

Imperial College London  
Department of Electrical and Electronic Engineering  
Communications and Signal Processing Group

# Sampling and Reconstruction of Finite Rate of Innovations Signals with Applications in Neuroscience and Sparse Representation

Jon Oñativia Bravo

2015

Supervised by Prof Pier Luigi Dragotti

Submitted in part fulfilment of the requirements for the degree of  
Doctor of Philosophy of Imperial College London



# Declaration

I declare that this thesis, and the research it contains, is the product of my own work under the guidance of my thesis supervisor Prof Pier Luigi Dragotti. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. The material of this thesis has not been submitted for any degree at any other academic or professional institution.

Jon Oñativia Bravo



*Zuri, nire maitia, Nekane*



# Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.



# Abstract

The sampling process is the link between continuous physical quantities and discrete sequences. Classical sampling theory restricts perfect reconstruction to bandlimited signals. During the past decade, a new theory is emerging which overcomes this limitation by describing a signal in terms of its innovation parameters per unit of time. This theory is known as Finite Rate of Innovation (FRI). This thesis extends the current theory with applications in neuroscience and sparse vector recovery.

First, we propose an algorithm to sample and reconstruct streams of Diracs. The FRI literature has only focused on the sampling of periodic or finite duration signals. The proposed method is able to reconstruct infinite streams where no clear separation between consecutive bursts can be established. We sequentially process the discrete samples and output locations and amplitudes of the Diracs in real-time. The algorithm achieves perfect reconstruction in the noiseless scenario. An extension for the noisy case is also proposed. Simulation results show that this novel method is able to reconstruct the original stream of Diracs very accurately even in very noisy situations.

Next, we present a novel application of the FRI theory to infer the spiking activity of individual neurones. Fluorescence sequences are obtained from two-photon imaging of calcium signals in regions of the brain of in vivo mice. Action potentials are well characterised by decaying exponentials in this type of data. A novel method to sample and reconstruct streams of decaying exponentials is developed which is directly applied to fluorescence sequences to infer the timing of action potentials. The algorithm is tested with both real and surrogate data and outperforms state of the art methods for spike train inference from calcium imaging data

Finally, we analyse the problem of finding the sparse representation of a finite-dimensional signal in an overcomplete dictionary. Recently, a new algorithm, ProSparse, has been presented which solves the sparse representation problem using Prony's method. We provide a probabilistic analysis of the algorithm and demonstrate that it presents a phase transition behaviour. We validate the analysis with extensive simulations and compare the performance of this approach against another sparse recovery algorithm: Basis Pursuit. We also propose a variation of ProSparse for the noisy scenario. This approach outperforms state of the art algorithms in a number of different scenarios.



# Acknowledgements

Firstly I would like to thank my supervisor Pier Luigi for all his help and encouragement during my PhD. This long journey would not have been possible without his guidance, wisdom and enthusiasm. Despite his busy schedule he always finds time for the weekly meetings with his students. It has been an honour to have him as my supervisor during my Master's project and PhD thesis.

I am very grateful to Yue M. Lu from Harvard University who has been like a second supervisor during the last stage of my PhD. His mathematical rigour and sharpness have been essential to find some of the results that are presented here. I am also thankful to Simon Schultz from the Bioengineering Department at Imperial College London who provided all the data and knowledge about action potentials, neural activity and all neuroscience related topics. I am also grateful to the examiners Mike Brooks from Imperial College and Jared Tanner from Oxford University who contributed to the improvement of this thesis by carefully reading the initial version, suggesting modifications and indicating further references.

Additionally, I would like to thank my good friends from Imperial, Felipe and David, for all the great moments on the South Kensington area during the endless days that started in the lab and finished in some nearby pub. John and Hamdi, my *bro*s who always put a smile on my face. Toni, who taught me all the FRI tricks when I started my PhD. Sira for encouraging me to come to Imperial. My Spanish friends in London who made these four years far from my beloved ones much easier: Diego and Carmen for adopting me when I arrived in London, special thanks to *gente que mola*, Azu, Sam and Eli for all the sushi evenings and *amigos de la cerveza*, Alberto, Arturo and Pablo for all the beer tasting Saturdays. Iñigo for all the enriching discussions in the four corners of the world and Julen for feeding my entrepreneurial dreams. The *cuadrilla* from San Sebastian, Maria, Iosu, Xabi, Paul, Martin, Iulen, ..., who are always there to share some *pintxos* when going back home.

Finally, I would like to thank my family for their love and endless support in all my projects. Elaia and Unai who made the world a more beautiful place. And the biggest thanks of all goes to Nekane who gave me strength and love through the ups and downs of my PhD.

*Jon Oñativia Bravo*  
*Donostia, December 2015.*



# Contents

|                                                                                                 |           |
|-------------------------------------------------------------------------------------------------|-----------|
| <b>Notations and formulas</b>                                                                   | <b>19</b> |
| <b>1 Introduction</b>                                                                           | <b>21</b> |
| 1.1 Original contribution and outline of the thesis . . . . .                                   | 22        |
| 1.2 Reproducible research . . . . .                                                             | 23        |
| 1.3 Publications . . . . .                                                                      | 24        |
| <b>2 Classical sampling theory and sampling FRI signals</b>                                     | <b>27</b> |
| 2.1 Classical sampling theory . . . . .                                                         | 27        |
| 2.1.1 The aliasing effect . . . . .                                                             | 29        |
| 2.1.2 Acquisition process . . . . .                                                             | 31        |
| 2.2 Signals with Finite Rate of Innovation . . . . .                                            | 32        |
| 2.2.1 Examples of FRI signals . . . . .                                                         | 34        |
| 2.3 Sampling kernels . . . . .                                                                  | 35        |
| 2.3.1 Strang-Fix conditions and splines for polynomial reproduction . . . .                     | 36        |
| 2.3.2 Generalised Strang-Fix conditions and splines for exponential repro-<br>duction . . . . . | 39        |
| 2.4 Sampling and perfect reconstruction of FRI signals . . . . .                                | 42        |
| 2.4.1 Sampling with an exponential reproducing kernel . . . . .                                 | 42        |
| 2.4.2 Reconstruction of a stream of Diracs . . . . .                                            | 44        |
| 2.5 Sampling FRI signals in the presence of noise . . . . .                                     | 48        |
| 2.5.1 Performance of the noisy reconstruction algorithms . . . . .                              | 51        |
| 2.5.2 On the effect of coefficients $c_{m,n}$ on the statistics of the noise . . . .            | 52        |
| 2.6 Summary . . . . .                                                                           | 53        |
| <b>3 Sampling infinite streams of Diracs</b>                                                    | <b>55</b> |
| 3.1 Introduction . . . . .                                                                      | 55        |
| 3.2 Sampling FRI signals . . . . .                                                              | 56        |
| 3.3 Sampling an infinite stream of Diracs . . . . .                                             | 58        |
| 3.3.1 Noiseless case . . . . .                                                                  | 59        |
| 3.3.2 Noisy scenario . . . . .                                                                  | 62        |
| 3.4 Simulation results . . . . .                                                                | 63        |

|          |                                                                           |            |
|----------|---------------------------------------------------------------------------|------------|
| 3.5      | A note on periodicity and reducing the number of temporal samples . . . . | 65         |
| 3.6      | Summary . . . . .                                                         | 68         |
| <b>4</b> | <b>Neural activity monitoring</b>                                         | <b>69</b>  |
| 4.1      | Introduction . . . . .                                                    | 70         |
| 4.2      | Methods . . . . .                                                         | 71         |
| 4.2.1    | Experimental methods . . . . .                                            | 71         |
| 4.2.2    | Mathematical model . . . . .                                              | 72         |
| 4.2.3    | Spike detection . . . . .                                                 | 74         |
| 4.2.4    | Generating surrogate data . . . . .                                       | 82         |
| 4.2.5    | Real-time processing . . . . .                                            | 82         |
| 4.3      | Results . . . . .                                                         | 83         |
| 4.3.1    | Real data . . . . .                                                       | 84         |
| 4.3.2    | Surrogate data . . . . .                                                  | 84         |
| 4.3.3    | Comparison with existing methods . . . . .                                | 85         |
| 4.4      | Summary . . . . .                                                         | 87         |
| <b>5</b> | <b>ProSparse average performance and phase transition</b>                 | <b>89</b>  |
| 5.1      | The sparse representation problem . . . . .                               | 89         |
| 5.2      | Overview of ProSparse . . . . .                                           | 94         |
| 5.3      | Exact probabilistic average-case analysis of ProSparse . . . . .          | 96         |
| 5.3.1    | Probability model . . . . .                                               | 97         |
| 5.3.2    | Exact probability for maximum gap size . . . . .                          | 97         |
| 5.3.3    | Probability of success of ProSparse . . . . .                             | 99         |
| 5.4      | Asymptotic analysis and phase transition . . . . .                        | 100        |
| 5.5      | Numerical validation and simulations . . . . .                            | 104        |
| 5.5.1    | Simulation results . . . . .                                              | 105        |
| 5.6      | Summary . . . . .                                                         | 105        |
| <b>6</b> | <b>Sparse pattern recovery in the presence of noise</b>                   | <b>107</b> |
| 6.1      | Introduction . . . . .                                                    | 107        |
| 6.2      | Finite-dimensional Cadzow and circulant matrices . . . . .                | 111        |
| 6.2.1    | The union of bases case, fast circulant Cadzow algorithm . . . . .        | 112        |
| 6.3      | Noisy sparse recovery based on Cadzow and Prony . . . . .                 | 113        |
| 6.3.1    | Complexity analysis of ProSparse Denoise . . . . .                        | 114        |
| 6.4      | Guaranteed performance analysis . . . . .                                 | 115        |
| 6.4.1    | Signal-to-noise ratio of the signal . . . . .                             | 115        |
| 6.4.2    | Sufficient condition for sparse recovery with $K_p = K_q = 1$ . . . . .   | 116        |
| 6.5      | Simulation results . . . . .                                              | 119        |
| 6.6      | Summary . . . . .                                                         | 121        |

|          |                                                                      |            |
|----------|----------------------------------------------------------------------|------------|
| <b>7</b> | <b>Conclusions</b>                                                   | <b>125</b> |
| 7.1      | Summary . . . . .                                                    | 125        |
| 7.2      | Future research . . . . .                                            | 127        |
| <b>A</b> | <b>Function reproduction with splines</b>                            | <b>129</b> |
| A.1      | Polynomial reproducing kernels and $c_{m,n}$ coefficients . . . . .  | 129        |
| A.2      | Exponential reproducing kernels and $c_{m,n}$ coefficients . . . . . | 130        |
| <b>B</b> | <b>Toeplitz matrices</b>                                             | <b>133</b> |
| B.1      | Rank deficiency of Toeplitz matrix . . . . .                         | 133        |
| B.2      | Generalised eigenvalues and rank reducing numbers . . . . .          | 134        |
| <b>C</b> | <b>CRB and parameter estimation</b>                                  | <b>135</b> |
| C.1      | Parameter estimation and Cramér-Rao lower bound . . . . .            | 135        |
| C.1.1    | Likelihood function and Fisher information matrix . . . . .          | 136        |
| C.2      | CRB in the FRI setup . . . . .                                       | 139        |
| C.2.1    | CRB of the measurements $s[m]$ . . . . .                             | 139        |
| <b>D</b> | <b>Exponential reproducing kernels and eMOMS</b>                     | <b>143</b> |
| D.1      | E-splines and eMOMS . . . . .                                        | 143        |
| D.1.1    | eMOMS and stability of reconstruction of FRI signals . . . . .       | 144        |
| D.1.2    | Dirichlet kernel and e-MOMS . . . . .                                | 146        |
| <b>E</b> | <b>Maximum of Rayleigh random variables</b>                          | <b>151</b> |
| E.1      | Maximum of $N$ random variables . . . . .                            | 151        |
| E.2      | Rayleigh distribution and maximum of random variables . . . . .      | 152        |
|          | <b>Bibliography</b>                                                  | <b>155</b> |

## List of Figures

|     |                                                               |    |
|-----|---------------------------------------------------------------|----|
| 1.1 | From continuous-time to discrete-time . . . . .               | 21 |
| 2.1 | Sampling and reconstruction of a bandlimited signal . . . . . | 29 |
| 2.2 | Acquisition process . . . . .                                 | 29 |
| 2.3 | $g_r(t)$ functions in bandlimited functions . . . . .         | 33 |
| 2.4 | Examples of signals with FRI . . . . .                        | 35 |
| 2.5 | B-splines . . . . .                                           | 38 |

|      |                                                                                            |     |
|------|--------------------------------------------------------------------------------------------|-----|
| 2.6  | Polynomial reproduction . . . . .                                                          | 39  |
| 2.7  | E-splines . . . . .                                                                        | 40  |
| 2.8  | Exponential reproduction . . . . .                                                         | 41  |
| 2.9  | Sampling and perfect reconstruction of a stream of Diracs . . . . .                        | 46  |
| 2.10 | Noisy recovery performance . . . . .                                                       | 50  |
| 3.1  | Support of the sampling kernel . . . . .                                                   | 56  |
| 3.2  | Border effects . . . . .                                                                   | 59  |
| 3.3  | Perfect reconstruction interval . . . . .                                                  | 60  |
| 3.4  | Noisy scenario, scatter plot and histogram . . . . .                                       | 62  |
| 3.5  | Sequential perfect reconstruction of stream of Diracs . . . . .                            | 63  |
| 3.6  | Noisy samples and reconstruction . . . . .                                                 | 65  |
| 3.7  | Critical sampling device . . . . .                                                         | 67  |
| 3.8  | Critical sampling signals . . . . .                                                        | 68  |
| 4.1  | Simultaneous multiphoton calcium imaging with electrophysiology . . . . .                  | 73  |
| 4.2  | Sampling a stream of decaying exponentials and equivalent problem . . . . .                | 75  |
| 4.3  | Sampling of a stream of decaying exponentials and perfect reconstruction . . . . .         | 77  |
| 4.4  | Singular values with noise . . . . .                                                       | 79  |
| 4.5  | Fluorescence signal processing with a sliding window . . . . .                             | 80  |
| 4.6  | Double consistency spike search with real data. . . . .                                    | 81  |
| 4.7  | Surrogate data. Temporal resolution $T_{\text{res}} = 147.2$ ms and $SNR = 15$ dB. . . . . | 83  |
| 4.8  | Algorithm's performance measurement with surrogate data . . . . .                          | 84  |
| 4.9  | Simulations results . . . . .                                                              | 86  |
| 5.1  | Bounds comparison . . . . .                                                                | 91  |
| 5.2  | Phase transition in ProSparse . . . . .                                                    | 93  |
| 5.3  | Gaps between consecutive spikes . . . . .                                                  | 97  |
| 5.4  | Exact probability <i>vs</i> Monte Carlo simulation . . . . .                               | 104 |
| 5.5  | ProSparse <i>vs</i> Basis Pursuit . . . . .                                                | 106 |
| 6.1  | Cadzow denoise for spike removal . . . . .                                                 | 111 |
| 6.2  | Guaranteed performance of ProSparse Denoise . . . . .                                      | 118 |
| 6.3  | Simulation results, union of Fourier and identity bases . . . . .                          | 122 |
| 6.4  | Simulation results, union of Fourier and identity bases . . . . .                          | 123 |
| C.1  | CRB of two nearby Diracs . . . . .                                                         | 141 |
| D.1  | $P = 16$ , E-spline, eMOMS and Dirichlet kernel. . . . .                                   | 147 |
| D.2  | Shifted Dirichlet kernel and stable eMOMS . . . . .                                        | 148 |
| E.1  | PDF of the maximum of Rayleigh distributed r.v. . . . .                                    | 154 |

## List of Tables

|     |                                                             |     |
|-----|-------------------------------------------------------------|-----|
| 3.1 | Sequential algorithm's performance . . . . .                | 64  |
| 6.1 | Average execution time in sparse recovery (N=128) . . . . . | 120 |
| 6.2 | Average execution time in sparse recovery (N=256) . . . . . | 120 |



# Notations and formulas

## Sets and numbers

|                     |                                                                         |
|---------------------|-------------------------------------------------------------------------|
| $\mathbb{N}$        | Positive integers including 0                                           |
| $\mathbb{Z}$        | Integers                                                                |
| $\mathbb{R}$        | Real numbers                                                            |
| $\mathbb{R}^+$      | Positive real numbers                                                   |
| $\mathbb{C}$        | Complex numbers                                                         |
| $z^*, \bar{z}$      | Complex conjugate of $z \in \mathbb{C}$                                 |
| $[a, b]$            | Closed interval: $\{x \in \mathbb{R} : a \leq x \leq b\}$               |
| $(a, b)$            | Open interval: $\{x \in \mathbb{R} : a < x < b\}$                       |
| $[a, b)$            | Left-closed, right-open interval: $\{x \in \mathbb{R} : a \leq x < b\}$ |
| $(a, b]$            | Left-open, right-closed interval: $\{x \in \mathbb{R} : a < x \leq b\}$ |
| $\lfloor x \rfloor$ | Floor function: $\max \{n \in \mathbb{Z} : n \leq x\}$                  |
| $\lceil x \rceil$   | Ceil function: $\min \{n \in \mathbb{Z} : n \geq x\}$                   |
| $n \bmod N$         | Remainder of the integer division of $n$ by $N$                         |

## Signals, vectors and matrices

|                                          |                                                                                   |
|------------------------------------------|-----------------------------------------------------------------------------------|
| $f(t)$                                   | Continuous-time signal                                                            |
| $f[n]$                                   | Discrete-time signal                                                              |
| $L^2(\mathbb{R})$                        | Finite energy functions: $\int_{-\infty}^{+\infty}  f(t) ^2 dt < \infty$          |
| $\ell^2(\mathbb{Z})$                     | Finite energy discrete signals: $\sum_{n=-\infty}^{+\infty}  f[n] ^2 < \infty$    |
| $\langle f(t), g(t) \rangle$             | Inner product: $\int_{-\infty}^{+\infty} f(t) g^*(t) dt$                          |
| $\langle f[n], g[n] \rangle$             | Inner product: $\sum_{n=-\infty}^{+\infty} f[n] g^*[n]$                           |
| $\delta(t)$                              | Dirac distribution: $\int_{-\infty}^{+\infty} \delta(t) f(t) dt = f(0)$           |
| $\delta[n]$                              | Discrete Dirac: 1 if $n = 0$ , 0 otherwise                                        |
| $\mathbf{1}_A(t)$                        | Indicator function: 1 if $t \in A$ , 0 otherwise                                  |
| $f(t) * g(t)$                            | Continuous-time convolution: $\int_{-\infty}^{+\infty} f(\tau) g(t - \tau) d\tau$ |
| $f[n] * g[n]$                            | Discrete-time convolution: $\sum_{m=-\infty}^{+\infty} f[m] g[n - m]$             |
| $\mathbf{x} \in \mathbb{C}^N$            | Finite-dimensional signal of size $N$ : $\mathbf{x} = (x[n])_{n=0}^{N-1}$         |
| $\mathbf{A} \in \mathbb{C}^{M \times N}$ | Matrix of size $M \times N$                                                       |
| $\mathbf{A}^T$                           | Transpose of matrix $\mathbf{A}$                                                  |
| $\overline{\mathbf{A}}$                  | Element-wise complex conjugate of $\mathbf{A}$                                    |
| $\mathbf{A}^H$                           | Conjugate transpose or Hermitian transpose of matrix $\mathbf{A}$                 |

## Transforms

|                                                  |                                                                                                                       |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| $\mathcal{F}\{f(t)\} = \hat{f}(\omega)$          | Fourier transform: $\int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt$                                                  |
| $\mathcal{F}^{-1}\{\hat{f}(\omega)\} = f(t)$     | Inverse Fourier transform: $\frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega t} d\omega$            |
| $\mathcal{F}\{f[n]\} = F(e^{i\omega})$           | Discrete-time Fourier transform: $\sum_{n=-\infty}^{+\infty} f[n] e^{-i\omega n}$                                     |
| $\mathcal{F}^{-1}\{F(e^{i\omega})\} = f[n]$      | Inverse discrete-time Fourier transform of: $\frac{1}{2\pi} \int_{2\pi} F(e^{i\omega}) e^{i\omega t} d\omega$         |
| $\mathcal{Z}\{f[n]\} = F(z)$                     | Z-transform: $\sum_{n=-\infty}^{+\infty} f[n] z^{-n}$                                                                 |
| $\text{DFT}_N\{\mathbf{x}\} = \hat{\mathbf{x}}$  | Discrete Fourier transform: $\left(\sum_{n=0}^{N-1} x[n] e^{-i2\pi kn/N}\right)_{k=0}^{N-1}$                          |
| $\text{IDFT}_N\{\hat{\mathbf{x}}\} = \mathbf{x}$ | Inverse discrete Fourier transform: $\frac{1}{N} \left(\sum_{k=0}^{N-1} \hat{x}[k] e^{i2\pi kn/N}\right)_{n=0}^{N-1}$ |

## Formulas

|                     |                                                                                                                                                                         |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fourier series      | For a periodic signal $f(t)$ of period $T$ :<br>$f(t) = \sum_{k=-\infty}^{+\infty} a_k e^{i2\pi kt/T}, \text{ where } a_k = \frac{1}{T} \int_T f(t) e^{-i2\pi kt/T} dt$ |
| Poisson summation   | $\sum_{n=-\infty}^{+\infty} f(t - nT) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \hat{f}\left(\frac{2\pi k}{T}\right) e^{i2\pi kt/T}$                                     |
| Parseval's relation | $\int_{-\infty}^{+\infty} f(t) g^*(t) dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) \hat{g}^*(\omega) d\omega$                                           |

## Probability

|                   |                                       |
|-------------------|---------------------------------------|
| $\mathbb{P}\{A\}$ | Probability of event $A$              |
| $\mathbb{E}\{X\}$ | Expected value of random variable $X$ |

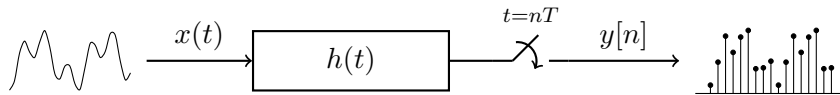
## Asymptotic notation

|                     |                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\mathcal{O}(g(x))$ | $\mathcal{O}(g(x)) = \{f(x) : \text{there exist positive constants } c \text{ and } x_0$<br>such that $0 \leq f(x) \leq c g(x)$ for all $x \geq x_0\}$   |
| $o(g(x))$           | $o(g(x)) = \{f(x) : \text{for any constant } c > 0 \text{ there exists a constant}$<br>$x_0 > 0$ such that $0 \leq f(x) < c g(x)$ for all $x \geq x_0\}$ |

# Chapter 1.

## Introduction

Real-world signals are usually continuous-time and in order to process them with digital systems we first have to discretise them, this is known as the sampling process. Sampling devices from the twentieth century have been driven by the theory developed by Nyquist, Shannon, Whittaker and Kotelnikov. This approach imposes a limit on the type of signals that can be acquired based on their frequency content, making perfect reconstruction only achievable for bandlimited signals. Figure 1.1 is a schematic illustration of a system that *samples* a continuous-time signal  $x(t)$ . The filter  $h(t)$  accounts for the modifications that the analogue signal suffers during the process. Samples are acquired every  $T$  seconds leading to the discrete-time signal  $y[n]$ . One obvious question that arises at this point is whether it is possible to recover  $x(t)$  from the only knowledge of the sequence of samples  $y[n]$ .



**Figure 1.1.:** From continuous-time to discrete-time.

The sampling process inherently produces a loss of information, and if  $x(t)$  is an arbitrary signal we will never be able to reconstruct it from the samples  $y[n]$ . However, depending on the nature of the signal  $x(t)$ , the properties of the filter  $h(t)$ , and the sampling period  $T$ , there are some situations where the original signal can be recovered. The traditional approach in the signal processing literature has been to consider signals that are limited in frequency, that is, signals that do not present rapid variations. If the sampling rate  $1/T$  is at least twice the bandwidth of the signal, then the signal can be perfectly reconstructed from its samples. In the past decade, a new sampling theory has emerged where signals are characterised in terms of their degrees of freedom. In this new framework, we say that a signal can be recovered when it presents a limited number of degrees of freedom. The terminology *rate of innovation* is coined to refer to the degrees of freedom of the signal per unit of time. This new sampling theory considers that the signals that can be perfectly reconstructed are those that present a finite rate of innovation. Thus, the name of this

new framework: *finite rate of innovation* (FRI) theory.

The idea that a signal can be reconstructed from some partial information can also be applied to the entirely discrete-time and finite-dimensional case. In this context, the signal that we are after is a finite-dimensional vector  $\mathbf{x}$  of size  $L$  and we consider that we have access to a vector  $\mathbf{y}$  of size  $N$ , with  $N < L$ . In other words, we want to reconstruct  $L$  samples  $x[0], x[1], \dots, x[L-1]$  that form the vector  $\mathbf{x}$  from the knowledge of  $N$  samples  $y[0], y[1], \dots, y[N-1]$ . We further assume that the two vectors have a linear relationship that can be written in matricial form as  $\mathbf{y} = \mathbf{D} \mathbf{x}$ , where  $\mathbf{D}$  is a matrix of size  $N \times L$ . In general, this is an underdetermined linear system of equations that cannot be solved since  $N < L$ . We thus have to impose some conditions on the vector  $\mathbf{x}$ . Similarly to the finite rate of innovation concept for the continuous-time case, here we consider *sparse* vectors, that is, vectors that have a small number of non-zero elements. Depending on the number of non-zero elements and the specific structure of the matrix  $\mathbf{D}$ , there are cases where  $\mathbf{x}$  can be perfectly reconstructed from the observed vector  $\mathbf{y}$ . This setup has a wide range of applications that include signal compression, denoising or separation. Moreover, if we consider that a two-dimensional image can be represented by a one-dimensional vector, for instance, by stacking all the pixel values in a single column, this framework is also extensible to image processing applications such as image super-resolution or inpainting, as well as image compression or denoising.

In this thesis we analyse and propose some novel algorithms and theoretical results in the context of these two scenarios:

1. the recovery of a continuous-time signal  $x(t)$  from some discrete-time samples  $y[n]$ ,
2. and, the reconstruction of a finite-dimensional vector  $\mathbf{x}$  of size  $L$  from an observed vector  $\mathbf{y}$  of size  $N$  with  $N < L$ .

In particular, in the continuous-time case, we analyse the problem where the signal  $x(t)$  corresponds to the activity of a neuron and the goal is to estimate when this neuron has fired an *action potential*. In the finite-dimensional case, we consider the problem of estimating vectors that correspond to a mixture of sinusoidal and spike elements.

## 1.1. Original contribution and outline of the thesis

The work presented in this thesis contains the contribution of several persons. The first page of each chapter contains a footnote acknowledging the particular contribution of the persons that collaborated to obtain the results that are described.

In Chapter 3 we present a sequential algorithm to sample and reconstruct streaming trains of Diracs. In the noiseless scenario, the algorithm is able to achieve perfect reconstruction, even if there is no clear separation between consecutive burst of Diracs. An

extension is also proposed for the noisy scenario that achieves a very high success rate with very few false positives.

Chapter 4 is a novel application of the FRI theory to neuroscience. We extend the sequential algorithm of the previous chapter to the case where the streaming signal is a train of decaying exponentials. This model is applied to fluorescence sequences obtained from the measurement of calcium ion concentration in neurones of in vivo mice. This algorithm operates in real-time and accurately estimates the spiking activity of neurones outperforming state of the art algorithms.

In Chapter 5 we move to the finite-dimensional scenario. We present a probabilistic analysis of the performance of a sparse reconstruction algorithm. We characterise a sharp phase transition behaviour of the algorithm which allows us to predict the performance of the algorithm depending on the sparsity level of the vector to be estimated.

Chapter 6 is a novel extension of the algorithm presented in the previous chapter for the noisy scenario. The proposed method is faster than state of the art algorithms and presents better performances in a number of scenarios.

## 1.2. Reproducible research

The results presented in this thesis are illustrated with figures and numerical experiments that should be reproducible. The text in this manuscript contains theoretical descriptions and diagrams that should allow the readers to implement their own versions of the different algorithms. However, a full description of all the parameters and details that are involved in these complex computations is not always possible. In order to guarantee the reproducibility of all the results, the code and data that has been used to generate the figures and experiments is publicly available online. Please follow this link to obtain the files: <https://github.com/jonativia/thesis> (click on the button *Download ZIP* on the lower right corner of the website to obtain all the content in a single compressed file).

This address corresponds to a public repository of the source code. Simulations and figures have been generated with MATLAB scripts, and when needed, data is stored in `.mat` files. The `README.md` text file describes which scripts reproduce the figures and the results of the thesis. The source code is organised in different folders that follow the chapters structure of the thesis, and in general, there is one script per figure. In addition to the scripts that reproduce the figures, there are a number of other source files that contain MATLAB functions that are used within these scripts. The publicly available source code corresponds to the contents that have already been published. The source code to reproduce the results of unpublished work is available upon request.

## 1.3. Publications

### To be submitted

- J. Oñativia, Y. M. Lu, and P. L. Dragotti. “Sparsity According to Prony: Average Performance Analysis and Phase Transition”. *to be submitted* (2015).

### Peer-reviewed journals

- J. Oñativia and P. L. Dragotti. “Sparse sampling: theory, methods and an application in neuroscience”. *Biological cybernetics* 109.1 (Feb. 2015), pp. 125–139.
- J. Oñativia, S. Schultz, and P. L. Dragotti. “A Finite Rate of Innovation Algorithm for Fast and Accurate Spike Detection from Two-Photon Calcium Imaging”. *Journal of Neural Engineering* 10.4 (July 2013), pp. 1–14.

### Peer-reviewed conferences

- S. Reynolds, J. Oñativia, C. S. Copeland, S. R. Schultz, and P. L. Dragotti. “Spike Detection Using FRI Methods and Protein Calcium Sensors: Performance Analysis and Comparisons”. *2015 International Conference on Sampling Theory and Applications (SampTA)*. May 2015, pp. 533–537.
- J. Oñativia, Y. M. Lu, and P. L. Dragotti. “Sparsity Pattern Recovery Using FRI Methods”. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’15 — Brisbane, Australia)*. Apr. 2015.
- J. Oñativia, Y. M. Lu, and P. L. Dragotti. “Finite Dimensional FRI”. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’14 — Florence, Italy)*. May 2014, pp. 1808–1812.
- P. L. Dragotti, J. Oñativia, J. A. Urigüen, and T. Blu. “Approximate Strang-Fix: Sampling infinite streams of Diracs with any kernel”. *Proceedings SPIE 8858, Wavelets and Sparsity XV*. vol. 8858. Sept. 2013, 88580Y–88580Y–8.
- J. Oñativia, J. A. Urigüen, and P. L. Dragotti. “Sequential Local FRI Sampling of Infinite Streams of Diracs”. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’13 — Vancouver, Canada)*. May 2013, pp. 5440–5444.

### Conference abstracts

- J. Oñativia, P. L. Dragotti, and Y. M. Lu. “Sparsity According to Prony, average performance analysis”. *SPARS Workshop 2015 (Cambridge, UK)*. July 2015.

- J. Oñativia and P. L. Dragotti. “Sparsity According to Prony: From Structured to Unstructured Representations and Back”. *International BASP Frontiers workshop (Villars-sur-Ollon, Switzerland)*. Jan. 2015.
- J. Oñativia and P. L. Dragotti. “Finite dimensional FRI for reconstruction of sparse signals”. *UCL-Duke workshop on Sensing and Analysis of High-Dimensional Data (London, UK)*. Sept. 2014.
- S. Schultz, J. Oñativia, J. A. Urigüen, and P. L. Dragotti. “A Finite Rate of Innovation Algorithm for Spike Detection from Two-Photon Calcium Imaging”. *Neuroscience annual meeting (New Orleans, USA)*. Oct. 2012.



## Chapter 2.

# From classical sampling theory to sampling signals with finite rate of innovation

The world is analogue but computation is digital. The process that bridges this gap is known as the sampling process and has been instrumental to the digital revolution of the past 60 years. Without the sampling process we could not convert real-life signals in digital form and without digital samples we could not use computers for digital computation. The sampling process is also ubiquitous in that it is present in any mobile phone or digital camera but also in sophisticated medical applications like magnetic resonance imaging or ultrasound machines, in sensor networks and in digital microscopes just to name a few examples. Our understanding of the conversion of continuous-time signal in discrete form has been heavily influenced by the Shannon-Whittaker-Kotelnikov sampling theorems [13–16]. This approach restricts the class of signals that can be sampled and perfectly reconstructed to bandlimited signals. During the past few years, a new framework has emerged that overcomes these limitations and extends sampling theory to a broader class of signals named signals with Finite Rate of Innovation (FRI) [17]. Instead of characterising a signal by its frequency content, FRI theory describes it in terms of the innovation parameters per unit of time. As we shall see, bandlimited signals are a subset of this more general definition. In this chapter, we first revisit the classical sampling theory. We then introduce the theory of FRI signals and show how to sample and perfectly reconstruct classes of band unlimited signals. We also present extensions to this framework that deal with the noisy scenario, that is, when samples are corrupted with noise. In this case perfect reconstruction cannot be achieved, however, robust algorithms exist that perform reliable reconstructions of the original signal.

### 2.1. Classical sampling theory

The conversion of real world signals into streams of samples that can be stored or transmitted has been central to communications and signal processing. Under certain conditions, a continuous-time signal can be completely reconstructed from the knowledge of its values at points equally spaced in time. This surprising property follows from a result that is

referred to as the *sampling theorem* [18, Chapter 7]. In 1949, Shannon published one of his seminal papers, “Communications in the presence of noise” [13] where he stated the following sampling theorem:

**Theorem 1** (Shannon, 1949 [13]). *If a function  $x(t)$  contains no frequencies higher than  $W$  cycles per second, it is completely determined by giving its ordinates at a series of points spaced  $1/2W$  seconds apart.*

As mentioned in the paper, this was common knowledge in the *communication art*. He called the quantity  $1/2W$  the *Nyquist interval* in recognition to Nyquist’s work in communication theory during the 1920s [19, 20]. If we denote by  $f_s = 2W$  the sampling rate measured in Hz, the Nyquist interval  $T = 1/f_s$  corresponds to the sampling period in seconds. In the ideal bandlimited case, we can obtain an unambiguous discrete-time representation of the signal by just storing its values every  $T$  seconds:

$$x[n] = x(t)|_{t=nT}. \quad (2.1)$$

From samples  $x[n]$ , we can perfectly reconstruct the original signal as follows:

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t}{T} - n\right), \quad (2.2)$$

where  $\operatorname{sinc}(t) \stackrel{\text{def}}{=} \sin(\pi t)/\pi t$  is the cardinal sine function or sinc function. In his paper, Shannon provides a concise proof for the sampling theorem in terms of a Fourier series expansion. If the signal is bandlimited, we can write its time domain expression in terms of the inverse Fourier transform as follows

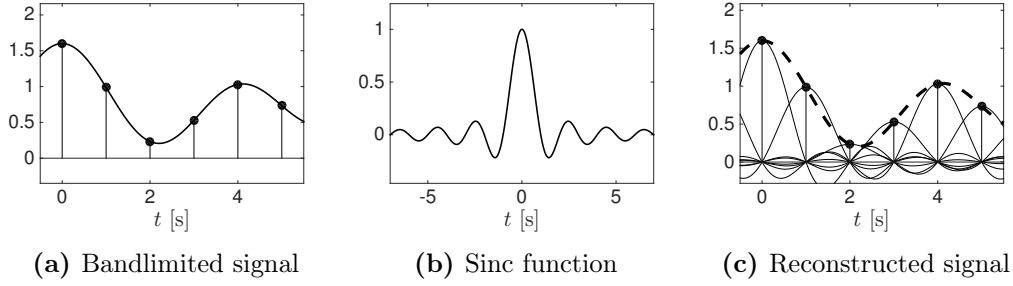
$$x(t) = \frac{1}{2\pi} \int_{-2\pi W}^{+2\pi W} \hat{x}(\omega) e^{i\omega t} d\omega, \quad (2.3)$$

where we have used the fact that the Fourier transform of  $x(t)$  is zero outside  $[-2\pi W, +2\pi W]$  since  $x(t)$  is, by hypothesis, bandlimited. If we sample the analogue signal at regular intervals of time  $T = 1/2W$  we obtain

$$x[n] = x(t)|_{t=n/2W} = \frac{1}{2\pi} \int_{-2\pi W}^{+2\pi W} \hat{x}(\omega) e^{i\omega n/2W} d\omega \stackrel{(a)}{=} \int_{-W}^W \hat{x}(2\pi\nu) e^{i\frac{2\pi\nu n}{2W}} d\nu, \quad (2.4)$$

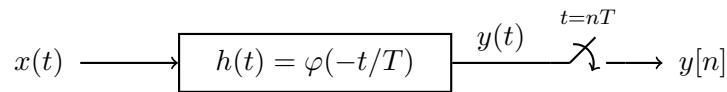
where (a) follows from the change of variable  $\omega = 2\pi\nu$ . This expression corresponds to the Fourier series coefficients of a  $2W$ -periodic version of  $\hat{x}(2\pi\nu)$  up to a scaling factor  $2W$ . The sequence  $x[n]$  is therefore a sufficient representation of  $\hat{x}(\omega)$  in the interval  $\omega \in [-2\pi W, +2\pi W]$ , and therefore also determines completely the function  $x(t)$ . Figure 2.1 illustrates the sampling and reconstruction process of a bandlimited signal.

This approach does not apply strictly to real world signals since it is well known that for



**Figure 2.1.:** The bandlimited signal  $x(t)$  in (a) is sampled at regular intervals of time (black dots) leading to the discrete-time signal  $x[n]$ . The reconstruction is performed using the sinc function (b). The signal is reconstructed by weighting shifted versions of the sinc function with the discrete-time signal  $x[n]$  (c).

a function to be bandlimited it must have infinite time duration [21]. Moreover, in practice Shannon's reconstruction formula is rarely used due to the slow decay of the sinc function. If  $x(t)$  is not bandlimited, prefiltering with an ideal lowpass filter ( $h(t) = \text{sinc}(t/T)$ ) and reconstructing applying (2.2) provides a lowpass approximation of  $x(t)$ . This is the best approximation in the least squares sense of  $x(t)$  in the space spanned by  $\{\text{sinc}(t/T - n)\}_{n \in \mathbb{Z}}$  [16]. However, it is an approximation, and perfect reconstruction of the original signal is not achieved. Moreover, the ideal lowpass filter is not realisable. The acquisition process where the continuous-time signal is prefiltered before taking samples at regular intervals of time  $T$  is illustrated in Figure 2.2. We refer the reader to Unser's paper [16] for a comprehensive review of classical sampling theory with a Hilbert space formulation. A different interpretation of the classical sampling theory is to say that the space of all bandlimited functions is spanned by  $\{\text{sinc}(t/T - n)\}_{n \in \mathbb{Z}}$ .



**Figure 2.2.:** Acquisition process. The continuous-time signal  $x(t)$  is filtered with  $h(t)$  in the analogue domain and then sampled at regular intervals of time  $t = nT$ . If the filter can be expressed as  $h(t) = \varphi(-t/T)$  the discrete-time samples are given by  $y[n] = \langle x(t), \varphi(t/T - n) \rangle$ .

### 2.1.1. The aliasing effect

The last step of the acquisition process, that is, sampling the continuous-time signal  $y(t)$  at regular intervals of time  $T$ , can also be modelled by multiplying  $y(t)$  with an impulse-train signal  $p(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$  [18]:

$$y_p(t) = y(t) p(t) = \sum_{n=-\infty}^{+\infty} y(nT) \delta(t - nT). \quad (2.5)$$

The Dirac delta function  $\delta(t)$  is useful in making the transition from real variable functions to discrete sequences. A Dirac is an idealisation of a pulse whose duration is reduced to zero (it is equal to zero everywhere except at  $t = 0$ ) with an integral over the real line equal to unity. It was introduced by Paul Dirac in the context of theoretical physics to represent a point mass or a point charge [22]. Strictly speaking it is not a function since its integral should be equal to zero. However, symbolic calculations with Diracs simplify notations and are justified by the theory of distributions (see [23, Appendix A.7]). We can see  $y_p(t)$  as the continuous-time representation of the discrete-time signal  $y[n] = y(nT)$ . This function is useful to understand what happens in the frequency domain when we discretise a continuous-time function. The Fourier transform of the impulse-train signal is given by

$$\hat{p}(\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{+\infty} \delta(\omega - k\omega_s), \quad (2.6)$$

where  $\omega_s = 2\pi f_s = 2\pi/T$ . Multiplying two functions in time is equivalent to convolving their Fourier transforms in the frequency domain and applying a factor of  $1/2\pi$ . Therefore,

$$\hat{y}_p(\omega) = \frac{1}{2\pi} \hat{y}(\omega) * \hat{p}(\omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \hat{y}(\omega - k\omega_s). \quad (2.7)$$

That is, the spectrum of the Dirac train  $y_p(t)$  is a periodic repetition of  $\hat{y}(\omega)$  with period  $\omega_s$  and scaled by  $1/T$ . If the frequency content of  $\hat{y}(\omega)$  is limited to  $\omega \in [-W, +W]$  and  $W < \omega_s/2$ , the conditions of the sampling theorem are satisfied, and the original signal can be recovered by low-pass filtering  $y_p(t)$ , which is equivalent to applying the reconstruction formula given in (2.2). However, if  $W > \omega_s/2$  the replicas  $\hat{y}(\omega - k\omega_s)$  overlap and  $y(t)$  is no longer recoverable by lowpass filtering. This effect is referred to as *aliasing*.

Let  $y(t) \in L^2(\mathbb{R})$  be a finite energy signal and  $y_p(t)$  be given as in (2.5). The relation between the discrete-time Fourier transform of the signal  $y[n] = y(t)|_{t=nT}$ ,

$$Y(e^{i\omega}) = \mathcal{F}\{y[n]\} = \sum_{n \in \mathbb{Z}} y[n] e^{-i\omega n}, \quad (2.8)$$

and the Fourier transform of  $y_p(t)$ , is given by

$$Y(e^{i\omega}) = \hat{y}_p\left(\frac{\omega}{T}\right), \quad (2.9)$$

since the Fourier transform of  $y_p(t)$  can also be written as

$$\hat{y}_p(\omega) = \int_{-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} y[n] \delta(t - nT) e^{-i\omega t} dt = \sum_{n=-\infty}^{+\infty} y[n] e^{-i\omega nT}. \quad (2.10)$$

We assume that the sequence  $y[n]$  is of finite energy, that is  $y[n] \in \ell^2(\mathbb{Z})$ , which allows

us to swap the order of the summation and the integral in the previous equation without having convergence issues. It follows that the aliasing effect is directly reflected in the discrete-time Fourier transform of  $y[n]$ :

$$Y(e^{i\omega}) = \mathcal{F}\{y[n]\} = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \hat{y}\left(\frac{\omega - 2\pi k}{T}\right). \quad (2.11)$$

The signal  $y(t)$  is obtained by filtering the input signal  $x(t)$  with  $h(t)$ . In the frequency domain, we have

$$\hat{y}(\omega) = \hat{x}(\omega) \hat{h}(\omega). \quad (2.12)$$

If the input signal is not limited in frequency to  $[-W, W]$ , the aliasing effect can be reduced having a filter  $h(t)$  that attenuates all frequencies  $|\omega| > W$ . Thus the name of anti-aliasing filter.

### 2.1.2. Acquisition process

Consider the typical acquisition process illustrated in Figure 2.2. This is usually modelled as a filtering stage followed by a sampling stage. The filter accounts for the modifications that the analogue signal  $x(t)$  experiences before being sampled. It may model an anti-aliasing filter or it might be due to the distortion introduced by the acquisition device, for example, in the case of a digital camera the distortion due to the lens. If the filter is expressed in terms of a scaled and time reversed function  $\varphi(t)$ , we have that filtering signal  $x(t)$  with  $h(t) = \varphi(-t/T)$  and retrieving samples at instants of time  $t = nT$  is equivalent to computing the inner product between  $x(t)$  and  $\varphi(t/T - n)$ . Specifically, the filtered signal is given by

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau) h(t - \tau) d\tau = \int_{-\infty}^{+\infty} x(\tau) \varphi\left(-\frac{t - \tau}{T}\right) d\tau. \quad (2.13)$$

Moreover, sampling  $y(t)$  at regular intervals of time  $t = nT$  leads to

$$y[n] = y(t)|_{t=nT} = \int_{-\infty}^{+\infty} x(\tau) \varphi\left(\frac{\tau}{T} - n\right) d\tau = \left\langle x(t), \varphi\left(\frac{t}{T} - n\right) \right\rangle. \quad (2.14)$$

The function  $\varphi(t)$  is called the sampling kernel. The classical sampling approach does not achieve perfect reconstruction of the original signal  $x(t)$  unless it is a perfectly bandlimited signal. However, in the next section we present some classes of non-bandlimited signals where perfect reconstruction is achievable when the input signal and the sampling kernel satisfy some conditions.

We tend to forget that the Shannon sampling theorem provides sufficient but not necessary conditions for perfect reconstruction. In other words, this theorem does not claim that it is not possible to sample and reconstruct classes of non-bandlimited signals. It is

therefore incorrect to assume that the bandwidth of a signal is related to its information content. A first attempt to reconcile these two notions: sampling rate and information content, was made in 2002 by Vetterli et al. [17]. Here, they introduced a new class of signals called signals with finite rate of innovation (FRI) which includes both bandlimited signals as well as non-bandlimited functions. They showed that classes of FRI signals can be sampled and perfectly reconstructed using an appropriate acquisition device. The literature presents a variety of scenarios where perfect reconstruction of FRI signals is achieved with different types of kernels. The sampling scheme presented in Equations (2.13) and (2.14), and illustrated in Figure 2.2, is therefore extensible to modern sampling schemes such as FRI theory.

## 2.2. Signals with Finite Rate of Innovation

Consider now a new class of signals that extend the one in (2.2):

$$x(t) = \sum_{k \in \mathbb{Z}} \sum_{r=0}^R a_{r,k} g_r(t - t_k), \quad (2.15)$$

where  $\{g_r(t)\}_{r=0}^R$  is a set of known functions. We note that, since  $g_r(t)$  are known, signals in (2.15) are uniquely determined by the set of parameters  $a_{r,k}$  and  $t_k$ . We denote these parameters as degrees of freedom. Let  $C_x(t_a, t_b)$  be a counting function that counts the number of degrees of freedom in  $x(t)$  over the interval  $[t_a, t_b]$ . For  $x(t)$  defined as in (2.15) we have that

$$C_x(t_a, t_b) = (R + 2) |\{t_k \in [t_a, t_b]\}|, \quad (2.16)$$

where  $|\{t_k \in [t_a, t_b]\}|$  corresponds to the number of  $t_k$  that belong to the interval  $[t_a, t_b]$ . In other words, there are  $R + 2$  degrees of freedom per index  $k$ :  $R + 1$  amplitudes  $a_{r,k}$  and one location  $t_k$ . We define the rate of innovation  $\rho$  as follows [17, 24, 25]:

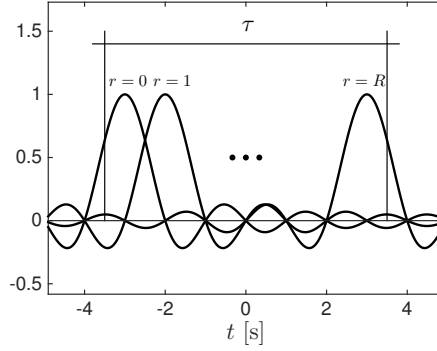
$$\rho = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} C_x\left(-\frac{\tau}{2}, +\frac{\tau}{2}\right). \quad (2.17)$$

We can now give a proper definition of FRI signals:

**Definition 1** (FRI signals [17]). *A signal with finite rate of innovation is a signal whose parametric representation is given in (2.15) and with a finite  $\rho$ , as defined in (2.17).*

It is of interest to note that bandlimited signals fall under this definition. Let  $\tau$  be an odd multiple of the sampling period  $T$ , that is,  $\tau = NT$  for some odd integer  $N$ . For a given time interval  $[-\tau/2, +\tau/2]$ , let

$$g_r(t) = \text{sinc}\left(\frac{t}{T} - \left(r - \frac{R}{2}\right)\right), \quad r = 0, 1, \dots, R, \quad (2.18)$$



**Figure 2.3.:**  $g_r(t)$  functions from equation (2.18) for  $r = 0$ ,  $r = 1$  and  $r = R$ . These functions cover the entire temporal interval  $[-\tau/2, +\tau/2]$ . The  $R$  parameter is assumed to be even in order to have a sinc function located at the origin for  $r = R/2$ ,  $R/2$  replicas of the sinc function in the negative half of the temporal interval and another  $R/2$  in the positive half.

be the set of  $R + 1$  known functions, where we assume that  $R$  is an even integer that satisfies  $(R + 1) = \tau/T$ . This set of functions is illustrated in Figure 2.3 where we can see that they cover the entire temporal interval of duration  $\tau$ . The following function

$$x_\tau(t) = \sum_{r=0}^R a_r g_r(t), \quad (2.19)$$

has  $R + 1$  degrees of freedom in  $[-\tau/2, +\tau/2]$  due to the unknown amplitudes  $a_r$ . Note that in this context we do not have the degree of freedom that corresponds to the temporal delay  $t_k$ , and therefore

$$C_x(-\tau/2, \tau/2) = R + 1. \quad (2.20)$$

If we set  $a_r = x[n]|_{n=(r-R/2)}$  and make  $\tau$  tend to infinity, the function  $x_\tau(t)$  coincides with the expression for bandlimited functions given in (2.2). By considering that  $R + 1 = \tau/T$ , we have that

$$\rho = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} (R + 1) = \frac{1}{T}, \quad (2.21)$$

and conclude that bandlimited functions have a rate of innovation of  $1/T$ . Therefore, one possible interpretation is that we can reconstruct them because they have a finite rate of innovation (rather than because they are bandlimited). We can also define a *local* rate of innovation with respect to a sliding window:

$$\rho_\tau(t) = \frac{1}{\tau} C_x\left(t - \frac{\tau}{2}, t + \frac{\tau}{2}\right). \quad (2.22)$$

In this case, we are usually interested in the maximum local rate of innovation:

$$\rho_{max}(\tau) = \max_{t \in \mathbb{R}} \rho_\tau(t). \quad (2.23)$$

Note that, for bandlimited functions and provided that  $\tau$  is equal to a multiple of  $T$ , the local, maximum and global rates of innovation are equal and have a value of  $1/T$ .

