A 32-Channel MCU-based Feature Extraction and Classification for Scalable On-node Spike Sorting

Deren Y. Barsakcioglu* and Timothy G. Constandinou*†
*Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK
†Centre for Bio-Inspired Technology, Institute of Biomedical Engineering, Imperial College London, SW7 2AZ, UK
Email: {deren.barsakcioglu10, t.constandinou}@imperial.ac.uk

***Abstract*—This paper describes a new hardware-efficient method and implementation for neural spike sorting based on selection of a channel-specific near-optimal subset of features given a larger predefined set. For each channel, real-time classification is achieved using a simple decision matrix that considers the features that provide the highest separability determined through off-line training. A 32-channel system for on-line feature extraction and classification has been implemented in an ARM Cortex-M0+ processor. Measured results of the hardware platform consumes 268μW per channel during spike sorting (includes detection). The proposed method provides at least x10 reduction in computational requirements compared to literature, while achieving an average classification error of less than 10% across wide range of datasets and noise levels.**

I. Introduction

Hidden among the interactions of billions of neurons in the brain are the cures for neurological diseases and disability. Over the past decades, many neural recording systems have been proposed in order to extract this wealth of information for advancing neuroscience research as well as developing better assistive devices and neural prosthetics [1]–[3]. Improvements in microfabrication techniques, as well as in microelectronics, present a unique opportunity to monitor large groups of neurons at an individual level. It has been projected that thousands of channels will be monitored in future neural recording systems [4]. However, biological limits on allowable thermal dissipation (to prevent tissue damage) imposes strict constraints on the wireless transmission (required to eliminate percutaneous wires to minimise risk of infections) of such large amounts of data, hence dimensionality reduction is required. On-node spike sorting prior to wireless transmission is a way of overcoming this power and bandwidth bottleneck.

Traditionally done off-line, spike sorting is a source separation task during which spiking neurons are identified based on their distinguishing characteristics (e.g. features). The typical processing flow comprises spike detection, feature extraction and spike classification. Since on-node spike sorting trades-off processing power to reduce wireless power, feasibility of such approach requires developing computationally efficient and accurate spike sorting methods. Hence, several feature extraction methods have been proposed in literature within this context. Among these are discrete derivatives (DD) [5], zero-crossing features (ZCF) [6] and first and second derivative extrema (FSDE) [7], and feature de-noising filter together with optimal features [8].

In this paper, a new method for online spike sorting is proposed and demonstrated in a low power embedded platform. The remainder of this paper is organized as follows: Section II provides a high level description of the proposed method, while Section III details algorithm design and hardware implementation. In Section IV the spike sorting accuracy and computational efficiency of the proposed method is verified through Matlab simulations and circuit measurements, followed by conclusions in Section V. From here on, the proposed method is referred as waveform and derivative features (WDF).

II. Proposed Method for Spike Sorting

The high level description of WDF is presented in Fig. 1. It consists of two parts: calibration and real-time classification. During calibration, neural data is streamed out to an external device where initial clustering of the data establishes available neurons (N) and their associated waveforms (i.e. templates). The training algorithm then finds the positions of local extrema for each template. These positions of extrema are considered as possible regions of maximum separation between neurons, denoted as S (see Section III). Training algorithm then searches for a subset of features (denoted by P) within S that maximises separation of all N. For example, $P_{1,2}$ is the sample index of the feature that maximally separates neuron $\#1$ and neuron $\#2$. This approach ensures channel-specific selection of near-optimal features. For P, a corresponding comparison threshold vector, T, is also computed.

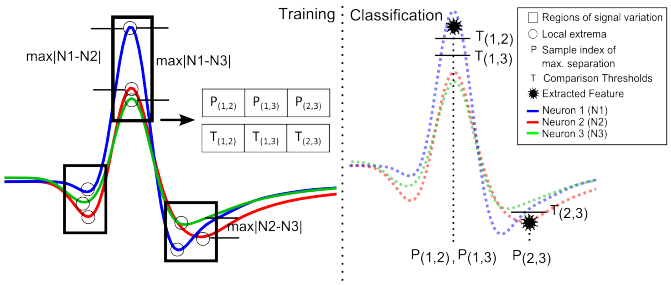

Fig. 1. A high level description of the proposed method for spike sorting.

