SPIKE SORTING AT SUB-NYQUIST RATES

Jose Caballero†, Jose Antonio Urigüen†, Simon R. Schultz* and Pier Luigi Dragotti†

†Department of Electrical and Electronic Engineering, Imperial College of London
*Department of Bioengineering, Imperial College of London
{ jose.caballero06, jose.uriguen08, s.schultz, p.dragotti } @imperial.ac.uk

ABSTRACT

Spike sorting relies on the ability to establish the temporal occurrence of action potentials and their relation to specific neurons. Neural information is intrinsically compressible and as such suitable for sparse sampling. Potentially, this should allow for the use of multi-channel recordings, which is particularly advantageous to improve spike sorting.

In this paper we propose a novel algorithm capable of sampling neural data at sub-Nyquist rates, yielding the same performance for spike sorting as traditional schemes.

Index Terms— FRI, spike sorting, sparse sampling

1. INTRODUCTION

Communication between neurons is carried out by action potentials (spikes) propagating as electrochemical impulses along the nervous system. Spike sorting is the ability to detect the temporal occurrence of action potentials and their relation to specific neurons, helping the analysis of brain activity.

Spike sorting has been shown to be successful at monitoring a limited number of neurons. However, substantial information in order to study how communication inside the brain occurs requires the problem to be scaled up. According to Shannon, since spikes typically contain frequencies up to 8KHz, sampling rates of at least 16KHz are normally required. This poses fundamental problems for simultaneous multichannel spike sorting in terms of energy consumption, computational complexity and hardware demands.

The activity of a neuron can be viewed as a temporal point process of identical spikes. Furthermore, the firing rate of neurons is by nature very low and action potentials can be shown to be approximately sparse in the wavelet domain. These conditions make neural information suitable to modern sampling techniques, such as finite rate of innovation (FRI) sampling [1] or compressed sensing (CS) [2], advocating for an economic acquisition of information.

In this paper we propose a novel algorithm capable of sampling and reconstructing neural data at sub-Nyquist rates,

preserving enough features of the original signal so that spike sorting is performed equally reliably.

The paper is organised as follows: in the next section we introduce the motivation for the paper. In Section 3 we explain the technical background required for the rest of the paper. In Section 4 we describe the modules of our proposed low-sampling-rate acquisition and reconstruction of neuronal activity signals. We then show the simulation results in Section 5 and conclude in Section 6.

2. SPIKE SORTING

A spike is the electrochemical action potential fired by a neuron for data transmission through the nervous system and their distinctive signature shapes largely depend on their morphology and the recording process. The objective of spike sorting algorithms is to detect action potentials and identify which neuron generated them. The vast majority follow three basic steps. They begin with a spike detection stage, mainly achieved by voltage thresholding with respect to an estimation of the noise amplitude in the signal. Then, a feature extraction step characterises detected spikes, the main property looked for among these features being that they present a multimodal distribution that ideally allows to separate spikes fired by different neurons. Principal Component Analysis (PCA) and wavelet decomposition have widely been used in the literature for feature extraction [3–6]. To end, and based on these features, a clustering step is necessary to relate each spike to a particular neuron.

Existing algorithms suffer from scalability issues due to high sampling rates. Neural activity from one neuron has been shown to be compressible [7, 8], and as such it is suitable for sparse sampling. Reducing the sampling frequency would imply that large simultaneous extracellular recordings could be obtained and the additional reliability provided by multi-channel recordings would be exploited in the sorting. Improvement in sampling techniques is also crucial to scaling up the recording technologies to hundreds or thousands of neurons, i.e. to the population sizes really necessary to understand brain function and to control the next generation of neuroprostheses.

3. TECHNICAL BACKGROUND

3.1. Sampling signals with FRI

In [1] Vetterli et al. demonstrated that certain classes of signals that are characterised by a finite number of degrees of freedom (signals with finite rate of innovation) can be sampled and perfectly reconstructed using appropriate sampling schemes. In this section we review the basic scenario, following the development as treated in [9].

