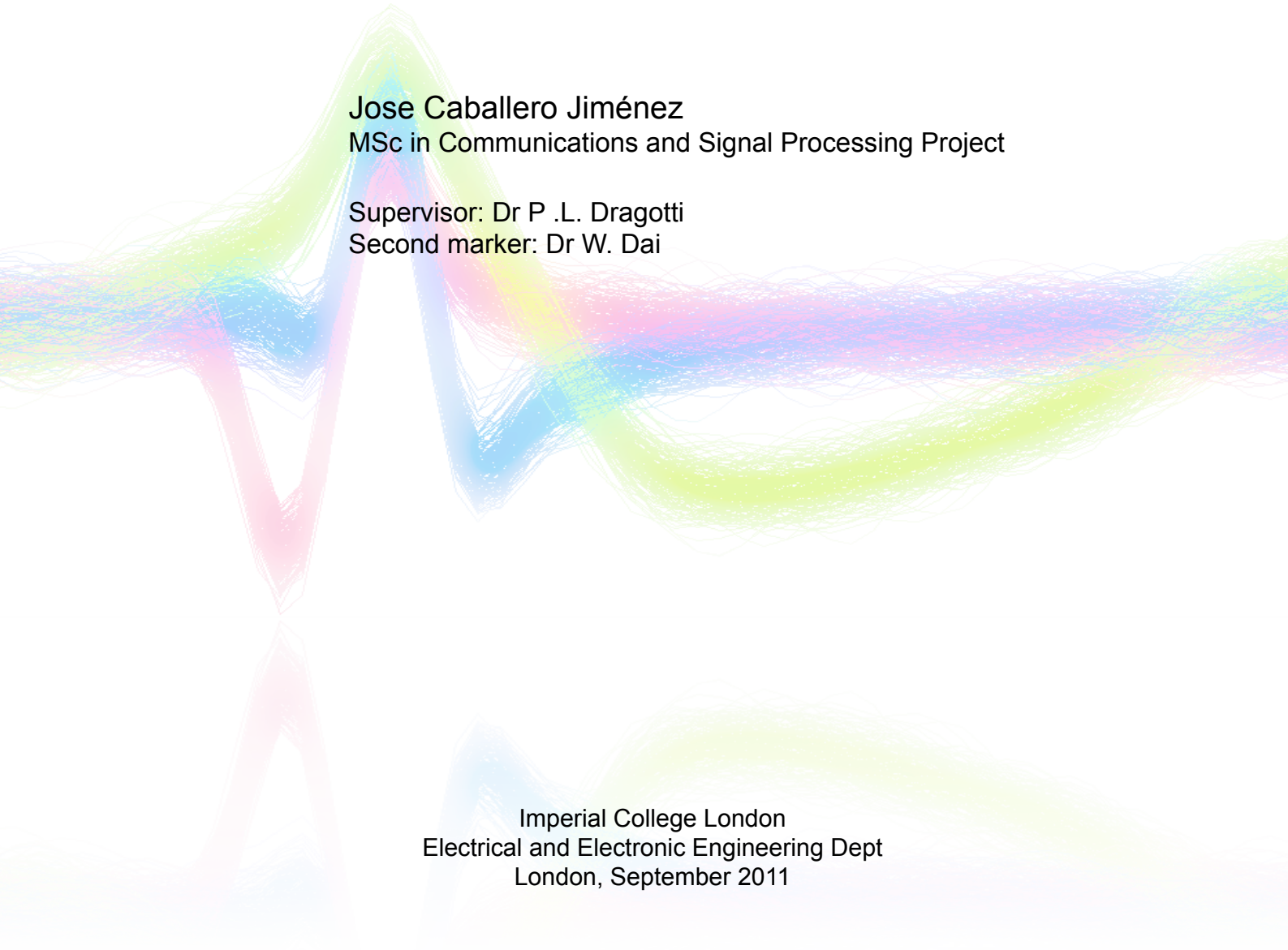
Imperial College London

# Spike Sorting at sub-Nyquist Rates

Jose Caballero Jiménez
MSc in Communications and Signal Processing Project

Supervisor: Dr P .L. Dragotti
Second marker: Dr W. Dai



Imperial College London
Electrical and Electronic Engineering Dept
London, September 2011

# Abstract

Spike sorting can encourage medical and neurophysiological research by providing a tool to investigate how neurons communicate between them. The reliability and robustness of spike sorting algorithms is known to increase if multiple channels monitor the same area of the brain simultaneously. There is a technological constraint in that the Nyquist frequency imposes a high minimum sampling frequency which makes computational complexity and energy demands increase to unacceptable levels for the use of multiple recording channels. In this report, a sparse sampling algorithm based on concepts from FRI sampling and compressed sensing is presented. The data acquisition targets specifically neural recorded activity and relies on the observation that it can be represented parametrically. Frequency reductions of more than 4x the sampling frequency typically used for these purposes are shown to reliably capture characteristic features of spikes driving the sorting process, therefore imitating state-of-the-art spike sorting results obtained with data sampled at above the Nyquist rate.

# Acknowledgment

First and foremost, I would like to thank Dr. Pier Luigi Dragotti for giving me the opportunity to work on this very interesting subject, as well as for his invaluable guidance, mentoring and support throughout the entire project.

I am also extremely grateful to Jose Antonio Urigüen for his expertise in the subject and for helping me countless times. His interest in my progress has undoubtedly been essential for the completion of this work.

Lastly, my special acknowledgments go to my friends, family and in particular to my parents, for their constant support and for teaching me how to always aim higher. Without them, I would not be where I am today.

To all of the above mentioned, I would like to express my deepest gratitude.

# Contents


**1 Introduction** **7**
- 1.1 Objectives and specifications . . . 8
- 1.2 Report outline . . . 9

**2 Spike sorting** **11**
- 2.1 Features of extracellular recordings . . . 12
  - 2.1.1 Spike shapes . . . 12
  - 2.1.2 Noise characteristics . . . 13
- 2.2 Spike sorting methodology . . . 14
- 2.3 Spike sorting algorithms . . . 15
  - 2.3.1 Spike detection and alignment . . . 16
  - 2.3.2 Feature extraction . . . 17
  - 2.3.3 Clustering . . . 18
- 2.4 Limitations and potential improvements . . . 19
  - 2.4.1 Overlapping spikes . . . 19
  - 2.4.2 Bursting neurons . . . 19
  - 2.4.3 Spike alignment . . . 20
  - 2.4.4 Sparse sampling . . . 20
- 2.5 Performance metrics . . . 20
- 2.6 Conclusion . . . 21

**3 Technical background** **22**
- 3.1 FRI sampling . . . 22
  - 3.1.1 Exponential reproducing kernels . . . 24
  - 3.1.2 Sampling streams of Diracs using exponential reproducing kernels . . . 25
  - 3.1.3 Robustness against noise . . . 26
- 3.2 Compressed Sensing . . . 28

3.2.1 $l1$ minimization . . . 29
3.2.2 Basis Pursuit . . . 30
3.3 The wavelet transform . . . 31
3.3.1 Fourier vs wavelet transform . . . 31
3.3.2 The discrete wavelet transform . . . 32

**4 Algorithm design** **35**
4.1 Signal modelling . . . 35
4.2 FRI E-Spline sampling for neural data . . . 38
4.2.1 Signal $d(t)$ information extraction . . . 39
4.2.2 Signal $p(t)$ information extraction . . . 41
4.3 Basis Pursuit for spike shape recovery . . . 43
4.4 Complete algorithm . . . 45
4.5 Conclusion . . . 47

**5 Algorithm results** **48**
5.1 Recovery of a single Dirac results . . . 48
5.1.1 Sampling a known pulse . . . 49
5.1.2 Sampling an unknown pulse . . . 50
5.2 Recovery of pulse shape through Basis Pursuit results . . . 51
5.3 Complete algorithm results . . . 53
5.3.1 Algorithm convergence . . . 54
5.3.2 Estimation robustness to noise . . . 57
5.3.3 Spike sorting at sub-Nyquist rates performance . . . 58

**6 Conclusion and future work** **61**

# List of Figures


2.1 Microwire action potential recording. . . . . 12
2.2 Real neuron action potential recorded (left) and its wavelet decomposition (bottom right) using a quadratic spline (top right). . . . 13
2.3 Typical spike sorting algorithm setup. . . . . 15
2.4 Two action potentials with the same peak amplitude. Window discriminators help classifying spikes. . . . . 16
2.5 Trade-off in spike detection using voltage thresholds. . . . . 17
2.6 Feature extraction from aligned spike shaped (left). It is difficult to recognize three different spikes from the peak amplitude histogram (top right) but the first PCA component score provides clear separation (bottom right). . . . . 18

3.1 Typical sampling setup. Signal $g(t)$ is the continuous-time signal, $h(t)$ the impulse response of the acquisition device and $y_n$ are the samples obtained through the process. . . . . 22
3.2 Complex exponential function reproduction using a general E-Spline of order $P = 30$. The length of the kernel support creates a wide unstable reconstruction region. . . . . 25
3.3 Modified E-Splines for $P = 18$ and $P = 30$. . . . . 28
3.4 Filter bank representation of the DWT. . . . . 33

4.1 Example of the neural data modelling proposed. The pulse shapes of two neurons (left) are convolved with two streams of delta functions to generate signal $g(t)$ (right). . . . . 36
4.2 Example of signal $g(t)$. Any functions $p(t)$ and $d(t)$ can generate it as long as $t_p + t_d = t_g$. . . . . 37
4.3 High level design of the sub-Nyquist sampling algorithm relying on signal decomposition into the convolution of signals $d(t)$ and $p(t)$. 38
4.4 FRI sampling of a known pulse setup. . . . . 40

4.5 Magnitude (top) and phase (bottom) DFT of signal $g(t)$ together with the DFT coefficients which are available through parameter $s_m$. 42
4.6 Complete FRI sampling algorithm proposed for the acquisition of neural data. . . . . 46

5.1 Original signals $d(t)$ and $p(t)$ (left) and original and reconstructed signal $g(t)$ (right). . . . . 49
5.2 Known pulse FRI estimation performance vs SNR. . . . . 50
5.3 Estimation of an unknown pulse. . . . . 51
5.4 Basis Pursuit performance in recovering a 1-sparse vector $\mathbf{w}$ with respect to its location for $\mathbf{F}_r$ (left) and $\mathbf{F}$ (right). . . . . 52
5.5 Basis Pursuit performance vs SNR. . . . . 52
5.6 Reconstruction of signal $g(t)$ using the sampling algorithm proposed. 55
5.7 Evolution of $\tilde{d}(t)$ and $\tilde{p}(t)$. . . . . 55
5.8 Pulses estimated with the FRI sampling method proposed. . . . . 57
5.9 Estimation performance vs SNR. . . . . 57
5.10 Capture of spike features with $P = 10$ (left) and $P = 30$ (right). . 58
5.11 Spike shapes being sorted. Set 1 (left) and set 2 (right). . . . . 60

# Chapter 1

# Introduction

Communication between neurons is carried out by action potentials or spikes propagating as electrical and chemical pulses along the nervous system. Much is still to be learned from the way that neurons are interconnected at a neuron-to-neuron scale for communication purposes as well as at the scale of local brain regions. Cracking the neural code could provide invaluable medical information boosting research on neurological diseases such as epilepsy or Alzheimer's disease and our understanding on the brain's physiological structure. The analysis of brain activity at individual neuron resolutions is called spike sorting and relies on the ability to detect the temporal occurrence of action potentials and to identify a relationship between each one of these and a unique specific neuron.

Spike sorting has been shown to be successful at monitoring a restricted number of neurons. The activity of two or three neurons can be analyzed using a microwire implanted in the brain. However, substantial information in order to analyze communication inside the brain requires the spike sorting problem to be scaled up to monitor tens to hundreds of neurons simultaneously. According to Shannon's theorem [1], the natural morphology of neural spikes, which typically set their maximum frequency at 8 KHz, require sampling rates above 16 KHz. This restricted design requirement poses fundamental problems for simultaneous multichannel spike sorting in terms of energy consumption, computational complexity and hardware demands.

The activity of a neuron can be viewed parametrically as a temporal point process of identical spikes. Furthermore, the firing rate of neurons is by nature very low and action potentials can be shown to be sparse in the wavelet domain. These are conditions that make neural information suitable to modern sampling techniques, such as finite rate of innovation (FRI) sampling or compressed sens-

ing, advocating for an economic acquisition of information. The idea underlying these techniques is that, while classical sampling captures information at high rates and then immediately compresses it throwing away all redundancies in it, modern sampling puts forward the intuitive thinking that if data collected is compressible in some sense, compression should be intrinsically performed already at the sampling stage to avoid the processing of unnecessary information.

While these sampling methods have enjoyed great acceptance for electroencephalography (EEG) [2] and electrocardiography (ECG) [3], its application to high spatial and temporal resolution brain activity recordings needed for spike sorting is very limited in the literature. In [4], a proposal is presented in which compressed sensing is used for the wireless transmission of neural data. On top of developing an acquisition technique, a spike sorting algorithm is developed based on the prior that data is being acquired through compressed sensing. Although satisfactory results are obtained with sampling rates up to 3.2 times lower than classical algorithms, compressed sensing does not make use of the fact that neural information can be parametrically represented as FRI sampling does. This makes a proposal based on compressed sensing suitable to a wider range of signals, but if this information is known a priori, it can reduce minimum sampling rates of the acquisition device.

In [5], FRI sampling using this prior knowledge is adapted for spike sorting purposes achieving much lower sampling frequencies with satisfactory results down to 400 Hz. However, the problem tackled by the authors is very much simplified because the sorting criterion used is simply the peak-to-peak amplitude of action potentials. This clearly reduces the information required for the sorting which in turn imposes a much more relaxed constraint on the minimum sampling rate. Spike sorting based on such simplistic features is known to break down in many scenarios, notably for noisy recordings.

## 1.1 Objectives and specifications

In this project, a novel algorithm for data acquisition at sub-Nyquist rates is presented for the acquisition of neural activity information in spike sorting algorithms. It is motivated by the observation that the activity of each neuron can be represented as a train of Diracs convolved by a distinctive pulse shape, which is on its own sparse in the wavelet domain. In that sense, the rate of innovation in neural activity that represents the degrees of freedom per time unit, can be separated and self-contained in these two different signals. Combining results from FRI sampling and compressed sensing, data can be sampled at a much lower rate

than that imposed by the Nyquist frequency and then reconstructed for further processing by a spike sorting algorithm.

It is important to note that, contrary to previous literature on spike sorting at sub-Nyquist rates, the proposal in this report does not deal with spike sorting itself. Rather than comparing the output of the project to existing solutions, the main objective is to analyze the viability of this novel algorithm as an economic way in terms of energy consumption to sample neural activity recordings while keeping the characteristic features of spike shapes necessary for reliable spike sorting. The success of the project will therefore be based on evidence that the proposed algorithm is capable of

1. reducing considerably the sampling rate of current spike sorting acquisition devices, and

2. reproducing faithfully the distinctive features of the sampled data on which spike sorting algorithms rely.

Comparing the frequency rates at which this novel algorithm can operate with those used in classical spike sorting techniques is enough to determine if the first objective is met. On the other hand, evidence confirming the second objective is much harder to analyze quantitatively. Nevertheless, it is possible to determine if the sub-Nyquist sampled data incorporates all spike features by comparing the sorting performance of an available spike sorting software using data at rates above Nyquist and reconstructed data from samples at low rates. A similar degree of sorting performance would suggest that the algorithm proposed is a valid option.

## 1.2 Report outline

Chapter 2 in this report will introduce the problem of spike sorting. Although the development of a new spike sorting algorithm is not the goal of the project, it is important to analyze available methods and understand their limitations. Of direct relevance is the description of physiological characteristics of neural recordings and the sampling process usually involved. The following chapter will present the background theory on FRI sampling, compressed sensing and wavelet analysis necessary to the comprehension of the algorithm proposed. A detailed description of the sub-Nyquist rates acquisition device designed throughout the project will be provided, introducing at the same time its potential advantages and flaws. Chapter 5 will analyze the performance of this algorithm. Firstly, particular attention will be given to the signal reconstruction accuracy of individual modules

as well as that of the complete algorithm. Then, an experiment will be conducted in order determine the viability of this method as an acquisition device for a spike sorting algorithm. Conclusions drawn from this project as well as potential improvements for the algorithm developed will be summarized in chapter 6.

# Chapter 2

# Spike sorting

The measurement of nerve activity has been at physiological research's disposal for almost a century. The detection and understanding of neural spike activity is however still a challenging problem whose development has been lagging surprisingly far behind capabilities of current acquisition and processing hardware. To a large extent, progress in neuroscience relies on the study of communication patterns between neurons and the interpretation of neural codes. To that effect, activity of individual neurons belonging to a local region of the brain must be isolated from that of nearby neurons and analyzed within the context of a complex brain process.

Recording neural activity is not a technical challenge, but the proximity of neurons in the same local region often makes it difficult to acquire activity from a single neuron in only one channel. Usually, microwires suitable for extracellular recordings are able to reduce the number of detected neurons to 2 or 3 per channel. Spike sorting is the procedure of detecting, characterizing and classifying electrochemical neural impulses based on their shapes. This is often where the challenge lies, since it is common that neurons in the same local area have action potentials of similar shape or that high noise levels distort neural impulses so much that sorting performance falls below acceptable thresholds.

