On-Chip Feature Extraction for Spike Sorting in High Density Implantable Neural Recording Systems

Awais M. Kamboh, and Andrew J. Mason

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, Michigan, USA
{kambohaw, mason}@msu.edu

Abstract—Modern microelectrode arrays acquire neural signals from hundreds of neurons in parallel that are subsequently processed for spike sorting. It is important to identify, extract and transmit appropriate features that allow accurate spike sorting while using minimum computational resources. This paper describes a new set of spike sorting features, explicitly framed to be computationally efficient and shown to outperform PCA based spike sorting. A hardware friendly architecture, feasible for implantation, is also presented for detecting neural spikes and extracting features to be transmitted for off chip spike classification.

I. INTRODUCTION

Advances in microelectronics and nanostructures have enabled scientists to combine thousands of electrodes into microelectrode arrays [1], permitting capture of neural signals from hundreds of neurons simultaneously for use in neuroprosthetics applications and neuroscience research. However, existing implantable wireless transceivers lack the capability to transmit the large amount of data generated by these electrodes. This mandates the use of implantable signal processing systems that enable on-chip data reduction while maintaining necessary information embedded in the neural spikes. On-chip feature extraction for spike sorting has been identified as a way to reduce transmission bandwidth [2].

Existing methods of off-chip feature extraction for spike sorting, e.g. template matching, principal component analysis (PCA), and time-frequency transforms are computationally too demanding to be implantable, as reported in the comprehensive survey [3]. For implantable systems, the area required by circuitry is also a major concern in addition to the power consumption during computations. It has been shown that memory elements are the most area consuming blocks in systems incorporating on-chip neural signal processing [4]. Thus, it is highly desired to reduce the number of computations, while using minimum number of memory elements. Most high performance algorithms are not scalable to 1024 channels, keeping within the area and power limitations, due to the incurred hardware cost of complex processing [3,4]. An implanted system has to be low-power, low-area, highly accurate, automatic, and able to operate in real-time [3].

This paper presents a new set of features for spike sorting, named Zero-Crossing Features (ZCF), and compares their sorting accuracy against PCA features for several data sets. ZCF have been explicitly framed to be extremely computationally efficient and to not require any offline training. Figures of merit for different stages of the system have been established. The effect of non-ideal spike detection on the performance of ZCF and PCA was studied. In addition, the computational complexity was analyzed and a scalable hardware architecture suitable for processing hundreds of channels in an implantable system is presented.

II. THEORY

Figure 1 shows the general data flow of a spike sorting system. Input signal X[n] is the recording from electrodes containing spikes from multiple neurons. S[k] is the result of spike detection where only spikes samples are kept while noise is discarded, as a result reducing the number of samples. The spikes are fed to a feature extractor which generates F[p], where the number of values generated per spike equals the number of features.

The features are then used either on-chip or off-chip to classify the incoming spike as belonging to one of the possible neurons on that channel. All multi-channel systems keep track of the spikes by recording a ‘spike arrival time’ and a ‘channel identifier’ in addition to the features. Most systems have two phases of operation. The ‘training’ phase followed by the ‘acquisition’ phase [3, 5]. During the training phase the neural data is transmitted uncompressed and is used by external processors to calculate different parameters, e.g. detection thresholds and feature vectors. These parameters are then relayed back to the implanted module where they are used in acquisition phase. These steps are repeated for all the channels. Nevertheless, ideally, a system feasible for recording from hundreds of channels should require minimum user intervention.

It is important to note that the ZCF feature extraction presented in this paper does not require any training. Throughout this paper, the ‘training’ phase is only used to train detection thresholds and classification algorithms. Although this paper primarily discusses feature extraction, the performance of feature extraction is also dependent on the quality of spike detector used while the overall spike sorting performance is also dependent on the spike classifier.
A. Spike Detection and Classification

Several methods for spike detection have been reported in literature including data transformations, derivatives and template matching [5]. All methods employ a thresholding step for spike detection at the front-end or later in the system. Most high performance methods are computationally very intense and require offline training in addition to a large number of memory elements to maintain templates or temporary data. Thus, time-domain front-end thresholding is the most suitable option for high density multi-channel implanted systems [3], where the thresholds are computed off chip in the training phase. Using two independent thresholds, one positive and one negative, referred to herein as Dual Thresholds (DT), provides significantly improved performance as compared to a single ‘absolute’ threshold. DT is chosen as the spike detector for this study because of its better detection performance.

Once the features have been extracted, several algorithms are available for off-chip spike classification. K-Means and Mahalanobis distance based classification were considered for this study. Mahalanobis classification was found to perform better for almost all tests and was used for classification henceforth. Off-chip Mahalanobis clustering also allows for adaptability in clustering as channel statistics change and spike amplitudes vary over time.

B. Feature Selection and Extraction

PCA and Wavelet Transforms (WT) have traditionally been used for off-chip spike sorting; however, they are deemed unsuitable for implantable on-chip multi-channel feature extraction because of the high computational cost and hardware resources required to map the neural data onto the feature space. Furthermore, both methods require offline training, and neither of the methods provides a generic set of features applicable across channels. This means each channel requires separate training and results in separate sets of features. This high computational cost mandates the design of a new set of features that is explicitly framed to be computationally efficient and do not require any offline training while maintaining good sorting performance.

