Feature Extraction using First and Second Derivative Extrema (FSDE) for Real-time and Hardware-Efficient Spike Sorting

Sivylla E. Paraskevopoulou[a,b], Deren Y. Barsakcioglu[a,b], Mohammed R. Saberi[a], Amir Eftekhar[a,b], and Timothy G. Constandinou[a,b]

[a] *Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK*
[b] *Centre for Bio-Inspired Technology, Institute of Biomedical Engineering, Imperial College London, SW7 2AZ, UK*
Email:{s.paraskevopoulou09, deren.barsakcioglu10, mohammad.saberi06, a.eftekhar, t.constandinou}@imperial.ac.uk

Abstract

Next generation neural interfaces aspire to achieve real-time multi-channel systems by integrating spike sorting on chip to overcome limitations in communication channel capacity. The feasibility of this approach relies on developing highly-efficient algorithms for feature extraction and clustering with the potential of low-power hardware implementation. We are proposing a feature extraction method, not requiring any calibration, based on first and second derivative features of the spike waveform. The accuracy and computational complexity of the proposed method are quantified and compared against commonly used feature extraction methods, through simulation across four datasets (with different single units) at multiple noise levels (ranging from 5 to 20% of the signal amplitude). The average classification error is shown to be below 7% with a computational complexity of $2N - 3$, where N is the number of sample points of each spike. Overall, this method presents a good trade-off between accuracy and computational complexity and is thus particularly well-suited for hardware-efficient implementation.

Keywords: feature extraction, spike sorting, derivatives, hardware implementable

1. Introduction

Neural prostheses (or neuroprosthetics) have already demonstrated a significant impact to individuals with damage/dysfunction to sensory and ‘cognitive’ pathways (Humayun et al., 2003; Georgiou and Toumazou, 2005; Perlmutter and Mink, 2006; Constandinou et al., 2008, 2009). For example, cochlear implants, deep brain stimulation devices and more recently, vagus nerve stimulation therapy are now in use in over 350,000 individuals.[1] On the other hand, neural prostheses that interface to the motor pathway (i.e. monitoring devices) have not yet delivered such success with therapeutic devices in clinical applications (i.e. in human populations). However, instruments that facilitate electrophysiological recording in the brain are now commonplace in neuroscience laboratories. Here, brain machine interfaces (BMIs) provide a unique opportunity to deliver next generation research tools by providing more channels (Stevenson and Kording, 2011) with the capability of local/real-time processing at lower power consumption and within a miniature form factor.

Realising such devices is now possible thanks to recent advances in microtechnology (i.e. silicon probes and CMOS integrated circuits). This has allowed both electronics and electrodes to be scalable and manufacturable with sub-micron resolution features. Systems including multi-electrode arrays (MEAs) (Maynard et al., 1997) and multi-channel neural interface chips (Harrison et al., 2007; Rodriguez-Perez et al., 2012; Paraskevopoulou and Constandinou, 2012; Gao et al., 2012) can now be engineered for recording single unit activity with excellent precision, repeatability and reliability.

However, the requirement for such devices to be implantable sets critical restrictions particularly on the electronics design. Although no extensive study has been performed on the effect of long term device implantation, the currently cited limit on thermal dissipation sets a maximum of 1^{o}C temperature increase and 40 mW/cm^2 heat flux (Wolf, 2008). This in itself sets a very stringent power budget. Other constraints include the limited implant size, battery capacity/lifetime and communication bandwidth. Thus, in or-

[1] NICDC/NIH Publication No. 11-4798 specifies 219,000 cochlear implants, Medtronic and Cyberonics state 80,000 DBS and 60,000 VNS devices respectively (on their corporate website).

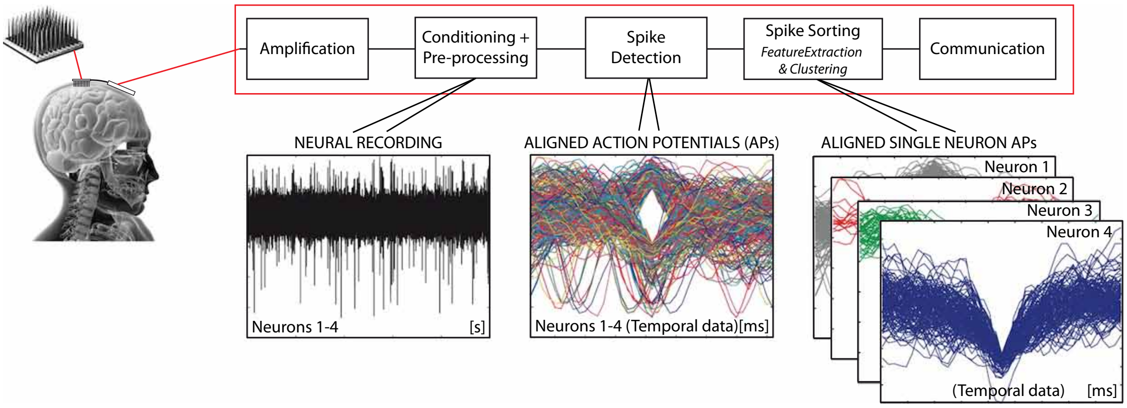

Amplification
Conditioning + Pre-processing
Spike Detection
Spike Sorting
FeatureExtraction & Clustering
Communication
NEURAL RECORDING
ALIGNED ACTION POTENTIALS (APs)
ALIGNED SINGLE NEURON APs
Neurons 1-4
[s]
Neurons 1-4 (Temporal data)[ms]
Neuron 1
Neuron 2
Neuron 3
Neuron 4
(Temporal data)
[ms]

Figure 1: A typical single channel architecture for recording extracellular action potentials and determining single unit activity.

der to achieve a scalable high channel count while maintaining a high resolution of recording, the information transmitted must be of high quality (i.e. equivalent to extracting the raw data), and the information extraction methods must be hardware implementable in ultra-low power electronics (Eftekhar et al., 2010).

Each electrode can observe activity from up to 5-10 units (Pedreira et al., 2012; Buzsáki and Draguhn, 2004). Thus by being able to identify the neuron firing, a high information quality can be maintained, with reduced data rate. The information extraction chain that achieves this is illustrated in Fig. 1. In a conventional system, the analogue front-end provides the signal amplification and (depending on implementation methodology) conditioning (filtering), and digitisation (data conversion). The rationale to implementing more functions within the analogue front-end, is that resource (i.e. power, size and data) requirements can be significantly reduced. This is however at the expense of reduced versatility, i.e. software can be completely reprogrammed, whereas power-optimised custom hardware tends to be function specific. The subsequent spike sorting involves spike detection followed by feature extraction and classification.

