Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance

J. Neural Eng. 12 (2015) 016009 (10pp) doi:10.1088/1741-2560/12/1/016009

Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance

Breanne P Christie[1], **Derek M Tat**[1], **Zachary T Irwin**[1], **Vikash Gilja**[2,3,4], **Paul Nuyujukian**[5,6], **Justin D Foster**[7], **Stephen I Ryu**[7,8], **Krishna V Shenoy**[5,7,9,10], **David E Thompson**[1,11] and **Cynthia A Chestek**[1,12,13]

[1] Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI, USA
[2] Department of Computer Science, Stanford University, Stanford, CA, USA
[3] Department of Electrical & Computer Engineering, University of California, San Diego, La Jolla, CA, USA
[4] Neurosciences Program, University of California, San Diego, La Jolla, CA, USA
[5] Department of Bioengineering, Stanford University, Stanford, CA, USA
[6] School of Medicine, Stanford University, Stanford, CA, USA
[7] Department of Electrical Engineering, Stanford University, Stanford, CA, USA
[8] Department of Neurosurgery, Palo Alto Medical Foundation, Palo Alto, CA, USA
[9] Neurosciences Program, Stanford University, Stanford, CA, USA
[10] Department of Neurobiology, Stanford University, Stanford, CA, USA
[11] Department of Electrical and Computer Engineering, Kansas State University, Manhattan, KS, USA
[12] Neurosciences Program, University of Michigan, Ann Arbor, MI, USA
[13] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

E-mail: cchestek@umich.edu

Received 15 September 2014, revised 12 November 2014
Accepted for publication 17 November 2014
Published 11 December 2014

Abstract

Objective. For intracortical brain–machine interfaces (BMIs), action potential voltage waveforms are often sorted to separate out individual neurons. If these neurons contain independent tuning information, this process could increase BMI performance. However, the sorting of action potentials (‘spikes’) requires high sampling rates and is computationally expensive. To explicitly define the difference between spike sorting and alternative methods, we quantified BMI decoder performance when using threshold-crossing events versus sorted action potentials. *Approach.* We used data sets from 58 experimental sessions from two rhesus macaques implanted with Utah arrays. Data were recorded while the animals performed a center-out reaching task with seven different angles. For spike sorting, neural signals were sorted into individual units by using a mixture of Gaussians to cluster the first four principal components of the waveforms. For thresholding events, spikes that simply crossed a set threshold were retained. We decoded the data offline using both a Naïve Bayes classifier for reaching direction and a linear regression to evaluate hand position. *Main results.* We found the highest performance for thresholding when placing a threshold between −3 and $-4.5 \times V_{\mathrm{rms}}$. Spike sorted data outperformed thresholded data for one animal but not the other. The mean Naïve Bayes classification accuracy for sorted data was 88.5% and changed by 5% on average when data were thresholded. The mean correlation coefficient for sorted data was 0.92, and changed by 0.015 on average when thresholded. *Significance.* For prosthetics applications, these results imply that when thresholding is used instead of spike sorting, only a small amount of performance may be lost.

The utilization of threshold-crossing events may significantly extend the lifetime of a device because these events are often still detectable once single neurons are no longer isolated.

Keywords: spike sorting, threshold, brain–machine interface

(Some figures may appear in colour only in the online journal)

1. Introduction

Brain–machine interfaces (BMIs) translate electrophysiological signals into command signals for assistive technology [1, 2] to aid individuals with severe neurological disorders. This information can be acquired via multi-electrode arrays that penetrate 1–2 mm into the cortex. Research in this field has allowed nonhuman primates [3–8] and individuals with paralysis [1, 9–12] to use signals from the brain to move a computer cursor. Related research has also demonstrated that nonhuman primates [13] and people [14, 15] can control a robotic arm for self-feeding. In order to control these devices, action potentials from the motor cortex are used because they are associated with kinematic [16] and kinetic [17] movement parameters.

Spike sorting is a process that involves examining the voltage deflections on each recording electrode and differentiating the action potentials ('spikes') with distinctive waveforms. It is commonly used to analyze electrical signals from each electrode in order to determine which action potentials were emitted from a given neuron [18]. For BMI applications, this procedure is most useful on a subset of electrodes where neurons contain independent information [19], meaning they each fire preferentially for a specific target or task. However, if electrodes contain primarily one single neuron each or contain neurons with similar tuning, spike sorting would not be expected to provide an appreciable performance gain relative to other methods. In addition, our previous study found that although single neurons may remain above the noise for years, they become less discriminable from each other with time [20].

