PAPER

In vivo neural spike detection with adaptive noise estimation

Journal of Neural Engineering

PAPER

In vivo neural spike detection with adaptive noise estimation

Daniel Valencia[1,2,*], Patrick P Mercier[2] and Amir Alimohammad[1]

[1] Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA, United States of America
[2] Department of Electrical and Computer Engineering, University of California San Diego, San Diego, CA, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: dlvalencia@sdsu.edu

RECEIVED
31 January 2022

REVISED
24 June 2022

ACCEPTED FOR PUBLICATION
12 July 2022

PUBLISHED
22 July 2022

Keywords: neural signal processing, application-specific integrated circuits, brain-machine interfaces

Abstract

Objective. The ability to reliably detect neural spikes from a relatively large population of neurons contaminated with noise is imperative for reliable decoding of recorded neural information. *Approach.* This article first analyzes the accuracy and feasibility of various potential spike detection techniques for *in vivo* realizations. Then an accurate and computationally-efficient spike detection module that can autonomously adapt to variations in recording channels' statistics is presented. *Main results.* The accuracy of the chosen candidate spike detection technique is evaluated using both synthetic and real neural recordings. The designed detector also offers the highest decoding performance over two animal behavioral datasets among alternative detection methods. *Significance.* The implementation results of the designed 128-channel spike detection module in a standard 180 nm CMOS process is among the most area and power-efficient spike detection ASICs and operates within the tissue-safe constraints for brain implants, while offering adaptive noise estimation.

1. Introduction

Damage to the spinal cord can disrupt the pathway of signals sent between the brain and the body and may result in partial or complete loss of both motor and sensory functions. The loss of these functions can have devastating implications on the quality of one's life. Currently, most patients living with paralysis require around-the-clock assistance to fulfill their daily tasks, which can be cost-intensive and deprive their sense of independence. Extraction of motor intent directly from the brain using brain-machine interfaces (BMIs) have shown to be promising in generating control signals for external assistive devices. However, complex brain processes are reflected by the activity of large neural populations and that the study of a few neurons provide relatively limited information [1].

Biological neurons communicate information among each other via electrical pulses, called action potentials or spikes. Conventional micro-electrode arrays (MEAs), such as Utah Array, are able to record from hundreds of electrodes [2]. Each electrode records spikes from multiple neurons close to the electrode's tip. In fact, the neural signal on a recording electrode is the cumulative electrical activity of various nearby neurons contaminated with noise, which is referred to as multi-unit activity (MUA), offering an extracellular recording. In general the background noise is a non-Gaussian random process due to various issues including electrode drift during operation, tissue-electrode interface noise, electronics noise, variation in the spike shape, the presence of overlapping spikes, and correlations between spikes and local field potentials (LFPs). Using conventional MEAs, it is not possible to precisely place electrodes to individually record from a single neuron. Extracellular electrodes detect changes in electrical potentials from a vicinity of a neuron (about 140 micrometers) where generally tens of neurons are present [3]. Researchers, however, often require single-neuron activity for the study of how neurons are correlated with each other for specific stimulus [4, 5]. Also, the algorithms employed for accurate neural decoding typically process spike trains, which represent the action potentials of individual neurons over time [6].

For detecting the spiking activity of neurons, the continuous recorded analog signal by the MEA is first amplified and then converted into a digital signal. The digitized signal is then passed through a band-pass filter typically between 300 and 3000 Hz. Frequencies below 300 Hz are filtered to remove

low-frequency activity and the upper cutoff frequency is set to diminish the noisy appearance of the spike shapes [7]. After filtering, spike detection attempts to separate the high amplitude spike signals from the low amplitude neural background noise. The background noise can be either fixed to a specific value or it can be estimated dynamically. The noise threshold is often set to a scaled version of the background noise. Detecting neural spikes is performed in two phases, the pre-processing to emphasize the spikes from the background noise and then applying the estimated threshold to the pre-emphasized signal. Spikes are interpreted as occurring when the pre-emphasized signal crosses the noise threshold. The spike detection process can thus return the spike waveform itself in addition to the time an action potential occurs.

This article focuses on the development of an accurate spike detection module toward implantable *in vivo* operation that can adapt to changes in channel statistics autonomously. The rest of this article is organized as follows. Section 2 presents the impact of various filtering methods on the performance of spike detection as well as the commonly employed signal pre-emphasis algorithms and alternative techniques for noise estimation. The computational complexity and feasibility of the pre-emphasis and noise estimation techniques for *in vivo* spike detection are discussed and compared. Section 3 quantifies the performance of various combinations of the pre-emphasis and noise estimation methods for spike detection using the widely-employed WaveClus synthetic datasets [8]. Section 4 presents the design and hardware implementation of our designed detection technique with adaptive noise estimation. Section 5 quantifies the reliability of our designed and implemented spike detection module and its application in neural decoding. Finally, section 6 makes some concluding remarks.

2. Filtering, pre-processing, and noise estimation algorithms

After the amplification and analog-to-digital conversion of the recorded neural signals, spike detection, which consists of filtering, pre-emphasis, noise estimation, and thresholding, is applied. Filtering is used to remove unwanted frequency components from the recorded neural signals. For example, low frequency LFP, with the frequency of about 250 Hz, are removed. Pre-emphasis involves processing of the filtered neural signals to discern the neural activity from ambient background noise. The ambient background noise is estimated dynamically, which is used to derive a threshold value that the pre-emphasized neural signal must exceed in order to be detected as spikes.

From the perspective of *in vivo* signal processing, low-order filters are preferable due to their lower computational complexity and memory requirements. Using Matlab's FilterDesigner toolbox, we designed five candidate band-pass causal filters, Equiripple FIR, Butterworth IIR, Chebyshev Types I and II IIR, and Elliptical IIR filters. Each of the filters had the following characteristics: sampling rate of 24 kHz, high-pass frequency of 300 Hz, low-pass frequency of 3000 Hz, 60 dB of attenuation in both stop bands, unity gain and 1 dB of ripple in the pass band, and the filter orders between 4 and 10.

Figures 1(a) and (b) illustrate the impact of causal and non-causal filtering, respectively, on the spike waveforms with various sixth-order filters. Non-causal filtering was realized using MATLAB's `filtfilt` function. It can be seen that the causal filter realizations reduce the spike amplitude and more importantly, impose a phase delay to the signal. In the case of the Cheby2 realization, the signal is severely attenuated when using a relatively low order of six. Figure 1(b) shows the benefit of utilizing non-causal filtering, which ensures zero phase shift. Also, the amplitudes and shapes of the spike waveforms are better preserved compared to the causal filtering. Figures 1(c) and (d) show the bloxplots of the Euclidean distance between actual spike waveforms and the filtered spikes employing causal and non-causal filtering, respectively. It is shown that the Equiripple FIR filter provides the least Euclidean distance among the various filter realizations.