Assume that $x(t)$ is a stream of K Diracs with amplitudes a_k located at instants of time t_k smaller than τ seconds, i.e. $x(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k)$. Then, the measurements obtained sampling the signal with the kernel $\varphi\left(-\frac{t}{T}\right)$, for $n = 0, 1, \ldots, N - 1$, are

$$y_n = \left\langle x(t), \varphi\left(\frac{t}{T} - n\right)\right\rangle = \sum_{k=0}^{K-1} a_k \varphi\left(\frac{t_k}{T} - n\right), \quad (1)$$

where we assume the sampling period is $T = \frac{\tau}{N}$.

In this paper we concentrate on a specific class of kernels, used in [9], that are able to reproduce exponentials. An exponential reproducing kernel is any function $\varphi(t)$ that satisfies

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t - n) = e^{\alpha_m t} \quad \text{with } \alpha_m \in \mathbb{C}, \quad (2)$$

for a proper choice of coefficients $c_{m,n}$ (see [9]).

Exponential splines (E-Splines) [10] are central to the exponential reproduction property. An E-Spline $\beta_{\alpha_m}(t)$ of order P and parameters α_m is able to reproduce the exponentials $e^{\alpha_m t}$, $m = 0, \ldots, P$; and so is any composite function of the form $\gamma(t) * \beta_{\alpha_m}(t)$ [10].

In the reconstruction scheme of [9] the samples y_n are first combined linearly with the coefficients $c_{m,n}$ to obtain

$$s_m = \sum_{n=0}^{N-1} c_{m,n} y_n = \hat{a}_k u_k^m, \quad (3)$$

for $m = 0, \ldots, P$, and where $\hat{a}_k = a_k e^{\alpha_0 \frac{t_k}{T}}$ and $u_k = e^{\lambda \frac{t_k}{T}}$. This follows from the fact that the original signal is a stream of Diracs and we use $\alpha_m = \alpha_0 + m\lambda$. The values s_m are exponential moments of the continuous-time signal $x(t)$.

Then, the new pairs of unknowns $\{\hat{a}_k, u_k\}_{k=0}^{K-1}$ can be retrieved from the power series in (3) using the classical Prony's method. The key ingredient is the annihilating filter h_m, which has the property that the roots of its z-transform $\hat{h}(z)$ correspond precisely to the locations u_k. The filter annihilates the series s_m [9], which can be written in matrix-vector form as

$$\mathbf{Sh} = \mathbf{0}. \quad (4)$$

By solving the above system, we find h_m and then retrieve u_k. Finally, we obtain the weights a_k by solving K consecutive equations in (3) (note that $P \geqslant 2K - 1$).

We thus conclude that perfect reconstruction of a stream of K Diracs is possible with any kernel able to reproduce exponentials: $\varphi(t) = \gamma(t) * \beta_{\alpha_m}(t)$.

3.2. Modified E-Spline kernels

When noise is present in the acquisition process we do not have access to the ideal measurements. In contrast, we get the samples $\hat{y}_n = y_n + \epsilon_n$, which translate into

$$\hat{s}_m = s_m + \sum_{n=0}^{N-1} c_{m,n} \epsilon_n \quad m = 0, \ldots, P. \quad (5)$$

One way to control how noise affects the measurements is by designing a specific kernel that can act on (5) through appropriate coefficients $c_{m,n}$. In [11] it was shown how to modify E-Splines in order for the properties of the noise ϵ_n to be preserved in their contribution to the moments s_m.

In this paper we make use of the modified E-Spline kernels proposed in [11] to sample the neural signals. We choose even orders P, for which the parameters α_m are purely imaginary and equal to $j\omega_m = j\frac{\pi}{N}(2m - P)$, $m = 0, \ldots, P$.

4. DESIGN OF THE ALGORITHM

4.1. Modelling the neural signal

The activity of a neuron can be interpreted as a point process in which roughly the same spike is fired at different instants of time. Assuming stationary neurons and no bursting exists, the amplitudes of the spikes are constant for the same neuron. Thus, the neural signal to be sampled from neuron j can be thought of as the result of the convolution of its spike shape with a train of Diracs, i.e. $x_j(t) = p_j(t) * \sum_{k=0}^{K-1} a_j \delta(t - t_{j,k})$, where $p_j(t)$ is the spike, a_j is the constant amplitude and $t_{j,k}$ are the firing instants. If we consider the contribution of J different neurons, which add up linearly, we can write the complete model for the neural signal as $x(t) = \sum_{j=0}^{J-1} x_j(t) + n(t)$, where $n(t)$ accounts for any source of noise.

