygun and A. Hero III. Optimal simultaneous detection and estimation under a false alarm constraint. *IEEE Transaction on Information Theory*, 41 (3):688 – 703, 1995. [cited at p. 19]

[11] R. Benitez and Z. Nenadic. Robust unsupervised detection of action potentials with probabilistic models. *IEEE Transactions on Biomedical Engineering*, 55(4):1344–1354, April 2008. [cited at p. 90]

[12] O. Besson, A.A. Monakov, and C. Chalus. Signal waveform estimation in the presence of uncertainties about the steering vector. *IEEE Transactions on Signal Processing*, 52(9):2432–2440, 2004. [cited at p. 41]

[13] O. Besson, L.L. Scharf, and F. Vincent. Matched direction detectors and estimators for array processing with subspace steering vector uncertainties. *IEEE Transactions on Signal Processing*, 53(12):4453–4463, 2005. [cited at p. 22, 41]

[14] O. Besson and F. Vincent. Performance analysis of beamformers using generalized loading of the covariance matrix in the presence of random steering vector errors. *IEEE Transactions on Signal Processing*, 53(2):452–459, 2005. [cited at p. 40]

[15] B. Bharath and V.S. Borkar. Stochastic approximation algorithms: Overview and recent trends. *Sadhana*, 24 (Part 4 & 5):425–452, 1999. [cited at p. 102]

[16] E. Biffi, D. Ghezzi, A. Pedrocchi, and G. Ferrigno. Development and validation of a spike detection and classification algorithm aimed at implementation on hardware devices. *Computational Intelligence and Neuroscience*, 2010, 2010. [cited at p. 64, 71, 92]

[17] R.S. Blum and K.F. McDonald. Analysis of STAP algorithms for cases with mismatched steering and clutter statistics. *IEEE Transactions on Signal Processing*, 48(2):301–310, Feb. 2000. [cited at p. 40]

[18] T. Borghi, R. Gusmeroli, A.S. Spinelli, and G. Baranauskas. A simple method for efficient spike detection in multiunit recordings. *J Neurosci Methods*, 163(1):176–180, Jun 2007. [cited at p. 71]

[19] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. [cited at p. 14]

[20] E.A. Branchaud. *A control system for positioning recording electrodes to isolate neurons in extracellular recordings*. PhD thesis, California Institute of Technology, 2006. [cited at p. 97, 103]

[21] E.A. Branchaud, J.W. Burdick, and R.A. Andersen. An algorithm for autonomous isolation of neurons in extracellular recordings. In *Proc. First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics BioRob 2006*, pages 939–945, 2006. [cited at p. 32, 45, 72, 78, 93, 97]

[22] E.A. Branchaud, J.G. Cham, Z. Nenadic, J.W. Burdick, and R.A. Andersen. A miniature robot for autonomous single neuron recordings. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1920 – 1926, 2005. [cited at p. 93]

[23] S. Broda and M. Paolella. Evaluating the density of ratios of noncentral quadratic forms in normal variables. *Computaional Statistics and Data Analysis*, 53:1264 – 1270, 2009. [cited at p. 41]

[24] R. Butler and M. Paolella. Uniform saddlepoint approximations for ratios of quadratic froms. Technical report, Department of Statistical Science, Southern Methodist University, 2007. [cited at p. 36]

[25] R. Butler and M. Paolella. Uniform saddlepoint approximations for ratios of quadratic forms. *Bernoulli*, 14:140 – 154, 2008. [cited at p. 34, 36]

[26] J.F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proceedings F: Radar and Signal Processing*, 140(6):362–370, Dec. 1993. [cited at p. 41]

[27] M.A. Carreira-Perpinan. Mode-finding for mixtures of gaussian distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1318–1323, Nov. 2000. [cited at p. 74]

[28] M. Castella, S. Rhioui, E. Moreau, and J. C. Pesquet. Quadratic higher order criteria for iterative blind separation of a MIMO convolutive mixture of sources. *IEEE Transactions on Signal Processing*, 55(1):218–232, Jan. 2007. [cited at p. 5, 114]

[29] J. Cham, E. Branchaud, Z. Nenadic, B. Greger, R. Andersen, and J. Burdick. Semi-chronic motorized microdrive and control algorithm for autonomously isolating and maintaining optimal extracellular action potentials. *J Neurophysiol*, 93(1):570–579, Jan 2005. [cited at p. 45, 93]

[30] F. Champagnat, Y. Goussard, and J. Idier. Unsupervised deconvolution of sparse spike trains using stochastic approximation. *IEEE Transactions on Signal Processing*, 44(12):2988–2998, 1996. [cited at p. 33, 117]

[31] Hsiao-Lung Chan, Ming-An Lin, Tony Wu, Shih-Tseng Lee, Yu-Tai Tsai, and Pei-Kuang Chao. Detection of neuronal spikes using an adaptive threshold based on the max-min spread sorting method. *J Neurosci Methods*, 172(1):112–121, Jul 2008. [cited at p. 71]

[32] Hsiao-Lung Chan, Tony Wu, Shih-Tseng Lee, Ming-An Lin, Shau-Ming He, Pei-Kuang Chao, and Yu-Tai Tsai. Unsupervised wavelet-based spike sorting with dynamic codebook searching and replenishment. *Neurocomputing*, 2009. (in press). [cited at p. 64]

[33] R. Chandra and L.M. Optican. Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network. *IEEE Trans Biomed Eng*, 44(5):403–412, May 1997. [cited at p. 44, 65]

[34] Xi Chen, A. R. Leyman, and Jun Fang. A geometric method for blind separation of digital signals with finite alphabets. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2007*, volume 3, pages III–941–III–944, 15–20 April 2007. [cited at p. 66]

[35] Chong-Yung Chi, Ching-Yung Chen, Chil-Horng Chen, and Chih-Chun Feng. Batch processing algorithms for blind equalization using higher-order statistics. *IEEE Signal Processing Magazine*, 20(1):25–49, Jan. 2003. [cited at p. 73, 114]