Spike sorting also presents several obstacles that make the transition of neural prosthetics from research to the clinic more difficult [21]. It requires high sampling rates (~30 000 s^{-1}) and is computationally expensive, which can increase the power consumption of an implantable device. While several groups have automated this process on large sets of recording electrodes [7, 19, 22, 23], automated algorithms are rarely applied in real-time because they involve software development and hardware resources that may not be readily available. For these reasons, it is currently unclear whether it is worth the time and resources required to spike sort.

There are currently several alternatives to spike sorting. Upcoming movement can be accurately predicted by using multiunit activity (MUA), which is estimated by band-pass filtering the data, eliminating extreme values, and computing the sample-by-sample root mean square (rms) voltages [24]. Cortical local field potentials (LFPs) are summation signals of excitatory and inhibitory dendritic potentials that are also commonly used to predict movement [25]. Another method, thresholding, has become an increasingly popular alternative to spike sorting. This process involves setting a threshold for every electrode channel and detecting each time the voltage potential exceeds the threshold. This procedure ignores whether the action potential activity results from more than one neuron [8, 20, 21]. It is advantageous because it requires no human intervention and it is neither time-consuming nor computationally expensive. If performance is still sufficiently high, it could be beneficial for chronically implanted systems because it would allow for lower per-channel power requirements. This could be used either to reduce power supply constraints or to add additional channels with the same power supply.

Several studies have shown that complex movements can be decoded without the detection of isolated single units. A recent closed-loop study used threshold-crossing events obtained multiple years after implantation, when spikes were not well isolated [8, 20, 26]. They used a new algorithm, the ReFIT-Kalman filter, which performed 2.5x better than the original velocity-KF algorithm and 86% as well as the real hand for center-out-and-back cursor control [8]. Stark and Abeles demonstrated that it is possible to predict upcoming movement accurately with a small number of electrodes by utilizing thresholded spikes and by estimating MUA [24]. Related research by Fraser and colleagues has also suggested that thresholding is comparable to spike sorting [21]. In their study, four data sets from one day of online experimental testing demonstrated that thresholded data could sometimes outperform spike sorted data. On the other hand, a recent study by Todorova *et al* concluded that basic spike sorting outperforms low-threshold waveform-crossing methods [27]. Because these disagreeing studies employ different threshold placement techniques, a general conclusion about the performance of sorted data versus unsorted data cannot yet be reached. To gain a better understanding of these conflicting results, it will be necessary to evaluate a wide range of thresholds and the contributions of different neuron subpopulations in a large number of data sets.

In this study, we quantified the differences in performance between spike sorting and thresholding in two animals using 58 daily data sets. Other studies have analyzed the performance of different spike sorting methods [28], MUA [24, 28], and LFPs [24, 28–36]. To our knowledge, this is the first study that uses a comprehensive data set to find an optimal threshold level in addition to quantifying the differences in decoder performance between spike sorting and thresholding. It is also one of the few studies to compare the decoder performances of single neurons, single neurons with minimal contamination from noise, multiple neurons, and combinations of these unit types. We used Naïve Bayes classification and linear regression to analyze the performance of single neurons versus multiple neurons, rms voltage thresholds, fixed voltage thresholds, and spike sorting only a subset of channels. When partially spike sorting, we also

