34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Low-cost intracortical spiking recordings compression with classification abilities for implanted BMI devices

Bertrand Coppa[1], Rodolphe Héliot[1], Olivier Michel[2], Eric Moisan[2] and Dominique David[1]

***Abstract*— Within Brain-Machine Interface systems, cortically implanted microelectrode arrays and associated hardware have a low power budget for data sampling, processing and transmission. It is already possible to reduce neural data rates by on-site spike detection; we propose a method to further compress spiking data at a low computational cost, with the objective of maintaining clustering and classification abilities. The method relies on random binary vector projections, and simulations show that it is possible to achieve a compression ratio of 5 at virtually no cost in terms of classification errors.**

***Index Terms*— Random embeddings; Compressive Sensing; Neural signal clustering**

I. Introduction

Brain-Machine Interfaces (BMI) aim at establishing a direct communication pathway between the brain and an external actuator, such as computer cursor or a robotic device [1–3]. Potential applications include the restoration of sensorimotor functions for patients suffering from spinal cord injuries, stroke, and other neurological disorders [4]. To this aim, a BMI system generally embeds four components: a recording device capturing neurophysiological signals from the brain, a decoding algorithm converting these signals into a variable representing an action to be performed, an actuator, and a feedback provided to the user. Cortically implanted microelectrode arrays allow to collect spiking data from neural ensembles, thus allowing to control external devices with great accuracy. Single unit and multi-unit activity are recorded on each electrode, meaning that the activity of each recording site must be sorted in real-time to be used in a BMI paradigm [5]. In the end, units with a clearly distinguishable waveform are isolated and clustered.

Intracortical neural activity is typically recorded using sampling frequencies between 30 kHz and 50 kHz. Since the number of channels can be high as well (128 or more electrodes), this generates large amounts of data to be processed in real time. For embedded or implanted systems, transmitting this data in real time would require a huge wireless bandwidth, in the order of hundreds of Mbits/sec. Thus, data must be compressed before transmission; however, due to very low power budget, there is a need for a low computational cost compression technique and associated hardware. Spike detection addresses the problem of data reduction: there are algorithms that allow real-time adaptive discrimination threshold and spike detection [6]. Yet, more compression is possible at a low computational cost: *Compressive Sensing* (CS) has been known to allow simple data compression at the cost of expensive decompression [7–10] under the hypothesis that there exists a (known) sparse representation of the data.

The principle of CS is to project data on few vectors that collect the information. Random vectors usually work well for any kind of signal [11]. This projection on random vectors is similar to dimensionality reduction techniques proposed by [12–15], where the final objective is clustering. Indeed, as stated earlier, spikes clustering is generally performed after data acquisition within a BMI setup, in order to determine which neuron fired the detected spike. We propose here a simple compression method, inspired by CS, using only binary operations within the embedded or implanted system, to compress the detected spikes. Section II describes the experimental dataset and the compression method; section III presents classification results, followed by a discussion.

II. Material and methods

A. Dataset

In this paper, we used simulated neural data made publicly available by the authors of [16]. The background noise is simulated using spikes shapes from a database of around 600 recorded and averaged spike shapes, set at random times and amplitudes. A train of data from 3 spike shapes is then superimposed with normalized amplitude. The noise variance is scaled to various values relative to the normalized amplitude of the spike trains. The data are simulated at 96 kHz and interpolated so that spikes are set continuously (to machine precision) in time. This allows to generate spikes at arbitrary starting time (not necessarily matching a sample time). It was then down-sampled to 24 kHz to imitate actual recording conditions. More information is available in [16].

The data is available in the form of a 10-seconds-long simulated signal containing 507 spikes, of which figure 1 show an excerpt with highlighted spikes. The location and cluster class of the spikes in the train of data are available. The data also provides synchronized spike traces as shown on figure 2.

A *Principal Component Analysis* (PCA) of the spike traces allows to distinguish three well-separated clusters (fig. 3) on the first two components of the PC, providing a basis for classification purposes. The k-means algorithm is often used for classification (see section II-B.2).

