A 41 µW real-time adaptive neural spike classifier

Zjajo, A.; Leuken, R. van

DOI
10.1109/bhi.2016.7455941

Publication date
2016

Document Version
Accepted author manuscript

Published in
2016 IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI 2016

A 41 µW Real-Time Adaptive Neural Spike Classifier

Amir Zjajo, Rene van Leuken

***Abstract*—Robust, power- and area-efficient spike classifier, capable of accurate identification of the neural spikes even for low SNR, is a prerequisite for the real-time, implantable, closed-loop brain-machine interface. In this paper, we propose an easily-scalable, 128-channel, programmable, neural spike classifier based on nonlinear energy operator spike detection, and a boosted cascade, multiclass kernel support vector machine classification. The power-efficient classification is obtained with a combination of the algorithm and circuit techniques. The classifier implemented in a 65 nm CMOS technology consumes less than 41 µW of power, and occupy an area of 2.64 mm^2.**

I. Introduction

Neural prosthetic devices require a large number of parallel electrodes to be implanted into relevant cortical regions [1]. However, very frequently an electrode records the action potentials from multiple surrounding neurons (e.g., due to the background activity of other neurons, slight perturbations in electrode position, or external electrical or mechanical interference), and the recorded waveforms/spikes consist of the superimposed potentials fired from these neurons [2]. Clustering spike-derived features is, due to the contaminating noise, a challenging task; the degree of overlap between the annotated clusters increases as a function of the noise variance. The ability to distinguish spikes from noise, and to distinguish spikes from different sources from the superimposed waveform, therefore depends on both the discrepancies between the noise-free spikes from each source, and the signal-to-noise level (SNR) in the recording system.

The space to host a multi-channel, implantable, neural recording systems is restricted to ensure minimal tissue damage and tissue displacement during implantation. Furthermore, power density of the entire system (including the analog front-end, signal sorting, wireless telemetry, energy harvesting, etc.) is limited to 800 µW/mm^2 [3] to prevent possible heat damage to the tissue surrounding the device (and subsequently, limited power consumption prolong the battery's longevity and evade recurrent battery replacements surgeries). In addition, for high-performance neural prosthetic devices, the high-density, raw data rate recording is required. A 128-channel, 10-bit-precise digitization of neural waveforms sampled at 40 kHz generates ~ 51 Mbs^{-1} of data; the power costs in signal conditioning, quantization and wireless communication all scale with the data rate.

This research was supported in part by the European Union and the Dutch government, as part of the CATRENE program under Heterogeneous INCEPTION project.

A. Zjajo and R. van Leuken are with Circuits and Systems Group, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands (e-mail: amir.zjajo@ieee.org).

In this paper, we propose a 128-channel, programmable, neural spike classifier based on nonlinear energy operator spike detection, and multiclass kernel support vector machine (SVM) classification. The power-efficient, multi-channel clustering is achieved by a combination of the several algorithm and circuit techniques, namely, the Kesler's transformation, a boosted cascade reduced set vectors approach, a two-stage pipeline processing units, the power-scalable kernels, the register-bank memory, a high-V_T devices, and a near-threshold supply. The results obtained in a 65 nm CMOS technology show that an efficient, large-scale neural spike data classification can be obtained with a low power (less than 41 µW, corresponding to a 15.5 µW/mm^2 of power density), compact, and a low resource usage structure (31k logic gates resulting in a 2.64 mm^2 area).

II. Real-Time Adaptive Spike Classification

A. Architectural Overview of the Neural Interface

The data acquired by the recording electrodes in 128-channel (8×16 arrangement) neural recording interface is conditioned using analog circuits, as illustrated in Figure 1. Each channel consists of an electrode, a low noise pre-amplifier (LNA), a band-pass filter, and a programmable gain post-amplifier (PGA), while an 10-bit A/D converter (ADC) is shared by 16 post-amplifiers through time-multiplexing. The ADC output is fed to a back-end signal processing unit, which provides additional filtering and executes a spike sorting. Several previous spike-sorting DSP realizations [4]-[6] have implemented spike detection and feature extraction, however, most spike sorting clustering algorithms, e.g., means, and superparamagnetic clustering, are offline, unsupervised algorithms not usable for real-time data streams.

