Deflation technique for neural spike sorting in multi-channel recordings

Zoran Tiganj, Mamadou Mboup

DEFLATION TECHNIQUE FOR NEURAL SPIKE SORTING IN MULTI-CHANNEL RECORDINGS

Zoran Tiganj

Non-A, INRIA Lille - Nord Europe,
parc Scientifique de la Haute Borne 40,
59650 Villeneuve d'Ascq, France

Mamadou Mboup

Université de Reims Champagne-Ardenne,
CReSTIC, UFR SEN, BP 1039
Moulin de la Housse 51687 Reims, France

ABSTRACT

We propose an ICA based algorithm for spike sorting in multi-channel neural recordings. In such context, the performance of ICA is known to be limited since the number of recording sites is much lower than the number of the neurons around. The algorithm uses an iterative application of ICA and a deflation technique in two nested loops. In each iteration of the external loop, the spiking activity of one neuron is singled out and then deflated from the recordings. The internal loop implements a sequence of ICA and spike detection for removing the noise and all the spikes that are not coming from the targeted neuron. We validate the performance of the algorithm on simulated data, but also on real simultaneous extracellular-intracellular recordings. The results show that the proposed algorithm performs significantly better than when only ICA is applied.

Index Terms— Spike sorting, Iterative ICA, Deflation

1. INTRODUCTION

In cortical systems information between neurons is transmitted using action potentials (spikes). These are commonly recorded with extracellular electrodes which capture a mixture of activity of a vast number of neurons around. Identifying and isolating the activity of individual neurons in the mixture is one mandatory step for many tasks in neuroscience. This is known as the *spike sorting* problem.

Even though the basic form of an action potential is the same for all neurons, the spikes recorded with an extracellular electrode have different shapes and amplitudes if they are coming from different neurons [1]. This property, which is due to the propagation and the velocity effects, is on the basis of spike sorting methods (for review, see: [2] and [3]).

The majority of algorithms consist of several common steps. The first step is spike detection: The goal is to detect and find the time occurrences of the action potentials emitted by the neurons close to the electrode. Depending on the signal to noise ratio, spike detection is done either by simple thresholding or by using a more advanced technique *e.g.* [4], or [5]. The next step is to apply some feature extraction technique, such as Principal Component Analysis (PCA) [6] or wavelet based methods [7], to represent each detected spike in a reduced dimensional space. Finally, the features are used as sort keys for the classification of the spikes into different clusters, each of which is assumed to contain the spiking activity of a single neuron. Among the more popular spike sorting algorithms, we may cite those based on k-means [8], expectation maximization (EM) [9], template matching [10] and also some more complicated approaches such as super-paramagnetic clustering [7]. The main problem in spike sorting is the lack of robustness. This is due to very unbalanced feature space: the number of elements per cluster can vary significantly from cluster to cluster, also mutual distances between clusters can be very different. When multi-site recordings are available, to make the feature vector space more clear, Independent Component Analysis (ICA) is sometimes applied (see [11], [12], [8] and [13] for examples). This improves the sorting results. However, the contribution of ICA is limited by the fact that number of neurons largely exceeds the number of electrodes, even though some *more-sources-than-sensors* versions of ICA are available [14].

In this paper we present a new algorithm for spike sorting in multi-channel extracellular recordings. We use a deflation technique to improve the performance of ICA: after we isolate the firing instants of a single neuron, we remove them from the original recording and repeat the procedure until the algorithm becomes unable to isolate any more neurons. Moreover, within each iteration we implement, in an internal loop, another iterative algorithm for removing the noise and all the spikes that are not coming from the neuron that appears as the closest to the electrode. At the end of the algorithm, we detect the overlapped spikes. To do this we take all the possible pairs of the isolated neurons and apply ICA only on the segments of the signal while one of the neurons in the pair was active. Each step of the algorithm is analyzed in the following section.