Efficient spike sorting algorithms are available for the study of a few neurons per channel whereas commercially available acquisition systems allow recordings of up to hundreds of channels simultaneously [6]. Therefore, there is the urgent need to be able to scale up spike sorting algorithms for the analysis of large populations of neurons keeping reliable levels of performance. A natural constraint when scaling up these analysis processes is the computational load they entail. The more channels are used for data acquisition, the more information needs to

be processed during the sorting stage. Therefore, it would be ideal to keep sorting performance levels as high as possible while acquiring the minimum information necessary from the continuous time signal.

The following subsections are intended to introduce the motivations behind the development of spike sorting and to describe classical and modern techniques available to this end. State-of-the-art methods will be described and current limitations listed with a particular focus on the scalability issue mentioned above. For a comprehensive review of spike sorting techniques the reader is referred to the surveys in [7-9].

## 2.1 Features of extracellular recordings

### 2.1.1 Spike shapes

Extracellular recordings use one or more electrodes to detect the sum of all electrical responses in the axons of the nerve. Electrodes with a tip of about $1\mu m$ in size will normally record the activity of a single neuron (single-unit recording) while slightly larger tips can record activity from several neurons (mulit-unit recording). The latter is generally what is obtained for spike sorting purposes. Electrical activity navigating from one neuron to another in the vicinity of the electrode is detected by a voltmeter, the result of which is a typical biphase shape of an action potential of about 1mV peak amplitude (see figure 2.1). Each neuron has its own distinctive action potential shape depending on its morphology and on the distance and orientation relative to the tip of the electrode.

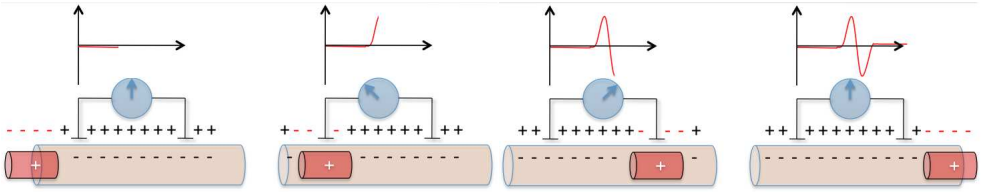


Figure 2.1: Microwire action potential recording.

The firing activity of a neuron can be seen as a temporal Poisson process with a mean firing rate of about 20 Hz. The issue regarding bursting neurons, which for a short time fire at a much higher rate than usual, will later be introduced. It has been empirically shown that spikes have substantial frequency contents up to 8 KHz. Another concept of interest while evaluating sorting performance is

the refractory period of a neuron. This is a physiological characteristic of these type of cells that limits the interfering time between consecutive spikes to about $t_{ref} \geq 2.5$ ms [10].

Central to the development and motivation of the algorithm developed in this project is the idea that neural action potentials have an approximately sparse wavelet transform. The sparsity of the transform is dependent both on the input data and the wavelet family used for the decomposition. The more similar the wavelet family is to the input data, the sparser its wavelet transform. Figure 2.2 shows the shape of a real (noisy) action potential and its wavelet decomposition using the shown quadratic spline. In [4], it is empirically shown that about 1/6 of wavelet transform coefficients of a spike gather around 99% of the signal energy.

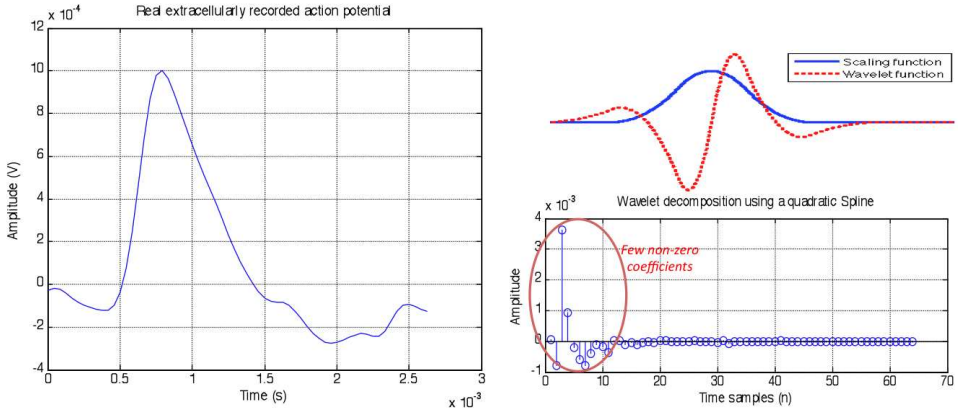



Figure 2.2: Real neuron action potential recorded (left) and its wavelet decomposition (bottom right) using a quadratic spline (top right).

### 2.1.2 Noise characteristics

Many spike sorting algorithms use assumed a priori knowledge on the background noise to improve spike classification. It is common practice to consider that data is corrupted by additive white Gaussian noise. Even though this might be handy because there are Bayesian clustering techniques dealing with this kind of noise, it is not always a good assumption. Noise in extracellular recordings is mostly due to the activity of distant neurons picked up by a recording electrode [11, 12]. Therefore, there is some correlation between the desired signals and the background noise, and the latter is not white but rather shares a similar power spectrum with the spikes.

Furthermore, there are many difficulties in the recording that indicate that clusters' shapes must necessarily deviate from gaussianity, even if original data is corrupted by AWGN. These can be electrodes moving during recording (non-stationary data), spike shapes varying due to bursting, spikes from two or more neurons overlapping or spike misalignment. These phenomena contribute to the elongation of detected clusters, making Bayesian clustering techniques suboptimal for classification. Some of these issues will be addresses in more detail below.

## 2.2 Spike sorting methodology

The range of approaches to tackle the spike sorting problem is large and covers rudimentary manual approaches as well as unsupervised algorithms. Complexity varies a lot depending on the approach taken, but there is not a universal solution for spike sorting yet. As is often the case, the compromise between complexity and performance is dictated by the requirements of a particular experiment.

Despite this diversity, most of the algorithms follow the same fundamental steps and only differ in the choice of their implementation. The general framework is schematically summarized in figure 2.3. Steps of spike sorting algorithms can be grouped into the four blocks shown in the schematic, but it is of fundamental importance to understand the data acquisition process that occurs prior to sorting and is the matter of discussion for this project. The analog filtering is necessary in order to prepare the data for the sorting process and the analog-to-digital conversion determines the computational burden that is passed on to the sorting algorithm.

The acquisition of neural activity is a fairly standardized procedure. Microwires of $C$ electrodes, usually stereotrodes ($C = 2$) or tetrodes ($C = 4$), record the activity of a local region of about a hundred micrometers. After amplification, raw data is bandpass filtered so as to remove underlying low frequency activity and prevent aliasing in the analog-to-digital conversion. The choice of the filter can modify the features of spikes. Consequently, it influences posterior sorting performance, but it is common practice to use for instance a causal filter with cut-off frequencies of about 500 Hz and 6 KHz [6]. Notice that the upper cut-off frequency is set slightly below the maximum frequency in the spike spectrum in order to reduce the noisy appearance of action potentials.

The use of multiple channel recordings, either through the use of various electrodes within a bundle or monitoring the same local area with more than a single bundle, has been shown to increase the robustness of spike sorting algorithms [13]. Until very recently, all spike sorting algorithms were based solely on Nyquist's

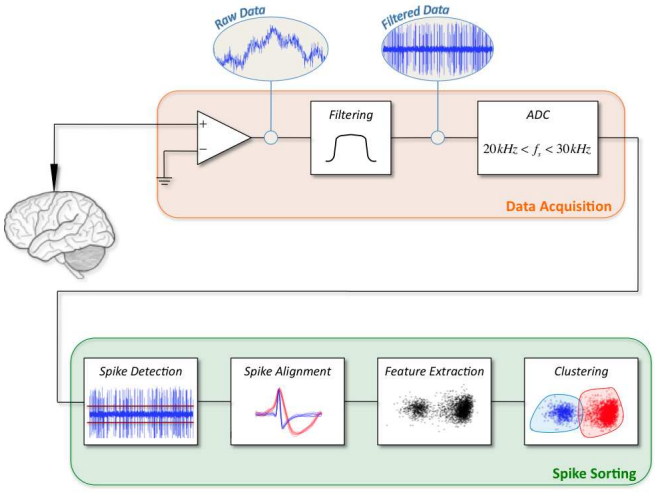



Figure 2.3: Typical spike sorting algorithm setup.

sampling theorem to digitize analog information, which sets 16 KHz as the minimum sampling rate for the analog-to-digital conversion of neural spikes. Typical sampling frequencies range between 20 KHz and 30 KHz, which means that for a common spike length of about 2 ms a minimum of 40 samples are needed to capture its variations in time.

FRI sampling and compressed sensing techniques are increasingly being adapted to extracellular brain activity recordings showing that action potentials can very well be estimated with a number of samples below the Nyquist rate, therefore acquiring the same information in a much more economic way [2, 4, 5]. A successful adjustment to these techniques to spike sorting would imply that a larger set of neurons could be simultaneously monitored, hence improving the robustness of the spike sorting process. The goal of this project is to analyze the suitability of a proposed sparse sampling algorithm to acquire neural data. However, even if the development of a spike sorting algorithm is out of scope, it is also important to understand the steps undergone by sampled data inside them and the principal problems they face.

## 2.3 Spike sorting algorithms

The most straightforward solution for discriminating amongst neural spikes is to classify them based on features available in time domain such as peak or peak-to-

peak amplitude, energy or width. There might be a few cases where sorting based on such basic properties is enough but nothing prevents different neurons to fire action potentials sharing many time domain properties. For instance, in figure 2.4 two different spikes share the same peak amplitude. Template matching [14] and window discriminators can be used in order to reinforce characterization of these basic methods, but have the big disadvantage that every channel of a new experiment has to be supervised by the user. It is therefore not practical when the aim is to analyze hundreds of channels simultaneously.

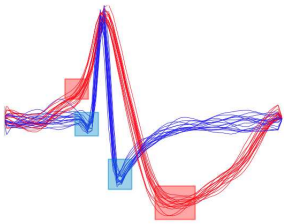


Figure 2.4: Two action potentials with the same peak amplitude. Window discriminators help classifying spikes.

The development of alternative spike sorting algorithms has prioritized three basic properties:

1. Increased reliability and measured performance with respect to basic methods,
2. unsupervised operation in which user intervention should at most be required for final validation, and
3. reasonably fast sorting that can ideally be implemented on-line.

The vast majority of spike sorting algorithms can be structured in three modules.

### 2.3.1 Spike detection and alignment

Spike shapes must be extracted from the filtered recording before they can be analyzed. By far, the most widely implemented technique for spike detection is voltage thresholding. Recorded data is compared with a threshold whose value is proportional to an estimation of the noise level, and any data point crossing the threshold is assumed belonging to a spike. Again, there is not a universal criterion for optimally setting a detection threshold given a data set. In any case, an inevitable compromise exists trading errors caused by unusually high noise instances (low thresholds) or errors due to missed real spikes (high thresholds). This inherent unreliability of threshold detection is shown in figure 2.5.

Detected spikes must be stored and aligned using typically 64 samples per spike so as to advantage from fast transform processing in posterior stages. Alignment of spikes is a crucial step in order to avoid interpreting various instances of a single

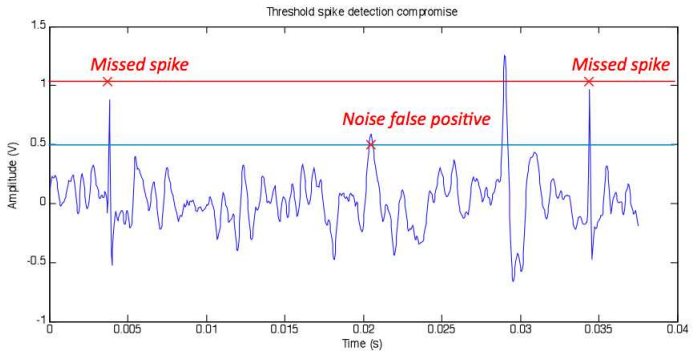



Figure 2.5: Trade-off in spike detection using voltage thresholds.

spike as different spikes. Alignment techniques include oversampling through interpolated waveforms.

### 2.3.2 Feature extraction

All spike sorting algorithms must go through a dimensionality reduction step in which two or three coefficients are assumed to be enough to completely characterize each spike and to determine which ones look alike and which ones do not. Time domain characteristics being discarded for their unreliability, spikes must be analyzed in different domains than the time domain. The main property looked for among features for discrimination is that it presents a multimodal distribution like the ones shown in figure 2.6.

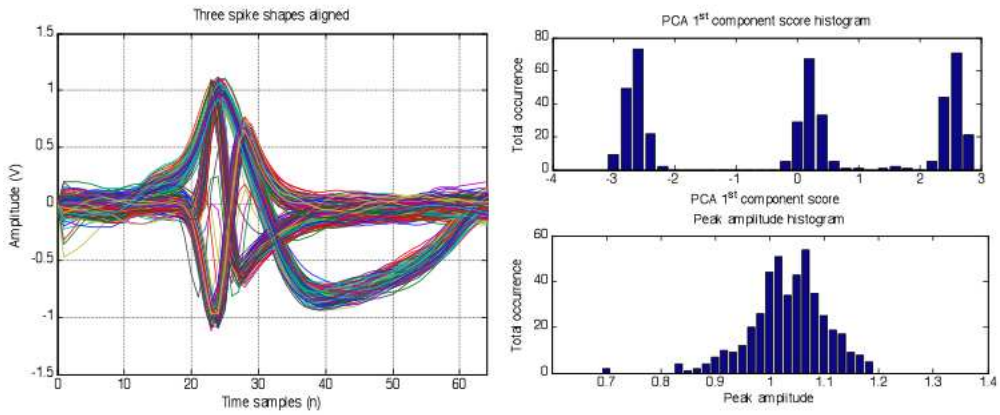



Figure 2.6: Feature extraction from aligned spike shaped (left). It is difficult to recognize three different spikes from the peak amplitude histogram (top right) but the first PCA component score provides clear separation (bottom right).

Available spike sorting algorithms having the best performances normally use either Principal Component Analysis (PCA) or wavelet decomposition for feature extraction. The final results of the report will assess the suitability of the proposed data acquisition method based on PCA feature extraction. This technique transforms a number of variables that are possibly correlated into a reduced number of uncorrelated variables. The orthogonal transformation is based on the criterion that the first component accounts for as much of the variability in the data as possible, while following components successively have the highest variance possible while adhering to the orthogonality constraint.

To a large extent, variations in spike sorting algorithms concentrate on finding new features to efficiently and reliably characterize spike shapes. However, the fact that spikes of more than 40 samples are being classified using only two or three variables usually does not raise any questions. A big dimensionality reduction takes part in all classical algorithms, which suggests that if the distinctive features interesting for the sorting of spikes could already be captured during the sampling, an enormous computational cost would be saved.

### 2.3.3 Clustering

The last step in any spike sorting procedure is the clustering of spikes sharing the same characteristics in the extracted features. Ideally, plotting all spikes features extracted in either two or three dimensions clearly produces a reduced number of

clusters representing each spike class present in the recording. In practice, noise spreads out clusters so much that they can overlap. Automated solutions also exist for this step of spike sorting. The approach of superparamagnetic clustering (SPC) presented in [6] will be used for the analysis of results in chapter 5. SPC assesses the interactions between nearest neighbors to determine the local density and groups together sets of points in high-density regions.

## 2.4 Limitations and potential improvements

There is a huge room for improvement in order to get to an optimal spike sorting algorithm. In what follows some of the technical challenges encountered are summarized. These will help in the interpretation of final results.

### 2.4.1 Overlapping spikes

Many recordings will experience overlap between spikes from different neurons. If this phenomenon is not accounted for, resulting spikes will most probably be identified as low populated separate clusters. Overlapping spikes contain precious information regarding functional connectivity in local regions of the brain and therefore the need for reliable techniques dealing with this issue can't be sufficiently stressed. Some attempts have been done by linearly superimposing representative shapes of identified clusters or by using search trees [7], although these are far from being optimal.

### 2.4.2 Bursting neurons

The assumption that spike shapes are stationary is usually made in spike sorting algorithms. This is generally the case in extracellular brain recordings except for bursting cells. These are neurons firing at very high rates for a short period of time. Amplitudes of successive spikes are typically decreasing, creating a smearing or elongation of the spike's cluster. If too much variation is found between spikes, they might be interpreted as action potentials from more than a single neuron. Within the scope of this project, bursting neurons will not be analyzed. Results obtained can be extended to include the possibility of bursting cells using cross-correlograms.

### 2.4.3 Spike alignment

Spike alignment is defined as an algorithm module on its own because it is of crucial importance and usually a challenging step in any spike sorting algorithm. Detected spikes must be aligned before their features are extracted and the performance of the sorting method will vary according to the accuracy of the alignment. If traces are poorly aligned, features of similar shapes will not match and clusters will not be clearly distinguishable. Spikes are normally aligned with respect to their peaks, but due to insufficient sampling, the peak of sampled spikes can be at different positions. One solution is to check adjacent alignments [7]. In the results of this project a more reliable approach will be used, which consists in oversampling spike shapes using interpolated waveforms, aligning them to their peaks and then decimating the resulting spikes back to their original sampling frequency [6].