In this work we consider two of the most commonly employed pre-emphasis methods, non-linear energy operator (NEO) [9], and absolute value (ABS) [8]. Energy-based signal pre-emphasis methods such as NEO accentuate the spikes by computing the energy of the signal. NEO is given as

$$\psi[n] = x[n]^2 - x[n-1] \times x[n+1], \tag{1}$$

where $\psi[n]$ denotes the energy of the signal $x[n]$, and is commonly employed as it amplifies spikes from background noise. The ABS pre-emphasis method is given as

$$\hat{x}[n] = |x[n]|, \tag{2}$$

where $\hat{x}[n]$ denotes the ABS of $x[n]$. Depending on the type of pre-emphasis method, it may induce additional phase delay, and in some cases, no pre-emphasis is applied to the filtered neural signal. Figures 2(a) and (b) show the impact of causal and non-causal filtering, respectively, on the NEO pre-emphasized signal with various sixth order filter realizations. Figures 2(c) and (d) show the boxplots of the Euclidean distance between the actual and filtered energy waveforms utilizing causal and non-causal filtering, respectively. Note that in the energy domain, the causal filters perform similarly while the non-causal FIR filter retains the highest amplitude.

While calculating the ABS and the energy are trivial, estimation of the threshold is not. The scaling factor is often between 1.5 and 4 and is chosen

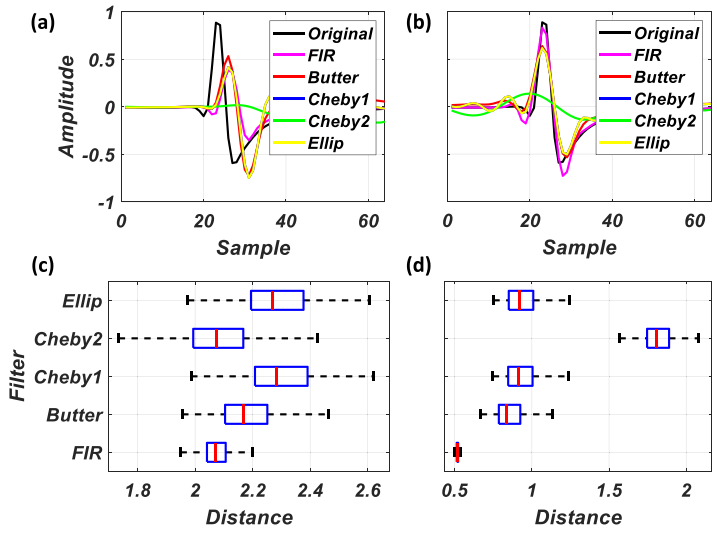

Figure 1. The impact of (a) causal filtering and (b) non-causal filtering on the spike waveform shapes. The boxplots (c) and (d) show the variations of the Euclidean distance between actual spike waveforms and the filtered spikes employing causal and non-causal filtering, respectively.

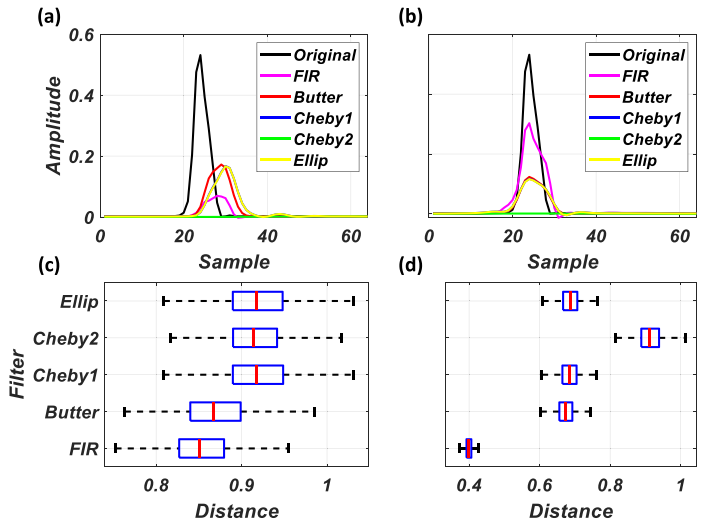

Figure 2. The impact of (a) causal filtering and (b) non-causal filtering on the spike energy waveform shapes. The boxplots (c) and (d) show the variations of the Euclidean distance between actual and the filtered energy waveforms employing causal and non-causal filtering, respectively.

empirically. In some applications, reducing the scaling factor to allow more noisy waveforms to be considered as spikes has resulted in a greater performance [10]. We used a constant scaling factor of 4, and the scaling factor was not tuned to optimize the performance of individual methods for a fair comparison among detection schemes. Thus, any process following the spike detection will interpret threshold crossings as if they were genuinely occurring spikes.

As threshold crossings are interpreted as spikes, the selection of the threshold is a crucial decision in the detection process for two main reasons: (i) the threshold should be set sufficiently high such that noisy or erratic behavior in the neural signal is not interpreted as genuine spiking activity; and (ii) the threshold should be accurate enough so that the threshold crossings behave similar to genuine spike trains.

In this article, we consider four alternative noise estimation methods, Median (MAD) [8], root-mean square (RMS) [11], Ada-BandFlt (ABF) [12], and AdaFlt (AF-128) [12]. The Median (MAD) is a commonly employed algorithm for estimating the noise and is given as:

$$\sigma_e = \text{med}\left(\frac{|x|}{0.6745}\right),$$

where med denotes the median of the signal and 0.6745 denotes the 75th percentile of the standard

Table 1. The computational complexity of various noise estimation methods over a one second window.

Method	Adds./Mults. per window	Comparisons per window	Update frequency	Ops/s.
MAD	1	f_s^2	1 Hz	f_s^2+4
RMS	f_s-1	0	1 Hz	$6f_s+2$
ABF	$f_s/500$	$3f_s^2/10e3$	4 Hz	$(3f_s^2+60f_s)/2.5e3-14$
AF-128	$f_s/500$	$1.28f_s^2/10e3$	100 Hz	$(1.28f_s^2+60f_s)/100-580$

normal distribution [8]. The median of the ABS of the neural signal reduces the interference of spikes based on the assumption that spikes seldom appear in the signal. The median is often computed over a relatively long recording, on the order of one minute. However, storing relatively long neural recordings is infeasible for *in vivo* BMIs, in which the area- and energy dissipation constraints severely limit the amount of memory storage and the algorithms that can be realized.

An alternative noise estimation methods is the RMS [11], where the estimated noise is given as:

$$\sigma_e = \sqrt{\frac{1}{M}\sum_{i=0}^{M}[x_i^2]}, \tag{3}$$

and M denotes the number of samples of the band-pass filtered and pre-emphasized signal x. When estimating the noise using the RMS, it is a common practice to perform summation over one second. The spike threshold is typically considered as a scaled version of the estimated noise, such as $2\sigma_e$ or $4\sigma_e$. In practice, the square root operation can be avoided by squaring the pre-emphasized neural signal when compared to the estimated noise as $\hat{x}[n]^2 > \sigma_e^2$, where $\hat{x}[n]$ denotes the pre-emphasized neural signal.