We also propose a clustering algorithm (see [15] for de-

This work was supported by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council and FEDER through the 'Contrat de Projets Etat Region (CPER) 2007-2013'.

tails) that is applied within each iteration of the internal loop. As this clustering algorithm is specifically designed for neural recordings, it presents an improved robustness compared to classical approaches in this specific application. The first step of the algorithm is standard: the detected spikes to be sorted are first processed with some feature extraction technique, such as PCA, and then represented in a space with reduced dimension by keeping only a few most important features. The resulting space is next filtered in order to emphasis the differences between the centers and the borders of the clusters (we assume that the extracted features have unimodal, non-uniform distribution). Using some prior knowledge on the lowest level of activity of a neuron, as *e.g.* the minimal firing rate, we find the number of clusters and the centers of each cluster. For classifying the spikes, we use a simple approach which also relies on the non-uniform distribution of the data: each cluster is initialized by its singleton center and the clustering is performed by classifying one spike at a time, in such a way that if an element belongs to a cluster then so does its nearest neighbor. This procedure leads to a very robust solution as shown in [15].

In section 3 we test the proposed algorithm on two types of signal: 1) Artificially created neural mixture, where we simulated 10 four-channel, two million samples long, extracellular recordings in population of 1000 neurons distributed around the electrode. 2) Five real simultaneous intracellular-extracellular recordings (see [1] where a detailed description of the recording process is given). We test the sorting algorithms on the extracellular recordings and use the intracellular ones to verify the results of the sorting for the neurons which were recorded intracellularly. The results confirm that ICA (alone) improves significantly the spike sorting. They also show that further significant improvement is afforded by the proposed method.

2. ALGORITHM DESCRIPTION

In this section we describe the proposed algorithm for the multi-channel spike sorting. The algorithm iteratively removes the activity of distant neurons from the original recording as well as the activity of neurons close to the electrode once they are isolated from the mixture. By doing so we reduce the number of sources, which brings the application of ICA in a more confortable setting.

Apart from the recorded signal, to use the algorithm we only need to provide a lower bound for the firing rate, call it G, and the spike detection threshold level. Minimal firing rate is usually easy to obtain since, generally, we are not interested in separating the activity of neurons that fired only a few spikes. The choice of the threshold level, on the other side, is not very critical, since we will simply neglect all the spikes that belong to the cluster which contains most of the spikes with lowest amplitude. This will make the algorithm very robust and reliable, since the spikes with the lowest amplitudes usually come from many different neurons, which activity is only partially detected.

We assume that a four-channel recording is given, but the algorithm is the same for any number of channels. The four recorded signals are labeled as $E1$, $E2$, $E3$ and $E4$. Example of the four-channel recording is shown on figure 1 (spikes from the neuron recorded intracellularly are marked with the red starts).

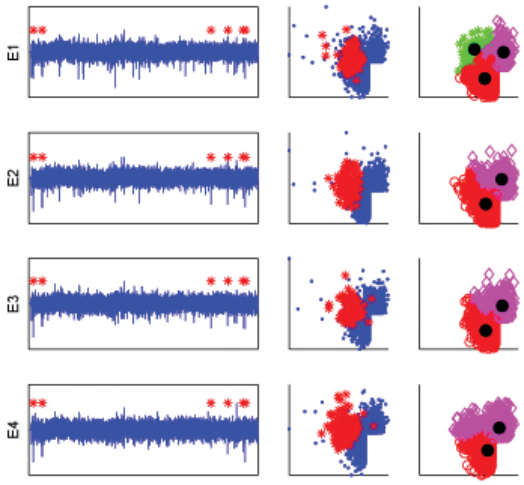

Fig. 1. 50000 samples from the real four-channel extracellular recording (first column). 2D feature vector space with the extracted spikes - positive peak amplitude vs negative peak amplitude (second column). Spikes that come from the neuron recorded intracellularly are labeled the with the red stars. Results of sorting algorithm from [15] are given in the third column, where the black dots mark the detected centers of the clusters.

Only to demonstrate that spike sorting is very difficult for such recording we apply directly the method based on a feature extraction, described in [15]. First, we detect the spikes from each recording site using a simple spike detection method based on filtering of the recorded signal with a specially designed Volterra filter that emphasis spike shaped waveforms [5], [16]. We project the spikes in a 2D vector space by keeping only two features per spike: the positive and the negative peak amplitudes. Notice that one could also use the *e.g.* the two first principal components obtained from PCA. The plots of the features are shown next to the corresponding signals, in the second column on figure 1. In the third column we give the outputs of the algorithm from [15]. The spikes that are left unsorted by the algorithm are automatically eliminated and not plotted. The colors/symbols chosen for the representation of the clusters for the plots in the second column are not related to the one in the third column. It is clear that from such feature vector space it is not possible to identify directly the activity of the neuron recorded intracellularly.