[36] Hansang Cho, D. Corina, J.F. Brinkley, G.A. Ojemann, and L.G. Shapiro. A new template matching method using variance estimation for spike sorting. In *Proc. 2nd International IEEE EMBS Conference on Neural Engineering*, pages 225–228, 16–19 March 2005. [cited at p. 63]

[37] J.H. Choi and T. Kim. Neural action potential detector using multi-resolution TEO. *Electronics Letters*, 38(12):541–543, 2002. [cited at p. 71]

[38] Joon Hwan Choi, Hae Kyung Jung, and Taejeong Kim. A new action potential detector using the MTEO and its effects on spike sorting systems at low signal-to-noise ratios. *IEEE Trans Biomed Eng*, 53(4):738–746, Apr 2006. [cited at p. 44, 51, 52, 53, 71]

[39] C. M. Coviello and L. H. Sibul. Blind source separation and beamforming: algebraic technique analysis. *IEEE Transactions on Aerospace and Electronic Systems*, 40(1):221–235, Jan 2004. [cited at p. 66]

[40] S.A. Cruces-Alvarez, A. Cichocki, and S. Amari. From blind signal extraction to blind instantaneous signal separation: criteria, algorithms, and stability. *IEEE Transactions on Neural Networks*, 15(4):859–873, 2004. [cited at p. 113]

[41] S. Dandapat and G.C. Ray. Spike detection in biomedical signals using midprediction filter. *Med Biol Eng Comput*, 35(4):354–360, Jul 1997. [cited at p. 71]

[42] M. Delescluse and C. Pouzat. Efficient spike-sorting of multi-state neurons using inter-spike intervals information. *J Neurosci Methods*, 150(1):16–29, Jan 2006. [cited at p. 44, 65]

[43] K.I. Diamantaras and T. Papadimitriou. Blind deconvolution of multi-input single-output systems with binary sources. *IEEE Transactions on Signal Processing*, 54(10):3720–3731, Oct. 2006. [cited at p. 91, 117]

[44] Weidong Ding and Jingqi Yuan. Spike sorting based on multi-class support vector machine with superposition resolution. *Med Biol Eng Comput*, 46(2):139–145, Feb 2008. [cited at p. 44]

[45] R.L. Dougherty, A. Edelman, and J.M. Hyman. Nonnegativity-, monotonicity-, or convexity-preserving cubic and quintic hermite interpolation. *Mathematics of Computation*, 52(186):471–494, 1989. [cited at p. 103]

[46] S.C. Douglas, M. Gupta, H. Sawada, and S. Makino. Spatio temporal FastICA algorithms for the blind separation of convolutive mixtures. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1511–1520, July 2007. [cited at p. 5, 113]

[47] S.C. Douglas and S. Haykin. On the relationship between blind deconvolution and blind source separation. In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, volume 2, pages 1591–1595, 2–5 Nov. 1997. [cited at p. 117]

[48] Lin Du, T. Yardibi, Jian Li, and P. Stoica. Review of user parameter-free robust adaptive beamforming algorithms. *Digital Signal Processing*, 19:567–582, 2009. [cited at p. 32]

[49] R. Eckhorn and U. Thomas. A new method for the insertion of multiple microprobes into neural and muscular tissue, including fiber electrodes, fine wires, needles and microsensors. *J Neurosci Methods*, 49(3):175–179, Sep 1993. [cited at p. 55]

[50] A. El-Keyi, T. Kirubarajan, and A.B. Gershman. Robust adaptive beamforming based on the kalman filter. *IEEE Transactions on Signal Processing*, 53(8):3032–3041, 2005. [cited at p. 32]

[51] Y.C. Eldar, A. Nehorai, and P.S. La Rosa. An expected least-squares beamforming approach to signal estimation with steering vector uncertainties. *IEEE Signal Processing Letters*, 13(5):288–291, 2006. [cited at p. 41]

[52] D.L. Evans and L.M. Leemis. Algorithms for computing the distributions of sums of discrete random variables. *Mathematical and Computer Modelling*, 40(13):1429–1452, 2004. [cited at p. 41]

[53] R. Evans, T. Fortmann, and A. Cantoni. Envelope-constrained filters–i: Theory and applications. *IEEE Transactions on Information Theory*, 23(4):421–434, 1977. [cited at p. 19]

[54] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006. [cited at p. 15]

[55] M.S. Fee, P.P. Mitra, and D. Kleinfeld. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability. *J Neurosci Methods*, 69(2):175–188, Nov 1996. [cited at p. 44, 51, 65]

[56] M.S. Fee, P.P. Mitra, and D. Kleinfeld. Variability of extracellular spike waveforms of cortical neurons. *J Neurophysiol*, 76(6):3823–3833, Dec 1996. [cited at p. 61]

[57] N.I. Fisher and J.S. Marron. Mode testing via excess mass estimate. *Biometrika*, 88:499–517, 2001. [cited at p. 90]

[58] E. Fishler and H. Messer. Detection and parameter estimation of a transient signal using order statistics. *IEEE Transactions on Signal Processing*, 48, 2000. [cited at p. 19]

[59] G. Forchini. The exact cumulative distribution function of a ratio of quadratic forms in normal variables with application to the AR(1) model. *Econometric Theory*, 18:823–852, 2002. [cited at p. 41]

[60] G. Forchini. The distribution of a ratio of quadratic forms in noncentral normal variables. *Communications in Statistics. Theory and Methods*, 34:999 – 1008, 2005. [cited at p. 34, 36, 41]

[61] B. Friedlander and B. Porat. Performance analysis of transient detectors based on linear data transformations. *IEEE Transactions on Information Theory*, 38 (2):665–673, 1992. [cited at p. 20]

[62] F. Gamboa and E. Gassiat. Source separation when the input sources are discrete or have constant modulus. *IEEE Transactions on Signal Processing*, 45(12):3062–3072, Dec. 1997. [cited at p. 66]

[63] S. Gazor, S. Affes, and Y. Grenier. Robust adaptive beamforming via target tracking. *IEEE Transactions on Signal Processing*, 44(6):1589–1593, June 1996. [cited at p. 32, 33]