### 2.2.1. Examples of FRI signals

From now on we are going to focus our attention on signals that are not bandlimited and therefore cannot be sampled and perfectly reconstructed within the classical sampling framework. Examples of FRI signals which are not bandlimited and that are of interest to us include

- *Stream of pulses*:  $x(t) = \sum_k a_k p(t - t_k)$ . For instance, stream of decaying exponentials:

$$x(t) = \sum_k a_k e^{-(t-t_k)/\tau} \mathbf{1}_{t \geq t_k}, \quad (2.24)$$

which are a good fit for calcium transient signals induced by neural activity in two-photon calcium imaging. Here  $\mathbf{1}_{t \geq t_k}$  is zero for  $t < t_k$  and 1 otherwise. This is the type of signal that we are going to deal with in the application of FRI theory in neuroscience presented in Chapter 4. Figure 2.4(a) and 2.4(b) are examples of such signals.

- *Stream of Diracs* (see Figure 2.4(c)):

$$x(t) = \sum_k a_k \delta(t - t_k). \quad (2.25)$$

The Dirac delta function is defined by the effect it has when applied to another function:  $\int_{-\infty}^{+\infty} \delta(t) f(t) dt = f(0)$ . This integral satisfies the same properties as usual integrals such as change of variables and integration by parts. This allows us to represent a stream of pulses in terms of a stream of Diracs as follows:

$$\sum_k a_k p(t - t_k) = p(t) * \sum_k a_k \delta(t - t_k). \quad (2.26)$$

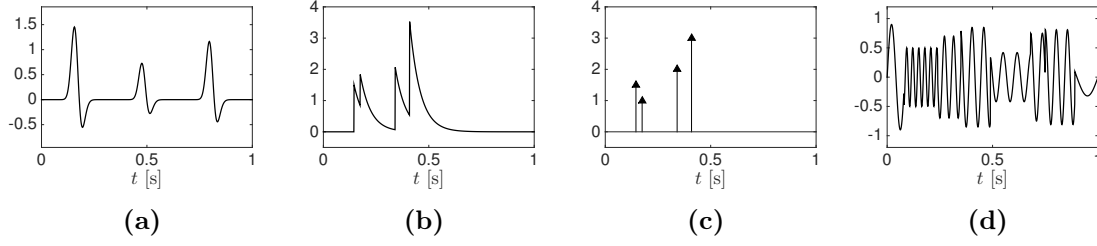
- *Piecewise sinusoidal signals* (see Figure 2.4(d)):

$$x(t) = \sum_k \sum_r a_{k,r} e^{i(\omega_{k,r}t + \varphi_{k,r})} \mathbf{1}_{[t_k, t_{k+1})}(t). \quad (2.27)$$

This is another example of not bandlimited signal and FRI theory has successfully been applied to this type of signals in [26].

- *Piecewise polynomial signals*:

$$x(t) = \sum_{k=0}^{\infty} \sum_{r=0}^R a_{k,r} (t - t_k)_+^r, \quad (2.28)$$



**Figure 2.4.:** Examples of signals with FRI. When the shape of the pulse is known the signal depends only on the amplitude and location of such pulses.

where  $t_+^r = \max(t, 0)^r$ . This type of model is particularly interesting when we are dealing with piecewise smooth signals and is another example of signal that can be perfectly recovered using FRI theory.

The literature presents different strategies to sample and perfectly reconstruct these signals. These strategies are based on imposing some conditions on the sampling kernels and then applying some specific reconstruction algorithms to recover the free parameters. Next, we present some of the sampling kernels that are used in the FRI theory.

## 2.3. Sampling kernels

FRI theory shows that it is possible to sample and perfectly reconstruct not bandlimited signals. In order to achieve this goal, different reconstruction schemes are based on different sampling kernels. We present here the main ones.

- *Infinite support kernels:* sinc and Gaussian ( $h_\sigma(t) = e^{-t^2/2\sigma^2}$ ). The original paper on FRI theory [17] presents a perfect reconstruction strategy to determine the locations and amplitudes of a stream of Diracs based on these two kernels. However, these two kernels have infinite support and this fact leads to reconstruction algorithms that are potentially unstable.
- *Polynomial and exponential reproducing kernels.* An extension to the original framework based on kernels that are able to reproduce polynomial and exponential functions was presented in [24]. A kernel  $\varphi(t)$  is said to reproduce polynomials of maximum degree  $P$  if, together with its shifted versions, it satisfies

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t - n) = t^m, \quad m = 0, 1, \dots, P. \quad (2.29)$$

Similarly, a kernel  $\varphi(t)$  is said to reproduce exponentials when there exists a linear combination of  $\varphi(t)$  with its integer shifts that reproduces functions of the form  $e^{\alpha_m t}$ , that is,

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t - n) = e^{\alpha_m t}, \quad m = 0, 1, \dots, P. \quad (2.30)$$

These kernels normally have compact support. The exponential reproducing kernels are of particular interest, especially when the parameters  $\alpha_m$  are purely imaginary. This leads to the most stable reconstruction schemes in the presence of noise [27]. For this reason, the work in the following sections and chapters focuses mostly on this type of kernels.

- *Rational kernels:* Finally, the authors in [24] also extended the FRI framework to kernels with a rational Fourier transform of the form

$$\hat{\varphi}(\omega) = \frac{\prod_{n=0}^Q (i\omega - b_n)}{\prod_{m=0}^P (i\omega - a_m)}, \quad Q < P. \quad (2.31)$$

This class of kernels is quite general and includes, for instance, the transfer function of any linear electric circuit.

The family of kernels that are able to reproduce a set of functions are of particular interest since many FRI reconstruction algorithms are based on this property. It is therefore important to provide a formal definition of this property:

**Definition 2** (Reproduction [28]). *A family of functions  $\{f_n(t)\}_{n \in \mathbb{Z}}$  reproduces a function  $f(t)$  if and only if there exists a sequence  $(c_n)_{n \in \mathbb{Z}}$  such that the equality*

$$f(t) = \sum_{n \in \mathbb{Z}} c_n f_n(t) \quad (2.32)$$

*holds almost everywhere.*

### 2.3.1. Strang-Fix conditions and splines for polynomial reproduction

The study of functions that are able to reproduce other functions stems from different areas such as approximation theory in numerical analysis or the finite element method in engineering. In 1971, Strang and Fix established the conditions that the Fourier transform of a function has to satisfy in order to reproduce polynomials of order up to  $P$ . These are the so-called Strang-Fix conditions [29]:

**Proposition 1** (Strang-Fix conditions). *A function  $\varphi(t)$  is able to reproduce polynomials, that is,*

$$t^m = \sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t - n), \quad m = 0, 1, \dots, P, \quad (2.33)$$

*if and only if*

$$\hat{\varphi}(0) \neq 0 \quad \text{and} \quad \hat{\varphi}^{(m)}(2\pi\ell) = 0 \quad \text{for} \quad \begin{cases} \ell \in \mathbb{Z} \setminus \{0\}, \\ m = 0, 1, \dots, P. \end{cases} \quad (2.34)$$

The superscript  $(m)$  stands for the  $m$ th derivative of  $\hat{\varphi}(\omega)$  and for  $m = 0$  we adopt the following convention:  $\hat{\varphi}^{(0)}(\omega) = \hat{\varphi}(\omega)$ .

A versatile family of functions that satisfy these conditions are the basis splines or B-splines, term that was coined by Schoenberg [30]. In general, splines are piecewise polynomial functions of a certain maximum degree  $P$ . The extremal points of the intervals that define the pieces are called knots. At these knots, the splines also present some degree of smoothness: they are continuous as well as their derivatives up to order  $P - 1$ . These functions were developed for mathematical approximation and interpolation and knew a great success for modelling automobile bodies or smooth surfaces in the aeronautic industry in the 1950s and 1960s (see Ferguson's work published while he was at Boeing [31], De Boor's work while at General Motors [32] as well as De Casteljau's work at Citroën and Bézier's at Renault which lead to the Bézier curves that are widely used in computer graphics). In the 1990s polynomial splines became popular in the signal processing community thanks to their fit within the wavelet framework and their application in image processing [33–35]. B-splines are a type of splines of minimal support for a given degree of smoothness. In 1946, Schoenberg proved that any polynomial spline of degree  $P$  with knots at the integers could be represented as a linear combination of shifted B-splines, thus the name of basis splines or B-splines [36].

Consider the following function:

$$\beta_0(t) = \begin{cases} 1, & 0 \leq t < 1, \\ 0, & \text{otherwise,} \end{cases} \quad \xrightarrow{\mathcal{F}} \quad \hat{\beta}_0(\omega) = \frac{1 - e^{-i\omega}}{i\omega}. \quad (2.35)$$

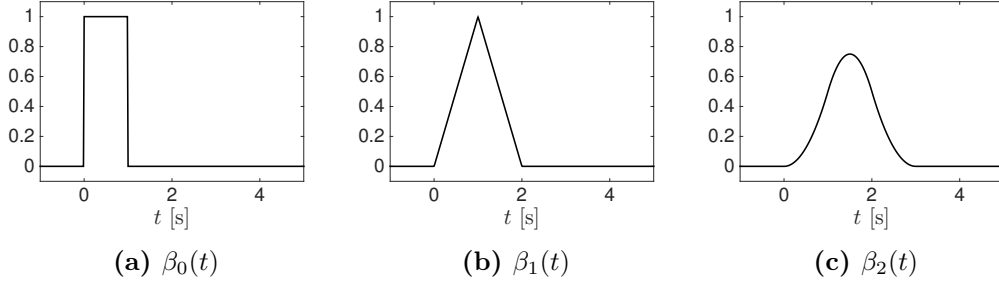
This is the zero order B-spline. It is easy to verify that it satisfies the Strang-Fix conditions given in (2.34) with  $P = 0$ . Hence,  $\beta_0(t)$  is able to reproduce zero order polynomials, that is, constant functions. This is pretty obvious since  $\beta_0(t)$  corresponds to a unity width box function, and therefore, by setting the weights  $c_n$  applied to each shifted  $\beta_0(t - n)$  equal to the amplitude of the constant function to be reproduced, equation (2.32) is perfectly satisfied. Higher order B-splines are constructed by convolving recursively lower order ones, that is,

$$\beta_P(t) = \beta_{P-1}(t) * \beta_0(t). \quad (2.36)$$

Convolving two functions in the time domain corresponds to computing their product in the frequency domain. This leads to a closed-form expression for the Fourier transform of the  $P$ th order B-spline:

$$\hat{\beta}_P(\omega) = \hat{\beta}_{P-1}(\omega) \hat{\beta}_0(\omega) = \left( \frac{1 - e^{-i\omega}}{i\omega} \right)^{P+1}. \quad (2.37)$$

We can also verify that this function satisfies the Strang-Fix conditions of order  $P$ , and therefore,  $\beta_P(t)$  can reproduce polynomials of order up to  $P$ . B-splines of order 0, 1 and



**Figure 2.5.:** B-splines of orders  $P = 0, 1, 2$ . Note that the support of a B-spline of order  $P$  is  $[0, P + 1]$ .

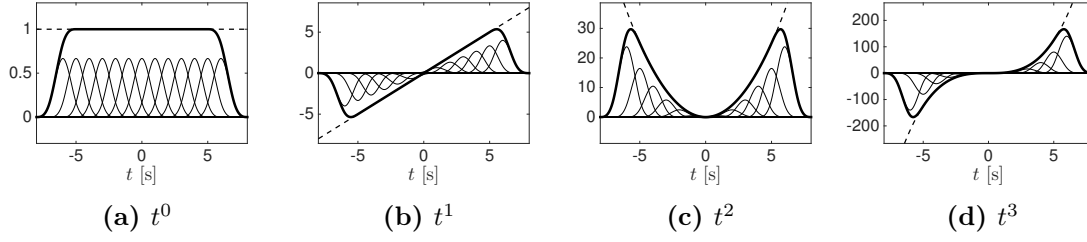
2 are shown in Figure 2.5. As the order increases,  $\beta_P(t)$  is smoother since higher order derivatives are also continuous. Moreover, it has been shown that the B-splines converge to a shifted and scaled Gaussian function as  $P \rightarrow \infty$  [37]. Note that the support of the B-splines increase with their order. We have that  $\text{supp}(\beta_P(t)) = [0, P + 1]$ , where

$$\text{supp}(f(t)) \stackrel{\text{def}}{=} \{t \in \mathbb{R} : f(t) \neq 0\}. \quad (2.38)$$

Sampling kernels of compact support are of particular interest in our scenario since they lead to a finite number of non-zero samples  $y[n]$  if the continuous-time signal that is being acquired is also of compact support.

In order to reproduce a specific polynomial  $t^m$  we have to compute the  $c_{m,n}$  coefficients that correspond to the weights that are applied to the shifted versions of the kernel  $\varphi(t - n)$  (see Equation (2.29)). Note that the index  $m$  refers to the polynomial that is reproduced and the index  $n$  to the specific shift of the kernel. The steps to compute these coefficients are detailed in the Appendix A.1. In practice, the summation in (2.29) is truncated in order to cover a given temporal interval. The reproduction of the polynomial is thus not satisfied for the entire real line. However, since the support of the kernel is compact, there is a temporal region where the reproduction is exact. Figure 2.6 illustrates the reproduction of polynomials up to order  $P = 3$  with a B-spline of order  $P = 3$ . Since the summation over the index  $n$  has been truncated, we can appreciate these border effects near the beginning and the end of the temporal interval of interest. It can easily be verified that if we are considering B-splines of support  $[0, P + 1]$  and we restrict the summation in the polynomial reproduction formula to indices  $n \in \{n_0, n_0 + 1, \dots, n_0 + N - 1\}$ , the polynomials are perfectly reproduced in the temporal interval  $[n_0 + P, n_0 + N]$ .

The amplitude of the coefficients  $c_{m,n}$  is directly linked to the amplitude of the polynomials in the temporal region of interest, and therefore, become very large when we move away from the origin. As it will become clear in Section 2.4, these coefficients play a crucial role in the reconstruction process of a sampled FRI signal. Having  $c_{m,n}$  coefficients that become very large leads to numerical instabilities and this effect penalises heavily the performance of reconstruction algorithms in the presence of noise. It is therefore preferable to



**Figure 2.6.:** Polynomial reproduction with B-spline of order  $P = 3$ . This function is able to reproduce polynomials of order up to  $P = 3$ . The thin lines represent the shifted and weighted kernels, the thick line represents their sum and the dashed line the true polynomials. The border effects are due to the fact that we truncate the summation in (2.29).

consider reproducing functions that are bounded in order to improve performances. This leads to the family of kernels that are presented in the next subsection: the exponential reproducing kernels.

### 2.3.2. Generalised Strang-Fix conditions and splines for exponential reproduction

The notion of reproduction of functions can also be extended to the exponential case, that is, we assume that linear combinations of shifted versions of a given function  $\varphi(t)$  are able to reproduce exponential functions as in (2.30). The exponential functions to be reproduced can have arbitrary complex parameters  $\alpha_m \in \mathbb{C}$ , however, we will restrict our presentation to the purely imaginary case, that is,  $\alpha_m = i\omega_m$  where  $\omega_m \in \mathbb{R}$ . Besides simplifying derivations, having purely imaginary parameters leads to the most resilient to noise FRI reconstruction schemes.

Strang-Fix conditions can be extended to the exponential case leading to the generalised Strang-Fix conditions [28, 38]. Here, we provide a formulation that was presented by Urig  en et al. in [27] and is more suited to our context:

**Proposition 2** (Generalised Strang-Fix conditions). *A function  $\varphi(t)$  is able to reproduce exponential polynomials, that is,*

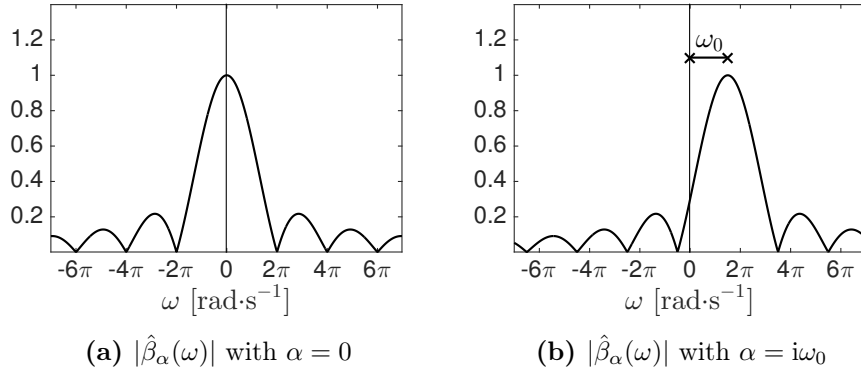
$$t^r e^{i\omega_m t} = \sum_{n \in \mathbb{Z}} c_{m,n,r} \varphi(t - n), \quad (2.39)$$

*if and only if*

$$\hat{\varphi}(\omega_m) \neq 0 \quad \text{and} \quad \hat{\varphi}^{(r)}(\omega_m + 2\pi\ell) = 0, \quad (2.40)$$

*for  $\ell \in \mathbb{Z} \setminus \{0\}$ ,  $r = 0, 1, \dots, R_m$  and  $m = 0, 1, \dots, P$ .*

The parameters  $R_0, R_1, \dots, R_P$ , are some positive integers. If we restrict to the case where all  $R_m = 0$ , we obtain the conditions that a kernel has to satisfy to reproduce purely exponential functions as in (2.30). The proof follows from considering a function  $\psi(t)$  that



**Figure 2.7.:** Absolute value of the Fourier transform of zero order E-splines given by (2.41). If the parameter  $\alpha$  is equal to zero, the E-spline corresponds to the zero order B-spline and the Fourier transform corresponds to a sinc function. For  $\alpha = i\omega_0$  purely imaginary, the Fourier transform of the E-spline is a shifted version of the sinc function.

satisfies the Strang-Fix conditions for the polynomial case and by multiplying this function by  $e^{i\omega_m t}$ . This condition can easily be extended to the more general case where  $\alpha \in \mathbb{C}$  by imposing conditions similar to (2.40) to the bilateral Laplace transform of  $\varphi(t)$ . A direct consequence of the generalised Strang-Fix conditions is that distinct  $\omega_m$  parameters have to satisfy  $\omega_n - \omega_m \neq 2\pi\ell$ ,  $\ell \in \mathbb{Z}$ . If this condition is not satisfied the zeros introduced by one of the parameters  $\omega_m$  in  $\hat{\varphi}(\omega_m + 2\pi\ell)$  will make the condition  $\hat{\varphi}(\omega_n) \neq 0$  fail for the other parameter  $\omega_n$ .

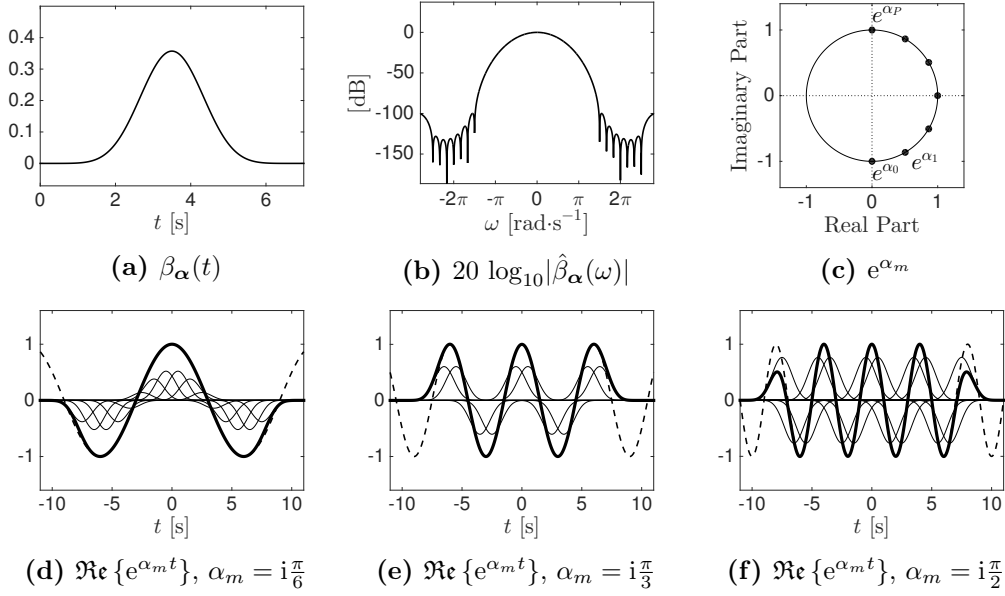
Similarly to the polynomial case, there exists a family of functions that originate from spline theory that are well suited for exponential reproduction, these are the exponential B-splines [39, 40] or E-splines [24]. Consider the following function:

$$\beta_\alpha(t) = \begin{cases} e^{\alpha t}, & 0 \leq t < 1, \\ 0, & \text{otherwise,} \end{cases} \quad \xrightarrow{\mathcal{F}} \quad \hat{\beta}_\alpha(\omega) = \frac{1 - e^{\alpha - i\omega}}{i\omega - \alpha}. \quad (2.41)$$

This function satisfies the generalised Strang-Fix conditions, and therefore, is able to reproduce the exponential function  $e^{\alpha t}$ . This corresponds to the zero order E-spline with parameter  $\alpha$  and can also be written as  $\beta_\alpha(t) = e^{\alpha t} \beta_0(t)$ . For purely imaginary parameter  $\alpha = i\omega_0$ , this can be seen as a modulated zero-order B-spline. Figure 2.7 illustrates this function for two different values of the parameter  $\alpha$ . Note that for  $\alpha = 0$ , the E-spline corresponds to the zero order B-spline.

As for the polynomial case, higher order E-splines can be built by convolving zero order ones. That is, an E-spline that can reproduce a set of exponential functions  $\{e^{i\omega_m t}\}_{m=0}^P$  is given by

$$\beta_\alpha(t) = \beta_{i\omega_0}(t) * \beta_{i\omega_1}(t) * \dots * \beta_{i\omega_P}(t), \quad (2.42)$$



**Figure 2.8.:** Exponential reproduction with E-spline with  $\alpha = [-i\frac{\pi}{2}, -i\frac{\pi}{3}, -i\frac{\pi}{6}, 0, i\frac{\pi}{6}, i\frac{\pi}{3}, i\frac{\pi}{2}]$ . The first row represents the E-spline in the temporal and frequency domain and the parameters  $e^{\alpha_m}$ . The second row depicts the reproduction of the real part of the exponentials  $e^{\alpha_m t}$ . The thin lines represent the shifted and weighted kernels, the thick line represents their sum and the dashed line the true exponential functions.

where  $\alpha = (i\omega_0, i\omega_1, \dots, i\omega_P)$ . It follows that the Fourier transform of  $\beta_\alpha(t)$  is given by:

$$\hat{\beta}_\alpha(w) = \prod_{m=0}^P \frac{1 - e^{-i(\omega - \omega_m)}}{i(\omega - \omega_m)}, \quad (2.43)$$

which satisfies the generalised Strang-Fix conditions with  $R_m = 0$ . So far, we have assumed that all  $\omega_m$  are distinct which leads to E-splines that reproduce exponential functions  $e^{i\omega_m t}$ . If we want to reproduce polynomial exponentials  $t^r e^{i\omega_m t}$  up to a given degree  $R_m$  it is necessary to repeat each coefficient  $\omega_m$   $R_m$  times.

Figures 2.8(a) and (b) depict an E-spline of order  $P = 6$  in the temporal and frequency domains respectively. Since the parameters  $\alpha_m = i\omega_m$  considered are purely imaginary, the different  $e^{\alpha_m}$  lie on the unit circle as illustrated in Figure 2.8(c). The Fourier transform of the E-spline shows that the generalised Strang-Fix conditions are satisfied, that is, the function is non-zero at the frequencies  $\omega_m$ , but vanishes at the frequencies  $\omega_m + 2\pi\ell$  with  $\ell \in \mathbb{Z} \setminus \{0\}$ .

E-splines of order  $P$  have also compact support and have continuous derivatives up to order  $P - 1$ . The construction of higher order E-splines through convolution is justified by the following proposition:

**Proposition 3** (Unser and Blu, 2005 [41]). *Let  $\varphi_\alpha$  be a function that reproduces the exponential polynomials in  $\text{span}\{e^{\alpha t}, \dots, t^p e^{\alpha t}\}$ . Then, for any  $\varphi$  such that  $\int_{-\infty}^{+\infty} \varphi(t) e^{-\alpha t} dt \neq$*

0, the composite function  $\varphi(t) * \varphi_\alpha(t)$  also reproduces these exponential polynomials.

Moreover, there is a strong relation between exponential reproducing functions and E-splines since it can be shown that any compactly supported function  $\varphi(t)$  that reproduces exponential polynomials satisfy

$$\varphi(t) = \gamma(t) * \beta_\alpha(t), \quad (2.44)$$

where  $\beta_\alpha(t)$  is the corresponding E-spline that reproduces the same exponential polynomials [42].

In the same way as in the reproduction of polynomials, in order to reproduce a specific exponential function  $e^{\alpha_m t}$  we have to obtain the coefficients  $c_{m,n}$  that correspond to the weights that are applied to the shifted versions of the kernel  $\varphi(t-n)$ . The steps to compute these coefficients are detailed in Appendix A.2. Figures 2.8(a), (b) and (c) illustrate the reproduction of some exponential functions. Since the parameters  $\alpha_m$  are complex, the resulting function is also complex:  $e^{i\omega_m t} = \cos(\omega_m t) + i\sin(\omega_m t)$ ; the figures only depict the real parts.

## 2.4. Sampling and perfect reconstruction of FRI signals

In the previous section we have seen some properties of exponential reproducing kernels. In this section we analyse how we can make use of these properties in order to perfectly reconstruct streams of Diracs. Streams of Diracs are the canonical FRI signals because many phenomena can be modelled as a pulse shape convolved with a stream of Diracs. Hence, the framework presented to sample and reconstruct streams of Diracs can easily be extended to a wide range of different scenarios.

### 2.4.1. Sampling with an exponential reproducing kernel

We consider an acquisition device where the sampling kernel is able to reproduce exponential functions  $\{e^{i\omega_m t}\}_{m=0}^P$ . When we have control over the acquisition device, the parameters  $\alpha_m = i\omega_m$  are part of the design problem. As will become clear in what follows, these parameters will determine the information in the input analogue signal  $x(t)$  that we can retrieve. Specifically, the different  $\omega_m$  correspond to the frequencies of the Fourier transform of  $x(t)$  that we will be able to retrieve from knowledge only of the samples  $y[n]$ . We assume that the signal  $x(t)$  is localised, in other words, that  $x(t)$  has compact support. Moreover, if we assume that  $\text{supp}(x(t)) = [0, L)$ , and that the transfer function of the acquisition device is also of compact support with  $h(t) = \varphi(-t/T)$  and  $\text{supp}(\varphi(t)) = (-P, 0]$ , from (2.14), we have that the number of non-zero samples  $y[n]$  is finite. Specifically,  $y[n] \neq 0$  for  $n = 0, 1, \dots, N-1$  where  $N = \lceil L/T \rceil + P$ .

Let  $s[m]$ ,  $m = 0, 1, \dots, P$ , be a new sequence obtained by linearly combining samples  $y[n]$  with the coefficients  $c_{m,n}$  from (2.30), that is,

$$s[m] = \sum_{n=0}^{N-1} c_{m,n} y[n], \quad m = 0, 1, \dots, P, \quad (2.45)$$

where  $N$  is the number of non-zero samples  $y[n]$ . If we make use of the fact that the sampling kernel is able to reproduce exponentials, it follows that

$$s[m] \stackrel{(a)}{=} \sum_{n=0}^{N-1} c_{m,n} \langle x(t), \varphi(t/T - n) \rangle \quad (2.46)$$

$$\stackrel{(b)}{=} \int_0^L x(t) \sum_{n=0}^{N-1} c_{m,n} \varphi(t/T - n) dt \quad (2.47)$$

$$\stackrel{(c)}{=} \int_0^L x(t) e^{i\omega_m t/T} dt = \hat{x} \left( -\frac{\omega_m}{T} \right), \quad (2.48)$$

where (a) follows from (2.14), (b) from the linearity of the inner product and the fact that  $\text{supp}(x(t)) = [0, L]$  and (c) from the exponential reproduction property. The quantity  $s[m]$  therefore corresponds to the Fourier transform of  $x(t)$  evaluated at  $\omega = -\omega_m/T$ .

#### Remark on generalised Strang-Fix conditions and alias-free sampling

At this point, it might seem that we are facing a contradiction, since we are retrieving exact Fourier information of a signal at some frequencies but we are not imposing the classical sampling theorem's conditions that the input signal is bandlimited or that the acquisition filter is an ideal lowpass filter. The only condition we have imposed is that the sampling kernel be able to reproduce exponentials. In Figure 2.8(b) we can observe that an E-spline that reproduces the exponentials  $e^{i\omega_m t}$  is not bandlimited. Yet due to the generalised Strang-Fix conditions, it is possible to retrieve uncorrupted Fourier information at the frequencies that the sampling kernel reproduces. In what follows we show that the generalised Strang-Fix conditions are essentially anti-aliasing conditions at specific frequencies.

From (2.11) and (2.12) it follows that

$$Y(e^{i\omega}) = \mathcal{F}\{y[n]\} = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \hat{x} \left( \frac{\omega - 2\pi k}{T} \right) \hat{h} \left( \frac{\omega - 2\pi k}{T} \right). \quad (2.49)$$

Note that  $x(t)$  and  $h(t)$  are of compact support in the temporal domain, and therefore both functions are unlimited in frequency, which produces aliasing in  $Y(e^{i\omega})$ . The sampling kernel  $\varphi(t)$  and the impulse response of the acquisition device  $h(t)$  are related by  $h(t) =$

$\varphi(-t/T)$ . We thus have that  $\hat{h}(\omega) = T \hat{\varphi}(-\omega T)$ , and

$$Y(e^{i\omega}) = \sum_{k=-\infty}^{+\infty} \hat{x}\left(\frac{\omega - 2\pi k}{T}\right) \hat{\varphi}(2\pi k - \omega). \quad (2.50)$$

Appendix A.2 shows that the  $c_{m,n}$  coefficients can be expressed as  $c_{m,n} = e^{i\omega_m n} / \hat{\varphi}(\omega_m)$ . If we plug this expression in (2.48) we have that

$$s[m] = \sum_{n=0}^{N-1} c_{m,n} y[n] = \frac{1}{\hat{\varphi}(\omega_m)} \sum_{n=0}^{N-1} y[n] e^{i\omega_m n} = \frac{1}{\hat{\varphi}(\omega_m)} Y(e^{i\omega})|_{\omega=-\omega_m}. \quad (2.51)$$

Replacing the expression of  $Y(e^{i\omega})$  obtained in (2.50) into equation (2.51) yields

$$s[m] = \frac{1}{\hat{\varphi}(\omega_m)} \sum_{k=-\infty}^{+\infty} \hat{x}\left(\frac{2\pi k - \omega_m}{T}\right) \hat{\varphi}(\omega_m - 2\pi k). \quad (2.52)$$

From the generalised Strang-Fix conditions (see Proposition 2) we know that if  $\varphi(t)$  is an exponential reproducing kernel that reproduces the functions  $e^{i\omega_m t}$  then  $\hat{\varphi}(\omega_m + 2\pi k) = 0$  for  $k = \mathbb{Z} \setminus \{0\}$ . This is the condition that allows us to recover exact Fourier information of the original signal  $x(t)$  at these specific frequencies without having any aliasing. In fact, by using Strang-Fix conditions in (2.52) we obtain:

$$s[m] = \hat{x}\left(-\frac{\omega_m}{T}\right). \quad (2.53)$$

We can thus conclude that, despite the aliasing effect in  $Y(e^{i\omega})$ , we can obtain uncorrupted information of the Fourier transform of  $x(t)$  at some specific frequencies thanks to the fact that the sampling kernel satisfies the generalised Strang-Fix conditions.

### 2.4.2. Reconstruction of a stream of Diracs

We assume that the input signal is a stream of  $K$  Diracs:

$$x(t) = \sum_{k=1}^K a_k \delta(t - t_k). \quad (2.54)$$

The signal is filtered and sampled at regular intervals of time which leads to the discrete-time signal  $y[n]$ . The continuous-time signal  $x(t)$  is perfectly determined by the pairs  $\{(a_k, t_k)\}_{k=1}^K$ . We assume that all the locations are distinct and that the amplitudes are non-zero. Thus, the goal is to estimate this set of parameters from samples  $y[n]$ . We consider the case where the sampling kernel  $\varphi(t)$  satisfies the exponential reproduction property for a choice of  $\boldsymbol{\alpha} = (\alpha_m)_{m=0}^P$  such that  $\alpha_m = i\omega_m$ , where  $\omega_m \in \mathbb{R}$  for  $m = 0, 1, \dots, P$ . We further impose the frequencies  $\omega_m$  to be equispaced, that is  $\omega_{m+1} - \omega_m = \lambda$ .

We can thus express the frequencies as

$$\omega_m = \omega_0 + m \lambda. \quad (2.55)$$

Since  $x(t)$  is a sum of Diracs, we have that the Fourier transform is given by a sum of exponentials:

$$\hat{x}(\omega) = \int_{-\infty}^{+\infty} \sum_{k=1}^K a_k \delta(t - t_k) e^{-i\omega t} dt = \sum_{k=1}^K a_k e^{-i\omega t_k}. \quad (2.56)$$

This is clearly a band unlimited signal. The signal  $x(t)$  in the temporal domain and its Fourier transform are illustrated in Figures 2.9(a) and (e).

We now consider the sequence  $s[m]$  that is obtained by linearly combining samples  $y[n]$  with the coefficients  $c_{m,n}$  from the exponential reproducing property (2.30). From (2.48) we have that  $s[m] = \hat{x}(-\omega_m/T)$  and therefore:

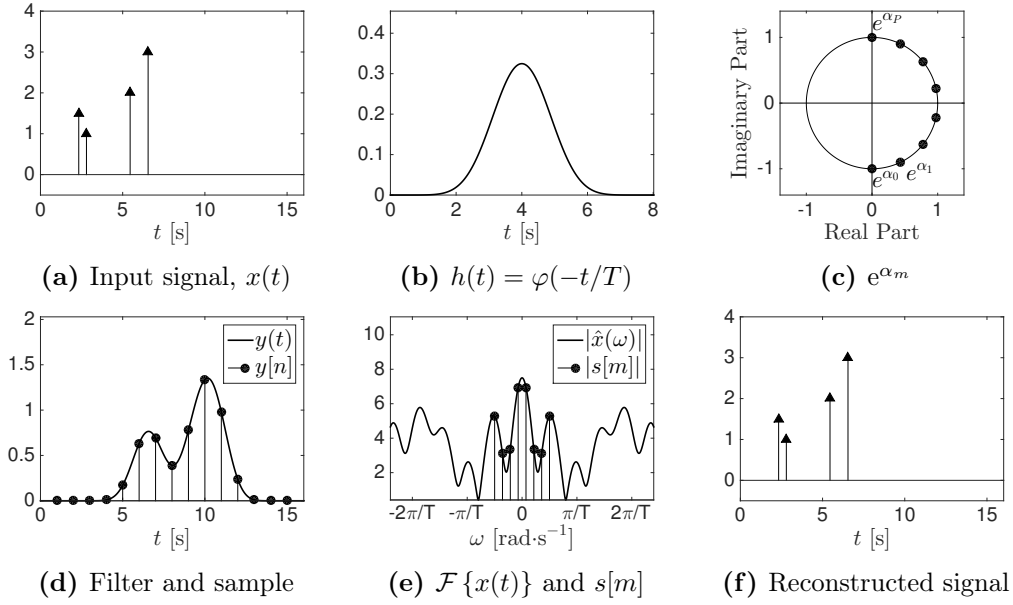
$$s[m] = \sum_{k=1}^K a_k e^{i\omega_m t_k/T} = \sum_{k=1}^K \underbrace{a_k e^{i\omega_0 t_k/T}}_{b_k} \left( \underbrace{e^{i\lambda t_k/T}}_{u_k} \right)^m = \sum_{k=1}^K b_k u_k^m, \quad (2.57)$$

where  $b_k \stackrel{\text{def}}{=} a_k e^{i\omega_0 t_k/T}$  and  $u_k \stackrel{\text{def}}{=} e^{i\lambda t_k/T}$ . Note that we have also used the fact that the frequencies can be expressed as  $\omega_m = \omega_0 + \lambda m$ . The perfect recovery of the original stream of Diracs, that is, the estimation of the locations  $t_k$  and the amplitudes  $a_k$  of the  $K$  Diracs, is now recast as the estimation of parameters  $b_k$  and  $u_k$  from the knowledge of values  $s[m]$ . The problem of estimating the parameters of a sum of exponentials from a set of samples arises in a variety of fields and has been analysed for several years by the spectral estimation community [43]. It arises in a variety of applications such as identifying an auto-regressive system from its output [44], estimating exponentially damped sinusoids in noise [45, 46], estimating the direction of arrival of signals in array processing [47–49] and reconstructing polygons from its moments [50, 51]. The problem is linear in the parameters  $b_k$ , but is nonlinear in the parameters  $u_k$ . The main difficulty is in finding the nonlinear terms. There are two main approaches to solve this problem, one based on homogeneous recurrence relations, known as Prony's method, and the other based on signal subspace estimation techniques, which is solved using matrix pencils.

### Prony's method

One way to solve the problem of estimating  $\{(b_k, u_k)\}_{k=1}^K$  from  $s[m]$ , is by realising that the sequence  $s[m]$  given as in (2.57) is the solution to the following linear homogeneous recurrence relation

$$s[m] + h[1] s[m-1] + \dots + h[K] s[m-K] = 0. \quad (2.58)$$



**Figure 2.9.:** Sampling and perfect reconstruction of a stream of Diracs. (a) is the continuous-time stream of Diracs, (b) the sampling kernel  $h(t) = \varphi(-t/T)$  where  $\varphi(t)$  is an E-spline of order  $P = 7$  that reproduces the exponentials illustrated in (c). (d) is the continuous-time signal  $y(t) = x(t) * h(t)$  and the corresponding discrete samples  $y[n] = y(t)|_{t=nT}$ . In (e), absolute value of the Fourier transform of  $x(t)$  and of the sequence  $s[m]$  obtained from samples  $y[n]$  linearly combined with coefficients  $c_{m,n}$ . (f) perfect reconstruction of the Diracs from the sequence  $s[m]$ .

This approach is referred to as Prony's method or the annihilating filter method. Baron de Prony developed this method in 1795 to estimate the frequency, phase and amplitudes of a finite sum of sinusoids [52]. The name of annihilating filter method comes from the fact that the coefficients  $(1, h[1], \dots, h[K])$  can be seen as a finite impulse response filter with  $K+1$  taps that annihilates the sequence  $s[m]$ , that is, the output of this filter vanishes when the input signal corresponds to  $s[m] = \sum_{k=1}^K b_k u_k^m$ . Note that the coefficients  $(h[m])_{m=1}^K$  are still unknown. In order to estimate the coefficients of the annihilating filter, we assume that it has zeros at  $z = u_k$ , that is, its z-transform satisfies

$$H(z) = \mathcal{Z}\{h[m]\} = \sum_{m=0}^K h[m] z^{-m} = \prod_{k=1}^K (1 - u_k z^{-1}), \quad (2.59)$$

where  $h[0] = 1$ . Note that  $H(z)|_{z=u_k} = 0$ . Filtering the sequence  $s[m]$  with  $h[m]$  corresponds to convolving both sequences in the temporal domain. If we plug the expression of  $s[m]$  obtained in (2.57) into the discrete convolution formula we obtain

$$h[m] * s[m] = \sum_{\ell=0}^K h[\ell] s[m-\ell] = \sum_{\ell=0}^K h[\ell] \sum_{k=1}^K b_k u_k^{m-\ell} = \sum_{k=1}^K b_k u_k^m \underbrace{\sum_{\ell=0}^K h[\ell] u_k^{-\ell}}_{H(z)|_{z=u_k}=0} = 0. \quad (2.60)$$

This convolution can also be written in matricial form as follows

$$\mathbf{0} = \begin{bmatrix} s[K] & s[K-1] & \dots & s[0] \\ s[K+1] & s[K] & \dots & s[1] \\ \vdots & \vdots & \ddots & \vdots \\ s[2K-1] & s[2K-2] & \dots & s[K-1] \end{bmatrix} \begin{bmatrix} 1 \\ h[1] \\ \vdots \\ h[K] \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{S} \mathbf{h}. \quad (2.61)$$

The matrix  $\mathbf{S}$  is a Toeplitz matrix of size  $K \times (K+1)$  built from  $2K$  consecutive values  $s[m]$ . It can be shown that if all  $b_k$  are non-zero and all  $u_k$  are distinct, the matrix  $\mathbf{S}$  has full row rank  $K$  (see Appendix B.1). Hence, the null space has dimension one and the coefficients of vector  $\mathbf{h}$  are unique. This vector can be obtained by taking the singular value decomposition (SVD) of  $\mathbf{S}$  and choosing the singular vector associated with the zero singular value. If we scale this singular vector in order to have the first element equal to 1 the solution is unique. The parameters  $u_k$  are then obtained from the roots of the polynomial

$$H(z) = 1 + \sum_{m=1}^K h[m] z^{-m} = z^{-K} (h[K] + h[K-1]z + \dots + h[1]z^{K-1} + z^K). \quad (2.62)$$

Note that the roots can be obtained numerically from the eigenvalues of the companion matrix

$$C(\mathbf{h}) = \begin{bmatrix} 0 & 0 & \dots & 0 & -h[K] \\ 1 & 0 & \dots & 0 & -h[K-1] \\ 0 & 1 & \dots & 0 & -h[K-2] \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -h[1] \end{bmatrix}, \quad (2.63)$$

since the characteristic polynomial of this matrix is equivalent to the polynomial  $z^K H(z)$ . Once the nonlinear parameters  $u_k$  are recovered, the amplitudes  $b_k$  of the exponentials can be obtained from (2.57) by solving  $K$  linear equations. The parameters that uniquely determine the stream of Diracs are then obtained as follows:

$$t_k = \frac{T \log u_k}{i\lambda} \quad \text{and} \quad a_k = b_k e^{-i\omega_0 t_k/T}, \quad \text{for } k = 1, \dots, K. \quad (2.64)$$

The perfect reconstruction of the stream of Diracs is therefore achieved. Note that we only require  $2K$  consecutive samples of the sequence  $s[m]$  to estimate the  $2K$  parameters. This establishes a minimum requirement on the order of the sampling kernel, since each sample  $s[m]$  is related to one of the exponential functions that are reproduced by  $\varphi(t)$ . Hence, in order to reconstruct  $K$  Diracs, the sampling kernel has to reproduce at least  $P+1 = 2K$  exponentials. The entire sampling and reconstruction process is illustrated in Figure 2.9.

### Matrix pencil method

An alternative method to retrieve the parameters  $\{(b_k, u_k)\}_{k=1}^K$  from the sequence  $s[m]$  is based on subspace techniques for estimating generalised eigenvalues of matrix pencils [45, 46]. Matrix pencils are the matricial counterpart of polynomials. For some nonnegative integer  $P$  and a set of square matrices  $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_P$  of the same size, the matrix pencil of order  $P$  is defined as

$$\mathbf{L}_P(x) = \mathbf{A}_0 + \mathbf{A}_1 x + \dots + \mathbf{A}_P x^P. \quad (2.65)$$

A variation of the first order matrix pencil that has the form  $\mathbf{A}_0 - x\mathbf{A}_1$  is known as linear matrix pencil. The approach that estimates the parameters of a sum of exponentials from a linear matrix pencil is based on the particular structure of Toeplitz matrices where each element is given by a sum of exponentials. We build a Toeplitz matrix  $\mathbf{S}$  from  $2K$  consecutive samples of the sequence  $s[m]$  similarly to (2.61), but of size  $(K+1) \times K$  (instead of  $K \times (K+1)$ ). Let  $\mathbf{S}_0$  be the matrix constructed from  $\mathbf{S}$  by removing the first row and  $\mathbf{S}_1$  the matrix constructed from  $\mathbf{S}$  by removing the last row. These two matrices are of size  $K \times K$  and have rank  $K$ . It can be shown that in the matrix pencil  $\mathbf{S}_0 - \mu\mathbf{S}_1$  the parameters  $\{u_k\}_{k=1}^K$  from (2.57) are rank reducing numbers, that is, the matrix  $\mathbf{S}_0 - \mu\mathbf{S}_1$  has rank  $K-1$  for  $\mu = u_k$  and rank  $K$  otherwise (see Appendix B.2). The parameters  $\{u_k\}_{k=1}^K$  are thus obtained from the eigenvalues of the generalised eigenvalue problem  $\mathbf{S}_0 \mathbf{v} = \mu \mathbf{S}_1 \mathbf{v}$ . This generalised eigenvalue problem can be solved by computing the eigenvalues of the matrix  $\mathbf{S}_1^{-1} \mathbf{S}_0$ , since

$$(\mathbf{S}_0 - \mu\mathbf{S}_1)\mathbf{v} = \mathbf{0} \quad \Leftrightarrow \quad (\mathbf{S}_1^{-1} \mathbf{S}_0 - \mu\mathbf{I})\mathbf{v} = \mathbf{0}. \quad (2.66)$$

Note that the matrix  $\mathbf{S}_1$  is of size  $K \times K$  and has rank  $K$ , hence it is invertible. In practice, the problem is not solved by finding the eigenvalues of  $\mathbf{S}_1^{-1} \mathbf{S}_0$  since this matrix might be ill-conditioned. Moreover, there are algorithms based on the Schur decomposition that compute the generalised eigenvalues without actually computing the inverse of  $\mathbf{S}_1$  (see Chapter 7 in [53] for a detailed description of such algorithms). The algorithm described in the next section is much more robust and the approach that is used in the following chapters when we refer to the matrix pencil method. Maravic and Vetterli presented an extension to the FRI theory using an approach based on this principle in 2005 [54].

## 2.5. Sampling FRI signals in the presence of noise

The acquisition process inevitably introduces noise, making the solutions described so far only ideal. Perturbations may arise in the analogue and digital domain. We model the noise of the acquisition process as a white Gaussian process that is added to the ideal

samples. The noisy samples are therefore given by

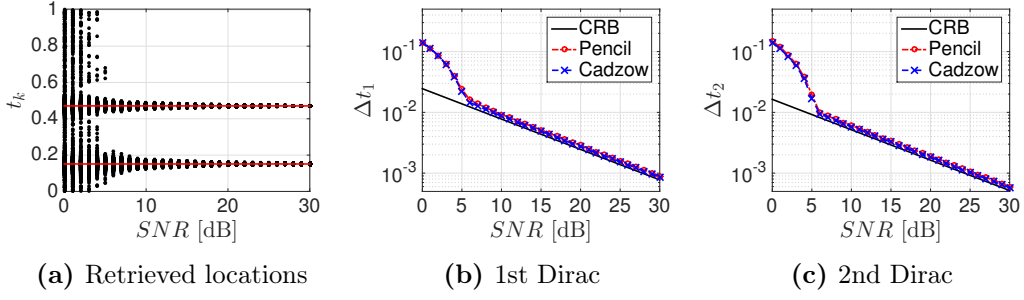
$$\tilde{y}[n] = y[n] + \varepsilon[n], \quad (2.67)$$

where  $y[n]$  are the ideal noiseless samples and  $\varepsilon[n]$  are independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance  $\sigma_\varepsilon^2$ . In order to have a more robust reconstruction, we can increase the number of samples  $s[m] = \sum_{n=0}^{N-1} c_{m,n} y[n]$  by designing a sampling kernel that reproduces more exponentials than the critical number  $P + 1 = 2K$ . The denoising strategies that can be applied to improve the performance of the reconstruction process come from the spectral analysis community, where the problem of finding sinusoids in noise has been extensively studied. From now on we assume that the length  $P + 1$  of the sequence  $s[m]$  is larger than  $2K$ . We can thus build a bigger Toeplitz matrix  $\mathbf{S}$ :

$$\mathbf{S} = \begin{bmatrix} s[M] & s[M-1] & \dots & s[0] \\ s[M+1] & s[M] & \dots & s[1] \\ \vdots & \vdots & \ddots & \vdots \\ s[P] & s[P-1] & \dots & s[P-M] \end{bmatrix}. \quad (2.68)$$

In the noiseless case, this matrix has still rank  $K$  provided that the number of rows and the number of columns are at least equal to  $K$ . Note that in the previous section we have only considered Toeplitz matrices of size  $K \times (K + 1)$  or  $(K + 1) \times K$ .

One approach is based on *cleaning* the samples  $s[m]$  before using Prony's method by applying an algorithm proposed by Cadzow for signal enhancement in 1988 [55]. It is an iterative procedure applied to the Toeplitz matrix constructed from samples  $s[m]$  as in (2.68). This enhancement approach works better when the Toeplitz matrix is made as square as possible, for instance, by setting  $M = \lceil P/2 \rceil$ . By construction the matrix  $\mathbf{S}$  is Toeplitz, and in the noiseless case it is of rank  $K$ . The presence of noise makes this matrix full rank. The Cadzow algorithm looks for the closest rank deficient matrix which is Toeplitz. At each step we force matrix  $\mathbf{S}$  to be of rank  $K$  by computing the singular value decomposition (SVD) and only keeping the  $K$  largest singular values and setting the rest to zero. This new matrix is not Toeplitz anymore, we thus compute a new Toeplitz matrix by averaging the diagonal elements. This last matrix might not be rank deficient and we can thus iterate again. After *cleaning* the sequence  $s[m]$ , the next step is to solve the homogeneous system (2.61). This is done by computing the total least squares solution that minimises  $\|\mathbf{S}\mathbf{h}\|^2$  subject to  $\|\mathbf{h}\|^2 = 1$ . If this vector is normalised with respect to the first element we have that the following  $K$  elements correspond to the coefficients of the annihilating filter  $H(z)$  in (2.59). Note that  $\mathbf{h}$  is of size  $K + 1$ , and therefore, when applying Prony's method we have to build a *tall* Toeplitz matrix  $\mathbf{S}$  of size  $(P - K + 1) \times (K + 1)$  from the previously cleaned sequence  $s[m]$ . We say the matrix is tall because now  $P + 1$  is



**Figure 2.10.:** Performance of noisy recovery of 2 Diracs from  $N = 22$  samples  $y[n]$  with an E-spline of order  $P = 10$ . For each level of noise 10 000 realisations of the noise have been generated. (a) Scatter plot of the retrieved locations, the horizontal lines represent the true locations of the Diracs. (b) and (c) depict the measured standard deviation of the retrieved locations for the “Cadzow+Prony” and matrix pencil algorithms compared to Cramér-Rao lower bounds.

larger than  $2K$  and therefore the matrix has more rows than columns. This approach has successfully been applied in the FRI setup in [25].