We will now demonstrate the performance of the proposed algorithm using step-by-step description. The flowchart in figure 2 describes the proposed algorithm.

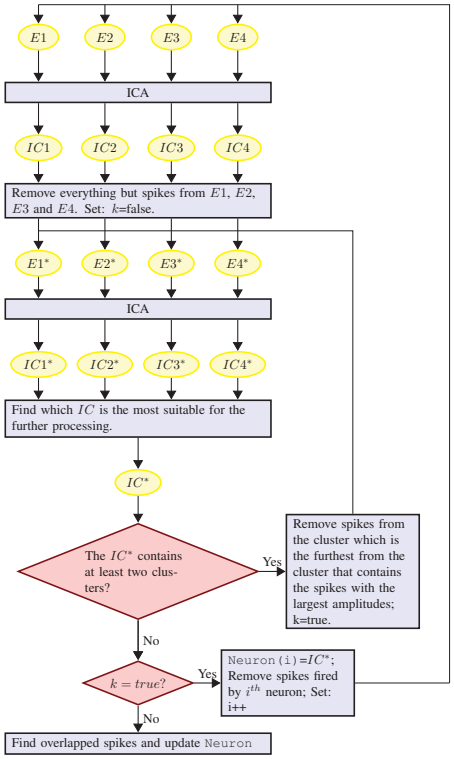

Fig. 2. Flowchart of the proposed multi-channel spike sorting algorithm.

• `Step 1:` **ICA.** We process the input signal with the classical fast-ICA algorithm [17][1] and obtain $IC1$, $IC2$, $IC3$ and $IC4$ as the outputs (figure 3). The second and the third column on the figure are again shown only to demonstrate of what the results would be if we would apply the feature extraction based sorting method directly on the outputs of the ICA algorithm. The difference between the clusters is more obvious than on figure 1.

• `Step 2:` **Remove everything but spikes from $E1$, $E2$, $E3$ and $E4$. Set: k=false.** The IC with highest energy will be kept for for the further processing. To determine it, we compute the average amplitude of the G dominant spikes on each IC. The one with the highest average amplitude is retained. This IC is indeed likely to contain the activity of the neurons closest to the electrode. We then detect all the spikes from that IC using the method mentioned before in this section and described in [5] and [16]. Next, from all the 4 input recordings we remove (set to zero) all but the detected spikes. We assume that the removed parts of the signal correspond

[1]We used the contrast function $g = u^3$ what gave reasonably good results, but comparing with different choices of g and also with different versions of ICA algorithm would be interesting to investigate in the future.

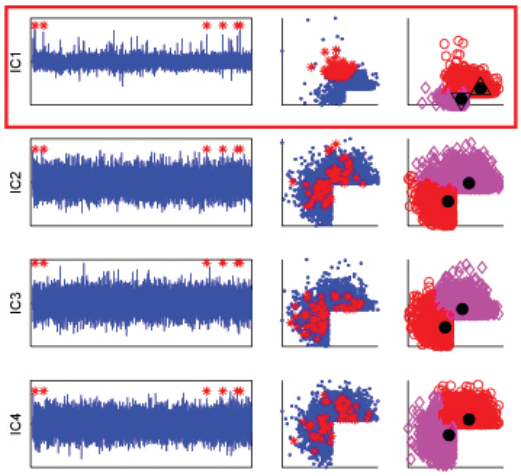

Fig. 3. ICA results, on the recording shown of figure 1. The four ICs are shown in the first column (50000 samples). 2D feature vector space with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from [15] on the vector space from the second column. The red rectangle marks the IC that has G spikes of the highest amplitude.

only to the noise and the activity of the neurons that are farthest from the electrode. The outputs are called $E1^*$, $E2^*$, $E3^*$ and $E4^*$.

• `Step 3:` **ICA.** We apply ICA on $E1^*$, $E2^*$, $E3^*$ and $E4^*$. The resulting ICs, called $IC1^*$, $IC2^*$, $IC3^*$ and $IC4^*$, are shown on figure 4. To demonstrate the results as before, we plot the spikes in the feature vector space (second column) and output of the feature extraction based sorting algorithm (third column). As the number of the sources is now lower, the ICA algorithm performed a little better and a little bigger difference between clusters is visible, in comparison with the results from figure 3.

