Power Feasibility of Implantable Digital Spike-Sorting Circuits for Neural Prosthetic Systems

Zachary S. Zumsteg[1], Rizwan E. Ahmed[1], Gopal Santhanam[1], Krishna V. Shenoy[1,2], Teresa H. Meng[1]
[1]Department of Electrical Engineering, [2]Neurosciences Program, Stanford University, Stanford, California, USA

Abstract— **A new class of neural prosthetic systems aims to assist disabled patients by translating cortical neural activity into control signals for prosthetic devices. Based on the success of proof-of-concept systems in the laboratory, there is now considerable interest in increasing system performance and creating implantable electronics for use in clinical systems. A critical question that impacts system performance and the overall architecture of these systems is whether it is possible to identify the neural source of each action potential (spike sorting) in real-time and with low power. Low power is essential both for power supply considerations and heat dissipation in the brain. In this paper we report that several state-of-the-art spike sorting algorithms implemented in modern CMOS VLSI processes are expected to be power realistic.**

Keywords— **Neural prosthetics, spike sorting algorithms, spike sorting circuits, power estimation**

I. Introduction

Neural prosthetic systems that assist disabled patients by translating neural activity from the brain into control signals for prosthetic devices rely on neural recordings from implanted electrode arrays. Electrode arrays are implanted neurosurgically in the brain region of interest, but the precise distance between each electrode tip and surrounding neurons is uncontrolled. As a result, implantable electrodes are manufactured with only moderately high impedances (e.g., 100-300 KO) to ensure recordings from at least one neuron. In practice, such electrodes typically record action potentials from more than one neuron. It is standard scientific practice to assign each distinct action potential waveform to a different neuron. This so-called spike sorting process results in neural activity attributable to identified neurons, and is widely thought to allow maximum information extraction from a given data set since individual neurons are believed to be the fundamental information coding units in the brain.

If all of the action potentials appearing on one electrode are considered to come from a single neuron, two problems arise. First, it is incorrect to assume that just because two neurons are close enough to each other to influence the same electrode that they have identical tunings for movement. Just because one neuron responds strongly to rightward arm movements does not mean that all of the nearby neurons respond in the same fashion. Properly attributing action potentials to their neural source can lead to significantly greater information extraction, which is essential for high-performance neural prosthetic systems. Second, even if nearby neurons do have similar tunings, if the different waveforms are all considered to come from one neuron the activity of this "neuron" is overestimated. Such overestimates, and incorrect attribution in general, impedes answering questions about how neural activity changes and adapts through time. Taking such adaptation into account is another essential feature in modern neural prosthetic system design.

There appears to be significant doubt in the neural prosthetic system community that implanted real-time spike sorting is possible. The concern is that spike sorting, though valuable if not essential for high-performance systems, may be too power consuming, large and even complex to be implemented in an implantable neural prosthetic system.

Implantable spike sorting is important for one additional reason, beyond those mentioned above. Even with a relatively conservative sampling rate of 10kHz, it is not realistic to telemeter out broadband data for hundreds or thousands of electrodes simultaneously [1]. Data compression techniques are needed. An implantable spike sorting integrated circuit could be placed between the ADC and the telemetry circuits resulting in a system that broadcasts the time and neural identity of every action potential.

In this paper, we investigate the practicality of implementing a digital spike-sorting integrated circuit. We consider two modern spike sorting algorithms and argue that the number of computational operations required and the energy consumed by using standard CMOS VLSI make implantable spike-sorting chips realistic even for arrays with very large numbers of electrodes.

II. Methodology

A. Spike Sorting Overview

Most spike sorting algorithms consist of two separate stages. The first stage trains the algorithm on a fixed amount of recorded data so that a set of parameters can be learned. These parameters will be used for real-time data classification in the second stage. Certain features are common to nearly all spike sorting algorithms. First, a high pass filter must be used to eliminate the 0.5Hz to 100Hz low frequency local potential signal (LFP). Then, spike times

This work was supported by MARCO/DARPA (T.H.M.), an NDSEG fellowship (G.S.) and the following awards to K.V.S.: NSF Center for Neuromorphic Systems Engineering at Caltech, ONR Adaptive Neural Systems, Whitaker Foundation, Center for Integrated Systems at Stanford, Sloan Foundation, and Burroughs Wellcome Fund Career Award in the Biomedical Sciences. Please address correspondence to thm@stanford.edu.

must be identified. Often, events are aligned to their exact peaks through interpolation techniques. Most algorithms also use dimensionality reduction to reduce the computational burden and prevent over fitting the training data set with a confining, complex model.

