Journal of Neuroscience Methods 198 (2011) 125–134

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Fuzzy logic-based spike sorting system

Karthikeyan Balasubramanian*, Iyad Obeid

Temple University, Department of Electrical and Computer Engineering, College of Engineering and Architecture, Philadelphia, PA 19122, United States

ARTICLE INFO

Article history:
Received 17 June 2010
Received in revised form 17 March 2011
Accepted 23 March 2011

Keywords:
Fuzzy logic
Real-time spike processing
Shift invariance
Spike sorting

ABSTRACT

We present a new method for autonomous real-time spike sorting using a fuzzy logic inference engine. The engine assigns each detected event a 'spikiness index' from zero to one that quantifies the extent to which the detected event is like an ideal spike. Spikes can then be sorted by simply clustering the spikiness indices. The sorter is defined in terms of natural language rules that, once defined, are static and thus require no user intervention or calibration. The sorter was tested using extracellular recordings from three animals: a macaque, an owl monkey and a rat. Simulation results show that the fuzzy sorter performed equal to or better than the benchmark principal component analysis (PCA) based sorter. Importantly, there was no degradation in fuzzy sorter performance when the spikes were not temporally aligned prior to sorting. In contrast, PCA sorter performance dropped by 27% when sorting unaligned spikes. Since the fuzzy sorter is computationally trivial and requires no spike alignment, it is suitable for scaling into large numbers of parallel channels where computational overhead and the need for operator intervention would preclude other spike sorters.

Published by Elsevier B.V.

1. Introduction

Spike sorting is a common signal processing procedure that is frequently necessary in biomedical applications ranging from the analysis of electrocardiograms and electroencephalograms to speech processing and neuronal signal processing. It is performed primarily to classify spikes amongst themselves as well as to separate biologically relevant spikes from noisy spike-like artifacts. In general, spike sorting is a two-step process involving feature extraction of detected spikes followed by feature clustering; numerous algorithms exist for both steps (see Lewicki, 1998). Common spike sorting feature extraction algorithms include wave shape analysis, principal component analysis (PCA), and independent component analysis (ICA). Despite their widespread application, existing spike sorters are not well suited for applications involving large numbers of channels, i.e. >1000, especially in environments where computational resources may be limited. For example, the pursuit of implantable or wearable hardware for extracting high channel counts of brain machine interface signals has been hindered by the complexity of the signal processing algorithms and the sheer number of channels that must be integrated (Nicolelis, 2001). In this context, existing spike sorters suffer from two key limitations: (i) shift dependence, which demands a computationally expensive process of temporal alignment of spikes and (ii) the necessity of generating a template for each channel from accumulated trial data under user supervision.

We propose a feature extraction and spike characterization method that uses fuzzy logic at its core. The implementation of this method is computationally negligible and requires no user intervention after an initial rule-forming phase. The rules are specific to the particular spikes of interest (i.e. extracellular neural spikes) and must be determined ahead of time only once with no intervention required afterwards. The same rules apply across all channels and do not vary over time. Briefly, each detected spike is assigned a fuzzy score from zero to one, where "1" represents maximum *spikiness*. Scores are determined based on *fuzzified* spike features that are then operated on by a series of fuzzy rules to determine the spike's degree of membership in the output space. The fuzzy score can be clustered to sort the spikes. The fuzzy logic-based feature extraction system can be used as a standalone spike sorter, or alternatively, used prior to conventional spike sorting to easily eliminate detected events that do not merit sorting. We demonstrate the value of this technique by sorting single unit cortical action potentials. We also show that our technique has the benefits of being shift-invariant and of obviating the need for templates based on accumulated spikes.

Although this work was developed within the context of Brain Machine Interfaces that operate on trains of sorted extracellular spikes, our technique is general and can easily be applied to other modalities (i.e. EEG, EKG, EMG, etc.) by simply redefining the natural language spike descriptions that comprise the Fuzzy-Logic setup.

* Corresponding author at: Temple University, 1947, North 12th Street, #214 Neural Instrumentation Lab, College of Engineering and Architecture, Philadelphia, PA 19122, United States. Tel.: +1 215 204 3795; fax: +1 215 204 5960.
E-mail address: bkintex@temple.edu (K. Balasubramanian).

0165-0270/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.jneumeth.2011.03.016

2. Background

Spike sorting techniques have been studied extensively for over three decades (Lewicki, 1998; Schmidt, 1984; Vibert and Costa, 1979). Semi-automatic spike sorters, i.e. those that require an operator to establish sorting parameters for each individual channel, have become common in research labs, as manual methods have been shown to be notably inconsistent. Wood et al. (2004) studied the error in manual sorting on a synthetic data set comprising neural signals developed using probabilistic waveform models. They reported an average sorting error of 23% false positives and 30% false negatives and emphasized the necessity for automatic spike sorters. Of the several techniques used in semi-automatic sorting, template matching and principal components are used widely in application (Oweiss and Anderson, 2002). Template matching involves extracting direct or transformed spike features of known average waveforms to form templates. Each detected spike event is then compared with these templates and classified, accordingly. Commonly extracted features include simple amplitudes (Lewicki, 1998), Fourier coefficients (Schmidt, 1984), and wavelet coefficients (Hulata et al., 2002; Letelier and Weber, 2000). Principal component analysis (PCA) is a classical linear feature extraction method. A set of aligned spike waveforms of dimension $m \times n$ is used to determine the principal orthogonal axes in the R^n space so that the actual matrix can be represented adequately with reduced dimensionality. These axes capture most of the variability in the matrix, thereby reducing the dimensionality of the data. Each spike waveform is then projected into the principal component space (Diamantaras, 2001).

While the PCA method is considered the benchmark spike sorting process, it is computationally expensive and requires a user-managed training phase for each channel (Chen et al., 2009). This includes redefining the cluster boundaries for each recording session. While this is not a problem for desktop-PC based systems with modest (~500) number of channels, it is a fundamental limitation for systems monitoring large (i.e. >1000) channels implemented in implantable or portable hardware. Gibson et al. (2008) made a comparative analysis on the computational complexity of principal components versus that of the discrete wavelet transform and found that the principal component method was approximately ten times more computationally complex than the wavelet transform method. Their estimate of computational complexity did not include offline training, which involves matrix decompositions that are significantly computationally intensive for implantable hardware. We have provided our own estimate of the computational needs of the PCA method in the Section 5. Linderman et al. (2008) compared the K-means/PCA method of spike sorting with a reported Sahani algorithm, claiming that their algorithm estimated noise-whitened principal component space for real-time spike classification. However, the reported algorithm still needed (i) a training phase to determine the mixture model and (ii) accumulation of spike waveforms to determine the principal components. Their report therefore underscores the fact that PCA-based methods are not completely real-time.