[1] B. Coppa, R. Héliot and D. David are with CEA-LETI, Minatec Campus, Grenoble, France. {bertrand.coppa, rodolphe.heliot, dominique.david} at cea.fr

[2] O. Michel and E. Moisan are with GIPSA-Lab, University of Grenoble, France. {olivier.michel, eric.moisan} at grenoble-inp.fr

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

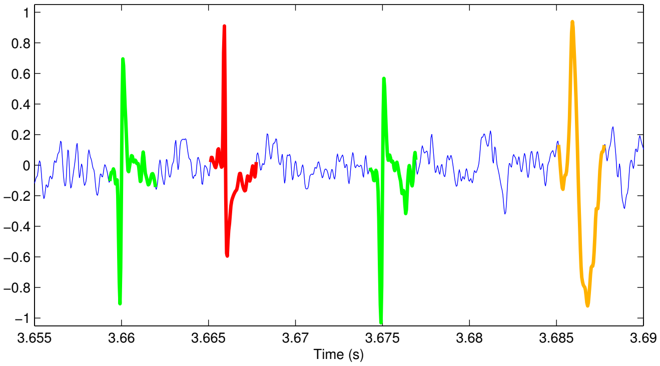

Fig. 1. Excerpt of simulated raw data, sampled at 24 kHz, where the data spikes are shown in bold and colored as per cluster class.

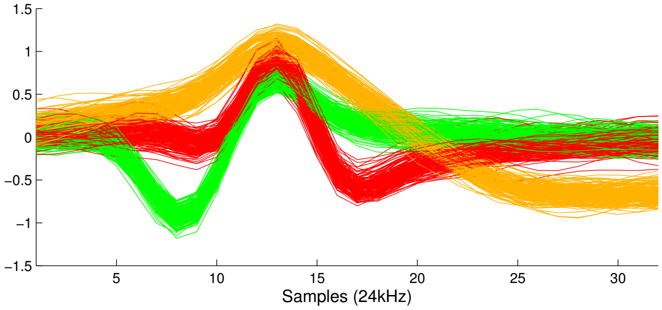

Fig. 2. Synchronized and superimposed spikes, with color scheme corresponding to attributed cluster class.

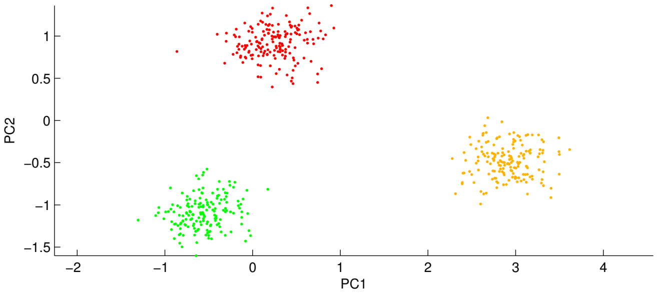

Fig. 3. Projection on first two principal components of the spikes, with color corresponding to the attributed cluster class.

B. Methods

1) Compression: The principle of CS is simple, as illustrated by figure 4: given a signal $x \in \mathbb{R}^N$ and a $m \times N$, $m < N$ matrix Φ, the compressed signal is Φx.

In our particular situation, the signal is a digitally sampled, quantified version of the real signal. The matrix Φ is a binary *Plus Minus One* matrix, built randomly with *iid* matrix elements. The probability for each state (± 1) is 0.5. This kind of matrix is statistically orthogonal, and even for the small dimensions considered in this paper, the rows are close to orthogonal. Using such a binary *Plus Minus One* matrix and a binary encoded digital signal (the most frequent case of encoding) is highly efficient from a computational point of view, since the projection on the matrix is reduced to

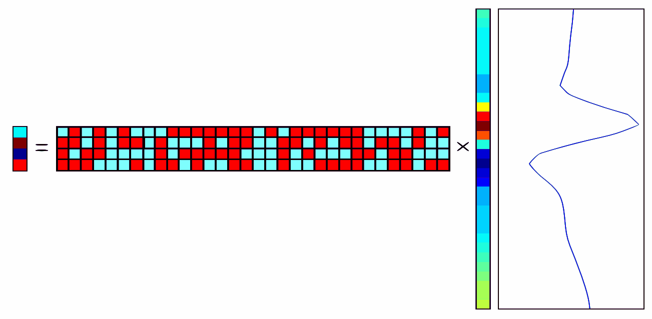

Fig. 4. Illustration of CS principle. The compressed data on the left is obtained by a projection on a random binary matrix (center) of the signal (right), here with dimensions $m = 4$ and $N = 32$.

simple binary operations: sign switch and addition. As a consequence, it makes compressive sensing very attractive for low-power, embedded data compression systems.

2) Classification: Classification is performed using the k-means algorithm [17]. This is an iterative algorithm that attributes each point to the closest (here, in Euclidean distance) cluster center, then updates the cluster centers by setting each to the mean of all points in the cluster, until there is no more change in the clusters. To automatically initialize the cluster centers, we used a method inspired by [18]. The principle is to build a Minimum Spanning Tree (MST): the tree is initialized to a random point of the set, then sequentially built by adding the closest (in Euclidean distance) point to any point already belonging to the tree. The sequence of distances between each added point and the tree is kept in memory, and the clusters are chosen by setting a threshold on those distances. The centers of those clusters are used to initialize the k-means algorithm.