• `Step 4:` **Find which IC is the most suitable for the further processing.** In the further steps we will need only one IC. Thus in this step we want to chose the one that contains well preserved spike waveforms. It is likely that the best choice would be the one for which the detected spikes show the largest dynamics. As a measure of spike dynamics we use the average difference between the positive and the negative peak. Since spikes naturally contain a large peak, such difference should be a good measure of the dynamics. For the further processing we chose IC which has the largest value of the difference and call it IC^*.

• *`Condition 1:`* **The IC^* contains at least two clusters?**

• `If` *`Condition 1`* `is true:` **Remove spikes from the cluster which is the furthest from the cluster that contains the spikes with the largest amplitudes; k=true.** In this step we have to apply the algorithm proposed in [15], this time it is not only for demonstration, but to find which spikes

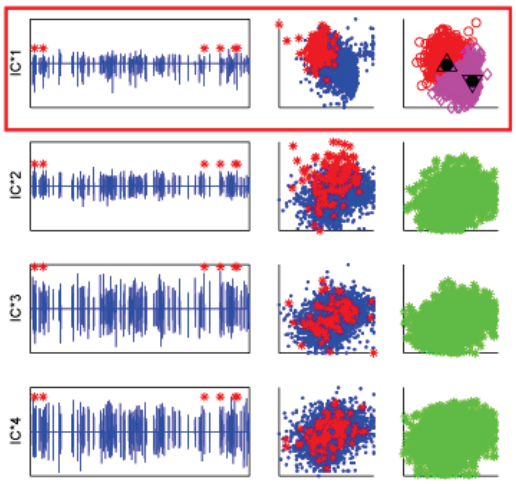

Fig. 4. Results of ICA after all but spikes was removed from the signal (set to zero) - 50000 samples are displayed. The four ICs are shown in the first column. 2D feature vector spaces with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from [15] on the vector space from the second column. IC that has been chosen for the further processing is marked with a red rectangle. The black triangle, with the tip pointing up, marks the center of the cluster that contains the spikes generally higher in the amplitude than the spikes from the other clusters. Opposite, the black triangle with the tip pointing down marks the center of the cluster that contains the spikes with generally lower amplitudes.

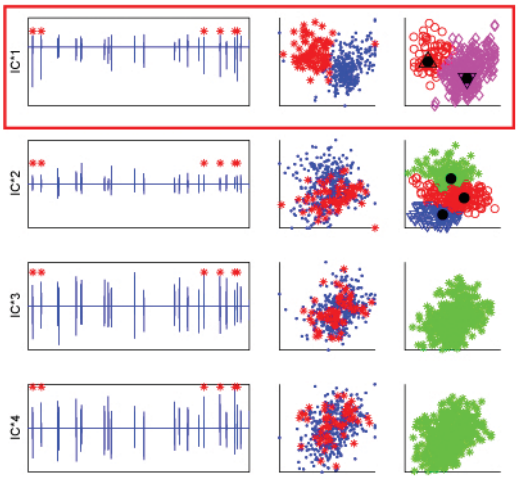

Fig. 5. Result of applying ICA on the real recording, after removing (setting to zero) of the low amplitude spikes (see figure 4) from $E1^*$, $E2^*$, $E3^*$ and $E4^*$ (50000 samples). The four ICs are shown in the first column. 2D feature vector spaces with the spikes extracted from the ICs are in the second column. The plots in the third column display the results of applying the algorithm from [15] on the vector spaces from the second column. All the labels are the same as for figure 4.

we can remove from the signal. We identify the cluster that contains majority of the spikes of the largest amplitudes. The spikes that have been classified in the furthest cluster are now removed from $E1^*$, $E2^*$, $E3^*$ and $E4^*$. Next, the algorithm goes back to the `step 3`. Figure 5 shows the outputs after applying ICA on the new $E1^*$, $E2^*$, $E3^*$ and $E4^*$. On $IC1^*$ and $IC2^*$ we have detected more than one cluster. We use the same criterion as before to chose which IC we should keep. In this case it is $IC1^*$. The cluster that will be removed is the one labeled with a black triangle whose tip is pointing down. After that cluster is removed, spikes from the neuron recorded intracellularly will be almost the only one remaining in the signal. Only a few spikes from the intracellularly recorded neuron were removed. We set in this step the flag k, in order to track if the algorithm converged just after a new neuron is isolated, what would indicate generally that no more neurons can be separated from the given recording and the iterative part of the algorithm would ended.