Feature extraction simplifies the classification process of spikes by selecting the features that best describe them. This leads to dimensionality reduction; instead of comparing n variables, where n is the number of sample points, we have m variables ($m < n$) , where m is the number of selected features. For example, some basic features include the amplitude, width, power, or phase of the spike (Balasubramanian and Obeid, 2011). Typical analytical methods to obtain features are: principal components analysis (PCA) (Abeles and Goldstein Jr, 1977), discrete wavelet transform (DWT) (Letelier and Weber, 2000), discrete derivatives (DD) (Gibson et al., 2008), template matching (TM) (Yuan et al., 2012; Zhang et al., 2004), zero-crossing features (ZCF) (Kamboh and Mason, 2012) and integral transform (IT) (Zviagintsev et al., 2005).

Once the features have been extracted they can be projected into their m-dimensional space, usually referred to as classification (or clustering). There exist several clustering algorithms (based on different partitioning methods) including k-means clustering (Chan et al., 2008), Bayesian clustering (Dai et al., 2008), valley detection (Kim and McNames, 2007; Mahmud et al., 2012), superparamagnetic clustering (Quiroga et al., 2004), expectation maximization (Harris et al., 2000; Pouzat et al., 2002) and artificial neural networks (ANN). A comprehensive review of clustering methods is provided in (Xu and Wunsch, 2010).

The overall effective complexity of spike sorting depends on two factors: the computational complexity of the specific feature extraction and clustering algorithms and the number of dimensions in feature space.

This paper proposes a new feature extraction method, based on the first and second derivatives of the spike waveform. This method is shown to achieve a relatively high accuracy, however with a low computational complexity, and thus is suitable for low power hardware implementation. Another advantage is that it runs in real-time, not requiring any off-line training. This is particularly useful for when monitoring a large number of channel as no calibration is necessary.

1.1. Spike modelling

Our initial motivation for the use of derivative features is based on the dipole model of an action potential(Mechler and Victor, 2012). Eq. 1 describes the potential generated by a dipole of charge $\overrightarrow{p}$ at location $\overrightarrow{r}$ from the centre of the dipole.

$$V(\overrightarrow{r}) = \frac{\overrightarrow{p}\,\overrightarrow{r}}{4\pi\epsilon_0 r^3} = \frac{pr \cdot cos\theta}{4\pi\epsilon_0 r^3} \tag{1}$$

Where p is the dipole operator, r is the electrode-to-neuron-distance, cosθ the cosine of the angle between the two vectors $\overrightarrow{p}$ and $\overrightarrow{r}$, and ϵ_0 the dielectric constant. This dipole is formed by the extracellular current balancing the influx of Na$^+$ cations from the neuronal extracellular to the intracellular space (Buzsáki et al., 2012), thus maintaining equilibrium at the membrane.

The extracellular potentials generated by a spiking pyramidal cell and measured at different locations in it's vicinity are illustrated in Fig. 2. An initial observation is that the orientation of the electrode relative to the spiking neuron changes the spike shape and its distance reduces the amplitude of the spikes.

Based on this observation, we examine the paradigm that a recording electrode and two identical spiking neurons (i.e. with identical morphology) are aligned in the x-direction, but the distance between the electrode and the first neuron is smaller than the distance between the electrode and the second neuron. In that case, the recorded spikes from the distinct neurons will exhibit 'identical' spike shapes but differ in amplitude. In fact, the spike amplitude will be decreasing inversely proportionally to r^2, according to Eq. 1. As the two recorded spikes have the same expansion in time and different amplitude, the spike amplitude can be used as a feature to separate the two spikes. However, this difference will also be evident in their first derivatives, additionally providing some indication to the spike morphology. Therefore, intuitively the first derivative appears to be an eligible feature for spike sorting.

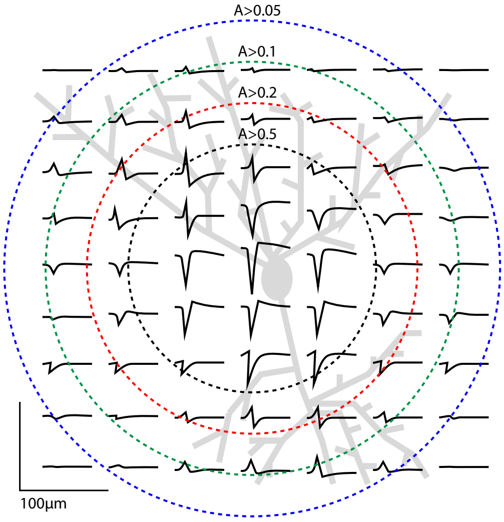


Figure 2: Variation in the observed extracellular action potential profile with spatial position. Shown is an example in the vicinity of a spiking pyramidal cell. Adapted from (Buzsáki et al., 2012). Dotted lines illustrate the regions of relative spike amplitudes larger than 0.5 (black), 0.2 (red), 0.1 (green) and 0.05 (blue).

2. Materials and Methods

The previous section provided an intuitive rationale behind using derivatives to extract spike features for subsequent classification. The use of such features has been shown analytically (Yang et al., 2009), and through simulations (Yang et al., 2009; Karkare et al., 2011). Derivatives have also been proven an efficient filter for noise shaping (highlighting the recorded spiking activity against background spiking activity) (Yang et al., 2009). In this paper, however, we will explore the efficiency of a new feature extraction method based on using both the first and second derivatives. All the methods reported herein are tested using Mathworks Matlab v7.12.

2.1. Proposed method for featured extraction

The derivatives are computed as the difference between the current and previous sample points of the spike waveform (s), according to Eq. 2 for the first derivative (FD) and Eq. 3 for the second derivative (SD).

$$FD(n) = s(n) - s(n-1) \tag{2}$$

$$SD(n) = FD(n) - FD(n-1) \tag{3}$$

The effect of taking the first and second derivatives of typical action potential signals is illustrated in Fig. 3. Here, onto the spike waveform are annotated the occurrences of the following features: (a) negative, and (b) positive peaks of the first derivative (FD$_{min}$ and FD$_{max}$), and (c) negative, and (d) positive peaks of the second derivative (SD$_{min}$ and SD$_{max}$). In this example, it can be clearly observed that these features together can be used to distinguish between these three spike profiles.