Training algorithm then sends P and T back on-chip. During real-time classification, features are extracted using P, compared to corresponding T, and a vote is cast for either of the relevant neurons. For example, considering $P_{1,2}$ (Fig. 1), feature extracted at this position differentiates neurons $\#1$ and $\#2$. Since feature at $P_{1,2}$ is above $T_{1,2}$, it is likely to belong neuron $\#1$. The same is done for all P and the detected spike is assigned to neuron with maximum votes. It should be noted that derivative space is also included in WDF. The same procedure above is also applied to second derivative waveforms.

III. Algorithm Design and Hardware Implementation

A. Feature Selection

The selected feature space is based on the waveform dynamics of EAPs relative to the electrode position. It has been shown that EAP can be modelled as a dipole charge, and its amplitude scales with distance from the source [9]. In fact, the paradigm where identical neurons laterally aligned would have the same waveform expansion time, however, would scale differently in amplitude has been the motivation for previous derivative based feature extraction studies such as FSDE [7].

Besides amplitude scaling, the electrode position relative to different sections of neuronal structure also affects the expression of other critical features — initial positive capacitive peak and re-polarisation phase — of the EAP due to variations in current components across the neuronal structure [10]. For example, the amplitude of the initial capacitive peak can range along apical trunk from none (closer to soma) to almost as same as the main peak (closer to more distant dendrites) [11]. Furthermore, it has been shown that differences in conduction densities is another major source of variability observed in the main peak and re-polarisation phase. These result in variations in the width and amplitude of the main peak and re-polarisation phase, as well as variations in inflection points associated between their transition [11].

Based on the underlying analysis of waveform dynamics, one can hypothesize that the main features affected are at the local extrema and the inflection points of the EAP and its derivatives. In other words, characterisation — thus separation — of signals may be done using EAP amplitudes and rates of change at such local extrema and inflection points. In fact, an empirical study of the recorded EAPs reveal that the majority of separation between EAPs are in fact at these local extrema. Fig. 2 presents the amplitude and temporal variation profiles of local extrema of EAP and their derivatives up to third order. Based on waveform dynamics and empirical evidence, the features with highest variability are considered as potential candidates for proposed algorithm (features 1,2 and 3 for EAP waveform; 2 and 3 for first derivative; 2,3, and 4 for both second and third derivatives).

Among the determined feature space above, combining EAP waveform and second derivative features are found to provide the best trade-off in terms of accuracy and computational complexity across various datasets at varying levels of background and added white Gaussian noise (AWGN) noise (see Section IV for datasets). This result is attributed to derivatives' superior performance at high background activity, while waveform features are more tolerant to increases in AWGN. In fact, the rate of sorting accuracy loss of derivatives for AWGN is observed to be are 0.44%, 0.60% and 0.72% per dB (increasing order of derivatives), while this is 0.25% per dB for EAP waveform features. On the other hand, it can be observed that higher derivative orders are better at rejecting background activity. Although from the perspective of computational requirements first derivative is preferable, the average classification accuracy of second and third derivatives are better than the first by 4.6% and 3.6% respectively across all datasets and noise levels (7.57% and 6.20% for "Difficult1" and "Difficult2" only). Hence, second derivatives provide a significant accuracy improvement with minimal increase in computational complexity (extra addition and shift operation per feature) over first derivatives. The reason why third derivatives are 1% worse than second is attributed to the fact that differentiation at the presence of AWGN reduces overall SNR. Therefore, features 1,2,3 for EAP waveform and 2,3,4 for second derivatives is the chosen feature space for training algorithm to search (denoted by S previously).

