UNIVERSITÀ DI PADOVA FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione

Scuola di Dottorato di Ricerca in Ingegneria dell'Informazione
Indirizzo: Automatica e Ricerca Operativa

XXI Ciclo

NEURAL SPIKES CLASSIFICATION IN MULTICHANNEL RECORDINGS

Direttore della Scuola: Ch.mo Prof. Matteo Bertocco

Supervisore: Ch.mo Prof. Alessandro Beghi

Dottoranda: Martina Camarda

“Every thing should be as simple as possible, but not simpler.”

A. Einstein

Sommario

Per studiare il comportamento dei neuroni vengono utilizzate sonde multicanale in cui ciascun elettrodo misura la sovrapposizione di treni di spike generati dai neuroni circostanti. Un primo passo necessario è quello di individuare e separare i segnali provenienti da diverse sorgenti associando ciascuno spike al neurone che lo ha generato. A questo scopo sono stati sviluppati molti algoritmi di *spike sorting* (classificazione di spike) che si basano su differenti principi, ma nessun metodo è, finora, stato riconosciuto come migliore degli altri.

Questa tesi affronta il problema della classificazione di segnali impulsivi nel contesto Neurofisiologico presentando un nuovo algoritmo di *spike sorting* denominato *Multi-Channel Inversion for Spike Classification* (MCI4SC). Il nuovo metodo sfrutta l'informazione proveniente da più canali (legata alla posizione dei neuroni) e fa un uso distintivo della matrice di mixing associata al canale di misura. In particolare, invertendo più matrici derivate da quella di mixing, il metodo è in grado di gestire la sfavorevole ma tipica situazione in cui sono presenti più neuroni registrati che sensori, sotto la ragionevole ipotesi che il numero di neuroni contemporaneamente attivi sia minore od uguale al numero di sensori. Un'altra caratteristica distintiva dell'algoritmo MCI4SC e della sua implementazione è l'uso della Trasformata Wavelet Packet. Questo strumento è stato impiegato per stimare il rapporto tra le ampiezze degli spike nei differenti canali, dando così luogo a una stima delle componenti della matrice di mixing che risulta consistente anche nel caso di basso rapporto segnale rumore.

L'algoritmo MCI4SC è stato applicato a dati sperimentali impostando manualmente le soglie. Buoni risultati di classificazione sono stati ottenuti sia nel caso di Purkinje Cells con spikes di differenti ampiezze e forme d'onda, sia nel caso di neuroni nel lobo antennale della Locusta con segnale molto rumoroso. Nel confronto con un algoritmo basato sul metodo *Markov Chain Monte Carlo*, l'algoritmo MCI4SC presenta una efficienza almeno comparabile con un più basso tempo computazionale, oltre alla importante capacità di risolvere spike sovrapposti. Ciò rende il nuovo algoritmo, presentato in questa tesi, uno strumento affidabile e competitivo nel contesto dello *spike sorting*.

Parole chiave: classificazione di spike, registrazione multicanale, rapporto di ampiezza, wavelet.

Abstract

In order to study the neural behavior, scientists employ multichannel probes where each electrode records a mixture of spike trains from surrounding neurons. A first necessary step is the individuation and the separation of signal from different sources, associating each detected spike to the neuron of origin. Many spike sorting algorithms based on different principles have been developed for this purpose, but there is still no consensus on which is the best method.

This thesis addresses the issue of impulsive signal classification in the Neurophysiological framework presenting a novel spike sorting algorithm named Multi-Channel Inversion for Spike Classification (MCI4SC). The new method exploits multichannel information related to neuron positions, and makes a distinctive use of the mixing matrix associated to the measurement channel. In particular, inverting many matrices derived by the mixing one, the method is able to handle the disadvantageous, but typical, situation where there are more recorded neurons than recording sensors, under the reasonable hypothesis that the number of simultaneously firing neurons is lower than or equal to the number of sensors. Another distinguishing feature of MCI4SC algorithm and its implementation is the use of the Wavelet Packet Transform. This tool has been used to estimate the ratio between spike amplitudes in different channels, thus leading to a consistent estimation of the mixing matrix components even in case of low signal to noise ratio.

The MCI4SC algorithm has been applied on experimental data with human supervision on the threshold setting. Good spike sorting results have been obtained for bursting Purkinje Cells with varying waveform and amplitude spikes, as well as for noisy neurons in Locust antennal lobe. Compared with an algorithm based on the Markov Chain Monte Carlo, the MCI4SC algorithm has at least comparable efficiency with a much lower computational time, in addition to the important capability of overlapping spike resolution. This makes the new algorithm presented in this thesis a reliable and competitive tool in the spike sorting context.

Keywords: spike sorting, multichannel recording, amplitude ratio, wavelet.

Contents

Introduction

This thesis has been developed in the context of neural signal processing, where the initial step toward the study of neural behavior is the individuation and the separation of signals from different neurons. Neurons typically communicate one with the others by firing brief voltage spikes. These electrical pulses, and in particular their timing, are the precious information that neurophysiologists aim to decode in order to study the nervous system.

The extracellular measurement of electrical spikes is carried out by placing an electrode in close proximity to active neurons and by simultaneously recording several spike trains. Due to improvements in implantable hardware technology, multiple electrodes on the same probe are nowadays feasible and multichannel recordings have become a standard tool in the neurophysiological context.

As each extracellular electrode records a mixture of spike trains generated from surrounding neurons, a crucial step is to separate the single neural traces to determine when each neuron fired. In other words, it is necessary to identify all spike events and to assign each of them to the neuron that has produced it. In the last decades a lot of *spike sorting* algorithms have been developed for this purpose, but there is still no consensus on which is the best method.

The purpose of this thesis is to develop a new spike sorting algorithm that has been named Multi-Channel Inversion for Spike Classification (MCI4SC). The new algorithm exploits multichannel information related to neuron positions in order to assign each spike to its own neuron of origin. The most attractive feature of MCI4SC is its capability to resolve overlapping spikes. This challenging aspect is often unattended by the most part of spike sorting algorithms, as well as the problem of neurons that fire spikes with amplitude and waveform variations. As shown in Chapter 4, the application of MCI4SC on experimental data confirms its efficiency potentialities even in case of overlapping spikes and spike variations. This makes MCI4SC a reliable and competitive tool in the spike sorting context.

The main idea of this algorithm is to separate original neural signals inverting the mixing matrix associated to the measurement channel between neurons and sensors. The number of recorded neurons is typically greater than the number of recording sensors, thus leading to a rectangular mixing matrix and preventing a straightforward solution. Nevertheless, thanks to the sparsity and to the finite duration of neural spikes, under the reasonable hypothesis that the number of simultaneously firing neurons is lower than or equal to the number of sensors, MCI4SC is able to handle this unfavorable situation reaching good spike sorting results.

Summarizing, the developed algorithm consists of two subsequent phases: first, the *learning phase* estimates the mixing matrix and the neural waveform templates; then, the *classification phase* associates the neuron of origin and the arrival time to each detected spike. To reach this last purpose, MCI4SC algorithm inverts many matrices derived by the mixing one. Since each inverted matrix is associated to a reduced measurement channel made by a number of neurons equal to the number of sensors, the original signals separation can be reached.

Besides the original use of matrices associated to reduced measurement channels, another distinguishing feature of MCI4SC algorithm and its implementation is the use of the Wavelet Packet Transform. This tool has been used in the learning phase to estimate the ratio between spike amplitudes in different channels, thus leading to a consistent estimation of the mixing matrix components even in case of low signal to noise ratio.

The thesis outline is the following:

Chapter 1 — The spike sorting problem in Neurophysiology — introduces the reader to the context of spike sorting problem. Purposes and difficulties are described in this first chapter, as well as the neural signal features.

Chapter 2 — Spike sorting methods — presents a brief review of spike sorting algorithms pointing out the respective assumptions, worthiness and weakness. In particular, this chapter considers the basic spike sorting methods, the methods based on Wavelet Transformations, those that exploit multichannel information, and those that aim to resolve overlapping spikes.

Chapter 3 — Multi-Channel Inversion for Spike Classification — describes the new method, named MCI4SC, to sort neural spikes recorded by multichannel probes. As well as explaining the main ideas and the algorithm details, this chapter also illustrates the linear mixing model adopted for the measurements, the necessary assumptions, and the preliminary steps. Finally, qualitative comparisons with other two similar approaches are illustrated.

Chapter 4 — Analysis of experimental data and Results — reports the application of this new spike sorting algorithm on experimental data from a Purkinje Cells recording. The main results are compared with those of another algorithm, based on the Markov Chain Monte Carlo, that has been applied on the same data set. The two algorithms results almost comparable in efficiency, but MCI4SC takes much lower computational time, and it is able to resolve total overlapping spikes.

Appendix A — Wavelet Analysis — gives a brief introduction to the Wavelet Analysis whose properties result very useful in the context of neural spike recordings, and thus have been exploited by MCI4SC algorithm.

Appendix B — Locust data analyzed by MCI4SC algorithm — illustrates the details of the MCI4SC application on a different set of experimental data from the Locust antennal lobe.

Chapter 1

The spike sorting problem in Neurophysiology

Neurophysiology is the part of physiology that studies the nervous system function in order to understand neural signal codification and relationships between stimulus and response. A lot of studies in this field have produced advances in our knowledge of how nervous systems code sensory and motor information. Still much work, however, remains to be done to explain complex mechanisms of neural behavior.

Neurons typically communicate one with the others by firing brief voltage spikes (named action potentials). These electrical pulses, and in particular their timing, are the precious information that neurophysiologists aim to decode. Since brain processes are the result of a large neural populations activity, the study of single neurons in isolation gives an extremely poor vision of the whole system: therefore scientists require to observe the interaction of a large number of entities [13] [10].

The extracellular measurement of electrical spikes is carried out by placing an electrode in close proximity to active neurons and simultaneously recording several spike trains from as many sources as possible. Obtaining such “multi-unit” recording is easier and cheaper than performing more “single-unit” recordings with as many probes as observed neurons. On the contrary, due to improvements in implantable hardware technology, multiple electrodes on the same probe are nowadays feasible and multichannel multi-unit recordings have become a standard tool in the neurophysiological context [39][40].

As each extracellular electrode of the probe records a mixture of spike trains generated from surrounding neurons, the first crucial step is to separate the single neural traces to determine when each neuron fired. Switching the perspective, it is necessary to identify all spike events and to assign each recorded spike to the neuron that has produced it. This kind of analysis generally goes by the name of *spike sorting* (see Figure 1.1).

Because the accuracy of spike sorting critically affects the accuracy of all

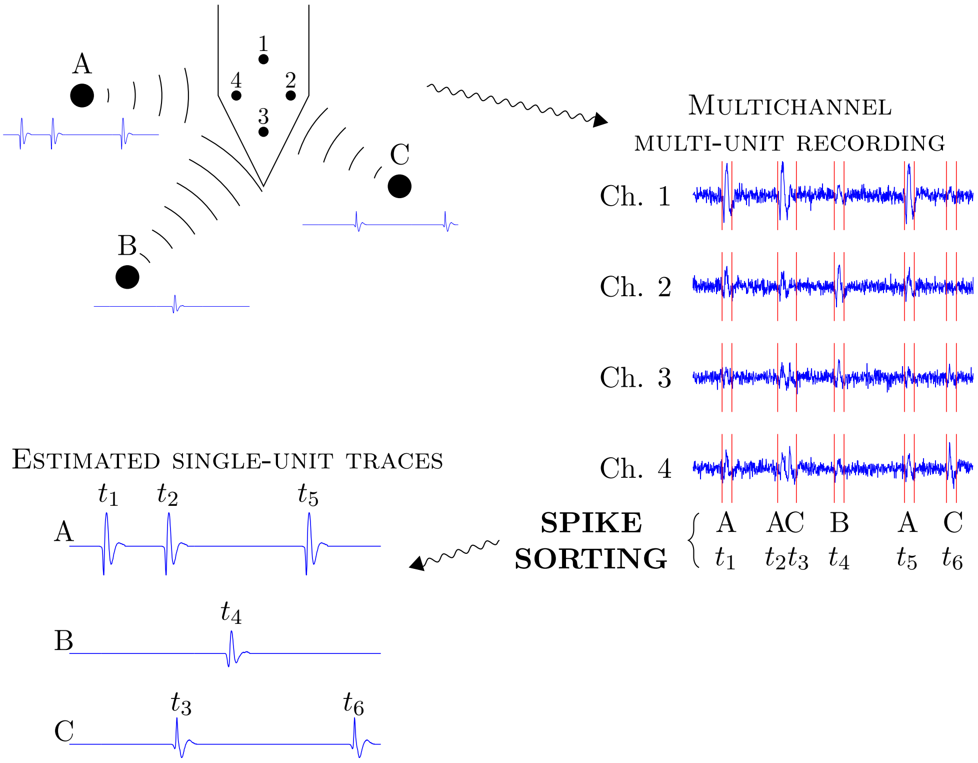


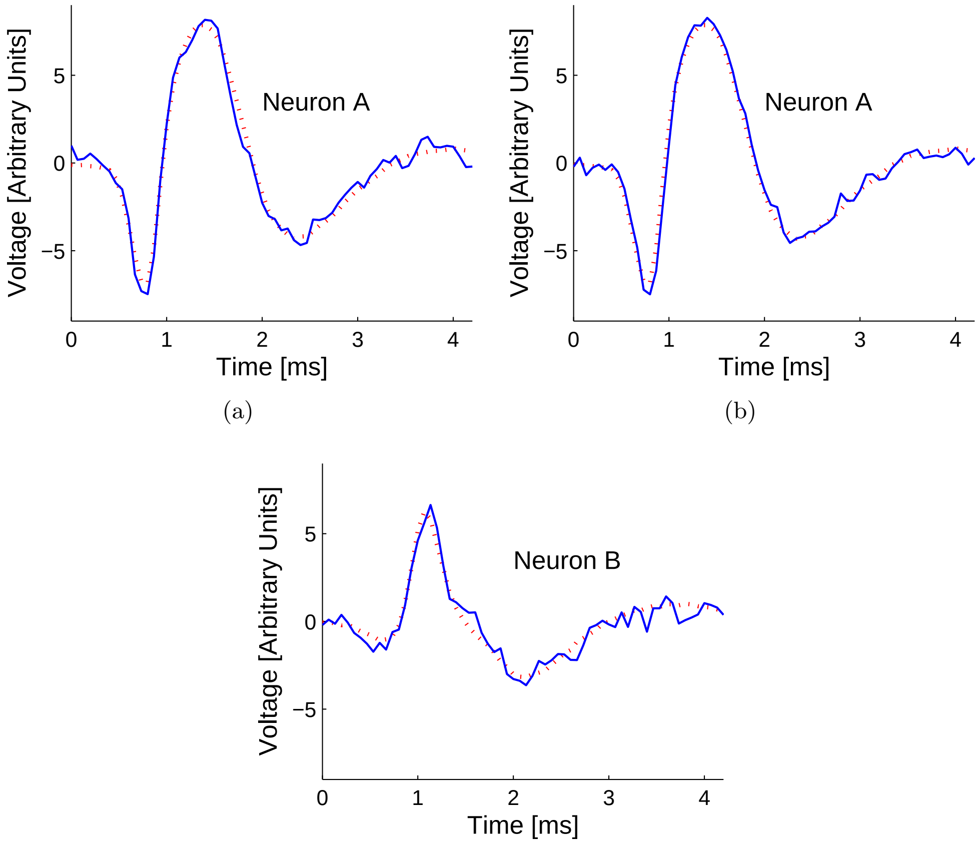
Figure 1.1: In the top left corner a probe with 4 electrodes records the activity generated by surrounding neurons A, B, C (no scale). On the right side, a simulated multichannel multi-unit recording where each channel presents a mixture of the three neural traces. The spike sorting algorithm detects spikes and assigns them to their neurons specifying the arrival times. As a result, the procedure gives an estimation of the original single-unit spike trains (on the bottom left corner).

subsequent analyses on neuron interactions and response codification, developing a good classification algorithm is a very important target in the study of nervous system function. In the last decades, this challenging problem has attracted the attention of scientists beyond the field of neurophysiology; a lot of algorithms have been developed in order to achieve the classification of neural pulses from a multi-neural activity recordings, but there is still no consensus on which is the best method.

1.1 The signal in neural recordings

To better understand potentialities and limits of spike sorting algorithms, this section describes some important neuron peculiarities that allow, in principle, to differentiate spikes of one neuron from those of other neurons. Spike sorting methods that have been developed until now, exploit one or more of these aspects making the due ideal assumptions.

Spike Waveform — The recorded waveform provides, in principle, a means to classify spikes as belonging to the same neuron or not: in general, spikes from the same neuron look similar and spikes from different neurons look different (see Figure 1.2).



(a)

(b)

(c)

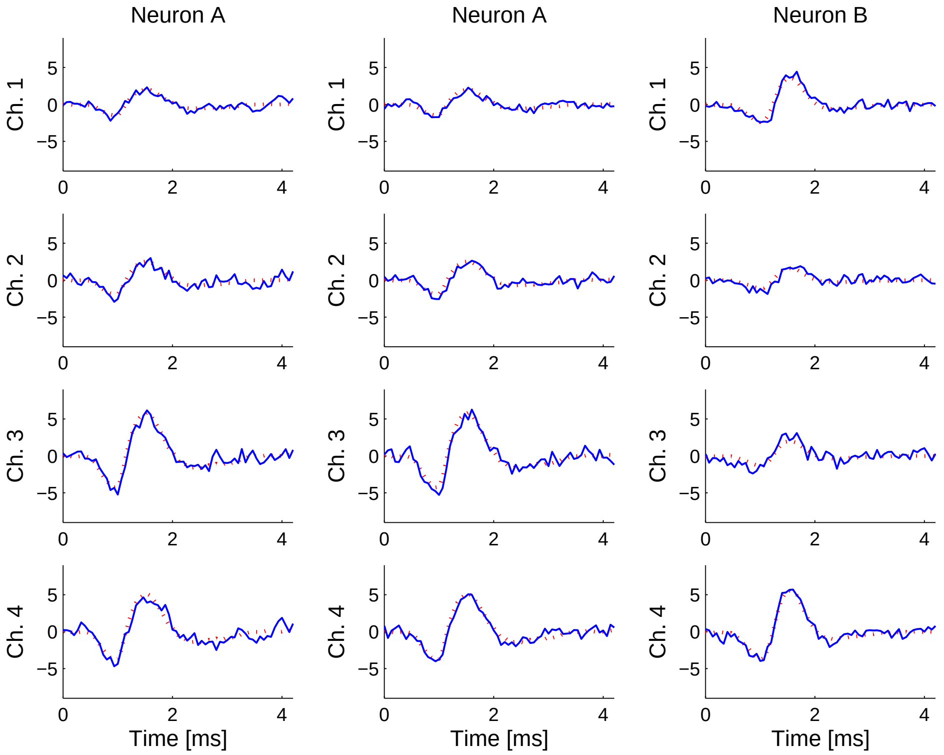
Figure 1.2: Some examples of spike waveforms recorded in the locust (*Schistocerca americana*) antennal lobe: spikes (a) and (b) are from the same neuron and look similar; spike (c) is from another neuron and appears different. Solid lines are for raw data and dotted lines are for estimated spikes. The vertical scale is voltage in arbitrary units.

Actually, neurons belonging to a certain population generate spikes with almost identical waveforms. Nevertheless, due to the anisotropic properties of the extracellular medium, waveforms from distinct neurons differ at the electrode location where they are recorded [38]. To exploit this aspect, many spike sorting methods (see for example [67]) take advantage of the ideal assumption that each neuron has its own characteristic waveform even if, in practice, different neurons may present indistinguishable spike shapes. To differentiate such ambiguous neurons, it is necessary to jointly exploit other peculiarities described in the following paragraphs.

Moreover, the assumption that the neuron waveform remains unchanged in time is an idealization which usually fits well the actual situation, but it is not always true; for example, bursting neurons generate spikes that may change their shape over the course of the burst. Despite that, many algorithms that classify spikes by waveforms, assume that the spike shapes are relatively stable during the recording time.

Spike Amplitude and Neuron Position — The amplitude of recorded waveform decreases inversely with the square of the distance between the neuron and the electrode [65]. Therefore, neurons that fires with mostly similar spike amplitude but have different distances from the sensor, will be recorded with consequently scaled waveforms. For this reason, early spike sorting algorithms (see for example [28]) widely exploited the spike amplitude as a distinctive feature assuming it constant through subsequent firings from the same neuron (on condition that the probe position remains steady during the recording time). Although the constant amplitude assumption is true for many neurons, spike amplitude may actually be a decreasing function of the time between two consecutive spikes and some methods take in account even this variability.

The introduction of multi-electrodes probes made available further interesting point of views: since electrodes are in a different position, they may measure the same spike with different amplitudes, different delay times, and even different waveforms. On the other hand, neuron position strongly affect the appearing of these differences through the sensors and since neuron are necessarily located in distinct places this provide useful information for spike discrimination. Nevertheless, it is worth noticing that delay times and waveforms on different electrodes can be fairly similar when sensors are quite close to each others (as in some commonly used probes), while different spike amplitudes are usually still representative of distinct neuron position (see Figure 1.3). In particular, ratios of spike amplitude from several sensors have an interesting property, i.e. they remain relatively constant with changes in spike amplitude and spike shape. In fact these spike changes, that often occur for example during a burst firing, are proportionally the same on all channels under the condition that the probe position remains steady. Another aspect to point out is the capability, in principle, to distinguish the spike source without any ambiguity: under the idealization that neurons are point sources in a homogeneous medium, ratios of spike amplitude from at least 4 electrodes not located in a plane are unique for each neuron and allow to locate it in the tri-dimensional space. Nevertheless, if the electrodes are less than four and/or are in a planar — or even worst linear — configuration, amplitude ratios cannot univocally determine the neuron position and some sources may possibly result indistinguishable from the spike-amplitude ratios point of view. This problem arises not only when less than four electrodes are available but also when the neuron signal is too weak and it is recorded just



(a) (b) (c)

Figure 1.3: Some examples of spikes recorded by 4 sensors in the locust (*Schistocerca americana*) antennal lobe: the same spike is seen by more sensors with different amplitudes related to the distance between neuron and sensor (solid lines for recorded measures and dotted lines for estimated spikes). Spikes (a) and (b) are from the same neuron and show similar amplitude trend along the four channels. Spike (c) is from another neuron: it has a similar waveform but it can be distinguished by its different amplitude pattern. The vertical scale is voltage in arbitrary units.

by the nearest sensors.

Inter-Spike Interval Distribution — Additional information for spike classification comes from arrival times and the intervals between them. In fact, the Inter-Spike Interval (ISI) distribution of a single neuron has a trend that reflect the biophysical properties of the neuron itself. The most characteristic property is the refractory period: once a neuron has fired, it needs a recharging period before being able to fire the next spike. The duration of this period depends on neuron typology (for example the refractory period for neocortical cells is about $1 \div 2\,ms$ [52]). The refractory period is clearly highlighted in an ISI histogram that starts at zero and stays at zero for a finite time (see Figure 1.4). The presence or not of a clear refractory period in the ISI histogram indicates if involved

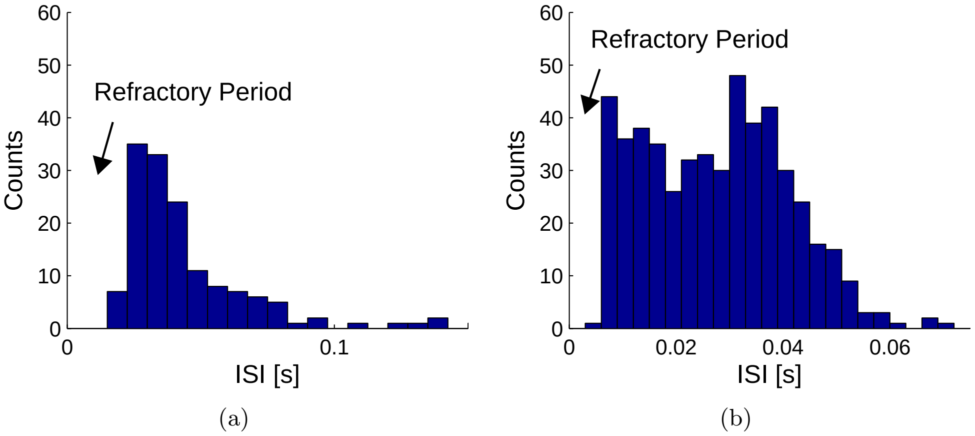


Figure 1.4: Two examples of Inter-Spike Interval histogram. In (a) the ISI histogram of a spike train from a projection neuron in the locust (*Schistocerca americana*) antennal lobe (20 seconds, 176 spikes). In (b) the multi-modal ISI histogram of a spike train from a Purkinje cell in young rat cerebellar slices (58 seconds 644 spikes). The bin width in the horizontal axis is $7.5\,ms$ and $3\,ms$, respectively. The vertical axis represents the spike count in a bin.

spikes are generated by only one or more neurons. This information can be used either to carry out the spike sorting procedure in combination with others kind of distinctive information [32], or as a final reliability check for spike sorting results. More sophisticated methods take into account, not only the refractory period, but the whole — opportunely modeled — ISI distribution [26].

The above aspects (spike waveform, spike-amplitude ratios from several electrodes and ISI distribution) are conceptually independent and provide complementary information to potentially differentiate one neuron from the others. In order to achieve the best spike sorting result it is wise — although complex — to exploit all these features at the same time because, in general, they are unable to univocally discriminate the source on their own: for example, neurons whose recorded spike waveforms are similar, can be distinguished by different spike amplitude ratios from several sensors and, vice versa, neurons whose recorded spike amplitude ratios are similar, can be distinguished by different waveforms.

1.2 Difficulties concerning spike sorting in neurophysiology

This section illustrates some technical and physiological issues that make the spike sorting problem non-trivial. As it is pointed out in the following chapter, developed methods deal more or less successfully with these aspects.

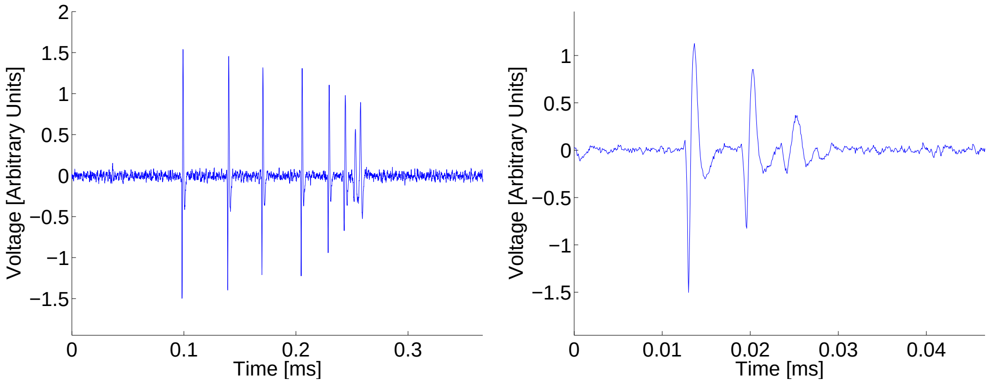
Unknown Number of Involved Neurons — Typically the recorded spikes come from an a priori unknown number of different neurons. It is necessary to infer even this information by data themselves, paying attention to neuron that may rarely fire, and avoiding to consider two distinct neurons as the same one or a single neuron as two.

Noise in the recorded data — Recorded neural spikes are embedded in noise due both to recording hardware and to activity of distant neurons (background noise). The signal to noise ratio, that varies depending on fired spike amplitude and neuron distance, can be low for some neurons in certain conditions. In particular, this happens in in-vivo recordings where a lot of surrounding neurons give rise to background activity. Moreover, the background noise, due to its nature, may have spectral density similar to that of the spikes under analysis, can be correlated to the spiking of foreground neurons and hence can also be non-stationary.

Electrode drifts during the recording — During the recording time, experimental condition may evolve and the probe may move in the tissue. This implies changes in neurons-electrodes relative positions, and hence changes in spike waveforms and amplitude ratios between channels. Nevertheless, if the drift is quite slow, it is possible to consider time interval where conditions remain stationary.

Bursting Neurons generate spikes varying in amplitude and shape — Bursting neurons are those neurons that repeatedly fires burst or fast sequence of spikes. Bursts can have a variable number of firings (a burst of two spikes is called a doublet, of three spikes is called a triplet, of four a quadruplet, etc.) and each burst is followed by a period of quiescence before the next occurs. Bursts can be visually identified in single-unit recordings, but it is difficult to distinguish them from random events when small bursts, like doublets or triplets, are embedded in spike trains or when recordings contain spikes from many neurons. Bursts can also be identified with the help of ISI histograms: short ISIs occur more frequently when are induced by burst dynamics, thus creating a first high peak in the corresponding multi-modal ISI histogram. The main problem concerning bursting neuron is that, as mentioned in the previous section, their spikes can vary in both shape and amplitude (see Figure 1.5). Therefore, the spike sorting procedure must be careful that spikes generated by a single bursting neuron are not classified as coming from more separate neurons.

Presence of Overlapping Spikes — In the spike sorting context, overlapping spikes are one of the main issue to deal with. They occur when two or more neurons fire simultaneously — or one after another within a very short delay — producing a superposition of voltage with different characteristic from time



(a)

(b)

Figure 1.5: Two examples of burst from Purkinje cells in young rat cerebellar slices. In (a) a burst with 8 spikes and in (b) a triplet. Notice that, in both cases, spike amplitude and waveform vary along the burst. The vertical scale is voltage in arbitrary units.

to time. The number of different superpositions is very large and theoretically infinite: it depends not only on the large number of combinations with more than 2 involved neurons, but also on variations in the overlap phase. Amplitude and waveform of the overlapping spike are more or less distorted with respect to the original spike, according to the percentage of spike superposition (see Figure 1.6). Therefore, ordinary spike sorting methods, based only on waveform or amplitude recognition, are prevented from efficiently resolving overlapping spikes, unless the superposition is little. It should be remarked, however, the relevance of dealing accurately with overlapping spikes, since they may give important information about the correlation of neural activity; on the other hand, errors in their classification give biased results in the reconstruction of original spike trains.

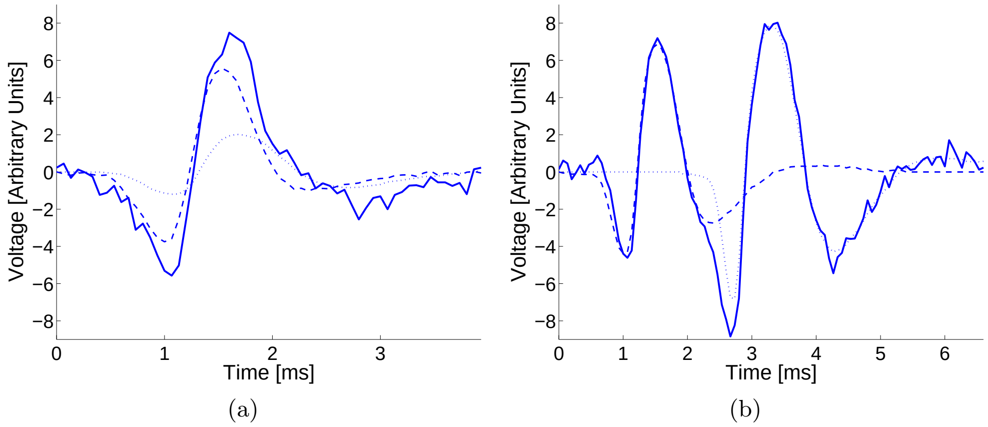


Figure 1.6: Two examples of overlapping spikes recorded in the locust (*Schistocerca americana*) antennal lobe. Solid lines for raw data; dashed and dotted lines for the two estimated spike waveforms. In (a) a total superposition of two spikes: notice that the resulting waveform and the peak amplitude are different from both the original ones, and the overlapping spike looks like the firing of a third neuron. In (b) a partial superposition: notice that double peaks are still visible and almost with their original amplitude, but the two spike waveform are distorted. The vertical scale is voltage in arbitrary units.

Chapter 2

Spike sorting methods

As already seen, a spike sorting algorithm takes one or more voltage traces containing spikes from several neurons and attempts to produce, for each source present in raw data, a collections of times corresponding to respective events. There are some ideal requirement that these algorithms should satisfy. Generally speaking they should be:

- Highly *efficient* in spike classification even in case of overlapping spikes, bursting neurons, and nearly silent neurons (correctly identifying the number of active neurons).
- *Robust* to non-stationary conditions, like drifting of the electrodes, variations of neural properties, noise variability, and in general all the violations of explicit and implicit assumptions on which the algorithm is based.
- *Unsupervised* or at least with a hybrid approach where most of the process is carried out automatically and the human operator is required only for supervision or validation. The completely manual approach, as well as subjective and unreproducible, is even slow and inapplicable when the number of electrodes in the recording system and the number of recorded neurons are too large (also for the human inability to visualize the high-dimensional cluster space). Nevertheless, given the complexity of the problem and our current knowledge, a totally unsupervised approach is nowadays not much reliable.
- *Scalable* with electrode numbers, and independent of different probe geometries. Current acquisition system allow the simultaneous recording from few or hundreds of channels in different configurations: there should be algorithms that are able to operate in these various situations.
- *Real-time* algorithms with low computational complexity, in order to use multiple spike trains as inputs to neural prosthetic devices or brain-machine interfaces. Real-time spike sorting could also lead to real-time data analysis, and to real-time changes during experiments.

Many spike sorting algorithms have been developed until now, from early manual pattern recognition to more automated methods, but meeting the above mentioned ideal requirements is still an open problem. This chapter presents a brief review of spike sorting algorithms pointing out the respective assumptions, worthiness and weakness, without any pretension to be exhaustive in the presence of a so large variety of proposed methods. For more detailed reviews about spike sorting methods and their comparisons see [2][76][72][48][10].

As it will be shown in the following, the various approaches differ in the criteria for identifying single neurons, in assumptions about noise statistics and signal characteristics. Besides differences in their implementations, the majority of spike sorting algorithms have some main steps in common: the event detection in raw data, the extraction of distinctive features from each spike and the feature clustering that groups spikes from the same neuron. It is interesting to give some more details about those steps, schematically represented in Figure 2.1.

Spike Detection — Extracellular spike recordings are inevitably corrupted by noise and that makes necessary an efficient spike detection to individuate and isolate events. A simple and widely used technique is the amplitude thresholding of sampled voltage, but its performance degradate rapidly with low signal to noise ratio. Other detection methods have been investigated: for example, the power detection method applies a threshold on the sum of squared samples over a window of fixed length [40][4], the matched filter detection considers instead the inner product with a certain spike template [4][26], while other detection methods are based on Haar Transformation [77], Continuous Wavelet Transform [56] or the point-wise product of Wavelet Transform coefficients of multiple scales [43]. In each case, the threshold value is set by a trade-off between the maximum number of detected spikes and the minimum number of false positive.

Feature Extraction — This is a key step in which a few distinctive characteristics (or features) are selected from each spike. Those features can be related to spike waveform and/or to neuron position. For example some simple choices are the peak amplitude, the spike width, the peak-to-peak amplitude, or the signal energy. More elaborated choices are the first principal components, some selected wavelet coefficients or amplitude ratios in multi-channels recordings. As a limit case, features may be directly represented by recorded samples in their original high dimensional space [32]. The crucial point is that extracted features have to maintain and eventually exalt information that allow to classify events: ideally, they get rid of noise and best represent differences among spikes from different neurons.

In early methods, because of limited computational resources and manual intervention, the purpose was to choose the minimal set of features that yielded the best discrimination [73][29]. In fact, the dimensionality reduction of the object to sort — going from a space of dimension equal to the number of data

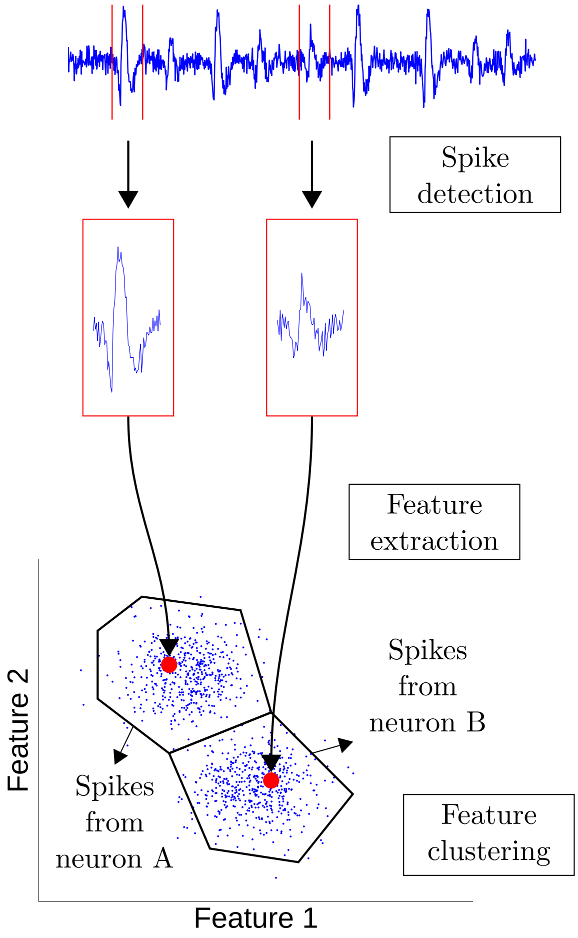


Figure 2.1: Spikes are detected in a simulated multi-unit recording; for each detected spikes, some distinctive features are extracted and clustered in order to assign each spike to a certain group or neuron.

points per spike, to a lower dimensional space of a few features — saves computational time and it is necessary for some clustering algorithms that can not handle too many inputs in a reasonable time, or that need a manual classification through a visual representation on bi- or tri-dimensional plots. Nevertheless, in general, there are better possibilities to distinguish spike form different neurons when more features are used.

Feature Clustering — In principle, similar features correspond to the same neuron and, consequently, features extracted by all spikes in a data set ideally create as many separate groups as many involved neurons. Spike sorting can then be performed through clustering algorithms that isolate each group of features. A straightforward solution is the manual clustering where the human operator identify polygons that frame and separate clusters in bi-dimensional projections of the spike features (see for examples [66] [46]). This method is obviously very time-consuming, subjective — especially when clusters overlap — and limited by the bi-dimensional visualization of the cluster cutting space [40].

Many unsupervised clustering methods have been developed. Among them, K-means is one of the simpler, faster and thus widely used algorithm (see for example [41] [42] [70] [74]). Given an a priori fixed number of clusters (i.e. given the number of involved neurons), this algorithm iteratively assigns each point to the cluster whose center (or centroid) is the nearest — according to a certain metric — and re-calculate the cluster centroids until they are stable [31]. As a drawback, the result significantly depends on the selection of initial cluster centers.

Other, more refined, solutions have been proposed, based for example on Bayesian classification, or on Expectation Maximization procedures. The Bayesian classification proposed in [47], models spikes as the sum of an average waveform plus white noise. Under the assumption of Gaussian noise, it is possible to define quantitatively the probability that a certain number of spike models describe the spikes in the observed signal. The final spike models may then be chosen as that with the highest probability, given the data. In [69], an iterative Expectation-Maximization procedure estimates the maximum likelihood model parameters (mean and covariance matrix) for each cluster, as well as mixing parameters describing the relative frequency of the different clusters. These last two approaches presume a Gaussian distribution of the clusters, assuming that the spike variability is determined just by additive and Gaussian stationary background noise. Although this assumption may be plausible in many recordings, there are several technical and physiological issues that can make it unreasonable, giving rise to elongated non-Gaussian clusters [33] that are erroneously split in more groups. As already seen, some issues are, for example, electrode drifts during the recording, variation in the spike shape due to bursting, presence of overlapping spikes, non-stationary background noise, correlations among spikes and background noise, and misalignments in the spike detection.

One approach to avoid the assumption of Gaussian distributions is the hierarchical clustering algorithm proposed in [32] that uses spike time information to determine cluster boundaries: it first classifies the events into an exceedingly large number of clusters and then merges these clusters according to both spike shape similarities as well as the statistics of ISI distributions (such as imposed by the refractory period). Another solution to the problem of non-Gaussian clusters are the algorithms based on nearest neighbour interactions: without assuming any particular distribution of the data, they group together contiguous set of points when the local density is larger than a certain value. For example, the super-paramagnetic clustering is a stochastic algorithm of this kind [8]; it was initially developed in the context of statistical mechanics and it has been subsequently introduced in the neural spike framework by [64].

Finally, an important distinction in spike sorting algorithms must be pointed out: some methods perform an unique clustering phase that classifies all spikes under analysis; whereas some other methods perform first a partial clustering — in order to identify characteristic features and/or representative waveform of involved neurons — and then, in a second phase, they reach the final and complete classification of all spikes — scanning all events one by one and determine which cluster best describes them. The method presented in this thesis adopts this second approach, which can be find also in [77] [68] [67] [74].

2.1 Basic Spike Sorting Methods

This section introduces classical and widely used approaches to spike sorting. Those methods have been firstly and mainly thought for recordings with one single channel and they are straightforwardly extended to the multichannel situation.

Amplitude and Window Discriminator — The first and simpler spike sorting method, born when data acquisition and analysis systems where still analog, is the Amplitude Discriminator: each detected event is associated to its own peak amplitude and classification is done grouping similar values of this parameter. For example [28] considers histograms of spike amplitude, and then classifies spikes on the base of different peaks in the histograms. Obviously, the underlying assumption is that each neuron generated spikes with different amplitudes that do not change over time. The main difficult arises when spikes from different neurons have nearly equal amplitudes.