However, various training algorithms differ in several key aspects, including the method used for dimensionality reduction, the metric used to decide which neuron a spike is associated with, and the iterative clustering algorithm used to obtain an optimal set of parameters in the training stage.

B. Spike Sorting Algorithms Selected

In this paper, two different spike sorting algorithms of different complexity were implemented. The algorithms selected are not necessarily representative of the vast number of spike sorting algorithms available, but by focusing on a small set of algorithms, it allows us to show that real time spike sorting is possible, even in the most complex cases. Both training algorithms begin with a high pass filter with a 600 Hz cutoff, threshold at 3s, and use oversampling and interpolation to align spikes to their centers of mass.

1) The K-means Algorithm

The training stage of the K-means algorithm is shown in Fig. 1. The set of aligned events are projected into the space spanned by the first three principle components of the data. The spikes are randomly grouped into M data sets, where M is the number of neurons per electrode. The means of the partitions are calculated, and then each spike is regrouped according to which mean it is closest to. This process is iterated until convergence.

For real time classification, detected events are projected along the stored principle components and classified as belonging to the cluster whose mean has the smallest squared Euclidean distance from the projected point.

2) The Sahani Spike Sorting Algorithm

The training algorithm for Sahani's spike sorting algorithm (SA) is shown in Fig. 2. SA uses samples of the background noise to project the events into the noise whitened principle component analysis (PCA) space [2].

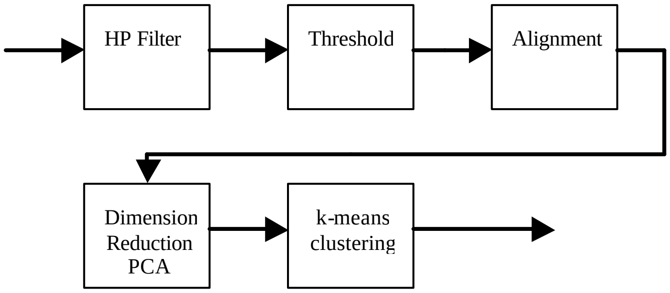

Fig 1. High level block diagram of K-means training

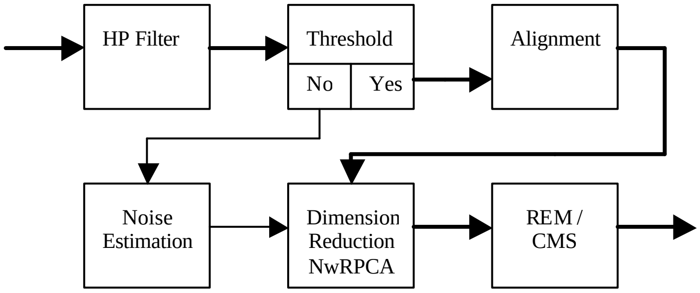

Fig 2. High level block diagram of SA Training

Relaxation Expectation-Maximation (REM) and cascading model selection (CMS) are used to cluster the data and fit the clusters to a mixture model. Real time classification consists of filtering, thresholding, projecting into the noise-whitened robust PCA space, and then using maximum *a posteriori* classification for assignment.

Our justification for choosing Sahani's algorithm is as follows: SA is one of the most sophisticated and computationally intensive algorithms published. By analyzing the estimated power consumption of this algorithm, we can set an upper bound on the power consumption of current algorithms.

C. Power Estimate Methods Used

Power estimation of the above mentioned spike sorting algorithms is calculated by recasting the operations performed to simple instructions that can be implemented in hardware. A detailed analysis of the algorithms was carried out and approximate figures for the number of operations (namely adds and multiplies) required for each task were obtained. Operation counts for some complex linear algebra functions used in the algorithms, like matrix decompositions, were taken from standard texts on numerical linear algebra [6]. Operation counts were then translated to power by using the figure 1mWatt/GOPS [3]. This figure is used as the standard power consumption per operation for ASICs implemented in 0.13µm technology. Finally, an approximate power usage from memory accesses was added. This was taken as double the power from instruction execution [4]. The figures should be taken as an "order of magnitude" indication. However, we believe that these figures are indicative of the power consumption, and thus achieve the objective of showing that these systems can be implemented in an implantable neural prosthetic.

D. Generation of Realistic Synthetic Data

Realistic synthetic data sets were generated in order to obtain the error rates for the various algorithms. While the data tested is not representative of every possible situation that might arise in a neural prosthetic application, it does give one a general of idea of how well the algorithms could perform in certain situations and how their error rates might be expected to compare. Templates were obtained from several sets of actual neural data recorded from a rhesus macaque monkey. The relative and absolute firing frequencies of each neuron were noted.

