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

Evaluation Study of Compressed Sensing for Neural Spike Recordings

Christoph Bulach, Ulrich Bihr and Maurits Ortmanns

Abstract— **In this paper, an evaluation study of compressed sensing (CS) for neural action potential (spike) signals in MATLAB is presented. State-of-the-art neural recorders use 100 or more parallel channels to measure neural activity resulting in a data rate of 16 – 20 Mbit/s. Since a low-power design is required for an implanted neural recorder, it seems advantageous to compress the neural data prior to the wireless transmission. The continuous neural spike signals are compressed and transmitted to facilitate the possibility of an unrestricted data analysis at the receiver. Synthesized and recorded neural data sets are used to test the performance of CS. The 6-level Daubechies-8 wavelet decomposition matrix and two learned dictionary matrices are utilized as dictionaries for CS. The compression results are evaluated with the spike sorting program OSort. CS is shown to work for the compression of low-noise synthesized neural spike signals with a compression rate of 2.05, but cannot be recommended for the compression of neural spike signals in general.**

I. Introduction

Implanted neural recorders are widely used to monitor neural activity in the human brain. Due to the heat sensitivity of brain tissue, the power dissipation of the recording chip should be around 10 mW [1] and thus, a power efficient design is necessary. Neural recorders use a small electrode array of thin needles [2] to measure 100 or more extracellular signals from multiple neurons in parallel. These extracellular recordings can be decomposed into local field potential (LFP) and the spike signal. The LFP is concentrated between 10 – 200 Hz and the spike signal between 300 Hz – 5 kHz [2]. The acquired neural signal of each channel is separated into LFP and spike signal, which are then processed separately. This paper only considers the neural spike signal. With a common quantization of 8 – 10 bits [3] and a sampling frequency of 20 kHz, the simultaneous acquisition of 100 channels results in a data rate of at least 16 Mbit/s for the spike signal. This data has to be transmitted wirelessly out of the body, since no cables can be attached to the human being to guarantee maximal mobility during the experiments. Due to the power constraints of implanted neural recorders, the wireless transmission of the calculated spike data rate favors a compression of the spike signal prior to the transmission.

A neural recording system with 100 recording channels was shown in [1], where the timestamps of the detected spikes are transmitted to the receiver. Furthermore, the spike waveform of one selectable channel is transmitted. Therefore, a spike sorting of the neural signals from all channels is not possible at the receiver. The use of compressed sensing (CS) [4] for the compression of neural recordings

C. Bulach, U. Bihr and M. Ortmanns are with the Institute of Microelectronics, University of Ulm, 89081 Ulm, Germany `ulrich.bihr@uni-ulm.de`

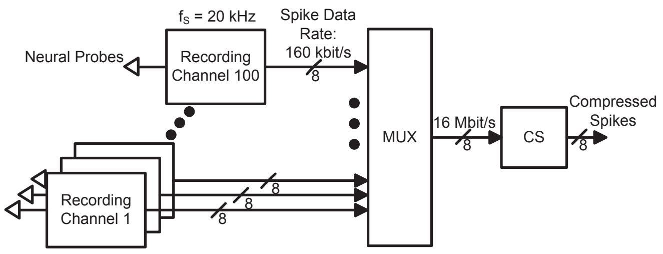


Fig. 1. Embedding of CS into the 100-channel neural recorder frontend

was demonstrated to work for low-noise neural spike signals in [5], but the results of the signal reconstruction after CS were not evaluated by means of a spike sorting algorithm; even though this is mandatory when a lossy compression is used for neural spike compression. The compression of detected and extracted neural spikes in a neural recording with CS was reported in [6].

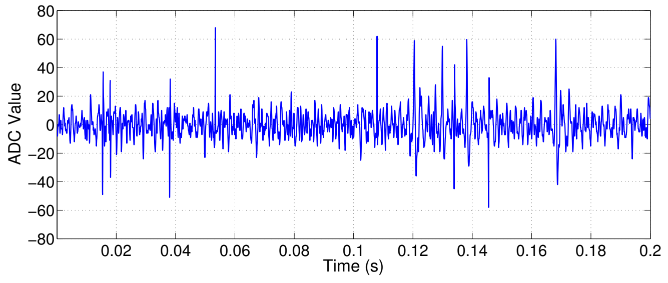
In this paper, CS is implemented for the compression of neural spike signals and the compression results are evaluated with the spike sorting program OSort [7]. The neural spike signal and not only extracted spikes are compressed and transmitted for the possibility of an unrestricted data analysis at the receiver. Fig. 1 shows the embedding of CS into the neural recorder frontend. The advantages of CS are the high possible compression rate and the low complexity of the compression scheme. Because of the sparse nature of neural spike signals, CS seems to be suited for their compression.