A natural improvement, applicable when spikes with the same peak amplitude have instead different waveforms, is the Window Discriminator: classification is based on spikes crossing one or several windows, i.e. according to whether spike amplitude is above or below some reference levels at particular instants (see, for example [14][73][71]). In addition to spike amplitude other parameters may be considered: for example, in case of biphasic spike shape, the time between two peaks and the time from the first peak to the next zero crossing have been used [55].

In general, those simple techniques are fast and can be implemented on-line. For this reason their approach is still employed in commercial acquisition systems although the significant degradation of their performances when signal to noise ratio is low or when spikes overlap. One more disadvantage is that windows discriminators require an user to manually set or readjust the reference windows and this do not allow to work with more than few channels simultaneously. Finally they may miss sparsely firing neurons and obviously they make mistakes when neurons fire spikes with different amplitudes.

Template Matching — Another way to separate spikes corresponding to different neurons is the Template Matching. First of all, this method individuates a set of template waveforms, i.e. a set of characteristic spike shapes — one for each neuron — that may be recorded spikes or averages of spikes. Then, each recorded spike is compared — through a similarity or dissimilarity measure — with all template waveforms, and it is assigned to the corresponding neuron. This method was first presented in [35] where the user selects the templates by means of a visual inspection, and the spikes are sorted using a mean square distance metric. Many variations of Template Matching technique have been presented. For example, in [6] a fixed template is used to scan data to find events similar to the template both in shape and amplitude; in [1] the cross-correlation technique finds events of different amplitude with shapes similar to the template; in [21] both amplitude and similarity are used for detection of events.

In earlier and simpler methods a human operator has to manually choose a small set of spikes that would serve as templates, with consequent problems of biases and ill-defined selection criteria [35] [6]. Spike template can also be chosen automatically: for example, in [77] it is described a template learning phase where two features are extracted from each events and then clustered in order to determine an averaged waveform for each recorded neuron; in [70] templates are constructed automatically by combining principle component analysis and K-means clustering; in a more sophisticated approach, templates are modeled in the high-dimensional space of the sampled waveform as continuous piece-wise linear functions plus white noise with Gaussian distributed amplitudes [47].

Also the classification phase presents many variations: the algorithm in [27] employs the nonparametric nearest neighbor technique using the Euclidean metric to calculate the distance to the template, so assuming a spherical cluster around the template; the Bayesian approach of [47] takes instead into account the variation around the mean spike shape to give more accurate decision boundaries; in [4] data are passed through a whitening filter before matching filtering and there is a comparison between squared euclidean distance and city block distance; the method proposed in [18] uses the Levenberg-Marquardt algorithm to estimate the variances of each cluster based on distribution shape, and a weighted Euclidean metric to refine the classification results.

Unlike the Window Discriminator, the Template Matching strategy has the advantage to consider the whole spike waveform and not only some of its points. In general, the technique is quite simple to be nowadays implemented on-line, and there are even some attempt to improve its speed: for example [70] curtails the matching process when the distance between the template and waveform exceeds some threshold. Nevertheless, Template Matching methods, being strictly related to the spike waveform, assume a stationary spike shape and fail when neuron spikes change their waveforms (in case of bursting neurons or electrode drift) and when distinct neurons has similar spike shape. Moreover, classification errors arise when spikes highly overlap; there are anyway some variations, as will be shown in the following, that attempt to resolve the overlapping problem.

Principal Components Analysis — The Principal Component Analysis (PCA) finds an orthogonal linear transformation that transforms the data to a new coordinate system such that the first coordinate (called the first Principal Component) accounts for as much as possible of the variability in the data, and each succeeding component accounts for as much as possible of the remaining variability. In the spike sorting context, one of the most used method for automatically choosing features is to take the first 2 or 3 Principal Components usually containing more than 80% of the energy of the signal and capturing the directions of largest variation in the data. Some examples about application of this methods are in [37] [36] [2] [34].

Methods based on Principal Component Analysis, as Template Matching methods, found their classification power on spike waveform discrimination assuming a stationary spike shape. As a consequence, errors arise when neuron spikes change their waveforms (in case of bursting neurons or electrode drift), when distinct neurons has similar spike shape, and when spikes highly overlap. Considering Principal Components means essentially summarizing all waveform information in few parameters with maximum variance. As each compression procedure they may loose information, therefore it is not surprising that a study, comparing spike sorting methods, found the Principal Components not as accurate as Template Matching, although faster [76]. The same study found also that Principal Components yield more accurate classification than other features. Nevertheless, Principal Component Analysis selects the directions of maximum variance of the data, which are not necessarily the directions of best waveform separation. In other words, it may well be that the information for separating the features clusters is in some lower Principal Components that are disregarded.

2.2 Methods based on Wavelet Transformations

Since neural spikes are functions well localized both in time and frequency, Wavelet Analysis should provide a good signal representation for them: i.e. the spike energy should be concentrated in few Wavelet coefficients, thus resulting in a separation of the signal from the noise (spreads in more coefficients). See, as an example, the Wavelet Packet decomposition of a neural spike in Figure 2.2. Moreover, Wavelet Transformations supplies almost uncorrelated noise coefficients, variable time-frequency resolution, and have low computational complexity. In particular, the Wavelet Packet (WP) Transformation, that allows a more adaptable time-frequency resolution, is a very promising approach and it has been employed in the new spike sorting algorithm, introduced in Chapter 3. This section presents, for the sake of completeness, some other recent algorithms based on Wavelet Analysis.

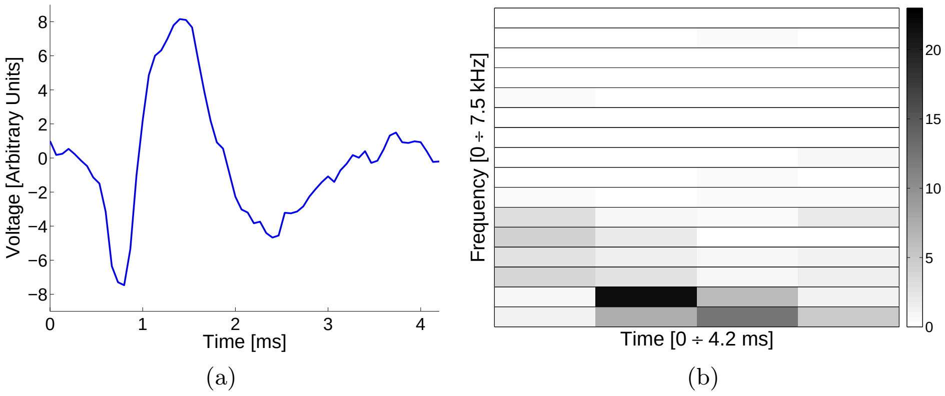


Figure 2.2: In subfigure (a), 64 samples of a neural spike recorded in the locust (*Schistocerca americana*) antennal lobe. The vertical scale is voltage in arbitrary units. In subfigure (b), the corresponding Wavelet Packet Transformation, in absolute value, over a complete tree of depth 4, Symlet of order 5 as mother wavelet. Notice that the signal energy is concentrated in few of the 64 Wavelet Packet coefficients.

The pioneer paper [79] describes a method based on a shift-invariant Wavelet Transformation and a particular "magnitude" and "phase" representation of the transformed signals. The magnitude vector is obtained by simply sorting the values of the Wavelet coefficients vector in increasing order. The phase vector is obtained by indicating the position that each component of the magnitude vector had prior to sorting. The main advantage of this representation is that it is phase (shift) invariant: a transient signal and all of its time-shifted versions have identical magnitudes. The phase-corrected approximation of templates is then compared with each recorded spike.

The method presented in [46] is based on a visual selection of m Discrete Wavelet coefficients with large standard deviation, large average, and a bimodal o multimodal distribution of the detected spikes population. According to the authors, these coefficients should exhibit the greatest potential to differentiate spikes. Then clusters are manually delimited in bi-dimensional projections plots of selected Wavelet coefficients.

In [41], using the Local Discriminant Bases with a Shannon's mutual information maximization, authors select 9 Wavelet Packets, tested for many different spike waveforms on different electrodes in different experiments, that are able to enhance differences between two spikes and best separate them. According to the authors, these wavelet packets appear universal for recording of this kind. The spike classification is carried out clustering Wavelet coefficients in the 9-dimensional space.

The method in [64] combines Wavelet Transform with Superparamagnetic clustering. Primarily, from each spike, the first 10 Wavelet coefficients with the largest deviation from normality are chosen as distinctive features then Superparamgnetic clustering, based on simulated interactions between each data point and its nearest neighbors, classifies the spikes according to the selected set of Wavelet coefficients.

The method described in [45] is based on the Discriminant Pursuit algorithm. It extracts a reduced set of features from the Wavelet Packet decomposition on the series of residual difference vectors. According to the authors, these features turn out to be an efficient representation that can be subsequently analyzed with statistical classifiers.

In [60] it is presented a method for features extraction based on a combination of Principal Component Analysis and Continuous Wavelet Transform that aims to compensate faults of the two approaches in the capture of spike structure. First, it finds a representative waveform for each cluster in the Principal Component space, averaging spike waveforms in a small neighborhood of each cluster center. Then, it applies the Wavelet Transform to all representative waveforms and searches for the scale and translation parameters that maximizes distance between representative waveforms in the first and second halves of the spike time window. Finally, for all spikes, it evaluates the Wavelet Transformation value for the two found parameter sets thus extracting features.

All previous methods in this section have been conceived for single-electrode recordings and their use of the Wavelet Analysis tool has been in order to efficiently describe the spike waveform: essentially, they aim to capture those aspects of the waveform structure that best allow to differentiate spike shapes. Therefore, all these algorithms do the same a priori assumptions of methods based on Matched Filter and Principal Component Analysis: in particular they need stationary different waveforms associated to different neurons. As a consequence they have, in general, difficulties to resolve overlapping spikes when the superposition is complete.

The new method proposed in this thesis exploits the advantages of Wavelet Analysis in a different way from methods previously described. The idea is to code amplitude information rather than waveform information, and thus Wavelet Packet Transform is used to fairly estimate amplitude ratios between waveforms recorded by several sensors.

2.3 Multichannel Methods

In the development of spike sorting algorithms, the most significant progress is been the evolution of recording techniques that introduced multiple electrodes. Many problems encountered with single-electrode recordings vanish with multiple-electrode recordings: the result is an increasing of the discriminant power and a dramatic improvement of spike sorting outcomes, as shown in [53], with two electrodes (stereotrode), and in [30] [39] [40], with four electrodes (tetrode). In fact, having multiple recordings of the same event from distinct positions allows additional information to be used for more accurate spike sorting; for example, spikes from distinct neurons may appear similar, in amplitude and shape, when recorded by a particular sensor, but they differ on recordings by sensors in different physical locations. Even the new algorithm proposed in this thesis, exploits and combines information from more channels. As a reference, this section describes some other methods using potentiality of multichannel recordings with different approaches.

Waveform features — Many spike sorting techniques have been developed for single electrode and then extended to multiple electrodes. For example, in case of algorithms that cluster spike waveform features, a natural extension is done extracting features separately from each recorded channel and combining them — eventually weighted according to different signal to noise ratio in different channels — to form the inputs for the clustering algorithm.

According to this line, the authors of paper [64], anticipate the generalization of their single-electrode method to tetrode recordings: they will select Wavelet coefficients, with the largest deviation from normality, not only from one channel but also from adjacent channels, and will group them by means of the Superparamagnetic clustering in a larger dimension space.

Principal Components methods can also be performed on multiple-electrode recordings to obtain features for clustering: for example [40] considers the first 3 Principal Components for each channel obtaining a 12-dimensional feature vector for each spike.

Intensity features — Nevertheless, when electrodes are closely spaced as in some commonly used tetrodes, the discriminant power of multi-channel recordings lives mostly in the different amplifications of the same recorded waveforms. This justifies the development of spike sorting algorithms that exploit multichannel amplitude features, rather than waveform features, in order to classify spikes. Actually, this kind of algorithms take account of information about neuron position, but they do it in a roundabout way.

The study in [39], that compares the performance of tetrodes with the best electrode pair and best single electrode, considers for each waveform the peak-to-peak amplitude (i.e., maximum-to-minimum amplitude), the integral of the spike waveform, as well as the spike width (defined as the time difference from peak negativity to peak positivity), and the first 2 Principal Components. This study founds that bi-dimensional projections plots of peak-to-peak amplitude (i.e., maximum-to-minimum amplitude) provided the greatest resolving power for the tetrode and the stereotrode data, while the principal components gave the best separation in the single-electrode data.

Even the method in [66] associates, to each spike, the 4 peak amplitudes in the tetrode channels and visualize them in 6 bi-dimensional projections. It also suppresses noise common to each of the 4 electrodes by means of the Hadamard transformation, that perform a scaling and a rotation in the 4-dimensional space giving less elongated clusters and allowing to reveal the differential part of the signal.

With the idea to consider a measure of the spike intensity more robust than the noisy peak amplitude, the method in [42] computes, for each spike, 4 power measures of the waveforms on the tetrode channels and classify events using the K-means algorithm on this 4-dimensional power space. The mean power values of each cluster are then used to compute the neuron location with respect to the tetrode position.

Concatenation — There is also another idea to approach the spike-sorting problem with multiple-electrode recordings: the two methods described in [32] and [26] concatenate the separate waveforms of various channels in an unique vector considered, in its entirety, as a single waveform for sorting purposes; the algorithm in [67] links instead together the Fourier Transforms of the voltage waveforms from each electrodes and sorts concatenated spectra.

Neuron position and measurement channel — Some other techniques conceived for multiple electrodes, take more directly into consideration the neuron position with respect to the sensors and/or the measurement channel between neurons and sensors. This approach, unlike those based on waveform and amplitude features, allows in principle to classify bursting cells with varying waveform and amplitude.

For example the algorithm in [17] computes, from tetrode recordings, the spatial position of neural spike activity and exploit its physical meaning to discriminate the spike sources. This algorithm, with the due assumptions on signal propagation through the extracellular medium, considers a system of equations that is function of the 4 positive peak amplitudes on tetrode channels. The solution of this system, using the Newton's iterative method, returns back the three spatial coordinates of the neuronal sources with respect to the tetrode tips. These approximations of spatial locations are grouped in clusters by the Self-Organizing Map algorithm [44], whose clustering performances are not dependent on the algorithm initialization. According to the authors, this spike sorting method performs better than the ones using tetrode peak amplitudes and it is roughly equivalent with the ones using amplitude ratios.

Another approach, that takes rather into account the measurement channel between neurons and sensors, is the one in [68], where the sorting is done exploiting properties of the surrounding tissue and the resulting differences in how signals are measured on different sensors. The first step of this method classifies spikes in stereotrode data using ratios of transfer functions between the same neuron and two electrodes about $2\,mm$ apart. The sensors are distant enough to give rise to a perceptible frequency dependence in the transfer function ratio.

The multiple-electrode spike sorting problem can also be seen in the framework of Blind Source Separation (BSS), where the aim is to separate n independent signals that have been linearly mixed onto m channels, with poor information about the source signals or the mixing process [15] [20]. One of the classical method to successfully solve this problem, is the Independent Component Analysis (ICA) that is been applied also in the context of neural signals [11]. The idea behind ICA is to find the matrix that transforms the m recorded mixtures into n statistically independent signals that correspond to the source signals. This matrix corresponds to the inverse of the measurement channel, and its estimation is done without any attempt to directly identify the source positions. Nevertheless, besides the assumption that the sources are mixed linearly, the ICA technique has other two important limitations: it demands that the source signals are independent, and the number of observations (or electrodes) is at least equal to the number of recorded sources.

The method proposed in [74] tries to overcome the last limitation on the number of electrodes and sources, combining the ICA technique and the K-means clustering. It relaxes the limitation with the assumption that no more than m neurons fire simultaneously in a segment window, where m is the number of electrodes. The idea is to apply the ICA to subsets of events that presumably are generated by one neuron or a little fraction of those recorded in the entire data set. The subsets of events are created through the K-means clustering on maximum and minimum spike amplitudes. Since it is more reliably to create a number of cluster larger than the expected number of neurons, the method implements also an aggregation step where the ICA decomposition of each subset is compared in order to merge clusters that contain spike from the same neuron.

Another sophisticated method in the framework of Blind Source Separation, that attempts to deal with the case of correlated signal sources in the presence of strong correlated noise, is the spike-sorting technique initially proposed in [57] and detailed in [59]. It performs an undecimated (or Stationary) Discrete Wavelet Packet Transform of the multichannel spike waveforms, and select a characteristic Wavelet Packet subtree for each spike by searching for tree nodes having the spatial covariance matrix of their coefficient with largest subset of maximum singular values. The technique relies on separating sources with highest spatial energy distribution in each frequency subband spanned by the corresponding wavelet basis: the eigenvector coefficients, corresponding to the largest eigenvalue, would yield the spatial diversity of the communication channel as a basis for discrimination.

Also the method proposed in this thesis have been conceived for multiple electrodes and, like other methods in the framework of blind source separation, it takes into account the measurement channel between neurons and sensors. The approach, however, is different from ICA, since the new method directly estimates the components of the mixing matrix (i.e. the matrix that describes how neural signals combine in the recorded channels).

2.4 Methods for the resolution of overlapping spikes

One of the most challenging open problems in the spike sorting context is the one of overlapping spikes that occur when two or more neurons fire very close in time: an efficient spike sorting algorithm must be able to determine how many and which neurons have contributed to the superposition.

The majority of the developed procedures do not explicitly deal with such kind of events: they assume to treat only single spikes (i.e. non-overlapping spikes) and limit their application to data set where the superposition rarely occurs. Although these aforementioned methods may correctly classify superimposed spikes if they are sufficiently separated in time, their performance degrades severely when the spikes fire simultaneously, giving rise to a completely different shape (see Figure 1.6(a)). Some of these approaches can individuate overlapping spikes, thus having some gauge of their frequency and potentially keeping them outside the sorting process: for example, spike with multiple close peaks or outlier points far from clusters center may be labeled as overlapping spikes.

Nevertheless, there are some techniques that directly deal with overlapping spikes. This section describes some approaches that attempt to recover spike superpositions.

Comparison among all linear combinations — The most standard solution is to evaluate the distance between each recorded waveform and all possible linear combinations of the previously identified templates with all possible delays — multiple of the sampling time — in a certain range. Metrics used with this intent are, for example, the Euclidean distance in [63] [4], and the city block distance in [9]. When the minimal distance among all combinations is below the acceptance threshold, then the templates corresponding to that combination are considered as the constituent of the overlapping spikes.

Under Gaussian noise assumption, another approach for recover overlaps is the one described in [3] that selects, among all possible combinations of spike models with all relative timings of their peaks, the combination with the maximum likelihood. A similar approach is been exploited also in [62], [78] and, from the spectra point of view, in [67].

The drawback of these procedures is that they are computationally expensive when the number of recorded neurons is large and moreover when they consider overlapping spikes with contributions from more than two neurons.

Waveform subtraction methods — Another simple approach to decompose overlaps is to subtract spikes from the recorded waveform once they have been classified, in the hope that this will improve the classification of remaining spikes. This approach requires templates of the spikes. It gives reasonable results when spikes are separated well enough so that at least one can be accurately classified, but fails when spikes are closer. Another problem with this approach is that the subtraction can introduce noise or spurious spike-like shape if the spike model or the occurrence time are not accurately estimated.

Under the assumption of stationary and Gaussian noise, a sophisticated approach has been proposed in [47]. It uses k-dimensional search trees to quickly find the possible combinations of spike waveform that could account for a given recorded spike. For each detected event, the method selects a region around the first peak, fits this region with a spike model and subtracts the model from the recorded samples thus removing the contribution of its waveform tail. The method continues by fitting the residual waveform (i.e. the raw data minus the model) until other peaks are found. The algorithm returns for each event a list of all plausible spike combinations along with their associated probability.

Neural Network — Another solution for the overlapping spike problem can be found in [16], where neural networks have been used. A serious drawback, however, is that the network must be trained using labeled spikes; thus the learned decision boundaries are as accurate as the initial labeling. ling. Like the subtraction methods, these method can only identify overlaps that have separated peaks.

Wavelet Packet decomposition — An alternative tool to resolve superposition is the Wavelet Packet decomposition, as long as the constituent spikes of the superposition are separated in the time-frequency plane.

In [41] it is shown that by using the same Wavelet Packet coefficients that are employed for single spike classification, a number of overlapping spikes can also be classified, if multiple peaks can be detected.

A more sophisticated approach for multichannel recordings is given in [58][59]; it exploits, in addition to the Wavelet Packet decomposition, also the eigendecomposition of the sample covariance matrix of the transformed observation matrix.

Multichannel approaches for overlapping spike resolutions — Surely, having multiple recording of the same spike gives additional information useful to decompose spike superpositions. An example of multichannel solution to the overlapping problem is the ICA. As shown in the previous section, the method introduced in [74] tries to overcome the ICA limitation on the number of electrodes and sources using K-means clustering. Under the assumption that the number of simultaneously firing neurons is lower or equal to the number of sensors, the method attempts to separate the original spike trains thus resolving overlaps. The method plans, first of all, a division of spikes between those in “stable condition” and those in “unstable condition”; this two set of spikes are then considered one after the other. In the stable condition, spike waveforms show isotropic Gaussian variability, and their density in the feature clustering space is large enough that they can be sorted, by K-means, in many initial clusters that presumably contain spikes generated by a single neuron; afterwords, through ICA, clusters are aggregated and identified. In the unstable condition, there are instead a small number of spikes with irregular waveforms that are sorted as well (they can be generated by neurons that show waveform variability or they can be overlapping spikes); although they cannot be sorted in clusters containing only one single-neuronal activity, the authors assume that each cluster contains a combination of few similar single neuronal activities, thus allowing the ICA application and the overlapping resolution.

An alternative multichannel approach is shown in [19]. First, it classifies the recorded spikes by means of any existing spike-sorting algorithm; such initial clustering does not concern about misclassified overlapping spikes under the assumption that their number is much smaller than that of non-overlapping spikes. Then, the method estimates the matrix that describes the linear mixing of single neural activities into the multichannel recordings: coefficients of each column are represented by normalized average amplitudes of the multichannel waveforms in each cluster. The estimated mixing matrix allows to calculate a set of projection matrices that eliminate, from the measured data, one neuron activity at the time. Running a second spike sorting on each projection, and combining the results of the two spike sortings, the method can resolve all spike superposition involving two neurons.

Like the last two methods ([74] and [19]), the algorithm presented in this thesis aims to resolve overlapping spikes exploiting multichannel recording. Section 3.5 highlights differences of the new algorithm with respect to these other two approaches.

Chapter 3

Multi-Channel Inversion for Spike Classification

This thesis proposes a new method, named Multi-Channel Inversion for Spike Classification (MCI4SC), to sort neural spikes recorded by multichannel probes. As well as explaining the main ideas and the algorithm details, this chapter also illustrates the linear mixing model adopted for recorded measurements, the necessary assumptions, and the preliminary steps. The application to experimental data is shown in the next chapter.

3.1 Assumptions and model for recordings

The MCI4SC algorithm, as it is presented in this thesis, works on multichannel recordings from electrodes that are close enough to guarantee the following assumptions:

1) **the neural signal is recorded simultaneously in each channel**, i.e. the signal arrives with the same delay on each sensor;

2) **each channel records the same spike waveform**, i.e. signal distortions among sensors are negligible.

For many of the available data sets the assumptions hold. For example, the widely used tetrode from the Center for Neural Communication Technology of the University of Michigan (also known as Michigan probe), has a diagonal length equal to $50\,\mu m$ [30]. Electrical neural signals traveling in the extracellular medium, usually sampled with a frequency of order $10^4\,Hz$, are recorded by this probe with a negligible delay among electrodes, so that spikes can be considered aligned in time on the different synchronized channels (thus satisfying the first assumption). Sensors proximity, like that of Michigan probe, bounding the relative group velocity dispersion, allows as well to assume similar transmission channels between a certain neuron and each sensor, so that all electrodes provide the same waveform images, with different amplification (second assumption is satisfied, too).

Under this hypothesis on closely spaced electrodes, and reasonably assuming that **the medium between neurons is linear** so that signals in each measurement sum linearly, an m-channel recording of n neurons ($m \lesseqgtr n$) can be simply represented as a linear mixing model (see Figure 3.1):

$$x(k) = As(k), \quad k \in \mathbb{Z}, \tag{3.1}$$

where the column vector $x(k) = [x_1(k), \dots, x_m(k)]'$ denotes the noise-free multichannel recording, the matrix

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \tag{3.2}$$

is the m by n *mixing matrix* that describes the measurement channel between neurons and sensors, and $s(k) = [s_1(k), \dots, s_n(k)]'$ denotes the snapshot of single neuronal activities. To be more precise, each signal $s_j(k)$ is the delayed and distorted version of the original source signal with equal delay and distortion on all sensors.

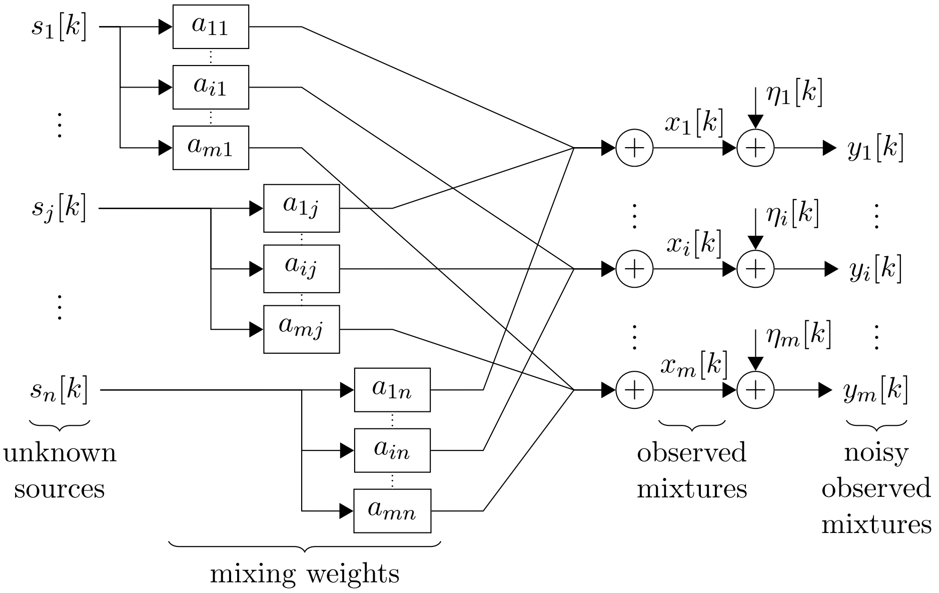


Figure 3.1: An illustration of the linear mixing model that represents the recording of n neural signals with m electrodes. The n unknown sources are linearly mixed to form m observed mixtures; $s_j(k)$ indicates the sampled activity of the j-th neuron as successions of spikes, $x_i(k)$, $\eta_i(k)$, and $y_i(k)$ indicates respectively the noise-free observation, the additive noise, and the noisy data recorded on the i-th sensor.

Considering a finite set of samples ($k \in [1, 2, \ldots, K]$), the model in Eq. (3.1) can also be expressed in matrix form as:

$$X = AS, \tag{3.3}$$

where $X^{m \times K} = [x(1), \ldots x(K)]$ is the noise-free multichannel recording, and $S^{n \times K} = [s(1), \ldots s(K)]$ is the sampled neural activity.

Since the interest is just in spikes occurrences, and not in reconstruction of the source model and its emitted waveform, the measurement model of Eq. (3.1) and the MCI4SC algorithm do not require absence of delays and distortions introduced by the transmission channels between neurons and sensors, provided that, for every neuron, delay and distortion are the same in all electrodes. Given that each channel is affected by the same distortion rate, the mixing matrix components a_{ij} are constant in frequency. Nevertheless, in some recordings — due to the large distance between sensors, high anisotropy and inhomogeneity of the extracellular space, or no validity of the point source approximation — distortion and delays can even be different from one sensor to another with respect to the same source. When these differences are significant, the measurement model and the MCI4SC algorithm can be opportunely extended by introducing proper traces realignment in correspondence to each spike and frequency-dependent mixing matrix components.

In the present work, another assumption is that the mixing matrix components a_{ij} are, not only constant in frequency, but also constant in time, and the measurement channel described by Eq. (3.1) is stationary. This aspect is guaranteed if **the probe holds steady with respect to neurons during all recording time**. During the measurement, however, electrodes may drift to a new position as the neural tissue settles in response to pressure from advancement of probe. If the change is slow, the MCI4SC algorithm can be reapplied on contiguous segment of data where the measurement channel can be considered stationary.

Equation (3.1) describes a noise-free model, but, since recordings are corrupted by noise, they are better describe as follows (see Figure 3.1):

$$y(k) = x(k) + \eta(k) = As(k) + \eta(k), \quad k \in \mathbb{Z}, \tag{3.4}$$

where $y(k) = [y_1(k), \ldots, y_m(k)]'$ is the noisy multichannel recording, and $\eta(k) = [\eta_1(k), \ldots, \eta_m(k)]'$ is the additive noise in the measurement channel. Considering a finite set of samples ($k \in [1, 2, \ldots, K]$), recordings can also be described in the following matrix form:

$$Y = X + H = AS + H, \tag{3.5}$$

where $Y^{m \times K} = [y(1), \ldots y(K)]$ is the noisy multichannel recording, and $H^{m \times K}$ equal to $[\eta(1), \ldots \eta(K)]$ is the additive noise in the measurement channel.

As with many other spike sorting algorithms, MCI4SC makes the following assumptions on the noise present in the recorded data:

- **the noise sums linearly with the signal**;
- **the noise is statistically independent of the signal**;
- **the noise is white, Gaussian, and stationary**; if not white, the measurement can be whitened.
- **the noise is zero-mean**; if not, the estimated mean is subtracted from data to obtain a zero-mean noise.

These assumption are quite well satisfied by that part of the noise coming from recording hardware. The neural signal, however, is corrupted also by environmental noise due to surrounding neurons that perturb the extracellular medium at each firing. This background noise can be statistically dependent of the signal, and thus not white Gaussian nor stationary. In those recording where the environmental noise is predominant and these assumption are violated, the MCI4SC algorithm capability to well estimate the mixing matrix may degradate, as well as the sorting result.

Finally, MCI4SC algorithm requires the following assumption regarding the neural activity:

- **Each neuron has a different relative position with respect to electrodes**; this assumption is certainly guaranteed when the neural signal is recorded by at least four electrodes that are not located in a plane. In the unlikely event that two neurons are at the same relative distance from sensors, they are indistinguishable from the MCI4SC point of view, unless the spike shape can be used to discriminate different neurons.
- **Each neuron fires a finite set of spike waveforms**; under the hypothesis that all neurons are in distinguishable positions, neurons that fire spikes with different waveforms can be handled by MCI4SC algorithm if these waveforms are in a finite number. In case of a possible ambiguity in neuron positions with respect to electrodes, a human intervention is required to evaluate whether spikes with different waveforms come from an unique neuron or more.
- **Each neuron fires more than one time**; since, like other algorithms in this framework, also the MCI4SC algorithm needs to cluster spikes from the same neuron, it requires therefore a minimum number of recurrences for each neuron, and for each of its spike shape.
- **No more than m neurons fire simultaneously**, where m is the number of employed sensors. If a larger number of neurons fire at the same time, the MCI4SC algorithm itself is unable to correctly decompose the overlap. The probability to have superposition of more than m neurons depend

obviously on the firing frequency, but these occurrences are as rarer as m is larger.

3.2 Main ideas behind the MCI4SC algorithm

The MCI4SC algorithm has two distinct phases: i) the *learning phase* estimates the number n of neurons contributing to a particular recording, their waveform templates, and the mixing matrix A that represents the measurement channel between neurons and sensors; ii) the *classification phase*, inverting matrices derived by A, analyzes each detected event with the aim of associating the spikes with the corresponding firing neurons and determining the times of occurrence, even in case of overlapping of up to m spikes.

The whole algorithm is centered on the mixing matrix A. It is worth noticing that each of the n columns $A_j = [a_{1j} \dots a_{mj}]'$, $j = 1, \dots, n$ is related to one particular neuron. To be more precise the m column components are the coefficients that multiply the same neural signal s_j in each of the m recorded traces:

$$\begin{bmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{bmatrix} = \sum_{j=1}^{n} \left(\begin{bmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{bmatrix} s_j(k) \right) = \sum_{j=1}^{n} A_j s_j(k). \tag{3.6}$$

Learning phase — A crucial point is that the sparsity and the finite duration of neural spikes allow to isolate intervals $[K_i, \dots, K_f]$ in which the data contain the signal coming from just one neuron h, while other neural signals are equal to zero ($s_j(k) = 0$, $\forall j \neq h$, $k \in [K_i, \dots, K_f]$):

$$\begin{bmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{bmatrix} = \begin{bmatrix} a_{1h} \\ \vdots \\ a_{mh} \end{bmatrix} s_h(k) = A^h s_h(k), \quad \text{if} \quad k \in [K_i \dots, K_f]. \tag{3.7}$$

It is worth noticing that this condition is specific to the neural recordings framework, and it does not apply to other relevant contexts.

When in a free-noise segment of data there is only one neural signal s_h, it is possible to calculate $m-1$ ratios of the m mixing matrix components belonging to the same column h, as ratios of spike amplitude, or more in general as ratios of spike samples:

$$\frac{a_{ih}}{a_{rh}} = \frac{a_{ih} \cdot s_h(k)}{a_{rh} \cdot s_h(k)} = \frac{x_i(k)}{x_r(k)}, \quad \forall k \in [K_i \dots, K_f] \text{ and } k | s_h(k) \neq 0, \tag{3.8}$$

where $r \in \mathbb{Z}$ denotes the reference channel ($1 \leq r \leq m$), and i is equal to $1, \dots, m$ with $i \neq r$; when $i = r$ the ratio is equal to 1 by definition. Notice that the ratios in Eq. (3.8) can be calculated without directly considering the effective waveform s_h because, due to assumptions in Section 3.1, it appears equal in all channels. Moreover, spike-sample ratios $x_i(k)/x_r(k)$ are invariant with respect to spike waveform variations typical of bursting neurons, as well as they are invariant to amplitude changes from spike to spike.

Component ratios calculated in Eq. (3.8), allow to construct a mixing matrix $\tilde{A}$, that is a particular normalization of the original mixing matrix A, where each column is rescaled according to a different and arbitrary multiplicative factor:

$$\tilde{A}^{m\times n} = \begin{bmatrix} a_{11}/a_{r1} & a_{12}/a_{r2} & \cdots & a_{1n}/a_{rn} \\ \vdots & \vdots & & \vdots \\ a_{m1}/a_{r1} & a_{m2}/a_{r2} & \cdots & a_{mn}/a_{rn} \end{bmatrix}. \tag{3.9}$$

The introduction of these n freedom degrees in the mixing matrix, is like considering a linear mixing model whit an arbitrary scaling of each neural signal:

$$x(k) = \sum_{j=1}^{n} \left(\begin{bmatrix} a_{1j}/a_{rj} \\ \vdots \\ a_{mj}/a_{rj} \end{bmatrix} a_{rj} \cdot s_j(k) \right) = \tilde{A}\tilde{s}(k), \tag{3.10}$$

where $\tilde{s}(k) = [a_{r1}s_1(k), \ldots, a_{rn}s_n(k)]' = \text{diag}(a_{r1}, \ldots, a_{rn}) \cdot s(k)$ is the scaled neural signal; and $\tilde{S}^{n\times K} = [\tilde{s}(1), \ldots \tilde{s}(K)]$ is the corresponding scaled signal in the matrix form, when a finite time interval ($k \in [1, 2, \ldots, K]$) is considerd. The only lost information, with this column rescaling, is the actual spike amplitude at the source, that it is not strictly required to classify spikes, since a neuron can emit spikes with different amplitudes from time to time. Therefore, this arbitrary normalization of mixing matrix columns does not affect the subsequent spike sorting analysis based on mixing matrix $\tilde{A}$ instead of A. On the other hand, disregarding actual spike amplitudes, the methods is robust to gain calibration errors of sensors; moreover, it does not care about the attenuation coefficient of the medium where signal is propagating, nor the exact mutual electrodes position.

In the analysis of experimental data, it must be considered the noisy recording model $Y = X + H$, and hence ratios of mixing matrix components are not calculated, but estimated from the data samples $y_i[k]$. To obtain a more accurate estimation it is appropriate to consider, for each neuron, a sufficient number of not-overlapping spikes, and to consider, through average operations, more samples for each spike.

In conclusion, the learning phase of MCI4SC, consists in:

A1. Selection of data segments with spikes generated by the activity of a single neuron (called *single spikes* in the following).

A2. For each single spike, estimation of spike-amplitude ratios between different channels.

A3. Single spikes sorting that associate each single spike to the neuron of origin.

A4. For each neuron, estimation of the corresponding normalized mixing matrix column, and of the corresponding characteristic waveform templates.

An important step, in the above mentioned procedure, is the sorting of single spikes. Notice that the $m - 1$ ratios of Eq. (3.8), used to estimate each column of the mixing matrix $\tilde{A}$, can also be used as distinctive features to cluster single spikes: they are ratios of spike amplitudes in different channels, and they are related to the characteristic neuron positions, as already said in Section 1.1.

It is also worth underlining that, in the learning phase, just single spikes are clustered, thus carrying out a partial sorting. Only with the subsequent classification phase, and the overlaps resolution, the complete sorting of all spikes is achieved. This division in two phases (learning and classification) allows to initially considering a subset of spikes that can be correctly classified even without knowledge about the measurement channel, and that it is easier to handle when it is by itself. In fact, single spikes naturally create well defined clusters, while overlapping spikes, coming from a huge number of possible combinations, in general tend to create scattered and confusing clouds of features that disturbs the single spike clustering (see Figure 3.2).

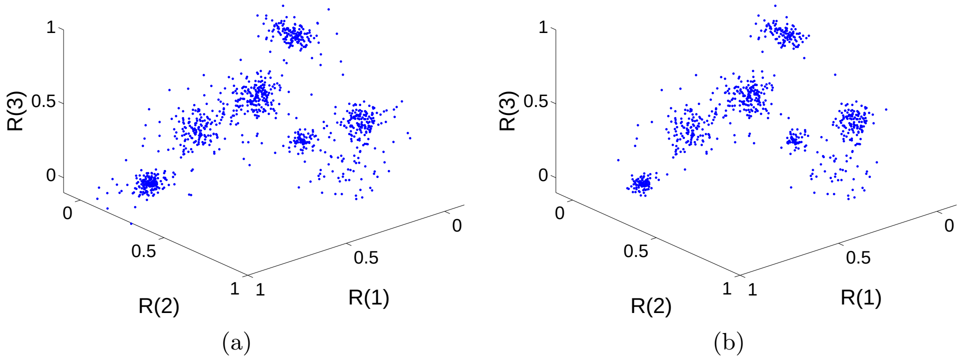


Figure 3.2: Tri-dimensional plot of three spike-amplitude ratios from tetrode recording in the locust (*Schistocerca americana*) antennal lobe. In (a), spike-amplitude ratios of all detected spikes in the recordings (990 spikes). In (b), spike-amplitude ratios of that part of detected spikes considered as single spikes (745 spikes).

Classification phase — When recordings are done with as many sensors as recorded neurons ($m = n$), the respective mixing matrix is square and its inversion is straightforward, if there are no ill-conditioning problems. In this favorable case, once the mixing matrix $\tilde{A}$ is been estimated, it is possible to separate original source signals ($\tilde{S} = \tilde{A}^{-1}X = \tilde{A}^{-1}\tilde{A}\tilde{S}$), and automatically resolve spike superpositions finding spike occurrences in each reconstructed neural trace.

Nevertheless, with the current state of technology, the number of recorded neuron is larger than the number of sensors that can be used ($n > m$), thus the mixing matrix has more columns than rows, and the inverse matrix is not defined. Some other methods attempts to overcome the limitation of having less sensors than neurons and a rectangular mixing matrix. For example, the algorithm in [74] uses the ICA on subset of spikes generated by a fraction of the total recorded neurons, while the algorithm in [19] eliminates one neuronal activity at the time, by means of projection matrices, thus resolving overlapping of at most two spikes. Anyway, both these two methods avoid to directly invert the measurement channel, as instead MCI4SC algorithm does.

A generalization of the inverse matrix, that may be defined even in case of rectangular matrices, is the Moore-Penrose pseudoinverse. The pseudoinverse of a real m-by-n matrix A, is defined as the unique n-by-m matrix $A^\dagger$ satisfying the following four conditions: i) $AA^\dagger A = A$; ii) $A^\dagger AA^\dagger = A^\dagger$; iii) $(AA^\dagger)' = AA^\dagger$; iv) $(A^\dagger A)' = A^\dagger A$.[1] See [5] and [7] for a more detailed discussion on generalized inverses theory. The pseudoinverse is successfully applied to solve overdetermined systems of linear equations ($n < m$), since it provides the least square solution. When $n > m$, however, the problem is undetermined, and the reconstruction of the n original signals $\tilde{S}$, once they have been mapped by a linear transformation matrix $\tilde{A}^{m\times n}$ with rank lower than n, cannot be achieved applying $\tilde{A}^\dagger$ to the measurements $X = \tilde{A}\tilde{S}$: each original signal still appears in more than one row of $\tilde{A}^\dagger X$, thus remaining mixed to the others. (see Figure 3.4).

