[Sarah Gibson, Jack W. Judy, and Dejan Marković]

©ISTOCKPHOTO.COM/GUIDO VROLA

Spike Sorting

[The first step in decoding the brain]

Extracellular recording, the technique of inserting an electrode into the extracellular tissue of the brain to record the activity of individual neurons ("single-unit activity"), is a common experimental method used by neuroscientists to study how the brain works. In recent years, researchers have also demonstrated its potential use in medical technologies for the treatment of disorders such as paralysis, epilepsy, and memory loss. Although most of these applications require single-unit activity, these electrodes record the activity from multiple neurons surrounding the electrode. Spike sorting is the process of separating this signal into single-unit activity. A number of algorithms for this purpose have been published over the years, but there is still no universally accepted solution. In this article, we will present an overview of the spike-sorting problem, its current solutions, and the challenges that remain. Because of the increasing demand for chronically implanted spike-sorting hardware, we will also discuss implementation considerations.

Digital Object Identifier 10.1109/MSP.2011.941880
Date of publication: 8 December 2011

1053-5888/12/$31.00©2012IEEE

INTRODUCTION

For centuries, scientists have been using electrophysiology to study the electrical properties of biological cells and tissues. In 1791, Luigi Galvani discovered that he could induce contraction in a frog leg muscle by applying an electric current [1]. In 1952, Hodgkin and Huxley, using an experimental technique they developed called the "voltage clamp," made a number of groundbreaking discoveries on the movement of ions across the membranes of nerve cells during action potential generation for which they eventually received a Nobel Prize [2]–[5]. In 1977, Hubel and Wiesel (also Nobel Prize laureates) used electrophysiological recordings to provide the first information about how the activity of individual neurons contribute to higher visual processing [6].

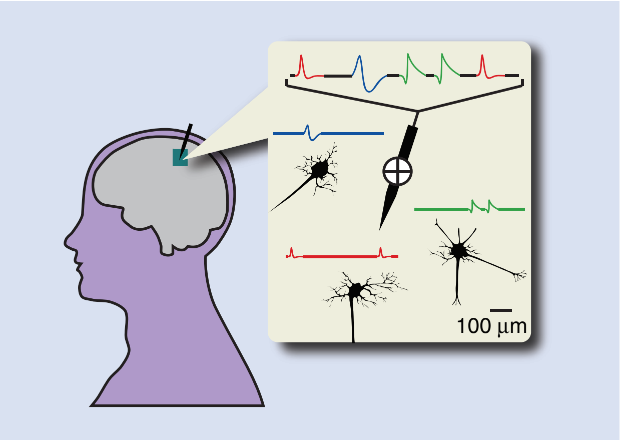
WHETHER THE APPLICATION IS BASIC SCIENCE RESEARCH OR MEDICAL TECHNOLOGY, THE SIGNALS FROM INDIVIDUAL NEURONS ("SINGLE-UNIT ACTIVITY") ARE OFTEN OF PARTICULAR INTEREST.

Electrophysiological recordings can be made from within cells (intracellular) or from outside cells (extracellular). In studies of the central nervous system, small-diameter electrodes can be positioned in the extracellular space to record electrical signals from surrounding neurons (Figure 1). These electrodes are able to detect action potentials from individual neurons. The ability of extracellular recording to provide researchers with neuron-level activity combined with its relatively low level of difficulty to perform (as compared to intracellular recording, for example) have led extracellular recording to become the dominant experimental technique in many studies. For example, there has been a movement in neuroscience research to study not only individual neurons but networks of neurons to understand how the activity of interconnected neurons results in higher-order functions such as perception, understanding, movement, and memory. Such studies require extracellular recording from ensembles of neurons using multichannel electrode arrays. In *Methods for Neural Ensemble Recordings*, contributing authors Sameshima and Baccalá go so far as to claim that "extracellular recordings are the only practical choice in experiments that intend to establish correlations between neural ensemble responses and behaviors involving awake animals" [7].

Electrophysiology is also used in clinical settings. For example, in patients with severe pharmacologically intractable epilepsy who require surgical resection of the affected brain tissue, electrophysiological recordings from depth electrodes placed inside the brain are used to localize brain areas where seizures begin. These larger electrodes mainly record electroencephalogram (EEG) signals, but often microelectrodes are implanted as well for use in research (e.g., [8]–[10]), since single-unit activity provides greater detail on changes in signal transmission that could distinguish normal from abnormal activity. And over the past decade, the technique of extracellular recording has received additional attention as researchers have begun to tap into its potential use in medical technologies for the treatment of disorders such as paralysis [11], [12], epilepsy [11], and even cognitive and memory loss [13].

Whether the application is basic science research or medical technology, the signals from individual neurons ("single-unit activity") are often of particular interest. In basic science, for example, the researcher may require knowledge of single-unit activity to study how a type of neuron responds to a specific stimulus. Similarly, most neural prosthetic technologies employ some sort of "decoding" algorithm—which may decode movement [11], [12], [14], intentions [15], or memories [13]—that typically operates on signals from individual neurons. But because of the sizes of recording electrodes, the recorded signal is the sum of the signals from several (two to ten) neurons surrounding the electrode ("multiunit activity," illustrated in Figure 1). Microwire electrode tips used in extracellular recording typically have diameters of 13–80 μm [16]. The Cyberkinetics implementation of the popular Utah silicon microelectrode array has conical electrode tips with radii of 3–5 μm and lengths 35–75 μm. In such cases, spike sorting, the process of separating multiunit activity into groups of single-unit activity, is necessary.

Beyond the functional role that spike sorting serves, spike sorting is important in providing the data reduction required of on-chip, multichannel processors. Recent advances in data-acquisition technology allow for the recording of hundreds of channels simultaneously, but a higher number of channels leads to a higher data rate. Harrison presents the example of a 100-channel system using a sampling rate of 30 kSa/s and a resolution of 10 b, which would produce data at 30 Mb/s [17]. Wireless transmission at this data rate cannot be achieved under the strict power limits to which implantable electronics are subject. The present solution is to transfer data from the subject to a computer via thick cables. In the research setting, cables restrict the physical movement of subjects, thereby limiting the quality and diversity of experiments that can be



[FIG1] **The electrical signal recorded from a microelectrode is the sum of the postsynaptic and action-potential activity of many neurons in the surrounding area.**

performed. Furthermore, these cables can increase the severity of noise and motion artifacts seen in the recording. Performing spike sorting in implanted hardware such that only spike IDs are retained, as opposed to the entire raw data, would achieve enough data reduction to enable the wireless transmission of data, thereby eliminating the need for cables altogether.

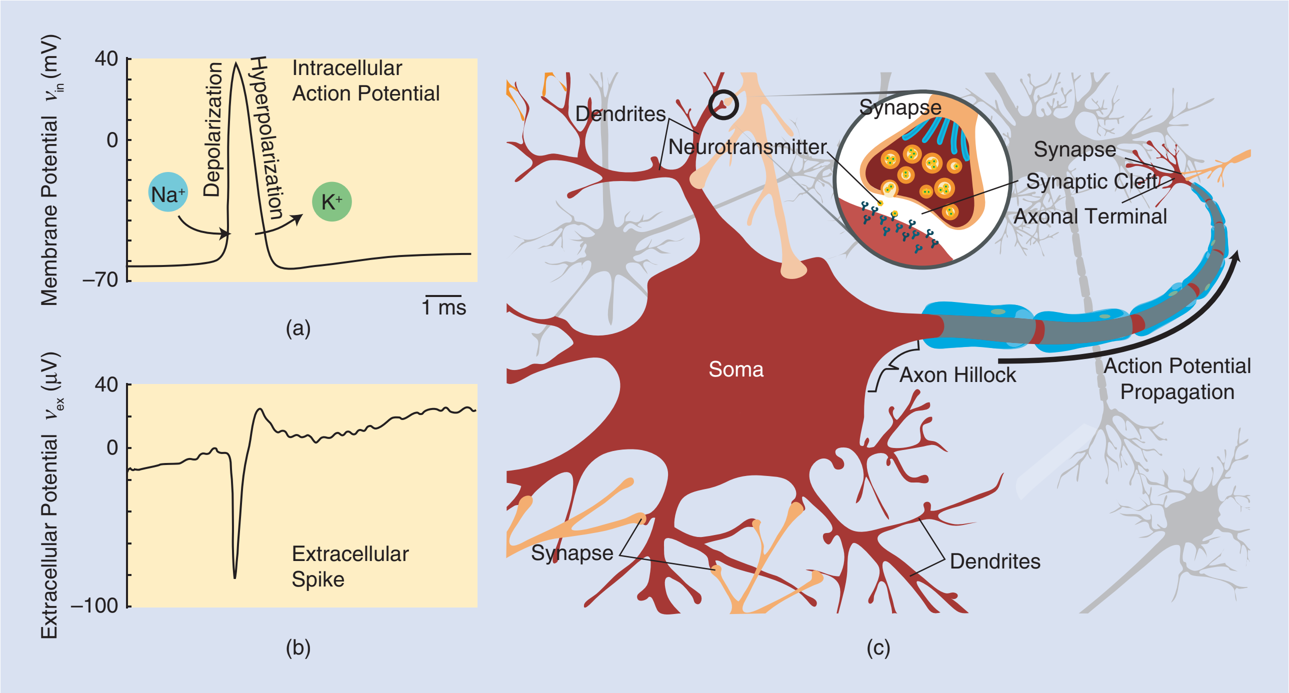
SIGNAL COMPOSITION

Let us begin by looking at the composition of the signals that are recorded by extracellular electrodes. These signals are usually prelowpass-filtered in the analog domain and then sampled at a rate of 20–30 kHz. Different cellular mechanisms are responsible for different frequency components of the recorded signals. The high-frequency content (about 300–6,000 Hz) is referred to as unit activity, while the low-frequency signal content (below about 600 Hz) is referred to as local field potential.

UNIT ACTIVITY

If unit activity is the signal of interest, as it is in this article, then the sampled signal is bandpass-filtered with a low cutoff frequency of 100–300 Hz and a high cutoff frequency of 3,000–10,000 Hz. As indicated by its name, the source of this "unit activity" is action potentials from individual neurons.

On some level, the action potential can be thought of as the discrete, binary event by which neurons communicate. Much of what we know about the mechanism was discovered in the seminal works of Hodgkin and Huxley [2]–[5]. Neuronal membranes are impermeable to charged ions except at sites of ligand- and voltage-gated channels, which allow the passage of charged ions between the intra- and extra-cellular space. When the ion channels are closed or inactive, the concentrations of potassium (K^+) and chloride (Cl^-) ions inside the cell are high relative to oustide the cell, while the concentration of sodium (Na^+) ions is high outside the cell relative to inside the cell. At rest, the cell membrane potential, defined with respect to the inside of the cell, is about -70 mV. When a cell's membrane is depolarized, e.g., by excitatory synaptic input, this depolarization "activates" (opens) the Na^+ channels, causing Na^+ ions to rush into the cell along the concentration gradient. This influx of Na^+ causes the membrane to become even more depolarized, consequently causing more Na^+ channels to become activated. Eventually the membrane potential reaches threshold, at which point external input is no longer needed to depolarize the cell, and the positive feedback caused by the Na^+ current continues the depolarization at an even faster rate. This sharp influx of Na^+ into the cell results in the rising phase of the action potential shown in Figure 2(a). Once the cell reaches a peak depolarization of about 40 mV, two things happen: the Na^+ channels become "inactivated" such that no more Na^+ions can pass through, and the voltage-gated K^+ channels open. Now, K^+ ions flow out of the cell along the concentration gradient, and the cell membrane begins to "hyperpolarize"; this efflux of K^+ results in the falling phase of the action potential [Figure 2(a)]. This hyperpolarization continues until the cell has returned to its resting potential. In some cases, the hyperpolarization is



[FIG2] Parts (a) and (b) are adapted from [18] (used with permission). (a) Illustration of the change in membrane potential during a typical action potential. (b) Illustration of the corresponding change in potential as seen by an extracellular electrode. Note the difference in vertical-axis scales. (c) Diagram of a neuron. Action potentials begin at the axon hillock and propagate down the axon. Depolarization of the axon terminal then triggers the release of neurotransmitters into the synaptic cleft, in turn depolarizing the postsynaptic cell.

followed by a slow after-hyperpolarization, where the resting potential is overshot, before the membrane potential returns to rest. The action potential usually begins at the axon hillock, near the cell body (soma), and propagates down the axon [Figure 2(b)]. Depolarization of the axon terminal then triggers the release of neurotransmitters into the synaptic cleft (the gap between the presynaptic cell's axon terminal and the postsynaptic cell's dendrite), in turn depolarizing the postsynaptic cell. It is in this way that neurons communicate with each other.

Extracellularly recorded action potentials are called *spikes*. (The terms *action potential* and *spike* are sometimes used interchangeably; to be precise, however, we will use *action potential* to refer to the intracellular event and *spike* to refer to the captured extracellular waveform.) As shown in Figure 2(b), a spike looks slightly different from an intracellular action potential. First, because the recording electrode is placed outside of the cell rather than inside the cell, the polarity is reversed. Second, the filtering properties of the extracellular medium result in an extracellular signal that is about two to three orders of magnitude smaller than the corresponding intracellular signal (~10 to 100 μV compared to ~10 mV). Third, because the membrane acts like a resistor and capacitor in parallel, that is, a highpass filter [Figure 3(a)], the recorded extracellular potential is approximately equal to the derivative of the intracellular potential [19].