This paper is structured as follows. The methods used for the evaluation of the compressed data sets are described in section II. The theory of CS and the implementation are given in section III. The performance of CS is shown in section IV. Section V concludes the paper.

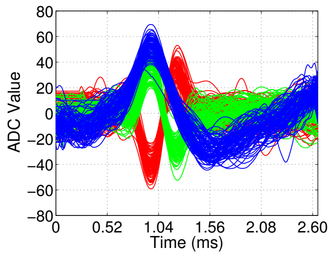
II. Methods

A. Evaluation of CS

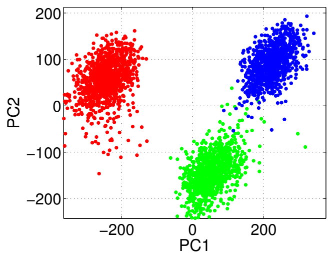
A suitable evaluation scheme is necessary to test whether a reconstructed signal after any kind of compression contains the relevant information of the uncompressed signal. In a neural recording, the number, timestamps and shapes of the spikes are relevant to decode either a spike rate code or a code based on the spike timings. The analysis of a spike signal is commonly performed with a spike sorting program. The spikes emitted by different neurons are distinguishable by the shape of their waveforms and amplitudes [7]. In a spike sorting program, the spikes are first detected and then, based on the spikes' waveforms, sorted into different clusters representing the active neurons. The spike sorting software ”OSort” [7] was used to evaluate the spike reconstruction quality after compression and subsequent decompression of



(a)



(b)



(c)

Fig. 2. (a) Time-domain signal excerpt of a synthesized neural signal [8]. (b) Spikes detected in the neural signal (60 sec.) containing the excerpt shown in (a). The colors show the presence of three different spike waveforms in the recording. (c) Plot of the first two principal components (PC) of the spikes in (b). The colors signalize the different detected clusters.

the neural signal. Fig. 2a shows an excerpt of a neural spike signal [8]. After sorting of a neural signal, OSort plots the waveforms of the detected spikes and assigns the waveforms to different clusters (see Fig. 2b). The principal component analysis is used to generate a cluster plot of the spikes (see Fig. 2c).

The evaluation of a compressed data set is performed as shown in Fig. 3. At first, the neural spike signal is sorted without compression (*reference* sorting) and then after compression and subsequent decompression (*measured* sorting). Both results are compared according to a set of parameters reflecting the relevant features of a neural signal:

- Percentage of recovered spikes
- Percentage of correctly assigned spikes
- Percentage of additionally detected spikes
- Number of additionally detected clusters

The recovered spikes appear in both the reference and the measured sorting. If a spike is assigned to the same cluster in both sortings, it is counted as correctly assigned spike. Since the sorting algorithm tends to overcluster after a signal distortion, it is sufficient to check for additional clusters. A missing cluster is detected by a significant drop of correctly assigned spikes. The thresholds for a measured sorting to be considered similar enough to a reference sorting were defined and validated by a large number of sortings. The percentage of recovered and correctly assigned spikes had to be greater than 95 %, the percentage of additionally detected spikes less than 5 % and no additional clusters were permitted.

B. Selection of neural data sets

Freely available data sets were used for the simulation of CS to facilitate the reproduction of the results. The first group of data sets is referred to as *synthesized data sets* [8]. For each of these data sets, a spike train was synthesized using 3 spike templates which were created from neocortex and basal ganglia recordings. Then noise created by the unused spike templates with a standard deviation of 0.05 – 0.4 times the normalized spike templates' amplitudes was added [9]. The number of active neurons and emitted spikes is known for every data set. The second group of data sets is called *recorded data sets* which are measured recordings of the hippocampus of anesthetized rats [10]. The correct number of active neurons and spikes in these data sets is unknown.

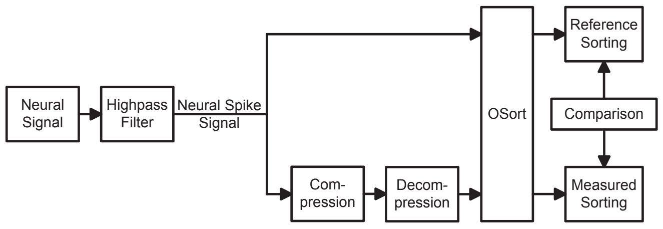


Fig. 3. Block diagram for the performance evaluation of CS with OSort

C. Neural signal processing

Since only the neural spike signal had to be compressed, it was separated from the LFP before the simulations. Therefore, the neural data sets were highpass filtered using a Butterworth highpass filter with a cutoff frequency of 300 Hz. After filtering, the spike signal was quantized with 8 bits, because simulations of spike quantizations in OSort showed, that with a quantization of less than 8 bits the percentage of recovered and correctly assigned spikes dropped significantly. These quantized spike signals were used for CS and subsequent performance evaluations.

