# Feature Extraction Using Extrema Sampling of Discrete Derivatives for Spike Sorting in Implantable Upper-Limb Neural Prostheses

Majid Zamani, *Student Member, IEEE*, and Andreas Demosthenous, *Senior Member, IEEE*

***Abstract*—Next generation neural interfaces for upper-limb (and other) prostheses aim to develop implantable interfaces for one or more nerves, each interface having many neural signal channels that work reliably in the stump without harming the nerves. To achieve real-time multi-channel processing it is important to integrate spike sorting on-chip to overcome limitations in transmission bandwidth. This requires computationally efficient algorithms for feature extraction and clustering suitable for low-power hardware implementation. This paper describes a new feature extraction method for real-time spike sorting based on extrema analysis (namely positive peaks and negative peaks) of spike shapes and their discrete derivatives at different frequency bands. Employing simulation across different datasets, the accuracy and computational complexity of the proposed method are assessed and compared with other methods. The average classification accuracy of the proposed method in conjunction with online sorting (O-Sort) is 91.6%, outperforming all the other methods tested with the O-Sort clustering algorithm. The proposed method offers a better tradeoff between classification error and computational complexity, making it a particularly strong choice for on-chip spike sorting.**

***Index Terms*—Discrete derivatives, extrema sampling, feature extraction, implantable neural interface, neural recording, online sorting, spike sorting.**

## I. Introduction

ALTHOUGH mankind has provided artificial limbs for amputees for millennia, modern upper limb prostheses are far from ideal. There have been major advances in the design of life-like hands that have many joints and look natural. The current state-of-the-art uses myoelectric signals recorded from the skin which is unreliable due to noise caused by residual muscles in the stump. The patient must learn to operate the hand in a slow and unnatural way handicapped by the fact that there is no tactile feedback. Using motor cortical signals to control hand prostheses is an alternative approach [1], [2] but is very invasive. An appropriate site for detecting the intended movement is in the peripheral nerves of the amputation stump. At that site, there are many motor neurons corresponding to the lost muscles, and sensory afferents which could provide sensory feedback and reduce phantom pain. Recently Tombini *et al.* [3] demonstrated acutely the use of four intrafascicular (nerve-penetrating) probes as a bidirectional interface to a robotic hand. In that study, each probe contained eight electrodes detecting signals with amplitudes of the order of a few microvolts from nearby nerve fibres. Signals were transferred via percutaneous cables and processed external to the body. Having an implantable neural interface with more electrodes could be advantageous.

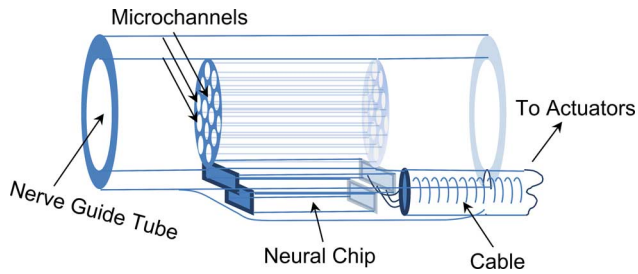


Fig. 1. Sketch of a microchannel nerve interface with a neural chip intimately connected. Chip provides amplification and spike sorting.

One other type of nerve-electrode interface suitable for long-term use with peripheral nerves is the microchannel nerve interface [4]. It combines features of both sieve and cuff electrode designs by confining axons in approximately 100 $\mu$m diameter microchannels. Since the microchannels greatly increase the extracellular resistance, a natural amplification of recordable voltage potentials is provided. Fitzgerald *et al.* [5] have shown that peripheral nerves grow into microchannels, giving large ($\sim$ 100 $\mu$V) action potentials (spikes), characteristic of the active neurons, and are consistent for months. Using such an implantable neural interface might offer the possibility of controlling upper-limb prostheses with many actuators, thus enabling more natural and wide ranging movements rather than just a few basic grasps. The concept of an "active" microchannel nerve interface is shown in Fig. 1 where a custom integrated circuit is mounted very close to the microchannels. The chip must amplify and reduce the data rate needed to represent the many spike signals and perform real-time clustering of spike characteristics. Stimulation may also be necessary for feedback. To achieve real-time online (on-chip) processing it is important to design an accurate feature extraction with low-power hardware resources, suitable for implantable devices (including, but not limited to, the microchannel nerve interface).

Manuscript received July 30, 2013; revised January 08, 2014 and February 20, 2014; accepted February 24, 2014. Date of publication March 05, 2014; date of current version July 03, 2014. This work was supported by a University College London Graduate School Research Studentship.

The authors are with the Department of Electronic and Electrical Engineering, University College London, WC1E 7JE London, U.K. (e-mail: m.zamain@ucl.ac.uk; a.demosthenous@ucl.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNSRE.2014.2309678

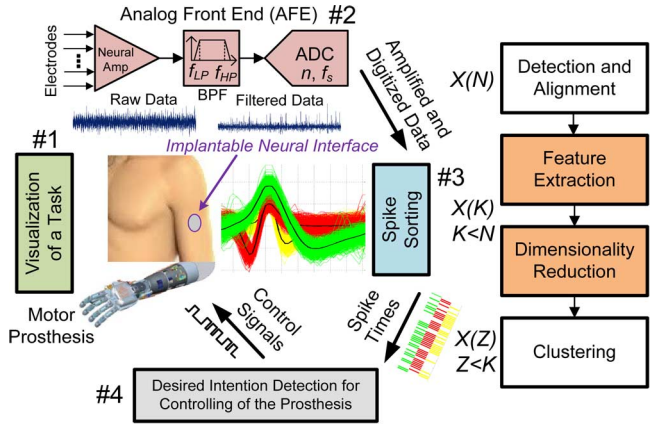


Fig. 2. Overview of closed-loop control of a prosthetic hand.

With the aim of developing an energy-efficient neural recording and spike sorting chip for the targeted application, this paper reports a new feature extraction method based on extrema analysis (positive and negative peaks) of spike shapes and their discrete derivatives [6] with different sampling intervals. The proposed method runs in real-time and does not require any offline training. Compared to other methods it offers a better tradeoff between accuracy and computational complexity using online sorting. Unlike in other systems, the spike sorting procedure eliminates multiplications which are computationally expensive, power hungry and require appreciable silicon area. These are important features in implantable devices particularly where there is a high channel count.

The paper is organized as follows. Section II describes prior-art and the new method. Section III outlines the synthetic datasets used for evaluation. Section IV presents the results and discussion of comparative performance analysis. These include simulation results for classification accuracy using synthetic data, clustering using synthetic and recorded *in vivo* neural data, and complexity analysis. A metric based on the projection test is proposed for quantifying the discrimination degree of clusters. It is used to compare the sorting quality of the proposed method against other work. In addition, the overall complexity for sorting is optimized using the $\ell 1$-norm distance calculation. Finally, conclusions are discussed in Section V.

## II. Algorithms

The prosthetic hand signal processing control chain including spike sorting is illustrated in Fig. 2. Spike sorting is the process of grouping the recorded action potentials (spikes) into clusters based on the similarity of their shapes. The process can be divided into the following steps: 1) *spike detection and alignment*, separating spikes from noise and aligning the spikes to a common point, 2) *feature extraction*, extracting features of the spike shapes which gives a *dimensionality reduction*, i.e., going from a space of dimension $N$ (with $N$ the number of datapoints per spike) to a low dimensional space of a few features, and 3) *clustering*, grouping spikes with similar features into clusters, corresponding to the different neurons. In this paper, the focus is on the last two steps.

### A. Off-Chip Feature Extraction Methods

Principal component analysis (PCA) [7] has been the most commonly used algorithm for spike sorting because it yields an efficient coding of spikes (only the first 2–3 principal components need be retained). However, PCA requires offline training which is not compatible with online real-time spike sorting, and calculating the covariance matrix of the data demands high computational cost and hardware resources. In addition, with PCA there is no guarantee of optimal separation of clusters [8]. Another common technique is the discrete wavelet transform (DWT). It is a multi-resolution algorithm that provides good time resolution at high frequencies and good frequency resolution at low frequencies. But the convolution of the wavelet function with the original signal requires multiple multiplications and additions per spike, resulting in a high computational cost. Both PCA and DWT have traditionally been used for off-chip spike sorting.

