Computer Methods and Programs in Biomedicine 179 (2019) 104986

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Computationally efficient fully-automatic online neural spike detection and sorting in presence of multi-unit activity for implantable circuits

Taimoor Tariq[a,*], M. Hashim Satti[a], Hamid M. Kamboh[a], Maryam Saeed[a], Awais M. Kamboh[b,a]

[a] National University of Sciences and Technology, Islamabad, Pakistan
[b] University of Jeddah, Jeddah, Saudia Arabia

ARTICLE INFO

Article history:
Received 3 April 2019
Revised 28 June 2019
Accepted 14 July 2019
Available online xxx

Keywords:
Neural spike sorting
Spike detection
Neural interfaces

ABSTRACT

Background: Spike sorting is a basic step for implantable neural interfaces. With the growing number of channels, the process should be computationally efficient, automatic,robust and applicable on implantable circuits.

New Method: The proposed method is a combination of fully-automatic offline and online processes. It introduces a novel method for automatically determining a data-aware spike detection threshold, computationally efficient spike feature extraction, automatic optimal cluster number evaluation and verification coupled with Self-Organizing Maps to accurately determine cluster centroids. The system has the ability of unsupervised online operation after initial fully-automatic offline training. The prime focus of this paper is to fully-automate the complete spike detection and sorting pipeline, while keeping the accuracy high.

Results: The proposed system is simulated on two well-known datasets. The automatic threshold improves detection accuracies significantly($>15\%$) as compared to the most common detector. The system is able to effectively handle background multi-unit activity with improved performance.

Comparison: Most of the existing methods are not fully-automatic; they require supervision and expert intervention at various stages of the pipeline. Secondly, existing works focus on foreground neural activity. Recent research has highlighted importance of background multi-unit activity, and this work is amongst the first efforts that proposes and verifies an automatic methodology to effectively handle them as well.

Conclusion: This paper proposes a fully-automatic, computationally efficient system for spike sorting for both single-unit and multi-unit spikes. Although the scope of this work is design and verification through computer simulations, the system has been designed to be easily transferable into an integrated hardware form.

1. Introduction

As we enter a new era of technological progress, the upcoming multimodal intracranial recording systems offer high temporal and spatial resolution which is necessary for real time practical brain machine interfaces such as implantable neuro-prosthetics [1,2]. In a general neural acquisition and processing system, a multichannel analog front-end is used to acquire data, essentially in the form of spikes embedded in the background noise. These spikes are a result of the electrical activity of neurons communicating amongst each other and can be considered to be an electrical manifestation of cognitive processes taking place inside the brain. Ensembles of neurons work in cohesion with each other to perform meaningful functions and these require thousands of electrodes to record their activity, resulting in the need of automatic handling of parameters and data obtained through these electrodes. In most practical applications, the neuronal spikes need to be separated from the background noise and then classified on the basis of their source neurons by a process called Spike Sorting. It is the process of associating neuronal action potentials to corresponding source neurons by separating neural recordings into multiple spikes trains based on specific distinguishing features. The general flow of a spike sorting system consists of detection of neural spikes, extraction of

* Corresponding author.
E-mail address: 13beettariq@seecs.edu.pk (T. Tariq).

https://doi.org/10.1016/j.cmpb.2019.104986
0169-2607/

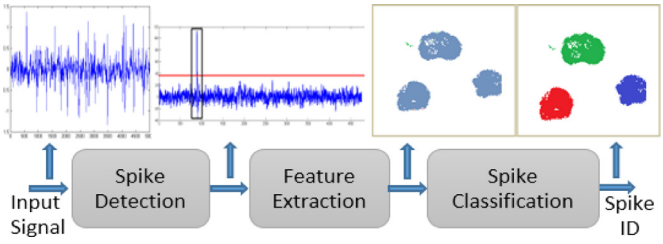


Fig. 1. Fundamental steps of spike sorting.

their distinguishing features and classification of these detected spikes on the basis of these features [3] as shown in Fig. 1

Most applications involving spike sorting such as implantable neuro-prostheses need data from multiple channels at high sampling rates for the processing to be done in real-time. However, it is not only infeasible, but even impossible in some cases to transmit the entire acquired data wirelessly for offline processing due to various constraints and trade-offs [4]. This gives rise to the need for on-chip spike sorting solutions that allow data transmission only when a spike is detected, thus, considerably reducing the transmission bandwidth and power consumption [5].

The characteristics of data being acquired from any of the multiple electrodes in an implanted array may vary considerably under different conditions from time to time. So the neural scientist may need to monitor channels and make adjustments accordingly for each channel and this may require a considerable amount of time and effort when the system extends to hundreds of channels [5–7], and this gives rise to an increased risk of human error as these regular manual interventions can cause error rates of upto 20% [8]. Moreover, it is not feasible for real time practical systems to be monitored and adjusted every time before and during operation. This increases the importance of unsupervised systems which can eliminate the need for manual technical supervision by performing accurately under varying conditions. The basic parameters that are usually required to be calculated automatically for real time processing include: recognition of presence of spike train, transform related parameters for spike detection, spike detection threshold, feature extraction parameters/templates, number of clusters and cluster centroids for accurate classification.

Multi-unit activity is also commonly found in intracranial recordings and represents the activity of several distant neurons whose spikes can be detected but are generally not large enough to be clustered or separated because of the difference in their spike shapes being masked by the background noise [9]. Most spike sorting architectures do not specifically handle multi-unit activity. New research indicates that handling multi-unit spikes is very important as they play a major role in spike sorting applications such as seizure prediction [10]. They have also been shown to be very informative in deciphering the brain's complex time-varying response to stimuli or to clinical insults [11], the analysis of the visual cortex [12] and the scrutiny of motor decision tasks [13]. Moreover, a latest study also shows their efficacy in real time limb state estimation from Dorsal Root Ganglion recordings [14].

In this paper, we propose a computationally efficient automatic spike detection and sorting system, designed to effectively handle multi-unit activity alongwith the single-unit activity, with some added variations provided alongside for applications centered solely on multi-unit activity. We have also compared techniques to determine the optimal number of clusters and concluded that the Gap statistic provides an ideal estimate of this number. A novel cluster verification block is proposed to identify clusters belonging to the same source neurons, thus, identifying and merging clusters that might have split during feature extraction. For the reduction of computational complexity, which has direct impact on area and power requirements for a chip, an offline, online co-design has been proposed in which optimal detection threshold and cluster centroids are initially calculated off-chip using automatic optimal threshold algorithm and Self Organizing Maps (SOM) respectively, and are thereafter transferred onto the chip for further classification. This system has the ability to work either in completely offline mode or online mode, depending on whether the user wants to analyze signals offline or sort them in real-time with the online - offline cooperative nature of the design. Although the scope of this work is design and verification of the proposed system using off-line computer simulations on well known datasets, the system has been designed considering factors such as computational complexity, robustness, breakdown detection and removing the need for human supervision. These factors are of prime importance to the spike sorting problem and its hardware implementability. In summary, the contributions of this paper are:

1. A novel methodology to determine the presence of a valid spike train on the electrodes and act accordingly.
2. A novel automatic spike detection threshold algorithm that is robust and can adapt to the signal.
3. Demonstration that higher order energy operators perform much better in the detection of multi-unit activity.
4. A computationally efficient feature extraction methodology that can effectively handle multi-unit activity.
5. A computationally efficient, fully-automatic spike sorting methodology designed for future hardware implementation.

Some issues that have not been addressed within the scope of this paper include:

1. Overlapping spikes.
2. Spatial information between electrodes of a multi-channel array.
3. We verify the algorithm using computer simulations on realistic datasets. An ASIC implementation and in-vivo testing is not been included in the scope of this work.
4. This work does not handle neurons that are unseen during the training phase and become active after deployment.

2. Related works

As the importance of spike sorting has increased significantly in recent times due to its widespread applications ranging from decoding the working of human brain for neuroscientific studies to playing a part in understanding and solving neuropsychiatric disorders by providing effective close-loop feedback for neural interface systems [15,16], more efforts are being undertaken to make this process unsupervised and fully automatic in real-time. Unsupervised spike sorting algorithms have been described in [17,18] which deal with large scale, high density microelectrode arrays and can sort a large amount of single-unit events in a short time while [7] provides a fully automatic spike sorting algorithm and a corresponding software package. These algorithms provide good spike sorting accuracies, yet they make use of computationally expensive dimensionality reduction techniques like PCA and Independent Component Analysis(ICA), which reduces their application only to offline analysis of data and makes them unfeasible to be used in implantable circuits. Keshtkaran and Yang [19] also presents a method that uses discriminative subspace learning to extract most discriminative features and then uses automatic detection of number of clusters for further clustering, but this method may again not be suitable for real time, implantable circuits due to its high computational complexity. A computationally efficient, unsupervised spike sorting algorithm is presented in [20], which highlights the importance of unsupervised, real-time spike sorting.