Segments of synthetic neural data were then generated. First, an action potential template is chosen at random from a given set, where the probability of choosing a particular template is given by that neuron's relative firing rate in the actual data set. Then, a spike time was chosen from a uniform random variable between zero and thirty seconds. The refractory period considerations were included. The template waveform was then run through a linear phase interpolating filter with a uniform random delay between zero and one sampling period to randomize the location of the peak with respect to the sampling interval. Noise and the LFP extracted from the actual data were then added.

Error rates were obtained by training the spike sorting algorithm on one set of synthetic data, and then running a real time classification algorithm on a second set of synthetic data generated from the same templates.

III. RESULTS

A. *Power Estimates*

1) Training

We assume that, since electrode movement is generally very small, the spike signals will be relatively stable over time, and thus training needs to be performed relatively infrequently. We assume that training is conducted once every twelve hours for each electrode. Also, we assume that we will train one electrode at a time, and that training will be conducted continuously. Even with 1000 electrodes, a relatively large number that anticipates electrode scaling trends, this allows training to take place over approximately 43 seconds for each electrode. Also, it is assumed that just before an electrode is to be trained, a thirty second segment of recent, filtered neural data is stored in memory.

To obtain power estimates, upper bounds were used for some parameters based on what would be observed in a typical data set. For example, the size of the training data is set to 900 spikes, a rather large number that assumes multiple, active neurons influencing one channel. Also, due to the fact that clustering and statistical fitting algorithms are iterative by nature, an upper bound of 20 iterations was assumed. The operation counts for various parts of SA and K-means training are listed in Table I. We counted basic multiply and add operations, and all other operations were recast to these fundamental units.

We see that the training algorithm requires approximately 2.74×10^8 operations per electrode. With our previous assumption of 43 seconds of training time per electrode, we can convert to a power number by the following expression:

$$Power = (Total\ ops / training\ time) * 1mW / (10^9 ops / sec) \quad (1)$$

TABLE I

Operation Estimates for Spike Sorting Algorithms

Step	SA Operations	k-means Operations
RMS Calculations	1.20E+02	1.20E+02
Spike Alignment	1.20E+08	1.20E+08
Noise Covariance	1.15E+05	---
PCA	6.37E+06	1.45E+06
Model Training	1.48E+08	7.27E+06
Total	**2.74E+08**	**1.29E+08**

In order to account for memory accesses, we double the resulting power number. Therefore, SA training power with 1000 electrodes is estimated as 13 µW, and the total power for K-means training is estimated to be 6 µW.

2) Real Time Classification

The classification process itself contributes relatively little to the overall power consumption of real time spike sorting. Most of the computational burden is dominated by the high pass filter and thresholding, both of which must be done in any digital neural prosthetic application, with or without spike sorting.

We will assume that an IIR filter consisting of two second order sections is used. Also, a 30kHz sampling rate is assumed. The IIR filter necessitates 6.3×10^5 ops/sec/electrode. Digital thresholding contributes 3×10^4 ops/sec/electrode. Combined, these sections should contribute about 1.32µW/electrode. Assuming a "worst case classification complexity scenario", in which there are five different neurons influencing an electrode and fire at a combined average rate of 50 spikes per second, classifying the data using a Euclidean distance metric requires 1.3×10^4 ops/sec/electrode. This corresponds to 0.026µW/electrode. In other words, even in the worst case for classification with a conservative filter, classifying the spikes requires only about a 2% increase in power over digital filtering and thresholding. In ordinary situations where only one or two neurons are recorded by an electrode and where the average firing rate is between 5 to 10 spikes/sec, the number of operations required for classification will typically be on the order of 1000 ops/sec/electrode.

Classification using maximum a posteriori yields similar results. Making similar "worst case assumptions" as before. MAP requires about 4.8×10^4 ops/electrode/sec, which corresponds to about 0.096µW/electrode. Classification power will again usually be about an order of magnitude smaller under normal conditions.

As such, the biggest hurdle to be overcome in real time classification, or any digital application, is the design of a power efficient method for high pass filtering broadband data over hundreds or thousands of channels at a time. The problem is made more difficult by the fact that the LFP is in the 0.5 - 100Hz frequency range, while much of the signal power is concentrated in the 1000-3000Hz range. With a sampling frequency of 30kHz, the necessary transition band is somewhat steep. Also, the amplitude of the LFP is often as large as the amplitude of the most prominent spike waveform, so the stopband attenuation must be fairly significant. Therefore, while FIR filters may give somewhat better performance results than IIR filters, the higher order they necessitate requires a prohibitive amount of computation for filtering hundreds of channels at a time.