• `If` *`Condition 1`* `is false:` *`Condition 2:`* **k = `true`?**

• `If` *`Condition 2`* `is true:` **Neuron(i)=IC^*; Remove the spikes fired by the i^{th} neuron; Set: $i++$.** If k is `true` that means that we have detected more than one cluster in the previous iteration so now, since only one cluster is left, we can conclude that the remaining spikes are coming from the same neuron. This is the i^{th} found neuron. To continue our iterative algorithm we set to zero all the samples from the recording (from $E1$, $E2$, $E3$ and $E4$) when any of already detected neurons (`Neuron(1)`, `Neuron(2)`,...,`Neuron(i)`) was active and go back to `step 1`. By doing this we removed an important high amplitude sources.

• `If` *`Condition 2`* `is false:` **Find the overlapped spikes and update** `Neuron`. The proposed deflation based algorithm is obviously not able to detect simultaneous firing, since once a spike has been sorted it is removed from the signal for the following iterations. To find simultaneously fired spikes we analyze the activity of the neurons two by two, for all possible pairs of the detected neurons. When analyzing the activity of each pair, we keep in the recording only the segments when one of the neurons in the pair fired a spike. Each such recording, call it $\tilde{s}_{i,j}$, where $i = 1, ..., S-1$ and $j = i+1, ..., S$ can be expressed as:

$$\tilde{s}_{i,j} = s_i + s_j + sgn(|s_i| + |s_j|) \sum_{k=1,k\neq i,j}^{S} s_k.$$

It is obvious that most of the energy in any $\tilde{s}_{i,j}$ comes from s_i and s_j. Thus, applying ICA on $\tilde{s}_{i,j}$ leads to the separation of s_i and s_j: one IC will represent mostly s_i, another one s_j and the remaining ICs will represent $sgn(|s_i| + |s_j|) \sum_{k=1,k\neq i,j}^{S} s_k$. Since the activity of s_i and s_j will be

separated, any spikes fired simultaneously by these two neurons should be visible on both of the ICs that represent the activity of these neurons. We demonstrate this by an example given on figure 6.

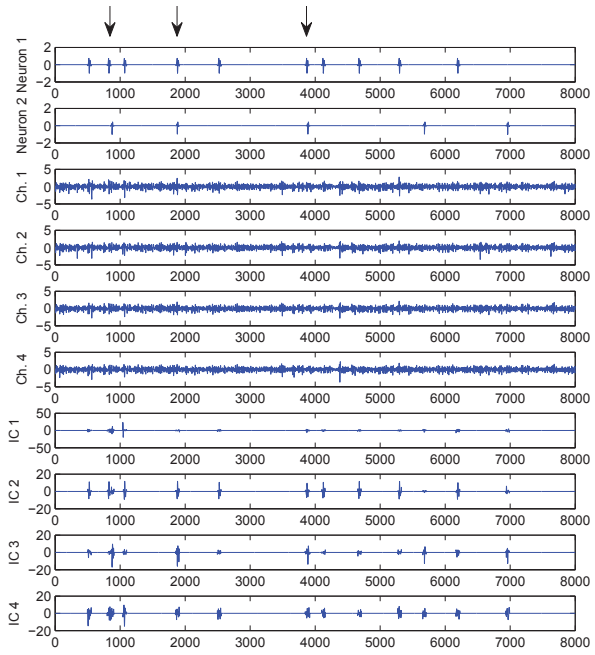

Fig. 6. The first two plots show part of the activity of the two simulated neurons. The following four plots show the part of the four-channel simulated recording. The four plots at the bottom show the results of applying ICA on the four channels. Before applying ICA the recording was modified by setting to zero all but the segments where one of the two neurons was active. It appears that the activity of *Neuron 1* is represented by *IC3* and the activity of *Neuron 2* is represented by *IC1*. The three arrows on the top of the figure point out the overlapped spikes from the two neurons. All of the three overlapped spikes are successfully sorted.