Moving onto the detailed dissection of parts of spike sorting, the first step in this process is spike identification. Many spike detection architectures have been reported in [21], but none of them explicitly handle electrodes which fail to capture any spike trains. We present a generic spike identification system which deals with such cases and identifies the absence or presence of valid spike trains, thus pinpointing the broken channels and pausing computation in case of non-existence of spike trains, so that the signals from these channels are not processed further. This can save valuable time and will help reducing the power consumption of future on-chip systems significantly, since they no longer have to process channels emitting pure noise.

After it has been successfully asserted that valid spikes are present in the data, the next step is to separate these spikes from noise through the process of spike detection. The accuracy of spike detection significantly influences performance of succeeding steps. The simplest technique used for spike detection is absolute thresholding in which spikes are detected if recorded real-time data samples cross a certain threshold. This technique does not perform well for low SNR data [22] or data that contains multi-unit spike trains. Wavelets have also been used for unsupervised spike detection [23] and have shown good accuracies even at low SNR levels [24]. However, algorithms involving wavelets are not only computationally extensive but also depend heavily on the choice of mother wavelet [22]. Similarly, many other spike detection techniques like cepstrum of bispectrum [22], morphological filters [25] and template based detection using normalized correlation [26] also work well at low SNR but are computationally extensive and are not viable for on-chip implementation. The Non-Linear energy Operator (NEO) has also been used widely in spike detection due to its instantaneous nature and low computational resource demand [27]. Gibson et al. [21] evaluates the performance of Stationary Wavelet Transform (SWT) and NEO-operator algorithm for spike detection by using ROC curves and shows NEO to outperform SWT at both low and high SNR's by being more noise robust and simple. However, the empirical threshold calculation techniques used in spike detection systems currently deliver suboptimal detection thresholds which limit detection accuracy especially when multi-unit activity is present. In some methods, the spike detection and sorting are entwined together to be completed in a single step [28] without an explicitly independent detection module.

After the detection of spikes from the recorded data, the next step is the extraction of features from these spikes. Over the years, a lot of research has been put into the selection of features for neural spike sorting. Features such as principal components and wavelet coefficients have been extensively used for sorting but the implementation of these features requires considerable resources and is not ideal for implantable chip systems. Zero Crossing Features (ZCF) [29] have been shown to give comparable performance to features like principal components, First Second Derivative Extremas (FSDE) and its variants FSDE2, FSDE3 and FSDE4 [30], with an added advantage of considerably less computational demand and simpler hardware implementation. When dealing with multi-unit intracranial data, there is a need for features which are able to appropriately separate multi-unit spikes from single units without adding to the computational expense. Keeping these constraints in view, an additional feature called NEO-Sum has been proposed in this paper for handling multi-unit spikes. The two ZCF features combined with the additional NEO-Sum feature gives us three distinct features which can then be used for classification.

Nearest-Neighbor and K-means are computationally the simplest classifiers available but it has been shown that they do not give optimal performance especially for medium/low SNR and high noise variance. Self-Organizing Maps (SOM) have been shown to outperform classification techniques such as Mahalanobis distance, PCA, Fuzzy C-Means and Cosine similarity etc. [31] by delivering the best accuracy-complexity trade-off, along-with demonstrating a stable performance over varying SNRs. An overview of some of the algorithms and their associated hardware costs is presented in [31].

3. Simulated datasets

The first dataset consists of neural spikes recorded from live experiments in conjunction with a neural signal simulator to produce signals that mimic invasive intracranial electrode data. A detailed description of experimental procedures employed to collect neural data are given in [32]. A total of 10 different spike templates were used to mimic 10 separate channels in this data, each channel further consisting of three spike classes. Fig 4 shows the spike templates for different channels. The generated recordings did not contain any overlapping spikes and were generated for different SNR values with SNRs of 5 dB, 7 dB, 10 dB, 12 dB, 15 dB and 18 dB, where each individual SNR dataset further made up of 10 channels, in order to give a comprehensive picture of performance of algorithms.

The second dataset was obtained from the University of Leicester Neuro-Engineering Lab [9]. This dataset consists of simulated extracellular recordings which contain a high proportion of realistic multi-unit spikes (20 Hz firing rate) in addition to less frequently occurring single unit spikes (firing rate between 0.5 and 5 Hz). These simulations were generated using 594 different averaged spike shapes taken from real recordings in monkey neocortex and basal ganglia. The multi-unit spikes are generally created by a combination of neurons occurring about 100–150μm from the electrode tip and were simulated in this dataset by combining the 594 spike shapes with their amplitude uniformly distributed between 0.5 and 1.5 times the scaled value of average standard deviation of background noise. A detailed explanation of these simulations is given in [9] and the three spike classes from a sample recording from the collection are shown in Fig 3, which clearly shows two classes of less frequently occurring single unit spikes of high amplitude(red and green) and one class of multi-unit low amplitude spikes(blue).

4. Evaluation metrics

The spike detection accuracy (DA) has been calculated using the formula:

$$DA = \frac{TDS}{MS + FA + TDS} * 100 \quad (1)$$

where TDS is the number of truly detected spikes, FA is the number of false alarms and MS is the number of missed spikes.

4.1. Truly Detected Spikes (TDS)

The TDS represents the number of detected spikes present in the simulated recordings which correspond to actual spikes (as defined by the provided labels).

4.2. False Alarms (FA)

Each FA is the noise which was mistaken to be a spike.

4.3. Missed Spikes (MS)

MS represents the number of actual spikes that were not detected.

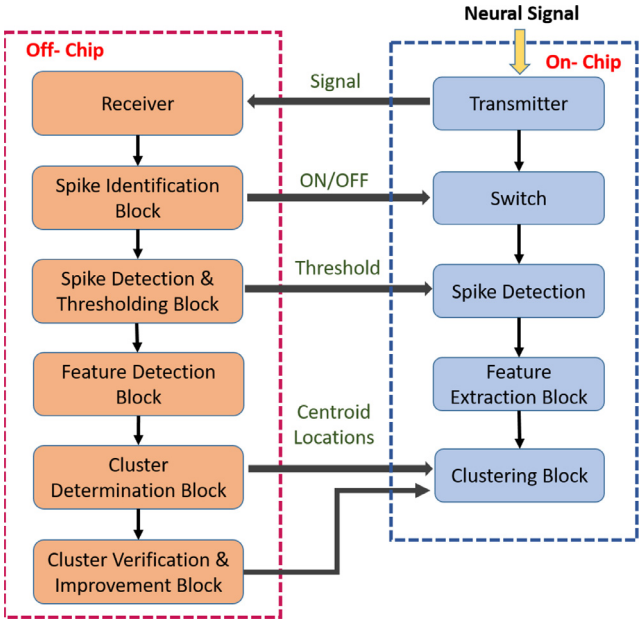


Fig. 2. Design of the system proposed in this paper, a combination of off-line and on-line processes.

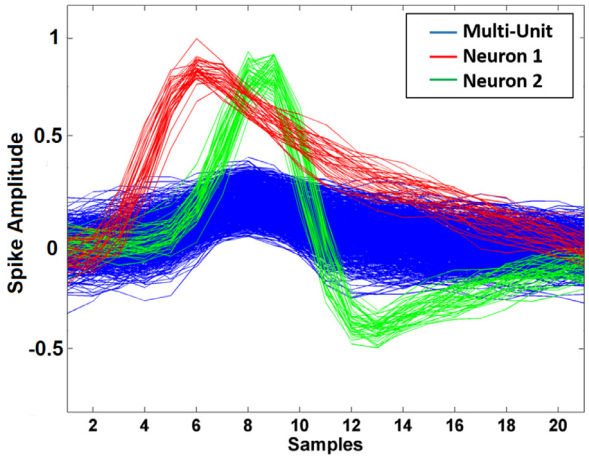


Fig. 3. Templates (with normalized spike amplitudes) and firing rates of three neurons present in a realistic simulation.

The classification accuracy (CA) has been calculated using Eq. (2) as:

$$CA = \frac{TCS}{TCS + MCS} * 100 \tag{2}$$

where TCS is the number of truly classified spikes and MCS is the number of mis-classified spikes.