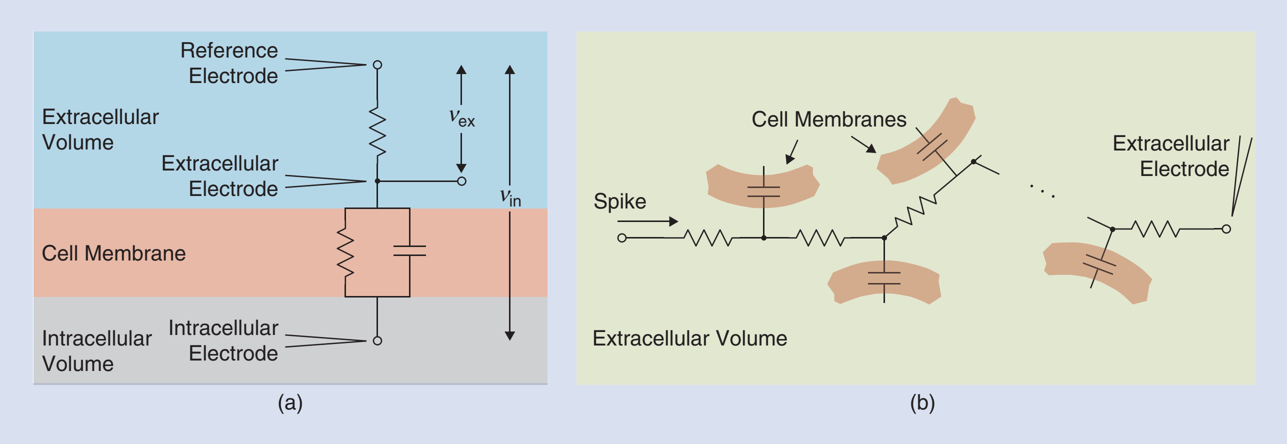
The shape of the intracellular action potential depends on a number of cell properties, including the cell type, the cell geometry, and the ion-channel distribution. This shape is generally considered to be constant for a given neuron, except in special cases such as burst (high-frequency) firing. Since the extracellular waveform is directly related to the intracellular waveform, the extracellular spike shape also depends on these properties, as well as on the position of the recording electrode relative to the cell, on the distance of the electrode from the cell, and on interference from other nearby neurons (background noise). This biological noise is the largest source of noise in a neural recording, having amplitudes approaching that of the unit activity. But the recording hardware itself, including the electrode, the amplifier, and the ADC, also adds a significant amount of noise, the scale of which is largely dependent on the given circuit implementation. It is usually assumed that the signal and the noise are statistically independent and that they sum linearly. Thus, in a given recording session where the electrode placement is assumed to be constant relative to the tissue, we assume that the extracellular spike shape for each neuron can be modeled as a deterministic waveform plus random noise. Note that while the recording noise usually is Gaussian, the background noise typically is not [21].

Spike trains can be treated as point processes with arrival times following a Poisson distribution. A neuron's firing rate, the frequency at which it generates action potentials, depends on the cell type and brain area. Neurons in the visual cortex, for example, which are either silent or firing at a base frequency of around five spikes per second (or simply Hz), respond to their preferred stimulus with firing rates of about 15–75 Hz [22]. A bursting neuron, on the other hand, can fire as many as 300–800 spikes per second [23].

LOCAL FIELD POTENTIALS

While local field potentials (LFPs) are not the focus of this article, a brief discussion of their properties is warranted here. We will also refer to these signals again later when we discuss alternative methods of decoding neural signals for brain–machine interfaces (BMIs).

A comprehensive description of the physiological basis for LFP was provided by Buzsáki and Traub in [24]. To summarize, LFPs come from several sources, the most significant of



[FIG3] **(a) A simple electrical-circuit model of extracellular recording [20]. Assume that an intracellular electrode is placed inside the cell, an extracellular electrode is placed outside but near the cell, and the reference electrode is placed very far away from the cell. The extracellular material can be modeled as pure resistance, while the cell membrane can be modeled as a resistor and a capacitor in series. The cell membrane highpass-filters the action potential signal, such that the extracellulary recorded signal (v_{ex}) is approximately equal to the derivative of the intracellularly recorded signal (v_{in}) [19]. (b) As the broadband signal passes through the extracellular medium, the capacitive membranes of nearby cells attenuate its high-frequency components. (Part (a) used with permission from Rockefeller University Press.)**

which is synaptic activity. Because the capacitive lipid membranes of cells in the brain act as a lowpass filter, the high-frequency components of neuronal signals are greatly attenuated as they travel through the extracellular medium; equivalently, slow signals are able to propagate much farther than are high-frequency signals [Figure 3(b)]. As a result, the low-frequency component of the signal recorded at any given point within the brain is a linear sum of the activity from large populations of cells. Thus, LFP can be interpreted as an indication of the "cooperative actions" of neurons.

Note that this signal is referred to as "LFP" when recorded by a microelectrode inserted into the brain (hence the "local" in LFP) but as "EEG" if recorded using scalp electrodes and as "electrocorticography (ECoG)" if recorded using epidural or subdural grid electrodes. Some scientists favor using ECoG and EEG because these methods are less invasive and easier to perform. However, because LFPs must propagate through a capacitive medium on their way to these recording sites, EEG and ECoG are actually "spatially smoothed" versions of the LFP. As such, EEG and ECoG contain very little information about the activity of the neurons that actually generate the signals. Furthermore, scalp and dural grid electrodes are only sensitive to signals originating in the superficial layers of the cortex; contributions to the signal from neurons in deeper layers of the cortex and from subcortical areas are virtually negligible. Thus, extracellular recording, which provides the experimenter with LFP measurements as well as unit activity, is the experimental technique most capable of providing information about the cooperative actions of neurons at high temporal and spatial resolutions.

SPIKE SORTING

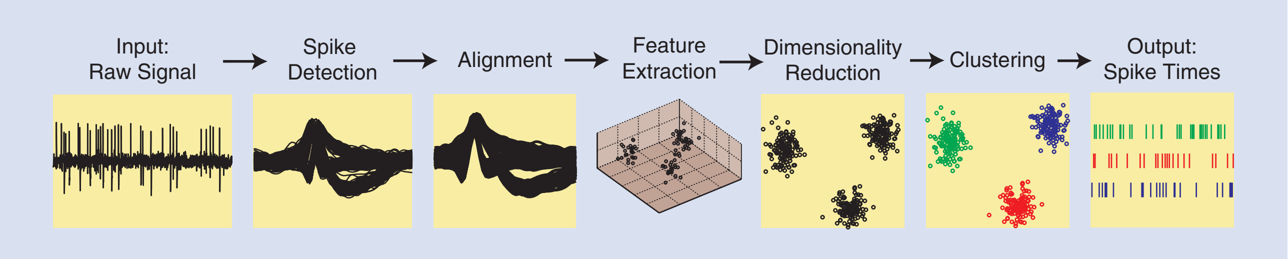
To obtain (multi)unit activity, the extracellular data is first bandpass filtered to remove the LFP and high-frequency noise (as described in the section "Unit Activity"). To then obtain single-unit activity, we must perform spike sorting by sending this raw data into the signal-processing chain shown in Figure 4. The first steps are spike detection, the process of separating spikes from background noise, and alignment, the process of aligning all detected spikes to a common temporal point relative to the spike waveform. Once the spikes have been identified, spike sorting can take place.

Most spike sorting methods—relying heavily on the previously mentioned assumption that each neuron produces a different, distinct shape (as seen by the electrode) that remains constant throughout a recording session—are based on spike waveform information. Thus, the first step in such methods is feature extraction, in which spikes are transformed into a certain set of features, such as principal components, that emphasize the differences between spikes from different neurons as well as the differences between spikes and noise. After feature extraction, some form of dimensionality reduction typically takes place, in which feature coefficients that best separate spikes are identified and stored for subsequent processing while the rest are discarded. Finally, spikes are classified into different groups, corresponding to different neurons, based on the extracted feature coefficients; this process is referred to as clustering. The result, the signal of interest to the experimenter and to BMIs, is the train of spike times for each neuron. This information can be represented graphically by a raster plot, where ticks are drawn to indicate spike occurrences versus time, as shown in the rightmost plot of Figure 4 for three neurons.

CLASSIFYING SPIKE-SORTING ALGORITHMS

Spike-sorting methods can be categorized according to a number of different characteristics. The first is the level of autonomy: methods can be "automatic/unsupervised" (fully autonomous) or "manual" (not at all autonomous). Automatic or unsupervised methods require no user input, while manual methods require constant supervision by an operator. Methods can also fall anywhere between these two extremes; a "semiautomatic" method is a method with both a manual stage and an automatic stage. For example, detection methods that require the manual setting of a threshold, or window-discriminator methods that require the manual definition of windows, but that then work automatically may be considered semiautomatic [25], as well as classification methods that require the user to manually reassign clusters after automatic cluster determination [26]. For neural prosthetic applications, spikes must be sorted in real time, thus precluding manual spike sorting. And because of the growing amount of data resulting from an increase in the number of simultaneously recorded channels, manual spike sorting is no longer a viable option in research settings either. Therefore, automatic methods are now usually required.

A second way to classify spike-sorting algorithms is by whether or not they are real time (also called online). The standard practice for many years has been to first record and store all the



[FIG4] The signal processing chain used to obtain single-unit activity.

SPIKE-SORTING METHODS CAN BE CATEGORIZED ACCORDING TO A NUMBER OF DIFFERENT CHARACTERISTICS.

data and then to perform spike sorting offline after the experiment. As a result, many of the spike-sorting methods that have been developed rely on access to all of the data at once. When using principal component analysis (PCA), for example, the principal components are often calculated using all of the detected spikes, and then each spike is projected onto these basis vectors before the actual classification takes place. It is increasingly common, however, for applications that require spike sorting to require that the spike sorting occur in real time. This requirement renders a number of hitherto commonly used methods inadequate. A compromise would be to modify offline algorithms to include an offline training period followed by a real-time classification period.

The third attribute by which spike-sorting methods can be categorized is adaptivity. As we will describe later, extracellular signals are not always stationary. In such cases it would be beneficial to use an algorithm that can adapt to a changing environment, as opposed to a static algorithm. There can be intermediate cases on this scale as well. For example, a static algorithm that requires a training period can be made adaptive by retraining it periodically.

Clustering algorithms specifically can also be classified as either parametric or nonparametric. Koontz et al. define a nonparametric clustering algorithm as "an algorithm for clustering multivariate data which is not based on a parametric model of an underlying probability density function. In particular, a nonparametric algorithm should identify clusters of arbitrary shape and size" [27]. In other words, any algorithm that assumes a certain structure to the data or that is biased towards a particular cluster shape, such as spherical or ellipsoidal, will be considered parametric. The underlying probability density function for neural data is not known a priori (see the section "Non-Gaussian Noise"), so nonparametric clustering algorithms are highly desirable.

Early spike-sorting algorithms were very simple, but not very accurate. In general, the more complex the method, the better the performance. This inherent tradeoff between algorithm accuracy and complexity leads to another characteristic by which to classify algorithms: the accuracy–complexity measure. As we described in the introduction, many applications require spike sorting in implantable hardware. Any implantable hardware is subject both to strict power-density constraints and to high reliability requirements. Thus, it is crucial to choose the spike-sorting methods with the optimal balance between accuracy and complexity to implement in hardware. As a step in that direction, our research group has performed such an analysis by evaluating a handful of methods using a biologically based, unbiased data set covering a wide range of SNRs [28], [29]. This work has allowed us to build a low-power (130-μW), 64-channel digital signal processor (DSP), which includes spike detection, alignment, and feature extraction [30], as well as a low-power (75-μW), 16-channel DSP, which includes spike detection and on-the-fly clustering [31], both of which are suitable for use in a real-time, implantable neural-recording system.

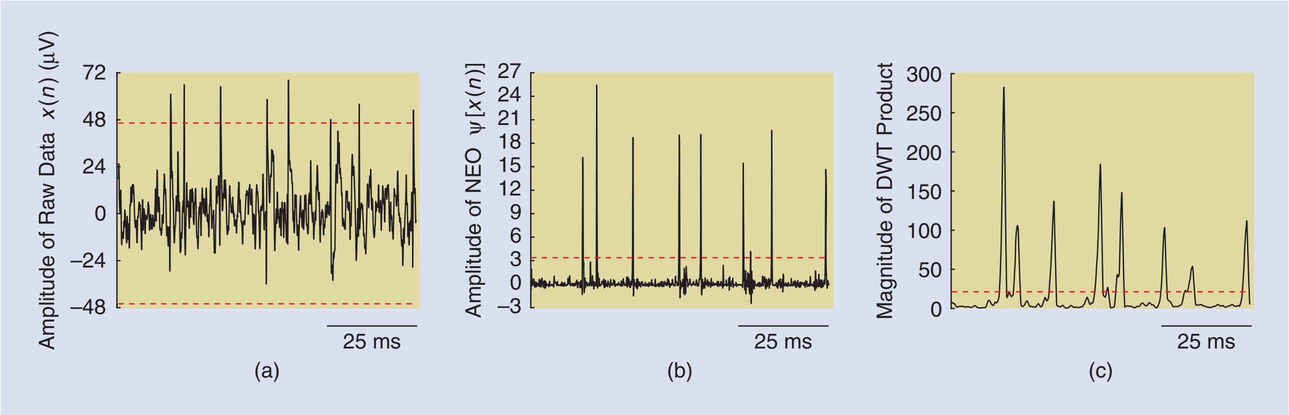
In the next section, we will give some examples of algorithms that have been used for each step of spike sorting. We will also mention some alternative methods, mostly statistical/probabalistic in nature, that do not conform to the block diagram shown in Figure 4.