In addition to template matching and PCA/feature extraction methods, researchers have shown the possibility of using Bayesian classification (Bar-Hillel et al., 2006; Wood et al., 2006), multivariate analysis (Shoham et al., 2003) and blind-source sorting with independent component analysis (ICA) (Brown et al., 1998; Madany Mamlouk et al., 2005) as viable techniques for spike sorting. Blind source separation (BSS) refers to the method of de-correlating and extracting statistically independent signals from a mixture of source signals. Membrane potentials from different neurons can be considered independent signal sources, and can be sorted using any of the BSS methods. Brown et al. (1998) used ICA to source sort photo-detector recordings of neural activity in Seaslug *Tritonia* via 448 channels, and reported that ICA outperformed other methods in existence at that time. Madany Mamlouk et al. (2005) tested the applicability of ICA on hippocampal C3 simulations and reported the possibility of using it to solve the "neural cocktail party" problem. In general, ICA implementation involves a series of matrix inversions to de-correlate the spike vectors, a process which is computationally demanding. Furthermore, in both the cases, the process of sorting was either off-line involving human intervention/judgment (as in Brown et al.) and/or required pre-learning (as in Mamlouk et al.). Also, ICA demands spatial correlation for reliable performance (Robert and Sawan, 2007). Recently, methods involving machine learning algorithms that use the extracted features or components to classify spikes have also appeared in literature (Hermle et al., 2004; Horton et al., 2007; Oghalai et al., 1994).

Unsupervised sorting techniques using wavelet coefficients (Quiroga et al., 2004) and non-linear oscillatory models of the spike signal (Aksenova et al., 2003) have also been reported. These methods have their suitability of application and advantaged in their respective domains, but with certain known limitations. The wavelet coefficient method used Haar wavelets and a 4-level decomposition of the simulated signal to determine a set of 64 coefficients out of which 10 coefficients were chosen for clustering. A custom-built clustering technique to determine the number of clusters, in an unsupervised manner, was used in their implementation. Two notable drawbacks of their system are (i) the levels of decomposition needed and the number of coefficients, if not the choice of mother wavelet, needed to define a class of spike can change significantly upon changes in the sampling frequency of the signal data and (ii) the use of intermediate iterative processes to choose the most significant coefficients and the iterative clustering process makes them unsuitable for a real-time implementation on a BMI. The other unsupervised sorting technique assumes that a neural recoding is an oscillatory signal and spikes are perturbations that occur in these oscillatory waveforms. Their system relies on the assumption that spike waveforms form a solution to a set of ordinary differential equations they have defined. This system essentially transforms the neural signal into phase trajectories and each class of neuron form a specific trajectory which is then compared to an unknown spike for classification. Limitations of this system are (i) higher order differential equations are needed to achieve distinguishable trajectories as the noise level increases, (ii) the choice of kernel selection to transform the spike signal onto a feature space (the phase space, according to its authors) and (iii) the need to train the sorter in determining the phase trajectories.

Errors in spike sorting can generally be attributed to (a) amplitude noise caused by background bioelectrical activity, circuit noise or electrode shift and (b) temporal noise manifested as misalignments in the waveforms. While it is generally agreed that the methods discussed above have improved efficiency and accuracy in spike sorting, errors or misclassifications still persist. Unfortunately, even modest spike sorting errors can introduce profound information loss. In analyzing cortical extracellular action potentials in the context of Brain Machine Interfaces, Won et al. (2007) reported a spike sorting error tolerance of only <10% before the transmitted mutual information dropped significantly, resulting in an unusable spike train. They employed Shannon information theory to estimate the mutual information rate and to derive the tolerance limit. By comparison, we report later in this paper that temporal alignment noise contributes to spike sorting errors of greater than 20% in PCA sorting, even between pairs of easily distinguished spikes.

In summary, the following limitations to spike sorting performance persist:

- Susceptibility to spike misalignment (Jung et al., 2006). Spike sorters (especially template-based) are also susceptible to amplitude noise, especially when the background noise shifts the spike maximum by a large extent.

- *Not fully autonomous*: Sorters typically require the operator-assisted accumulation of spike waveforms before actual classification can begin.
- *Computational complexity*: PCA, ICA and other recent methods for spike sorting are computationally complex, rendering them poorly suited for hardware implementation and non-scalable for parallel-processing of large numbers of channels.

The proposed Fuzzy-Logic based system overcomes all three of these issues by using extracted features to characterize detected spike waveforms with a fuzzy score. The fuzzy score is determined by combining the extracted features using a set of rules written in a simple, structural, natural language format. The score can be used in two ways, (a) to sort spike waveforms and (b) to determine the *worthiness* of a detected spike to be included for further processing. The proposed system has, in particular, the following advantages:

- Since the fuzzy score is determined on an individual spike basis, errors due to temporal shift or alignment noise are completely eliminated. The system does not demand any sort of alignment of spikes for classification, making it suitable for real-time implementation. Also, robustness to amplitude errors is inherent to the system.
- The fuzzy rules are common for any number of channels and will not vary with time, and the system requires no human intervention afterwards. This makes the system scalable to multiple parallel channels and completely real-time.
- The algorithm is computationally trivial and can easily be translated into customizable hardware such as FPGA or ASIC.

3. Methods

The fuzzy sorter essentially maps extracted spike features to an output space comprised of linearly separable clusters. The mapping engine, based on fuzzy logic, consists of twelve rules established based on subjective human interpretations of spike waveforms. These rules combine five extracted spike features to map onto the output space. The system was tested by sorting action potential waveforms extracted from extracellular recordings of a macaque, an owl monkey and a rat. Spikes were sorted by clustering their computed fuzzy scores. For comparison, the same data were sorted by clustering PCA weights. Data were tested both with and without spike alignment. The system was implemented and tested using Matlab.

3.1. Fuzzy Inference System

Fuzzy logic is a soft-computing technique that allows numerical representation of human reasoning and uncertainties associated with feature descriptors; a Fuzzy Inference System (FIS) is a decision system based on fuzzy logic. An FIS uses a set of classification rules to map the inputs onto the output space. Internally, each rule possesses a local mapping towards the output space. Assume that there are n decision variables or features (X_i, $i = 1, 2, 3, \ldots, n$) and one class variable (Y) in a given classification problem. If the entire space of each of the input variables is assumed to be U_i, then each of the k fuzzy rule produces,

If (X_1, is μ_{r1}) and (X_2, is μ_{r2}) and ... (X_n, is μ_{rn}) then Y is ρ_r with degree of certainty α_r; where, μ_{im} is the fuzzy set of U_i and ρ_r is the membership function and $\rho_r \in \{\rho_1, \rho_2, \ldots, \rho_k\}$; the factor α_r, $0 \le \alpha_r \le 1$, describes the strength of association of each antecedent to a given consequent in that particular rule. To determine the fuzzy reasoning from the ith rule and n attribute values ($x_1 \quad x_2 \quad x_3 \quad \ldots \quad x_n$), the degree of membership of each attribute is given as $\mu_{r1}(x_1)$, $\mu_{r2}(x_2)$, ..., $\mu_{rn}(x_n)$, and the association in the output space is calculated as,

$$w_r(X) = \alpha_r \prod_{q=1}^{n} \mu_{rq}(x_q) \tag{1}$$

The aggregated output of all the rules is calculated based on the winner-takes-all strategy, given as,

$$\hat{Y} = \rho_z, \quad Z = argmax_{1 \le z \le k} w_z(X) \tag{2}$$

Since spikes can be classified based on a set of features and defined if-then rules, fuzzy logic can be applied for the purpose of spike sorting. While it is not impossible to use the FIS to sort spikes based on only one feature, say the total amplitude alone, the accuracy diminishes rapidly with the addition of noise. The stability can, however, be improved with more extracted features. A set of five features were determined that allowed for optimal performance of the sorter. These chosen features (along with their membership functions) are (a) spike power {low, medium, high}, (b) spike amplitude range {low, medium, high}, (c) negative deflection amplitude {present, not-present}, (d) positive deflection amplitude {present, not-present} and (e) slope of the line connecting the negative minimum and positive maximum points {flat negative, steep, flat positive}. The output variable or the class variable is "Spikiness," with three membership functions {definitely not-a-spike, mostly-a-spike, definitely-a-spike}. The advantage of using fuzzy logic to classify spikes is that the system generates a comprehendible output based on both observable attributes such as range and negative/positive deflection, and derivable attributes such as spike power, slope, etc., needed to arrive at a classification. Furthermore, the system is inherently capable of handling uncertainties associated with the classification.