3. RESULTS

We apply the algorithm described in the previous section on 5 real recordings and on 10 simulated recordings. The results are given in tables 1 and 2. There are four key-operations in the proposed algorithm: 1) removing of the noise (*Noise rem.*); 2) removing of the spikes from the clusters furthest from the one that contains the spikes with highes amplitudes (*Clus. rem.*); 3) removing of the spikes from already separated neurons (*Sorted rem*); 4) detection of the overlapped spikes (*Over. det.*). We analyze contribution of each of these steps by simply comparing the proposed algorithm to algorithms which perform none, only one or more of these steps.

We express the results in terms of sorting accuracy (SA) and sorting mistake (SM). For the real recordings we can quantify the results only for one neuron per recording, the one which was simultaneously recorded intracellularly. Thus SA and SM are very simple to calculate: $SA(\%) = 100 * C/(F+C)$ and $SM(\%) = 100 * (T-C)/T$. Here C is the largest number of spikes from the neuron recorded intracellularly, which are sorted into the same cluster (correct detections). F is the number of spikes from other neurons which are wrongly placed into the same cluster for which C is calculated (false spikes). T is the total number of spikes fired by an intracellularly recorded neuron.

Table 1. Comparison of different spike sorting algorithms, in terms of Sorting Accuracy (SA) and Sorting Mistake (SM), on the real recordings. The column *Recording* contains the labels of each recording as in [1].

Recording	Feat. extr.		Feat. extr. ICA		Feat. extr. ICA Sorted rem.		Feat. extr. ICA Sorted rem. Noise rem.		Feat. extr. ICA Sorted rem. Noise rem. Clus. rem.		Feat. extr. ICA Sorted rem. Noise rem. Clus. rem. Over. det.	
	SA %	SM %	SA %	SM %	SA %	SM %	SA %	SM %	SA %	SM %	SA %	SM %
d561102	79	46	83	7	83	7	90	8	93	8	93	8
d561103	32	3	48	5	52	15	83	17	89	20	89	14
d561104	21	2	33	4	49	12	75	13	83	16	84	11
d533101	73	67	64	26	64	26	78	28	81	29	81	29
d533102	64	52	80	13	80	13	88	15	92	18	92	16

In table 1 we give SA and SM for each of the real recordings. Durations of the recordings were from 60 to 120 seconds with the sampling frequency of 10kHz for some recordings and 20kHz for others. The total number of the detected spikes per recording was from 800 to 2000, while the number of spikes fired by the intracellularly recorded neuron was from 200 to 500.

On the other side, using simulated recordings we can quantify the results for all clusters. We are not limited to only one cluster as in the example with the real signal. For each simulated recording we set the maximum number of clusters to 6. However, in some situations (especially for simpler methods like basic feature extraction and combination of feature extraction and ICA) less than 6 clusters have been found. Thus, we do a computation of SA and SM in a bit different way than for the real signal. We compute the average $\bar{SA}$ and the average $\bar{SM}$ over all 10 simulations for each j^{th} sorted cluster, where j goes from 1 to 6, which is the maximum number of sorted clusters:

$$\bar{SA}_j = \frac{\sum_{i=1}^{N_j} SA_{i,j}}{N_j}, \bar{SM}_j = \frac{\sum_{i=1}^{N_j} SM_{i,j}}{N_j}$$

where N_j is the number of iterations in which at least j clusters were detected. We give also a standard deviation (sd) of $\bar{SA}$ and $\bar{SM}$ in order to quantify the robustness of each algorithm. The results are given in table 2.

We can conclude from tables 1 and 2 that the proposed approach improves significantly the sorting results in comparison with the basic feature extraction approach as well as with

Table 2. Comparison of different spike sorting algorithms in terms of average sorting accuracy ($\bar{SA}$), average sorting mistake ($\bar{SM}$) and their standard deviations (sd) for first 6 sorted clusters from simulated recordings. Label *j* stands for the cluster index. N is the number of trials, out of 10, in which at least *j* clusters were detected.