III. Results

To evaluate the proposed method, we performed simulations on the dataset described in II-A. The experiment was repeated for multiple m values, each time using 1000 randomly-drawn binary $m \times N$ matrices.

A. Clustering Initialization

To initialize the cluster centers, the MST threshold was set *a posteriori* to the mean plus one standard deviation of the distances in the tree, and the minimal cluster size was set to $\#S/6$, where $\#S$ is the number of spikes in the dataset. This means that a maximum of 6 different clusters were allowed, which is coherent with physiological recordings where a maximum number of 4 units is generally observed on a given channel. Figure 5 shows the distances and threshold used to identify clusters, and figure 6 show the corresponding tree displayed on the principal component projection of the compressed spikes.

B. Classification results

Figure 7 shows the average proportion of misclassified spikes (over 1000 realisations) as a function of the number

Number of projections m Compression ratio	4 8	5 6.4	6 5.33	7 4.57	8 4	9 3.56	10 3.2	11 2.91	12 2.67
Average proportion of misclassified spikes	5.09%	2.38%	0.81%	0.48%	0.25%	0.07%	0.03%	0.02%	0.02%
Less than 0.5% misclassified	69.4%	83.1%	91.8%	95.2%	97.6%	98%	99%	99.6%	99.6%
Average number of clusters	2.855	2.934	2.979	2.988	2.994	2.999	3	3	3
Probability of having less than 3 clusters	14.3%	6.6%	2.1%	1.2%	0.6%	0.1%	0%	0%	0%

TABLE I

CLASSIFICATION PERFORMANCE IN FUNCTION OF THE NUMBER OF PROJECTIONS m, $N = 32$.

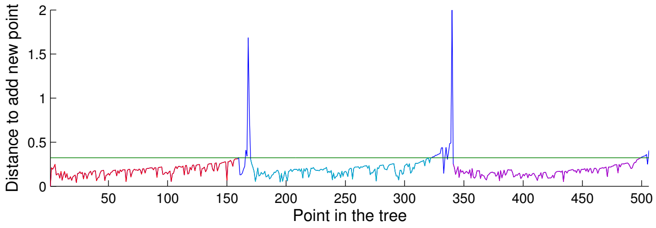

Fig. 5. Distances to add a new point in the minimum spanning tree, with a threshold line, and colored groups corresponding to identified clusters.

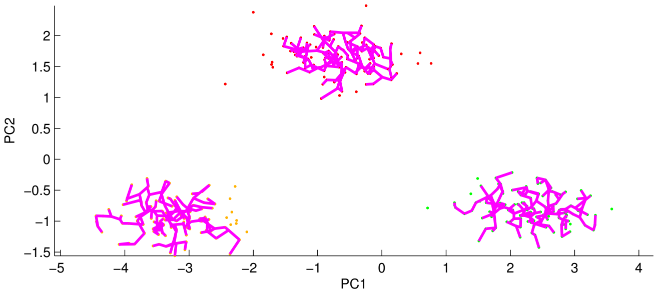

Fig. 6. Minimum spanning tree for the identified clusters.

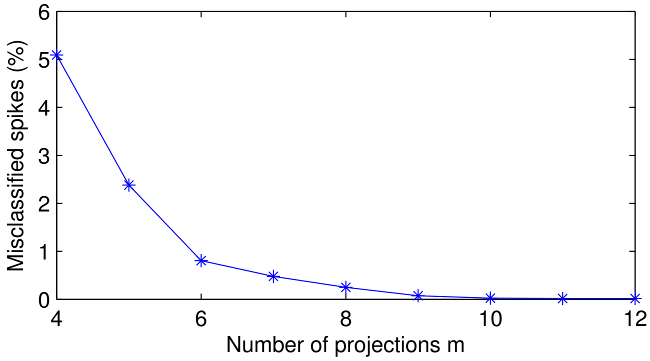

Fig. 7. Average percentage of misclassified spikes as a function of the number of projections m, with $N = 32$.