Fortunately, even in this case, the not simultaneous activation of all neurons, allows some alternative solutions. Since, typically, in the segment of data around each detected spike there is only a part of all neural signals present in the whole recordings, it is possible to slightly modify the recording model in Eq. (3.3), disregarding some mixing matrix columns associated to neurons actually inactive in each time interval. In order to select, from time to time, the necessary mixing matrix columns, thus considering as present or absent the corresponding neuron activities, it is convenient to introduce a new matrix $C^{n\times n} = \mathrm{diag}(c_1, \ldots, c_n)$, with

$$c_j = \begin{cases} 0 & \text{if neuron } j \text{ is selected} \\ 1 & \text{if neuron } j \text{ is unselected} \end{cases}, j = 1, \ldots, n, \tag{3.11}$$

where $\nu = \sum_{j=1}^{n} c_j \leq n$ is the total number of selected neurons. In each data segment, if C selects at least all the active neurons, the free-noise recording

[1] A computationally simple and accurate way to get the pseudoinverse is by using the singular value decomposition: if $A = U\Sigma V'$ is the singular value decomposition of A, then its pseudoinverse is $A^\dagger = V\Sigma^\dagger U'$; where the pseudoinverse of the diagonal matrix Σ, is obtained by taking the reciprocal of each non-zero element on the diagonal, and leaving the zeros in place. In numerical computation, only elements larger than some small tolerance are taken to be nonzero, and the others are replaced by zeros. If A has full row rank, then AA' is invertible and an explicit formula for the Moore-Penrose pseudoinverse is: $A^\dagger = A'(AA')^{-1}$.

model already introduced, $X = \tilde{A}\tilde{S}$, is exactly equivalent to the following model:

$$X = \tilde{A}C\tilde{S}. \tag{3.12}$$

On the other hand, the two previous models are no more equivalent when at least one neuron contributing to the data segment X is excluded by the matrix C.

The key point is that by calculating the Moore-Penrose pseudoinverse of the m-by-n matrix $\tilde{A}C$, instead of the matrix $\tilde{A}$, only the ν selected columns of $\tilde{A}$ are involved: this is equivalent to inverting the mixing matrix that represents the measurement channel between the same m sensors and only ν neurons. When the number ν of selected columns is lower or equal to m, the problem is no more underdetermined, and if the selected columns include all neurons effectively present in the segment of data (as already said, they must be no more than m), the original traces are separated by simply applying $(\tilde{A}C)^\dagger$ to the measurements X. In the following, they have been considered only matrices $C \in \mathcal{C}$ with ν exactly equal to m, thus keeping the rank of matrix $\tilde{A}C$ as larger as possible and at most equal to m:

$$\mathcal{C} = \{C \in \mathbb{R}^{n\times n} | C = \text{diag}(c_1, \dots, c_n), \text{ with } c_j = 0, 1 \text{ and } \textstyle\sum_{j=1}^{n} c_j = m\}. \tag{3.13}$$

To give an intuitive idea of how the pseudoinverses $\tilde{A}^\dagger$ and $(\tilde{A}C)^\dagger$ work, it is useful to consider the example of the simulated free-noise recording in Figure 3.3, where the two neurons N_C and N_D are active in the same time interval; in the following Figures 3.4, 3.5, and 3.6, there are 3 attempts to disentangle the N_C and N_D neural signals, by applying different Moore-Penrose pseudoinverses to the simulated recording.

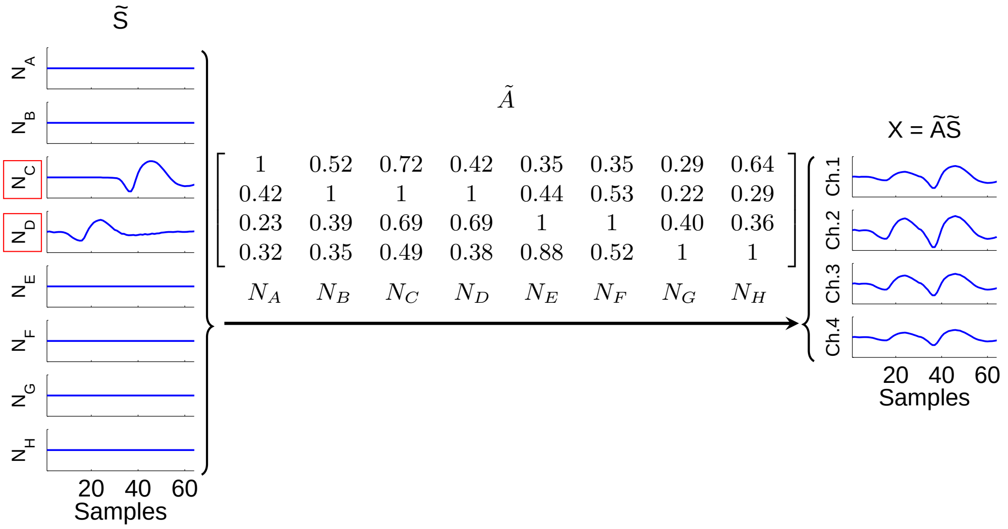
The Figure 3.3 shows a time interval where 2 of the 8 neurons are active. The mixing matrix $\tilde{A}$ mixes the 8 neural traces in 4 free-noise measurements.

The Figure 3.4 shows that the pseudoinverse $\tilde{A}^\dagger$, applied to the simulated measurements of Figure 3.3, is unable to separate and reconstruct the two original signals. The two waveforms are rather spread on more than one trace, thus remaining mixed together.

The Figure 3.5 shows, instead, that the pseudoinverse $(\tilde{A}C)^\dagger$, with C selecting both the two active neurons N_C and N_D, is able to separate and reconstruct the two original signals.

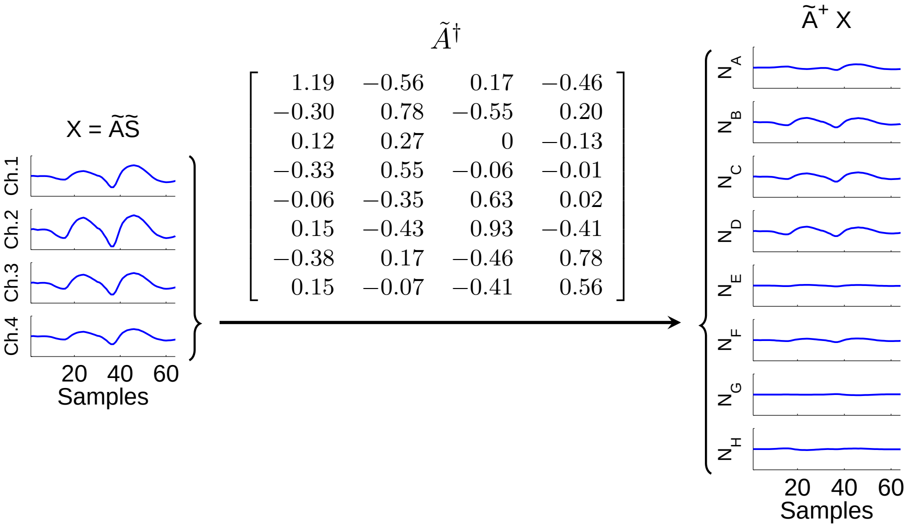
Finally, the Figure 3.6 shows that the pseudoinverse $(\tilde{A}C)^\dagger$ is unable to unmix the two original signals, when the matrix C does not select all the active neurons.

It seems that to disentangle original source it is necessary to know which neurons are effectively active in each segment of data under analysis; of course, this it is not known a priori, since it is actually the purpose of the whole procedure. Fortunately, there is another crucial point to exploit: the fact that the matrix C has selected, or not, all the active neurons is directly reflected in the



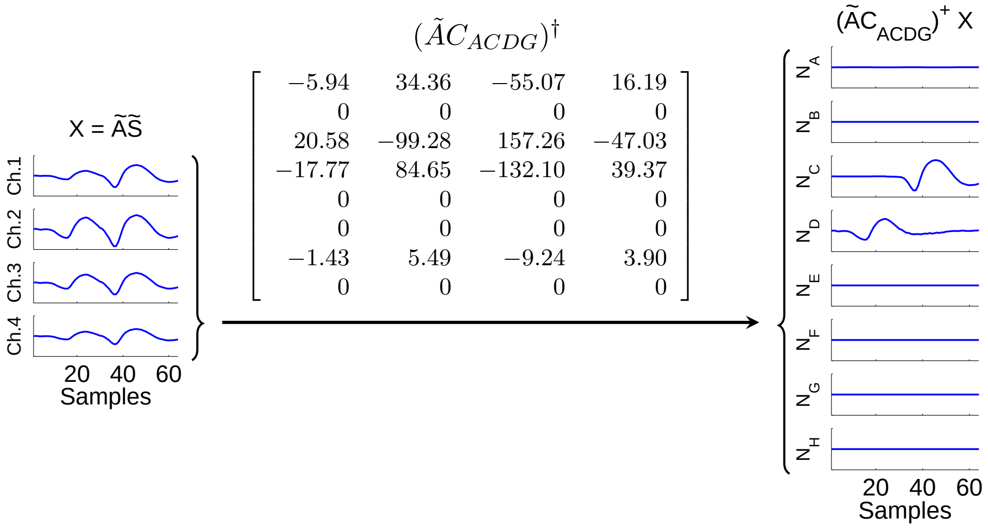
$$\begin{array}{cccccccc} 1 & 0.52 & 0.72 & 0.42 & 0.35 & 0.35 & 0.29 & 0.64 \\ 0.42 & 1 & 1 & 1 & 0.44 & 0.53 & 0.22 & 0.29 \\ 0.23 & 0.39 & 0.69 & 0.69 & 1 & 1 & 0.40 & 0.36 \\ 0.32 & 0.35 & 0.49 & 0.38 & 0.88 & 0.52 & 1 & 1 \\ N_A & N_B & N_C & N_D & N_E & N_F & N_G & N_H \end{array}$$

Figure 3.3: Example of a simulated free-noise recording with the superposition of two neural signals. On the left, the $n = 8$ neural signals $\tilde{S}$ generated by neurons $N_A, \ldots, N_H$ in a time interval where only neurons N_C and N_D are active; these 8 neural signals are mixed by the normalized mixing matrix $\tilde{A}^{4\times 8}$ thus giving rise to the $m = 4$ signals on the right, corresponding to the measurements $X = \tilde{A}\tilde{S}$ of 4 electrodes. Each column of the mixing matrix describes how the the corresponding neural signal is weighted in the observed measurement. The vertical scale is voltage in arbitrary units.



$$\begin{bmatrix} 1.19 & -0.56 & 0.17 & -0.46 \\ -0.30 & 0.78 & -0.55 & 0.20 \\ 0.12 & 0.27 & 0 & -0.13 \\ -0.33 & 0.55 & -0.06 & -0.01 \\ -0.06 & -0.35 & 0.63 & 0.02 \\ 0.15 & -0.43 & 0.93 & -0.41 \\ -0.38 & 0.17 & -0.46 & 0.78 \\ 0.15 & -0.07 & -0.41 & 0.56 \end{bmatrix}$$

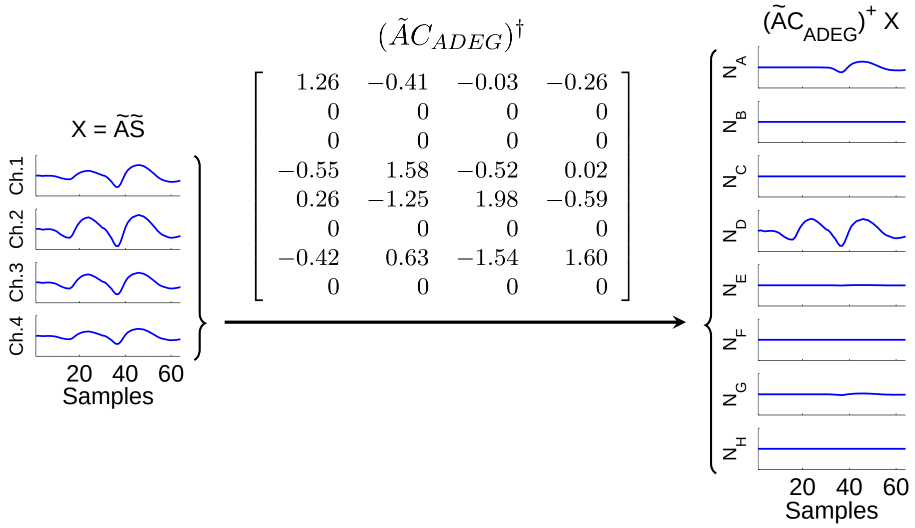
Figure 3.4: A bad attempt to disentangle the two neural signal in the simulated free-noise recording of Figure 3.3, by means of the Moore-Penrose pseudoinverses $\tilde{A}^{\dagger}$. On the left, the $m = 4$ measurements $X = \tilde{A}\tilde{S}$ of the simulated free-noise recording. On the right, the 8 signals obtained applying the pseudoinverse $\tilde{A}^{\dagger}$ to the measurements X. The two original neural signals do not appear reconstructed in the rows of $\tilde{A}^{\dagger}X$, each one is instead spread on more than one row, thus remaining still mixed with the other. The vertical scale is voltage in arbitrary units.



$$(\tilde{A}C_{ACDG})^{\dagger}$$

$$\begin{bmatrix} -5.94 & 34.36 & -55.07 & 16.19 \\ 0 & 0 & 0 & 0 \\ 20.58 & -99.28 & 157.26 & -47.03 \\ -17.77 & 84.65 & -132.10 & 39.37 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1.43 & 5.49 & -9.24 & 3.90 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 3.5: A good attempt to disentangle the two neural signal in the simulated free-noise recording of Figure 3.3, by means of the Moore-Penrose pseudoinverses $(\tilde{A}C_{ACDG})^{\dagger}$, where the matrix C_{ACDG} selects the 4 mixing matrix columns associated to neurons N_A, N_C, N_D and N_G. From left to right: the $m = 4$ measurements $X = \tilde{A}\tilde{S}$ of the simulated free-noise recording; the pseudoinverse $(\tilde{A}C_{ACDG})^{\dagger}$; the 8 signals obtained applying the pseudoinverse $(\tilde{A}C_{ACDG})^{\dagger}$ to the measurements X. In this case, since the columns selected by C include all those associated to effectively active neurons, the two original waveforms have been unmixed in correspondence of traces N_C and N_D, while signals in traces N_A and N_G result null and, the other 4 traces are zero by construction. The vertical scale is voltage in arbitrary units.



$$\begin{bmatrix} 1.26 & -0.41 & -0.03 & -0.26 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.55 & 1.58 & -0.52 & 0.02 \\ 0.26 & -1.25 & 1.98 & -0.59 \\ 0 & 0 & 0 & 0 \\ -0.42 & 0.63 & -1.54 & 1.60 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 3.6: A bad attempt to disentangle the two neural signal in the simulated free-noise recording of Figure 3.3, by means of the Moore-Penrose pseudoinverses $(\tilde{A}C_{ADEG})^{\dagger}$, where the matrix C_{ACDG} selects the 4 mixing matrix columns associated to neurons N_A, N_D, N_E and N_G. From left to right: the $m = 4$ measurements $X = \tilde{A}\tilde{S}$ of the simulated free-noise recording; the pseudoinverse $(\tilde{A}C_{ACDG})^{\dagger}$; the 8 signals obtained applying the pseudoinverse$(\tilde{A}C_{ADEG})^{\dagger}$ to the measurements X. In this case, since the columns selected by C do not include all those associated to active neurons, the two original neural signal are not correctly separated. The vertical scale is voltage in arbitrary units.

row of $(\tilde{A}C)^\dagger X$, and in the spike waveforms they show. This is because, in the context of neural recordings, each source recurrently fires one, or a set of, characteristic waveform(s) that have been estimated in the previous learning phase. Therefore, when the matrix C picks out all the effectively active neurons, and original signals are correctly separated in the rows of $(\tilde{A}C)^\dagger X$, as a consequence, the neural templates can be recognize in correspondence of the respective active neurons, while null signals appear in correspondence of inactive neurons. On the other hand, when, in at least one row of $(\tilde{A}C)^\dagger X$, there is a spike that not correspond to any templates of the associated neuron, this means that one or more active neurons have been excluded by the matrix C.

For example, in Figure 3.5, it is possible to recognize that the original signal reconstruction is done selecting a neuron combination that contains all the two active neurons: in fact, traces N_C and N_D show the characteristic waveforms fired by neurons N_C and N_D respectively; the other two involved traces, N_A and N_G, show a null signal. On the other hand, the original signal reconstruction in 3.6, is surely done selecting a neuron combination that excludes some of the active neurons, since there is not correspondence between obtained and expected waveforms: the spike in trace N_D has not the characteristic shape of neuron N_D, nor the spike in trace N_A has the characteristic shape of neuron N_A.

The idea of MCI4SC algorithm is to individuate the combination of active neurons that better explains measured data in terms of compatibility with the respective single neural waveforms. Obviously, in the analysis of experimental data, the presence of additive noise in the recordings $Y = X + H$, and the possibly ill-conditioning of some matrices $\tilde{A}C$, never allow to exactly find in the rows of $(\tilde{A}C)^\dagger Y$ the expected templates and null signals.

The proposed method, thus tests all the m-combinations of n neurons — by means of the all matrices $C \in \mathcal{C}$ — searching for the one that gives the minimum distance between the actual measurements Y and the simulated measurements $\tilde{A}\hat{S}^C$, where $\hat{S}^C$ is an hypothetical estimation of the original neural signal $\tilde{S}$. Hopefully, the matrix $\hat{C}$ that gives the minimum distance is the one that better describes the measurement channel; $\hat{C}$ shows as well, in the row of $(\tilde{A}\hat{C})^\dagger Y$, which are the active neurons and the arrival times of the respective spikes.

Figure 3.7 shows a segment of experimental data Y and two examples of simulated measurements $\tilde{A}\hat{S}^C$, one for a matrix C that selects all active neurons (top), and the other for a matrix C that does not select all active neurons (bottom). Notice that, in the first case, the original neural signals are correctly separated in the rows of $(\tilde{A}C)^\dagger Y$, and the simulated measurement $\tilde{A}\hat{S}^C$ is more similar to the actual measurement Y than in the second case.

In conclusion, the classification phase consists in examining every segment of data around each detected spikes, with the following procedure:

B1. Computation of the pseudoinverse $(\tilde{A}C)^\dagger$ for each $C \in \mathcal{C}$.

B2. Estimation of the hypothetical signal $\hat{S}^C$ from the rows of $(\tilde{A}C)^\dagger Y$, for each

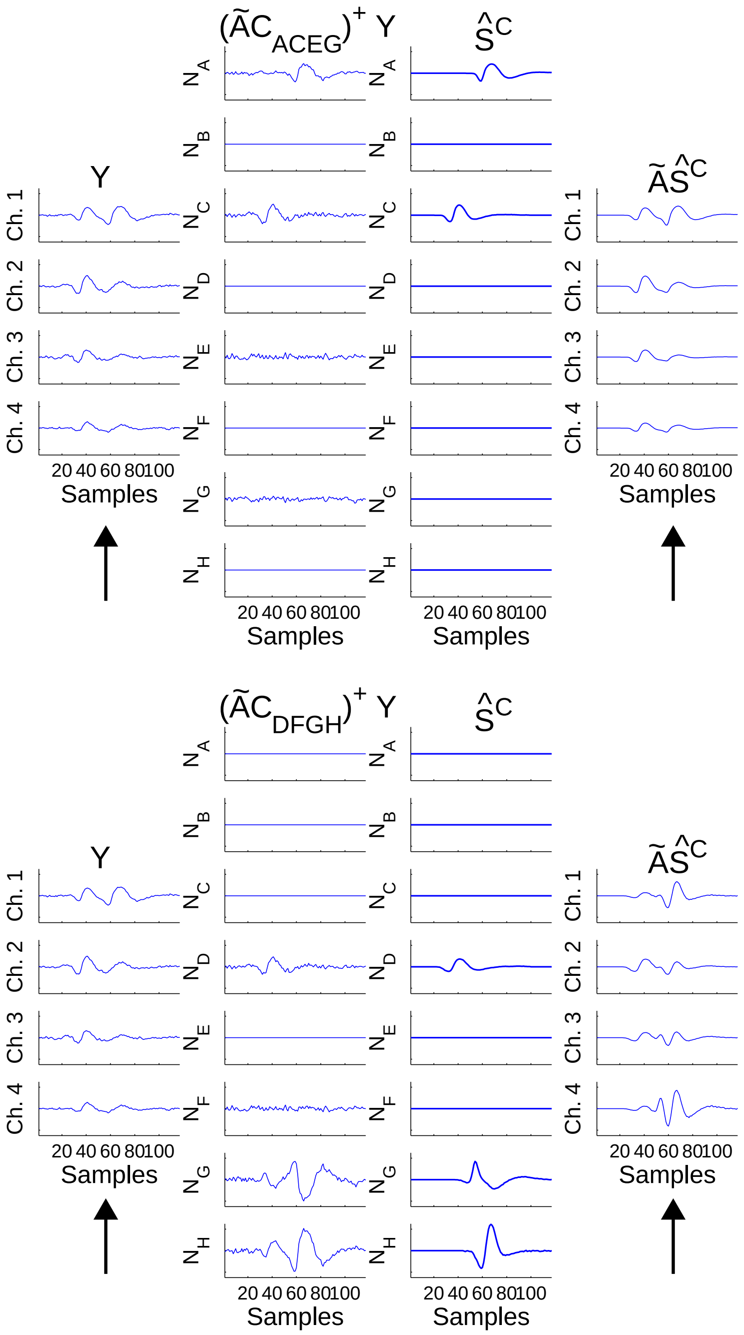


Figure 3.7: A segment of experimental data Y from a tetrode recording in the locust (*Schistocerca americana*) antennal lobe, and two examples of simulated measurements $\tilde{A}\hat{S}^C$. On the top, from left to right: the data segment Y; the unmixed signals $(\tilde{A}C_{ACEG})^{\dagger}Y$, where the matrix C_{ACEG} selects both the effectively active neurons A and C; the corresponding $\hat{S}^C$, i.e. the hypothetical estimation of the original neural signal $\tilde{S}$; the simulated measurement $\tilde{A}\hat{S}^C$, that are qualitatively near to the actual measurement Y. On the bottom, from left to right: the same data segment Y; the unmixed signals $(\tilde{A}C_{BDFH})^{\dagger}Y$, where the matrix C_{BDFH} does not select the two effectively active neurons; the corresponding $\hat{S}^C$, i.e. the hypothetical estimation of the original neural signal $\tilde{S}$; the simulated measurement $\tilde{A}\hat{S}^C$ that are qualitatively far from the actual measurement Y. The vertical scale is voltage in arbitrary units.

$C \in \mathcal{C}$.

B3. Selection of the matrix $\hat{C} \in \mathcal{C}$ that minimizes the distance $||\Sigma^{-1}(Y - \tilde{A}\hat{S}^C)||_F$ between measured data Y and the hypothetical signal $\hat{S}^C$ re-mixed by the recording matrix $\tilde{A}$:

$$\hat{C} = \arg\min_{C \in \mathcal{C}} ||\Sigma^{-1}(Y - \tilde{A}\hat{S}^C)||_F. \tag{3.14}$$

The norm $||\cdot||_F$ is the Frobenius norm, and the m by m matrix Σ^{-1} normalizes different noise levels in the channels.

B4. The neurons whose spikes were detected in the rows of $(\tilde{A}\hat{C})^\dagger Y$, in step B2, are assigned to the segment of data under analysis, together with the respective arrival times, estimated as well in step B2. If no one spike is been detected in the rows of $(\tilde{A}\hat{C})^\dagger Y$, the data segment is not classified, and have to be possibly analyzed with other procedures (it usually contains anomalies or spike superpositions with ill-conditioning problems).

It is interesting to notice that, in the ideal case where there are as many electrodes as recorded neurons, there is an unique inverse matrix $\tilde{A}^{-1}$ that applied to the whole recording allows to solve the inverse problem and to reconstruct the neural traces. On the contrary, the method here illustrated, in order to separate original signals, uses more matrices $(\tilde{A}C)^\dagger$, with C equal to a different $\hat{C}$ from time to time, each one solving the inverse problem in a delimited time interval where, in turn, some neurons are silent.

3.3 Learning Phase

This section illustrates more in details the first of the two phases of the MCI4SC algorithm. The analysis here described can be performed on a segment of the whole recording, on condition that the segment contains a minimum number of single spikes for each involved neuron, thus providing a reliable description of the measurement channel. Whenever some conditions change during the recording (e.g. the electrode positions with respect to recorded neurons) the learning phase has to be applied again on another segment that better describes the data.

A0 - Preliminary steps: filtering, spike detection, and analysis windows — To apply the MCI4SC algorithm, it is necessary to perform some preliminary steps illustrated in this paragraph.

First of all, data must be *filtered* with a band pass filter, typically between $300\,Hz$ and some kHz, in order to delete the slow component of the raw data, and to diminish the noisy appearance of the spike shapes.

It is also necessary to *detect*, in the noisy data, all events to put under analysis. The search for spikes must be performed on each electrode channel, since spikes are larger in some channel than in others. Any detection algorithm can be useful for the purpose, depending on its efficiency. In this thesis, in order to show the MCI4SC algorithm performances, the spike detection has been simply performed with an amplitude threshold on the samples recorded in each channel.

Finally, a suitable *window* must be cut around each detected spike to entirely contain the transient signal (or at least the most part of it). Since the learning phase uses the Wavelet Packet Transformation, it is convenient, at this stage, that the number K of datapoints in the analysis windows is fixed and equal to a power of 2. To choose which particular power of 2 will be used, it is necessary to exploit information about the possible spike duration and remember that this first phase aims to treat single spikes: thus, the window must be long enough to wholly contain one non overlapping spike (or at least the most part of it), but at the same time it must be not so long to contain part of adjacent spikes. On the other hand, the classification phase treats also with overlapping spikes that have highly variable durations — typically larger than single spikes — depending on the number of involved spikes and their respective delays. Therefore, in this second phase it is more appropriate to use analysis windows with different and adaptable lengths. The spike position inside the window analysis is not a crucial point for MCI4SC algorithm, since this parameter does not significantly affect the amplitude ratios estimation. The spike alignment, however, becomes important in waveform templates estimation, when the collection of single spike from the same neuron is averaged. Moreover, a rough alignment is anyway necessary in the learning phase, to guarantee that the most part of the single spike is inside the analysis window.

A1 - Single spikes versus overlapping spikes — As already said in Section 3.2, the components of the mixing matrix $\tilde{A}$ can be estimated as ratios of spike amplitude between different channels, under the condition that spikes are generated by a single neuron; thus, it is fundamental to distinguish between single and overlapping spikes. It is worth noticing that at this step it is not important to understand how many and which neurons are contributing to an overlapping spike, but only if the spike is single or not. Depending on the type of method used in step A3 to sort events, this distinction can be done contextually to the single spike sorting itself. For example, in the common case of feature clustering, the features far from clusters centers are easily overlaps; to obtain a more accurate clustering, however, it is convenient a preliminary distinction that discards all the spikes that surely are overlaps and usually create scattered clouds of points (as shown in figure 3.2).

A first consideration is that all the events with time length larger than the maximum single spike duration can be safely considered as overlapping spikes. Secondly, single and overlapping spikes can also be distinguished looking at the spike waveforms in each channel. According to the assumption 2) in Section

3.1, a single spike is recorded with the same spike waveform by each electrode, while an overlapping spike gives rise, in general, to a different mixture in every sensor, according to different neural signal amplifications in the channels.

To check for the presence of the same waveform in all channel, the most intuitive solution is to evaluate the cross-correlations, as a measure of similarity, among the recorded signals $y_1, \dots, y_m$. The channel that records the largest waveform is considered as the reference channel r, and m cross-correlation are calculated with respect to it. In particular, since the neural signal is recorded simultaneously in each channel according to the assumption 1) in Section 3.1, it suffices to evaluate the cross-correlation with zero-lag:

$$z_{r,i} = \frac{\sum_{k=1}^{K} y_r(k) y_i(k)}{\sqrt{\sum_{k=1}^{K} y_r^2(k) \cdot \sum_{k=1}^{K} y_i^2(k)}} \quad i = 1, \dots, m, \tag{3.15}$$

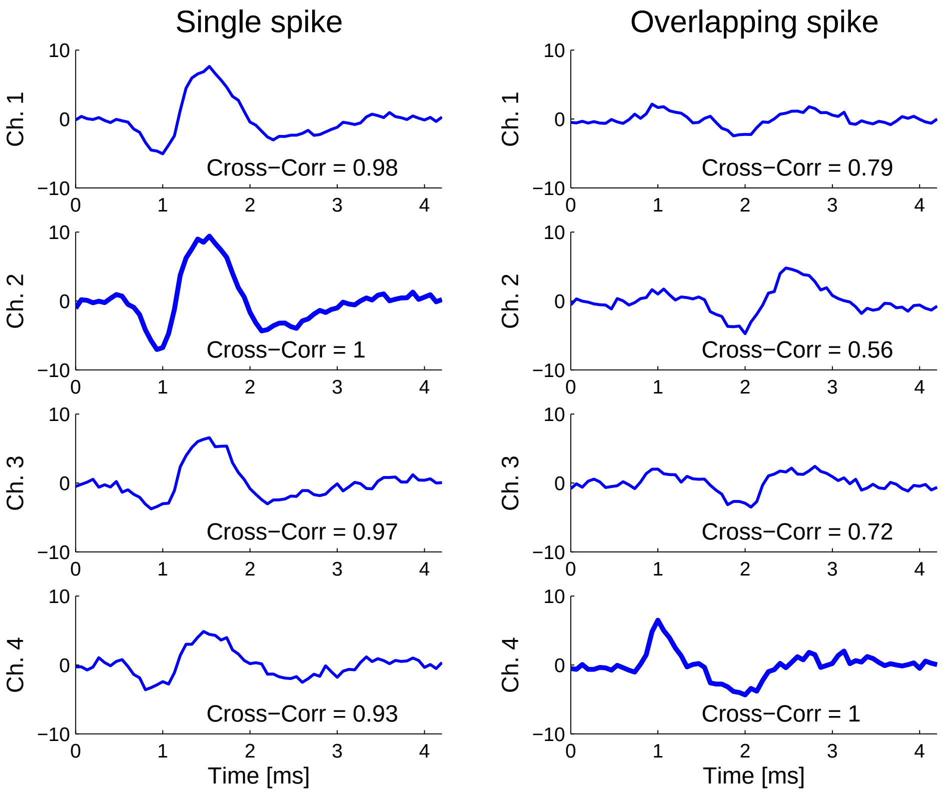
where K is the number of samples in the analysis windows, and $z_{r,r} = 1$ by normalization. Figure 3.8 shows two examples of single and overlapping spikes, with the respective cross-correlation values between the reference channel and the others.

It is worth noticing that, when two identical signals are embedded in noise, the cross-correlation depends on their signal to noise ratios (as the signal to noise ratios decrease, two different waveforms become more similar, and two similar waveform become more different); therefore, it is necessary to set an appropriate threshold $T_{cross}^{r,i}$ that takes into account this parameter. In summary, one spike is considered an overlap when at least one of its cross-correlations $z_{r,i}$, for $i = 1, \dots, m$, $i \neq r$, is under the corresponding threshold $T_{cross}^{r,i}$.

This approach, based on the same/different shape in the channels, does not attempt to exactly classify every spike as single or overlapping; it rather carries out just an approximate discrimination, that may often fail when neural signals are too weak. Nevertheless, this is not a critical issue, since this discriminating procedure can be efficiently completed contextually to the following spike clustering in step A3. Here, at this step, the aim is simply to discard all spikes that are obviously not single spikes.

A2 - Estimation of spike-amplitude ratios, based on Wavelet Coefficients — This paragraph describes a wavelet-based method to well estimate the amplitude ratios of the same impulsive signal recorded by m sensors differently located. In the following, these ratios will be simply named *amplitude ratios.*

In those segments of data that contain single spikes — and thus have, according to the assumption made, the same spike shape $w(k)$ recorded by all sensors — the amplitude ratios of the spike waveforms in different channel are well defined and meaningful parameters. As already said, on the one hand, these amplitude ratios correspond to ratios of column coefficients of the mixing matrix



(a) (b)

Figure 3.8: A single and an overlapping spike from a tetrode recording in the locust (*Schistocerca americana*) antennal lobe, and the respective cross-correlations with zero lag. Thicker lines for the reference channel. The vertical scale is voltage in arbitrary units. In (a) a single spike that shows similar waveforms in each channel: large values of cross-correlation between the signal in the second channel (reference channel) and the signals in other channels. In (b) an overlapping spike, superposition of two, that shows different waveforms in different channels: small values of cross-correlation between the signal in the forth channel (reference channel) and the signals in other channels.

that describes the multi-dimensional channel between neurons and sensors (see Eq. (3.8)). On the other hand, they provide also useful information to cluster single spikes. They are, in fact, strictly related to the neuron position, that is one of the peculiarities used to distinguish the spike source. This process of amplitude ratios estimation can be, therefore, considered as a feature extraction step — typical of many spike sorting algorithms — where a distinctive parameter of the source is extracted, from each spike.

The amplitude ratios as distinguishing features have already been — directly or indirectly — exploited by other spike classification methods that estimate them by considering maximum sample ratios or power ratios. The method proposed here exploits, instead, the Wavelet Transformations, and its capability to concentrate a transient signal in few large coefficients, thus less affected by the white noise. In particular, rather than the Wavelet Transformation, the Wavelet Packet (WP) Transformation has been preferred, since it is more versatile in terms of time-frequency resolution. Concisely, in the method proposed here, the amplitude ratio is estimated by dividing two significant WP coefficients, as will be better detailed in the following paragraphs. For a brief introduction to Wavelet Transformations see Appendix A.

Assuming that each of the m channels records the same spike waveform $w(k)$ with a different amplitude α_i (that is $y_i(k) = \alpha_i w(k) + \eta_i(k)$, $i = 1, \dots, m$), the following procedure estimates m ratios $R_{i/r} = \alpha_i/\alpha_r$ between the spike amplitude in the channel i and the spike amplitude in the channel r, taken as reference among the others.

First of all, the method performs the Discrete WP Transformation, associated to a complete binary tree of depth L, of each recorded spike $y_i(k)$, $k \in [1, \dots, K]$ with K power of 2 (see Figure 3.9). The calculation is initialized by considering the recorded samples $y_i(k)$ as coefficients of the root node $p = 1$ at the decomposition level $l = 0$ (i.e. $d^1_{y_i}(k) = y_i(k)$). The WP coefficients $\{d^{2p}_{y_i}(t)\}_t$ and $\{d^{2p+1}_{y_i}(t)\}_t$, respectively associated to nodes $2p$ and $2p+1$ at decomposition level $l+1$, are efficiently computed from the coefficients $\{d^p_{y_i}(t)\}_t$, of the parent node p at the level $l = \lfloor \log_2 p \rfloor$, through the following recursive

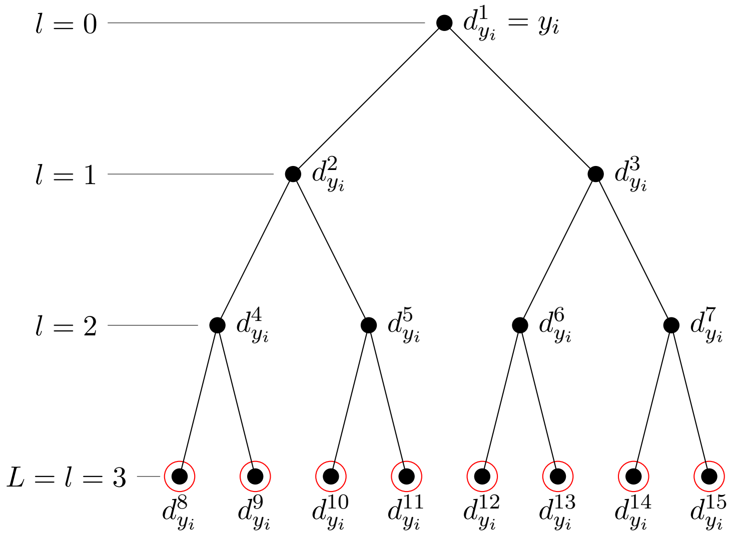


Figure 3.9: A complete binary tree of depth $L = 3$. The root (node $p = 1$) is associated to the K samples of the original signal y_i. In each decomposition level l there are 2^l nodes, each one associated with $K/2^l$ coefficients. The WP coefficients associated just to the open nodes $p = 8, \dots, 15$, are the Wavelet Packet Transformation related to this binary tree.

relations:

$$\begin{aligned} d_{y_i}^{2p}(t) &= \sum_{k \in Z} g_0(2t-k) d_{y_i}^{p}(k), \\ d_{y_i}^{2p+1}(t) &= \sum_{k \in Z} g_1(2t-k) d_{y_i}^{p}(k), \end{aligned} \tag{3.16}$$

where $t = 1, \dots, K/2^{l+1}$, and g_0, g_1 are two quadrature mirror filters related to the chosen mother wavelet.

What is usually named WP Transformation is made by the collection of all coefficients in the open nodes of an admissible binary tree. When the tree is complete, the open nodes are all at the same depth (see circled nodes in Figure 3.9), and the related WP Transformation has a uniform time-frequency resolution. The internal nodes contain redundant information with respect to the open nodes; nevertheless, they decompose the signal with different time-frequency resolutions according to their own level. In order to exploit a larger range of time-frequency resolutions and different signal decompositions, the amplitude ratio estimation takes into account also the WP coefficients associated to internal nodes. It is worth noticing that the computational time does not increase as a consequence of this choice, since the internal nodes coefficients are necessary to calculate the open node coefficients. The symbol $\mathcal{W}_{y_i}^{L}$ precisely indicates the collection of all $(L+1)K$ WP coefficients associated to each open and internal node of the Wavelet tree, from the root (corresponding to the original samples) to the open nodes at the last decomposition level L:

$$\mathcal{W}_{y_i}^{L} = \left\{ d_{y_i}^{p}(t),\, p = 1, \dots, 2^{L+1}-1,\, t = 1, \dots, K/2^{l+1}, \text{ with } l = \lfloor \log_2 p \rfloor \right\}. \tag{3.17}$$

As already said, the WP decomposition concentrates in few large coefficients the whole energy of those signals that, like the neural spikes, are well localized both in time and in frequency (while the white noise remains spread on all the time-frequency plane). Therefore, the WP coefficient in $\mathcal{W}_{y_i}^{L}$ with maximum absolute value typically represents a good percentage of the whole signal y_i, and its position (p, t) indicates that portion of the time-frequency plane where the signal to noise ratio is largest. Among all m recorded channels, the one which has the largest WP coefficients has presumably the largest spike amplitude, and it is taken as the reference channel r, so that:

$$\max(\{|d_{y_r}^{p}(t)| \in \mathcal{W}_{y_r}^{L}\}) \geq \max(\{|d_{y_i}^{p}(t)| \in \mathcal{W}_{y_i}^{L}\}), \quad \forall i = 1, \dots, m. \tag{3.18}$$

Since usually the noise standard deviation is comparable in the m channels, y_r is presumably the recorded spike with the best signal to noise ratio; thus, its maximum WP coefficient in $\mathcal{W}_{y_r}^{L}$ better individuates the time-frequency tile that contains alone the most part of the signal and where the respective coefficients, in other recorded channel, are less affected by hight level of noise. This tile position is indicated by the two values $\overline{p}_r$ and $\overline{t}_r$, that are its node and its coefficient number, respectively:

$$\overline{p}_r, \overline{t}_r = \arg\max_{p,n}(\{|d^p_{y_r}(t)| \in \mathcal{W}^L_{y_r}\}). \tag{3.19}$$

Finally we point out that, thanks to the linearity of the Wavelet Transformations, the amplitude ratios between scaled signals is preserved in the ratio of correspondent coefficients. Therefore, the desired estimation $\hat{R}_{i/r}$ can be simply given by the ratio between the maximum WP coefficient in the y_r decomposition, and the correspondent coefficient in the y_i decomposition:

$$\hat{R}_{i/r} = d^{\overline{p}_r}_{y_i}(\overline{t}_r)/d^{\overline{p}_r}_{y_r}(\overline{t}_r), \quad i = 1, \ldots, m. \tag{3.20}$$

The amplitude ratio estimation $\hat{R}_{r/r}$ is equal to 1 by construction; the other $m-1$ ratios are lower than 1, thus introducing a normalization with respect to the maximum recorded amplitude, that takes place in a different channel from neuron to neuron. This normalization corresponds to normalize the mixing matrix column in (3.9) by means of the maximum coefficient of the column itself, $a_{rj} = \max(\{a_{1j}, \ldots, a_{mj}\})$.

In Figure 3.10, the whole estimation procedure is summarized: on the top the recorded spikes y_1 and y_2, where the first is taken as reference signal; under each spike the respective WP coefficients at the decomposition level 1 up to $L = 4$. The procedure picks out the maximum WP coefficient of the y_1 decomposition, and the corresponding one in the y_2 decomposition. The ratio between these two coefficients gives the amplitude ratio estimation $\hat{R}_{2/1}$.

To give an idea of the benefits introduced by the WP-based method, a comparison with other estimators of amplitude ratios has been performed. The considered methods are:

- The estimator based on **ratios between peak samples** (it exploits the samples with maximum signal to noise ratio in the time domain).

  The reference channel r is the one with the maximum samples in absolute value, so that $\max_k(\{|y_r(k)|\}) \geq \max_k(\{|y_i(k)|\})$, $\forall i = 1, \ldots, m$; $\overline{k}_r = \arg\max_k(\{|y_r(k)|\})$ is the position of the maximum sample in the analysis window, and the samples ratio $y_i(\overline{k}_r)/y_r(\overline{k}_r)$ is the amplitude ratio estimation. For example, the method in [17], exploits the peak positive amplitudes of the recorded spikes, in order to calculate approximate positions of the spike sources around the tetrode; by clustering these spatial locations, the method obtains spike sorting results comparable to those obtained sorting ratios of peak positive amplitudes.