For the sake of clarity we assume that contributions do not overlap. In such a case, we can sequentially retrieve one pulse after the other and simply write that $x(t) = p(t) * d(t)$, where $p(t)$ is the pulse shape of the neuron, and $d(t)$ is one Dirac that represents the amplitude and location of the spike.

4.2. How to extract the information on $d(t)$ given $p(t)$

The signal $x(t)$ is acquired using an exponential reproducing kernel as discussed in Section 3.1 leading to the samples y_n as in (1). The Dirac signal $d(t)$ can be retrieved from prior knowledge on the pulse shape $p(t)$ and the samples y_n. If $p(t)$ is known exactly, then we can write

$$y_n = \left\langle x(t), \varphi\left(\frac{t}{T} - n\right)\right\rangle = \left\langle p(t) * d(t), \varphi\left(\frac{t}{T} - n\right)\right\rangle$$

$$= \left\langle d(t), \varphi_p\left(\frac{t}{T} - n\right)\right\rangle, \quad (6)$$

where now we have an equivalent exponential reproducing kernel $\varphi_p(t) = \varphi(t) * p^*(-t)$.

Then the locations and amplitudes that characterise $d(t)$ can be retrieved following the annihilating filter procedure explained in Section 3.1 but using the new set of moments $s_m^p = \sum_n c_{m,n}^p y_n$, where $c_{m,n}^p$ are the coefficients so that $\varphi_p(t)$ satisfies the exponential reproducing formula (2).

4.3. How to extract the information on $p(t)$ given $d(t)$

It is also possible to estimate the pulse shape with prior knowledge on $d(t)$ from the samples y_n. Consider the exponential moment s_m, for which the following holds

$$s_m = \sum_{n=0}^{N-1} c_{m,n} y_n = \sum_{n=0}^{N-1} c_{m,n} \left\langle x(t), \varphi\left(\frac{t}{T} - n\right)\right\rangle$$
$$= \left\langle x(t), e^{\alpha_m t}\right\rangle = \int_{-\infty}^{\infty} x(t) e^{\alpha_m t} \mathrm{d}t. \qquad (7)$$

Thanks to our choice of parameters α_m, (7) is precisely the Fourier transform (FT) of $x(t)$ at $\omega = \omega_m$. Therefore, the moments s_m satisfy

$$s_m = \hat{x}(\omega_m) = \hat{p}(\omega_m)\hat{d}(\omega_m), \qquad (8)$$

where $\hat{x}(\omega)$ is the FT of $x(t)$ and $\omega_m = \frac{\pi}{N}(2m - P)$, $m = 0, \dots, P$. If $d(t)$ is known, it is straightforward to obtain the FT of $p(t)$ at ω_m, $m = 0, \dots, P$ from the moments s_m. We then retrieve the pulse shape via an ℓ_1 minimisation as explained next.

4.4. Spike shape recovery

The aim of this processing block is to reconstruct a finely discretised version $\tilde{\mathbf{p}}$ of $p(t)$ from $\hat{p}(\omega_m)$, $m = 0, \dots, P$.

Assume $\tilde{\mathbf{p}}$ is the pulse shape discretised to a vector of length L ($L \gg P$) and denote with $\hat{\tilde{\mathbf{p}}}$ the approximated discrete Fourier transform (DFT) of $\tilde{\mathbf{p}}$ obtained from (8). We can write:

$$\hat{\tilde{\mathbf{p}}} = \mathbf{F}\tilde{\mathbf{p}} + \mathbf{n} \qquad (9)$$

where $\mathbf{F}$ is the DFT matrix of size $(P+1) \times L$, $\tilde{\mathbf{p}}$ is known and is obtained from (8), and $\mathbf{n}$ is additive noise used to approximate any model mismatch. We know that neuronal pulses can be sparsely described in the wavelet domain. We therefore rewrite (9) as

$$\hat{\tilde{\mathbf{p}}} = \mathbf{F}\mathbf{W}^{-1}\mathbf{w} + \mathbf{n} = \mathbf{A}\mathbf{w} + \mathbf{n}, \qquad (10)$$

where $\mathbf{W}$ is the $L \times L$ matrix representing the wavelet transform and $\mathbf{w} = \mathbf{W}\tilde{\mathbf{p}}$ is the wavelet representation of $\tilde{\mathbf{p}}$.