The first derivative of any function (or geometric object) can be described by its rate of change (or slope). Therefore, the second derivative is essentially the rate of change of the slope, representing the curvature of the signal. Spikes are characterised by several morphological changes including slopes, curvature and amplitude, all of which are dependent on the observation point relative to the neuron. The first and second derivative can

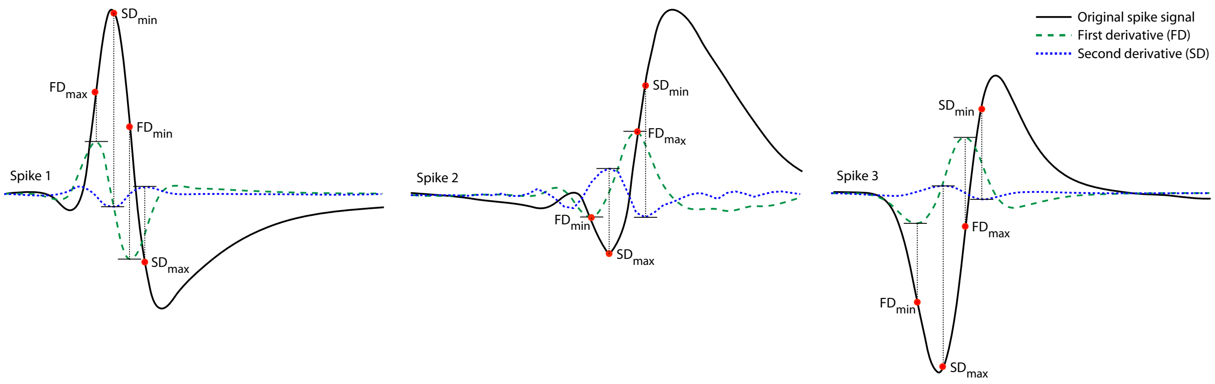


Figure 3: Typical action potential waveforms with their first and second derivatives (FD and SD). Annotated onto the original spike waveforms are the points corresponding to the extrema (i.e. minimum and maximum) of the first and second derivatives.

therefore provide, directly and indirectly, information on all three of these expected morphological variations.

2.2. *Other feature extraction methods (for comparison)*

In order to validate the viability of the proposed method (for hardware spike sorting), we compare (Section 3) its accuracy and computational complexity to other low complexity and derivative-based feature extraction methods, in addition to to Principal Component Analysis (PCA), which is widely regarded as a 'gold standard' for spike sorting.

2.2.1. *PCA*

PCA is a technique that extracts linearly uncorrelated components from the input signal. These principal components are the eigenvectors of the autocorrelation matrix of the original signal, and their number is equal to the number of samples. However, the most significant information is contained in the first few.

2.2.2. *Discrete derivatives (DD)*

DD is based on computing the derivatives at each sample point of the spike waveform according to Eq. 4 (Gibson et al., 2008).

$$dd_\delta(n) = s(n) - s(n-\delta) \tag{4}$$

Where s is the spike signal, n the sample point, and δ the time delay. Previous work used three different values for the delay: $\delta = 1, 3, 7$ (Karkare et al., 2011). Therefore, the dimensionality of the feature space increases by a factor of three compared to the original number of samples of the spikes. To decrease the dimensionality, the most significant DD coefficients are selected. This method has already been shown to be hardware implementable, demonstrated on a 64-channel DSP (Karkare et al., 2011).

2.2.3. *First derivative (FD)*

A method based on the derivation of the spike waveform in the discrete domain is proposed in (Yang et al., 2008). The selected features are the positive and negative peak of the spike first derivative, and the peak of the spike itself. In this paper, we will refer to this work as the FD method.

2.2.4. *Autoregressive model (AR)*

It has been shown that the spike signal can be modelled using an autoregressive (AR) model (described by Eq. 5) of p-th order (the coefficients are computed using the Burg algorithm), to separate the signal from background activity (Chan et al., 2006). In this paper, we investigate the efficiency of spike sorting when the features used are the coefficients of the AR model for each spike.

$$x_t = c + \sum_{(i=1)}^{p} \phi_i x_{t-1} + \epsilon_t \tag{5}$$

Where $\phi_1, ..., phi_p$ are the parameters of the model, c is a constant and ϵ_t is white noise.

2.2.5. *Point-to-point (PP) comparison*

A brute force approach to spike sorting is to use the sampled directly instead of extracting features (i.e. with no dimensionality reduction). This method is herein referred to as point-to-point comparison (PP).

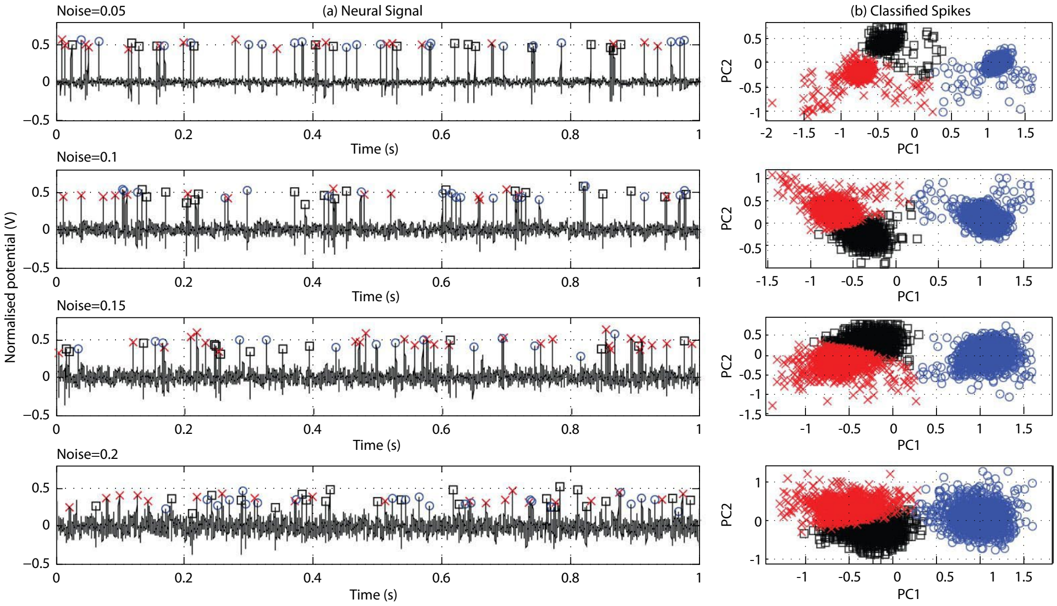


Figure 4: Test dataset 2 showing: (a) a 1 s segment at noise levels of 0.05 to 0.2 (from top to bottom), and (b) a two dimensional representation of the clustering space when using three principle components (z-axis is 3rd principle component - not shown).