- The estimation based on **ratios between powers** (it involves the whole signal rather than a single sample or a single WP coefficient).

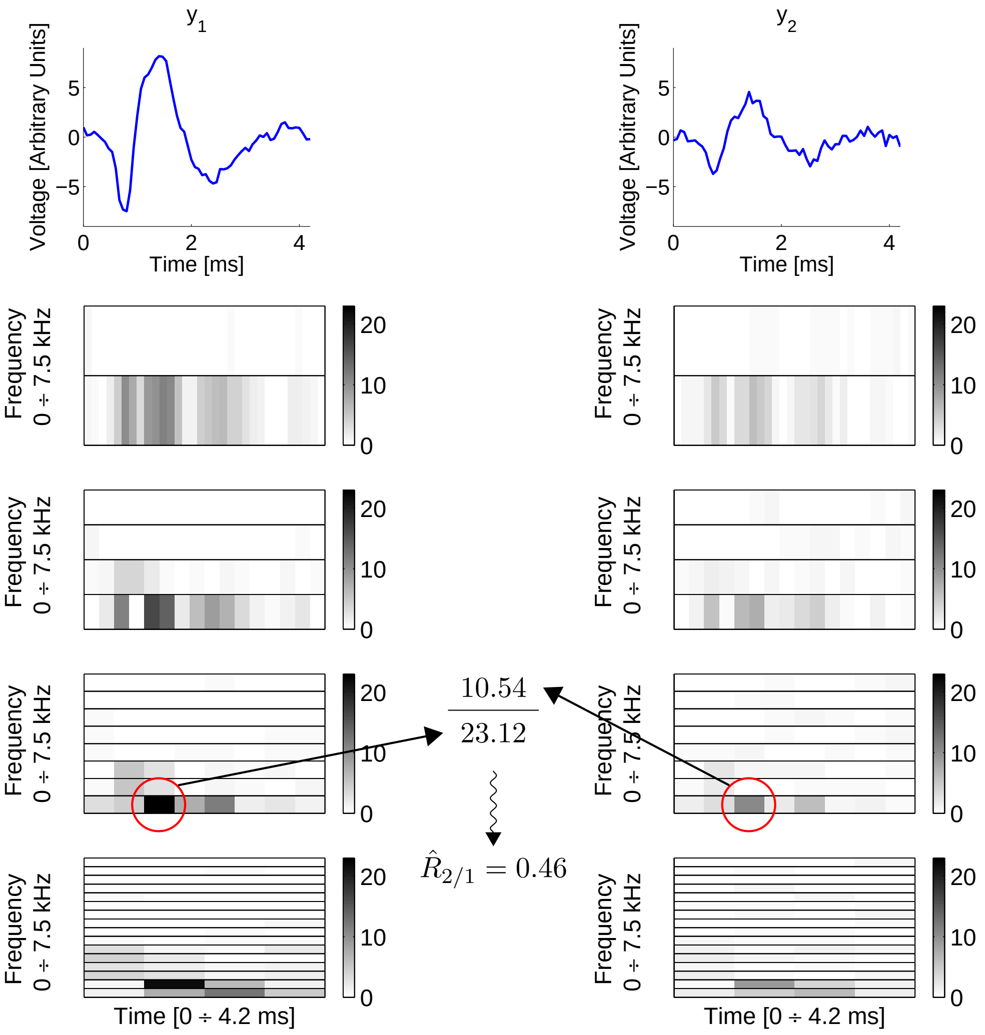


Figure 3.10: Amplitude ratio estimation between two spikes recorded in the locust (*Schistocerca americana*) antennal lobe. On the top, the two recorded spikes y_1 and y_2, each made by 64 samples (the vertical scale is voltage in arbitrary units). The first spike, that is larger, is taken as reference signal. Under each spike the respective WP coefficients, in absolute value, at the decomposition level 1 up to $L = 4$. The WP transformation is performed with the 5th order Symlet as mother wavelet. The procedure pick out the maximum WP coefficient of the y_1 decomposition, and the corresponding one in the y_2 decomposition. The ratio between these two coefficients, both circled in the figure, gives the amplitude ratio estimation $\hat{R}_{1/2}$.

The reference channel r is the one with the maximum power, so that $\sum_k y_r^2(k) \geq \sum_k y_i^2(k)$, $\forall i = 1, \ldots, m$, and the powers ratio is estimated by $\sqrt{\sum_k y_i^2(k) / \sum_k y_r^2(k)}$. For example, in order to estimate the mixing matrix coefficients, the method in [19] averages in time the absolute values of the samples in each channel and normalized them; this is comparable to an estimation based on powers ratios. Also the approach in [42], in order to estimate the 3 spatial coordinates of source position, exploits a weighted average of the amplitudes near the spike peak, that is proportional to spike power.

To compare the 3 methods (respectively based on maximum WP coefficient, maximum sample, and spike power) a spike template $w(k)$, estimated from a neural recording, has been used. In the simulation two channels have been considered: in the first channel the multiplicative factor of the spike template has been fixed to 1, and in the second channel the multiplicative factor has been varied between 0 and 1 ($0 \leq \alpha_2 \leq 1 = \alpha_1$). Gaussian noise with zero mean and the same standard deviation $\sigma_{\eta_1} = \sigma_{\eta_2} = \sigma_\eta$, has been added to both channels. Figure 3.11 shows an example of two simulated measurements $y_1 = \alpha_1 w + \eta_1$ and $y_2 = \alpha_2 w + \eta_2$, with amplitude ratio $\alpha_2/\alpha_1 = 0.5$ and $\sigma_\eta = 0.7$, that represents a situation easily found in neural recordings. The amplitude ratio has been estimated 10000 times, with the 3 estimators, for each considered value of the ratio α_2/α_1 and for each considered value of the standard deviation σ_η.

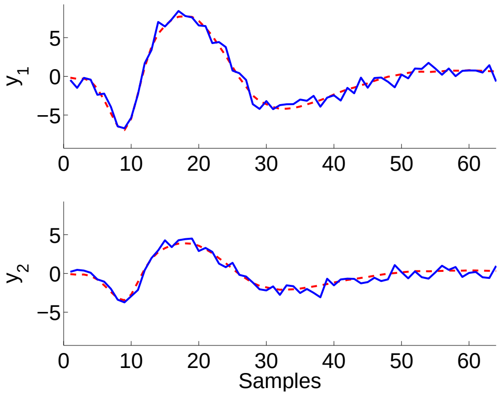


Figure 3.11: Two simulated measurements $y_1 = \alpha_1 w + \eta_1$ and $y_2 = \alpha_2 w + \eta_2$ respectively on the top and on the bottom (solid lines). The vertical scale is voltage in arbitrary units. The dashed lines refer to the waveform $w(k)$ estimated from a neural recording in the the locust (*Schistocerca americana*) antennal lobe. Taking the first channel as reference, the amplitude ratio α_2/α_1 is equal to 0.5. The noise in both channels is Gaussian with mean zero and standard deviation σ_η equal to 0.7.

It is worth noticing that the comparison has been performed to show the differences among the techniques on a qualitative basis only, using a specific example. Results may change when varying the template shape w or the analysis window length. The WP-based estimator performance depends in particular on the percentage of spike power captured by the maximum WP coefficient (for example, at the level of decomposition 4 using the 5th order Symlet as mother wavelet, the maximum WP coefficient of the template waveform w in Figure 3.11 captures the 39% of the whole power). The performance of the sample-based estimator depend as well on the percentage of spike amplitude captured by the maximum sample, and this percentage changes with the spike shift or the sampling period. The performance of the power-based estimator depends instead on the percentage of signal power in the analysis window with respect to the noise power, and this percentage changes with the length of analysis window.

Figure 3.12 shows the means of the 3 estimators, plus/minus 3 times their standard deviations, with the noise standard deviation σ_η fixed to 0.7, and with the amplitude ratio α_2/α_1 varying from 0 to 1. The WP-based estimator is compared with the sample-based estimator in (a), and with the power-based estimator in (b). Notice that the mean of the WP-based estimator is superimposed to the first quadrant bisector that represents the ideal estimator. On the contrary, the sample-based estimator is not consistent: the distance between its mean and the true value increases with the ratio α_1/α_2, due to the presence of noise. Moreover, the dispersion of the sample-based estimator is greater than the others. The power-based estimator has the advantage of a lower dispersion, due to the fact that it considers the signal in its entirety, and not in a single time instant or in a delimited time-frequency tile. But, even the power-based estimator, is not consistent: in particular, for low value of the ratio α_2/α_1, it is limited by the noise power.

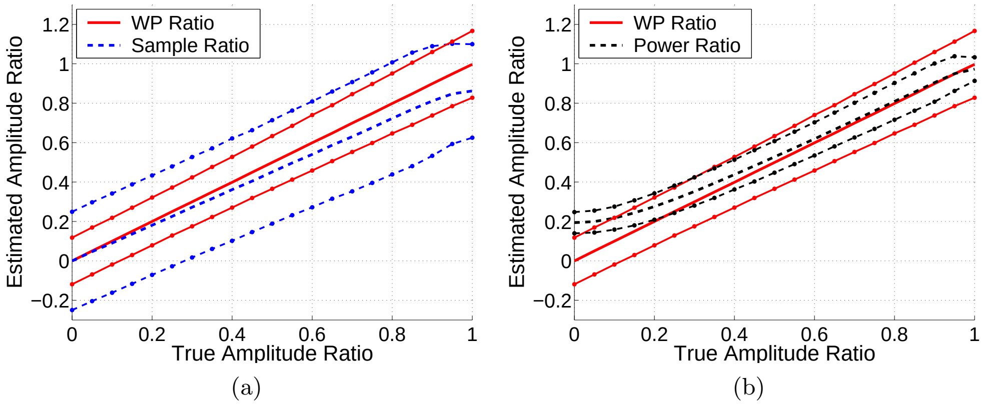
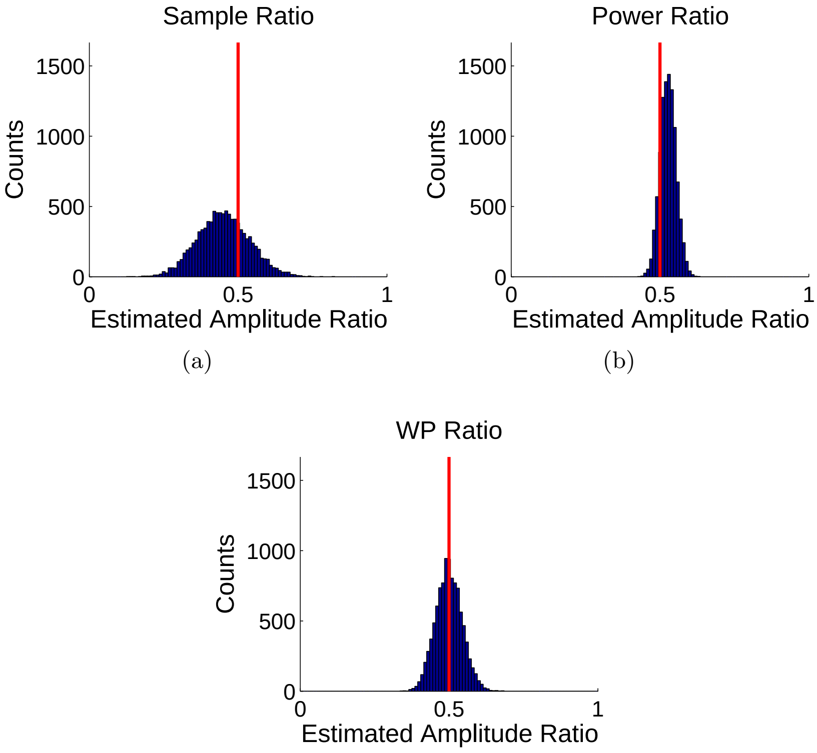


Figure 3.12: Means of the 3 estimators, plus/minus 3 times their standard deviations, with the noise standard deviation σ_η fixed to 0.7, and varying the amplitude ratio α_2/α_1 from 0 to 1. The WP-based estimator (solid lines) is compared with the sample-based estimator (dashed lines) in (a), and with the power-based estimator (dashed lines) in (b). The WP transformation is performed with the 5th order Symlet as mother wavelet, and decomposition level $L = 4$. For each considered value of the ratio α_2/α_1, 10000 trials have been done.

When amplitude ratios are clustered as distinctive features of the spike source, a low estimator dispersion is important, since it gives rise to more concentrated, and thus separated, clusters. On the other hand, consistency is even more important, since it allows to better estimate the mixing matrix $\tilde{A}$, on which depends the whole spike sorting procedure and its results.

Figure 3.13 shows the empirical distributions of the 3 estimators when amplitude ratio α_2/α_1 is equal to 0.5, and noise standard deviation σ_η is equal to 0.7. Also this figure highlights that the sample-based estimator, in (a), and the power-based estimator, in (b), are not consistent, while the WP-based estimator, in (c), has a distribution practically centered on the true value. At the same time, the figures show the large dispersion of the sampled-based estimator, the low dispersion of the power based-estimator, and the dispersion of the WP-based estimator in-between.



(a)

(b)

(c)

Figure 3.13: Empirical distributions of the 3 estimators when amplitude ratio α_2/α_1 is equal to 0.5, and noise standard deviation σ_η is equal to 0.7. In (a), the empirical distribution of the sampled-based estimator, in (b) the empirical distribution of the power-based estimator, and in (c) the empirical distribution of the WP-based estimator.

Figure 3.14 shows instead the means of the 3 estimators, plus/minus 3 times their standard deviations, when the amplitude ratio α_2/α_1 is fixed to 0.5, and the noise standard deviation σ_η, equal in both channels, varies from 0.1 to 2. In

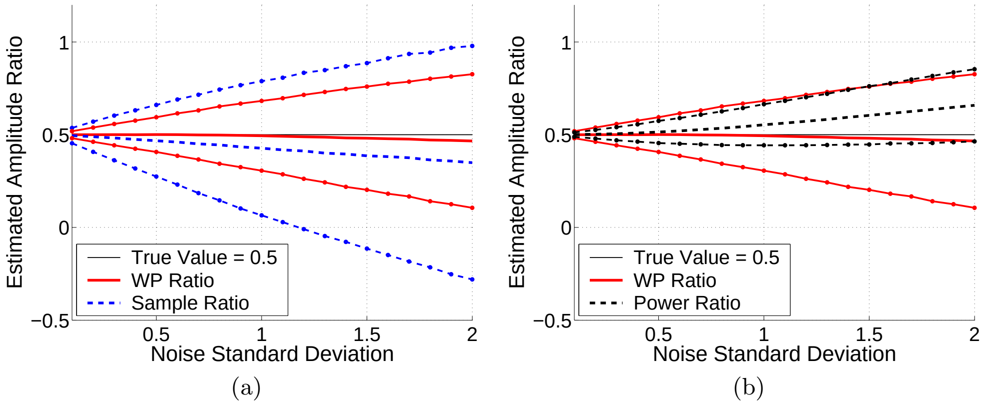


Figure 3.14: Means of the 3 estimators, plus/minus 3 times their standard deviations, with the amplitude ratio α_2/α_1 fixed to 0.5, and the noise standard deviation σ_η varying from 0.1 to 2. The WP-based estimator (solid lines) is compared with the sample-based estimator (dashed lines) in (a), and with the power-based estimator (dashed lines) in (b). The WP transformation is performed with the 5th order Symlet as mother wavelet, and decomposition level $L = 4$. For each considered value of the noise standard deviation σ_η, 10000 trials have been done.

this figure, the same comments about dispersion are valid. As for consistency, even the WP-based estimator moves away from the true value, but less than the other two estimators.

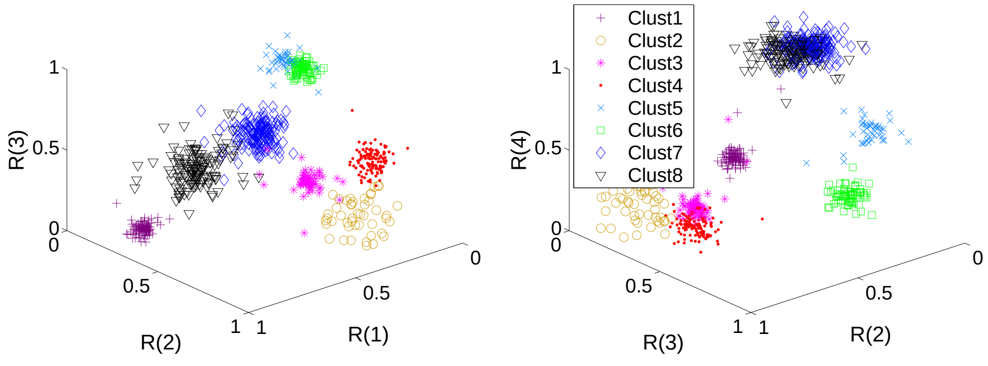
In conclusion, the comparison shows that the dispersion of the WP-based estimator is a compromise between the other two dispersions, and, above all, the WP-based estimator has the unquestionable advantage to give, in mean, an estimation closer to the true value. This happens thanks to the Wavelet ability to exalt the signal with respect to the noise, thus allowing a good estimation even when the signal to noise ratio is very low.

Clearly, the amplitude ratio estimation can likewise be applied to overlapping spikes, even though, in this case, the obtained values loose the meaning of amplitude ratio between two signals.

A3 - Single spike Sorting — In the previous step A2, a vector $R^e = [\hat{R}^e_{1/r}, \ldots, \hat{R}^e_{m/r}]'$ of m components normalized to 1, is been derived from each event e classified as non-overlapping spike. Clustering these vectors R^e in n cluster $Clust_1, \ldots, Clust_n$, i.e. grouping those vectors that are similar, and associating one neuron to each cluster, means as well to relate every single spike $e \in Clust_j$ to the neuron j that has generated it.

Many clustering algorithms, more or less complex, are available to this aim. Any of them can be used in this step, according to its efficiency, robustness, or independence of human supervision. In this thesis, in order to show performances of MCI4SC algorithm, the clustering step has been performed using the simple K-means algorithm. In this work, the clustering algorithm is opportunely initialized by a human operator that provides the number of clusters; as an alternative, more sophisticated algorithms may use a cost function to set this number.

To give an example of clustering, Figure 3.15 shows the results obtained by applying the K-means algorithm to all points in Figure 3.2b. Eight clusters, corresponding to as many neurons, are well separated by K-means, thus displaying that amplitude ratios can be good features to cluster and sort single spikes.



(a) (b)

Figure 3.15: Plots of amplitude ratios clusters that group each spike previously classified as single spike. In (a), a tri-dimensional plot of the first 3 amplitude ratios. In (b), a tri-dimensional plot of the last 3 amplitude ratios. 745 single spikes from a tetrode recording in the locust (*Schistocerca americana*) antennal lobe are grouped in 8 clusters.

A human supervisor can test the accuracy of the clustering procedure by looking at superimposed spikes waveforms, cluster by cluster, and checking the homogeneity of each cluster.

In the case where amplitude ratios, rather than other features, are grouped, the main problem is the presence of clusters made of spikes with different waveforms. In that instance, it is necessary to distinguish if they are single neurons that fire different waveforms (e.g. bursting neurons), or if they correspond to more neurons in a similar position with respect to the electrodes. Therefore, the ambiguity of such clusters that contain different waveforms, can generate two kind of errors: i) a single neuron that fires different waveforms can be considered as two or more neurons; ii) two, or more, neurons, with different waveforms and similar amplitude ratios can be considered as just one neuron.

To solve this kind of ambiguity, human intervention is particularly effective. A supervisor can evaluate the kind of electrode configuration (linear, planar, or not planar), if the number of electrodes that measure the spikes is lower than four, and the presence of burst with changing waveform in the data. From this kind of considerations, he can establish the nature of ambiguous clusters, and separate the various representative templates.

Obviously, in the unlikely case where two neurons fire a very similar waveform and are almost at the same relative distance from electrodes, they form an unique homogeneous cluster, and they result indistinguishable from the MCI4SC point of view. Such a situation can only be handled exploiting the information related to the Inter-Spike Interval distribution.

The case of recording in the locust (*Schistocerca americana*) antennal lobe, whose cluster are shown in Figure 3.15, has no ambiguous cluster: each group of spike mainly contains only one waveform (see Figure 3.16), and it is, therefore, unequivocally associated to a single neuron.

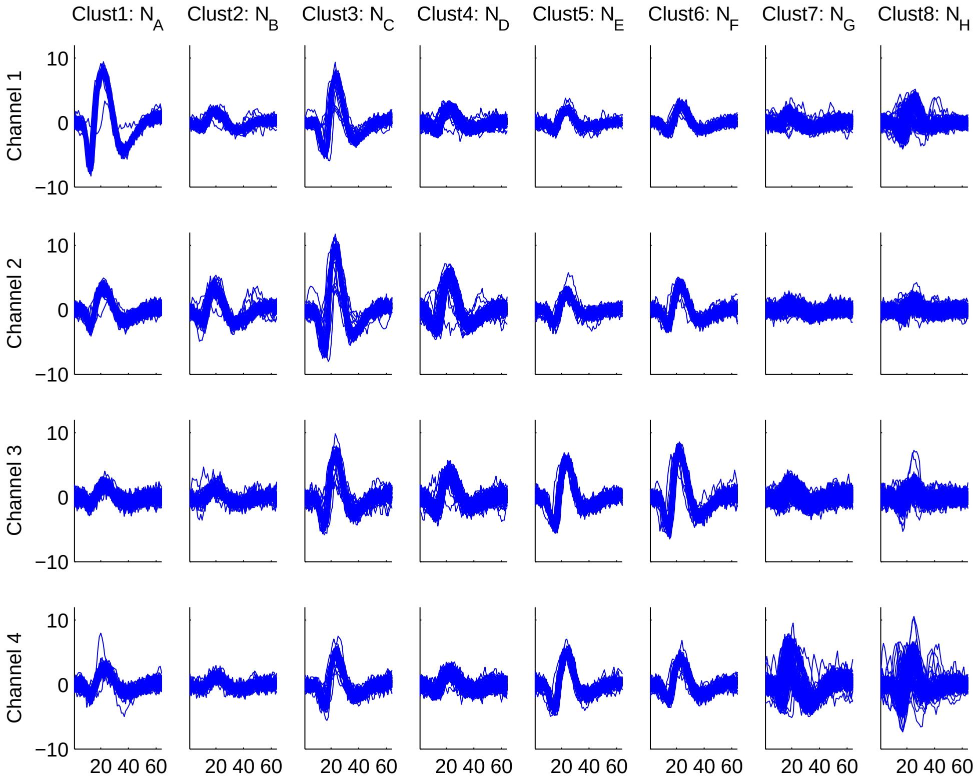


Figure 3.16: Superimposed spike waveforms, as they are clustered in Figure 3.15: each cluster mainly exhibits a unique waveform. Spikes are from a tetrode recording in the locust (*Schistocerca americana*) antennal lobe. The vertical scale is voltage in arbitrary units.

A4 - Waveforms and mixing matrix estimation — From each cluster $Clust_j$, $j = 1, \ldots, n$, identified in the previous step A3, it is possible to estimate the respective column $\hat{A}_j$ of the mixing matrix $\tilde{A}$, and the respective spike waveform template $\hat{w}_j(k)$, $k \in [1, \ldots, K]$, both characterizing the associated neuron. The two estimates are simply obtained by averaging the amplitude ratio vectors and the spike waveforms, among all single spikes in each cluster:

$$\hat{A}_j = \sum_{e \in Clust_j} \frac{1}{N_j} R^e; \tag{3.21}$$

$$\hat{w}_j(k) = \sum_{e \in Clust_j} \frac{1}{N_j} y^e_{r_j}(k), \quad k \in [1, \ldots, K], \tag{3.22}$$

where N_j is the number of events e collected in the cluster $Clust_j$. Notice that, since each cluster collects vectors with similar amplitude ratio, all spikes in the same cluster typically have the same reference channel r_j (different from neuron to neuron). The reference channel is the one where the spike is recorded with the largest amplitude and the largest signal to noise ratio, thus granting a better estimation of the mean spike waveform $\hat{w}_j(k)$. Notice also that template estimation is more accurate, when spikes are realigned before averaging.

Inevitably, into each cluster, there are also overlapping spikes erroneously identified as single spikes. Since it is crucial, for the subsequent spike classification, to well estimate the mixing matrix and the spike templates, it is convenient to exclude, from the average, the spikes having waveform too different from the mean one, or amplitude ratios too distant from the cluster centroid (i.e. all those spikes that probably are overlapping spikes). In fact, the learning phase does not attempt to classify the greatest number of spikes, it just aims to collect clean representatives of every spike type. Therefore better estimates can be achieved by excluding those spikes that are very likely overlapping spikes.

Even in this case, a human supervisor can test the quality of the outliers removal by looking at superimposed spikes.

Figure 3.17 shows the cluster of Figure 3.15, once they have been cleaned.

Figure 3.18 shows the superimposed spike waveform for each cleaned cluster, the mixing matrix components as they have been estimated, and, on the bottom, the corresponding estimated waveforms. The set of the estimated columns, arbitrarily ordered and rounded to nearest hundredths, gives rise to the normalized mixing matrix estimation:

$$\hat{A} = \begin{bmatrix} 1 & 0.52 & 0.72 & 0.42 & 0.35 & 0.35 & 0.29 & 0.64 \\ 0.42 & 1 & 1 & 1 & 0.44 & 0.53 & 0.22 & 0.29 \\ 0.23 & 0.39 & 0.69 & 0.69 & 1 & 1 & 0.40 & 0.36 \\ 0.32 & 0.35 & 0.49 & 0.38 & 0.88 & 0.52 & 1 & 1 \end{bmatrix}. \tag{3.23}$$

In the case of an ambiguous cluster, where more than a spike waveform is recognizable in the same group of spikes, it is necessary to estimate all the Q templates $\hat{w}^q_j(k)$, $q = 1 \ldots, Q$, whose shapes are visible in the cluster $Clust_j$. All these templates are associated to the same mixing matrix column $\hat{A}_j$, either when the cluster represents one single neuron that fires different waveforms,

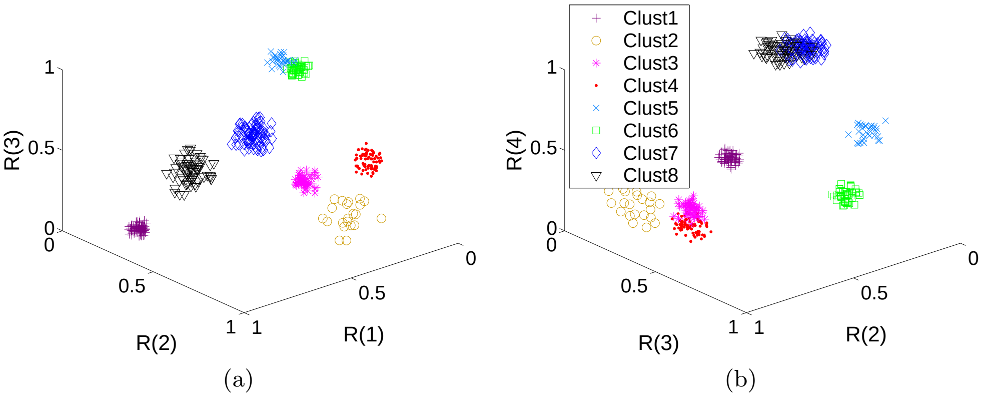


Figure 3.17: Plots of the amplitude ratios clusters of Figure 3.15, once they have been cleaned. In (a), a tri-dimensional plot of the first 3 amplitude ratios. In (b), a tri-dimensional plot of the last 3 amplitude ratios. 448 spikes in total are represented.

or when it represents more distinct neurons at the same relative distance from electrodes. In practice, it is necessary to perform a further clustering step on a smaller set of events that belong to the ambiguous cluster. This further classifying step, based on spike waveform rather than on spike-amplitude ratios, can be done more or less automatically, through any of the spike sorting algorithm already developed or through human intervention.

3.4 Classification Phase

This section illustrates in details the second of the two phases of the MCI4SC algorithm. In the classification phase all spikes are analyzed, one by one, trying to resolve the possible superposition.

B1 - Pseudoinversion of the mixing matrices $\hat{A}C$, for each $C \in \mathcal{C}$ — As already said in Section 3.2, when the number of sensors m is smaller than the number of neurons n, the original neural signals that contribute to a recorded data segment $Y = \tilde{A}\tilde{S} + H$, can be estimated through the Moore-Penrose pseudoinverse of a particular mixing matrix $\tilde{A}C$. The estimation is correct on condition that the mixing matrix $\tilde{A}C$, that describes a measurement channel with a number of neurons equal to the number of sensors, includes all active neurons, that are no more than m. Since it is not known a priori which neurons are effectively active, the idea of MCI4SC algorithm is to test all the matrices $C \in \mathcal{C}$ (see Eq. (3.13)), representing all the m-combinations of n neurons, in order to determine which one explains the measured data better than the others. The number of combinations to examine in $\mathcal{C}$, where m neurons

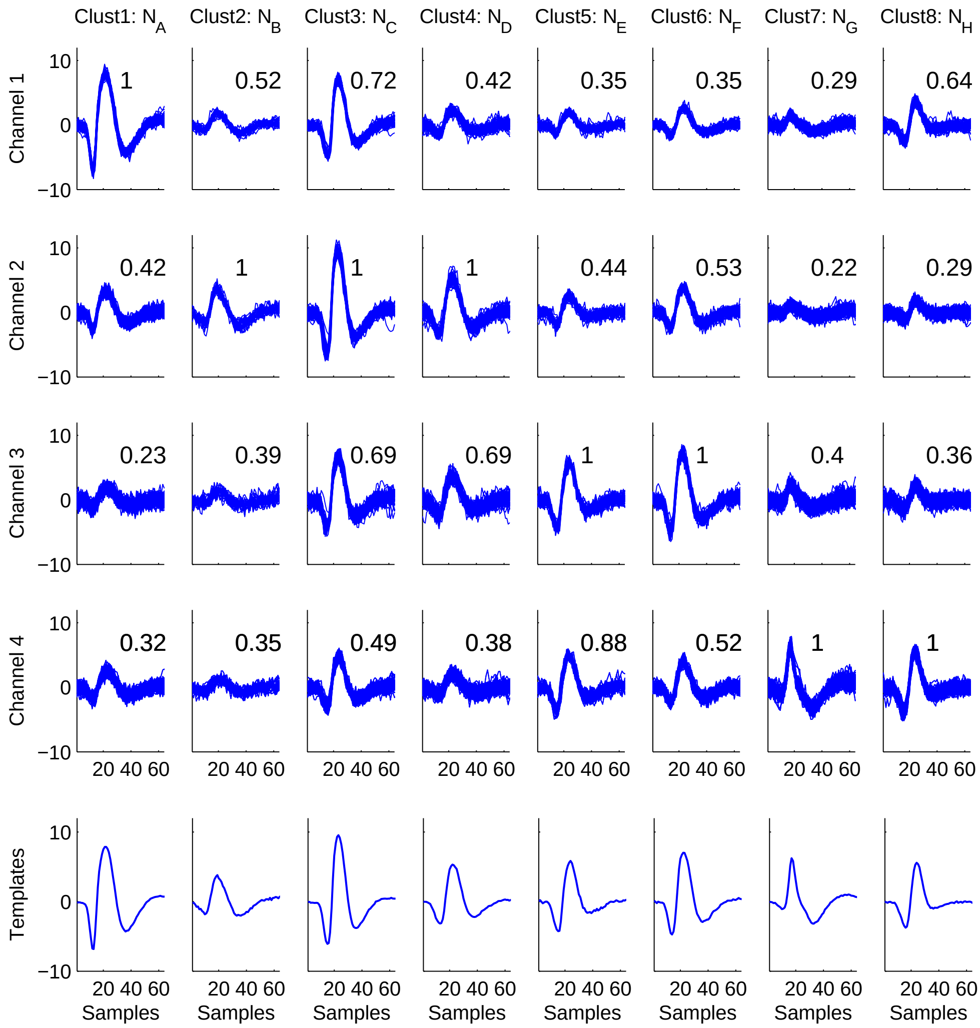


Figure 3.18: Superimposed spike waveforms for each cleaned cluster in Figure 3.17, and the mixing matrix components as they have been estimated. On the bottom, the corresponding templates, estimated from the respective reference channel. Spikes are from a tetrode recording in the locust (*Schistocerca americana*) antennal lobe. The vertical scale is voltage in arbitrary units.

are possibly active, and the other $m - n$ are silent, is:

$$\binom{n}{m} = \frac{n!}{(n-m)!m!}. \tag{3.24}$$

Let the mixing matrix $\hat{A}$ be estimated as in A4. In this first step, the MCI4SC algorithm calculates the Moore-Penrose Pseudoinverse $(\hat{A}C)^\dagger$ for each $C \in \mathcal{C}$, and applies it to the data segment Y under analysis, thus obtaining a neural signal reconstruction for each neuron combination in $\mathcal{C}$:

$$V^C = [v^C(1), \ldots, v^C(K)] = (\hat{A}C)^{\dagger} Y, \tag{3.25}$$

where V^C is a n by K matrix. In an equivalent form, it can be written as:

$$v^C(k) = [v_1^C(k), \ldots, v_n^C(k)]' = (\hat{A}C)^{\dagger} y(k), \quad k \in [1, \ldots, K], \tag{3.26}$$

where each row $v_j^C(k)$, $j = 1, \ldots, n$ of the matrix V^C is the neural signal reconstruction corresponding to neuron j.

As an example, consider the tetrode recording from the locust (*Schistocerca americana*) antennal lobe where 8 neurons have already been individuated (see Figure 3.18); in this case, the number of neuron combinations to examine is $\binom{8}{4} = 70$. Figure 3.19 shows a data segment of this recording, where two spikes from neurons N_A and N_F are slightly superimposed. Figure 3.20 shows the application of the pseudoinverse $(\hat{A}C)^{\dagger}$ to the measured data segment of Figure 3.19, for three of the 70 possible 4-combinations of different neurons. The crucial point is whether the neurons selected by C include, or not, all effectively active neurons. For example, the combination considered in Figure 3.20(top) includes both neuron N_A and neuron N_F: as a consequence, the respective traces show the spike waveforms correspondent to those of the two neurons. On the contrary, the other two combinations in Figure 3.20(bottom) exclude one or both the active neurons: in those case at least one trace exhibits a spike waveform that does not match to the respective neural template in Figure 3.18(bottom).

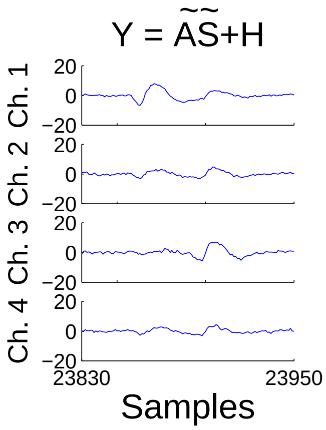


Figure 3.19: A data segment recorded by a tetrode in the locust (*Schistocerca americana*) antennal lobe. Two spikes from neurons N_A and N_F are slightly superimposed. The vertical scale is voltage in arbitrary units.

Notice that, when the number μ of active neurons is not equal to m ($\mu < m$), the m-combinations that contain all μ active neurons are more than one: they

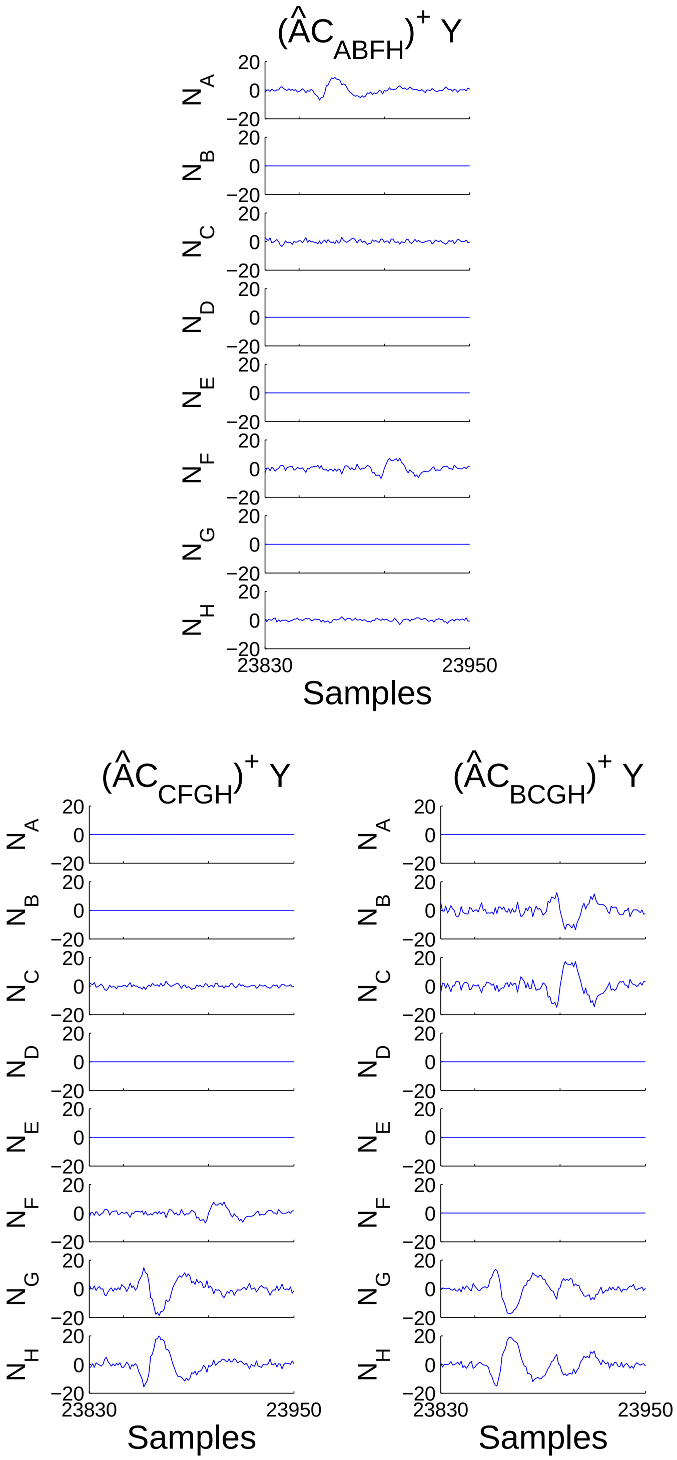


Figure 3.20: Application of the pseudoinverse $(\hat{A}C)^\dagger$ to the measured data segment Y of Figure 3.19, for three of the 70 possible 4-combinations of different neurons. Top: the two active neurons N_A and N_F are both included in the considered combination; bottom: one or both the active neurons are excluded by the considered combination. The vertical scale is voltage in arbitrary units.

are exactly $\binom{n-\mu}{m-\mu} = \frac{(n-\mu)!}{(n-m)!(m-\mu)!}$. For example, in the data segment of Figure 3.19, only $\mu = 2$ neurons are active, and thus there are $\binom{6}{2} = 15$ 4-combinations that include both neuron N_A and neuron N_F. Nevertheless, all these correct combinations are not equivalent from the numerical point of view, since they can be more or less ill-conditioned.

When a matrix is ill-conditioned, small changes in its components or in the recording (due to round-off errors, estimation error, or measurement errors) give rise to large perturbations in the solution of the inverse problem. A measure of the solution sensitivity to errors, is the condition number of the mixing matrix, that, using the 2-norm, is defined as the ratio of the largest singular value to the smallest: values near 1 indicate a well-conditioned matrix, and values much greater than 1 indicate an ill-conditioned matrix.

Figure 3.21 shows the result of considering a neuron combination, that like the one in Figure 3.20(top), includes both the two active neurons N_A and N_F, but with a higher condition number of the mixing matrix $\hat{A}C$. Notice that in this case where the condition number is higher, the two original neural templates are not recognizable in the row of V^C.

According to which neurons the matrix C selects or not, some columns rather than others contribute to the mixing matrix $\hat{A}C$, and the condition number changes as well. Therefore, some overlap can be better resolved than other, depending on the set of neurons that generated them. In particular, the more similar (or nearly linear dependent) are the columns associated with the involved neurons, the higher is the condition number, and the more difficult is to recover the original signals. This reflects the difficulty to separate signals coming from neurons with similar relative distances from sensors. Once again, it is worth underlining the importance of electrode configuration: to guarantee low condition numbers that better solve the inverse problem, electrodes must be as evenly separated as possible in all three dimensions.