### 2.4.4 Sparse sampling

One of the most plausible improvements for spike sorting algorithms is in their data acquisition methods, due to the natural suitability of extracellular recorded data towards sparse acquisition methods. It is enough to know the spike shape from a given neuron and the locations of occurrence (with their individual amplitudes at most if non-stationary data is considered) in order to completely define the activity of a given neuron. If an acquisition process is designed to look for this information already at the sampling stage instead of going through a classical Nyquist sampling process, acquisition could be simplified and made more economic. Moreover, the huge dimensionality reduction of data needed for feature extraction makes it also intuitive that distinctive features could probably be extracted already during sampling. That is, sampling could be interpreted as a useful tool for ad-hoc data analysis instead of only as a means to represent continuous-time signals as faithfully as possible. Wireless recording electrodes for instance could then transmit information at much lower rates to a local processing machine, reducing energy consumption.

## 2.5 Performance metrics

One of the main difficulties while assessing spike sorting performance is that access to a ground truth for sorting is very rare. As a consequence, clustering results are often subjective and rely heavily on the input data. Some algorithms might

perform extremely well for a particular data set while they might do a lot worse for others. This is the reason why researchers try to come up with simulated data that is as close as possible to real action potential recordings and clever tests must be run in order to find possible flaws in spike sorting algorithms.

It is common practice to use a priori knowledge from simulated data and to quantify the error rate therein [15, 16, 6, 7]. Spikes can be undetected, due to a missed spike or due to overlapping spikes, incorrectly classified or left unclassified. The presentation of performance evaluation varies depending on authors. In this report, errors will be classified as

1. Type I: False positive and unclassified spikes. These are action potentials that have been attributed to the wrong cluster or do not share enough similarities to belong to one of them.

2. Type II: False negative spikes. These are action potentials that have been missed in the detection phase.

Typical results for state-of-the-art spike sorting algorithms are about 90% correctly classified spikes for low noise values.

## 2.6 Conclusion

An overview of spike sorting methods has been provided in this section. The motivation behind the urge for efficient spike sorting solutions has been presented and can be summarized as a stepping stone towards the understanding of functional connectivity between individual neurons and local regions of the brain. Physiological properties of extracellular brain recordings, which is the data that will be dealt with throughout this project, has been analyzed and the general framework of all spike sorting algorithms has been examined.

Particular attention has been given to limitations of state-of-the-art techniques which are still far from being optimally addressed. The issue of particular interest within the scope of this project is the application of sparse data acquisition techniques to extracellular brain recordings. These techniques are particularly well suited for this type of recording not only because of the sparse nature of neural action potentials, but also because of the unavoidable and significant dimensionality reduction that they undergo in all available proposals.

# Chapter 3

# Technical background

## 3.1 FRI sampling

Sampling theory is at the heart of any modern signal processing and communications process. It is the essential tool through which continuous-time signals can be acquired, processed, transmitted and stored efficiently. Any sampling scheme can generally be visualized using the diagram in figure 3.1. A continuous time signal $g(t)$ is filtered with $h(t)$ and then sampled at a frequency $f_s = 1/T$. The filtering step may be an inherent characteristic of the acquisition device (such as a camera lens) or a design choice. The resulting samples $y_n$ are given by

$$y_n = \langle g(t), \varphi(t/T - n) \rangle \,. \tag{3.1}$$

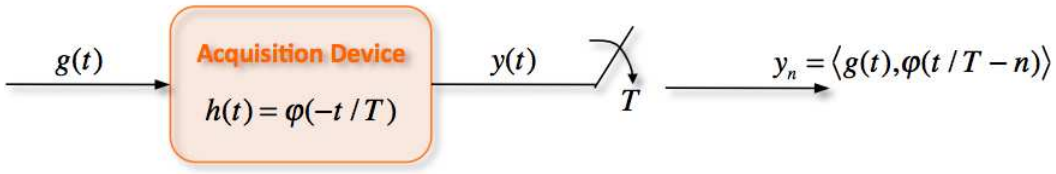



Figure 3.1: Typical sampling setup. Signal $g(t)$ is the continuous-time signal, $h(t)$ the impulse response of the acquisition device and $y_n$ are the samples obtained through the process.

The whole process can be interpreted as a projection of the continuous-time signal onto a set of bases determined by the filter $h(t)$, and hence, only the terms in $g(t)$ found on this set of bases can be recovered, the rest being regarded as

noise. It is sensible to think that the degrees of freedom of signal $g(t)$ must be limited in some way in order to be able to recover the signal without error using a finite set of bases. Questions that rise naturally when addressing this problem are [17]

1. What kind of signals $g(t)$ can be reconstructed?
2. What are optimal kernels $\varphi(t)$ allowing this reconstruction?
3. How should the reconstruction algorithms be designed?

Until recently, Shannon's sampling theorem [1] was considered to be the golden rule in classical sampling. It states that using the sinc function as kernel $\varphi(t)$, any bandlimited signal $g(t)$ can be reconstructed through a linear reconstruction process as long as the sampling rate $f_s$ satisfies $f_s \geq f_{max}$, where $f_{max}$ is the highest frequency component in $g(t)$. In the past decade, results have demonstrated how certain classes of non-bandlimited signals and which do not belong to a fixed subspace, can also be discretized and perfectly reconstructed using non-linear reconstruction algorithms, even though they have been acquired at sub-Nyquist rates.

The peculiarity of these modern sampling methods is that the sampling process is seen from the point of view of signal approximation operation. In that sense, the only condition for perfect recovery is that they possess a parametric representation with limited number of degrees of freedom. Signals satisfying this property are called signals with Finite Rate of Innovation (FRI) [18]. The definition of local rate of innovation will be preferred within the scope of this project, referring to signals that within a time window can be represented parametrically. Local reconstruction algorithms can then be designed an applied to recover an FRI sampled signal.

The observation motivating the proposed algorithm is that neural data can indeed be represented in parametric form and the rate of innovation within a time window can be separated. As will be explained, this makes neural recordings particularly well suited for sparse sampling as the degrees of freedom in the signal can be estimated separately and the original signal can then be reconstructed using its parametric representation.

Results using the sinc and the Gaussian kernels for FRI sampling appeared in [18] and were later extended in [17] to include the case of sampling kernels with compact support, notably to the case of exponential reproducing kernels such as exponential splines (E-Splines) [19]. One of the major issues of E-Splines is their

instability under noisy conditions and this was addressed in [20], where robustness to noise is taken into account in the design of modified E-Splines.

### 3.1.1 Exponential reproducing kernels

Depending on the properties of the kernel $\varphi(t)$ in the sampling process of figure 3.1, different sampling kernels can be designed which are appropriate for the sampling of particular types of signals $g(t)$. Exponential reproducing kernels encompass one class of these kernels.

An exponential reproducing kernel is any kernel $\varphi(t)$ that together with its shifted versions scaled accordingly is able to reproduce complex exponentials of the form $e^{\alpha_m}$ with $\alpha_m = \alpha_0 + m\lambda$ and $m = 0, 1, ..., P$ [17]. This is

$$\sum_{n\in\mathbb{Z}} c_{m,n}\varphi(t-n) = e^{\alpha_m t}, \tag{3.2}$$

for a appropriately chosen coefficients $c_{m,n}$. The parameter $P$ determines the maximum order of the exponentials that the kernel can reproduce and its choice will depend on the local rate of innovation of the sampled signal.

Coefficients $c_{m,n}$ can be found from

$$c_{m,n} = \int_{-\infty}^{\infty} e^{\alpha_m t}\tilde{\varphi}(t-n)dt, \tag{3.3}$$

where $\tilde{\varphi}(t)$ denotes a quasi-biorthogonal set of $\varphi(t)$ [17].

The central element in the theory related to exponential reproducing kernels is the notion of exponential splines [19]. The general E-Spline of order $P$ is the function $\beta_{\vec{\alpha}_P}(t)$ with Fourier transform

$$\hat{\beta}_{\vec{\alpha}_P}(\omega) = \prod_{m=0}^{P} \frac{1 - e^{\alpha_m - j\omega}}{j\omega - \alpha_m}, \tag{3.4}$$

with $\vec{\alpha}_P = (\alpha_0, \alpha_1, ..., \alpha_P)$. Furthermore, any composite function of the form $\gamma * \beta_{\vec{\alpha}_P}(t)$ is also an exponential reproduction kernel, simply due to the fact that the exponential reproduction formula is preserved through convolution [19].

It is useful to point out that, although E-Splines satisfy (3.2), the accuracy of the exponential reproduction is not the same throughout the time window considered. Typically a small region at the right end of the support considered will not represent the exponential correctly and signal features in this kind of

regions will not be well estimated. This comes from the fact that only kernels fully represented within the support contribute to the reproduction of the complex exponential functions. This becomes a problem when $P$ increases, because due to the generation of E-Splines through convolution, the support of kernels expands, thus taking into account fewer shifted versions for the reproduction of exponentials. The region for perfect reproduction becomes smaller as is shown in figure 3.2 for the support $t \in [0, \tau[$. These high orders $P$ can still be used for signal sampling but this phenomenon has to be taken into account when results are being analyzed.

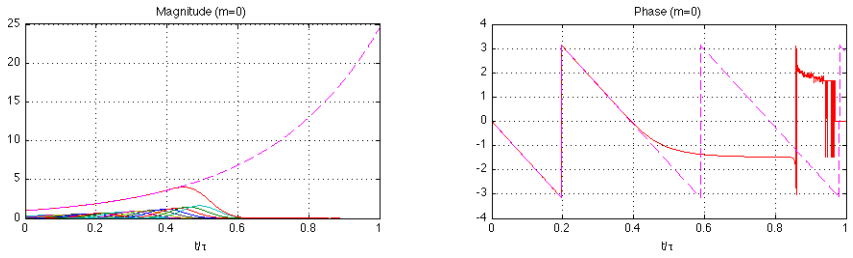



Figure 3.2: Complex exponential function reproduction using a general E-Spline of order $P = 30$. The length of the kernel support creates a wide unstable reconstruction region.

### 3.1.2 Sampling streams of Diracs using exponential reproducing kernels

Let us assume that a stream of $K$ Diracs $g(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k)$, $t_k \in [0, \tau[$, is sampled using an E-Spline of order $P$ to give $N$ samples $y_n = \langle g(t), \varphi(t - n) \rangle$ (where $T = 1$ for simplicity). It is shown here how an algorithm relying on the annihilating filter and a Vandermonde system is able to reconstruct the signal $g(t)$ from only these $N$ samples. Furthermore, a condition relating $P$ to the local rate of innovation of the signal $g(t)$ is given which ensures perfect reconstruction.

The exponential moments of signal $g(t)$ can be found from [17]

$$s_m = \sum_{n \in \mathbb{Z}} c_{m,n} y_n = \sum_{k=0}^{K-1} a_k e^{\alpha_0 + m\lambda t} = \sum_{k=0}^{K-1} \hat{a}_k u_k^m, \tag{3.5}$$

where $\hat{a}_k = a_k e^{\alpha_0 \frac{t_k}{T}}$ and $u_k = e^{\lambda \frac{t_k}{T}}$.

The annihilating filter method, also called Prony's method [21], can be applied to the samples $s_m$ so as retrieve parameters $u_k$. The key aspect of the annihilating

filter, let us denote it by $h_m$, $m = 0, ..., K$, is that it finds its roots at precisely the values of $u_k$. As a result, the time convolution $s_m * h_m$ is zero for the first $K$ elements, or to put in matrix form

$$\mathbf{Sh} = \begin{bmatrix} s_{K-1} & s_{K-2} & \cdots & s_0 \\ s_K & s_{K-1} & \cdots & s_1 \\ \vdots & \vdots & \ddots & \vdots \\ s_{N-1} & s_{N-2} & \cdots & s_{N-K} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix} = - \begin{bmatrix} s_K \\ s_{K+1} \\ \vdots \\ s_N \end{bmatrix}. \tag{3.6}$$

Notice that this system of linear equations requires $P \geq 2K - 1$ for it to have a unique solution. Furthermore, the Toeplitz matrix $\mathbf{S}$ is rank deficient in the noise free case, which is a property that can be exploited to increase robustness to noise when samples are not ideal. Given the values of the parameters $u_k$, the extraction of the Dirac locations $t_k$ is trivial. The amplitudes $a_k$ can then in turn be found using, for instance, the first $K$ consecutive equations in (3.5).

### 3.1.3 Robustness against noise

Noise cannot be avoided in any sampling process. E-Splines are particularly unstable under noisy conditions but proposals to get around this issue have been suggested in [20, 22, 23]. Consider the noisy samples

$$y'_n = y_n + \varepsilon_n, \tag{3.7}$$

where $\varepsilon_n$ is a set of independent and identically distributed (iid) noise values drawn from a white Gaussian distribution. In [20], two different techniques are studied to deal with noise in E-Spline sampling. The first one exploits the rank deficiency of matrix $\mathbf{S}$ in (3.6) in what is known as the Cadzow routine, and the design of what will be denoted here as modified E-Splines is introduced as a framework to build E-Splines which are genuinely resistant to noise.

#### Cadzow routine

The Cadzow routine addresses the problem of denoising from a subspace approach. When noise is considered, matrix $\mathbf{S}$ becomes $\mathbf{S}' = \mathbf{S} + \mathbf{B}$, which suggests that minimizing $\|\mathbf{S}'\mathbf{h}\|_2^2$ under the constraint that $\|\mathbf{h}\|_2^2 = 1$ should yield a good approximation of matrix $\mathbf{S}$ in a typical Total Least Squares (TLS) setup [24][1].

[1] $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$

The potential of this approach can however be notably increased if $\mathbf{S}'$ is denoised before solving this TLS problem. It is known that, in the absence of noise, matrix $\mathbf{S}$ is rank $K$ for $L \geq K$. Matrix $\mathbf{S}'$ can then be forced to be rank $K$ through an SVD approximation in which $\mathbf{S}' = \mathbf{U}\mathbf{\Lambda}\mathbf{V}$, where $\Lambda$ is a diagonal matrix of singular values of $\mathbf{S}'$. Forcing the $L - K$ lowest singular values to zero, a matrix $\tilde{\Lambda}$ can be built, and $\mathbf{S}$ approximated as $\tilde{\mathbf{S}} = \mathbf{U}\tilde{\mathbf{\Lambda}}\mathbf{V}$. This however breaks the Toeplitz nature of matrix $\mathbf{S}$, but this can also be forced by averaging the diagonal elements of $\tilde{\mathbf{S}}$. Again, $\tilde{\mathbf{S}}$ is no longer of rank $K$, and so the entire process can be repeated. This is known as the Cadzow routine and using 10 to 20 iterations can have a considerable impact on signal denoising.

### Modified E-Splines

The previous approach will work as long as the assumption that the additive noise is Gaussian is true. In the case of exponential reproducing kernels however, this assumption cannot be made, and noise is colored since equation (3.5) now becomes

$$s'_m = \sum_{n=0}^{N-1} c_{m,n} y'_n = s_m + \sum_{n=0}^{N-1} c_{m,n} \varepsilon_n, \tag{3.8}$$

and can be rewritten in matrix form as

$$\mathbf{s}' = \mathbf{s} + \mathbf{e}' = \mathbf{C}\mathbf{y} + \mathbf{C}\mathbf{e}, \tag{3.9}$$

where $\mathbf{C}_{m,n} = [c_{m,n}]_{(P+1)\times N}$. The autocorrelation matrix of vector $\mathbf{e}'$ is not proportional to the identity matrix and this makes the SVD separation of $\mathbf{S}'$ unreliable. To overcome this, minimum support modified E-Splines can be designed based upon the idea that if rows of $\mathbf{C}$ are made orthonormal, then $\mathbf{R}_{\mathbf{e}',\mathbf{e}'} = \sigma^2\mathbf{I}$ holds and the subspace approach can be used optimally. Details on the development of modified E-Splines are provided in [20].

Modified E-Splines will be used for the realization of this project exploiting their denoising capabilities. The most important characteristic for the comprehension of algorithmic design choices is that, in order to avoid resulting modified E-Splines to be complex-valued functions, it is necessary to impose

$$\alpha_m = j\omega_m = \begin{cases} j\frac{2\pi}{N}(2m - P) & \text{when } P \text{ is odd} \\ j\frac{\pi}{N}(2m - P) & \text{when } P \text{ is even} \end{cases}. \tag{3.10}$$

Figure 4 shows two examples of modified E-Splines for $P = 18$ and $P = 30$.

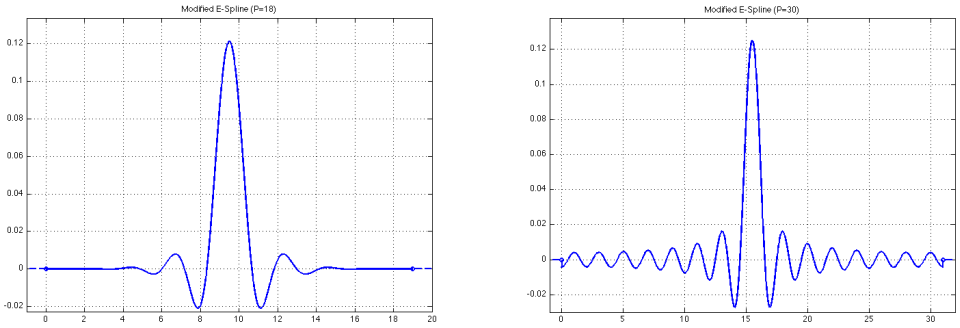



Figure 3.3: Modified E-Splines for $P = 18$ and $P = 30$.