OVERVIEW OF SPIKE-SORTING ALGORITHMS

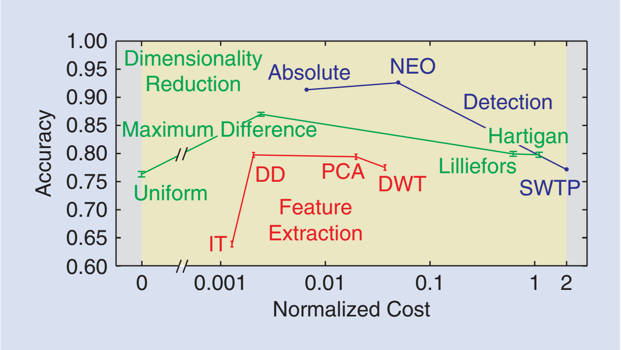
Note that Lewicki provided a nice review of spike-sorting methods in 1998 [32]. Here, we provide a relatively high-level description of the evolution of spike-sorting techniques as well as an update of more recent algorithms, and we present them in the context of hardware spike sorting.

DETECTION

Nearly all detection methods involve two main steps: the pre-emphasis of the signal and the application of a threshold. Spike-detection methods vary in how the signal is pre-emphasized and



[FIG5] Examples of pre-emphasized signals and threshold values (dashed red lines) for three different detection methods. (a) Absolute-value, (b) NEO, and (c) DWT product [34].



[FIG6] Accuracy versus normalized computational cost for each spike-detection, feature-extraction, and dimensionality-reduction algorithm tested. For spike-detection algorithms, "accuracy" represents the median choice probability. For feature-extraction and dimensionality-reduction algorithms, "accuracy" represents the mean classification accuracy after fuzzy *c*-means (the fuzzy-logic version of *k*-means) clustering, with error bars indicating the standard error of the mean. (Figure adapted from [29].)

in how the threshold is determined [36]. All of these methods run automatically given the detection threshold, so whether or not the algorithm is fully automatic depends on whether or not the threshold can be determined automatically. All of these methods are also real time (save a small delay for buffering spikes) given the detection threshold, but automatic calculation of the threshold usually involves a training period.

The early days of spike sorting came in a time before digital computers. Processing was done purely in analog hardware. As a result, spike-sorting methods were relatively primitive. Spike detection was typically performed using a simple voltage trigger or Schmitt trigger, where the voltage threshold was set manually by the user. Any time the voltage signal crossed that threshold, a pulse would be generated to indicate the presence of a spike [32]. Or, if the user needed the spike waveforms for subsequent spike sorting, a threshold crossing would trigger the capture of the spike waveform. This method is appealing because of its simplicity, and, as a result, is still used today by many experimenters. Some researchers have modified this method to include an absolute-value operation before the compare (or, equivalently, a compare to ± threshold, as shown in Figure 5), and to include automatic calculation of the threshold [33].

Another class of spike-detection algorithms is based on detecting changes in the energy of the signal. One such algorithm is called the nonlinear energy operator (NEO) or the Teager energy operator (TEO). Originally described in [35], the NEO has been proposed for use in spike detection by [36]–[38]. In discrete time, the NEO ψ is defined as

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

where $x(n)$ is a sample of the waveform at time n. The NEO is large only when the signal is both high in power (i.e., $x^2(n)$ is large) and high in frequency (i.e., $x(n)$ is large while $x(n+1)$ and $x(n-1)$ are small). Since a spike by definition is characterized by localized high frequencies and an increase in instantaneous energy [36], this method has an obvious advantage over methods that look only at an increase in signal energy or amplitude without regarding the frequency. This can be seen in Figure 5, which shows that the NEO operation increases the SNR of the signal, making detection less sensitive to the detection threshold. Another advantage of this method is that it is relatively simple to implement, whether in the digital or analog domain.

Other spike-detection algorithms are based on template matching. If the spike waveforms of interest are known a priori to the user, then matched filters can be used to correlate the incoming signal with the spike templates; if the correlation crosses a certain threshold then a spike has been detected. With known cluster templates, this method can also be used for the actual spike classification. A related method is detection using the discrete wavelet transform (DWT). The DWT, which is ideally suited for the detection of signals in noise (e.g., edge detection, speech detection), has recently also been applied to neural spike detection (see [34], [39], and [40]). This method has an intuitive appeal in that it is similar to template matching, where we correlate the signal with a known waveform, only it is scale-invariant. The DWT is also appealing because it can be implemented using a series of filter banks, keeping the complexity relatively low.

An example of one possible implementation is the DWT product [34]. First, the stationary wavelet transform (SWT) is calculated at five consecutive dyadic scales ($W(2^j, n)$, $j = 1, \ldots, 5$). Then the scale $2^{j_{\max}}$ with the largest sum of absolute values is found

$$j_{\max} = \underset{j\in\{3,4,5\}}{\operatorname{argmax}}\left(\sum_{n=1}^{N} |W(2^j, n)|\right). \tag{2}$$

From here, we calculate the point-wise product $P(n)$ (or "SWTP") between the SWT at this scale and the SWTs at the two previous scales

$$P(n) = \prod_{j=j_{\max}-2}^{j_{\max}} |W(2^j, n)|. \tag{3}$$

This product is then smoothed by convolving it with a Bartlett window $w(n)$ to eliminate spurious peaks, and a threshold is applied. The threshold Thr can be set automatically to a scaled version of the mean of this result:

$$Thr = C\frac{1}{N}\sum_{n=1}^{N} w(n) * P(n), \tag{4}$$

where N is the number of samples in the signal and C is a constant. Once again, Figure 5 shows that the pre-emphasized DWT signal has an increased SNR compared to the original signal, making detection less sensitive to the detection threshold.

In [29], we performed an analysis of accuracy versus computational complexity (logic and memory requirements) for various spike-sorting methods. The results of this analysis are reproduced in Figure 6, where the hardware cost has been normalized as follows. First, the number of operations per second (NOPS) and the area were estimated for each algorithm. These two metrics were then combined into a cost function so as to easily compare the complexities of different algorithms, as well as to provide a perspective on the complexity of individual processing steps (e.g., detection, feature extraction) relative to other processing steps. As shown in (5), first the NOPS for each algorithm was normalized by dividing by the maximum NOPS of any algorithm. Second, the same was done for area. Finally, these two numbers were summed. Thus, the cost function has a range from zero to two, zero being least costly and two most costly. This figure shows that the absolute-value ("Abs.") and NEO detection methods both have comparable accuracy and low complexity, with NEO at the knee of the curve. The results for the feature-extraction and dimensionality-reduction methods also displayed in this figure will be examined in later sections.

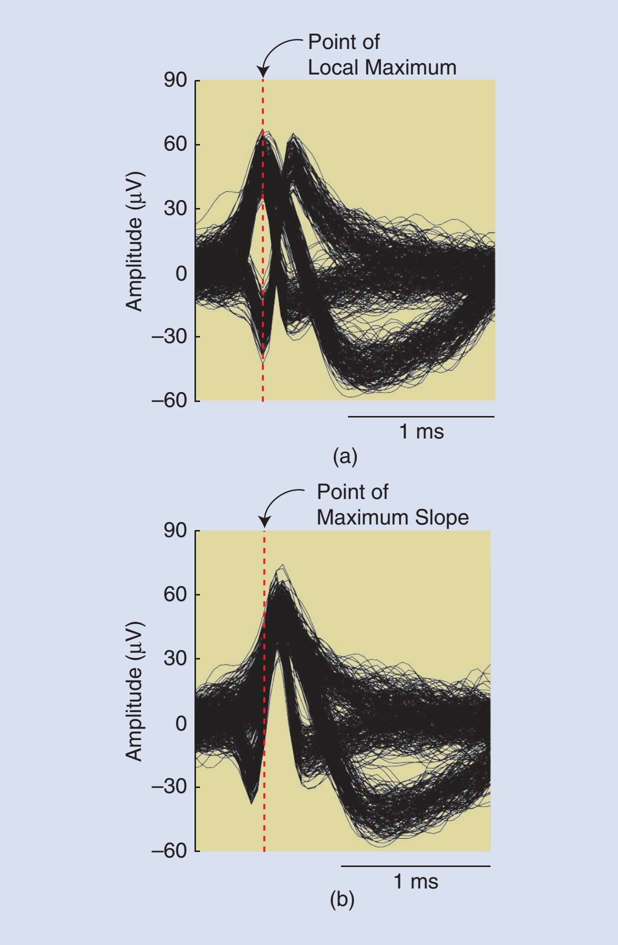
$$\text{Normalized Cost}_i = \frac{\text{NOPS}_i}{\max\limits_i \text{NOPS}} + \frac{\text{area}_i}{\max\limits_i \text{area}}. \tag{5}$$

ALIGNMENT

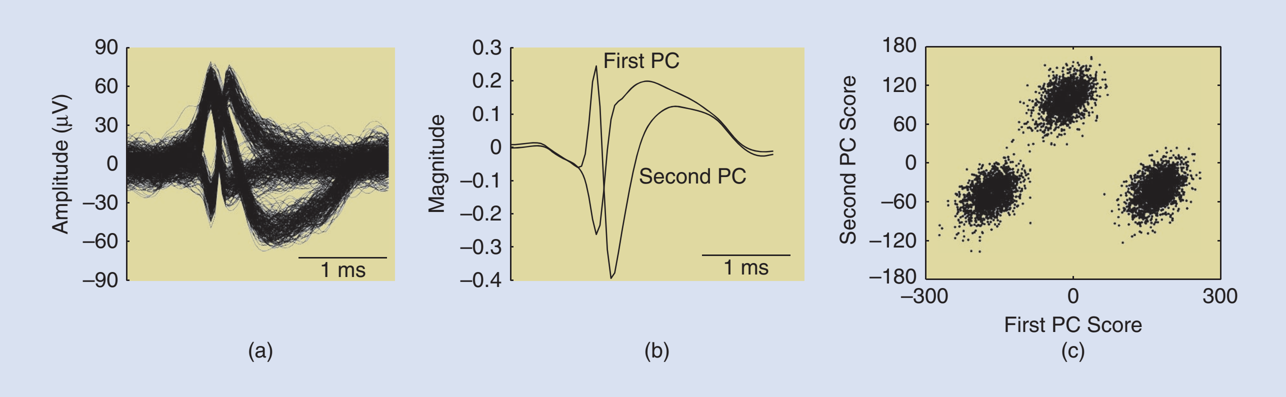
When spike detection is performed in the digital domain, whenever the voltage signal crosses a threshold, a window is applied and a spike waveform is captured. At this point, each spike is essentially aligned to the point of the threshold crossing. However, sampling jitter combined with noise effects may leave features of interest, such as maximum and minimum values, misaligned. Because this temporal misalignment can have a nasty effect on spike classification, alignment should be performed prior to classification.

The alignment process usually begins by upsampling the signal (using an interpolation method such as cubic spline) to help reduce the effects of sampling jitter. Then, the signal is aligned to some event in time. The aligned spikes may be downsampled to the original sampling rate after alignment.

The most common method of temporal alignment is to align each spike to the point of its maximum amplitude (Figure 7) [32]. Alignment to the point of maximum slope (Figure 7) has also been proposed [41], which is intuitive since the rising slope of the action potential has biological significance (Figure 2), unlike the peak amplitude. This method would be especially convenient if discrete derivatives (DDs) (described in the section "Feature Extraction") were already being used for feature extraction. Others have proposed alignment to the maximum of an



[FIG7] Examples of two different alignment methods. (a) The alignment to maximum amplitude and (b) the alignment to maximum slope.



[FIG8] Sample results of feature extraction using PCA. For the detected spikes in (a), PCs (b) are calculated, and each spike is expressed by (c) its first two PC coefficients.

energy measure such as the NEO [42], which would be convenient if NEO were already being used for spike detection. Similarly, alignment to the maximum integral [43] would be convenient if the integral transform (IT) (described in the next section) were being used for feature extraction. Indeed, it would be convenient to perform alignment with respect to any measure that is already being calculated in the sorting process.

IN THE 1970s, AS DIGITAL COMPUTERS GAINED POPULARITY AND PROCESSING CAPACITY, RESEARCHERS BEGAN USING MORE SOPHISTICATED ALGORITHMS FOR FEATURE EXTRACTION, SUCH AS PCA.

Although the aforementioned alignment methods will usually improve classification accuracy, alignment to a metric that is derived from the whole spike rather than from a single point may be less susceptible to the effects of background noise. One example of such a metric is the spike's center of mass [44]. Note that all of the algorithms that have been described in this section are completely automatic and real time.

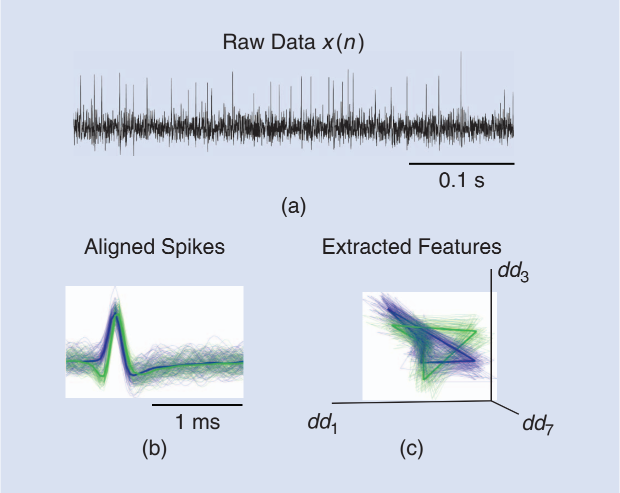
FEATURE EXTRACTION

Feature-extraction methods were also primitive in the early days of spike sorting. Often only very simple features such as the maximum spike amplitude, peak-to-peak amplitude, and spike width were used [32]. This approach, although simple, is quite susceptible to noise as well as to intrinsic variations in spike shapes.