In the proposed design, first threshold crossings of a local energy measurement [7] are used to detect spikes. A frequency-shaping filter significantly attenuates the low frequency noise and helps differentiating similar spikes from different neurons. The feature extraction based on maximum and minimum values of spike waveforms first derivatives [8] is employed due to its small computation and little memory requirement, while preserving high information score. Neural spikes are classified with multi-class support vector machine [9]. The relevant information is then transmitted to an outside receiver through the transmitter, or used for stimulation in a closed-loop framework.

B. Spike Detection

The 10-bit time-multiplexed neural data, sampled at 40 kS/s is applied to the control unit (Figure 2). A 4kB instruction memory and 8kB data memory offer spike detection algorithm programmability, and parameter set flexibility. The system control unit is loaded with 32 10-bit filter coefficients, and a 16-bit threshold value.

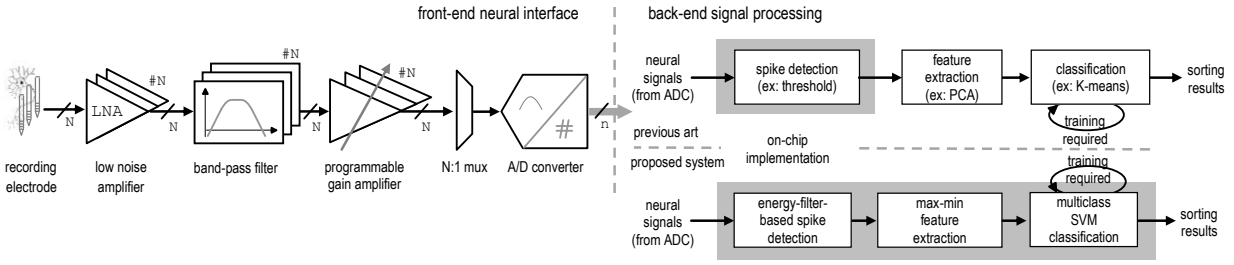

Figure 1: Block diagram of a brain machine interface with *N*-channel front-end neural recording interface and back-end signal processing.

The spike detector algorithm calculates the energy function for waveforms inside a slicing window; when a spike event reaches the threshold, a spike data is stored and transferred for the alignment process and further feature extraction. The noise shaping filter provides the spike waveforms derivatives to identify neurons' kernel signatures (including the positive and negative peaks of the spike derivative, and spike height). The filter coefficients are programmable through the coefficient register array. Consequently, a variety of noise profiles and spike widths can be precisely tuned. To attain the marginal phase distortion, we utilized Bessel filter structure. For real-time, high signal throughput, all spike processing operations, including detection, filtering, and feature extraction are performed in parallel.

The SRAM is implemented as the register-bank memory, since it can be scaled to sub-threshold voltages (i.e., to reduce the leakage power). In contrast, the compiled SRAM has limited read noise margin, and consequently, cannot be scaled below 0.7V. The register-bank memories are organized as spike registers [4], as shown in Figure 2b). Each spike register module consists of 10-bit registers to save the spike waveforms, and a delay line for clock gating. The decoder enables sequential, clock controlled selection of each spike sample *S* from a spike register. In each 10-bit spike register, only 1-bit D-flip-flops have an active clock. Accordingly, such delay-line-based clock-gating arrangement reduces the redundant clock transitions, and subsequently, allows 10 fold reduction in the clock-switching power (corresponding to a 32% reduction in the total power consumed by the memory).

C. Boosted Cascade SVM Classification

Let us consider labelled training spike trains $\{(x_i, y_i): i \in I\}$, where the discriminant function $f_m(x), m \in K=\{1,\ldots,k\}$ separates training data of the *m*-th class from the other training patterns, and *I* is set of indices.