2.3. *Classification (using k-means clustering)*

Once the features have been extracted, *k*-means clustering is utilised to separate the features and differentiate the spikes seen at an electrode. *k*-means clustering is a method that aims to partition the spikes (using the selected feature space) into *k* clusters, in which each spike belongs to the cluster with the nearest mean, as described by Eq. 6.

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} ||x_n - \mu_j||^2 \tag{6}$$

Where J is the objective function (i.e. the squared error function), n is the number of spikes, (x_1, x_2.. x_n) is the set of spike features (each of m-dimension), K is the number of clusters ($K \leq n$) and S_j are the different sets (i.e. spike classes). *k*-means aims to determine the clusters such as to minimise the objective function J.

This is generally implemented as an iterative algorithm that converges towards the solution. Depending on the feature types (and their dimensionality) the number of iterations required for convergence will vary. For all the classification presented herein, the Matlab ('kmeans') function was used with the number of iterations set to 10. This has been used to ensure that all methods converge to a near-optimum classification accuracy.

2.4. *Test datasets*

To evaluate the spike sorting accuracy for various feature extraction methods we are use simulated neural signals. The datasets (Quiroga, 2006) were originally used in (Quiroga et al., 2004) to evaluate spike sorting with wavelets and superparamagnetic clustering. The spike signals have been randomly selected from a database of over 594 waveforms recorded from the neocortex and basal ganglia in humans. Noise has been added onto the generated dataset by superimposing random spike signals to emulate the background activity. These have been randomly distributed at a ratio of 2:1 (spikes to 'noise'). The advantage of using simulated data is that the *ground truth* is established and is therefore particularly useful for testing different detection and sorting approaches.

For our simulations we are using four of these datasets, each containing 3 different types of spike shapes and at 4 different noise levels. The input signals have been normalised and the standard deviation of the added noise (i.e. inverse of the signal-to-noise ratio) is 0.05, 0.01, 0.15 and 0.2. An example portion of Dataset 2 is shown in Fig 4, at the different noise levels (from top to bottom). For each noise level is shown the raw dataset (on left) and cluster plots (on right) using the first three principal features. Also annotated (in red, blue and black) are the different spike classes (as known *a priori*). It should be noted that the although only a 1 s

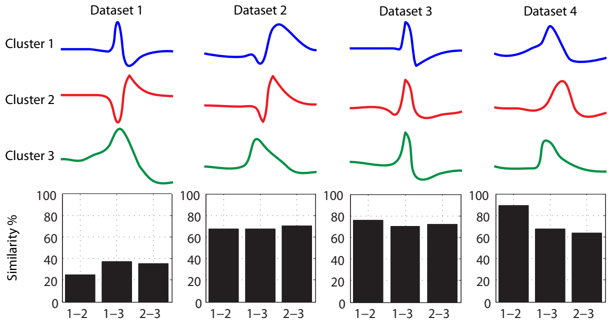


Figure 5: Single unit waveforms used within each dataset. Shown are the mean (peak-aligned) templates with the degree of separability, measured using the Bray-Curtis similarity index (shown at bottom).

portion of the inputs is shown, the cluster plots include all the spikes from the respective datasets. It can be observed that increasing the noise level (i.e. decreasing the SNR) adversely affects the cluster separation.

The spike waveforms for each of the 3 classes in the 4 test dataset are shown in Fig. 5. These have been generated by taking a mean across all spikes in each cluster. The spikes waveforms have been extracted using a 64 sample window aligned to the spike peak (set at sample 20). It can be observed the these datasets have varying levels of spike sorting difficulty (i.e. spikes with similar waveforms are harder to spike sort). Within this context, the Bray-Curtis similarity (Eq. 7) index (Lian et al., 2010) is used to quantify the separability of the spikes in each dataset.

$$S_{x,y} = 1 - \frac{\sum_{i=1}^{N} |x(i) - y(i)|}{\sum_{i=1}^{N} |x(i)| + |y(i)|} \tag{7}$$

Where x and y are the two spike waveforms being compared and N is the number of sample points. $S_{x,y}$ is in the range $(0 - 1)$, with 1 corresponding to identical signals. For each data set, the similarity measure between all three spike templates in a dataset are calculated and presented along with mean spike waveforms in Fig. 5.

3. Results and Discussion

In this section we apply the proposed method to spike sorting and compare its performance (accuracy) and computational complexity to the other hardware implementable methods described in the previous section.

3.1. Classification accuracy for different derivative feature combinations

In order to establish the optimum efficiency of using first and second derivative features (derived as described in Section 2.1), the classification accuracies between different feature combinations are compared. The six permutations tested are as follows:

- Method 1: Both the negative and positive peaks of the first derivative (FD$_{min}$ and FD$_{max}$), together with the negative peak of the second derivative (SD$_{min}$).
- Method 2: Both the negative and positive peaks of the first derivative (FD$_{min}$ and FD$_{max}$), together with the positive peak of the second derivative (SD$_{max}$).
- Method 3: Negative peak of the first derivative (FD$_{min}$), together with both the negative and positive peaks of the second derivative (SD$_{min}$ and SD$_{max}$).
- Method 4: Positive peak of the first derivative (FD$_{max}$), together with both the negative and positive peaks of the second derivative (SD$_{min}$ and SD$_{max}$).
- Method 5: Range of the first derivative (i.e. (FD$_{max}$-FD$_{min}$), together with range of the second derivative (i.e. SD$_{max}$-SD$_{min}$).
- Method 6: Average of the first derivative (i.e. (FD$_{min}$+FD$_{max}$)/2) together with average of the second derivative (i.e. (SD$_{min}$+SD$_{max}$)/2).
- Method 7: All FSDE features, i.e. both the negative and positive peaks of the first and second derivatives (FD$_{min}$, FD$_{max}$, SD$_{min}$ and SD$_{max}$).