An alternative approach to the “Cadzow+Prony” method is based on making the matrix pencil method more resilient to noise. In the matrix pencil method we can also consider the bigger Toeplitz matrix built as in (2.68). In this *robustified* matrix pencil method, we also solve the generalised eigenvalue problem  $\mathbf{S}_0 \mathbf{v} = \mu \mathbf{S}_1 \mathbf{v}$ , but we build the matrices  $\mathbf{S}_0$  and  $\mathbf{S}_1$  in a slightly different way. Instead of building  $\mathbf{S}_0$  and  $\mathbf{S}_1$  directly from matrix  $\mathbf{S}$  we now build them from the  $K$  singular vectors of  $\mathbf{S}$  that correspond to the  $K$  non-zero singular values. It can be shown that the generalised eigenvalues of this equivalent problem are also  $u_1, \dots, u_K$ . In the noisy scenario, the Toeplitz matrix is full rank, and therefore all the singular values are non-zero. We therefore consider the singular vectors that correspond to the  $K$  largest singular values. These are the steps of the noisy matrix pencil algorithm:

1. Build the Toeplitz matrix  $\mathbf{S}$  from noisy samples  $s[m]$ .
2. Perform the SVD of  $\mathbf{S}$ :  $\mathbf{S} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^H$ .
3. Keep the  $K$  left-singular vectors that correspond to the  $K$  largest singular values:  $\mathbf{U}_K = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K]$ , where  $\mathbf{u}_k$  are the columns of  $\mathbf{U}$ .
4. Build  $\mathbf{S}_0$  and  $\mathbf{S}_1$  from  $\mathbf{U}_K$  by dropping the first and last rows respectively.
5. Solve the following eigenvalue problem:

$$(\mathbf{S}_1^+ \mathbf{S}_0 - \mu \mathbf{I}) \mathbf{v} = \mathbf{0}, \quad (2.69)$$

where  $\mathbf{S}_1^+$  is the Moore-Penrose pseudoinverse of  $\mathbf{S}_1$ .

### 2.5.1. Performance of the noisy reconstruction algorithms

In order to measure and compare the performance of the noisy recovery algorithms we run the following experiment. A stream of 2 Diracs is sampled with an E-spline of order  $P = 10$  which leads to the noiseless sequence  $y[n]$ . For each level of noise, we add a realisation of the noise samples,  $\varepsilon[n]$ , in order to satisfy a given signal-to-noise ratio (SNR). The SNR is the ratio between signal power and noise power measured in decibels (dB):

$$SNR = 10 \log_{10} \left( \frac{P_y}{P_\varepsilon} \right) = 10 \log_{10} \left( \frac{\frac{1}{N} \sum_{n=0}^{N-1} |y[n]|^2}{\frac{1}{N} \sum_{n=0}^{N-1} |\varepsilon[n]|^2} \right). \quad (2.70)$$

The Diracs are then retrieved by applying the two methods previously described: “Cad-zow+Prony” and matrix pencil. In order to compare the two methods, the experiment is repeated 10 000 times for each level of noise. Let  $\hat{t}_k^{(i)}$  be the estimated location of the  $k$ th Dirac for the  $i$ th realisation of the noise. We measure the accuracy of each method by computing the standard deviation of the retrieved locations compared to the true locations, that is,

$$\Delta t_k = \sqrt{\frac{\sum_{i=1}^I \left( t_k - \hat{t}_k^{(i)} \right)^2}{I}}, \quad (2.71)$$

where  $t_k$  is the true location of the Dirac and  $I$  is the total number of realisations per noise level; in our case  $I = 10\,000$ . The results are shown in Figure 2.10 where (a) depicts the retrieved locations for all the realisations of the noise samples (each dot represents one retrieved location in one realisation of the experiment), and (b) and (c) present a comparison between the measured accuracies for each method. Note that as the noise power decreases (the SNR increases) the dots of the scatter plot in (a) concentrate around the true locations of the Diracs which correspond to the two horizontal lines. The straight lines in (b) and (c) represent the Cramér-Rao lower bounds (CRB) of this estimation problem. The CRB provides a theoretical lower bound of the standard deviation of unbiased estimators [43, 56–58]. See Appendix C.2 for details about how to derive this bound for the FRI framework. From Figures 2.10(b) and (c) it is clear that both approaches perform very similarly. This is not surprising since both are based on performing a SVD of the Toeplitz matrix  $\mathbf{S}$  and estimating the signal by considering the  $K$  largest singular values. Moreover, both algorithms are close to being optimal since the precision that is achieved reaches the CRB, and present a breakdown for SNR below 5 dB. The breakdown happens when the noise power goes beyond a certain threshold that makes the singular values due to the noise be larger than the singular values of the signal.

It is obvious that the performance of this type of algorithms is influenced by the sampling kernel. For a given type of kernel that reproduces exponentials, the performance is affected by the order  $P$  and the frequencies  $\omega_m$  that are reproduced. Urigüen’s PhD thesis [59] provides a good analysis of how these parameters affect the performance of the FRI

reconstruction algorithms. His work also explored the scenario where we have no control over the sampling kernel and proposed universal reconstruction schemes to recover FRI signals [27].

### 2.5.2. On the effect of coefficients $c_{m,n}$ on the statistics of the noise

In the noisy case, the coefficients  $c_{m,n}$  play a crucial role as they will be responsible for *colouring* the noise. The effect of noise on the measurements  $s[m]$  can be analysed more precisely as follows:

$$\tilde{s}[m] = \sum_{n=0}^N c_{m,n} \tilde{y}[n] = \sum_{n=0}^N c_{m,n} y[n] + \sum_{n=0}^N c_{m,n} \varepsilon[n] = s[m] + \eta[m], \quad (2.72)$$

for  $m = 0, 1, \dots, P$ . The  $(\ell + 1, m + 1)$  element in the covariance matrix  $\mathbf{\Sigma}_{\varepsilon}$  of a vector of random variables  $\boldsymbol{\varepsilon} = [\varepsilon[0], \varepsilon[1], \dots, \varepsilon[N - 1]]^T$  is defined as

$$[\mathbf{\Sigma}_{\varepsilon}]_{\ell,m} \stackrel{\text{def}}{=} \mathbb{E} \{ (\varepsilon[\ell] - \mathbb{E} \{ \varepsilon[\ell] \}) (\varepsilon[m] - \mathbb{E} \{ \varepsilon[m] \}) \}. \quad (2.73)$$

Since the noise samples  $\varepsilon[n]$  are assumed to be i.i.d. random variables drawn from a normal distribution  $\mathcal{N}(0, \sigma_{\varepsilon}^2)$ , their covariance matrix is given by  $\mathbf{\Sigma}_{\varepsilon} = \sigma_{\varepsilon}^2 \mathbf{I}_N$ , where  $\mathbf{I}_N$  is the identity matrix of size  $N \times N$ . The noise term in the samples  $\tilde{s}[m]$  is given by  $\eta[m] = \sum_{n=0}^{N-1} c_{m,n} \varepsilon[n]$ . It is easy to verify that the expected value of these random variables is also equal to zero:

$$\mathbb{E} \{ \eta[m] \} = \mathbb{E} \left\{ \sum_{n=0}^{N-1} c_{m,n} \varepsilon[n] \right\} = \sum_{n=0}^{N-1} c_{m,n} \mathbb{E} \{ \varepsilon[n] \} = 0. \quad (2.74)$$

Due to the coefficients  $c_{m,n}$ , the terms  $\eta[m]$  are complex. Hence, we consider the covariance using complex conjugation:

$$[\mathbf{\Sigma}_{\eta}]_{\ell,m} = \mathbb{E} \{ (\eta[\ell] - \mathbb{E} \{ \eta[\ell] \}) (\eta[m] - \mathbb{E} \{ \eta[m] \})^* \}. \quad (2.75)$$

If we replace  $\eta[m]$  by  $\sum_{n=0}^{N-1} c_{m,n} \varepsilon[n]$  and consider that  $c_{m,n} = e^{i\omega_m n} c_{m,0}$ , with  $\omega_m = \omega_0 + m\lambda$ , we obtain

$$[\mathbf{\Sigma}_{\eta}]_{\ell,m} = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} c_{\ell,n} c_{m,k}^* \underbrace{\mathbb{E} \{ \varepsilon[n] \varepsilon[k] \}}_{=\sigma_{\varepsilon}^2 \delta[n-k]} = \sigma_{\varepsilon}^2 c_{\ell,0} c_{m,0}^* \sum_{n=0}^{N-1} e^{i(m-\ell)\lambda n}. \quad (2.76)$$

If we do not impose any other condition on the frequencies  $\omega_m$ , the covariance matrix of  $\eta[m]$  is non-zero outside the main diagonal, which implies that the noise samples are

correlated. To avoid this, we impose the frequency separation  $\lambda$  to be equal to

$$\lambda = \frac{2\pi}{N}. \quad (2.77)$$

In this case, (2.76) becomes

$$[\Sigma_\eta]_{\ell,m} = \begin{cases} N \sigma_\varepsilon^2 |c_{m,0}|^2, & \text{if } \ell = m, \\ 0, & \text{otherwise.} \end{cases} \quad (2.78)$$

The noise samples  $\eta[m]$ ,  $m = 0, 1, \dots, P$ , are now uncorrelated. However, note that the variance of each  $\eta[m]$  is equal to  $N \sigma_\varepsilon^2 |c_{m,0}|^2$ . Since  $c_{m,0} = [\hat{\varphi}(\omega_m)]^{-1}$ , these variances might differ considerably for different values of  $m$ . This can be seen as the noise power being different for different frequencies. This is the reason why we say that the noise is coloured. If we choose frequencies  $\omega_m$  where the Fourier transform of the sampling kernel is very small, the variance of the corresponding  $\eta[m]$  becomes very large, making the recovery algorithms unstable. In order to have an accurate reconstruction it is therefore important to pick frequencies where  $\hat{\varphi}(\omega)$  is as flat as possible. If the sampling kernel is an E-spline, this can only be achieved by concentrating all the frequencies around the origin. To overcome this limitation we can consider the use of more advanced sampling kernels such as the exponential MOMS (Maximal Order Minimum Support) or eMOMS [60]. eMOMS are a variation of the E-splines and are still able to reproduce exponential functions. One of the main advantages of this type of function is that we can control the amplitude of the Fourier transform at  $\omega = \omega_m$ . We can even impose  $\varphi(\omega_m) = 1$  for all  $m = 0, 1, \dots, P$ . Urig  en et al. introduced this type of kernel in the FRI framework in [61] (see Appendix D for more details about this type of functions).

## 2.6. Summary

This chapter revisits the classical sampling theorem for bandlimited signals and introduces the notion of a more general class of functions known as finite rate of innovation (FRI) signals. The theory for sampling and perfectly reconstructing FRI signals is presented with a special emphasis on the sampling kernels that are able to reproduce exponentials. This framework is also extended to the case where samples are corrupted with additive noise and the performance of the best FRI reconstruction algorithms in noise are analysed.



## Chapter 3.

# Sequential local FRI sampling of infinite streams of Diracs

The theory of sampling signals with finite rate of innovation (FRI) has shown that it is possible to perfectly recover classes of non-bandlimited signals such as streams of Diracs from uniform samples. Most of the papers in the FRI literature, however, have to some extent focused only on the sampling of periodic or finite duration signals.

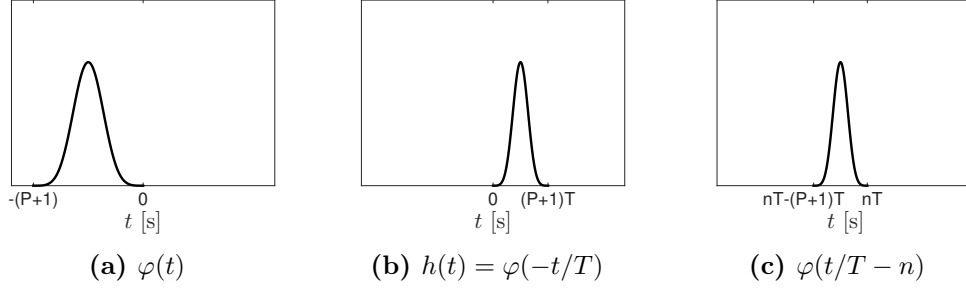
In this chapter we present a novel method that is able to reconstruct infinite streams of Diracs, even in high noise scenarios. We sequentially process the discrete samples and output locations and amplitudes of the Diracs in real-time. We first establish conditions for perfect reconstruction in the noiseless case and then present the sequential algorithm for the noisy scenario. We present simulation results that show that we can achieve a high reconstruction accuracy of 1000 Diracs for SNRs as low as 5dB. We also propose a different sampling scheme that takes advantage of the periodicity of the reproduced exponentials by the sampling kernel that allows us to reduce the number of required temporal samples.

### 3.1. Introduction

Streams of Diracs are the canonical example of signals with finite rate of innovation in that they are completely specified by a finite number of parameters per unit of time. Periodic streams of Diracs are sampled and perfectly reconstructed in [17] using the sinc kernel. Authors in [24, 62] instead propose the use of kernels that reproduce polynomials or exponentials and also propose a sequential algorithm to sample and perfectly reconstruct infinite streams of Diracs. The sequential algorithm, however, was designed to deal only with noiseless samples. The family of sum of sines (SoS) kernels was introduced in [63] for the sampling of periodic stream of pulses such as Diracs, authors also consider the case of infinite streams of Diracs. However, their method requires that the stream of Diracs be *bursty*. Specifically, a group of  $K$  Diracs must be followed by a long period of absence

---

<sup>0</sup>The work in this chapter includes research conducted jointly with Prof Dragotti and lead to the following publication [8]. The code to reproduce the results presented in this chapter is available online: <https://github.com/jonativia/thesis>.



**Figure 3.1.:** The impulse response of the acquisition device,  $h(t)$ , is given by  $h(t) = \varphi(-t/T)$ . In order to have a causal filter, the sampling kernel  $\varphi(t)$  is anticausal with support  $[-(P+1)T, 0]$ . Each sample  $y[n] = \langle x(t), \varphi(t/T - n) \rangle$  acquires information of  $x(t)$  on the temporal interval  $[nT - (P+1)T, nT]$ .

of Diracs in order for the method to work. They also assume that the reconstruction algorithm is synchronised with the sampling process in order to be automatically in phase with the time window containing the burst of Diracs.

In this chapter we present a novel approach to reconstruct infinite streams of Diracs, in high noise scenarios, with no clear separation between bursts. We sequentially process the discrete samples and output locations and amplitudes of the Diracs in real-time. We first establish conditions for perfect reconstruction in the noiseless case and then present the sequential algorithm for the noisy scenario. We show through simulations that the algorithm is able to process 10000 samples in about 100 seconds and that it can retrieve with high accuracy 1000 Diracs even in very low SNR regimes.

The chapter is organised as follows. In Section 3.2 we briefly review the case of sampling and reconstructing a finite stream of  $K$  Diracs as presented in the previous chapter. In Section 3.3 we explain our sequential algorithm in detail. We treat the noiseless and noisy scenarios separately, the former to establish conditions for perfect reconstruction, the latter for application in more realistic situations. We validate our algorithm with simulations in Section 3.4. In Section 3.5 we present an alternative sampling scheme that is able to reduce the number of required temporal samples to the critical value of  $N = 2K$  to estimate  $2K$  parameters. We summarise and conclude in Section 3.6.

## 3.2. Sampling FRI signals

We consider the case of a stream of  $K$  Diracs  $x(t) = \sum_{k=1}^K a_k \delta(t - t_k)$ , with  $a_k \in \mathbb{C} \setminus \{0\}$  and  $t_1 < t_2 < \dots < t_K$ . The continuous-time signal is filtered with a compact support kernel with impulse response  $h(t) = \varphi(-t/T)$  and uniformly sampled at regular intervals of time  $t = nT$ . The acquisition process was illustrated in Figure 2.2 and, as already

discussed in the previous chapter, the samples  $y[n]$  can be expressed as

$$y[n] = \left\langle x(t), \varphi\left(\frac{t}{T} - n\right) \right\rangle. \quad (3.1)$$

Figure 3.1(a) illustrates a sampling kernel  $\varphi(t)$  of compact support, the impulse response of the acquisition device  $h(t)$  is shown in Figure 3.1(b) and the scaled and shifted kernels  $\varphi\left(\frac{t}{T} - n\right)$  in Figure 3.1(c). When the sampling kernel is of compact support, each sample  $y[n]$  captures information about  $x(t)$  on the time interval which is covered by the support of the scaled and shifted kernel. In our setup, this interval corresponds to

$$t \in [nT - (P+1)T, nT] \quad (3.2)$$

as illustrated in Figure 3.1(c).

As described in the previous chapter, FRI theory shows that for a properly chosen filter  $h(t)$ , the signal  $x(t)$  can be perfectly reconstructed from the samples  $y[n]$ . We restrict our setup to exponential reproducing kernels which are functions that are able to reproduce exponentials:

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t - n) = e^{\alpha_m t}, \quad m = 0, 1, \dots, P, \quad (3.3)$$

where  $c_{m,n} \in \mathbb{C}$ . The  $\alpha_m$  are design parameters and are chosen to be purely imaginary and in complex conjugate pairs in order to have a real valued kernel  $\varphi(t)$ , that is,  $\alpha_m = i\omega_m$  with  $\omega_m \in \mathbb{R}$ . More specifically, we consider equispaced frequencies  $\omega_m$  that can be expressed as

$$\omega_m = \omega_0 + m\lambda. \quad (3.4)$$

Note that we require equispaced frequencies in order to apply Prony's method to the sequence  $s[m]$  introduced in the previous chapter and described in the next paragraph. Since the  $\alpha_m$  parameters appear in complex conjugate pairs, we have that

$$\omega_m = -\omega_{P-m}, \quad (3.5)$$

and it follows that

$$\omega_m = \lambda \left( m - \frac{P}{2} \right). \quad (3.6)$$

Note that if  $P$  is even we have that  $\alpha_{P/2} = i\omega_{P/2} = 0$ . There exist many functions of compact support in time that satisfy the exponential reproducing property (3.3), for instance, E-splines [24] and the modified E-splines introduced in [61]. The latter are the exponential reproducing kernels that are most resilient to noise [27, 61] and are the kernels of choice for our simulations. They are called eMOMS. Note that these kernels have support  $P+1$ . See Appendix D for details about how to construct these kernels and their relation with the Dirichlet kernel.

In order to recover the parameters  $\{(a_k, t_k)\}_{k=1}^K$ , we follow the procedure described in the previous chapter. First, the samples  $y[n]$  are linearly combined with coefficients  $c_{m,n}$  from (3.3). This leads to a new set of measurements

$$s[m] = \sum_{n \in \mathbb{Z}} c_{m,n} y[n], \quad m = 0, 1, \dots, P. \quad (3.7)$$

Combining (3.1) and (3.3) it follows that  $s[m]$  can be expressed as a power sum series [24]:

$$s[m] = \sum_{k=1}^K b_k u_k^m, \quad (3.8)$$

where  $b_k = a_k e^{i\omega_0 t_k/T}$  and  $u_k = e^{i\lambda t_k/T}$ . The unknown parameters are then estimated applying Prony's or matrix pencil methods. The critical number of measurements  $s[m]$  required to recover the  $2K$  parameters  $\{(a_k, t_k)\}_{k=1}^K$  is exactly  $2K$  [24]. It thus follows that the number of exponentials reproduced by the sampling kernel has to be  $P+1 \geq 2K$ .

### 3.3. Sampling an infinite stream of Diracs

We now consider the case of an infinite train of Diracs

$$x(t) = \sum_{k \in \mathbb{Z}} a_k \delta(t - t_k). \quad (3.9)$$

The signal  $x(t)$  is assumed to satisfy

$$\rho_{max}(\tau) = \max_{t \in \mathbb{R}} \rho_\tau(t) = \frac{2K}{\tau}, \quad (3.10)$$

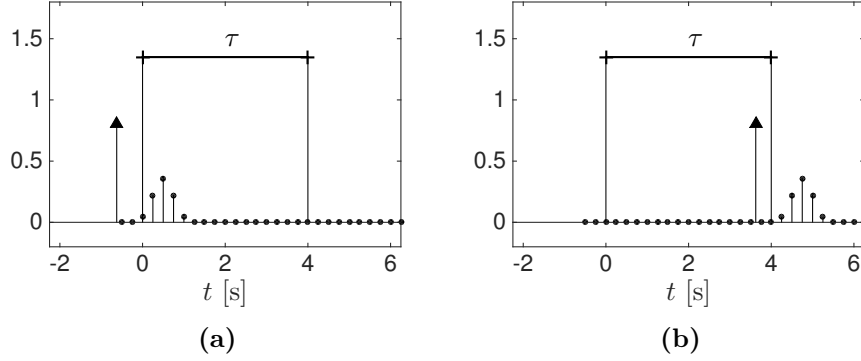
where  $\rho_{max}(\tau)$  is the the maximum local rate of innovation defined as in (2.23) and  $\tau > 0$ . This means that, if we consider a sliding window of size  $\tau$ , the number of Diracs that we see inside the window is always at most  $K$ . If we further assume that consecutive spikes have a minimum distance of  $\tau_0$ , that is,  $\tau_0 = \min_k t_{k+1} - t_k$ , it is easy to verify that

$$\rho_{max}(\tau) \leq \frac{4}{\tau_0}, \quad \text{if} \quad \tau \geq \tau_0. \quad (3.11)$$

We propose a sequential algorithm that estimates the locations of the Diracs of (3.9) by using a sliding window that sequentially covers the interval of time

$$t \in [n_i T, n_i T + \tau], \quad (3.12)$$

for  $n_i = 1, 2, \dots$ . The sliding window step is of  $T$  seconds, which equals the sampling period. We assume that  $\tau$  is an integer multiple of  $T$ , specifically,  $\tau = NT$ . In what



**Figure 3.2.:** Border effects. (a) A nearby Dirac located before the observation window  $\tau$  influences the samples  $y[n]$  of the window. (b) A Dirac inside the window but close to the right border generates non-zero samples outside the window.

follows, we first establish some conditions on the number of samples  $N$ , the sampling period  $T$  and the order of the E-splines necessary to achieve perfect reconstruction of the infinite stream (3.9). Second, we present a novel approach that is able to recover Diracs in high noise scenarios processing the stream (3.9) sequentially.

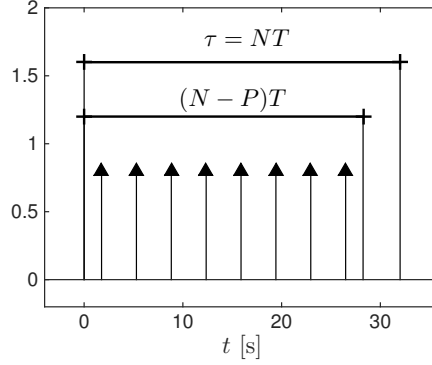
### 3.3.1. Noiseless case

We analyse the ideal scenario in the first place to determine the conditions on the number of samples of the sliding window  $N$ , the sampling period  $T$  and the support  $P + 1$  of the sampling kernel that allows our algorithm to be able to provide perfect reconstruction of (3.9). In our approach, we analyse sequentially sets of  $N$  samples  $y[n]$  that cover the temporal interval (3.12). This corresponds to samples with indices  $n = n_i + 1, \dots, n_i + N$ , since we assume that the support of the sampling kernel  $\varphi(t/T - n)$  is given by

$$\text{supp} \left( \varphi \left( \frac{t}{T} - n \right) \right) = [nT - (P + 1)T, nT]. \quad (3.13)$$

Note that due to the continuity of the sampling kernel  $\varphi(t)$ , at the extremal left point of the temporal interval  $t = n_i T$ , the sampling kernel with index  $n_i$  satisfies  $\varphi(t/T - n_i)|_{t=n_i T} = 0$ . Therefore, there is no need to consider the sample  $y[n]|_{n=n_i}$  since this sample does not capture any information of the interval  $[n_i T, n_i T + \tau]$ . This is why we consider samples starting at index  $n = n_i + 1$ .

Since we consider a finite number of samples, there exist border effects that may stop us from achieving perfect reconstruction. The sampling kernel  $\varphi(t/T)$  has compact support  $(P + 1)T$ . Consequently, a Dirac influences  $P + 1$  samples. This means that a Dirac located just before the window of interest will generate non-zero samples that will leak inside the window. Moreover, a Dirac located at the end of the window of interest will generate non-zero samples beyond the  $N$  samples we are considering, and therefore, cannot



**Figure 3.3.:** Perfect reconstruction interval. If all the Diracs are located in the temporal interval of size  $(N - P)T$  perfect reconstruction is achieved.

be reconstructed. This is illustrated in Figure 3.2.

We assume that the system is under the condition of initial rest. That is,  $x(t) = 0$  and  $y[n] = 0$  for  $t \leq t_0$  and  $n \leq n_0$  respectively for some  $t_0$  and  $n_0 = \lfloor t_0/T \rfloor$ . It follows that there is no *left* border effect for the first temporal window that includes the first non-zero sample  $y[n]$ . This border effect can be overcome in subsequent windows with the following reasoning. Since the algorithm operates sequentially, we can assume that when operating on the window  $t \in [n_i T, n_i T + \tau]$  we have already recovered Diracs up to the time instant  $n_i T$ . Therefore, their contributions can be removed which avoids the border effect on the *left* of the window.

To overcome the border effect of the *right* side we determine appropriate conditions on the number of samples  $N$ , the order  $P$  of the kernel and the sampling period  $T$ . The exact recovery of  $K$  Diracs requires  $P + 1 \geq 2K$ . Now, let us put ourselves in the worst case scenario, where we have  $K$  Diracs evenly spaced with constant separation  $\tau/K$ . As previously mentioned, when a Dirac is near the end of the interval, we are not able to perfectly reconstruct it. The size of this interval is  $PT$ . Therefore, we can only perfectly recover  $K$  Diracs when all of them are in a region of size  $(N - P)T$  which we refer to as the *perfect reconstruction interval* illustrated in Figure 3.3. In the case of constant separation  $\tau/K$ , we have to guarantee that there will be a position of the sliding window such that the  $K$  Diracs are in this perfect reconstruction interval. Since they can occupy an interval of maximum size  $(K - 1)\frac{\tau}{K}$  and to make sure they are within the perfect reconstruction interval for a certain window it follows that

$$(K - 1)\frac{\tau}{K} \leq (N - P - 1)T. \quad (3.14)$$

In order to maximise the perfect reconstruction interval we chose the smallest possible order for the sampling kernel, that is

$$P + 1 = 2K. \quad (3.15)$$

Replacing  $\tau$  by  $NT$  in (3.14) and imposing  $P + 1 = 2K$  we obtain

$$N \geq 2K^2. \quad (3.16)$$

It follows that

$$T \leq \frac{\tau}{2K^2}, \quad (3.17)$$

since  $T = \tau/N$  and  $N \geq 2K^2$ .

The reconstruction algorithm processes the stream of samples sequentially, retrieving the locations of each set of maximum  $K$  Diracs from  $N$  samples by applying the annihilating filter method. Provided we satisfy the previously described conditions, all Diracs will be located in the perfect reconstruction interval of a certain position of the sliding window, and thus recovered. From the recovered Diracs of the current window, we recalculate the  $N$  samples that correspond to this window, and only if the reconstructed samples are identical to the original ones, the Diracs are stored. The maximum number of Diracs  $K$  within a window has to be estimated. This is done by trying for all possible values of  $K$ , and only when the correct value is estimated the reconstructed samples will coincide with the original ones.

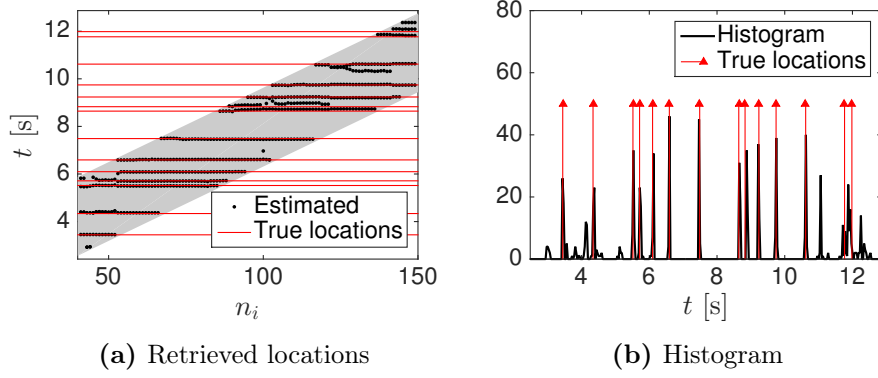
In the worst case scenario, Prony's method is applied  $K$  times to the  $N$  samples of the observed window. The overall cost of processing  $N$  samples to recover  $K$  Diracs is  $\mathcal{O}(K^4)$  as it is shown next. First, the sequence  $s[m]$  has to be built from the samples  $y[n]$ , which involves  $N$  products and additions. Then, an SVD is applied to the Toeplitz matrix of size  $K \times (K + 1)$  built from the sequence  $s[m]$  to obtain the coefficients of the annihilating filter. The complexity of this step is  $\mathcal{O}(K^3)$  [53]. The last step is to compute the roots of the annihilating filter to estimate the locations of the Diracs, which has as well a complexity of  $\mathcal{O}(K^3)$  (note that the roots are obtained from the eigenvalues of the companion matrix). If we assume that  $N = \alpha K^2$  with  $\alpha > 2$  and that this procedure is repeated at most  $K$  times, we obtain an overall complexity of  $\mathcal{O}(K^4)$ .

An additional point that has to be considered is that the Diracs are recovered from the measurements  $s[m] = \sum_{k=1}^K a_k e^{i\omega_0 t_k/T} (e^{i\lambda t_k/T})^m$ . Since the locations are retrieved from the phase of the complex numbers  $e^{i\lambda t_k/T}$  and the phase has a periodicity  $2\pi$ , we have to make sure that for Diracs located in the perfect reconstruction interval, that is  $t_k \in [0, (N - P)T)$ , we have  $\lambda t_k/T \in [0, 2\pi)$ . Which is satisfied when

$$\lambda = \frac{2\pi}{(N - P)}, \quad (3.18)$$

and determines the frequencies of the exponentials that are reproduced by the sampling kernel:

$$\omega_m = \frac{2\pi}{N - P} \left( m - \frac{P}{2} \right). \quad (3.19)$$



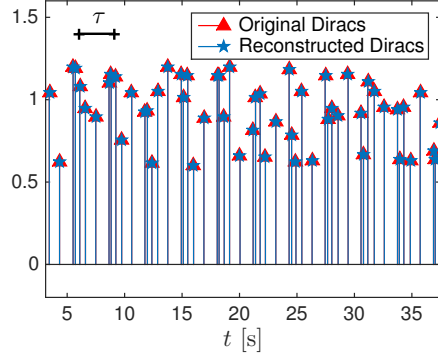
**Figure 3.4.:** Noisy scenario with SNR=15dB,  $N = 50$  and  $T = 1/16$ . The maximum rate of innovation of the streaming signal is  $2K/\tau = 3.2$  ( $K = 5$ ). (a) Plot of the sequentially estimated locations, the horizontal axis indicates the index of the sliding window and the vertical axis the location in time. (b) Histogram of the locations shown in (a). Horizontally aligned dots in (a) lead to peaks in the histogram in (b).

### 3.3.2. Noisy scenario

In the presence of noise, perfect reconstruction is not possible and the algorithm previously described becomes unstable. Moreover, the strict conditions on  $N$  and  $P$  come from imposing a critical sampling rate, since we have exactly  $2K$  values of the  $s[m]$  measurements to retrieve  $K$  Diracs. In the noisy case we relax this condition and allow larger values of  $P$ . This makes the denoising algorithms presented in the previous chapter more effective.

We thus develop a new strategy that is also based on using a sliding window and processing sets of  $N$  samples in sequential order. For each window and each group of  $N$  samples, we retrieve  $K$  Diracs using the algorithm in Section 3.2 coupled with matrix pencil and store all the locations and amplitudes retrieved in that window. We then slide the window by  $T$  and repeat the process. In other words, the  $i$ th window processes the noisy samples  $(\tilde{y}[n])_{n=n_i+1}^{n_i+N}$  and leads to the estimated locations  $\{\hat{t}_k^{(i)}\}$ . When the found locations correspond to real Diracs, they will be consistent among different positions of the sliding window that capture these Diracs. Otherwise, locations that are not correct and correspond to noise will normally be not consistent. Figure 3.4(a) shows the retrieved locations for different positions of the sliding window, where the horizontal axis corresponds to the window index,  $n_i$ , and the vertical axis to the locations in time, that is, for a given window index, each dot corresponds to an estimate of the set  $\{\hat{t}_k^{(i)}\}$ . Consistent locations among different windows appear as horizontally aligned dots. The shaded area represents the evolution in time of the observation window: for a given index  $n_i$ , the vertical cross-section of the shaded area represents the time interval  $\tau$  that is *seen* by this window. Consistent alignments of dots overlap with the horizontal straight lines that represent the true locations of the Diracs.

In order to detect which locations are consistent, a second step is to construct a histogram of detected locations. Only the peaks of the histogram are assumed to correspond to



**Figure 3.5.:** Sequential perfect reconstruction of a noiseless stream of 1000 Diracs with 10220  $y[n]$  samples. Only a small section of the stream is shown. Rate  $K = 5$  Diracs per  $\tau = 3.125$  s.  $N = 50$ ,  $T = 1/16$  and  $P = 9$ .

real Diracs. For a peak in the histogram above a certain threshold, the location of the corresponding Dirac is estimated averaging all the retrieved locations that contribute to this peak. This is illustrated in Figure 3.4(b).

Note that the robustness and accuracy of this algorithm depends as well on the minimum separation between consecutive spikes. More precisely, on the ratio between the minimum separation and the sampling rate. In the last part of Appendix C we analyse how the uncertainty of the retrieved locations for two Diracs changes with their separation. A theoretical uncertainty is derived from the Cramér-Rao lower bound and the actual performance of the noisy recovery algorithms is measured empirically. Note that a breakdown occurs when the separation goes below the sampling rate.

---

**Algorithm 1** Sequential FRI retrieval of Diracs

---

**Input:**  $(y[n])_{n=1}^{N_{TOT}}$ : stream of samples.

**Output:**  $\{(a_k, t_k)\}$ : Dirac locations and amplitudes.

- 1: **for**  $n_i = 0$  to  $N_{TOT} - N$  **do**
  - 2:     Retrieve  $\left\{ \left( \hat{a}_k^{(i)}, \hat{t}_k^{(i)} \right) \right\}$  from  $(y[n])_{n=n_i+1}^{n_i+N}$ .
  - 3: **end for**
  - 4: Construct histogram from retrieved locations  $\left\{ \hat{t}_k^{(i)} \right\}$ .
  - 5: Estimate Diracs from peaks of the histogram.
- 

### 3.4. Simulation results

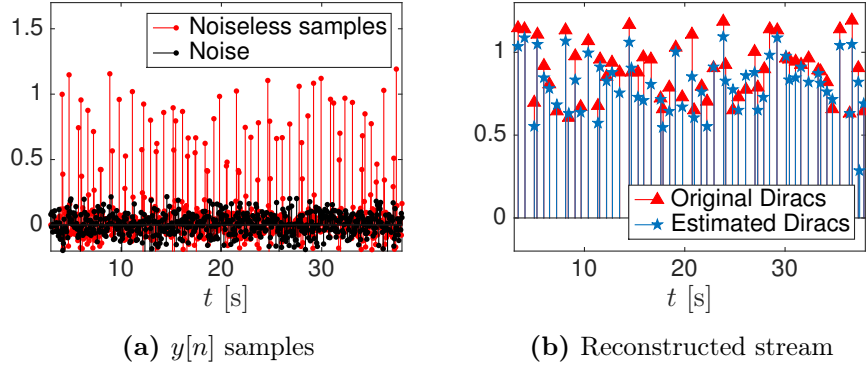
Both versions of the algorithm have been tested: the noiseless case for which perfect reconstruction is possible; and the noisy scenario, where locations are estimated from the histogram of the retrieved locations. In the noiseless case we always perfectly reconstruct the streams of Diracs with randomly generated locations and amplitudes. This is illustrated in Figure 3.5. The stream of Diracs is generated to satisfy the maximum rate of  $K$  Diracs per  $\tau$  interval.

In the noisy scenario not all the Diracs are always retrieved, and false positives may also happen. Note also that there is an uncertainty in the retrieved location. A retrieved Dirac is considered to correspond to a true Dirac if the difference between the real location and the estimated location is smaller than a threshold. Here we have set this threshold to  $T/2$ . We randomly generate the locations of a stream of 1000 Diracs for which we impose a minimal distance of  $2T$  between consecutive Diracs. Note that the uncertainty in the locations and the way we decide if a true Dirac has been detected would lead to distorted performance measurements if no minimal distance was imposed. The amplitudes of the Diracs are positive and are drawn from  $\mathcal{N}(1, 0.15)$ . We then take samples at regular intervals of time  $t = nT$  of the signal that results from convolving the stream of Diracs with the impulse response of the acquisition device, that is,  $y[n] = x(t) * h(t)|_{t=nT}$ . The samples  $y[n]$  are contaminated with additive white Gaussian noise where the amplitude of the noise is adjusted to satisfy a given SNR. The Diracs are recovered from the peaks of the histogram of the sequential algorithm described in the previous section. A peak is detected when the histogram goes beyond a threshold which has been set empirically to  $N/4$ . In theory, a Dirac leads to a peak in the histogram of height  $N$  since it is captured for  $N$  different locations of the sliding window. Due to noise and the border effects of the sliding window these peaks are lower. A threshold of  $N/4$  results in a good trade-off between detection rate and false positives. Figure 3.6 shows one realisation of the procedure explained before.

To further analyse the performance variation for different levels of noise we run the algorithm over 100 different realisations of noise for various levels of SNR. Table 3.1 summarises the obtained performances. The detection rate improves and the false positives are reduced when the SNR is increased (higher SNR implies lower noise). The results in Table 3.1 also show an increase in the accuracy of the retrieved locations. The algorithm has been implemented in MATLAB and tested using a commercial laptop (2.5 GHz Intel Core i5 CPU). The average time required to process 10220 samples corresponding to a stream of 630 seconds containing 1000 Diracs is about 105 seconds.

**Table 3.1.:** Noisy algorithm's performance. Stream of 1000 Diracs (630 seconds) and 10220 samples,  $T = 1/16$  s,  $N = 50$ ,  $P + 1 = 23$ . The experiment is repeated 100 for each SNR. The detection rate is given in percentage of detected true Diracs. The false positives are the average number of detected Diracs that do not correspond to true Diracs. The precision is the standard deviation of the retrieved locations with respect to the true locations averaged over the different realisations. The MSE corresponds to error in retrieving the amplitudes of the Diracs.

| SNR [dB]        | 5                   | 10                  | 15                  | 20                  |
|-----------------|---------------------|---------------------|---------------------|---------------------|
| Detection rate  | 97.69 %             | 99.97 %             | 100.00 %            | 100.00 %            |
| False positives | 351.7               | 37.8                | 0.5                 | 0.3                 |
| Precision [s]   | $0.0086 \pm 0.0002$ | $0.0049 \pm 0.0001$ | $0.0029 \pm 0.0001$ | $0.0018 \pm 0.0001$ |
| MSE             | $0.0325 \pm 0.0038$ | $0.0172 \pm 0.0013$ | $0.0129 \pm 0.0028$ | $0.0126 \pm 0.0069$ |



**Figure 3.6.:** Noisy samples with a SNR = 10 dB and reconstructed stream from the peaks of the histogram of the retrieved locations. The temporal locations are very accurately estimated.

### 3.5. A note on periodicity and reducing the number of temporal samples

So far, the critical sampling condition has been imposed for the number of measurements  $s[m] = \sum_n c_{m,n} y[n]$ , with  $m = 0, 1, \dots, P$ . The number of measurements  $s[m]$  has to be at least equal to the number of degrees of freedom, and therefore, in order to estimate the amplitudes and locations of  $K$  Diracs we have the following condition:  $P + 1 \geq 2K$ . However, this condition does not take into account the minimum number of temporal samples  $y[n]$  that are required and, in general, this number is larger than  $2K$ . Moreover, in the sequential reconstruction scheme presented in the previous sections perfect reconstruction is achieved when  $N \geq 2K^2$  (see Equation (3.16)). In this section, we present an alternative acquisition device that allows reducing the number of temporal samples to  $N = 2K$  by taking advantage of the underlying periodicity of the exponential functions that are reproduced by the sampling kernel.

We assume that we want to reconstruct a stream of  $K$  Diracs,  $x(t) = \sum_{k=1}^K a_k \delta(t - t_k)$ , that are located on a temporal interval of  $\tau$  seconds, that is,  $t_k \in [0, \tau)$ , for  $k = 1, \dots, K$ . Moreover, we acquire the signal with an exponential reproducing kernel  $\varphi(t)$  that is able to reproduce  $P + 1$  exponential functions, and therefore, the causal filter  $h(t) = \varphi(-t/T)$  is assumed to have support  $[0, (P + 1)T]$  as illustrated in Figure 3.1(b).

The locations of the Diracs are estimated from the phase of the parameters  $\{u_k = e^{i\lambda t_k/T}\}$  which are obtained from the measurements  $s[m] = \sum_{k=1}^K b_k u_k^m$ . The range of the phase of a complex number is  $[0, 2\pi)$  (or any other interval of size  $2\pi$ ). Therefore, in order to maximise the precision of the reconstruction process and avoid ambiguities we establish the following condition

$$0 \leq t_k < \tau \quad \Leftrightarrow \quad 0 \leq \lambda t_k/T < 2\pi, \quad (3.20)$$

which in turn leads to  $\lambda = 2\pi T/\tau$ . Since we want to establish conditions for the critical sampling rate, we require the order of the sampling kernel to be  $P + 1 = 2K$ . Moreover, we want the temporal interval  $\tau$  to be covered by exactly the same number of temporal samples, that is,  $N = 2K = P + 1$ . We thus have that

$$\lambda = \frac{2\pi}{P+1} \quad \text{and} \quad \omega_m = \frac{\pi}{P+1}(2m - P). \quad (3.21)$$

The samples  $y[n]$  can be written as follows

$$y[n] = \langle x(t), \varphi(t/T - n) \rangle = \sum_{k=1}^K a_k \varphi\left(\frac{t_k}{T} - n\right), \quad (3.22)$$

and since the support of  $\varphi(t/T - n)$  is given by (3.13), the non-zero samples correspond to indices

$$n = 1, \dots, \underbrace{2(P+1)}_{=2N=4K}. \quad (3.23)$$

This is exactly twice as many samples as the critical sampling rate of  $2K$ . Normally, in order to achieve perfect reconstruction, all  $2N$  samples have to be considered when computing the measurements  $s[m]$ , that is,

$$s[m] = \sum_{n=1}^{2(P+1)} c_{m,n} y[n]. \quad (3.24)$$

An important aspect of the reconstruction scheme is that it is based on the fact that  $\varphi(t)$  is able to reproduce exponentials. Moreover, these exponentials are purely imaginary, which leads to periodic functions. This periodicity is also reflected in the  $c_{m,n}$  coefficients:

$$c_{m,n-(P+1)} = e^{i\omega_m(n-(P+1))} c_{m,0} = e^{i\pi P} c_{m,n} = (-1)^P c_{m,n}. \quad (3.25)$$

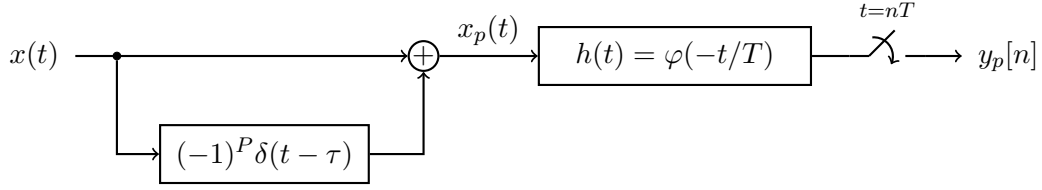
Samples  $y[n]$  for  $n = 1, \dots, (P+1)$  can also be written in terms of index  $n$  going from  $(P+1) + 1$  to  $2(P+1)$  as follows

$$y[n - (P+1)] = \left\langle x(t), \varphi\left(\frac{t}{T} - n + (P+1)\right) \right\rangle \quad (3.26)$$

$$= \sum_{k=1}^K a_k \varphi\left(\frac{t_k}{T} - n + (P+1)\right) \quad (3.27)$$

$$= \sum_{k=1}^K a_k \varphi\left(\frac{t_k + \overbrace{(P+1)T}^{=\tau}}{T} - n\right). \quad (3.28)$$

This expression is equivalent to sampling at instants of time  $t = nT$ , for  $n = (P+1) +$



**Figure 3.7.:** Critical sampling device. A replica of the signal  $x(t)$  is delayed by  $\tau$  seconds and multiplied by  $(-1)^P$  where  $P$  is the order of the sampling kernel. This replica, added to the original signal form  $x_p(t)$ . This new signal is then filtered and sampled with the traditional FRI acquisition device.

$1, \dots, 2(P+1)$ , a stream of Diracs located at  $t_k + \tau \in [\tau, 2\tau)$ , that is, the original stream  $x(t)$  delayed by  $\tau$  seconds. Let

$$y'[n] = y[n - (P+1)] = \langle x(t - \tau), \varphi(t/T - n) \rangle, \quad n = (P+1) + 1, \dots, 2(P+1). \quad (3.29)$$

From (3.25) and (3.29) it follows that

$$s[m] = \sum_{n=1}^{2(P+1)} c_{m,n} y[n] = \sum_{n=P+2}^{2(P+1)} c_{m,n} ((-1)^P y'[n] + y[n]) \quad (3.30)$$

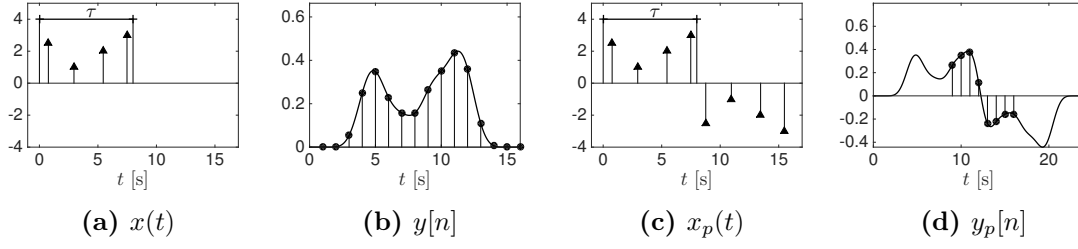
$$= \sum_{n=P+2}^{2(P+1)} c_{m,n} \left\langle (-1)^P x(t - \tau) + x(t), \varphi\left(\frac{t}{T} - n\right) \right\rangle. \quad (3.31)$$

We denote by  $x_p(t) \stackrel{\text{def}}{=} (-1)^P x(t - \tau) + x(t)$  the analogue signal that results from adding to the original signal  $x(t)$  a delayed and weighted version of it. By considering that we have imposed  $P+1 = 2K$ , we can rewrite the previous expression as follows

$$s[m] = \sum_{n=n_0+1}^{n_0+2K} c_{m,n} \underbrace{\left\langle x_p(t), \varphi\left(\frac{t}{T} - n\right) \right\rangle}_{\stackrel{\text{def}}{=} y_p[n]} = \sum_{n=n_0+1}^{n_0+2K} c_{m,n} y_p[n], \quad m = 0, 1, \dots, P, \quad (3.32)$$

where  $n_0 = P+1$ . Note that we only require  $N = 2K$  temporal samples  $y_p[n]$  that result from sampling the signal  $x_p(t)$ . The measurements  $s[m]$  in (3.32) are identical to the  $s[m]$  values in (3.8) obtained from  $4K$  samples  $y[n]$  after filtering and sampling  $x(t)$ . We can thus conclude that perfect reconstruction of  $K$  Diracs is achievable from only  $2K$  temporal samples given by the acquisition device illustrated in Figure 3.7.

Figure 3.8 depicts the signals involved in the sampling process of a stream of  $K = 4$  Diracs. Here (a) and (b) illustrate the original signal  $x(t)$  and the samples  $y[n]$  that are used in the traditional FRI setup. In order to achieve perfect reconstruction the samples  $y[n]$  have to capture the region where  $y(t)$  is non-zero and lead to  $4K = 16$  non-zero samples. The critical sampling scheme presented in this section is based on constructing



**Figure 3.8.:** (a) Stream of  $K = 4$  Diracs located in a temporal interval  $\tau = 4$  s. The traditional sampling scheme illustrated in (b) requires  $N = 4K = 16$  temporal samples  $y[n]$ . (c) intermediate signal  $x_p(t)$  of the critical sampling device illustrated in Figure 3.7. (d) Perfect reconstruction is achieved from only  $2K = 8$  temporal samples.

first the signal  $x_p(t)$  in the continuous-time domain, which is illustrated in (c). In this case perfect reconstruction is achieved with only the  $2K$  samples that are depicted in (d).

### 3.6. Summary

In this chapter we have presented a fast sequential algorithm to retrieve infinite streams of Diracs in noiseless and noisy environments. In the noiseless case, perfect reconstruction is achieved. In the noisy scenario we propose to retrieve sequentially groups of  $K$  Diracs and to retain only those Diracs whose locations have been consistently estimated in overlapping sliding windows. We have showed that the algorithm is able to process  $10K$  samples in about 100 seconds and can retrieve with high accuracy 1000 Diracs even in very low SNR regimes.

We have also presented an alternative acquisition device that is able to achieve perfect reconstruction of bursts of  $K$  Diracs from only  $2K$  temporal samples. This corresponds to the critical sampling rate since  $2K$  parameters have to be estimated:  $K$  amplitudes and  $K$  delays. This approach takes advantage of the fact that the exponential functions that are reproduced by the sampling kernel are periodic.

## Chapter 4.

# Neural activity monitoring from two-photon calcium imaging

Inferring the times of sequences of action potentials (spike trains) from neurophysiological data is a key problem in computational neuroscience. The detection of action potentials (APs) from two-photon imaging of calcium ion concentration offers certain advantages over traditional electrophysiological approaches, as up to thousands of spatially and immunohistochemically defined neurones can be recorded simultaneously. However, due to noise, dye buffering and the limited sampling rates in common microscopy configurations, accurate detection of APs from calcium time series has proved to be a difficult problem. Here we present a novel approach to the problem making use of FRI theory.