The pattern x_i is from an *n*-dimensional space $\mathcal{X}$, and its label attains a value from a set *K*. We transform the multiclass SVM problem to the single class problem with Kesler's construction [9]-[11]. Since the support vector $sv(x_i)$ appears only in the form of dot products in the dual form, we can construct the dot product (x_i,x_j) using the Kronecker delta, i.e., $\delta(i, j)=1$ for $i=j$, and $\delta(i, j)=0$ for $i \neq j$, and map it to a reproducing kernel Hilbert space [9]-[11]. The SVM classification is then composed of the set of discriminant functions

$$f_j(x) = \sum_{i \in I} \psi(sv(x_i) \cdot x)) \sum_{m \in K \setminus \{y_i\}} \alpha_i^m (\delta(j, y_i) - \delta(j, m)) + b_j, \; m \in K \setminus \{y_i\}, \quad (1)$$

where the vector b_j is given by

$$b_j = \sum_{i \in I} \sum_{m \in K \setminus \{y_i\}} \alpha_i^m (\delta(j, y_i) - \delta(j, m)), \qquad m \in K \setminus \{y_i\}, \quad (2)$$

α_i are weight vectors, $m \in K \setminus \{y_i\}$ are multiclass labels excluding y_i, and $\psi(.,.)$ is a symmetric, positive semidefinite Mercer kernel. For $\psi(.,.)$ one typically has the following choices: $x^T sv(x_i)$ ((weak) linear SVM)); $(x^T sv(x_i)+1)^d$ (polynomial SVM of degree *d*); $\tanh[\chi(x^T sv(x_i))+\upsilon]$ (multilayer perceptron (MLP) SVM); and $\exp\{-\gamma \|[sv(x_i)-x]\|_2^2/\sigma^2\}$ ((strong) radial basis function (RBF) SVM), where χ, υ, γ and σ are positive real constants. The kernels yield increasing levels of strength (e.g., false alarm for linear kernel of 18 per day decrease to 1.2 per day for RBF kernel [12]). However, the required power for each kernel (from simulation of the CPU) varies by orders of magnitude.

The complexity of the computation of (1) scales with the number of support vectors. To simplify the kernel classifier trained by the SVM, we extend iterative greedy optimization reduced set vectors approach [9],[13] with boosted cascade classifier (Figure 3). Consequently, we assess the reduced expansion in a cascaded way, such that in most cases a very small number of support vectors are applied.

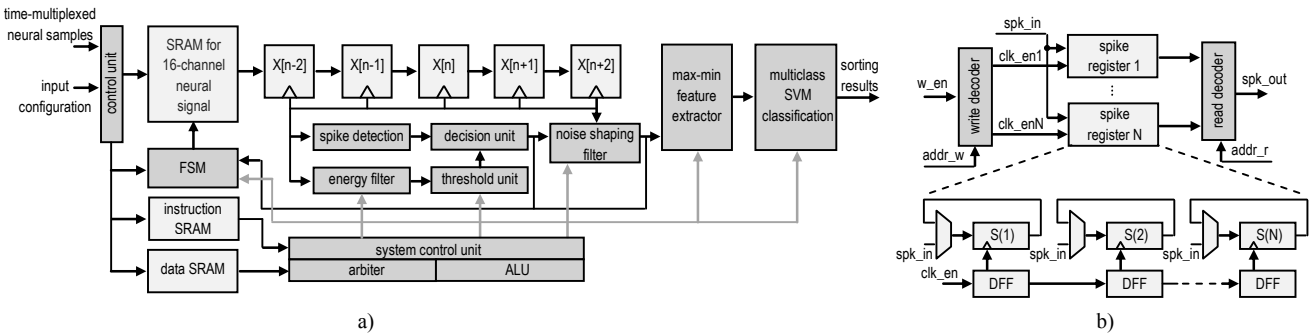

Figure 2: a) The architecture of the back-end signal processing, b) selectively-clocked register bank memory.