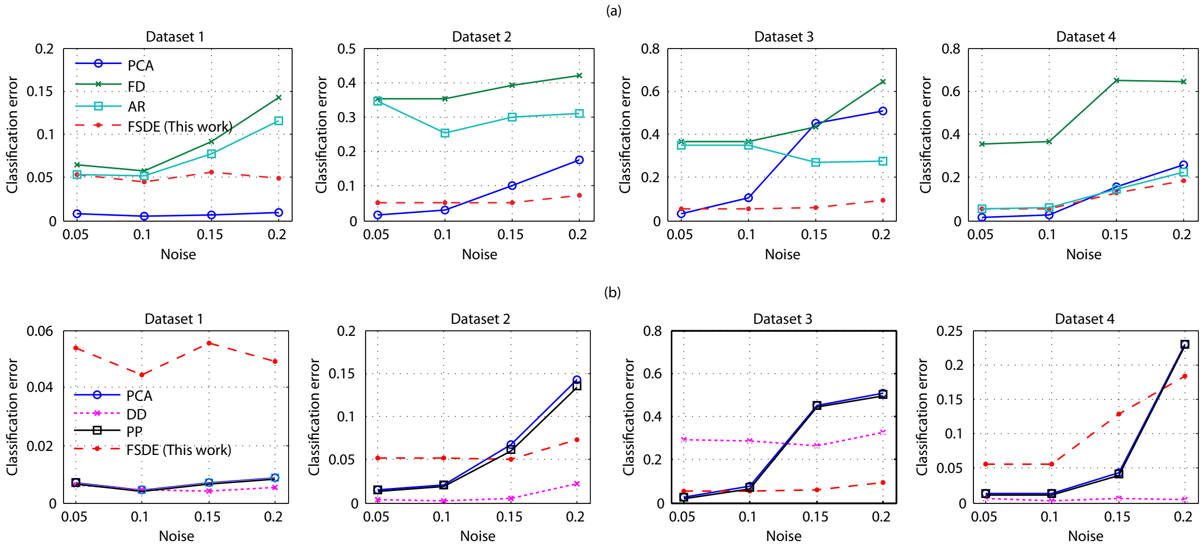
The main performance criteria that is quantified is the classification error, given by the number of misclassified spikes over the total number of detected spikes in each input signal. This must be balanced however, with the relative complexity of each method, i.e. the number of features taken. The results are presented in Table 1. The total error stated has been quantified as the average classification error across all datasets and noise levels.

The results show that Methods 6, 4, and 7 achieve the lowest classification errors for 2, 3 and 4 features respectively. As can be expected, the more features that are used, the lower the classification error. Assuming however, that the relative complexity between these methods is proportional to the number of features, and then comparing the relative classification errors, a much greater improvement in accuracy is observed when moving from 2 to 3 features than from 3 to 4.

Table 1: Comparison of the classification error for different combinations of derivative features. Results given across all the tested datasets and for all noise levels.

Dataset	Noise†	Classification Error						
		Method 1 (*m*=3)*	Method 2 (*m*=3)*	Method 3 (*m*=3)*	Method 4 (*m*=3)*	Method 5 (*m*=2)*	Method 6 (*m*=2)*	Method 7 (*m*=4)*
Dataset 1	0.05	0.0544	0.0561	0.0546	0.0538	0.0544	0.0572	0.0544
	0.10	0.0443	0.0468	0.0451	0.0446	0.0457	0.0477	0.0446
	0.15	0.0552	0.0581	0.0546	0.0555	0.0555	0.0601	0.0552
	0.20	0.0489	0.0535	0.0498	0.0492	0.0507	0.0596	0.0492
Dataset 2	0.05	0.0531	0.3457	0.0531	0.0519	0.0522	0.0575	0.0528
	0.10	0.0528	0.1247	0.0585	0.0517	0.0699	0.0622	0.0526
	0.15	0.0630	0.1850	0.0718	0.0504	0.2987	0.0988	0.0627
	0.20	0.0757	0.2283	0.0993	0.0729	0.3369	0.1469	0.0822
Dataset 3	0.05	0.0562	0.0437	0.0553	0.0550	0.3491	0.1339	0.0559
	0.10	0.0525	0.0415	0.0519	0.0522	0.3457	0.1180	0.0522
	0.15	0.0628	0.0441	0.0576	0.0619	0.3491	0.1489	0.0565
	0.20	0.0682	0.0539	0.0665	0.0940	0.3453	0.2293	0.0636
Dataset 4	0.05	0.3517	0.0386	0.3496	0.0562	0.3505	0.0612	0.0552
	0.10	0.0477	0.0576	0.0511	0.0552	0.3463	0.0661	0.0474
	0.15	0.0974	0.1180	0.3488	0.1282	0.3535	0.1308	0.0968
	0.20	0.1970	0.3458	0.2906	0.1829	0.3561	0.2030	0.1575
Total		**0.0863**	**0.1151**	**0.1099**	**0.0697**	**0.2350**	**0.1045**	**0.0649**

†Noise level relative to a normalised signal (i.e. noise=1/SNR), *m=number of features/dimensions


(a)

(b)

Figure 6: Comparison of the classification error between commonly used feature extraction methods (across all datasets and noise levels). Shown are comparison to methods with: (a) the same feature space dimensionality, and (b) different (higher) feature space dimensionality.

Method 4 is therefore chosen as it strikes a good trade-off between classification error and number of features, and additionally exhibits good noise immunity. The combination of features used in Method 4 are selected and will from hereon be referred to as the First and Second Derivative Extrema (FDSE) method.

3.2. Classification accuracy using FSDE features compared to other feature extraction methods

As mentioned in Section 1, the size of the feature space (m) directly impacts the computational complexity of spike sorting. Therefore, the classification accuracy of FSDE feature extraction with k-means has been considered separately for methods with the same and different m, shown in Figs. 6(a) and 6(b) respectively. The results have been averaged across all noise levels.

- Methods with the same feature space as FSDE (i.e. $m \leq 3$): The selected methods are FD, PCA$_3$, and AR$_3$ (i.e. subscript denoting the number of coefficients).

- Methods with a higher feature space than FSDE (i.e. $m > 3$): The selected methods are PCA$_{10}$, DD$_{21}$, and PP$_{64}$. For PCA, we have limited the number of coefficients to 10, as we have determined a negligible improvement (in classification error) for more coefficients. For DD, we are using 21 coefficients, which has been reported to be optimum (Karkare et al., 2011). The selection of coefficients is achieved through a blind training of 300 spikes per dataset (approximately 10% of the signal) by identifying the coefficients with the largest variance. Finally, for PP we are using all the sample points (i.e. m=64).