A calcium transient is a rapid increase of calcium ion concentration in a cell which peaks and gradually decreases during action potentials. For calcium transients well fit by a single exponential, the problem is reduced to reconstructing a stream of decaying exponentials. Signals made of a combination of exponentially decaying functions with different onset times are a subclass of FRI signals, for which much theory has recently been developed by the signal processing community. In this chapter we develop an algorithm based on the use of FRI theory to retrieve the timing of APs from calcium transient time series. The final algorithm is fast, non-iterative and parallelisable. Spike inference can be performed in real-time for a population of neurones and does not require any training phase or learning to initialise parameters. The algorithm has been tested with both real data (obtained by simultaneous electrophysiology and multiphoton imaging of calcium signals in cerebellar Purkinje cell dendrites), and surrogate data, and outperforms several recently proposed methods for spike train inference from calcium imaging data.

---

<sup>0</sup>This chapter includes research conducted jointly with Prof Dragotti, Dr Schultz and Dr Urigüen and lead to the following publications [3, 12]. All the real data that has been used in this work was acquired by Dr Schultz and his team. The extension of the FRI framework to decaying exponentials was derived by Dr Urigüen. The code to reproduce the results presented in this chapter is available online: <https://github.com/jonativia/thesis>.

## 4.1. Introduction

Understanding how information processing occurs in neural circuits is one of the principal problems of systems neuroscience. Information is encoded in the firing of action potentials (APs, or spikes) by individual neurones, and information processing involves the coordination of action potential firing by large populations of neurones organised into neural circuits. Key processing steps are thought to occur on the level of local microcircuits that contain on the order of 1,000–10,000 cells. These local circuits form highly connected three-dimensional networks [64]. To understand neural information processing, we thus must monitor the activity of neural circuits at a spatial resolution sufficient to resolve many individual neurones, and a temporal resolution sufficient to resolve individual action potentials on individual experimental trials. Of the currently available techniques for conducting neurophysiological experiments, only multiphoton calcium imaging [65–68] and multielectrode array electrophysiology [69–71] offer this capability. Of these, only multiphoton calcium imaging currently allows precise three-dimensional localisation of each individual monitored neurone within the region of tissue studied, in the intact brain.

During an action potential, the difference in electric potential between the interior and the exterior of a neurone rapidly rises and falls. These electric potential differences are induced by ion concentration gradients that are controlled by ion channels present in the membrane of the neurones. The occurrence of action potentials can therefore be inferred by measuring changes in intracellular concentration of calcium ion [72]. In order to monitor cellular activity, neurones must be labelled with a fluorescent  $\text{Ca}^{2+}$  indicator, and a number of approaches have been used to do this. Single cells can be labelled by filling the cell with dye during a whole-cell or intracellular recording [73, 74]. Alternatively, populations of neurones can be simultaneously labelled with acetoxymethyl (AM) ester calcium dyes [68], allowing simultaneous monitoring of action-potential induced calcium signals in a plane [75] or volume [64] of tissue. To investigate information processing in neural circuits, it is necessary to relate these calcium signals to the properties of the spike trains fired by the neurones, ideally by detecting the times of occurrence of spikes with single action potential resolution. A number of methods have previously been used to detect spike trains from calcium imaging data, including thresholding the first derivative of the calcium signal [72], and the application of template-matching algorithms based on either fixed exponential [76–78] or data-derived [79, 80] templates. Machine learning techniques [81] and probabilistic methods based on sequential Monte Carlo framework [82] or fast deconvolution [83] have also been proposed.

Some broadly used methods such as template matching or derivative-thresholding have the disadvantage that they do not deal well with multiple events occurring within a time period comparable to the sampling interval. Unfortunately, given that laser-scanning two-photon imaging systems are largely limited to scan rates of 8–30 Hz when frame-scanning

with sufficient spatial resolution to capture many neurones, and that neurones in many brain areas have a propensity to fire spikes in bursts, this is a relatively common occurrence in neurophysiological calcium signals. Bursts of spikes have been found to convey information with high reliability in some sensory systems [84, 85], and have been suggested to carry distinct sensory signals [86]. It is thus desirable to develop calcium transient detection algorithms that accurately detect multiple spike calcium events. As there is a trade-off between the area of tissue imaged and signal-to-noise ratio (zooming in on a region of tissue allows the collection of more photons per neurone, thus offering better signal-to-noise ratio, but limits the number of neurones that can be studied) and similarly between sampling rate and the area of tissue that can be imaged, it is desirable to improve algorithms for the detection of action potentials from calcium fluorescence time series.

In this chapter we present an extension to the FRI theory to sample and reconstruct streams of decaying exponentials. Since APs in calcium imaging data are well modelled by decaying exponentials, in the absence of noise, the proposed FRI algorithm perfectly retrieves the locations of APs using Prony’s method. We then combine this approach with a double consistency sliding window technique that improves performances in noisy scenarios. To reconstruct the time series we construct a Toeplitz matrix from the samples. The key characteristic of this matrix is that, in the noiseless case, it is rank deficient, and its rank is always equal to the number of APs in the observation window. We run the algorithm twice, firstly with a large time window to estimate the number of spikes by SVD, and secondly, with a time window containing only a small number of samples where we assume that a single spike is observed. In both cases, the sliding window approach is based on the theory presented in the previous chapter. We construct a joint histogram of the retrieved locations with the two different window sizes. The final spike time estimates are obtained from histogram peaks, corresponding to consistent positions among different windows.

The proposed algorithm is robust in high noise scenarios, and fast enough to allow real-time spike train inference for tens of neurones. We show that for surrogate data with a temporal resolution of 27 Hz and a SNR of 10 dB the algorithm presents a spike detection rate above 95 % with a false-positive rate below 0.02 Hz. Moreover, this algorithm is able to retrieve the spike locations with a precision higher than the temporal resolution of the acquired data.

## 4.2. Methods

### 4.2.1. Experimental methods

The data used in this study, and the experimental methods used to collect them, have been described in [79]. The data was provided by Professor Simon R. Schultz and his team from the Bioengineering Department at Imperial College London. Briefly, Sprague-Dawley

rats (P18-P29) were anaesthetised with urethane (1.2 g/kg) or with ketamine (50 mg/kg) / xylazine (5 mg/kg). A craniotomy was made over area Crus IIa of the cerebellum, filled with 1.5-2 % agarose in Ringer's solution, and a coverslip clamped above the agarose to suppress brain movement, while leaving a window open for microelectrode access. A micropipette was inserted to a depth of around 100-200  $\mu\text{m}$  below the pia mater, and AM-ester calcium dye (Oregon Green BAPTA-1 AM) pressure-ejected. Imaging was performed from 30 minutes following dye ejection, using a two-photon laser scanning microscope (Prairie Technologies). A pulsed Titanium:Sapphire laser was used for excitation, operating at 810 nm (MaiTai, SpectraPhysics) with  $< 100$  fs pulse width and 80 MHz repetition rate, and focused using a 40x, 0.8 Numerical Aperture objective lens (Olympus).

Image frames were acquired using ScanImage software [87] for MATLAB (MathWorks). Raster lines making up each frame were of 2 or 2.3 ms duration, resulting in frame rates of 7-16 Hz. For each region imaged, a high resolution reference image was first acquired (512x512 pixels, average of five frames), followed by movies at 256x64 or 256x32 pixel resolution. Fluorescence time series for each neurone were obtained by defining regions of interest (ROIs) using a combination of human operator and spatial independent component analysis [79, 88], and for each time bin, averaging the values of each pixel within the ROI.

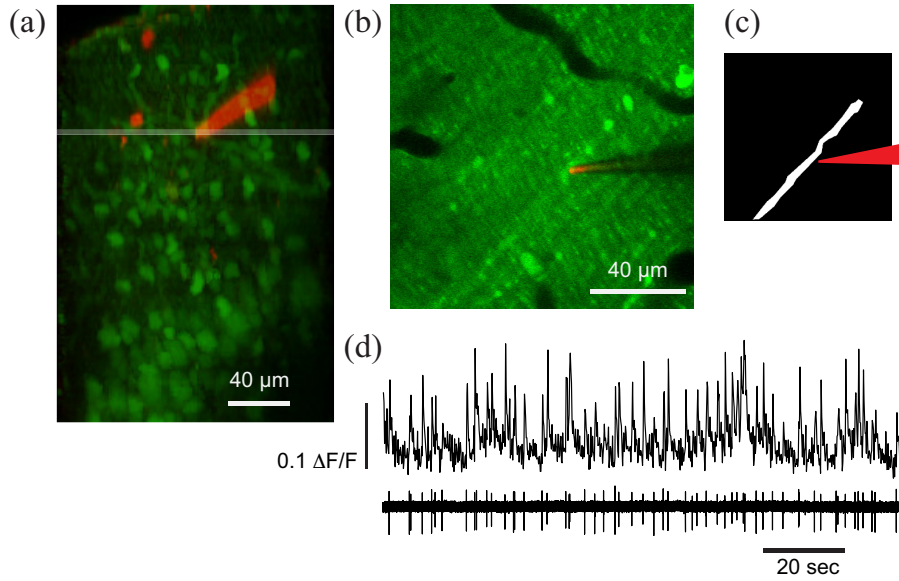
To validate our event detection algorithms, targeted extracellular recordings from imaged neurones were simultaneously performed. Patch micropipettes ( $\sim 4$  M $\Omega$ ) were filled with artificial cerebrospinal fluid (ACSF), together with Alexa 594 to aid visualisation of the pipette. The pipette was navigated until the tip was adjacent to a Purkinje cell soma or dendrites and complex spikes could be observed with high signal-to-noise ratio (SNR). We emphasise that we are using two-photon targeted (visualised) juxtacellular recording, using a patch-pipette filled with dye. Using this technique, we can observe that the pipette is attached to a cell in which fluorescence changes are observed for each action potential, meaning that there is no ambiguity concerning which cell is being recorded from. Electrophysiological and imaging data were then simultaneously acquired from the same neurone (Figure 4.1).

#### 4.2.2. Mathematical model

At time  $t$  we consider the fluorescence measurement for a given ROI to be proportional to the calcium concentration plus additive Gaussian noise [82]:

$$F(t) = \alpha[\text{Ca}^{2+}](t) + \beta + \varepsilon(t), \quad (4.1)$$

where  $[\text{Ca}^{2+}](t)$  is the intracellular calcium concentration at time  $t$ , constant  $\beta$  represents the baseline calcium concentration of a particular cell and  $\varepsilon(t)$  the noise at time  $t$ . The noise is independently and identically distributed according to a normal distribution with zero mean and  $\sigma^2$  variance.



**Figure 4.1.:** Simultaneous multiphoton calcium imaging with electrophysiology. (a) Maximum intensity projection showing juxtacellular recording from a Purkinje cell dendrite. The tissue was loaded with Oregon Green BAPTA-1 AM calcium indicator dye (green), and the pipette filled with Alex 594 (red) to aid visualisation during targeted recording. (b) Imaged location, corresponding to grey horizontal line in (a). (c) Mask showing region of interest for the recorded Purkinje cell. (d) Simultaneous acquisition of fluorescence time series (shown unfiltered) and dendritically recorded complex spikes (CS), showing CS-driven calcium transients.

The signal that we consider is the normalised fluorescence

$$\Delta F(t)/F_0 = \frac{F(t) - F_0}{F_0}, \quad (4.2)$$

where  $F_0$  is the average background pre-stimulus fluorescence.

To model mathematically the calcium dynamics  $[\text{Ca}^{2+}](t)$ , some assumptions have to be made [82]. We assume that when a neurone is activated, the calcium concentration jumps instantaneously, and each jump has the same amplitude  $A$ . This is justified by the fact that the amount of calcium influx per spike is constant [72]. The concentration then decays exponentially with time constant  $\tau$ , to a baseline concentration. The one-dimensional fluorescence signal can therefore be characterised by convolving the spike train with a decaying exponential and adding noise:

$$\Delta F(t)/F_0 = A \sum_k e^{-(t-t_k)/\tau} \mathbf{1}_{t \geq t_k} + \epsilon(t) \quad (4.3)$$

$$= A \sum_k \delta(t - t_k) * e^{-t/\tau} \mathbf{1}_{t \geq 0} + \epsilon(t), \quad (4.4)$$

where the index  $k$  represents different spikes and the different  $t_k$  their occurrence times. Hence, the goal of the spike detection algorithm is to obtain the values  $t_k$ .

### 4.2.3. Spike detection

Our spike detection algorithm is based on connecting the calcium transient estimation problem to the theory of FRI signals. Streams of Diracs are an idealization of streams of pulses. Although this example may seem limited, the framework presented in the previous chapters can be extended to a variety of signals. For instance, to calcium concentration measurements obtained from two-photon imaging to track the activity of individual neurones. In this model, the time delays correspond to the activation time of the tracked neurone, that is, the action potentials (AP).

### Extension of FRI theory to streams of decaying exponentials

Based on the traditional FRI framework presented in Chapter 2, we now develop a method to sample and reconstruct streams of decaying exponentials. We assume that there is a finite unknown number  $K$  of spikes within the observation period. Therefore the noiseless calcium concentration variation, denoted  $c(t)$ , can be expressed as

$$c(t) = \sum_{k=1}^K a_k e^{-(t-t_k)/\tau} \mathbf{1}_{t \geq t_k} = \underbrace{\sum_{k=1}^K a_k \delta(t - t_k)}_{\stackrel{\text{def}}{=} x(t)} * \underbrace{e^{-\alpha t} \mathbf{1}_{t \geq 0}}_{\stackrel{\text{def}}{=} p_\alpha(t)} = x(t) * p_\alpha(t), \quad (4.5)$$

where  $\alpha \stackrel{\text{def}}{=} 1/\tau$ . The function  $c(t)$  is also an FRI signal since it is perfectly determined by a finite number of parameters:  $\{(a_k, t_k)\}_{k=1}^K$ . Let us assume that  $c(t)$  is sampled with the acquisition device described in Section 2.4, that is, an exponential reproducing kernel  $h(t) = \varphi(-t/T)$ , followed by a sampling stage. The resulting samples  $y[n]$  can be expressed as the inner product between  $c(t)$  and  $\varphi(t/T - n)$ :

$$y[n] = \left\langle c(t), \varphi\left(\frac{t}{T} - n\right) \right\rangle. \quad (4.6)$$

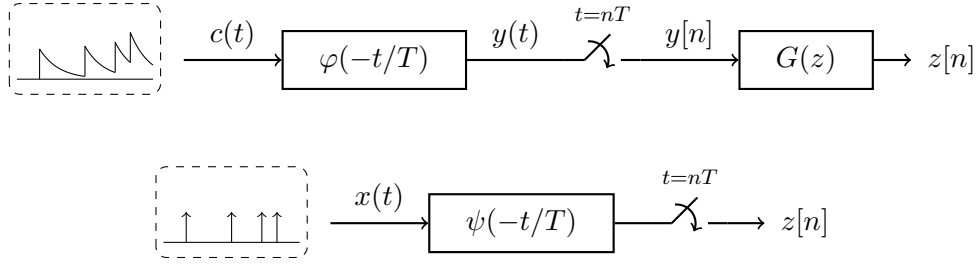
It is easy to verify that these samples are equivalent to sampling the stream of Diracs  $x(t)$  with the filter  $p_\alpha(-t) * \varphi(t/T - n)$ :

$$y[n] = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} x(\tau) p_\alpha(t - \tau) d\tau \right] \varphi(t/T - n) dt \quad (4.7)$$

$$= \int_{-\infty}^{+\infty} x(\tau) \int_{-\infty}^{+\infty} \underbrace{p_\alpha(t - \tau) \varphi(t/T - n)}_{= p_\alpha(-t) * \varphi(t/T - n)} dt d\tau \quad (4.8)$$

$$= \left\langle x(t), p_\alpha(-t) * \varphi\left(\frac{t}{T} - n\right) \right\rangle. \quad (4.9)$$

Note that since  $p_\alpha(t)$  is of infinite support, the equivalent filter  $p_\alpha(-t) * \varphi(t/T - n)$  is also of infinite support. This leads to an infinite number of non-zero samples  $y[n]$  even if the



**Figure 4.2.:** Sampling stream of decaying exponentials and equivalent problem. The first row represents the sampling scheme for a stream of decaying exponentials, the block  $G(z)$  represents the discrete-time filter  $1 - e^{-\alpha T} z^{-1}$  which leads to samples  $z[n] = y[n] - e^{\alpha T} y[n-1]$ . The second row represents the equivalent problem that leads to the same samples  $z[n]$  when the signal  $x(t)$  is a stream of Diracs with the same locations and amplitudes as the decaying exponentials in  $c(t)$ .

spikes are localised in time.

Let us also assume that the exponentials reproduced by the sampling kernel  $e^{i\omega_m t}$  can be expressed as  $e^{i(\omega_0 + \lambda m)t}$ , with  $m = 0, 1, \dots, P$ . Next, we show that sampling the signal in (4.5) with  $\varphi(-t/T)$  and computing the following finite differences

$$z[n] = y[n] - e^{-\alpha T} y[n-1], \quad (4.10)$$

is equivalent to the sequence that would result from sampling the stream of Diracs  $x(t) = \sum_{k=1}^K a_k \delta(t - t_k)$  with the following kernel

$$\psi(t) \stackrel{\text{def}}{=} \beta_{-\alpha T} \left( -\frac{t}{T} \right) * \varphi(t) \quad (4.11)$$

where  $\beta_{-\alpha T}(-t/T)$  is an E-spline with parameter  $-\alpha T$  [3]. The parameter  $\alpha$  is the constant of the exponential in (4.5). The sampling scheme for the stream of decaying exponentials and the equivalent problem of sampling a stream of Diracs are illustrated in Figure 4.2.

The weighted differences can be written as

$$z[n] = \left\langle x(t), p_\alpha(-t) * \varphi\left(\frac{t}{T} - n\right) - e^{-\alpha T} p_\alpha(-t) * \varphi\left(\frac{t}{T} - (n-1)\right) \right\rangle \quad (4.12)$$

since the inner product is linear. Applying Parseval's relation we can also write

$$z[n] = \frac{1}{2\pi} \left\langle \hat{x}(\omega), \mathcal{F} \left\{ p_\alpha(-t) * \varphi\left(\frac{t}{T} - n\right) - e^{-\alpha T} p_\alpha(-t) * \varphi\left(\frac{t}{T} - (n-1)\right) \right\} \right\rangle. \quad (4.13)$$

By noting that  $\mathcal{F}\{p_\alpha(-t)\} = \frac{1}{\alpha - i\omega}$  and that  $\mathcal{F}\{\varphi(t/T - n)\} = |T| \hat{\varphi}(\omega T) e^{-i\omega n T}$ , the

second term of this inner product can be expanded as follows:

$$\mathcal{F} \left\{ p_\alpha(-t) * \varphi \left( \frac{t}{T} - n \right) - e^{-\alpha T} p_\alpha(-t) * \varphi \left( \frac{t}{T} - (n-1) \right) \right\} \quad (4.14)$$

$$= \mathcal{F} \{ p_\alpha(-t) \} \mathcal{F} \left\{ \varphi \left( \frac{t}{T} - n \right) - e^{-\alpha T} p_\alpha(-t) * \varphi \left( \frac{t}{T} - (n-1) \right) \right\} \quad (4.15)$$

$$= \frac{1}{\alpha + i\omega} T \hat{\varphi}(\omega T) e^{-i\omega n T} [1 - e^{-\alpha T} e^{i\omega T}]. \quad (4.16)$$

Note that the absolute value of  $T$  has been replaced by  $T$ , this is justified by the fact that this value is always real and positive in our context. We thus have that

$$z[n] = \frac{1}{2\pi} \left\langle \hat{x}(\omega), \frac{1 - e^{-(\alpha - i\omega)T}}{\alpha - i\omega} T \hat{\varphi}(\omega T) e^{-i\omega n T} \right\rangle. \quad (4.17)$$

In the second part of the inner product we can recognise an expression which is similar to the Fourier transform of an E-Spline with parameter  $\alpha$ ,  $\hat{\beta}_\alpha(\omega) = \frac{1 - e^{\alpha - i\omega}}{i\omega - \alpha}$  (see (2.41)). We have that

$$\beta_{-\alpha T} \left( -\frac{t}{T} \right) = \begin{cases} e^{\alpha t}, & -T < t \leq 0, \\ 0, & \text{otherwise,} \end{cases} \quad \xrightarrow{\mathcal{F}} \quad \int_{-T}^0 e^{(\alpha - i\omega)t} dt = \frac{1 - e^{-(\alpha - i\omega)T}}{\alpha - i\omega}. \quad (4.18)$$

Therefore, applying Parseval's theorem again yields

$$z[n] = \left\langle x(t), \beta_{-\alpha T} \left( -\frac{t}{T} \right) * \varphi \left( \frac{t}{T} - n \right) \right\rangle. \quad (4.19)$$

We thus have that

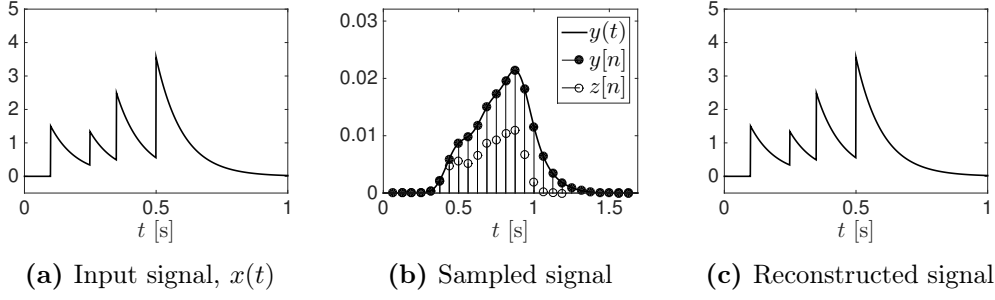
$$z[n] = \left\langle x(t), \psi \left( \frac{t}{T} - n \right) \right\rangle, \quad (4.20)$$

where  $\psi(t)$  is defined as in (4.11). This new kernel,  $\psi(t)$ , is still able to reproduce exponentials [41]. That is, there exists coefficients  $d_{m,n}$  such that

$$\sum_n d_{m,n} \psi(t - n) = e^{i\omega_m t}. \quad (4.21)$$

Note that these coefficients  $d_{m,n}$  are in general different to the  $c_{m,n}$  coefficients that correspond to the exponential reproducing kernel  $\varphi(t)$  of the actual acquisition device.

The problem of estimating the calcium transients and the problem of reconstructing an FRI signal are now equivalent. In fact, we now have a set of samples  $z[n] = \langle x(t), \psi(t/T - n) \rangle$  which are equivalent to those that we would obtain if we were sampling the stream of Diracs  $x(t)$  with the exponential reproducing kernel  $\psi(t)$ . We can therefore apply the FRI techniques described in the previous chapters to retrieve the parameters of these Diracs which correspond exactly to the locations and amplitudes of the decaying



**Figure 4.3.:** Sampling of a stream of decaying exponentials and perfect reconstruction. Since  $x(t)$  is an infinite duration signal, samples  $y[n]$  are non-zero for  $n \geq n_0$ , for some  $n_0$  that depends on the location of the first decaying exponential. However, if the number of decaying exponentials is finite, the number of non-zero samples  $z[n] = y[n] - e^{-\alpha T} y[n-1]$  is also finite since they are equivalent to sampling a stream of Diracs with a compact support kernel.

exponentials in  $c(t)$ . Specifically, we compute the new measurements  $s[m]$

$$s[m] = \sum_{n \in \mathbb{Z}} d_{m,n} z[n]. \quad (4.22)$$

Note that the stream of decaying exponentials is acquired with the sampling kernel  $\varphi(t)$ . However, the coefficients  $d_{m,n}$  that are applied to the samples  $z[n]$  correspond to the coefficients in (4.21) of the equivalent sampling kernel  $\psi(t)$  that is defined in (4.11). From  $s[m]$  we build the Toeplitz matrix  $\mathbf{S}$  as discussed in Section 2.4 and then apply Prony's method to retrieve the spikes. Figure 4.3 illustrates the perfect reconstruction of a stream of  $K = 4$  decaying exponentials. We summarise this inference method in Algorithm 2.

An important aspect of this approach is that the number of non-zero samples  $z[n]$  is finite when the sampling kernel  $\varphi(t)$  is of compact support. This is not the case for the samples  $y[n]$  as can be observed in Figure 4.3. This aspect allows the algorithm to perform the computation of the spike locations in real-time since the summation in Equation (4.22) is limited to a finite number of indices  $n$ .

Another relevant feature of this algorithm is that, in the noiseless scenario, we do not need to know the number of spikes. The number of spikes  $K$  can be obtained from the rank of the Toeplitz matrix built from the measurements  $s[m]$  as has been described in Chapter 2.

### Spike inference in practice

Due to the presence of noise, real data leads to Toeplitz matrices  $\mathbf{S}$  which are not rank deficient (see Algorithm 2 for details about how build  $\mathbf{S}$ ). Moreover, the number of spikes  $K$  within a time interval is unknown and cannot be estimated from the rank of matrix  $\mathbf{S}$ . Here, we describe a robust spike detection algorithm that is based on Algorithm 2 and the principles to sample infinite streams presented in the previous chapter.

In the noiseless case, the matrix  $\mathbf{S}$  has rank  $K$ . The SVD of this matrix has therefore

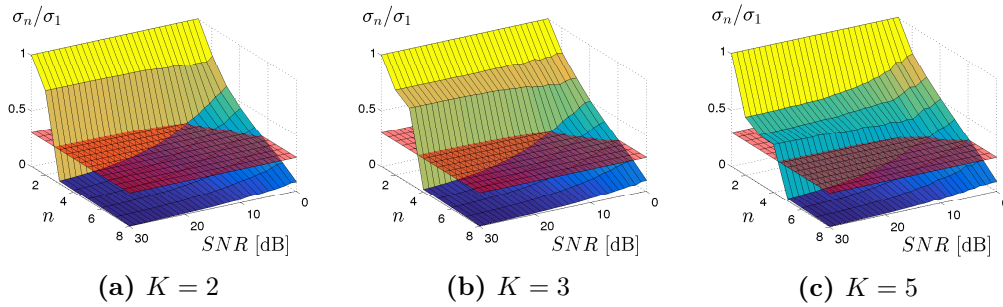
**Algorithm 2** FRI spike train inference (noiseless scenario)**Input:**  $c(t)$ : calcium concentration.**Output:**  $\{t_k\}_{k=1}^K$ : spike locations.

- 1: Filter with exponential reproducing kernel:  $y[n] = \langle c(t), \varphi(t/T - n) \rangle$ .
- 2: Compute weighted finite differences:  $z[n] = y[n] - e^{-\alpha T} y[n-1]$ .
- 3: Obtain new measurements to build Toeplitz matrix:  $s[m] = \sum_n d_{m,n} z[n]$ .
- 4: Estimate number of spikes  $K$  from Toeplitz matrix  $\mathbf{S}$ .
- 5: Compute the annihilating filter:  $\mathbf{S} \mathbf{h} = \mathbf{0}$ .
- 6: Retrieve locations from roots of the annihilating filter:  $H(z) = \sum_{m=0}^K h[m] z^{-m} = \prod_{k=1}^K (1 - u_k z^{-1})$ , where  $u_k = e^{i\lambda t_k/T}$ .

only  $K$  non-zero singular values. When noise is added to the input signal, the matrix  $\mathbf{S}$  becomes full rank and if we do not have prior knowledge of  $K$ , estimating its value becomes part of the problem. In a low noise scenario and when  $K$  is not zero,  $K$  can be estimated from the singular values of  $\mathbf{S}$ . In this case, the contribution of the signal in the singular value of  $\mathbf{S}$  is more important than the contribution of the noise, and a clear separation can be established to estimate the number of singular values that are due to the signal. Figure 4.4 illustrates the empirical estimation of the threshold that has been used to calculate the number of spikes from the singular values of the Toeplitz matrix  $\mathbf{S}$ . This threshold has been set to 0.3. Specifically, we keep the  $n$ th singular value if its normalised value with respect to the largest singular value is bigger than 0.3. For low levels of noise, the number of singular values above this threshold is a good estimate of the rank of the corresponding noiseless matrix. Another effect of the noise is that Prony's or matrix pencil methods do not lead to the exact solutions. These solutions can be made more accurate by applying the two denoising strategies presented in Section 2.5: "Cadzow+Prony" and the *robustified* matrix pencil. Both approaches lead to similar performances whilst the second is computationally more efficient, it is thus the method we have applied in our simulations.

Correct estimation of the number of spikes within the time window where we are searching for spikes is crucial to obtain good performances. The previously described approach, where  $K$  is estimated from the singular values of the matrix  $\mathbf{S}$ , has two main issues: firstly, we never detect the  $K = 0$  case, and secondly, in very noisy scenarios (low SNR), the estimation is not very accurate. To overcome these inaccuracies we perform a double consistency analysis. In order to extract the spikes from a long data stream, the signal is sequentially analysed with a sliding window. For each position of the window, we first estimate the number of spikes within the window, and we then extract the locations of the corresponding spikes. Figure 4.5 illustrates this procedure. If the window has size  $\Delta t$  and the window progresses by steps of  $t_{\text{step}}$ , the time interval processed within the  $i$ th window is

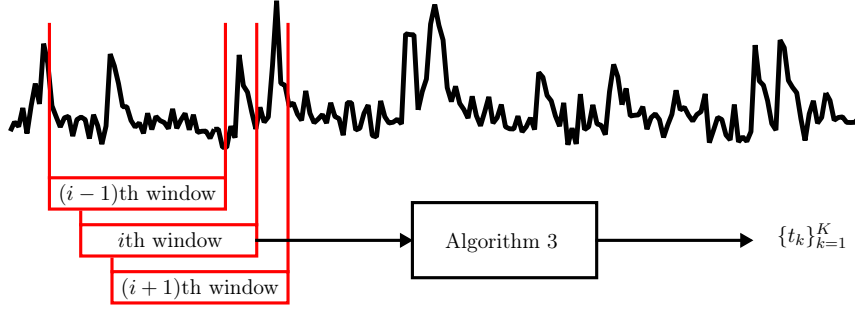
$$[t_0 + (i-1) t_{\text{step}}, \quad t_0 + (i-1) t_{\text{step}} + \Delta t), \quad (4.23)$$



**Figure 4.4.:** Singular values with noise, empirical threshold. Normalised singular values of Toeplitz matrices where the elements are given by a sum of exponentials with  $K = 2$  (a),  $K = 3$  (b) and  $K = 5$  elements (c). The horizontal axes represent different levels of noise and the index of the singular values. The vertical axis corresponds to the value of the normalised singular values. For each level of noise, 100 different realisations of noise have been generated. For each realisation of noise, the singular values of the matrix have been computed and normalised with respect to the largest one. The surface is obtained by averaging the normalised singular values over the different realisations for a particular level of noise. The red horizontal plane illustrates the threshold that has been empirically established at a value of 0.3.

where  $t_0$  is the instant of time of the first sample of the data stream. We select  $t_{\text{step}}$  to be equal to the temporal resolution of the data, so the window advances sample by sample. Consecutive windows overlap to guarantee that a spike is detected among different windows. Figure 4.6 illustrates this sequential processing of a real fluorescence sequence. In Figures 4.6(a) and 4.6(b) the red dots represent the retrieved locations for different positions of the sliding windows; the vertical axis represents the index of the window, and the horizontal axis the time location of the retrieved spikes. Figure 4.6(a) corresponds to a window size of 32 samples and Figure 4.6(b) to a window size of 8 samples. The blue lines represent the locations of the real spikes, this is the ground truth data. When a spike is detected among different windows, we can see that the red dots are aligned vertically because different windows output the same location.

The double consistency approach consists in running the algorithm twice following two different strategies in each execution. First, with a sufficiently large time window (32 samples of the input signal) we estimate the number of spikes from the singular values of the matrix  $\mathbf{S}$ . Second, with a sufficiently small window (8 samples of the input signal) we assume that we always have a single spike within this observation window. In both cases, for each position of the sliding window, the algorithm outputs the locations of the spikes assumed to be within that window. When the retrieved locations correspond to real spikes, the locations we retrieve are stable among the different positions of the window that capture these spikes, but when the locations correspond to noise they are not stable. We construct a joint histogram of the retrieved locations with the two different window sizes. This is shown in Figure 4.6(c). The locations of the real spikes are estimated from the peaks of the histogram. These peaks correspond to positions that are consistent among different windows. Figure 4.6(d) shows the fluorescence data with the real and the detected



**Figure 4.5.:** Fluorescence signal processing with a sliding window. For each time interval, the number of spikes within that interval is first estimated and then the location of each spike is retrieved.

spikes. The algorithm is summarised in Algorithm 3.

The sizes of the two windows of the double consistency approach have been empirically adjusted in order to maximise the performance of the algorithm. If larger windows are used more information about each spike is captured, however, the drawback of increasing the size of the window is that the estimate of the number of spikes that are present within the window becomes less accurate. It is thus important to keep these numbers within reasonable values. A window of size 32 resulted in a good detection rate but with a considerable amount of false positives. A window of size 8 presented a very low number of false positives but with the disadvantage of reducing the detection rate. The combination of the two lead to the best performances.

---

**Algorithm 3** FRI spike train inference (noisy scenario)

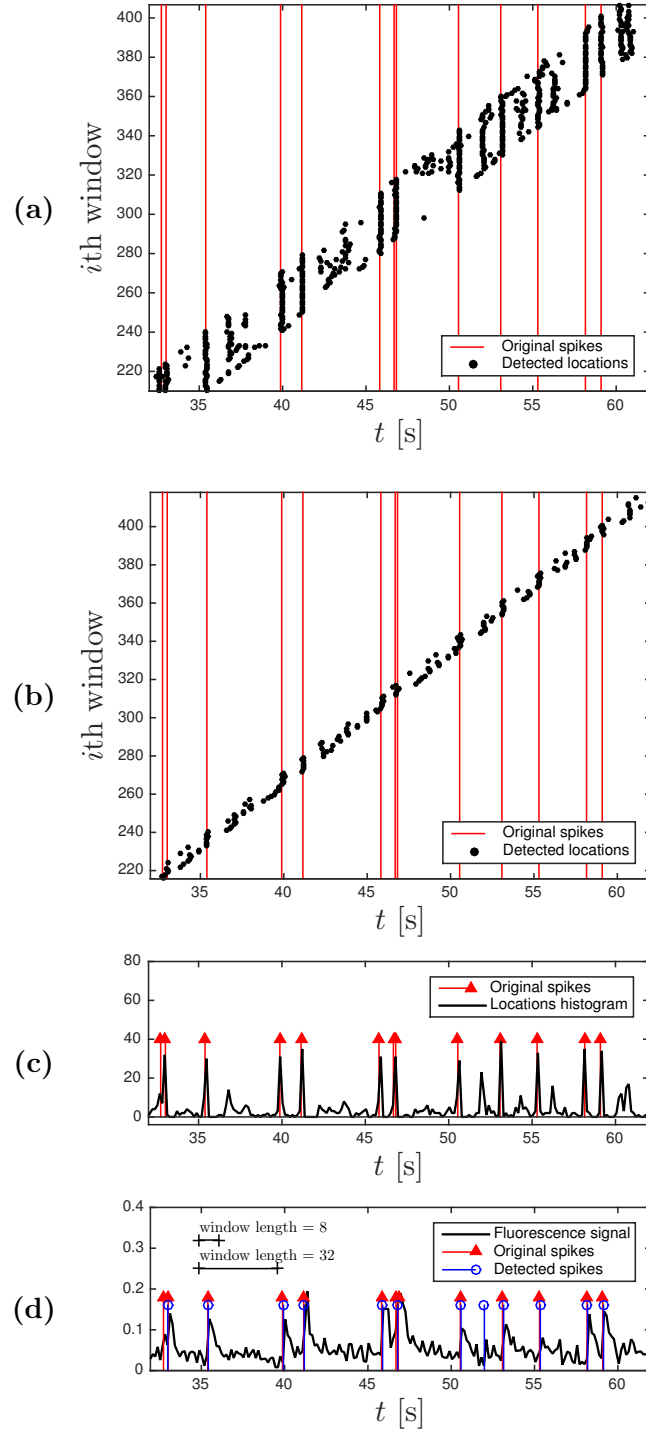
---

**Input:**  $\{c(n T_{\text{res}})\}_{n=i+1}^{i+N}$ :  $i$ th windowed calcium sequence ( $N = 32$  or  $8$ ).

Optional parameter  $K$ : number of spikes.

**Output:**  $\{t_k\}_{k=1}^K$ : spike locations.

- 1: Filter with exponential reproducing kernel:  $y[n] = \langle c(t), \varphi(t - n) \rangle$ .
  - 2: Compute weighted finite differences:  $z[n] = y[n] - y[n - 1] e^{-\alpha T}$ .
  - 3: Obtain new measurements:  $s[m] = \sum_n d_{m,n} z[n]$ .
  - 4: Create Toeplitz matrix  $\mathbf{S}$  from measurements  $s[m]$ .
  - 5: **if**  $K$  is not fixed **then**
  - 6:     Compute normalised singular values of  $\mathbf{S}$ .
  - 7:      $K$  is the number of singular values greater than 0.3.
  - 8: **end if**
  - 9: Create matrix  $\mathbf{S}_0$  from  $\mathbf{S}$  by dropping first row.
  - 10: Create matrix  $\mathbf{S}_1$  from  $\mathbf{S}$  by dropping last row.
  - 11: Retrieve  $\{u_k\}_{k=1}^K$  from the eigenvalues of the generalized eigenvalue problem  $\mathbf{S}_0 - \mu \mathbf{S}_1$ .
  - 12: Obtain  $\{t_k\}_{k=1}^K$  from  $u_k = e^{i\lambda t_k}$ .
-



**Figure 4.6.:** Double consistency spike search with real data. In (a) and (b) the dots represent the detected locations and the gray lines the locations of the original spikes for two different window sizes. In (a) the algorithm estimates the number of spikes within the sliding window (window size 32 samples). In (b) the algorithm assumes  $K = 1$  for each position of the sliding window (window size 8 samples). (c) shows the joint histogram of the detected locations. (d) shows the fluorescence signal with the original and the detected spikes.

#### 4.2.4. Generating surrogate data

In order to investigate the changes in performance of the spike detection algorithm in terms of parameters such as the signal-to-noise ratio and the sampling frequency, we have generated surrogate data with similar properties to the experimental data. We assume that the spike occurrence follows a Poisson distribution with parameter  $\lambda$  spikes/s. Experimental data presents occurrences between 0.45 and 0.5 spikes per second. The probability of having  $k$  spikes in the interval considered in parameter  $\lambda$  (one second) is given by the probability mass function of the Poisson distribution:

$$f_{\lambda}(k) = \frac{\lambda^k e^{-\lambda}}{k!}. \quad (4.24)$$

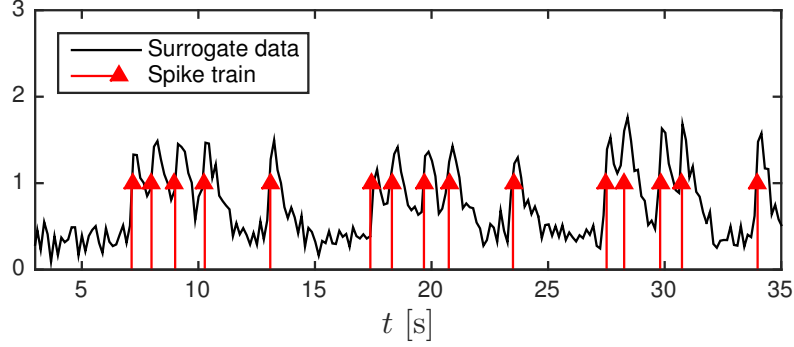
To generate a spike train for a time interval  $L$  we divide this interval in  $N$  slots. Each slot corresponds to a time interval of  $\Delta t = \frac{L}{N}$  seconds. The  $\lambda'$  parameter that corresponds to this new time interval is  $\lambda' = \lambda \Delta t$ . We then create a vector  $\mathbf{k} = (k_1, \dots, k_N)$  of size  $1 \times N$  where each  $k_i$  is a realisation of the independent random variables  $K_i \sim \text{Pois}(\lambda')$ . The  $i$ th element of this vector,  $k_i$ , gives the number of spikes that occurred during the  $i$ th time slot. We then have to generate the precise instant of time when the spike occurred. For a given time slot, we generate the  $k_i$  spike locations according to a uniform distribution. The total number of spikes in the time interval  $L$  is  $K = \sum_{i=1}^N k_i$ . Once we have generated the locations of the  $K$  spikes  $(t_k)_{k=1}^K$  the waveform given by the exponential decaying model is:

$$c(t) = A \sum_{k=1}^K e^{-\alpha(t-t_k)} \mathbf{1}_{t \geq t_k} \quad (4.25)$$

where  $\alpha = 1/\tau$ . We then generate the simulated fluorescence sequence sampling Equation (4.25) at instants  $t = nT_{\text{res}}$  for a temporal resolution of  $T_{\text{res}}$  seconds. The data sequence is slightly smoothed before sampling in order to have a differentiable function. We can then add white Gaussian noise to satisfy a certain SNR. The SNR is computed as the ratio between the power of the signal and the power of the noise, expressed in the logarithmic decibel scale. Figure 4.7 shows an example of generated data with a SNR of 15 dB.

#### 4.2.5. Real-time processing

The algorithm is fast enough to perform real-time spike inference. The most demanding stages in terms of computation requirements are the estimation of the number of spikes and the retrieval of the locations for each position of the sliding windows. The joint histogram's peak detection has a negligible complexity when compared to the previous stages. For each new data sample the algorithm has to perform the number of spikes estimation and locations retrieval for the 32 samples and 8 samples windows. Since previous locations are



**Figure 4.7.:** Surrogate data. Temporal resolution  $T_{\text{res}} = 147.2$  ms and  $SNR = 15$  dB.

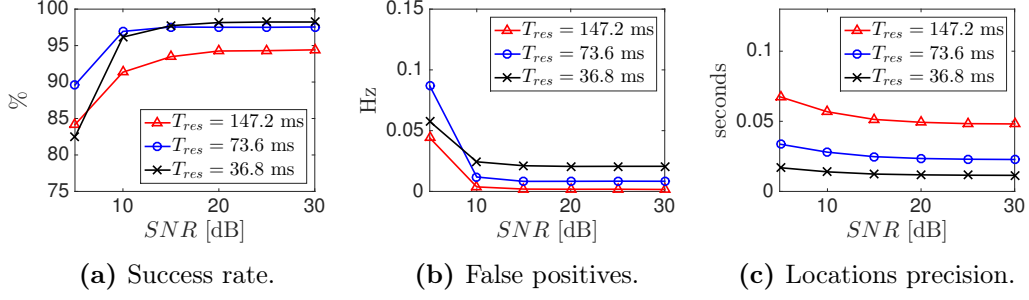
stored in memory, the histogram can be computed sequentially.

Performance measurements have been done for the current MATLAB implementation using a commercial laptop (tested on a 2.5 GHz Intel Core i5 CPU). In our setup, the 32 samples window takes on average (value obtained averaging the execution time of 1000 windows) 1.25 ms to perform the number of spikes estimation and location retrieval, and the 8 samples window takes 0.49 ms. Therefore, when a new data sample is available the algorithm takes 1.74 ms to process it. The sampling period is 147.2 ms, the current implementation can thus process up to 84 data streams in parallel. The algorithm requires the samples from a whole window before being able to output a location. Therefore the output has a maximum delay of  $32 \text{ samples} \times 147.2 \text{ ms/sample} = 4.71 \text{ s}$ .

### 4.3. Results

In this section we present the performance of the spike detection algorithm with real and surrogate data. The electrophysiological measurements give us a ground truth for the spiking activity of the monitored neurone which allows measuring the performance of the algorithm with real data. A detected spike is considered to correspond to a real spike if the difference between the real location and the estimated location is smaller than or equal to a threshold. We set this threshold to be equal to the temporal resolution of the data,  $T_{\text{res}}$ . If we denote by  $t_k$  the real location of a spike and  $\tilde{t}_j$  an estimated location, we consider that the real spike has been detected if  $\tilde{t}_j \in [t_k - T_{\text{res}}, t_k + T_{\text{res}}]$ . When a spike is assumed to correspond to a real spike, we can measure the error on the estimated location. From this error measurement we obtain a mean squared error (MSE) of the overall algorithm.

A limitation of the real data is the temporal resolution, which is imposed by the frame rate of the calcium imaging dataset. With the surrogate data we can control this resolution when we generate the data stream to measure the impact of this parameter to the algorithm's performance.



**Figure 4.8.:** Algorithm's performance measurement with surrogate data. The surrogate data contains 1000 spikes in a time interval of 2000 seconds. For each noise level, the experiment has been repeated for 100 different realisations of the noise. (a) The success rate is measured as the percentage of true spikes that have been correctly detected. (b) False positives are given as number of false positives per second (Hz). (c) Standard deviation of the retrieved locations with respect to the true locations.

#### 4.3.1. Real data

The real data is a datastream of 133 seconds with a temporal resolution  $T_{\text{res}} = 0.147$  s. Hence there are 903 samples. This data stream contains 62 spikes at a rate of 0.466 Hz.

The sliding window algorithm is performed twice, first with a big window of 32 samples estimating  $K$  from the estimated rank of the  $S$  matrix (thresholding of the singular values), and second with a small window of 8 samples and a fixed  $K = 1$ . The spikes are detected from the resulting histogram of the union of the locations retrieved in both iterations. The algorithm correctly detects 83.9 % of the spikes. The standard deviation of the locations is 0.0503 s. There are a total of 9 false positives, this corresponds to a false positive rate of 0.0598 Hz or 1.1 % if measured as the rate between false positives and total negative samples.

#### 4.3.2. Surrogate data

The real data presents a spike rate of 0.466 spikes per second. We have generated surrogate data assuming that the spike occurrence follows a Poisson distribution with a parameter  $\lambda = 0.5$  spikes/s and a total number of 1000 spikes. The noiseless calcium concentration signal has been generated once for a given spike distribution and with three different temporal resolutions. To analyse the performance variation for different levels of noise we have run the algorithm over 100 different realisations of noise for each level of SNR. Figure 4.8 summarises the obtained performances.

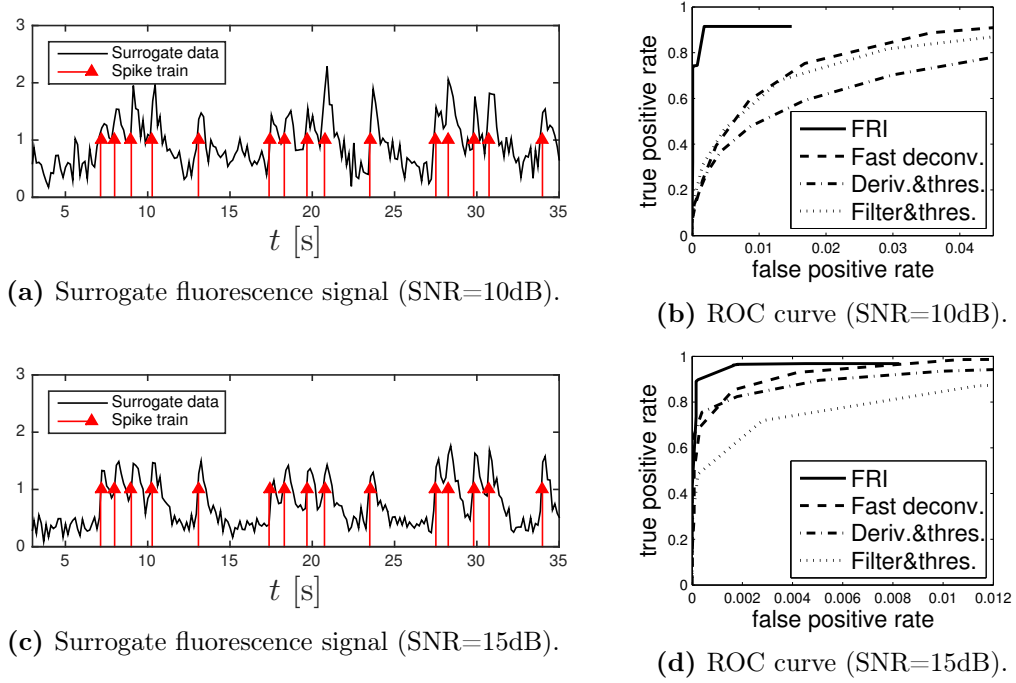
From Figure 4.8 it can be seen that the success rate of the algorithm depends strongly on the temporal resolution of the data. The higher the temporal resolution, the better the spike detection rate. The real data we have analysed presents a low temporal resolution because of the low frame rate of the calcium images ( $\frac{1}{0.147\text{s}} = 6.8$  Hz), but recent publications [89, 90] show that the acquisition techniques are improving, with in some situations

frame rates up to 125 Hz now available. At these frame rates, our algorithm presents success rates above 95 %. The performances of the detection algorithm are not particularly influenced by the noise for SNRs above 10 dB, and deteriorate slightly for lower SNRs. Increasing temporal resolution has a minor drawback, the amount of false positives slightly increases. However, the false positive rate is very low (about 15 false positives for a stream of 2000 seconds represents a rate of false positives below 0.01 Hz).

### 4.3.3. Comparison with existing methods

Various methods for spike inference from two-photon imaging have been presented in recent years, but to the best of our knowledge, none of them achieve these performances for real-time processing. [78] present a method based on finding a least-square solution to fit the observed fluorescence signal. With real data similar to ours, temporal resolution of 96 ms and neural activity with firing rate of 0.44 Hz, they obtain higher detection rates, 95 % detection of electrically confirmed AP with a false-positive rate of 0.012 Hz. However, this method is very slow and is not suitable for real-time processing. It also has to be noted that this data was acquired from cell bodies and ours from dendrites. [81] describe a new approach that combines principal component analysis and support vector machine. This method requires a learning phase to tune some parameters. The results show similar performances in terms of detection rate, with error rates  $< 10\%$ , but the precision of this method is lower as only a fraction of the detected spikes are detected in the correct time frame. [82] present a sequential Monte Carlo method to infer spike trains. Again, this method is not suitable for real-time processing due to its high computational complexity. [83] describe a fast nonnegative deconvolution filter to infer the most likely spike train given the fluorescence. The code that implements this method in MATLAB is publicly available and we have tested it with our data. The computational complexity of this method is comparable to ours. The output of this algorithm is a probability between 0 and 1 of having a spike in a given time frame. Thresholding this probability vector is how we decide if the neurone has been activated in a given time frame. The lower the threshold, the higher the detection rate, but this also increases the false positive rate.