## 3.2 Compressed Sensing

Compressed sensing is a mathematical field that has slowly developed from a theoretical point of view over the last few decades and has recently gained a lot of attention because of its powerful suitability in many engineering disciplines. From a signal processing perspective, it shares many of the objectives of FRI sampling. For instance, both theories try to answer questions such as how many samples are needed to perfectly reconstruct a given signal, or how should the sampling process be performed. The main difference lies in that compressed sensing acts on already existing samples to compress them, whereas FRI sampling creates a direct bridge between continuous and discrete-time signals.

Compressed sensing allows looking at these problems in a purely mathematical setup that can be seen as a revision of linear algebra. Given a wide complex sensing matrix $\mathbf{A} \in \mathbb{C}^{n \times m}$, $n < m$, which is assumed is full-rank, and the observation column vector $\mathbf{b} \in \mathbb{R}^n$, both known a priori such that

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.11}$$

the objective is to perfectly recover the column vector $\mathbf{x} \in \mathbb{C}^m$.

The problem here is that the vector $\mathbf{x}$ is in a way "hidden" within the coefficients of vector $\mathbf{b}$, which is available for processing and is simply the linear combination of the coefficients in vector $\mathbf{x}$ through matrix $\mathbf{A}$. The assumption that this transformation matrix is wide turns the expression in (3.11) into an underdetermined system of equations for an infinite number of solutions $\mathbf{x}$. Geometrically, the solution vector $\mathbf{x}$ can be seen as lying in an $m - n$ dimensional line, plane or a subspace of some kind. A formal description of this phenomenon

is that if $N := N(\mathbf{A})$ is the null space of $\mathbf{A}$ and $\mathbf{x_0}$ is any solution to (3.11), then the set of all solutions to (3.11) $F(\mathbf{y}) := \mathbf{A}^{-1}\mathbf{y}$ is given by $F(\mathbf{y}) := \mathbf{x_0} + N$ [25].

However, a prior on the data structure of vector $\mathbf{x}$ that is often the case in many applications, allows not only to perfectly reconstruct vector $\mathbf{x}$, but also to develop relatively simple algorithms for its recovery. The prior is that the signal $\mathbf{x}$ is sparse, which is equivalent to saying that it is compressible, meaning that its dimensions can be reduced without loosing any original information. The general model case is to consider an $S$-sparse signal , in which only $S$ coefficients are non-zero and $1 \leq S \ll m$. In this particular case, the previously infinite number of solutions for $\mathbf{x}$ reduces to only a few and in some cases to a unique one.

Compressible signals appear in many applications of signal processing. Medical or satellite images for instance are particularly well suited for compression in that they can be well defined by only the position of a few edges. To put a less common example, one could think of a heartbeat signal as repetitive, and hence compressible information.

Candès and Tao introduced in [26] the Restricted Isometry Property (RIP), which is a property that matrix $\mathbf{A}$ needs to ensure that the signal for $\mathbf{x}$ in (3.11) to be resolvable. This allows to define a relationship between parameters $n$, $m$ and $S$ so that if $n \geq cS\log(m/S)$, for $c$ a small constant, then solution $\mathbf{x}$ can be retrieved from $\mathbf{b}$ with high probability [26, 27, 28]. However, checking if the RIP is satisfied for a given matrix is a combinatorial problem, and except in some particular cases involving a random sampling, such as partial random Fourier matrices, the RIP condition must be checked experimentally.

### 3.2.1 $l1$ minimization

The great suitability of sampling at sub-Nyquist rates to many applications has put a lot of pressure on the signal processing community for the development of efficient algorithms for solving (3.11). As was mentioned above, the desired solution, let us denote it by $\mathbf{x}^{\#}$, lies in a translated null space. The aim of an algorithm looking for is to recover the sparsest solution out of all possible ones. This can be described by the following constrained problem

$$\min \ \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.12}$$

Unfortunately, this is an NP-hard problem and only brute force algorithms are known to date that can solve (3.12). The way around this problem, which has been exploited and studied the most in recent literature, considers the $l1$ norm minimization instead of $l0$. It turns out that in many cases, the solution to

$$\min \ \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.13}$$

coincides with that of (3.12).

Although a formal proof is not relevant for the comprehension of the project, it can be useful to try to get an intuition on why this is the case. Norms in different dimensions are geometrically different, but it is generally the case that spiky Euclidean balls of highest norm along dimensionality axes represent $l1$ norms. This minimization is therefore usually a good guess because it is susceptible to hit the solution subspace at a maximum sparsity solution with high probability.

### 3.2.2 Basis Pursuit

The problem of minimization has been tackled from many different perspectives, giving rise to many algorithms such as LASSO [35], Basis Pursuit (BP) [29], Matching Pursuit [36] and Iterated Shrinkage Thresholding [37] amongst others.

In this report, the Basis Pursuit framework for $l1$ minimization is adopted. BP was introduced in [29] as an optimization tool rather than an algorithm. The motivation comes from the signal decomposition problem, in which the objective is to decompose a signal $\mathbf{s} \in \mathbb{R}^m$ with a finite combination of waveforms $(\phi_\gamma)_{\gamma\in\Gamma}$, called atoms, drawn from an overcomplete dictionary $D$. Given that the dictionary of waveforms is known to allow for non-unique representations of $\mathbf{s}$, the sparsest representation of signal $\mathbf{s}$ is sought.

The novelty in BP is the identification of the equivalence between the compressed sensing $l1$ minimization problem with linear programming (LP). The problem in (3.13) can be restated as

$$\min \ \mathbf{c}^T\mathbf{x} \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.14}$$

This is a simple linear programming problem the solution of which can be found by many existing algorithms.

Considering the case of a noisy observation $\mathbf{b}$, a variant of (3.14) can be proposed as an unconstrained LP problem. Assuming the signal to be corrupted by additive Gaussian noise $\mathbf{n} \in \mathbb{C}^n$, the problem in (3.11) becomes $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{n}$. The condition $\mathbf{A}\mathbf{x} = \mathbf{b}$ that was previously taken as a constraint in the optimization problem is now an extra objective, and on top of looking for the sparsest representation of $\mathbf{x}$, the residual $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ is also to be minimized. A way to restate the problem is to consider the minimization of the contribution of both objectives while using a variable $\lambda$ to trade off between them. That is

$$\min\ \lambda \|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 . \tag{3.15}$$

The larger $\lambda$ is, the sparsest the solution found will be. Generally $\lambda$ is a small constant $0 \leq \lambda \leq 1$ [29].

This regularized version of the $l1$ minimization problem can also be restated as an LP in the same way as in the constrained compressed sensing problem, giving the equivalent problem

$$\min\ \mathbf{c}^T \|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{n}\|_2^2 \text{ s.t. } \mathbf{Ax} + \mathbf{n} = \mathbf{b}. \tag{3.16}$$

## 3.3 The wavelet transform

Any signal processing operation involves the retrieval of information from a signal that is not directly available in its current representation. This is done by means of signal transforms, which provide an alternative representation of a given signal by mapping its original dimension space to a different one. The choice of the transform simply depends on the particular purposes of the signal processing operation, which should optimally present particular characteristics of the signal.

### 3.3.1 Fourier vs wavelet transform

It is usually the case that the frequency contents of a signal $x(t)$ observed in the time domain need to be studied, and this link is provided by Fourier series as

$$x(t) = \sum_k a_k \phi_k(t) \tag{3.17}$$

where $\phi_k(t) = \Phi(kt) = e^{jk\omega_0 t}$ and the dilation index $k$ sets the frequency of $\phi_k$ [30]. Amplitude coefficients are given by $a_k = \langle \phi_k(t), x(t) \rangle$.

Fourier analysis however has a fundamental drawback that makes it unsuitable for a certain class of signals. A signal $x(t)$ in time domain exposes its amplitude variation as a function of time but does not reveal any explicit information about its frequency content, whereas its Fourier domain representation $\hat{X}(\omega)$ will show the magnitude and phase variations of the signal as a function of frequency but will not show the temporal characteristics of the signal.

This can be a problem for non-stationary signals for which it would be desirable to analyze its temporal and spectral variations jointly. Although there are

several techniques exploiting Fourier analysis to get around this temporal-spectral information problem, it is physically impossible to obtain complete information of the time and frequency domains at the same time. This effect is known as the Heisenberg uncertainty principle, as a reference to the law in quantum physics stating that it is not possible to measure simultaneously the momentum and the position of a particle.

The wavelet transform provides a trade-off between the resolutions in time and frequency. It does so by analyzing signal $x(t)$ using two compactly supported functions: the basis function and the mother wavelet. Different dilations of these two signals provide a notion of frequency content separation, whereas versions shifted in time keep track of the temporal variations within the signal. This must therefore be expressed as a function of two parameters $i$ and $k$ accounting respectively for frequency resolution and time shift. Signal $x(t)$ can then be decomposed as [30]

$$x(t) = \sum_{i,k} c_{i,k}\psi_{i,k}(t). \tag{3.18}$$

Just as happens with the Fourier series, wavelet functions $\psi_{i,k}$ span the space of finite-energy functions with an infinite orthonormal basis derived from the mother wavelet $\Psi$ as $\psi_{i,k}(t) = \frac{1}{2^{i/2}}\Psi(2^i t - k)$, and amplitude coefficients are found from $c_{i,k} = \langle \psi_{i,k}(t), x(t) \rangle$.

Wavelet analysis is a flexible signal transform in the sense that it does not restrict the choice of the mother wavelet. As long as a function satisfies some general criterions (smoothness, compact support, zero mean, etc) and requisites dependent on the algorithm used [30] it can be eligible as a mother wavelet for signal decomposition.

### 3.3.2 The discrete wavelet transform

Similar to Fourier transforms, the wavelet transform is available in continuous and discrete form. The continuous wavelet transform (CWT) uses an integral and convolution for the computation of wavelet coefficients. The discrete wavelet transform (DWT) on the other hand uses orthonormal or biorthogonal wavelets which are drawn from recursive multiplication of the discrete time signal x with DWT matrices.

The computation of scaling and wavelet functions for the DWT is not relevant for the comprehension of the project, but the way in which frequency information

is distributed among the DWT coefficients plays a major role in the following chapters and in the interpretation of the project's outcome.

The DWT is usually implemented as a filter bank using quadrature mirror filters. Consider the example of a 3 level wavelet decomposition on a vector of length 8. Each decomposition level in figure 3.4 separates frequency contents with a high-pass and a low-pass filter and feeds forward the low frequency contents of the signal to the next level in the filter bank.

Many real life signals, including neural action potentials, gather most of their energy in low frequencies. A finer resolution of the low frequency contents of the input signal is obtained at each level and the spectrum of the signal is left divided as shown in figure 3.4. The advantage due to this spectrum segmentation is that temporal variations have an influence on a limited number of wavelet coefficients (the cone of influence), hence information can be sparsely represented.

The DWT is a linear transform and as such is represented algebraically by the matrix operation

$$\mathbf{w} = \mathbf{W}\mathbf{x} = [c_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}, d_{2,0}d_{2,1}d_{2,2}d_{2,3}]^T , \tag{3.19}$$

where $\mathbf{W}$ is the DWT matrix. Notice that ideal bandpass filters cannot be realized and therefore although there exists a clear correspondence between coefficients in $\mathbf{w}$ with a particular frequency band, information from all frequencies appear to some degree in all coefficients of $\mathbf{w}$. This is very sensible and intuitive, but it is necessary to point it out at this stage because it will become a crucial aspect when interpreting the results of the algorithm proposed.

Fast recursive algorithms exist for the implementation of the DWT. A pyramidal Fast Wavelet Transform (FWT) [31] is used in the project for the computation of forward and inverse wavelet transforms. The recursive nature of the filter bank representation of the DWT is computed by iterative

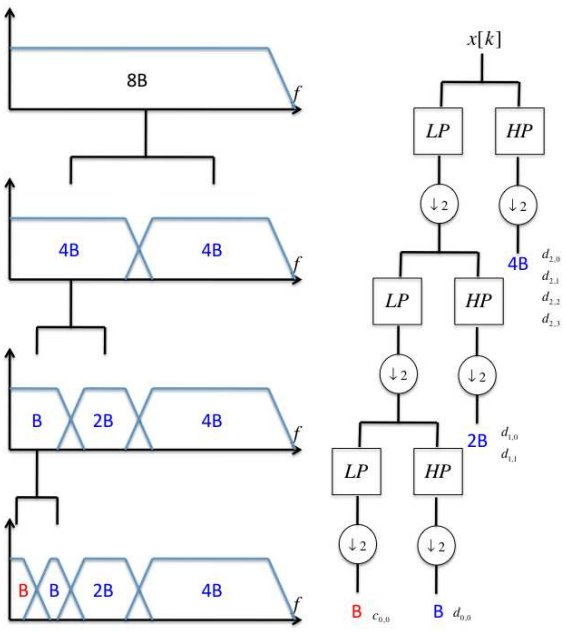



Figure 3.4: Filter bank representation of the DWT.

filtering and downsampling operations
on high-pass and low-pass filter matrices, which are effectively filters LP and HP in figure 3.4. This is nevertheless not a crucial aspect for the understanding of the project.

# Chapter 4

# Algorithm design

The application of modern sampling concepts to neural data has already demonstrated its viability [4, 5] and is simply a matter of time until extracellular recordings can be sampled at their minimum rate. In this chapter, a novel algorithm is presented as a potential solution to this engineering problem. A relatively realistic signal modelling allows to break down the input data into the convolution of two other signals. The problem is transformed into a simultaneous signal and system estimation, where concepts from FRI sampling and compressed sensing take part in the development of an iterative algorithm that permits the sampling of neural data at potentially very low rates.

In what follows, the signal modelling motivating the development of this sub-Nyquist sampling algorithm is described in detail and a high level design of the algorithm is provided. Then, the details of the main modules in the algorithm are analyzed individually. To conclude, the complete algorithm for neural data acquisition at low rate will then be described as a single processing block, giving a chance to foresee advantages and problems arousing in it.

## 4.1 Signal modelling

While studying the characteristics of neural extracellular recordings in chapter 2, it was suggested that the activity of a neuron could be interpreted as a point process in which the same spike shape is fired at different time instances. Assuming no bursting is present in the recorded data and that neural activity is stationary, spike amplitudes will be constant for each of the action potentials from an individual neuron. Based on these priors, the neural signal that is available for sampling with a microwire can be fully represented by the superposition of functions resulting

from the convolution between spike shapes and a train of Diracs. To put this mathematically, if a total of $K$ neurons are active in the vicinity of a recording electrode, firing $J$ spike shapes of peak amplitude $a_k$, $k = 0, ..., K-1$, at time instances $t_{k,j}$, $j = 0, ..., J-1$, the observed continuous-time signal $g(t)$, $t \in \left[-\frac{\tau}{2}, \frac{\tau}{2}\right[$, can be modeled as:

- Neuron $k$ activity:

$$\sum_{j=0}^{J-1} p_k(t) * a_k \delta(t - t_{k,j}) \tag{4.1}$$

- Total $K$ neurons:

$$g(t) = \sum_{k=0}^{K-1} \left[ \sum_{j=0}^{J-1} p_k(t) * a_k \delta(t - t_{k,j}) \right] + n(t) \tag{4.2}$$

where $p_k(t)$ is the characteristic spike shape of neuron $k$ at location $t_{p_k}$ with normalized DC component, and $n(t)$ is a zero mean additive noise signal which does not need to be Gaussian. This interpretation of neural activity data is illustrated in figure 4.1, showing an example where $K = 2$ and $J = 4$.

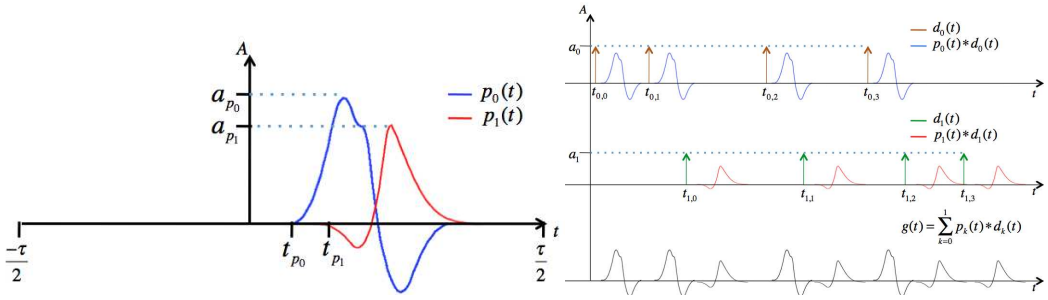



Figure 4.1: Example of the neural data modelling proposed. The pulse shapes of two neurons (left) are convolved with two streams of delta functions to generate signal $g(t)$ (right).

This modelling allows for a clear separation of the innovation rate of signal $g(t)$:

1. Pulse signals $p_k(t)$ capture information about individual spike *shapes* with normalized DC component.

2. Trains of Diracs $d_l(t)$ incorporate information about *time* instances and *amplitude* corresponding to a particular spike shape.