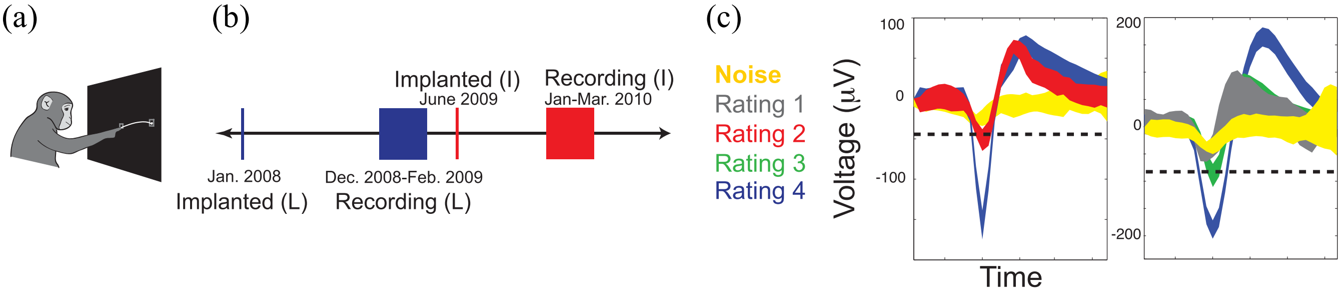

Figure 1. (a) A monkey performing the center-out reaching task on a fronto-parallel screen in complete darkness. (b) Timeline of the implantation and recording from Utah arrays. (c) A visual explanation of how waveforms are classified during spike sorting.

evaluated the characteristics of channels that gave the highest and lowest performances when spike sorted.

2. Methods

2.1. Data collection

The Stanford University Institutional Animal Care and Use Committee approved all protocols. The data and training methods used in this study are the same as in [20]. Two rhesus macaques, L and I, performed a center-out reaching task to 28 potential visual targets, with four distances and seven angles. These were collapsed into seven classified angles in our analysis. This task was performed on a fronto-parallel screen in complete darkness (figure 1(a)). Hand movements were optically tracked with reflective beads placed on the distal joint of the index or middle finger. Hand movements were acquired at a rate of 60 samples s^{-1} (nominal submillimeter resolution) using a Polaris system (NDI, Waterloo, Ontario, Canada). The task was sequenced using Tempo software (Reflective Computing, St. Louis, MO). The animals were trained for many months prior to the present study; therefore, it is likely that little to no learning occurred. Experiments typically lasted 1–5 h and animals were given a juice reward for successful trials.

One array was implanted in PMd/M1 of each animal via standard neurosurgical techniques. The data were recorded using silicon 'Utah' arrays (Blackrock Technologies, Salt Lake City, UT, USA) made up of 100 microelectrodes on a 10×10 grid with 400 μm center-to-center spacing. When data collection started, Monkey L and I's arrays were 11 and 7 months old with substantial single unit activity. Data were recorded in the rig using a Cerebus system (Blackrock Microsystems, Salt Lake City, UT) over six and eight weeks in Monkeys L and I (figure 1(b)). Sixty-nine daily data sets were acquired during these time periods, but just 58 were analyzed because we only used data sets with over 100 reaches per target.

Blackrock's built-in function was used for calculating the rms voltage, which is slightly different than the standard calculation. Their algorithm calculates a biased estimate of the rms voltage of the noise of the spike data stream. In the equations below, s_i represents the 30 kHz sampled spike stream after it is filtered, and x_i represents the mean-squared value. Equation (1) calculates one V_{rms} value for a set of 600 continuous samples. It does this for two seconds, which results in 100 mean-squared values. In equation (2), the lowest five of these values are discarded and the next 20 are used to calculate the final rms voltage. The rationale for this method is to avoid inflating the V_{rms} value due to the presence of large artifacts.

$$\left\{ x_j \,\middle|\, x_j = \frac{\sum_{i=1}^{600} s_i^2}{600} \right\}_{j=1\rightarrow 100} \tag{1}$$

$$\mathrm{rms} = \sqrt{\frac{\sum_{i=6}^{25} \min_i \{x_j\}}{20}} \tag{2}$$

$$\min_i \{x\} = i\text{th minimum of } \{x_j\} \tag{3}$$

2.2. Spike sorting technique

We used the spike sorting methods described in [19, 20, 37] to visualize and identify the action potentials associated with different neurons. All neural data were analyzed offline in MATLAB (MathWorks, Natick, MA, USA). Although a closed-loop study is currently the gold standard, we performed an open-loop analysis because we wanted to analyze the same data using different techniques. Briefly, the data were high-pass filtered in order to eliminate local field fluctuations. When the signal crossed a threshold relative to its rms voltage, a short voltage snippet was recorded before and after the threshold crossing. Very large 'juicer' artifacts were also removed by eliminating 'units' that stayed low for ~1 ms after the triggering peak. These rare inter-trial events would not have impacted decoder performance, but were problematic for another study using this data that involved identifying the largest single unit [20]. Next, the remaining data were shifted in time in order to align all of the peaks. The data were then noise-whitened and the top four principle components of the waveform snippets were obtained. A mixture-of-Gaussian model was fit to the data by a 'relaxation' variant of expectation–maximization, which reduces the chances of converging to local maxima [19, 37]. We manually rated the unit quality when forming our spike sorted data sets. The waveforms seen in every channel were categorized based on their shape, amplitude, and principal components (figure 1(c)). We classified waveforms as presumed single neurons (category '4'), single units with minimal contamination by other units (category '3'), and multi-units (category '2' or '1'). Category 1 units are likely contaminated by noise but still appear vaguely neural. Category 2 waveforms are more visibly neural, but they are depicted by a centroid that is not clearly delineated from the central 'hash' in principal component analysis (PCA). When analyzing the principal components, category 3 waveforms have a clearly delineated centroid with some overlapping units in the outer ring. This unit categorization did not occur when creating the thresholded data sets. For thresholded data sets, all waveforms that surpassed a set threshold level were analyzed regardless of their apparent quality. The same electrodes were used in both analyses.