One more comment about the condition number concerns the mixing matrix normalization. Once that the relative distances between neurons and electrodes are given, i.e. amplitude ratios are given, the condition number depends also from the mixing matrix normalization that has been used. Calculating the amplitude ratios with respect to the largest spike, as described in step A2, the MCI4SC algorithm normalizes each column so that its maximum value is 1. From the theoretical point of view, this column normalization correspond to rescale the neural signal at the source, and no effects are introduced in the signal reconstruction: dividing a column for a certain factor, the correspondent row in the pseudoinverse is multiplied by the same factor and the correspondent estimated neural signal is accordingly scaled. From the numerical point of view, however, the neural signal is not exactly scaled, and the perturbations vary with the effective value of matrix components. In particular, when the condition number is large, the matrix inversion can introduce large errors. It is worth noticing that the used normalization leading to mixing matrix compo-

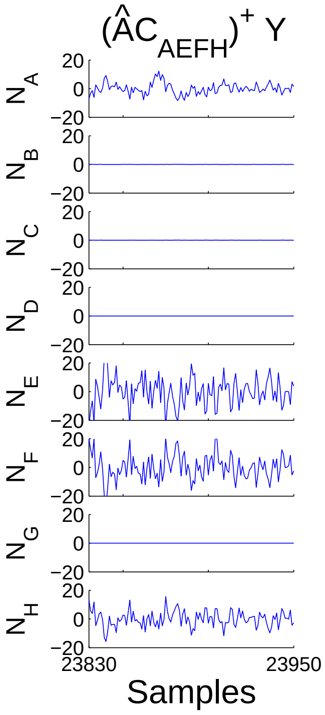


Figure 3.21: Application of the pseudoinverse $(\hat{A}C_{AEFH})^{\dagger}$ to the measured data segment Y of Figure 3.19. The neuron combinations includes both the two active neurons N_A and N_F, but the condition number of the mixing matrix $\hat{A}C_{AEFH}$ is higher than the one of $\hat{A}C_{ABFH}$ considered in the top of Figure 3.20: the values are $\sim$ 81.5 and $\sim$ 6.0 in the two respective cases. The vertical scale is voltage in arbitrary units.

nents between 0 and 1, keeps the condition number in a small range around its minimum.

B2 - Estimation of the hypothetical signals $\hat{S}^C$, for each $C \in \mathcal{C}$ — The MCI4SC algorithm works assuming that each combination of active neurons, represented by $C \in \mathcal{C}$, correctly describes the segment of measured data under analysis. Therefore, it expects to find, in the m traces $v_j^C(k)$ associated to the selected neurons, either noise or the spike waveform template of neuron j (the other $n - m$ traces are zero by construction).

In this step B2, the MCI4SC algorithm determines, for each $C \in \mathcal{C}$, and for each presumable active neuron j, the time shift k_j^C and the amplification B_j^C of the neural spikes in the corresponding j-th row of the matrix V^C. To evaluate both these two parameters, the algorithm calculates the cross-correlation between $v_j^C(k)$ and the correspondent template $\hat{w}_j(k)$, for each neuron j selected by the matrix C:

$$z_{v,w}^{j}(k) = \begin{cases} \sum_{f=k+1}^{K} v_j^C(f-k)\hat{w}_j(f), & 0 \le k \le K-1 \\ \\ \sum_{f=1}^{K-k} v_j^C(f-k)\hat{w}_j(f), & -K+1 \le k < 0 \end{cases}, \tag{3.27}$$

where the templates $\hat{w}_j(k)$ are those estimated in step A4. The time shift and the spike amplification are estimated as follows:

$$k_j^C = \arg\max_k(\{z_{v,w}^j(k), k \in [-K+1, \dots, K-1]\}), \tag{3.28}$$

and

$$B_j^C = \frac{z_{v,w}^j(k_j^C)}{\sum_k \hat{w}_j^2(k)}. \tag{3.29}$$

If the amplification B_j^C is under a certain threshold T_{B_j}, the algorithm assumes that only noise is present in the j-th row of V^C and the corresponding neuron is inactive, like those that are not selected by matrix C.

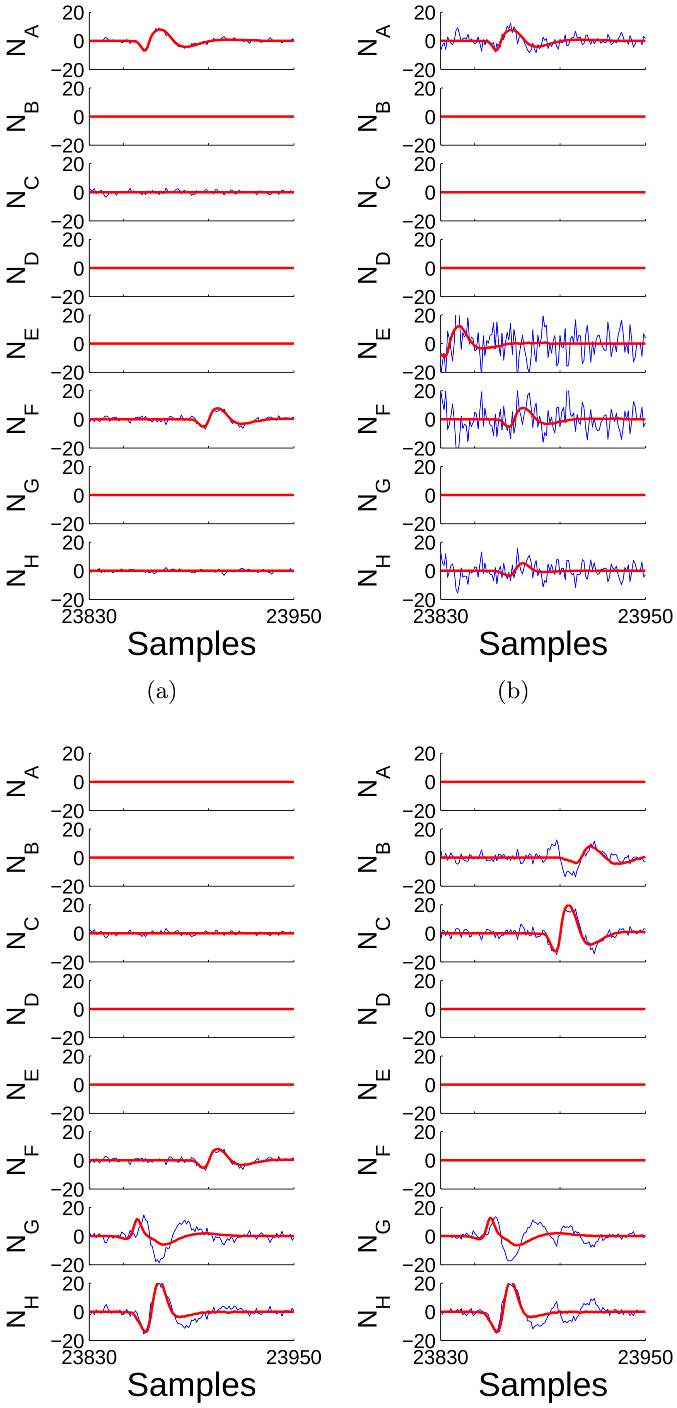
Once that k_j^C and B_j^C have been estimated for each presumed active neuron, also the hypothetical neural signal $\hat{s}^C(k) = [\hat{s}_1^C(k) \dots \hat{s}_n^C(k)]'$ is given:

$$\hat{s}_j^C(k) = \begin{cases} 0 & \text{if neuron } j \text{ is inactive} \\ B_j^C \hat{w}_j(k - k_j^C) & \text{if neuron } j \text{ is active} \end{cases}, j = 1, \dots, n. \tag{3.30}$$

The hypothetical signal $\hat{S}^C = [\hat{s}^C(1), \dots, \hat{s}^C(K)]$ represents the combination of neural signals that best explains the reconstruction V^C, under the assumption that only neurons selected by C can be active; $\hat{S}^C$ is no more than a set of consistent neural templates opportunely scaled and translated, where scales and translations are amplitudes and arrival times of the expected neural spikes in the rows of $V^C = (\hat{A}C)^{\dagger}Y$. Obviously, only when C includes all active neurons and $\hat{A}C$ is well conditioned, the rows of V^C exhibit the right waveform templates, and the parameters k_j^C and B_j^C really represent the time shift and the amplification of the template $\hat{w}_j$.

Figure 3.22 shows the hypothetical signals $\hat{S}^C$, superimposed to the traces V^C, in the 4 different cases of Figures 3.20 and 3.21, related to 4 different matrices C: in correspondence of neurons that are considered active, $\hat{S}^C$ shows the respective waveform templates, opportunely shifted and amplified. Notice that in (a), where the matrix C includes all the two effectively active neurons N_A and N_F, the hypothetical signal follows well the trend of traces in V^C. This does not happens in (b), where, even if the matrix C includes both the two active neurons, the matrix $\hat{A}C$ is not well-conditioned. Even in (c) and (d), the hypothetical signal $\hat{S}^C$ disagrees with the traces in V^C since one or both the active neurons are excluded by C.

The situation is slightly more complicated if an ambiguous cluster $Clust_j$ is individuated in the learning phase, and its mixing matrix column $\hat{A}_j$ is thus



(a)

(b)

(c)

(d)

Figure 3.22: The hypothetical signals $\hat{S}^C$ (thick lines), superimposed to the traces V^C (thin lines), in the 4 different cases considered in Figures 3.20 and 3.21, related to 4 different matrices C.

associated to Q waveform templates $\hat{w}_j^q(k)$, $q = 1\dots,Q$. In that case it is necessary to calculate the cross-correlation between the row $v_j^C(k)$ and each of the Q templates, estimating for each one the time shift, the amplification, and a different $\hat{S}^C$ that must be kept as well in consideration together with the others.

B3 - Selection of the minimum distance between hypothetical and original measurements — In this step, all the hypothetical signals $\hat{S}^C$, that have been estimated for each $C \in \mathcal{C}$ (and for each possible waveform of ambiguous clusters), are re-mixed with the matrix $\hat{A}$, thus obtaining an equal number of hypothetical measurements $\hat{Y}^C = \hat{A}\hat{S}^C$.

Figure 3.23 shows the hypothetical measurements $\hat{Y}^C$, corresponding to the 4 hypothetical signals $\hat{S}^C$ in Figure 3.22, related to 4 different matrices C. Notice that the hypothetical measurement in (a) is very similar to the original measurement in Figure 3.19, since it corresponds to a matrix C that includes all the two effectively active neurons N_A and N_F, and the corresponding matrix $\hat{A}C$ is well-conditioned; while the other hypothetical measurements in (b), (c), and (d), highly disagree with the original measurement.

All the hypothetical measurements $\hat{Y}^C$, for each $C \in \mathcal{C}$, are thus compared with the original measurement Y, by means of the Frobenius distance:

$$d_F(Y, \hat{Y}^C) = ||\Sigma^{-1}(Y - \hat{Y}^C)||_F = ||\Sigma^{-1}(Y - \hat{A}\hat{S}^C)||_F, \tag{3.31}$$

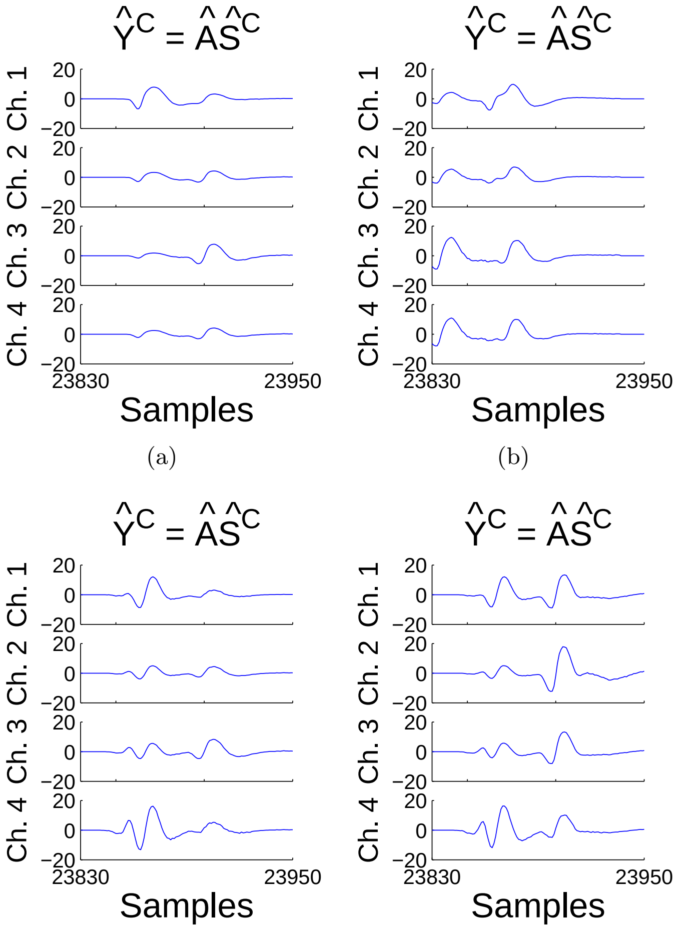
where $||A||_F = \sqrt{\sum_i \sum_j a_{ij}^2}$ is the Frobenius norm, and the diagonal matrix $\Sigma^{-1} = \mathrm{diag}(\sigma_{\eta_1}^{-1}, \dots, \sigma_{\eta_m}^{-1})$ normalizes the noise standard deviations of different sensors, so that they become comparable. For example the Frobenius distances in the 4 cases of Figure 3.23 are respectively: $\sim$ 14.5, $\sim$ 50.6, $\sim$ 106.0, and $\sim$ 61.8.

Among all the combinations of neurons $C \in \mathcal{C}$, the one that minimizes this distance between hypothetical and original measurements is chosen, as the one that best explains the observed measurements, both in terms of active neurons, and in terms of conditioning:

$$\hat{C} = \arg\min_{C \in \mathcal{C}} ||\Sigma^{-1}(Y - \hat{Y}^C)||_F. \tag{3.32}$$

For example, the Frobenius distance between the hypothetical measurement in 3.23a and the original measurements in 3.19 is the minimum among all the 70 hypothetical measurements obtained for each $C \in \mathcal{C}$.

B4 - Spike Classification — This step concludes the classification of the spike in the data segment under analysis. The hypothetical signal $\hat{S}^{\hat{C}}$ is considered as the neural signal estimation, and it indicates which neurons, among those selected by $\hat{C}$, are active and contribute to the observed data segment. The spike arrival times, defined as the time of the maximum positive peak, are



(a)

(b)

(c)

(d)

Figure 3.23: The hypothetical measurements $\hat{Y}^C = \hat{A}\hat{S}^C$, corresponding to the 4 hypothetical signals $\hat{S}^C$ in Figure 3.22, related to 4 different matrices C.

related to $k_j^{\hat{C}}$, known by step B2. Nevertheless, spikes that are too near the borders of the data segment, are not taken into account in the classification results.

For example, in the case of the measurement in Figure 3.19, the minimum distance is obtained with the matrix $\hat{C}$ that selects neurons N_A, N_B, N_F, and N_H. The corresponding hypothetical signal $\hat{S}^{\hat{C}}$, shown in Figure 3.22a, considers inactive the two neurons N_B, and N_H, thus concluding that neurons N_A and N_F are contributing to the segment of data. Considering the arrival time as the sample where the neural spike is maximum, the two arrival times are respectively at samples 23872, and 23905. Since the two spikes are almost inside the window analysis, they are both included in the classification results.

If no one spike has been detected in the rows of $V^{\hat{C}} = (\tilde{A}\hat{C})^{\dagger}Y$ — or if it is detected but it is too near the borders — the data segment is not classified, and have to be possibly analyzed with other procedures. This may be due to different reasons, such as some anomalies in the recording, or spike superpositions with ill-conditioning problems, or small contributions from neurons not modeled in step A4 because further and weaker than the foreground neurons.

Figures 3.24 and 3.25 show some other examples of overlapping spikes from the tetrode recording in the locust (*Schistocerca americana*) antennal lobe, that have been successfully resolved by MCI4SC algorithm. The first example shows a superposition of two spikes fired by neurons N_D and N_H; since the superposition is total, only one peak is visible in the recording. The second example shows a superposition of 3 spikes generated by neurons N_A, N_D, and N_G. Obviously, the classification phase of the MCI4SC algorithm is able to classifies single spike as well.

If there is an ambiguous cluster, the minimization (3.32) includes all its spike waveforms and the algorithm automatically selects the template that best explains the measured data. This works under the assumption that no more than a waveform from the same ambiguous cluster is present in the same segment; this is typically satisfied when the different waveforms come from the same neuron that cannot fire within a little period of time, while it is not guaranteed when the different waveforms come from different neurons at the same relative distances from electrodes.

3.5 Comparison with similar approaches

As already said, the MCI4SC algorithm has an approach similar to that of the two algorithms in [74] and [19], that likewise attempt to resolve overlapping spikes using multichannel recording (see Section 2.4). All these 3 methods adopt the linear mixing model of Eq. (3.1), but they handle it in different ways.

For example, the algorithm in [74] does not directly estimate the mixing matrix A: using ICA, it obtains the mixing matrix estimation as a consequence of source signals separation. ICA is usually adopted to solve the classical Blind

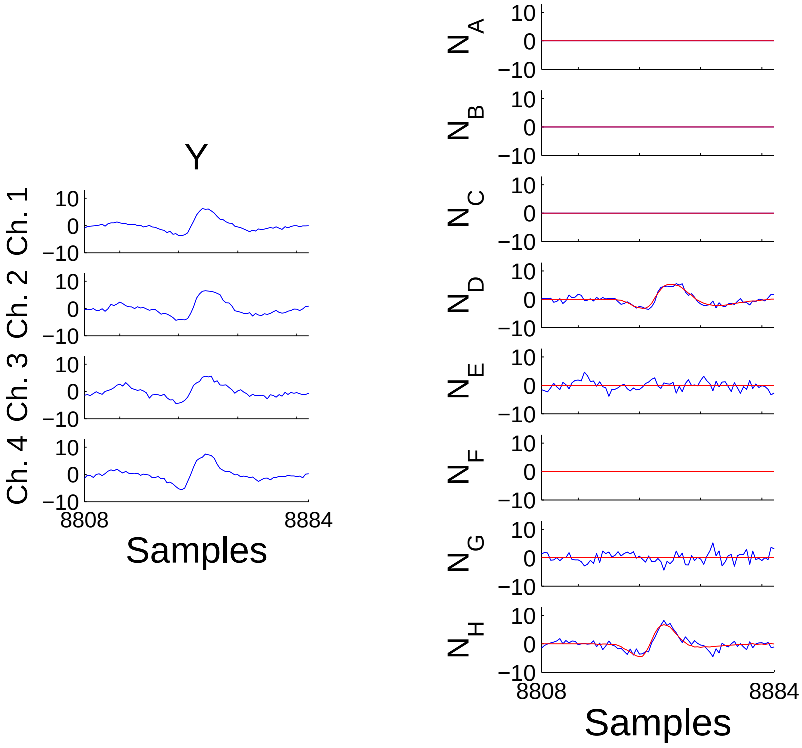


Figure 3.24: Classification of an overlap given by the total superposition of 2 spikes from neuron N_D and N_H. On the left, the data segment from the tetrode recording in the locust (*Schistocerca americana*) antennal lobe; on the right, the reconstructed signal $V^{\hat{C}}$ and the hypothetical signal $\hat{S}^{\hat{C}}$, where $\hat{C}$ represents the neuron combination N_D, N_E, N_G, and N_H, that minimizes the Frobenius distance $||Y - \hat{A}\hat{S}^C||_F$. The vertical scale is voltage in arbitrary units.

Source Separation (BSS) problem of the "cocktail party", where speech signals must be separated from a recording of people talking simultaneously in a room. ICA is usually applied under the assumption that source signals are independent, and the number of sensors m is equal to the number of sources n.

On one hand, neural traces from close neurons are not exactly independent, on the other hand, their recordings satisfy an additional condition that cannot be exploited in the "cocktail party" context: activities from different sources are not always present at the same time due to the sparsity and finite duration of neural spikes. Since it is possible to isolate, and recognize, data segments that contain a signal coming from just one neuron, the mixing matrix columns can be directly estimated, within a multiplicative factor, even if there are less sensors than neurons ($m < n$), and source signals are dependent.

MCI4SC algorithm and the one described in [19] take advantage of this neural recording characteristic, and directly estimate the mixing matrix after

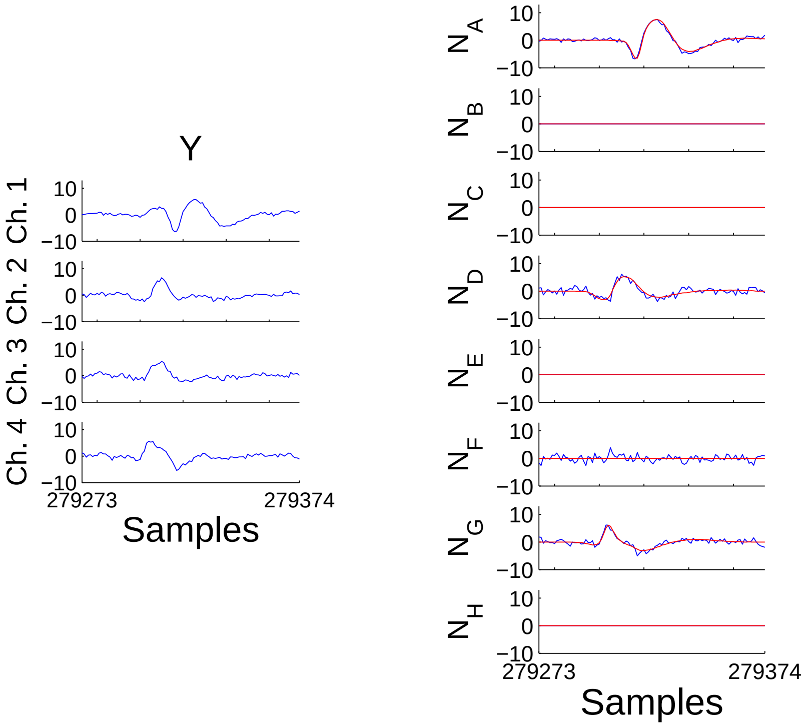


Figure 3.25: Classification of an overlap given by the superposition of 3 spikes from neuron N_A, N_D, and N_G. On the left, the data segment from the tetrode recording in the locust (*Schistocerca americana*) antennal lobe; on the right, the reconstructed signal $V^{\hat{C}}$ and the hypothetical signal $\hat{S}^{\hat{C}}$, where $\hat{C}$ represents the neuron combination N_A, N_D, N_F, and N_G, that minimizes the Frobenius distance $||Y - \hat{A}\hat{S}^C||_F$. The vertical scale is voltage in arbitrary units.

an initial clustering. Anyway, they do it in a different way: the method in [19] estimates the mixing matrix columns averaging all the waveforms in each cluster and then averaging in time and normalizing the absolute values in each channel (the method is somehow comparable to the estimation based on power ratios); while MCI4SC averages ratios of Wavelet Packet coefficients obtaining an estimation more reliable in case of low signal to noise ratio. The method in [19] differs from MCI4SC algorithm even because it does not care about the effect of overlapping spikes on the mixing matrix estimation: it assumes that their number is much smaller than that of single spikes, although this is not verified in some kind of recordings.

Like MCI4SC, also the algorithm described in [74] separately processes single spikes in a first phase, but — in the second phase — it differently treats overlapping spikes. To be more precise, it clusters spikes with irregular waveforms (overlapping spikes or spikes by neurons that show waveform variability) assuming that those generated by the the same subset of neurons exhibit similar features and tend to group together. Nevertheless, the shape of overlapping spikes strongly varies with the overlap phase, and, when the superposition is almost complete, the clustering of waveform/amplitude features does not necessarily groups events on the base of similar sources. On the contrary, MCI4SC algorithm, even if it deals less naturally than [74] with waveform variability, analyzes overlapping spikes one by one, without requiring their clustering.

All these 3 mentioned algorithms have to cope with the limitation of having less sensors than sources, yielding therefore a rectangular mixing matrix. To overcome this limitation, the algorithm in [74], uses the ICA on subset of spikes generated by a fraction of the total recorded neurons, thus resolving superpositions when the number of involved neurons is lower or equal to the number of sensors. The algorithm in [19] calculates, from the estimated mixing matrix, a set of projection matrices that in turn eliminate one neuron activity from the measured data: running a secondary spike sorting on each projection, the method can resolve all spike superposition involving two neurons (the complexity of the algorithm grows enormously to solve overlapping of more than two neurons). The method in [19] employs, as MCI4SC, the estimation of the mixing matrix, but, differently from MCI4SC, it does not exploit its inversion. Instead, under the assumption that less than m neurons are active at a time, MCI4SC algorithm inverts the matrix representing a reduced measurement channel, and is able to solve superpositions when the number of involved neurons is lower or equal to the number of sensors, as the method in [74] does.

Comparing, instead, MCI4SC algorithm with those spike sorting methods that computes the spatial position of neural spike activity, it is worth noticing that MCI4SC algorithm does not take explicitly into account the neuron positions, although it considers the amplitude ratio information associated to the relative distances between sources and electrodes. Therefore, MCI4SC algorithm does not even require to know the attenuation coefficient of the medium where signal is propagating, nor the exact mutual electrodes position, whereas such knowledge is required by those multichannel methods that exploit the spatial position of neural sources like the one in [17].

The last comparison is with those methods that explicitly deal with overlapping spikes by comparing each recorded waveform with all combinations of neural templates and all possible delays. MCI4SC algorithm has as well to try many combinations of neurons, but just those made of m neurons. In such a way, it can also resolve superposition of $2, 3, \ldots, m-1$ sources. Moreover, since MCI4SC makes use of combinations of neurons rather than combinations of templates, it does not require to try all the delays among waveforms, since the mixing matrix inversion allows to decompose superposition with every delay among spikes.

Chapter 4

Analysis of experimental data and Results

This chapter reports the application of the MCI4SC algorithm on a set of experimental data. The main results are compared with those of the Markov Chain Monte Carlo (MCMC) algorithm, that is presented in [26] and it is briefly described in the following. For the analyzed recording, MCI4SC has at least comparable efficiency with a much lower computational time, in addition to the important capability of overlapping spike resolution.

4.1 Evaluation of spike sorting algorithms

In order to evaluate the performances of a spike sorting algorithm, there are essentially two kind of data to use: i) simulated data, that unavoidably make assumptions on signal and noise characteristics, but assure the knowledge of the correct solution (i.e. the information of which spike correspond to which neuron); ii) or the experimental data that account for more realistic conditions, but the correct classification of their spikes is unknown. In the last case spike sorting results are usually compared with manual classification or better, when it is available, with simultaneous single-unit measurements.

A qualitative evaluation of spike sorting results on experimental data can also be given by the Inter-Spike Interval (ISI) distributions of classified spikes: if an ISI histogram clearly shows a refractory period, then their spikes are reasonably considered as coming from the same neuron. This, however, does not give any estimation of missed events, and a long collection period is required to be confident that spikes are indeed from a single neuron.

In the case of single spike classification, another qualitative evaluation is done by means of human inspection: when all spikes associated to the same neuron show the same trend, the sorting is reasonably assumed as correct.

Unfortunately, not so many simulated or experimental data are publicly available, and that limits the possibility to compare different algorithms. Christophe Pouzat and Matthieu Delescluse kindly share two set of experimental data on the website: www.biomedicale.univ-paris5.fr/SpikeOMatic/Data.html.

The first of the two data set is the tetrode recording from the locust (*Schistocerca americana*) antennal lobe, already shown and partially analyzed by MCI4SC algorithm in Chapter 3. The analysis details and the conclusive results are reported in Appendix B.

The following section illustrates the application of the MCI4SC algorithm on the second data set: the recording, by a linear probe, of Purkinje cells in a young rat cerebellar slice. This data set comes with a cell attached recording performed simultaneously on one of the Purkinje cells. Comparing the spike detected on this single-unit measurement with those associated by MCI4SC algorithm to the same cell, it is possible to quantitatively evaluate the spike sorting efficiency. The performances can be compared with those of the MCMC algorithm that, in [26], analyzes the same data set.

4.2 Purkinje cell data analyzed by MCI4SC algorithm

This section analyzes a multi-unit recording in a young rat (P12) cerebellar slice. The recording was performed positioning the probe along the Purkinje cells layer, in the presence of 40 μM DHPG (an agonist of metabotropic glutamatergic receptors) in order to make the Purkinje cells fire strong bursts.

The measurement is done using 4 of the 16 electrodes of a silicon probe provided by the Center for Neural Communication Technology of the University of Michigan. The 4 sensors are linearly placed on the probe 50 μm apart. The data were acquired at 15 kHz using a 16 bit A/D card. More details about the experimental procedure and the recording hardware can be found in [26].

The 58 seconds of data analyzed in this section are shown in Figure 4.1 (first four traces), together with a simultaneous single-unit recording of one of the recorded Purkinje cells (last trace). The single-unit recording comes from a patch-clamp pipette and serves as a reference signal to evaluate the spike-sorting performance.

Although the Purkinje cells activity has been recorded with an excellent signal-to-noise ratio, the analysis of this data set is very challenging: in these pharmacological conditions Purkinje cells fire burst of spikes with dramatically decreasing amplitudes and slightly different waveforms (see the triplet in the second channel of Figure 4.2).

Notice also that a spike recorded by a given sensor can be seen on its immediate neighboring sensors (50 μm apart) with reduced amplitudes, but never by further sensors. This is consistent with the exponential decay of the signal. Since the spikes are recorded by less then four sensors the neurons are not unequivocally distinguishable on the basis of the source position (or amplitude ratios). Anyway, even if MCI4SC potentialities are not fully exploited, the

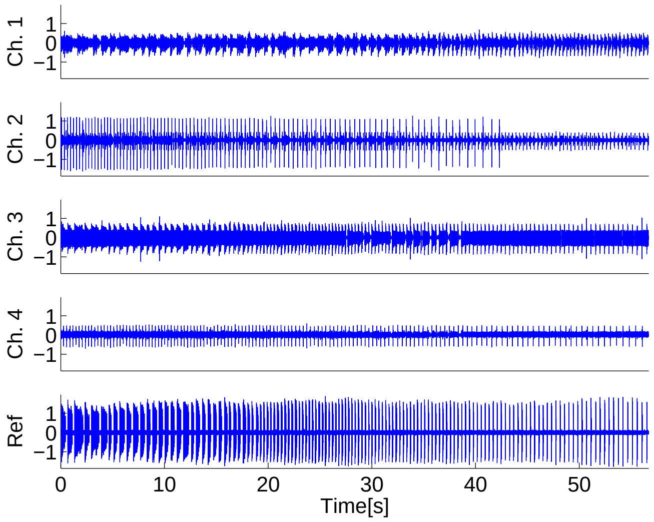


Figure 4.1: 58 seconds of a recording in the Purkinje cells layer of a young rat cerebellar slice; the data were acquired at 15 kHz and filtered between 300 Hz and 5000 Hz. The first 4 traces are the multi-unit recording from 4 sensors linearly placed in the silicon probe (amplified 2000 times). The last trace is the single-unit recording (amplified 1000 times). Voltage in vertical scale.

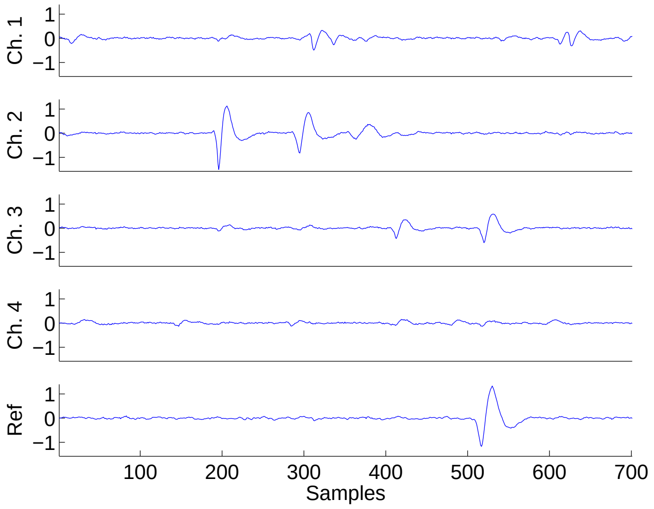


Figure 4.2: A segment of data from the Purkinje cells recording that is shown in Figure 4.1 (4 channels + reference channel). Notice the decreasing spike amplitudes in the short burst (triplet) of the second channel. Notice also that a spike recorded by a given sensor can be seen on its immediate neighboring sensors with reduced amplitudes, but never by further sensors. The reference neuron is primarily recorded by the third sensor. The vertical scale is voltage in arbitrary units.

obtained results are satisfactory.

Finally notice in Figure 4.2 that the reference neuron is primarily recorded by the third sensor. Another neuron is visible primarily in the third sensor, and its largest spikes are comparable in amplitude with the smallest spikes generated by the reference neuron at the end of each burst (see Figure 4.3).

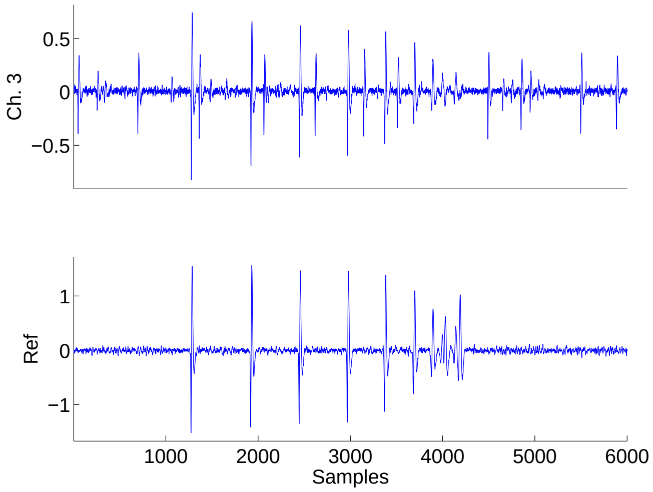


Figure 4.3: A segment of data from the Purkinje cells recording that is shown in Figure 4.1 (3rd channel + reference channel). Notice that the spikes in the reference channel are in correspondence with some of the spikes in the third channel. The remaining spikes in the third channel are generated by a different neuron. Its spikes are comparable in amplitude with the smallest spikes generated by the reference neuron at the end of each burst.

It is reasonable to assume that the measurement noise is Gaussian with practically zero mean. Figure 4.4 shows the normally distributed noise samples from data segments without spikes.

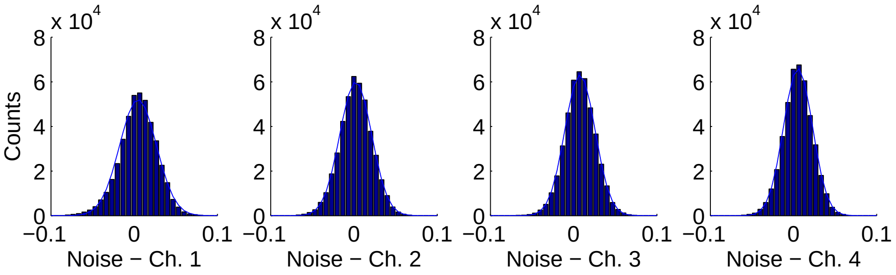


Figure 4.4: Histograms of the noise samples in data segments between one spike and another, in the four channel of the Purkinje cells recording that is shown in Figure 4.1. The distribution is fitted with a Gaussian approximately centered in zero.

On the other hand, it is worth noticing that the measurement noise is colored.

The MCI4SC algorithm requires the noise whiteness to exploit the capability of the Wavelet Transforms to separate the signal from the noise. The requirement, however, can be relaxed when the signal to noise ratio is high, or when the noise spectral support is spread with respect to the signal spectral support. This is the case of the Purkinje Cells recording under analysis.

A0 - Preliminary steps: filtering, spike detection, and analysis windows — First of all, the raw data, both the multi-unit and the single-unit recording, have been band-pass filtered between 300 and $5\,kHz$.

Secondly, the spike detection has been performed by an amplitude thresholding on the recorded samples. A different threshold T_d^i has been applied in each channel i ($i = 1, \dots, 4$) according to the respective level of additive noise η_i: the threshold T_d^i has been set as multiple of the standard deviation σ_{η_i} in the channel i. For a Gaussian distributed noise, the standard deviation σ_{η_i} can be estimated with the following formula based on the median:

$$\hat{\sigma}_{\eta_i} = median\left(|y_i|\right)/0.6745. \tag{4.1}$$

where y_i are the data recorded in the i-th channel. The standard deviation of the recorded data σ_{y_i}, including both noise and spikes, could be much more higher than the true noise standard deviation σ_{η_i}, especially when there are many large spikes. As shown in [64], the estimate of Eq.(4.1) diminishes the interference of the spikes, under the reasonable assumption that spikes amount to a small fraction of all samples.

A detection has occurred when a sample crosses, in absolute value, the threshold T_d^i in at least one channel. Two samples over the threshold and distant no more than T_{SP}, have been considered as belonging to the same spike, even if they exceed the threshold in distinct channels. In fact, since spikes oscillate around zero (especially in case of overlapping spikes), the samples over threshold can also be not contiguous. The check on the distance between samples over the thresholds T_d^i, by means of another threshold T_{SP}, allows to group distant samples of the same spike. In the following, k_F and k_L denote the first and the last sample over the detection threshold, thus defining the “detection window” for each detected spike.

In order to compare the results with those of the MCMC algorithm presented in [26], the thresholds T_d^i and T_{SP} have been set so that the number of detected spikes is equal to 2739. The threshold values that have been used are $T_d^i = 13.93 \cdot \hat{\sigma}_{\eta_i}$ (i.e. ~ 0.39, ~ 0.32, ~ 0.35, and ~ 0.31 respectively in the 4 channels) and $T_{SP} = 1.3\,ms$ (equal to 20 sampling times). In general, the choice of the thresholds must be done looking at the spikes in the recorded data. The thresholds T_d^i are chosen according to the number and amplitude of spikes to sort. Once T_d^i have been set, the threshold T_{SP} is chosen so that samples of the same spikes are, on average, grouped together.

The same thresholds T_d^i and T_{SP} have been used for the learning phase and for the classification phase, but in principle they may be different. The whole recording and all the 2739 detected spikes have been considered for the learning phase.

Finally a window is cut around each detected spike for further analysis. Since the learning phase uses the Wavelet Packet Transform to estimate amplitude ratios, it is convenient to use windows, named "learning windows", made of $K = 2^6 = 64$ samples ($\sim 4.3\,ms$). Notice that the maximum duration of detected spikes is about 60 samples, and thus the selected window length can entirely contain each single spike. All spikes have been aligned to their center of mass at sample 28. Usually, this rough alignment is sufficient for the purpose; otherwise, especially in case of low sampling, spike shapes can be oversampled, before the alignment, using interpolated waveforms.

In the classification phase, analysis windows with variable length have been adopted: they started $T_{SP} = 1.3\,ms$ (20 sampling times) before the first sample over the detection threshold, and they ended $T_{SP} = 1.3\,ms$ (20 sampling times) after the last sample over the detection threshold. In the following, k_B and k_E denote the samples where the "classification window" begins and ends for each detected spike ($k_B = k_F - 20$ and $k_E = k_L + 20$).

A1 - Single spikes versus overlapping spikes — First of all, 162 spikes with a detection window larger than $T_{long} = 25$ samples (i.e. $k_L - k_F + 1 \geq 25$), have been considered as overlapping spikes. The threshold T_{long} is chosen evaluating how many samples of the recorded spike waveforms are over the thresholds T_d^i.

Other 1773 spikes have been classified as overlapping spikes by checking the normalized cross-correlation $z_{r,i}$ with zero-lag between the reference channel r and the others 3 channels (see Eq. 3.15). At this stage, the reference channel r for a certain spike, is the one with the maximum sample in absolute value. The spike has been considered as an overlap when at least one of its cross-correlations $z_{r,i}$, for $i = 1, \ldots, m$, $i \neq r$, is under the corresponding threshold $T_{cross}^{r,i}$.

Consider the case in which the same signal w is recorded, with different amplitude, in the reference channel r and in the generic channel i (i.e. $y_r = \alpha_r w + \eta_r$ and $y_i = \alpha_i w + \eta_i$). Assuming that the additive noise in each channel is independent from the signal w, and it is Gaussian with zero mean (i.e. $\eta^i \sim \mathcal{N}(0, \sigma_{\eta_i})$ and $\eta^r \sim \mathcal{N}(0, \sigma_{\eta_r})$), the normalized cross-correlation between y_r and y_i can be approximate, on average, as:

$$\overline{z}_{r,i} = \frac{\sqrt{(\sum_{k=1}^{K} y_r^2(k) - K\sigma_{\eta_r}^2)(\sum_{k=1}^{K} y_i^2(k) - K\sigma_{\eta_i}^2)} + (K-1)cov(\eta_r, \eta_i)}{\sqrt{\sum_{k=1}^{K} y_r^2(k) \sum_{k=1}^{K} y_i^2(k)}}, \tag{4.2}$$

where K is the number of samples, and $cov(\eta_r, \eta_i)$ is the covariance between η_r

and η_i. The noise covariance is estimated from data segments without spikes:

$$cov\left(\begin{bmatrix}\eta_1\\ \vdots\\ \eta_4\end{bmatrix}\right)\sim\begin{bmatrix}0.49 & 0.19 & 0.09 & 0.07\\ 0.19 & 0.37 & 0.13 & 0.08\\ 0.09 & 0.13 & 0.33 & 0.12\\ 0.07 & 0.08 & 0.12 & 0.31\end{bmatrix}\cdot 10^{-3}. \tag{4.3}$$

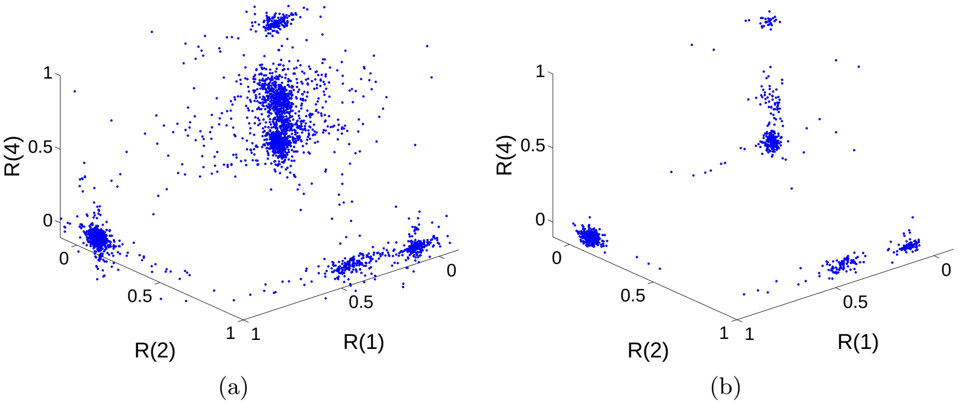
The standard deviation of the normalized cross-correlation $z_{r,i}$ can be as well approximated as:

$$\overline{\sigma}_{z_{r,i}}=\sqrt{\frac{\sigma^2_{\eta_r}(\sum_{k=1}^{K}y_i^2(k)-K\sigma^2_{\eta_i})+\sigma^2_{\eta_i}(\sum_{k=1}^{K}y_r^2(k)-K\sigma^2_{\eta_r})}{\sum_{k=1}^{K}y_r^2(k)\sum_{k=1}^{K}y_i^2(k)}}. \tag{4.4}$$

In Eq. (4.2) and (4.4), the quantities $\sum_k y_r^2(k)-K{\sigma_{\eta_r}}^2$ and $\sum_k y_i^2(k)-K{\sigma_{\eta_i}}^2$, that approximate $\sum_k w^2(k)$, are set equal to zero when they become negative. Concluding, the threshold for the cross-correlation $z_{r,i}$ is the set equal to:

$$T^{r,i}_{cross}=\overline{z}_{r,i}-3\overline{\sigma}_{z_{r,i}}. \tag{4.5}$$

Figure 4.5 represents spike-amplitude ratios of all detected spikes in (a), and spike-amplitude ratios of non overlapping spikes in (b). Notice that not all overlapping spikes have been excluded, but this first classification is sufficient to allow a right automatic clusterings in the following.