[64] J.E. Gentle and W. Haerdle. *Handbook of Computational Statistics - Concepts and Methods*. Springer, 2004. [cited at p. 102]

[65] P. Georgiev, F. Theis, and A. Chichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4):992 – 996, 2005. [cited at p. 66]

[66] A.B. Gershman. Robust adaptive beamforming: an overview of recent trends and advances in the field. In *Proc. 4th International Conference on Antenna Theory and Techniques*, volume 1, pages 30–35, 9–12 Sept. 2003. [cited at p. 32]

[67] A.B. Gershman and N.D. Sidiropoulos, editors. *Space-time processing for MIMO communications*. John Wiley & Sons, 2005. [cited at p. 6]

[68] A.G. Glen, D.L. Evans, and L.M. Leemis. APPL: A probabiliy programming language. *The American Statistican*, 55:156 – 166, 2001. [cited at p. 41]

[69] A.G. Glen, L.M. Leemis, and J.H. Drew. Computing the distribution of the product of two continuous random variables. *Computational Statistics and Data Analysis*, 44:451–464, 2004. [cited at p. 41]

[70] D.H. Glueck and K.E. Muller. On the trace of a wishart. *Communications in statistics. Theory and methods*, 27:2137–2141, 1998. [cited at p. 41]

[71] F. Goetze and A.N. Tikhomirov. Asymptotic distribution of quadratic forms. *Annals of probability*, 27(2):1072–1098, 1999. [cited at p. 41]

[72] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.2. 2008. [cited at p. 14]

[73] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. [cited at p. 14]

[74] M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In L. Liberti and N. Maculan, editors, *Global Optimization: From Theory to Implementation*, Nonconvex Optimization and its Applications, pages 155–210. Springer, 2006. [cited at p. 14]

[75] C. M. Gray, P. E. Maldonado, M. Wilson, and B. McNaughton. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *J Neurosci Methods*, 63(1-2):43–54, Dec 1995. [cited at p. 44, 71, 93]

[76] O. Grellier and P. Comon. Blind separation of discrete sources. *IEEE Signal Processing Letters*, 5(8):212–214, Aug. 1998. [cited at p. 116]

[77] J. Gurland. Distribution of quadratic forms and ratios of quadratic forms. *The Annals of Mathematical Statistics*, 24:416 – 427, 1953. [cited at p. 41]

[78] A.M. Haas, M.H. Cohen, and P.A. Abshire. Real-time variance based template matching spike sorting system. In *Proc. IEEE/NIH Life Science Systems and Applications Workshop LISA 2007*, pages 104–107, 8–9 Nov. 2007. [cited at p. 63]

[79] K.D. Harris, D.A. Henze, J. Csicsvari, H. Hirase, and G. Buzsaki. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J Neurophysiol*, 84(1):401–414, Jul 2000. [cited at p. 44, 56]

[80] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2008. [cited at p. 97]

[81] M.H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, 1996. [cited at p. 54]

[82] J.A. Herbst, S. Gammeter, D. Ferrero, and R.H.R. Hahnloser. Spike sorting with hidden markov models. *Journal of Neuroscience Methods*, 174:126–134, 2008. [cited at p. 64]

[83] M.L. Hines, A.P. Davison, and E. Muller. Neuron and python. *Frontiers in Neuroinformatics*, 3, 2009. [cited at p. 94]

[84] M.J. Hinich. Higher order cumulants and cumulant spectra. *Circuits, Systems, and Signal Processing*, 13(4):391–402, 1994. [cited at p. 10]

[85] G.R. Holt. *A critical reexamination of some assumptions and implications of cable theory in neurobiology*. PhD thesis, California Institute of Technology, Pasadena, CA, 1997. [cited at p. 94, 95]

[86] Yikun Huang and J. Miller. Phased-array processing for spike discrminination. *Journal of Neurophysiology*, 92:1944 – 1957, 2004. [cited at p. 65]

[87] E. Hulata, R. Segev, and E. Ben-Jacob. A method for spike sorting and detection based on wavelet packets and shannon's mutual information. *J Neurosci Methods*, 117(1):1–12, May 2002. [cited at p. 44, 65]

[88] N.P. Hurley and S.T. Rickard. Comparing measures of sparsity. *CoRR*, abs/0811.4706, 2008. [cited at p. 74]

[89] A. Hyvaerinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, May 1999. [cited at p. 113]

[90] A. Hyvaerinen and Y. Kano. Independent component analysis for nonnormal factor analysis. *New Developments in Psychometrics*, pages 649–656, 2003. [cited at p. 6]

[91] J.P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48:419 – 426, 1961. [cited at p. 34, 41]

[92] Y. Inouye and K. Tanebe. Super-exponential algorithms for multichannel blind deconvolution. *IEEE Transactions on Signal Processing*, 48(3):881–888, March 2000. [cited at p. 76, 116]

[93] M.C. Jones, J.S. Marron, and S.J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996. [cited at p. 75]

[94] S.Z. Kalson. An adaptive array detector with mismatched signal rejection. *IEEE Transactions on Aerospace and Electronic Systems*, 28(1):195–207, Jan. 1992. [cited at p. 19, 23, 24, 40]

[95] J. Karvanen and A. Cichocki. Measuring sparseness of noisy signals. *4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 125–130, 2003. [cited at p. 74]

[96] M. Kawamoto and Y. Inouye. A deflation algorithm for the blind source-factor separation of MIMO-FIR channels driven by colored sources. *IEEE Signal Processing Letters*, 10(11):343–346, Nov. 2003. [cited at p. 115]

[97] M. Kawamoto, M. Ohata, K. Kohno, Y. Inouye, and A.K. Nandi. Robust super-exponential methods for blind equalization in the presence of gaussian noise. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 52(10):651–655, Oct. 2005. [cited at p. 115]

[98] S.M. Kay. *Fundamentals of statistical signal processing Vol. 2 - Detection theory*. Prentice-Hall, 1998. [cited at p. 19]