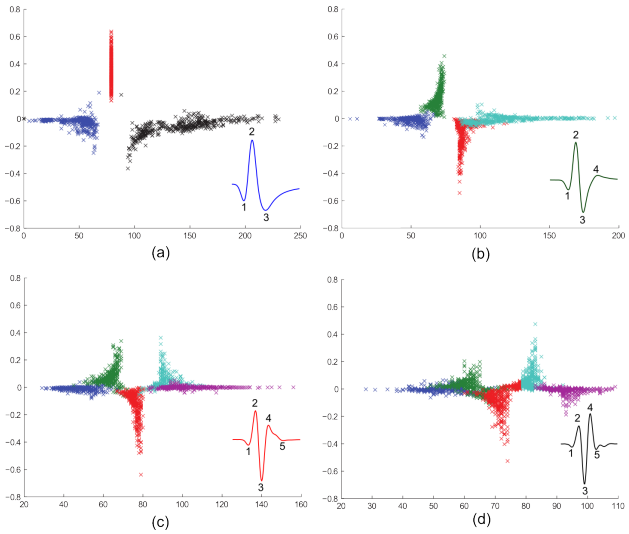

Fig. 2. Variations in (a) EAP waveforms (b) First Derivative (c) Second Derivative (d) Third Derivative. X-axis samples, Y-axis normalised amplitudes. In addition to distributions, given are they typical waveforms and numbered local extrema associated with each. Note that since EAPs are aligned with respect to their main peak, there is no temporal spread at peak position.

B. Off-chip Calibration

As described previously, the initial training clusters the EAPs and creates their template waveforms. Then, the positions of the local extrema of the waveform and derivative space are found for each neuron template. Among these positions of extrema, the training algorithm searches for sample positions $P_{i,j}$ for each neuron — where $i, j \in N$, $i \neq j$, and N is the number of neurons — such that features at $P_{i,j}$ has the largest L1-distance (hence separation).

For example, considering N=3, three feature positions $(P_{1,2}, P_{1,3}, P_{2,3})$ are required from both the EAP waveform and second derivative space (total of 6 features). In addition to feature positions, corresponding comparison thresholds $(T_{i,j})$ are also computed. $T_{i,j}$ is the mid-point of the features at $P_{i,j}$ scaled according to the spread (i.e. standard deviation) of feature j and i at $P_{i,j}$. Without such scaling, the threshold will be placed exactly at midpoint, and may cause errors in classification if there exists a significant difference in distribution of features j and i (see Fig. 3).

It should be noted that although training requires a clustering algorithm in order to establish neurons and their templates, perfect clustering and template creation is assumed. This is to prevent the choice of clustering method to affect the evaluation of WDF.

C. Feature Extraction and Classification

Feature extraction relies on the feature positions of maximum separation, $P_{i,j}$, determined during training. As the EAPs are detected, features at $P_{i,j}$ are compared with threshold values $T_{i,j}$. The procedure for comparison based classification

is illustrated in Fig. 3. Considering X is the feature extracted at $P_{i,j}$, it is compared to $T_{i,j}$. Since X is below $T_{i,j}$, it is highly likely that recorded action potential belongs to neuron $\#j$. Hence, neuron $\#j$ receives a vote. This procedure is done for all $P_{i,j}$, and the detected EAP is assigned to neuron with the majority of the votes.

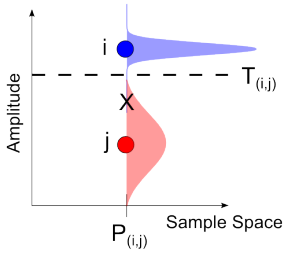

Fig. 3. Illustration of the threshold choice ($T_{i,j}$) based on distributions of features i and j, and the mechanics of comparison based classification.

In order to improve the classification accuracy, a weighted voting scheme is employed by taking into account the standard deviation of the noise. If X is within one standard deviation of $T_{i,j}$, the weighting of the vote is 1. If X is beyond one standard deviation of $T_{i,j}$, the weight of the vote is 2. Although this approach increases computational complexity by one comparison per $P_{i,j}$, the classification accuracy improves by 4.73% on average (8.19% for difficult only), while at some instances improving up to 11%. The standard deviations of noise are calculated separately for EAP waveform and second derivative space.