Let us now focus on the simplest case where $K = 1$ and $J = 1$ in order to understand the advantages of this data modelling. Assume the signal $g(t)$ has been generated from $p_0(t)$ convolved with $d_0(t)$ and is observed within a finite length continuous-time support like the one shown in figure 4.2. Adopting the data model introduced in (4.2) and dropping the subscripts $j$ and $k$, there are two degrees of freedom in the representation ($t_p$ and $t_d$) which lead to an infinite number of solutions for the locations of the individual Dirac and the pulse location as long as $t_p + t_d = t_g$.

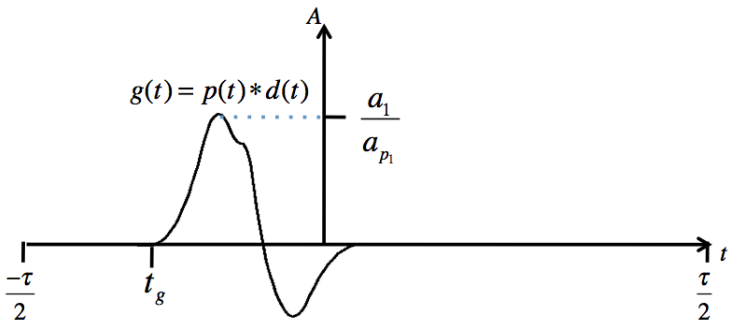



Figure 4.2: Example of signal $g(t)$. Any functions $p(t)$ and $d(t)$ can generate it as long as $t_p + t_d = t_g$.

Which representation is best? An extra condition must be taken into account in the estimation process in order to reduce the number of solutions. It has been discussed in chapter 2 that action potentials can have a sparse wavelet representation for a proper choice of a wavelet family. Since the wavelet domain is a time-frequency domain, the sparsity degree of the pulse $p(t)$ is directly dependent on its location $t_p$. The seek for sparsity in $p(t)$ is crucial for the development of the proposed algorithm as will become clearer later on, and that is why $t_p$ should be chosen in order to provide the sparsest wavelet representation of $p(t)$. This reduces the number of possible solutions for $t_p$ and $t_d$ and should normally create global minimums in the solution subspace from an optimization point of view.

To summarize the previous reasoning, an algorithm is to be designed which will estimate two signals $d(t)$ and $p(t)$ satisfying

1. $g(t) = d(t) * p(t)$,

2. $p(t)$ has unity DC component $\int_{-\tau/2}^{\tau/2} |p(t)|\, dt = 1$, and

3. $p(t)$ has a sparse wavelet decomposition.

The algorithm proposed for sub-Nyquist sampling finds its motivation on this modelling. Two different modules will estimate the signals $d(t)$ and $p(t)$. As schematically represented in figure 4.3, the estimations will use information from each other to iteratively refine themselves. The details of each module are given below.

In real life, it will only be possible to deal with discrete signals $g_l = g(lT_L)$, $p_l = p(lT_L)$ and $d_l = d(lT_L)$, $l = 0, ..., L-1$, at resolution $T_L = \tau/L$. Therefore, the estimation $\tilde{g}_l$ will have to be reconstructed using a discrete convolution.

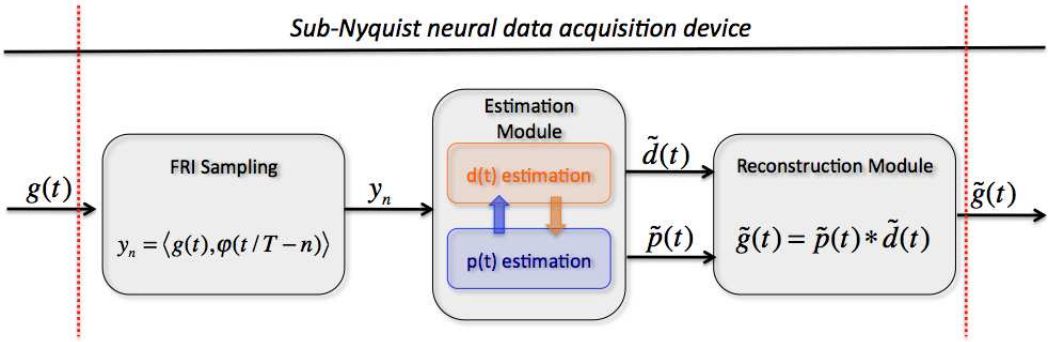



Figure 4.3: High level design of the sub-Nyquist sampling algorithm relying on signal decomposition into the convolution of signals $d(t)$ and $p(t)$.

## 4.2 FRI E-Spline sampling for neural data

Referring back to figure 4.3, the signal $g(t)$ coming directly from continuous-time recorded neural activity is fed to an FRI sampling module. The central operation at this stage is the transformation of continuous-time information into a discrete-time signal $y_n$, $n = 0, ..., N-1$. Exploiting the fact that $g(t)$ is equivalent to the convolution of $p(t)$ and $d(t)$, which are known to have local finite rate of innovation, it is sensible to think that the typical FRI sampling framework will be able to capture enough information for a faithful representation of $g(t)$ at rates which do not need to satisfy the Nyquist criterion.

Consider $g(t)$ being the result of a convolution between a single Dirac signal $d(t) = a_d \delta(t - t_d)$ and a pulse $p(t)$ of DC component unity. Assuming the

continuous-time support to be finite $t_g = t_p + t_d \in \left[-\frac{\tau}{2}, \frac{\tau}{2}\right[$, the samples $y_n$ are generated at a sampling frequency $f_{sN} = 1/T_{sN} = N/\tau$ lower than the Nyquist rate such that

$$y_n = \langle g(t), \varphi(t/T_{sN} - n) \rangle = \sum_{k=0}^{K-1} a_k \varphi(t_k/T_{sN} - n). \tag{4.3}$$

The sampling is performed using a known exponential reproducing kernel $\varphi(t)$ of order $P$, which implies that

$$\sum_n c_{m,n} \varphi(t - n) = e^{\alpha_m t}, \ \alpha_m = \alpha_0 + m\lambda \in \mathbb{C}, \tag{4.4}$$

for a proper choice of coefficients $c_{m,n}$ as was introduced in the previous chapter.

The measurements $y_n$ are the only information that will be available for the estimation of $d(t)$ and $p(t)$. As a consequence, it is crucial to choose the parameters $\alpha_m$ and $P$ carefully, because they largely determine the performance of this data acquisition algorithm as will become clear later on. For the sake of simplicity, let us already define them as

$$\alpha_m = -j\pi \frac{(P - 2m)}{N} \text{ and } P \text{ even}. \tag{4.5}$$

This choice is justified for two reasons:

1. It allows for the use of modified E-Splines as one choice for exponential reproducing kernel.

2. Exponential reproducing kernels with purely imaginary $\alpha_m$ create a powerful link between samples $y_n$ and the Fourier representation of $g(t)$.

### 4.2.1 Signal $d(t)$ information extraction

This subsection shows how the signal $d(t)$ can be estimated from the samples $y_n$ and previous knowledge on $p(t)$. Let us assume that perfect knowledge of the

signal spike shape $p(t)$ is available. Equation (4.3) can be rewritten as

$$\begin{aligned} y_n &= \langle g(t), \varphi(t/T_{sN} - n)\rangle = \langle p(t) * d(t), \varphi(t/T_{sN} - n)\rangle \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (d(\tau)p(t-\tau)d\tau)\,(\varphi^*(t/T_{sN} - n)dt) \\ &= \int_{-\infty}^{\infty} d(\tau)\left\{\int_{-\infty}^{\infty} \varphi^*(t/T_{sN} - n)p(t-\tau)dt\right\}d\tau \\ &= \int_{-\infty}^{\infty} d(\tau)\left\{\varphi^*(\tau/T_{sN} - n) * p(-\tau)dt\right\}d\tau \\ &= \langle d(t), \varphi^*(t/T_{sN} - n) * p(-t)\rangle \\ &= \langle d(t), \varphi_p(t/T_{sN} - n)\rangle\,. \end{aligned} \tag{4.6}$$

Notice that what has effectively been done is transforming the sampling of $g(t)$ into the sampling of $d(t)$, which is a single Dirac and can be retrieved using the FRI sampling as described in chapter 3 and a change of kernel as shown in figure 4.4.

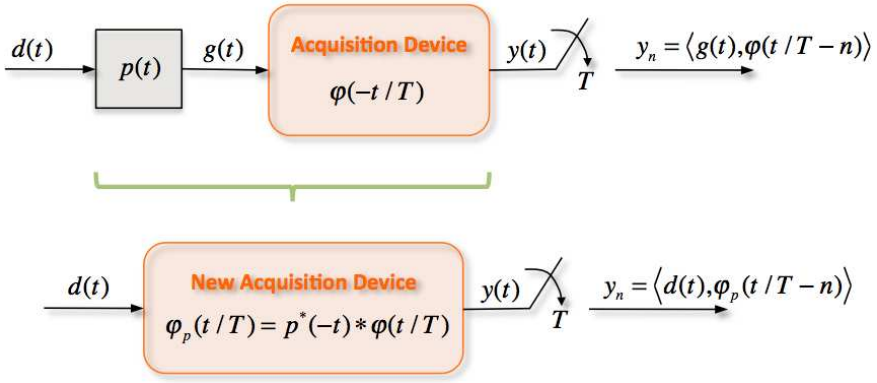



Figure 4.4: FRI sampling of a known pulse setup.

Taking advantage of the fact that any function convolved with an exponential reproducing kernel is another exponential reproducing kernel [19], the new kernel $\varphi_p(t)$ by definition satisfies

$$\sum_{n\in\mathbb{Z}} c^p_{m,n}\varphi_p(t-n) = e^{\alpha_m t}, \tag{4.7}$$

again for a proper choice of a new set of coefficients $c^p_{m,n}$.

The exponential moments $s^p_{m,n}$ of $d(t)$ can then in turn be computed as

$$s^p_m = \sum_n c^p_{m,n} y_n = \left\langle d(t), \sum_n c^p_{m,n} \varphi_p(t/T_{sN} - n) \right\rangle. \tag{4.8}$$

Signal $d(t)$ is known to have local finite rate of innovation because it is a single Dirac. The standard annihilating filter method can therefore directly be applied on coefficients $s^p_m$ in order to retrieve location $t_d$ and amplitude $a_d$, which completely defines signal $d(t)$.

In practice, perfect knowledge of $p(t)$ will never be available. An estimation $\tilde{p}(t)$ of the spike shape will provide estimations $\tilde{t}_d$ and $\tilde{a}_d$ and only an estimation $\tilde{d}(t)$ of the Dirac can be expected. Noise being omnipresent in any sampling process, the Cadzow routine will be used in order to denoise samples $s^p_m$ (20 Cadzow iterations are more than enough) before recovering the Dirac signal through Prony's method.

### 4.2.2 Signal $p(t)$ information extraction

Consider now the opposite case. Namely, assume that the signal $p(t)$ is to be estimated only from the samples $y_n$ and some prior knowledge on $d(t)$. The exponential moments of the signal $g(t)$ can also be retrieved using the original kernel $\varphi(t)$ such that

$$s_m = \sum_n c_{m,n} y_n = \left\langle d(t), \sum_n c_{m,n} \varphi(t/T_{sN} - n) \right\rangle. \tag{4.9}$$

Our choice of the parameter $\alpha_m$ is based on the fact that, on top of offering the construction of noise resistant E-Splines, $\alpha_m$ will be purely imaginary for all $m = 0, ..., P$. Replacing (4.4) in (4.9), it is found that [32]

$$s_m = \left\langle g(t), e^{\alpha_m t} \right\rangle = \int_{-\infty}^{\infty} g(t) e^{\alpha_m t} dt. \tag{4.10}$$

In words, there is an inherent relationship between the samples $y_n$ and the Fourier representation of the signal $g(t)$ attributed only to purely imaginary $\alpha_m$. Notice that $\alpha_m$ only takes discrete values. This implies that through (4.10) it is possible to obtain a finite number of discrete values of the Fourier transform of signal $g(t)$. This result is better interpreted if the DFT of the discrete signal $g_l$,

$l = 0, ..., L-1$, is considered. FRI sampling with a sampling period $T_{sN}$ of $g_l$ is equivalent to undersampling by $N/\tau$, which translates into an equivalent sampling period of signal $g(t)$ of $T' = \frac{\tau}{L}\frac{N}{\tau}$. The DFT of $g(t)$ can then be related to $s_m$ using

$$\begin{aligned} s_m &= \sum_{l=0}^{L-1} g(lT')e^{\alpha_m lT'} = \sum_{l=0}^{L-1} g\left(l\frac{L}{N}\right) e^{\alpha_m \frac{L}{N} l} \\ &= \sum_{l=0}^{L-1} g_l e^{j2\pi\frac{(P-2m)/2}{L}\frac{L}{N}l} = \sum_{l=0}^{L-1} g_l e^{-j2\pi\frac{(P-2m)/2}{N}l} \\ &= \hat{g}_{\left(\frac{P}{2}-m\right)} \end{aligned} \tag{4.11}$$

where $\hat{g}_i = \sum_{l=0}^{L-1} g_l e^{-j2\pi\frac{il}{L}}$, $i = 0, 1, ..., L-1$, is the DFT of $g(t)$ with resolution $1/L$ evaluated at $i$.

Given that $m = 0, 1, ..., P$, it can be noticed that $s_m$ is equivalent to the DFT of $g(t)$ at $i = \left[\frac{P}{2}, \frac{P}{2}-1, ..., -\frac{P}{2}\right]$. To illustrate this relationship, an example of the available DFT coefficients through the parameter $s_m$ is shown in figure 4.5. Therefore, this choice of will always provide the lowest frequency coefficients of the DFT of .

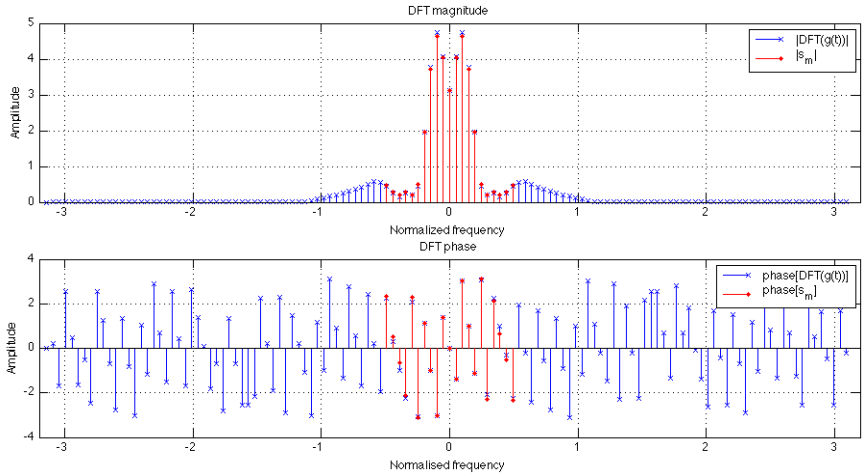



Figure 4.5: Magnitude (top) and phase (bottom) DFT of signal $g(t)$ together with the DFT coefficients which are available through parameter $s_m$.

Assuming perfect knowledge of the Dirac signal $d(t)$, the convolution in time domain translates into a product in the discrete Fourier domain, hence

$$s_m = \hat{g}_{\left(\frac{P}{2}-m\right)} = \hat{p}_{\left(\frac{p}{2}-m\right)}\hat{d}_{\left(\frac{P}{2}-m\right)}, \tag{4.12}$$

and the lowest frequency coefficients of the DFT of $p(t)$ can be computed as

$$\hat{p}_{\left(\frac{P}{2}-m\right)} = \frac{s_m}{\hat{d}_{\left(\frac{P}{2}-m\right)}}. \tag{4.13}$$

A total of $P+1$ coefficients might not seem enough at first glance to completely recover the spike shape $p(t)$, at least from a classical sampling theory perspective. However, as was introduced in the theoretical background section of the report, compressed sensing techniques based on $l1$ minimization can recover a signal from a limited number of samples provided the solution is sparse. It has been assumed that a sparse representation of the pulse $p(t)$ exists in the wavelet domain, so an individual module will be designed to overcome this limitation.

Again, only an estimation $\tilde{d}(t)$ of the Dirac signal will be available, which will lead through this process to the estimation $\tilde{\hat{p}}_{\left(\frac{P}{2}-m\right)}$.

## 4.3 Basis Pursuit for spike shape recovery

The aim of this processing block is to reconstruct the signal $\tilde{p}_l$ from $P+1$ coefficients of its DFT $\tilde{\hat{p}}_{\left(\frac{P}{2}-m\right)}$ using Basis Pursuit. It is assumed that the observed Fourier coefficients represent the DFT coefficients of the true pulse transform plus a residual signal $n_l$. The residual accounts for a zero mean but not necessarily Gaussian noise, which is inevitably picked up during the recording of neural activity, and estimation errors in $\tilde{\hat{p}}_{\left(\frac{P}{2}-m\right)}$.