A schematic of the Fuzzy Inference System (FIS) is shown in Fig. 1. Five characteristic features (attributes) are calculated for each detected spike; these form the crisp inputs. Each of these crisp features is then *fuzzified* by evaluating its degree of membership in multiple fuzzy sets. Unlike crisp sets, fuzzy sets allow a feature to have degrees of membership (μ_i) with multiple sets simultaneously. For example, the fuzzy set {low, medium, high} formed with spike power allows the crisp feature of a given spike to have both 60% membership in the category "medium power" and 40% membership in the "high power" category. A detailed treatment of *Fuzzy Logic* can be found in (Negnevitsky, 2004).

A set of 12 empirically determined natural language rules was devised to determine how a spike's degrees of membership in the various fuzzy sets correspond to its likelihood of being a spike (Table 1). These were selected from a much larger set of possible rules that we considered; only those rules that significantly enhanced the separation of neural populations were retained. It is obvious that there can be a total of 108 rules that can be possibly created using permutations of the five extracted features. Using a three step elimination process the listed 12 rules were retained while discarding the remaining rules. The steps of elimination are given as,

1. Elimination of self-contradicting rules. For example, a rule such as "If POWER is High and if RANGE is High and if NEGATIVE DEFLECTION is Not Present and if POSITIVE DEFLECTION is Not Present and if SLOPE is Steep" cannot occur to map to any output space because a spike with high power and high range cannot exist without either of the deflections present in it. Similarly, other rules that self-contradict were eliminated.
2. Elimination of rules that generate same consequent area. If two rules were to map the input space to the same consequent area, the rule that offered more human reasoning was retained while eliminating the other.

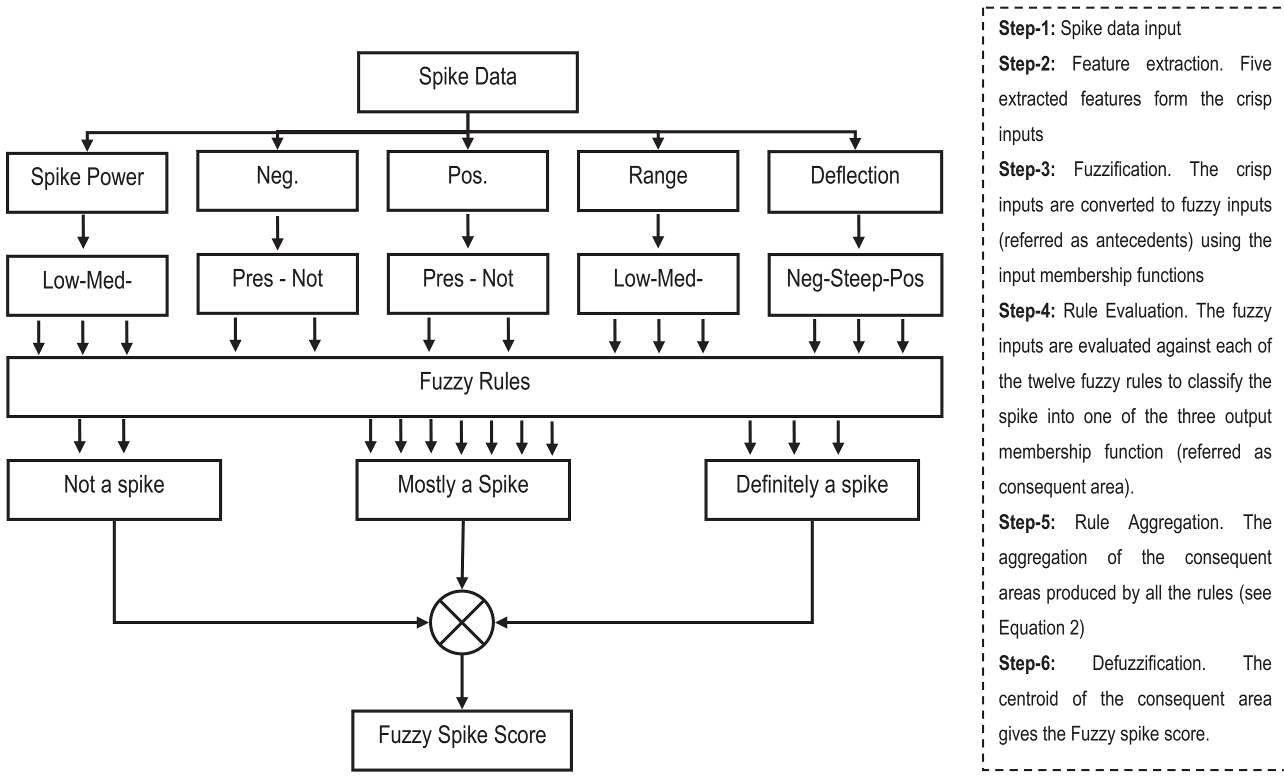

Fig. 1. Schematic representation of the Fuzzy Inference System (FIS). A set of 12 rules operates on the fuzzy inputs to map them to an output space, which in this case are the output membership functions. Refer to the side note for the steps involved in calculating the fuzzy score of a given spike using the FIS.

3. Elimination of rules that offer lesser discrimination when compared to their counterpart rules that point to the same output membership function. For example, a rule such as “If POWER is High and if RANGE is Low and if NEGATIVE DEFLECTION is Not Present and if POSITIVE DEFLECTION is Present and if SLOPE is Flat Positive” offers lesser discrimination towards the output class “Definitely Not a Spike” when compared to the rule “If POWER is Low and if RANGE is Low and if NEGATIVE DEFLECTION is Not Present and if POSITIVE DEFLECTION is Present”. Hence, the latter rule is retained.

For the steps 2 and 3 of the elimination process, extracellular neural spikes (600 spikes × 4 classes = 2400 spikes in total) from a species of rat were used. These data were different from the test signals, discussed in the following section, that were used to evaluate the Fuzzy Spike Sorter. Ours are not the only possible rules for spike sorting, but rather reflect a logical rationale based on the salient spike features needed for classification. These are general rules that will apply to a wide range of extracellular spike data. These rules need to be redefined only when the spike data is from a completely different kind such as the electromyogram or electrcorticogram spikes. Each rule determines the degree of membership in one of the output fuzzy sets, which in turn defines that rule’s *consequent area*. Refer to Fig. 1, in which a total of seven rules will produce consequent areas in the “mostly-a-spike” category, two rules in “not-a-spike” and three rules in “definitely-a-spike”. Finally, the aggregated contribution of all the rules determines the total output consequent area for a given spike, which is then *defuzzified* to produce the fuzzy score.

The Fuzzy Inference System is modular in design such that any additional classification criterion can be added without altering the existing setup. For example, if need arises to classify spikes based on their “phasiness” (i.e. biphasic or triphasic) in addition to the spikiness index, modular rules can be added independently of the existing rules. From Fig. 1, it can be seen that with additional features and rules, different kinds of output classifications can be included as additional modules. Here, two additional variables in conjunction with three of the existing variables and five modu-

Table 1
Fuzzy rules for spike characterization.