[99] E.J. Kelly. Performance of an adaptive detection algorithm; rejection of unwanted signals. *IEEE Transactions on Aerospace and Electronic Systems*, 25(2):122–133, March 1989. [cited at p. 39]

[100] Kyung Hwan Kim and Sung June Kim. Method for unsupervised classification of multiunit neural signal recording under low signal-to-noise ratio. *IEEE Trans Biomed Eng*, 50(4):421–431, Apr 2003. [cited at p. 44, 64]

[101] Kyung Hwan Kim and Sung June Kim. A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio. *IEEE Trans Biomed Eng*, 50(8):999–1011, Aug 2003. [cited at p. 71]

[102] Sunghan Kim and J. McNames. Automatic spike detection based on adaptive template matching for extracellular neural recordings. *J Neurosci Methods*, 165(2):165–174, Sep 2007. [cited at p. 32, 71, 76, 80]

[103] Z. Koldovskỳ and P. Tichavskỳ. Time-domain blind audio source separation using advanced component clustering and reconstruction. *Proceedings of Joint Workshop Hand-free Speech Communication and Microphone Arrays*, 2008. [cited at p. 91]

[104] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, 13(4):67–94, July 1996. [cited at p. 5]

[105] S.M. Kuo, B.H. Lee, and W. Tian. *Real-Time Digital Signal Processing*. John Wiley & Sons, 2006. [cited at p. 10]

[106] D. Langlois, S. Chartier, and D. Gosselin. An introduction to independent component analysis: Infomax and FastICA algorithms. *Tutorials in Quantitative Methods for Psychology*, 6:31–38, 2010. [cited at p. 6]

[107] H. Lebret and S. Boyd. Antenna array pattern synthesis via convex optimization. *IEEE Transactions on Signal Processing*, 45(3):526–532, 1997. [cited at p. 16]

[108] P. Lecumberri, M. Gomez, and A. Carlosena. Multichannel blind deconvolution of impulsive signals. In *EUSIPCO*, 2005. [cited at p. 115]

[109] P. Lecumberri, M. Gomez, and A. Carlosena. Multichannel blind deconvolution of transient impulsive signals. In *Proc. IEEE Instrumentation and Measurement Technology Conference IMTC 2006*, pages 1145–1150, 24–27 April 2006. [cited at p. 115]

[110] M. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9(4), 1998. [cited at p. 44, 62]

[111] Ta-Hsin Li. Blind indentification and deconvolution of linear systems driven by binary random sequences. *IEEE Transactions on Information Theory*, 38(1):26 – 38, 1992. [cited at p. 66]

[112] Ta-Hsin Li. Finite-alphabet information and multivariate blind deconvolution and identification of linear systems. *IEEE Transactions on Information Theory*, 49(1):330–337, Jan. 2003. [cited at p. 66, 91, 116]

[113] X. Li and M. Liu. A near optimum detection in alpha-stable impulsive noise. *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3305–3308, 2009. [cited at p. 5]

[114] Y. Li, A. Cichocki, and L. Zhang. Blind source estimation of FIR channels for binary sources: a grouping decision approach. *Signal Processing*, 84(12):2245–2263, 2004. [cited at p. 91, 116]

[115] Yuanqing Li, Jun Wang, and A. Cichocki. Blind source extraction from convolutive mixtures in ill-conditioned multi-input multi-output channels. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(9):1814–1822, Sept. 2004. [cited at p. 117]

[116] Yujun Li, Peter W. Tse, and Xiaojuan Wang. Recovery of vibration signal based on a super-exponential algorithm. *Journal of Sound and Vibration*, 311:537–553, 2008. [cited at p. 5, 74, 81, 115]

[117] O. Lieberman. A Laplace approximation to the moments of a ratio of quadratic forms. *Biometrika*, 81(4):681–690, 1994. [cited at p. 34, 41]

[118] W. S. Lu. Minimax design of nonlinear-phase FIR filters: a least-pth approach. In *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2002*, volume 1, pages I–409–I–412, 26–29 May 2002. [cited at p. 17]

[119] Zeng-Hua Lu. The numerical evaluation of the probability density function of a quadratic form in normal variables. *Computaional Statistics and Data Analysis*, 51:1986–1996, 2006. [cited at p. 41]

[120] Z.H. Lu and M.L. King. Improving the numerical technique for computing the accumulated distribution of a quadratic form in normal variables. *Econometric Reviews*, 21(2):149–165, 2002. [cited at p. 41]

[121] D. Luengo, I. Santamaria, and L. Vielva. A general solution to blind inverse problems for sparse input signals. *Neurocomputing*, 69:198 – 215, 2005. [cited at p. 66, 91]

[122] H. Luetkepohl. *Handbook of Matrices*. John Wiley and Sons, 1996. [cited at p. 29]

[123] R. Lugannani and S. Rice. Saddle point approximation for the distribution of the sum of independent random variables. *Advances in Applied Probability*, 12(2):475–490, 1980. [cited at p. 41]

[124] A. Maccione, M. Gandolfo, P. Massobrio, A. Novellino, S. Martinoia, and M. Chiappalone. A novel algorithm for precise identification of spikes in extracellularly recorded neuronal signals. *Journal of Neuroscience Methods*, 177(1):241–249, 2009. [cited at p. 71]

[125] J.R. Magnus. The exact moments of a ratio of quadratic forms in normal variables. *Annales d'economie et de statistique*, 4:95–109, 1986. [cited at p. 41]

[126] Z.F Mainen and T.J Sejnowski. Influence of dendritic structure on firing pattern in model neocortical neurons. *Nature*, 382(6589):363–366, 1996. [cited at p. 94]

[127] E. Mammen, J.S. Marron, and N.I. Fisher. Some asymptotics for multimodality tests based on kernel density estimates. *Probability Theory and Related Fields*, 91(1):115–132, 1992. [cited at p. 90]

[128] D.G. Manolakis, V.K. Ingle, and S.M. Kogon. *Statistical and Adaptive Signal Processing*. ARTECH HOUSE, 2005. [cited at p. 6]