### B. Proposed Method for On-Chip Feature Extraction

A simplified model of the DWT was presented in [6]. In it, discrete derivatives (DDs) are computed by calculating the slope at each sample point over a number of different time scales

$$\mathrm{DD}_{\delta}(n) = s(n) - s(n-\delta) \tag{1}$$

where $s$ is the spike waveform, $n$ is the sample point, and $\delta$ is the scaling factor (time delay). The equation shows subtraction between the samples $n$ and $(n-\delta)$. Normally $P$ DDs can be calculated per spike with different scaling factors to give multi-resolution spike decomposition, which corresponds to different frequency bands. This yields $P\times$ dimensionality expansion of feature space compared to the number of samples of an aligned spike (i.e., $N \rightarrow P \times N$). For example, the size of the feature space will be $N = (P = 3) \times (N = 45)$ samples per spike for DDs of three values of $\delta$. Feature space dimensionality directly impacts the computational complexity of spike sorting. As an illustration, the DDs of two typical spike waveforms with delay values of $\delta$ equal to 1, 3, and 7 are shown in Fig. 3.

Feature extraction based on extrema sampling (positive and negative peaks of DDs) is proposed here as an efficient approach not only in terms of computational simplicity but also accuracy. Retention of a subset of features significantly reduces the dimensionality from $P \times N$ to $K$, where $K$ is the number of selected features for the clustering stage $(K < N)$. For example, using $\delta = 1, 3, 7$ the following features are identified in Fig. 3: a) positive peaks $\mathrm{DD}|_{\delta=1,3,7(\max)}$; b) negative peaks $\mathrm{DD}|_{\delta=1,3,7(\min)}$; and c) peak-to-peak amplitude $(\mathrm{V_{p-p}})$ of each DD. Based on the concept of extrema selection, the following nine permutations of features are proposed for comparison with other work in terms of classification accuracy and computational complexity.

- Combination 1: $\mathrm{DD}|_{\delta=1,3,7(\max)}$ and $\mathrm{DD}|_{\delta=1,3,7(\min)}$.
- Combination 2: $\mathrm{V_{p-p}}$ of $\mathrm{DD}|_{\delta=1,3,7}$ (i.e., $\mathrm{DD}|_{\delta=1,3,7(\max)} - \mathrm{DD}|_{\delta=1,3,7(\min)}$).
- Combination 3: $\mathrm{DD}|_{\delta=3,5(\max)}$ and $\mathrm{DD}|_{\delta=3,5(\min)}$.
- Combination 4: $\mathrm{DD}|_{\delta=3,5(\max)}$ and $\mathrm{DD}|_{\delta=3,5(\min)}$, together with $\mathrm{V_{p-p}}$ of $\mathrm{DD}|_{\delta=3,5}$. $\mathrm{DD}|_{\delta=5}$ is not annotated in Fig. 3 for brevity.

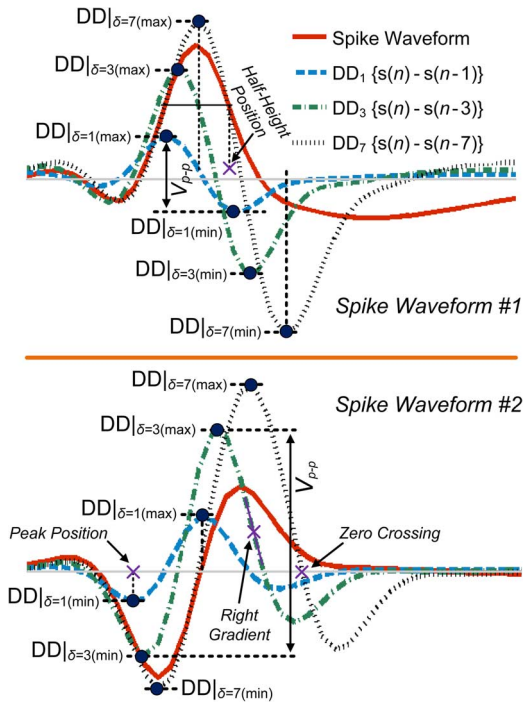


Fig. 3. Two spike waveforms (spike shapes) and their discrete derivatives. Positive peaks, negative peaks, and peak-to-peak amplitudes are annotated. Other features such as spike gradients and peak position are also depicted.

- Combination 5: $\mathrm{DD}|_{\delta=3,7(\max)}$ and $\mathrm{DD}|_{\delta=3,7(\min)}$.
- Combination 6: Features in Combination 5 together with $\mathrm{V_{p-p}}$ of $\mathrm{DD}|_{\delta=3,7}$.
- Combination 7: $\mathrm{DD}|_{\delta=3,7(\max)}$ and $\mathrm{DD}|_{\delta=3,7(\min)}$ together with original spike positive and negative peaks.
- Combination 8: $\mathrm{DD}|_{\delta=3,7(\max)}$ and $\mathrm{DD}|_{\delta=3,7(\min)}$ together with original spike height.
- Combination 9: $\mathrm{DD}|_{\delta=7(\max)}$ and $\mathrm{DD}|_{\delta=7(\min)}$ together with original spike positive and negative peaks.

In addition to the above permutations, various values of $\delta$ and geometric characteristics such as positive (or negative) signal energy, half-height position, right (or left) spike gradients, peak position and zero crossing points could be considered to define other possibilities. Some of the mentioned features are annotated in Fig. 3.

### C. Other Feature Extraction Methods for Comparison

*1) Waveclus [8]:* In this algorithm, the combination of the wavelet transform with super-paramagnetic clustering (unsupervised clustering method with offline training) is used for unsupervised and online spike sorting. Four-level multi-resolution decomposition using Haar wavelets is calculated as feature extraction which results in 64 wavelet coefficients for each spike. Then the Kolmogorov–Smirnov test for normality is used to select the first 10 coefficients with the largest deviation from normality for the sorting stage.

*2) Discrete Derivatives and Maximum Difference Test (DDs-MDT) [9]:* In this approach, the maximum difference test (MDT) is applied to each scaling factor of DDs to extract the multimodal coefficients. The MDT is a simplified model of the Lilliefors test for selecting the uncorrelated directions (minimum mutual information) for blind signal separation. Samples with bimodal distribution have deviation from unimodality or Gaussian distribution, thus they exhibit multiple peaks and valleys as a sign of multimodality.

*3) First and Second Derivatives:* The first and second derivatives of a spike represent its geometrical characteristics. The first derivative (FDV) interprets the gradient variations of a spike shape. It is defined as

$$\mathrm{FDV}(n) = s(n) - s(n-1). \tag{2}$$

The second derivative (SDV) highlights low frequency coefficients by computing the difference of the samples $n$ and $(n-1)$ of the FDV. That is

$$\mathrm{SDV}(n) = \mathrm{FDV}(n) - \mathrm{FDV}(n-1). \tag{3}$$

In [10] the FDV and SDV extrema (maximum and minimum peaks) were used to distinguish the clusters. This method is referred to as FDVSDV for the rest of the paper in the accuracy and complexity discussions.

*4) DDs and Uniform Sampling (DDs-USAMP) [11]:* In this method, after computing the DDs of the spike with three different values for the delay ($\delta = 1, 3, 7$), uniform sampling was performed to select the subset of features. Seven coefficients were selected for each DD level (resulting in 21 coefficients per spike) to achieve the highest median clustering accuracy across a signal-to-noise ratio range of 15-20 dB. Uniform sampling is a simple approach for decreasing the dimensionality from $N$ to $K$ (in this case from $3 \times 48$ to 21) but this type of sample selection could lead to non-segregation of clusters.

*5) Spike Shape [12]:* In this method all the samples of detected and peak-aligned spikes (without upsampling) are used for calculating the similarity measure between the mean of the spikes in clustering.

### D. Online Sorting (O-Sort Clustering)

For clustering O-Sort is used. It is the only online, automatic and unsupervised algorithm that is suitable for hardware implementation [13]. This algorithm provides real-time mapping of spikes to single neuron activity for closed-loop applications. The operation of O-Sort is as follows. 1) Initialization: Assign the first data point to its own cluster. 2) Calculate the distance between the next data point and each cluster centroid. The distance metric could use, for example, the Euclidean norm or the $\ell 1$-norm. 3) If the smallest distance is less than the merging threshold $T_M$, assign the point to the nearest cluster and recompute that cluster's mean. Otherwise, start a new cluster. 4) Check the distances between each cluster and every other cluster. If any distance is below the sorting threshold $T_S$, merge those two clusters and recomputed its mean. Steps 2–4 are then repeated indefinitely. In the simplified version of the algorithm (proposed in [13] and used herein), $T_M = T_S = T$. The threshold $T$ is defined as $T = S(\sigma_r)^2$, where $\sigma_r$ is the average standard deviation of the data computed continuously with a long ($\sim$1 min) sliding window, and $S$ is the number of datapoints of a single waveform.