Figure 4.9 presents receiver operating characteristic (ROC) curves in order to compare our algorithm (FRI) and the fast nonnegative deconvolution technique with surrogate data. We have also included simulation results for two other standard algorithms, derivative-thresholding and filter and thresholding. The latter method filters the fluorescence sequence with a derivative of a Gaussian filter in order to smooth the noise and detect spikes. All four methods have a thresholding stage to infer the spike train. A lower threshold provides a higher success rate but with the penalty of having more false positives. The simulations have been performed with the same spike train we generated to obtain the performance measurements in Figure 4.8 and with the same realisation of the noise in all four methods. We present the results for two different levels of noise. The two axis of the ROC



**Figure 4.9.:** Simulations showing FRI algorithm achieving better performances in spike train inference than the fast deconvolution technique from [83] and different filtering and thresholding approaches. (a) Surrogate data generated with a temporal resolution  $T_{\text{res}} = 147.2$  ms and  $SNR = 10$  dB. There are total of 1000 spikes with a rate of 0.5 spikes per second. (b) Receiver operating characteristic (ROC) curves comparing FRI (solid line), fast deconvolution (dashed line), derivative and thresholding (dashed-dotted line) and filtering and thresholding (dotted) techniques. (c) and (d) present the results of the same experiment in a lower noise scenario ( $SNR = 15$  dB). The  $x$  and  $y$  axis are unitless as they present a ratio between true positive or negative samples and obtained positive or negative samples.

curves are unitless as they present a ratio between true positive or negative samples and obtained positive or negative samples. The surrogate data contains 1000 true spikes and 13587 samples ( $2000 \text{ s} / T_{\text{res}}$ ). Thus an operating point with a false positive rate of 0.01 and a true positive rate of 0.9 correctly detects 900 spikes but throws 126 false positives. It can be observed that the FRI algorithm presents better performances although it has to be noted that the fast deconvolution algorithm is faster. The time required to process a 13600 samples stream (which corresponds to the 2000 seconds stream of surrogate data in Figure 4.9(a)) is around 3.85 seconds for the fast deconvolution algorithm and around 23.64 seconds for the FRI algorithm.

With real data, FRI achieves a success rate of 83.9 % (52 true spikes correctly detected out of 62) with only 9 false positives. To achieve similar success rates on the same data with the fast deconvolution method, we obtain more than 100 false positives, this is more false positives than true spikes. Derivative-thresholding presents more than 200 false positives for a success rate of 83.9 % and filter and threshold more than 110 false positives.

## 4.4. Summary

We have presented a novel spike inference technique based on FRI theory. Spikes are detected from calcium transients in fluorescence measurements. To do this, the existing FRI framework has been extended to a new class of signals that is formed by a stream of decaying exponentials. The data obtained in this type of measurements presents low temporal resolution and is corrupted with noise. To overcome these limitations we propose a sequential non-iterative algorithm that is able to detect spikes in real-time. The proposed algorithm achieves very high success rates with a low number of false positives. These promising results are a direct consequence of the fact that the fluorescence sequence can be parametrised as a signal recoverable in the FRI setup. FRI guarantees that the recovered signal is within a specific model, and this strong prior is what makes this algorithm very effective.

Techniques for spike train inference from two-photon imaging have begun attracting substantial attention in recent years due to the promise of being able to monitor spike trains from large numbers of localised neurones simultaneously. Improvements in acquisition techniques and increasing temporal resolution demand efficient spike inference algorithms to process all this information. Our algorithm is fast and parallelisable, and is thus well-suited to this context.



## Chapter 5.

# Sparsity according to Prony: average performance analysis and phase transition

In the previous chapters we have considered the problem of reconstructing continuous-time signals from sets of measurements. From now on we focus on reconstructing finite-dimensional discrete-time signals from a projection on lower dimensional spaces. The finite-dimensional counterpart of signals with finite rate of innovation are signals that have a sparse representation in some orthonormal basis or union of bases [91–93].

Finding the sparse representation of a finite-dimensional signal in an overcomplete dictionary has attracted a lot of attention over the past years, since many complex signals are not well represented in orthonormal bases and larger dictionaries can achieve sparser representations. Traditional approaches such as Basis Pursuit are based on relaxing the non-convex  $\ell_0$ -minimisation problem [94]. Recently, a new polynomial complexity algorithm, ProSparse, has been presented [95]. ProSparse solves the sparse representation problem when the dictionary is the union of Fourier and canonical bases and is based on Prony’s method. The original paper presents a bound on the sparsity that guarantees that the algorithm will find *all* the sparse representations that match a given observation provided that the sparsity is below this bound. This is a worst case scenario, and empirical evidence shows that the algorithm is able to find the correct sparse solutions with high probability well beyond the deterministic bound. Here, we present a probabilistic average-case analysis that characterises a sharp phase transition behaviour of the algorithm.

### 5.1. The sparse representation problem

Consider a matrix  $\mathbf{D} \in \mathbb{C}^{N \times L}$  with  $N < L$  and define the underdetermined linear system of equations

$$\mathbf{y} = \mathbf{D} \mathbf{x}, \tag{5.1}$$

---

<sup>0</sup>This chapter includes research conducted jointly with Prof Dragotti and Prof Lu who are the authors of the algorithm analysed in this chapter and has lead to the following publications [1, 5].

where  $\mathbf{y} \in \mathbb{C}^N$  corresponds to the observed signal, and  $\mathbf{x} \in \mathbb{C}^L$  to a representation of  $\mathbf{y}$  in the *dictionary*  $\mathbf{D}$ . The columns of the matrix  $\mathbf{D}$  are called atoms. The goal is to find an appropriate representation  $\mathbf{x}$  that matches the observation  $\mathbf{y}$ . The system in (5.1) has more unknowns than equations, and therefore, there exist infinitely many solutions if  $\mathbf{y}$  belongs to the span of the columns of  $\mathbf{D}$ , or there is no solution if  $\mathbf{y}$  does not belong to that span. We assume that the matrix  $\mathbf{D}$  has full rank  $N$  which avoids the no solution problem since then the columns of the matrix span the entire space  $\mathbb{C}^N$ .

We are interested in finding the sparsest solution, that is, a vector  $\mathbf{x}$  that has the fewest non-zero elements and satisfies (5.1). This is formalised as follows:

$$(P_0): \quad \arg \min_{\tilde{\mathbf{x}} \in \mathbb{C}^L} \|\tilde{\mathbf{x}}\|_0 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D} \tilde{\mathbf{x}}, \quad (5.2)$$

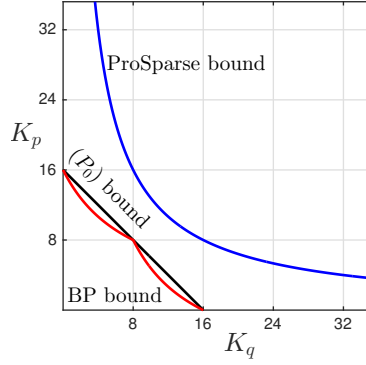
where  $\|\mathbf{x}\|_0 \stackrel{\text{def}}{=} |\{n : |x[n]| \neq 0\}|$  counts the non-zero elements in  $\mathbf{x}$  denoted by  $x[n]$ . Note that  $\|\cdot\|_0$  is not a norm because it does not satisfy the homogeneity property. Solving  $(P_0)$  directly is intractable since the  $\ell_0$ -“norm” is nonconvex. Moreover, for general dictionaries, finding the sparse representation is NP-hard [96].

One way to relax  $(P_0)$  is by replacing the  $\ell_0$ -“norm” with the  $\ell_1$ -norm which is convex. This leads to the following convex optimisation problem:

$$(P_1): \quad \arg \min_{\tilde{\mathbf{x}} \in \mathbb{C}^L} \|\tilde{\mathbf{x}}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D} \tilde{\mathbf{x}}. \quad (5.3)$$

This approach is known as Basis Pursuit and was proposed by Chen et al. in 1998 [94]. It is important to note that  $(P_1)$  can be solved using polynomial complexity algorithms [97]. The sparsity problem can also be solved with greedy algorithms such as Orthogonal Matching Pursuit [92, 98, 99] which is based on picking the atoms that are most correlated with the residue at each iteration. Greedy algorithms present a lower computational cost but are not designed to solve the optimisation problems mentioned above. However, if  $\mathbf{x}$  is sparse and the columns of  $\mathbf{D}$  are sufficiently incoherent, the OMP finds the sparsest solution [100]. In the next chapter we also present some other algorithms that are better suited for the noisy case. We refer the reader to Elad’s book [101] for a comprehensive overview of the sparse representation problem.

In this chapter we consider the problem of finding the  $K$ -sparse representation of a vector  $\mathbf{y}$  from the union of two dictionaries  $\Phi$  and  $\Psi$ , that is,  $\mathbf{D} = [\Phi, \Psi]$ . The use of union of dictionaries is motivated from applications such as image processing where patches of images can be represented in a more efficient manner if one dictionary accounts for the discontinuities and edges in the image and the other dictionary for oscillatory components that might arise in textures. If we express the vector  $\mathbf{x}$  in terms of the elements of each



**Figure 5.1.:** Comparison of bounds for sparse signal representation in union of Fourier and canonical bases for  $N = 256$ .

dictionary we have that

$$\mathbf{y} = \begin{bmatrix} \Phi & \Psi \end{bmatrix} \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_q \end{bmatrix} = \Phi \mathbf{x}_p + \Psi \mathbf{x}_q. \quad (5.4)$$

Let

$$K_p = \|\mathbf{x}_p\|_0 \quad \text{and} \quad K_q = \|\mathbf{x}_q\|_0 \quad (5.5)$$

be the number of atoms from  $\Phi$  and  $\Psi$  respectively. If we denote by  $K$  the overall sparsity of the vector  $\mathbf{x}$ , we have that

$$\|\mathbf{x}\|_0 = \|\mathbf{x}_p\|_0 + \|\mathbf{x}_q\|_0 = K_p + K_q = K. \quad (5.6)$$

Let  $\mu(\mathbf{D})$  denote the mutual coherence of a dictionary [100, 102]. This is defined as

$$\mu(\mathbf{D}) \stackrel{\text{def}}{=} \max_{1 \leq k, \ell \leq L, k \neq \ell} \frac{|\mathbf{d}_k^H \mathbf{d}_\ell|}{\|\mathbf{d}_k\|_2 \|\mathbf{d}_\ell\|_2}. \quad (5.7)$$

The mutual coherence characterises the dependency between the atoms of the dictionary  $\mathbf{D}$ . For the case of the overcomplete dictionary built from the union of Fourier and identity bases, it is easy to verify that  $\mu(\mathbf{D}) = 1/\sqrt{N}$ . For unitary matrices the mutual coherence is 0, but when the number of columns of  $\mathbf{D}$  is larger than the number of rows, that is  $L > N$ ,  $\mu(\mathbf{D})$  is strictly positive. For dictionaries that are the union of two unitary matrices we have that [103]

$$\frac{1}{\sqrt{N}} \leq \mu(\mathbf{D}) \leq 1. \quad (5.8)$$

We desire the smallest possible value to get a behaviour which is close to that exhibited by unitary matrices.

In 2001, Donoho and Huo analysed the sparse representation problem for the union of Fourier and canonical bases [102]. They showed that a  $K$ -sparse vector  $\mathbf{x}$  that satisfies the

observation model (5.1) is the unique solution to  $(P_0)$  when

$$K < \mu(\mathbf{D}) = \frac{1}{\sqrt{N}}. \quad (5.9)$$

They also showed that  $(P_0)$  and  $(P_1)$  are equivalent when

$$K < \frac{\sqrt{N}}{2}. \quad (5.10)$$

In 2002, these results were extended to generic orthogonal pairs of bases by Elad and Bruckstein [104]. The bound in (5.10) was improved to

$$2\mu(\mathbf{D})^2 K_p K_q + \mu(\mathbf{D}) \max\{K_p, K_q\} - 1 < 0. \quad (5.11)$$

The following year, Feuer and Nemirovsky showed that this bound is tight [105]. The expression in (5.11) is illustrated in red in Figure 5.1 and corresponds to the Basis Pursuit (BP) bound.

Recently, a new algorithm, ProSparse [95], has been presented for the case where the dictionary is the union of the Fourier and identity matrices. The ProSparse algorithm is a polynomial complexity algorithm that can recover  $\mathbf{x}$  from  $\mathbf{y} = [\mathbf{\Phi}, \mathbf{\Psi}]\mathbf{x}$ , provided that

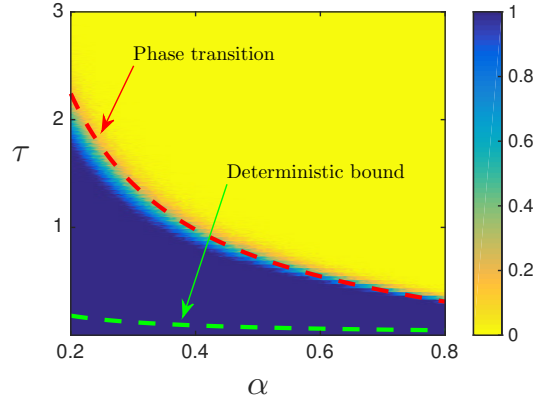
$$K_p K_q < \frac{N}{2}, \quad (5.12)$$

where  $K_p$  represents the number of non-zero elements from the Fourier matrix and  $K_q$  the number of atoms from the identity, leading to the overall sparsity level  $K = K_p + K_q$ . All these bounds are illustrated in Figure 5.1. This is a deterministic bound that guarantees perfect reconstruction when the sparsity constraint is satisfied. In this chapter, we analyse the probability of success of this approach when this sparsity constraint is relaxed.

ProSparse applies when  $\mathbf{\Phi}$  is a Vandermonde or a variation of a Vandermonde matrix of size  $N \times M$ . When  $M = N$ , this matrix models a basis, and a frame when  $M > N$ . The authors in [95] consider variations of Vandermonde matrices that correspond to the more general family of matrices that are characterised by

$$\mathbf{\Phi} = \mathbf{\Lambda} \mathbf{V} \mathbf{B}, \quad (5.13)$$

where  $\mathbf{\Lambda} \in \mathbb{C}^{N \times N}$  is a diagonal matrix,  $\mathbf{V} \in \mathbb{C}^{N \times M}$  is a Vandermonde matrix, and  $\mathbf{B} \in \mathbb{C}^{M \times M}$  is a matrix whose columns have sparse support. The other matrix  $\mathbf{\Psi}$  models a localised basis of size  $N \times N$ . Two specific instances of  $\mathbf{D}$  which are of particular interest to the work presented in this chapter are the case where  $\mathbf{D}$  is the union of the Fourier and the canonical bases, and the case where  $\mathbf{D}$  is the union of a Fourier frame of size  $N \times M$  with  $M > N$  and the canonical basis.



**Figure 5.2.:** Probability of success of ProSparse and phase transition. The colourmap represents probabilities computed via Monte Carlo simulation for  $N = 10^6$ . For each pair  $(\alpha, \tau)$ , 100 different spike locations have been generated uniformly at random. The number of spikes is given by  $K_q = \lfloor \alpha N \rfloor$ . At each point and for each realization of spike locations the existence of a maximum gap of size at least  $\tau \log N$  is checked. The dashed lines depict the deterministic bound and the predicted phase transition of ProSparse.

Simulations show that the algorithm is able to reconstruct the sparse vector well beyond the deterministic bound. Moreover, for high dimension, we observe a sharp phase transition in the behaviour of the algorithm. The main contribution of the work presented here is the precise characterisation of this phase transition. This characterisation is tight, that is, the algorithm succeeds with high probability when the sparsity is below a certain threshold, but fails with high probability when the sparsity goes beyond this threshold. Specifically, for large  $N$ , if there are  $K_q = \alpha N$  spikes, ProSparse can recover

$$K_p < \frac{-\log N}{2\log(1-\alpha)} \quad (5.14)$$

Fourier atoms with high probability and fails to recover

$$K_p > \frac{-\log N}{2\log(1-\alpha)} \quad (5.15)$$

atoms with high probability. A formal statement and proof of this phenomenon is given in Section 5.4. Figure 5.2 is an illustration of the phase transition phenomenon where the colourmap shows probabilities measured via Monte Carlo simulations for very large  $N$ . The  $x$ -axis represents the number of spikes as a fraction of  $N$ , that is,  $K_q = \lfloor \alpha N \rfloor$ , and the  $y$ -axis the number of Fourier atoms as  $K_p = \lfloor \tau \log N \rfloor$ . If the region is blue it means that ProSparse is able to reconstruct the sparse vector, if it is yellow, ProSparse fails. The green line depicts the curve  $\tau = 1/(2\alpha \log N)$ , which corresponds to the deterministic bound given in (5.12) when we replace  $K_p$  by  $\tau \log N$  and  $K_q$  by  $\alpha N$ . The guaranteed perfect reconstruction region is the area below this curve, which is much more pessimistic than the actual performance of the algorithm. The red curve depicts the phase transition

curve  $-1/2 \log(1 - \alpha)$ . It is clear that this expression predicts the location of this phase transition and identifies the region where the algorithm succeeds with high accuracy.

The characterisation of phase transition phenomena is also an active research topic in areas such as convex optimisation and high dimensional geometry [106–109]. Although related to the problem studied in this chapter, these results do not apply directly to our setup since they consider scenarios where the equivalent of the dictionary  $\mathbf{D}$  presents some type of randomness.

The rest of the chapter is organised as follows: Section 5.2 presents a brief overview of ProSparse, in Section 5.3 we present an average performance analysis of ProSparse by providing an exact formula to compute the probability of existence of gaps of a maximum size between consecutive spikes. Section 5.4 contains the main result of this work, that is, a proof of a phase transition behaviour on the probability of success of ProSparse for large  $N$ . Section 5.5 provides a numerical validation and simulation results where ProSparse is compared to Basis Pursuit. We conclude in Section 5.6.

## 5.2. Overview of ProSparse

ProSparse can be applied to any pair of bases whenever one of the bases can be expressed in terms of a Vandermonde matrix (or a variation of a Vandermonde matrix) and the other is *local* [95]. A typical example of this type of dictionary is the union of Fourier and identity matrices. In this section we provide a short overview of the algorithm for this scenario.

Let  $\mathbf{y} \in \mathbb{C}^N$  be a complex-valued finite-dimensional signal that has a  $K$ -sparse representation in an overcomplete dictionary formed by Fourier atoms and spikes. In matricial form this can be written as

$$\mathbf{y} = \mathbf{D} \mathbf{x}, \quad (5.16)$$

where  $\mathbf{D} = [\mathbf{F}, \mathbf{I}]$  and  $\mathbf{x}$  is  $K$ -sparse, that is, it has only  $K$  non-zero elements. The matrix  $\mathbf{F} \in \mathbb{C}^{N \times M}$  corresponds to a Fourier basis or frame and  $\mathbf{I} \in \mathbb{C}^{N \times N}$  to the identity matrix. The number of atoms in the dictionary  $\mathbf{D}$  is thus  $L = M + N$ . We further assume that  $K_p$  non-zero elements are due to Fourier atoms and that  $K_q = K - K_p$  elements are due to spikes. Given the synthesis model in (5.16) the  $n$ th entry of  $\mathbf{y}$  can also be written as follows:

$$y[n] = \frac{1}{\sqrt{N}} \sum_{k=1}^{K_p} a_k \exp(i\omega_k n) + \sum_{k=1}^{K_q} b_k \delta[n - n_k], \quad n = 0, 1, \dots, N-1, \quad (5.17)$$

where  $a_k, b_k \in \mathbb{C} \setminus \{0\}$ ,  $\omega_k = \frac{2\pi m_k}{M}$ ,  $0 \leq m_1 < \dots < m_{K_p} < M$  and  $0 \leq n_1 < \dots < n_{K_q} < N$ . If we consider the case where the overcomplete dictionary corresponds to the union of Fourier and identity bases, then  $M = N$  and  $\mathbf{x} \in \mathbb{C}^{2N}$ . When the dictionary is given by the union of a Fourier frame and the identity basis we have that  $M > N$  and  $\mathbf{x} \in \mathbb{C}^{M+N}$ .

In the case of the union of bases, it is shown in [95] that there is a polynomial complexity algorithm, *ProSparse*, that finds all  $(K_p, K_q)$ -sparse signals  $\mathbf{x}$  such that

$$\mathbf{y} = \mathbf{D} \mathbf{x} \quad \text{and} \quad K_p K_q < N/2. \quad (5.18)$$

This is a deterministic bound that guarantees perfect reconstruction when the sparsity constraint  $K_p K_q < N/2$  is satisfied and the observation  $\mathbf{y}$  is not corrupted with noise.

ProSparse is based on the fact that since one basis is local, many of the entries in  $\mathbf{y}$  are affected only by Fourier atoms. Specifically, we can find *clean* windows of consecutive entries of  $\mathbf{y}$  that are only due to Fourier atoms and that can be expressed as

$$y[n] = \frac{1}{\sqrt{N}} \sum_{k=1}^{K_p} a_k \exp(i \omega_k n) = \sum_{k=1}^{K_p} \alpha_k u_k^n, \quad (5.19)$$

where we restrict the temporal index  $n$  to the clean window,  $\alpha_k \stackrel{\text{def}}{=} a_k/\sqrt{N}$  and  $u_k \stackrel{\text{def}}{=} \exp(i \omega_k)$ . If such a window has at least  $2K_p$  consecutive entries, we can estimate the  $K_p$  pairs of parameters  $(\alpha_k, u_k)$  using Prony's method. Recall that this method has been presented in Chapter 2. Once the Fourier atoms are estimated from a clean window, we can build the residual vector by removing their contribution from the original signal. If the residual satisfies the sparsity constraint, it directly corresponds to the estimation of the spikes and therefore the algorithm has succeeded in retrieving the sparse vector.

The overall algorithm operates by performing an exhaustive search over all possible sliding windows  $y[\ell], y[\ell+1], \dots, y[\ell+2K_p-1]$  of size  $2K_p$  and for all the sparsity levels that satisfy the constraint (5.12). Specifically, for each  $\ell$  and  $K_p$ , the following Toeplitz matrix is built:

$$\mathbf{T}_{K_p, \ell} \stackrel{\text{def}}{=} \begin{bmatrix} y[\ell+K_p] & y[\ell+K_p-1] & \dots & y[\ell] \\ y[\ell+K_p+1] & y[\ell+K_p] & \dots & y[\ell+1] \\ \vdots & \vdots & \ddots & \vdots \\ y[\ell+2K_p-1] & y[\ell+2K_p-2] & \dots & y[\ell+K_p-1] \end{bmatrix}. \quad (5.20)$$

The Fourier atoms are then estimated applying Prony's method to  $\mathbf{T}_{K_p, \ell}$ . When the dictionary is the union of the Fourier and identity bases, ProSparse further assumes that  $K_p \leq K_q$  since the cases where  $K_p > K_q$  can be obtained from the dual, that is, the sparse vector can also be reconstructed from  $\overline{\mathbf{F}^H} \mathbf{y} = [\mathbf{I}, \mathbf{F}] \bar{\mathbf{x}}$ . In this dual case, the spikes become complex exponentials with amplitude  $b_k^*/\sqrt{N}$  and the Fourier atoms become spikes. The entire method is described in Algorithm 4.

*Remark 1.* When dealing with the Fourier basis, the problem becomes circular due to the periodicity of the exponentials  $e^{i2\pi n/N}$ . This means that the search of the clean windows can be performed as if the entry  $n = N-1$  was immediately followed by the entry  $n = 0$ .

---

**Algorithm 4** *ProSparse*—Prony’s based sparsity for union bases (Fourier-identity) [95]

---

**Input:** An observed vector  $\mathbf{y} \in \mathbb{C}^N$ .

**Output:** A set  $\mathcal{S}$  with all  $(K_p, K_q)$ -sparse signal  $\mathbf{x}$  that satisfies (5.12), with  $K_p \leq K_q$ .

Initialize  $\mathcal{S} = \{[\mathbf{0}^T, \mathbf{y}^T]^T\}$ . This is a trivial solution, with  $K_p = 0$  and  $K_q = \|\mathbf{y}\|_0$ .

**for**  $K_p = 1, 2, \dots, \lceil \sqrt{N/2} - 1 \rceil$  **do**

**for**  $\ell = 0, 1, \dots, N - 1$  **do**

        Build the Toeplitz matrix  $\mathbf{T}_{K_p, \ell}$  as in (5.20).

        Apply Prony’s method on  $\mathbf{T}_{K_p, \ell}$  to find parameters  $(\alpha_k, u_k)$ , where  $0 \leq k < K_p$ .

**if**  $\{u_k\}$  contains  $K_p$  distinct values, with each  $u_k \in \{e^{i2\pi m/N} : m \in \mathbb{Z}\}$  **then**

            Compute the estimated Fourier contribution  $\tilde{\mathbf{y}} = \left( \sum_{k=1}^{K_p} \alpha_k u_k^n \right)_{n=0}^{N-1}$ .

            Compute the residual  $\mathbf{r} = \mathbf{y} - \tilde{\mathbf{y}}$  and let  $K_q = \|\mathbf{r}\|_0$ .

**if**  $K_p \leq K_q$  and  $K_p K_q < N/2$  **then**

                Obtain the sparse signal  $\mathbf{x}$  from  $\tilde{\mathbf{y}}$  and  $\mathbf{r}$ .

$\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{x}\}$ .

**end if**

**end if**

**end for**

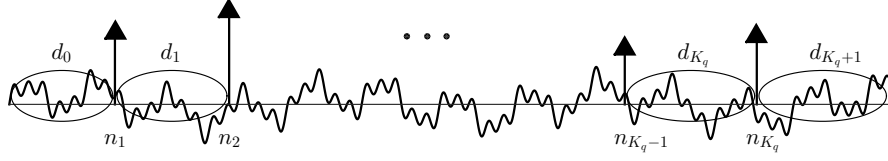
**end for**

---

When the sparsity satisfies (5.12), there always exists at least one clean window, and there might even be a large number of them. This constraint on the sparsity is a worst-case bound, since we can think of counter examples with  $K_p K_q \geq N/2$  where these clean windows do not exist. However, for sparsity levels which are small but are beyond the deterministic bound, these counter examples have a negligible probability of occurrence. The next section provides the exact probability for the existence of these clean gaps.

### 5.3. Exact probabilistic average-case analysis of ProSparse

ProSparse operates by first recovering  $K_p$  Fourier atoms from a *clean* window of uncorrupted samples of  $\mathbf{y}$  and then retrieving  $K_q$  spikes from the residual, that is, the difference between  $\mathbf{y}$  and the recovered Fourier atoms. The  $K_p$  Fourier atoms are recovered by applying Prony’s method and this requires a clean window of at least  $2K_p$  consecutive samples. In this section, we provide an exact formula to compute the probability of finding this clean window when the sparsity constraint is not satisfied. Note that when the sparsity constraint  $K_p K_q < N/2$  is satisfied this probability is equal to one. This expression is then specialised to determine the probability of success of ProSparse for the Fourier frame and the Fourier basis cases.



**Figure 5.3.:** Samples  $y[n]$  located between consecutive spikes are only due to the  $K_p$  Fourier atoms. The sum of all the gaps  $d_k$ ,  $k = 0, 1, \dots, K_q$ , is equal to the size of the vector  $\mathbf{y}$ ,  $N$ , minus the number of spikes,  $K_q$ . ProSparse is able to recover  $K_p$  Fourier atoms if there is a gap of size at least  $2K_p$ .

### 5.3.1. Probability model

For given  $N$  and  $K_q$ , we select  $K_q$  distinct spike locations uniformly at random. Let

$$\sigma \stackrel{\text{def}}{=} \{n_1, n_2, \dots, n_{K_q}\}, \quad (5.21)$$

be the set of spike locations, where  $0 \leq n_1 < n_2 < \dots < n_{K_q} < N$ . ProSparse is able to recover  $K_p$  Fourier atoms if there exists a gap between consecutive spikes of at least  $2K_p$  samples. For any choice  $\sigma$  of  $K_q$  distinct locations, we define the  $K_q + 1$  gaps as follows

$$d_k \stackrel{\text{def}}{=} n_{k+1} - n_k - 1, \quad (5.22)$$

for  $k = 0, \dots, K_q$ , where we assume  $n_0 = -1$  and  $n_{K_q+1} = N$ . Note that the sum of all the  $d_k$  is equal to  $N - K_q$ . These gaps are illustrated in Figure 5.3 where we can observe that the samples between consecutive spikes are only due to Fourier atoms.

Let

$$\Delta \stackrel{\text{def}}{=} \max_{0 \leq k \leq K_q} \{d_k\} \quad (5.23)$$

be the maximum gap between consecutive spikes when circularity is not taken into account, that is, when we are in the Fourier frame case, and let

$$\Gamma \stackrel{\text{def}}{=} \max \left\{ \max_{1 \leq k < K_q} \{d_k\}, d_0 + d_{K_q} \right\} \quad (5.24)$$

be the maximum gap between consecutive spikes when circularity is considered, which corresponds to the Fourier basis case.

### 5.3.2. Exact probability for maximum gap size

The following two propositions provide the exact probability that the maximum gaps  $\Delta$  and  $\Gamma$  are smaller than a given value  $S$ .

**Proposition 4** (Non-circular case). *For a given problem of size  $N$ , and  $K$  distinct locations drawn uniformly at random, let  $\Delta$  be the maximum gap between consecutive locations*

defined as in (5.23). The exact probability that  $\Delta < S$ , for some constant  $S$ , is given by

$$F_N(K, S) \stackrel{\text{def}}{=} \mathbb{P}\{\Delta < S\} = \frac{\overbrace{(f_S[n] * \dots * f_S[n])}^{K+1 \text{ times}}|_{n=N-K}}{\binom{N}{K}}, \quad (5.25)$$

where

$$f_S[n] = \begin{cases} 1, & \text{if } 0 \leq n < S, \\ 0, & \text{otherwise.} \end{cases} \quad (5.26)$$

*Proof.* If we consider the experiment of drawing  $K$  spike locations uniformly at random, we are interested in the probability that the maximum gap between consecutive spikes is smaller than  $S$ . This probability is given by the ratio between the number of outcomes with  $\Delta < S$  and the total number of possible outcomes. The second number is given by the binomial coefficient  $\binom{N}{K}$ . The number of outcomes with  $\Delta < S$  corresponds to the sets  $\sigma$  defined as in (5.21) that satisfy  $\max_{0 \leq k \leq K} \{d_k\} < S$ , where the integers  $d_k$  are defined as in (5.22). This quantity can also be computed as the cardinality of the following set

$$\{(d_0, d_1, \dots, d_K) \in \{0, 1, \dots, S-1\}^{K+1} : \sum_{k=0}^K d_k = N-K\}. \quad (5.27)$$

This set provides all possible combinations of gaps such that the maximum gap is always smaller than  $S$  (note that all  $d_k$  are between 0 and  $S-1$ ) and the total sum of the gaps corresponds to a valid spikes distribution since the sum is equal to  $N-K$ . This counting problem can be solved by constructing i.i.d. discrete uniform random variables  $d_0, \dots, d_K$  supported on  $\{0, 1, \dots, S-1\}$ , and by computing the probability that the sum of these random variables is equal to  $N-K$ . The cardinality of this set is then given by  $S^{K+1} \mathbb{P}\{\sum_k d_k = N-K\}$ . Since the  $d_k$  are uniformly distributed, the probability mass function (PMF) of the random variables  $d_k$  is given by  $\mathbb{P}\{d_k = n\} = f_S[n]/S$ . Since the PMF of the sum of random variables corresponds to the convolution of their PMF, this cardinality is given by  $S^{K+1} (f_S[n]/S * \dots * f_S[n]/S)|_{n=N-K}$ . This leads to the term in the numerator:

$$S^{K+1} \mathbb{P}\left\{\sum_k d_k = N-K\right\} = S^{K+1} \overbrace{\frac{f_S[n]}{S} * \dots * \frac{f_S[n]}{S}}^{K+1 \text{ times}}|_{n=N-K} \quad (5.28)$$

$$= \overbrace{(f_S[n] * \dots * f_S[n])}^{K+1 \text{ times}}|_{n=N-K}. \quad (5.29)$$

□

**Proposition 5** (Circular case). *For a given problem of size  $N$ , and  $K$  distinct locations drawn uniformly at random, let  $\Gamma$  be the maximum gap between consecutive locations defined*

as in (5.24). The exact probability that  $\Gamma < S$ , for some constant  $S$ , is given by

$$G_N(K, S) \stackrel{\text{def}}{=} \mathbb{P}\{\Gamma < S\} = \frac{(g_S[n] * \overbrace{f_S[n] * \dots * f_S[n]}^{K-1 \text{ times}})|_{n=N-K}}{\binom{N}{K}}, \quad (5.30)$$

where

$$g_S[n] = \begin{cases} n+1, & \text{if } 0 \leq n < S, \\ 0, & \text{otherwise,} \end{cases} \quad (5.31)$$

and  $f_S[n]$  is defined as in Proposition 4.

*Proof.* The proof is analogous to the proof for Proposition 4. The term of the numerator is given by the cardinality of the following set:

$$\{(d_0, d_1, \dots, d_K) \in \{0, 1, \dots, S-1\}^{K+1} : \sum_{k=0}^K d_k = N-K \text{ and } d_0 + d_K \leq S-1\}. \quad (5.32)$$

The additional constraint  $d_0 + d_K \leq S-1$  is required to impose the maximum gap size restriction when circularity is considered. In order to take this constraint into account in the cardinality of the set, we consider that the sum of 2 out of the  $K+1$  i.i.d. discrete uniform random variables must be strictly smaller than  $S$ . This leads to the function  $g_S[n]$  which comes from convolving  $f_S[n]$  with itself and restricting the indices  $n$  from 0 to  $S-1$ . The remaining  $K-1$  gaps are considered by convolving  $K-1$  times the function  $f_S[n]$ .  $\square$

### 5.3.3. Probability of success of ProSparse

When the Fourier atoms in  $\mathbf{y}$  come from a Fourier frame, circularity cannot be applied. The probability of success of ProSparse is therefore given by the probability that there is at least one gap of size larger than or equal to  $2K_p$ . We thus have the following result:

$$\mathbb{P}\{\text{ProSparse (Fourier frame) succeeds}\} = \mathbb{P}\{\Delta \geq 2K_p\} \quad (5.33)$$

$$= 1 - F_N(K_q, 2K_p), \quad (5.34)$$

where  $F_N(K_q, 2K_p)$  is the function defined in Proposition 4 with  $K_q$  spikes and a gap  $S = 2K_p$ .

When the Fourier atoms in  $\mathbf{y}$  come from a Fourier basis, that is, the matrix  $\mathbf{F}$  in the dictionary  $\mathbf{D} = [\mathbf{F}, \mathbf{I}]$  is a unitary Fourier matrix, the problem becomes circular since the elements in vector  $\mathbf{y} \in \mathbb{C}^N$  have an underlying periodicity of period  $N$  samples. Moreover, the problem can also be solved from the dual vector  $\overline{\mathbf{F}^H \mathbf{y}} = [\mathbf{I}, \mathbf{F}] \overline{\mathbf{x}}$ . In the dual vector, spikes lead to complex exponentials and Fourier atoms lead to spikes. Thus, ProSparse succeeds either if there is a gap of size at least  $2K_p$  samples between the  $K_q$  locations of the spikes or, in the dual domain, if there is a gap of size at least  $2K_q$  samples between the

$K_p$  locations of the Fourier atoms. Consequently, the probability of success of ProSparse is given by:

$$\mathbb{P}\{\text{ProSparse (Fourier basis) succeeds}\} = 1 - G_N(K_q, 2K_p) G_N(K_p, 2K_q), \quad (5.35)$$

where  $G_N(K_q, 2K_p)$  is the function defined in Proposition 5 with  $K_q$  spikes and a gap  $S = 2K_p$ .

## 5.4. Asymptotic analysis and phase transition

Given the expression in (5.25) for the function  $F_N(K, S)$ , we are now interested in understanding the behaviour of the maximum gap  $\Delta$  in the asymptotic regime, that is, when  $N \rightarrow \infty$ . The probability given by the function  $G_N(K, S)$  in (5.30) is slightly smaller than  $F_N(K, S)$  because of circularity. However, in the asymptotic regime this difference is negligible and therefore the asymptotic analysis focuses on the more general case of the Fourier frame. We also assume that  $K_q = \lfloor \alpha N \rfloor$  for fixed  $\alpha \in (0, 1)$ . If we want to reconstruct a sparse vector with  $K_p$  Fourier atoms and  $K_q$  spikes, ProSparse presents a phase transition in the probability of success which is characterised by the following proposition:

**Proposition 6.** *Let  $\mathbf{y} \in \mathbb{C}^N$  be a mixture of  $K_p$  Fourier atoms and  $K_q$  spikes. If  $K_q = \alpha N$  and  $K_p = -(1 - \varepsilon) \frac{\log N}{2 \log(1 - \alpha)}$ , then, for any  $\varepsilon > 0$  the probability that ProSparse succeeds goes to 1 as  $N \rightarrow \infty$ . Similarly, if  $K_q = \alpha N$  and  $K_p = -(1 + \varepsilon) \frac{\log N}{2 \log(1 - \alpha)}$ , then, for any  $\varepsilon > 0$  the probability that ProSparse succeeds goes to 0 as  $N \rightarrow \infty$ .*

*Proof.* The proof is structured in two parts. The first part provides a lower bound on the probability of success. This bound tends to 1 when the number of Fourier atoms is below a threshold. The second part provides an upper bound that tends to 0 when the number of Fourier atoms goes beyond the same threshold. We can therefore conclude that there is a phase transition in the probability of success.

*Lower bound:* Proposition 4 proves that the probability of success can be obtained convolving  $K_q + 1$  times the function  $f_S[n]$  defined as in (5.26) and evaluating the resulting function at  $n = N - K_q$ .

Getting a closed-form expression of an entry of a discrete-time sequence that results from various discrete convolutions is not an easy task. However, we note that the convolution of two discrete sequences can also be obtained by computing the product of their generating functions. These generating functions are polynomial representations where the coefficients of the polynomials correspond to the discrete sequences [110]. Therefore, the probability we are after is also given by the coefficient of order  $N - K_q$  of the polynomial that results from raising to the  $K_q + 1$  power the polynomial  $p_S(x) = 1 + x + x^2 + \dots + x^{S-1}$ , where  $p_S(x)$  is the generating function of the discrete sequence  $f_S[n]$ . To refer to the  $p$ th order

coefficient of a polynomial of order  $n$ , we adopt the notation used in [111]:

$$\text{coef} \{(a_0 + a_1 x + \dots + a_n x^n), x^p\} = a_p, \quad (5.36)$$

with  $0 \leq p \leq n$ . We can therefore write the probability of success in terms of the  $N - K_q$  coefficient of the generating function:

$$\mathbb{P} \{\Delta \geq S\} = 1 - \mathbb{P} \{\Delta < S\} \quad (5.37)$$

$$= 1 - \frac{\text{coef} \left\{ (1 + x + \dots + x^{S-1})^{K_q+1}, x^{N-K_q} \right\}}{\binom{N}{K_q}}. \quad (5.38)$$

Since the polynomial has positive coefficients, it is easy to verify that ([111, Appendix D])

$$\text{coef} \left\{ (1 + x + \dots + x^{S-1})^{K_q+1}, x^{N-K_q} \right\} \leq \frac{(1 + x + \dots + x^{S-1})^{K_q+1}}{x^{N-K_q}}, \quad \forall x > 0. \quad (5.39)$$

The polynomial of order  $S - 1$  is a geometric series, and therefore, can be replaced by  $(1 - x^S)/(1 - x)$ , which leads to

$$\frac{(1 + x + \dots + x^{S-1})^{K_q+1}}{x^{N-K_q}} = \frac{(1 - x^S)^{K_q+1}}{x^{N-K_q} (1 - x)^{K_q+1}}. \quad (5.40)$$

Setting  $x = 1 - \alpha$  yields

$$\text{coef} \left\{ (1 + x + \dots + x^{S-1})^{K_q+1}, x^{N-K_q} \right\} \quad (5.41)$$

$$\leq \frac{[1 - (1 - \alpha)^S]^{K_q+1}}{(1 - \alpha)^{N-K_q} \alpha^{K_q+1}} \quad (5.42)$$

$$= e^{-(N-K_q) \log(1-\alpha) - (K_q+1) \log \alpha} [1 - (1 - \alpha)^S]^{K_q+1} \quad (5.43)$$

$$= \frac{1}{\alpha} e^{N H(\alpha)} [1 - (1 - \alpha)^S]^{K_q+1}, \quad (5.44)$$

where  $H(\alpha) \stackrel{\text{def}}{=} -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha)$  is the binary entropy function.

From Stirling's formula, we have that

$$\binom{N}{K_q} = \binom{N}{\alpha N} \geq \frac{\sqrt{2\pi}}{e^2} [\alpha(1 - \alpha)N]^{-1/2} e^{N H(\alpha)}. \quad (5.45)$$

Applying (5.44) and (5.45) to the expression in (5.38) we obtain

$$\mathbb{P} \{\Delta \geq S\} \geq 1 - \frac{e^2}{\sqrt{2\pi}} \sqrt{\frac{(1 - \alpha)N}{\alpha}} [1 - (1 - \alpha)^S]^{K_q+1}. \quad (5.46)$$

If we set again  $S = \tau \log N$  we have that

$$\left[1 - (1 - \alpha)^{\tau \log N}\right]^{K_q+1} = e^{(K_q+1) \log [1 - (1 - \alpha)^{\tau \log N}]} \quad (5.47)$$

$$= e^{(K_q+1) \log [1 - N^{\tau \log(1-\alpha)}]} \quad (5.48)$$

$$= e^{-(\alpha + \frac{1}{N}) N^{1+\tau \log(1-\alpha)} (1+o(1))}. \quad (5.49)$$

If  $\tau < -1/\log(1 - \alpha)$ , this last expression tends to 0 as  $N \rightarrow \infty$ . Therefore, we have proved the following lower bound: For  $\lfloor \alpha N \rfloor$  spikes distributed uniformly at random in distinct locations over a grid of size  $N$ , the probability of having a maximum gap between consecutive spikes of size at least  $\tau \log N$  satisfies

$$\lim_{N \rightarrow \infty} \mathbb{P} \{ \Delta \geq \tau \log N \} = 1, \quad \text{if } \tau < \frac{-1}{\log(1 - \alpha)}. \quad (5.50)$$

*Upper bound:* We have that

$$\mathbb{P} \{ \Delta \geq S \} = \mathbb{P} \left\{ \bigcup_{k=0}^{K_q} \{d_k \geq S\} \right\} \quad (5.51)$$

$$\leq \sum_{k=0}^{K_q} \mathbb{P} \{d_k \geq S\}, \quad (5.52)$$

where the right hand side in (5.51) represents the probability of the union of sets of events  $d_k \geq S$ , that is, the probability that at least one of the  $d_k$  is greater than or equal to  $S$ , and (5.52) follows from the union bound. From the symmetry of the problem we have that

$$\sum_{k=0}^{K_q} \mathbb{P} \{d_k \geq S\} = (K_q + 1) \mathbb{P} \{d_0 \geq S\}. \quad (5.53)$$

This last probability is given by  $\binom{N-S}{K_q} / \binom{N}{K_q}$ , since the event of having the first gap of at least size  $S$  is equivalent to having the  $K_q$  spikes in the last  $N - S$  locations. It follows that

$$\sum_{k=0}^{K_q} \mathbb{P} \{d_k \geq S\} = (K_q + 1) \frac{(N - K_q)! (N - S)!}{N! (N - K_q - S)!} \quad (5.54)$$

$$= (K_q + 1) \frac{(N - K_q) \dots (N - K_q - S + 1)}{N(N - 1) \dots (N - S + 1)}. \quad (5.55)$$

By defining the following parameter,  $\beta \stackrel{\text{def}}{=} 1 - \frac{K_q}{N}$ , we can rewrite the previous expression

as follows,

$$\sum_{k=0}^{K_q} \mathbb{P}\{d_k \geq S\} = (K_q + 1) \frac{(\beta - 0)(\beta - \frac{1}{N}) \dots (\beta - \frac{S-1}{N})}{(1 - 0)(1 - \frac{1}{N}) \dots (1 - \frac{S-1}{N})} \quad (5.56)$$

$$= e^{\log(K_q+1)} e^{\sum_{i=0}^{S-1} f(\frac{i}{N})}, \quad (5.57)$$

where  $f(x) \stackrel{\text{def}}{=} \log \frac{\beta-x}{1-x}$ . We also have that

$$\sum_{k=0}^{K_q} \mathbb{P}\{d_k \geq S\} = e^{\log(K_q+1)+f(0)+[\sum_{i=1}^{S-1} f(\frac{i}{N}) \frac{1}{N}] N} \quad (5.58)$$

$$\leq e^{\log(K_q+1)+f(0)+N \int_0^{(S-1)/N} f(x) dx}. \quad (5.59)$$

Here, we have used the fact that for the values of  $x$  we are considering inside the integral in (5.59) (positive and smaller than  $\beta$ ), the ratio  $\frac{\beta-x}{1-x}$  is positive, decreasing with  $x$ , and smaller than 1. Therefore,  $f(x) = \log \frac{\beta-x}{1-x}$  is negative and monotonically decreasing. It follows that the summation  $\sum_{i=1}^{S-1} f(\frac{i}{N}) \frac{1}{N}$  is upper bounded by the integral  $\int_0^{(S-1)/N} f(x) dx$ .

Let

$$g(x) \stackrel{\text{def}}{=} \int_0^x f(\tau) d\tau, \quad (5.60)$$

then  $g(x)$  is smooth around  $x = 0$ . Moreover,  $g(0) = 0$ ,  $g'(0) = f(0) = \log \beta$  and  $g''(0) = f'(0) = 1 - \frac{1}{\beta}$ . The Taylor expansion at  $x = \frac{S-1}{N}$  is given by

$$g\left(\frac{S-1}{N}\right) = (\log \beta) \frac{S-1}{N} + \frac{1-1/\beta}{2} \frac{(S-1)^2}{N^2} (1+o(1)). \quad (5.61)$$

Inserting (5.61) in (5.59) and noting that  $f(0) = \log \beta$  yields

$$\mathbb{P}\{\Delta \geq S\} \leq e^{\log(K_q+1)} e^{S \log \beta + \frac{1-1/\beta}{2N} (S-1)^2 (1+o(1))}. \quad (5.62)$$

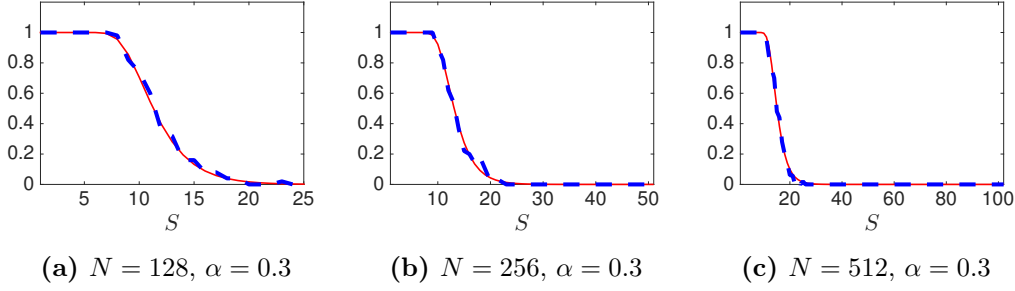
If we set  $S = \tau \log N$  and replace the values of  $K_q$  and  $\beta$  by  $\alpha N$  and  $1 - \alpha$  respectively, we have that

$$e^{\log(K_q+1)+S \log \beta} e^{\frac{1-1/\beta}{2N} (S-1)^2 (1+o(1))} \quad (5.63)$$

$$= (\alpha N + 1) e^{\tau \log N \log(1-\alpha)} e^{\frac{\alpha}{2N(\alpha-1)} (\tau \log N - 1)^2 (1+o(1))} \quad (5.64)$$

$$= \left(\alpha + \frac{1}{N}\right) N^{1+\tau \log(1-\alpha)} e^{\frac{\alpha}{2N(\alpha-1)} (\tau \log N - 1)^2 (1+o(1))}. \quad (5.65)$$

If  $\tau > -1/\log(1-\alpha)$  the above expression tends to 0 as  $N \rightarrow \infty$ . We thus have proved the following upper bound: For  $\lfloor \alpha N \rfloor$  spikes distributed uniformly at random in distinct locations over a grid of size  $N$ , the probability of having a maximum gap between consecutive



**Figure 5.4.:**  $\mathbb{P}\{\Delta \geq S\}$ , exact probability *vs* Monte Carlo simulation for different sizes of the problem. The exact probability (red continuous line) is computed using the expression in Proposition 4. The Monte Carlo approximation (blue dashed line) is obtained by randomly generating 50 realisations of the spike locations for each pair  $(K_q = \lfloor \alpha N \rfloor, S)$  and checking the existence of a maximum gap of size at least  $S$ .

spikes of size at least  $\tau \log N$  satisfies

$$\lim_{N \rightarrow \infty} \mathbb{P}\{\Delta \geq \tau \log N\} = 0, \quad \text{if } \tau > \frac{-1}{\log(1 - \alpha)}. \quad (5.66)$$

The threshold of the lower bound in (5.50) and the upper bound in (5.66) coincide and are equal to  $-1/\log(1 - \alpha)$ . Since ProSparse requires a gap  $\Delta$  of size at least  $2K_p$  to succeed, the phase transition characterised in Proposition 6 has been proved.  $\square$

## 5.5. Numerical validation and simulations

Proposition 4 provides an expression to compute the probability of failure or success of the ProSparse algorithm. This expression is exact, and for a given problem of size  $N$ , and sparsity levels  $(K_p, K_q)$  we can compute this probability by making use of the discrete convolution formula. However, this expression does not give an insight into how the performance of the algorithm evolves for different values of these parameters. Moreover, when  $N$  gets large, this computation becomes very slow making an exhaustive analysis of the type presented in Figure 5.2, where  $N = 10^6$ , intractable.

Monte Carlo simulations provide a reliable alternative that allow us to compute these probabilities in an efficient manner. Figure 5.4 compares the exact and Monte Carlo approaches for  $N = 128, 256$  and  $512$ . The red continuous line depicts the exact probabilities and the blue dashed line the probabilities computed via the Monte Carlo method with 100 realisations per point. It is clear that the approximated probabilities provide a faithful picture of the evolution of the true probability. Moreover, we can observe that for a number of spikes fixed to be a fraction of  $N$ , that is,  $K_q = \lfloor \alpha N \rfloor$ , the transition from the region of high probability of success to the region of low probability of success becomes sharper as  $N$  increases and this behaviour is well characterised in both, the exact and approximated

curves. This phenomenon is the phase transition of the asymptotic regime that is given in Proposition 6. The curves in Figure 5.4 can also be seen as vertical cuts of the probabilities that are represented in Figure 5.2.