The BandFlt algorithm [12] is similar to the RMS algorithm, but rather than a single window of one second, it utilizes 300 ten millisecond windows. The RMS is computed for each window and then sorted in ascending order. An improved version of the BandFlt algorithm is the Ada-BandFlt, where the input signal is split into 10 ms sample windows and the RMS is computed for each window [12]. The RMS values are then collected for 100 windows and an initial noise estimation is set as the 25th percentile of the distribution of RMS values of the 100 windows. After this initial noise estimation, the estimated noise is updated every following 250 ms as:

$$\sigma_e(n) = 0.8\times\sigma_e(n-1) + 0.2\times M_{0.25},$$

where $M_{0.25}$ denotes the 25th percentile of the RMS values.

The AdaFlt algorithm [12] works on 128 ten millisecond windows. The maximum and minimum values of each window are found and the 40th percentile of the maximum and minimum values are computed, and each of these are taken as initial estimates for positive and negative thresholds, respectively. Following the initial estimation, AdaFlt calculates one set of maximum and minimum values once for every ten millisecond windows. After 128 sets of maximum and minimum values are calculated, the noise estimates are given as:

$$\sigma_{p,e}(n) = 0.9\times\sigma_{p,e}(n-1) + 0.1\times M_{0.4},$$

$$\sigma_{n,e}(n) = 0.9\times\sigma_{n,e}(n-1) + 0.1\times m_{0.4},$$

where $M_{0.4}$ and $m_{0.4}$ denote the 40th percentile of maximum and minimum values, respectively.

Table 1 gives the computational complexity of four different noise estimation methods over a one second window. For algorithms that require sorting to compute the median, we assumed that sorting M values requires an average of M^2 comparisons [13]. To normalize the computational complexity of different operations, we estimated the complexity of addition and multiplication as 2 and 4 times the complexity of a comparison [14]. Note that while the RMS requires the most number of addition and multiplication operations, the lack of sorting significantly reduces the required number of operations per second. For example, for a sampling rate of $f_s = 10$ kHz, the MAD, RMS, ABF, and AF-128 algorithms require approximately 100e6, 60e3, 120e3, and 1.3e6 operations per second, respectively. However, we can measure the performance of the noise estimation method as the accuracy of the threshold crossings. However, the input to the noise estimation is the pre-emphasized signal. Thus, there are various combinations of signal pre-emphasis and noise estimation methods and we will evaluate their performance in the next section.

3. Performance of *in vivo* potential spike detection techniques

To evaluate the performance of the candidate pre-emphasis and noise thresholding methods, we use the widely employed WaveClus datasets, which consists of 20 simulated neural recordings with three single-units in each recording. There are four difficulty levels: Easy1, Easy2, Difficult1, and Difficult2, which refers to the similarity of the three single-unit spike waveforms present in the recordings. The simulated

recordings have varying levels of noise with a standard deviation between 0.05 and 0.20 (up to 0.40 for Easy1) relative to the amplitude of the spikes. The recordings are first simulated at 96 kHz and then downsampled to 24 kHz. These datasets have been widely used as they also contain annotated spike timings and the classes of each spike, referred to as ground truth information.

The performance of the candidate spike detection methods can thus be given by comparing it is interpreted detected spike times to those present in the ground truth dataset. Using the ground truth information, we can evaluate whether a spike detected at time t_i is a genuine spike (i.e. a true positive *TP*), or an errant/noise spike (i.e. a false positive *FP*). Also, ground truth spikes that are missed by the detector can be denoted as missed spikes (i.e. a false negative *FN*). For our analysis, a spike is taken as a true positive if it is detected within two milliseconds of the ground truth spike time. Some commonly employed spike detection measures involve computing the probability of detection P_d, the probability of missed spikes P_m, and the probability of false alarms P_{fa}, which are defined as $P_d = \frac{TP}{TP+FN+FP}$, $P_{fa} = \frac{FP}{TP+FP}$, and $P_m = \frac{FN}{TP+FN}$, respectively. Some literature also uses the sensitivity metric $\frac{TP}{TP+FP}$, which quantifies the ratio of valid spikes given all detected spikes. A sensitivity value close to one represents that a large portion of the detected spikes are genuine. However, a high sensitivity value in conjunction with a high number of false positives may induce an artificially high sensitivity value. Another measure is the F-Score metric, defined as $F = \frac{TP}{TP+0.5(FN+FP)}$, which is a combination of the recall and precision of detection. For a fair comparison with the state-of-the-art work, the F-Score is our chosen metric in this article. For the first analysis, we employed a relatively-low order a fourth order Elliptical IIR filter is employed, following conventional low-order filtering methods in BMI systems [15, 16]. If alternative filters have relatively similar frequency and phase characteristics, then it follows that the performance differences among the alternative detection methods is primarily related to the employed detection methods. Table 2 gives the mean F-scores of the candidate detection schemes over the four WaveClus datasets. It is apparent that the combination of the NEO with the RMS or the ABF outperforms other candidate methods. It can also be seen that the NEO with RMS offers the lowest standard deviation across the datasets, offering a robust performance compared to the alternative methods. While the ABF noise estimation method performs well, it is requirement of sorting 100 RMS values makes it infeasible for efficient, real-time *in vivo* realization.

For the NEO and RMS-based detection, we also quantify the impact of alternative filters on spike detection performance by computing the F-Scores over the WaveClus datasets with ground-truth information. Figures 3(a)–(d) show the performance of the NEO and RMS-based spike detection using different filters over the Easy1, Easy2, Difficult1, and Difficult2 datasets, respectively. It is shown that the FIR, Butterworth, Chebyshev Type I, and Elliptical filters outperform the Chebyshev Type II filter. Interestingly, all filters other than Chebyshev Type II do not necessarily see improved performance for increased filter order. While we found that non-causal filtering is preferred for retaining waveform shapes, such details are not relevant for applications that only require spike timings, such as BMIs that employ MUA threshold crossings rather than single-unit action potentials. While the NEO pre-emphasis reduces the amplitude of the spike waveforms, as shown in figure 2, it was also shown that causal filtering further reduces the amplitude of the spike waveforms. Figures 4(a)–(d) show the F-scores of employing fourth order causal and non-causal Equiripple FIR filters for various noise levels over the Easy1, Easy2, Difficult1, and Difficult2 datasets, respectively. It can be seen that the difference between the two filtering approaches is insignificant. The appeal of the low-order causal filtering is due to their relatively low computational complexity and memory requirements. Additionally, from the hardware implementation perspective, the causal realization is applicable for real-time processing. Therefore, for the BMI applications where the integrity of waveform shapes is not a concert and only threshold crossing rates are required, using a relatively low-order causal FIR filter is adequare.

Note that the F-score metric measures the ratio of true spike detections over all detected spikes and is dependent on the application for which a value can be deemed acceptable. For example, in the case of spike sorting, acceptable values are 0.8 or higher, i.e. the majority of detected spikes are genuine. However, for neural decoding, it has been shown that voltage threshold settings can be tuned to optimize the encoding of movements' kinematics [10]. Thus, detecting more false positives as spikes and hence, a smaller F-score, may yield a better decoding.

Table 2. The mean F-scores of the candidate detection methods over the WaveClus datasets.