5. Methodology

The proposed system is a co-design of off-line training and on-line operation. Fig. 2 shows an overview of the proposed automatic spike sorting system with clear indication of the coordination between on-chip and off-chip processes. In the proposed system, a certain amount of raw neural data will be initially transmitted off-chip for system training. The spike presence block (Section 5.1) will process the data and detect the existence of a spike train. This block will trigger operation of each channel so that processing is halted in the event of a malfunction. Subsequently for spike detection, the optimal detection threshold is determined using a gradient based technique (Section 5.2). ZCF and NEO-sum features are

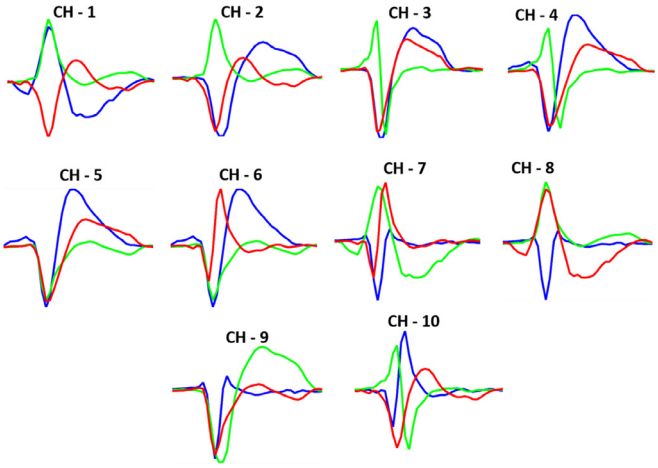


Fig. 4. Templates for various spike classes present in different channels.

then extracted from the detected spikes and these features are further utilized in classification. The number of clusters, determined using the Gap statistic (Section 5.4) are used in Self-Organizing Maps (SOM), a type of neural network which are applied on the extracted features to provide cluster centroids. A cluster verification block (Section 5.5) identifies the different clusters belonging to the same neural source. The optimal detection threshold and cluster centroids will now be transmitted onto the chip for further processing.

5.1. Valid spike presence algorithm

The presence or absence of a spike train in a data stream is a problem that has not been a focus of much research yet. Before the system starts the actual spike detection and sorting process, it is important to validate the existence of a spike train at a channel because sometimes the farther electrodes might not capture any viable neuronal activity. Moreover, the electrodes may malfunction/break or get damaged, resulting in no detection of any valid neuronal activity on that channel. In such a case, the processing for that channel is halted using the spike presence system and this system can thus also serve as an indicator to the neural scientist about potential system problems. In case of electrode drifts, where the electrodes are not damaged, but shifted during the operation, a viable signal is still available and function will continue. Adjustment of clusters for electrode drifts can be made using the Moving Cluster K-Means classifier in which the cluster centroid after each classification is updated using a moving cumulative average of the previous 32 samples [31].

The proposed spike presence algorithm is to be applied on the raw neural data initially transmitted off-chip and is based on the Non-Linear Energy Operator defined in Eq. (3). The NEO filter output is a high-SNR, non-linear mapping of the spike train and is used commonly for spike detection. The various steps of the spike identification algorithm are as follows:

1. Initially, apply the NEO filter to a 5-second window of the raw neural data x[n].

$$NEO(x[n]) = x[n]^2 - x[n-1]x[n+1] \tag{3}$$

2. Sweep the threshold from zero to the maximum NEO value obtained, keeping the step size small. Store the number of detection's in a vector $d[n]$, in which each entry corresponds to a threshold value from the threshold vector $t[n]$.
3. Compute the first and second gradient of this vector.
4. After the maximum first gradient, If the first zero crossing of the second gradient of the number of detection's (s[n])

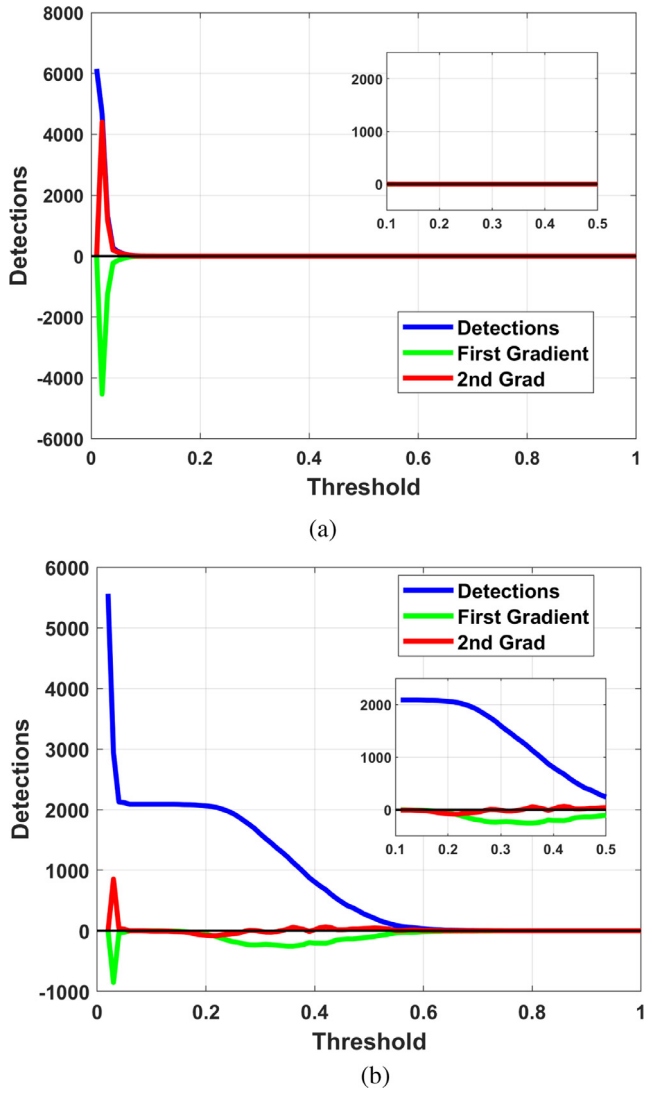


Fig. 5. (a) The number of detection's for a channel without any valid spike train. Note that once the value of second gradient touches zero, it remains zero with increasing threshold (b) Graphs for a channel with a valid spike train. Note that the second gradient does not remain consistently zero as the threshold increases.

occurs at a very small number of detection's, it indicates the absence of a valid spike train.

$$s[n] = 0 \quad \Rightarrow \quad d[n] \leq \varepsilon \tag{4}$$

where ε represents the detection of a very few small number of spikes e.g. 15, which is an insignificant number in a 5-second data. Fig. 5(a) shows a case where a valid spike train does not exist, and the second gradient zero crossing corresponds to almost zero defections. Fig. 5(b) shows a case where a valid spike train exists, and there are a significant number of detection's at the first zero crossing of the second gradient.

5. In case a valid spike train is absent, an *OFF* signal is sent on-chip while the algorithm proceeds to work on the next 5-seconds of the neural data recording.

The algorithm operates on the principle that the number of detections is high when the threshold is low. This is because noise is also detected along with the spikes. As the threshold increases, the detections start reducing rapidly. A period of slowdown and then again an increase in the rate of reduction of number of detections indicates the presence of one or more spike trains, because valid spike trains can be detected accurately across a suitable range of thresholds without any massive reduction in number of detections. Finally, for very high threshold values the number of detections drops to zero. First gradient of this curve shows the rate of change in the detections w.r.t. threshold. A deviation in the first gradient signifies the presence of a valid spike train, as it corresponds to the portion of further decrease in number of detections after the period of sustained detections, as shown in Fig. 5(b). To identify the location of this peak the second gradient is required. The number of detections at this point will be quite significantly greater than zero and will correspond to the number of valid detected spikes. On the contrary, in cases when no spike train is present, once the second gradient goes to zero, it remains zero for all higher values of threshold as seen in Fig. 5(a). Instead of keeping ε zero, we give it a very small value to cater for some ambiguities such as a small number of false alarms. In our study, empirical analysis suggests that for a 5-second data segment, a value of ε equal to 15 suffices the objective.

5.2. Spike detection

Most of the contemporary on-chip spike detection architectures employ energy operators for detection. *Discrete Energy Operators* measure the cross energy between a signal and its derivatives [33], with the general kth order discrete energy operator defined as:

$$\gamma_k(x[n]) = x[n]x[n+k-2] - x[n-1]x[n+k-1], \quad k = 1, 2, 3, \ldots \tag{5}$$

For $k = 2$, the Discrete Energy Operator transforms into the Non-Linear Energy Operator (NEO) in (3), which is the most widely used energy operator for low complexity spike detection modeled for circuits.