To demonstrate how decoder performance would change if action potentials were sorted using other sophisticated techniques, we also performed the unsupervised, wavelet spike sorting algorithm described by Quiroga *et al* [38]. Each spike was decomposed using a 4-level Haar wavelet transform. Next, ten wavelet coefficients with the largest deviations from normality were selected from each channel based on a modified Kolmogorov–Smirnov test. These were then clustered using the Potts model superparamagnetic clustering (SPC) method. We used the software provided by Quiroga *et al* to perform this analysis (www.vis.caltech.edu/~rodri/Wave_clus/Wave_clus_home.htm).

To detect neural units, we initially spike sorted each of the 96 channels using the automated wavelet sorting algorithm. Afterwards, we manually adjusted the parameters of the SPC to further improve our sort results based on visual inspection of the similarity of clustered waveforms. The SPC was constrained to yield no more than 8 clusters per channel, but the typical number of discerned clusters was 2–4. We applied this algorithm to the first day of recording in Monkey I and the first two days of recording in Monkey L.

2.3. Neural data analysis

The same two types of offline neural decoders used in [20] were also applied in this study. We used Naïve Bayes classification as a discrete neural decoder to calculate how accurately we could predict target selection [7]. We modeled the firing rate of each channel during a 500 ms time window after target presentation as a Poisson distribution. The likelihood of each target was calculated using equations (4) and (5), where Θ is the actual target angle, θ is the predicted angle, y_n is the number of spikes that occurred on neuron n, $\vec{Y}$ represents a vector of observed spikes, and N is the number of neurons.

$$P\left(\vec{Y}\middle|\Theta=\theta\right)=\prod_{n=1}^{N}\frac{\lambda_{n,\theta}^{y_n}\mathrm{e}^{-\lambda_{n,\theta}}}{y_n!} \tag{4}$$

$$P\left(\Theta=\theta\mid\vec{Y}\right)=\frac{P\left(\vec{Y}\mid\Theta=\theta\right)P(\Theta=\theta)}{P(\vec{Y})}\propto P\left(\vec{Y}\mid\Theta=\theta\right) \tag{5}$$

The target that maximized this likelihood was selected. We trained and tested this model within the same day using ten-fold cross validation. We evaluated the performance using a percent correct variable that indicates how often we were able to correctly predict which target was selected.

In addition, we used a continuous offline linear decoder to predict the animal's hand position [39]. For each trial, the data were divided into 100 ms bins and we found the average firing rate and hand position for every bin. We also used the firing rate information given at ten sequential 100 ms time lags. Using a linear Wiener filter, we modeled hand position as a function of firing rate for every channel. In equation (6), matrix X includes the average horizontal and vertical position for every bin. Every row of matrix Y contains the firing rates for each neuron at the ten sequential time lags. Matrix B is calculated by linear regression and is the resulting linear decoder.

$$\begin{gathered}X=YB\\ \begin{bmatrix}\mathrm{pos}_x(1) & \mathrm{pos}_y(1)\\ \cdots & \cdots\\ \mathrm{pos}_x(t) & \mathrm{pos}_y(t)\end{bmatrix}=\begin{bmatrix}\mathrm{neur}_1\mathrm{lag}_1(1) & \mathrm{neur}_1\mathrm{lag}_2(1) & \ldots & \mathrm{neur}_N\mathrm{lag}_{10}(1)\\ \cdots & \cdots & \cdots & \cdots\\ \mathrm{neur}_1\mathrm{lag}_1(t) & \ldots & \ldots & \mathrm{neur}_N\mathrm{lag}_{10}(t)\end{bmatrix}\\ \times\begin{bmatrix}B_{x,\mathrm{neur1lag}_1} & B_{y,\mathrm{neur1}l\mathrm{ag}_1}\\ \cdots & \cdots\\ B_{x,\mathrm{neur}N\mathrm{lag}_{10}} & B_{y,\mathrm{neur}N\mathrm{lag}_{10}}\end{bmatrix}\end{gathered} \tag{6}$$