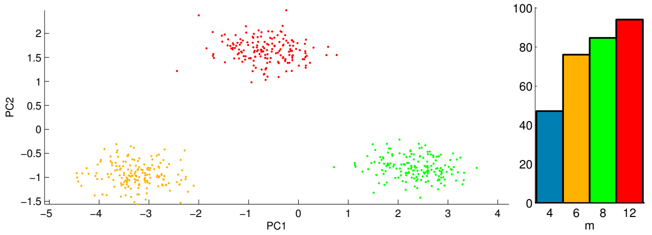

Fig. 8. Projection on first two principal components of the compressed ($m = 4$, $N = 32$) spikes, no error in clustering; and proportion of realisations that do as good as that for $m = 4$, 6, 8 or 12.

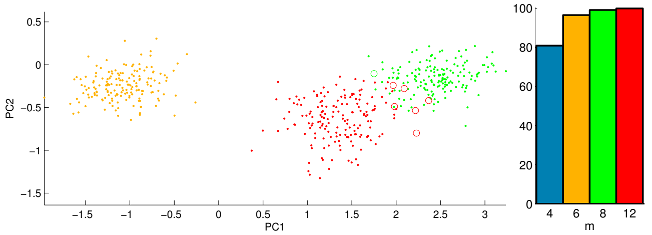

Fig. 9. Projection on first two principal components of the compressed ($m = 4$, $N = 32$) spikes, 7 error in clusterings; and proportion of realisations that do as good as (or better than) that for $m = 4$, 6, 8 or 12.

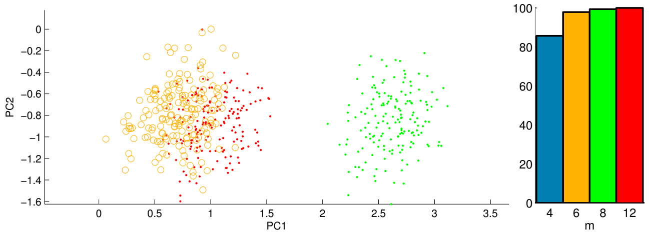

Fig. 10. Projection on first two principal components of the compressed ($m = 4$, $N = 32$) spikes, missing cluster; and proportion of realisations that do better than that, for $m = 4$, 6, 8 or 12.

of projections, from 4 to 12 (i.e. compression ratio from 8 to 2.67). In our example, there are 507 spikes, so 0.02% corresponds approximately to a single misclassified spike. The results indicates that for $m \geq 6$ (compression ratio of approximately 5.33), there is in average less than 5 misclassified spikes. For $m \geq 9$, there is much less than one misclassified spike.

Figure 8 shows an example of a good result for a high compression ratio (8): the projection on the principal components shows the tree clusters but they are a little less distinctly separated and a little more scattered compared to figure 3. Note that PCA is only used for visualization convenience, and that classification is always performed on the full compressed data and not only the two principal components. This is the result in most of the cases, as shown by table I, line 2, even when $m = 4$. However, for such compression ratio, it happens about 3 out of 10 times that there are errors between two clusters, as shown on figure 9. In most cases (85% of the realisations with errors for

$m = 4$ and up to 100% for $m > 10$), misclassification happens between only two clusters. In the worst cases, two clusters are mixed as seen on figure 10: this leads to a high error rate as a full cluster is wrongly classified and each cluster has approximately one third of the population. This happens 14.1% of the time in the case of $m = 4$ and only in this case does it happen that there was only one cluster (0.2% of occurrence). With larger m, this happens less and less often and for $m \geq 10$ (compression ratio 3.2), it does not happen any more. When it happens, in most cases (over 83%), only two clusters are affected but they are still well-separated from the third. As shown by the right bar-graphs (fig. 8-10), the perfect result already occurs a lot (except for the case $m = 4$), but in the majority of cases, the result is as good or better than the situation on figure 9.

Table I sums up the classification results. The first line indicates the proportion of misclassified spikes over the 1000 realisations, as shown in figure 7, while the second line shows the proportion of realisations where the number of misclassified spikes is lower than 0.5%, that is, at most 2 errors. Starting from $m = 6$, it happens with probability higher than 90%, and the average error rate is less than 1%. The two last lines deal with the number of clusters, and show that while it is an occasional issue for high compression ratios, when $m \geq 10$, there is no cluster detection error.

C. Discussion

The results show that the CS-based method has good performance for compressing data while conserving clustering properties of data. Furthermore, the computational cost of the compression is only $m \times N$ additions. This means that the higher the compression ratio, the lower the cost, although this comes at the expense of clustering quality. Choosing $m = 6$ (i.e. a compression ratio of $32/6 \approx 5.33$) seems like a good trade-off between clustering performance and compression cost. Further clustering improvement induced by adding more projections (i.e. $m > 6$) is less significant than improvements from $m = 4$ to $m = 6$. We did not attempt in this study to fully reconstruct the signal, instead focusing on clustering capabilities that are of importance in real-time BMI systems. However, with the projection matrix dimensions used here (6×32), basic simulations using IRLS [9] as reconstruction algorithm indicates that it would be difficult to have a near-perfect reconstruction. One solution would be to reduce the compression ratio (from 5.33 to 3, for example), or to seek a representation basis that would require only one (or few) significant coefficient to represent a spike.