(a)

(b)

Figure 4.5: Tri-dimensional plot of three spike-amplitude ratios (the 1st, the 2nd, and the 4th) from the Purkinje cells recording. In (a), the spike-amplitude ratios of 2739 detected spikes. In (b), the spike-amplitude ratios of the 804 spikes not classified as overlapping spikes.

A2 - Estimation of spike-amplitude ratios — Following the procedure described in Section 3.3, the amplitude ratios of the 804 non overlapping spikes have been estimated as ratios of Wavelet Packet coefficients.

The Discrete Wavelet Packet Transform has been performed on the previously selected windows of $K = 64$ samples. To deal with the border distortion problem, the analysis windows has been extended according to the periodization mode.

The Wavelet theory suggests, in general, to use spike-shaped mother wavelets but there is not a more precise indication. Many different mother wavelets have been used for neural spike sorting: for example the 4th order Daubechies in [46], 3rd order Coiflet in [41], and the Haar in [64]. The mother wavelet used here to test MCI4SC algorithm is the 5th order Symlet, an orthogonal wavelet with compact support. The 5th order is a trade-off solution between frequency and time-resolution: larger orders ensure a better frequency-resolution, but imply a worse time-resolution and request larger analysis windows.

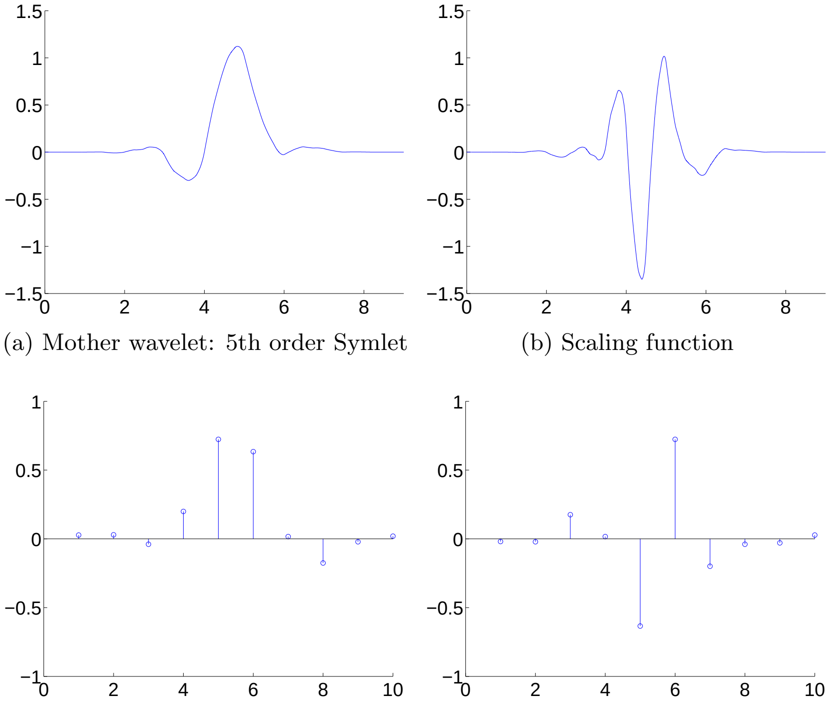
The WP Transformation has been performed at the decomposition level $L = 4$, that is, using periodization extension mode, the maximum level satisfying the necessary condition $K/2^{(L-1)} < DFL/2$, with window length $K = 64$, and decomposition filter length $DFL = 10$ (twice the wavelet mother order).

Figure 4.6 shows in (a) the 5th order Symlet, in (b), the associated scaling function, in (c) and (d) the decomposition low-pass filter g_0 and the decomposition high-pass filter g_1 associated to the 5th order Symlet.

A3 - Single spike Sorting — Using K-means algorithm the amplitude ratios of non-overlapping spikes have been clustered in 6 groups associated to an equal number of neurons (see Figure 4.7). The clustering algorithm has been opportunely initialized with a human intervention that provides the number of clusters and their approximate centroids.

Figure 4.8 shows the superimposed spike waveforms, cluster by cluster, as they are grouped in Figure 4.7. Notice the presence of different waveforms in the clusters. Human intervention is required to establish if different waveforms in the same cluster belong to only one bursting neuron or to more neurons at the same relative distances from the sensors. Since in this recording, that has been performed with electrodes in linear configuration, the spikes are measured just by 2 or 3 electrodes, their amplitude ratios does not unambiguously identify the neuron positions. Nevertheless, it is known a priori that these recorded neurons fire burst of spikes. Therefore it is reasonable to consider each cluster as generated by a unique neuron. For example, notice that the second cluster clearly exhibits 3 different spike waveforms, and remember that this recording contains a bursting neuron that fires triplets in the second channel (see Figure 4.2).

The MCI4SC algorithm naturally handles neurons that fire the same spike waveform with different amplitudes, but its complexity increases handling neurons that fire different spike waveforms. Since in this case the spike waveforms of the bursts change mostly in amplitude and slightly in shape, it is possible to treat those of the same cluster/neuron as the same waveform differently scaled,



(a) Mother wavelet: 5th order Symlet

(b) Scaling function

(c) Decomposition low-pass filter

(d) Decomposition high-pass filter

Figure 4.6: In (a) the 5th order Symlet. In (b) the associated scaling function. In (c) and (d) the decomposition low-pass filter g_0 and the decomposition high-pass filter g_1 associated to the 5th order Symlet.

still obtaining good results. This simplifies the spike sorting procedure and allows to save computational time. A more accurate (but time-expensive) analysis can be performed separately modeling all the spike shapes visible in each cluster/neuron.

A4 - Waveforms and mixing matrix estimation — To well estimate the spike templates and the mixing matrix, it is convenient to 'clean' the clusters found in the previous step by excluding overlapping spikes erroneously classified as single spikes. The cleaning operation has been performed in two step: i) first, outlier points with amplitude ratios too distant from cluster centroids have been removed; ii) then, spikes with shapes too different from the average waveform have been removed.

A well-known distance measure, often used to detect outliers, is the Mahalanobis distance which takes into account the correlations of the data set and does not depend on the scale of measurements.

Consider the cluster $Clust_j$, $j = 1, \ldots, 6$, that collects N_j events e associated

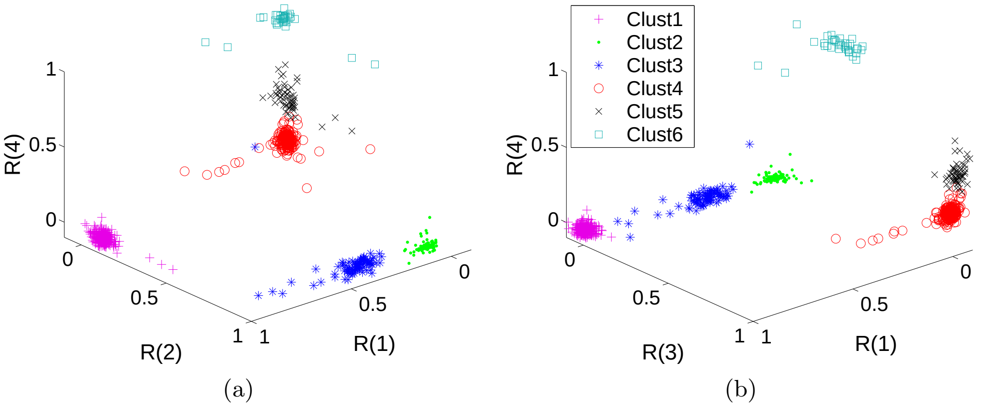


Figure 4.7: The amplitude ratios of the 804 non overlapping spikes from the Purkinje cells recording are grouped in 6 clusters. In (a), a tri-dimensional plot of the 1st, the 2nd, and the 4th spike-amplitude ratio. In (b), a tri-dimensional plot of the 1st, the 3rd, and the 4th spike-amplitude ratio. The numbers of spikes collected by the clusters from 1 to 6 are 303, 95, 87, 231, 53, and 35 respectively.

to an equal number of multivariate amplitude ratio vectors R^e. The cluster centroid $\overline{R}_j$ is usually estimated as the arithmetic mean of its sample, and the cluster covariance matrix S^j is usually estimated as the sample covariance matrix. The Mahalanobis distance between the vector R^e, $e \in Clust_j$, and the centroid $\overline{R}_j$ of its cluster is defined as:

$$D_M(R^e, \overline{R}_j) = \sqrt{(R^e - \overline{R}_j)'(S^j)^{-1}(R^e - \overline{R}_j)}. \tag{4.6}$$

The outliers typically exhibit Mahalanobis distances from their cluster centroids that are much greater than the Mahalanobis distances of the remaining points in the same cluster. Therefore, in the analysis of the Purkinje cells recording, the spikes $e \in Clust_j$ with distance $D_M(R^e, \overline{R}_j)$ greater than the threshold T^j_{Mahal} have been excluded by their cluster $Clust_j$. The threshold T^j_{Mahal} has been set to the mean distance from the $Clust_j$ centroid:

$$T^j_{Mahal} = \sum_{e \in Clust_j} \frac{1}{N_j} D_M(R^e, \overline{R}_j), \quad j = 1, \dots, 6. \tag{4.7}$$

Figure 4.9 shows the clusters of Figure 4.7 after this first cleaning step.

The second cleaning step calculates, first of all, an average waveform $\overline{y}_j(k)$ for each cluster $Clust_j$, $j = 1, \dots, 6$. The average considers the waveforms of the events e survived from the first cleaning step, as they have been recorded in the reference channel r_j (the channel with the largest spike amplitude). The

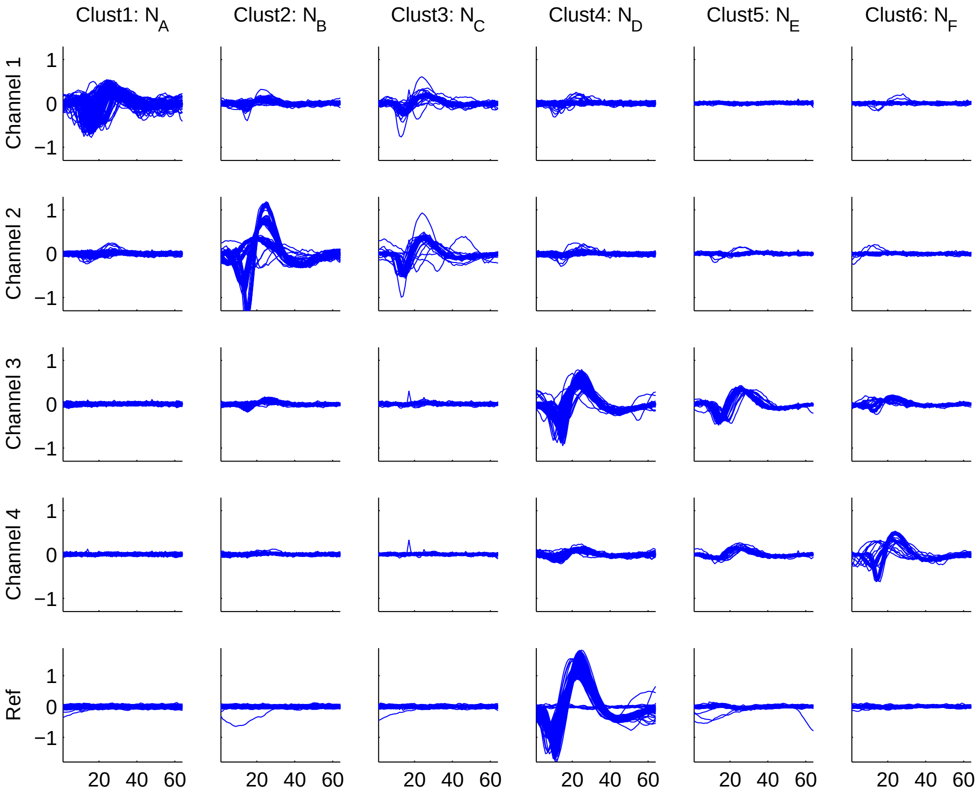


Figure 4.8: Superimposed spike waveforms from the Purkinje cells recording, as they are clustered in Figure 4.7. Notice the presence of different waveforms into the clusters. Since recorded neurons fire burst of spikes, each cluster has been considered as generated by a unique neuron. The vertical scale is voltage in arbitrary units.

cleaning operation comes to an end evaluating the maximum cross-correlations between the spike waveforms and the average waveforms $\overline{y}_j(k)$ of the respective cluster $Clust_j$. If the maximum cross-correlation is under a certain threshold T_w^j, the spike has been excluded by the cleaned cluster. Moreover, exploiting the information about the maximum cross-correlation lag, all spikes have been realigned to the average waveform.

In the analysis of the Purkinje cells recording, the thresholds T_w^j, with $j = 1, \ldots, 6$, have been set to 0.97, 0.80, 0.97, 0.97, 0.97, and 0.90 respectively. Notice that the second and the sixth thresholds are lower than the others. This allows to save the slightly different waveforms of the bursting neuron N_B and N_E, without separately modeling them. Figure 4.10 shows the superimposed spike waveforms for each cluster after the cleaning procedure. The numbers of spikes in the cleaned clusters associated to neurons $N_A, \ldots, N_F$ are 143, 70, 71, 185, 35, and 29 respectively. The percentages of removed events are 53%,

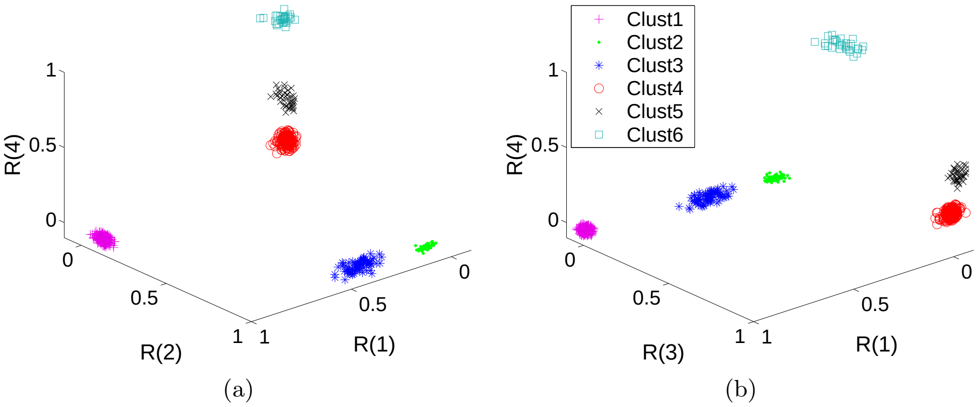


Figure 4.9: The amplitude ratios clusters of Figure 4.7 after the first cleaning step that has removed points too distant from the respective cluster centroid. 642 spikes in total are represented. In (a), a tri-dimensional plot of the 1st, the 2nd, and the 4th spike-amplitude ratio. In (b), a tri-dimensional plot of the 1st, the 3rd, and the 4th spike-amplitude ratio.

26%, 18%, 20%, 34%, 17%, respectively. Notice that the reference neuron of the single-unit recording corresponds to the neuron N_D.

Finally, the learning phase has been concluded by averaging the spike waveforms and the amplitude ratio vectors among all single spikes in each cleaned cluster of Figure 4.10. The estimated waveform templates $\hat{w}_1, \ldots, \hat{w}_6$ are on the bottom of Figure 4.10, and the estimated mixing matrix $\hat{A}$ (rounded to nearest hundredths) is the following:

$$\hat{A} = \begin{bmatrix} 1 & 0.13 & 0.46 & 0.01 & -0.02 & 0.01 \\ 0.13 & 1 & 1 & 0.04 & 0.00 & 0.04 \\ 0.02 & 0.10 & 0.10 & 1 & 1 & 0.39 \\ 0.01 & 0.02 & 0.04 & 0.20 & 0.43 & 1 \end{bmatrix}. \tag{4.8}$$

The two cleaning steps have been carried out automatically with the exception of the threshold setting. The human supervision is used to set the thresholds T^j_{Mahal} and T^j_w so that the cleaning result is visually good (see waveform superposition in Figure 4.10). In particular, it is necessary to reach a compromise between an accurate cleaning of the clusters and the numbers of spikes saved in each cluster (the greater is the number the better will be the related estimates).

According to desired accuracy in the neuron models, other methods can be used to clean the clusters, from the manual selection of single spikes to completely unsupervised techniques.

B1 - Pseudoinversion of the mixing matrices $\hat{A}C$, for each $C \in \mathcal{C}$ — For this Purkinje cells recording, since 6 neurons have been measured by 4

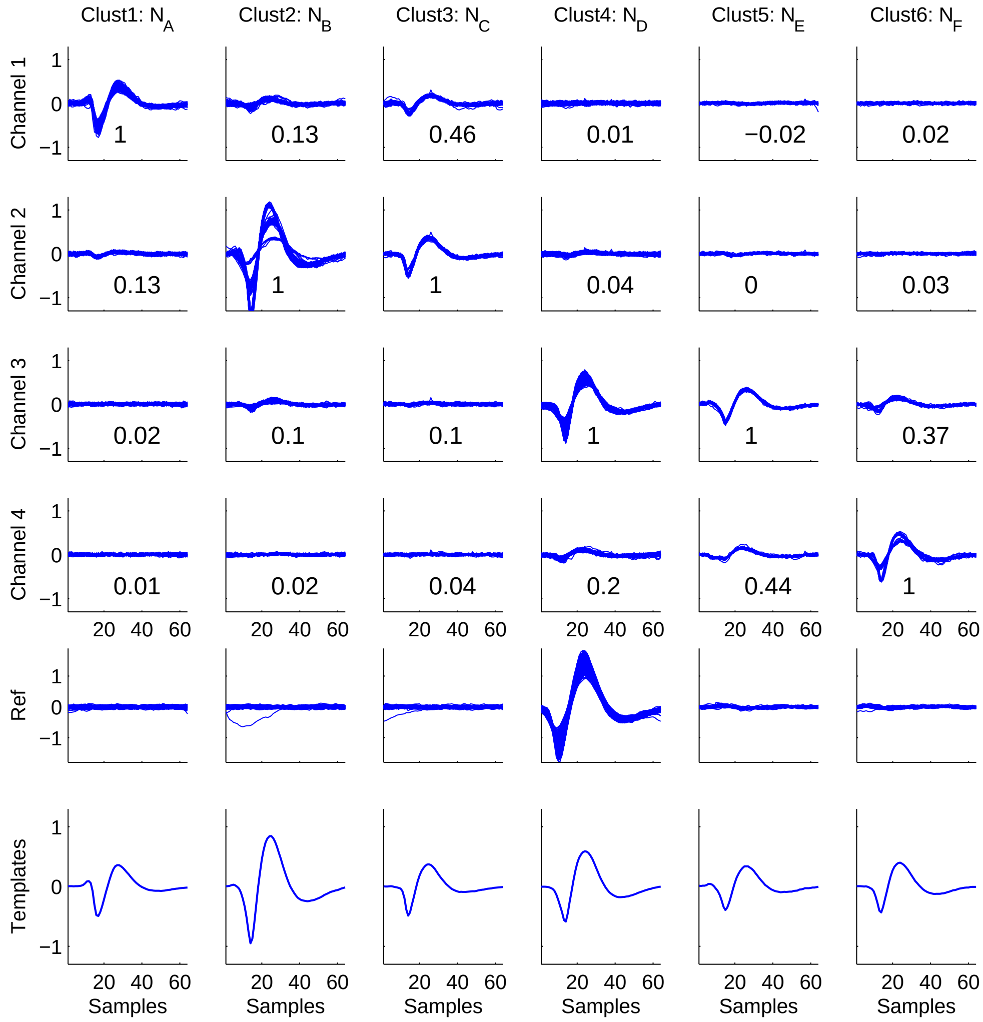


Figure 4.10: Superimposed spike waveforms of the clusters in Figure 4.9 after that they have been cleaned. The Figure shows also the estimated mixing matrix components in correspondence of Channel 1–4 and, on the bottom, the estimated waveform templates. Notice that the reference neuron of the single-unit recording named 'Ref' corresponds to the neuron N_D. The vertical scale is voltage in arbitrary units.

sensors, the number of neuron combinations in $\mathcal{C}$ is $\binom{6}{4} = 15$. Therefore, in this first step of the classification phase, 15 pseudoinverses $(\hat{A}C)^{\dagger}$ have been calculated. $\hat{A}$ is the estimated mixing matrix in Eq. (4.8), and C is a 6 by 6 diagonal matrix with 4 components equal to 1 in correspondence to the selected neurons in each of the 15 possible combinations.

According to which neurons the matrix C has selected or not, some columns of $\hat{A}$ rather than others contribute to the mixing matrix $\hat{A}C$, and the condition

Combination of neurons selected by C	Condition Number of the matrix $\hat{A}C$
$N_A\ N_B\ N_C\ N_D$	~ 127.5
$N_A\ N_B\ N_C\ N_E$	~ 131.8
$N_A\ N_B\ N_C\ N_F$	~ 264.7
$N_A\ N_B\ N_D\ N_E$	~ 9.7
$N_A\ N_B\ N_D\ N_F$	~ 1.9
$N_A\ N_B\ N_E\ N_F$	~ 2.4
$N_A\ N_C\ N_D\ N_E$	~ 9.7
$N_A\ N_C\ N_D\ N_F$	~ 2.0
$N_A\ N_C\ N_E\ N_F$	~ 2.4
$N_A\ N_D\ N_E\ N_F$	~ 55.2
$N_B\ N_C\ N_D\ N_E$	~ 10.1
$N_B\ N_C\ N_D\ N_F$	~ 6.8
$N_B\ N_C\ N_E\ N_F$	~ 6.9
$N_B\ N_D\ N_E\ N_F$	~ 91.3
$N_C\ N_D\ N_E\ N_F$	~ 248.2

Table 4.1: For each one of the 15 possible 4-combinations of 6 neurons selected by the matrix C, the respective condition number of the matrix $\hat{A}C$ has been shown.

number changes as well. The Table 4.1 shows the respective condition numbers of the matrix $\hat{A}C$, for each one of the 15 possible 4-combinations of 6 neurons selected by the matrix C.

Notice that the combinations with the largest condition numbers are those that contain at the same time the neurons N_A-N_B-N_C, or the neurons N_D-N_E-N_F. These two subsets of neurons are mostly visible in the same channels (N_A, N_B, N_C are mostly visible only in the first two channels; N_D, N_E, N_F are mostly visible only in the last two channels), and that makes their columns very "similar" (see the matrix in Eq. (4.8)). As a consequence, it will be more difficult to correctly classify the spikes made by the superposition of N_A, N_B, and N_C, or N_D, N_E, and N_F. This kind of limitation is not related to the method itself, but to the reciprocal positions between sensors and neurons.

B2 - Estimation of the hypothetical signals $\hat{S}^C$, for each $C \in \mathcal{C}$ — In this step, the 2739 detected spikes have been scanned, one by one, in their own classification window $y(k)$, $k \in [k_B, \dots, k_E]$. The classification window has been cut around the detected spike as described in A0, and has adaptive length.

For each spike, and for each combination $C \in \mathcal{C}$, the hypothetical signal reconstruction $v^C(k) = [v_1^C(k), \dots, v_6^C(k)]'$ has been calculated as $(\hat{A}C)^\dagger y(k)$. In correspondence to the two neurons that have not been selected by C, the rows $v_j^C(k)$ are null by construction. Instead, for the other 4 neurons that the matrix C has selected, the cross-correlations $z^j_{v,w}(k)$ — between the reconstruction rows $v^C_j(k)$ and the templates $\hat{w}_j(k)$ — have been evaluated (see Eq. 3.27) to determine the hypothetical signals $\hat{S}^C$ associated to the neuron combination C.

To be more precise, the hypothetical signal $\hat{S}^C$ is a collection of waveform templates $\hat{w}_j(k)$, opportunely amplified and translated (see Eq. 3.30). The template amplification B^C_j and the template translation k^C_j are related to the maximum of the cross-correlation $z^j_{v,w}(k)$ as described by Eq. (3.28) and (3.29). But, only if the amplification B^C_j is larger than the respective threshold T^j_B, the correspondent waveform template $\hat{w}_j$ has been accepted to contribute to the hypothetical signal $\hat{S}^C$. Otherwise, when the amplification B^C_j is too small, a null signal is been associated to the neuron j, as in the case of the two neurons not selected by C.

For each cluster j, the amplification threshold T^j_B has been set to the minimum amplification among those of spikes in the cleaned cluster $Clust_j$ of Figure 4.10. The correspondent values were: ~ 0.76, ~ 0.35, ~ 0.81, ~ 0.71, ~ 0.92, and ~ 0.70, for $j = 1, \ldots, 6$ respectively. Notice the low value of T^2_B that allows to accept a small spike amplification for the neuron N_B that fires spikes much smaller than the waveform template $\hat{w}_2$ (see Figure 4.10)

The thresholds T^j_B are very important for a correct classification. On the one hand, if they are too small, many little oscillations in the data (due to noise or anomalies) can be erroneously considered as neural spikes; on the other hand, if they are too large, they prevent some neural spikes to be considered because replaced by a null signal. The T^j_B setting is a very delicate matter especially when different neurons fire spikes with very similar shape that are recorded by few sensors. In this unlucky case, exploiting information about the possible amplification range is useful to avoid ambiguity and distinguish different sources. Whereas, in the case of unambiguous recordings, the choice of T^j_B is less determining and constraining.

In the case of the Purkinje cells recording under analysis, the reference neuron N_D fires spikes that are similar in shape to those of neuron N_E, and both of the two neurons are recorded only in the third and in the fourth channel. Unfortunately, the smallest spikes of neuron N_D have, in the third channel, an amplitude comparable to the greatest of neuron N_E. The adopted thresholds T^4_B avoids this ambiguity since it is large enough to disregard spikes from neuron N_E, even though it disregards also a percentage of small spikes from neuron N_D. Despite this lack in the classification of small spikes from the reference neuron, good classification results have been obtained.

B3 - Selection of the minimum distance between hypothetical and original measurements — For each detected spikes, the 15 hypothetical signals $\hat{S}^C$, one for each $C \in \mathcal{C}$, have been re-mixed with the matrix $\hat{A}$, thus obtaining an equal number of hypothetical measurements $\hat{Y}^C = \hat{A}\hat{S}^C$.

All these hypothetical measurements $\hat{Y}^C$ have been compared with the original measurement Y, by means of the Frobenius distance:

$$d_F(Y, \hat{Y}^C) = ||\Sigma^{-1}(Y - \hat{Y}^C)||_F = ||\Sigma^{-1}(Y - \hat{A}\hat{S}^C)||_F, \tag{4.9}$$

where the diagonal matrix $\Sigma^{-1} = \text{diag}(\sigma_{\eta_1}^{-1}, \ldots, \sigma_{\eta_4}^{-1})$ normalizes the noise standard deviations of different sensors, so that they become comparable.

Among the 15 combinations of neurons $C \in \mathcal{C}$, the one that minimizes this distance between hypothetical and original measurements has been chosen, as the one that best explains the observed measurements, both in terms of active neurons, and in terms of conditioning:

$$\hat{C} = \arg\min_{C \in \mathcal{C}} ||\Sigma^{-1}(Y - \hat{Y}^C)||_F. \tag{4.10}$$

B4 - Spike Classification — The sorting procedure has been concluded associating to each detected spikes a list of neurons $\{N_{j'}, N_{j''}, \ldots\} \subset \{N_A, \ldots, N_F\}$, and a list of corresponding arrival times $\{k_{j'}^{\text{MAX}}, k_{j''}^{\text{MAX}}, \ldots\}$ defined as the samples where the maximum positive peaks occur.

For each detected spike, the matrix $\hat{C}$, selected in the previous step, individuated the hypothetical signals $\hat{S}^{\hat{C}}$ that better explains the observed data segment. The neurons considered active in $\hat{S}^{\hat{C}}$ have been included in the neuron list, but only if their spikes were in central positions inside the analysis windows of the classification phase. For each neuron j the corresponding spike arrival time k_j^{MAX} can be calculated through $k_j^{\hat{C}}$, that has been found in step B2:

$$k_j^{\text{MAX}} = k_B + k_j^{\hat{C}} - 1 + \arg\max_k(\hat{w}_j(k)), \tag{4.11}$$

where k_B is the first sample of the classification windows, and $\hat{w}_j(k)$ is the spike waveform of neuron j.

To be more precise, the spikes have been considered in a central position inside the analysis windows when the maximum positive peak or the minimum negative peak were inside the detection window extended by 3 samples on the left and on the right, i.e. only if the spikes satisfied the following condition:

$$k_F - 3 \leq k^{\text{MAX}} \leq k_L + 3 \quad \text{or} \quad k_F - 3 \leq k^{\text{min}} \leq k_L + 3, \tag{4.12}$$

where k_F and k_L were the first and the last samples over the detection threshold, and k^{MAX}, k^{min} were the maximum positive and the minimum negative peak position respectively.

Thanks to this condition, the spikes that contributed to the data segment under analysis with a little portion of the whole spike waveform (the tail or the header) were not taken into consideration, since they were not properly part of the detected spike. Moreover, this condition guaranteed to consider each spike once, and not twice: in other words, each spike has been considered in one analysis window and not in the previous/subsequent window $T_{SP} = 20$ sampling times apart (notice that the estimated waveform had at most the maximum and the minimum peaks 11 samples apart, that is less than $T_{SP} - 3 \cdot 2$).

If no one neuron has been considered active by $\hat{S}^{\hat{C}}$, or if no one spike of active neurons has satisfied the condition in (4.12), the data segment under analysis has not been classified. As already said, this may be due to different reasons, such as some anomalies in the recording, or spike superpositions with ill-conditioning problems, or small contributions from neurons not modeled in step A4 because further and weaker than the foreground neurons.

Figure 4.11 and 4.12 show two examples of overlapping spikes, from the Purkinje Cells recording, that have been successfully classified by MCI4SC algorithm. The first example is a superposition of two spikes fired by neurons N_E and N_F. The second example is a superposition of 3 spikes generated by neurons N_B, N_C, and N_D.

Classification results — Among the 2739 detected spikes, 2407 have been classified as single spikes, 205 as superposition of 2 spikes, 11 as superposition of 3 spikes, and the remaining 116 have not been classified. The total number of sorted spikes is, therefore, 2850.

About the spikes considered as superpositions, the arrival time distance between two overlapping spikes varies as shown in the histogram of Figure 4.13 Notice that 33 overlapping spikes have arrival times distance < 5 sampling time ($\sim 0.33\,ms$). It is very difficult to correctly classify these spikes with algorithms that do not explicitly consider the overlapping spike problem.

Figure 4.14 shows, for each sorted neuron, the ISI histograms in semi-log scale, and the respective number of associated spikes. Notice that the ISI histograms show refractory periods of at least $3\,ms$ (45 sampling time).

The code was written in MATLAB programming language and run on 1.73 GHz Pentium PC running Microsoft Windows XP. The Analysis of 58 seconds Purkinje Cell recording (with 2739 detected spikes) takes, on this system, about 5 seconds for the preliminary step $A0$, about 24 seconds for the learning phase, and about 105 seconds for the classification phase (134 seconds in total).

The computational time linearly increases with the number of detected spikes, but heavily increases with the number n of neuron to sort. In fact, increasing n, the number of sensors m being equal, the possible number of m-combinations grows according to Eq. (3.24). At the same time, a larger number of sensors able to record the same group of spikes, heavily reduces the combination number.

Reference Channel — First of all, to match the reference waveforms with those in the extracellular measurements, the single-unit recording has been upturned, so that positive values were negative and vice versa. Moreover, the single-unit recording has been realigned with the extracellular recording through a shift of 5 sampling times toward the time origin.

The same detection procedure described in step A0 for the extracellular recording is applied to the single-unit recording: a detection has occurred when a sample crosses, in absolute value, the threshold T_d^{ref}; two samples over the threshold and distant no more than T_{SP}, have been considered as belonging to the same spike. In order to compare the results with those of the MCMC algorithm presented in [26], the thresholds T_d^{ref} and T_{SP} have been set so that the number of detected spikes is equal to 766. In particular, T_d^{ref} has been set to 0.832, and T_{SP} has been set to $1.3\,ms$ (20 sampling times).

Not all the 766 spikes that have been detected in the reference channel have also been recorded/detected in the third sensor of the extracellular recording.

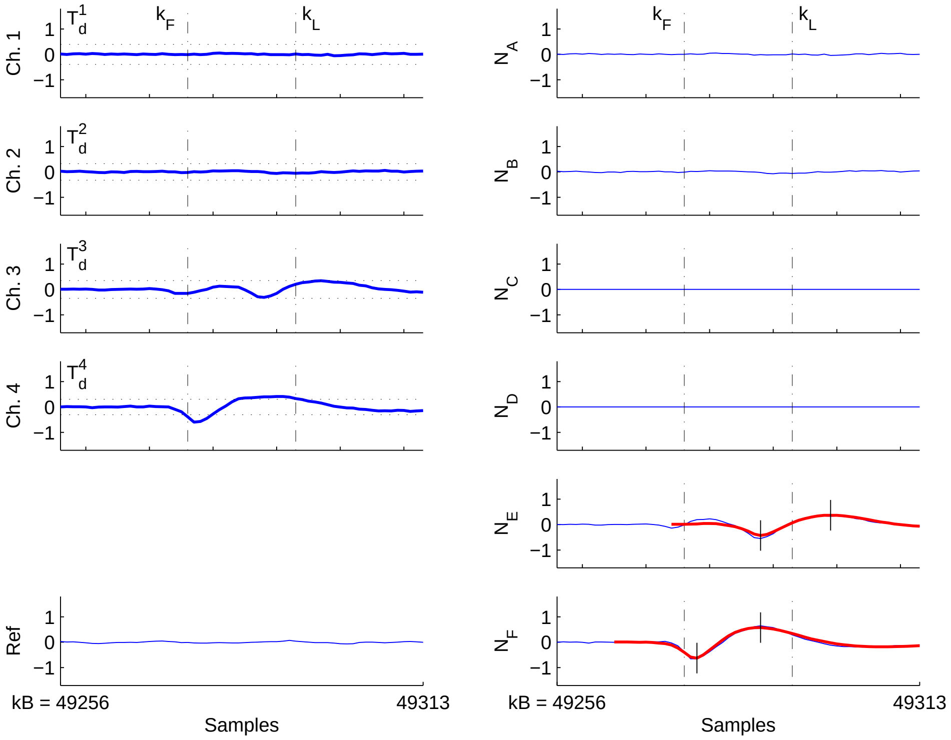


Figure 4.11: Classification of an overlap given by the superposition of 2 spikes from neurons N_E and N_F. On the left, the data segment from the Purkinje Cells recordings: the extracellular multi-unit recordings (Y) and the single-unit recording. On the right, the reconstructed signal $V^{\hat{C}}$ (thin lines), and the hypothetical signal $\hat{S}^{\hat{C}}$ (thick line). The matrix $\hat{C}$ represents the neuron combination N_A, N_B, N_E, and N_F that minimizes the Frobenius distance $||\Sigma^{-1}(Y - \hat{A}\hat{S}^{C})||_F$. The horizontal lines in the recorded data plots represent the detection threshold $\pm T_d^i$, $i = 1, \ldots, 4$. The sample k_B is the first sample of the classification window. The vertical lines, k_F and k_L, delimit the detection window. Both the two spikes have been included in the classification result because the maximum positive peak or the minimum negative peak are inside the detection window and satisfy the condition (4.12). The vertical scale is voltage in arbitrary units.

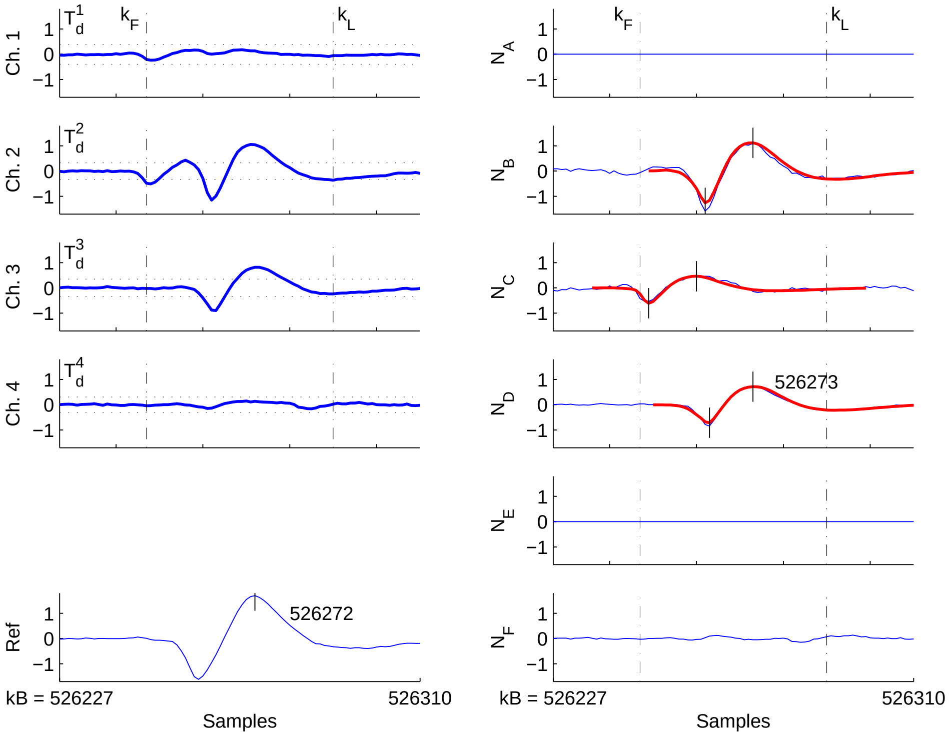


Figure 4.12: Classification of an overlap given by the superposition of 3 spikes from neurons N_B, N_C and N_D. On the left, the data segment from the Purkinje Cells recordings: the extracellular multi-unit recordings (Y) and the single-unit recording. On the right, the reconstructed signal $V^{\hat{C}}$ (thin lines), and the hypothetical signal $\hat{S}^{\hat{C}}$ (thick line). The matrix $\hat{C}$ represents the neuron combination N_B, N_C, N_D, and N_F that minimizes the Frobenius distance $||\Sigma^{-1}(Y - \hat{A}\hat{S}^C)||_F$. The horizontal lines in the recorded data plots represent the detection threshold $\pm T_d^i$, $i = 1, \ldots, 4$. The sample k_B is the first sample of the classification window. The vertical lines, k_F and k_L, delimit the detection window. All the 3 spikes have been included in the classification result because the maximum positive peak and the minimum negative peak are inside the detection window and satisfy the condition (4.12). The vertical scale is voltage in arbitrary units.

See, for example, that the last of the 3 detected spikes in Figure 4.15 is over the detection threshold T_d^{ref} in the reference channel, but it does not overcome the detection threshold T_d^3 in the third channel.

As already seen in Figure 4.10, the reference neuron corresponds to the neuron N_D. The MCI4SC algorithm has assigned 644 spikes to the neuron N_D. Among these spikes, 636 correspond to as many spikes detected in the reference channel. The correspondence is been accepted when the arrival time of the N_D spike is distant at most 3 samples ($\sim 0.2\,ms$) from the arrival time of the reference spikes. The others 8 spikes assigned to N_D are False Alarm.

Figure 4.16 show, as an example, the classification results of spikes detected

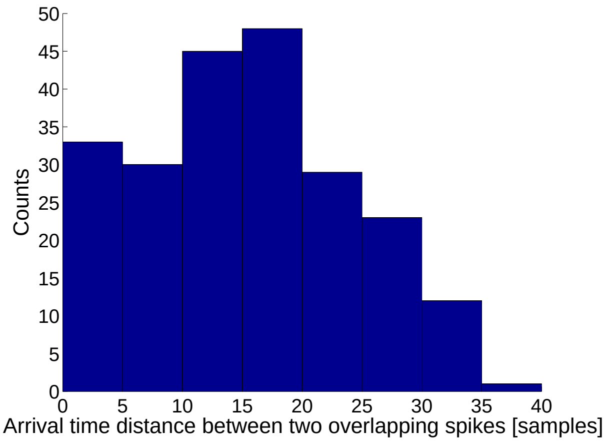


Figure 4.13: Histogram of the arrival time distance between two overlapping spikes. Bin width: 5 samples.