We trained and tested this model within the same day using 20-fold cross validation. Continuous performance is given using a correlation coefficient between predicted and actual hand position.

In order to validate the accuracy of the linear regression decoder, we also measured the 'mean distance to the target' during a decoded reach and compared the two metrics in the threshold analyses. The mean distance to target metric represents the distance to the peripheral target averaged across all of the timesteps during the reach [40]. An optimal reach with uniform speed will have a mean distance to target equal to half the total distance.

3. Results

3.1. Neural data unit quality and performance

To ensure that the data we are analyzing are representative of common data sets, we evaluated several characteristics. Figure 2 displays the spike panels on day 1 for both animals and table 1 gives the distribution of unit types. Both animals show a similar trend, in that most of the waveforms contain a mixture of multiple units and the fewest represent single units. For Monkey L, 40% percent of sorted waveforms are noisy category 1 waveforms and only 19% are single units. 47% percent of Monkey I's waveforms are category 1 and only 12% are single units. These unit distributions are typical for a 'good' array. Figure 3(a) displays the rms voltages across all data sets. Monkey L's average rms voltage was 14.53 μV and Monkey I's was 9.22 μV.

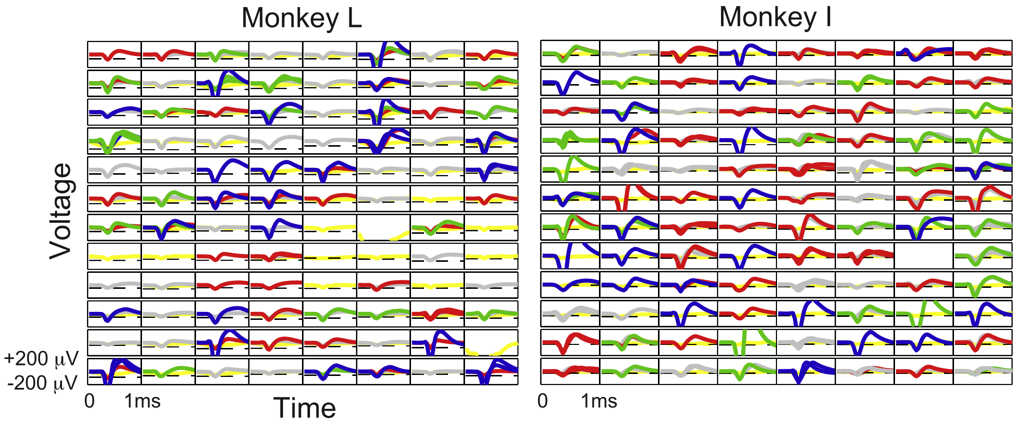

Figure 2. Spike panels of Monkey L (left) and I (right) on the first day of recording. Waveform colors correspond to the legend shown in figure 1(c) (single units are in blue, green waveforms represent single units with minimal contamination by other units, multi-units are in red, gray waveforms are multiple units with potential noise contamination, and noise is yellow). Sample thresholds are placed at $-4.5 \times V_{rms}$ for Monkey L and $-3 \times V_{rms}$ for Monkey I.

Table 1. Distribution of waveforms on day 1.

	Number of waveforms	
Category	Monkey L	Monkey I
1	60 (39.47%)	102 (47.00%)
2	39 (25.66%)	59 (27.19%)
3	24 (15.79%)	31 (14.29%)
4	29 (19.08%)	25 (11.52%

Figure 3(b) displays both decoders' performances when using different subsets of spike sorted data. When analyzing spike sorted data, channels with low firing rates were not removed. Because there were likely large low-firing units, performance was higher when they were included. Their presence ensured that we did not specifically disadvantage spike sorting when comparing it to thresholding. Category 2 waveforms, which likely correspond to multiple neurons, gave the highest performance when isolated and analyzed compared to other individual categories. In addition, the performance values when using categories 2 and 3 were only minimally improved when the single units were added into the analysis, though the improvements were statistically significant (t-tests, $p < 0.01$). When units with ratings of 2 through 4 were analyzed, they demonstrated the best overall performance with the smallest standard deviation. The Naïve Bayes percent correct was 94% for Monkey L and 83% for Monkey I. Correlation coefficients were 0.94 (L) and 0.90 (I). This subset of spike sorted data is what we used as a comparison for the remaining analyses.