The NEO filter, when applied to a signal, enhances its high-energy, high-frequency parts which usually correspond to neural spikes. It is widely used for on-chip neural spike detection due to its instantaneous nature and low computational demand. Fig. 6 shows the application of the NEO filter to a spike train. It can be seen that the NEO filter enhances the spike regions owing to their high energy and frequency. A threshold is then applied to this NEO filter output to detect neural spikes (shown as a green line in Fig. 6(a)). Traditionally, this threshold is obtained by taking the scaled mean of the NEO output [33] and is given by Eq. (6) as:

$$Thr = C\frac{1}{N}\sum_{n=1}^{N} NEO[n] \tag{6}$$

The calculation of the optimal value of C is usually a huge challenge because this value varies from data to data and thus, it needs to be calculated experimentally with trial and error method [34]. Therefore, instead of using this traditional method of threshold calculation (referred to as 'normal threshold' in rest of the paper), we have proposed a new optimal threshold calculation technique that is data independent and unsupervised, without requiring any ground truth because the ground truth is usually not available for real-time systems. The optimal threshold is calculated off-line and just the value of threshold is to be sent on-chip to be further applied on the NEO-transformed signal. This algorithm works as follows:

1. Raw unfiltered data is sent off-chip for training. The training data is expected to be representative of the actual expected signal, thus it must contain spikes from all the neurons in the electrode's vicinity.
2. Apply NEO filter to this data.
3. Sweep the threshold starting from zero to max(NEO) and record the number of detected spikes for each threshold

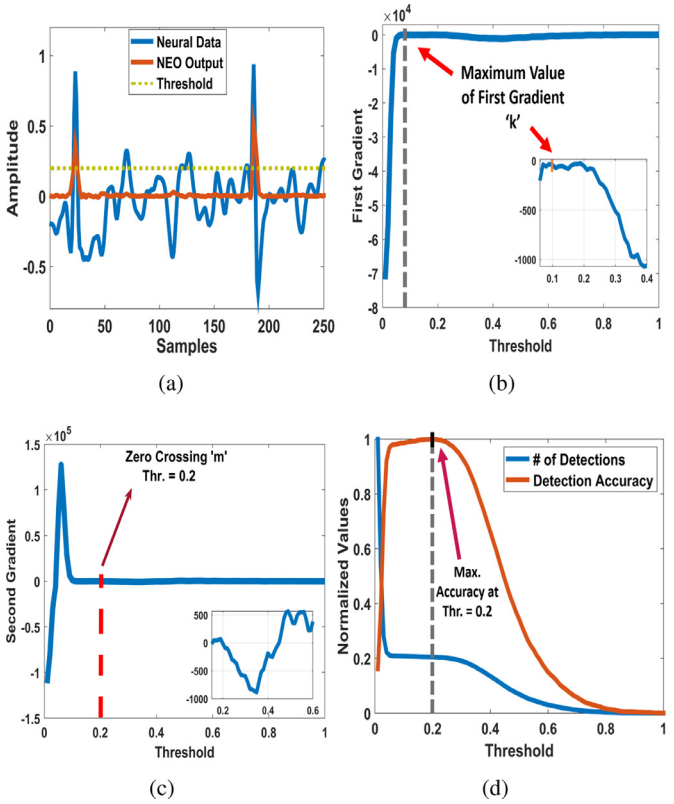


Fig. 6. (a) Application of NEO filter and consequent threshold on a neural signal (b) First Gradient of Number of Detections (c) Result after applying Second Gradient, clearly showing a zero crossing corresponding to optimal threshold (d) Normalized Comparison of number of detections and detection accuracy.

value. These number of detections are represented by $d[n]$ while $t[n]$ represents the corresponding threshold, where n is the threshold index. The first gradient of the number of detections, $f[n]$, is then computed with respect to the threshold and is shown in Fig. 6(b) as

$$f[n] = d[n+1] - d[n] \tag{7}$$

4. Locate the threshold index corresponding to the maximum first gradient, as given in Fig. 6(b). It will be

$$k = argmax(f[n]) \tag{8}$$

5. Compute $s[n]$, the second gradient of the number of detections with respect to the threshold, where

$$s[n] = f[n+1] - f[n] \tag{9}$$

6. Starting from the index of maximum first gradient k, locate the index m where the second gradient becomes zero or changes sign. The stopping condition is represented as

$$s[m] * s[m-1] \leq 0 \;\&\; m \geq k \tag{10}$$

7. The optimal threshold is the threshold corresponding to this index m, as evident from Fig. 6(c) and can be confirmed from Fig. 6(d) as well.

$$optimal\ threshold = t[m] \tag{11}$$

The optimal threshold value is now to be transmitted onto the chip for further spike detection. Fig. 6(d) shows a normalized number of detections vs threshold curve. The curves have been normalized to signify the position of the optimal threshold. It can be seen that as the threshold increases, there comes a period during which the number of detected spikes does not vary significantly with varying threshold. During this period, the signal noise has just been overcome while the threshold value is still less than the peaks of neuronal spikes and thus the optimal detection threshold lies here. This is the same point at which the value of second gradient attains a zero crossing (Fig. 6(c)).

Table 1
Detection accuracy in presence of multi-unit spikes for different k.

k	Detection accuracy (%)					
	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Mean
$k = 2$ (NEO)	41.36	54.86	54.26	54.56	36.6	48.33
$k = 3$ (ADEV)	46.88	58.36	58.68	58.69	41.55	52.88
$k = 4$ (DEAO)	51.49	63.73	63.73	63.53	48.6	58.22
$k = 5$	53.66	64.94	66.48	66.29	51.75	60.62
$k = 6$	52.89	64.33	64.69	65.63	50.83	59.67
$k = 7$	49.92	62.02	62	62.19	48.1	56.85

Table 2
Percentage of multi-unit and single-unit activity in the realistic multi-unit data-set.

Simulation	Percentage of spikes		
	Multi-unit	Single-unit 1	Single-unit 2
simulation_1	90.23	5.217	4.55
simulation_2	67.08	16.46	16.46
simulation_3	66.86	17.20	15.93
simulation_4	67.11	15.94	16.94
simulation_5	94.93	2.66	2.41

5.2.1. Multi-unit activity based detection

It has been observed that higher order energy operators show enhanced performance compared to NEO based spike detection when it comes to multi-unit activity detection. If we use a value of k greater than 2 in Eq. (5), we get higher order discrete energy operators. For $k = 3$, it gives us the Asymmetric Discrete Energy Velocity Operator (ADEV) defined in Eq. (12) as

$$ADEV(x[n]) = x[n]x[n+1] - x[n-1]x[n+2] \tag{12}$$

For $k = 4$, we get the Discrete Energy Acceleration Operator (DEAO) as defined in Eq. (13)

$$DEAO(x[n]) = x[n]x[n+2] - x[n-1]x[n+3] \tag{13}$$

As we increase the value of k further, it is observed that the detection accuracy keeps on increasing up to k=5, after which it starts decreasing as shown in Table 1. The 5th-order discrete energy operator is expressed as

$$\gamma_5[n] = x[n]x[n+3] - x[n-1]x[n+4] \tag{14}$$

It is evident from Table 1 that, for a sampling rate of 25Ksps, as the value of k is increased beyond 5, the detection accuracy starts decreasing. Thus, $k = 5$ can be used as an optimal value for pure multi-unit detection's at this sampling rate. At higher sampling rates, the energy will be distributed across a higher number of samples and we would require operators with $k > 5$ to achieve optimal performance. The main idea demonstrated in this section is that higher order energy operators are better for higher contrast between noise and multi-unit activity, the exact quantitative analysis of the transitions for sampling rates is not within the scope of this paper.

It is important to state that the realistic multi-unit data-set has a very high proportion of low SNR multi-unit spikes, as stated in Table 2. The multi-unit spikes are formed by super-position of single-unit activity and thus have a very low energy profile which makes it difficult to separate them from noise, thus resulting in a high false alarm rate. It can be seen in Table 3 that higher order operators have a better ability to differentiate between the energy profile of multi-unit activity and noise, reducing the false alarm

Table 3
Comparison of detection performance for the multi-unit data-set.

Detection percentages averaged over five realistic simulations			
Operator	Multi-unit	Single-unit	False alarms
$k = 2$ (NEO)	39.576	100	18.392
$k = 5$	56.334	100	12.96

Table 4
Comparison of detection performance for three channels of the single-unit data-set.

Operator	Detection accuracy at different SNR		
	5 dB	7 dB	10 dB
$k = 2$ (NEO)	97.83	100	100
$k = 5$	95.44	99.2	100

rate and improving detection of multi-unit spikes. The single unit activity has a much higher SNR compared to the noise and multi-unit activity thus it is relatively easy to detect all the single-unit spikes using both $k = 2$ and $k = 5$, as shown in Table 3. However, if the data-set consists solely of single unit activity or if the multi-unit activity is not required for further analysis, higher order energy operators might be detrimental to performance, as shown in Table 4. Higher order energy operators are only recommended when multi-unit activity is of central importance to the application, as in [10–14].

For rest of the paper, we will use a value of $k = 2$, which is the most common and widely used energy operator, so the results can be generalized for both single and multi-unit spikes.