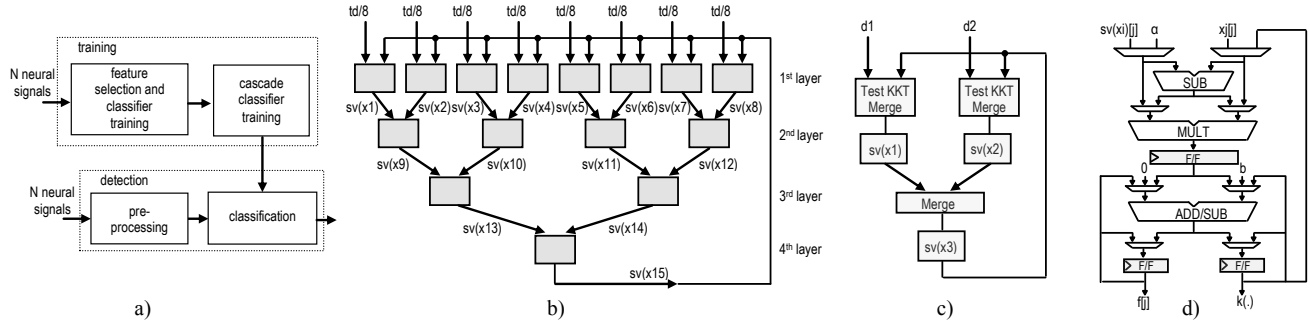

Figure 3: a) Cascaded SVM framework, b) binary boosted cascade architecture, c) a cascade with two input sets, d) two stage pipeline processing unit.

Consider a set of reduced set vectors classification functions, where the *j*-th function is an approximation with *j* vectors, chained into a sequence. A query vector is then evaluated by every function in the cascade, and if classified negative, the evaluation stops

$$f_{c,j}(x) = \text{sgn}(\ f_1(x))\ \text{sgn}(\ f_2(x)) \dots , \tag{3}$$

where $f_{c,j}(x)$ is the cascade evaluation function of (1), i.e., we bias each cascade level in a way that one of the binary decisions is very confident, while the other is uncertain and propagates the data point to the next, more complex cascade level. Biasing of the functions is performed by setting the offset parameter b_j in (2).

The training data (*td*) in Figure 3b) are split into subsets, and each one is evaluated individually for support vectors in the first layer [14]. Hence, eliminating non-support vectors early from the classification, significantly accelerates SVM procedure. The scheme requires only modest communication from one layer to the next, and a satisfactory accuracy is often obtained with a single pass through the cascade. When passing through the cascade, merged support vectors are used to test data *d* for violations ε of the Karush-Kuhn-Tucker (KKT) conditions [11] (Figure 3c). Violators are then combined with the support vectors for the next iteration. The required arithmetic over feature vectors (the element-wise operands as well as SVM model parameters) is executed with two-stage pipeline (i.e. to reduce glitch propagation) processing unit (Figure 3d). Flip-flops are inserted in the pipeline to lessen the impact of active-glitching [2], and to reduce the leakage energy.

III. Experimental Results

Design simulations on the transistor level were performed at body temperature (37 ºC) on Cadence Virtuoso using industrial hardware-calibrated TSMC 65nm CMOS technology. In the classifier design, most of the circuit is idle (zero switching activities) at any clock cycle. Consequently, the leakage dominates the power consumption. To minimize the leakage, the classifier is synthesized with high-V_T devices. For minimal power consumption, the circuit operates at near-threshold (0.4 V) supply. The test dataset is based on recordings from the human neocortex and basal ganglia (Figure 3). The neural data was input to RTL simulations to obtain switching activity estimates for the design. These estimates were then annotated into the synthesis flow to obtain energy estimates for the digital spike-classification module.

To improve the data structure from the numerical point of view, the system in (1) is firstly pre-processed by reordering of the nonzero patterns for bandwidth reduction (Figure 4a). The information encoded in the spike trains is subsequently classified with RBF SVM kernel. Figure 4b) gives a three classes classification graphical illustration, where the bold lines represent decision boundaries. The SVM spike sorting performance has been summarized and benchmarked (Figure 4c) versus four different, relatively computationally-efficient methods for spike sorting: template matching, principle component analysis, Mahalanobis and Euclidean distance. The performance is quantified using the effective accuracy, i.e., total spikes classified versus spikes correctly classified (excluding spike detection). The SVM classifier consistently outperforms benchmarked methods over the entire range of SNRs tested, although it only exceeds the Euclidean distance metric by a slight margin reaching an asymptotic success rate of ~ 97 %. The estimation error varies with the number of spikes detected (Figure 5a), and it reaches -60 dB with normalized distribution at around 700 spikes over the entire dataset. The convergence period is ~0.1 s assuming a firing rate at 20 spikes/s from 3 neurons.