Detection method	Mean F-score	Standard deviation
NEO + MAD	0.45	0.012
NEO + RMS	0.92	0.001
NEO + ABF	0.94	0.002
ABS + MAD	0.91	0.034
ABS + RMS	0.89	0.104
ABS + ABF	0.91	0.075

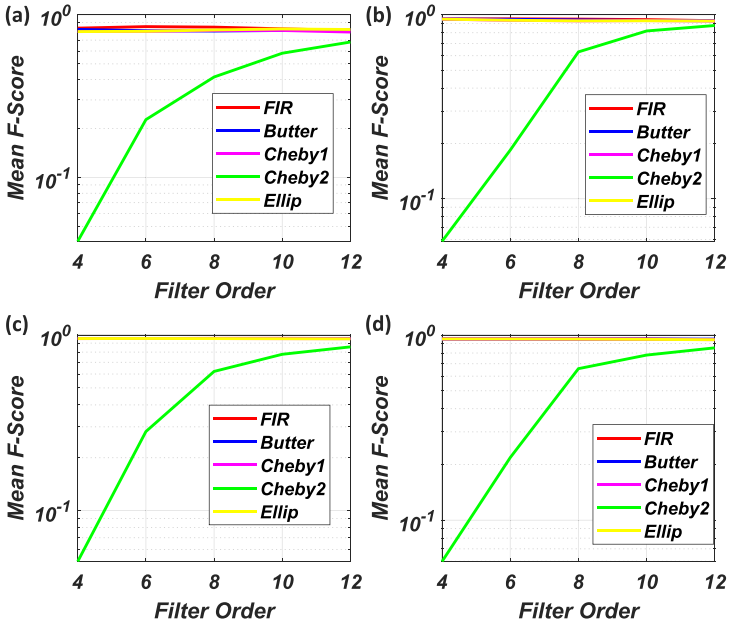

Figure 3. The F-scores of the NEO and RMS detection with various filters for the (a) Easy1, (b) Easy2, (c) Difficult1, and (d) Difficult2 datasets, respectively.

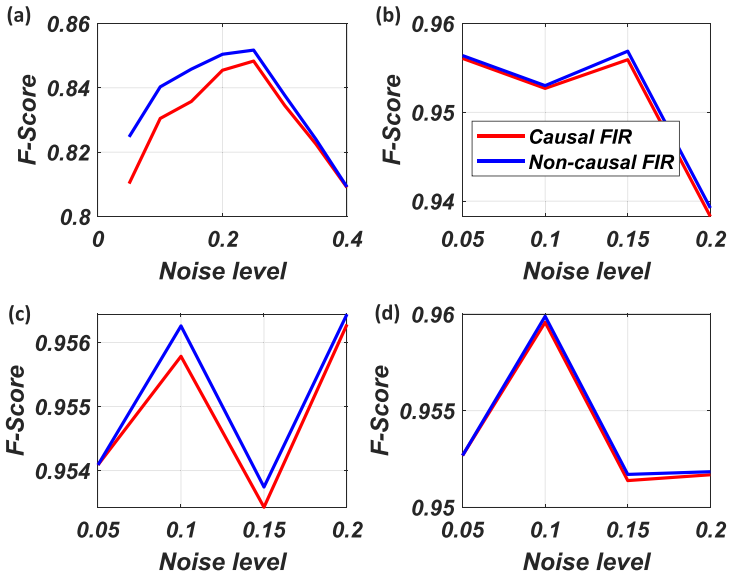

Figure 4. The F-Scores of the fourth-order causal and non-causal Equiripple FIR filter over the (a) Easy1, (b) Easy2, (c) Difficult1, and (d) Difficult2 datasets.

4. Hardware realization of the candidate detector

Based on the discussed findings, we suggest that using the NEO pre-emphasis with the RMS thresholding provides the optimal detection performance.

In vivo spike detection must work with limited computational resources that are shared among hundreds to thousands of channels. For efficient hardware realization of the RMS thresholding, equation (3) is simplified to avoid the division and square root operations. The division is replaced with an arithmetic shift operation and the square root is avoided by squaring the thresholding inequality $x[n]_{NEO} > \sigma_e$. The simplified noise estimation is then given as:

$$\sigma_e = \sum^{M}(x^2_{NEO}) \gg \left(\log_2\left(\frac{M}{C^2}\right)\right), \qquad (4)$$

where M denotes the number of samples to accumulate before computing the estimated noise σ_e, C denotes the noise scalar, and $x \gg y$ denotes the

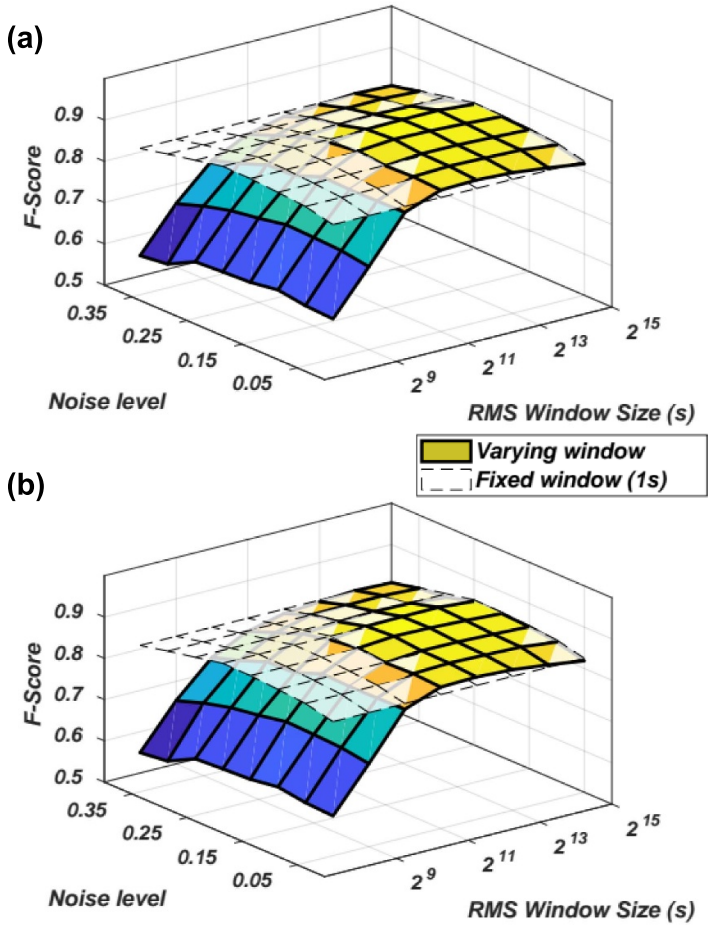

Figure 5. The F-Score of the NEO and RMS-based spike detector for the (a) Easy1 and (b) Difficult1 datasets over various noise levels and varying RMS window sizes using the simplified equation (4).

arithmetic right shift of x by y bit positions. To employ the right shift operation, the values of M and C are constrained as powers of two. To quantify the impact of the applied simplification, we computed the F-Scores for the Easy1 and Difficult1 datasets over values of M ranging between 8 and 15, with the scalar $C = 4$. As shown in figures 5(a) and (b), the performance difference between the one second RMS window and the simplified equation (4) is negligible for values of $M \geqslant 13$. In some cases, the performance of the simplified computation outperforms that of the fixed window. To demonstrate the noise adaptation over a real dataset, a single channel of data from the 'indy_20 170 131_02' dataset is passed to the NEO and RMS-spike detector using the Xilinx Vivado simulator. Figure 6 shows the continuous update of the RMS noise threshold values in response to the signal's variations. For example, when the signal increases in amplitude, the noise threshold is increased to avoid detecting larger noise values as spikes. The noise threshold, shown as a sample-and-hold line, indicates that the RMS changes occur in fixed intervals (i.e. over 2^{13} samples). For clarity, a smoothed version of the noise adaptation using an eight-sample moving average filter is also shown.