5.3. Feature extraction

After a spike has been detected, it is then sent to the feature extraction block. Zero crossing features are computationally very efficient and have been shown to give comparable performance to principal components [29]. This makes ZCF ideal for implantable integrated brain circuits. The two ZCF features are computed in accordance with Eq. (15):

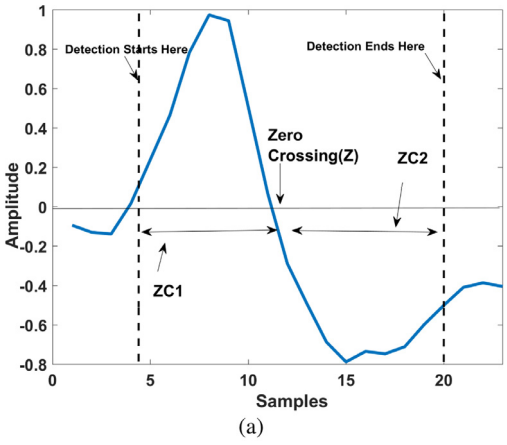
$$ZC1 = \sum_{n=0}^{Z-1} x[n] \quad \& \quad ZC2 = \sum_{n=Z}^{K-1} x[n] \tag{15}$$

where Z is the first zero crossing index and K is the length of the detected spike. The underlying assumption behind Zero Crossing Features is that every spike has a different energy profile before and after the zero crossing and this profile depends on the neuron generating that spike, as shown in the graphical explanation of ZCF in Fig. 7(a).

Moreover, multi-unit spikes usually have low SNR and high variance thus, giving rise to variant clusters that merge with single unit clusters and make classification very difficult. Therefore, in order to separate multi-unit activity while keeping in view the constraint of computational complexity of features, we have introduced a third feature called NEO-Sum. This third feature makes use of the previously computed NEO-coefficients from the detection block to separate multi-unit activity in the feature extraction step.

$$Neo - Sum = \sum_{k=0}^{N} NEO[n] \tag{16}$$

where N is the length of the detected spike and $NEO[n]$ is the output of the NEO filter applied on this spike. The NEO-Sum feature has been introduced keeping in view the lower energy-frequency profiles of multi-unit spikes as compared to single-unit spikes. Fig. 7(b) shows ZCF applied on a recording containing multi-unit



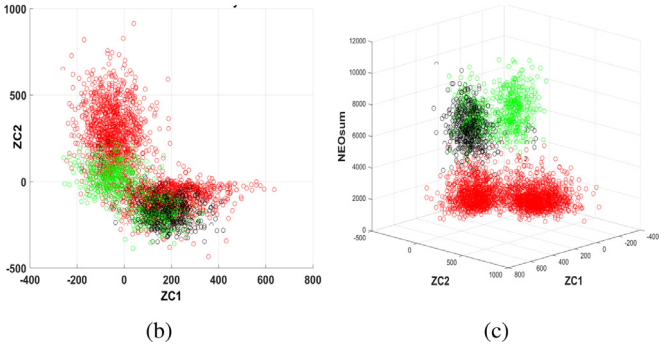


Fig. 7. (a) A graphical explanation of ZCF (b) Results after using only ZCF as features (c) Results after using a combination of ZCF and NEO-Sum as features.

activity. It can be seen that due to the variant nature of the multi-unit spikes, the multi-unit cluster (red) merges with both the single unit clusters (black and green) making it difficult to separate them. When the third NEO-Sum feature is added, the multi-unit spikes are separated from the single units owing to a difference in energy-frequency profiles as shown in Fig. 7(b).

5.4. Cluster evaluation

To accurately classify spikes according to their corresponding neuronal sources, we use a combination of K-means and Self Organizing Maps (SOM). Beforehand, the number of clusters is evaluated using a cluster evaluation criteria using one of the cluster evaluation techniques given below:

1. Calinski Harabasz
2. Davies Bouldin
3. Gap Statistic
4. Silhouette

Now the task is to select the best cluster evaluation technique keeping in view our system design. This is done by comparing the errors in the number of clusters given by each technique as shown in Table 5. It is to be noted that the 'error' described in the table is the difference between the number of clusters that the evaluation technique predicts and the actual number of clusters i.e. **3** as given in the ground truth. The results seen in Table 5 are from all the channels with data-sets of varying SNR.

From Table 5, it is preferred to choose a technique that gives a higher number of clusters than the actual number of clusters as the extra clusters can be merged accordingly using the succeeding cluster verification block. However, if the initial prediction of clusters is even slightly less than the actual, separating these merged clusters will be an impossible task later on. Whereas, even

Table 5
Comparison of various cluster evaluation techniques.

Difference b/w calculated and actual no. of clusters [Calculated Error]											
SNR	Met	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7	Ch8	Ch9	Ch10
5 dB	**GAP**	1	0	0	0	1	2	−1	1	0	0
	Cal.	0	−1	−1	0	0	0	−1	−1	−2	0
	Dav.	−1	−2	−1	−1	−1	0	−1	−1	−3	0
	Silh.	−1	−2	−1	−1	0	0	−1	−1	−2	0
7 dB	**GAP**	1	1	0	0	1	1	2	3	1	0
	Cal.	1	1	−1	0	1	1	−1	4	0	0
	Dav.	1	1	−1	−1	1	1	−1	1	−1	−1
	Silh.	1	1	−1	−1	1	1	−1	1	0	0
12 dB	**GAP**	2	2	0	0	1	1	3	3	3	0
	Cal.	2	2	0	0	1	1	1	4	1	0
	Dav.	1	1	−1	0	1	1	−1	2	−1	0
	Silh.	1	1	−1	0	1	1	−1	2	0	0
15 dB	**GAP**	2	2	0	0	2	1	3	4	4	0
	Cal.	2	2	0	0	1	1	1	4	4	0
	Dav.	1	1	−1	0	1	1	−1	2	−1	0
	Silh.	1	1	−1	1	1	1	1	2	1	0

significantly more number of clusters can be merged later on using the technique proposed in Section 5.5. Keeping this in mind, Gap was selected as the best choice for cluster evaluation, as this technique has a tendency of almost never delivering lower number of clusters then the actual. Delivering a higher number is not a problem as these clusters can be merged in post-processing in the next step. All the other techniques have a tendency to deliver a lower number than the actual number of clusters (negative error), thus making separation impossible even in post processing. Moreover, Gap gave better results as compared to all the other techniques even at higher SNRs.

The number of clusters and feature space are then input to the Self Organizing Map classifier. Self-Organizing Maps are a type of neural network that relies on competitive learning. Weight vectors corresponding to number of clusters returned by GAP are initialized randomly. When a training example is fed into the network, its euclidean distance with all the weight vectors is computed. The neuron which has a weight vector closest to the input is adjusted towards the input vector. A typical update rule is formulated as

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s)) \tag{17}$$

where s is the current iteration, t is the index of the input data vector, $D(t)$ is the input data vector, v is the index of the centroid neuron, W_v is the weight associated with the neuron v, $\alpha(s)$ is the learning rate and $\theta(u, v, s)$ is the neighbourhood function. A detailed description is provided in [35].

The SOM classifies the training data and delivers us the cluster centroids, which along with the number of clusters are transferred on-chip for further classification. These cluster centroids are then further updated online using K-means, as the classification process progresses. Given a feature vector, the online K-means algorithm assigns a spike to the nearest centroid. Furthermore, to cater to the inconvenience caused by electrode drift, we can use a computationally efficient Moving Cluster K-Means (MCK) classifier [31], in which the cluster centroid after each classification is updated using a cumulative moving average of the previous 32 samples, thus, enabling the classifier to adapt to non-stationary clusters. Suppose at iteration stamp $i+1$, the feature data vector $D_t(i+1)$ from a spike is assigned to the cluster v with centroid $\mu_v(i)$. The MCK moving average update, based on the previous 32 samples will be:

$$\mu_v(s+1) = \frac{D_t(i+1) + 31 \cdot \mu_v(i)}{32} \tag{18}$$

Our system also includes a cluster verification block so that if the number of clusters delivered by the evaluation technique and classified by SOM is more than the actual number of spike sources, it will merge clusters to deliver optimal results, as elaborated in the next section.

5.5. Cluster verification

The ZCF extraction block detects the first zero crossing after a spike has been detected and uses it for feature extraction. Some neural spikes found in actual recordings have an uncanny shape and they show an initial rise and fall just before the major spike shape occurs. The initial rise is relatively small and sometimes the detection threshold detects these as a part of the spike while at other times these are not detected in the spike signal. This means that on some occasions, the spike is detected four to five samples earlier because of these rise and falls. If these abnormalities are significant enough to be occasionally considered as a part of the spike, the existence of a zero-crossing in-between these initial rise and falls is detected by the ZCF block, resulting in a variation in the values of Zero-Crossing features for the same cluster. The consequence of this is the splitting of the cluster belonging to a specific spike shape into sub clusters. This greatly influences the process of classification and causes a significant drop in classification accuracy.

We have introduced a correlation based cluster verification block to cater for this problem. The cluster verification algorithm is stated below:

1. The feature space is clustered using Self-Organizing Maps (SOM) based on the number of clusters delivered by the cluster evaluation block.
2. Spike templates are extracted for each different class cluster.
3. If the normalized linear correlation between any spike templates is greater than a pre-selected threshold, the clusters are assumed to belong to the same neural source.