[129] A. Mansour, A.K. Barros, and N. Ohnishi. Blind separation of sources: Methods, assumptions and applications. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E83-A:1498–1512, 2000. [cited at p. 113]

[130] P. Marsh. Saddlepoint approximations for noncentral quadratic forms. *Economic Theory*, 14:539 – 559, 1998. [cited at p. 34]

[131] F. Marvasti, A. Amini, F. Haddadi, M. Soltanolkotabi, B.H. Khalaj, A. Aldroubi, S. Holm, S. Sanei, and J.A. Chambers. A unified approach to sparse signal processing. *CoRR*, abs/0902.1853, 2009. [cited at p. 91]

[132] H. Mathis and S. C. Douglas. Bussgang blind deconvolution for impulsive signals. *IEEE Transactions on Signal Processing*, 51(7):1905–1915, July 2003. [cited at p. 115]

[133] E.M. Maynard, C.T. Nordhausen, and R.A. Normann. The utah intracortical electrode array: A recording structure for potential brain-computer interfaces. *Electroencephalography and clinical Neurophysiology*, 102(3):228–239, 1997. [cited at p. 2]

[134] V. Mazet, D. Brie, and C. Caironi. Sparse spike train deconvolution using the hunt filter and a thresholding method. *IEEE Signal Processing Letters*, 11(5):486–489, 2004. [cited at p. 117]

[135] K.F. McDonald and R.S. Blum. Analytical analysis of STAP algorithms for cases with mismatched steering and clutter statistics. In *Proc. Record of the 1999 IEEE Radar Conference*, pages 267–272, 20–22 April 1999. [cited at p. 40]

[136] K.C. McGill. Optimal resolution of superimposed action potentials. *IEEE Trans Biomed Eng*, 49(7):640–650, Jul 2002. [cited at p. 44]

[137] http://github.com/pmeier82/Neural-Simulation Meier, P. [cited at p. 95]

[138] W.L. Melvin. A STAP overview. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):19–35, 2004. [cited at p. 5, 19, 47]

[139] J.M. Mendel. Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305, 1991. [cited at p. 10]

[140] C.E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, 1978. [cited at p. 15]

[141] D. Middleton and R. Esposito. Simultaneous optimum detection and estimation of signals in noise. *IEEE Transactions on Information Theory*, 14, 1968. [cited at p. 19]

[142] M.C. Minnotte. Nonparametric testing of the existance of modes. *The Annals of Statistics*, 25(4):1646 – 1660, 1997. [cited at p. 90]

[143] M.C. Minnotte, D.J. Marchette, and E.J. Wegman. The bumpy road to the mode forest. *Journal of Computational and Graphical Statistics*, 7(2):239–251, 1998. [cited at p. 90]

[144] N. Mtetwa and L. Smith. Smoothing and thresholding in neuronal spike detection. *Neurocomputing*, 69:1366–1370, 2006. [cited at p. 51, 71]

[145] S. Mukhopadhyay and G.C. Ray. A new interpretation of nonlinear energy operator and its efficacy in spike detection. *IEEE Trans Biomed Eng*, 45(2):180–187, Feb 1998. [cited at p. 65, 71, 82]

[146] D.W. Muller and G. Sawitzki. Excess mass estimates and tests for multimodality. *Journal of the American Statistical Association*, 86:738 – 746, 1991. [cited at p. 90]

[147] Z. Nenadic and J. Burdick. Spike detection using the continuous wavelet transform. *IEEE Trans Biomed Eng*, 52(1):74–87, Jan 2005. [cited at p. 71, 80, 81, 82, 90, 96]

[148] Z. Nenadic and J. Burdick. A control algorithm for autonomous optimization of extracellular recordings. *IEEE Transactions on Biomedical Engineering*, 53(5):941 – 955, 2006. [cited at p. 93, 94, 96, 97, 101, 103, 106, 108, 140]

[149] www.neuron.yale.edu NEURON. [cited at p. 94]

[150] F. Nicolls. *Constraints and invariance in target detection*. PhD thesis, University of Cape Town, 2000. [cited at p. 19]

[151] S. Nordebo and Z. Zang. Semi-infinite linear programming: A unfied approach to digital filter design with time- and frequency-domain specifications. *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, 46(6):765 – 775, 1999. [cited at p. 18]

[152] I. Obeid and P.D. Wolf. Evaluation of spike-detection algorithms for a brain-machine interface application. *IEEE Trans Biomed Eng*, 51(6):905–911, Jun 2004. [cited at p. 44, 51, 70]

[153] P.D. O'Grady, B.A. Pearlmutter, and S.T. Rickard. Survey of sparse and non-sparse methods in source separation. *International Journal of Imaging Systems and Technology*, 15(1):18–33, 2005. [cited at p. 91, 112]

[154] J. O'Keefe and M.L. Recce. Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus*, 3(3):317–330, Jul 1993. [cited at p. 43]

[155] P. Paeaejaervi. *Adaptive blind deconvolution using third-order moments: Exploiting asymmetry*. PhD thesis, Lulea University of Technology, Sweden, 2005. Licentiate Thesis. [cited at p. 74, 115]

[156] M. Paolella. Computing moments of ratios of quadratic forms in normal variables. *Computational Statistics and Data Analysis*, 42:313 – 331, 2003. [cited at p. 41]

[157] T.W. Parks and C.S. Burrus. *Digital Filter Design*. John Wiley & Sons, 1987. [cited at p. 9, 10]

[158] C. Paterson, R. Curry, A. Purvis, and S. Johnson. Detection of action potentials in the presence of noise using phase-space techniques. *World Academy of Science, Enineering and Technology*, 44:76–79, 2008. [cited at p. 71]

[159] A. Pazienti and S. Gruen. Robustness of the significance of spike synchrony with respect to sorting errors. *Journal of Computational Neuroscience*, 21(3):329–342, 2006. [cited at p. 78]

[160] M.S. Pedersen, J. Larsen, U. Kjems, and L.C. Parra. *A Survey of Convolutive Blind Source Separation Methods*. Springer Handbook of Speech Processing. Springer Press, Nov 2007. [cited at p. 6, 20, 66, 113]