3.2. Best rms threshold for decoding

It is common practice for researchers to use the rms voltage to set a threshold [8, 21, 41, 42]. To quantitatively determine the best placement, we tested the decoder performances of different threshold levels from $-3 \times V_{rms}$ to $-18 \times V_{rms}$ (figure 4(a)). Channels with a firing rate <1 Hz were removed. For the Naïve Bayes decoder, the best threshold level was $-4.5 \times V_{rms}$ for Monkey L and $-3 \times V_{rms}$ for Monkey I. These thresholds produced Naïve Bayes percent correct values of 90% (L) and 89% (I), with chance ≈14%. The best correlation coefficient was seen at $-4 \times V_{rms}$ for Monkey L, giving $\rho = 0.92$. For Monkey I, it was seen at $-3 \times V_{rms}$ and produced $\rho = 0.89$. We used the mean distance to target metric to corroborate the trends seen with correlation coefficient, and we found that both metrics displayed similar performance trends. Monkey L's optimal threshold level was higher possibly because the array was older and may have contained more noise. For Monkey I, spike sorted data performance was slightly lower than that seen with threshold-crossing events. In addition, Monkey I's threshold analyses did not contain a peak. The $-3 \times V_{rms}$ threshold value, which was approximately $-28\,\mu V$ on average, was initially set during recording and may not have been sufficiently low for this newer array. For Monkey L, the decoder performances of optimally thresholded data are only slightly lower than the spike sorted data, though the differences are statistically significant ($p < 0.001$). For Monkey I, the Naïve Bayes performance of thresholded data is statistically significantly higher than that of the spike sorted data ($p = 2.4 \times 10^{-11}$), but the linear regression performance is not significantly different ($p = 0.908$).

3.3. Best fixed voltage threshold for decoding

Some chip designs use a fixed voltage threshold for every channel instead of using the channels' rms voltages. This approach may work well if the mean and standard deviation of the rms voltage are fairly tight. To test how our data sets would perform using this technique, and to determine which fixed voltage threshold would maximize performance, we tested the decoders for thresholds between $-10\,\mu V$ and $-200\,\mu V$. Channels with a firing rate <1 Hz were removed. The performances are given in the bottom row of figure 4. For the Naïve Bayes decoder, Monkey L had the best performance at $-70\,\mu V$, producing PC = 88%. Monkey I was best at $-10\,\mu V$, the lowest possible value, giving PC = 89%. The best correlation coefficient was seen at $-50\,\mu V$ for Monkey L and