The problem can be stated in matrix form as

$$\tilde{\hat{\mathbf{p}}} = \mathbf{F}\mathbf{p} + \mathbf{n}, \tag{4.14}$$

where $\tilde{\hat{\mathbf{p}}} = \left[\tilde{\hat{p}}_{P/2}, \tilde{\hat{p}}_{P/2-1}, ..., \tilde{\hat{p}}_{-P/2}\right]^T \in \mathbb{C}^{(P+1)}$, $\mathbf{p} = [p_0, p_1, ..., p_{L-1}]^T \in \mathbb{R}^L$ and $\mathbf{F}_{r,c} = \left[\frac{1}{\sqrt{L}} e^{-j2\pi\frac{(P-r-1)(c-1)}{L}}\right]$ is a wide Fourier transform matrix having $P+1$ rows and $L$ columns. It is known that the pulse shape $\mathbf{p}$ is sparse in the wavelet domain, hence decomposing it as $\mathbf{p} = \mathbf{W}^{-1}\mathbf{w}$ permits the restatement of (4.14) as

$$\tilde{\hat{\mathbf{p}}} = \mathbf{F}\mathbf{W}^{-1}\mathbf{w} + \mathbf{n} = \mathbf{A}\mathbf{w} + \mathbf{n}, \tag{4.15}$$

where $\mathbf{w} \in \mathbb{R}^L$is the wavelet decomposition of $\mathbf{p}$ and $\mathbf{W}^{-1}$ is the inverse wavelet transform matrix.

This transform is fully determined by the wavelet family and levels of decomposition chosen for the decomposition, but what is of most interest here from a mathematical viewpoint is that the vector $\mathbf{w}$ is sparse. Restating the problem into this form means that it is now directly applicable to and solvable through the Basis Pursuit framework.

Notice that matrix $\mathbf{A}$ is not drawn randomly from the dictionary $\mathbf{FW}^{-1}$ but what is imposed by the choice of $\alpha_m$ is rather a very precise structure of the sensing matrix $\mathbf{A}$. Therefore theoretical results based on the RIP condition cannot be extrapolated directly to this case because they rely on a random sensing matrices. The implications of this choice of matrix $\mathbf{A}$ will restrict the potential of this algorithm considerably as is studied in the results section. An important observation however is that the order $P$ of the modified E-Spline used for the FRI sampling is what determines the number of DFT observations available for this Basis Pursuit problem. Hence, a higher order $P$ will naturally lead to a better recovery of the pulse shape $p(t)$.

The residual vector, which has been denoted as $\mathbf{n}$, is not avoidable and therefore only the use of the unconstrained Basis Pursuit problem is justified in order to solve the underdetermined system of linear equations in (4.15). As has been discussed in the technical background of the report, solving it with the prior that $\mathbf{w}$ is as sparse as possible comes down to considering the optimization problem

$$\min \ \lambda \left\|\mathbf{w}\right\|_1 + \frac{1}{2}\left\|\mathbf{A}\mathbf{w} - \tilde{\tilde{\mathbf{p}}}\right\|_2^2. \tag{4.16}$$

That is, it is required to minimize both the sparsity degree of $\mathbf{w}$ and the power of the residual. The weight that each condition has on the result is regularized through parameter $\lambda$, which should be dependent of the noise power. Experimentally $\lambda = 0.5$ has given close to optimal results for all noise conditions tested, so this design choice is already defined.

There are many known algorithms of varying complexity available to solve the linear programme derived in BP constrained. For the purpose of this project, an approach based on a primal-dual log-barrier algorithm introduced in [29] is chosen. It is an interior point method algorithm that has been further developed by Saunders in the available software package Sparselab [33]. The details of this algorithm are not essential to the understanding of this project and can be found in the references provided.

## 4.4 Complete algorithm

It has been shown how $N$ samples obtained through FRI sampling are enough to retrieve all information in the signal $g(t)$. The procedure involves breaking the problem down into two separate signals $d(t)$ and $p(t)$ and estimating them individually. More precisely, subsections 4.2.1 and 4.2.2 respectively detail how $\tilde{d}(t)$ can be calculated using $y_n$ and $\tilde{p}(t)$ and how $\tilde{p}(t)$ can in turn be derived from $y_n$ and $\tilde{d}(t)$. Hence, the estimation of one of these signals is directly dependent on the estimation of the other, suggesting that an iterative algorithm could simultaneously estimate both signals.

The design schematic of the proposed algorithm is shown in figure 4.6. Initialization could either come from an initial guess of $d(t)$ or $p(t)$. It is chosen that a first guess of $p(t)$ be a Dirac located at the centre of the support $\left[-\frac{\tau}{2}, \frac{\tau}{2}\right[$, i.e. the reference point $t = 0$. This implies that in the first iteration, the kernel $\varphi_p(t)$ will be exactly the same as $\varphi(t)$. At this stage, the $d(t)$ estimation module will look for the location and amplitude of a single Dirac when actually the signal being fed to the acquisition device is $g(t)$, i.e. a Dirac convolved with a pulse. The first guess of $d(t)$ will therefore be a lousy guess, but it is enough to obtain a first estimate of the pulse $p(t)$ through coefficients $s_m$ and Basis Pursuit. This information is then in turn used to update the equivalent kernel $\varphi_p(t)$, yielding a new set of coefficients $c^p_{m,n}$ in (4.8). The updated exponential moments $s^p_m$ will be closer to their true value, hence providing a refined estimation of $d(t)$ again through Prony's method.

This process can be iterated until

1. a maximum number of iterations is reached, or

2. convergence to a solution with predefined tolerance is achieved.

Without even testing this algorithm, one crucial design choice can already be foreseen. It is clear that the higher the order $P$ of the modified E-Spline is, the more coefficients of the DFT of $p(t)$ are available in the Basis Pursuit module for the recovery of the pulse $p(t)$. To be more specific, although theoretical results relating the dimensions of the observation vector $P+1$ with the dimensions of the solution sought $L$ cannot be strictly applied because the RIP condition cannot be tested for matrix $\mathbf{A}$, it is simple common sense that more DFT observations necessarily translate into a better $p(t)$ reconstruction. However, it was mentioned in chapter 3 that kernels of high orders can become unstable and provide unreliable FRI sampling. Therefore, parameter $P$ offers a trade-off in the algorithm and

it needs to be explored whether errors in the FRI sampling due to sampling instability can be traded for better compressed sensing results.

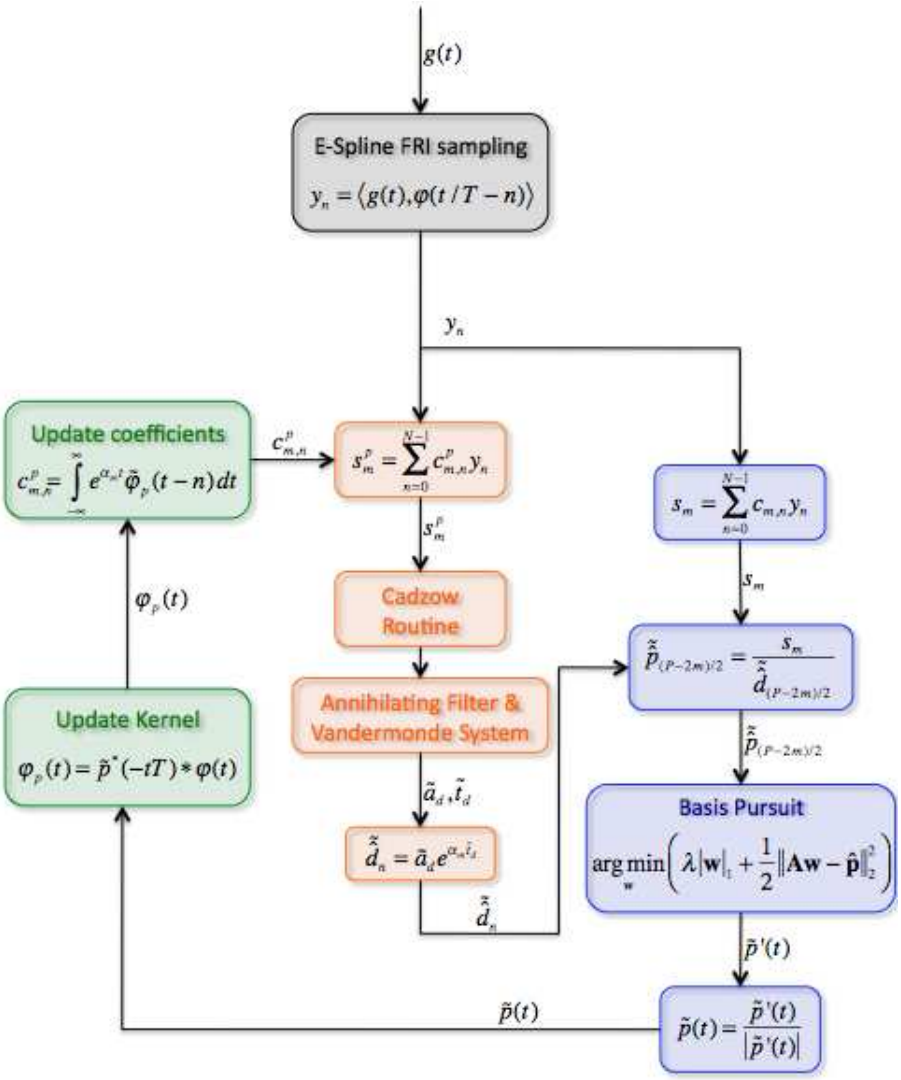



Figure 4.6: Complete FRI sampling algorithm proposed for the acquisition of neural data.

## 4.5 Conclusion

This chapter has described in detail the algorithm proposed for neural data acquisition at sub-Nyquist rates. A signal modelling based on physiological properties of extracellular recordings of neural activity motivates the representation of the continuous time signal $g(t)$ as $d(t) * p(t)$, because it separates the innovation rate in recorded data. This facilitates the use of FRI sampling and Basis Pursuit techniques to sample and recover original data at rates potentially below the Nyquist threshold. The algorithm is broken down into a simultaneous signal and system estimation modules for the recovery of $d(t)$ and $g(t)$.

The intrinsic critical trade-off imposed by the choice of parameter $P$ has been anticipated as one of the most important considerations in the design of the algorithm. While moderate orders of E-splines might lead to a more reliable FRI sampling, the Basis Pursuit module used for the recovery of signal $p(t)$ from just $P + 1$ DFT coefficients suggests that a higher $P$ will translate into a better estimation.

# Chapter 5

# Algorithm results

The performance of the algorithm this report proposes for neural recording sampling at sub-Nyquist frequencies relies on the accuracy of the individual estimations $\tilde{d}(t)$ and $\tilde{p}(t)$, as well as on the convergence behavior of the complete algorithm. It has already been mentioned that the necessary initialization of the algorithm will produce an erroneous guess of signal $d(t)$ from which a good estimation of $p(t)$ is expected stabilizing in this way the iterative pattern of the algorithm. This chapter will analyze the operation of the algorithm and will display general performance results.

First, results of the two most important modules of the algorithm, which estimate the single Dirac signal and the pulse shape, will be analyzed individually as their operation is central to the performance of the estimation algorithm. Then, the entire algorithm will be tested against noise, and particular emphasis will be given to the convergence pattern it exhibits. Based on experiments, the optimal parameter $P$ trading the performance of the algorithm will have to be determined. To conclude, an available spike sorting software will be used in order to compare the performance of a state-of-the-art spike sorting algorithm, for datasets sampled at above and below the Nyquist rate.

## 5.1 Recovery of a single Dirac results

In this section, the performance of the estimation of $d(t)$ is examined as an individual module. Recall that this estimation is very easily obtained if a priori information on the signal $p(t)$ is available. The problem then becomes a simple question of FRI sampling of a train of Diracs. As was introduced in the FRI sampling background, a virtually zero error can be obtained in the noise free scenario

of FRI sampling of a stream of Diracs long as the following conditions hold:

1. The E-spline order $P \geq 2K - 1$, where $K$ is the number of Diracs in $d(t)$.
2. The $K$ Diracs are within the region of the support where $\sum_{n\in\mathbb{Z}} c_{m,n}\varphi(t-n) = e^{\alpha_m t}$ holds.

This applies both to the sampling of a train of Diracs and to the sampling of a train of Diracs convolved with a known pulse. Given that only the case $K = 1$ is being considered, condition 1 will be met for all $P \geq 1$. Condition 2 must be ensured manually.

### 5.1.1 Sampling a known pulse

A signal $g(t)$ defined inside the support $t \in \left[-\frac{\tau}{2}, \frac{\tau}{2}\right[$, is sampled with a period $T = \tau/N$ and reconstructed from samples $y_n$. The FRI sampling is executed with a modified E-Spline of order $P$ as the sampling kernel $\varphi(t)$. In the process, the samples are corrupted by an additive white Gaussian noise of power $\sigma^2$.

Let us assume $N = 31$ and $\tau = 1$. This test assumes that the signal to be sampled is a known pulse convolved with a single Dirac $d(t) = a_k\delta(t - t_k)$. Given that the pulse is known a priori, the local rate of innovation of the signal reduces to that of the single Dirac. Figures 5.1 provide an example in which a rectangular pulse function of length 11/128 is being convolved with the Dirac $d(t) = \delta(t - 0.23)$.

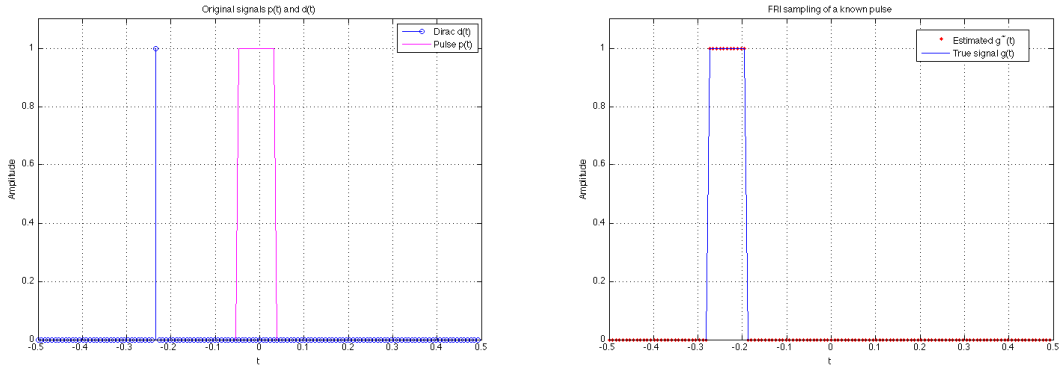



Figure 5.1: Original signals $d(t)$ and $p(t)$ (left) and original and reconstructed signal $g(t)$ (right).

The test is performed using modified E-Splines of different orders and putting into practice the Cadzow routine (see figure 5.2). The plot was obtained by averaging the result of 1000 realizations. A comparison is made with the general E-Spline of order $P = 13$, which is the order that best results gives for this example using classical E-Splines. It is clear that modified E-Splines have a considerable advantage in terms of signal denoising for FRI sampling. These results also suggest that, even though high orders $P$ might become unstable for FRI sampling, as long as conditions 1 and 2 above are satisfied they will provide a better estimation of signal $d(t)$.

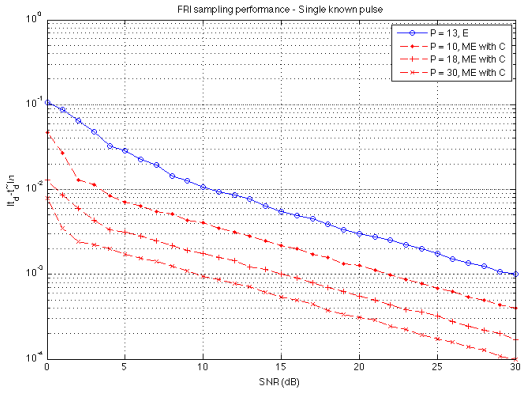



Figure 5.2: Known pulse FRI estimation performance vs SNR.

### 5.1.2 Sampling an unknown pulse

Special attention must be given to the case where an unknown pulse is being sampled and reconstructed as if it was a single Dirac. This is the case of the first iteration in the algorithm, when no information about the pulse shape is available and a first guess on $d(t)$ must be made. Figure 5.3 shows how the algorithm reconstructs a simple signal shape $p(t)$ using a modified E-Spline of order $P = 30$.

Although no formal proof is given in the scope of this report, it can be seen that the algorithm reconstructs the unknown pulse as a Dirac located at the region of roughly highest energy in $g(t)$ with a wrong amplitude estimate. This can be simply explained from the intuition that the algorithm is tricked into interpreting the region of higher energy as the region of highest probability for the location of the Dirac. This is obviously a wrong guess for signal $d(t)$ as was explained that it would be in the design of the algorithm, but allows to launch the algorithm by generating an estimate of $p(t)$. The iterative nature of the algorithm will

then be expected to look for the combination of $p(t)$ and $d(t)$ satisfying the three conditions enumerated in the algorithm design ($g(t) = p(t) * d(t)$ ,$\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} |p(t)|\, dt = 1$ , and sparsest wavelet decomposition of $p(t)$).

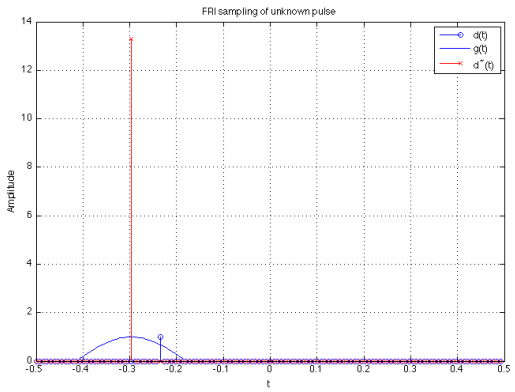



Figure 5.3: Estimation of an unknown pulse.