j	Feat. extr.			Feat. extr. ICA			Feat. extr. ICA Sorted rem.			Feat. extr. ICA Sorted rem. Noise rem. Clus. rem. Over. det.		
	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N	$\bar{SA}$ %	sd %	N
1	18 15	3 5	10	68 12	7 9	10	76 10	9 7	10	88 6	12 6	10
2	45 7	20 7	2	62 13	10 10	10	74 11	14 8	10	86 7	16 6	10
3	0 0	0 0	0	54 14	15 18	5	69 13	21 11	10	82 7	20 8	10
4	0 0	0 0	0	44 16	18 12	2	60 15	23 12	10	79 9	24 10	10
5	0 0	0 0	0	0 0	0 0	0	54 16	26 16	9	75 11	26 11	10
6	0 0	0 0	0	0 0	0 0	0	39 17	30 18	7	73 11	30 12	9

the approach based on using only the ICA algorithm. We can say that each of the steps of the algorithm has beneficial effect on the final sorting results. As it is expected, the proposed algorithm generally decreases the number of falsely sorted spikes and increases the number of spikes which are left unsorted (missed spikes). In the trade-off between the number of missed spikes and the number of false spikes, in neural coding applications, it is highly preferable to have more missed spikes [18]. Finally we can say that the proposed algorithm shows improved robustness and gives reliable results as compared to ICA. However, it does not have an error compensation step. A mistake in a given iteration will likely have an influence in the following iterations. But, as only one neuron is targeted at each time, an error does not affect the sorting results of all the neurons.

4. REFERENCES

[1] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsaki, "Intracellular Features Predicted by Extracellular Recordings in the Hippocampus In Vivo," *J. Neurophy.*, vol. 84, pp. 390–400, 2000.

[2] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials.," *Network*, vol. 9, pp. 53–78, 1998.

[3] S. Gibson, J. W. Judy, and D. Markovic, "Comparison of spike-sorting algorithms for future hardware implementation," *IEEE EMBS*, vol. 1, pp. 5015–20, 2008.

[4] Z. Nenadic and J. W. Burdick, "Spike detection using the continious wavelet transform," *IEEE Trans. in Biomed. Eng.*, vol. 52, pp. 74–87, 2005.

[5] Z. Tiganj and M. Mboup, "Spike detection and sorting: Combining algebraic differentiations with ica," in *ICA and Sig. Sep., 8th Int. Conf.*, Brazil, 2009, pp. 475–482.

[6] D. A. Adamos, E. K. Kosmidis, and G. Theophilidis, "Performance evaluation of pca-based spike sorting algorithms," *Comp. Meth. and Prog. in Biomed.*, vol. 91, pp. 232–244, 2008.

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

[8] S. Takahashi, Y. Anzai, and Y. Sakurai, "A new approach to spike sorting for multi-neuronal activities recorded with a tetrode–how ICA can be practical.," *Neur. Res.*, vol. 46, pp. 265–272, 2003.

[9] F. Wood, M. Fellows, J. Donoghue, and M. Black, "Automatic spike sorting for neural decoding," in *IEEE EMBS*, 2004, vol. 26 I, pp. 4009–4012.

[10] C. Vargas-Irwin and J. P. Donoghue, "Automated spike sorting using density grid contour clustering and subtractive waveform decomposition.," *J. Neurosci. Meth.*, vol. 164, pp. 1–18, 2007.

[11] Y. Shiraishi, N. Katayama, T. Takahashi, A. Karashima, and M. Nakao, "Multi-neuron action potentials recorded with tetrode are not instantaneous mixtures of single neuronal action potentials.," *IEEE EMBS.*, pp. 4019–4022, 2009.

[12] G. D. Brown, S. Yamada, and T. J. Sejnowski, "Independent component analysis at the neural cocktail party.," *Trends in Neurosci.*, vol. 24, no. 1, pp. 54–63, 2001.

[13] S. Takahashi, Y. Anzai, and Y. Sakurai, "Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes.," *J. Neurophysiology*, vol. 89, no. 4, pp. 2245–2258, 2003.

[14] P. Comon and C. Jutten, *Handbook of Blind Source Separation, Independent Component Analysis and Applications*, Academic Press (Elsevier), 2010.

[15] Z. Tiganj and M. Mboup, "A non-parametric method for automatic neural spikes clustering based on the non-uniform distribution of the data," *Submitted*, 2011.

[16] M. Mboup, "A volterra filter for neuronal spike detection," *Research report, INRIA, http://hal.inria.fr/inria-00347048/fr*, 2008.

[17] A. Hyvärinen, "Fast and robust Fixed-Point algorithms for independent component analysis," *IEEE Trans. on Neur. Networks*, vol. 10, pp. 626–634, 1999.

[18] N. G. Ilan and H. J. Don, "Information theoretic bounds on neural prosthesis effectiveness: The importance of spike sorting," in *ICASSP*, 2008, pp. 5204–5207.