The results in Fig. 6 show that for methods of the same feature space as FSDE, it generally achieves higher classification accuracy than the other methods. However, for methods with a higher feature space, they generally achieve a higher classification accuracy. Another observation is that the FSDE and DD methods (across all feature space) exhibits the best noise immunity amongst all methods.

Furthermore, by defining the dimensionality factor as the number of features over the number of samples in the spike waveform (i.e. m/n), the relationship between dimensionality and classification error can be directly established. This is illustrated in Fig. 7, showing the classification error averaged across all datasets and noise levels. It can be observed that the classification error of FSDE is comparable to DD (i.e. 21 coefficients with maximum variance) and PCA (despite a significantly lower complexity), and is only larger than that of DD, when using all coefficients (m=181).

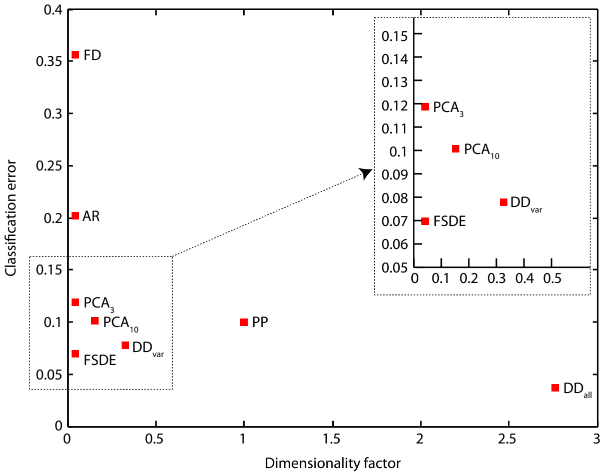


Figure 7: Observing the effect of dimensionality on classification error for the different feature extraction methods used in this work.

3.3. Clustering FSDE features using k-means

To demonstrate the effectiveness of spike sorting using FSDE features (with 10-iteration k-means), the actual clusters are compared to those determined. This is shown in Fig. 8, also illustrating the windowed spike waveforms (and corresponding mean templates) of the actual and determined classes.

3.4. Computational complexity

Within the context of implantable spike sorting hardware, computational complexity is as important as the classification accuracy of the spike sorting method. This is due to previously mentioned resource constraints (i.e. power, bandwidth and volume). We define the computational complexity in terms of the number arithmetic operations (i.e. additions, multiplications, etc) required to compute each feature. Table 2 estimates the relative complexity of each feature extraction method, along with any additional computation required for selecting a subset of features (i.e. in dimensionality reduction).

As mentioned previously, k-means has been chosen as the clustering algorithm used in our simulations. Since the k-means algorithm is mainly dominated by calculating point to cluster mean distances (using Squared Euclidean metric), its computational complexity is estimated to be $k(2m-1)$ additions and km multiplications per spike, where m is the number of features per spike and k is the number of clusters.

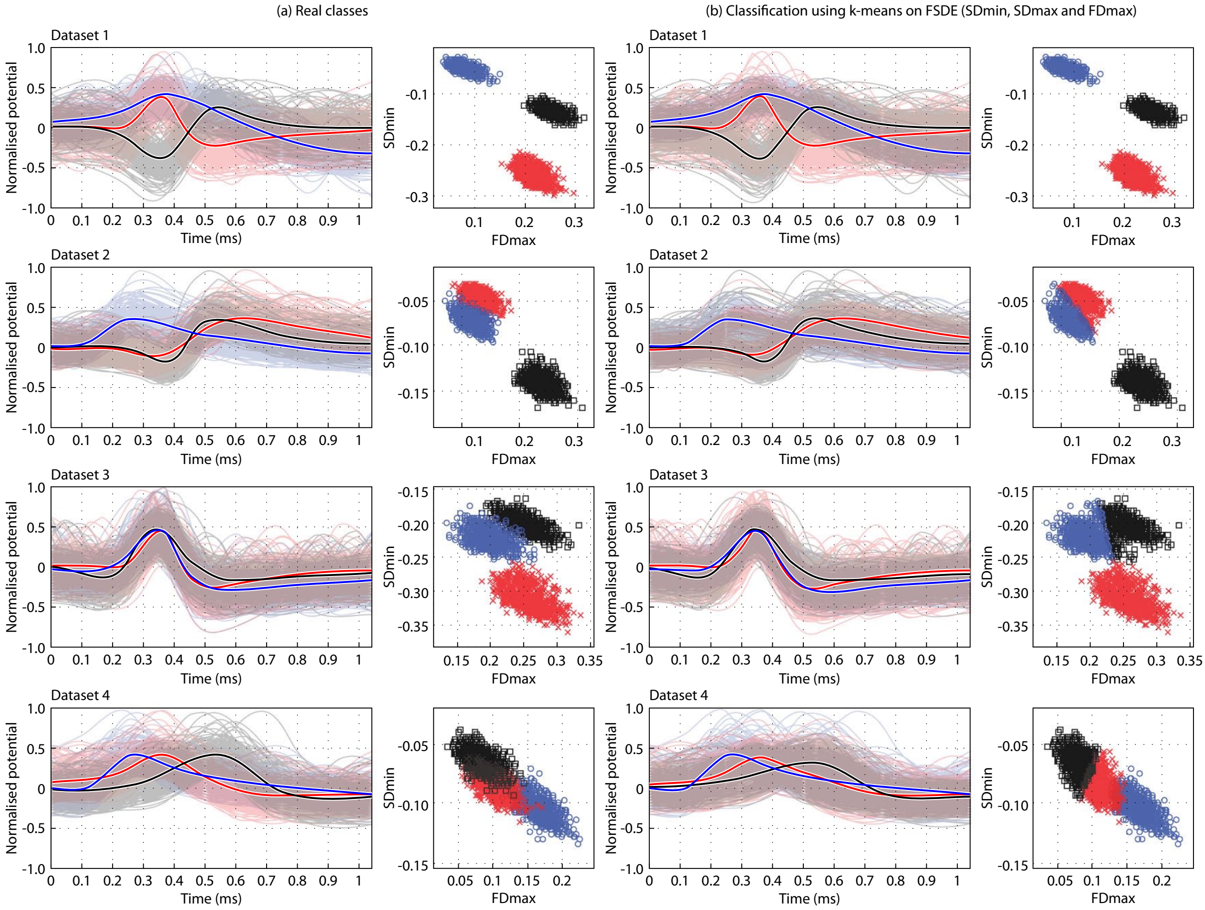


Figure 8: Spike classification utilising FSDE features. Shown are: (a) the actual classes (as specified in each dataset at a noise level of 0.2), and (b) classes as generated using FSDE features with *k*-means clustering. Shown for both sets are the spike waveforms (with mean templates overlaid) and clusters in FSDE feature space (z-axis is SD$_{max}$ - not shown). In each case, the 3 different clusters are coloured in red, blue and black.