Rule	If POWER is	And if RANGE is	And if NEGATIVE DEFLECTION is	And if POSITIVE DEFLECTION is	And if SLOPE is	Then SPIKE is
1	Low	Low	Not Present	Not Present	–	Definitely not a spike
2	High	High	Present	Present	Steep	Definitely a spike
3	High	High	Present	Present	Flat negative	Mostly a spike
4	High	Med	Present	–	Steep	Mostly a spike
5	High	Med	–	Present	Steep	Mostly a spike
6	Med	Low	Not Present	Not Present	Flat positive	Definitely not a spike
7	High	High	Present	Present	Flat positive	Definitely a spike
8	High	Med	Present	Present	Flat positive	Mostly a spike
9	Med	Med	Present	Present	Flat positive	Mostly a spike
10	Med	High	Present	Present	Flat positive	Mostly a spike
11	Med	Med	Present	Present	Steep	Mostly a spike
12	Med	High	Present	Present	Steep	Definitely a spike

Table 2
Fuzzy rules for spike phase characterization.

Rule	If NEGATIVE DEFLECTION is	And if FORWARD POSITIVE DEFLECTION is	And if BACKWARD POSITIVE DEFLECTION is	And if FORWARD SLOPE is	And if BACKWARD SLOPE is	Then SPIKE PHASE is
13	Present	Present	Present	Steep	Flat negative	Partly triphasic
14	Present	Present	Present	Flat positive	Steep	Partly triphasic
15	Present	Present	Present	Steep	Steep	Triphasic
16	Present	Present	Not Present	–	–	Biphasic
17	Present	Not Present	Present	–	–	Biphasic

lar rules (see Table 2) were devised to classify spike phasiness as "biphasic", "partly triphasic" and "triphasic".

3.2. Test signals

The test signals used to evaluate the spike sorters were in vivo extracellular cortical recordings sampled at 31.25 kHz from three animals: an owl monkey, a macaque, and a rat. For each animal, signals were identified that were comprised of spikes from two, three or four neurons. This yielded a total of nine signals (3 animals × 3 neuron groupings). Spikes from each of these signals were detected using a standard threshold method and were accumulated as separate banks. Each of these banks was then manually sorted by three experts in order to establish a ground truth. Both the proposed fuzzy logic spike sorter and the benchmark PCA sorter were evaluated on the spikes in these nine data banks.

4. Sample fuzzy score calculation

The functionality of the Fuzzy Inference System is further explained by detailing part of a fuzzy score calculation for the sample spike shown in Fig. 2. The five crisp inputs are calculated from the spike samples and are then fuzzified by determining their degrees of membership, μ, in the input fuzzy sets. The fuzzy sets for each of the five crisp inputs are shown in Fig. 3a. For example, the spike power is computed to be 0.35. The spike therefore has a membership of 0.40 in the "Low Power" set, 0.73 in the "Medium Power" set and 0.00 in the "High Power" set. Table 3 lists all five crisp input values for the sample spike and the resulting membership values in the corresponding fuzzy sets.

The next step is rule evaluation and aggregation. Rule 9 is evaluated here as an example: If power is medium and range is medium and negative deflection is present and positive deflection is present and slope is flat positive then the signal is "Mostly a spike." The rules of Fuzzy Logic stipulate that "and" (i.e. intersection) conjunctions are computed by taking the minimum membership value, whereas "or" (i.e. union) conjunctions are evaluated as the maximum membership value. In the case of Rule 9:

$$\mu_{Pwr \cap Rng \cap NDef \cap PDef \cap Slope} = \min[\mu_{Pwr}, \mu_{Rng}, \mu_{NDef}, \mu_{PDef}, \mu_{Slope}]; \\ = \min[0.73, 1.00, 0.31, 0.52, 0.52] = 0.31 \quad (3)$$

Thus, the value 0.31 (see Fig. 3a) refers to the degree of membership in the output fuzzy membership function "Mostly a Spike", and the area of the curve below the value 0.31 of this output membership function is referred to as the consequent area produced by Rule 9. Similarly, each fuzzy rule produces a consequent area that refers to the degree of membership in the respective output membership function. In the process of rule aggregation, the maximum consequent area under each output membership function is used to determine the aggregated consequent area (according to Eq. (2)). A typical output fuzzy set made of aggregated consequent areas is shown in Fig. 3b. The aggregated area comprises the partial area under the curves defined by degrees of membership 0.31 (with "Mostly a Spike") and 0.07 (with "Definitely a Spike"). Consequent areas from other rules were either zeros or less than that of 0.31 or 0.07 in the respective output membership set.

From the aggregated consequent area of the output space, a crisp number associated with a given spike can be calculated using several defuzzification methods (Negnevitsky, 2004). The most common technique is the "Centroid" method and is used here. Essentially, it is the center of gravity of the aggregated consequent area that is determined and used as fuzzy score for the given spike. Thus, the *defuzzified* output, or fuzzy score, for the sample spike is 0.60.

Similarly, to show the modular nature of the system, the phasiness index of the spike is calculated with the additional five rules given in Table 2. While Rule 16 results in a consequent area in the "biphasic" output membership function for the given spike, all the other rules (rules 13, 14, 15 and 17) result in zero consequent area. Thus, the consequent area of Rule 16 becomes the aggregated area in the phasiness output space for the sample spike, which is shown in Fig. 3c. The *defuzzified* output of phasiness index is 0.15, which categorizes the spike as "biphasic". Thus, a given spike can be classified based on the spikiness, and also with additional criterion such as the phasiness as in the cases of "triphasic" spikes.

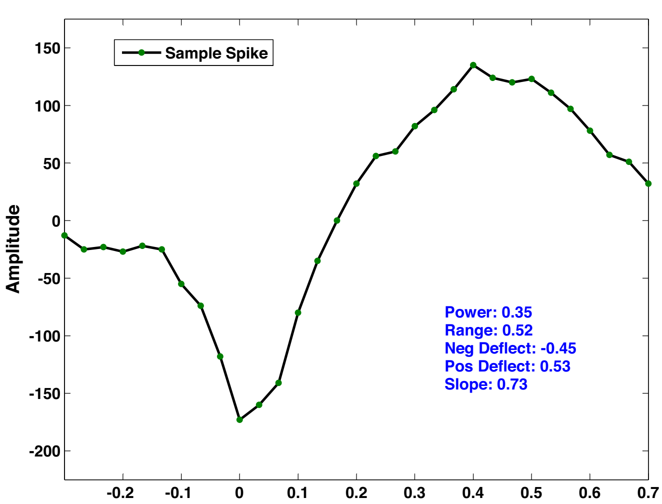

Fig. 2. A sample spike with its extracted crisp feature values. This spike has moderate "spiky" qualities and subsequently receives a fuzzy score of 0.6.

Table 3
Crisp and fuzzified extracted features for a sample spike.