The number of support vectors required is partly governed by the complexity of the classification task. The kernels yield increasing levels of strength; however, the required energy for each kernel varies by orders of magnitude as illustrated in Figure 5b). As the SNR decreases more support vectors are needed in order to define a more complex decision boundary. For our dataset, the number of support vectors required is reduced within the range of 300-310 (Figure 5c).

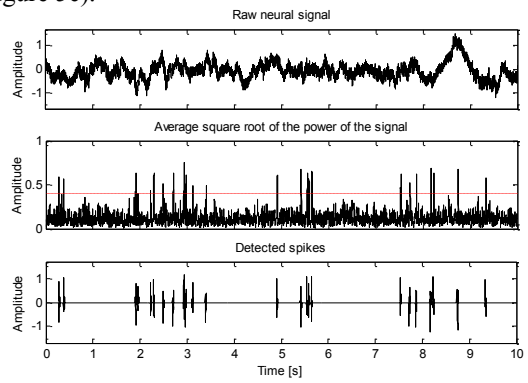

Figure 3: Spike detection from continuously acquired data, the *y* axis is arbitrary; a) top: raw signal after amplification, not corrected for gain, b) middle: threshold (line) crossings of a local energy measurement with a running window of 1ms, and c) bottom: detected spikes.

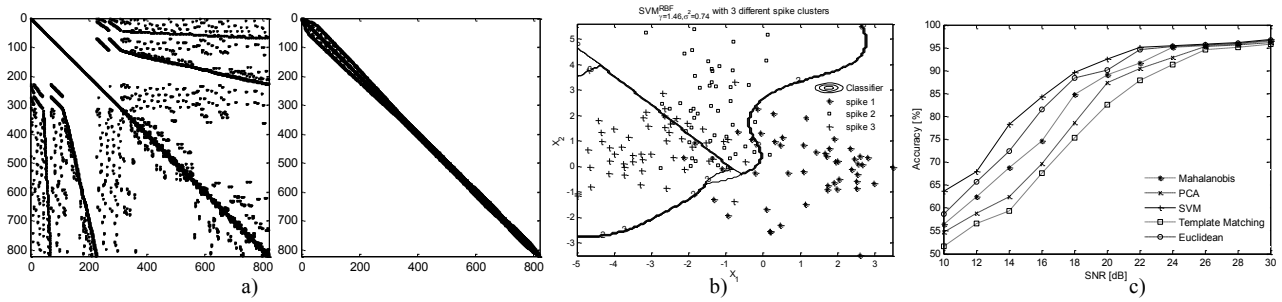

Figure 4: a) Nonzero pattern before (left) and after (right) reordering, b) the SVM separation hypersurface for the RBF kernel, c) effect of SNR on spike sorting accuracy of the BMI system.

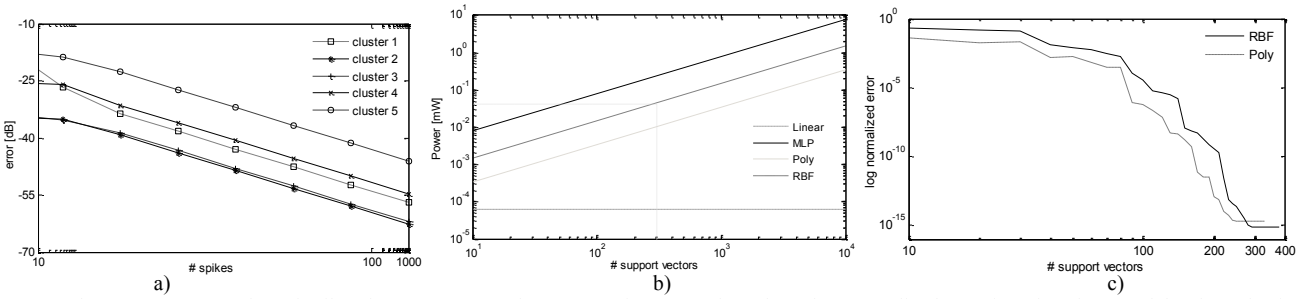

Figure 5: a) The error versus number of spikes, b) energy per cycle versus various SVM kernels, c) log normalized error in reduced set model order reduction versus number of support vectors.