The most prominent and visually distinctive features of a set of different spikes are the relative amplitude (or energy) of the spikes, the relative position of positive and negative peaks in the spikes, and the widths of the spikes. Each of these features can be used to sort spikes of one kind from the others, however, the ability to sort spikes based on these features falls sharply with an increase in noise level, in signals with low signal to noise ratio (SNR). However, all these features can be combined together to form a new set of features that are distinctive, resilient to noise and computationally feasible. Mathematically this set of features can be expressed as

\[ ZC1 = \sum_{n=0}^{K-1} x(n) \quad ZC2 = \sum_{n=Z}^{K-1} x(n) \quad (1) \]

where \( ZC1 \) and \( ZC2 \) are the two features, \( K \) is the number of samples in a spike and \( Z \) is the index of first zero crossing after the spike has been detected. Note that in a DT based spike detection system, the spikes are detected when the value of the spike is larger than either of the specified thresholds. The value of \( Z \) in (1) is thus the first zero crossing after a significant amount of energy in the spike has already been recorded. Features in (1) can also be seen as recording the pre-zero-crossing and post-zero-crossing energies of the spike. Collectively, these features are referred to as Zero-Crossing Features (ZCF) for the rest of the paper.

The underlying assumption in selection of these quantities as a feature set is that the spikes from different neurons have different ‘area’ before and after the first zero crossing. Though, in theory, it is possible to have multiple neurons generating spikes with same energy before and after zero crossing, in practice, experimentally recorded spike waveforms show considerable differences in these two values allowing for the use of these features for spike sorting. In case the spikes have no zero crossing, \( ZC1 \) contains all the area of the spike while \( ZC2 \) remains zero. Even if multiple non-zero-crossing spikes are present in the channel, the features still maintain a degree of separability. An average spike is about 1.5ms long, which at 20Kaps sampling rate amounts to 30 samples. The factor ‘\( K \)’ in (1) is thus set to 30 and is dependent on the sampling rate.

The ZCF are graphically represented in the example of Fig. 2. When a spike is generated, the electrode potential changes very rapidly. In this example, the spike is detected when the electrode potential crosses the negative threshold. There are still some samples before the detection that belong to the spike and may contain useful information and thus need to be stored in a buffer. For the sampling rate of 20Kaps, empirical results show that a buffer length of only 3 successfully captures all the useful samples and captures the onset of the spike, as shown in Fig. 2.

![Figure 2. Graphical representation of ZCF features for a spike detected using dual thresholds.](image-url)
C. Complexity Calculations and Hardware Resources

The computation complexity of implementing the two features of ZCF was compared to that of implementing the two most significant components of PCA, where each of the PCA components has $K$ dimensions. Computations for spike detection algorithms are for each input sample, while the computations for feature extraction are for each detected spike. To compare complexity in terms of addition operations, each multiplication is considered equal to 10 additions [5] whereas each ‘comparison’ operation is considered equal to one addition. For the case of DT spike detection, use of two thresholds means that every sample is compared against either positive or negative threshold, thus only one computation per sample.

From Table I, it can be seen that for each detected spike, the ZCF needs less than 5% computations and requires less than 8% memory as compared to the PCA. Fig. 3 gives the hardware architecture of the proposed signal detection and feature extraction algorithms. Light colored blocks represent memory elements.

### Table I. Comparison of Complexity and Required Memory for Different Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Add.</th>
<th>Mult.</th>
<th>Complexity</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>ZCF</td>
<td>30*</td>
<td>0</td>
<td>30*</td>
<td>5</td>
</tr>
<tr>
<td>PCA</td>
<td>60*</td>
<td>60*</td>
<td>660*</td>
<td>65</td>
</tr>
</tbody>
</table>

* Computations per spike

A three word FIFO is updated every new sample. When a spike is detected the control logic uses the data stored in FIFO and the subsequent input values directly to accumulate ZC1. When the zero-crossing-detector (ZCD) detects a change in sign then ZC2 is accumulated until all the 30 samples representing a spike are received.

Figure 3. Hardware architecture of proposed signal detection and feature extraction stages.

It is important to note that, because the system is being prototyped for 1024 channels; one additional byte of storage per channel translates to 1Kbyte of storage for the system. Similarly, each additional computation per channel means 1024 additional system computations.

### III. Methods

Neural spikes recorded from live experiments were used in conjunction with a neural signal simulator to generate signals that mimic electrode recordings. A detailed discussion of experimental procedures to collect neural data can be found in [6]. A total of ten different spike shapes were used to mimic ten separate neural channels, each consisting of three spikes. Figure 4 shows the 10 spike channels used in these simulations. For each dataset, one second of ‘training’ phase was completed to determine thresholds for DT, principal components for PCA and initial statistics for Mahalanobis clustering. The signals generated did not contain any overlapped spikes, and are generated for different values of SNR, ranging from 18dB to -5dB, as computed by

$$\text{SNR} = 20 \log_{10} \left( \frac{\sigma_x}{\sigma_n} \right)$$  \hspace{1cm} (2)

where $\sigma_x$ and $\sigma_n$ are the signal and noise standard deviations.