Crisp extracted feature values	Fuzzified extracted feature values
Spike power 0.35	{low, medium, high} ⇒ {0.40, 0.73, 0.00}
Spike amplitude range 0.52	{low, medium, high} ⇒ {0.02, 1.00, 0.07}
Negative deflection −0.45	{present, not present} ⇒ {0.31, 0.57}
Positive deflection 0.53	{present, not present} ⇒ {0.52, 0.35}
Gradient slope 0.73	{flat negative, steep, flat positive} ⇒ {0.00, 0.52, 0.52}

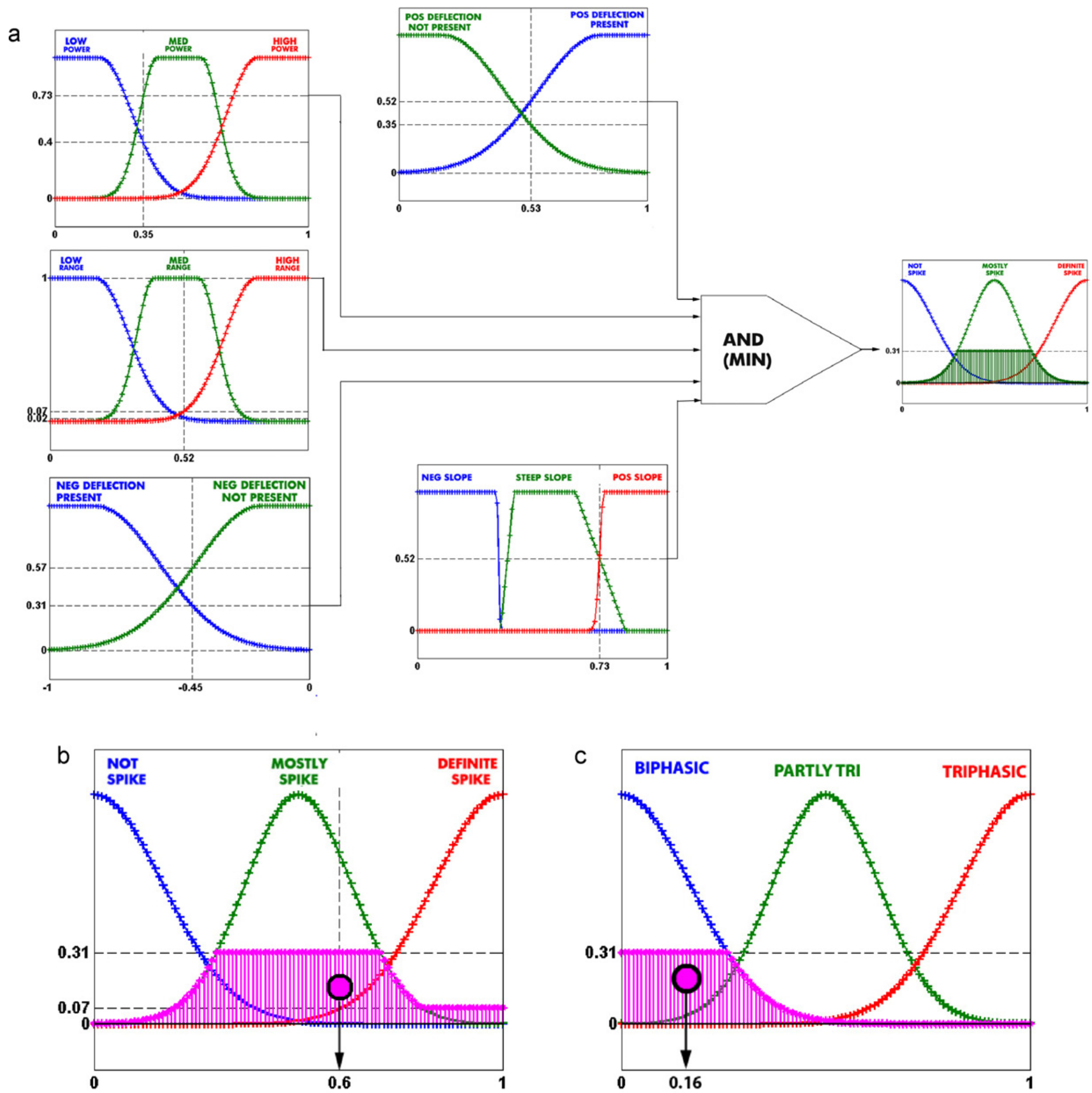

Fig. 3. Fuzzy Inference Engine specifically developed for extracellular neural action potentials. (a) Five crisp spike features are extracted and fuzzified by determining their degree of membership in various fuzzy sets. The set memberships can then be combined according to natural language rules. In the case of Rule 9 (depicted here) the set memberships are combined by taking their minimum value. (b) Aggregated consequent areas for the three output membership functions, and their centroid, which is the final fuzzy score. (c) Aggregated consequent area for the spike phasiness, and the centroid is 0.16 thereby making the spike biphasic.

5. Results

Each bank of spike waveforms was subjected to 50 trials and during each trial a random set of 200 spike waveforms was chosen to be sorted using the proposed Fuzzy Inference System. For each trial, the fuzzy scores were calculated for all of the 200 constituent spikes and these scores were autonomously clustered using a generic Fuzzy C-means clustering algorithm. The resulting clusters were compared against the true spike identities (established using manual clustering) to determine the Percent Correctly Classified (PCC). The procedure was repeated for all nine data banks (see Section 3.2). For comparison, the same test signals were also sorted by clustering the first two principal components (PCA). The principal components were extracted using a standard routine in which the test spike vectors arranged as a matrix were mean subtracted, covariance estimated and rotated using Jacobian method to determine the Eigen vectors. Scalar multiplication of each spike vector with the leading two Eigen vectors produced the two principal components. In this manner, we evaluated spike sorting under conditions where a given signal contained either of two, three, or four neurons.

The sorting results presented in Fig. 4 show sorting performance (PCC) of fuzzy sorter and PCA sorter for the classification of n-neurons ($n = 2, 3, 4$) from the three different animals. A 3-way ANOVA was conducted to test the statistical significance of the differences in PCC caused by factors such as the number of neurons, type of the animal and the type of sorter. Three important inferences were be drawn from the ANOVA results:

(a) There is no significant difference in the performance of the sorters due to animals (p-value = 0.6201).
(b) Significant difference in performance is observed due to the number of neurons (p-value = 0.0032).
(c) The performance of Fuzzy Spike Sorter and PCA sorter are not significantly different (p-value = 0.1235).

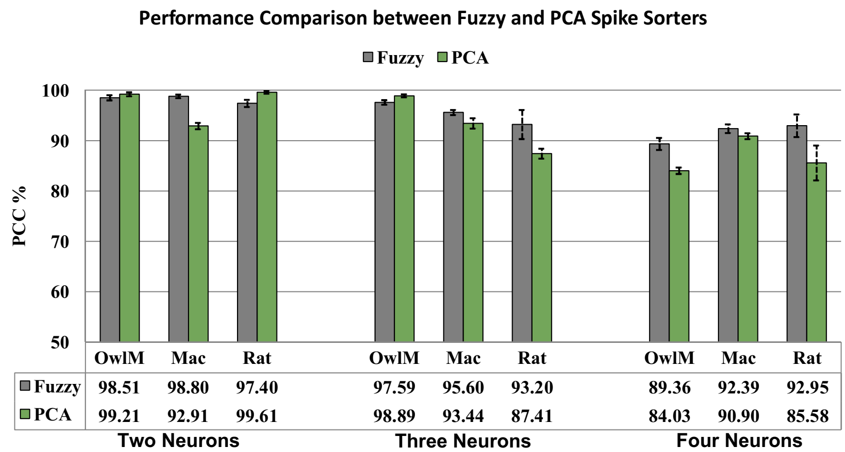

Fig. 4. Comparison of mean percent correct classification (PCC) by Fuzzy Spike Sorter and PCA Sorter on the spike data from the three different animals. Three-way ANOVA showed that the difference in performance of fuzzy sorter caused by change in animal is statistically insignificant; however, the performance deteriorated as the number of neuron classes increased from two to four. Except one case (four neuron classification of Owl Monkey recording), fuzzy sorter reached the required 90% PCC in all the other cases.