III. Theory and Implementation

CS can be viewed as a compression scheme, since it enables the reconstruction of a signal from a small number of measurements corresponding to the information rate of this signal [11].

A. Sparsity

The requirement for CS is the existence of a basis or dictionary (collection of time-discrete waveforms), wherein the signals which have to be compressed are sparse. A time-discrete signal x ($x \in \mathbb{R}^N$) is called *sparse* in a basis Ψ ($\Psi \in \mathbb{R}^{N\times N}$), if it can be described with a linear combination of only a small number K ($K \ll N$) of the basis functions [11]. Therefore, x can be written as [12]:

$$x = \Psi\alpha \;\; with \;\; \alpha \in \mathbb{R}^N \,, \tag{1}$$

with the basis functions ψ_i written in the columns of the matrix Ψ ($\Psi = [\psi_1, \psi_2, ..., \psi_N]$). Vector α is called *K-sparse* for having K nonzero entries. If α only has K very large entries and $N - K$ small entries, it is called *K-compressible*.

B. Compressive Sampling

The compression of x is performed by multiplication of x with an $M \times N$ ($M \ll N$) sensing matrix Φ, resulting in a vector y ($M \times 1$) [12]:

$$y = \Phi x = \Phi\Psi\alpha = \Theta\alpha \tag{2}$$

Θ is obtained by the multiplication of Φ and Ψ and y is the compressed version of x, which is sent to the receiver. Thus, the fixed compression rate of CS is $\frac{N}{M}$.

C. Signal reconstruction

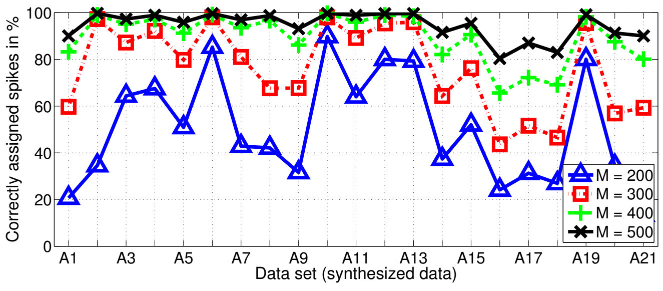
A stable reconstruction of the original signal x from $M \geq K$ measurements is possible, if the restricted isometry property (RIP) is fulfilled [12] and the coherence between the matrices Φ and Ψ is minimal [12]. CS theory shows, that random Gaussian and Bernoulli matrices fulfill the incoherence property and RIP with high probability [12] independently of the dictionary matrix Ψ. If the incoherence property and RIP are fulfilled for Θ, an exact reconstruction of α is possible with large probability. The recovery of the sparse vector α is performed by solving the convex optimization problem called *Basis Pursuit* [13]:

$$\widehat{\alpha} = \min_{\widetilde{\alpha} \in \mathbb{R}^M} \|\widetilde{\alpha}\|_{l_1} \text{ subject to } y = \Theta\widetilde{\alpha} \ , \tag{3}$$

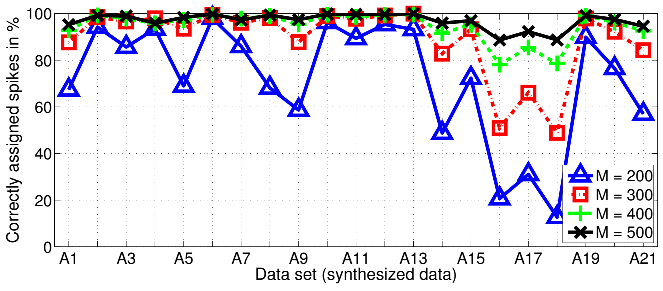
where $\|x\|_{l_1} = \sum_i |x_i|$ denotes the l_1-norm of x. This optimization problem can be rewritten as a linear program [13] and thus, solved in polynomial time. The original signal x is recovered as $\widehat{x} = \Psi\widehat{\alpha}$. This recovery is not computed on the implanted chip and thus, sufficient computing power can be assumed for its computation.