In the 1970s, as digital computers gained popularity and processing capacity, researchers began using more sophisticated algorithms for feature extraction, such as PCA [45]. In PCA, the orthogonal basis [i.e., the "principal components" (PCs)] that captures the directions in the data with the largest variation is calculated by performing eigenvalue decomposition of the covariance matrix of the data. Each spike is then expressed as a series of PC coefficients c_i

$$c_i = \sum_{n=1}^{N} PC_i(n) \cdot s(n), \qquad (6)$$

where s is a spike, N is the number of samples in a spike/PC, and PC_i is the ith PC (Figure 8). These coefficients are then clustered to obtain the spike classifications. This method raised the bar on the classification performance that could be achieved, especially for noisier data. An added bonus is that, because most of the variance is captured in the first few components, the dimensionality can be reduced by only keeping the first two or three PCs, thereby reducing the computation time of PC coefficient calculation and of subsequent clustering. Even today, PCA is the most trusted and most commonly used method of spike sorting. The downside to PCA is that it is not a real-time algorithm. It is usually performed offline after the acquisition of the entire data set, but it can be modified to include a training period during which the PCs are calculated followed by a real-time PC-coefficient-calculation period. We confirmed in [29] that PCA achieves a high accuracy but at a high computational cost (Figure 6).



[FIG9] Sample results of feature extraction using DDs. For the synthetic data shown in (a), (b) spikes were detected and aligned and each spike was expressed by (c) three discrete derivative "coefficients." Spikes have been colored according to the ground truth. (Figure adapted from [30].)

Besides for spike detection, the DWT has also been proposed for feature extraction by [33]. The DWT should work well for feature extraction since it is a multiresolution technique that provides good localization in both time and frequency. As in PCA, performing the DWT on spike waveforms results in a set of "expansion coefficients," which can then be clustered to achieve spike classification. Figure 6 shows that this method has a fairly high accuracy but a relatively high cost.

Methods have also been developed with the accuracy–complexity tradeoff in mind. One example is called the DD method and is like a simplified version of DWT [46]. DDs are calculated by computing the slope at each sample point, over a number of different time scales

$$dd_\delta(n) = s(n) - s(n-\delta), \qquad (7)$$

where s is a spike and δ is an integer related to the time scale. Our studies have shown this to be a reliable yet inexpensive method for spike sorting (Figure 6). An example showing how this method can emphasize differences between spike classes is shown in Figure 9 for synthetic data, where spikes have been colored according to the ground truth.

Another such method is called the IT [47], in which spikes are classified based on the areas under the positive and negative phases of the spike, I_A and I_B, respectively

$$I_A = \frac{1}{N_A} \sum_{n=n_A}^{n_A+N_A-1} s(n)\,, \quad I_B = \frac{1}{N_B} \sum_{n=n_B}^{n_B+N_B-1} s(n), \qquad (8)$$

where s is the spike, n_A is the first sample of the positive phase, N_A is the total number of samples in the positive phase, n_B is the first sample of the negative phase, and N_B is the total number of samples in the negative phase of the spike. N_A, N_B, n_A, and n_B are all determined by offline training. This method is appealing because of the simple hardware implementation presented. Since only two features are extracted from each spike (I_A and I_B), the resulting dimensionality of this method is two, and no dimensionality reduction is required before clustering. The drawback to this method is its poor performance (Figure 6), comparable to or worse than that of primitive methods of the early days.

[DIMENSIONALITY REDUCTION IS A CRITICAL STEP IN SPIKE SORTING FOR A NUMBER OF REASONS.]

DIMENSIONALITY REDUCTION

Dimensionality reduction is a critical step in spike sorting for a number of reasons. The most obvious reason is that it will significantly reduce the required memory and computational complexity of clustering, resulting in significant reductions in the area and power of the spike-sorting hardware. Another obvious benefit is that it reduces the output data rate of spike-sorting hardware configured to output features only. A third reason that makes dimensionality reduction critical is that it improves the accuracy of clustering. Adding dimensions in clustering improves the performance only up to a certain point, after which adding more dimensions can cause the performance of the clusterer to degrade. One reason for this may be that dimensions in which the data is not separated introduce noise or confusion into the clusterer.

The most primitive way that the dimensionality of features can be reduced is with uniform sampling, in which to reduce the dimensionality from N to D, we simply choose D evenly spaced samples, for example, by choosing every N/Dth sample beginning with sample number $D/2$. This is essentially the same as choosing D random samples. As one might guess, this method is not very reliable (Figure 6, "Unif.").

A smarter way to choose the features that can best separate clusters, as shown in Figure 10, is by finding those features that have mulitmodal distributions across spikes, as multimodal distributions are an indication that more than one population (collection of spikes from the same neuron) is present in the data set. Next, we present three dimensionality-reduction algorithms that use this approach.

The first of these algorithms is called the Lilliefors test [48], a modification of the Kolmogorov-Smirnov test. The null hypothesis is that the data under question comes from any normally distributed population (whereas the Kolmogorov-Smirnov test tests the null hypothesis that the data comes from a standard normal distribution). The basic steps of the test areas follows:

1) Calculate the population mean and population variance of the data.

2) Calculate the empirical distribution function (EDF) of the data.

3) Test statistic: The maximum discrepancy between the EDF and the cumulative distribution function (CDF) of a normal distribution having the mean and variance calculated in 1).

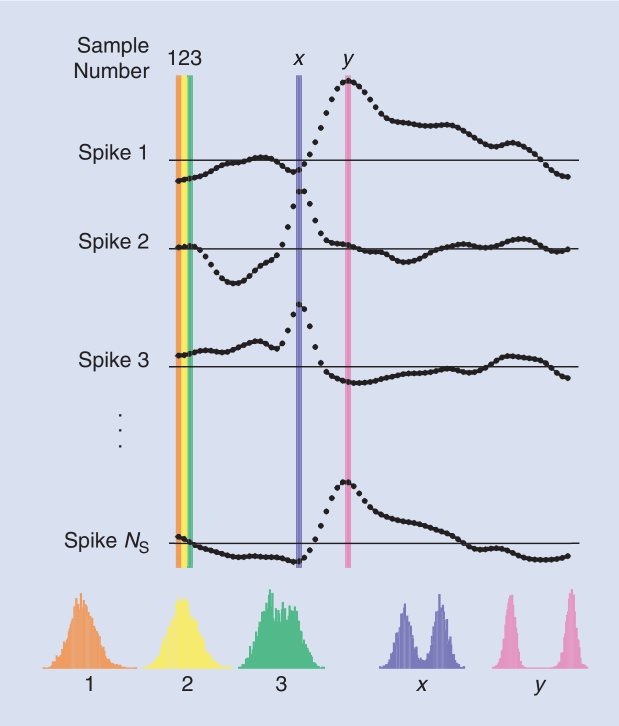
The Lilliefors test has been used by [33] for dimensionality reduction in spike sorting, where the test statistic was used to find the coefficients whose distributions differed most from the normal distribution. The assumption is that the null hypothesis will be rejected for coefficients with multimodal distributions but not for coefficients with unimodal distributions. Our study showed that this method has good performance but prohibitive complexity (logic and memory requirements) (Figure 6, "Lillie.").

Hartigan's dip test [49], [50] is a statistical test that looks specifically for multimodality. The CDF of a unimodal distribution has only one mode and is convex before the mode and concave after the mode. On the other hand, CDFs of multimodal distributions have more than one mode, and therefore have regions alternating between concave and convex. The basic steps in the dip test are as follows:

1) Calculate the EDF of the data.

2) Calculate the greatest convex minorant (GCM) and the least concave majorant (LCM).

3) Test statistic: The maximum distance ("dip") between the EDF and the GCM or LCM.



[FIG10] An illustration of how feature distribution information can be used in dimensionality reduction. For visualization purposes, we use the time samples of the spikes as features. In this example, the distributions of the amplitudes for samples one, two, and three are unimodal, so they would not be good choices of features to be used in clustering. Samples *x* and *y*, on the other hand, have bimodal distributions, indicating that clustering of these features would reveal the two underlying populations within the data.

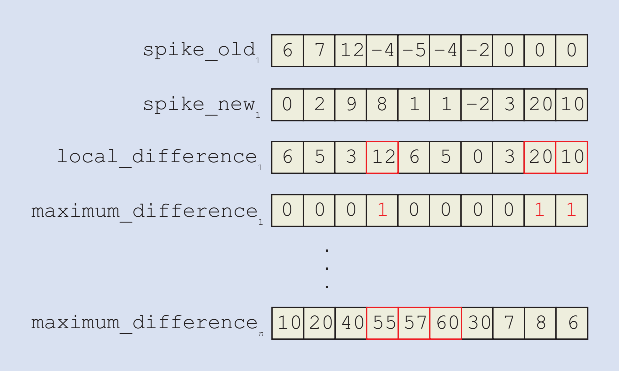
The coefficients whose distributions are "more multimodal" will have larger test statistics. Thus, we choose the coefficients that have the largest test statistics for use in clustering. Our study showed that this method also has good performance but an even greater hardware cost than the Lilliefors test (Figure 6, "Hart.").

CLUSTERING, ESPECIALLY UNSUPERVISED CLUSTERING, IS OFTEN THE MOST DIFFICULT AND MOST COMPLEX PART OF THE SORTING PROCESS.

We proposed an alternative to the above algorithms with comparable accuracy yet far less complexity in [29]. In the maximum-difference test (illustrated in Figure 11), as in the Lilliefors test, we seek the coefficients with the most variability, only now under the limited-memory conditions typical of implantable hardware. For an initial feature dimensionality N, four N-sample arrays of memory are initialized to zero: `maximum_difference`, `local_difference`, `spike_new`, `spike_old`. Throughout the training period, the ith iteration of the algorithm is as follows:

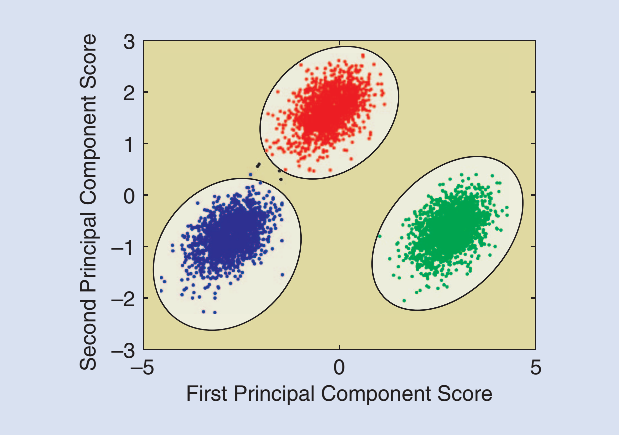
1) Write the current feature samples to the array `spike_new`.
2) Subtract the values in `spike_new`, coefficient by coefficient, from the values in `spike_old`, and write the absolute value of the result to `local_difference`.
3) Find the indices corresponding to the three largest values in `local_difference`.
4) Increment the values in `maximum_difference` indexed by these three indices.
5) Overwrite the values in `spike_old` with the values in `spike_new`.

These steps are repeated until the end of the training period. At this point, assuming that the goal is to reduce the dimensionality from N to D, `maximum_difference` is scanned for the locations corresponding to the D largest values, and the coefficients corresponding to these indices are identified as the coefficients that will be used in clustering. Figure 6 ("Max. Diff.") shows that this method is about three orders of magnitude less complex than both the Lilliefors test and the dip test yet even more accurate.



`spike_old`$_1$	6	7	12	-4	-5	-4	-2	0	0	0
`spike_new`$_1$	0	2	9	8	1	1	-2	3	20	10
`local_difference`$_1$	6	5	3	12	6	5	0	3	20	10
`maximum_difference`$_1$	0	0	0	1	0	0	0	0	1	1
⋮										
`maximum_difference`$_n$	10	20	40	55	57	60	30	7	8	6

[FIG11] Example execution of the maximum-difference test. At the beginning of the algorithm, the first spike would be stored in `spike_old` and the second spike in `spike_new`. The difference between the two arrays is calculated and its absolute value stored in `local_difference`. The indices corresponding to the three largest values in `local_difference` are four, nine, and ten; the values in `maximum_difference` indexed by these indices are each incremented by one. These steps are repeated until the end of the training period, when we have the final value of `maximum_difference`. Assuming that we want to reduce the dimensionality from ten to three, we choose the features indexed by the indices corresponding to the three largest values of `maximum_difference`, four, five, and six, to be used in clustering.



[FIG12] Example of results of manual cluster cutting on PCA features.

CLUSTERING

Clustering, especially unsupervised clustering, is often the most difficult and most complex part of the sorting process.

In the early days of spike sorting, the most common method of clustering was manual cluster cutting; extracted features were plotted on a scatter plot and cluster boundaries were defined by hand [32]. Even today, most commercial software packages for spike sorting provide the user with the capability to define cluster boundaries by drawing polygons in the chosen feature space using a mouse pointer (see the example in Figure 12). But because this method is prone to human errors [25], [26], not to mention the time that is required of the operator, automatic and semiautomatic methods are desirable. Another primitive but at least semiautomatic technique is that of window discriminators, in which spike waveforms that intersect one or several user-defined windows are assigned to the same neuron.