### 5.5.1. Simulation results

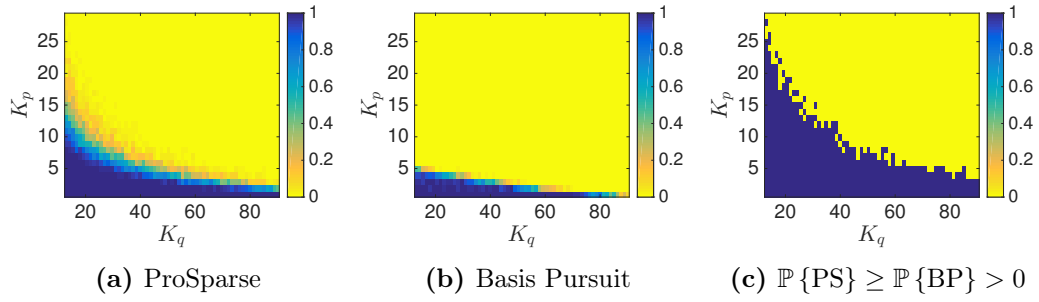
Figure 5.5 presents simulation results where ProSparse is compared to Basis Pursuit for the union of Fourier frame and identity case. The frame has a size of  $M = 2048$  times  $N = 128$  and the identity matrix is square with size  $N \times N$ . Thus, the sparse vector  $\mathbf{x}$  has a dimension of  $L = M + N = 2176$  and the observation  $\mathbf{y}$  is of size  $N = 128$ . The vertical and horizontal axes correspond to the sparsity of the vector  $\mathbf{x}$  in each dictionary. Performances of both algorithms are compared by generating 50 different realisations of the sparse vector  $\mathbf{x}$  for each sparsity combination  $(K_p, K_q)$ . The locations of the non-zero elements in  $\mathbf{x}$  are obtained by drawing random permutations from the sets  $\{1, 2, \dots, M\}$  and  $\{1, 2, \dots, N\}$  and keeping the first  $K_p$  and  $K_q$  elements respectively for the location of the Fourier atoms and spikes. The amplitudes of these elements are complex-valued and the real and imaginary parts are drawn from  $\mathcal{N}(0, 1)$ .

The ProSparse results in Figure 5.5(a) are obtained by checking that the maximum gap between consecutive spikes is at least equal to  $2K_p$ . When this is the case, the Fourier atoms can be recovered and therefore perfect reconstruction of the vector  $\mathbf{x}$  is achieved. The Basis Pursuit results in Figure 5.5(b) are obtained by solving the  $\ell_1$ -minimisation problem  $(P_1)$  in (5.3) with CVX, a MATLAB package for specifying and solving convex programs [112]. The success of the algorithm is measured by computing  $\|\tilde{\mathbf{x}} - \mathbf{x}\|_2^2 / \|\mathbf{x}\|_2^2$ , where  $\tilde{\mathbf{x}}$  is the reconstructed sparse vector, and checking that this value is below a negligible value of  $10^{-5}$ .

Both algorithms present a phase transition behaviour since the 2D plane is clearly split in two regions, one where the algorithms achieve perfect reconstruction with very high probability, and the other where the algorithms fail almost surely. It is also clear that in this scenario, ProSparse outperforms Basis Pursuit since it is able to recover the sparse vectors in a larger region of the  $(K_p, K_q)$  plane. This is confirmed in Figure 5.5(c) where we can see that the probability that ProSparse succeeds is always larger than or equal to the probability that Basis Pursuit succeeds.

## 5.6. Summary

We considered the problem of finding the sparse representation of a signal in the union of two dictionaries. A recently proposed polynomial complexity algorithm, *ProSparse*, is guaranteed to find all the sparse representations for the union of Fourier and identity matrices when the constraint  $K_p K_q < N/2$  is satisfied. Here, we have presented a probabilistic analysis of the probability of success of this algorithm when the sparsity constraint is not



**Figure 5.5.:** Comparison between ProSparse and Basis Pursuit. Union of Fourier frame of size  $2048 \times 128$  and identity  $128 \times 128$ . Probabilities obtained via Monte Carlo simulation. For each pair  $(K_p, K_q)$  50 different realisations of the locations of the atoms have been generated. The amplitudes of the atoms are complex-valued and the real and imaginary parts are drawn from  $\mathcal{N}(0, 1)$ . (c) presents the comparison between the two algorithms for the region where the probabilities are non-zero.

satisfied. We have provided an expression for the exact probability and showed that there is a phase transition in the asymptotic regime.

## Chapter 6.

# Sparse pattern recovery in the presence of noise

The previous chapter presents an average performance analysis of ProSparse, an algorithm to solve the sparsity recovery problem using Prony's method. In this chapter, we present a variation of this algorithm to solve the sparsity problem when noise is present. ProSparse is based on finding a clean window between consecutive spikes where Prony's method is applied to estimate first the Fourier atoms. Then, the spikes are retrieved from the difference between the observation and the estimated Fourier atoms. This approach is not well suited for the noisy case since the search of these clean windows becomes unreliable when the samples are corrupted with noise.

Inspired by the Cadzow denoising algorithm that was introduced in Chapter 2, we present here a novel algorithm that is able to solve the sparsity problem in the presence of noise. Our approach outperforms state of the art algorithms such as Basis Pursuit Denoise [94] and Subspace Pursuit [113] when the dictionary is the union of Fourier and identity matrices. The algorithm is based on the fact that the Cadzow denoising algorithm is well suited to remove noise from a sequence that is given by a sum of exponentials. Moreover, this algorithm is also capable of removing spikes from the observed sequence. We take advantage of this feature to design an algorithm that removes the spikes iteratively to obtain a robust estimate of the Fourier atoms. The spikes can be estimated in the same way from the Fourier transform of the observed signal. The algorithm has low complexity compared to state of the art algorithms for sparse recovery since it relies on the Fast Fourier Transform (FFT) algorithm.

### 6.1. Introduction

In this chapter we consider the same problem of the previous chapter, that is, finding the sparse representation of a given signal. However, we now consider the scenario where the observation has been corrupted with additive noise.

Let the noiseless observed signal be given by

$$y[n] = \frac{1}{\sqrt{M}} \sum_{k=1}^{K_p} a_k \exp\left(i \frac{2\pi}{M} m_k n\right) + \sum_{k=1}^{K_q} b_k \delta[n - n_k], \quad 0 \leq n < N, \quad (6.1)$$

where  $0 \leq m_1 < \dots < m_{K_p} < M$  and  $0 \leq n_1 < \dots < n_{K_q} < N$  are integers that correspond to the indices of the atoms that form the observed signal and  $a_k, b_k \in \mathbb{C} \setminus \{0\}$  their amplitudes. When noise is present, the observed signal is given by

$$\tilde{y}[n] = y[n] + \varepsilon[n], \quad 0 \leq n < N, \quad (6.2)$$

where  $\varepsilon[n] = \varepsilon^R[n] + i \varepsilon^I[n]$  models complex-valued noise. We assume that  $\varepsilon^R[n]$  and  $\varepsilon^I[n]$  are i.i.d. random variables for  $0 \leq n < N$ . We also assume that the noise is Gaussian with the real and imaginary parts drawn from  $\mathcal{N}(0, \sigma_\varepsilon^2)$ .

The case where  $M = N$  corresponds to the union of bases, and the case  $M > N$  corresponds to the union of a Fourier frame and the identity matrix. In both cases, the observed signal can be written in matricial form as follows:

$$\mathbf{y} = \mathbf{D} \mathbf{x} + \boldsymbol{\varepsilon} = \begin{bmatrix} \mathbf{F} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_q \end{bmatrix} + \boldsymbol{\varepsilon} \quad (6.3)$$

where  $\mathbf{y}, \boldsymbol{\varepsilon} \in \mathbb{C}^N$ ,  $\mathbf{x}_p \in \mathbb{C}^M$ ,  $\mathbf{x}_q \in \mathbb{C}^N$  and  $\mathbf{x} \in \mathbb{C}^L$ , where  $L = M + N$ .

The Compressed Sensing framework [114–116] presents a wide range of algorithms to solve the noisy sparsity recovery problem. Here, we briefly present the traditional methods that are still in use nowadays and some of the more recent algorithms that are well suited for our context. We refer the interested reader to the books by Foucart and Rauhut [117] or by Eldar and Kutyniok [118] for comprehensive descriptions and recent applications of such techniques.

In [94], where Basis Pursuit was first presented, the noisy scenario was also analysed and an extension named Basis Pursuit Denoise (BPDN) was described. BPDN relaxes the constraint of the  $(P_1)$  problem defined in (5.3) as follows

$$(BPDN) : \quad \arg \min_{\tilde{\mathbf{x}} \in \mathbb{C}^L} \|\tilde{\mathbf{x}}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D} \tilde{\mathbf{x}}\|_2^2 < \eta, \quad (6.4)$$

where  $\eta$  is an error tolerance that depends on the power of the noise. In [119], Tibshirani et al. presented the Least Absolute Shrinkage and Selection Operator (LASSO) method that minimises the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. For a particular regularisation parameter  $\lambda > 0$ , the LASSO estimate is equivalent to the unconstrained version given by

$$\min_{\tilde{\mathbf{x}}} \frac{1}{2} \|\mathbf{y} - \mathbf{D} \tilde{\mathbf{x}}\|_2^2 + \lambda \|\tilde{\mathbf{x}}\|_1. \quad (6.5)$$

A more recent approach named Gradient Projection for Sparse Recovery (GPSR) [120] is based on formulating the sparse recovery problem as a quadratic program by splitting vector  $\mathbf{x}$  into its positive and negative parts:  $\mathbf{x} = \mathbf{u} - \mathbf{v}$ , where  $\mathbf{u} = (u[n])_{n=0}^{L-1}, u[n] \geq 0$  and  $\mathbf{v} = (v[n])_{n=0}^{L-1}, v[n] \geq 0$ . The resulting quadratic program is given by

$$\min_{\mathbf{z}} \mathbf{c}^T \mathbf{z} + \frac{1}{2} \mathbf{z}^T \mathbf{B} \mathbf{z} \stackrel{\text{def}}{=} F(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{z} \geq 0, \quad (6.6)$$

where  $\mathbf{z} = [\mathbf{u}^T, \mathbf{v}^T]^T$ ,  $\mathbf{c} = \tau \mathbf{1}_{2N} + [-\mathbf{y}^T \mathbf{D}, \mathbf{y}^T \mathbf{D}]$ , with  $\mathbf{1}_{2N} = [1, \dots, 1]^T$ , and

$$\mathbf{B} = \begin{bmatrix} \mathbf{D}^T \mathbf{D} & -\mathbf{D}^T \mathbf{D} \\ -\mathbf{D}^T \mathbf{D} & \mathbf{D}^T \mathbf{D} \end{bmatrix}. \quad (6.7)$$

In this formulation all the vectors and matrices are assumed to be real. This approach can be easily extended to the complex case by splitting the atoms of the dictionary  $\mathbf{D}$  and the elements in vectors  $\mathbf{x}$  and  $\mathbf{y}$  into their real and complex parts. The basic idea of GPSR is to iterate  $\mathbf{z}^{(i)}$  along the negative gradient  $-\nabla F(\mathbf{z}^{(i)})$ .

The Orthogonal Matching Pursuit (OMP) algorithm [92, 100, 121], briefly mentioned in the previous chapter, naturally extends to the noisy case by modifying the tolerance parameter that is used as a stopping condition (see Algorithm 5 for a full description of OMP). If we denote by  $\mathbf{D}_{\mathcal{S}^{(k)}}$  and  $\tilde{\mathbf{x}}_{\mathcal{S}^{(k)}}$  the matrix and vector that result from respectively selecting the columns and elements in  $\mathbf{D} \in \mathbb{C}^{N \times L}$  and in  $\tilde{\mathbf{x}} \in \mathbb{C}^L$  given by the indices  $\mathcal{S}^{(k)} \subseteq \{1, \dots, L\}$ , the step in OMP at the  $k$ th iteration where the new solution is updated can be written as

$$\mathbf{x}^{(k)} = \arg \min_{\tilde{\mathbf{x}} \in \mathbb{C}^L} \|\mathbf{D}_{\mathcal{S}^{(k)}} \tilde{\mathbf{x}}_{\mathcal{S}^{(k)}} - \mathbf{y}\|_2^2. \quad (6.8)$$

This is a least squares problem, and the solution is obtained by solving the following system of normal equations:

$$\mathbf{D}_{\mathcal{S}^{(k)}}^H \mathbf{D}_{\mathcal{S}^{(k)}} \tilde{\mathbf{x}}_{\mathcal{S}^{(k)}} = \mathbf{D}_{\mathcal{S}^{(k)}}^H \mathbf{y} \quad \Leftrightarrow \quad \mathbf{D}_{\mathcal{S}^{(k)}}^H \underbrace{(\mathbf{D}_{\mathcal{S}^{(k)}} \tilde{\mathbf{x}}_{\mathcal{S}^{(k)}} - \mathbf{y})}_{= -\mathbf{r}^{(k)}} = \mathbf{0}. \quad (6.9)$$

We thus have that the residual at iteration  $k$  is orthogonal to the elements of the dictionary that have been selected up to this iteration, that is, the elements in  $\mathbf{D}_{\mathcal{S}^{(k)}}$ . This is the reason why the OMP algorithm has the word *orthogonal* in its name. The tolerance parameter imposes a maximum to the norm of this residual. More recently, a new set of greedy algorithms have been developed that present lower complexity than OMP with similar performance guarantees: Compressive Sampling Matching Pursuit (CoSaMP) [122] and Subspace Pursuit [113]. Both algorithms are also based on updating iteratively the support of the sparse vector from the residual. However, in each iteration the candidate locations that are added to the support is larger than in OMP, where only one location is added at each iteration. The reconstruction complexity of OMP is  $\mathcal{O}(KLN)$  while the complexity

**Algorithm 5** OMP—Orthogonal Matching Pursuit [92, 98, 99]

**Input:** Dictionary  $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_L] \in \mathbb{C}^{N \times L}$ , observation  $\mathbf{y} \in \mathbb{C}^N$  and error threshold  $\eta$  [optional argument, maximum number of iterations  $K_{\max}$ ].

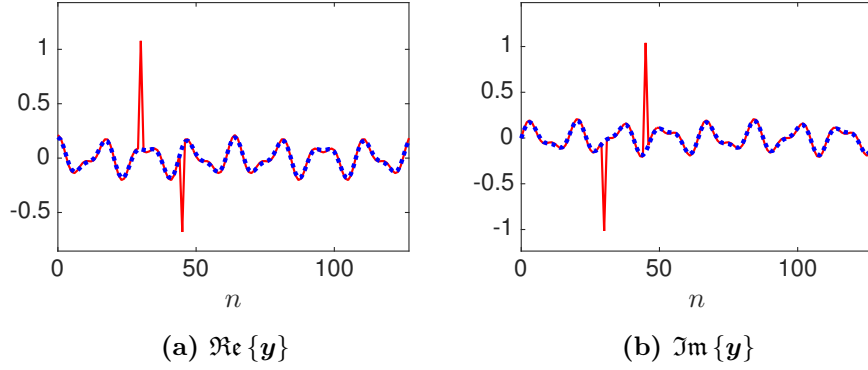
**Output:** Sparse vector  $\mathbf{x} \in \mathbb{C}^L$ .

- 1: Initialise index  $k = 0$ .
- 2: Initialise solution  $\mathbf{x}^{(0)} = \mathbf{0}$ .
- 3: Initialise residual  $\mathbf{r}^{(0)} = \mathbf{y} - \mathbf{D} \mathbf{x}^{(0)} = \mathbf{y}$ .
- 4: Initialise support  $\mathcal{S}^{(0)} = \emptyset$ .
- 5: **while**  $\|\mathbf{r}^{(k)}\|_2^2 > \eta$  [optional:  $k < K_{\max}$ ] **do**
- 6:    $k \leftarrow k + 1$
- 7:   Compute  $e[i] = \|z[i] \mathbf{d}_i - \mathbf{r}^{(k-1)}\|_2^2$  for  $i \in \{1, \dots, L\} \setminus \mathcal{S}^{(k-1)}$ , where  $z[i] = \frac{\mathbf{d}_i^H \mathbf{r}^{(k-1)}}{\|\mathbf{d}_i\|_2^2}$ .
- 8:   Find index  $i_0 = \arg \min_{i \in \{1, \dots, L\} \setminus \mathcal{S}^{(k-1)}} \{e[i]\}$ .
- 9:   Update support  $\mathcal{S}^{(k)} = \mathcal{S}^{(k-1)} \cup \{i_0\}$ .
- 10:   Compute solution  $\mathbf{x}^{(k)} = \arg \min_{\tilde{\mathbf{x}} \in \mathbb{C}^L} \|\mathbf{D} \tilde{\mathbf{x}} - \mathbf{y}\|_2^2$  subject to  $\text{supp}\{\tilde{\mathbf{x}}\} = \mathcal{S}^{(k)}$ .
- 11:   Update residual  $\mathbf{r}^{(k)} = \mathbf{y} - \mathbf{D} \mathbf{x}^{(k)}$ .
- 12: **end while**

of CoSaMP and Subspace Pursuit are  $\mathcal{O}(L \log^2 L)$  and  $\mathcal{O}(NL \log K)$  respectively. There also exist other families of algorithms based on iterative thresholding methods such as the Normalized Iterative Hard Thresholding (NIHT) [123, 124].

Inspired by the denoising strategies that are applied in the Finite Rate of Innovation framework, here we present an extension of the sparse recovery algorithm that has been analysed in detail in the previous chapter for the noisy case. The Cadzow signal enhancement algorithm [55] is a generic strategy that seeks to denoise a signal before applying an actual signal recovery algorithm. The approach is based on imposing sequentially, at each iteration, a set of properties that the signal is hypothesised to possess. For the observed signal given as in (6.1) and (6.2), if we assume that the spikes are also part of the noise, we can apply Cadzow's algorithm to try to recover the original Fourier atoms. This is the principle that is applied in the algorithm that is presented in this chapter.

The rest of the chapter is organised as follows: Section 6.2 presents the Cadzow denoising algorithm particularised to our setup. Section 6.3 describes the novel noisy sparse recovery algorithm. Section 6.4 is an analysis of the algorithm from a probabilistic point of view to establish a guaranteed performance condition for the simple case where there is one Fourier atom and one spike. Section 6.5 presents simulation results where the algorithm is compared against state of the art algorithms for sparse recovery. We then conclude in Section 6.6.



**Figure 6.1.:** Cadzow denoise for spike removal. The thin lines in (a) and (b) depict the the real and imaginary parts respectively of the observed signal  $\mathbf{y}$  that is made of two complex exponentials and two complex-valued spikes. The dotted lines illustrate the denoised signal where we can appreciate that the spikes have been successfully removed.

## 6.2. Finite-dimensional Cadzow and circulant matrices

In our context, the Cadzow algorithm is used to denoise samples that are given by a sum of exponentials corrupted with additive noise. If we consider that the spikes are also part of the noise, the noisy samples  $\tilde{y}[n]$  can be written as follows:

$$\tilde{y}[n] = \sum_{k=1}^{K_p} \alpha_k u_k^n + \eta[n], \quad (6.10)$$

where the signal of interest corresponds to the sum of  $K_p$  exponentials and the term  $\eta[n]$  includes the spikes and the actual Gaussian noise. The Cadzow algorithm is based on building the following Toeplitz matrix:

$$\tilde{\mathbf{Y}}^{toe} = \begin{bmatrix} \tilde{y}[M] & \tilde{y}[M-1] & \dots & \tilde{y}[0] \\ \tilde{y}[M+1] & \tilde{y}[M] & \dots & \tilde{y}[1] \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{y}[P] & \tilde{y}[P-1] & \dots & \tilde{y}[P-M] \end{bmatrix}, \quad (6.11)$$

where the matrix is built with a number of rows and columns which are larger than the number of exponentials  $K_p$ , and is made as square as possible. Then, the following two properties are imposed sequentially at each iteration:

1. Find the closest matrix of rank  $K_p$  (in the Frobenius norm sense) by computing the SVD of  $\tilde{\mathbf{Y}}^{toe}$  and setting to zero the smallest singular values.
2. Impose a Toeplitz structure by averaging the diagonal elements.

These two properties are applied until a stopping condition is reached. This condition can be a maximum number of iterations or the difference between the singular values of

different iterations being below a threshold. An interesting property of this algorithm is that it can also be used to remove the spikes of a signal that is made of Fourier atoms and spikes. Figure 6.1 illustrates the removal of the spikes by applying Cadzow to a signal that is made of two Fourier atoms and two spikes. The dotted line depicts the resulting *denoised* signal which still represents faithfully the Fourier atoms and where we can see that the spikes have been removed.

### 6.2.1. The union of bases case, fast circulant Cadzow algorithm

In our particular case, the parameters  $u_k$  are complex-valued and lie on a grid of size  $M$  on the unit circle:

$$u_k = \exp\left(i\frac{2\pi}{M}m_k\right), \quad k = 1, \dots, K_p, \quad (6.12)$$

with  $0 \leq m_1 < \dots < m_{K_p} < M$ . Moreover, when the Fourier atoms in  $\mathbf{y} \in C^N$  come from the Fourier basis we have that  $M = N$ . In this case, since we have access to  $N$  samples  $\tilde{y}[n]$ , and due to the periodicity of the Fourier atoms, the Toeplitz matrix can be extended to the following noisy circulant matrix of size  $N \times N$ :

$$\tilde{\mathbf{Y}}^{circ} = \begin{bmatrix} \tilde{y}[0] & \tilde{y}[N-1] & \dots & \tilde{y}[1] \\ \tilde{y}[1] & \tilde{y}[0] & \dots & \tilde{y}[2] \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{y}[N-1] & \tilde{y}[N-2] & \dots & \tilde{y}[0] \end{bmatrix}. \quad (6.13)$$

This matrix is of full rank  $N$  due to the presence of noise. The *noiseless* matrix  $\mathbf{Y}^{circ}$  with samples  $y[n]$  that are only due to the complex exponentials, satisfies the rank deficiency and the Toeplitz structure properties that are required to apply Cadzow denoising algorithm. Therefore, we can also apply the Cadzow strategy to denoise the circulant matrix  $\tilde{\mathbf{Y}}^{circ}$ . It is easy to verify that the noiseless matrix is of rank  $K_p$  since it can be diagonalised as follows:

$$\mathbf{Y}^{circ} = \mathbf{F} \mathbf{\Lambda} \mathbf{F}^H, \quad (6.14)$$

where  $\mathbf{\Lambda} = \text{diag}(\hat{y}[m])_{m=0}^{N-1}$  and  $\hat{y}[m] = \text{DFT}_N \{y[n]\} = \sum_{n=0}^{N-1} y[n] \exp(i2\pi mn/N)$  has exactly  $K_p$  non-zero elements at  $m = m_k$ . Due to the circulant structure of this matrix, it follows that imposing the first property to  $\tilde{\mathbf{Y}}^{circ}$  (rank deficiency) yields a circulant matrix, and therefore the resulting algorithm is not iterative. That is, we do not need to impose the second property since it is already satisfied and the Cadzow denoising algorithm stops after imposing the first property in the first iteration. The resulting denoising algorithm is described in Algorithm 6. This algorithm is fast since it only needs to compute one DFT and one inverse DFT which are performed using the Fast Fourier Transform (FFT) algorithm. The complexity of this algorithm is  $\mathcal{O}(N \log N)$ . This is considerably better than the complexity of the original Cadzow approach since it requires computing an SVD

at each iteration which, in general, has a complexity of  $\mathcal{O}(N^3)$  [53]. Note that  $\mathcal{O}(N^3)$  corresponds to the complexity of deterministic algorithms that compute the full SVD. However, in our context we are interested in a partial SVD that comprises the largest singular values. There exist modern randomised algorithms that are able to compute an approximated SVD for the  $K$  dominant components with complexity  $\mathcal{O}(MN \log K)$  for matrices of size  $M \times N$  [125, 126].

---

**Algorithm 6** *Fast Cadzow*—Denoising algorithm for circulant matrices

---

**Input:**  $N$  noisy samples  $\tilde{y}[n]$  and number of complex exponentials  $K$ .

**Output:** Denoised samples  $y'[n]$ .

- 1:  $\hat{y}[m] = \text{DFT}_N \{ \tilde{y}[n] \}$ .
  - 2: Set to zero  $N - K$  samples  $\hat{y}[m]$  that correspond to the smallest values of  $|\hat{y}[m]|$ .
  - 3:  $y'[n] = \text{IDFT}_N \{ \hat{y}[m] \}$
- 

### 6.3. Noisy sparse recovery based on Cadzow and Prony

The strategy to remove spikes that has been presented in the previous section is of particular interest to recover sparse vectors in the scenario where the dictionary is the union of Fourier and identity matrices. Note that when the spikes are removed, part of the energy, and therefore of the amplitudes, of the Fourier atoms is also removed. Therefore, instead of removing all the spikes at once and getting an estimate of the Fourier atoms, we can follow a different strategy where only one spike is removed at each iteration. The general idea of the algorithm is that if we compute the difference between the original observation  $\mathbf{y}$  and the Cadzow denoised vector  $\mathbf{y}'$ , this residual will mainly contain the spikes and the noise. We also assume that the power of the noise is small compared to the spikes. We estimate one spike at each iteration, remove the contribution of this spike, and iterate again until all the spikes have been removed. Once the spikes have been removed, the Fourier atoms are estimated from this *cleaned* vector  $\mathbf{y}'$  by applying Prony's method. Note that Prony's method finds *continuous* frequencies that might not correspond exactly to the frequencies of the atoms that form the vector  $\mathbf{y}$ . These frequencies are rounded to the nearest multiple of  $2\pi/M$  in order to be able to represent the observation with the dictionary  $\mathbf{D}$ . The algorithm to estimate the sparse vector based on this approach and that we have named ProSparse Denoise is described step by step in Algorithm 7.

This approach can be applied regardless of the fact that the Fourier atoms come from a Fourier basis ( $M = N$ ) or from a Fourier frame ( $M > N$ ). The advantage of the Fourier basis case, is that we can apply the *fast* version of the Cadzow algorithm that has been presented in Algorithm 6. Algorithm 7 provides an estimate for the entire vector  $\mathbf{x}$ , that is, it estimates both, the Fourier atoms and the spikes. However, in practice, we use this algorithm only to estimate the Fourier atoms. A more robust estimate of the spikes is

---

**Algorithm 7** *ProSparse Denoise*—Union of Fourier and identity matrices
 

---

**Input:** Noisy vector  $\mathbf{y} = [\mathbf{F}, \mathbf{I}] \mathbf{x} + \varepsilon$  and sparsity levels  $(K_p, K_q)$ .

**Output:**  $(K_p, K_q)$ -sparse vector  $\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_q \end{bmatrix}$ .

- 1: Initialise spikes  $\mathbf{x}_q = \mathbf{0}$ .
  - 2: Initialise spikes locations  $\Omega = \{0, 1, \dots, N - 1\}$ .
  - 3: Denoise  $\mathbf{y}' = \text{Cadzow}(\mathbf{y}, K_p)$ .
  - 4: **for**  $i = 1$  **to**  $K_q$  **do**
  - 5:   Compute residual  $\mathbf{r} = \mathbf{y} - \mathbf{y}'$ .
  - 6:   Estimate spike location from residual  $n_0 = \arg \max_{n \in \Omega} \{|r[n]|\}$ .
  - 7:   Store spike  $x_q[n_0] = r[n_0]$ .
  - 8:   Add spike location to support  $\Omega \leftarrow \Omega \setminus \{n_0\}$ .
  - 9:   Remove spikes to the observation  $\mathbf{y}' = \mathbf{y} - \mathbf{x}_q$ .
  - 10:   Denoise  $\mathbf{y}' \leftarrow \text{Cadzow}(\mathbf{y}', K_p)$ .
  - 11: **end for**
  - 12: Estimate Fourier atoms  $\mathbf{x}_p = \text{Prony}(\mathbf{y}', K_p)$ .
- 

obtained with the approach that is described next.

In the case of the Fourier basis, Algorithm 7 is applied together with the *circulant* Cadzow denoising approach described in Algorithm 6 to estimate the Fourier atoms from  $\mathbf{y}$ . The same approach is applied to  $\overline{\mathbf{F}} \mathbf{y}$  to estimate the spikes since, in  $\overline{\mathbf{F}} \mathbf{y}$ , the spikes correspond to complex exponentials. Note that both computations can be performed in parallel.

In the case of the Fourier frame, Algorithm 7 is first applied to  $\mathbf{y}$  with the traditional Cadzow approach. This provides an estimate of the Fourier atoms  $\mathbf{x}_p$ . Then, Algorithm 7 is applied again to the Fourier transform of the residual, which ideally only contains the spikes. That is, we compute  $\mathbf{r} = \mathbf{y} - \mathbf{F}_{N,M} \mathbf{x}_p$  to estimate the spikes from  $\overline{\mathbf{F}}_N \mathbf{r}$ . To avoid ambiguities we have added subscripts to the Fourier matrices. We denote by  $\mathbf{F}_N$  the square Fourier matrix of size  $N \times N$  and  $\mathbf{F}_{N,M}$  the rectangular Fourier frame of size  $N \times M$  (which is given by the first  $N$  rows of a square Fourier matrix of size  $M \times M$ ).

### 6.3.1. Complexity analysis of ProSparse Denoise

The key parts of the ProSparse Denoise algorithm rely on two algorithms of known complexity, Cadzow denoising and Prony's method. For the case where the Fourier atoms correspond to the Fourier basis, the complexity of the denoising step is  $\mathcal{O}(N \log N)$ . It follows that the complexity of the loop that iterates  $K_q$  times is  $\mathcal{O}(K_q N \log N)$ . The next step of the algorithm is to apply Prony's method to retrieve the  $K_p$  Fourier atoms, which has a complexity of  $\mathcal{O}(K_p^3)$  (see Section 3.3). In practice the algorithm is applied to  $\mathbf{y}$  to estimate the  $K_p$  Fourier atoms and to  $\overline{\mathbf{F}} \mathbf{y}$  to estimate the  $K_q$  spikes. Since we have that  $K = K_p + K_q$ , the overall complexity of the algorithm for the union of bases case is  $\mathcal{O}(K^3 + K N \log N)$ . For the case of the union of Fourier frame and identity matrix, the

Cadzow algorithm has a higher complexity since the fast version based on the FFT cannot be applied. Making use of the modern randomised algorithms that we have mentioned in the previous section that compute a partial SVD for the  $K$  dominant components with complexity  $\mathcal{O}(N^2 \log K)$ , the complexity of the algorithm becomes  $\mathcal{O}(K^3 + KN^2 \log N)$ . Note that the complexity of this algorithm does not depend on the dimension of the sparse vector  $\mathbf{x} \in \mathbb{C}^L$ . This is a great gain in performance if compared to the complexity of the algorithms described in the introduction of this chapter.

## 6.4. Guaranteed performance analysis

The signal model presented in (6.1) and (6.2) allows us to perform some probabilistic analysis of the performance of the algorithm from a worst case scenario point of view. In this section, first, we present an analysis of the SNR of the signal in terms of the amplitudes of the atoms and the power of the noise. Then, we establish a sufficient condition for the algorithm to succeed for the simple case where there is only one Fourier atom and one spike. This condition is then validated with numerical simulations. We restrict the analysis to the case where the Fourier atom is drawn from a Fourier basis.

### 6.4.1. Signal-to-noise ratio of the signal

The power of the signal is given by

$$P_y = \frac{1}{N} \sum_{n=0}^{N-1} |y[n]|^2 = \frac{1}{N} \sum_{n=0}^{N-1} y[n] y^*[n]. \quad (6.15)$$

If we denote by  $\omega_k = \frac{2\pi}{N} m_k$  the frequencies of the Fourier atoms, it follows that

$$|y[n]|^2 = \left( \frac{1}{\sqrt{N}} \sum_{k=1}^{K_p} a_k e^{i\omega_k n} + \sum_{k=1}^{K_q} b_k \delta[n - n_k] \right) \left( \frac{1}{\sqrt{N}} \sum_{\ell=1}^{K_p} a_\ell^* e^{-i\omega_\ell n} + \sum_{\ell=1}^{K_q} b_\ell^* \delta[n - n_\ell] \right). \quad (6.16)$$

If we expand this product we obtain

$$\begin{aligned} |y[n]|^2 &= \frac{1}{N} \sum_{k=1}^{K_p} \sum_{\ell=1}^{K_p} a_k a_\ell^* e^{i(\omega_k - \omega_\ell)n} + \frac{1}{\sqrt{N}} \sum_{k=1}^{K_p} \sum_{\ell=1}^{K_q} a_k b_\ell^* e^{i\omega_k n} \delta[n - n_\ell] \\ &\quad + \frac{1}{\sqrt{N}} \sum_{k=1}^{K_q} \sum_{\ell=1}^{K_p} b_k a_\ell^* e^{-i\omega_\ell n} \delta[n - n_k] + \sum_{k=1}^{K_q} \sum_{\ell=1}^{K_q} b_k b_\ell^* \delta[n - n_k] \delta[n - n_\ell]. \end{aligned} \quad (6.17)$$

Summing over  $n$  yields

$$P_y = \frac{1}{N} \left( \sum_{k=1}^{K_p} |a_k|^2 + \frac{1}{\sqrt{N}} \sum_{k=1}^{K_p} \sum_{\ell=1}^{K_q} a_k b_\ell^* e^{i\omega_k n_\ell} + \frac{1}{\sqrt{N}} \sum_{k=1}^{K_q} \sum_{\ell=1}^{K_p} a_k b_\ell^* e^{-i\omega_\ell n_k} + \sum_{k=1}^{K_q} |b_k|^2 \right). \quad (6.18)$$

Applying the fact that for  $z \in \mathbb{C}$  we have  $z z^* = 2 \Re \{z\}$ , we have that

$$P_y = \frac{1}{N} \left( \sum_{k=1}^{K_p} |a_k|^2 + \sum_{k=1}^{K_q} |b_k|^2 + \frac{2}{\sqrt{N}} \sum_{k=1}^{K_p} \sum_{\ell=1}^{K_q} \Re \{a_k b_\ell^* e^{i\omega_k n_\ell}\} \right). \quad (6.19)$$

In order to analyse the power of the noise we have to take into account that this term corresponds to a stochastic process. Therefore, a meaningful measure of the noise power is given by its expected value:

$$P_n = \mathbb{E} \left\{ \frac{1}{N} \sum_{n=0}^{N-1} |\varepsilon[n]|^2 \right\}. \quad (6.20)$$

Due to the independence of the noise samples, it follows that

$$P_n = \frac{1}{N} \sum_{n=0}^{N-1} \mathbb{E} \{ \varepsilon^R[n]^2 + \varepsilon^I[n]^2 \} = \frac{1}{N} \sum_{n=0}^{N-1} (\mathbb{E} \{ \varepsilon^R[n]^2 \} + \mathbb{E} \{ \varepsilon^I[n]^2 \}) = 2 \sigma_\varepsilon^2. \quad (6.21)$$

The resulting SNR is given by

$$SNR = \frac{P_y}{P_n} = \frac{\sum_{k=1}^{K_p} |a_k|^2 + \sum_{k=1}^{K_q} |b_k|^2 + \frac{2}{\sqrt{N}} \sum_{k=1}^{K_p} \sum_{\ell=1}^{K_q} \Re \{a_k b_\ell^* e^{i\omega_k n_\ell}\}}{2 N \sigma_\varepsilon^2}. \quad (6.22)$$

This expression is going to be useful to assess what is the actual SNR of the signal when we establish some conditions on  $N$  and  $\sigma_\varepsilon$ .

#### 6.4.2. Sufficient condition for sparse recovery with $K_p = K_q = 1$

ProSparse Denoise depends on estimating spikes and Fourier atoms from the maximum of the absolute value of the signal  $\mathbf{y}$  and its Fourier transform. For this reason, it is important to understand how the maximum of the absolute value of the noise behaves. Since our noise model is based on complex-valued i.i.d. Gaussian random variables, the absolute value of the noise term, that is  $|\varepsilon[n]| = |\varepsilon^R[n] + i \varepsilon^I[n]|$ , follows a Rayleigh distribution. In Appendix E it is shown that for large  $N$ , the maximum value of a collection of  $N$  random variables that follow a Rayleigh distribution with parameter  $\sigma_\varepsilon$  is equal to  $\sigma_\varepsilon \sqrt{2 \log N}$  with high probability. However, it has to be noted that this convergence is extremely slow [127]. Note that in the case of the Fourier basis, we consider unitary Fourier transforms, and therefore, the noise terms in the Fourier transform of  $\mathbf{y}$  follow the same distribution as in

the temporal domain.

The first step of the denoising algorithm operates in the Fourier domain and selects the Fourier atoms by picking the samples with the largest amplitudes. When there is one Fourier atom and one spike, the worst case scenario corresponds to the case where the noise and the Fourier transform of the spike sum destructively with the Fourier atom, the amplitude of the noise at this location is maximum, and there is another location where the noise also reaches the maximum amplitude of  $\sigma_\varepsilon \sqrt{2 \log N}$ .

Let us assume that the Fourier atom and the spike have the same amplitude:

$$a_1^R = a_1^I = b_1^R = b_1^I = 1/\sqrt{2}. \quad (6.23)$$

If the Fourier atom is located at  $m = m_1$ , the noiseless sample of the Fourier transform at  $m = m_1$  is given by

$$\hat{y}[m_1] = (a_1^R + i a_1^I) + \frac{1}{\sqrt{N}}(b_1^R + i b_1^I) e^{-i 2 \pi n_1 m_1 / N} \quad (6.24)$$

$$= e^{i \pi / 4} + \frac{1}{\sqrt{N}} e^{i(\pi / 4 - 2 \pi n_1 m_1 / N)}. \quad (6.25)$$

The Fourier atom and the spike sum destructively when the phase of the second complex exponential in the previous equation has a difference of  $\pi$  radians (modulo  $2\pi$ ) with the phase of the first complex exponential, that is:

$$\frac{\pi}{4} - \frac{2 \pi n_1 m_1}{N} = \frac{5 \pi}{4} + 2 \pi \ell \quad (6.26)$$

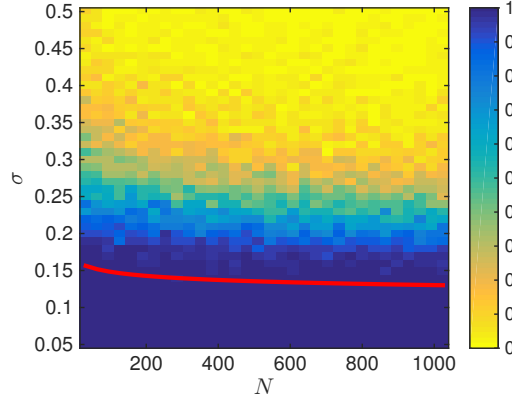
$$\Leftrightarrow -n_1 m_1 = \frac{N}{2} + \ell N \quad (6.27)$$

$$\Leftrightarrow \text{mod}(n_1 m_1, N) = \frac{N}{2}. \quad (6.28)$$

This establishes a condition on  $m_1$  and  $n_1$  to have a destructive interaction between the Fourier atom and the Fourier transform of the spike. In that case, the resulting amplitude is given by

$$|y[m_1]| = 1 - \frac{1}{\sqrt{N}}. \quad (6.29)$$

In the worst case scenario, we assume that the amplitude of the noise at location  $m = m_1$  is maximum, that is  $|\varepsilon[m_1]| = \sigma_\varepsilon \sqrt{2 \log N}$ , and sums destructively with the Fourier atom's amplitude. Moreover, we assume that another location  $m \neq m_1$  also presents a noise sample with this same amplitude. We can therefore establish the following sufficient success condition that guarantees that in the worst case scenario we will still be able to detect the



**Figure 6.2.:** Guaranteed performance of ProSparse Denoise with 1 Dirac and 1 Fourier atom for different levels of noise. 50 iterations per noise level. The success is measured by checking that the support is correctly retrieved at each iteration. The amplitudes of the atoms are fixed and are equal to  $1/\sqrt{2}$  for the real and imaginary parts. The location of the Fourier atom is  $m_1 = 1$ . The location of the Fourier atom is  $m_1 = 1$  and the location of the spike,  $n_1$ , satisfies (6.28). The red line depicts the equation  $\sigma = (1 - 1/\sqrt{N})(\sqrt{8 \log N})$  from (6.31).

Fourier atom:

$$1 - \frac{1}{\sqrt{N}} - \sigma_\varepsilon \sqrt{2 \log N} > \sigma_\varepsilon \sqrt{2 \log N} \quad (6.30)$$

$$\Leftrightarrow \quad \sigma_\varepsilon < \frac{1 - 1/\sqrt{N}}{\sqrt{8 \log N}}. \quad (6.31)$$

Equation (6.31) establishes an upper bound for the standard deviation of the noise in order to guarantee the success of the algorithm. For a given size of the problem  $N$ , if there are one Fourier atom and one spike, having a noise with a  $\sigma_\varepsilon$  which is below this upper bound guarantees, in theory, that the algorithm will succeed. Figure 6.2 illustrates an empirical validation of this argument. For different sizes of the problem, and different levels of noise, the sparse vector is reconstructed applying the algorithm described in the previous section. The experiment is repeated 50 times for each combination of  $N$  and  $\sigma$  by generating different realisations of the noise vector. We consider that the algorithm has succeeded if the correct support of the original vector  $\mathbf{y}$  is recovered. The red line depicts the bound established in Equation (6.31). We can observe that when the noise has a power below this bound the algorithm always succeeds, confirming the predicted result.

If we replace the value of  $\sigma_\varepsilon$  obtained in (6.31) in the expression of the SNR, with the constraint on  $m_1$  and  $n_1$  established in (6.28), it follows that

$$SNR > \frac{8 \log N}{N \left(1 - 1/\sqrt{N}\right)}. \quad (6.32)$$

## 6.5. Simulation results

ProSparse Denoise has been tested against the following state of the art algorithms: BPDN, OMP, LASSO and Subspace Pursuit. Figures 6.3 and 6.4 present the results where we have measured the MSE of the recovered sparse vector and the probability that the algorithms recover the exact support of the original sparse signal for different sizes of the problem. Specifically, for a given sparsity level and SNR, 100 different supports of the sparse vector  $\mathbf{x} \in \mathbb{C}^L$  are generated uniformly at random; and for each realisation of the support, 10 different realisations of the amplitudes of  $\mathbf{x}$  and of the noise vector  $\boldsymbol{\varepsilon}$  are generated such that the observation  $\mathbf{y} = \mathbf{D}\mathbf{x} + \boldsymbol{\varepsilon}$  satisfies the specified SNR. For each sparsity level we thus have 1000 realisations. All the amplitudes are Gaussian distributed and complex-valued. At each realisation, the five algorithms reconstruct a sparse vector from the noisy observation  $\mathbf{y}$ . Simulations have been performed for 8 different scenarios that correspond to SNR levels of 5 and 10 dB with a bias in the number of atoms from one dictionary with respect to the other of 25 and 50 % and sizes  $N = 128$  and  $N = 256$ . A bias of 25 % means that there are  $K/4$  spikes and  $3K/4$  Fourier atoms (the different values of  $K$  are chosen so that these numbers are always integers).

The greedy algorithms, OMP and Subspace Pursuit, and the novel algorithm presented in this chapter, ProSparse Denoise (denoted by PSDN in the plots), inherently take advantage of the fact that the sparsity level is known. The other two algorithms that have been tested, which are based on convex relaxation techniques (BPDN and LASSO), do not make use of the information of the sparsity level in their execution. However, this information has also been used in the reconstruction of BPDN and LASSO by applying a final debiasing phase. The debiasing phase consists of two steps. First, the support of the sparse vector is estimated from the largest entries of the solution that is estimated by each algorithm, this is where the information of the sparsity level is exploited. Then, the amplitudes of the locations that correspond to this support are computed with a least squares approach (all the other elements are set to zero). This approach improves considerably the MSE that is measured for these algorithms.

From the results, it is clear that the proposed sparse recovery algorithm consistently outperforms state of the art algorithms at all noise and sparsity levels. We can also see that, in this scenario, the greedy algorithms perform better than convex relaxation techniques. The differences in performance become negligible for large  $K$ , where the probability of retrieving the correct support drops for all the methods. Another algorithm was also tested before running the extensive simulations, GPSR. The performance of this algorithm is very sensible to a parameter that has to be tuned (see parameter  $\tau$  embedded in vector  $\mathbf{c}$  in Equation 6.6). Despite the efforts to tune it for this scenario, the obtained results were considerably worse than the other algorithms and therefore the algorithm was discarded for these simulations. The code of the algorithm was downloaded from the author's website

[120], and approaches that are mentioned in the paper such as "warm-start" were also tested (start with a larger value of  $\tau$  to obtain a first approximation of the solution and use this approximation as an initial estimate for a subsequent execution with a smaller  $\tau$ ).

Besides the gain in performance, it is also important to note that this novel algorithm is faster than the other algorithms in the majority of scenarios. Execution times have been measured during these simulations and are summarised in Tables 6.1 and 6.2. These measurements are obtained by averaging over the multiple realisations of each sparsity level. The experiments have been run using a commercial laptop (tested on a 2.5GHz Intel Core i5 CPU) and all the algorithms were implemented in MATLAB. We used the CVX package to implement the BPDN optimisation problem because it was giving the best performance compared to other optimisation toolboxes [112]. LASSO was tested using MATLAB's implementation. OMP has been implemented for the simulations and the implementation of Subspace Pursuit downloaded from the authors' website (<http://www.ee.imperial.ac.uk/wei.dai/Software/SubspacePursuit.zip>). Note that only Subspace Pursuit runs faster than ProSparse Denoise and only for very low sparsity levels. For sparsity levels that go beyond 12 for  $N = 128$  and 16 for  $N = 256$  ProSparse Denoise is the fastest of all the algorithms.

**Table 6.1.:** Average execution time of sparse recovery algorithms,  $N = 128$ .

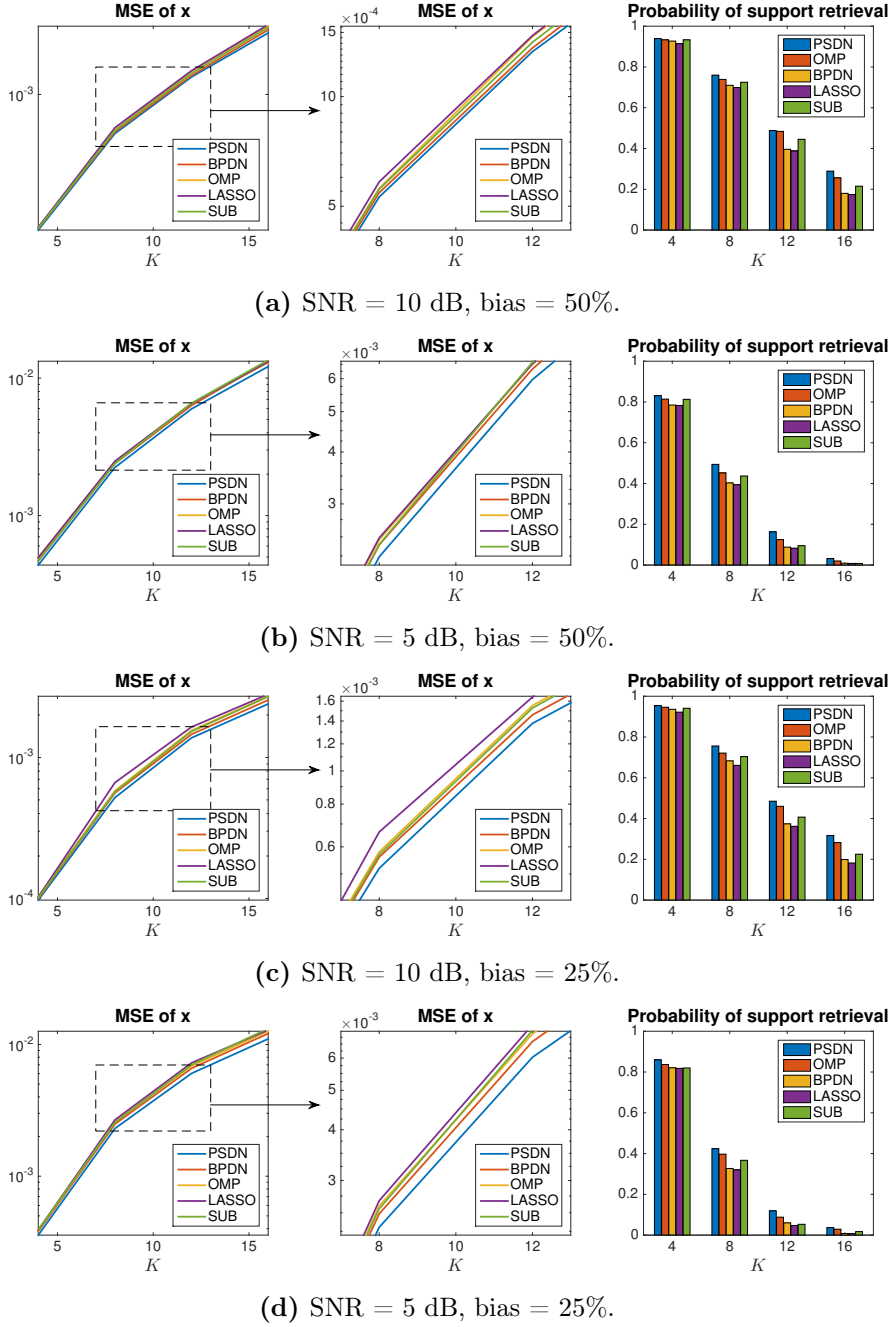
| K  | ProSparse Denoise | BPDN   | OMP    | LASSO  | Subspace Pursuit |
|----|-------------------|--------|--------|--------|------------------|
| 4  | 0.0030            | 3.1480 | 0.0308 | 0.0566 | <b>0.0024</b>    |
| 8  | 0.0037            | 2.9903 | 0.0575 | 0.0763 | <b>0.0035</b>    |
| 12 | <b>0.0054</b>     | 3.1420 | 0.0916 | 0.1007 | 0.0059           |
| 16 | <b>0.0059</b>     | 2.9097 | 0.1187 | 0.1184 | 0.0081           |

**Table 6.2.:** Average execution time of sparse recovery algorithms,  $N = 256$ .

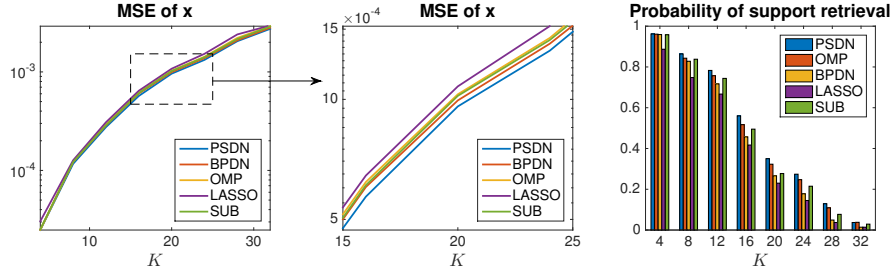
| K  | ProSparse Denoise | BPDN    | OMP    | LASSO  | Subspace Pursuit |
|----|-------------------|---------|--------|--------|------------------|
| 4  | 0.0098            | 33.0813 | 0.2442 | 0.1503 | <b>0.0055</b>    |
| 8  | 0.0112            | 29.3087 | 0.4779 | 0.1565 | <b>0.0087</b>    |
| 12 | 0.0136            | 27.8360 | 0.7159 | 0.1722 | <b>0.0110</b>    |
| 16 | <b>0.0151</b>     | 27.2135 | 0.9560 | 0.1921 | 0.0165           |
| 20 | <b>0.0171</b>     | 26.7477 | 1.1906 | 0.2038 | 0.0195           |
| 24 | <b>0.0202</b>     | 26.7861 | 1.4324 | 0.2183 | 0.0230           |
| 28 | <b>0.0216</b>     | 25.8907 | 1.6440 | 0.2318 | 0.0251           |
| 32 | <b>0.0203</b>     | 22.9313 | 1.6568 | 0.2218 | 0.0250           |

## 6.6. Summary

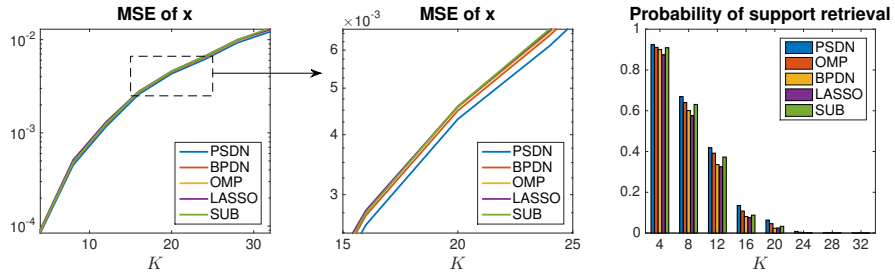
A novel algorithm has been presented to solve the sparse recovery problem in the noisy scenario. This new approach is based on an extension of the Cadzow denoising algorithm for the finite-dimensional case. This extension is combined with an iterative spike removal algorithm to obtain a *cleaned* signal that only contains Fourier atoms. These atoms are then estimated using Prony's method. The overall algorithm is able to solve the sparsity problem faster, and with higher precision, than state of the art algorithms. We note that traditional compressed sensing methods are more flexible, in the sense that they can solve the sparsity problem for generic dictionaries. However, our method outperforms these algorithms because it fully exploits the particular structure of the dictionary at hand: the union of Fourier and identity matrices.