## 5.2 Recovery of pulse shape through Basis Pursuit results

The Basis Pursuit block is responsible for the estimation of $p(t)$ and is therefore the second essential part of this algorithm together with the module estimating $d(t)$. The refinement of estimation $\tilde{d}(t)$ relies on the capacity of this model to obtain a close estimation of $p(t)$ even with the lousy guess of the Dirac in the first iteration of the system.

Referring back to section 3.3 on the DWT, the matrix $\mathbf{A}$ is restricting the observations to the $P+1$ lowest digital frequencies and by doing so it is excluding any information represented at high frequencies (except a small percentage that will be leaking into the low frequency region due to imperfections in the transform bandpass filters). It is therefore reasonable to think that only the reconstruction of Diracs in the low resolutions of the wavelet decomposition of $\mathbf{p}$ will be possible, for it will be the only ones from which some information will haven been captured in the observation vector $\tilde{\hat{\mathbf{p}}}$.

This phenomenon is illustrated in figure 5.4 using an $L = 128$ samples wavelet decomposition. In both plots, a vector $\mathbf{w}$ of sparsity $S = 1$ is reconstructed from an observation vector $\tilde{\hat{\mathbf{p}}} = \mathbf{A}\mathbf{w}$. The position of the non-zero coefficient is varied

and noise is not taken into account. The only difference in both figures lies in the sensing matrix $\mathbf{A}$. In the left-hand side plot the matrix $\mathbf{A} = \mathbf{F}_r\mathbf{W}$, where $\mathbf{F}_r$ is a partial random Fourier matrix of dimensions $(P+1)\times L$, which is known to satisfy the RIP condition. Results are from an average of 500 realizations of matrix $\mathbf{F}_r$. The right-hand side figure shows results using the sensing matrix obtained for the algorithm proposed. The matrix $\mathbf{W}$ is the wavelet transform matrix related to a 4 level quadratic biorthogonal spline decomposition, giving different resolutions in the sets $[1, 16]$, $[17, 32]$, $[33, 64]$ and $[65, 128]$.

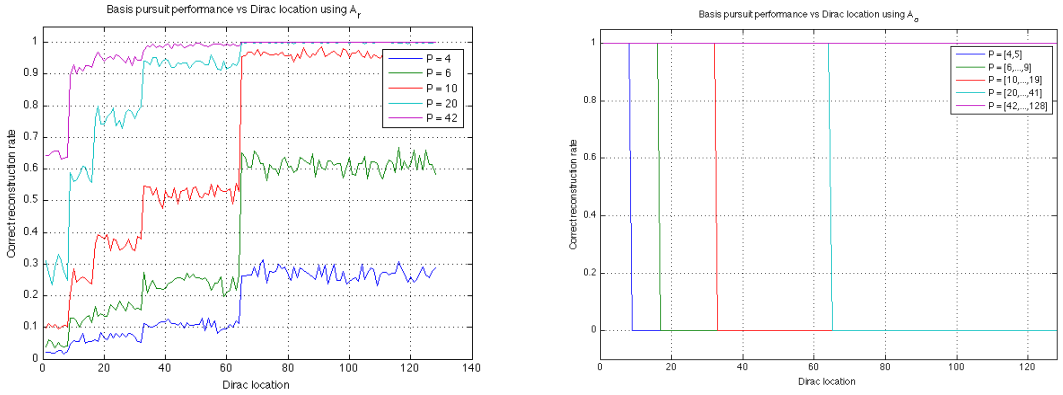



Figure 5.4: Basis Pursuit performance in recovering a 1-sparse vector $\mathbf{w}$ with respect to its location for $\mathbf{F}_r$ (left) and $\mathbf{F}$ (right).

Notice that in both figures the rate of successful recovery is dependent on the location of the non-zero coefficient in the vector $\mathbf{w}$. In the case of $\mathbf{F}_r$, the rate of correct estimation increases with respect to the resolution in which this non-zero coefficient lies although there is always a non-zero probability for perfect recovery in all values of $P$. There is a fundamental difference in the results of $\mathbf{F}$, where the deterministic choice in the rows of produces in a very clear pattern for this dependency, allowing only either 100% or 0% successful recovery in low and high wavelet

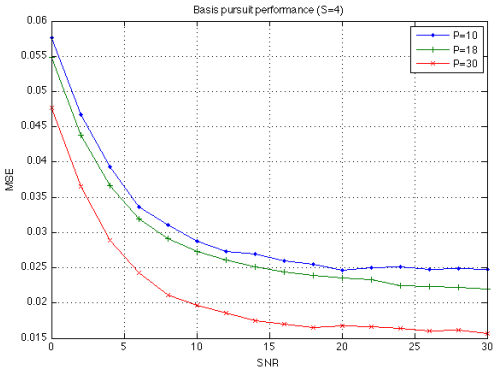



Figure 5.5: Basis Pursuit performance vs SNR.

resolutions respectively. Hence, increasing $P$ as much as possible is a crucial point in the algorithm design, because it allows the reconstruction of non-zero coefficients at high resolutions.

On top of this reasoning, the performance of Basis Pursuit against noise also suggests that high orders $P$ should be favored. Figure 5.5 displays the results of an experiment testing the robustness of Basis Pursuit against noise for the particular case of a vector $\mathbf{w}$ with sparsity $S = 4$. Locations of non-zero coefficients are drawn uniformly at random out of the 128 wavelet coefficients.

At this stage, it is convenient for the analysis of further results to settle with a convenient choice for the parameters $P$, $D$ and $L$, given that they fully determine whether the Basis Pursuit processing block will be working optimally. Keeping in mind that $P$ should be less than or equal to 30 in order to provide a stable FRI sampling, the length of the wavelet decomposition $L$ should be kept relatively small in order to have a chance at reconstructing non-zero coefficients in high resolutions of $\mathbf{w}$. Early on, the choice of a quadratic biorthogonal spline with $D = 4$ was proposed for a wavelet decomposition of length $L = 128$. Based on results for the estimation of $d(t)$ and $p(t)$ it is recommended that $P$ be in the range $[20, 41]$, allowing possible reconstruction for the first 64 wavelet coefficients of $\mathbf{w}$. It is obviously possible to come up with a different combination of these parameters giving rise to different results, but for the rest of this chapter only the recommended choice will be considered for the sake of conciseness.

## 5.3 Complete algorithm results

Three main questions are addressed concerning the performance of the complete algorithm. To begin, it is useful to put into practice the algorithm in the noise free case to see if convergence can be guaranteed, and if so, if converges to the optimal solution is possible. As was discussed in the previous chapter, there are two degrees of freedom in the estimation algorithm ($t_p$ and $t_d$) leading to a large number of solutions to reconstruct signal $g(t)$. It is nevertheless expected that the Basis Pursuit scheme can find the sparsest wavelet representation of $p(t)$ reducing the number of optimal solutions.

Once the convergence pattern of the iterative algorithm has been studied, the quality of the estimation will be assessed as a function of the input noise. The estimation of signal $g(t)$ from a restricted amount of samples will inevitably introduce an estimation error and this section aims at quantifying it. The most interesting results involve applying this estimation algorithm to a spike sorting software. This will determine if this estimation error, which would be introduced

in the data acquisition stage of the spike sorting process, has important repercussions on posterior sorting processing. In this way, it will be possible to see if spike sorting of data reconstructed from samples acquired at low sampling rates is possible or not.

### 5.3.1 Algorithm convergence

In this subsection, the algorithm operation is traced through a simple example in order to have a clear feeling of how it is using information fed back and forward between the two estimation modules. Based on the results deduced from this example, the convergence behavior of the algorithm will be interpreted and suggestions will be given on how to reinforce this convergence. This should not be regarded as a rigorous convergence analysis of an iterative algorithm, but rather as a simple insight on the intrinsic interdependencies between modules of the algorithm. A clear understanding of the issues within the algorithm is indispensable to determine whether the proposed sampling technique is a viable option to operate as the acquisition device of a spike sorting algorithm.

Consider signal $g(t)$ shown in figure 5.6 and the algorithm parameters $P = 20$, $N = 31$ and $L = 128$. This signal $g(t)$ has been generated from signal $d(t) = 0.5\delta(t - 0.039)$ convolved with a pulse $p(t)$ also shown in the figure. Notice that there are only non-zero coefficients in vector $\mathbf{w}$ at locations 3, 4, 11 and 13, and since the transform uses $D = 4$ decomposition levels and the quadratic biorthogonal spline, figure 5.4 shows that these coefficients are within the perfect recovery region in $\mathbf{w}$ for $P = 30$.

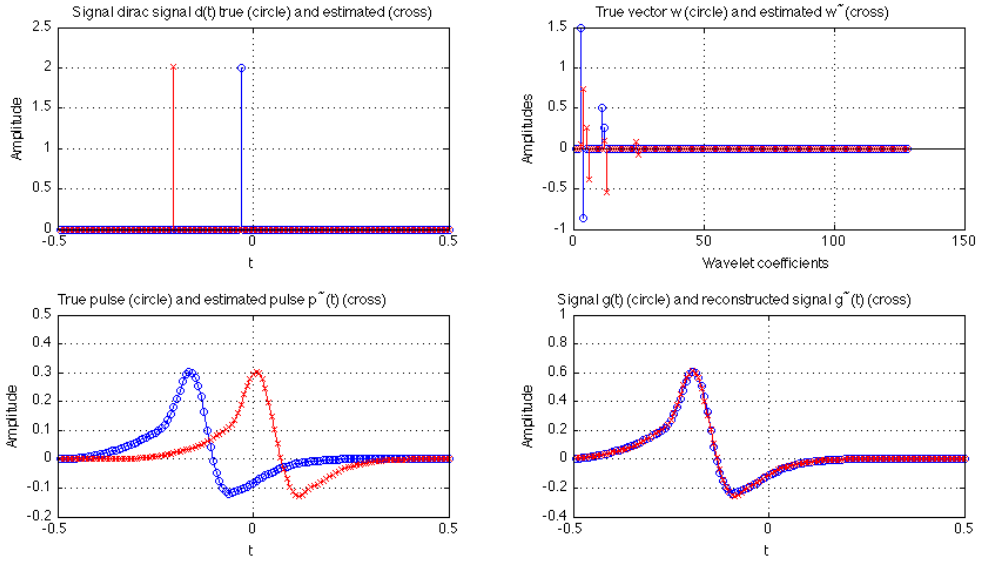



Figure 5.6: Reconstruction of signal $g(t)$ using the sampling algorithm proposed.

The algorithm is stopped after only 4 iterations for a mean square difference between two consecutive iterations tolerance set at $10^{-5}$. Notice nonetheless that the algorithm reconstructs the signal $\tilde{p}(t)$ around the reference point $t = 0$, and even though it is able to obtain a good estimation of the original pulse the wavelet transform of the estimation has a lower sparsity degree than the original.

Figure 5.7 shows both the evolution of $\tilde{d}(t)$ and $\tilde{p}(t)$. It has been seen previously that the location of the first Dirac is roughly the region of highest power in $g(t)$, in this example it is the positive polarity of the action potential giving a first guess $\tilde{t}_d \approx -0.2$. This estimation is then used to obtain a pulse shape $\tilde{p}(t)$ very similar to through Basis Pursuit but located around $t = 0$, which due to the wrong location of $\tilde{d}(t)$ is needed in order to satisfy $g(t) = p(t) * d(t)$. However, even with a fair estimation of the pulse shape, the estimated location of the Dirac does not

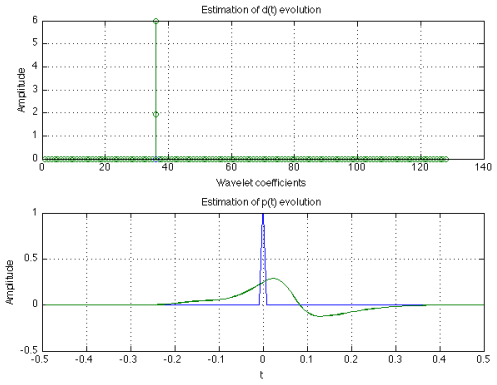



Figure 5.7: Evolution of $\tilde{d}(t)$ and $\tilde{p}(t)$.

change in posterior iterations, which would be necessary in order to allow $\tilde{p}(t)$ to change its position accordingly, and since Basis Pursuit promotes sparsity, it would ideally converge to a position where $\tilde{p}(t)$ has the sparsest possible wavelet decomposition.

Although rigorous analysis of algorithm convergence is out of scope of this report, it is believed that with the information available to the algorithm ($y_n$ FRI samples and $P+1$ observations for Basis Pursuit), the solution subspace has many stable regions when seen as an optimization problem. The solution to which the algorithm converges depends greatly on the location of the pulse that initializes the algorithm given that it will converge to the closest solution available for which $\tilde{p}(t)$ is located the closest to this initial position. The Basis Pursuit block does not seem to be enough to guarantee that the solution found will reconstruct a pulse shape with the sparsest possible wavelet representation.

This suboptimal solution is nevertheless not a big problem because a relatively accurate estimation of $g(t)$ is obtained. The only issue is that the signal $p(t)$ found will rarely have a wavelet decomposition as sparse as the original one. The reason for this is very simple. Because the wavelet domain is a time-frequency domain, if a pulse with sparsest wavelet representation is shifted in time, their wavelet coefficients change and a larger number of coefficients will have to be non zero in order to represent the new pulse. Only a limited number of time shifts will use the same number of non zero coefficients. All pulses located between them will give rise to coefficients at higher resolutions. Therefore, the performances of the Basis Pursuit processing block are diminished since they are proportional to the sparsity degree in $\mathbf{w}$.

The estimated Dirac position $\tilde{t}_d$ can be intentionally perturbed through simulated annealing so that other solutions are visited and a global minimum with respect to the sparsity degree in $\mathbf{w}$ could then be found. However experiments showed that a very small advantage in estimation accuracy can be obtained from this and therefore the extension of the algorithm to incorporate simulated annealing is not justified.

The following figures show more examples of estimated pulses using the algorithm proposed.

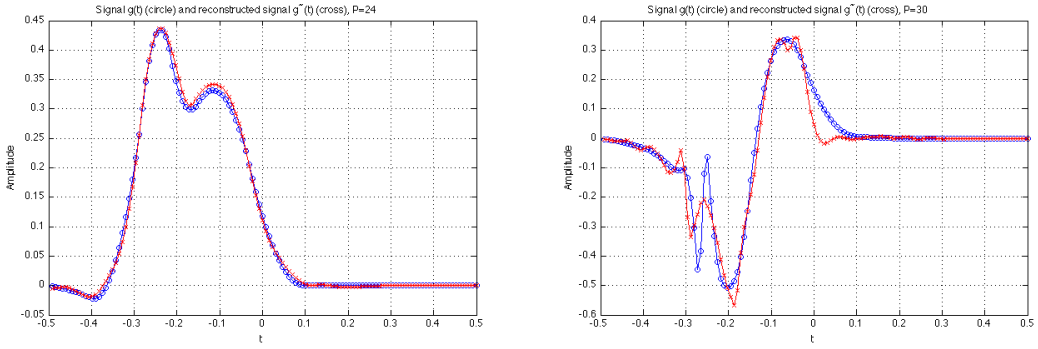



Figure 5.8: Pulses estimated with the FRI sampling method proposed.

### 5.3.2 Estimation robustness to noise

In the previous sections, the accuracy of the estimation modules of $d(t)$ and $p(t)$ has been tested against noise individually. Both coincide in that the highest the order of the modified E-Spline, the better the performance of the estimations. This does not necessarily mean however that the same must be true for the estimation of signal $g(t)$. The algorithm being iterative, there exist inherent interdependencies between successive estimations of $d(t)$ and $p(t)$ that have an impact on the overall performance of the algorithm.

The following experiment uses the same signal $g(t)$ introduced in the previous subsection under different noise conditions and measures the estimation accuracy as the mean square difference of the original and estimated signals.

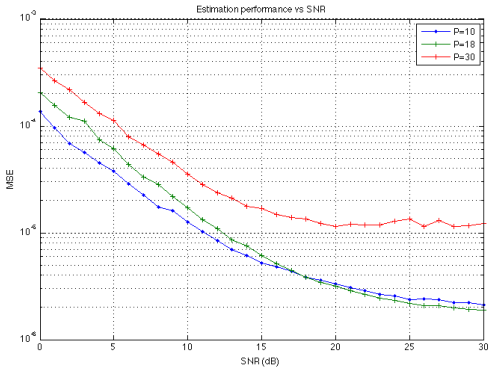



Figure 5.9: Estimation performance vs SNR.

Figure 5.9 shows an advantage for lower values of E-Splines order $P$ even at very low noise values. The reason for this can be explained referring back to the results of $p(t)$ estimation. Small orders of $P$ can reconstruct the shape of $p(t)$ through basis pursuit using only a restricted number of non-zero wavelet coefficients. In particular $P = 10$ and $P = 18$ can recover a pulse using only the first 32 wavelet coefficients out of an $L = 128$ wavelet decomposition. Any

coefficients above will be missed and set to 0 in the reconstruction. For $P = 30$, up to 64 coefficients can be used. This is an advantage in that it is possible to capture high frequency features of the pulse, but can as well be a source of error. This phenomenon is illustrated in figure 5.10, where the same pulse incorporating a non zero coefficient at the high resolution location 40 is being estimated using $P = 10$ and $P = 30$.