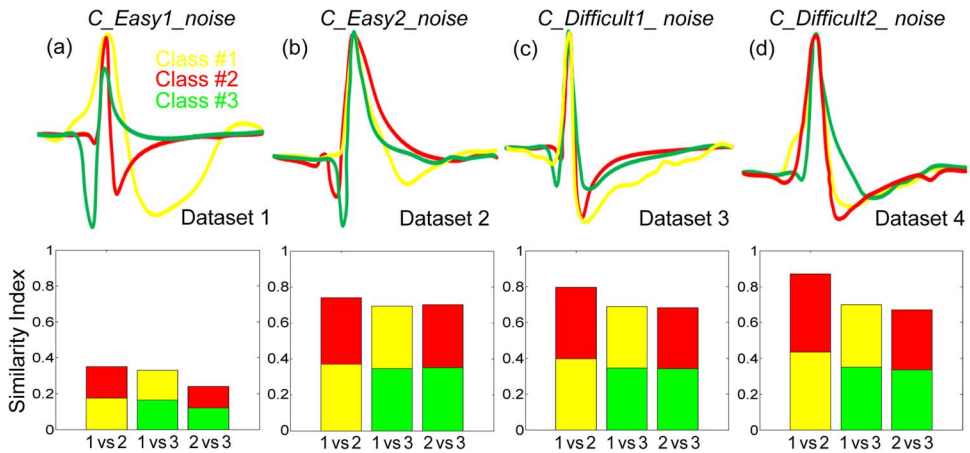


Fig. 4. Spike bank mean waveforms (peak-aligned) used for testing with corresponding Bray–Curtis similarity index (shown at the bottom). (a) *C_Easy1_noise*, (b) *C_Easy2_noise*, (c) *C_Difficult1_noise*, and (d) *C_Difficult2_noise*.

O-Sort is simple in operation with good complexity-accuracy tradeoff and satisfies online sorting constraints (memory and power). This algorithm is adaptive thus nonstationarity of data in time is applied to the cluster position and number of clusters. A disadvantage of O-Sort is that it may split clusters into sub-clusters leading to a reduction in clustering performance. The created sub-clusters are not matched with any source and are considered as noise clusters.

## III. Test Datasets

To compare the performance of the proposed method with other work the Waveclus spike bank[1] was used. The database contains different average spike waveforms recorded from the neocortex and basal ganglia in humans. To emulate the background noise activity, spike waveforms randomly chosen from the data library were added to the generated datasets. There are advantages in using this database. Firstly, each dataset provides true spike classes, which is useful for accuracy calculations, and the ground truth can be established. Secondly, the diversity of the data allows evaluation of spike sorting algorithms from a constant source. Datasets with different degrees of difficulty (e.g., similarity of spike shape) and noise levels are provided. The four datasets are *C_Easy1_noise*, *C_Easy2_noise*, *C_Difficult1_noise*, and *C_Difficult2_noise*, where *noise* denotes the noise level in terms of standard deviation, namely, 0.01, 0.05, 0.15, and 0.2. "*Easy*" and "*Difficult*" is the similarity index between the spike shapes in each dataset. Thirdly, the characteristics of the datasets are similar to real practical recordings.

Fig. 4 shows the three different types of spike shape present in each of the four test datasets and the calculated Bray–Curtis similarity indexes [14] between all the spike shapes in each dataset. The sorting difficulty becomes more demanding with increasing the similarity between the spike shapes (width and amplitude fluctuations). Fig. 5(a) shows color-coded spikes corresponding to different neurons from dataset 2 (*C_Easy2_0.05*), Fig. 5(b) shows the two-dimensional (2-D) projection of spike clusters, and Fig. 5(c) shows a short recording segment with colored markers (with prior knowledge from simulation). Each color corresponds to a single-unit activity. As the difficulty of a dataset increases, sorting becomes increasingly challenging. For realistic assessment, a 7-bit, 30 kS/s second-order delta sigma modulator was used to digitize the raw data.

[1]Available online: http://www2.le.ac.uk/centres/csn/spike-sorting

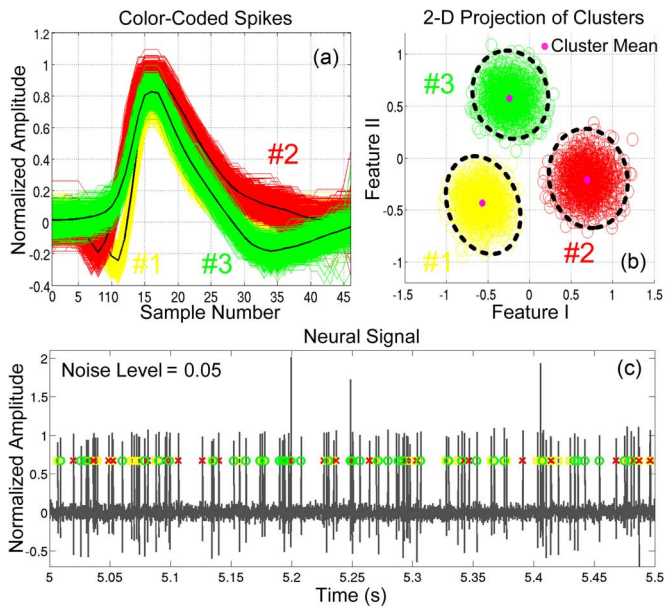


Fig. 5. Test dataset 2 (*C_Easy2_0.05*) showing. (a) Color-coded spikes corresponding to different neurons (#1 yellow, #2 red, #3 green). (b) 2-D projection of spike clusters. (c) Segment of simulated dataset 2.

## IV. Results and Discussion

### A. Determining the Optimal Threshold

The procedure adopted to determine the optimal threshold $T$ is as follows.

1) The range of threshold values is determined via the dataset with the highest similarity index between the spike shapes. The maximum limit for threshold $(T_{\max})$ is defined using dataset 4 (which has the highest similarity measure) when no clusters are missed and there is no artificial clustering (Fig. 6). Artificial clustering erroneously allocates spikes to a cluster. The threshold value can be decreased to a minimum limit $(T_{\min})$ below which overclustering occurs.

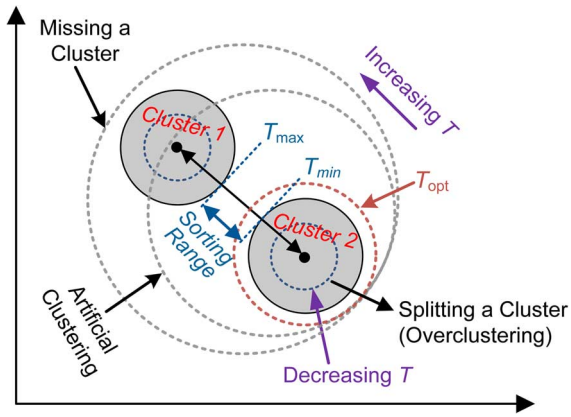


Fig. 6. 2-D representation of feature space for two clusters from dataset 4 with the highest similarity. The effect of increasing (or decreasing) the threshold $T$ is depicted. Different threshold levels ($T_{\rm opt}$, $T_{\rm max}$ and $T_{\rm min}$) and sorting range are indicated. For $T > T_{\rm max}$ the risk of missing a cluster and artificial clustering is high. For $T < T_{\rm min}$ the main cluster would artificially be split into two or more sub-clusters.

Overclustering is the splitting of a single cluster into multiple clusters. $T_{\rm min}$ is determined when clusters in dataset 4 start to overcluster. For $T_{\rm min} < T < T_{\rm max}$ no clusters will be missed and no overclustering will occur.

2) The optimal threshold ($T_{\rm opt}$) for each method (published and proposed) is found by sweeping the threshold value through the range $T_{\rm max} - T_{\rm min}$ and determining the sorting accuracy.

Using this procedure determines $T_{\rm opt}$ for each method when investigating its effectiveness using different metrics.

### B. Classification Accuracy

In this subsection the nine feature combinations proposed in Section II-B are evaluated in terms of classification accuracy to determine in conjunction with O-Sort the best one. This is then compared with the feature extraction methods outlined in Section II-C. Classification accuracy is defined as

$$\mathrm{CA_{cc}} = \frac{\mathrm{TPCC}}{\mathrm{NTS}} \times 100\% \qquad (4)$$

where TPCC is the number of truly detected and correctly classified spikes and NTS is the number of truly detected spikes. NTS = DTS − (FPS + MS), where DTS is the number of detected spikes, FPS is the number of false alarm spikes due to noise or overlapping spikes, and MS is the number of missed spikes.