D. Implementation of CS

The number of rows and columns N in the dictionary matrix Ψ was chosen to be 1024. With a sampling frequency of 20 kHz and a typical spike duration of 2.5 ms, a spike uses about 50 samples. Dependent on the firing rate of neurons and the number of active neurons, the percentage of 1024 samples which is occupied by spikes is very variable. N was selected so large, because at least one spike was supposed to be present in every data set, even for lower spike emission rates of the neurons. The parameter M was chosen to range from 200 to 500, since the compressibility of the neural spikes in the analyzed dictionaries was never below 10 % of the 1024 coefficients. A multiple of 2 – 4 of the compressibility is usually used for the selection of M. Therefore, compression rates from 2.05 to 5.12 can be realized. Extracted and aligned neural spikes were shown to be compressible in the Daubechies-8 (db8) wavelet domain [6]. In the current paper continuous neural spike signals are transmitted. Thus, each neural signal was partitioned into segments of length $N = 1024$ containing an unknown number of spikes. The compressibility of those signal segments was analyzed with db8 wavelet decomposition matrices utilizing several decomposition levels. The 6-level db8 wavelet decomposition matrix was chosen as dictionary matrix, since it delivered the best compressibility. Furthermore, two dictionary matrices were created by a sparse dictionary learning program [14] to increase the compressibility of the analyzed signals. For both the recorded and the synthesized data sets a separate dictionary was generated using the data sets of each group in the learning process. A Gaussian random matrix with entries of a $\mathcal{N}(0, \frac{1}{M})$ distribution was selected as sensing matrix.



(a)



(b)

Fig. 4. Percentage of correctly assigned spikes for the synthesized data sets after CS with a Gaussian random sensing matrix. (a) Results obtained with the 6-level db8 dictionary matrix. (b) Results obtained with the learned dictionary matrix for synthesized data. $M = 200 - 500$ corresponds to a compression rate of 5.12 – 2.05.

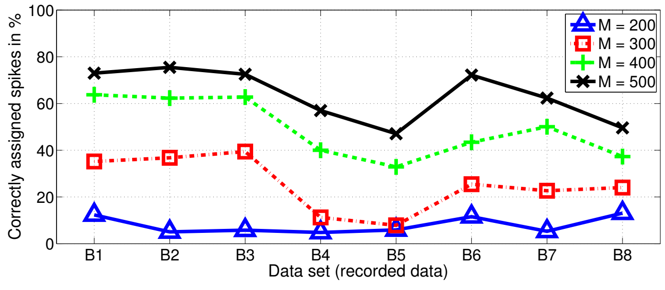
IV. Results

The results of the selected evaluation parameters in OSort show a high correlation and therefore, only the results for the correctly assigned spikes are shown here.

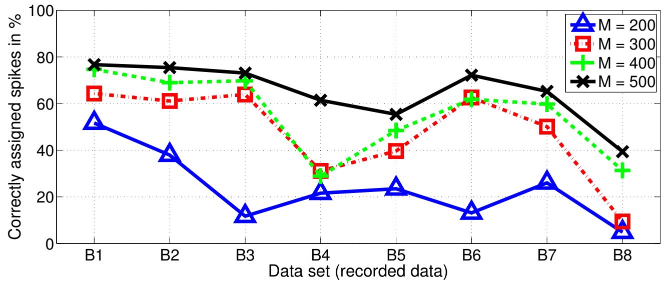
A. CS results with synthesized data sets

Fig. 4a shows the percentage of correctly assigned spikes for the synthesized data sets after CS with the 6-level db8 dictionary matrix. The data sets with the lowest noise level of 0.05 times the spike templates' amplitudes (A2, A6, A10 and A19) perform best. For the two highest values of M, the percentage of correctly assigned spikes exceeds 95 % for most data sets, but the percentage of correctly assigned spikes in the data sets A16 – A18 remain far below 90 %. These are the data sets with the highest noise levels of 0.35, 0.3 and 0.4 respectively.