Table 2: Computational complexity of each feature extraction method (in terms of the number of arithmetic operations).

Method	**Additions**	**Multiplications**	**Dimensionality Reduction**
FD	N-1	-	-
SD	2N-3	-	-
DD†	3N - 11	-	(3N-11)(1-5T) additions, T(6N-22) multiplications‡
PP	-	-	-
AR	$[N-1] + \sum_{i=1}^{p} 5(N-i) + i + 1$	$[N+1] + \sum_{i=1}^{p} 5(N-i) + i + 3$	-
PCA	$N^2 + 2N + 1$	$N^2 + N$	-

N is the number of samples in each spike, T is the number of spikes for training, and p is the model order for the Autoregressive model.
†Additional complexity associated for dimensionality reduction, ‡variance calculation

In order to assess the accuracy and computational complexity of different feature extraction methods, the aforementioned factors contributing to computational demands has been combined to produce a single complexity figure-of-merit (CFOM) as in (Gibson et al., 2008; Zviagintsev et al., 2005):

$$C = N_{add} + 10 \times N_{mult} \tag{8}$$

where N_{add} is the number of additions, and N_{mult} is the number of multiplications required. It should be noted that, only the complexities associated with feature extraction and clustering are considered in CFOM. Any Computation associated with choosing a subset of features extracted (such as variance calculations as in DD) are only calculated once, and therefore not included. CFOM is given (per spike) assuming 64-sample spikes (n=64) of three different clusters (k=3), and a fourth order autoregressive model (p=4).

Fig. 9 illustrates the trade-off between classification error and computational complexity. Here, it can be clearly observed that the FSDE method achieves both a good classification accuracy and low computational complexity. In fact, the only method (from those compared) that outperforms FSDE is D$_{all}$, which has a 35× higher computational complexity. On the other hand, when compared to DD_{var} (considered to be both accurate and computationally efficient), the FSDE method has a 10% lower classification error with 4× lower complexity. In other words, amongst all the feature extraction methods considered, FSDE provides the best trade-off between accuracy and complexity.

Additionally, the number of features (i.e. dimensions) selected to describe the dataset indirectly impact the complexity through the added resource required for the increased clustering computation. Therefore methods with fewer features are preferable to those with more. As FDSE uses only 3 features, this further reduces its computational complexity compared to methods with higher dimensionality.

4. Conclusion

In this paper, we have proposed a new feature extraction method for spike sorting, based on first and second derivative extrema (FSDE) features. Specifically, the selected method utilises the (positive) peak of the first derivative together with the positive and negative peaks of the second derivatives. This method has been shown to achieve a classification error of only 6.97% (averaged across 4 datasets with varying degrees of sorting difficulty and noise levels), for a computational complexity of $2N - 3$.

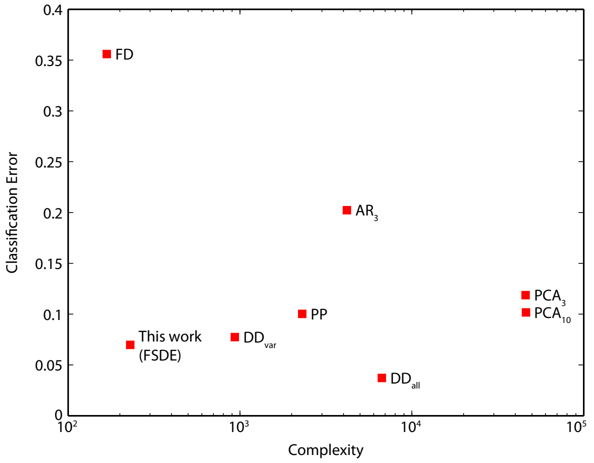


Figure 9: Comparing classification error (averaged over all datasets and noise-levels) with computational complexity for the different feature extraction methods considered herein.

We have tested the classification accuracy of k-means for different feature extraction methods. To ensure sufficient convergence for all feature extraction methods, the number of k-means iterations is set at 10. The proposed method outperforms PCA$_3$ (which achieves an average classification error of 10.17%), with a 100× lower computational complexity. Furthermore, when compared to another hardware-efficient derivative-based method (i.e. DD with dimensionality reduction), the proposed method performs marginally better (DD_{var} average classification error is 7.74%), with a 3× lower computational complexity. This unique balance between good accuracy and low complexity makes the FSDE method a particularly good candidate for hardware (i.e. on-chip) spike sorting.

Acknowledgments

This work was in part supported by the UK Engineering and Physical Sciences Research Council (EPSRC Grant ref. EP/I000569/1 and DTA award).

Appendix A. Abbreviations in the paper

Here is an alphabetical list of abbreviations used in this paper.

ANN	Artificial Neural Networks
AR	Auto-Regressive
BMI	Brain Machine Interface
CFOM	Complexity Figure-of-Merit
DD	Discrete Derivative
DSP	Digital Signal Processor
DWT	Discrete Wavelet Transform
FD	First Derivative
FSDE	First and Second Derivative Extrema
IT	Integral Transform
MEA	Multi-Electrode Array
PCA	Principle Component Analysis
PP	Point-to-Point
SD	Second Derivative
SNR	Signal-to-Noise Ratio
ZCF	Zero Crossing Features

References

Abeles M, Goldstein Jr M. Multispike train analysis. Proceedings of the IEEE 1977;65(5):762–773.

Balasubramanian K, Obeid I. Fuzzy logic-based spike sorting system. Journal of neuroscience methods 2011;198(1):125–134.

Buzsáki G, Anastassiou C, Koch C. The origin of extracellular fields and currentseeg, ecog, lfp and spikes. Nature Reviews Neuroscience 2012;13(6):407–420.

Buzsáki G, Draguhn A. Neuronal oscillations in cortical networks. Science 2004;304(5679):1926–1929.