This validates that the Fuzzy Spike Sorter functions as effectively (under these conditions) as the gold-standard benchmark PCA sorter.

To demonstrate the shift-invariance property of the fuzzy sorter, the same set of spike waveforms were again sorted using precisely the same sorting procedures as before but without being temporally aligned to the minimum sample value. Spikes were misaligned in time by random offsets of up to 0.3ms (i.e. ~10 samples). No significant difference was found between the fuzzy scores of aligned and misaligned spikes, since the fuzzy sorting system assigns the scores independent of the other spikes. On the contrary, PCA's performance dropped 27% while sorting the misaligned spikes.

To better illustrate the FIS robustness to spike misalignment, we present results from a single test signal comprising 100 spikes from each of two neurons. Fig. 5a and b present the sorting results from a single sample test signal with 100 spike waveforms. This particular set of spikes was chosen because the two component action potentials are both qualitatively and quantitatively quite different and therefore ought to be easily sorted. Fig. 5 shows the results for the aligned signals for both the sorting methods tested. Both

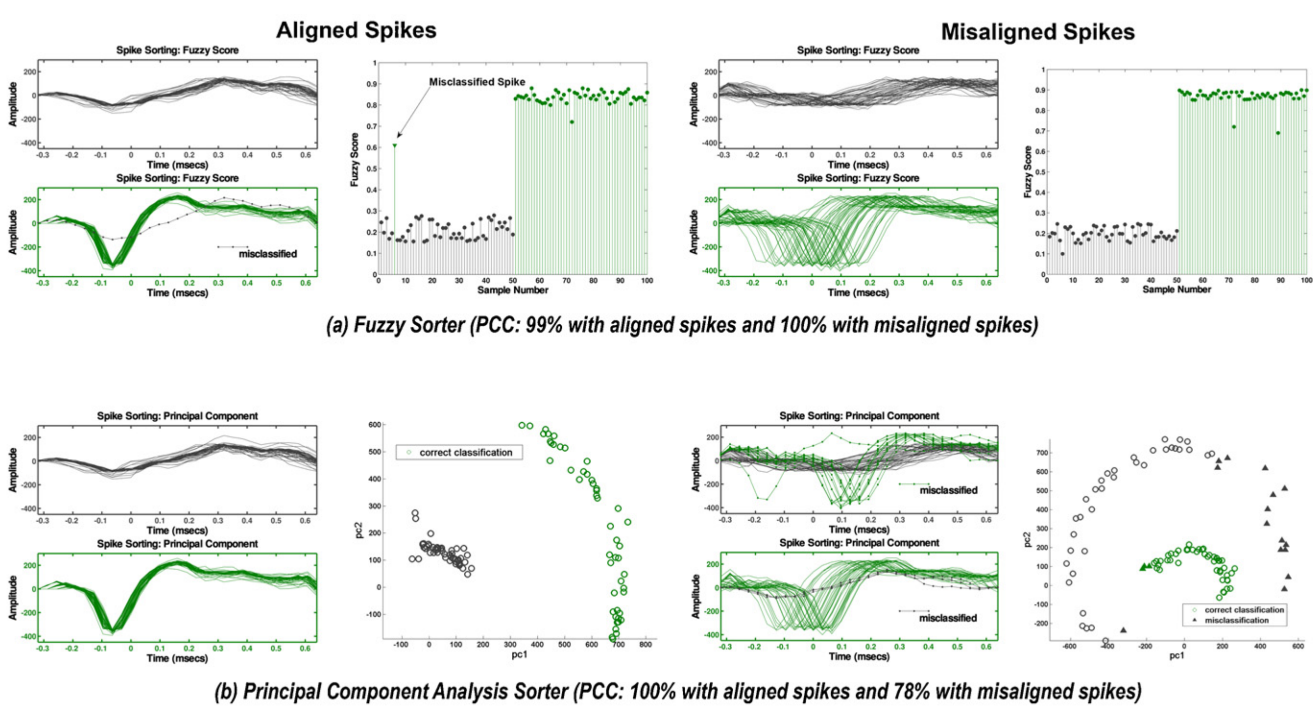

Fig. 5. Spike clustering for a single sample test signal. (a) (top left) Plots show the sorted spikes and cluster space sorted by fuzzy sorter in cases of aligned and (top right) misaligned spikes. In both the cases, fuzzy sorter produced linearly separable clusters. (b) Sorter spikes and cluster space generated by PCA sorter (bottom left shows aligned spikes and bottom right shows misaligned spikes, subjected to PCA based sorting). With aligned spikes, PCA produced linearly separable cluster space, but failed in case of misaligned spikes. Spikes and feature markers are correspondingly color coded; a grey circle corresponds to a correctly sorted grey spike, whereas a grey triangle corresponds to a green spike incorrectly sorted as a grey one.

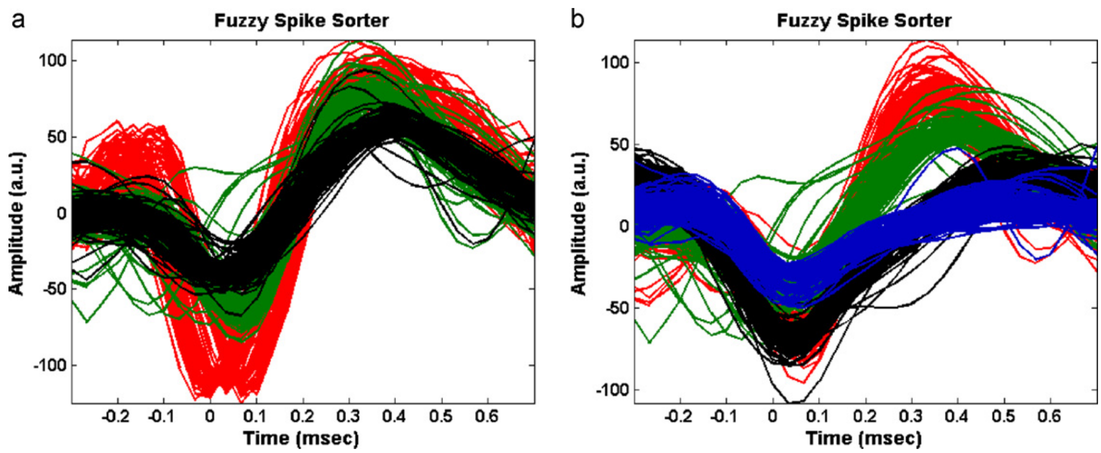

Fig. 6. Fuzzy Spike Sorter results for classification of (a) three neurons and (b) four neurons.

fuzzy sorter and PCA produce two distinct feature sets which are easily clustered; sorting is perfect or near perfect in both cases. To compare the robustness of the two techniques towards spike misalignment, sorting was performed on the same data set as before but with random temporal misalignment. Results show that, even with misaligned data, the fuzzy sorter continues to produce distinct feature sets that can be easily and accurately clustered; in this case, all 100 spikes were correctly sorted. In contrast, Fig. 5b illustrates how the PCA sorter is poorly suited to sort the misaligned spikes. The temporal offsets between the spikes greatly affect the proper computation of the principal component scores, thereby producing a feature space that is not easily partitioned by the Fuzzy C-Means clustering algorithm; in this case, only 78% of the spikes were correctly sorted (Fig. 5b).