The datapath of the designed multi-channel NEO and RMS-based spike detector is shown in figure 7. The datapath consists of two main components: the squaring NEO unit and the RMS estimator. The squaring NEO unit has an input signal shift register *CBuf*, radix-2 booth-encoded multipliers *B* to compute the NEO value, followed by a fixed-point multiplier to compute the square of the NEO signal. The depth of the shift register *CBuf* is equal to $2N$, where N denotes the number of supported channels. Rather than reusing the datapath by increasing its operating frequency, we instead choose to realize multiple datapath units that each process a relatively low number of channels. One caveat, however, is that implementing the squared NEO would be costly, especially when instantiated several times across datapath instances. We thus implement two of the three NEO multiplications with sequential radix-2 booth-encoded multipliers [17], as shown in figure 8. The RMS Estimator consists of a set of N registers for accumulating the squared NEO values of each channel, as well as a set of N latches for storing the RMS value once every M input samples, as per equation (4). A control unit *CTRL* manages when the booth-encoded multipliers, accumulator

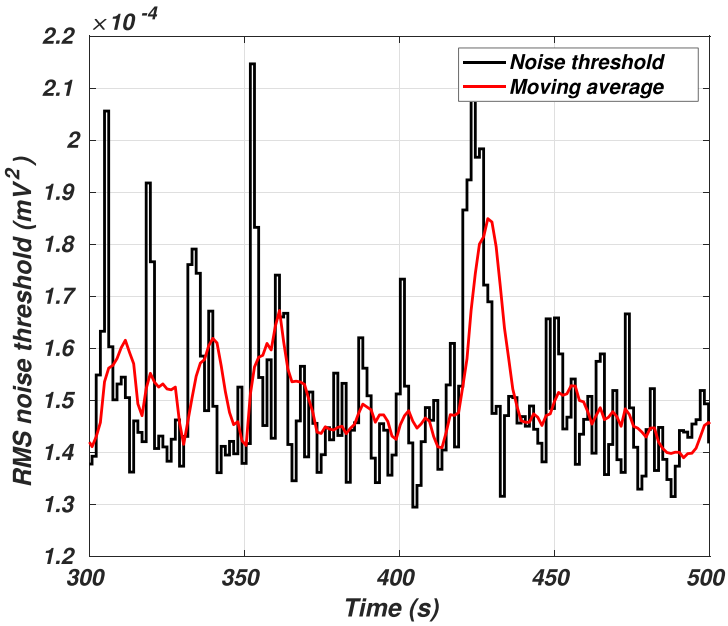

Figure 6. Adaptive noise threshold estimation.

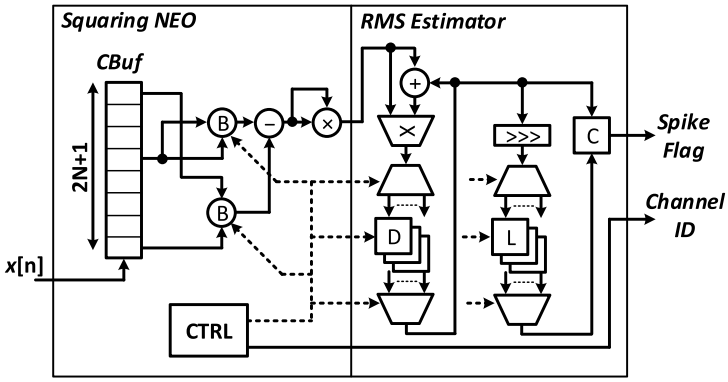

Figure 7. The datapath of the multi-channel NEO and RMS-based spike detector.

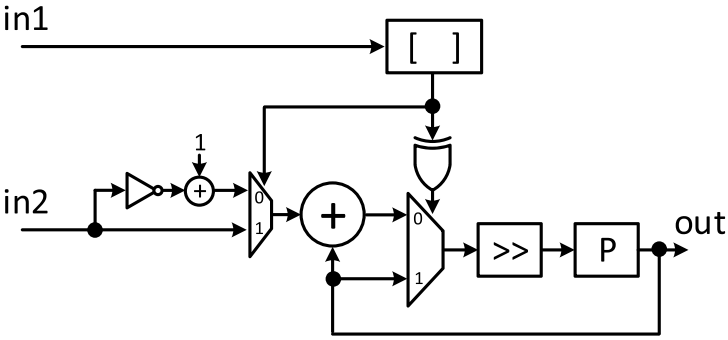

Figure 8. The datapath of the sequential radix-2 booth multiplier.

registers, and RMS value latches are enabled for writing.

We have designed and implemented the NEO and RMS-based spike detector in a standard 180 nm CMOS process with 16 datapaths processing 8 channels each, for a total of 128 channels. The ASIC layout, shown in figure 9, is estimated to consume 639 μW of power from a 1.8 V supply when operating at 800 KHz and is estimated to occupy 3.44 mm^2 of silicon area. The implemented spike detector thus

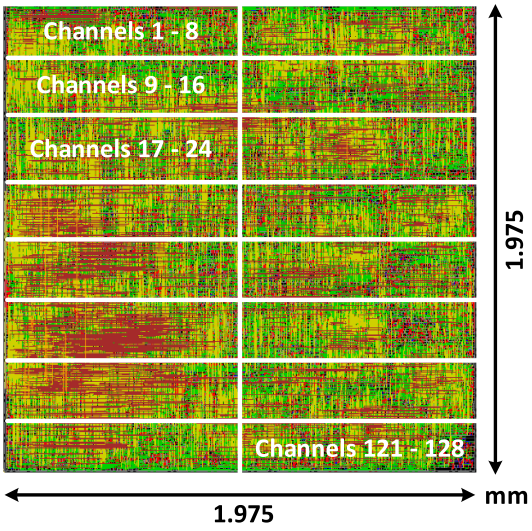

Figure 9. The ASIC layout of the designed 128-channel NEO and RMS-based spike detector.

consumes 4.9 μW per channel and occupies 0.02 mm^2 of silicon area per channel. Note that the designed and implemented NEO and RMS-based spike detector supports arbitrary window sizes of 2^B while the threshold is updated more frequently for smaller window sizes. Synthesis was performed with Synopsys Design Compiler and the place and route was performed with Cadence Innovus. After routing, the netlist is simulated to obtain a realistic switching activity for estimating the dynamic power consumption. The NEO and RMS-based spike detector has a power density of 18.58 mW cm^{-2}, which is within the tissue safe constraints (i.e. 40 mW cm^{-2}) [18]. We have also implemented an alternate version of the NEO and RMS spike detector in the same 180 nm CMOS process by replacing all multiplications with approximate log-based multipliers [19]. As opposed to the serial Booth-encoded multipliers, the log-based multipliers do not require an increased operation frequency and can operate at only 80 KHz, which significantly reduces the power consumption. The approximated NEO and RMS-based spike detector only consumes 0.64 μW of power per channel from a 1.8 V supply and occupies 0.02 mm^2 of silicon area. The power density of the approximated design is only 3.07 mW cm^{-2}, which is well within the tissue safe constraints.