A more sophisticated method of clustering, which is now the benchmark clustering method in this field, is called k-means [51]. The k-means algorithm is based on a distance metric. The main steps of the algorithm are described in "k-Means Clustering." The main benefit of using k-means is that it is a very simple and fast algorithm. However, a major drawback to this algorithm is that it is not unsupervised, as it requires the user to input k. For applications such as BMIs, the spike sorting must be completely automatic, so there will be no user to input this information. Moreover, even if there were a user,

SPC HAS BENEFITS SIMILAR TO THOSE OF VALLEY SEEKING: IT IS UNSUPERVISED (AGAIN, NO A PRIORI KNOWLEDGE OF THE NUMBER OF CLUSTERS IS REQUIRED) AND NONPARAMETRIC.

determining the number of neurons is a nontrivial, often difficult task. Efforts are being made, however, to find ways of automatically and reliably determining the number of neurons in a recording [52]; perhaps these techniques could be used to initialize *k*-means, making it a fully unsupervised algorithm. Another drawback of *k*-means is that it is not real time, making it unsuitable for real-time applications. A compromise would be to adapt the algorithm to have a training period, where the cluster centroids are defined, followed by a real-time classification period, but this would only be appropriate for stationary data. Yet another drawback of this algorithm is that it is parametric; since each point is assigned to a cluster based solely on its Euclidean distances from the cluster centroids, the determined clusters will necessarily be spherical. There are many instances when the distribution of neural data will not be spherical. For example, during electrode drift, data tends to form ellipsoidal clusters. *K*-means would force spherical clusters, possibly dividing ellipsoidal clusters into two.

One unsupervised clustering algorithm is called valley seeking [53]. The idea in valley seeking is to first calculate the normalized density derivative (NDD) and then to find the peaks of this function. The cluster boundaries are then identified as the regions between the peaks (i.e., the valleys). An overview of the algorithm is provided in "Valley-Seeking Clustering." The benefits of the valley-seeking algorithm are that it is unsupervised (not even the number of clusters is required to be provided by the user) and nonparametric, giving it the ability to cluster data sets that have nontrivial shapes, such as donuts and spirals. The algorithm is not real time, however, making it unsuitable for real-time applications. Additionally, from a hardware point of view, the algorithm has the serious drawback of high complexity. It requires the computation and storage in memory of at least six N_S-by-N_S matrices, where N_S is the number of spikes being clustered. For large values of N_S valley seeking may not be a viable choice for hardware implementation.

Superparamagnetic clustering (SPC) [54] is another unsupervised clustering algorithm that has found application to spike sorting [33]. In SPC, the data is modeled as a granular magnet, where each point is assigned a spin. The model is heated from low temperatures to high temperatures. At very low temperatures, all the spins will be aligned; this is referred to as the "ferromagnetic region." At high temperatures, the system is disordered and all the spins are random; this is called the "paramagnetic region." At temperatures that lie between these regions (called the "superparamagnetic region"), spins within the same high-density region are aligned while the spins of different high-density regions are not aligned; here the clusters are revealed. A summary of the algorithm steps is provided in "SPC."

SPC has benefits similar to those of valley seeking: it is unsupervised (again, no a priori knowledge of the number of clusters is required) and nonparametric. It also has the

k-MEANS CLUSTERING

1) Define k (number of clusters/neurons).
2) Randomly define the k cluster centroids.
3) Assign each data point to the cluster with the closest (usually by Euclidean distance measure) centroid.
4) Recompute each cluster centroid as mean of that cluster.

Steps 3–4 are repeated until a convergence criterion (either that the assignments stop changing or that the maximum number of iterations has been reached) is met.

VALLEY-SEEKING CLUSTERING

Definitions. Let x and x' be two data points. Denote the neighbor number (NN) of x' with respect to x as $NN(x,x')=k$ and $NN(x',x)=l$. Denote the ith neighbor of x as $x_{(i)}$, i.e., $NN(x, x_{(i)}) = i$, and $NN(x_{(i)},x) = a_{(i)}$, $i = 1,2, \ldots ,k-1$. Similarly, denote the jth neighbor of x' as $x'_{(j)}$, $NN(x',x'_{(j)}) = j$, $NN(x'_{(j)},x') = b_{(j)}, j = 1,2, \ldots ,l-1$.

Algorithm steps:

1) Input threshold parameters t_1, t_2, and t_3.
2) Calculate the Euclidean distance matrix $D = (d_{ij})$.
3) Determine the neighbor number (NN) matrix $L = (l_{ij})$, where $l_{ij} = NN(x_i, x_j)$.
4) Calculate the matrix $S = (s_{ij})$, where

$$s_{ij} = \frac{(l_{ij} + l_{ji})}{2}.$$

5) Estimate the NDD matrix $J = (J_{ij})$, where

$$J_{ij} = \frac{|l_{ij} - l_{ji}|}{s_{ij}^{1+1/d}}, \; s_{ij} \leq t_1.$$

and d is the dimensionality of the feature space.

6) Estimate the convexity $D2 = (d2_{ij})$, where

$$d2_{ij} = \frac{l_{ij}\sum_{i=1}^{k-1} a_{(i)} + l_{ji}\sum_{j=1}^{l-1} b_{(j)}}{l_{ij}\sum_{i=1}^{k-1} i + l_{ji}\sum_{j=1}^{l-1} j}, \; s_{ij} \leq t_1.$$

7) Determine the discretized connectivity matrix $C = (c_{ij})$, where

$$c_{ij} = I(s_{ij} \leq t_1, J_{ij} \leq t_2, d2_{ij} \leq t_3)$$

is the indicator of whether x_i and x_j belong to the same cluster

8) Assign cluster labels to observations according to the discretized connectivity matrix.

SPC

1) Calculate the Euclidean distance matrix $D = (d_{ij})$.
2) Identify the K nearest neighbors of each point v_i. v_i and v_j are considered neighbors iff v_i is one of the K nearest neighbors of v_j and v_j is one of the K nearest neighbors of v_i.
3) Assign a Potts spin $s_i = 1, 2, \ldots, q$ to each point v_i randomly (or set all $s_i = 1$), where q is a constant representing the number of possible spins. Note: The value chosen for q does not imply anything about the number of clusters. The authors of [54] used $q = 20$.
4) Calculate the interaction strength J_{ij} between neighboring points v_i and v_j, where

$$J_{ij} = \begin{cases} \frac{1}{K}\exp\left(-\frac{d_{ij}^2}{2a^2}\right) & \text{if } v_i \text{ and } v_j \text{ are neighbors,} \\ 0 & \text{otherwise,} \end{cases}$$

and a is the average of all d_{ij}'s between neighboring points.

5) For each temperature (e.g. $T = 0{:}0.02{:}0.2$), perform the following Monte Carlo simulation; of iterations $m = 1{:}\,M$:
   a) Assign a frozen bond between nearest-neighbor points v_i and v_j with probability:

   $$p_{i,j}^f = 1 - \exp\left(-\frac{J_{ij}}{T} \cdot \delta_{s_i, s_j}\right),$$

   where $\delta_{s_i, s_j} = \begin{cases} 1 & \text{if } s_i = s_j, \\ 0 & \text{otherwise.} \end{cases}$

   b) Generate a random number x from a uniform distribution on $[0, 1]$. If $x < p_{i,j}^f$, there is a; bond between v_i and v_j.
   c) Define clusters as all points that are connected by a bond.
   d) Define c^m, where

   $$c_{ij}^m = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are in the same cluster,} \\ 0 & \text{otherwise.} \end{cases}$$

6) Calculate the two-point connectedness C_{ij}, where

$$C_{ij} = \frac{1}{M}\sum_{m=1}^{M} c_{ij}^m.$$

7) Calculate the spin-spin correlation function:

$$G_{ij} = \frac{(q-1)c_{ij}+1}{q}.$$

8) If $G_{ij} > \theta$, where θ is a predefined threshold, v_i and v_j belong to the same cluster.
9) Assign cluster labels to observations according to G.

OSORT CLUSTERING

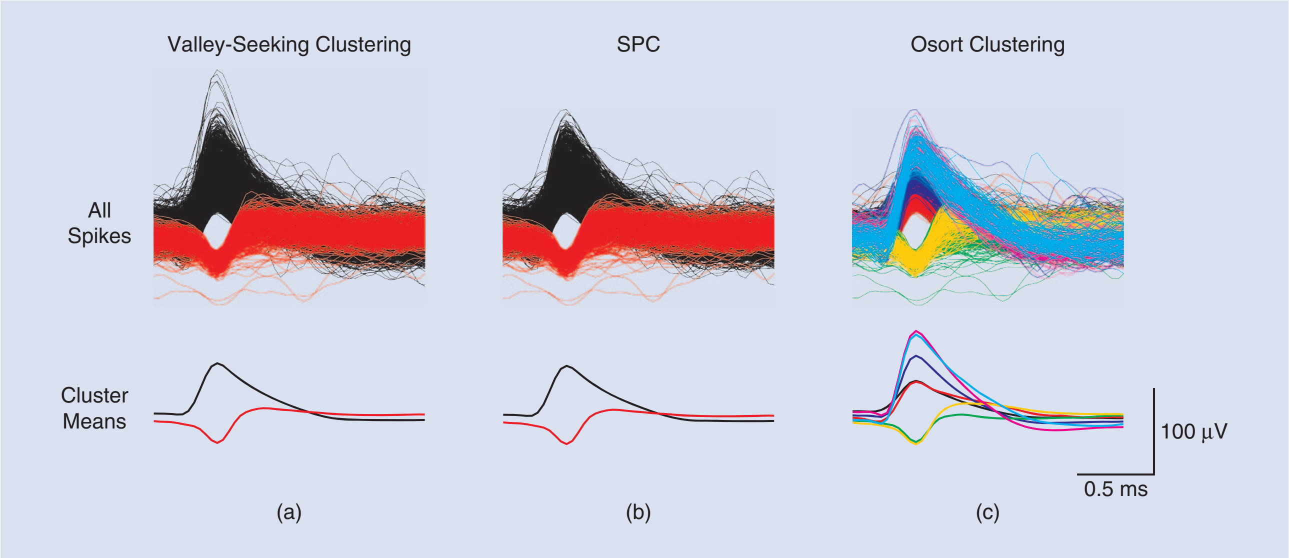
1) Initialization: Assign the first data point to its own cluster.
2) Calculate the Euclidean distance between the next data point and each cluster centroid.
3) If the smallest distance is less than the merging threshold T_M, assign the point to the nearest cluster and recompute that cluster's mean using the N most recent points. Otherwise, start a new cluster.
4) Check the distances between each cluster and every other cluster. If any distance is below the sorting threshold T_S, merge those two clusters and recompute its mean.

Steps 2–4 are then repeated indefinitely. In the simplified version of this algorithm, $T_M = T_S$, which is equal to the variance of the data computed continuously on a long (~1 min) sliding window. Note that when computing cluster centroids, only the N most recent points are used. This helps to account for electrode drift, since the clusters are allowed to drift as well.

same major drawback: complexity. It requires the computation of at least nine N_S-by-N_S matrices, where N_S is the number of spikes being clustered. Again, a large N_S requires a prohibitive number of operations and amount of memory. SPC also requires a Monte Carlo simulation, which increases the computation time. The algorithm can be simplified by using a mean-field approximation in place of the Monte Carlo simulations. But although this simplification reduces the run time, it actually increases the complexity. Furthermore, like the valley-seeking method, SPC is an offline algorithm. Thus, SPC is also not a practical choice for hardware implementation.

The only clustering algorithm known to the authors at this time that is both automatic and online and that has a good accuracy–complexity tradeoff is called Osort [55]. This method was developed by researchers who needed to isolate single neurons during their experiments, which requires processing large amounts of data in real time. Out of necessity, they proposed a much simpler way of clustering, where each data point is assigned to a cluster "on-the-fly." The algorithm is described in "Osort Clustering." This method of clustering appears to be extremely efficient. Very little memory is required. Therefore, of the three unsupervised clustering algorithms presented here, this method is the only one suitable for implementation in hardware. The main drawback to this method is that, like in k-means, it bases its decisions on a distance metric, essentially assuming a spherical distribution of data. So while it can track spherical clusters moving in time to form ellipsoidal clusters, it cannot resolve a stationary ellipsoidal cluster (which would result, for example, from multivariate noise).

An example showing the results of clustering using valley-seeking clustering, SPC, and Osort is shown in Figure 13. Valley seeking and SPC give similar results, whereas Osort appears to overcluster—that is, to find too many clusters, or to divide a single-unit cluster into subclusters. A summary of various characteristics of each of the algorithms described in this section is given in Table 1.



[FIG13] Example results from clustering 30 s of real data (human entorhinal cortex) using three different clustering methods. Note that for the valley-seeking and SPC methods, PCA was performed for feature extraction prior to clustering, whereas Osort uses only time-domain samples for clustering by default.

ALTERNATIVE METHODS FOR SPIKE SORTING

While most spike-sorting methods to date involve feature extraction followed by clustering of these features using nonparametric, nearest-neighbor methods as described above, other, more sophisticated methods have been developed based on Gaussian mixture models [44], [56]–[58] or t-distribution mixture models [59] in attempt to provide optimal solutions to the clustering problem based on statistics and probability theory. Many of these methods involve calculating a noise model for a particular data set, performing noise whitening, and using Bayesian or maximum-likelihood estimation.

An example of one such method was presented in [58]. In summary, the method begins with calculating an empirical model for the recording noise and using this model to perform noise whitening on the data. Next, a "data generation model" (which includes the number of clusters and their positions in the event space) that maximizes the a posteriori probability to observe the samples that are actually observed (i.e., to maximize the likelihood function) is calculated as follows:

1) Specify a model M by specifying the number of neurons K, their discharge frequencies π_j, $j = 1, \dots, K$, and their template waveforms $\boldsymbol{u}_j$, $j = 1, \dots, K$.

2) Compute the probability for unit (neuron) j to have generated the event (spike) $\boldsymbol{e}_i$, $p(\boldsymbol{e}_i|\boldsymbol{u}_j)$, as follows:

a) Define the residual vector $\boldsymbol{\Delta}_{ij} = \boldsymbol{e}_i - \boldsymbol{u}_j$.

b) Then $p(\boldsymbol{e}_i|\boldsymbol{u}_j) = 1/(2\pi)^{D/2} \cdot \exp(-1/2 \cdot \boldsymbol{\Delta}_{ij}^T \boldsymbol{\Delta}_{ij})$, where D is the dimensionality of the event space.