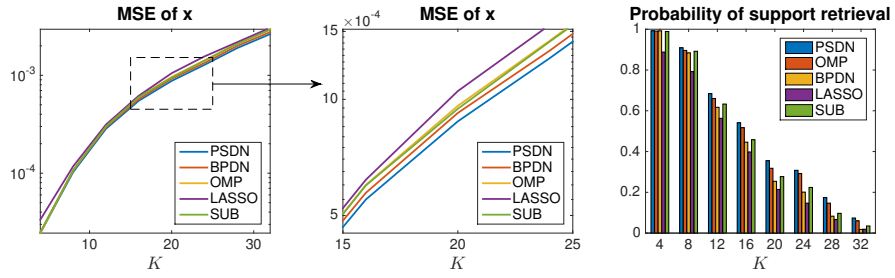
**Figure 6.3.:** Union of Fourier and identity bases, each of size  $N \times N$  with  $N = 128$ . Simulation results with  $K_q = \text{bias} \cdot K$  spikes and  $K_p = K - K_q$  Fourier atoms. 1000 realisations per sparsity level (100 non-zero locations uniformly drawn at random and for each realisation of the non-zero locations 10 realisations of the amplitudes drawn from  $\mathcal{N}(0, 1)$  for the real and imaginary parts). The results of the novel algorithm presented in this chapter correspond to PSDN (ProSparse Denoise).



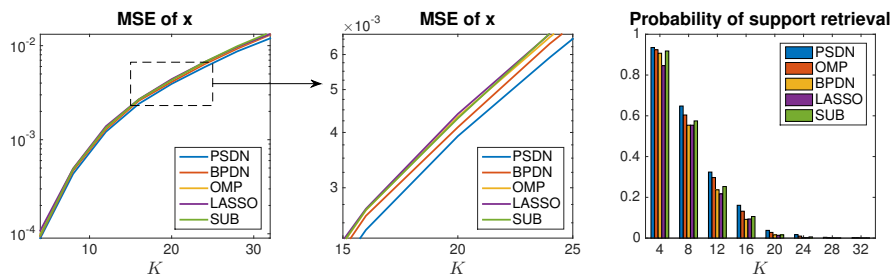
(a) SNR = 10 dB, bias = 50%.



(b) SNR = 5 dB, bias = 50%.



(c) SNR = 10 dB, bias = 25%.



(d) SNR = 5 dB, bias = 25%.

**Figure 6.4.:** Union of Fourier and identity bases, each of size  $N \times N$  with  $N = 256$ . Simulation results with  $K_q = \text{bias} \cdot K$  spikes and  $K_p = K - K_q$  Fourier atoms. 1000 realisations per sparsity level (100 non-zero locations uniformly drawn at random and for each realisation of the non-zero locations 10 realisations of the amplitudes drawn from  $\mathcal{N}(0,1)$  for the real and imaginary parts). The results of the novel algorithm presented in this chapter correspond to PSDN (ProSparse Denoise).



# Chapter 7.

## Conclusions

### 7.1. Summary

In this thesis we have studied sampling and reconstruction methods for signals that present a finite number of degrees of freedom. These signals are known as signals with finite rate of innovation (FRI) and the framework that is being developed around this idea during the last decade has revolutionised the way we acquire and reconstruct some classes of signals. From the classical sampling theory point of view, band unlimited signals cannot be sampled and perfectly recovered. However, FRI theory presents a new sampling scheme that achieves perfect reconstruction for some classes of band unlimited signals. These ideas can also be applied to recover finite-dimensional signals. First, we have analysed the sampling process of streaming continuous-time signals. Next, we have presented an extension of this procedure with an application in neuroscience to monitor the activity of individual neurones. Last, we have examined the case where the unknown signal is finite-dimensional.

In Chapter 2, we have revisited the classical theory for sampling bandlimited signals in order to introduce the FRI theory. FRI theory presents a more general characterisation of signals, and therefore, bandlimited signals can be seen as a subset of FRI signals. We have seen how perfect reconstruction of some band unlimited signals can be achieved. Specifically, we have presented the case where the input is a stream of Diracs and the sampling kernel satisfies the exponential reproducing property. The core of the reconstruction algorithms is based on Prony's method. We have also described some state of the art algorithms to make the reconstruction procedure more robust when samples are contaminated with noise.

The canonical example of FRI signals are streams of Diracs. A Dirac delta is an idealisation of an impulsive signal where we are only interested in its amplitude and location. A set of  $K$  Diracs is thus perfectly recovered if we are able to retrieve the  $2K$  parameters that correspond to their amplitudes and locations. FRI literature focuses on sampling and reconstructing bursts of Diracs, but has not analysed streaming signals where no clear separation between consecutive impulses can be detected. In Chapter 3, we have presented a novel algorithm that achieves perfect reconstruction for this type of streaming signals.

First, we have established some conditions on the support of the sampling kernel and the sampling period in order to achieve perfect reconstruction of noiseless signals. We have then presented a robust algorithm to reconstruct streaming signals in the presence of noise. This algorithm is based on sampling and reconstructing sequentially sets of Diracs with a sliding window. Since each Dirac is captured by several positions of the sliding window, we can construct a histogram of the retrieved locations. The peaks of this histogram correspond to real Diracs.

Neuroscientists are trying to identify neural circuits in regions of the brain. In order to understand the connectivity of individual neurones it is important to infer their activation times or action potentials. Calcium fluorescence imaging techniques allow the monitoring of populations of neurones. Action potentials are well characterised by decaying exponentials in this type of data. In Chapter 4, we have presented an extension of the FRI theory to sample and reconstruct streams of decaying exponentials. We have combined this extension with the ideas presented in the previous chapter to infer the times of action potentials from fluorescence sequences. The algorithm is able to monitor tens of neurones in real time with a commercial computer and outperforms state of the art algorithms.

The parametrisation of FRI signals is based on defining a continuous-time signal in terms of the location and amplitude of some known pulses. This idea can also be applied to the finite-dimensional case to solve the sparse reconstruction problem. In Chapter 5, we have presented a probabilistic analysis of the performance of an algorithm that reconstructs finite-dimensional signals based on Prony's method, *ProSparse*. These signals have a sparse representation in a dictionary that is given by sinusoidal atoms and spikes. The probabilistic analysis shows a phase transition behaviour of the algorithm, that is, the algorithm succeeds with high probability when some conditions are satisfied, and fails with high probability when they are not.

In the last technical chapter, Chapter 6, we have proposed an iterative algorithm to solve the sparse representation problem when the observed signal is corrupted with noise. We have also considered the context where the observed signal is given by a sum of sinusoidal elements and spikes. This novel approach is based on applying a denoising algorithm that reduces noise in signals that are given by sums of exponentials. This denoising method keeps the sinusoidal part of the observed signal but also removes the spikes. We make use of this property to iteratively remove the spikes and then estimate the Fourier atoms from the cleaned signal. The spikes can then be estimated from the residual, or if the sinusoidal atoms are drawn from a Fourier basis, we can also apply the same approach to the Fourier transform of the observed signal. In that case, the spikes become sinusoidals and thus can be estimated with the same approach. This method outperforms state of the art algorithms for sparse recovery such as Basis Pursuit Denoise or Subspace Pursuit.

## 7.2. Future research

To conclude the thesis we propose some future research topics.

- An interesting extension of the FRI theory was proposed in the year 2013 by Urigüen et al. [27] where the strict conditions on the sampling kernels were relaxed, and still, very good performances were achieved. This idea can be applied to the problem we have studied in Chapter 4 to monitor neural activity. This application would require to estimate the transfer function of the entire acquisition device, but has the potential of improving the accuracy and performance of the inference algorithm.
- The sparse recovery problem has been analysed for dictionaries that correspond to the union of Fourier and identity matrices. This problem has very practical applications in image processing if the algorithms are extended to other pairs of bases such as wavelet and DCT basis. The joint use of these two bases can lead to better approximation results when used to approximate images, where wavelet atoms can represent very accurately edges and discontinuities and DCT atoms can represent regular patterns in textures.
- In Chapter 6 we have presented a limited guaranteed performance analysis for the case where the observed signal is made of one Fourier atom and one Dirac. Analysing from a probabilistic point of view the real performance of the algorithm for general signals with more atoms would lead to a better understanding of its behaviour.



## Appendix A.

### Function reproduction with splines

#### A.1. Polynomial reproducing kernels and $c_{m,n}$ coefficients

In this section we provide an efficient way of computing the  $c_{m,n}$  involved in the reproduction of polynomial functions. A polynomial reproducing kernel satisfies:

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t - n) = t^m, \quad m = 0, 1, \dots, P. \quad (\text{A.1})$$

If the set  $\{\varphi(t - n)\}_{n \in \mathbb{Z}}$  does not form an orthogonal basis, the  $c_{m,n}$  coefficients can be obtained from the quasi-biorthogonal set  $\{\tilde{\varphi}(t - n)\}_{n \in \mathbb{Z}}$  as  $c_{m,n} = \langle t^m, \tilde{\varphi}(t - n) \rangle$  ([128]). This includes the particular case where  $\tilde{\varphi}(t)$  is the dual of  $\varphi(t)$ , that is, they satisfy  $\langle \varphi(t), \tilde{\varphi}(t - n) \rangle = \delta[n]$ . The dual is not always easy to obtain, thus, we can try to compute directly the  $c_{m,n}$  coefficients without having knowledge of  $\tilde{\varphi}(t)$ .

$$\begin{aligned} c_{m,n} &= \int_{-\infty}^{\infty} t^m \tilde{\varphi}(t - n) dt \\ &= \int_{-\infty}^{\infty} (t + n)^m \tilde{\varphi}(t) dt \\ &= \sum_{k=0}^m \binom{m}{k} n^{m-k} \underbrace{\int_{-\infty}^{\infty} t^k \tilde{\varphi}(t) dt}_{=c_{k,0}} \\ &= \sum_{k=0}^m \binom{m}{k} n^{m-k} c_{k,0}. \end{aligned} \quad (\text{A.2})$$

This expression provides an easy way to compute the  $c_{m,n}$  coefficients for any  $n$  from the  $c_{k,0}$  coefficients. We still need to compute the  $c_{k,0}$  coefficients for  $k = 0, 1, \dots, m$ . If we

plug this expression for  $c_{m,n}$  in the polynomial reproducing formula we obtain:

$$\begin{aligned}
 t^m &= \sum_{n \in \mathbb{Z}} \left( \sum_{k=0}^m \binom{m}{k} n^{m-k} c_{k,0} \right) \varphi(t-n) \\
 &= \sum_{n \in \mathbb{Z}} \left( c_{m,0} + \sum_{k=0}^{m-1} \binom{m}{k} n^{m-k} c_{k,0} \right) \varphi(t-n) \\
 &= c_{m,0} \sum_{n \in \mathbb{Z}} \varphi(t-n) + \sum_{k=0}^{m-1} \binom{m}{k} c_{k,0} \sum_{n \in \mathbb{Z}} n^{m-k} \varphi(t-n),
 \end{aligned} \tag{A.3}$$

where we can isolate  $c_{m,0}$  to obtain a recursive formula to compute these coefficients for any  $m$  up to  $m = P$ :

$$c_{m,0} = \frac{t^m - \sum_{k=0}^{m-1} \binom{m}{k} c_{k,0} \sum_{n \in \mathbb{Z}} n^{m-k} \varphi(t-n)}{\sum_{n \in \mathbb{Z}} \varphi(t-n)}. \tag{A.4}$$

This formula is valid for any  $t$ . We thus have the freedom to choose, for instance  $t = 0$ , which makes computations more stable. Note that we have assumed that  $\varphi(t)$  only reproduces polynomials up to degree  $m = P$  and therefore this formula is only valid for  $0 \leq m \leq P$ . The first coefficients are given by:

$$c_{0,0} = \frac{1}{\sum_{n \in \mathbb{Z}} \varphi(t-n)}, \tag{A.5}$$

$$c_{1,0} = \frac{t - c_{0,0} \sum_{n \in \mathbb{Z}} n \varphi(t-n)}{\sum_{n \in \mathbb{Z}} \varphi(t-n)}, \tag{A.6}$$

$$c_{2,0} = \frac{t^2 - c_{0,0} \sum_{n \in \mathbb{Z}} n^2 \varphi(t-n) - 2 c_{1,0} \sum_{n \in \mathbb{Z}} n \varphi(t-n)}{\sum_{n \in \mathbb{Z}} \varphi(t-n)}. \tag{A.7}$$

## A.2. Exponential reproducing kernels and $c_{m,n}$ coefficients

In the previous section we have explained how to compute the  $c_{m,n}$  coefficients involved in the reproduction of polynomials. In this section we are interested in the reproduction of exponential functions and provide a similar explanation about how to compute the  $c_{m,n}$  coefficients to achieve this goal. A kernel that reproduces a set of exponential functions  $\{e^{\alpha_m t}\}_{m=0}^P$  satisfies:

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t-n) = e^{\alpha_m t}, \quad m = 0, 1, \dots, P. \tag{A.8}$$

If the set  $\{\varphi(t-n)\}_{n \in \mathbb{Z}}$  does not form an orthogonal basis, the  $c_{m,n}$  coefficients can be obtained from the quasi-biorthogonal set  $\{\tilde{\varphi}(t-n)\}_{n \in \mathbb{Z}}$  as  $c_{m,n} = \langle e^{\alpha_m t}, \tilde{\varphi}(t-n) \rangle$ .

This includes the particular case where  $\tilde{\varphi}(t)$  is the dual of  $\varphi(t)$ , that is, they satisfy  $\langle \varphi(t), \tilde{\varphi}(t-n) \rangle = \delta[n]$ . In this case, we have that

$$\begin{aligned}
 c_{m,n} &= \int_{-\infty}^{\infty} e^{\alpha_m t} \tilde{\varphi}(t-n) dt \\
 &= \int_{-\infty}^{\infty} e^{\alpha_m(t+n)} \tilde{\varphi}(t) dt \\
 &= e^{\alpha_m n} \underbrace{\int_{-\infty}^{\infty} e^{\alpha_m t} \tilde{\varphi}(t) dt}_{c_{m,0}} \\
 &= e^{\alpha_m n} c_{m,0}.
 \end{aligned} \tag{A.9}$$

We can thus compute the coefficients  $c_{m,n}$  for any  $n$  from the sole knowledge of the coefficients  $c_{m,0}$ . In order to compute the set of  $c_{m,0}$  coefficients for  $m = 0, 1, \dots, P$ , we can insert this expression of  $c_{m,n}$  in the exponential reproduction formula which yields

$$c_{m,0} = \frac{e^{\alpha_m t}}{\sum_{n \in \mathbb{Z}} e^{\alpha_m n} \varphi(t-n)} = \left[ \sum_{n \in \mathbb{Z}} e^{-\alpha_m(t-n)} \varphi(t-n) \right]^{-1}. \tag{A.10}$$

Assuming that the parameters  $\alpha_m = i\omega_m$  are purely imaginary and applying the Poisson summation formula it follows that

$$c_{m,0} = \left[ \sum_{k \in \mathbb{Z}} \hat{\varphi}(2\pi k + \omega_m) e^{i2\pi k t} \right]^{-1}. \tag{A.11}$$

From the generalised Strang-Fix conditions (see Proposition 2) we know that if  $\varphi(t)$  is an exponential reproducing kernel that reproduces the functions  $e^{i\omega_m t}$  then  $\hat{\varphi}(\omega_m + 2\pi k) = 0$  for  $k = \mathbb{Z} \setminus \{0\}$ . Therefore, we obtain the following compact expressions for the  $c_{m,0}$  and  $c_{m,n}$  coefficients:

$$c_{m,0} = [\hat{\varphi}(\omega_m)]^{-1} \quad \text{and} \quad c_{m,n} = e^{i\omega_m n} [\hat{\varphi}(\omega_m)]^{-1}. \tag{A.12}$$



## Appendix B.

# Toeplitz matrices with elements given by sums of exponentials

### B.1. Rank deficiency of Toeplitz matrix

Let  $\mathbf{S}$  be the following  $(P - M + 1) \times (M + 1)$  Toeplitz matrix:

$$\mathbf{S} = \begin{bmatrix} s[M] & s[M-1] & \dots & s[0] \\ s[M+1] & s[M] & \dots & s[1] \\ \vdots & \vdots & \ddots & \vdots \\ s[P] & s[P-1] & \dots & s[P-M] \end{bmatrix}, \quad (\text{B.1})$$

where the number of rows and columns are greater than or equal to some nonnegative integer  $K$ , that is,  $(P - M + 1) \geq K$  and  $(M + 1) \geq K$ . We consider the case where each element  $s[m]$  is given by a sum of  $K$  exponentials:

$$s[m] = \sum_{k=1}^K b_k u_k^m, \quad (\text{B.2})$$

with all  $b_k$  non-zero and all  $u_k$  distinct. The matrix  $\mathbf{S}$  can be decomposed as follows:

$$\mathbf{S} = \underbrace{\begin{bmatrix} 1 & \dots & 1 \\ u_1 & \dots & u_K \\ \vdots & \ddots & \vdots \\ u_1^{P-M} & \dots & u_K^{P-M} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} b_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & b_K \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_1^M & u_1^{M-1} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ u_K^M & u_K^{M-1} & \dots & 1 \end{bmatrix}}_{\mathbf{C}}. \quad (\text{B.3})$$

Since  $\mathbf{B}$  and  $\mathbf{C}$  are Vandermonde matrices with distinct elements, both are of rank  $K$ .  $\mathbf{A}$  is a  $K \times K$  diagonal matrix where the entries of the main diagonal correspond to the amplitudes  $b_1, b_2, \dots, b_K$ . Therefore, if elements  $b_1, b_2, \dots, b_K$  are all non-zero, matrix  $\mathbf{S}$  has rank  $K$ .

Note that this property is also valid when matrix  $\mathbf{S}$  is built from any  $P + 1$  consecutive

samples  $s[m]$  given as in B.2 and starting from an arbitrary index  $\ell \neq 0$ , that is, we build a Toeplitz  $\mathbf{S}$  from  $s[\ell], s[\ell + 1], \dots, s[\ell + P]$ . In this case, the decomposition in (B.3) is still valid by just adding a premultiplication by a diagonal matrix with the elements of the main diagonal given by  $u_1^\ell, u_2^\ell, \dots, u_K^\ell$ , which is also of rank  $K$ .

## B.2. Generalised eigenvalues and rank reducing numbers

Let  $\mathbf{S}$  be a Toeplitz matrix built as in (B.1) where the elements are given as in (B.2). We further impose the number of rows to be at least equal to  $K + 1$  and the number of columns to be equal to  $K$ . Let  $\mathbf{S}_0$  be the matrix constructed from  $\mathbf{S}$  by dropping the first row and  $\mathbf{S}_1$  the matrix constructed from  $\mathbf{S}$  by dropping the last row. From the decomposition presented in (B.3) we have that  $\mathbf{S}_0$  and  $\mathbf{S}_1$  have full column rank  $K$ . We now show that  $\{u_k\}_{k=1}^K$  are the generalised eigenvalues of the following generalised eigenvalue problem:

$$\mathbf{S}_0 \mathbf{v} = \mu \mathbf{S}_1 \mathbf{v} \quad \Leftrightarrow \quad (\mathbf{S}_0 - \mu \mathbf{S}_1) \mathbf{v} = \mathbf{0}. \quad (\text{B.4})$$

We have that (B.4) has a nontrivial solution  $\mathbf{v} \in \mathbb{C}^K$  only if  $\mathbf{S}_0 - \mu \mathbf{S}_1$  has rank strictly smaller than  $K$ , that is, we want to find the values of  $\mu$  that reduce the rank of  $\mathbf{S}_0 - \mu \mathbf{S}_1$ . Thus the name of rank reducing numbers. If we decompose matrix  $\mathbf{S}$  as in (B.3), that is,  $\mathbf{S} = \mathbf{B} \mathbf{A} \mathbf{C}$ , we can also express  $\mathbf{S}_0$  and  $\mathbf{S}_1$  as

$$\mathbf{S}_0 = \mathbf{B}_0 \mathbf{A} \mathbf{C} \quad \text{and} \quad \mathbf{S}_1 = \mathbf{B}_1 \mathbf{A} \mathbf{C}, \quad (\text{B.5})$$

where  $\mathbf{B}_0$  and  $\mathbf{B}_1$  are built from  $\mathbf{B}$  by removing the first and last rows respectively. We have that

$$\mathbf{S}_0 - \mu \mathbf{S}_1 = (\mathbf{B}_0 - \mu \mathbf{B}_1) \mathbf{A} \mathbf{C}. \quad (\text{B.6})$$

It is easy to verify that  $\mu = u_k$ ,  $k = 1, \dots, K$ , are rank reducing numbers since setting  $\mu = u_k$  introduces a column of zeros at the  $k$ th column of  $\mathbf{B}_0 - \mu \mathbf{B}_1$ . Hence, the parameters  $u_k$  correspond to the generalised eigenvalues of (B.4).

## Appendix C.

# Cramér-Rao lower bound and parameter estimation in the FRI context

### C.1. Parameter estimation and Cramér-Rao lower bound

We refer the reader to the Stoica's book [43, Appendix B] for a comprehensive analysis of the Cramér-Rao bound tools in the context of spectral estimation. Here, we present a brief introduction to these statistical tools in order to apply them in the context of FRI.

Let  $f(n, \boldsymbol{\theta})$  be a real-valued parametric model that depends on a real-valued vector  $\boldsymbol{\theta} = [\theta_1, \dots, \theta_K]^T \in \mathbb{R}^K$ . Vector  $\boldsymbol{\theta}$  represents the parameters to be estimated and  $n$  accounts for the temporal variable. Let  $\hat{\boldsymbol{\theta}}$  denote the estimated parameters from  $N$  samples. If the estimate satisfies  $\mathbb{E}\{\hat{\boldsymbol{\theta}}\} \rightarrow \boldsymbol{\theta}$  as  $N \rightarrow \infty$ , we say that the estimator is unbiased. For small errors we can approximate the estimation error  $f(n, \hat{\boldsymbol{\theta}}) - f(n, \boldsymbol{\theta})$  by

$$f(n, \hat{\boldsymbol{\theta}}) - f(n, \boldsymbol{\theta}) \simeq (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T \nabla f(n, \boldsymbol{\theta}), \quad (\text{C.1})$$

where  $\nabla f(n, \boldsymbol{\theta}) \in \mathbb{R}^K$  is the gradient of  $f(n, \boldsymbol{\theta})$  for a fixed  $n$ , that is, each element of  $\nabla f(n, \boldsymbol{\theta})$  is given by the partial derivative of  $f(n, \boldsymbol{\theta})$  with respect to each element in vector  $\boldsymbol{\theta}$ . It follows that the mean squared error (MSE) of  $f(n, \hat{\boldsymbol{\theta}})$  is approximately given by

$$\begin{aligned} \text{MSE}\{f(n, \hat{\boldsymbol{\theta}})\} &= \mathbb{E}\left\{[f(n, \hat{\boldsymbol{\theta}}) - f(n, \boldsymbol{\theta})]^T [f(n, \hat{\boldsymbol{\theta}}) - f(n, \boldsymbol{\theta})]\right\} \\ &\simeq \mathbb{E}\left\{\nabla f(n, \boldsymbol{\theta})^T (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T \nabla f(n, \boldsymbol{\theta})\right\} \\ &= \nabla f(n, \boldsymbol{\theta})^T \underbrace{\mathbb{E}\{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\}}_{\mathbb{Cov}\{\hat{\boldsymbol{\theta}}\}} \nabla f(n, \boldsymbol{\theta}). \end{aligned} \quad (\text{C.2})$$

For an unbiased estimator, that is  $\mathbb{E}\{\hat{\boldsymbol{\theta}}\} \rightarrow \boldsymbol{\theta}$  as  $N \rightarrow \infty$ , let  $\mathbf{P}$  denote the covariance matrix of  $\hat{\boldsymbol{\theta}}$ :

$$\mathbf{P} \stackrel{\text{def}}{=} \mathbb{Cov}\{\hat{\boldsymbol{\theta}}\} = \mathbb{E}\left\{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\right\}. \quad (\text{C.3})$$

The Cramér-Rao lower bound (CRB) [129, 130] states that there is a matrix such that

$\mathbf{P}$  is lower bounded, that is,  $\mathbf{P} \geq \mathbf{P}_{cr}$  in the sense that the difference matrix  $\mathbf{P} - \mathbf{P}_{cr}$  is positive semidefinite. The matrix  $\mathbf{P}_{cr}$  is given by

$$\mathbf{P}_{cr} = [\mathbf{I}(\boldsymbol{\theta})]^{-1} \quad (\text{C.4})$$

where  $\mathbf{I}(\boldsymbol{\theta})$  is the Fisher information matrix which is defined next. The diagonal terms in the matrix  $\mathbf{P}$  correspond to the uncertainty of our estimate and therefore the CRB provides the best achievable performance of an unbiased estimator.

### C.1.1. Likelihood function and Fisher information matrix

Let  $\mathbf{y} = (\tilde{y}[n])_{n=0}^{N-1} \in \mathbb{R}^N$  denote the vector of available noisy measurements. Each available sample has a deterministic part which corresponds to the parametric model  $f(n, \boldsymbol{\theta})$  and a stochastic part that accounts for the noise and uncertainties introduced by the acquisition device. We thus have

$$\tilde{y}[n] = f(n, \boldsymbol{\theta}) + \varepsilon[n], \quad (\text{C.5})$$

where  $\varepsilon[n]$ ,  $n = 0, 1, \dots, N-1$ , are random variables. If the parameters and measurements are complex-valued,  $\mathbf{y}$  and  $\boldsymbol{\theta}$  are obtained by concatenating the real and imaginary parts of the complex data and parameters.

We assume that the probability density function (PDF) of the random variables  $\varepsilon[n]$  is known, we can thus also obtain the pdf of  $\mathbf{y}$  which we denote  $p_{\mathbf{y}}(\mathbf{x}; \boldsymbol{\theta})$ . It follows that, if  $\mathcal{Y}$  is a subset of the possible values that the vector  $\mathbf{y}$  can take, the probability that the observation  $\mathbf{y}$  is in  $\mathcal{Y}$  is given by

$$\mathbb{P}\{\mathbf{y} \in \mathcal{Y}\} = \int_{\mathcal{Y}} p_{\mathbf{y}}(\mathbf{x}; \boldsymbol{\theta}) d\mathbf{x}. \quad (\text{C.6})$$

The function  $p_{\mathbf{y}}(\mathbf{x}; \boldsymbol{\theta})$  is called the *likelihood function* which leads to an important parameter estimation method called *maximum likelihood* (ML) estimator. The ML estimate of  $\boldsymbol{\theta}$  is obtained by maximizing the likelihood function as a function of  $\boldsymbol{\theta}$  for a given observation  $\mathbf{x} = \mathbf{y}$ :

$$\hat{\boldsymbol{\theta}}_{ML} = \arg \max_{\boldsymbol{\theta}} p_{\mathbf{y}}(\mathbf{y}; \boldsymbol{\theta}). \quad (\text{C.7})$$

It is often more practical to maximize the log-likelihood function:

$$\ell(\mathbf{y}; \boldsymbol{\theta}) \stackrel{\text{def}}{=} \log p_{\mathbf{y}}(\mathbf{y}; \boldsymbol{\theta}). \quad (\text{C.8})$$

The Fisher information matrix is defined in terms of the log-likelihood function:

$$\mathbf{I}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \mathbb{E} \{ \nabla \ell(\mathbf{y}; \boldsymbol{\theta}) (\nabla \ell(\mathbf{y}; \boldsymbol{\theta}))^T \}. \quad (\text{C.9})$$

### Gaussian and i.i.d. noise

Let us assume that the noise terms  $\varepsilon[n]$ ,  $n = 0, 1, \dots, N-1$ , are i.i.d. and normally distributed, that is,  $\varepsilon[n] \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ . The pdf of each observation  $\tilde{y}[n]$  is thus given by

$$p_{\tilde{y}[n]}(x; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} \exp\left(-\frac{(x - f(n, \boldsymbol{\theta}))^2}{2\sigma_\varepsilon^2}\right). \quad (\text{C.10})$$

Since the samples are independent, the pdf of  $\mathbf{y}$  is given by the product of each pdf:

$$p_{\mathbf{y}}(\mathbf{x}; \boldsymbol{\theta}) = \prod_{n=0}^{N-1} p_{\tilde{y}[n]}(x_n) = \left(\sqrt{2\pi\sigma_\varepsilon^2}\right)^{-N} \exp\left(-\frac{1}{2\sigma_\varepsilon^2} \sum_{n=0}^{N-1} (x_n - f(n, \boldsymbol{\theta}))^2\right), \quad (\text{C.11})$$

where  $\mathbf{x} = [x_0, \dots, x_{N-1}]^T \in \mathbb{R}^N$ . The log-likelihood function is therefore given by

$$\ell(\mathbf{y}; \boldsymbol{\theta}) = -N \log\left(\sqrt{2\pi\sigma_\varepsilon^2}\right) - \frac{1}{2\sigma_\varepsilon^2} \sum_{n=0}^{N-1} (\tilde{y}[n] - f(n, \boldsymbol{\theta}))^2, \quad (\text{C.12})$$

where  $\mathbf{x} = \mathbf{y}$ . The derivative of  $\ell(\mathbf{y}; \boldsymbol{\theta})$  with respect to each parameter  $\theta_k$  is given by

$$\frac{\partial \ell(\mathbf{y}; \boldsymbol{\theta})}{\partial \theta_k} = \frac{1}{\sigma_\varepsilon^2} \sum_{n=0}^{N-1} (y[n] - f(n, \boldsymbol{\theta})) \frac{\partial f(n; \boldsymbol{\theta})}{\partial \theta_k}. \quad (\text{C.13})$$

Note that the term  $\tilde{y}[n] - f(n, \boldsymbol{\theta})$  is equal to  $\varepsilon[n]$ . The gradient of  $\ell(\mathbf{y}, \boldsymbol{\theta})$  is thus given by

$$\nabla \ell(\mathbf{y}, \boldsymbol{\theta}) = \frac{1}{\sigma_\varepsilon^2} \sum_{n=0}^{N-1} \varepsilon[n] \nabla f(n; \boldsymbol{\theta}). \quad (\text{C.14})$$

We can now compute the Fisher information matrix:

$$\begin{aligned} \mathbf{I}(\boldsymbol{\theta}) &= \mathbb{E} \left\{ \frac{1}{\sigma_\varepsilon^4} \sum_{n=0}^{N-1} \varepsilon[n] \nabla f(n, \boldsymbol{\theta}) \sum_{m=0}^{N-1} \varepsilon[m] \nabla f(m, \boldsymbol{\theta})^T \right\} \\ &= \frac{1}{\sigma_\varepsilon^4} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \underbrace{\mathbb{E} \{ \varepsilon[n] \varepsilon[m] \}}_{=\sigma_\varepsilon^2 \delta[n-m]} \nabla f(n, \boldsymbol{\theta}) \nabla f(m, \boldsymbol{\theta})^T \\ &= \frac{1}{\sigma_\varepsilon^2} \sum_{n=0}^{N-1} \nabla f(n, \boldsymbol{\theta}) \nabla f(n, \boldsymbol{\theta})^T. \end{aligned} \quad (\text{C.15})$$

This gives a closed-form expression to compute the Fisher information matrix.

### Correlated noise

We now assume that  $\boldsymbol{\varepsilon} = [\varepsilon[0], \dots, \varepsilon[N-1]]^T$  follows a multivariate normal distribution where samples are correlated. We thus have to take into account the covariance matrix of

the noise. In what follows we assume that  $\mathbb{E}\{\varepsilon[n]\} = 0$  and therefore  $\mathbf{\Sigma}_\varepsilon = \mathbb{E}\{\varepsilon\varepsilon^T\}$ . The pdf of vector  $\mathbf{y}$  is given by

$$p_{\mathbf{y}}(\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^N |\mathbf{\Sigma}_\varepsilon|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{u})^T \mathbf{\Sigma}_\varepsilon^{-1}(\mathbf{x} - \mathbf{u})\right), \quad (\text{C.16})$$

where  $\mathbf{u} = (f(n, \boldsymbol{\theta}))_{n=0}^{N-1}$  and  $|\mathbf{\Sigma}_\varepsilon|$  is the determinant of the covariance matrix. Note that this expression also accounts for the case where the  $\varepsilon[n]$  are i.i.d. random variables. In that case, the covariance matrix is given by  $\mathbf{\Sigma}_\varepsilon = \text{diag}(\sigma_\varepsilon^2, \dots, \sigma_\varepsilon^2)$  and expression (C.16) is then equal to (C.11).

The log-likelihood function of the multivariate normal distribution is given by

$$\ell(\mathbf{y}, \boldsymbol{\theta}) = -\frac{1}{2} \log((2\pi)^N |\mathbf{\Sigma}_\varepsilon|) - \frac{1}{2}(\mathbf{y} - \mathbf{u})^T \mathbf{\Sigma}_\varepsilon^{-1}(\mathbf{y} - \mathbf{u}). \quad (\text{C.17})$$

Therefore, the derivative with respect to  $\theta_k$  is given by

$$\begin{aligned} \frac{\partial \ell(\mathbf{y}; \boldsymbol{\theta})}{\partial \theta_k} &= \frac{1}{2} \left( \frac{\partial \mathbf{u}}{\partial \theta_k} \right)^T \mathbf{\Sigma}_\varepsilon^{-1}(\mathbf{y} - \mathbf{u}) + \frac{1}{2}(\mathbf{y} - \mathbf{u})^T \mathbf{\Sigma}_\varepsilon^{-1} \left( \frac{\partial \mathbf{u}}{\partial \theta_k} \right) \\ &= \left( \frac{\partial \mathbf{u}}{\partial \theta_k} \right)^T \mathbf{\Sigma}_\varepsilon^{-1}(\mathbf{y} - \mathbf{u}). \end{aligned} \quad (\text{C.18})$$

where we have applied the fact that  $\mathbf{\Sigma}_\varepsilon$  is symmetric. Note that the term  $(\mathbf{y} - \mathbf{u})$  is equal to the noise vector  $\boldsymbol{\varepsilon}$ . If we explicitly write the matrix products we have

$$\frac{\partial \ell(\mathbf{y}; \boldsymbol{\theta})}{\partial \theta_k} = \sum_{m=0}^{N-1} \frac{\partial f(m, \boldsymbol{\theta})}{\partial \theta_k} \sum_{n=0}^{N-1} [\mathbf{\Sigma}_\varepsilon^{-1}]_{m,n} \varepsilon[n], \quad (\text{C.19})$$

where  $[\cdot]_{m,n}$  represents the  $(m+1, n+1)$  element of a matrix for indices  $m$  and  $n$  starting at  $m=0$  and  $n=0$ . The gradient of the log-likelihood is therefore given by

$$\nabla \ell(\mathbf{y}; \boldsymbol{\theta}) = \sum_{m,n=0}^{N-1} \nabla f(m, \boldsymbol{\theta}) [\mathbf{\Sigma}_\varepsilon^{-1}]_{m,n} \varepsilon[n] = \mathbf{G}^T \mathbf{\Sigma}_\varepsilon^{-1} \boldsymbol{\varepsilon}. \quad (\text{C.20})$$

where matrix  $\mathbf{G}$  is a  $N \times K$  matrix where each row is given by vector  $\nabla f(n, \boldsymbol{\theta})^T$ . By substituting this expression of the gradient of the log-likelihood function in the definition of the Fisher information matrix (C.9) we obtain the following

$$\mathbf{I}(\boldsymbol{\theta}) = \mathbb{E} \left\{ \mathbf{G}^T \mathbf{\Sigma}_\varepsilon^{-1} \boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T \underbrace{(\mathbf{\Sigma}_\varepsilon^{-1})^T}_{=\mathbf{\Sigma}_\varepsilon^{-1}} \mathbf{G} \right\} = \mathbf{G}^T \mathbf{\Sigma}_\varepsilon^{-1} \mathbb{E}\{\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T\} \mathbf{\Sigma}_\varepsilon^{-1} \mathbf{G}. \quad (\text{C.21})$$

Note that the expectation term coincides with the definition of the covariance matrix for

random variables with zero mean. We thus have

$$\mathbf{I}(\boldsymbol{\theta}) = \mathbf{G}^T \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1} \mathbf{G}. \quad (\text{C.22})$$

## C.2. CRB in the FRI setup

In the FRI framework, we have access to the following set of noisy samples

$$\tilde{y}[n] = \sum_{k=1}^K a_k \varphi\left(\frac{t_k}{T} - n\right) + \varepsilon[n], \quad (\text{C.23})$$

where the parameters to be estimated are the  $K$  pairs  $(a_k, t_k)$ . The parameters vector is therefore given by  $\boldsymbol{\theta} = [a_1, \dots, a_K, t_1, \dots, t_K]^T \in \mathbb{R}^{2K}$  and the parametric model corresponds to

$$f(n, \boldsymbol{\theta}) = \sum_{k=1}^K a_k \varphi(t_k/T - n) \quad n = 0, 1, \dots, N-1. \quad (\text{C.24})$$

We thus have that

$$\nabla f(n, \boldsymbol{\theta}) = \begin{bmatrix} \varphi(t_1/T - n) \\ \vdots \\ \varphi(t_K/T - n) \\ \frac{a_1}{T} \varphi'(t_1/T - n) \\ \vdots \\ \frac{a_K}{T} \varphi'(t_K/T - n) \end{bmatrix}. \quad (\text{C.25})$$

By replacing this expression of the gradient in matrix  $\mathbf{G}$  from Equation (C.22) and inverting  $\mathbf{I}(\boldsymbol{\theta})$  we can compute the CRB in the FRI setup. This is the approach that has been applied to obtain the CRB plots in Figure 2.10.

### C.2.1. CRB of the measurements $s[m]$

The FRI reconstruction procedures do not operate directly on the samples  $\tilde{y}[n]$ , but on the set of complex-valued measurements  $\tilde{s}[m]$  that are obtained by linearly combining the real-valued samples  $\tilde{y}[n]$  with the complex-valued coefficients  $c_{m,n}$  from (2.30). This leads to the following parametric model:

$$h(m, \boldsymbol{\theta}) = \sum_{k=1}^K a_k e^{i\omega_m t_k/T}, \quad m = 0, 1, \dots, P. \quad (\text{C.26})$$

where  $\omega_m$  is a design parameter. We can therefore compute the CRB for this parametric model which might provide a more accurate estimate of the uncertainty of the reconstruction algorithms.

The influence of the noise in the samples  $\tilde{y}[n]$  directly affects the measurements  $\tilde{s}[m]$  as

follows:

$$\tilde{s}[m] = \sum_{k=1}^K a_k e^{i\omega_m t_k/T} + \sum_{n=0}^{N-1} c_{m,n} \varepsilon[n], \quad m = 0, 1, \dots, P. \quad (\text{C.27})$$

In matricial form we can write

$$\mathbf{s} = \mathbf{C} \mathbf{y} = \mathbf{C} \mathbf{u} + \mathbf{C} \boldsymbol{\varepsilon}, \quad (\text{C.28})$$

where  $\mathbf{y} \in \mathbb{R}^N$  is the column vector of the noisy measurements,  $\mathbf{u} = (f(n, \boldsymbol{\theta}))_{n=0}^{N-1}$  is the parametric model of the real-valued samples,  $\mathbf{s} \in \mathbb{C}^{(P+1)}$  is the new sequence expressed in vectorial form and  $\mathbf{C} \in \mathbb{C}^{(P+1) \times N}$  is the matrix constructed from coefficients  $c_{m,n}$ .

When dealing with complex-valued measurements and parameters, the CRB can be obtained from the real-valued vectors built by concatenating the real and imaginary parts. However, in our setup, it can be shown [27, 59] that the Fisher information matrix is given by the following more compact expression

$$\mathbf{I}(\boldsymbol{\theta}) = \frac{1}{\sigma_\varepsilon^2} \mathbf{G}^H (\mathbf{C} \mathbf{C}^H)^{-1} \mathbf{G}, \quad (\text{C.29})$$

where  $\mathbf{G}$  is the matrix of size  $(P+1) \times 2K$  given by

$$\mathbf{G} = \begin{bmatrix} i \frac{a_1 \omega_0}{T} e^{i\omega_0 t_1/T} & \dots & i \frac{a_K \omega_0}{T} e^{i\omega_0 t_K/T} & e^{i\omega_0 t_1/T} & \dots & e^{i\omega_0 t_K/T} \\ i \frac{a_1 \omega_1}{T} e^{i\omega_1 t_1/T} & \dots & i \frac{a_K \omega_1}{T} e^{i\omega_1 t_K/T} & e^{i\omega_1 t_1/T} & \dots & e^{i\omega_1 t_K/T} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ i \frac{a_1 \omega_P}{T} e^{i\omega_P t_1/T} & \dots & i \frac{a_K \omega_P}{T} e^{i\omega_P t_K/T} & e^{i\omega_P t_1/T} & \dots & e^{i\omega_P t_K/T} \end{bmatrix}. \quad (\text{C.30})$$

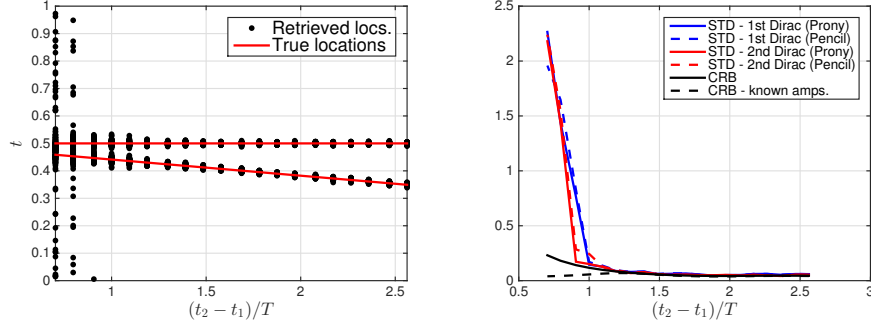
The Fisher information matrix can be computed with the compact form given in Equation (C.29) only if the coefficients  $c_{m,n}$  satisfy  $c_{m,n} = c_{P-m,n}^*$ . This is the case in our setup because we impose that the exponentials that are reproduced appear in complex conjugate pairs. Again, the CRB is obtained by inverting the Fisher information matrix  $\mathbf{I}(\boldsymbol{\theta})$ .

### Uncertainty of the location of two Diracs in terms of their distance

We now focus on the uncertainty of the location estimation, and we assume that the amplitudes of the Diracs are known and have a constant value equal to unity. In order to estimate the precision of the estimates of two Diracs that are close to each other, we restrict  $K = 2$ . For a proper choice of the sampling kernel and of the frequencies  $\omega_m$ , we can impose  $\mathbf{C} \mathbf{C}^H = \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix. It follows that the Fisher information matrix is a  $2 \times 2$  matrix and is given by

$$\mathbf{I}(\boldsymbol{\theta}) = \frac{1}{T^2 \sigma_\varepsilon^2} \begin{bmatrix} \sum_{m=0}^P \omega_m^2 & \sum_{m=0}^P \omega_m^2 \exp(i\omega_m \Delta t/T) \\ \sum_{m=0}^P \omega_m^2 \exp(-i\omega_m \Delta t/T) & \sum_{m=0}^P \omega_m^2 \end{bmatrix}, \quad (\text{C.31})$$

where  $\Delta t = t_2 - t_1$ . Note that the uncertainty in the locations depends only on the time difference of the Diracs  $\Delta t = t_2 - t_1$ . Moreover, due to the particular symmetry of the matrix  $\mathbf{I}(\boldsymbol{\theta})$ , the uncertainty in the location of both Diracs are equal. Figure C.1 illustrates the uncertainty of one of the Diracs in terms of the time difference  $\Delta t/T$ . We can observe that there is a breakdown effect when  $\Delta t$  is of the order of  $T$ .



**Figure C.1.:** Sampling and reconstruction of two Diracs with an eMOMS kernel ( $P=16$ ),  $\sigma = 0.06$  (approx. SNR = 15dB). Retrieved locations for varying distance between the two Diracs.



## Appendix D.

# Exponential reproducing kernels, eMOMS and the Dirichlet kernel

We refer the reader to Urig  en’s PhD thesis [59] for a detailed description on using eMOMS kernels to sample and reconstruct FRI signals. Here, we provide a brief introduction and their link with the Dirichlet kernel. This kernel is the most resilient to noise and the one that has been used to obtain the results in Chapters 3 and 4.

As introduced in Chapter 2, an exponential reproducing function is a function, that together with its shifted versions, is able to reproduce exponential functions:

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t - n) = e^{\alpha_m t}, \quad m = 0, 1, \dots, P \quad \text{and} \quad \alpha_m \in \mathbb{C}. \quad (\text{D.1})$$

All the functions that are considered in this appendix satisfy this property.

### D.1. E-splines and eMOMS

E-splines are a family of functions that reproduce exponentials. Moreover, E-splines are the functions of minimal support that are able to reproduce exponentials. An E-spline of order  $P$  reproduces  $P + 1$  exponentials. Let  $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_P)$  be the vector of the parameters of the exponential functions to be reproduced. The Fourier transform of an E-spline of order  $P$  is given by:

$$\hat{\beta}_{\boldsymbol{\alpha}}(\omega) = \prod_{m=0}^P \frac{1 - e^{\alpha_m - i\omega}}{i\omega - \alpha_m}. \quad (\text{D.2})$$

An interesting property of E-splines is that they preserve the exponential reproduction property through convolution. This means that if  $\beta_{\boldsymbol{\alpha}}(t)$  reproduces exponentials  $\{e^{\alpha_m t}\}_{m=0}^P$ , any other function  $\varphi(t) = \beta_{\boldsymbol{\alpha}}(t) * \gamma(t)$  also reproduces the same exponentials (provided that  $\int_{-\infty}^{+\infty} \gamma(t) e^{-\alpha_m t} dt \neq 0$ ). If we take the Fourier transform of  $\varphi(t)$  we have that

$$\hat{\varphi}(\omega) = \hat{\beta}_{\boldsymbol{\alpha}}(\omega) \hat{\gamma}(\omega). \quad (\text{D.3})$$

We have a particular case that leads to a family of functions called MOMS (minimal support maximal order) [60] when  $\hat{\gamma}(\omega)$  is equal to

$$\hat{\gamma}(\omega) = \sum_{\ell=0}^P a_{\ell} (i\omega)^{\ell}, \quad (\text{D.4})$$

for some coefficients  $a_{\ell}$ . From the Fourier transform properties we can see that  $\gamma(t)$  is a differential operator since

$$\mathcal{F} \left\{ \frac{d^n f(t)}{dt^n} \right\} = (i\omega)^n \hat{f}(\omega). \quad (\text{D.5})$$

Therefore, the expression in time of  $\varphi(t)$  is given by

$$\varphi(t) = \sum_{\ell=0}^P a_{\ell} \frac{d^{\ell} \beta_{\alpha}(t)}{dt^{\ell}}. \quad (\text{D.6})$$

Since  $\beta_{\alpha}(t)$  is a function of minimal support that reproduces the exponentials  $\{e^{\alpha_m t}\}_{m=0}^P$ , and  $\varphi(t)$  is constructed from a linear combination of  $\beta_{\alpha}(t)$  and its derivatives,  $\varphi(t)$  is also of minimal support and reproduces the same exponentials. This is why  $\varphi(t)$  is called eMOMS, where the letter “e” stands for *exponential*.