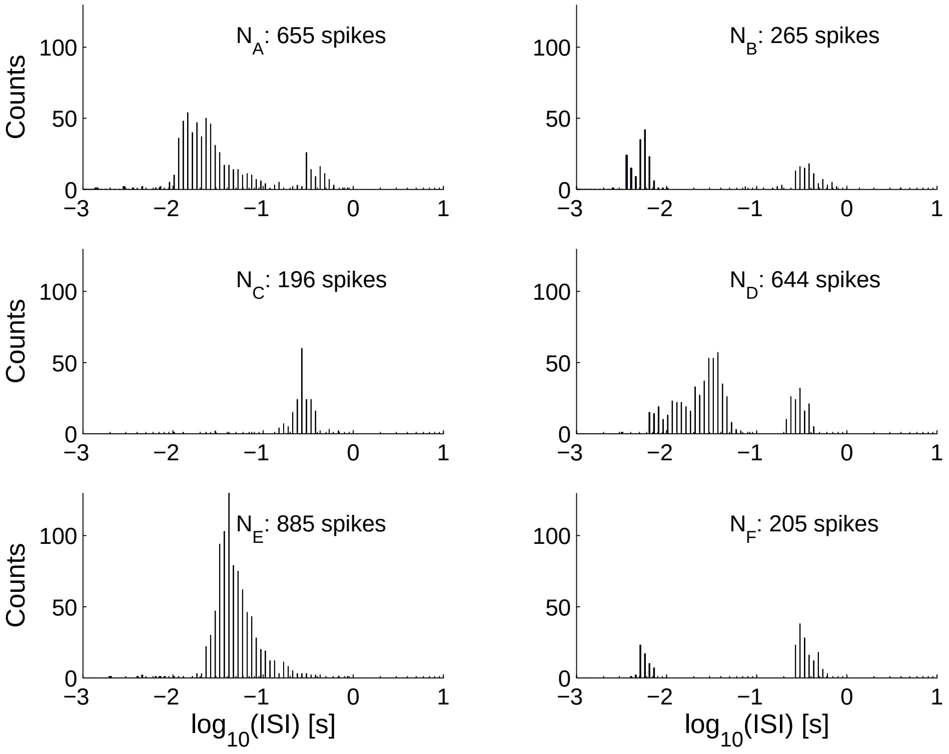


Figure 4.14: ISI histogram in semi-log scale of the 6 sorted neurons and the respective number of associated spikes. Notice a refractory period of at least 3 ms. The ISI is in seconds.

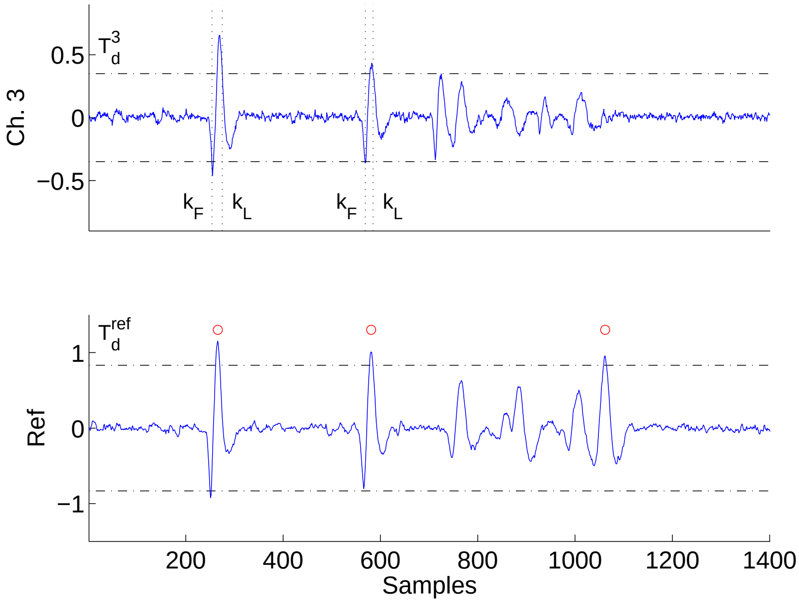


Figure 4.15: A segment of data from the Purkinje cells recording (3rd channel + reference channel). The horizontal lines represent the detection thresholds $\pm T_d^3$ and T_d^{ref}. The vertical lines, k_F and k_L, delimit the detection windows in the extracellular recording. Three spikes have been detected in the reference channel (3 circles indicates their arrival times). Notice that the last of the 3 spikes has not been detected in the third channel. The vertical scale is voltage in arbitrary units.

in the data segment of Figure 4.3.

4.3 Comparison with MCMC Algorithm

This section compares the results obtained by the MCI4SC algorithm, with those obtained by the MCMC algorithm described in [26], when both have been applied on the same data set (the Purkinje Cells recording). Before comparing results, the MCMC algorithm is described in its essential features.

The MCMC algorithm — The method developed by Delescluse and Pouzat, is built on a data generation model that takes into account both the firing statistics and the spike amplitude dynamics. To perform spike-sorting, this algorithm makes statistical inference on the parameters of the data generation model using a Markov Chain Monte Carlo (MCMC) approach, under the assumption that the measured spike amplitudes are corrupted by a Gaussian white noise which sums linearly with the spikes and is statistically independent of them.

To account for the empirical ISI density of the Purkinje Cells in DHPG, the method resorts to a Hidden Markov Model (HMM) with 3 log-Normal states.

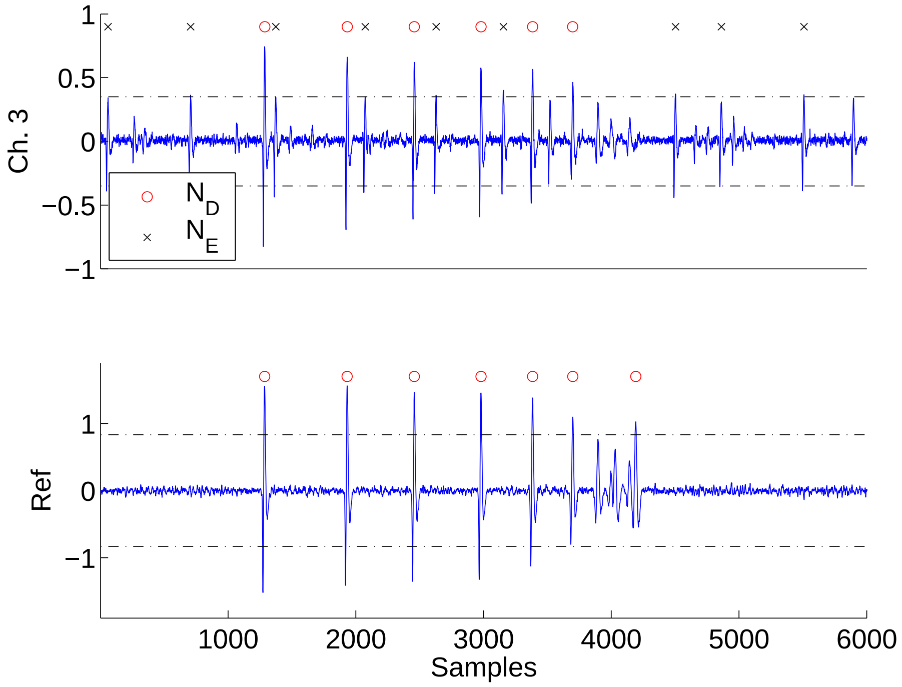


Figure 4.16: Spike sorting results on a segment of data from the Purkinje cells recording (3rd channel + reference channel). On the top, circles indicates the arrival times of spikes classified as coming from neuron N_D; on the bottom, circles indicates the arrival times of spikes detected in the reference channel. Crosses indicates the arrival times of spikes classified as coming from neuron N_E. The horizontal lines represent the detection thresholds $\pm T_d^3$ and $\pm T_d^{ref}$ respectively. Notice that some spikes do not overcome the thresholds and have not been detected. The vertical scale is voltage in arbitrary units.

In general, the sequence of ISIs produced by a given neuron can be seen as the observable output of a hidden sequence of states of this neuron (“hidden” in the sense that the state in which the neuron is, is not directly observable from the data). The 3 states of the model are related to 3 modes in the ISI distribution, and each event is fired so that its ISI is generated by one of the 3 possible probability densities according to the state in which the neuron is.

The 3 modes in the ISI distribution are respectively associated to bursty firing (first mode at high frequency with short ISI), tonic firing (second mode at intermediate frequency with intermediate ISI) and pauses between them (third mode at low frequency with long ISI). These 3 modes reflect the the Purkinje Cells activity in DHPG that typically evolves from a tonic firing to a bursty firing separated by intervals of several hundreds of ms. The ISI density of each state is been modeled as a log-Normal density:

$$f(isi) = \frac{1}{isi\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{1}{2}\left(\frac{log(\frac{isi}{s})}{\sigma}\right)^2}, \tag{4.13}$$

where s is a scale parameter (in seconds), and σ is a shape parameter (dimensionless). Therefore, 6 parameters are necessary to describe the 3 log-Normal densities.

Beyond these 6 parameters, other 6 parameters are necessary to describe the transition matrix between the 3 states: after the generation of each event, a transition from the current state to any other (including itself) is performed stochastically. In conclusion, 12 parameters specify the ISI distribution for each neuron.

Events are described not only by their occurrence time, but also by their peak amplitude measured on the 4 channels. The spike amplitude, that depends on elapsed time from the previous spike, has been modeled by an exponential relaxation:

$$A(isi) = P(1 - \delta e^{-\lambda\, isi}), \tag{4.14}$$

where isi is the ISI, λ is the inverse of the relaxation time constant (measured in seconds^{-1}), P is the 4-dimension vector of the maximal possible amplitude with the same modulation on the 4 channels, and δ is the dimensionless maximal modulation. Adding these other 6 quantities, the total number of parameters per neuron amounts to 18.

As it is explained in [26], the statistical inference on the model parameters relies on the construction of a Markov chain whose space $\mathcal{S}$ is the product of two spaces: the space of the model parameters defined in the data generation model, and the space of spike train configurations that specifies a neuron of origin and a neuron state for each spike. Therefore, a state of the Markov chain in this space is determined by two vectors: the vector θ of model parameters (a $(18 \cdot K)$-dimensional vector where K is the number of recorded neurons), and the vector C of the configuration, specifying a neuron of origin and a neuron state for each spike (a $(2 \cdot N)$-dimensional vector, where N is the number of detected spikes being analyzed). The construction of this Markov chain is done in such a way that it samples the space $\mathcal{S}$ from the posterior density $\pi_{post}(\theta, C|Y)$ of the model parameters and configurations, given the recorded data Y: at each step t of the algorithm a new state $[\theta^{(t)}, C^{(t)}]$ of the Markov chain is generated from the state at step $t-1$, $[\theta^{(t-1)}, C^{(t-1)}]$.

Once the simulated Markov chain has reached equilibrium — i.e. the chain is sampling from its stationary distribution which is the desired posterior density — the values of the parameters and the neurons of origin can be estimated, as well as errors on these estimates. This is done by averaging the value of a given parameter i over the performed algorithm steps, discarding the first steps necessary to reach equilibrium.

The excellent performance of this method relies on its ability to take into account the information provided by the occurrence time of the spikes, as well as their amplitude dynamics. It is moreover built on a proper probability model for data generation which, in that case, implies that convergence proofs of the algorithm do exist. This MCMC based approach provides, as well, meaningful confidence intervals on the model parameters and on the neuron of origin.

Nevertheless, the sorting of each detected spike, is not independently performed, since it is related to the time elapsed from the previous spikes. This means that an error in the spike detection or classification has consequence on the classification of subsequent spikes.

The MCMC algorithm can be easily adapted to different discharge models, or different numbers of states, but it needs a priori assumptions on them (possibly done on the base of single-unit recordings).

The method, as it is described in [26], is not fully automatic since it requires the user to choose the number of active neurons in the data, by individually scanning models with different numbers of neurons. The authors are working on a way to reliably compare models with different numbers of neurons, thus making automatic this initialization.

Comparison of results — It is worth noticing that the comparison between the two algorithms, MCI4SC and MCMC, cannot be straightforward, since the spike detection, both in the extracellular and in the single-unit recordings, has been performed in two different way.

As it is been described in the previous section (step A0), the spike detection implemented to evaluate the MCI4SC algorithm has been performed by thresholding the raw data, and considering two samples over the threshold, distant no more than T_{SP}, as belonging to the same spike.

The spike detection implemented to evaluate the MCMC algorithm has been differently performed. As described in [26], a first set of large events were detected as local maxima with a peak value exceeding a preset high threshold (5 times the standard deviation of the whole trace). These detected events were normalized (peak amplitude, at 1, temporal average, at 0) to give a “spike template”. Each trace was then filtered with this template (by convolution with the template in reversed time order). Finally, events were detected on the filtered trace as local maxima whose peak value exceeded a preset threshold (a multiple of the standard deviation of the filtered trace). To simplify calculations and reduce the complexity of the algorithm, the peak amplitudes were “noise whitened”.

In particular, notice that the MCMC algorithm does not consider spike superposition. Each detected peak is treated as a single spike, and two peaks detected onto different channels within less than 5 samples have been considered as only one spike [61] [25]. Therefore, when 2739 spikes have been detected, 2739 spikes have been classified as well by the MCMC algorithm. On the contrary the MCI4SC algorithm, analyzing 2739 detected spikes, attributed to their neurons of origin a different number of spikes (2850). Moreover, MCI4SC is able, in principle, to correctly classify overlapping spikes even in case of total superposition of more waveforms (up to the number of sensors), while MCMC algorithm assumes the presence of a single spike, and risks to introduce error in the ISI and in the classification of subsequent spikes.

The analysis results of MCMC algorithm (on 2739 detected spikes) were obtained after a 1000 MC steps with the Replica Exchange Method. All parameters had reached their equilibrium value after roughly 500 MC steps, and the average value of each model parameter has been computed using the last 200 MC steps. This required about 33 minutes on a $3\,GHz$ PC (Pentium IV) running Linux (codes were written in C). On the other hand, the MCI4SC algorithm requires about 134 seconds to analyze an equal number of detected spikes belonging to the same recording (considering 6 active neurons as MCMC has done). It is worth noticing, however, that the indicated time is valid once that all thresholds have been set. To obtain the best results, MCI4SC requires the human supervision during the preliminary step and the learning phase, in order to assure a proper threshold setting.

MCMC algorithm detected 766 spikes in the single-unit recording, using the same detection procedure used for the multichannel recording. At the end of the spike sorting procedure, it assigned 637 spikes to the reference neuron. Among these spikes, 629 correspond to as many spikes detected in the reference channel, and the others 8 spikes assigned to the reference neuron are False Alarms. For comparison remember that MCI4SC algorithm has assigned to the reference neuron 644 spikes; among these, 636 are considered as True Alarms, and the others 8 as False Alarms. The classification results provided by the two algorithms are more or less comparable, even though a fair comparison should be done adopting the same detection procedure.

Notice that not all the 766 spikes that have been detected in the reference channel are effectively recorded/detected in the multichannel recording (see Figure 4.15). Among the 766 reference events detected on the single-unit recording, only 641 are detected on the third channel by the MCMC detection procedure, thus leading to an efficiency equal to $629/641 = 98.1\%$. Nevertheless, the number of reference spikes detected in the third channel is not fully significant because it obviously depends on the detection procedure, and on the way that the coincidences third/reference channel are defined.

As a final consideration on the sorting results, notice that the neurons N_B and N_F produce triplets of spikes of very different amplitudes, that are clearly visible in the recorded data (see for example Figure 4.2).

In the case of neuron N_F, both the two algorithms correctly classify its detected spikes. Nevertheless, only the first one or the first two spikes of each triplet are effectively detected.

The case of neuron N_B is more challenging, since the third spike of the triplet has often been detected and it has, in the second channel, amplitude comparable to those of spikes fired by neuron N_C.

A total of 100 triplets fired by neuron N_B are present in the second channel. The third spike of each burst has often been detected by MCMC algorithm, but it has been wrongly attributed to neuron N_C, instead of being attributed to neuron N_B like the first two spikes of the burst. This misclassification is essentially due to the fact that the model of spike waveform dynamics is not sufficiently supported by data from unit N_B. With the adopted model, the third spikes in bursts are more likely to come from unit N_C, whose events are of similar amplitude.

On the other hand, MCI4SC algorithm has detected all the three spikes in 60 cases, while in 40 cases only the first two spikes have been detected: a total of $60 \cdot 3 + 40 \cdot 2 = 260$ spikes generated by N_B have been detected. Among the 60 triplets completely detected, in 58 cases all the three spikes have been attributed to N_B, and in the 2 remaining cases the third spikes have been wrongly attributed to N_C (notice that, in this recording, MCI4SC is able to correctly classify small spikes that experimentalists usually do not take into account). Among the 40 detected doublet, in 39 cases the two spikes have been attributed to N_B, e in 1 case the second spike has been wrongly attributed to N_C. Summarizing, 257 spikes (over the 260 detected) have been correctly attributed to N_B (98.8%); 8 spikes (over the 265 attributed to N_B) do not belong to the typical triplet of N_B, and must be considered as False Alarm (3%).

Figure 4.17 shows, as an example, the classification results in a data segment (the first 700 samples are the same of Figure 4.2). In the second channel, there is a triplet from neuron N_B correctly classified, and, in the fourth channel, there is a doublet from neuron N_F. It is worth noticing that many existing methods, based on waveform or amplitude features, wrongly assign the 3 spikes of the triplet to 3 different neurons, and a supervisor has to subsequently group together these spikes.

In conclusion, the two algorithms, MCMC and MCI4SC, both obtain good results with two completely different approaches that exploit different neural features. The first is based on firing statistics and spike amplitude dynamics; the second is based on amplitude ratios and spike waveforms. Since the two algorithms use complementary information, in principle, they can both obtain better results somehow integrating the other approach. In other words, a more complex algorithm that conjugates these two methods, simultaneously exploiting both the temporal information (MCMC) and the information related to neuron position (MCI4SC), can provide even better results, although more computational expensive.

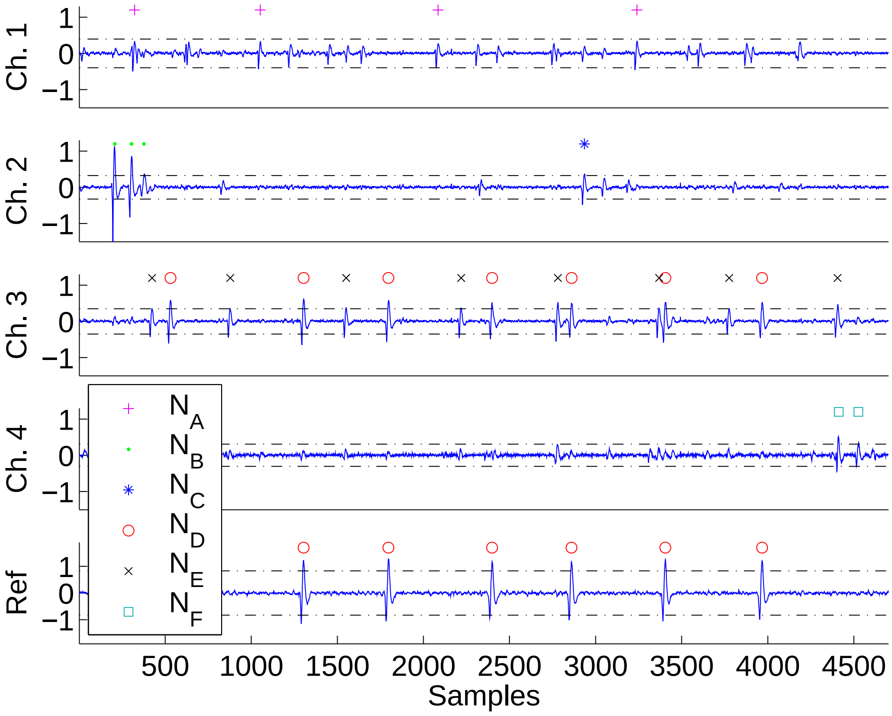


Figure 4.17: Spike sorting results on a segment of data from the Purkinje cells recording (4 channels + reference channel). Symbols are in correspondence of arrival times of classified spikes (in the channel where the spike is more visible). The horizontal lines represent the detection thresholds $\pm T_d^1$, $\pm T_d^2$, $\pm T_d^3$, $\pm T_d^4$, and $\pm T_d^{ref}$. In the second channel, there is a triplet from neuron N_B correctly classified, and, in the fourth channel, there is a doublet from neuron N_F. The vertical scale is voltage in arbitrary units.

Conclusions

In this thesis a new method, named MCI4SC, for the neural spike classification in multichannel recording is presented. In this context more extracellular electrodes simultaneously record a mixture of spike trains generated by surrounding neurons. The purpose is to separate the single neural traces in order to determine when each neuron fired. In other words, the request is to assign all the detected spikes to the neurons of origin, specifying the arrival times.

Many spike sorting algorithms, more or less efficient, have been developed until now: from early manual pattern recognition to more automated methods. The new spike sorting algorithm presented in this thesis, is unique in its use of the mixing matrix associated to the measurement channel. In particular, it is able to handle the disadvantageous, but typical, situation where there are more recorded neurons than recording sensors (under the reasonable hypothesis that the number of simultaneously firing neurons is lower than or equal to the number of sensors). In synthesis, this new algorithm inverts many mixing matrices associated to as many reduced measurement channel, individuating the one that best explains the recorded data. The reduced measurement channel involves only a number of neurons that is equal to the number of sensors, thus allowing the original signals separation.

The most attractive feature of MCI4SC is its capability to resolve overlapping spikes. This challenging aspect is often unattended by the most part of spike sorting algorithms, although it is extremely important for a correct spike classification, especially when the firing rate is high and superpositions are frequent.

Another challenging aspect, in the spike sorting framework, is that of bursting neurons. Many developed algorithms are unable to correctly classify such neurons, since they usually exhibit changing in the spike amplitudes and in the spike waveforms. On the one hand, MCI4SC naturally deals with amplitude variability, that does not affect the amplitude ratios and the correspondent mixing matrix. On the other hand, MCI4SC less naturally deals with waveform variability. Even if some passages, like mixing matrix estimation and inversion, are independent from the spike shape, nevertheless step B2 requires the knowledge of neural waveforms. Until, however, the different waveforms of a bursting neuron are in a finite number and detectable, the algorithm is able to reach a good classification even though it has a higher computational complexity. Moreover, if the different waveforms of a burst are similar enough, as in the case of the Purkinje Cells recording, they can be grouped together thus keeping low the computational time.

The application of MCI4SC on experimental data has confirmed its efficiency potentialities in spike classification. Good spike sorting results have been obtained for bursting Purkinje Cells, as well as for noisy neurons in Locust antennal lobe, even in case of total overlapping spikes. MCI4SC has also been compared with the MCMC algorithm described in [26], showing that MCI4SC efficiency is at least comparable with a much lower computational time.

Moreover, the amplitude ratio estimation by means of the Wavelet Packet Transform is another distinguishing feature of the new algorithm and its implementation. The Wavelet properties provide a consistent estimation of the amplitude ratios between channels even in case of low signal to noise ratio, where others estimator typically fail. This leads, in general, to a more reliable mixing matrix estimation.

Speaking about some other ideal requirements for the spike sorting algorithms, it is worth noticing that the nearly silent neurons, i.e. neurons that rarely fires, are not properly handled by MCI4SC algorithm. In fact, it requires a minimum number of single spikes to individuate and model each neuron. Anyway, it is reasonable to suppose that spike from a nearly silent neuron will be moved in the set of non classified spikes, that can be subsequently analyzed with other procedures.

About robustness to non stationary conditions, it is known that during the recording time, the electrodes may drift in the neural tissue. This results in a gradual change in the spike amplitudes and waveforms. If the drift is quite slow, it is possible to reestimate the mixing matrix and the spike template in every contiguous time interval where condition can be assumed stationary. Another assumption that can be violated during the course of the recording is the constant level of background noise. Some threshold values, that are related to it, have to be adaptively reset.

Although unsupervised algorithms are obviously preferable, given the complexity of the spike sorting problem, the best results are obtained with human supervision on quasi-automatic algorithms. MCI4SC obtains better results with manual intervention in order to set thresholds and to validate the single spike sorting of the learning phase.

Scalability is another interesting property for spike sorting algorithm. In principle, MCI4SC has not limitations on the number of sensors and recorded neuron. Nevertheless when the number of recorded neuron is much bigger than the number of sensors, MCI4SC may present computational challenges in light of the large number of neuron combinations to consider. In general, if the data analysis is limited to a small number of foreground neurons, it may be easy to keep low the computational time, possibly allowing the real-time operation.

Moreover, it is worth noticing that MCI4SC is independent from sensor geometry, even though it has better performances with non-planar configurations, that allow to univocally locate the neuron position. Planar or linear configuration, however, do not prevent the application of MCI4SC to the data: at most, they may possibly introduce some ambiguities in the neuron differentiation.

For the sake of completeness, two final remarks have to be done. Although this thesis focuses specifically on spike classification of neural signals, MCI4SC is applicable to other contexts where sparse impulsive signals are recorded by multiple sensors, and the measurements are well describe by linear mixing models with additive noise.

As a second remark, it is worth noticing that the new algorithm essentially exploits the information related to the neuron position and the spike waveform. In principle, however, better spike sorting results can be obtained exploiting all the features that potentially differentiate one neuron from the others. Therefore, MCI4SC improvements are expected somehow including temporal information related to the Inter-Spike Interval.

Appendix A

Wavelet Analysis

In the framework of signal processing, mathematical transformations are applied to recorded samples in order to obtain information that are not readily available in the raw data. The Wavelet Transforms, for example, allow to enhance time-frequency structures otherwise hidden in the time domain. This appendix gives a brief introduction to the versatile mathematical tools of Wavelet Analysis whose properties result very useful in the context of neural spike recordings. See [75], [12], and [51] for a more detailed discussion on Wavelet Transformations and their properties.

A.1 The Continuous Wavelet Transform

While the well-known Fourier Transform decomposes a signal as a sum of sinusoids $\{e^{i\omega t}\}_{\omega\in\mathbb{R}}$, the Wavelet Transform decomposes a signal as a sum of wavelet functions $\{\psi_{s,\tau}(t)\}_{s,\tau\in\mathbb{R}}$. Contrary to the sinusoids, that are well localized in frequency but have infinite duration, the wavelets are well localized both in time and in frequency (they are oscillating functions with practically finite duration). As a consequence, the main feature of Wavelet Analysis is that it allows to localize in time the spectral components. For this reason the Wavelet Transform results particularly suitable for transient signals analysis.

To be more precise, the *Continuous Wavelet Transform* (CWT) decomposes a signal $x(t) \in L^2(\mathbb{R})$ into the following basis of functions:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{|s|}}\psi\left(\frac{t-\tau}{s}\right) \qquad s,\tau \in \mathbb{R},\ s \neq 0, \tag{A.1}$$

that is built translating and scaling the *mother wavelet* $\psi(t)$. The translation parameter τ, that shifts $\psi(t)$ in time, allows to examine different segment of the signal; the scale parameter s, that dilates and compresses $\psi(t)$, allows to examine different frequency range with different time-frequency resolutions. Thanks to the factor $1/\sqrt{|s|}$, all the basis functions have the same energy, when varying the scale s.

The continuous wavelet transform of a square-integrable signal $x(t) \in L^2(\mathbb{R})$ is expressed by the following integral:

$$CWT_x^{\psi}(s,\tau) = \int_{-\infty}^{+\infty} x(t)\psi^*_{s,\tau}(t)\,\mathrm{dt} \equiv < x(t), \psi_{s,\tau}(t) >, \tag{A.2}$$

where $< \cdot, \cdot >$ is the scalar product in $L^2(\mathbb{R})$.

It is possible to exactly reconstruct the original signal $x(t)$ from its wavelet transform $CWT_x^{\psi}(s,\tau)$, if the mother wavelet $\psi(t)$ satisfies the following admissibility condition:

$$\int_0^{\infty} |\Psi(\omega)|^2/\omega\,\mathrm{d}\omega < \infty. \tag{A.3}$$

A.2 Discrete Wavelet Transform and Multiresolution Analysis

The CWT is useful because it allows analytical evaluations, but its signal representation is very redundant. It is rather possible to consider the CWT on a discrete set of the parameters s and τ without redundancy and loss of information. In particular, the *Discrete Wavelet Transform* (DWT) is defined discretizing s and τ on a dyadic grid ($s = 2^l$, $\tau = k2^l$ with $l, k \in \mathbb{Z}$) and, when the chosen mother wavelet is orthogonal, the DWT is non redundant, as well as complete.

An efficient algorithm to calculate the DWT has been implemented when Mallat and Mayer discovered a fundamental relation between the Discrete Wavelet Analysis and the *Multi-Resolution Analysis* (MRA) [49] [50] [54]. In few words, the MRA provides a systematic method to construct particular functions named scaling functions, and from those the correspondent wavelets. In fact, scaling functions and wavelets can be expressed, in the frequency domain, as product of low pass and high pass FIR filters. As a consequence, the Discrete Wavelet Transform can be fast and efficiently calculated by means of a FIR filter bank, avoiding to directly calculate the mother wavelet.

To give some more details, the MRA is equivalent to constructing a hierarchy of signal approximations that live in a sequence of nested linear vector subspaces like the following:

$$\ldots \subset V_{l+1} \subset V_l \subset V_{l-1} \subset \ldots \subset L^2(\mathbb{R}) \quad l \in \mathbb{Z}, \tag{A.4}$$

where each subspace is associated to the scale $s = 2^l$, and has to satisfy suitable properties [51]. In particular, MRA is dyadic if $x(t) \in V_l \implies x(2t) \in V_{l-1}$. It is worth noticing that, in one direction, these successive subsets approximate the signal with greater and greater precision (i.e. $\lim_{l\to-\infty} V_l = L^2(\mathbb{R})$) while, in the other direction, they approach the null function, carrying less and less information (i.e. $\lim_{l\to+\infty} V_l = \{0\}$).

The MRA also defines the *scaling function* $\phi(t)$ as a function orthogonal to its discrete translations $\phi(t-k)$, thus producing the following orthogonal basis for each subspace V_l:

$$\{\phi_{l,k}(t)\}_{k\in\mathbb{Z}} \quad = \quad \{2^{-l/2}\phi(2^{-l}t-k)\}_{k\in\mathbb{Z}} \quad l\in\mathbb{Z}. \tag{A.5}$$

As already said, the scaling function $\phi(t)$ can be calculated in the frequency domain as the product of low-pass FIR filters:

$$\Phi(\omega)=\prod_{k=1}^{+\infty} H_0\left(\frac{\omega}{2^k}\right), \tag{A.6}$$

where $H_0(\omega)$ is the Fourier Transform of a Quadrature Mirror Filter (QMF) $h_0(n)$.

From the low-pass filter h_0, the corresponding high-pass FIR filter h_1 is constructed by reversing and multiplying by $(-1)^k$ the coefficients of h_0:

$$h_1(k)=(-1)^k h_0(-k-1). \tag{A.7}$$

Notice that, by construction, the QMFs h_0 and h_1 are half-band filters that satisfy the power complementarity condition $|H_0(\omega)|^2+|H_1(\omega)|^2=1$.

The corresponding mother wavelet $\psi(t)$ is obtained from the scaling function $\phi(t)$ and the high pass filter h_1, as described by the following product in the frequency domain:

$$\Psi(\omega)=H_1\left(\frac{\omega}{2}\right)\Phi\left(\frac{\omega}{2}\right). \tag{A.8}$$

Coming back to the vectorial linear spaces, let W_{l+1} be the orthogonal complement of V_{l+1} in V_l:

$$V_l=V_{l+1}\oplus W_{l+1}. \tag{A.9}$$

It is possible to demonstrate that, with minor conditions on $H_0(\omega)$ [24], the wavelet ψ obtained by Eq. (A.8), creates an orthogonal basis for W_l, through scaling and translations:

$$\{\psi_{l,k}(t)\}_{k\in\mathbb{Z}} \quad = \quad \{2^{-l/2}\psi(2^{-l}t-k)\}_{k\in\mathbb{Z}} \quad l\in\mathbb{Z}. \tag{A.10}$$

Recursively using Eq. (A.9), a generic vectorial space V_0 can be defined as the orthogonal sum of an *approximation space* V_L at the level $L\geq 1$, and every *detail space* W_L, W_{L-1}, …, W_1 from the level L to 1:

$$V_0=V_L\oplus W_L\oplus W_{L-1}\ldots\oplus W_1. \tag{A.11}$$

An orthogonal basis for V_0 is thus provided by the scaling functions $\{\phi_{l,k}(t)\}$ at the level $l=L$, and by the wavelets $\{\psi_{l,k}(t)\}$ at the level $l=L,L-1,\ldots,1$.

Let $\{a_{l,k}\}_{k\in\mathbb{Z}}$ be the orthogonal projection on V_l of a certain signal x, with respect to the basis $\{\phi_{l,k}(t)\}_{k\in\mathbb{Z}}$:

$$\{a_{l,k}\}_{k\in\mathbb{Z}}=\{<x,\phi_{l,k}>\}_{k\in\mathbb{Z}}; \tag{A.12}$$

and let $\{d_{l,k}\}_{k\in\mathbb{Z}}$ be the orthogonal projection on W_l of the signal x with respect to the basis $\{\psi_{l,k}(t)\}_{k\in\mathbb{Z}}$:

$$\{d_{l,k}\}_{k\in\mathbb{Z}} = \{< x, \psi_{l,k} >\}_{k\in\mathbb{Z}}. \tag{A.13}$$

The coefficients $\{a_{l,k}\}_{k\in\mathbb{Z}}$ and $\{d_{l,k}\}_{k\in\mathbb{Z}}$ are named *approximation* and *detail* coefficients, respectively. Notice that the coefficients $\{d_{l+1,k}\}$ represent the additional detail required to pass from the resolution level $l+1$ to l. In other words $\{d_{l+1,k}\}$ is the difference between the two approximations $\{a_{l,k}\} \in V_l$ and $\{a_{l+1,k}\} \in V_{l+1}$ of the signal x (see Figure A.1).

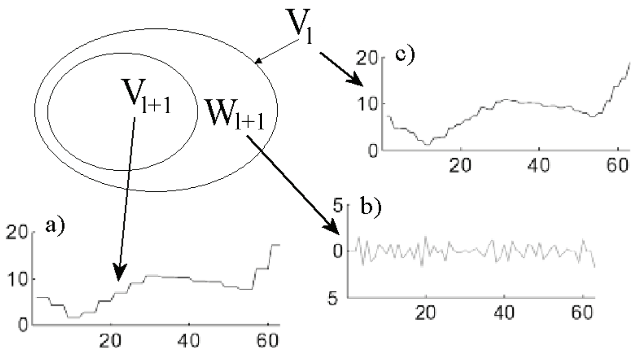


Figure A.1: The subset W_{l+1} is the orthogonal complement of V_{l+1} in V_l. As an example, in (a) and (c), two approximations $\{a_{l+1,k}\}$ and $\{a_{l,k}\}$ of a certain signal x at the level $l+1$ and l respectively. In (b), the detail $\{d_{l+1,k}\}$ of the signal x at the resolution level $l+1$, that added to (a) gives (c).

In conclusion, it is possible to decompose any signal $x(t) \in V_0$ as a unique linear combination of scaling functions and wavelets, by means of its approximation coefficients $\{a_{L,k}\}_{k\in\mathbb{Z}}$ at the last level of decomposition L and its detail coefficients $\{d_{L,k}\}_{k\in\mathbb{Z}}, \ldots, \{d_{1,k}\}_{k\in\mathbb{Z}}$ from level L to 1. This complete set of approximation and detail coefficients represents the DWT:

$$\{\mathcal{W}^L\} = \left\{\{a^L(k)\}_{k\in Z}, \{d^L(k)\}_{k\in Z}, ..., \{d^1(k)\}_{k\in Z}\right\}. \tag{A.14}$$

As a general remark, approximation coefficients have low frequency contents, while detail coefficients contain higher spectral components as the detail level l decreases.

The Multiresolution theory plays a crucial rôle in the implementation of the DWT as it provides a fast algorithm (analogous to the Fast Fourier Transform) that avoids unnecessary calculations of mother wavelets and scaling functions. Such decomposition algorithm — known as *Mallat's algorithm* — is based on a classical two channel subband coder implemented by the QMF bank, and its computational complexity is $\mathcal{O}(K)$, where K is the number of signal samples.

In particular, at each decomposition level $l + 1$, approximation and detail coefficients ($\{a_{l+1}\}$ and $\{d_{l+1}\}$) are obtained by the approximation coefficients of the previous level ($\{a_l\}$):

$$a_{l+1}(t) = \sum_{k \in Z} g_0(2t - k) a_l(k) \tag{A.15}$$

$$d_{l+1}(t) = \sum_{k \in Z} g_1(2t - k) a_l(k), \tag{A.16}$$

where $g_0 = h_0(-k)$ and $g_1 = h_1(-k)$ are two QMFs related to h_0 and h_1. Usually, the recursive MRA of the signal $x(t)$ is initialized identifying the set of samples $\{x_k\}_{k \in \mathbb{Z}}$ as the approximation coefficients $\{a_0, k\}$ in the space V_0.

A useful representation of signal decompositions over basis functions is the partition of the time-frequency plane induced by the basis functions themselves. Roughly speaking, any basis function can be identified by a tile in the time-frequency plane where the most of its energy is localized. For instance, the DWT is represented by constant area tiles with vertical height that increases as the frequency increases (see Figure A.2a). This time-frequency subdivision makes the DWT suitable for the analysis of slow signals in presence of fast noise fluctuations.

A.3 The Wavelet Packet Transform

The Wavelet analysis is not suitable for all signals, because the characteristic logarithmic frequency resolution is no more efficient when signals have high frequency spectral contents (e.g. bandpass signals). On the contrary, the *Wavelet Packet* (WP) Transform provides a more flexible division of the frequency domain; in particular, it allows “fine” frequency resolution also in the high frequency bands. In fact, the Wavelet Packets are localized in time like the Wavelets but, at the same time, they may have many oscillations to ensure higher frequency resolutions.

The idea of Coifmann, Meyer and Wickerhauser [22] [23] was to extend the subdivision of the approximation subspaces ($V_l = V_{l+1} \oplus W_{l+1}$) also to the detail subspaces, so that even W_l is divided in two orthogonal subspaces. This detail subdivision can be iterated at any level of the analysis, when it is opportune.

Since the detail subdivision may be done or not, the WP Transform is not unique. Each different WP Transform can be represented by an *admissible* binary tree, where open nodes (the leaves) identify the respective partition of the time-frequency plane. A binary tree is admissible when its nodes have either 0 or 2 branches.

The vectorial subspaces associated to the open nodes of an admissible binary tree are mutually orthogonal and their union is equal to the original space V_0, identified with the root of the tree. Therefore, the union of the Wavelet Packet bases of all the leaves in an admissible binary tree provides an orthogonal basis for V_0. Figure A.2 shows four examples of admissible binary trees with the associated partition of the time-frequency plane into tiles.

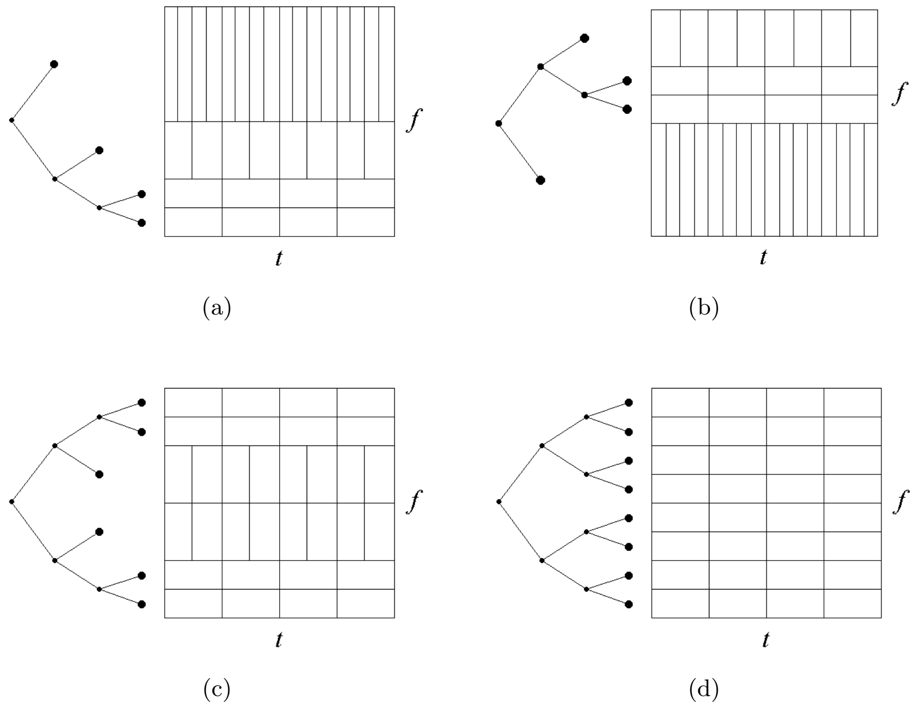


Figure A.2: Example of four admissible binary trees and the corresponding time-frequency tiles related to the open nodes. In particular, (a) shows the Dyadic binary tree associated to the DWT, and (d) shows a complete binary tree.