IV. Conclusion

We proposed in this paper a system to compress intracortical spiking signals at a very low computational cost. Rather than transmitting the full recorded information, compressed data is transmitted that allows to perform spike classification without the need for reconstructing the original signal. The system is based on random binary matrix projections, which require sign switch and addition only, making the method very easy to implement. The simulations showed the classification results remain very good even when using compressed data: a compression ratio of 5.33 allows spikes classification with little to no error. Future work will include the design of dedicated digital hardware; we plan to design an integrated circuit implementing the described method in CMOS 65nm technology, in order to precisely evaluate the area and power consumption of such a system.

References

[1] F Galan, M Nuttin, et al. "A brain-actuated wheelchair : Asynchronous and non-invasive Brain computer interfaces for continuous control of robots". In: *Clinical Neurophysiology* 119 (2008), pp. 2159–2169. DOI: 10.1016/j.clinph.2008.06.001.

[2] G Schalk, K J Miller, et al. "Two-dimensional movement control using electrocorticographic signals in humans". In: *Journal of neural Engineering* 5 (2008), pp. 75–84. DOI: 10.1088/1741-2560/5/1/008.

[3] Meel Velliste, Sagi Perel, et al. "Cortical control of a prosthetic arm for self-feeding". In: *Nature* 453.June (2008), pp. 1098–1101. DOI: 10.1038/nature06996.

[4] Leigh R Hochberg, Mijail D Serruya, et al. "Neuronal ensemble control of prosthetic devices by a human with tetraplegia". In: *Nature* 442.July (2006), pp. 164–171. DOI: 10.1038/nature04970.

[5] MS Lewicki. "A review of methods for spike sorting: the detection and classification of neural action potentials". In: *Network: Computation in Neural Systems* 9.4 (1998), R53–R78.

[6] J.F. Beche, S. Bonnet, et al. "Real-time adaptive discrimination threshold estimation for embedded neural signals detection". In: *Neural Engineering, 2009. NER'09. 4th International IEEE/EMBS Conference on*. IEEE. 2009, pp. 597–600.

[7] DL Donoho. "Compressed sensing". In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306.

[8] E. Candes and J. Romberg. "l1-magic: Recovery of sparse signals via convex programming". In: *California Institute of Technology, Tech. Rep* (2005).

[9] R. Chartrand and W. Yin. "Iteratively reweighted algorithms for compressive sensing". In: *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*. 2008, pp. 3869–3872.

[10] J.A. Tropp and A.C. Gilbert. "Signal recovery from random measurements via orthogonal matching pursuit". In: *IEEE Transactions on Information Theory* 53.12 (2007), p. 4655.

[11] EJ Candes and T. Tao. "Near-optimal signal recovery from random projections: Universal encoding strategies?" In: *IEEE Transactions on Information Theory* 52.12 (2006), pp. 5406–5425.

[12] S. Dasgupta. "Learning mixtures of Gaussians". In: *Foundations of Computer Science, 1999. 40th Annual Symposium on*. 1999, pp. 634 –644. DOI: 10.1109/SFFCS.1999.814639.

[13] S. Dasgupta. "Experiments with random projection". In: *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*. 2000, pp. 143–151.

[14] X.Z. Fern and C.E. Brodley. "Random projection for high dimensional data clustering: A cluster ensemble approach". In: *Proceedings of 20th International Conference on Machine learning*. 2003.

[15] A. Bertoni and G. Valentini. "Ensembles Based on Random Projections to Improve the Accuracy of Clustering Algorithms". In: *Neural nets 2005* 3931 (2006), p. 31.

[16] R. Quian Quiroga, Z. Nadasdy, and Y. Ben-Shaul. "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering". In: *Neural Computation* 16.8 (2004), pp. 1661–1667.

[17] S. Lloyd. "Least squares quantization in PCM". In: *Information Theory, IEEE Transactions on* 28.2 (1982), pp. 129 –137. ISSN: 0018-9448. DOI: 10.1109/TIT.1982.1056489.

[18] L. Galluccio, O.J.J. Michel, et al. "Graph Based k-Means Clustering". In: *Elsevier Signal Processing* (2011). DOI: 10.1016/j.sigpro.2011.12.009.