An explanation of this cluster verification can be derived from Fig. 8. Only the two ZCF dimensions have been shown in these plots for better illustration. Fig. 8(a) shows the ground truth labeling of the feature space. The two distinct yellow clusters belong to the same spike class but have split up as per the explanation provided. When Gap was applied to this feature space, it provided an optimal cluster number of **four** which was then used by the SOM Classifier. Fig. 8(b) shows the result of SOM classifier applied on the feature space. It can be seen that the two distinct yellow clusters seen in Fig. 8(a) have been classified into separate classes in Fig. 8(b). Fig. 8(c) shows the extracted spike templates from the

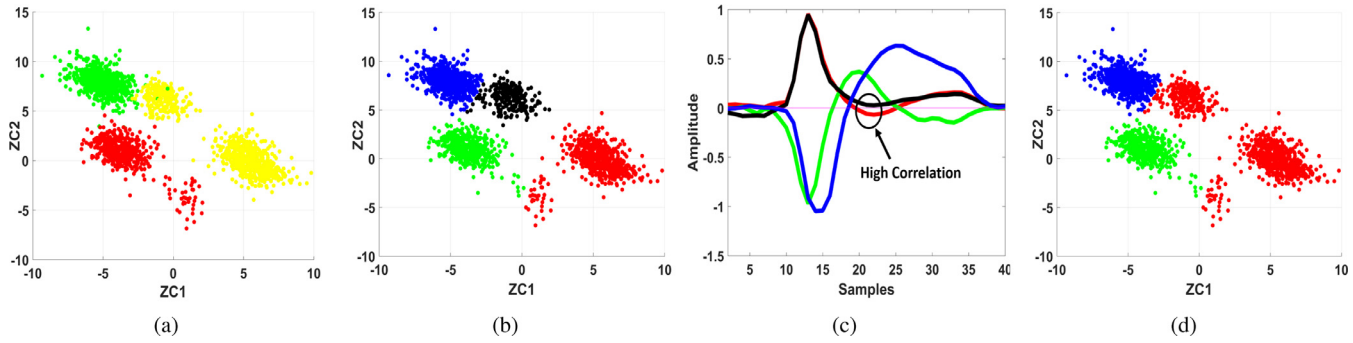


Fig. 8. (a) Ground truth labeling of feature space (b) Result of initial SOM classification on feature space c) Extracted cluster templates signifying high correlation between some clusters d) Post cluster verification and cluster merger results.

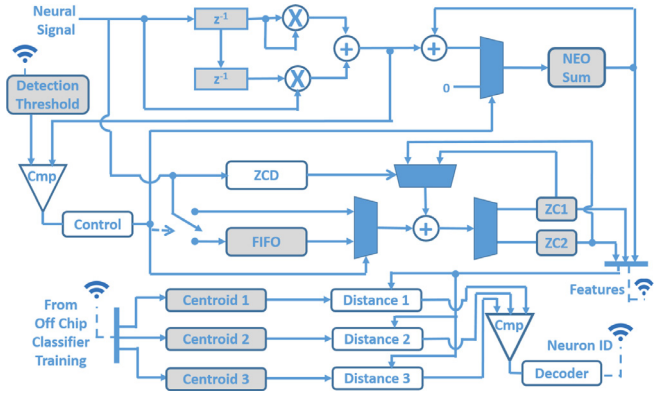


Fig. 9. Block diagram representation of online process architecture (where the white blocks represent computations or control units and gray blocks represent registers).

clusters shown in Fig. 8(b). It is clearly visible that the red and black clusters are highly correlated. The normalized linear correlation between these two clusters is **0.9622** which is above our pre-selected threshold **0.9**. Thus, the system concludes that these two clusters belong to the same spike class and can be merged together. Fig. 8(d) shows this merger after cluster verification, verifying that the end results approximately match the ground truth result in Fig. 8(a). The more elaborate correlation plots are not shown in Fig. 8 for brevity, as the overall method can be easily explained without them as well.

A conceptual block diagram depicting the operational flow of the proposed online process is shown in Fig. 9. The detection threshold, number of clusters and cluster centroids are to determined by initial offline training and then transferred onto the online processor. The detection, feature extraction and classification is to be thereafter performed online. For simplicity, the classification block is shown for only three spike classes but a number of Centroid registers can be used and activated depending on the number of clusters determined. This diagram should serve as an indicator of the simplicity and computational efficiency of the proposed method. Although the scope of this paper is design, analysis and verification of the algorithm using computer simulations, Fig. 9 may aid in future practical implementation on platforms such as FPGA or ASIC.

6. Experimental setup

Our system was designed to handle both single unit and multi-unit activity. Keeping this in view, we tested our work on two different simulated datasets: one consisting of only single unit activity and the other consisting of a very large proportion of realistic multi-unit spikes.

In the spike detection experiment, we demonstrate that our proposed automatic threshold improves detection performance compared to the commonly used threshold. In the spike classification experiment, we demonstrate the effectiveness of our proposed feature in separation of multi-unit activity and show that our cluster verification technique indeed performs as expected, mitigating the cluster splitting issue presented by ZCF features. The classification results represent the simulation for the online classifier after the required initial offline training. The system is fully automatic, with considerably low computational demand.

7. Results and discussions

7.1. Spike detection

As explained earlier, the Non-Linear Energy Operator is widely used for on-chip neural spike detection due to its instantaneous nature and low computational demand. For determining the threshold by standard normal threshold functions, the ideal value of the constant C needs to be calculated experimentally using ground truth and varies significantly from data to data. On the other hand, the optimal threshold algorithm we propose is automatic and does not require ground truth.

Various methods [1] use the value of C=4 for hardware based spike detection systems. Thus, just for comparison we will compare the proposed automatic threshold technique with the normal threshold function with the constant factor being C=4 for both datasets.

For dataset 1, Fig. 10(a) shows the average detection accuracy results for all channels ranging over multiple SNRs. It can be seen that for medium and higher SNRs, the performance of both methods is good and acceptable, but as the SNR decreases, the optimal threshold technique outperforms the constant threshold by a considerable margin. Table 6 summaries the average detection accuracy over all SNRs.

Moreover, the two threshold techniques were also applied to the realistic dataset dominated by multi-unit activity. The normal scaling factor C=4 had been calculated empirically for specific single-unit activity which generally has a much higher energy profile, most of the multi-unit spikes fall below the threshold, thus it shows poor performance. It strengthens our proposition that the same value of C cannot be used for different recordings, resulting in the need of a generalized automatic threshold technique that can adapt according to the signal. Due to the presence of a large number of multi-unit low SNR spikes in the data, the detections are more sensitive and susceptible to the threshold and even small changes in threshold can cause drastic changes in accuracy. The difference between the normal and automatic threshold accuracies

Table 6
Mean accuracies for spike detection and classification.

Results	Methods	Mean accuracy (%)	
		Single unit datasets	Single unit & embedded multi-unit activity datasets
Spike Detection	Normal Threshold	77.46	9.42
	Automatic Threshold	98.43	43.63
Spike Classification	Simple ZCF as features	94.9	40.41
	ZCF & NEO_Sum as features	94.908	92.62

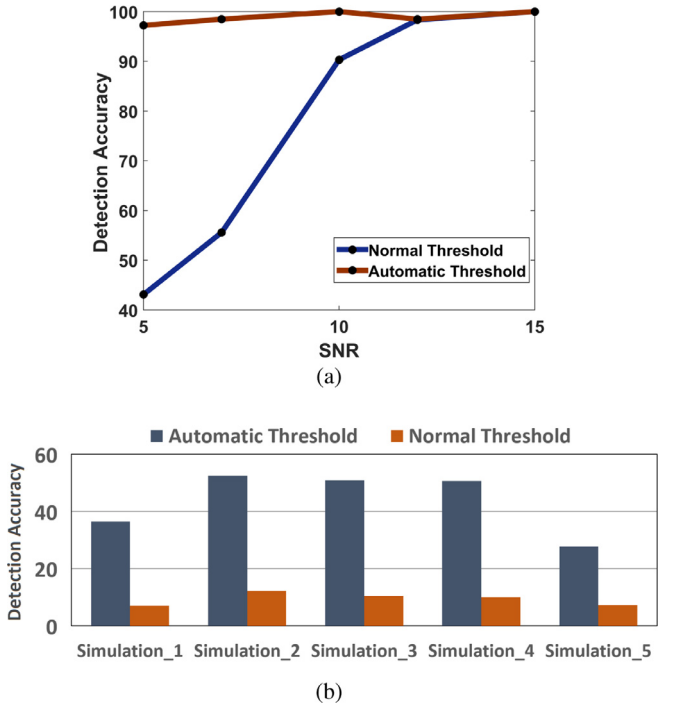


Fig. 10. Comparison of detection accuracies between normal and automatic threshold for (a) Single-Unit High SNR dataset (b) Multi-unit low SNR dataset.