To demonstrate the noise immunity of the fuzzy sorter, fuzzy scores of 100 spikes representing two neuron classes were calculated at different noise levels. In the absence of noise, the centroids of the fuzzy scores of these two spike classes were 0.28 and 0.75, respectively. The signal to noise ratio (SNR) was varied by adding noise comprised of far-field extracellular neural signals sampled at the same frequency as the spikes themselves. Fig. 7 shows the resulting fuzzy scores as a function of SNR. The addition of noise to the spike signals increased the signal power as well the signal amplitude, therefore resulting in an increase in the fuzzy score of a typical spike. However, the distance between the centroids of the two clusters was reduced from 0.48 to 0.1, as the noise level increased. At a very low SNR (~1), the cluster centroids were separated only by 0.1, resulting in a PCC of ~65%.

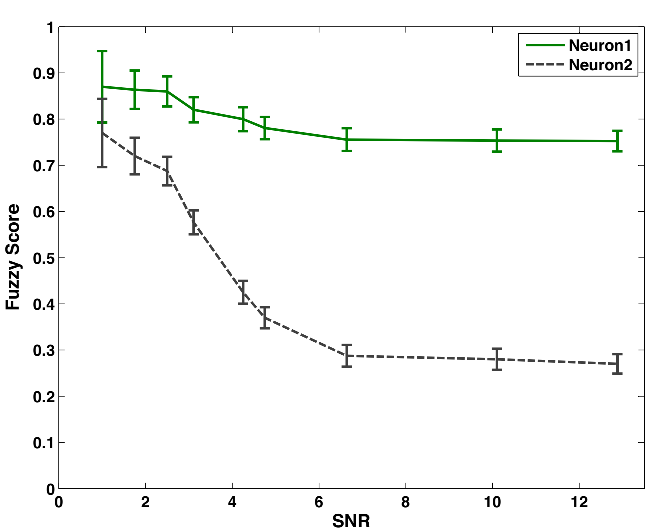

Fig. 7. Cluster centroids of the fuzzy scores of two neurons under different SNR are plotted. At higher SNR, the centroids are widely separated, and as the noise level increased (SNR ~1), the cluster separation narrowed down to 0.1 fuzzy score units resulting in a PCC of ~68%.

A comparison of the computational complexity of the fuzzy sorter and PCA shows the computations needed for a typical implementation of these techniques (Table 4). The PCA implementation can be divided as (i) the training phase, including the Eigen vector estimations and (ii) the testing phase during which the principal coefficients are determined. The processes of mean estimation, covariance calculation and matrix rotation all need spike accumulation and alignment to perform properly. We developed a typical hardware implementation protocol of the PCA method which involved (a) estimation of mean of each spike vector and storing the mean-subtracted spike vector, (b) covariance calculation of the spike matrix and storing the resultant upper triangular matrix in memory and (c) estimation of Eigen vectors using singular value decomposition obtained by the Jacobian process (Bravo et al., 2006). Table 4 compares the computational resources needed for the PCA and fuzzy sorter for a 31-sample spike. The resources needed for the PCA method include the computations to calculate the principal coefficients as well the memory requirements for the intermediate matrix storage. If the recording parameters are altered, the entire process of Eigen vector calculation must be repeated for the PCA technique to work. On the other hand, the membership functions of the fuzzy sorter are normalized (note in Fig. 3a that the membership functions range between 0 and 1) so that any new recording with different recording parameters can still be processed using the same set of membership functions, but with a normalizing unit (not discussed here) prior to feature extraction. The stark difference in the number of operations suggests that the fuzzy sorter would be better suited for implementation in a high channel count application (such as a brain machine interface) where computational resources, power, and/or physical hardware space may be limited.

6. Discussion

Spike sorting is an important and ubiquitous component of biomedical signal processing. In a number of emerging applications, particularly ensemble neural signal processing, spike sorting must occur simultaneously in ever-increasing numbers of channels. Hence, there is a real need for spike sorting algorithms that can operate autonomously and that can be implemented with a minimum of hardware resources. Furthermore, spike sorter accu-

Table 4
Comparison of various computational operations needed for the PCA technique and the fuzzy spike sorter method. The training phase of PCA is an intermittent process demanding computationally expensive matrix operations. The calculation of principal coefficients alone uses relatively more operations as compared to the fuzzy sorter.

PCA function	Operations					FSS function	Operations				
	Add	Sub	Mult	Comp and data shift	Memory (bits)		Add	Sub	Mult	Comp	LUTs (bits)
Spike alignment[a]				1700 and 3100	$100 \times 31 \times 16$						
Mean	3000	400	200		$100 \times 31 \times 16$	Feature extraction	32	9	39	2	–
Covariance	3200	6400	3200		4550×16	Fuzzification	–	–	–	–	13×1212
Singular value decomposition (Jacobi method with five sweeps)	9900	9900	4950		$100 \times 100 \times 16$						
$100 \times 1 \times 16$	Rule Map and aggregation	–	–	–	48	–					
Principal coefficients (three coefficients) per spike	90	–	93		$100 \times 3 \times 16$	Defuzzification	17	18	2	–	2×6912

[a] Assuming that the minimum point occurs within the first 18 samples.

racy is vital, since even a low rate of incorrectly sorted spikes can markedly degrade the underlying encoded information (Won et al., 2007). In this context, the fuzzy sorter presented here is well suited for systems with large numbers of channels such as might be found in a brain computer interface with hundreds or even thousands of electrodes. In such existing systems, the user must typically manually set spike sorting parameters for each individual channel which is time consuming and potentially error prone. In contrast, our design can simply be turned on and used. Our design is also computationally negligible, meaning that a simple low-power processor or embedded logic core could potentially handle dozens or even hundreds of channels.

Alignment of spikes to a common point is a fundamental requirement for PCA sorting to work. Furthermore, PCA can operate only after a set of accumulated training spikes is collected. These two fundamental limitations of PCA sorting make it unsuitable for completely autonomous, real-time processing. However, since our technique evaluates each spike individually, computationally expensive processes such as spike accumulation and alignment are entirely eliminated. Our system is therefore well-suited for real-time processing of multichannel spike data.

We have tested our system extensively with in vivo extracellular neural recordings from three different animals that are comprised of combination of neural and electrical noises along with the action potential spikes. Sorting performance has been quantified under two main conditions: with and without spike alignment. Further, since our system uses a feature extraction procedure that is normalized for spike lengths, the setup is immune to sampling rate changes. We have also compared our results to spike sorting performed using clustered principal components under both the conditions. With the case of aligned spikes, we have proven that our fuzzy sorter is as effective as or better in certain cases than the benchmark principle component sorter (see Fig. 4). ANOVA results showed that the same set of rules, without any change in the membership functions or parameters, has produced statistically similar performance across three different animals. This shows that the Fuzzy Spike Sorter has the potential for being more general in terms of applicability for sorting spikes from various sources. Additionally, one of the main merits of our system was observed when sorting misaligned spikes. For those signals, the fuzzy sorter significantly outperformed the principal component sorter in all the cases, even for clusters of spikes that were clearly different and that should therefore have been easily sorted. Whereas the fuzzy sorter performance was not affected by the temporal misalignment, the PCA sorter dropped to an unacceptable PCC of 78%. The limitations of spike sorting performance can be directly observed via the extracted feature clusters (see Fig. 5b). The cluster spaces were linearly separable in both the cases when aligned spikes were sorted. However, when the misaligned spikes were used, the fuzzy sorter clusters remained linearly separable, whereas the PCA cluster space was not. Our system was also tested towards classifying neurons which originated from three or four sources (see Fig. 6a and b for illustration) and is shown to be performing adequately within the required limit of 90% PCC in most of the cases. Finally, the proposed system was shown to be modular by design in which additional constraints (via fuzzy rules) were imposed to classify spikes based on their phase variations.