3) Calculate the probability P_i for the model to have generated event $\boldsymbol{e}_i : P_i = \sum_{j=1}^{K} \pi_j \cdot p(\boldsymbol{e}_i|\boldsymbol{u}_j)$, where π_j is the probability for unit j to occur.

4) Maximize the a posteriori probability, $\mathcal{P}(S;M) = \prod_{i=1}^{N} P_i$, or the likelihood function, $\mathcal{L}(S;M) = \sum_{i=1}^{N} \ln P_i$.

5) Use an iterative algorithm such as expectation-maximization to maximize $\mathcal{L}$.

Once the model is established, each event $\boldsymbol{e}_i$ is attributed to one of the K units by finding the j that minimizes $|\boldsymbol{\Delta}_{ij}|^2$, which is equivalent to choosing the unit with the highest probability to have generated the event.

The assumptions built in to this method are that the spike waveforms generated by a given neuron are constant, that the signal and the noise are statistically independent, that the signal and the noise sum linearly, and that the noise is well described by its covariance matrix. Nonstationary noise would violate this last assumption, but the authors claim that in such cases several noise covariance matrices could be used successively to describe the noise. The same assumption would be violated if the noise covariance matrix were to have third or

[TABLE 1] SUMMARY OF CLUSTERING ALGORITHM CHARACTERISTICS.

	MANUAL	WINDOW DISCRIMINATORS	*k*-MEANS	VALLEY SEEKING	SPC	OSORT
NONPARAMETRIC	NO	YES	NO	YES	YES	NO
AUTOMATIC/UNSUPERVISED	NO	NO	NO	YES	YES	YES
REAL-TIME/ONLINE	NO	YES	NO	NO	NO	YES
ADAPTIVE	NO	NO	NO	NO	NO	YES
ACCURACY	LOW+	?	0.90*	0.74*	0.85*	0.74*
COMPLEXITY	–	–	LOW	HIGH	HIGH	LOW

+[26], * [31]

higher order moments. However, the authors showed that, at least for their data, the background noise is well described by its covariance matrix.

CHALLENGES IN SPIKE SORTING

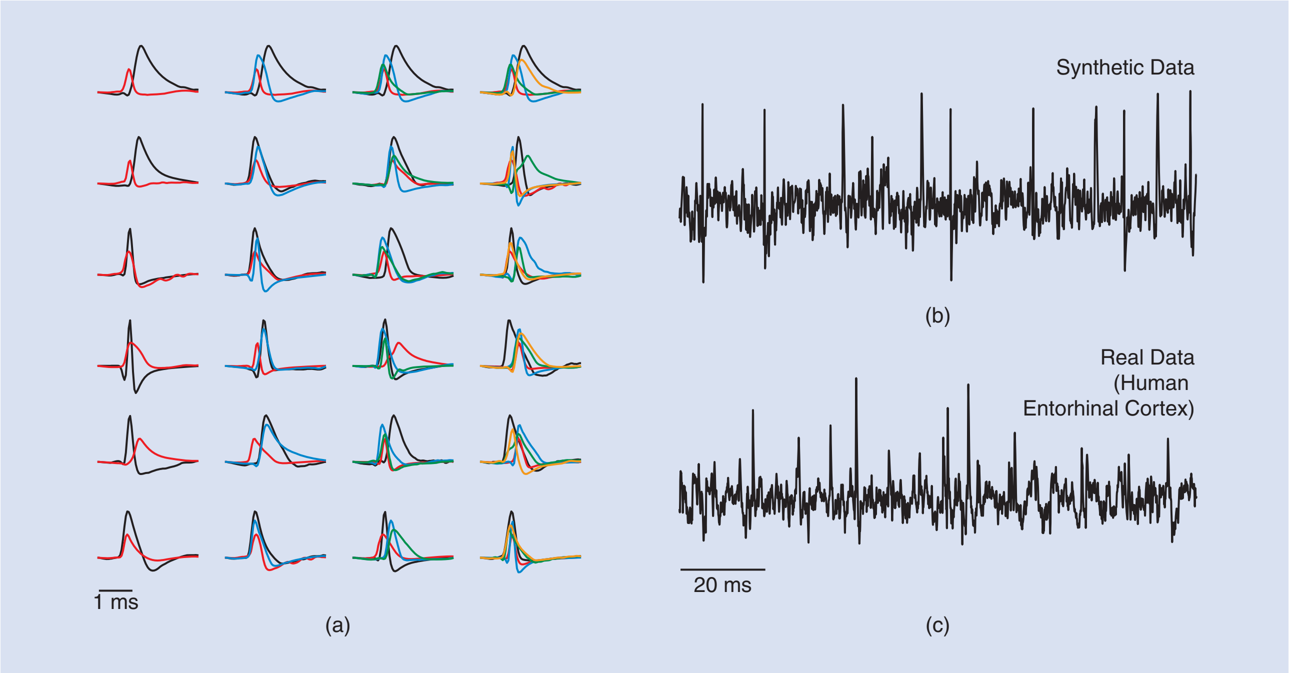
NO GROUND TRUTH

There are many unique characteristics of neural recording that make classification of neural data more difficult than for other types of data. One such characteristic is that there is almost always a lack of any sort of "ground truth." Many popular classification techniques, such as support vector machines, rely on a training period that uses known data to define cluster boundaries before the automatic classification period begins. In extracellular recording, however, experimenters typically must play a more passive role; we can only observe the neural activity, we cannot influence it. (Neural activity can be influenced by electrical stimulation, but usually not with single-cell precision.)Thus, we have no ground truth to be utilized in training the algorithms.

[THERE ARE MANY UNIQUE CHARACTERISTICS OF NEURAL RECORDING THAT MAKE CLASSIFICATION OF NEURAL DATA MORE DIFFICULT THAN FOR OTHER TYPES OF DATA.]

The lack of a ground truth also makes it nearly impossible to quantify the performance of the classifier. Let's revisit the problem, illustrated in Figure 1. The recording electrode is inserted into the neural tissue. Although each neuron in the tissue is generating its own, often independent, train of spikes, the recording electrode receives only the sum of the activity from all neurons in its vicinity. We want to use spike sorting to separate the composite signal into the individual spike trains. Now consider an analogous problem in classical communications theory. Let's say we want to quantify the performance of an error-correcting code (spike-sorting algorithm). To do so, we would generate a known test vector (signals from individual neurons), encode the signal, corrupt it with noise (mix the signals from individual neurons together), decode the signal (perform spike sorting), and finally calculate the bit error rate (classification performance) as the percentage of correctly received bits (percentage of correctly classified spikes). The problem is that in extracellular recording, we have no control over, or even knowledge of, the input signals. If we have no access to known test vectors, how can we quantify the performance of the classifier?

The best-known solution to this problem has been to perform simultaneous intracellular/extracellular recordings [60]. Although we still have no control over the input signals, the intracellular recordings at least provide us with some knowledge of them, so we can to some degree evaluate the clustering performance. The problem with this method is that intracellular recordings are very difficult to make, and there are very few such data sets already in existence, making thorough algorithm evaluations difficult. Furthermore, for each of the paired intracellular/extracellular recordings in [60], although the extracellular electrodes (tetrodes) may have



[FIG14] Examples of the synthetic data used in [28] and [29]. (a) Average spike waveforms used as templates in the test data sets, (b) sample of synthetic data generated from the spike templates in (a), and compare to real data from human entorhinal cortex in (c).

recorded signals from multiple neurons, only one neuron's spikes were intracellularly confirmed. This limits the degree to which the accuracy of an algorithm can be assessed. Another solution to the problem has been to use a real data set that has been annotated by an expert according to spike occurrences and classes. However, studies have shown that the performance of human operators is actually much lower than that of semiautomatic clustering tools [25], [26]. Therefore, it does not make sense to take the performance of a human operator as ground truth, particularly when calculating the accuracy of automatic methods that are likely to outperform the human operators. A third option has been to create biologically accurate synthetic data sets, which would provide both a ground truth and the flexibility to manipulate the signal variance and feature complexity in a way that is not possible using real data [28], [29], [33], [55], [61], [62]. An example of the synthetic data used by our group to perform the algorithm evaluations in [28] and [29] is shown in Figure 14.

AN ENTIRELY DIFFERENT APPROACH TO EVALUATING THE PERFORMANCE OF SPIKE SORTING IS TO USE POST-PROCESSING TECHNIQUES BASED ON OUR KNOWLEDGE OF THE STATISTICAL PROPERTIES OF FIRING NEURONS.

An entirely different approach to evaluating the performance of spike sorting is to use post-processing techniques based on our knowledge of the statistical properties of firing neurons. For example, one can look at the distribution of interspike intervals (ISIs) for each neuron after spike sorting [32], [55], [63], [64]. Under most circumstances, after firing an action potential a neuron cannot fire again until after a refractory period, typically 1–3 ms. An ISI histogram showing a significant number of samples fewer than 3 ms would indicate bad clustering (e.g., that spikes from two neurons were combined into one multiunit cluster, or that the cluster is a noise cluster).

Tankus et al. developed a method specifically to identify a cluster as a single cell or multiple units [64]. This task is normally performed by human visual inspection of the distributions of spike waveforms around the spike mean (the variation around the mean for single units should be small). As such, their approach was to mimic the performance of the human classifier. The method is composed primarily of two parts. First, the ISI distribution of each cluster is examined as described above, and a cluster is declared multiunit if more than 1% of ISIs are fewer than 3 ms. For each remaining cluster, the variance of the spike waveforms around the main rise in voltage of the mean waveform is quantified. Then clusters whose variances exceed a certain threshold are also declared multiunit.

Pouzat et al. also developed several additional clever post-processing techniques, including the standard deviation (S.D.) test, the χ^2 test, and the projection test [58]. The idea behind the S.D. test is that, assuming that the spike waveform generated by a given neuron is constant and that the signal and noise sum linearly, the sample-by-sample S.D. over all the spikes from one cluster should be equal to the standard deviation of the noise. So after spike classification, any cluster whose S.D. differs significantly from the noise S.D. can be either further scrutinized or discarded. The χ^2 test tests the hypothesis that each cluster of spikes forms a D-dimensional Gaussian distribution. (In the next subsection, however, we will examine whether the assumption of this test is valid or not.) The test is performed by first calculating the squared D-dimensional distance of every spike in a given cluster from its cluster mean and then by checking whether or not this distribution follows a χ^2 distribution with D degrees of freedom. A distribution that deviates significantly from the expected distribution may indicate the clustering of two similar units into one cluster. Finally, in the projection test, we again assume that the distribution of spikes in D-dimensional space should be a multivariate Gaussian with a covariance matrix equal to the identity matrix (assuming that noise whitening has been performed, as in the section "Alternative Methods for Spike Sorting"), and that the projection of all spikes onto all possible axes joining any pair of units should form Gaussian distributions centered on the cluster centroids with standard deviations equal to one. The "distinguishability" of any two given units can then be defined by setting a limit on the acceptable overlap between these two distributions, and a user can declare that units with less than a certain degree of distinguishability not be used in further analysis. This test also reveals when two clusters have been combined into one, as the projections between these two clusters will form a single Gaussian distribution centered around the true cluster mean.

NON-GAUSSIAN NOISE

Much of classical signal-detection theory is based on the assumption of channels having additive white Gaussian noise, and, as a result, most of the classical signal-detection techniques have been built around this assumption. Noise in extracellular recordings, on the other hand, has been shown to be both nonwhite [45] and non-Gaussian [21], [59], so many of these classical techniques cannot be applied. Even signal-detection techniques that do not assume Gaussian noise, just that the distribution of the noise is known, are difficult to apply to neural data due to a lack of accurate noise models, especially models that are valid across various experimental setups.

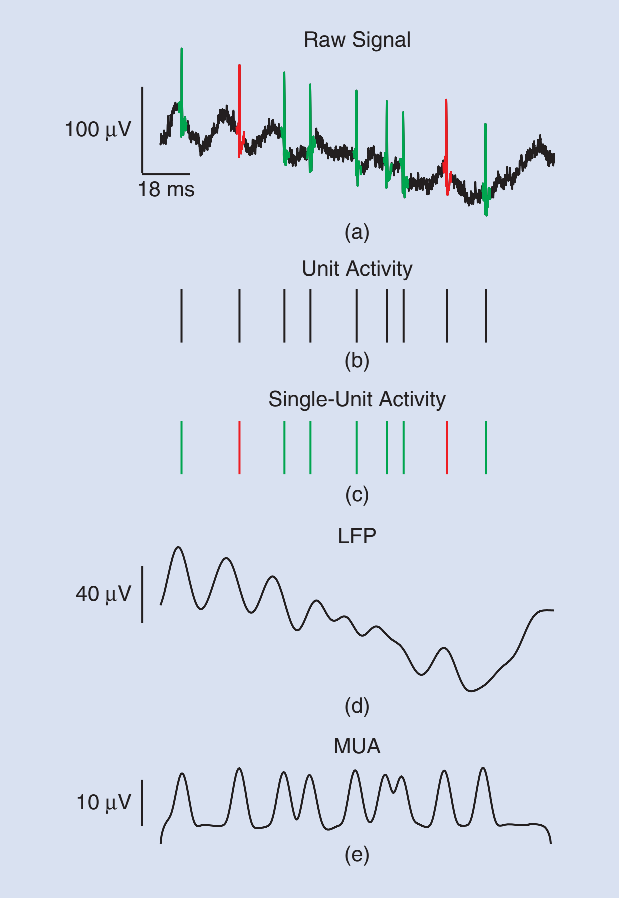
NONSTATIONARITIES

To make matters worse, neural data can be nonstationary. Fee et al. showed both that background noise is nonstationary and that a neuron's spike waveform varies as a function of the time since its preceding action potential [21]. This change in a neuron's spike waveform over time is especially dramatic during burst firing [32], [65], during which the peak amplitude of the spike will decrease, since it is firing before completely returning to its resting state. Other causes of nonstationarites are electrode drift [32], [65], when the stiff electrode drifts within the fluid tissue with respect to the neurons being recorded, and cortical pulsation due to heartbeat or respiration [65]. Despite all of these known contributors to nonstationarity, data stationarity is still an assumption built into most spike-sorting methods. Until spike-sorting methods are developed to combat this problem, classification results will suffer. Efforts in this direction include [65] and [66].