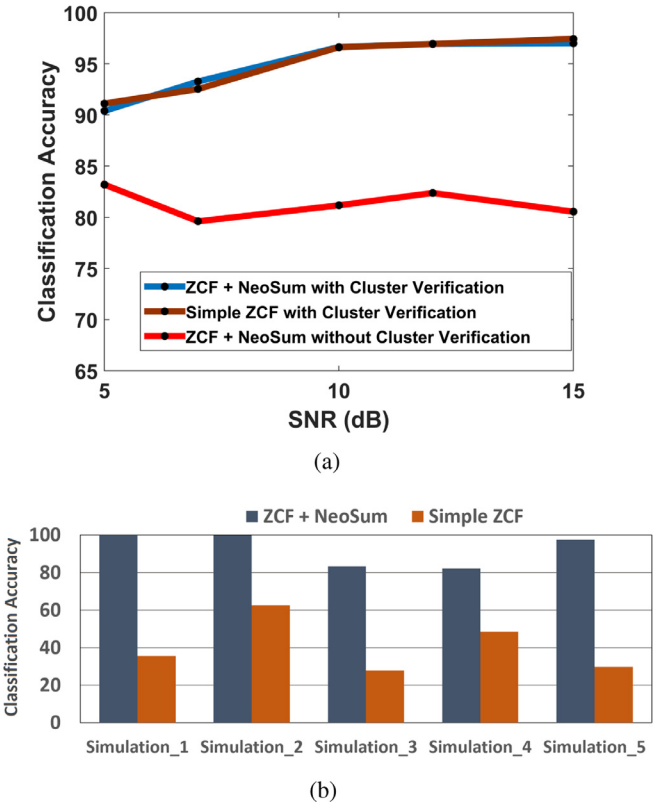


Fig. 11. Comparison of various classification techniques when applied on (a) various SNRs of a single-unit dataset (b) a multi-unit realistic dataset.

can be seen impeccably in Fig. 10(b) and Table 6. It can be observed that there was a considerable improvement in the detection performance as compared to normal threshold but the overall detection accuracy was still lower as compared to recordings consisting solely of high SNR single units (Fig. 10(a)). The reason for this is that low SNR multi-unit spikes are very difficult to detect because of their variant nature and low energy profile.

Most applications are usually more focused on the detection of single-unit spikes only so multi-unit spikes can be discarded in these cases. But if the system is to be used for applications that specifically require the detection of multi-unit activity, it is recommended to use a detection architecture based on the 5th-order discrete energy operator as explained in Section 5.2.1 as it gives multi-unit detection results better than NEO operator as evident in Table 1 as well. The efficacy of our proposed features in separation of multi-unit activity from single unit activity is explained in the next section.

7.2. Spike classification

Our proposed classification technique consists of an initial off-line training using Self-Organizing Maps and cluster verification block and on-line transmission of cluster centroids and further classification based on those centroids using k-means.

To study the classification results accurately, we assumed that all spikes had already been detected and thus used the ground truth spike time data. Fig. 11(a) demonstrates the efficacy of our classification technique for various SNRs ranging from 5 dB to 12 dB. It can be seen that ZCF+NEO-Sum classification with cluster verification and simple ZCF with cluster verification give much better results than simple ZCF without a cluster verification block. This is due to the fact that ZCF in some cases splits clusters to give multiple clusters of a single spike class. The results also show that cluster splitting reduces with a decrease in SNR as there is a rise in classification accuracy at the low SNR of 5dB.

Although the classification of individual single unit activity does not benefit much with the incorporation of the additional NEO-Sum feature, it greatly improves the separation of multi-unit and single unit activity as shown in Fig. 11(b) and Table 6. The additional feature takes advantage of the fact that multi-unit spikes

Table 7
Complexity comparison of spike detection operators.

Algorithms	N_{add}	N_{mult}	Comp. complexity	Detection accuracy (%)
Untransformed Input Signal & Absolute Threshold	2	-	2	82
NEO Transformed Input Signal & Automatic Threshold	2	2	22	97.2

have a lower energy profile or content than single unit spikes. Although other features may also be available to separate multi-unit and single unit activity, but NEO-Sum is computationally efficient owing to the fact that NEO coefficients are already available to us in the detection phase and we only have to compute a simple sum to extract this feature.

7.3. Complexity comparison

In this paper, the proposed combination of off-line & on-line processes result in the implementation of only the NEO operator, Zero-Crossing and Neo-Sum features and K-means on the chip while the threshold calculation, cluster evaluation, verification and initial SOM classification happen off-chip, thus, greatly reducing the on-line complexity. The blocks that are proposed to be a part of the on-line process consist of simple processes which should not be an issue when transferring the proposed system into a hardware prototype.

As in [21], we define the computational complexity as:

$$Complexity = N_{additions} + 10 * N_{multiplications} \quad (19)$$

7.3.1. Spike detection

Absolute threshold applied directly on real-time signal and thresholding the NEO-transformed signal are the most commonly used low complexity methods for spike detection. Table 7 shows an accuracy-complexity comparison of these two methods (including complexity required for comparing threshold on-chip) without taking into account the requirements for threshold calculation as they are computed off-chip. The number of additions (N_{add}) and multiplications (N_{mult}) in Table 7 are for a single sample of spike. The detection results are for signals with a comparatively realistic low SNR of 5dB.

It is evident from Table 7 that NEO with automatic threshold provides a very high accuracy of 97% as compared to Absolute Threshold at a small expense of complexity.

Many detection algorithms also employ Stationary Wavelet Transform (SWT) for spike detection. However, the computational complexity of SWT goes up to approximately 450 [21], which is very high as compared to that of NEO. Thus, NEO becomes an appropriate choice since it gives a high accuracy and fairly low complexity.

7.3.2. Feature extraction and classification

This paper employs ZCF with Neo-Sum and K-means as the on-line feature extraction and classification unit respectively. Most of the contemporary unsupervised implantable architectures use O-Sort with Euclidean Distance or *l1*-norm distance as the main classifier because of its capability to cluster features without any supervision. But, the main drawback of using O-Sort is that its classification performance rarely exceeds 70% [36], thus reducing its effectiveness in practical on-chip spike sorting applications. On the other hand, K-means gives a classification performance exceeding 90% but it requires a prior knowledge of number of clusters. Our proposed off-chip & on-chip combination can provide just that, with the off-chip system working unsupervised to determine the number of clusters along with the cluster centroids and then sending these to the chip for enhanced performance of the complete system. Thus, the whole architecture not only works unsupervised, but effectively uses off-line training and on-chip testing for a better performance. Moreover, [36] also provides an elaborate accuracy-complexity comparison of different spike sorting architectures and its results show that ZCF with K-means is amongst the top methods with a low computational complexity and very high accuracy. As shown in Table. 8, Our proposed Zero-Crossing and Neo-Sum features require only 2^*M^*N additions and no multiplications for their computation (where M = number of spikes, N = samples per spike) which are low as compared to PCA, Discrete Wavelet Transforms or Discrete Derivatives, making them an optimal choice for our system to deal with a wide variety of spikes including multi-unit activity.

Table 8
Complexity comparison of feature extraction algorithms.

Algorithm	N_{add}	N_{mult}	Computational complexity
ZCF	30	0	30
PCA2	60	60	660
DWT	58	60	658

The separation of multi-unit activity from single unit activity is very important in a lot of applications and their non-separation can also be a cause of many problems as explained in [2]. Our proposed method is efficient with regards to both accuracy and computation while dealing with the classification of multi-unit spikes as well by employing a combination of ZCF and Neo-Sum features. We have come across no other implantable spike sorting architecture that specifically deals with this multi-unit activity separation until now.

8. Conclusions

With the rise in the number of channels in neural interfaces and the increase in sampling rates, it is imperative to develop new computationally efficient spike sorting methods to reduce the transmission bandwidth and power consumption. Furthermore, the systems need to be completely unsupervised in order to remove human supervision, so that they can be applied to practical problems in real-time.

We have proposed, and through the use of computer simulations, verified a new spike sorting system that is computationally efficient and completely unsupervised, modeled for future hardware implementation. Furthermore, it can also handle multi-unit activity which is usually a part of actual recordings and can provide a unique window of opportunity to understanding the human brain, if handled appropriately. No other reported architecture we came across specifically handles multi-unit activity. We also present a cluster verification method to mitigate some issues with Zero Crossing features.

While there may be other methods which have a better quantitative/accuracy performance than our proposed system, we have effectively addressed issues such as computational complexity, removal of the need for human supervision and effective handling of multi-unit activity. These issues are of prime importance to the spike sorting problem and practical implementation of ASIC systems that may build upon these techniques.