The average sorting results are summarized in Table I. The methods were evaluated across all datasets and noise levels. Combination 5 with dimensionality ($K$) of 4 achieves the highest $\mathrm{CA_{cc}}$ whereas Combination 8 (with $K = 9$) achieves the lowest $\mathrm{CA_{cc}}$. The methods were also examined with overclustering ratio criteria. It was observed that Combination 5 achieves the lowest overclustering whereas Combination 1 in conjunction with O-Sort generally tends to divide a cluster into sub-clusters. Combination 5 is therefore selected and will from hereon be referred to as the $\mathrm{DD}|_2$-Extrema method.

TABLE I
CLASSIFICATION ACCURACY COMPARISON OF THE EXAMINED FEATURE COMBINATIONS

| Average Classification Accuracy | | | | | |
|---|---|---|---|---|---|
| Combination* | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 | Mean |
| Combination 1 (**$K$ = 6) | 94.8% | 92% | 88% | 85.8% | 90.1% |
| Combination 2 ($K$ = 3) | 76.2% | 78.6% | 73.4% | 68.6% | 74.2% |
| Combination 3 ($K$ = 4) | 91% | 86% | 77.2% | 79.8% | 83.5% |
| Combination 4 ($K$ = 6) | 91.4% | 86.8% | 83. 6% | 81.4% | 85.8% |
| Combination 5 ($K$ = 4) | 95.8% | 93.4% | 87.8% | 89.6% | 91.6% |
| Combination 6 ($K$ = 6) | 91.2% | 89.6% | 80.4% | 79.8% | 85.2% |
| Combination 7 ($K$ = 6) | 89.6% | 84.6% | 79.2% | 74.8% | 82% |
| Combination 8 ($K$ = 5) | 75.6% | 77.4% | 70.4% | 68.8% | 73% |
| Combination 9 ($K$ = 4) | 80.4% | 76.4% | 75.4% | 71.2% | 75.8% |

*See Section II-B for the features used in each Combination

**$K$ = number of features

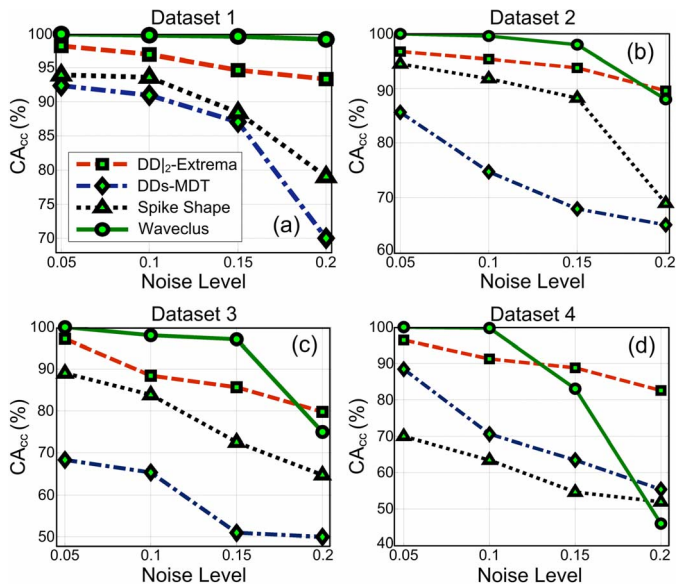


Fig. 7. Comparison of classification accuracy between the $\mathrm{DD}|_2$-Extrema method and other methods as a function of noise level for the four datasets. The result of Waveclus in [8] is used for comparison.

For comparison, the $\mathrm{CA_{cc}}$ of the methods in Section II-C was investigated, in particular, Waveclus ($K = 10$), DDs-MDT ($K = 21$), DDs-USAMP ($K = 21$), Spike Shape ($K = 45$) and PCA3 (projection of first three principal components). The results (averaged across all noise levels) are shown in Fig. 7. The $\mathrm{CA_{cc}}$ of DDs-MDT drops significantly in datasets 2 and 3 due to the intense overclustering effect. The number of coefficients ($K$) representing deviation from normality is 21, i.e., 7 from each scaling factor ($\delta = 1, 3, 7$). The results show that $\mathrm{DD}|_2$-Extrema and DDs-MDT have similar performance in dataset 1. The former has better similarity tolerance in datasets

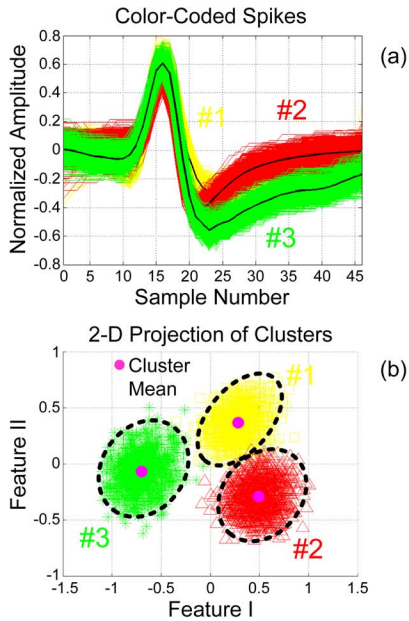


Fig. 8. Test dataset 3 (*C_Difficult1_0.05*) showing: (a) Color-coded spikes corresponding to different neurons (#1 yellow, #2 red, #3 green). (b) 2-D projection of spike clusters.

2 and 3 which means significant reduction in complexity or required memory. Spike Shape works satisfactorily for datasets 1 and 2 only. It significantly increases the computational complexity without improvement in performance. In this method the simplest metric to sort the spikes is the distance (e.g., Euclidean distance) between the unclassified spikes and the stored templates. As discussed in [15] the classification accuracy of Spike Shape declines when the spike waveforms have similar patterns (increasing the similarity index).

PCA uses maximum variance, correlated coefficients, unlike the Waveclus and DDs-MDT. In the test for *C_Difficult1_noise* and *C_Difficult2_noise* the sorting algorithm did not distinguish one of the three clusters with PCA feature vectors. Although PCA is computationally complex, there is no guarantee of efficient results. Selecting the coefficients with the largest variance does not necessitate deviation from normality and it may compromise the sorting performance. Feature space probability density function of two different clusters using PCA overlap (correlated directions) was investigated in [16]. PCA and DDs-USAMP did not perform well in the tests. Their average $\mathrm{CA_{cc}}$ with the O-Sort classifier was 66.4% and 61%, respectively, and therefore they are not plotted in Fig. 7.

In conclusion, the best performance for accuracy-dimensionality is demonstrated by $\mathrm{DD}|_2$-Extrema ($K = 4$). It exhibits superior similarity tolerance and noise immunity.

### C. Clustering Results With Synthetic Data

The evaluation of spike sorting with $\mathrm{DD}|_2$-Extrema is shown in Fig. 8 using dataset 3 (*C_Difficult1_0.05*). Fig. 8(a) shows color-coded spikes in each cluster and Fig. 8(b) shows color-coded mapping to distinguish between the neurons. The clusters are distinguished with the knowledge of true identities from the neural simulator. For clarity, noise events and overlapping spikes are not plotted. The typical way of illustrating the isolated units is by superimposing detected spikes with different colors.

The detailed sorting results using dataset 4 are shown in Fig. 9. The two statistical tests, namely the inter-spike interval histogram (ISIH) test and the projection test as discussed in [17], were used to quantitatively assess the sorting quality. A total of 3040 raw waveforms were detected, 2929 (96.4%) of which were assigned to one of the three well-separated single units (969, 986, and 974 for each cluster, respectively). Fig. 9(a) shows the normalized raw waveforms and the mean waveform for each of the three clusters. Fig. 9(b) shows the firing pattern of each neuron across a time window of 0–28 s. The mean firing rate (FR) [18] of neurons #1, #2, and #3 is 10.28 Hz, 10.64 Hz, and 9.98 Hz, respectively. Each diagram in Fig. 9(b) also has the raster plot (or spike times) corresponding to the temporal firing of each neuron. Fig. 9(c) shows the ISIH for each neuron. The ISIH window should not be less than the action potential refractory period (<2 ms after alignment). Single-unit activity is stated when the spike waveform is clearly distinguishable with no ISIH less than the refractory period. Fig. 9(d) shows from left to right the 2-D projection of clusters with increasing noise level (0.05, 0.1, and 0.15). Increasing the noise level adversely affects the projection and may result in some degree of overlap, but it is observed that the borders of clusters are clear even in the most difficult dataset with noise level of 0.15. Fig. 9(e) shows the results of the projection test using probability density functions for three combinations of cluster in *C_Difficult2_005*. The projection test [13], [19] is a one-dimensional representation between two known means with a distance indicator $(D)$ which assesses the quality of clustering. This test shows whether or not spikes from multiple neurons are artificially assigned to a particular cluster by the sorting algorithm. Furthermore, it detects the invalid merging of two clusters. In Fig. 9(e) the normalized distance between probability density functions for the three combinations of clusters are sufficiently large to permit correct assignment of spikes to unique neurons.