The above system is underdetermined but we only need to search for a sparse vector $\mathbf{w}$ that satisfies (10). This modelling is reminiscent of the traditional CS framework where, in our context, the acquisition matrix is a 'fat' Fourier matrix rather than a more conventionally used random matrix. We therefore assume a sufficiently large P and a sufficiently sparse vector $\mathbf{w}$ in order to solve for $\tilde{\mathbf{p}}$ using an ℓ_1 minimisation technique such as Basis Pursuit (BP).

4.5. Complete algorithm

In the previous sections it has been shown how N samples are enough to recover $x(t)$ by breaking down the problem into estimating $d(t)$ and $p(t)$ separately. An iterative algorithm can thus be applied to retrieve $x(t)$ without any prior knowledge.

Assume we initialise the algorithm by setting $p(t) = \delta(t)$, meaning that at the first iteration the kernel $\varphi_p(t)$ coincides with $\varphi(t)$. The $d(t)$ estimation module will look for the location and amplitude of the Dirac, although the signal is actually a spike. The first estimation of $d(t)$ will therefore be inaccurate, but it is enough to obtain a good estimation of $p(t)$ using the recovery technique of Sections 4.3 and 4.4. Once there is useful information of $p(t)$, it can be used to update $\varphi_p(t)$ and the new set of coefficients $c_{m,n}^p$ to compute s_m^p. Using the updated moments the process can be repeated again, the convergence criterion being a maximum number of iterations or that a solution within a predefined tolerance is reached.

5. RESULTS

The algorithm has proven to converge experimentally to the sought pulse shape at the desired location in about 5 iterations for a mean square error (MSE) convergence tolerance of 10^{-5}. The estimation of $p(t)$ is however suboptimal from the point of view of sparsity in the wavelet domain. The reason for this is probably that the algorithm is able to find a solution out of various stable regions in the solution space. One example of the estimation of $x(t)$ can be seen in Fig. 1.
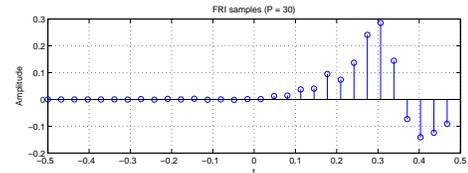
(a) Samples obtained using the FRI sampling scheme
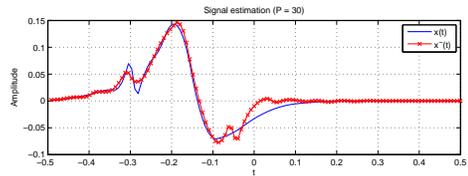
(b) Original pulse $x(t)$ and estimated version $\hat{x}(t)$

Fig. 1. Reconstruction of signal $x(t)$ using the proposed sampling algorithm.

The order P of the E-Spline is a relevant design choice that influences the performance of the algorithm. Low values provide better estimation accuracy but high values are able to capture high resolution details of the spike shapes. We choose the latter because fine details are relevant for sorting.

Finally, we have assessed the suitability of the sub-Nyquist sampling algorithm for spike sorting, working with

Spike set	Noise s.d.	Missed spikes		False positives		Misclassified spikes		Unclassified spikes		Success Rate	
		24K C	5.8K F	24K C	5.8K F	24K C	5.8K F	24K C	5.8K F	24K C	5.8K F
Easy (1)	0.05	111	135	0	2	22	21	30	20	83.7	82.2
	0.1	93	91	6	9	29	34	9	4	86.3	86.2
	0.15	143	129	7	21	50	56	1	2	79.9	79.2
	0.2	248	216	1	18	37	44	1	2	71.3	72
Difficult (2)	0.05	140	149	0	0	17	7	70	71	77.3	77.3
	0.1	101	80	0	16	418	199	0	16	48.1	69.9
	0.15	115	86	1	20	346	454	0	0	53.8	44
	0.2	160	108	3	19	441	420	0	0	39.6	45.3
(Av.)	0.125	138.88	124.25	2.24	13.13	170	154.38	13.88	14.38	67.5	69.51