The WP coefficients $\{d_{l+1}^{2p}(t)\}_t$ and $\{d_{l+1}^{2p+1}(t)\}_t$, respectively associated to nodes $2p$ and $2p+1$ at the decomposition level $l+1$, are efficiently computed from the coefficients $\{d_l^p(t)\}_t$, of the parent node p at the level $l = \lfloor \log_2 p \rfloor$ by the following recursive relations:

$$\begin{aligned} d_{l+1}^{2p}(t) &= \sum_{k \in Z} g_0(2t-k) d_l^p(k) \\ d_{l+1}^{2p+1}(t) &= \sum_{k \in Z} g_1(2t-k) d_l^p(k), \end{aligned} \tag{A.17}$$

where g_0 and g_1 are the two QMFs related to h_0, h_1, ϕ, and ψ. The computational complexity of the algorithm is $\mathcal{O}(K \log_2 K)$ where K is the number of signal samples.

Figure A.3 shows an example of a WP Transform associated to a binary tree of depth 4. It is worth noticing that narrow time-frequency tiles are associated to innermost leaves and the outermost leaves correspond to broad tiles.

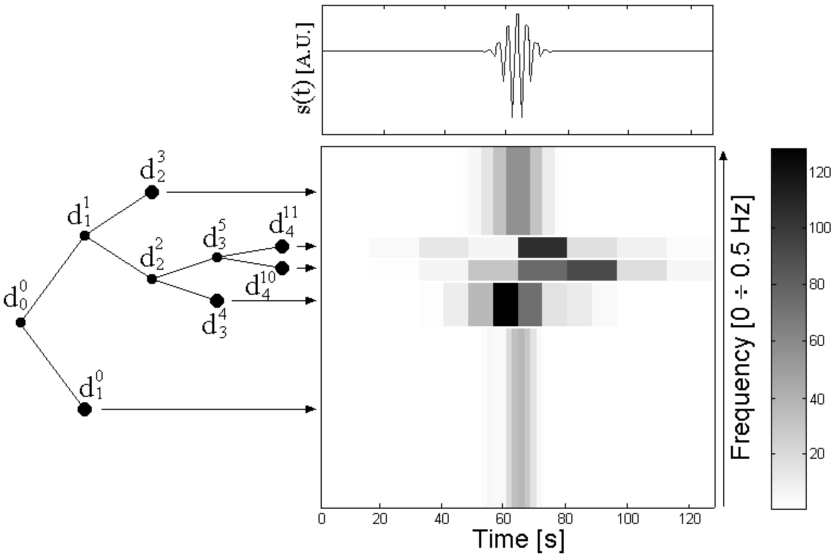


Figure A.3: An admissible binary tree of depth 4 and the corresponding WP Transform of the signal $s(t)$ plotted on the top. The coefficient absolute values are represented by tiles of the time-frequency plane using a gray-scale palette.

Appendix B

Locust data analyzed by MCI4SC algorithm

This appendix illustrates the application of MCI4SC on a tetrode recording from the locust (*Schistocerca americana*) antennal lobe. Some intermediate results have already been shown, as examples, in Chapter 3, and this appendix reports the analysis details and the conclusive results. No quantitative evaluation can be done on the analysis of this data set, since the true classification is unknown and no simultaneous reference recording is available.

The measurement is done using one of the 4 tetrodes on a silicon probe from the Center for Neural Communication Technology of the University of Michigan [30]. The probe has two shanks with 2 tetrodes each, and a tetrode is made by 4 electrodes in a diamond-shaped configuration. The width of each shank is about $80\,\mu m$, the diagonal length of each tetrode is $50\,\mu m$. The probe has been gently pushed into the antennal lobe (the first olfactory relay of the insect, whose diameter is approximately $400\mu m$), so that the lowest two tetrodes are roughly $50 - 100\,\mu m$ below the surface. The data were acquired at $15\,kHz$.

The 20 seconds of data analyzed in this section are shown in Figure B.1. Contrary to the Purkinje Cells recording analyzed in Chapter 4, in this data set there are not bursting neurons with dramatically decreasing amplitudes, but the signal to noise ratio is worse.

It is reasonable to assume that the measurement noise is Gaussian with practically zero mean. Figure B.2 shows the normally distributed noise samples from data segments without spikes. Also in this case the colored noise does not prevent to reach good spike sorting results. The noise covariance is been estimated from data segments without spikes:

$$cov\left(\begin{bmatrix}\eta_1\\ \vdots\\ \eta_4\end{bmatrix}\right) \sim \begin{bmatrix}0.19 & 0.07 & 0.06 & 0.07\\ 0.07 & 0.28 & 0.07 & 0.06\\ 0.06 & 0.07 & 0.41 & 0.07\\ 0.07 & 0.07 & 0.08 & 0.34\end{bmatrix}. \tag{B.1}$$

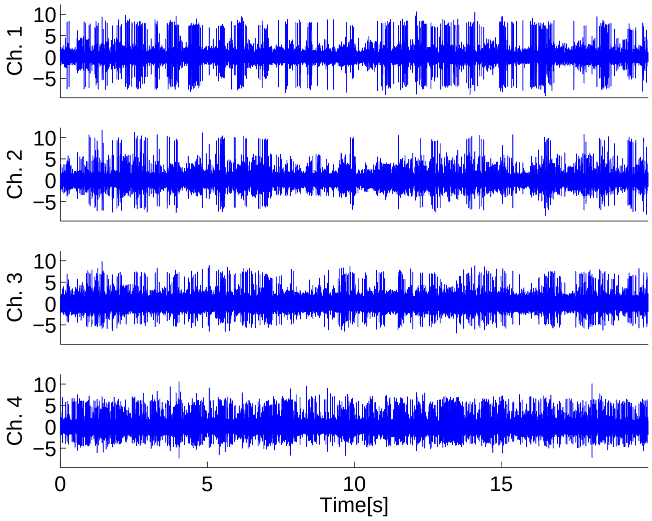


Figure B.1: 20 seconds of a tetrode in vivo recording from the locust (*Schistocerca americana*) antennal lobe; the data were acquired at 15 kHz and filtered between 300 Hz and 5000 Hz. The vertical scale is voltage in arbitrary units.

A0 - Preliminary steps: filtering, spike detection, and analysis windows — First of all, the raw data has been band-pass filtered between 300 and 5 kHz.

Secondly, spikes have been detected when a sample crosses, in absolute value, the threshold T_d^i in at least one channel i. The threshold T_d^i has been set as multiple of the standard deviation σ_{η_i}, estimated in each channel i according to Eq. (4.1). In particular, it has been set to $T_d^i = 6 \cdot \hat{\sigma}_{\eta_i}$ (i.e. ~ 3.12, ~ 3.77, ~ 4.39, and ~ 4.12 respectively in the 4 channels). Two samples over the threshold and distant no more than $T_{SP} = 2\,ms$ (30 sampling times), have been

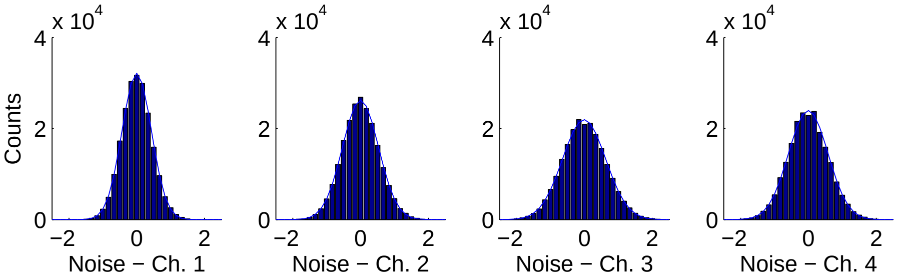


Figure B.2: Histograms of the noise samples in data segments between one spike and another, in the four channel of the recording that is shown in Figure B.1. The distribution is fitted with a Gaussian approximately centered in zero.

considered as belonging to the same spike.

With these threshold setting, used both in the learning phase and in the classification phase, 990 spikes have been detected. All these detected spikes have been considered in the learning phase.

Finally a learning window of $K = 2^6 = 64$ samples ($\sim 4.3\,ms$) is been cut around each detected spike for further analysis. Notice that since the duration of detected spikes is about 70 samples, the learning window contains the most part of each single spike. All spikes have been aligned to their center of mass at sample 28.

In the classification phase, analysis windows with variable length have been adopted: they started $T_{SP} = 2\,ms$ (30 sampling times) before the first sample over the detection threshold, and they ended $T_{SP} = 2\,ms$ (30 sampling times) after the last sample over the detection threshold.

A1 - Single spikes versus overlapping spikes — First of all, 77 spikes with a detection window length larger than or equal to $T_{long} = 35$ samples have been considered as overlapping spikes.

Other 168 spikes have been classified as overlapping spikes checking the normalized cross-correlation $z_{r,i}$ with zero-lag between the reference channel r and the others 3 channels (see Eq. (3.15)). At this stage, the reference channel r for a certain spike, is the one with the maximum sample in absolute value. The spike has been considered as an overlap when at least one of its cross-correlations $z_{r,i}$, for $i = 1, \ldots, m$, $i \neq r$, is under the corresponding threshold:

$$T^{r,i}_{cross} \overline{z}_{r,i} - \overline{\sigma}_{z_{r,i}}, \tag{B.2}$$

where $\overline{z}_{r,i}$ is defined as in Eq. (4.2), and $\overline{\sigma}_{z_{r,i}}$ is defined as in Eq. (4.4). This threshold his higher and thus more restrictive than the threshold in Eq. (4.5), that has been used in the analysis of the Purkinje Cells recordings where the signal to noise ratio was better.

Figure 3.2a shows the spike-amplitude ratios of all the 990 detected spikes, and Figure 3.2b shows the spike-amplitude ratios of the 745 spikes not classified as overlapping spikes.

A2 - Estimation of spike amplitude ratios — The amplitude ratios have been estimated as ratios of Wavelet Packet coefficients, following the procedure described in Section 3.3.

The Discrete Wavelet Packet Transform has been performed on the previously selected windows of $K = 64$ samples, using the 5th order Symlet as mother wavelet, and the periodization extension mode. The maximum decomposition level that has been considered was $L = 4$.

A3 - Single spike Sorting — Using K-means algorithm the amplitude ratios of non-overlapping spikes have been clustered in 8 groups associated to an equal number of neurons (see Figure 3.15). The clustering algorithm has been opportunely initialized with a human intervention that provides the number of clusters and their approximate centroids.

The numbers of spikes collected by the clusters from 1 to 8 are 90, 46, 65, 125, 44, 68, 166, and 141 respectively.

Figure 3.16 shows the superimposed spike waveforms, cluster by cluster, as they are grouped in Figure 3.15. Since each cluster exhibits mainly a unique waveform, it is unequivocally associated to a single neuron.

A4 - Waveforms and mixing matrix estimation — To well estimate the spike templates and the mixing matrix, the clusters found in the previous step have been cleaned by excluding overlapping spikes erroneously classified as single spikes. The cleaning operation has been performed in two step, as explained in Chapter 4: i) first, outlier points with amplitude ratios too distant from cluster centroids have been removed; ii) then, spikes with shapes too different from the average waveform have been removed.

For each cluster $Clust_j$, the first cleaning step removed all spikes with Mahalanobis distance from the cluster centroid greater than the threshold T^j_{Mahal}, where the threshold T^j_{Mahal} has been set to the mean distance from the centroid (see Eq. (4.7)).

The second cleaning step evaluates the maximum cross-correlations between each spike waveforms and the average waveforms $\overline{y}_j(k)$ of the respective cluster $Clust_j$. If the maximum cross-correlation is under a certain threshold T^j_w, the spike have been excluded by the cleaned cluster. Moreover, exploiting the information about the maximum cross-correlation lag, all spikes have been realigned to the average waveform. In the analysis of the Locust recording, the thresholds T^j_w, with $j = 1, \dots, 8$, have been set to 0.90.

Figure 3.17 shows the amplitude ratio clusters of Figure 3.15 after the two cleaning steps. The numbers of spikes in the cleaned clusters associated to neurons $N_A, \dots, N_H$ are 68, 21, 55, 75, 31, 42, 86, and 70 respectively (448 spikes in total). The percentages of removed events are 24%, 54%, 15%, 40%, 30%, 38%, 48%, and 50% respectively.

Figure 3.18 shows the superimposed spike waveforms of the cleaned clusters in Figure 3.17. The Figure shows also the estimated mixing matrix components (rounded to nearest hundredths) and, on the bottom, the estimated waveform templates $\hat{w}_1, \dots, \hat{w}_8$. They are obtained by respectively averaging the amplitude ratio vectors and the spike waveforms among all single spikes in each cleaned cluster. In this case the templates have been averaged on a window of 80 samples in order to entirely contain the waveform evolution.

The thresholds T^j_{Mahal} and T^j_w have been set with human supervision in order to reach a good cleaning result and thus a good single spike classification. Since all spikes associated to the same neuron show in Figure 3.18 the same trend, the sorting is reasonably assumed as correct.

B1 - Pseudoinversion of the mixing matrices $\hat{A}C$, for each $C \in \mathcal{C}$ — For this tetrode recording, since 8 neurons have been measured by 4 sensors, the number of neuron combinations in $\mathcal{C}$ is $\binom{8}{4} = 70$. Therefore, in this first step of the classification phase, 70 pseudoinverses $(\hat{A}C)^\dagger$ have been calculated. $\hat{A}$ is the estimated mixing matrix in Eq. (3.23), and C is a 8 by 8 diagonal matrix with 4 components equal to 1 in correspondence to the selected neurons in each of the 70 possible combinations.

According to which neurons the matrix C has selected or not, some columns of $\hat{A}$ rather than others contribute to the mixing matrix $\hat{A}C$, and the condition number changes as well between ~ 5.04 and ~ 4289 (49 neuron combinations have the condition number lower than 50).

The combinations with the largest condition numbers are those that contain at the same time neurons mostly visible in the same channel, like for example N_B, N_C, N_D mostly visible in the second channel, or N_E, N_F mostly visible in the third channel, or N_G, N_H mostly visible in the fourth channel.

B2 - Estimation of the hypothetical signals $\hat{S}^C$, for each $C \in \mathcal{C}$ — In this step, the 990 detected spikes have been scanned, one by one, in their own classification window.

For each spike, and for each combination $C \in \mathcal{C}$, the hypothetical signals $\hat{S}^C$ has been estimated as explained in section 3.4.

For each cluster j, the amplification threshold T_B^j have been set to the minimum amplification among those of spikes in the cleaned cluster $Clust_j$ of Figure 3.18. The correspondent values were: ~ 0.92, ~ 0.86, ~ 0.90, ~ 0.75, ~ 0.88, ~ 0.88, ~ 0.85, and ~ 0.77, for $j = 1, \ldots, 8$ respectively.

B3 - Selection of the minimum distance between hypothetical and original measurements — Among the 70 combinations of neurons $C \in \mathcal{C}$, the one that minimizes the Frobenius distance between hypothetical and original measurements has been chosen, as the one that best explains the observed measurements, both in terms of active neurons, and in terms of conditioning (see Eq. (3.32)).

B4 - Spike Classification — The sorting procedure has been concluded associating to each detected spikes a list of neurons and a list of corresponding arrival times defined as the samples where the maximum positive peaks occur.

The neurons considered active have been included in the neuron list only if the maximum positive peak or the minimum negative peak of their spikes were inside the detection window extended by 3 samples on the left and on the right. This guarantees that each spike has been considered in one analysis window and not in the previous/subsequent window $T_{SP} = 30$ sampling times apart (notice that the estimated waveform had at most the maximum and the minimum peaks 15 samples apart, that is less than $T_{SP} - 3 \cdot 2$).

If no one neuron has been considered active, or if no one spike of active neuron was in a central position inside the analysis windows, the data segment under analysis has not been classified.

Figures 3.24 and 3.25 show two examples of overlapping spikes from the tetrode recording in the locust (*Schistocerca americana*) antennal lobe, that have been successfully resolved by MCI4SC algorithm (they are respectively superpositions of 2 and 3 spikes).

Classification results — Among the 990 detected spikes, 816 have been classified as single spikes, 110 as superposition of 2 spikes, 16 as superposition of 3 spikes, 1 as superposition of 4 spikes, and the remaining 47 have not been classified. The total number of classified spikes is, therefore, 1088.

Figure B.3 shows, for each sorted neuron, the ISI histograms, in linear scale, and the respective number of associated spikes. For a qualitative evaluation, notice that the ISI histograms show refractory periods of about 20 ms (300 sampling time). The percentages of spikes from each neuron $N_A, \ldots, N_H$ with

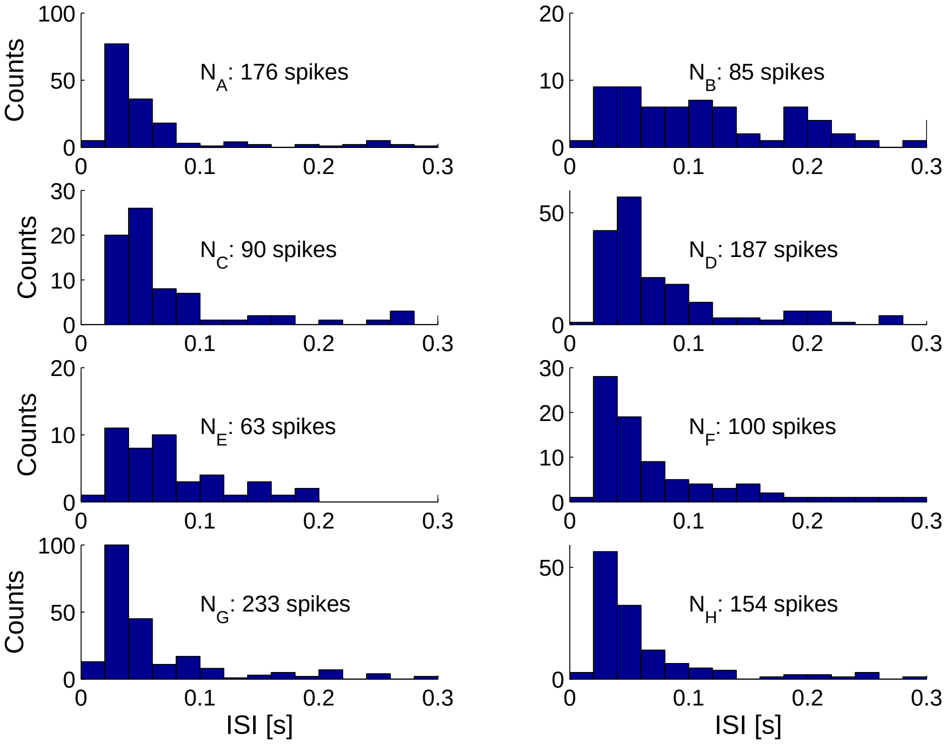


Figure B.3: ISI histogram of the 8 sorted neurons and the respective number of associated spikes. Notice a refractory period of about 20 ms that correspond to the first bin.

ISI smaller than 20 ms are $\sim$ 3.3%, $\sim$ 1.9%, $\sim$ 0%, $\sim$ 0.1%, $\sim$ 2.2%, $\sim$ 1.3%,

∼ 6.3%, and ∼ 2.4% respectively. Since they are all less than 10% the spike sorting is commonly considered as acceptable. Nevertheless, some collection of spikes are not so large to be confident that spikes are indeed from a single neuron.

The code was written in MATLAB programming language and run on 1.73 *GHz* Pentium PC running Microsoft Windows XP. Once that all thresholds have been set, the analysis of 20 seconds from the tetrode recording takes, on this system, about 1.6 seconds for the preliminary step $A0$, about 8.4 seconds for the learning phase, and about 175 seconds for the classification phase (185 seconds in total).

Figure B.4 shows an example of spike classification in a segment of data.

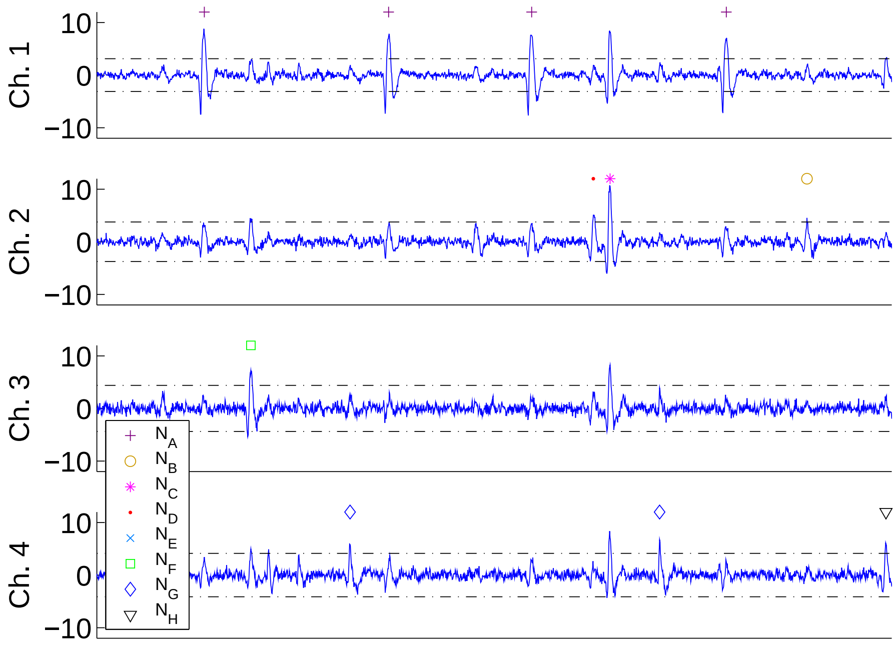


Figure B.4: Spike sorting results on a segment of data from the locust (*Schistocerca americana*) antennal lobe recording. Symbols are in correspondence of arrival times of classified spikes (in the channel where the spike is more visible). The horizontal lines represent the detection thresholds $\pm T_d^1$, $\pm T_d^2$, $\pm T_d^3$, and $\pm T_d^4$. The vertical scale is voltage in arbitrary units.

Bibliography

[1] M. A. Abdul-Ghani, T. A. Valiante, and P. S. Pennefather. Sr2+ and quantal events at excitatory synapses between mouse hippocampal neurons in culture. *Journal of Physiology*, 495(1):113–125, August 1996.

[2] M. Abeles and M. H. Goldstein. Multispike train analysis. *Proceedings of the IEEE*, 65(5):762–773, May 1977.

[3] A. F. Atiya. Recognition of multi-unit neural signals. *IEEE Transactions on Biomedical Engineering*, 39(7):723–729, July 1992.

[4] I. N. Bankman, K. O. Johnson, and W. Schneider. Optimal detection, classification, and superposition resolution in neural waveform recordings. *IEEE Transaction on Biomedical Engineering*, 40(8):836–841, August 1993.

[5] A. Ben-Israel and T. N. E. Greville. *Generalized inverses: theory and applications.* John Wiley and Sons, New York, 1974.

[6] H. Bergman and M. R. DeLong. A personal computer-based spike detector and sorter: implementation and evaluation. *Journal of Neuroscience Methods*, 41(3):187–97, March 1992.

[7] D. S. Bernstein. *Matrix mathematics: theory, facts, and formulas with application to linear systems theory.* Princeton University press, Princeton, 2005.

[8] M. Blatt, S. Wiseman, and E. Domany. Superparamagnetic clustering of data. *Physical Review Letters*, 76(18):3251–3254, April 1996.

[9] J. M. Bower, Y.-F. Wong, and J. Banik. Neural networks for template matching: Application to real-time classification of the action potentials of real neurons. In *Neural Information Processing Systems*, pages 103–113. American Institue of Physics, Denver, Colorado, USA, 1987.

[10] E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, May 2004.

[11] G. D. Brown, S. Yamadab, and Sejnowskic T. J. Independent component analysis at the neural cocktail party. *Trends in Neurosciences*, 24(1):54–63, January 2001.

[12] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms: A Primer.* Prentice Hall PTR, Upper Saddle River (NJ), 1998.

[13] G. Buzsáki. Large-scale recording of neuronal ensembles. *Nature Neuroscience*, 7(5):446–451, May 2004.

[14] W. H. Calvin. Some simple spike separation techniques for simultaneously recorded neurons. *Electroencephalography and Clinical Neurophysiology*, 34(1):94–96, January 1973.

[15] J.-F. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, October 1998.

[16] 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 Transactions on Biomedical Engineering*, 44(5):403–412, May 1997.

[17] M. I. Chelaru and M. S. Jog. Spike source localization with tetrodes. *Journal of Neuroscience Methods*, 142(2):305–315, March 2005.

[18] H. Cho, D. Corina, J. F. Brinkley, G. A. Ojemann, and L. G. Shapiro. A new template matching method using variance estimation for spike sorting. *Neural Engineering, 2005. Conference Proceedings. 2nd International IEEE EMBS Conference*, pages 225–228, March 2005.

[19] J. H. Choi, D. Kim, and T. Kim. A new overlapping resolution method for multi-channel spike sorting. In *Conference Proceedings. 2nd International IEEE EMBS Conference on Neural Engineering, 2005*, pages 683–686, March 2005.

[20] S. Choi, A. Cichocki, H. M. Park, and S. Y. Lee. Blind source separation and independent component analysis: A review. *Neural Information Processing - Letters and Reviews*, 6(1):1–57, January 2005.

[21] J. D. Clements and J. M. Bekkers. Detection of spontaneous synaptic events with an optimally scaled template. *Biophysical Journal*, 73(1):220–229, July 1997.

[22] R. R. Coifman, Y. Meyer, and M. V. Wickerhauser. Adapted waveform analysis, wavelet-packets and applications. In Robert E. O'Malley Jr., editor, *ICIAM 91: Proceedings of the 2nd International Conference on Industrial and Applied Mathematics, 8–12 July, 1991*, pages 41–50, Philadelphia, 1992. SIAM, SIAM Press.

[23] R. R. Coifman and M.V. Wickerhauser. Entropy based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 32:712–718, March 1992.

[24] I. Daubechies. *Ten Lectures on Wavelets (C B M S - N S F Regional Conference Series in Applied Mathematics)*. Society for Industrial and Applied Mathematics, Philadephia, December 1992.

[25] M. Delescluse and C. Pouzat. SpikeOMatic Tutorial 2 — SpikeOMatic goes MCMC. http://www.biomedicale.univ-paris5.fr/physcerv/C_Pouzat/Code_folder/SpikeOMaticTutorial2.pdf, July 2005.

[26] M. Delescluse and C. Pouzat. Efficient spike-sorting of multi-state neurons using inter-spike intervals information. *Journal of Neuroscience Methods*, 150(1):16–29, January 2006.

[27] E. H. D'Hollander and G. A. Orban. Spike recognition and on-line classification by unsupervised learning system. *IEEE Transaction on Biomedical Engineering*, 26(5):279–284, May 1979.

[28] J. C. Dill, P. C. Lockemann, and K. Nab. An attempt to analyze multi-unit recordings. *Electroencephalography and Clinical Neurophysiology*, 28(1):79–82, January 1970.

[29] G. J. Dinning and A. C. Sanderson. Real-time classification of multiunit neural signals using reduced feature sets. *IEEE Transaction on Biomedical Engineering*, 28(12):804–812, December 1981.

[30] K. L. Drake, K. D. Wise, J. Farraye, D. J. Anderson, and S. L. Bement. Performance of planar multisite microprobes in recording extracellular single-unit intracortical activity. *IEEE Transactions on Biomedical Engineering*, 35(9):719–732, September 1988.

[31] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.*, chapter 6, pages 217–219. John Wiley and Sons, New York, 1973.

[32] 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. *Journal of Neuroscience Methods*, 69(2):175–188, November 1996.

[33] M. S. Fee, P. P. Mitra, and D. Kleinfeld. Variability of extracellular spike waveforms of cortical neurons. *Journal of Neurophysiology*, 76(6):3823–3833, December 1996.

[34] G. L. Gerstein, M. J. Bloom, I. E. Espinosa, S. Evanczuk, and M. R. Turner. Design of a laboratory for multineuron studies. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):668–677, 1983.

[35] G. L. Gerstein and W. A. Clark. Simultaneous studies of firing patterns in several neurons. *Science*, 143(3612):1325–1327, March 1964.

[36] E. M. Glaser. Separation of neuronal activity by waveform analysis. *Advances in Biomedical Engineering*, 1:77–136, 1971.

[37] E. M. Glaser and W. B. Marks. On-line separation of interleaved pulse sequences. *Data Acquisition and Processing in Biology and Medicine*, 5:137–156, 1968.

[38] C. Gold, D. A. Henze, C. Koch, and G. Buzsáki. On the origin of the extracellular action potential waveform: A modeling study. *Journal of Neurophysiology*, 95(5):3113–3128, February 2006.

[39] 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. *Journal of Neuroscience Methods*, 63(1):43–54, December 1995.

[40] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsáki. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of Neurophysiology*, 84(1):401–414, July 2000.

[41] E. Hulata, R. Segev, and E. Ben-Jacob. A method for spike sorting and detection based on wavelet packets and shannon's mutual information. *Journal of Neuroscience Methods*, 117(1):1–12, May 2002.

[42] M. S. Jog, C. I. Connolly, Y. Kubota, D. R. Iyengar, L. Garrido, R. Harlan, and Graybiel A. M. Tetrode technology: advances in implantable hardware, neuroimaging, and data analysis techniques. *Journal of Neuroscience Methods*, 117(2):141–152, June 2002.

[43] K. H. Kim and S. J. Kim. A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio. *IEEE Transactions on Biomedical Engineering*, 50(8):999–1011, August 2003.

[44] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[45] M. Laubach. Wavelet-based processing of neuronal spike trains prior to discriminant analysis. *Journal of Neuroscience Methods*, 134(2):159–168, April 204.

[46] J. C. Letelier and P. P. Weber. Spike sorting based on discrete wavelet transform coefficients. *Journal of Neuroscience Methods*, 101(2):93–106, September 2000.

[47] M. S. Lewicki. Bayesian modeling and classification of neural signals. *Neural Computation*, 6(5):1005–1030, September 1994.

[48] M. S. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Computation in Neural Systems*, 9(4):R53–R78, November 1998.

[49] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of l2(r). *Transactions of the American Mathematical Society*, 315(1):69–87, September 1989.

[50] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[51] S. G. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, San Diego, 1999.

[52] D.A. McCormick, Connors B.W., Lighthall J.W., and D.A. Prince. Comparative electrophysiology of pyramidal and sparsely spiny stellate neurons of the neocortex. *Journal of Neurophysiology*, 54(4):782–806, 1985.

[53] B. L. McNaughton, J. O'Keefe, and C. A. Barnes. The stereotrode: a new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records. *Journal of Neuroscience Methods*, 8(4):391–397, August 1983.

[54] Y. Meyer. *Ondelettes et Opérateurs.* Hermann, Paris France, 1990.

[55] D. J. Mishelevich. On-line real-time digital computer separation of extracellular neuroelectric signals. *IEEE Transactions on Biomedical Engineering*, 17(2):147–150, April 1970.

[56] Z. Nenadic and J. W. Burdick. Spike detection using the continuous wavelet transform. *IEEE Transaction on Biomedical Engineering*, 52(1):74–87, January 2005.

[57] K. G. Oweiss and D. J. Anderson. Spike sorting: A novel shift and amplitude invariant technique. *Neurocomputing*, 44–46:1133–1139, July 2002.

[58] K. G. Oweiss and D. J. Anderson. Spike superposition resolution in multichannel extracellular neural recordings: a novel approach. In *Conference Proceedings. First International IEEE EMBS Conference on Neural Engineering, 2003*, pages 130–133, March 2003.

[59] K. G. Oweiss and D. J. Anderson. Tracking signal subspace invariance for blind separation and classification of nonorthogonal sources in correlated noise. *EURASIP Journal on Advances in Signal Processing*, 2007.

[60] A. Pavlov, V. A. Makarov, I. Makarova, and F. Panetsos. Sorting of neural spikes: When wavelet based methods outperform principal component analysis. *Natural Computing*, 6(3):269–281, September 2007.

[61] C. Pouzat. SpikeOMatic Tutorial 1. http://www.biomedicale.univ-paris5.fr/ physcerv/ C_Pouzat/ Code_folder/ SpikeOMaticTutorial1.pdf, July 2005.

[62] C. Pouzat, O. Mazor, and G. Laurent. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *Journal of Neuroscience Methods*, 122(1):43–57, December 2002.

[63] V. J. Prochazka and H. H. Kornhuber. On-line multi-unit sorting with resolution of superposition potentials. *Electroencephalography and clinical neurophysiology*, 34(1):91–93, January 1973.

[64] R. Quian Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, 16(8):1661–1687, August 2004.

[65] W. Rall. Electrophysiology of a dendritic neuron model. *Biophysical Journal*, 2:145–167, March 1962.

[66] S. Rebrik, S. Tzonev, and K. D. Miller. Analysis of tetrode recordings in cat visual system. In *CNS '97: Proceedings of the sixth annual conference on Computational Neuroscience: trends in research, 1998*, pages 491–496, New York, NY, USA, 1998. Plenum Press.

[67] D. Rinberg, W. Bialek, H. Davidowitz, and N. Tishby. Spike sorting in the frequency domain with overlap detection. *arXiv:physics/0306056*, June 2003.

[68] D. Rinberg, H. Davidowitz, and N. Tishby. Multi-electrode spike sorting by clustering transfer functions. In *Proceedings of the 1998 conference on Advances in Neural Information Processing Systems II*, pages 146–152, Cambridge, MA, USA, 1999. MIT Press.

[69] M. Sahani. *Latent variable models for neural data analysis.* PhD thesis, Pasadena, CA, USA, 1999.

[70] T. Sato, T. Suzuki, and K. Mabuchi. Fast automatic template matching for spike sorting based on davies-bouldin validation indices. *Engineering in Medicine and Biology Society, 2007. 29th Annual International Conference of the IEEE*, pages 3200–3203, August 2007.

[71] E. M. Schmidt. An instrument for separation of multiple unit neuro-electric signals. *IEEE Transactions on Biomedical Engineering*, 18(2):155–157, March 1971.

[72] E. M. Schmidt. Computer separation of multi-unit neuroelectric data: a review. *Journal of Neuroscience Methods*, 12(2):95–111, December 1984.

[73] W. Simon. The real time sorting of neuro-electric action potentials in multiple unit studies. *Electroencephalography and Clinical Neurophysiology*, 18:192–195, February 1965.

[74] S. Takahashi, Y. Anzai, and Y. Sakurai. Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes. *Journal of Neurophysiology*, 89(4):2245–2258, April 2003.

[75] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding.* Prentice-Hall, Englewood Cliffs (NJ), 1995.

[76] B. C. Wheeler and W. J. Heetderks. A comparison of techniques for classification of multiple neural signals. *IEEE Transactions on Biomedical Engineering*, 29(12):752–759, December 1982.

[77] X. Yang and S. A. Shamma. A totally automated system for the detection and classification of neural spikes. *IEEE Transaction on Biomedical Engineering*, 35(10):806–816, October 1988.

[78] P.-M. Zhang, J.-Y. Wu, Y. Zhou, P.-J. Liang, and J.-Q. Yuan. Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. *Journal of Neuroscience Methods*, 135(1–2):55–65, May 2004.

[79] G. Zouridakis and D. C. Tam. Multi-unit spike discrimination using wavelet transforms. *Computers in Biology and Medicine*, 27(1):9–18, January 1997.

Ringraziamenti

Ora che questa tesi è giunta alla sua conclusione, è il momento di ringraziare le numerosissime persone che direttamente o indirettamente hanno contribuito alla sua realizzazione. Il primo grande, enorme, ringraziamento va alla mia famiglia, a mia madre Antonietta e a mio padre Nino che hanno saputo starmi vicino nei momenti difficili quando l'impresa sembrava impossibile e che in tutti questi tre lunghi e allungati anni non mi hanno mai fatto mancare il loro confortante supporto. E grazie anche a mio fratello Stefano che da lontano ha sempre risposto alle mie richieste di aiuto con lunghissime telefonate intercontinentali.

Tra coloro poi che più direttamente hanno contribuito alla realizzazione di questa tesi devo esprimere la più sincera gratitudine al mio supervisore Prof. Alessandro Beghi per i suoi preziosi suggerimenti, le discussioni, tutto il lavoro di supervisione, e soprattutto per gli incoraggiamenti e la fiducia accordatami in questi tre anni. Ringrazio anche i Proff. Giorgio Picci e Gianluigi Pillonetto per le utili discussioni scientifiche.

Al di là di quanto esposto in questa tesi, buona parte del mio dottorato è stata dedicata allo studio di algoritmi per la rivelazione delle onde gravitazionali presso i Laboratori Nazionali di Legnaro dell'Istituto Nazionale di Fisica Nucleare. Relativamente a questa parte del dottorato, desidero ringraziare in modo particolare il Dott. Antonello Ortolan per tutto il lavoro svolto insieme, per i suoi incoraggiamenti e la sua comprensione. Ringrazio i Laboratori Nazionali di Legnaro e in primo luogo il Prof. Massimo Cerdonio, direttore dell'esperimento AURIGA, per l'ospitalità. Ringrazio Tito dal Canton per tutte le discussioni su algoritmi e wavelet. Stefano Longo per l'assistenza informatica. Marco Drago per l'amena lettura. Giulio Mazzolo per le chiacchierate in corriera. Jean-Pierre Zendri e Antonella Maraner per pranzi e cene. Livia Conti per i consigli non ascoltati. Luca Taffarello e Gabriele Vedovato per aver sempre mangiato tutto quello che ho portato. Nicola Liguori per aver sposato un brava cuoca. E Michele Bignotto per i suoi incredibili racconti.

Il lavoro descritto in questa tesi ha avuto invece inizio presso l'Ecole Polytechnique Fédérale de Lausanne in Svizzera, sotto la supervisione del Prof. Martin Vetterli, che ringrazio per la sua ospitalità, e del Dott. Ridolfi che mi ha introdotto a questa, per me nuova, tematica. Ringrazio i Proff. Roberto Rinaldo e Riccardo Bernardini per avermi messo in contatto con il mondo svizzero.

Ringrazio il Dr. Christophe Pouzat per aver condiviso i dati delle sue registrazioni multicanale e per la discussione telefonica. Un grazie anche a Luciano Sbaiz per aver messo a mia disposizione il suo accogliente appartamento. Ma soprattutto grazie al collega di ufficio e amico Fritz Menzer per la sua compagnia in terra straniera, la sua generosità e la sua apprezzata competenza culinaria. Un pensiero va anche ai due grandissimi buongustai d'oltreoceano Alexandra e Paco Estrada, agli amici italiani Samuele Bandi e Pietro Azzari, all'amico iraniano Pooyan Manoochehrnia e ai colleghi dell'Audiovisual Communications Laboratory Ivana Jovanovic, Radhakrishna Achanta, Ali Hormati, Christoff Faller, Clement Fredembach, Francisco Pinto e Gunnar Schaefer.

Poiché gran parte del mio dottorato è stata vissuta nell'ufficio padovano, vorrei ringraziare tutti i miei colleghi del Dipartimento di Ingegneria dell'Informazione che si sono avvicendati in questi anni e che hanno condiviso con me gioie e dolori. Partendo dai miei due compagni di avventura Federico Ramponi, a cui devo l'iscrizione al dottorato e Stefano Gamba, a cui sono riconoscente per non aver avuto troppa fretta di finire. A seguire con Maura Pasquotti, custode di mille segreti e correttrice di bozze, Lucia Seno e i picciu-picciu, Federica Garin e le sue mitiche ricette, Francesca Carli, Giulia Ortolan, Andrea Masiero e il suo momento topico, Francesco Ticozzi e il soprannome "Gravitina", Alessandro Saccon e Arianna De Lazzari, Eugenio Cinquemani e Lucia Galasso, Giuseppe Notarstefano e Mariella, Luca Burelli, Ruggero Carli, Alberto Silletti, Mirco Rampazzo, Simone Del Favero, Saverio Bolognani, Mattia Bruschetta, Damiano Varagnolo, e Diego Fiorin. Un ringraziamento speciale va alla segretaria della Scuola di Dottorato, Alessandra Calore, per la sua disponibilità e dolcezza. Infine, allontanandomi un poco dall'ufficio padovano, ma restando sempre in zona, vorrei ringraziare Silvana Cardelle e Piero che mi hanno nutrito di gelati nei giorni in cui non c'era il tempo per un pranzo vero.

Fuori dal mondo accademico desidero ringraziare i miei tre gatti, Pancio, Desi e il coraggioso Nemo, per la loro compagnia nelle lunghe notti di scrittura e nelle calde giornate estive. Ringrazio la zia Elvira per il maglione viola contro il freddo svizzero e per essersi sempre interessata al mio lavoro di dottorato. Un ricordo va anche ad Adriana Breda perché il suo pensiero mi ha accompagnato nel momento più critico della scrittura di questa tesi. Ringrazio le zie venete e siciliane per il supporto morale, le mie cugine Mariangela e Cristiana per gli incoraggiamenti e gli ultimatum di questi ultimi mesi. Un ringraziamento speciale alla zia Francesca e allo zio Pierluigi esperto di virgole e altri segni di punteggiatura.

Infine desidero ringraziare l'infermiere per le sue cure, e tutti gli amici vicini e lontani perché hanno sopportato le mie bizzarrie, le mie nevrosi, le mie sparizioni e nonostante tutto continuano a mostrarmi il loro affetto.

Per concludere ringrazio tutti quelli che mi sono involontariamente dimenticata di ringraziare. Non se ne abbiano a male, ma si considerino comunque inclusi in questo delirio di ringraziamenti.