Chan H, Lin M, Wu Y, Lai H, Lee S, Yang J, Fang S. Detection and characterization of neural spikes. In: Proc. IET Conference on Advances in Medical, Signal and Information Processing (MEDSIP). 2006. p. 1–4.

Chan H, Wu T, Lee S, Fang S, Chao P, Lin M. Classification of neuronal spikes over the reconstructed phase space. Journal of neuroscience methods 2008;168(1):203–211.

Constandinou T, Georgiou J, Toumazou C. A fully-integrated semicircular canal processor for an implantable vestibular prosthesis. In: Proc. IEEE Int. Conf. Electronics, Circuits and Systems (ICECS). 2008. p. 81–84.

Constandinou T, Georgiou J, Toumazou C. A neural implant asic for the restoration of balance in individuals with vestibular dysfunction. In: Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS). 2009. p. 641–644.

Dai J, Liu X, Yi Y, Zhang H, Wang J, Zhang S, Zheng X. Experimental study on neuronal spike sorting methods. In: Proc. IEEE International Conference on Future Generation Communication and Networking. volume 2; 2008. p. 230–233.

Eftekhar A, Paraskevopoulou S, Constandinou T. Towards a next generation neural interface: Optimizing power, bandwidth and data quality. In: Proc. IEEE Biomedical Circuits and Systems Conference (BioCAS). 2010. p. 122–125.

Gao H, Walker R, Nuyujukian P, Makinwa K, Shenoy K, Murmann B, Meng T. Hermese: A 96-channel full data rate direct neural interface in 0.13¡ formula formulatype=. IEEE Journal of Solid-State Circuits 2012;47(4):1043–1055.

Georgiou J, Toumazou C. A 126-μw cochlear chip for a totally implantable system. Solid-State Circuits, IEEE Journal of 2005;40(2):430–443.

Gibson S, Judy J, Markovic D. Comparison of spike-sorting algorithms for future hardware implementation. In: Proc. IEEE International Conference of the Engineering in Medicine and Biology Society (EMBS). 2008. p. 5015–5020.

Harris K, Henze D, Csicsvari J, Hirase H, Buzsáki G. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. Journal of Neurophysiology 2000;84(1):401–414.

Harrison R, Watkins P, Kier R, Lovejoy R, Black D, Greger B, Solzbacher F. A low-power integrated circuit for a wireless 100-electrode neural recording system. IEEE Journal of Solid-State Circuits 2007;42(1):123–133.

Humayun M, Weiland J, Fujii G, Greenberg R, Williamson R, Little J, Mech B, Cimmarusti V, Van Boemel G, Dagnelie G, et al. Visual perception in a blind subject with a chronic microelectronic retinal prosthesis. Vision research 2003;43(24):2573–2581.

Kamboh A, Mason A. Computationally efficient neural feature extraction for spike sorting in implantable high-density recording systems. IEEE Transactions on Neural Systems and Rehabilitation Engineering 2012;.

Karkare V, Gibson S, Markovic D. A 130-μw, 64-channel neural spike-sorting dsp chip. IEEE Journal of Solid-State Circuits 2011;46(5):1214–1222.

Kim S, McNames J. Automatic spike detection based on adaptive template matching for extracellular neural recordings. Journal of neuroscience methods 2007;165(2):165–174.

Letelier J, Weber P. Spike sorting based on discrete wavelet transform coefficients. Journal of neuroscience methods 2000;101(2):93–106.

Lian J, Garner G, Muessig D, Lang V. A simple method to quantify the morphological similarity between signals. Signal Processing 2010;90(2):684–688.

Mahmud M, Bertoldo A, Girardi S, Maschietto M, Vassanelli S. Sigmate: A matlab–based automated tool for extracellular neuronal signal processing and analysis. Journal of neuroscience methods 2012;.

Maynard E, Nordhausen C, Normann R. The utah intracortical electrode array: A recording structure for potential brain-computer interfaces. Electroencephalography and clinical Neurophysiology 1997;102(3):228–239.

Mechler F, Victor J. Dipole characterization of single neurons from their extracellular action potentials. Journal of computational neuroscience 2012;32(1):73–100.

Paraskevopoulou S, Constandinou T. An ultra-low-power front-end neural interface with automatic gain for uncalibrated monitoring. In: Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS). 2012. p. 193–196.

Pedreira C, Martinez J, Ison M, Quiroga R. How many neurons can we see with current spike sorting algorithms? Journal of Neuroscience Methods 2012;211:58–65.

Perlmutter J, Mink J. Deep brain stimulation. Annu Rev Neurosci 2006;29:229–257.

Pouzat C, Mazor O, Laurent G. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. Journal of neuroscience methods 2002;122(1):43–57.

Quiroga R. last accessed: November 2012. *wwwviscaltechedu/rodri/Wave_clus/Wave_clus_homehtm* 2006;.

Quiroga R, Nadasdy Z, Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Computation 2004;16(8):1661–1687.

Rodriguez-Perez A, Ruiz-Amaya J, Delgado-Restituto M, Rodriguez-Vazquez A. A low-power programmable neural spike detection channel with embedded calibration and data compression. IEEE Transactions on Biomedical Circuits and Systems 2012;6(2):87–100.

Stevenson I, Kording K. How advances in neural recording affect data analysis. Nature neuroscience 2011;14(2):139–142.

Wolf P. Thermal considerations for the design of an implanted cortical brain-machine interface (bmi). Indwelling Neural Implants: Strategies for Contending with the In Vivo Environment 2008;:33–38.

Xu R, Wunsch D. Clustering algorithms in biomedical research: A review. Biomedical Engineering, IEEE Reviews in 2010;3:120–154.

Yang Z, Chen T, Liu W. A neuron signature based spike feature extraction algorithm for on-chip implementation. In: Proc. IEEE International Conference of the Engineering in Medicine and Biology Society (EMBS). 2008. p. 1716–1719.

Yang Z, Zhao Q, Liu W. Improving spike separation using waveform derivatives. Journal of Neural Engineering 2009;6(4):2–12.

Yuan Y, Yang C, Si J. The m-sorter: An automatic and robust spike detection and classification system. Journal of Neuroscience Methods 2012;210:281–290.

Zhang P, Wu J, Zhou Y, Liang P, Yuan J. Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. Journal of Neuroscience Methods 2004;135(1):55–65.

Zviagintsev A, Perelman Y, Ginosar R. Low power architectures for spike sorting. In: Proc. IEEE International Conference of the Engineering in Medicine and Biology Society (EMBS). 2005. p. 162–165.