Inspired by [16] a quantitative metric based on the projection test is proposed for assessing the level of distinctness of the generated clusters. The metric for quantifying the discrimination degree of clusters is the ratio of intercluster distance to intracluster distance, defined as

$$\mathrm{DisDeg} = \frac{\mathrm{inter}}{\mathrm{intra}} = \frac{\min_{\substack{i=1,2,\ldots,y \\ j=1,2,\ldots,y;}} \quad _{i \neq j} \left( \sum_{i=1}^{y} \sum_{j=1}^{y} PD_{ij} \right)}{T} \tag{5}$$

where $PD_{ij}$ is the projection test distance between clusters $i$ and $j$. This ratio is a metric for cluster quality measurements and is useful for optimization. The higher the value of DisDeg the better the cluster separation quality.

Fig. 10 compares the separation confidence level of the Graph–Laplacian feature (GLF) [16], $\mathrm{DD}|_2$-Extrema and DDs-MDT. Increasing the similarity in each dataset from *Easy1* to *Difficult2* and noise level adversely affects the cluster

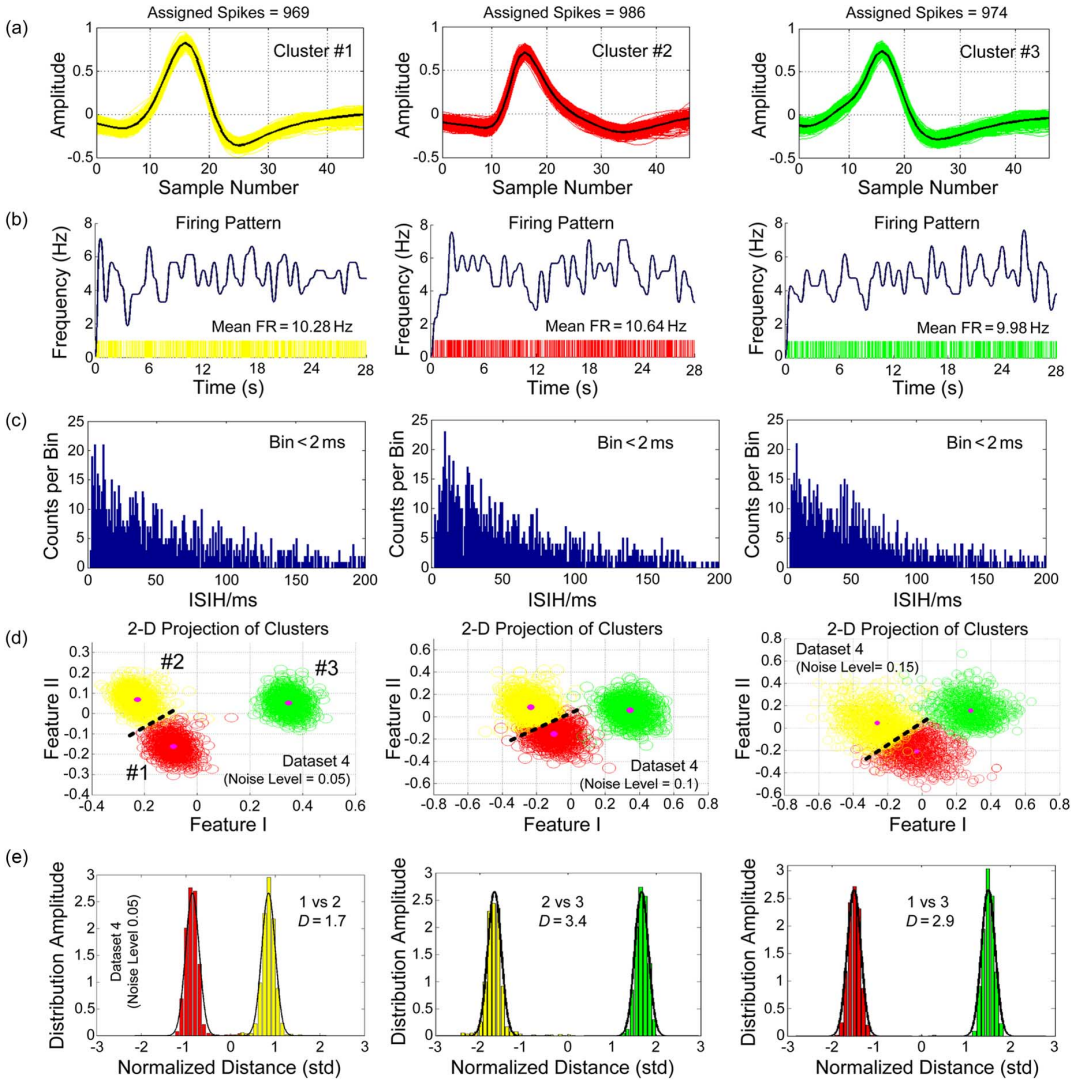


Fig. 9. Sorting results of *C_Difficult2_005*. (a) Color-coded clusters with number of assigned spikes in each cluster. Amplitude is of arbitrary units. (b) Corresponding firing pattern which depicts firing rate of each neuron (0–28 s). Approximated firing rate determined by the Gaussian window function. The mean firing rate (FR) is annotated in each plot. (c) ISIH of each cluster. (d) 2-D projection of clusters for *C_Difficult2_0.05*, *C_Difficult2_0.01* and *C_Difficult2_0.15*. (Spikes have been colored according to the ground truth). (e) Illustrates projection test using probability density functions for the three combinations of clusters (*C_Difficult2_0.05*) in (d). For each combination of neurons the distance between the two distributions is described by how many standard deviations they are apart (*D* value in each plot). Each neuron is color-matched across (d) and (e).

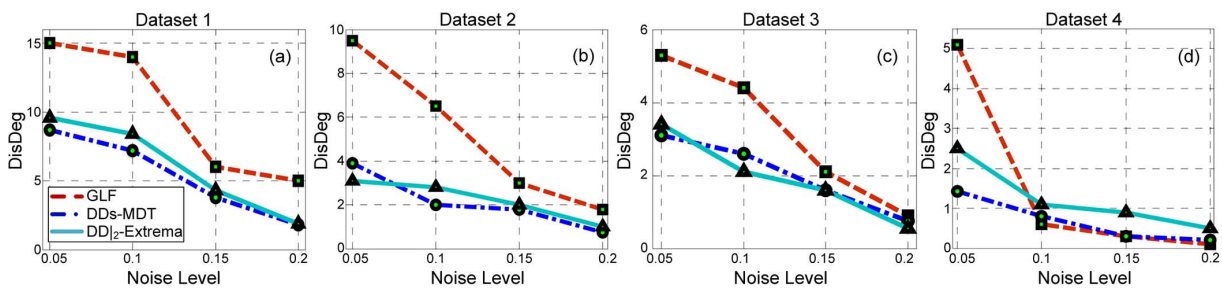


Fig. 10. Discrimination degree of GLF, DDs-MDT and $\mathrm{DD}|_2$-Extrema. For simplicity $T = 1$. DisDeg of GLF was calculated using the quality metric in [16].

separation. The DisDeg results in Fig. 10 verify the sorting efficiency of the proposed method.