Conflicts of interest

The authors have conflicts of interest with the following institutions:

- National University of Sciences and Technology, Islamabad, Pakistan. [nust.edu.pk].
- University of Jeddah, Jeddah, Saudi Arabia. [uj.edu.sa].
- Korea Advanced Institute of Science and Technology. [kaist.ac.kr].

Acknowledgments

This work has been supported by Ministry of Science and Technology, Pakistan and is undertaken at the NeuroInformatics Lab, School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad.

References

[1] S.M.A. Zeinolabedin, A.T. Do, K.S. Yeo, T.T.-H. Kim, Design of a hybrid neural spike detection algorithm for implantable integrated brain circuits, in: 2015 IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 794–797.

[2] T. Tariq, M.H. Satti, M. Saeed, A.M. Kamboh, Low SNR neural spike detection using scaled energy operators for implantable brain circuits, in: 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2017, pp. 1074–1077.

[3] H.G. Rey, C. Pedreira, R.Q. Quiroga, Past, present and future of spike sorting techniques, Brain Res. Bull. 119 (2015) 106–117.

[4] A. Eftekhar, E.P. Sivylla, G.C. Timothy, Towards a next generation neural interface: optimizing power, bandwidth and data quality, in: 2010 Biomedical Circuits and Systems Conference (BioCAS), 2010, doi:10.1109/biocas.2010.5709586.

[5] U. Rutishauser, E.M. Schuman, A.N. Mamelak, Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo, J. Neurosci. Methods 154 (1–2) (2006) 204–224, doi:10.1016/j.jneumeth.2005.12.033.

[6] C. Rossant, S.N. Kadir, D.F.M. Goodman, J. Schulman, M.L.D. Hunter, A.B. Saleem, A. Grosmark, M. Belluscio, G.H. Denfield, A.S. Ecker, et al., Spike sorting for large, dense electrode arrays, Nature Neurosci. 19 (4) (2016) 634–641, doi:10.1038/nn.4268.

[7] J.E. Chung, J.F. Magland, A.H. Barnett, V.M. Tolosa, A.C. Tooker, K.Y. Lee, K.G. Shah, S.H. Felix, L.M. Frank, L.F. Greengard, A fully automated approach to spike sorting, Neuron 95 (6) (2017) 1381–1394.

[8] F. Wood, M. Black, C. Vargas-Irwin, M. Fellows, J. Donoghue, On the variability of manual spike sorting, IEEE Trans. Biomed. Eng. 51 (6) (2004) 912–918, doi:10.1109/tbme.2004.826677.

[9] J. Martinez, C. Pedreira, M.J. Ison, R. Quian Quiroga, Realistic simulation of extracellular recordings, J. Neurosci. Methods 184 (2) (2009) 285–293.

[10] E.M. Merricks, E.H. Smith, G.M. McKhann, R.R. Goodman, L.M. Bateman, R.G. Emerson, C.A. Schevon, A.J. Trevelyan, Single unit action potentials in humans and the effect of seizure activity, Brain 138 (10) (2015) 2891–2906.

[11] Y.-S. Choi, M.A. Koenig, X. Jia, N.V. Thakor, Quantifying time-varying multiunit neural activity using entropy-based measures, IEEE Trans. Biomed. Eng. 57 (11) (2010) 2771–2777.

[12] R. Land, G. Engler, A. Kral, A.K. Engel, Response properties of local field potentials and multiunit activity in the mouse visual cortex, Neuroscience 254 (2013) 141–151.

[13] M. Mattia, S. Ferraina, P. Del Giudice, Dissociated multi-unit activity and local field potentials: a theory inspired analysis of a motor decision task, NeuroImage 52 (3) (2010) 812–823.

[14] S. Han, J.-U. Chu, H. Kim, J.W. Park, I. Youn, Multiunit activity-Based real-Time limb-State estimation from dorsal root ganglion recordings, Scient. Rep. 7 (December 2016) (2017) 44197.

[15] L. Hamilton, M. Mcconley, K. Angermueller, D. Goldberg, M. Corba, L. Kim, J. Moran, P.D. Parks, S. Chin, A.S. Widge, et al., Neural signal processing and closed-loop control algorithm design for an implanted neural recording and stimulation system, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015.

[16] J. Park, G. Kim, S.-D. Jung, A 128-channel FPGA based real-time spike-sorting bidirectional closed-loop neural interface system, IEEE Trans. Neural Syst. Rehabil. Eng. (2017).

[17] G. Hilgen, M. Sorbaro, S. Pirmoradian, J.-O. Muthmann, I. Kepiro, S. Ullo, C.J. Ramirez, A.P. Encinas, A. Maccione, L. Berdondini, et al., Unsupervised spike sorting for large scale, high density multielectrode arrays, Cell Rep. 18 (10) (2017) 2521–2532.

[18] C. Leibig, T. Wachtler, G. Zeck, Unsupervised neural spike sorting for high-density microelectrode arrays with convolutive independent component analysis, J. Neurosci. Methods 271 (2016) 1–13.

[19] M.R. Keshtkaran, Z. Yang, Noise-robust unsupervised spike sorting based on discriminative subspace learning with outlier handling, J. Neural Eng. 14 (3) (2017), doi:10.1088/1741-2552/aa6089.

[20] S.E. Paraskevopoulou, D. Wu, A. Eftekhar, T.G. Constandinou, Hierarchical adaptive means (ham) clustering for hardware-efficient, unsupervised and real-time spike sorting, J. Neurosci. Methods 235 (2014) 145–156, doi:10.1016/j.jneumeth.2014.07.004.

[21] S. Gibson, J.W. Judy, D. Markovic, Comparison of spike-sorting algorithms for future hardware implementation, in: 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, pp. 5015–5020.

[22] S. Shahid, L.S. Smith, A novel technique for spike detection in extracellular neurophysiological recordings using cepstrum of bispectrum, Eur. Signal Process. Conf. 96 (2008). 33–5

[23] N. Zoran, J. Burdick, Spike detection using the continuous wavelet transform, IEEE Trans. Biomed. Eng. 52 (1) (2005) 74–87, doi:10.1109/tbme.2004.839800.

[24] V. Shalchyan, W. Jensen, D. Farina, Spike detection and clustering with unsupervised wavelet optimization in extracellular neural recordings, IEEE Trans. Biomed. Eng. 59 (9) (2012) 2576–2585.

[25] X. Liu, X. Yang, N. Zheng, Automatic extracellular spike detection with piecewise optimal morphological filter, Neurocomputing 79 (2012) 132–139.

[26] W.J. Hwang, S.H. Wang, Y.T. Hsu, Spike detection based on normalized correlation with automatic template generation, Sensors (Switzerland) 14 (6) (2014) 11049–11069.

[27] S. Mukhopadhyay, G.C. Ray, A new interpretation of nonlinear energy operator and its efficacy in spike detection., IEEE Trans. Bio-Med. Eng. 45 (2) (1998). 180–7

[28] K.G. Oweiss, A. Mason, Y. Suhail, A.M. Kamboh, K.E. Thomson, A scalable wavelet transform VLSI architecture for real-Time signal processing in high-Density intra-Cortical implants, IEEE Trans. Circuits Syst. I 54 (6) (2007) 1266–1278, doi:10.1109/tcsi.2007.897726.

[29] A.M. Kamboh, A.J. Mason, Computationally efficient neural feature extraction for spike sorting in implantable high-Density recording systems, IEEE Trans. Neural Syst. Rehabil. Eng. 21 (1) (2013) 1–9.

[30] S.E. Paraskevopoulou, D.Y. Barsakcioglu, M.R. Saberi, A. Eftekhar, T.G. Constandinou, Feature extraction using first and second derivative extrema (fsde) for real-time and hardware-efficient spike sorting, J. Neurosci. Methods 215 (1) (2013) 29–37, doi:10.1016/j.jneumeth.2013.01.012.

[31] M. Saeed, A.A. Khan, A.M. Kamboh, Comparison of classifier architectures for online neural spike sorting, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (4) (2017) 334–344.

[32] M. Aghagolzadeh, K. Oweiss, Compressed and distributed sensing of neuronal activity for real time spike train decoding, IEEE Trans. Neural Syst. Rehabil. Eng. 17 (2) (2009) 116–127.

[33] P. Maragos, A. Potamianos, Higher order differential energy operators, IEEE Signal Proc. Lett. 2 (8) (1995) 152–154.

[34] M.H. Zarifia, N.K. Ghalehjogh, M. Baradaran-Nia, A new evolutionary approach for neural spike detection based on genetic algorithm, Expert Syst. Appl. 42 (1) (2015) 462–467.

[35] T. Kohonen, The self-organizing map, Neurocomputing 21 (1990) 1–6.

[36] M. Zamani, A. Demosthenous, Feature extraction using extrema sampling of discrete derivatives for spike sorting in implantable upper-limb neural prostheses, IEEE Trans. Neural Syst. Rehabil. Eng. 22 (4) (2014) 716–726.