#### D.1.1. eMOMS and stability of reconstruction of FRI signals

The extra degree of freedom that is provided by the coefficients  $a_{\ell}$  in (D.4) can be useful to design sampling kernels that are more resilient to noise. In Chapter 2 we have seen that the coefficients  $c_{m,n}$  of the exponential reproducing formula play a crucial role in the FRI algorithms. Moreover, when the parameters  $\alpha_m$  are purely imaginary, that is  $\alpha_m = i\omega_m$ , these coefficients  $c_{m,n}$  are directly linked to the amplitude of the Fourier transform of the sampling kernel at the frequencies that are reproduced.

From (A.12) and (D.3) we have that

$$c_{m,0} = [\hat{\varphi}(\omega_m)]^{-1} = \left[ \hat{\gamma}(\omega_m) \hat{\beta}_{\alpha}(\omega_m) \right]^{-1}. \quad (\text{D.7})$$

The most resilient to noise sampling scheme is obtained when the noise in the measurements  $\tilde{s}[m] = \sum_n c_{m,n} (y[n] + \varepsilon[n])$  is *white*, and this is achieved when we have

$$|c_{m,0}| = 1 \quad (\text{D.8})$$

(see Equation (2.78)). We thus impose

$$|\hat{\gamma}(\omega_m)| = \left| \hat{\beta}_{\alpha}(\omega_m) \right|^{-1}. \quad (\text{D.9})$$

This condition is used to find the values of the coefficients  $a_{\ell}$ . Note that we have an extra

degree of freedom on the phase of  $\hat{\gamma}(\omega_m)$ . This degree of freedom leads to a variety of sampling kernels that satisfy the condition  $|c_{m,0}| = 1$ .

If we impose the more restrictive condition

$$c_{m,0} = 1 \quad \Rightarrow \quad \hat{\varphi}(\omega_m) = 1, \quad (\text{D.10})$$

it follows that  $\hat{\gamma}(\omega_m) = [\hat{\beta}_{\alpha}(\omega_m)]^{-1}$ , and, under some extra conditions on the frequencies  $\omega_m$ , the kernel  $\varphi(t)$  corresponds to a period of the Dirichlet kernel as it is shown next.

We have that

$$b_m^{-1} \stackrel{\text{def}}{=} \hat{\beta}_{\alpha}(\omega_m) = \prod_{\substack{n=0 \\ n \neq m}}^P \frac{1 - e^{-i(\omega_m - \omega_n)}}{i(\omega_m - \omega_n)}, \quad (\text{D.11})$$

and considering that  $i\omega_m = \alpha_m$ , this can also be written as

$$b_m = \prod_{\substack{n=0 \\ n \neq m}}^P \frac{\alpha_m - \alpha_n}{1 - e^{\alpha_n - \alpha_m}}, \quad m = 0, 1, \dots, P. \quad (\text{D.12})$$

Since  $\hat{\gamma}(\omega)$  is a polynomial in  $i\omega$ , it is more convenient to express  $\hat{\gamma}(\omega)$  in terms of a complex variable  $s$  that will be evaluated in  $i\omega$ :

$$\Gamma(s) = \sum_{\ell=0}^P a_{\ell} s^{\ell}. \quad (\text{D.13})$$

The function  $\Gamma(s)$  is a polynomial of order  $P$  in the complex variable  $s$  and we have that  $\hat{\gamma}(\omega) = \Gamma(s)|_{s=i\omega}$ . Note that  $\Gamma(s)$  corresponds to the bilateral Laplace transform of  $\gamma(t)$ . The polynomial  $\Gamma(s)$  can also be seen as an interpolating polynomial that interpolates the set of points

$$\{(\alpha_0, b_0), (\alpha_1, b_1), \dots, (\alpha_P, b_P)\}, \quad (\text{D.14})$$

that is,

$$\Gamma(s)|_{s=\alpha_m} = b_m, \quad m = 0, 1, \dots, P. \quad (\text{D.15})$$

We can thus build  $\Gamma(s)$  from the polynomial interpolation formula:

$$\Gamma(s) = \sum_{m=0}^P \left( \prod_{\substack{n=0 \\ n \neq m}}^P \frac{s - \alpha_n}{\alpha_m - \alpha_n} \right) b_m. \quad (\text{D.16})$$

By replacing (D.12) in (D.16) and setting  $s = i\omega$  we obtain

$$\hat{\gamma}(\omega) = \Gamma(i\omega) = \sum_{m=0}^P \prod_{\substack{n=0 \\ n \neq m}}^P \frac{i\omega - \alpha_n}{1 - e^{\alpha_n - \alpha_m}}. \quad (\text{D.17})$$

This expression provides a closed-form expression for  $\hat{\gamma}(\omega)$ . It follows that the closed-form expression for the Fourier transform of the eMOMS is given by:

$$\hat{\varphi}(\omega) = \hat{\gamma}(\omega) \hat{\beta}_{\alpha}(\omega) = \sum_{m=0}^P \prod_{\substack{n=0 \\ n \neq m}}^P \frac{i\omega - \alpha_n}{1 - e^{\alpha_n - \alpha_m}} \prod_{\ell=0}^P \frac{1 - e^{\alpha_{\ell} - i\omega}}{i\omega - \alpha_{\ell}}. \quad (\text{D.18})$$

Next, it is shown that this approach leads to numerical instabilities. Moreover, the resulting kernel is not continuous. We also show how to overcome these limitations in order to obtain a stable and continuous eMOMS.

### D.1.2. Dirichlet kernel and e-MOMS

The Dirichlet kernel of order  $N$  is given by

$$D_N(t) \stackrel{\text{def}}{=} \sum_{n=-N}^N e^{int} = \frac{\sin((N + 1/2)t)}{\sin(t/2)}. \quad (\text{D.19})$$

The Dirichlet kernel, in its original form, is  $2\pi$ -periodic. Since the sampling kernel  $\varphi(t)$  has a support of  $P + 1$ , we consider the following modified function:

$$D_N\left(\frac{2\pi}{P+1}t\right) = \sum_{n=-N}^N e^{in\frac{2\pi}{P+1}t}, \quad (\text{D.20})$$

which is  $(P + 1)$ -periodic.

Let us consider the  $(P + 1)$ -periodic repetition of the function  $\varphi(t)$  with Fourier transform given by Equation (D.18). Applying the Poisson summation formula we can obtain an expression of this periodic function in terms of its Fourier transform:

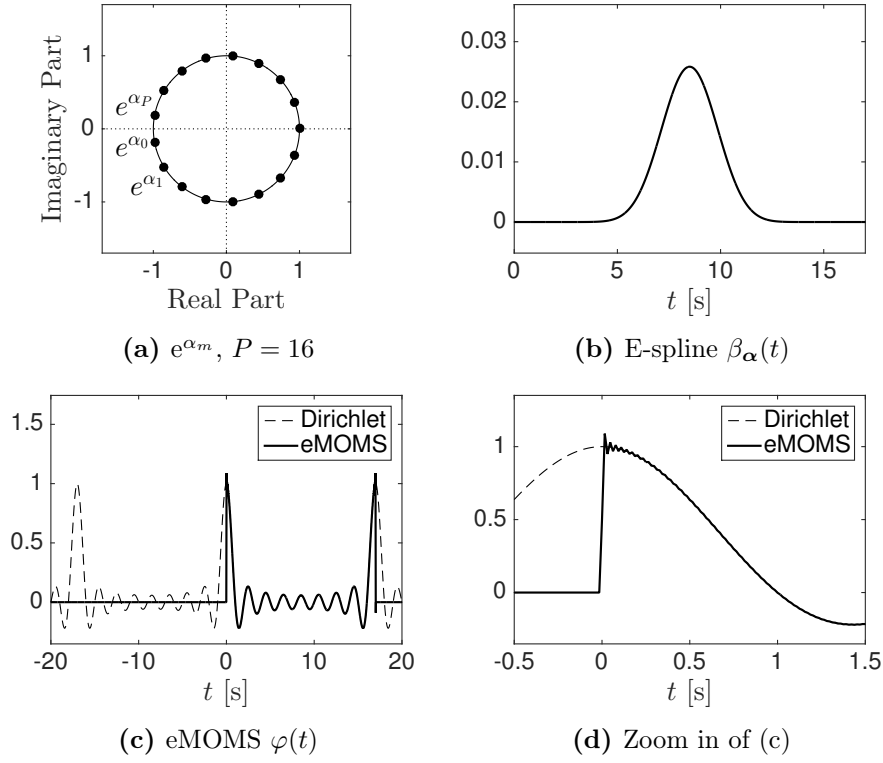
$$\sum_{n \in \mathbb{Z}} \varphi(t - n(P + 1)) = \frac{1}{P + 1} \sum_{k \in \mathbb{Z}} \hat{\varphi}\left(\frac{2\pi k}{P + 1}\right) e^{ik\frac{2\pi}{P+1}t}. \quad (\text{D.21})$$

If we choose the parameters  $\alpha_m$  so that  $e^{\alpha_m}$  spans the entire unit circle, for instance,

$$\alpha_m = i\omega_m = i\frac{2\pi}{P + 1} \left(m - \frac{P}{2}\right), \quad (\text{D.22})$$

where  $m = 0, 1, \dots, P$ ; and if  $P$  is even, we have that

$$\hat{\varphi}\left(\frac{2\pi k}{P + 1}\right) = \begin{cases} 1 & \text{for } k = -P/2, \dots, P/2, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{D.23})$$



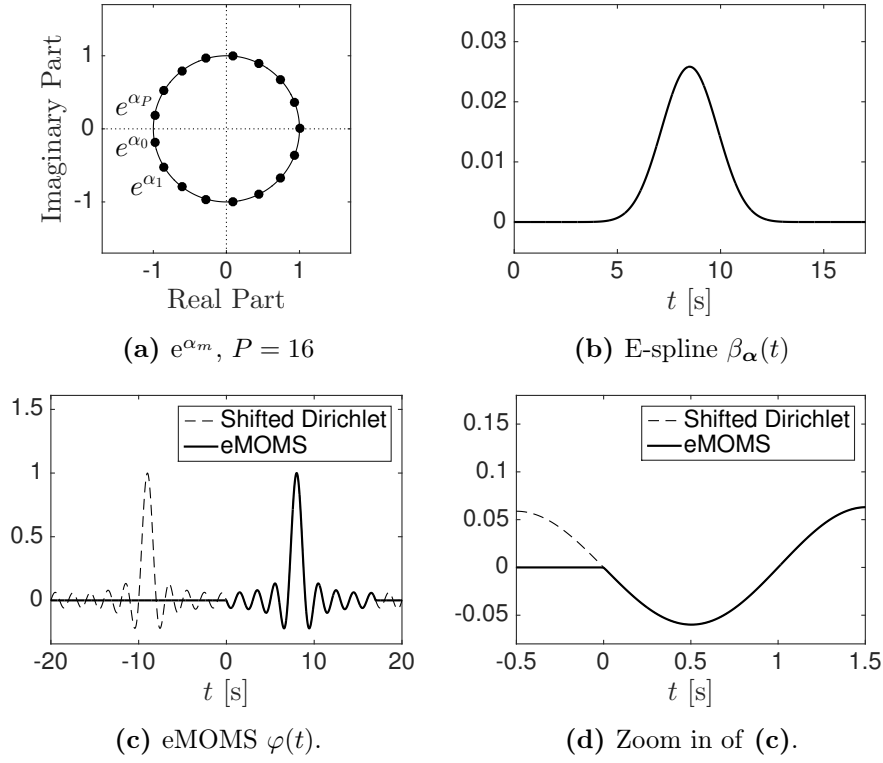
**Figure D.1.:**  $P = 16$ , E-spline, eMOMS and Dirichlet kernel. (a) Parameters  $\alpha_m$  are spread over the entire unit circle. (b) E-spline that reproduces the exponential functions with parameter  $e^{\alpha_m}$  given in (a). (c) eMOMS  $\varphi(t)$  of compact support  $[0, (P + 1)]$  and scaled Dirichlet kernel. The resulting kernel  $\varphi(t)$  is not continuous at  $t = 0$  and  $t = P + 1$  and presents some numerical instabilities as can be seen in (d).

We thus have

$$\sum_{n \in \mathbb{Z}} \varphi(t - n(P + 1)) = \frac{1}{P + 1} \sum_{k=-P/2}^{P/2} e^{ik \frac{2\pi}{P+1} t} = \frac{1}{P + 1} D_{P/2} \left( \frac{2\pi}{P+1} t \right), \quad (\text{D.24})$$

where  $D_{P/2}(t)$  is the Dirichlet kernel of order  $P/2$ . We can thus conclude that the function  $\varphi(t)$ , with Fourier transform given in Equation (D.18), corresponds to one period of the normalised and scaled Dirichlet kernel of order  $P/2$ . Figure D.1 illustrates these functions for  $P = 16$ . In (c) and (d) we can see that this approach leads to an unstable reproduction of the Dirichlet kernel, since the maximum of the scaled version of  $D_N(\frac{2\pi}{P+1} t)$  is located at  $t = 0$  and  $t = P + 1$ , however, the B-spline vanishes in this region as can be seen in (b). It is therefore very unstable to reproduce these maxima with a linear combination of  $\beta_\alpha(t)$  and its derivatives.

Note that this eMOMS has been obtained by imposing the more restrictive constraint  $\hat{\varphi}(\omega_m) = 1$ . This condition can be relaxed to  $|\hat{\varphi}(\omega_m)| = 1$  and still the FRI algorithms benefit from the fact that the noise in the measurements  $s[m]$  is white. In order to avoid



**Figure D.2.:** Shifted Dirichlet kernel and stable eMOMS. Note that this eMOMS is continuous, which was not the case in the eMOMS depicted in Figure D.1.

the numerical instabilities, we now reproduce one period of a shifted version of the Dirichlet kernel so that the maximum is located in the region where  $\beta_\alpha(t)$  has its maximum and the resulting eMOMS is continuous.

Consider the following scaled and shifted by  $\frac{P}{2}$  Dirichlet kernel:

$$D_{P/2} \left( \frac{2\pi}{P+1} \left( t - \frac{P}{2} \right) \right) = \sum_{n=-P/2}^{P/2} e^{in \frac{2\pi}{P+1} (t - P/2)} = \sum_{n=-P/2}^{P/2} e^{-ik\pi \frac{P}{P+1}} e^{in \frac{2\pi}{P+1} t}. \quad (\text{D.25})$$

It is easy to verify that:

- $D_{P/2} \left( \frac{2\pi}{P+1} \left( t - \frac{P}{2} \right) \right) \Big|_{t=0} = 0,$
- $D_{P/2} \left( \frac{2\pi}{P+1} \left( t - \frac{P}{2} \right) \right) \Big|_{t=P+1} = 0.$

From the expression of the periodised  $\varphi(t)$  obtained in (D.25) and the term  $e^{-ik\pi \frac{P}{P+1}}$  that appears in (D.25), it follows that the eMOMS will correspond to one period of the shifted and scaled Dirichlet kernel if we impose

$$\hat{\varphi}(\omega_m) = e^{-i \left( m - \frac{P}{2} \right) \pi \frac{P}{P+1}} = e^{-i\omega_m P/2}, \quad (\text{D.26})$$

where we have applied the fact that  $\omega_m = \frac{2\pi}{P+1} (m - \frac{P}{2})$ . It follows that the Fourier transform of the function  $\gamma(t)$  is given by

$$\hat{\varphi}(\omega) = \sum_{m=0}^P e^{-i\omega_m P/2} \prod_{\substack{n=0 \\ n \neq m}}^P \frac{i\omega - \alpha_n}{1 - e^{\alpha_n - \alpha_m}}. \quad (\text{D.27})$$

This leads to the following eMOMS:

$$\hat{\varphi}(\omega) = \hat{\gamma}(\omega) \hat{\beta}_{\alpha}(\omega) = \sum_{m=0}^P e^{-i\omega_m P/2} \prod_{\substack{n=0 \\ n \neq m}}^P \frac{i\omega - \alpha_n}{1 - e^{\alpha_n - \alpha_m}} \prod_{\ell=0}^P \frac{1 - e^{\alpha_{\ell} - i\omega}}{i\omega - \alpha_{\ell}}. \quad (\text{D.28})$$

This eMOMS is illustrated in Figure D.2 where we can appreciate that this function is continuous and do not present the numerical instabilities of the eMOMS in Figure D.1.



## Appendix E.

### Maximum of Rayleigh random variables

Here we analyse the statistics of the maximum of a collection of i.i.d. random variables which are Rayleigh-distributed. In probability theory and statistics, the study of the distribution of the maximum (or minimum) of a number of samples is known as extreme value theory. In 1928, Fisher and Tippett [127] showed that the minimum or the maximum of a very large collection of i.i.d. random variables have a limiting distribution that can be categorised into three different classes. These three classes correspond to the Gumbel, Fréchet and Weibull distributions [131, 132]. In general, this type of analysis can present considerable difficulties, however, the derivations are hugely simplified when the random variables are Rayleigh-distributed.

#### E.1. Maximum of $N$ random variables

Let  $X_1, \dots, X_N$  be  $N$  i.i.d. random variables. These random variables are characterised by the following two functions:

- *Probability density function (PDF) of  $X_n$ :*  $f_X(x)$ .
- *Cumulative distribution function (CDF):*  $F_X(x) = \mathbb{P}\{X_n \leq x\} = \int_{-\infty}^x f_X(t) dt$ .

Note that the  $N$  random variables are characterised by the same  $f_X(x)$  and  $F_X(x)$  since we assume that they are identically distributed. The PDF can also be expressed in terms of the CDF as follows:

$$f_X(x) = F'_X(x). \quad (\text{E.1})$$

We define the following new random variable:

$$A_N = \max(X_1, \dots, X_N). \quad (\text{E.2})$$

The CDF of  $A_N$  is given by

$$F_{A_N}(x) = \mathbb{P}\{X_1 \leq x, \dots, X_N \leq x\} \stackrel{\text{indep.}}{=} \mathbb{P}\{X_1 \leq x\} \dots \mathbb{P}\{X_N \leq x\} = [F_X(x)]^N. \quad (\text{E.3})$$

It follows that the PDF is given by

$$f_{A_N}(x) = F'_{A_N}(x) = N [F_X(x)]^{N-1} f_X(x). \quad (\text{E.4})$$

## E.2. Rayleigh distribution and maximum of random variables

We now assume that the random variables follow a Rayleigh distribution with parameter  $\sigma$ . We thus have:

$$f_X(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \quad x \geq 0, \quad (\text{E.5})$$

$$F_X(x) = 1 - e^{-\frac{x^2}{2\sigma^2}}, \quad x \geq 0. \quad (\text{E.6})$$

The PDF of the random variable  $A_N$  is therefore given by

$$f_{A_N}(x) = N \left(1 - e^{-\frac{x^2}{2\sigma^2}}\right)^{N-1} \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \quad x \geq 0. \quad (\text{E.7})$$

If we apply the following change of variable  $x = \sigma\sqrt{2 \log y}$ , where we have that, if  $x \geq 0$  then  $y \geq 1$ , the PDF of  $A_N$  becomes

$$f_{A_N}(y) = \frac{\sqrt{2} N}{\sigma} \left(1 - \frac{1}{y}\right)^{N-1} \frac{\sqrt{\log y}}{y}, \quad y \geq 1. \quad (\text{E.8})$$

Taking the derivative with respect to  $y$  we have that

$$\begin{aligned} f'_{A_N}(y) &= \frac{\sqrt{2} N (N-1)}{\sigma} \left(1 - \frac{1}{y}\right)^{N-2} \frac{\sqrt{\log y}}{y^3} \\ &\quad + \frac{\sqrt{2} N}{\sigma} \left(1 - \frac{1}{y}\right)^{N-1} \frac{1/2}{\sqrt{\log y} y^2} - \frac{\sqrt{2} N}{\sigma} \left(1 - \frac{1}{y}\right)^{N-1} \frac{\sqrt{\log y}}{y^2}, \end{aligned} \quad (\text{E.9})$$

which in turn leads to

$$f'_{A_N}(y) = \frac{\sqrt{2} N}{\sigma} \left(1 - \frac{1}{y}\right)^{N-1} \frac{\sqrt{\log y}}{y^2} \left(\frac{N-1}{y-1} + \frac{1}{\log y} - 1\right). \quad (\text{E.10})$$

Let  $y = \alpha N$ , for some  $\alpha > 0$ , it follows that

$$f'_A(\alpha N) = \frac{\sqrt{2}}{\sigma} \left(1 - \frac{1}{\alpha N}\right)^{N-1} \frac{\sqrt{\log \alpha N}}{\alpha^2 N} \left(\frac{N-1}{\alpha N-1} + \frac{1}{\log \alpha N} - 1\right). \quad (\text{E.11})$$

For large  $N$ , if  $\alpha = 1$ , then

$$\frac{N-1}{\alpha N-1} + \frac{1}{\log \alpha N} - 1 \rightarrow 0. \quad (\text{E.12})$$

Moreover, for large  $N$  and  $y$

$$\frac{N-1}{y-1} + \frac{1}{\log y} - 1 \rightarrow \frac{N-1}{y-1} - 1 \quad (\text{E.13})$$

and

$$y < N \Rightarrow \frac{N-1}{y-1} - 1 > 0, \quad (\text{E.14})$$

$$y > N \Rightarrow \frac{N-1}{y-1} - 1 < 0. \quad (\text{E.15})$$

Therefore, for large  $N$ ,  $f_A(y)$  presents a maximum at  $y = N$ . Which leads to,

$$\lim_{N \rightarrow \infty} \arg \max_x f_A(x) = \sigma \sqrt{2 \log N}. \quad (\text{E.16})$$

Moreover, if we replace this value for  $x$  in (E.7) we obtain

$$f_{A_N}(\sigma \sqrt{2 \log N}) = \frac{\sqrt{2}}{\sigma} \sqrt{\log N} \left(1 - \frac{1}{N}\right)^{N-1}. \quad (\text{E.17})$$

It follows that

$$\lim_{N \rightarrow \infty} f_{A_N}(\sigma \sqrt{2 \log N}) = \frac{\sqrt{2 \log N}}{\sigma e}. \quad (\text{E.18})$$

Note that

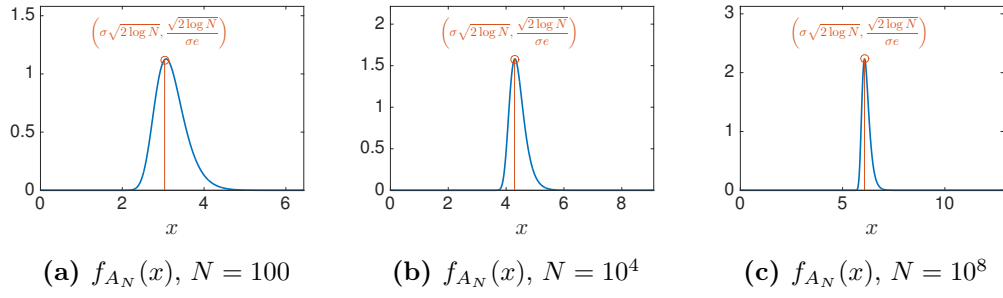
$$\lim_{N \rightarrow \infty} f_{A_N}(\sigma \sqrt{2 \log N}) \rightarrow \infty. \quad (\text{E.19})$$

However,

$$\int_{-\infty}^{+\infty} f_{A_N}(t) dt = 1. \quad (\text{E.20})$$

We can thus conclude that, for large  $N$ , the maximum value of a set of  $N$  i.i.d. samples that follow a Rayleigh distribution with parameter  $\sigma$  is equal to  $\sigma \sqrt{2 \log N}$  with high probability.

The PDF of the maximum value of a collection of  $N$  Rayleigh random variables is illustrated in Figure E.1 for different values of  $N$ . We can observe that, as  $N$  increases, the PDF is more concentrated around the maximum value of  $\sigma \sqrt{2 \log N}$ . However, the convergence is extremely slow.



**Figure E.1.:** PDF of the random variable defined as the maximum of  $N$  Rayleigh distributed random variables for different values of  $N$  with parameter  $\sigma = 1$ .

# Bibliography

- [1] J. Oñativia, Y. M. Lu, and P. L. Dragotti. “Sparsity According to Prony: Average Performance Analysis and Phase Transition”. *to be submitted* (2015).
- [2] J. Oñativia and P. L. Dragotti. “Sparse sampling: theory, methods and an application in neuroscience”. *Biological cybernetics* 109.1 (Feb. 2015), pp. 125–139.
- [3] J. Oñativia, S. Schultz, and P. L. Dragotti. “A Finite Rate of Innovation Algorithm for Fast and Accurate Spike Detection from Two-Photon Calcium Imaging”. *Journal of Neural Engineering* 10.4 (July 2013), pp. 1–14.
- [4] S. Reynolds, J. Oñativia, C. S. Copeland, S. R. Schultz, and P. L. Dragotti. “Spike Detection Using FRI Methods and Protein Calcium Sensors: Performance Analysis and Comparisons”. *2015 International Conference on Sampling Theory and Applications (SampTA)*. May 2015, pp. 533–537.
- [5] J. Oñativia, Y. M. Lu, and P. L. Dragotti. “Sparsity Pattern Recovery Using FRI Methods”. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’15 — Brisbane, Australia)*. Apr. 2015.
- [6] J. Oñativia, Y. M. Lu, and P. L. Dragotti. “Finite Dimensional FRI”. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’14 — Florence, Italy)*. May 2014, pp. 1808–1812.
- [7] P. L. Dragotti, J. Oñativia, J. A. Urigüen, and T. Blu. “Approximate Strang-Fix: Sampling infinite streams of Diracs with any kernel”. *Proceedings SPIE 8858, Wavelets and Sparsity XV*. Vol. 8858. Sept. 2013, 88580Y–88580Y–8.
- [8] J. Oñativia, J. A. Urigüen, and P. L. Dragotti. “Sequential Local FRI Sampling of Infinite Streams of Diracs”. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’13 — Vancouver, Canada)*. May 2013, pp. 5440–5444.
- [9] J. Oñativia, P. L. Dragotti, and Y. M. Lu. “Sparsity According to Prony, average performance analysis”. *SPARS Workshop 2015 (Cambridge, UK)*. July 2015.
- [10] J. Oñativia and P. L. Dragotti. “Sparsity According to Prony: From Structured to Unstructured Representations and Back”. *International BASP Frontiers workshop (Villars-sur-Ollon, Switzerland)*. Jan. 2015.
- [11] J. Oñativia and P. L. Dragotti. “Finite dimensional FRI for reconstruction of sparse signals”. *UCL-Duke workshop on Sensing and Analysis of High-Dimensional Data (London, UK)*. Sept. 2014.
- [12] S. Schultz, J. Oñativia, J. A. Urigüen, and P. L. Dragotti. “A Finite Rate of Innovation Algorithm for Spike Detection from Two-Photon Calcium Imaging”. *Neuroscience annual meeting (New Orleans, USA)*. Oct. 2012.

- [13] C. E. Shannon. “Communication in the Presence of Noise”. *Proceedings of the IEEE* 37.1 (Jan. 1949), pp. 10–21.
- [14] J. M. Whittaker. “The “Fourier” Theory of the Cardinal Function”. *Proceedings of the Edinburgh Mathematical Society (Series 2)* 1.3 (July 1928), pp. 169–176.
- [15] V. A. Kotelnikov. “On the Transmission Capacity of “Ether” and Wire in Electrocommunications”. *Izd. Red. Upr. Svyazzi RKKA (Moscow)* (1933).
- [16] M. Unser. “Sampling – 50 Years After Shannon”. *Proceedings of the IEEE* 88.4 (Apr. 2000), pp. 569–587.
- [17] M. Vetterli, P. Marziliano, and T. Blu. “Sampling Signals with Finite Rate of Innovation”. *IEEE Transactions on Signal Processing* 50.6 (June 2002), pp. 1417–1428.
- [18] A. V. Oppenheim and A. S. Willky. *Signals and Systems*. 2nd ed. Prentice Hall, 1996.
- [19] H. Nyquist. “Certain Factors Affecting Telegraph Speed”. *Transactions of the American Institute of Electrical Engineers* 43 (Jan. 1924), pp. 412–422.
- [20] H. Nyquist. “Certain Topics in Telegraph Transmission”. *Transactions of the American Institute of Electrical Engineers* 47.2 (Apr. 1928), pp. 617–644.
- [21] D. Slepian. “On Bandwidth”. *Proceedings of the IEEE* 64.3 (Mar. 1976), pp. 292–300.
- [22] P. A. M. Dirac. *The Principles of Quantum Mechanics*. 4th ed. Oxford: Oxford University Press, 1958.
- [23] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd ed. Academic Press, Jan. 2009.
- [24] 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* 55.5 (May 2007), pp. 1741–1757.
- [25] T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot. “Sparse Sampling of Signal Innovations”. *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 31–40.
- [26] J. Berent, P. L. Dragotti, and T. Blu. “Sampling Piecewise Sinusoidal Signals with Finite Rate of Innovation Methods”. *IEEE Transactions on Signal Processing* 58.2 (Feb. 2010), pp. 613–625.
- [27] J. A. Urigüen, T. Blu, and P. L. Dragotti. “FRI Sampling with Arbitrary Kernels”. *IEEE Transactions on Signal Processing* 61.21 (Nov. 2013), pp. 5310–5323.
- [28] C. Vonesch, T. Blu, and M. Unser. “Generalized Daubechies Wavelet Families”. *IEEE Transactions on Signal Processing* 55.9 (Sept. 2007), pp. 4415–4429.
- [29] G. Strang and G. J. Fix. “A Fourier Analysis of the Finite Element Variational Method”. In: *Constructive Aspects of Functional Analysis*. Edizioni Cremonese, Rome, 1971, pp. 793–840.
- [30] I. J. Schoenberg. *Cardinal Spline Interpolation*. Society for Industrial and Applied Mathematics, 1973.
- [31] J. Ferguson. “Multivariable Curve Interpolation”. *Journal of the Association for Computing Machinery* 11.2 (Apr. 1964), pp. 221–228.

- 
- [32] G. Birkhoff and C. R. De Boor. “Piecewise Polynomial Interpolation and Approximation”. *Proc. General Motors Symposium of 1964*. Ed. by H. L. Garabedian. New York and Amsterdam: Elsevier, 1965, pp. 164–190.
  - [33] M. Unser, A. Aldroubi, and M. Eden. “Fast B-Spline Transforms for Continuous Image Representation and Interpolation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.3 (Mar. 1991), pp. 277–285.
  - [34] M. Unser. “Splines: A Perfect Fit for Signal and Image Processing”. *IEEE Signal Processing Magazine* 16.6 (Nov. 1999), pp. 22–38.
  - [35] M. Unser. “Ten Good Reasons for Using Spline Wavelets”. *Proceedings SPIE 3169, Wavelet Applications in Signal and Image Processing V*. Vol. 3169. Oct. 1997, pp. 422–431.
  - [36] I. J. Schoenberg. “Contributions to the problem of approximation of equidistant data by analytic functions”. *Quarterly of Applied Mathematics* 4 (1946), pp. 45–99, 112–141.
  - [37] M. Unser, A. Aldroubi, and M. Eden. “On the Asymptotic Convergence of B-Spline Wavelets to Gabor Functions”. *IEEE Transactions on Information Theory* 38.2 (Mar. 1992), pp. 864–872.
  - [38] I. Khalidov, T. Blu, and M. Unser. “Generalized L-Spline Wavelet Bases”. *Proceedings SPIE 5914, Wavelets XI*. Vol. 5914. Sept. 2005, 59140F–59140F–8.
  - [39] M. Sakai and R. A. Usmani. “On Exponential Splines”. *Journal of Approximation Theory* 47.2 (June 1986), pp. 122–131.
  - [40] B. J. McCartin. “Theory of Exponential Splines”. *Journal of Approximation Theory* 66.1 (July 1991), pp. 1–23.
  - [41] M. Unser and T. Blu. “Cardinal Exponential Splines: Part I – Theory and Filtering Algorithms”. *IEEE Transactions on Signal Processing* 53.4 (Apr. 2005), pp. 1425–1438.
  - [42] R. Delgado-Gonzalo, P. Thévenaz, and M. Unser. “Exponential Splines and Minimal-Support Bases for Curve Representation”. *Computer Aided Geometric Design* 29.2 (2012), pp. 109–128.
  - [43] P. Stoica and R. Moses. *Spectral Analysis of Signals*. 1st ed. Prentice Hall, Mar. 2005.
  - [44] V. F. Pisarenko. “The Retrieval of Harmonics from a Covariance Function”. *Geophysical Journal of the Royal Astronomical Society* 33.3 (Sept. 1973), pp. 347–366.
  - [45] Y. Hua and T. K. Sarkar. “Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise”. *IEEE Transactions on Acoustics, Speech and Signal Processing* 38.5 (May 1990), pp. 814–824.
  - [46] Y. Hua and T. K. Sarkar. “On SVD for Estimating Generalized Eigenvalues of Singular Matrix Pencil in Noise”. *IEEE Transactions on Signal Processing* 39.4 (Apr. 1991), pp. 892–900.
  - [47] A. Paulraj, R. H. Roy, and T. Kailath. “Estimation of Signal Parameters via Rotational Invariance Techniques – ESPRIT”. *19th Asilomar Conference on Circuits, Systems and Computers, 1985*. Nov. 1985, pp. 83–89.

- [48] T.-J. Shan, M. Wax, and T. Kailath. “On Spatial Smoothing for Direction-of-Arrival Estimation of Coherent Signals”. *IEEE Transactions on Acoustics, Speech and Signal Processing* 33.4 (Aug. 1985), pp. 806–811.
- [49] R. O. Schmidt. “Multiple Emitter Location and Signal Parameter Estimation”. *IEEE Transactions on Antennas and Propagation* 34.3 (Mar. 1986), pp. 276–280.
- [50] P. Milanfar, G. C. Verghese, W. C. Karl, and A. S. Willsky. “Reconstructing Polygons from Moments with Connections to Array Processing”. *IEEE Transactions on Signal Processing* 43.2 (Feb. 1995), pp. 432–443.
- [51] M. Elad, P. Milanfar, and G. H. Golub. “Shape from Moments — An Estimation Theory Perspective”. *IEEE Transactions on Signal Processing* 52.7 (July 2004), pp. 1814–1829.
- [52] R. Prony. “Essai expérimental et analytique sur les lois de la dilatabilité de fluides élastiques et sur celles de la force expansive de la vapeur de l’eau et de la vapeur de l’alkool, à différentes températures”. *Journal de l’École Polytechnique* 1.22 (1795), pp. 24–76.
- [53] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 4th ed. The Johns Hopkins University Press, 2013.
- [54] I. Maravić and M. Vetterli. “Sampling and Reconstruction of Signals with Finite Rate of Innovation in the Presence of Noise”. *IEEE Transactions on Signal Processing* 53.8 (Aug. 2005), pp. 2788–2805.
- [55] J. A. Cadzow. “Signal Enhancement—A Composite Property Mapping Algorithm”. *IEEE Transactions on Acoustics, Speech and Signal Processing* 36.1 (Jan. 1988), pp. 49–62.
- [56] H. Cramér. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press, 1951.
- [57] C. R. Rao. *Linear Statistical Inference and Its Applications*. New York: Wiley, 1965.
- [58] B. Porat and B. Friedlander. “Computation of the Exact Information Matrix of Gaussian Time Series with Stationary Random Components”. *IEEE Transactions on Acoustics, Speech and Signal Processing* 34.1 (Feb. 1986), pp. 118–130.
- [59] J. A. Urigüen. *Exact and Approximate Strang-Fix Conditions to Reconstruct Signals with Finite Rate of Innovation from Samples Taken with Arbitrary Kernels*. PhD Thesis, Imperial College London, 2013.
- [60] T. Blu, P. Thévenaz, and M. Unser. “MOMS: Maximal-Order Interpolation of Minimal Support”. *IEEE Transactions on Image Processing* 10 (July 2001), pp. 1069–1080.
- [61] J. A. Urigüen, P. L. Dragotti, and T. Blu. “On the Exponential Reproducing Kernels for Sampling Signals with Finite Rate of Innovation”. *9th International Workshop on Sampling Theory and Applications (SampTA’11)*. May 2011.
- [62] P. L. Dragotti, M. Vetterli, and T. Blu. “Exact Sampling Results for Signals with Finite Rate of Innovation Using Strang-Fix Conditions and Local Kernels”. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*. Vol. 4. Mar. 2005, pp. iv/233–iv/236.
- [63] R. Tur, Y. C. Eldar, and Z. Friedman. “Innovation Rate Sampling of Pulse Streams with Application to Ultrasound Imaging”. *IEEE Transactions on Signal Processing* 59.4 (Apr. 2011), pp. 1827–1842.

- 
- [64] W. Göbel and F. Helmchen. “In Vivo Calcium Imaging of Neural Network Function”. *Physiology* 22.6 (Dec. 2007), pp. 358–365.
  - [65] W. Denk, J. H. Strickler, and W. W. Webb. “Two-Photon Laser Scanning Fluorescence Microscopy”. *Science* 248.4951 (Apr. 1990), pp. 73–76.
  - [66] W. Denk, K. R. Delaney, A. Gelperin, D. Kleinfeld, B. W. Strowbridge, D. W. Tank, and R. Yuste. “Anatomical and functional imaging of neurons using 2-photon laser scanning microscopy”. *Journal of Neuroscience Methods* 54.2 (Oct. 1994), pp. 151–162.
  - [67] K. Svoboda, F. Helmchen, W. Denk, and D. W. Tank. “Spread of dendritic excitation in layer 2/3 pyramidal neurons in rat barrel cortex in vivo”. *Nature Neuroscience* 2.1 (Jan. 1999), pp. 65–73.
  - [68] C. Stosiek, O. Garaschuk, K. Holthoff, and A. Konnerth. “In vivo two-photon calcium imaging of neuronal networks”. *Proceedings of the National Academy of Sciences of the United States of America* 100.12 (June 2003), pp. 7319–7324.
  - [69] J. Csicsvari, D. A. Henze, B. Jamieson, K. D. Harris, A. Sirota, P. Barthó, K. D. Wise, and G. Buzsáki. “Massively Parallel Recording of Unit and Local Field Potentials with Silicon-Based Electrodes”. *Journal of Neurophysiology* 90.2 (2003), pp. 1314–1323.
  - [70] T. J. Blanche, M. A. Spacek, J. F. Hetke, and N. V. Swindale. “Polytrodes: High-Density Silicon Electrode Arrays for Large-Scale Multiunit Recording”. *Journal of Neurophysiology* 93.5 (May 2005), pp. 2987–3000.
  - [71] J. Du, I. H. Riedel-Kruse, J. C. Nawroth, M. L. Roukes, G. Laurent, and S. C. Masmanidis. “High-Resolution Three-Dimensional Extracellular Recording of Neuronal Activity with Microfabricated Electrode Arrays”. *Journal of Neurophysiology* 101.3 (Mar. 2009), pp. 1671–1678.
  - [72] D. Smetters, A. Majewska, and R. Yuste. “Detecting Action Potentials in Neuronal Populations with Calcium Imaging”. *Methods* 18.2 (1999), pp. 215–221.
  - [73] K. Kitamura, B. J. and Masanobu Kano, W. Denk, and M. Häusser. “Targeted patch-clamp recordings and single-cell electroporation of unlabeled neurons in vivo”. *Nature Methods* 5.1 (Jan. 2008), pp. 61–67.
  - [74] F. Helmchen, K. Svoboda, W. Denk, and D. W. Tank. “In vivo dendritic calcium dynamics in deep-layer cortical pyramidal neurons”. *Nature Neuroscience* 2.11 (Nov. 1999), pp. 989–996.
  - [75] K. Ohki, S. Chung, Y. H. Ch’ng, P. Kara, and R. C. Reid. “Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex”. *Nature* 433 (Feb. 2005), pp. 597–603.
  - [76] J. N. D. Kerr, D. Greenberg, and F. Helmchen. “Imaging input and output of neocortical networks in vivo”. *Proceedings of the National Academy of Sciences of the United States of America* 102.39 (Sept. 2005), pp. 14063–14068.
  - [77] J. N. D. Kerr, C. P. J. de Kock, D. S. Greenberg, R. M. Bruno, B. Sakmann, and F. Helmchen. “Spatial Organization of Neuronal Population Responses in Layer 2/3 of Rat Barrel Cortex”. *Journal of Neuroscience* 27.48 (Nov. 2007), pp. 13316–13328.

- [78] D. S. Greenberg, A. R. Houweling, and J. N. D. Kerr. “Population imaging of ongoing neuronal activity in the visual cortex of awake rats”. *Nature Neuroscience* 11.7 (July 2008), pp. 749–751.
- [79] S. R. Schultz, K. Kitamura, A. Post-Uiterweer, J. Krupic, and M. Häusser. “Spatial Pattern Coding of Sensory Information by Climbing Fiber-Evoked Calcium Signals in Networks of Neighboring Cerebellar Purkinje Cells”. *Journal of Neuroscience* 29.25 (June 2009), pp. 8005–8015.
- [80] I. Ozden, H. M. Lee, M. R. Sullivan, and S. S.-H. Wang. “Identification and Clustering of Event Patterns From In Vivo Multiphoton Optical Recordings of Neuronal Ensembles”. *Journal of Neurophysiology* 100.1 (July 2008), pp. 495–503.
- [81] T. Sasaki, N. Takahashi, N. Matsuki, and Y. Ikegaya. “Fast and Accurate Detection of Action Potentials From Somatic Calcium Fluctuations”. *Journal of Neurophysiology* 100.3 (Sept. 2008), pp. 1668–1676.
- [82] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynek, and L. Paninski. “Spike Inference from Calcium Imaging Using Sequential Monte Carlo Methods”. *Biophysical Journal* 97 (July 2009), pp. 636–655.
- [83] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski. “Fast Nonnegative Deconvolution for Spike Train Inference From Population Calcium Imaging”. *Journal of Neurophysiology* 104.6 (Dec. 2010), pp. 3691–3704.
- [84] P. Reinagel, D. Godwin, S. M. Sherman, and C. Koch. “Encoding of Visual Information by LGN Bursts”. *Journal of Neurophysiology* 81.5 (May 1999), pp. 2558–2569. URL: <http://jn.physiology.org/content/81/5/2558.abstract>.
- [85] F. Gabbiani, W. Metzner, R. Wessel, and C. Koch. “From stimulus encoding to feature extraction in weakly electric fish”. *Nature* 384.6609 (Dec. 1996), pp. 564–567.
- [86] X. Wang, Y. Wei, V. Vaingankar, Q. Wang, K. Koepsell, F. T. Sommer, and J. A. Hirsch. “Feedforward Excitation and Inhibition Evoke Dual Modes of Firing in the Cat’s Visual Thalamus during Naturalistic Viewing”. *Neuron* 55.3 (Aug. 2007), pp. 465–478.
- [87] T. A. Pologruto, B. L. Sabatini, and K. Svoboda. “ScanImage: Flexible software for operating laser scanning microscopes”. *BioMedical Engineering OnLine* 2.1 (May 2003), p. 13.
- [88] J. Reidl, J. Starke, D. B. Omer, A. Grinvald, and H. Spors. “Independent component analysis of high-resolution imaging data identifies distinct functional domains”. *NeuroImage* 34.1 (2007), pp. 94–108.
- [89] A. J. Sadosky, P. B. Kruskal, J. M. Kimmel, J. Ostmeier, F. B. Neubauer, and J. N. MacLean. “Heuristically optimal path scanning for high-speed multiphoton circuit imaging”. *Journal of Neurophysiology* 106.3 (Sept. 2011), pp. 1591–1598.
- [90] G. Katona, G. Szalay, P. Maák, A. K. M. Veress, D. Hillier, Balázs, E. S. Vizi, B. Roska, and B. Rósa. “Fast two-photon in vivo imaging with three-dimensional random-access scanning in large tissue volumes”. *Nature Methods* 9.2 (Feb. 2012), pp. 201–211.
- [91] D. L. Donoho and P. B. Stark. “Uncertainty Principles and Signal Recovery”. *SIAM Journal on Applied Mathematics* 49.3 (June 1989), pp. 906–931.

- 
- [92] S. G. Mallat and Z. Zhang. “Matching Pursuits with Time-Frequency Dictionaries”. *IEEE Transactions on Signal Processing* 41.12 (Dec. 1993), pp. 3397–3415.
  - [93] B. K. Natarajan. “Sparse Approximate Solutions to Linear Systems”. *SIAM Journal on Computing* 24.2 (Apr. 1995), pp. 227–234.
  - [94] S. S. Chen, D. L. Donoho, and M. A. Saunders. “Atomic Decomposition by Basis Pursuit”. *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 33–61.
  - [95] P. L. Dragotti and Y. M. Lu. “On Sparse Representation in Fourier and Local Bases”. *IEEE Transactions on Information Theory* 60.12 (Dec. 2014), pp. 7888–7899.
  - [96] G. Davis, S. Mallat, and M. Avellaneda. “Greedy Adaptive Approximations”. *Constructive Approximation* 13.1 (1997), pp. 57–98.
  - [97] S. Boyd and L. Vandenberghe. *Convex Optimization*. 1st ed. Cambridge University Press, Mar. 2004.
  - [98] Y. C. Pati, R. Rezaifar, and P. Krishnaprasad. “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition”. *27th Asilomar Conference on Signals, Systems and Computers*. Vol. 40-44. Nov. 1993.
  - [99] G. Davis, S. G. Mallat, and Z. Zhang. “Adaptive Time-frequency Decompositions”. *SPIE Journal on Optical Engineering* 33.7 (July 1994), pp. 2183–2191.
  - [100] J. A. Tropp. “Greed is Good: Algorithmic Results for Sparse Approximation”. *IEEE Transactions on Information Theory* 50.10 (Oct. 2004), pp. 2231–2242.
  - [101] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. 1st ed. Springer, Aug. 2010.
  - [102] D. L. Donoho and X. Huo. “Uncertainty Principles and Ideal Atomic Decomposition”. *IEEE Transactions on Information Theory* 47.7 (Nov. 2001), pp. 2845–2862.
  - [103] A. M. Bruckstein, D. L. Donoho, and M. Elad. “From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images”. *SIAM Review* 51.1 (2009), pp. 34–81.
  - [104] M. Elad and A. M. Bruckstein. “A Generalized Uncertainty Principle and Sparse Representation in Pairs of Bases”. *IEEE Transactions on Information Theory* 48.9 (Sept. 2002), pp. 2558–2567.
  - [105] A. Feuer and A. Nemirovski. “On Sparse Representation in Pairs of Bases”. *IEEE Transactions on Information Theory* 49.6 (June 2003), pp. 1579–1581.
  - [106] D. Donoho and J. Tanner. “Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing”. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 367.1906 (2009), pp. 4273–4293.
  - [107] D. L. Donoho and J. Tanner. “Precise Undersampling Theorems”. *Proceedings of the IEEE* 98.6 (June 2010), pp. 913–924.
  - [108] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová. “Statistical-Physics-Based Reconstruction in Compressed Sensing”. *Physical Review X* 2.2 (May 2012), p. 021005.

- [109] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp. “Living on the edge: phase transitions in convex programs with random data”. *Information and Inference* 3.3 (2014), pp. 224–294.
- [110] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. 3rd ed. Oxford University Press, 2001.
- [111] T. Richardson and R. Urbanke. *Modern Coding Theory*. 1st ed. Cambridge University Press, Mar. 2008.
- [112] M. Grant and S. Boyd. “CVX: Matlab software for disciplined convex programming, version 2.1”. <http://cvxr.com/cvx> (Sept. 2015).
- [113] W. Dai and O. Milenkovic. “Subspace Pursuit for Compressive Sensing Signal Reconstruction”. *IEEE Transactions on Information Theory* 55.5 (May 2009), pp. 2230–2249.
- [114] D. L. Donoho. “Compressed Sensing”. *IEEE Transactions on Information Theory* 52.4 (Apr. 2006), pp. 1289–1306.
- [115] E. J. Candès, J. Romberg, and T. Tao. “Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information”. *IEEE Transactions on Information Theory* 52.2 (Feb. 2006), pp. 489–509.
- [116] E. J. Candès and M. B. Wakin. “An Introduction To Compressive Sampling”. *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 21–30.
- [117] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, Aug. 2013.
- [118] Y. C. Eldar and G. Kutyniok. *Compressed Sensing: Theory and Applications*. 1st ed. Cambridge University Press, June 2012.
- [119] R. Tibshirani. “Regression Shrinkage and Selection via the Lasso”. *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. URL: <http://www.jstor.org/stable/2346178>.
- [120] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. “Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems”. *IEEE Journal of Selected Topics in Signal Processing* 1.4 (Dec. 2007), pp. 586–597.
- [121] J. A. Tropp and A. C. Gilbert. “Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit”. *IEEE Transactions on Information Theory* 53.12 (Dec. 2007), pp. 4655–4666.
- [122] D. Needell and J. A. Tropp. “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. *Applied and Computational Harmonic Analysis* 26.3 (May 2009), pp. 301–321.
- [123] T. Blumensath and M. E. Davies. “Iterative Hard Thresholding for Compressed Sensing”. *Applied and Computational Harmonic Analysis* 27.3 (Nov. 2009), 265–274.
- [124] T. Blumensath and M. E. Davies. “Normalized Iterative Hard Thresholding: Guaranteed Stability and Performance”. *IEEE Journal of Selected Topics in Signal Processing* 4.2 (Apr. 2010), pp. 298–309.

- 
- [125] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. “Clustering Large Graphs via the Singular Value Decomposition”. *Machine Learning* 56.1 (July 2004), pp. 9–33.
  - [126] N. Halko, P. G. Martinsson, and J. A. Tropp. “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions”. *SIAM Review* 53.2 (2011), pp. 217–288.
  - [127] R. A. Fisher and L. H. C. Tippett. “Limiting Forms of the Frequency Distribution of the Largest and Smallest Member of a Sample”. *Mathematical Proceedings of the Cambridge Philosophical Society* 24.2 (Apr. 1928), pp. 180–190.
  - [128] T. Blu and M. Unser. “Approximation error for quasi-interpolators and (multi-)wavelet expansions”. *Applied and Computational Harmonic Analysis* 6 (1999), pp. 219–251.
  - [129] C. R. Rao. “Information and the Accuracy Attainable in the Estimation of Statistical Parameters”. *Bulletin of the Calcutta Mathematical Society* 37.3 (1945), pp. 81–91.
  - [130] H. Cramér. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press, 1946.
  - [131] E. J. Gumbel. “Les valeurs extrêmes des distributions statistiques”. *Annales de l’Institut Henri Poincaré* 5.2 (1935), pp. 115–158.
  - [132] M. Fréchet. “Sur la loi de probabilité de l’écart maximum”. *Annales de la Société Polonaise de Mathématique* 6 (1927), pp. 93–116.