### *D. Clustering Results With Recorded In Vivo Neural Data*

Collected neural signals from the peripheral median nerve in pig (obtained with a multi-electrode cuff *in vivo*) were used to test the sorting performance of the $\mathrm{DD}|_2$-Extrema. 6564 spike waveforms were detected from 24 single neurons in four different channels: eight in channel 1, five in channel 2, five in channel 3, and six in channel 4. The analysis of the sorting results is shown in Fig. 11 for three of the channels. Fig. 11(a) shows the mean spike waveforms of channels l–3; each color

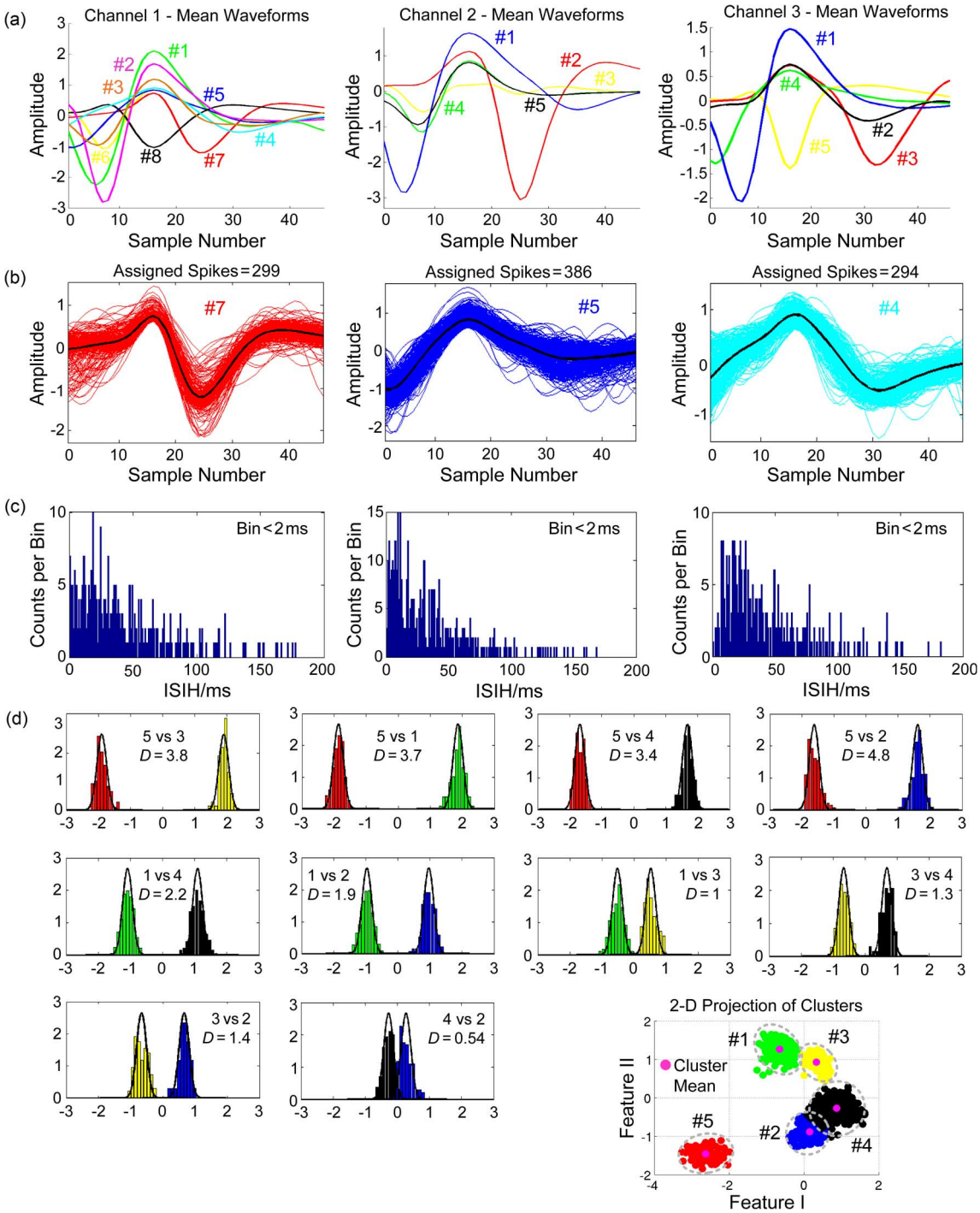


Fig. 11. Sorting results of the recorded *in vivo* neural data. (a) Illustration of found mean waveforms in channels 1–3. The amplitude is of arbitrary units. (b) Color-coded clusters with number of assigned spikes in each cluster from channel 1 (#7 red, #5 blue, #4 cyan). (c) ISIH of the neurons from channel 1. (d) Results of projection test using estimated probability density functions for all possible combinations in channel 2. 2-D projection of clusters is included for visual clarity. The vertical axis shows the distribution amplitude and the horizontal axis is the distance between the two distributions.

corresponds to a unique neuron. Fig. 11(b) shows the sorting results of three similar neurons from channel 1. The ISIH test shown in Fig. 11(c) verifies the accurate segregation between the chosen neurons. In Fig. 11(d), the projection test quantifies the distance between every pair of clusters in channel 2. The normalized distance between each pair (standard deviation criterion) is large enough to conclude the efficient separation of clusters. In total 5973 (91%) of all detected spikes were assigned to the identified neurons.

### *E. Complexity Analysis*

In order to assess the hardware requirements of different methods the computation complexity metric was used. It is defined as [9], [20]

$$\mathrm{ComputComp} = N_{\mathrm{add(sub)}} + 10N_{\mathrm{mult(div)}} \tag{6}$$

where $N_{\mathrm{add}}$ is the number of additions (or subtractions), and $N_{\mathrm{mult}}$ is the number of multiplications (or divisions)

TABLE II
COMPUTATIONAL COMPLEXITY COMPARISON OF VARIOUS FEATURE EXTRACTION AND DIMENSIONALITY REDUCTION METHODS

| Method | Feature Extraction Additions | Feature Extraction Multiplications | Dimensionality Reduction | Clustering Algorithm | *Average $CA_{cc}$ | Number of Features ($K$) | ADC Simulator |
|---|---|---|---|---|---|---|---|
| $DD\|_2$-Extrema | $M(2N-10)$ | - | Max/Min/$V_{p\text{-}p}$ of $DD\|_{2(\delta=3,7)}$ | O-Sort | 91.6% | 4 | ✓ |
| PCA | $M(N^2+1)$ | $M(N^2+N)$ | - | O-Sort | 66.4 % | 3 first PCs | ✓ |
| FDVSDV | $M(2N-3)$ | - | Max/Min of FDV and SDV | O-Sort | 73.6% | 4 | ✓ |
| [a]DDs-MDT | $M(3N-11)$ | - | MDT$\rightarrow(M-1)(3N)$ | O-Sort | 71.8% | 21 | ✓ |
| [a]DDs-USAMP | $M(3N-11)$ | - | Uniform Sampling | O-Sort | 61% | 21 | ✓ |
| Spike Shape | - | - | - | O-Sort | 68.4% | 45 | ✓ |
| [c]ZCF [23] | $M(N)$ | - | - | $k$-means | >94.0% | 2ZCFs = 3 | × |
| **DWT | $M(4N)$ | $M(8N-10)$ | Kolmogorov-Smirnov | SPC | [b]92.6% | 10 | × |

$M$ = number of spikes; $N$ = sample number per spike
* Averaged across 4 datasets with varying degrees of noise level
** DWT (four-level Haar wavelet) and Kolmogorov–Smirnov used in Waveclus
$a$—DDs-MDT and DDs-USAMP with scaling factors $\delta = 1, 3, 7$
$b$—The results in [8] are used for comparison
$c$—ZFC performance was evaluated with a different spike bank

required. Table II compares the proposed method and seven other published methods in terms of estimated computational complexity, clustering algorithm used, average classification accuracy $(\mathrm{CA_{cc}})$ and number of features (dimensionality). Compared to DDs-MDT, $\mathrm{DD}|_2$-Extrema has 3.6× lower complexity with 19.8% higher average accuracy. FDVSDV has 5.76× and 1.62× higher complexity compared with $\mathrm{DD}|_2$-Extrema and DDs-MDT, respectively. It was reported in [10] that FDVSDV has 6.97% classification error with varying noise levels across all four datasets and $M(2N-3)$ complexity with the $k$-means classifier.

Since in FSVSDV extrema are selected using the attenuated projections of a spike shape, the O-Sort sorting threshold needs to be reduced to distinguish the clusters. The threshold level for FDVSDV was found using $T = S(\alpha \cdot \sigma)^2$ where $\alpha(= 0.25)$ is a factor derived from simulation. There are disadvantages with the calculated threshold. They include: 1) the possible creation of an impractically small threshold value (e.g., 0.001) which is then very sensitive to noise variations, and 2) determining the sorting threshold involves an extra multiplication (evaluating $T$ for recalculating the average standard deviation when using a sliding window).