[161] K.H. Pettersen, A. Devor, I. Ulbert, A.M Dale, and G.T. Einevoll. Current-source density estimation based on inversion of electrostatic forward solution: Effects of finite extent of neuronal activity and conductivity discontinuities. *Journal of Neruscience Methods*, 154:116 – 133, 2006. [cited at p. 94]

[162] K.H. Pettersen and G.T. Einevoll. Amplitude variability and extracellular low-pass filtering of neuronal spikes. *Biophysical Journal*, 94:784 – 802, 2008. [cited at p. 94, 95]

[163] K.H. Pettersen and G.T. Einevoll. Neurophysics: what the telegrapher's equation has taught us about the brain. *An anthology of developments in clinical engineering and bioimpedance: Festschrift for Sverre Grimnes*, 2009. [cited at p. 94]

[164] K.H. Pettersen, E. Hagen, and G.T. Einevoll. Estimation of population firing rates and current source densities from laminar electrode recordings. *Journal of Computational Neuroscience*, 24:291 – 313, 2008. [cited at p. 94]

[165] B. Porat and B. Friedlander. Performance analysis of a class of transient detection algorithms - a unified framework. *IEEE Transactions on Signal Processing*, 40 (10), 1992. [cited at p. 20]

[166] A. Potchinkov and R. Reemtsen. The design of FIR filters in the complex plane by convex optimisation. *Signal Processing*, 46:127 – 146, 1995. [cited at p. 16]

[167] C. Pouzat, M. Delescluse, P. Viot, and J. Diebolt. Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: a markov chain monte carlo approach. *J Neurophysiol*, 91(6):2910–2928, Jun 2004. [cited at p. 44, 64]

[168] C. Pouzat, O. Mazor, and G. Laurent. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *J Neurosci Methods*, 122(1):43–57, Dec 2002. [cited at p. 4, 12, 52, 59, 65, 72, 77, 97]

[169] Wanzhi Qiu, Syed Khsro Saleem, and Minh Pham. Blind identification of multichannel systems driven by impulsive signals. *Digital Signal Processing*, 2009. [cited at p. 5, 115]

[170] Meng-Hong Qu and Ying Wu. A steering vector estimation algorithm in uncalibrated arrays. In *Proc. 3rd International Conference on Innovative Computing Information and Control ICICIC '08*, pages 132–132, 18–20 June 2008. [cited at p. 41]

[171] R.Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput*, 16(8):1661–1687, Aug 2004. [cited at p. 57]

[172] S. Rao, J.C. Sanchez, Seungju Han, and J.C. Principe. Spike sorting using non parametric clustering via cauchy schwartz pdf divergence. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2006*, volume 5, pages V–V, 14–19 May 2006. [cited at p. 63]

[173] S.P. Rebrik, B.D. Wright, A.A .Emondi, and K.D. Miller. Cross-channel correlations in tetrode recordings: implications for spike-sorting. *Neurocomputing*, 2627:1033–8, 1999. [cited at p. 44]

[174] C.W. Reed and K. Yao. Performance of blind beamforming algorithms. In *Proc. Ninth IEEE SP Workshop on Statistical Signal and Array Processing*, pages 256–259, 14–16 Sept. 1998. [cited at p. 66]

[175] C.D. Richmond. Performance of a class of adaptive detection algorithms in nonhomogeneous environments. *IEEE Transactions on Signal Processing*, 48(5):1248–1262, May 2000. [cited at p. 40]

[176] F. Riera-Palou, J.M. Noras, and D. Cruickshank. Segmented equalizers with dynamic length selection. *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, 2001*, 2:951–958, 2001. [cited at p. 81]

[177] F.C. Robey, D.R. Fuhrmann, E.J. Kelly, and R. Nitzberg. A CFAR adaptive matched filter detector. *IEEE Transactions on Aerospace and Electronic Systems*, 28(1):208–216, 1992. [cited at p. 22]

[178] E.A. Robinson and S. Treitel. *Geophysical Signal Analysis*. Prentice Hall, 1980. [cited at p. 45, 46]

[179] K. Roeder. A graphical technique for determining the number of components in a mixture of normals. *Journal of the American Statistical Association*, 89(426):487–495, 1994. [cited at p. 75]

[180] U. Rutishauser, E. Schuman, and A. Mamelak. Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo. *J Neurosci Methods*, 154(1-2):204–224, Jun 2006. [cited at p. 44, 52, 57, 59, 61, 64, 97]

[181] M. Sahani, J.S. Pezaris, and R.A. Andersen. On the separation of signals from neighboring cells in tetrode recordings. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 222–228, Cambridge, MA, USA, 1998. MIT Press. [cited at p. 4, 72]

[182] Y. Sakurai and S. Takahashi. Dynamic synchrony of firing in the monkey prefrontal cortex during working-memory tasks. *J Neurosci*, 26(40):10141–10153, Oct 2006. [cited at p. 44]

[183] I. Santamaria-Caballero, C. Pantaleon-Prieto, F. Diaz-de Maria, and A. Artes-Rodriguez. A new inverse filter criterion for blind deconvolution of spiky signals using gaussian mixtures. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-96*, volume 3, pages 1680–1683, 7–10 May 1996. [cited at p. 115]

[184] D.P. Scholnik. Mixed-norm FIR filter optimization using second-order cone programming. In *ICASSP*, 2002. [cited at p. 16]

[185] D. Schuhmacher, B.T. Vo, and B.N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, 2008. [cited at p. 27]

[186] H. Schulze and C. Lueders. *Theory and applications of OFDM and CDMA*. John Wiley & Sons, 2005. [cited at p. 6]

[187] S. Shahid and L.S. Smith. A novel technique for spike detection in extracellular neurophysiological recordings using cepstrum of bispectrum. *Proceedings of the 16th European Signal Processing Conference (EUSIPCO-08)*, 2008. [cited at p. 71, 82, 89]

[188] S. Shahid, J. Walker, and L.S. Smith. A new spike detection algorithm for extracellular neural recordings. *IEEE Transactions on Biomedical Engineering*, 57(4):853 – 866, 2010. [cited at p. 71, 82, 89]