D. Embedded Hardware Platform

A high level description of the implementation is given in Fig. 4. WDF is implemented and verified on an MCU-based platform, FRDM-KL25Z, running an ARM Cortex-M0+ processor. MCU-implementation also incorporates single-amplitude spike detection as described in [12].

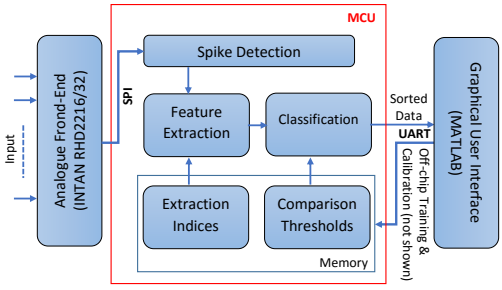

Fig. 4. High level block diagram of the implementation.

Following verification of WDF, RHD2216/32 analogue front-end (AFE) by Intan Technologies is chosen for complete recording system implementation. The AFE provides 16/32 amplifier channels with software configurable bandwidth (0.1 Hz to 20 kHz) and sampling rates (1kS to 30kS). Furthermore, a Graphical User Interface (GUI) is developed in MATLAB which allows users to: (1) stream neural data and perform training, (2) monitor the spike sorting output, (3) display neural recording statistics (e.g. ISI diagrams and spike count rates), and (4) save or load a recording session. It should be noted that Kinetis KL43Z is used for the complete system implementation. This choice is due to the fact that Intan requires 16-bit SPI connection to communicate instructions and data, and KL25Z MCU only has 8-bit SPI option.

IV. Results

The investigations and verification of WDF are performed in Matlab™ R2011b v7.13. It is then followed by hardware specific verification and measurements. For the real-time demonstration of WDF, Keil μVision 5 embedded development environment is used. Synthetic datasets at varying background activity levels (σ_{noise}: 0.05, 0.10, 0.15 and 0.20) are used to assess spike sorting accuracy [12]. In addition to background activity, AWGN arising from electronic noise, axons, dendrites and synaptic currents — can also be significant. Even only considering the contribution from electronics, noise contributions up to 11 μV_{rms} are observed in literature [13]. Therefore, additional AWGN are added to the above datasets.

WDF is compared to other hardware-efficient feature extraction methods such as DD [5] and FSDE [7], as well as template matching (TM) which uses complete sample space. In addition, principle component analysis (PCA) is also included since it is regarded as the "gold standard". WDF is assessed both in terms of sorting accuracy and computational complexity. For the feature extraction methods (from literature) compared herein, K-means has been used as the clustering/classification method. The spike sorting accuracy is evaluated as $\%\ Accuracy = \frac{No.\ of\ correctly\ sorted\ spikes}{Total\ no.\ of\ spikes} \times 100\%$. The computational requirements are assessed through a complexity figure-of-merit $CFOM = N_{add} + 10 \times N_{mult}$, where N_{add} is the number of additions and N_{mult} is the number of multiplications required [5].

The trade-off between sorting error and computational complexity is shown in Fig. 5. It is clearly observed from Fig. 5 (a) that the feature extraction proposed performs significantly better than others. While the two derivative based methods (FSDE and DD) are closest in terms computational complexity, computational efficiency of WDF is more than an order of magnitude. This primarily due to the fact that WDF trades-off memory ($P_{i,j}$) for extracting features. The only computation that dominates CFOM is calculating the second derivative features. However, unlike other methods such as FSDE that computes the derivatives for whole sample space, WDF calculates the second derivatives only at $P_{i,j}$ resulting in better CFOM. Besides feature extraction, classification method is also very efficient. In fact, the classification of each detected EAP takes 18 additions(1 comparison = 1 addition).