Various hardware realizations of neural spike detection have been reported recently [20–22]. Table 3 lists the characteristics and implementation results of various spike detection ASICs. For a fair comparison, the implementation results have been scaled to a 180 nm CMOS process with a 1.8 V supply voltage, as described in [24]. In [20], the authors present a 16-channel BMI with a digital implementation of window discriminator-based spike detection. Window discriminators involve two threshold values and detect a spike when an action potential crosses an upper and lower threshold, which correspond to the de-polarization and re-polarization of the spike waveform, respectively [25]. Unfortunately, the two threshold values are not adaptive to changes in the signal during run-time, and are programmed through a communication interface. In [21], the authors present an analog implementation of a NEO-based spike detector. The threshold is considered as the peak value of the NEO pre-emphasized signal, and it adapts to the signal if new values of the NEO signal exceed the current peak value. In [22], the authors present a 64-channel neural signal acquisition system-on-chip (SoC). Spikes are detected using a NEO-based pre-emphasis and a fixed threshold that is uploaded to the SoC, and unfortunately cannot adapt to real-time changes in channel statistics.

It can be seen in table 3 that our designs are among the most power- and area-efficient spike detection circuits, while supporting adaptive noise estimation. Compared to the design in [21], which also employs the NEO pre-emphasis, our digital design naturally consumes more power, however, our design offers a slightly lower probability of detecting false positive spikes over the same WaveClus datasets. It can be seen in table 3 that the NEO and RMS detector utilizing approximated multipliers dissipates less power than the detector using booth-encoded multipliers, and also consumes the least energy per channel among the state of the art spike detection ASICs. While power is a commonly employed metric for comparing ASICs, energy consumption is a vital metric for *in-vivo* BMIs, as it directly impacts the battery life for implantable circuits. For example, an *in-vivo* detector based on booth-encoded multipliers operating at 800 KHz, employing the SAFT LS14250 battery with a nominal capacity of 1200 mAh, would operate for approximately 433 $\times$ 10^3 h, while the approximate-based NEO and RMS detector would operate for approximately 3.37 $\times$ 10^6 h. The designs in [21–23] would operate for approximately 1.44 $\times$ 10^6 h, 463 $\times$ 10^3, and 192 $\times$ 10^3–287 $\times$ 10^3 h, respectively.

5. Spike detection performance analysis

In a BMI system, the output of the spike detection is used by the subsequent neural signal processing modules. For example, spike sorting groups individual neurons' spikes into individual clusters [25]. Additionally, the dynamics of groups of neurons and behavioral data, such as motor movement [26], can be explored. Neural decoding translates spiking information into a quantifiable representation, such as movement kinematics for a robotic prosthesis [27]. We assess the performance of the candidate spike detection methods in a neural decoding task using trial-based behavioral recordings without ground truth information.

Table 3. The characteristics and implementation results of various spike detection ASICs.

Work	Ours Booth	Ours Approx.	[20]	[21]	[22]	[23]
Technology (nm)	180	180	180	180	65	130
Implementation	Digital	Digital	Analog	Analog	Analog	Digital
Clock frequency (KHz)	800	80	—	24	20	800
Supply voltage (V)	1.8	1.8	—	1.8	0.8	1.2
Channels	128	128	16	1	64	64
Power per Ch. (μW)	4.9	0.64	4	1.5	1.21	3.04–4.54
Area per Ch. (mm^2)	0.02	0.02	—	0.03	0.0105	—
Pre-emphasis	NEO	NEO	None	NEO	NEO	None
Thresholding	RMS	RMS	Fixed	Peak	Fixed	Dual-Mean
Adaptive	Y	Y	N	Y	N	Y
Scaled power per Ch. (μW)[a]	4.9	0.64	4	1.5	4.67	7.53–11.26
Scaled area per Ch.[a]	0.02	0.02	—	0.03	0.05	—
Scaled energy per Ch. (pJ)[a,b]	6.24	0.8	—	1.87	5.83	9.4–14

[a] Scaled to a 180 nm process with a 1.8 V supply voltage, as described in [24].

[b] Normalized to a clock frequency of 800 kHz.

The advantage of synthetic datasets is that we can compute the F-Score as a quantitative measure for assessing the performance of the detection schemes due to the known times of action potentials, referred to as 'ground truth' information. A real neural recording usually does not offer ground truth information and hence, it is subjective when comparing the performance among different spike detection methods. One approach to verify the spikes of real recordings is to use intra-cellular recordings, which senses the voltage inside the neuron itself [28]. When simultaneous intra- and extra-cellular recordings are available, the spike times detected in the extracellular recording can thus be verified using the intracellular recording [29]. Unfortunately, intracellular recording is challenging due to the lack of stability and long-term reliability of the recordings [30]. One useful method when employing high-density MEAs is to exploit the spatio-temporal correlations of spikes detected at different recording sites [31]. However, this is not applicable for a low-density clinical recording interface. Another method that can be used to quantify the performance of a spike detector is to employ a widely accepted toolset, such as WaveClus [8], as a 'gold standard' [15]. The detected spike times from such tools could be used as a reference label to evaluate the probability of detection, missed spikes, and false alarms of alternative detection methods. One caveat, however, is that due to the nature of neural recordings, these tools cannot offer guaranteed true labeling of data since users need to set different parameters for different neural datasets. Due to the use of MUA spike trains for BMI applications, another method to quantify the performance of various spike detection methods is to model the separability and distinctness of spike trains for different source stimuli. For example, neurons in the motor cortex modulate their firing rates in response to specific movement direction [32]. Thus, a spike detection that can produce spike trains with activity more accurately correlated with the intended stimulus/action is preferable for BMI applications.