[189] O. Shalvi and E. Weinstein. Super-exponential methods for blind deconvolution. *IEEE Transactions on Information Theory*, 39(2):504–519, March 1993. [cited at p. 5, 27, 73, 74, 114]

[190] B.A. Shenoi. *Introduction to Digital Signal Processing and Filter Design*. John Wiley & Sons, 2006. [cited at p. 9]

[191] S. Shoham, M. Fellows, and R. Normann. Robust, automatic spike sorting using mixtures of multivariate t-distributions. *J Neurosci Methods*, 127(2):111–122, Aug 2003. [cited at p. 44, 65]

[192] D. Shpak. Design of mixed-norm FIR filters using an unconstrained least-pth algorithm. *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1:253 – 255, 2003. [cited at p. 18]

[193] M. T. M. Silva and V. H. Nascimento. Improving the tracking capability of adaptive filters via convex combination. *IEEE Transactions on Signal Processing*, 56(7):3137–3149, July 2008. [cited at p. 114]

[194] B.W. Silverman. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society. Series B (Methodological)*, 43(1):97 – 99, 1981. [cited at p. 90]

[195] M.I. Skolnik. *Introduction to Radar Systems*. McGraw-Hill Book, 1981. [cited at p. 5]

[196] J.C. Spall. *Introduction to Stochastic Search and Optimization*. JOHN WILEY & SONS, 2003. [cited at p. 102, 103]

[197] O. Sporns, D.R. Chialvo, M. Kaiser, and C.C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, 9(9):418–425, 2004. [cited at p. 2]

[198] J.P. Stitt, R.P. Gaumond, J.L. Frazier, and F.E. Hanson. A comparison of neural spike classification techniques. In *Proc. 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 3, pages 1092–1094, 30 Oct.–2 Nov. 1997. [cited at p. 63]

[199] A. Swami, J.M. Mendel, and C.L. Nikias. *Higher-Order Spectral Analysis Toolbox User's Guide*. United Signal and Systems, Inc., 2001. [cited at p. 10, 79]

[200] S. Takahashi, Y. Anzai, and Y. Sakurai. Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes. *J Neurophysiol*, 89(4):2245–2258, Apr 2003. [cited at p. 44, 65]

[201] S. Takahashi and Y. Sakurai. Real-time and automatic sorting of multi-neuronal activity for sub-millisecond interactions in vivo. *Neuroscience*, 134(1):301–315, 2005. [cited at p. 45]

[202] S. Takahashi, Y. Sakurai, M. Tsukuda, and Y. Anzai. Classification of neural activities using independent component analysis. *Neurocomputing*, 49:289–298, 2002. [cited at p. 45]

[203] T. Takekawa, Y. Isomura, and T. Fukai. Accurate spike sorting for multi-unit recordings. *European Journal of Neuroscience*, 31:263–272, 2010. [cited at p. 64, 92]

[204] P. Thakur, H. Lu, S. Hsiao, and K. Johnson. Automated optimal detection and classification of neural action potentials in extra-cellular recordings. *J Neurosci Methods*, 162(1-2):364–376, May 2007. [cited at p. 20, 32, 44, 45]

[205] http://www.thomasrecording.com Thomas Recording GmbH. [cited at p. 2]

[206] Z. Tiganj and M. Mboup. Spike detection and sorting: Combining algebraic differentiations with ICA. *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, pages 475–482, 2009. [cited at p. 65, 71]

[207] S. Timofeev, A.R.S. Bahai, and P. Varaiya. Adaptive acoustic beamformer with source tracking capabilities. *IEEE Transactions on Signal Processing*, 56(7):2812–2820, 2008. [cited at p. 32]

[208] Lang Tong and S. Perreau. Multichannel blind identification: from subspace to maximum likelihood methods. *Proceedings of the IEEE*, 86(10):1951–1968, Oct. 1998. [cited at p. 114]

[209] L. Traver, C. Tarin, P. Marti, and N. Cardona. Adaptive-threshold neural spike detection by noise-envelope tracking. *Electronics Letters*, 43(24), 2007. [cited at p. 71]

[210] H.L. Van Trees. *Detection, Estimation, and Modulation Theory Part IV - Optimum Array Processing*. JOHN WILEY & SONS, 2002. [cited at p. 5, 20, 22, 71, 77]

[211] J. Treichler and B. Agee. A new approach to multipath correction of constant modulus signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31(2):459–472, Apr 1983. [cited at p. 114]

[212] M. K. Tsatsanis and Zhengyuan Xu. Performance analysis of minimum variance CDMA receivers. *IEEE Transactions on Signal Processing*, 46(11):3014–3022, Nov. 1998. [cited at p. 116]

[213] D. Tse and P. Viswanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005. [cited at p. 6]

[214] P.W. Tse, J.Y. Zhang, and X.J. Wang. Blind source separation and blind equalization algorithms for mechanical signal separation and identification. *Journal of Vibration and Control*, 12(4):395–423, 2006. [cited at p. 114]

[215] Ching-Yih Tseng, D.D. Feldman, and L.J. Griffiths. Steering vector estimation in uncalibrated arrays. *IEEE Transactions on Signal Processing*, 43(6):1397–1412, June 1995. [cited at p. 41]

[216] J.K. Tugnait and U. Gummadavelli. Blind channel estimation and deconvolution in colored noise using higher order cumulants. *J. Franklin Inst.*, 333(3):311 – 337, 1996. [cited at p. 22]

[217] G. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3):311–329, 1960. [cited at p. 45]

[218] R. Vaidya and S. Bhatnagar. Correlation based optimization of random early detection. In *IEEE India Anual Conference (INDICON)*, 2004. [cited at p. 102]

[219] R. Vaidya and S. Bhatnagar. Optimized RIO for diffserv networks. In *Proceedings of International Conference on Information and Computer Science (ICICS)*, 2004. [cited at p. 102]

[220] C. Vargas-Irwin and J.P. Donoghue. Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. *J Neurosci Methods*, 164(1):1–18, Aug 2007. [cited at p. 44, 63]