FSVSDV was implemented with O-Sort to provide a fair comparison with the method proposed in this paper. The average accuracy of FDVSDV is 73.6%. DDs-MDT has an average accuracy of 71.8% with O-Sort. Six of the eight methods listed in Table II use the O-Sort clustering algorithm. It should be mentioned that the average accuracy of O-Sort is around 70% while both $k$-means and fuzzy $c$-means have over 90% discrimination power. However, the only online and unsupervised algorithm in the context of implantable sorting hardware is O-Sort. In addition, $k$-means and fuzzy $c$-means clustering require prior knowledge of the number of clusters.

To test the accuracy of the classifiers, original spike shapes were applied (without upsampling) to establish the accuracy with unprocessed data. Waveclus is very efficient with an acceptable classification error (7.4%). However, its computational complexity is much higher than that of either $\mathrm{DD_2}$-Extrema or DDs-MDT. $\mathrm{DD}|_2$-Extrema without offline training requires <5% of the computational complexity of Waveclus (the computational complexity of Kolmogorov–Smirnov is not considered). Although $\mathrm{DD}|_2$-Extrema and Waveclus have comparable similarity immunity, there is a sustained improvement in $\mathrm{DD}|_2$-Extrema performance with increasing noise level (noise immunity).

The overall sorting complexity consists of creating clusters and merging phases. It is defined as

$$\mathrm{SortComp} = \overbrace{MK(3C-3)+K(3C-3)}^{\mathrm{Add}} + \overbrace{20MKC}^{\mathrm{Mult}} \quad (7)$$

where $M$ is the number of feature vectors, $K$ is the number of features representing each spike in feature space, and $C$ is the number of clusters. The Euclidean distance calculation requires the multiplication operation in (7) which leads to high sorting complexity. The $\ell 1$-norm distance metric makes O-Sort a particularly good choice for hardware implementation since it is much less dependent on the dimensionality of feature space. The $\ell 1$-norm distance is less susceptible to biological noise than the Euclidean distance [21], hence resulting in a better average $\mathrm{CA_{cc}}$. Fig. 12 shows the classification error versus computational complexity of feature extraction and sorting. The best tradeoff between classification error and complexity belongs to the proposed method.

Fig. 13 displays the average classification error versus dimensionality factor for various methods. The dimensionality factor is the ratio of feature space dimensions to the number of samples per spike. As can be seen, there are significant differences between $\mathrm{DD}|_2$-Extrema $(K = 4)$ and FDVSDV $(K = 4)$, Spike Shape $(K = 45)$, DDs-MDT $(K = 21)$, DDs-USAMP $(K = 21)$ and PCA $(K = 3)$. The competing methods for $\mathrm{DD}|_2$-Extrema are ZCF [22] $(2\mathrm{ZCFs} = 3)$ and Waveclus $(K = 10)$ with 2.4% and 1% lower classification error, respectively. It can be clearly observed that $\mathrm{DD}|_2$-Extrema outperforms all the other methods and provides the best tradeoff in complexity, accuracy and dimensionality.

### F. Proposed Data Reduction Application Example

As noted in Section I, one potential application is the development of a custom integrated circuit for an implantable multi-electrode neural interface for upper-limb prostheses. The

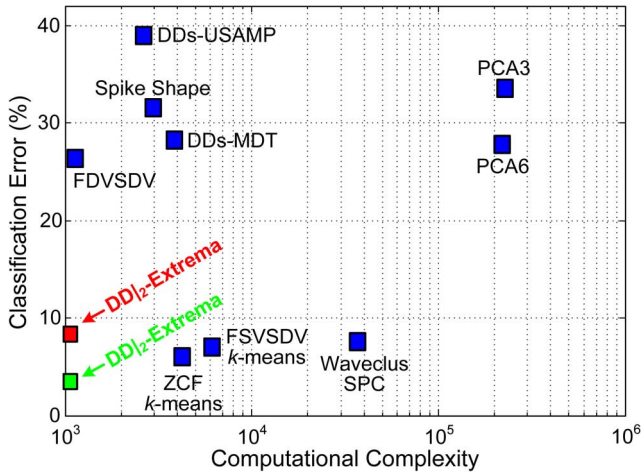


Fig. 12. Classification error versus computational complexity for the different feature extraction and sorting methods considered herein. Red square ($\mathrm{DD}|_2$-Extrema) shows the average accuracy across all datasets and all noise levels. Green square ($\mathrm{DD}|_2$-Extrema) shows the average accuracy across all datasets using noise with a standard deviation of 0.05.

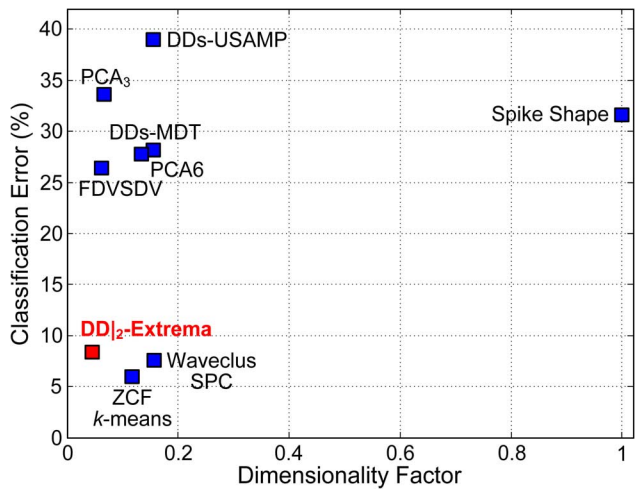


Fig. 13. Classification error versus dimensionality factor for the different feature extraction methods considered herein.

chip will amplify and reduce the data rate needed to represent the many spike signals. Each interface will typically have 20+ channels (microelectrodes) and there are four nerves in the upper arm which carry most of the motor axons. Prior to spike sorting the recorded data will be digitized by an analog-to-digital converter (ADC, see Fig. 2). Assuming each channel is sampled at 30 kHz with 7-bit ADC resolution, the average data rate at the input of the digital spike sorting processor will be 21 Mb/s. A typical neuron generates on average 40 spikes per second when active [23], and up to 25 active neurons are estimated in each channel. Therefore, after detection and alignment the data rate will become 1.26 Mb/s (100 neurons $\times$ 40 spikes/(neuron.second) $\times$ 45 samples/spike $\times$ 7 bits/sample). Using $\mathrm{DD}|_2$-Extrema to encode the spikes into four features (28 bits/spike) the extracted coefficients will yield a data rate of 112 kb/s or 0.53% of the original data rate. In the final step typically the cluster number and channel number will be processed via O-Sort, which will yield a final transmission data rate of 56 kb/s (i.e., 0.265% of the original data rate). Such a low data rate is feasible for wireless transmission using a single pair of coils for both power and data [24]. Reducing the data rate to only 0.265% of the original data rate should also be attractive for high channel count recording microsystems for other applications [25], [26].

## V. Conclusion

This paper has investigated a new feature extraction method based on spike waveforms and their discrete derivatives. Nine combinations of extrema features have been examined with the proposed $\mathrm{DD}|_2$-Extrema method offering the highest average classification accuracy. Specifically, $\mathrm{DD}|_2$-Extrema with $M(2N-10)$ complexity and dimensionality factor of 4, achieves 91.6% average classification accuracy. It requires $>1\%$ of the computational complexity of PCA while providing higher accuracy. The combination of $\mathrm{DD}|_2$-Extrema with O-Sort provides a considerable accuracy-complexity tradeoff. This should allow on-chip processing without any assumption of high level of fidelity for sorting multi-unit activity. The results confirm that the average classification error is less than 4% across all the datasets tested using noise with a standard deviation of 0.05. This theoretical limit can be used to determine the design parameters of the analog front end (AFE, see Fig. 2) including data converter resolution and sampling rate, filter type, bandwidth and order, and amplifier noise, bandwidth and gain. The overall complexity of spike sorting has been optimized using the $\ell 1$-norm distance calculation. The clustering performance of the proposed method has been evaluated using both synthetic and recorded *in vivo* neural data. $\mathrm{DD}|_2$-Extrema outperforms various online and offline algorithms which have significantly more complexity. In addition, it offers a better tradeoff between complexity, accuracy and dimensionality than all the other methods considered herein. $\mathrm{DD}|_2$-Extrema could be used as the basis of a future neural amplifying and spike sorting chip for upper-limb prostheses or other applications such as cortical neural recording [27] and bladder control after spinal cord injury [28].

## Acknowledgment

The authors would like to thank Prof. N. Donaldson from the Implanted Devices Group at University College London for providing the recorded *in vivo* neural data.