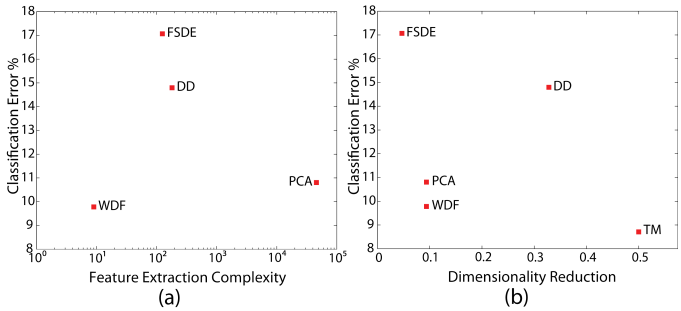

Fig. 5. Comparison of the methods (averaged across datasets and noise levels) in terms of classification error and (a) feature extraction complexity, (b) dimensionality reduction (no. of extracted features / no. of samples per spike)

On the other hand, better sorting accuracy of WDF compared to other derivative features is attributed to two factors. The primary factor is the use of EAP waveform features which are least affected by increasing AWGN levels (compared to

derivative features). The second is the extraction method itself. Since the feature selection is done on averaged waveforms during training (i.e. neuron templates), the effects of AWGN on feature selection is minimised. On-node feature extraction step is merely finding the relevant position of the feature. This is in contrary to derivative methods that needs to operate on derivatives. For example, FSDE relies on calculating max/min on the derivative waveforms which with increasing AWGN may introduce artificial spikes that can be picked up as the max/min resulting in incorrect feature extraction.

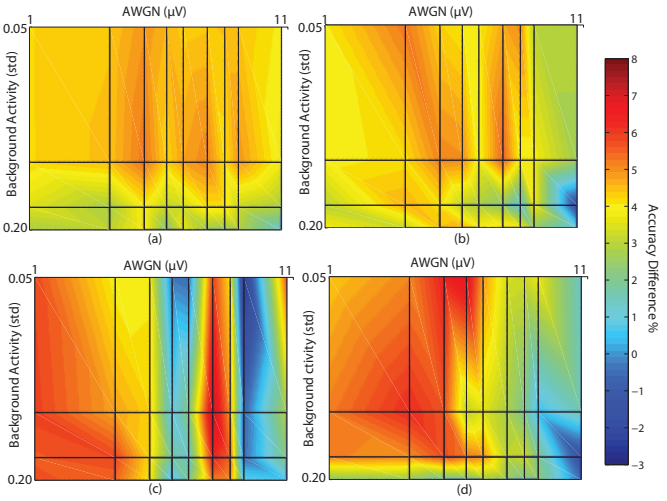

Fig. 6. Spike sorting accuracy difference between Matlab and MCU implementations at various background activity and AWGN levels of test datasets(a-d). (a) Easy1 (b) Easy2 (c) Difficult1 (d) Difficult2

The sorting accuracy difference between Matlab simulations and MCU-implementation is presented in Fig. 6. The average accuracy loss is 5.52% (σ=1.74%) compared to Matlab simulations, while the maximum loss observed is 9.32%. Considering that the sample resolution for real-time implementation is 8-bits compared to double-precision in MATLAB, 5.52% average accuracy loss is an expected trade-off between performance and hardware resource utilisation.

TABLE I. MEASURED POWER CONSUMPTION PER CHANNEL (IN μW)

Ch. No.	Idle	Comm.	Det. & Sort	Total
1	487	4610	700	5800
8	2210	525	388	3130
32	588	110	268	975

The combined power consumption for detection and sorting is measured to be 268μW/channel, which sorting (i.e. WDF algorithm) consumes 5% of (see Table I[1]). Despite the WDF algorithm being very efficient and scalable, implementation is nevertheless off-the-shelf and sub-optimal. For example, 268μW/channel could further be reduced by monolithically integrating the ARM-core (for WDF) and an ASIC realisation of detection. Such implementation is estimated to consume 33.8μW[2] for detection and 0.3μW[3] for sorting code executed in IP Core (without I/O), hence reducing the detection and sorting power (by$\times$8) to 34μW/channel. The total power consumption is also reduced as such implementation eliminates the redundant MCU-peripherals.