5.1. Spike detection performance analysis over real neural recordings

A spike train can be considered as a stochastic point process that consists of binary events in time. The conditional intensity function (CIF) $\lambda(t|H_{t-})$ of a point process provides a complete probability model of the process and describes the instantaneous firing probability conditioned on its firing history. Following the approach in [33], we can compute the CIF for a channel of neural data over various repeated trials of a specific stimulus S_i. Then, we can compute the probability $P(N(t)|S_i)$ of observing a particular set of spike times $N(t)$ given the stimulus S_i. For this analysis, we use the raw broadband data recorded from an adult male Rhesus macaque monkey in the Sabes Lab at the University of California, San Francisco [34]. The monkey was trained to perform self-paced reaches by controlling the position of a cursor on a screen. Data was recorded from area M1 of the primary motor cortex using a chronically implanted 96-channel Utah Array. Spikes were detected using the candidate spike detection methods described above. Using the relative starting and ending positions of the cursor target, we discretized the monkey's reach into four possible directions: left and upwards, left and downwards, right and upwards, and right and downwards. The detected spike times were also discretized as follows. The spike times were converted into spike trains $N(t)$ into K bins of width $\Delta = TK^{-1}$, where T denotes the total time of each reach, in our case aligned to the final 500 ms before the monkey finished its reach. The spiking activity during a reach is thus given as $N_{1:k} = [\Delta N_1, \ldots, \Delta N_k]$, where ΔN_k is one if there is a spike in the time interval $((k-1)\Delta, k\Delta)$ and zero otherwise. The discretized CIF $\lambda(k\Delta|N_{1:k-1})$ is given

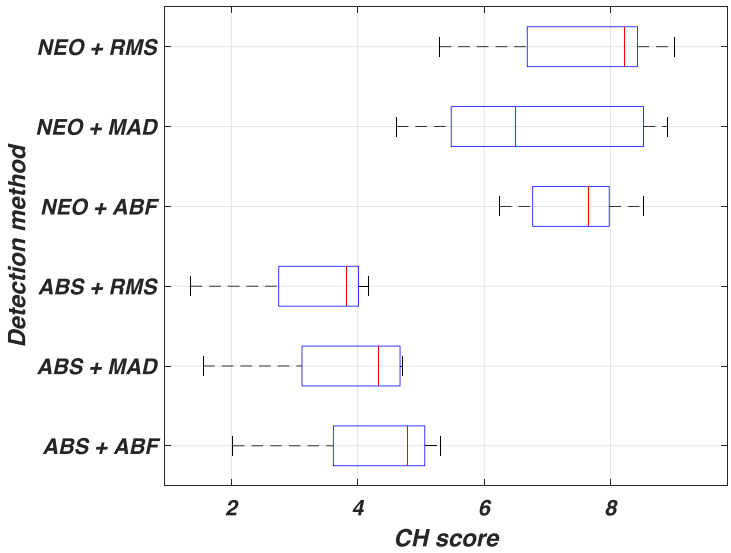

Figure 10. The CH score boxplots of the projected spike trains using various spike detection methods for the macaque monkey's reaching tasks.

as the mean of the spike history $N_{1:k-1}$ divided by Δ. The probability of the observed spike train $N(t)$ conditioned on the presented stimulus S_i can be written as:

$$P(N(t)|S_i) = \exp\left[\sum_{k=0}^{K}\log(\lambda(k\Delta|N_{1:k-1})\Delta)\Delta N_k - \sum_{k=0}^{K}\lambda(k\Delta|N_{1:k-1})\Delta\right].$$

After computing the mean CIF for all four reach types, the probability of each trial conditioned on each of the four mean CIFs, i.e. $P(N(t)|S_1)$, $P(N(t)|S_2)$, $P(N(t)|S_3)$, and $P(N(t)|S_4)$ are computed, where S_1–S_4 denotes reach types left and upwards, left and downwards, right and upwards, and right and downwards, respectively. We then compute a log likelihood feature vector given as $Y_{N(t)} = \log[P(N(t)|S_1), \ldots, P(N(t)|S_4)]$. $Y_{N(t)}$ can thus be considered as a projection of the spike train $N(t)$ onto the likelihood space of the four reach types. Thus, for each reach, there is a feature vector that represents a point in the likelihood space. A more useful spike detection yields a well-defined set of clusters for each type of reach. To quantify the resulting likelihood space clusters of projected spike trains, we employ the Calinski–Harabasz (CH) metric [35], which is defined as the ratio of the distances among clusters and the distances within a cluster. For the spike train analysis, we employed six spike detection methods using five 'indy' datasets, 20 161 220_02, 20 170 123_02, 20 170 124_01, 20 170 127_03, 20 170 131_02. The spike trains were partitioned into sets associated with the respective reach type for a particular movement. Figure 10 shows the CH score boxplots of the projected spike trains generated by the candidate detection methods over the five datasets. It can be seen that the NEO-based methods outperform the ABS-based methods. Also, one can note that the combination of the NEO with RMS yields a higher median CH score. While the NEO and ABF method has a smaller variability, the computational complexity of the RMS is approximately half of the ABF's complexity, as given in table 1. This implies that the projected spike trains generated by the NEO and RMS detection produce better defined clusters, which may be preferable for neural decoding.

5.2. The impact of spike detection on neural decoding applications

To study the effect of alternative spike detection methods on the performance of neural decoding, we utilize the publicly available hc-2 dataset [36]. The dataset consists of the neural recordings from the CA1 region of the hippocampus of freely moving rodents over various experiments. The rodents were provided with water or food as a reward at random locations throughout a platform. The positions of the rodent was determined using video footage by tracking the position of LEDs on the rodents' heads. The aim of decoding is to predict the location of the rodent based on the spiking activity of neurons in the hippocampus. The tip of each recording shank, which are the channels with the highest signal amplitude, were used for detecting spikes using the candidate methods. We designed and trained a gated recurrent unit (GRU)-based recurrent neural network (RNN) decoder to map the binned spike counts onto the rodent's positions. The training data contained 7772 samples, each representing 1.92 s of data over 75 bins (i.e. bin size = 25.6 ms). The training data was partitioned into 80% for training, 10% for validation, and 10% for testing. The validation data is used to evaluate the

Table 4. The validation and testing performance of the GRU-based RNN decoder using various potential *in vivo* spike detection methods.

Detection method	Validation (R^2) score	Testing (R^2) score
ABS + MAD	0.73	0.75
ABS + RMS	0.79	0.76
NEO + MAD	0.73	0.74
NEO + RMS	0.85	0.86

performance of the model on data not observed during training, while the testing data is used to evaluate the performance of the model after training. The RNN was trained using the Tensorflow framework for up to 250 epochs, using early stopping on the R^2 metric to avoid over-fitting to the training data. The R^2 metric, also known as the coefficient of determination, quantifies the amount of variance in the dataset that can be account for by the model, with a score of 1.0 being perfect. Table 4 gives the testing performance of the GRU-based RNN over various spike detection methods. One can see that the combination of NEO and RMS offers the highest performance for both the validation and testing sets.

6. Conclusion

This article investigated the efficiency of various potential spike detection techniques for *in vivo* implantation. It was found that the NEO-based preprocessing in combination with the RMS noise estimation outperforms other state-of-the-art spike detection methods. It was shown that the combination of NEO with RMS resulted in the highest F-Score when evaluated on the commonly employed Wave-Clus datasets. The design and implementation of a 128-channel NEO and RMS-based spike detector in a standard 180 nm CMOS process was presented. The synthesized NEO and RMS spike detector was estimated to occupy 0.2 mm^2 of silicon area and consume 4.9 μW of power from a 1.8 V supply while operating at 800 KHz. It was shown that our design is among the most power, area, and energy-efficient spike detectors. The designed and implemented spike detector was also employed for neural decoding and it was found that the NEO and RMS-based detection also offers the highest decoding performance across two different animal behavioral datasets.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

This work was supported by the Center for Neurotechnology (CNT), a National Science Foundation (NSF) Engineering Research Center (EEC-1028725) and the NSF Award #2007131. The authors would like to thank Jose Berroteran for developing software implementations of the various noise estimation algorithms.