Hardware realization of the proposed system in customizable hardware is relatively straightforward. In contrast, for the PCA method, both the Eigen value decomposition process and the need to collect several hundred training spikes per channel is cumbersome for hardware realizations. Since our system neither involves any matrix decompositions nor accumulates spikes for training, it is computationally inexpensive. For a 31-sample spike, our system requires approximately 36 multiplications, 36 additions and 18 memory access operations (i.e. Look-Up Table access), which is a fraction of the computational operations required of the PCA method (see Table 4). A proof-of-concept implementation of the entire fuzzy system on a Xilinx® Virtex-5 FPGA board utilized around seven percent of the resources available on the FPGA. Scaling the fuzzy system to several hundred channels will depend on the FPGA's clock frequency and a time-multiplexed data acquisition system rather than the fuzzy system itself.

7. Conclusion

Fuzzy scoring is demonstrated as a viable method for spike sorting. It has been shown that the Fuzzy Logic-based feature extraction method and the fuzzy score can be used to form an efficient spike sorting method. The technique of Fuzzy Logic-based spike sorting has performed equivalently or better in certain cases than the method of principal components, and has been proven robust against temporal alignment errors. Also, the fuzzy score computation does not demand accumulation or alignment of spikes making is feasible for real-time applications. Moreover, the system is shown to be modular which allows addition of criteria without disturbing the existing setup. The use of fuzzy score is not limited to spike sorting, but could be valuable as a general coefficient to estimate the spikiness of any given spike. This could allow one to use the score as a criterion towards inclusion or exclusion of a detected spiking event in the subsequent processes.

References

Aksenova TI, Chibirova OK, Dryga OA, Tetko IV, Benabid AL, Villa AEP. An unsupervised automatic method for sorting neuronal spike waveforms in awake and freely moving animals, methods. Recording in Awake and Freely Moving Animals 2003;30(June (2)):178–87.

Bar-Hillel A, Spiro A, Stark E. Spike sorting: Bayesian clustering of non-stationary data. Journal of Neuroscience Methods 2006;157(2):303–16.

Bravo I, Jimenez P, Mazo M, Lazaro JL, Gardel A. Implementation in Fpgas of Jacobi method to solve the Eigenvalue and Eigenvector problem. FPL'06. International Conference. Field Programmable Logic and Applications 2006(August):1–4.

Brown GD, Yamada S, Lübben H, Sejnowski TJ. Independent component analysis of optical recordings from the Seaslug Tritonia. In: Paper Presented at the Proceedings of the 5th Joint Symposium on Neural Computation; 1998.

Chen T-C, Chen K, Wentai L, Chen L-G. On-chip principal component analysis with a mean pre-estimation method for spike sorting. In: Paper Presented at the Circuits and Systems, ISCAS 2009, IEEE International Symposium on; 2009.

Diamantaras IK. Neural networks and principal component analysis. In: Hu YHaH, Jenq-Neng, editors. Handbook of Neural Network Signal Processing. Florida: CRC Press; 2001.

Gibson S, Judy JW, Markovic D. Comparison of spike-sorting algorithms for future hardware implementation. In: Paper Presented at the Engineering in Medicine and Biology Society, EMBS 2008, 30th Annual International Conference of the IEEE; 2008.

Hermle T, Schwarz C, Bogdan M. Employing ICA and SOM for spike sorting of multielectrode recordings from CNS. Journal of Physiology-Paris 2004;98(46):349–56.

Horton PM, Nicol AU, Kendrick KM, Feng JF. Spike sorting based upon machine learning algorithms (SOMA). Journal of Neuroscience Methods 2007;160(1):52–68.

Hulata E, Segev R, Ben-Jacob E. A method for spike sorting and detection based on wavelet packets and Shannon's mutual information. Journal of Neuroscience Methods 2002;117(1):1–12.

Jung HK, Choi JH, Kim T. Solving alignment problems in neural spike sorting using frequency domain PCA. Neurocomputing 2006;69(7–9):975–8.

Letelier JC, Weber PP. Spike sorting based on discrete wavelet transform coefficients. Journal of Neuroscience Methods 2000;101(2):93–106.

Lewicki MS. A review of methods for spike sorting: the detection and classification of neural action potentials. Network: Computation in Neural Systems 1998;9(4):R53–78.

Linderman MD, Santhanam G, Kemere CT, Gilja V, O'Driscoll S, Yu BM, et al. Signal Processing Challenges for Neural Prostheses. IEEE Signal Processing Magazine 2008;25(1):18–28.

Madany Mamlouk A, Sharp H, Menne KML, Hofmann UG, Martinetz T. Unsupervised spike sorting with ICA and its evaluation using GENESIS simulations. Neurocomputing 2005;65–66:275–82, doi:10.1016/j.neucom.2004.10.019.

Negnevitsky M. Artificial Intelligence: A Guide to Intelligent Systems. 2nd edition Addison Wesley; 2004.

Nicolelis MAL. Actions from thoughts. Nature 2001;409(6818):403–7.

Oghalai JS, Street WN, Rhode WS. A neural network-based spike discriminator. Journal of Neuroscience Methods 1994;54(1):9–22.

Oweiss KG, Anderson DJ. Spike sorting: a novel shift and amplitude invariant technique. Neurocomputing 2002;44–46:1133–9, doi:10.1016/S0925-2312(02)00436-8.

Quiroga RQ, Nadasdy Z, Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Computation 2004;16(August (8)):1661–87.

Robert PY, Sawan M. An independent-component-analysis-based time-space processor for the identification of neural stimulation sources. In: Paper Presented at the Engineering in Medicine and Biology Society, EMBS 2007, 29th Annual International Conference of the IEEE; 2007.

Schmidt EM. Computer separation of multi-unit neuroelectric data: a review. Journal of Neuroscience Methods 1984;12(2):95–111.

Shoham S, Fellows MR, Normann RA. Robust, automatic spike sorting using mixtures of multivariate t-distributions. Journal of Neuroscience Methods 2003;127(2):111–22.

Vibert JF, Costa J. Spike separation in multiunit records: a multivariate analysis of spike descriptive parameters. Electroencephalography and Clinical Neurophysiology 1979;47(2):172–82.

Won D.S. DS, Tiesinga PHE, Henriquez CS, Wolf PD. An analytical comparison of the information in sorted and non-sorted cosine-tuned spike activity. Journal of Neural Engineering 2007;4:322–35.

Wood F, Black MJ, Vargas-Irwin C, Fellows M, Donoghue JP. On the variability of manual spike sorting. IEEE Transactions on Biomedical Engineering 2004;51(6):912–8.

Wood F, Goldwater S, Black MJ. A non-parametric Bayesian approach to spike sorting. In: Paper Presented at the Engineering in Medicine and Biology Society, EMBS'06, 28th Annual International Conference of the IEEE; 2006.