## References

[1] T. Yanagisawa, M. Hirata, Y. Saitoh, T. Goto, H. Kishima, R. Fukuma, H. Yokoi, Y. Kamitani, and T. Yoshimine, "Real-time control of a prosthetic hand using human electrocorticography signals," *J. Neurosurg.*, vol. 114, no. 6, pp. 1715–1722, Jun. 2011.

[2] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, pp. 372–375, May 2012.

[3] M. Tombini, J. Rigosa, F. Zappasodi, C. Porcaro, L. Citi, J. Carpaneto, P. M. Rossini, and S. Micera, "Combined analysis of cortical (EEG) and nerve stump signals improves robotic hand control," *Neurorehabil. Neural Repair*, vol. 26, no. 3, pp. 275–281, Mar.–Apr. 2012.

[4] S. P. Lacour, J. J. Fitzgerald, N. Lago N, E. Tarte, S. McMahon, and J. Fawcett, "Long micro-channel electrode arrays: A novel type of regenerative peripheral nerve interface," *IEEE Trans. Neural. Syst. Rehabil. Eng.*, vol. 17, no. 5, pp. 454–460, Oct. 2009.

[5] J. J. Fitzgerald, N. Lago, S. Benmerah, J. Serra, C. P Watling, R. E Cameron, E. Tarte, S. P. Lacour, S. B. McMahon, and J. W Fawcett, "A regenerative microchannel neural interface for recording from and stimulating peripheral axons *in vivo*," *J. Neural Eng.*, vol. 9, p. 016010, Jan. 2012.
[6] Z. Nadasdy, R. Quian Quiroga, Y. Ben-Shaul, B. Pesaran, D. A. Wagenaar, and R. Andersen, "Comparison of unsupervised algorithms for on-line and off-line spike sorting," presented at the 32nd Annu. Meet. Soc. Neurosci., Orlando, FL, 2002 [Online]. Available: http://www.vis.caltech.edu/~zoltan/
[7] M. Abeles and M. H. Goldstein, Jr., "Multispike train analysis," *Proc. IEEE*, vol. 65, no. 5, pp. 762–773, May 1977.
[8] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *J. Neural Comp.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.
[9] S. Gibson, J. W. Judy, and D. Marković, "Technology-aware algorithm design for neural spike detection, feature extraction, dimensionality reduction," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 5, pp. 469–478, Oct. 2010.
[10] S. E. Paraskevopoulou, D. Y. Barsakcioglu, M. R. Saberi MR, A. Eftekhar A, and T. G. Constandinou, "Feature extraction using first and second derivative extrema (FSDE) for real-time and hardware-efficient spike sorting," *J. Neurosci. Methods*, vol. 215, no. 1–2, pp. 29–37, Jan. 2013.
[11] V. Karkare, S. Gibson, C.-H. Yang, H. Chen, and D. Marković, "A 130-$\mu$W, 64-channel neural spike-sorting DSP chip," *IEEE J. Solid-State Circuits*, vol. 46, no. 5, pp. 1214–1222, May 2011.
[12] Y. Yuan, C. Yang, and J. Si, "The m-sorter: An automatic and robust spike detection and classification system," *J. Neurosci. Methods*, vol. 210, no. 2, pp. 281–290, Sep. 2012.
[13] U. Rutishauser, E. M. Schuman, and A. N. Mamelak, "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, *in vivo*," *J. Neurosci. Methods*, vol. 154, no. 1–2, pp. 204–224, Jun. 2006.
[14] J. Lian, G. Garner, D. Muessig, and V. Lang, "A simple method to quantify the morphological similarity between signals," *J. Signal Process.*, vol. 90, no. 2, pp. 684–688, Feb. 2010.
[15] D. M. Schwarz, M. S. A. Zilany, M. Skevington, N. J. Huang, B. C. Flynn, and L. H. Carney, "Semi-supervised spike sorting using pattern matching and a scaled Mahalanobis distance metric," *J. Neurosci. Methods*, vol. 206, no. 2, pp. 120–131, May 2012.
[16] Y. Ghanbari, P. E. Papamichalis, and L. Spence, "Graph-Laplacian features for neural waveform classification," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 5, pp. 1365–1372, Jun. 2011.
[17] C. Pouzat, O. Mazor, and G. Laurent, "Quality metrics to accompany spike sorting of extracellular signals," *J. Neurosci. Methods*, vol. 31, no. 24, pp. 8699–8705, Jun. 2011.
[18] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press, 2001.
[19] C. Pouzat, O. Mazor, and G. Laurent, "Using noise signature to optimize spike-sorting and to assess neuronal classification quality," *J. Neurosci. Methods*, vol. 122, no. 1, pp. 43–57, Dec. 2002.
[20] A. Zviagintsev, Y. Perelman, and R. Ginosar, "Low-power architectures for spike sorting," in *Proc. 2nd IEEE Int. Conf. EMBS Neural Eng.*, Arlington, VA, 2005, pp. 162–165.
[21] V. Karkare, S. Gibson, and D. Marković, "A 75-$\mu$W, 16-channel neural spike-sorting processor with unsupervised clustering," *IEEE J. Solid-State Circuits*, vol. 48, no. 9, pp. 2230–2238, Sep. 2013.
[22] A. Kamboh and A. Mason, "Computationally efficient neural feature extraction for spike sorting in implantable high-density recording systems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 1, pp. 1–9, Jan. 2013.
[23] R. R. Harrison, P. T. Watkins, R. J. Kier, R. J. Lovejoy, B. Greer, and F. Solzbacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 123–133, Jan. 2007.
[24] D. Cirmirakis, D. Jiang, A. Demosthenous, N. Donaldson, and T. Perkins, "A fast passive phase shift keying modulator for inductively coupled implanted medical devices," in *Proc. 38th Eur. Solid-State Circuits Conf.*, Bordeaux, France, 2012, pp. 301–304.
[25] I. H. Stevenson and K. P. Kording, "How advances in neural recording affect data analysis," *Nat. Neurosci.*, vol. 14, no. 2, pp. 139–142, Feb. 2011.
[26] A. Berényi, Z. Somogyvari, A. J. Nagy, L. Roux, J. D. Long, S. Fujisawa, E. Stark, A. Leonardo, T. D. Harris, and G. Buzsaki, "Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals," *J. Neurophysiol.*, Dec. 2013.
[27] D. J. Chew, L. Zhu, E. Delivopoulos, I. R. Minev, K. M. Musick, C. A. Mosse, M. Craggs, N. Donaldson, S. P. Lacour, S. B. McMahon, and J. W. Fawcett, "A microchannel neuroprosthesis for bladder control after spinal cord injury in rat," *Sci. Transl. Med.*, vol. 5, no. 210ra155, Nov. 2013.
[28] D. A. Borton, M. Yin, J. Aceros, and A. Nurmikko, "An implantable wireless neural interface for recording cortical circuit dynamics in moving primates," *J. Neural Eng.*, vol. 10, p. 026010, Feb. 2013.

**Majid Zamani** (S'13) was born in Tehran, Iran, in 1984. He received the M.Sc. degree in microelectronics from the Islamic Azad University, Science and Research Branch, Tehran, Iran, in 2011. He then joined the Analog and Biomedical Electronics Group in the Department of Electronic and Electrical Engineering at University College London, London, U.K., to study for the Ph.D. degree.

His research interests are the design of implantable devices for neural prostheses, low-power mixed-signal integrated circuit design, and data converters in advanced CMOS technologies.

Mr. Zamani was awarded the best researcher M.Sc. student award. He was also awarded a UCL Overseas Research Scholarship and a Graduate Research Scholarship to pursue his Ph.D. degree.

**Andreas Demosthenous** (S'94–M'99–SM'05) received the B.Eng. degree in electrical and electronic engineering from the University of Leicester, Leicester, U.K., in 1992, the M.Sc. degree in telecommunications technology from Aston University, Birmingham, U.K., in 1994, and the Ph.D. degree in electronic and electrical engineering from University College London, London, U.K., in 1998.

He is a Professor in the University College London Department of Electronic and Electrical Engineering, where he leads the Analog and Biomedical Electronics Group. He has published over 190 articles in journals and international conference proceedings. His current research interests include analog and mixed-signal integrated circuits for biomedical, sensor, and signal-processing applications.

Dr. Demosthenous is the Deputy Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS and an Associate Editor for the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS. He is a member of the Technical Programme Committee of several IEEE conferences including ESSCIRC and VLSI-SoC. He was on the organizing committee of the 2013 IEEE Biomedical Circuits and Systems Conference (BioCAS 2013). He is a Fellow of the Institution of Engineering and Technology and a Chartered Engineer.