SPIKE SORTING WILL ALWAYS BE NECESSARY FOR ELECTROPHYSIOLOGICAL EXPERIMENTS THAT ARE DESIGNED TO STUDY THE BEHAVIOR OF INDIVIDUAL CELLS OR NETWORKS OF CELLS.

OVERLAPPING SPIKES

A final, very tricky problem in spike sorting is that of overlapping spikes. The refractory period forces a neuron to rest for at least 1 ms between successive action potentials. But remember that our recorded signal is the sum of signals from several nearby neurons that are assumed to be firing independently. Thus, it is possible for two different neurons to fire at or around the same time, such that their spikes overlap with one another in the recorded signal. At best, conventional spike-sorting methods may be able to identify such a detection as an outlier and, therefore, to classify it as noise. At worst, this overlap would be misclassified entirely. Ideally, we would like to be able to detect when an overlap occurs and to resolve which neurons the overlapping spikes have come from. Some techniques have been developed towards this goal; see [41], [56], and [67]–[71].



[FIG15] Parts (a)–(e) show an example of an MUA signal, compared to the raw signal, unit activity, and LFP. Signals in this figure were generated by the authors and plotted in the same manner as in [76].

SINGLE- VERSUS MULTICHANNEL-RECORDING SIGNAL PROCESSING

In multichannel recordings, adjacent channels sometimes receive activity from the same neurons. Examples of these types of multichannel recordings are stereotrode/tetrode recordings, where the recording probes (made from microwires) have two/four closely spaced electrodes ($\sim 10\ \mu$m between centers). In these cases, correlations between channels can be exploited to separate single-unit activity, similarly to how triangulation can be used to determine the position of one object with respect to two other objects. Several algorithms have been developed to make use of this information, including independent component analysis (ICA) [72].

The idea behind ICA is that if N sources (neurons) have been mixed onto N detectors (electrodes) using a linear combination, a matrix can be found to "unmix" the data such that each channel is independent from every other channel, i.e., each channel contains the spike train from each neuron. The assumptions behind this method are that, ideally, the number of electrodes equals the number of neurons (or, in the less ideal case, the number of detectors is greater than the number of sources), and that each neuron is seen by at least two electrodes. This method, when it can be applied, has many benefits, including the ability to automatically detect artifacts and overlapping spikes and to correctly sort spikes from a neuron whose amplitude changes with time (i.e., to handle waveform nonstationarities).

Note that multichannel silicon arrays have much larger spacing between electrodes ($\sim 400\ \mu$m for the Utah array [73]), so it is much less likely for the same neuron to be recorded on two channels here. As such, algorithms exploiting correlations between channels can typically not be used for multichannel recordings of this type.

CONTROVERSY

The last thing that we will mention about spike sorting is the ongoing debate within the field of neural prosthetics—still a relatively immature field—over whether or not spike sorting is really necessary for reliable decoding. Spike sorting in the traditional sense seeks the single best spike train for each observed neuron. As a result, ambiguous spike trains often get discarded, which may be undesirable or unacceptable in some cases such as chronic recordings. To mitigate this problem, Wood and Black propose using an infinite Gaussian mixture model to instead generate a distribution of spike trains, i.e., multiple different spike-sorting results with varying probabilities [74]. Then, this distribution of spike trains, rather than a single spike train, would serve as input to subsequent processing steps. The authors postulate that these results may be useful in certain types of neural signal analysis such as decoding algorithms which rely on cosine tuning. This approach has the benefit of quantifying the certainty of spike-sorting results and of improving single-unit yield. However, it is unclear how straightforward it would be to use this approach in neural signal analysis; downstream algorithms would likely have to be modified.

[IMPLANTABLE SPIKE-SORTING HARDWARE WOULD BRING MEDICAL TECHNOLOGIES FOR THE TREATMENT OF DISORDERS SUCH AS PARALYSIS, EPILEPSY, AND EVEN COGNITIVE AND MEMORY LOSS CLOSER TO A REALITY.]

A number of other researchers have actually reported sufficient decoding performance when multiunit, rather than single-unit, activity is decoded. Ventura, for example, presented a paradigm for using multiunit spike trains in conjunction with existing decoding algorithms, such as the population-vector and maximum-likelihood decoding algorithms, to predict movement with comparable performance to traditional methods that use single-unit spike trains [75]. By bypassing spike sorting, this method saves time and computational effort, making it more appropriate for use in real-time neural prosthetics. This method has the added advantage of performing well in low SNR, where accurate spike sorting can be unreliable. Actually, though, spike sorting is implicitly built in to this method, in that each constituent neuron's identity is revealed through information about tuning.

Stark and Abeles, on the other hand, came up with a decoding paradigm that involves no spike sorting at all, whether explicit or implicit [76]. They introduce a quantity called multiunit activity (MUA), which is calculated by bandpass-filtering the signal from 300–6,000 Hz and taking the RMS. They then used the MUA as input to classification algorithms such as support vector machines, Fisher's linear discriminant analysis, Poisson probability density estimation, and artificial neural networks, which traditionally use single-unit activity. For each of these decoding methods, they found MUA to give better motion-prediction performance than either single-unit activity or LFP. Other advantages of this method are that MUA is more easily obtained than single-unit activity, MUA recordings are more stable over time, and MUA is informative even in the absence of spikes. An example of an MUA signal, compared to spikes and LFP, is shown in Figure 15.

Another study showed that, while decoding is still better when single units are used, an acceptable level of performance can also be achieved using multiple units [77]. A number of other researchers have also reported success in movement decoding using LFP [78], [79]. The primary advantages to using LFPs over spikes are that they are easier to acquire, are more stable over time, and are less susceptible to noise. Many other studies have suggested using a combination of LFPs and spikes to achieve high decoding performance [80]–[82].

Still, the majority of published studies in the field of neural prosthetics have used single-unit activity as input to their decoding algorithms [14], [15], [83], [84]. Furthermore, neural prosthetics is just one of many applications for spike sorting. Spike sorting will always be necessary for electrophysiological experiments that are designed to study the behavior of individual cells or networks of cells.

OUTLOOK

Spike sorting is an important processing step for many of the scientific and clinical applications that involve the extracellular recording of neuronal activity. Work still remains in finding optimal automatic, real-time, efficient, and accurate spike-sorting algorithms that address all the remaining challenges described in the section "Challenges in Spike Sorting." Finding such a solution to the spike-sorting problem would finally allow reliable spike sorting to be performed in implantable hardware. Performing spike sorting in hardware, simultaneously on many channels, would provide researchers with whole new experimental paradigms. For example, on-site spike sorting would aid in providing experimenters with instantaneous information about the neurons, such as their tuning functions as a stimulus is varied. These signals could also be used to "close the loop" by delivering signals back to the brain, enabling a whole new class of neurophysiological and neuropsychological experiments. Performing spike sorting in hardware would also achieve enough data reduction to enable the wireless transmission of data, thereby eliminating the need for cables. This would open the door for new types of experiments in which the activity of the brain is investigated as animals move freely in enriched (and possibly even natural) environments. It may also allow for recording from species that have never before been recorded, such as freely flying bats. Finally, implantable spike-sorting hardware would bring medical technologies for the treatment of disorders such as paralysis, epilepsy, and even cognitive and memory loss closer to a reality.

ACKNOWLEDGMENTS

The authors thank Dr. R. Staba for reviewing the manuscript and providing invaluable feedback, especially with regards to the biology and for providing access to human electrophysiological recordings. We also thank V. Karkare, R. Chandler, and M. Rachid for their comments. This work was supported in part by the National Science Foundation under grant numbers 0847088 and 0824275.

AUTHORS

Sarah Gibson (sarah@ee.ucla.edu) received a B.S. degree in electrical and computer engineering from Baylor University in 2005 and an M.S. degree in electrical engineering in 2008 from the University of California, Los Angeles, where she is currently working toward a Ph.D. degree in electrical engineering. Her research interests are in systems and techniques for neural signal processing.

Jack W. Judy (jjudy@ee.ucla.edu) received the B.S.E.E. degree (summa cum laude) from the University of Minnesota, Minneapolis in 1989, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley in 1994 and 1996, respectively. He has been on the faculty of the Electrical Engineering Department at the University of California, Los Angeles (UCLA), since 1997, where he is currently an associate professor. At UCLA, he is the chair of the MEMS and nanotechnology major field of the Electrical Engineering Department and the director of the UCLA Neuroengineering Training Program.

Dejan Marković (dejan@ee.ucla.edu) received the Dipl.Ing. degree from the University of Belgrade, Serbia, in 1998 and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 2000 and 2006, respectively, all in electrical engineering. In 2006, he joined the faculty of the Electrical Engineering Department at the University of California, Los Angeles as an assistant professor. His current research is focused on digital integrated circuits and DSP architectures for parallel data processing in future radio and healthcare systems, design with post-CMOS devices, design optimization methods, and CAD flows.

REFERENCES

[1] N. Kipnis, "Luigi Galvani and the debate on animal electricity, 1791–1800," *Ann. Sci.*, vol. 44, no. 2, pp. 107–142, 1987.

[2] A. L. Hodgkin and A. F. Huxley, "Currents carried by sodium and potassium ions through the membrane of the giant axon of Loligo," *J. Physiol.*, vol. 116, no. 4, pp. 449–472, 1952.

[3] A. L. Hodgkin and A. F. Huxley, "The components of membrane conductance in the giant axon of Loligo," *J. Physiol.*, vol. 116, no. 4, pp. 473–496, 1952.

[4] A. L. Hodgkin and A. F. Huxley, "The dual effect of membrane potential on sodium conductance in the giant axon of Loligo," *J. Physiol.*, vol. 116, no. 4, pp. 497–506, 1952.

[5] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, 1952.

[6] D. H. Hubel and T. N. Wiesel, "Ferrier lecture: functional architecture of macaque monkey visual cortex," *Proc. R. Soc. Lond. B*, vol. 198, no. 1130, pp. 1–59, 1977.

[7] K. Sameshima and L. A. Baccalá, "Trends in multichannel neural ensemble recording instrumentation," in *Methods for Neural Ensemble Recordings,* M. A. L. Nicolelis, Ed. Boca Raton, FL: CRC, 1999, ch. 3, pp. 47–60.

[8] A. Bragin, C. L. Wilson, R. J. Staba, M. Reddick, I. Fried, and J. Engel, Jr., "Interictal high-frequency oscillations (80–500 Hz) in the human epileptic brain: entorhinal cortex," *Ann. Neurol.*, vol. 52, no. 4, pp. 407–415, 2002.

[9] R. J. Staba, C. L. Wilson, A. Bragin, I. Fried, and J. Engel, "Quantitative analysis of high-frequency oscillations (80–500 Hz) recorded in human epileptic hippocampus and entorhinal cortex," *J. Neurophysiol.*, vol. 88, no. 4, pp. 1743–1752, Oct. 2002.

[10] R. J. Staba, L. Frighetto, E. J. Behnke, G. W. Mathern, T. Fields, A. Bragin, J. Ogren, I. Fried, C. L. Wilson, and J. Engel, "Increased fast ripple to ripple ratios correlate with reduced hippocampal volumes and neuron loss in temporal lobe epilepsy patients," *Epilepsia*, vol. 48, no. 11, pp. 2130–2138, Nov. 2007.

[11] M. A. L. Nicolelis, "Actions from thoughts," *Nature*, vol. 409, no. 6818, pp. 403–407, 2001.

[12] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.

[13] T. W. Berger, A. Ahuja, S. H. Courellis, S. A. Deadwyler, G. Erinjippurath, G. A. Gerhardt, G. Gholmieh, J. J. Granacki, R. Hampson, M. C. Hsaio, J. Lacoss, V. Z. Marmarelis, P. Nasiatka, V. Srinivasan, D. Song, A. R. Tanguay, and J. Wills, "Restoring lost cognitive function," *IEEE Eng. Med. Biol. Mag.*, vol. 24, no. 5, pp. 30–44, 2005.

[14] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, June 2002.

[15] K. V. Shenoy, D. Meeker, S. Cao, S. A. Kureshi, B. Pesaran, C. A. Buneo, A. P. Batista, P. P. Mitra, J. W. Burdick, and R. A. Andersen, "Neural prosthetic control signals from plan activity," *Neuroreport*, vol. 14, no. 4, pp. 591–596, 2003.

[16] E. M. Schmidt, "Electrodes for many single neuron recordings," in *Methods for Neural Ensemble Recordings,* M. A. L. Nicolelis, Ed. Boca Raton, FL: CRC, 1999, ch. 1, pp. 1–23.

[17] R. R. Harrison, "A low-power integrated circuit for adaptive detection of action potentials in noisy signals," in *Proc. 25th Ann. Int. Conf. IEEE EMBS,* Cancun, Mexico, 2003, pp. 3325–3328.

[18] G. Buzsáki, M. Penttonen, Z. Nádasdy, and A. Bragin, "Pattern and inhibition-dependent invasion of pyramidal cell dendrites by fast spikes in the hippocampus in vivo," *Proc. Natl. Acad. Sci. USA*, vol. 93, no. 18, pp. 9921–9925, Sept. 1996.