[1]The idle power, consumed by having KL25Z on, is measured in the absence of communication, detection and sorting.

[2]Estimation (per channel) includes spike alignment and memory. It is based on implementation synthesized for Igloo Nano AGLN250V2.

[3]Excludes memory access. Assumes 3 neurons/channel (each spiking at 20 Hz).

V. CONCLUSION

A new feature extraction and a classification method have been presented and implemented on an MCU-based platform. When compared against other hardware efficient methods in literature, WDF achieves the best trade-off in spike sorting accuracy and computational requirements.

The high accuracy of the WDF is due to the selection of a channel-specific near-optimal subset of features based on the local extrema of the EAPs and their second derivatives, which are shown to provide the highest separability and noise immunity (both background activity and AWGN). The computational efficiency of WDF is due to simplicity of feature extraction and classification. Feature extraction computes second derivatives only at feature positions, while classification is based only on simple comparison operations.

ACKNOWLEDGMENT

This work was funded by the EPSRC grant EP/I000569/1 and EP/K015060/1.

REFERENCES

[1] M. Rizk, et al., "A fully implantable 96-channel neural data acquisition system," *J. Neural Eng.*, vol. 6, no. 2, p. 026002, 2009.

[2] M. Chae, et al., "A 128-channel 6mW wireless neural recording ic with on-the-fly spike sorting and uwb tansmitter," *Proc. IEEE ISSCC*, pp. 146–603, 2008.

[3] V. Karkare, S. Gibson, and D. Marković, "A 130-μW, 64-channel neural spike-sorting dsp chip," *IEEE J. Solid-State Circuits*, vol. 46, no. 5, pp. 1214–1222, 2011.

[4] I. H. Stevenson and K. P. Kording, "How advances in neural recording affect data analysis," *Nature Neuroscience*, vol. 14, no. 2, pp. 139–142, 2011.

[5] S. Gibson, J. W. Judy, and D. Marković, "Comparison of spike-sorting algorithms for future hardware implementation," *Proc. IEEE EMBS Conf.*, pp. 5015–5020, 2008.

[6] K. M. Awais and M. J. Andrew, "On-chip feature extraction for spike sorting in high density implantable neural recording systems," *Proc. IEEE BioCAS*, pp. 13–16, 2010.

[7] S. E. Paraskevopoulou, D. Y. Barsakcioglu, M. R. Saberi, A. Eftekhar, and T. G. Constandinou, "Feature extraction using first and second derivative extrema (fsde) for real-time and hardware-efficient spike sorting," *J. Neuroscience Methods*, vol. 215, no. 1, pp. 29–37, 2013.

[8] Y. Yang, S. Boling, A. Eftekhar, S. E. Paraskevopoulou, T. G. Constandinou, and A. J. Mason, "Computationally efficient feature denoising filter and selection of optimal features for noise insensitive spike sorting," *Proc. IEEE EMBS Conf.*, pp. 1251–1254, 2014.

[9] F. Mechler and J. D. Victor, "Dipole characterization of single neurons from their extracellular action potentials," *J. Comp. Neuroscience*, vol. 32, no. 1, pp. 73–100, 2012.

[10] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currents — EEG, ECOG, LFP and spikes," *Nature Rev. Neuroscience*, vol. 13, no. 6, pp. 407–420, 2012.

[11] C. Gold, D. A. Henze, C. Koch, and G. Buzsáki, "On the origin of the extracellular action potential waveform: a modeling study," *J. Neurophysiology*, vol. 95, no. 5, pp. 3113–3128, 2006.

[12] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.

[13] D. Y. Barsakcioglu, Y. Liu, P. Bhunjun, J. Navajas, A. Eftekhar, A. Jackson, R. Quian Quiroga, and T. G. Constandinou, "An analogue front-end model for developing neural spike sorting systems," *IEEE Trans. BioCAS*, vol. 8, no. 2, pp. 216–227, 2014.