B. Algorithm Performance Assessment

The most important characteristic of any spike sorting algorithm is its ability to accurately classify action potentials. How well the algorithms perform this task depends on the characteristics of the signal, like the SNR, the number of neurons, and how different the waveforms are. Using an IIR filter consisting of two second order sections, detected spikes were classified with greater than 90% accuracy in all tested cases with SA and greater than 85% accuracy in all tested cases with K-means.

As expected, both SA and the K-means algorithm performed similarly for high SNR channels. However, for lower SNR channels, or channels with spike waveforms that are fairly similar in amplitude and shape, SA's performance is vastly superior to the K-means algorithm.

The effects of different SNR levels were investigated by using several high pass FIR filters of different orders with a data set containing two action potential shapes of similar amplitudes. With a 56th order filter, both algorithms classified over 90% of the detected spikes correctly. SA also classified over 94% of detected spikes correctly with greatly reduced SNR using a 10th order filter. The K-means algorithm, on the other hand, fails catastrophically at low SNR. Even with a 32nd order high pass filter, it classifies less than 50% of the action potentials correctly.

Both algorithms are equally susceptible to missed spikes with low order FIR filters. For example, both failed to detect 67% of the total spikes with a 10th order FIR filter. Thus, although accurate classification can be obtained with low order FIR filters, the large amount of missed spikes prohibits using them in high performance systems.

Another problem that arises in spike sorting algorithms with outlier elimination is that they can mistakenly throw away legitimate action potentials, especially when two neurons fire temporally close enough such that their action potentials overlap. However, our experiments show that both algorithms classify actual spikes as outliers rarely. Less than 1% of spikes were eliminated in most cases. However, in cases in which neurons are highly correlated, special signal processing techniques for handling overlapping spikes not included in the tested algorithms may be needed.

SA offers many other advantages over simpler spike sorting algorithms. By using cascading model selection, SA can actually determine the number of neurons on its own, which is a crucial feature of any unsupervised spike sorting algorithm. Also, by projecting the data in the noise whitened principle component space, SA maximizes the separability of the clusters. In other words, this projection has the greatest ratio of the average distance between clusters to the average spread of the data. It is therefore possible to separate clusters that would be indistinguishable in regular principle component analysis.

This performance assessment combined with the previous result on power estimation indicates that more sophisticated spike-sorting algorithms should be used, because their power penalty is very small compared to that of data filtering.

IV. DISCUSSION

We have shown that currently available spike sorting algorithms can be both reliable and power efficient. With 100 electrodes, we can upper bound the power consumption to be about 143µW. Assuming a 5mm^2 chip, this gives a power to area ratio of about 2.9mW/cm^2, which is well below the 80mW/cm^2 chronic heat dissipation threshold believed to cause tissue damage [5].

In addition, we have found that the high pass filter stage both dominates power consumption and greatly affects algorithm performance. Alternative methods of multichannel filtering for spike sorting should be investigated to further reduce the power consumption and improve performance as the number of available electrodes expands. Two immediate solutions are to use a lower sampling rate and analog bandpass filters that simultaneously prevent aliasing and eliminate the LFP.

V. CONCLUSION

We have shown that digital spike sorting is feasible using currently available algorithms and technology. In addition, we have shown that algorithm training and real time classification combine to give less than a 10% power increase over simple filtering and thresholding even when assuming worst case conditions for every electrode.

REFERENCES

[1] R. Harrison, "A low-power integrated circuit for adaptive detection of action potentials in noisy signals," *Proc. 2003 Intl. Conference of the IEEE EMBS*, September 17-21, 2003.

[2] M. Sahani, *Latent Variable Models for Neural Data Analysis*, California Institute of Technology, 1999.

[3] A.P. Chandrakasan, S. Sheng and R.W. Brodersen, "Low Power CMOS Digital Design", IEEE Solid State Circuits Society Quarterly Newsletter, April 3, 2003.

[4] Teresa H. Meng, ``Low-Power Signal Processing System Design for Wireless Applications," invited paper, IEEE Personal Communication Magazine, Vol. 5, No. 3, pp. 20-31, June 1998.

[5] T. M. Sees, H. Harasake, G. M. Saidel, and C. R. Davies, "Characterization of tissue morphology, angiogenesis, and temperature in adaptive response of muscle tissue to chronic heating," *Lab. Investigation,* vol. 78(12), pp. 1553-1562, 1998.

[6] G. Golub, C.F.Van Loan, *Matrix Computations* , Baltimore, MD: John Hopkins University Press, 1983