The required cycle count (0.14 kcycles) and memory (0.2 kB) for linear kernel, versus (4.86 kcycles) and (6.7 kB) for RBF kernel, highlight the memory-usage dependence on the kernels. The spike detection implementation includes 31k logic gates resulting in a 2.64 mm^2 area, and consumes only 41 µW of power from a 0.4 V supply voltage. The consumed power corresponds to a temperature increase of 0.11 °C (i.e., assuming the 0.029 °C/mW model [3]), which is ~ 9 times lower than the required consumed power in a neural implants safe range (<1 °C). In Table II, we compare the state of the spike sorting systems to this work.

	[4]	*[5]*	*[6]*	***[this work]****
Technology [nm]	*65*	*90*	*65*	*65*
Programmability	*no*	*yes*	*no*	*yes*
V_{DD} [V]	*0.27*	*1*	*0.3*	*0.4*
No. of channels	*16*	*128*	*1*	*128*
Pow. Dens. [µW/mm^2]	*60.9*	*9.8*	*43.4*	*15.5*
Power [µW]	*75*	*87*	*2.17*	*41*
Area [mm^2]	*1.23*	*8.9*	*0.05*	*2.64*

TABLE I- COMPARISON WITH PRIOR ART, *-SIMULATED DATA.

IV. CONCLUSION

In this paper, we propose a programmable neural spike classifier based on multiclass kernel SVM for 128-channel spike sorting system that tracks the evolution of clusters in real-time, and offers high accuracy, has low memory requirements, and low computational complexity. The implementation results show that the spike classifier operates on-line, without compromising on required power and chip area, even in a neural interfaces with a low SNR.

REFERENCES

[1] M.A. Lebedev, M.A.L. Nicolelis, "Brain-machine interfaces: past, present and future", *Trends Neurosc.*, vol. 29, no. 9, pp. 536-546, 2006.

[2] K.H. Lee, N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals", *IEEE J. Solid-State Circ.*, vol. 48, no. 7, pp 1625-1637, 2013.

[3] S. Kim, R. Normann, R. Harrison, F. Solzbacher, "Preliminary study of the thermal impact of a microelectrode array implanted in the brain", *Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 2986-2989, 2006.

[4] V. Karkare, S. Gibson, D. Marković, "A 75-µW, 16-channel neural spike-sorting processor with unsupervised clustering", *IEEE J. Solid-State Circ.*, vol. 48, no. 9, pp. 2230-2238, 2013.

[5] T.-C. Ma, T.-C. Chen, L.-G. Chen, "Design and implementation of a low power spike detection processor for 128-channel spike sorting microsystem", *IEEE Int. Conf. Acous., Speech Sig. Proc.,* pp. 3889-3892, 2014.

[6] Z. Jiang, Q. Wang, M. Seok, "A low power unsupervised spike sorting accelerator insensitive to clustering initialization in sub-optimal feature space", *IEEE Des. Autom. Conf.*, pp. 1-6, 2015.

[7] K.H. Kim, S.J. Kim, "A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio", *IEEE Trans. Biomed. Eng.*, vol. 50, no. 8, pp. 999-1011, 2003.

[8] T. Chen, *et al.*, "NEUSORT2.0: A multiple-channel neural signal processor with systolic array buffer and channel-interleaving processing schedule", *Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 5029-5032, 2008.

[9] A. Zjajo, R. van Leuken, "Iterative learning cascaded multiclass kernel based support vector machine for neural spike data classification", *IEEE Int. Conf. Comp. Intelligence in Bioinformatics and Comp. Biology*, pp. 1-6, 2015.

[10] V. Franc, V. Hlavac, "Multi-class support vector machine", *IEEE Int. Conf. Pattern Recogn.*, pp. 236-239, 2002.

[11] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, John Wiley and Sons, 2000.

[12] [Online]. Available: http://www.physionet.org,Physionet.

[13] J. Vlach, K. Singhal, *Computer methods for circuit analysis and design*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1983.

[14] H.P. Graf, *et al.*, "Parallel support vector machines: the cascade SVM", *Adv. Neural Inf. Proc. Syst.*, pp. 521-528, 2004.