The percentage of correctly assigned spikes obtained after CS with the learned dictionary is shown in Fig. 4b. For $M = 500$ the performance is on average 2.3 % better than with the wavelet dictionary. For both dictionaries the percentage of correctly assigned spikes is correlated with the known noise level in the data sets. The lower the noise level, the higher the percentage of correctly assigned spikes. To summarize the results, CS for the synthesized data sets utilizing $M = 500$ and a 6-level db8 wavelet dictionary matrix works for all data sets with the lowest noise level of 0.05 and for some data sets with a noise level of 0.1, 0.15 and 0.2 according to the defined criteria in OSort. Thus, CS can be used to compress 52.4 % of the synthesized data sets with a compression rate of 2.05. CS for the learned dictionary works for 76.2 % of the synthesized data sets



(a)



(b)

Fig. 5. Percentage of correctly assigned spikes for the recorded data sets after CS with a Gaussian random sensing matrix. (a) Results obtained with the 6-level db8 dictionary matrix. (b) Results obtained with the learned dictionary matrix for recorded data. $M = 200 - 500$ corresponds to a compression rate of 5.12 – 2.05.

using $M =$ 500. These are all data sets (except A4 and A21) with a noise level lower than or equal to 0.25.

B. CS results with recorded data sets

The percentage of correctly assigned spikes for the recorded data sets after CS with the 6-level db8 wavelet dictionary matrix is shown in Fig. 5a. Data set B5 shows the poorest performance for $M =$ 500. This is the data set, in which the largest number of clusters of all recorded data sets is detected in the reference sorting. The noise level in the recorded data sets is higher than in the synthesized data sets, but difficult to measure exactly, since no ground truth data are available. Because of the higher noise level, CS might not be able to reconstruct the spikes with the required variability. Consequently, many spikes are assigned to wrong clusters by OSort. For all data sets the percentage of correctly assigned spikes is below 95 % and thus, the requirements for a good sorting result in OSort are not met by any data set.

The percentage of correctly assigned spikes using the learned dictionary is shown in Fig. 5b. For $M =$ 500 the performance is on average 1.2 % better than with the wavelet dictionary. The learned dictionary provides better compressibility than the wavelet matrix, since the waveforms of the dictionary were learned from the original spike waveforms. However, no data sets meets the defined requirements for a good sorting result in OSort and thus, CS does not work satisfactorily with $M =$ 200 – 500 for both dictionaries.

V. Conclusion

This paper showed the implementation of CS for the compression of synthesized and recorded neural spike signals in MATLAB. The performance of CS was evaluated using the spike sorting software OSort. The simulations showed the performance of CS to be highly dependent on the noise level in the neural data set. With increasing noise level the original compressibility assumption gets invalid. If the standard deviation of the noise level of the synthesized data sets is lower than or equal to 0.05 times the amplitude of the spike templates, CS utilizing the 6-level db8 wavelet decomposition matrix as dictionary matrix was shown to work satisfactorily with a compression rate of 2.05. In contrast to the compression of continuous neural spike signals shown in [5], CS for noisy recorded signals is found to be not working satisfactorily.

However, CS could be used to compress extracted and aligned neural spikes as shown in [6]. In that case, the variability of the data can be controlled and the compressibility of the analyzed signal is guaranteed, since only one spike is compressed at a time. If only detected and extracted spikes were compressed and transmitted to the receiver, the compression rate would increase, but the possibility of unrestricted data analysis at the receiver would be lost. The use of CS to compress neural spike signals, while keeping the entire information of the signal, is not recommended.

References

[1] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, and F. Solzbacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 123–133, Jan. 2007.

[2] R. Harrison, "The design of integrated circuits to observe brain activity," *Proceedings of the IEEE*, vol. 96, no. 7, pp. 1203–1216, July 2008.

[3] A. Sodagar, K. Wise, and K. Najafi, "A fully integrated mixed-signal neural processor for implantable multichannel cortical recording," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 6, pp. 1075–1088, June 2007.

[4] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] F. Chen, A. Chandrakasan, and V. Stojanović and, "A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors," in *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, Sept. 2010, pp. 1–4.

[6] Z. Charbiwala, V. Karkare, S. Gibson, D. Markovic, and M. Srivastava, "Compressive sensing of neural action potentials using a learned union of supports," in *International Conference on Body Sensor Networks (BSN), 2011*, May 2011, pp. 53–58.

[7] U. Rutishauser, E. Schuman, and A. Mamelak, "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo," *J Neurosci*, vol. 154, no. 1-2, pp. 204–224, 2006.

[8] R. Q. Quiroga. (2012, Mar.) Wave_clus – simulator data. [Online]. Available: http://www.vis.caltech.edu/~rodri/Wave_clus/Simulator.zip

[9] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, pp. 1661–1687, Aug. 2004.

[10] J. Teeters and F. Sommer. (2012, Mar.) Collaborative research in computational neuroscience – data sharing. [Online]. Available: https://crcns.org/data-sets/hc/hc-1

[11] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[12] R. G. Baraniuk, "Compressive sensing," *Lecture Notes in IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–120, July 2007.

[13] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.