[19] Z. Nádasdy, J. Csicsvari, M. Penttonen, J. Hetke, K. Wise, and G. Buzsáki, "Extracellular recording and analysis of neuronal activity: From single cells to ensembles," in *Neuronal Ensembles: Strategies for Recording and Decoding,* H. B. Eichenbaum and J. L. Davis, Eds. New York: Wiley-Liss, 1998, ch. 2, pp. 17–55.

[20] W. H. Freygang and K. Frank, "Extracellular potentials from single spinal motoneurons," *J. Gen. Physiol*, vol. 42, no. 4, pp. 749–760, 1959.

[21] M. S. Fee, P. P. Mitra, and D. Kleinfeld, "Variability of extracellular spike waveforms of cortical neurons," *J. Neurophysiol.*, vol. 76, no. 6, pp. 3823–3833, 1996.

[22] F. W. Campbell, B. G. Cleland, G. F. Cooper, and C. Enroth-Cugell, "The angular selectivity of visual cortical cells to moving gratings." *J. Physiol.*, vol. 198, no. 1, pp. 237–250, Sept. 1968.

[23] C. M. Gray and G. Viana Di Prisco, "Stimulus-dependent neuronal oscillations and local synchronization in striate cortex of the alert cat," *J. Neurosci.*, vol. 17, no. 9, pp. 3239–3253, May 1997.

[24] G. Buzsáki and R. D. Traub, "Physiologic basis of the electroencephalogram and local field potentials," in *Epilepsy: A Comprehensive Textbook, 2nd ed.*, J. Engel, Jr. and T. A. Pedley, Eds. Philadelphia, PA: Lippincott Williams & Wilkins, 2008, ch. 72, pp. 797–807.

[25] F. Wood, M. J. Black, C. Vargas-Irwin, M. Fellows, and J. P. Donoghue, "On the variability of manual spike sorting," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 912–918, 2004.

[26] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsáki, "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements," *J. Neurophysiol.*, vol. 84, no. 1, pp. 401–414, 2000.

[27] W. L. G. Koontz, P. M. Narendra, and K. Fukunaga, "A graph-theoretic approach to nonparametric cluster analysis," *IEEE Trans. Comput.*, vol. C-25, no. 9, pp. 936–944, Sept. 1976.

[28] S. Gibson, J. W. Judy, and D. Marković, "Comparison of spike-sorting algorithms for future hardware implementation," in *Proc. 30th Ann. Int. Conf. IEEE EMBS,* Vancouver, Canada, 2008, pp. 5015–5020.

[29] S. Gibson, J. W. Judy, and D. Marković, "Technology-aware algorithm design for neural spike detection, feature extraction, and dimensionality reduction." *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 5, pp. 469–78, Oct. 2010.

[30] V. Karkare, S. Gibson, and D. Marković, "A 130-μW, 64-channel spike-sorting DSP chip," in *Proc. IEEE Asian Solid-State Circuits Conf.,* Taipei, Taiwan, 2009, pp. 289–292.

[31] V. Karkare, S. Gibson, C.-H. Yang, H. Chen, and D. Marković, "A 75 μW, 16-channel neural spike-sorting processor with on-the-fly clustering," in *Proc. Int. Symp. VLSI Circuits,* 2011, pp. 252–253.

[32] M. S. Lewicki, "A review of methods for spike sorting: The detection and classification of neural action potentials," *Network: Comput. Neural Syst.*, vol. 9, no. 4, pp. R53–R78, Nov. 1998.

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

[34] K. H. Kim and 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, Aug. 2003.

[35] J. F. Kaiser, "On a simple algorithm to calculate the 'energy' of a signal," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '90),* Albuquerque, NM, 1990, vol. 1, pp. 381–384.

[36] S. Mukhopadhyay and G. C. Ray, "A new interpretation of nonlinear energy operator and its efficacy in spike detection," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 2, pp. 180–187, Feb. 1998.

[37] K. H. Kim and S. J. Kim, "Neural spike sorting under nearly 0-dB signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 10, pp. 1406–1411, Oct. 2000.

[38] I. Obeid and P. D. Wolf, "Evaluation of spike-detection algorithms for a brain-machine interface application," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 905–911, June 2004.

[39] E. Hulata, R. Segev, Y. Shapira, M. Benveniste, and E. Ben-Jacob, "Detection and sorting of neural spikes using wavelet packets," *Phys. Rev. Lett.*, vol. 85, no. 21, pp. 4637–4640, 2000.

[40] R. J. Brychta, S. Tuntrakool, M. Appalsamy, N. R. Keller, D. Robertson, R. G. Shiavi, and A. Diedrich, "Wavelet methods for spike detection in mouse renal sympathetic nerve activity," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 1, pp. 82–93, 2007.

[41] R. Chandra and L. M. Optican, "Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 5, pp. 403–412, May 1997.

[42] J. H. Choi, H. K. Jung, and T. Kim, "A new action potential detector using the MTEO and its effects on spike sorting systems at low signal-to-noise ratios," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 4, pp. 738–746, 2006.

[43] A. Zviagintsev, Y. Perelman, and R. Ginosar, "Algorithms and architectures for low power spike detection and alignment," *J. Neural Eng.*, vol. 3, no. 1, pp. 35–42, 2006.

[44] M. Sahani, "Latent variable models for neural data analysis," Ph.D. dissertation, California Inst. Technol., Pasadena*, CA,* May 1999.

[45] M. Abeles and M. H. Goldstein, Jr., "Multispike train analysis," *Proc. IEEE*, vol. 65, no. 5, pp. 762–773, May 1977.

[46] Z. Nadasdy, R. Q. Quiroga, Y. Ben-Shaul, B. Pesaran, D. A. Wagenaar, and R. A. Andersen. (2002). Comparison of unsupervised algorithms for on-line and off-line spike sorting. *Proc. 32nd Annu. Meeting Soc. for Neurosci.* [Online]. Available: http://www.vis.caltech.edu/~zoltan/

[47] A. Zviagintsev, Y. Perelman, and R. Ginosar, "Low-power architectures for spike sorting," in *Proc. 4th Int. IEEE EMBS Conf. Neural Eng.,* Arlington, VA, Mar. 2005, pp. 162–165.

[48] H. W. Lilliefors, "On the Kolmogorov-Smirnov test for normality with mean and variance unknown," *J. Amer. Statist. Assoc.*, vol. 62, no. 318, pp. 399–402, June 1967.

[49] J. A. Hartigan and P. M. Hartigan, "The dip test of unimodality," *Ann. Stat.*, vol. 13, no. 1, pp. 70–84, Mar. 1985.

[50] P. M. Hartigan, "Computation of the dip statistic to test for unimodality," *J. Appl. Statist.*, vol. 34, no. 3, pp. 320–325, 1985.

[51] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, 1967, vol. 1, pp. 281–297.

[52] D. Novák, J. Wild, T. Sieger, and R. Jech, "Identifying number of neurons in extracellular recording," in *Proc. 4th Int. IEEE EMBS Conf. Neural Eng.,* Antalya, Turkey, 2009, pp. 742–745.

[53] C. Zhang, X. Zhang, M. Q. Zhang, and Y. Li, "Neighbor number, valley seeking and clustering," *Pattern Recognit. Lett.*, vol. 28, no. 2, pp. 173–180, 2007.

[54] M. Blatt, S. Wiseman, and E. Domany, "Data clustering using a model granular magnet," *Neural Comput.*, vol. 9, no. 9, pp. 1805–1842, 1997.

[55] U. Rutishauser, E. M. Schuman, and A. N. Mamelak, "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo," *J. Neurosci. Methods*, vol. 154, no. 1–2, pp. 204–224, 2006.

[56] M. S. Lewicki, "Bayesian modeling and classification of neural signals," *Neural Comput.*, vol. 6, no. 5, pp. 1005–1030, 1994.

[57] M. Sahani, J. S. Pezaris, and R. A. Andersen, "Extracellular recording from multiple neighboring cells: a maximum-likelihood solution to the spike-separation problem," in *Proc. 6th Ann. Conf. Comput. Neurosci. (CNS97)*. New York: Plenum, 1998, pp. 619–625.

[58] C. Pouzat, O. Mazor, and G. Laurent, "Using noise signature to optimize spike-sorting and to assess neuronal classification quality," *J. Neurosci. Methods*, vol. 122, no. 1, pp. 43–57, 2002.

[59] S. Shoham, M. R. Fellows, and R. A. Normann, "Robust, automatic spike sorting using mixtures of multivariate t-distributions," *J. Neurosci. Methods*, vol. 127, no. 2, pp. 111–122, 2003.

[60] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsáki, "Intracellular features predicted by extracellular recordings in the hippocampus in vivo," *J. Neurophysiol.*, vol. 84, no. 1, pp. 390–400, 2000.

[61] L. S. Smith and N. Mtetwa, "A tool for synthesizing spike trains with realistic interference," *J. Neurosci. Methods*, vol. 159, no. 1, pp. 170–180, 2007.

[62] J. Martinez, C. Pedreira, M. J. Ison, and R. Quian Quiroga, "Realistic simulation of extracellular recordings." *J. Neurosci. Methods*, vol. 184, no. 2, pp. 285–293, Nov. 2009.

[63] D. H. Perkel, G. L. Gerstein, and G. P. Moore, "Neuronal spike trains and stochastic point processes. I. The single spike train," *Biophys. J.*, vol. 7, no. 4, pp. 391–418, 1967.

[64] A. Tankus, Y. Yeshurun, and I. Fried, "An automatic measure for classifying clusters of suspected spikes into single cells versus multiunits," *J. Neural Eng.*, vol. 6, no. 5, p. 056001, Oct. 2009.

[65] R. K. Snider and A. B. Bonds, "Classification of non-stationary neural signals," *J. Neurosci. Methods*, vol. 84, no. 1-2, pp. 155–166, 1998.

[66] A. Bar-Hillel, A. Spiro, and E. Stark, "Spike sorting: Bayesian clustering of non-stationary data," *J. Neurosci. Methods*, vol. 157, no. 2, pp. 303–316, 2006.

[67] V. J. Prochazka and H. H. Kornhuber, "On-line multi-unit sorting with resolution of superposition potentials," *Electroencephalogr. Clin. Neurophysiol.*, vol. 34, no. 1, pp. 91–93, 1973.

[68] A. F. Atiya, "Recognition of multiunit neural signals," *IEEE Trans. Biomed. Eng.*, vol. 39, no. 7, pp. 723–729, 1992.

[69] S. Takahashi, Y. Anzai, and Y. Sakurai, "Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes," *J. Neurophysiol.*, vol. 89, no. 4, pp. 2245–2258, Apr. 2003.

[70] W. Ding and J. Yuan, "Spike sorting based on multi-class support vector machine with superposition resolution," *Med. Biol. Eng. Comput.*, vol. 46, no. 2, pp. 139–145, 2008.

[71] K. Y. Kwon, S. Eldawlatly, and K. G. Oweiss, "NeuroQuest: A comprehensive tool for large scale neural data processing and analysis," in *Proc. 4th Int. IEEE EMBS Conf. Neural Eng.*, Antalya, Turkey, 2009, pp. 622–625.

[72] G. D. Brown, S. Yamada, and T. J. Sejnowski, "Independent component analysis at the neural cocktail party," *Trends Neurosci.*, vol. 24, no. 1, pp. 54–63, 2001.

[73] E. M. Maynard, C. T. Nordhausen, and R. A. Normann, "The Utah intracortical array: A recording structure for potential brain–computer interfaces," *Electroencephalogr. Clin. Neurophysiol.*, vol. 102, pp. 228–239, 1997.

[74] F. Wood and M. J. Black, "A nonparametric Bayesian alternative to spike sorting," *J. Neurosci. Methods*, vol. 173, no. 1, pp. 1–12, Aug. 2008.

[75] V. Ventura, "Spike train decoding without spike sorting," *Neural Comput.*, vol. 20, no. 4, pp. 923–963, Apr. 2008.

[76] E. Stark and M. Abeles, "Predicting movement from multiunit activity," *J. Neurosci.*, vol. 27, no. 31, pp. 8387–8394, 2007.

[77] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain–machine interface for reaching and grasping by primates," *PLoS Biol.*, vol. 1, no. 2, p. E42, Nov. 2003.

[78] B. Pesaran, J. S. Pezaris, M. Sahani, P. P. Mitra, and R. A. Andersen, "Temporal structure in neuronal activity during working memory in macaque parietal cortex," *Nat. Neurosci.*, vol. 5, no. 8, pp. 805–811, 2002.

[79] N. F. Ince, R. Gupta, S. Arica, A. H. Tewfik, J. Ashe, and G. Pellizzer, "Movement direction decoding with spatial patterns of local field potentials," in *Proc. 4th Int. IEEE EMBS Conf. Neural Engineering,* Antalya, Turkey, Apr. 2009, pp. 291–294.

[80] C. Mehring, J. Rickert, E. Vaadia, S. C. De Oliveira, A. Aertsen, and S. Rotter, "Inference of hand movements from local field potentials in monkey motor cortex," *Nat. Neurosci.*, vol. 6, no. 12, pp. 1253–1254, Dec. 2003.

[81] H. Scherberger, M. R. Jarvis, and R. A. Andersen, "Cortical local field potential encodes movement intentions in the posterior parietal cortex," *Neuron*, vol. 46, no. 2, pp. 347–354, Apr. 2005.

[82] M. Mollazadeh, V. Aggarwal, N. V. Thakor, A. J. Law, A. Davidson, and M. H. Schieber, "Coherency between spike and LFP activity in M1 during hand movements," in *Proc. 4th Int. IEEE EMBS Conf. Neural Engineering*, Antalya, Turkey, Apr. 2009, pp. 506–509.

[83] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, no. 6810, pp. 361–365, 2000.

[84] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.