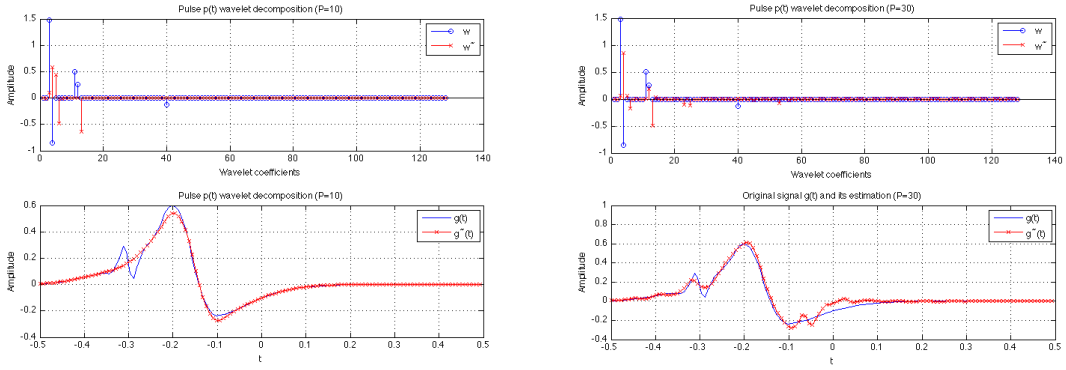



Figure 5.10: Capture of spike features with $P = 10$ (left) and $P = 30$ (right).

Characteristic features of neuron action potentials are needed for reliable spike sorting. These happen to be typically at high frequencies and as Basis Pursuit results have shown, they can only be reconstructed using high values of $P$ within this setup. Therefore, even though high E-Spline orders might provide worse estimation results, they are essential to posterior spike sorting. This justifies that, for a given value of samples $N$, the highest possible E-Spline order $P$ should be chosen for the sampling in the hope of capturing information valuable for spike sorting, even if it comes at the cost of a small extra estimation error.

### 5.3.3 Spike sorting at sub-Nyquist rates performance

The simultaneous signal and system estimation algorithm developed in this project are now tried as the data acquisition module of a spike sorting algorithm. The main concern is to see if this data estimation technique has any repercussions on the spike sorting process. Because the sampling performed is a reconstruction process, data is not faithfully represented as acquired. Therefore, if the distinguishing features of a spike shape that are required for reliable spike sorting are lost in the sampling process, this algorithm will not be suitable as acquisition

device for neural data. If on the contrary, these features happen to be enhanced by the reconstruction, current spike sorting methods could be outperformed.

To this end, the publicly available software Wave_Clus from [34] is used. Details and results obtained with this software can be found in [6].

- Spike detection is done using a metric based on the median of the input data.
- Spike alignment is obtained through oversampling and interpolation.
- Feature extraction is made using the first 3 PCA components.
- Spike clustering is based on Superparamagnetic Clustering (SPC).

The data analyzed is a long artificial train of spikes belonging to three different neurons under realistic noise conditions [6]. Two sets of simulations are analyzed (see figure 5.11), under the effect of an additive noise of standard deviation $\sigma$ relative to the peak spike amplitudes.

The FRI sampling is simulated by resampling the data that is used for the original programme. The data is originally sampled at 24 KHz and the location and class of each spike is known a priori. One by one, a total of 1000 action potentials are estimated using the algorithm described above, effectively estimating an $L = 128$ sample pulse with only $N$ FRI samples. The reconstructed spikes are then placed back into their respective locations inside the long train of action potentials producing a data set with an effective sampling rate of $f_{sN} = 24000\frac{N}{128}$. The rates investigated are 5.8 KHz ($N = 31$ with $P = 30$) and 4.6 KHz ($N = 25$ with $P = 24$).

Errors in the classification are classified as Type I (unclassified and false positives) and Type II errors (missed spikes) and the metric that is used for performance comparison simply counts the number of errors with respect to the number of spikes being analyzed. That is

$$\%correct = \frac{\#spikes - \#errors}{\#spikes}. \tag{5.1}$$

A third data set is generated by simply downsampling the original data by 4. This is essentially the dataset at a similar rate as one of the FRI sampled datasets, but in which data was acquired using the classical sampling framework and not using a reconstruction algorithm. This is obviously not a fair comparison because the software is explicitly designed to work at frequencies above Nyquist. Only nonlinear reconstruction methods such as the one proposed in this project can

be expected to provide a neural recording suitable for this software. Nonetheless, results for this dataset are also provided.

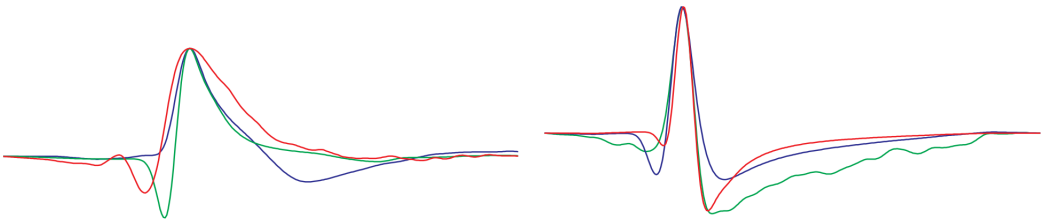


Figure 5.11: Spike shapes being sorted. Set 1 (left) and set 2 (right).

| | | Type II errors (#) | | | | Type I errors (#) | | | | Success rate (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Spike set | Noise variance | 24kHz C | 6kHz C | 5.8kHz FRI | 4.6kHz FRI | 24kHz C | 6kHz C | 5.8kHz FRI | 4.6kHz FRI | 24kHz C | 6kHz C | 5.8kHz FRI | 4.6k FRI |
| Set 1 | 0.05 | 111 | 78 | 135 | 153 | 52 | 114 | 43 | 48 | 83.7 | 80.8 | 82.2 | 79.9 |
| | 0.1 | 93 | 178 | 91 | 143 | 44 | 298 | 47 | 564 | 86.3 | 52.4 | 86.2 | 29.3 |
| | 0.15 | 143 | 352 | 129 | 176 | 58 | 229 | 79 | 95 | 79.9 | 41.9 | 79.2 | 72.9 |
| | 0.2 | 248 | 382 | 216 | 269 | 39 | 226 | 64 | 94 | 71.3 | 39.2 | 72 | 63.7 |
| Set 2 | 0.05 | 140 | 36 | 149 | 146 | 87 | 424 | 78 | 77 | 77.3 | 54 | 77.3 | 77.7 |
| | 0.1 | 101 | 56 | 80 | 84 | 418 | 672 | 221 | 449 | 48.1 | 27.2 | 69.9 | 46.7 |
| | 0.15 | 115 | 93 | 86 | 91 | 347 | 621 | 474 | 404 | 53.8 | 28.6 | 44 | 50.5 |
| | 0.2 | 160 | 158 | 108 | 121 | 444 | 585 | 439 | 490 | 39.6 | 25.7 | 45.3 | 38.9 |
| Average | 0.125 | 138.88 | 166.62 | 124.25 | 147.87 | 186.12 | 396.12 | 180.62 | 277.62 | 67.5 | 43.72 | 69.51 | 57.45 |

Table 5.1: Spike sorting comparison for datasets acquired at different frequency rates. Letters C and FRI stand for classical and FRI sampling respectively.

The data reconstructed from the sampling at 5.8 KHz using the proposed estimation algorithm shows to be as suitable for spike sorting as the original data simulated at 24 KHz. This gives a reduction of 4.1x the original sampling frequency for no decrease in sorting performance. This implies that distinctive features of action potentials that are used for their sorting are not lost in the sampling process. At lower sampling frequencies such as 4.6 KHz, the performance of the sorting decays.

Although these results should be taken cautiously because all spike sorting techniques depend very much on input data, it demonstrates the fact that, at least in specific cases, the use of sub-Nyquist frequencies for the sampling of neural spikes features can provide satisfactory sorting performances with a much more economic acquisition device.

# Chapter 6

# Conclusion and future work

Spike sorting algorithms could reveal crucial information about the diagnosis of neurological diseases and our understanding of how neurons communicate between them. However, their performances are lagging behind the capabilities of current technology. One of the main challenges is the use of multichannel recordings to increase robustness in sorting techniques. Physiological properties from neuron action potentials set a Nyquist rate for data acquisition that increases energy consumption, computational complexity and hardware demands to unacceptable levels for multichannel recordings.

This report has introduced a novel algorithm for neural data sampling at sub-Nyquist rates. The motivating observation for its implementation is that the electrochemical activity of each neuron can be represented parametrically as the convolution of a stream of Diracs with a distinctive pulse shape. Sampling such signals can be viewed as a simultaneous signal and system estimation process. Combining concepts from FRI sampling and compressed sensing and based on evidence that both the signal and the system are in some sense sparse, this estimation can be performed at potentially much lower rates than that imposed by Shannon's theorem.

The algorithm presented has been shown to reconstruct spike shapes up to an accuracy determined by the kernel used for the sampling. It was found that, although high order E-Splines introduce a small error in the estimation compared to lower order E-Splines, their use was necessary in order to ensure that the sampling process captured unique features of each spike. Without these, spike sorting algorithms are unable to classify action potentials from different neurons. Sampling frequencies 4x lower than the Nyquist rate were demonstrated to retain characteristics necessary for state-of-the-art spike sorting results.

Future work should mainly focus on ways to overcome the limitations of the Basis Pursuit module. The fact that the deterministic sensing matrix limits the reconstruction of the pulse shape to use only the lowest wavelet resolutions was not anticipated. As a result, estimation accuracy had to be traded by increasing E-Spline orders so as to ensure the sampling did not miss features necessary for the sorting, which are usually high frequency. The sampling kernel determines the sensing matrix, so the design of new E-Splines to this end is not discarded. Also, the implementation of spike sorting algorithms particularly suited to the acquisition device presented is suggested as future development. The method proposed saves energy consumption in the sampling process, meaning that data can for instance be transmitted wirelessly to a local processing machine economically, but algorithms making use of the parametric signal modeling exploited here could greatly reduce computational complexity also in the sorting process.

Overall, this project has been successful in that state-of-the-art spike sorting performances have been achieved at rates considerably lower than the Nyquist threshold. This is a step further in the application of sparse sampling techniques to brain activity recordings, whose ongoing development is hoped to help answering many questions about the most complex organ in the human body.

# Bibliography


[1] C.E. Shannon, "Communication in the presence of noise", Proc. Institute of Radio Engineers, vol. 37 (1), pp. 10-21, January 1949. Reprint as classic paper in: Proc. IEEE, vol. 86 (2), Feb. 1998.

[2] K.-K. Poh and P. Marziliano, "Compression of neonatal EEG seizure signals with finite rate of innovation," Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2008), pp. 433–436, Mar.-Apr. 2008.

[3] Y. Hao, P. Marziliano, M. Vetterli and T. Blu, "Compression of ECG as a signal with finite rate of innovation," Proc. 27th Annu. Int. Conf. Eng. Medicine Biology Soc. (IEEE-EMBS 2005), pp. 7564–7567, Jan. 2006.

[4] Z.M. Charbiwala, V. Karkare, S. Gibson, D. Markovic and M.B. Srivastava, "Compressive Sensing of Neural Action Potentials Using a Learned Union of Supports," Proceedings of the International Conference on Body Sensor Networks (BSN 2011), Dallas, TX , May 2011.

[5] L. Srinivasan, L.R. Varshney and J. Kusuma, "Acquisition of Action Potentials with Ultra-Low Sampling Rates," Conf. Proc. IEEE Eng. Med. Biol. Soc., pp. 4213-6, Aug.-Sep. 2010.

[6] R. Quian Quiroga, Z. Nadasdy and Y. Ben-Shaul, "Unsupervised Spike Detection and Sorting with wavelets and superparamagnetic clustering," Neural Comp., vol. 16, pp. 1661-1687, 2004.

[7] M. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," Network: Computation in Neural Systems, vol. 9, pp. R53-R78, 1998.

[8] M. Abeles and M. Goldstein Jr, "Multispike train analysis," Proceedings of the IEEE, vol. 65, pp. 762-773, 1977.

[9] E.M. Schmidt, "Computer separation of multi-unit neuroelectric data: a review", Journal of Neuroscience Methods, vol. 12, pp. 95-111, 1984.

[10] P.R. Gray, "Conditional probability analyses of the spike activity of single neurons," Biophysical Journal 1, vol. 7 (6), pp. 759-777, Nov. 1967.

[11] 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, vol. 69, pp. 175-188, 1996.

[12] C. Pouzat, O. Mazor and Laurent G, "Using noise signature to optimize spike-sorting and to assess neuronal classification quality," Journal of Neuroscience Methods, vol. 122, pp. 43-57, 2002.

[13] 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, vol. 63, pp. 43-54, 1995.

[14] G.L. Gerstein and W.A. Clark, "Simultaneous studies of firing patterns in several neurons," Science, vol. 143, pp. 1325-1327, 1964.

[15] D.A. Adamos, E. K. Kosmidis and G. Theophilidis, "Performance evaluation of PCA-based spike sorting algorithms," Computer methods and programs in Biomedicine, vol. 91(3), pp. 232-244, 2008.

[16] P.P. Mitra and H. Bokil, Observed Brain Dynamics, Oxford Press (9), pp. 257-270, 2007.

[17] P.L. Dragotti, M. Vetterli and T. Blu, "Sampling moments and reconstructing signals of Finite Rate of Innovation: Shannon meets Strang-Fix," IEEE Transactions on Signal Processing, vol. 55 (5), pp. 1741-1757, 2007.

[18] M. Vetterli, P. Marziliano and T. Blu, "Sampling signals with finite rate of innovation," IEEE Trans. Signal Processing, vol. 50 (6), pp. 1417-1428, June 2002.

[19] M. Unser and T. Blu, "Cardinal Exponential Splines: Part I – Theory and filtering algorithms," IEEE Transactions on Signal Processing, vol. 53, pp. 1425-1438, Apr. 2005.

[20] J.A. Uriguen, P.L. Dragotti and T. Blu, "On the exponential reproducing kernels for sampling signals with Finite Rate of Innovation," Proc. of International Conference on Sampling Theory and Applications (SampTA), Singapore, May 2011.

[21] P. Stoica and R. Moses, "Introduction to Spectral Analysis," Engelwood Cliffs, NJ, Prentice-Hall, 2000.

[22] I. Maravic and M. Vetterli, "Sampling and reconstruction of signals with finite rate of innovation in the presence of noise," IEEE Transactions on Signal Processing, vol. 53, pp. 2788-2805, 2005.

[23] T. Blu, P.L. Dragotti, M. Vetterli, P. Marziliano and L. Coulot, "Sparse sampling of signal innovations," IEEE Signal Processing Magazine, vol.25 (2), pp. 31-40, 2008.

[24] J.A. Cadzow, "Signal enhancement - A composite property mapping algorithm," Acoustics, Speech and Signal Processing IEEE Transactions, vol. 36, pp. 49-62, Jan. 1988.

[25] I. Daubechies, R. DeVore, M. Fornasier and S. Güntürk, "Iteratively re-weighted least squares minimization for sparse recovery," Commun. Pure Appl. Math., pp. 35, June 2008.

[26] E. Candès, J. Romberg and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," IEEE Trans. on Information Theory, vol. 52(2), pp. 489-509, Feb. 2006.

[27] R. Baraniuk, "Compressive sensing," IEEE Signal Processing Mag, pp. 118-120, 2007.

[28] D.L. Donoho, "Compressed sensing," IEEE Trans. on Information Theory, vol. 52(4), pp. 1289-1306, Apr. 2006.

[34] University of Leicester, Department of Engineering, R. Quian Quiroga, "Spike Sorting - Wave_Clus": http://www2.le.ac.uk/departments/engineering/research/bioengineering/neuroengineering-lab/spike-sorting [accessed on 06/08/20011].

[29] S.S. Chen, D.L. Donoho and M.A. Saunders, "Atomic decomposition by basis pursuit," SIAM Journal on Scientific Computing, vol. 20, pp. 33-61, 1998.

[30] J.C. Letelier, P.P. Weber, "Spike sorting based on discrete wavelet transform coefficients," Journal of Neuroscience Methods, vol. 101 (2), pp. 93-106, 2000.

[31] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical recipes in C: The art of scientific computing, Cambridge: Cambridge University Press, 1993.

[32] H. Akhondi and P.L. Dragotti, "Simultaneous estimation of sparse signals and systems at sub-Nyquist rates," 4th Workshop on Signal Processing with Adaptive Sparse Structured Representations, Edinburgh, June 2011.

[33] Stanford University, "SparseLab – Seeking sparse solutions to linear systems of equations": http://sparselab.stanford.edu/ [accessed on 27/07/2011].

[35] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," Journal of the Royal Statistical Society, Series B, vol. 58, pp. 267-288, 1994.

[36] S. Mallat and Z. Zhong, "Matching Pursuit in a time-frequency dictionary," IEEE Transactions on Signal Processing, vol. 41, pp. 3397-3415, 1993.

[37] M. Elad, B. Matalon, J. Shtok and M. Zibulevsky, "A wide-angle view at iterated shrinkage algorithms," in SPIE (Wavelet XII), pp. 26-29, 2007.