Table 1. *Spike sorting comparison for datasets acquired at different rates*. C and F stand for classical and FRI sampling respectively. We measure (i) undetected spikes, (ii) noise detected as spikes, (iii) spikes in the wrong cluster, and (iv) spikes that cannot be identified.

surrogate data available from the NeuroEngineering Lab at the University of Leicester. Spikes are simulated using a database of 594 different average action potentials recorded in the neocortex and basal ganglia. Three distinct spike shapes are placed at arbitrary times with normalised peak amplitude of 1 and background noise is generated with a standard deviation relative to 1 from superimposed spikes selected at random. Difficulties for sorting mainly come from similarities among spike shapes, realistic background noise and overlapping spikes susceptible of generating errors.

We compare the performance achieved feeding the original data (@24KHz) and an FRI subsampled version to the spike sorting algorithm "Wave_Clus" [6], estimating a total of 1000 action potentials. We use $N = 31$ FRI samples to represent pulses of length $L = 128$, and an E-Spline of order $P = 30$, to achieve a sampling rate reduction by a factor 4. The reconstruction of individual spikes assumes that their location is known a priori. The results are presented in Table 1, and they show that our method is able to preserve the performance achieved by traditional spike sorting algorithms. We have noticed that there is a decrease in missed spikes and an increase in false positives. We believe this is due to the detection threshold value, chosen proportional to the median of the absolute value of the recording [6], which is lowered since the reconstruction process slightly smoothes spikes out.

6. CONCLUSIONS

In this paper, we have proposed an iterative reconstruction algorithm that can estimate a neural signal from FRI samples that have been obtained using a modified E-Spline at reduced sampling rates. The design of the algorithm is motivated by the sparse representation of the neural activity signal. Our main contribution is that we show that state-of-the-art spike sorting performances can be reached with a reduction in the sampling rate of a factor 4 compared to traditional methods. Future work will analyse MSE spike shape reconstruction results and compare the proposed method with respect to existing undersampling spike acquisition techniques.

7. REFERENCES

[1] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1417–1428, 2002.

[2] E. J. Candès and T. Tao, "Robust Uncertainty Principle: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Theory*, vol. 52, pp. 489–509, February 2006.

[3] D. A. Adamos, E. K. Kosmidis, and G. Theophilidis, "Performance evaluation of PCA-based spike sorting algorithms," *Computer methods and programs in Biomedicine*, vol. 91, no. 3, pp. 232–244, 2008.

[4] J. C. Letelier and P. P. Weber, "Spike sorting based on discrete wavelet transform coefficients," *Journal of neuroscience methods*, vol. 101, no. 2, pp. 93–106, 2000.

[5] E. Hulata, R. Segev, and E. Ben-Jacob, "A method for spike sorting and detection based on wavelet packets and Shannon's mutual information," *Journal of neuroscience methods*, vol. 117, no. 1, pp. 1–12, 2004.

[6] R. Q. Quiroga, Z. Nadasdy, and Y. B. Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comp.*, vol. 16, pp. 1661–1687, 2004.

[7] Z. Charbiwala, V. Karkare, S. Gibson, D. Markovic, and M. B. Srivastava, "CS of Neural APs Using a Learned Union of Supports," *International Workshop on WIBSN*, pp. 53–58, 2011.

[8] S. Lakshminarayan, L. R. Varshney, and J. Kusuma, "Acquisition of Action Potentials with Ultra-Low Sampling Rates," in *Annual International Conference of the IEEE EMBS*, pp. 4213–4216, 2010.

[9] P. L. Dragotti, M. Vetterli, and T. Blu, "Sampling Moments and Reconstructing Signals of Finite Rate of Innovation: Shannon Meets Strang-Fix," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1741–1757, 2007.

[10] M. Unser and T. Blu, "Cardinal Exponential Splines: Part I—Theory and Filtering Algorithms," *IEEE Transactions on Signal Processing*, vol. 53, pp. 1425–1438, April 2005.

[11] J. A. Urigüen, P. L. Dragotti, and T. Blu, "On The Exponential Reproducing Kernels for Sampling Signals with Finite Rate of Innovation," in *Proceedings of the 9th International Workshop on SampTA*, May 2–6, 2011.