[221] B.N. Vo, S. Singh, and V. Tadic. Adaptive envelope-constrained filtering. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, 2003. [cited at p. 19]

[222] R. Vollgraf. *Unsupervised Learning Methods for Statistical Signal Processing*. PhD thesis, 2006. [cited at p. 12, 22, 114]

[223] R. Vollgraf, M. Munk, and K. Obermayer. Optimal filtering for spike sorting of multi-site electrode recordings. *Network*, 16(1):85–113, Mar 2005. [cited at p. 20, 45]

[224] R. Vollgraf and K. Obermayer. Improved optimal linear filters for the discrimination of multichannel waveform templates for spike-sorting applications. *IEEE Signal Processing Letters*, 13(3):121–124, 2006. [cited at p. 22, 68]

[225] Guang-Li Wang, Yi Zhou, Ai-Hua Chen, Pu-Ming Zhang, and Pei-Ji Liang. A robust method for spike sorting with automatic overlap decomposition. *IEEE Trans Biomed Eng*, 53(6):1195–1198, Jun 2006. [cited at p. 44]

[226] Z. Wang and P. Willett. A performance study of some transient detectors. *IEEE Transactions on Signal Processing*, 48 (9):2682–2685, 2000. [cited at p. 20]

[227] P.T. Watkins, G. Santhanam, K.V. Shenoy, and R.R. Harrison. Validation of adaptive threshold spike detector for neural recording. In *Proc. 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society IEMBS '04*, volume 2, pages 4079–4082, 1–5 Sept. 2004. [cited at p. 71]

[228] M. Wax and Y. Anu. Performance analysis of the minimum variance beamformer. *IEEE Transactions on Signal Processing*, 44(4):928–937, 1996. [cited at p. 40]

[229] M. Wax and Y. Anu. Performance analysis of the minimum variance beamformer in the presence of steering vector errors. *IEEE Transactions on Signal Processing*, 44(4):938–947, 1996. [cited at p. 40]

[230] M. Welling. Robust higher order statistics. *Tenth International Workshop on Artificial Intelligence and Statistics*, pages 405–412, 2008. [cited at p. 90]

[231] M.T. Wolf, J.G. Cham, E.A. Branchaud, and J.W. Burdick. A miniature robot for isolating and tracking neurons in extracellular cortical recordings. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2008*, pages 1594–1601, 19–23 May 2008. [cited at p. 63, 93, 101]

[232] M.T. Wolf, J.G. Cham, E.A. Branchaud, G.H. Mulliken, J.W. Burdick, and R.A. Andersen. A robotic neural interface for autonomous positioning of extracellular recording electrodes. *The International Journal of Robotics Research*, 28(9):1240–1256, 2009. [cited at p. 63, 93, 97]

[233] Dan Wu, Xuemai Gu, and Qing Guo. Improved super-exponential algorithm for blind equalization. *Digital Signal Processing*, 19:444–451, 2009. [cited at p. 115]

[234] W. Wu, D. Wheeler, E. Staedtler, M. Munk, and G. Pipa. Behavioral performance modulates spike field coherence in monkey prefrontal cortex. *Neuroreport*, 19(2):235–238, Jan 2008. [cited at p. 55]

[235] www.sci.utah.edu/~gk/abstracts/bisti03/. [cited at p. 2]

[236] R. Xu and D. II Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005. [cited at p. 52]

[237] Xian-Feng Xu, Da-Zheng Feng, Wei Xing Zheng, and Hua Zhang. Convolutive blind source separation based on joint block toeplitzation and block-inner diagonalization. *Signal Processing*, 90:119 – 133, 2010. [cited at p. 91, 113]

[238] Zhengyuan Xu, Ping Liu, and Xiaodong Wang. Blind multiuser detection: from MOE to subspace methods. *IEEE Transactions on Signal Processing*, 52(2):510–524, Feb. 2004. [cited at p. 116]

[239] Zhengyuan Xu and M.K. Tsatsanis. Blind adaptive algorithms for minimum variance CDMA receivers. *IEEE Transactions on Communications*, 49(1):180–194, Jan. 2001. [cited at p. 115]

[240] A. Yamazaki, T. Tajima, and K. Matsuoka. Convolutive independent component analysis of EEG data. *SICE 2003 Annual Conference*, 2:1227– 1231, 2003. [cited at p. 20]

[241] Zhi Yang, Qi Zhao, and Wentai Liu. Improving spike separation using waveform derivatives. *Journal of Neural Engineering*, 6(4):046006+, 2009. [cited at p. 65]

[242] D. Yellin and B. Porat. Blind identification of FIR systems excited by discrete-alphabet inputs. *IEEE Transactions on Signal Processing*, 41(3):1331–1339, March 1993. [cited at p. 91, 117]

[243] Ke-Hai Yuan and P.M. Bentler. Two simple approximations to the distributions of quadratic forms. *Department of Statistics, UCLA. Department of Statistics Papers*, Paper 2007010106, 2007. [cited at p. 41]

[244] Yuan Yuan, Chenhui Yang, and J. Si. An advanced spike detection and sorting system. In *Proc. International Joint Conference on Neural Networks IJCNN 2009*, pages 3477–3484, 14–19 June 2009. [cited at p. 90]

[245] Pu-Ming Zhang, Jin-Yong Wu, Yi Zhou, Pei-Ji Liang, and Jing-Qi Yuan. Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. *J Neurosci Methods*, 135(1-2):55–65, May 2004. [cited at p. 44]

[246] B. Zhao, J.A. Yang, and M. Zhang. Research on blind source separation and blind beamforming. In *Proc. International Conference on Machine Learning and Cybernetics*, volume 7, pages 4389–4393, 18–21 Aug. 2005. [cited at p. 66]

[247] A. Zviagintsev, Y. Perelman, and R. Ginosar. Algorithms and architectures for low power spike detection and alignment. *Journal of Neural Engineering*, 3:35–42, 2006. [cited at p. 71]

List of Figures

List of Tables

Index