Figure 4. Channels and spikes used in simulations. Ten different channels consisting of ten different spike shapes are used with each channel containing three spike shapes.

The performance accuracy of a spike sorting system depends on the collective and individual performance of each of the components. Performance accuracy of the detection algorithms is computed based on the number of spikes missed and the false alarms in relation to the number of original spikes generated by the simulator. Detection Error for Dual Threshold is represented by [7].

$$DE_{DT} = \frac{MS + FA}{OS + FA} \times 100$$  \hspace{1cm} (3)

where OS is the number of original spikes, FA is the number of false alarms and MS is the number of missed spikes. Performance of ZCF is compared against the performance of PCA; however, PCA requires a training phase to find the two most significant principal components. When ‘perfect’ detection is used then Classification Error can be calculated as

$$CE_p = \frac{MCS}{OS} \times 100$$  \hspace{1cm} (4)

where MCS is the number of mis-classified spikes. Accuracy of the feature extraction algorithm is computed through (4) treating misses or false alarms as misclassified spikes. The effects of non-ideal spike detector on classification accuracy are reflected in the classification error formula

$$CE_{DT} = \frac{MCS - FA}{DS - FA} \times 100$$  \hspace{1cm} (5)

where DS is the total number of spikes detected. The performance of feature extraction is isolated from the performance of non-ideal spike detector by (5). Since
different sets of features are effected differently, (5) represents the effects of mis-alignment caused due to imperfect spike detection on the classification accuracy. Here number of misclassified spikes includes all the false alarms since false alarms are always wrongly classified. The performance of the complete system including the non-ideal DT spike detector, feature extractor and classifier can be computed as

\[ DCE_{DT} = \frac{MS + MCS}{OS + FA} \times 100 \quad (6) \]

IV. RESULTS AND DISCUSSION

Figure 5 shows the performance results of detection and classification stages using the four figures of merit described by (3)-(6). Ten different data sets were used in simulations, and the resulting averages have been plotted. Figure 5(i) shows that the positive and negative DT threshold values that generate the minimum detection error (DEDT) in the initial training phase result in comparable error in the acquisition phase. The detection error increases sharply for a SNR of 3dB and below where noise power becomes comparable to signal power. Fig. 5(ii) compares the performance of ZCF and PCA and presents the classification error when ideal detection is assumed (CETP). It can be seen that ZCF and PCA perform equally well even under low SNR conditions. This plot shows that, even though ZCF requires only 5% of the resources as PCA, it can perform as well as PCA-based features. Fig. 5(iii) shows that when non-ideal spike detection is employed, the resulting mis-alignment of detected spikes adversely effects the classification performance (CETD) of both algorithms. The errors have increased by about 50% for each SNR. However, it can be seen that there is no performance loss at high SNRs and that both ZCF and PCA still perform equally well for low SNRs. Fig 5(iv) shows the system level results including spike detection, feature extraction and classification. The performance of both algorithms is nearly indistinguishable. It can be seen that even though the ZCF and PCA are capable of performing well under low SNR conditions (e.g. 5% errors at 0dB), the system performance (55% errors at 0dB) is compromised by the performance of the spike detector (45% errors at 0dB in addition to the mis-alignment of detected spikes). Note that this analysis does not reject the possibility that if more than two features are used in PCA, at the cost of increased hardware resources, then its classification performance may improve.

A 1024 channel system with a sampling frequency of 25Ks/sec/channel and 10 bits ADC resolution would generate raw data at about 250Mbits per second. Using ZCF feature extraction, the information to be transmitted could be reduced to a time stamp, a channel ID and the two computed features (around 50 bits per spike). Assuming the average number of neurons on each channel to be three and the neuron firing rate to be 30 spikes per second, data would only need to be transmitted at the rate of \( \approx 4.7 \)Mbits using ZCF, which is less than 2% of the original rate. The resulting savings to transmission power is purchased by the cost of ZCF computations, which average to, 2.86 million computations per second during steady state operation. In comparison, PCA provides similar data reduction but costs 60.91 million computations per second, many times dynamic power of ZCF.

![Figure 5](image-url)  
Figure 5. Percentage values of (i) DEDT, (ii) CEP, (iii) CETD and (iv) DCEDT versus different values of SNR. Data represents performance results for the DT spike detector, the ZCF and PCA feature extractor and the combined spike sorting system.

V. CONCLUSIONS

A new set of computationally efficient features for spike sorting has been presented and compared against PCA for spike sorting accuracy and hardware resource demand. ZCF provides equally good performance to PCA while consuming only 5% of the resources. In addition, ZCF does not require any offline training and is applicable across all channels. A scalable hardware friendly architecture has also been presented, resulting in huge savings in transmission bandwidth for wireless implanted neural recording systems.

REFERENCES