ORCID iD

Daniel Valencia https://orcid.org/0000-0003-4539-8166

References

[1] Ohiorhenuan I E, Mechler F, Purpura K P, Schmid A M, Hu Q and Victor J D 2010 Sparse coding and high-order correlations in fine-scale cortical networks *Nature* **466** 617–21
[2] Maynard E M, Nordhausen C T and Normann R A 1997 The Utah intracortical electrode array: a recording structure for potential brain–computer interfaces *Electroencephalogr. Clin. Neurophysiol.* **102** 228–39
[3] Buzsáki G 2004 Large-scale recording of neuronal ensembles *Nat. Neurosci.* **7** 446–51
[4] Mukamel R and Fried I 2012 Human intracranial recordings and cognitive neuroscience *Annu. Rev. Psychol.* **63** 511–37
[5] Boraud T, Bezard E, Bioulac B and Gross C E 2002 From single extracellular unit recording in experimental and human Parkinsonism to the development of a functional concept of the role played by the basal ganglia in motor control *Progr. Neurobiol.* **66** 265–83
[6] Tam W-K, Wu T, Zhao Q, Keefer E and Yang Z 2019 Human motor decoding from neural signals: a review *BMC Biomed. Eng.* **1** 1–22
[7] Quiroga R Q 2009 What is the real shape of extracellular spikes? *J. Neurosci. Methods* **177** 194–8
[8] Quiroga R Q, Nadasdy Z and Ben-Shaul Y 2004 Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering *Neural Comput.* **16** 1661–87
[9] Kaiser J F 1990 On a simple algorithm to calculate the 'energy' of a signal *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing* pp 381–4
[10] Oby E R, Perel S, Sadtler P T, Ruff D A, Mischel J L, Montez D F, Cohen M R, Batista A P and Chase S M 2016 Extracellular voltage threshold settings can be tuned for optimal encoding of movement and stimulus parameters *J. Neural Eng.* **13** 036009
[11] Guillory K and Normann R 1999 A 100-channel system for real time detection and storage of extracellular spike waveforms *J. Neurosci. Methods* **91** 21–29
[12] Biffi E, Ghezzi D, Pedrocchi A and Ferrigno G 2010 Development and validation of a spike detection and classification algorithm aimed at implementation on hardware devices *Comput. Intell. Neurosci.* **2010** 1–15
[13] Al-Kharabsheh K S, AlTurani I M, AlTurani A M I and Zanoon N I 2013 Review on sorting algorithms a comparative study *Int. J. Comput. Sci. Secur.* **7** 120–6 (available at: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.3357&rep=rep1&type=pdf)
[14] Mora-Mora H, Mora-Pascual J, García-Chamizo J M and Jimeno-Morenilla A 2006 Real-time arithmetic unit *Real-Time Syst.* **34** 53–79
[15] Zhang Z and Constandinou T 2021 Adaptive spike detection and hardware optimization towards autonomous, high-channel-count BMIs *J. Neurosci. Methods* **354** 109103
[16] Willett F R, Avansino D T, Hochberg L R, Henderson J M and Shenoy K V 2021 High-performance brain-to-text communication via handwriting *Nature* **593** 249–54
[17] Koren I 2018 *Computer Arithmetic Algorithms* (Natick, MA: AK Peters/CRC Press)

[18] Wolf P D and Reichert W 2008 Thermal considerations for the design of an implanted cortical brain–machine interface (BMI) *Indwelling Neural Implants: Strategies for Contending with the In Vivo Environment* (Boca Raton, FL: CRC Press) pp 33–38
[19] Kim M S, Del Barrio A A D, Oliveira L T, Hermida R and Bagherzadeh N 2019 Efficient mitchell's approximate log multipliers for convolutional neural networks *IEEE Trans. Comput.* **68** 660–75
[20] Liu X, Zhang M, Richardson A G, Lucas T H and Van der Spiegel J 2017 Design of a closed-loop, bidirectional brain machine interface system with energy efficient neural feature extraction and PID control *IEEE Trans. Biomed. Circuits Syst.* **11** 729–42
[21] Koutsos E, Paraskevopoulou S E and Constandinou T G 2013 A 1.5 μw NEO-based spike detector with adaptive-threshold for calibration-free multichannel neural interfaces *IEEE Int. Symp. on Circuits and Systems* pp 1922–5
[22] Biederman W, Yeager D J, Narevsky N, Leverett J, Neely R, Carmena J M, Alon E and Rabaey J M 2015 A 4.78 mm^2 fully-integrated neuromodulation SoC combining 64 acquisition channels with digital compression and simultaneous dual stimulation *IEEE J. Solid-State Circuits* **50** 1038–47
[23] Delgado-Restituto M, Rodriguez-Perez A, Darie A, Soto-Sánchez C, Fernández-Jover E and Rodriguez-Vazquez A 2017 System-level design of a 64-channel low power neural spike recording sensor *IEEE Trans. Biomed. Circuits Syst.* **11** 420–33
[24] Stillmaker A, Xiao Z and Baas B 2011 Toward mode accurate scaling estimates of cmos circuits from 180nm to 22nm *Technical Report* ECE-VCL-2011-4 vol 4 (Davis, CA: VLSI Compuitation Lab, ECE Department, University of California) p m8
[25] Lewicki M S 1998 A review of methods for spike sorting: the detection and classification of neural action potentials *Netw., Comput. Neural Syst.* **9** R53–R78
[26] Trautmann E M *et al* 2019 Accurate estimation of neural population dynamics without spike sorting *Neuron* **103** 292–308
[27] Chestek C A *et al* 2011 Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex *J. Neural Eng.* **8** 045005
[28] Taketani M and Baudry M 2010 *Advances in Network Electrophysiology* (Berlin: Springer)
[29] Hamilton F, Berry T and Sauer T 2018 Tracking intracellular dynamics through extracellular measurements *PLoS One* **13** e0205031
[30] Long M A and Lee A K 2012 Intracellular recording in behaving animals *Curr. Opin. Neurobiol.* **22** 34–44
[31] Pachitariu M, Steinmetz N A, Kadir S N, Carandini M and Harris K D 2016 Fast and accurate spike sorting of high-channel count probes with kilosort *Advances in Neural Information Processing Systems* vol 29 pp 4448–56
[32] Teka W W, Hamade K C, Barnett W H, Kim T, Markin S N, Rybak I A and Molkov Y I 2017 From the motor cortex to the movement and back again *PLoS One* **12** e0179288
[33] Salimpour Y, Soltanian-Zadeh H, Salehi S, Emadi N and Abouzari M 2011 Neuronal spike train analysis in likelihood space *PLoS One* **6** e21256
[34] O'Doherty J E, Cardoso M M B, Makin J G and Sabes P N 2017 Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology (https://doi.org/10.5281/zenodo.583331)
[35] Caliński T and Harabasz J 1974 A dendrite method for cluster analysis *Commun. Stat. Theory Methods* **3** 1–27
[36] Mizuseki K, Sirota A, Pastalkova E and Buzsáki G 2009 Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop *Neuron* **64** 267–80