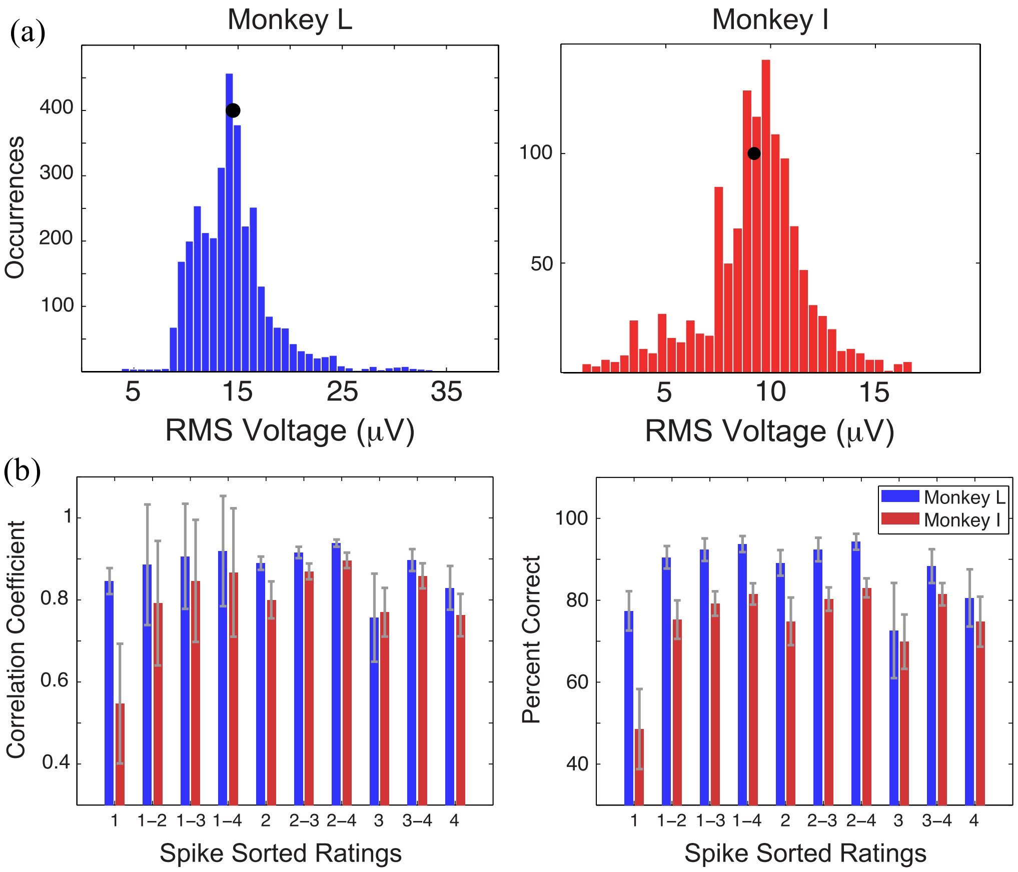

Figure 3. (a) Histograms of the rms voltages across all electrode channels with firing rate >1 Hz over all data sets. The black dot represents that animal's mean rms voltage. (b) We investigated how classes of waveforms performed individually, and then we combined waveform classes in order to determine which subset of spike sorted data would give the highest performance. Ratings of 4 represent single units and ratings of 1 contain a mixture of multiple units that is likely contaminated by noise. The gray error bars represent standard deviation.

$-10\,\mu$V for Monkey I. This generated $\rho = 0.92$ (L) and $\rho = 0.90$ (I). The optimal thresholds in the linear regression analysis matched those seen in the mean distance to target analysis. For both animals, the performances associated with threshold levels of -10 to $-30\,\mu$V were nearly identical due to the threshold level that was initially set during recording. For Monkey L, the performances of optimally thresholded data are only slightly lower than the spike sorted data, though the differences are statistically significant ($p < 0.001$). For Monkey I, the Naïve Bayes performance of thresholded data is statistically significantly higher than that of the sorted data ($p = 2.7E - 11$), though the linear regression performance is not significantly different ($p = 0.912$).

3.4. Partial spike sorting

If decoder performance is impaired when thresholding is used instead of spike sorting, it may be possible to spike sort a subset of the data to regain that performance. To determine which channels were best to spike sort, we tested the individual decoder performances of each channel on day 1. For every electrode, performance was measured when thresholded data were removed and replaced with the corresponding spike sorted data. We then ranked those channels by which ones gave the highest improvement [43] on day 1, and tested them on day 2 and beyond. We measured the performance after spike sorting the best channel, second best channel, etc on the remaining data sets. For the partial spike sorting analysis, channels with low firing rates were not removed. The channels with low firing rates are different from day to day, which would interfere with our ability to rank and test the same electrodes. This analysis is primarily used as a tool to identify the channels in both animals that were most affected by spike sorting or thresholding.

Figure 5(a) demonstrates that as more electrodes were spike sorted, the decoder performances increased for Monkey L. These trends were anticipated based on our previous finding that sorted data outperformed thresholded data for this animal. The results from Monkey I are not displayed because the thresholded data outperformed sorted data, and this analysis is primarily focused on regaining performance lost during thresholding. If 26 of Monkey L's channels were spike sorted, the linear regression performance was no longer statistically different than the fully spike sorted data ($p = 0.06$). After spike sorting only six channels, the correlation coefficient was within 1% of that seen when fully spike sorting. On the other hand, 76 channels had to be sorted until the achieved Naïve Bayes performance was not significantly different ($p = 0.051$). Spike sorting 61 channels brought the Naïve

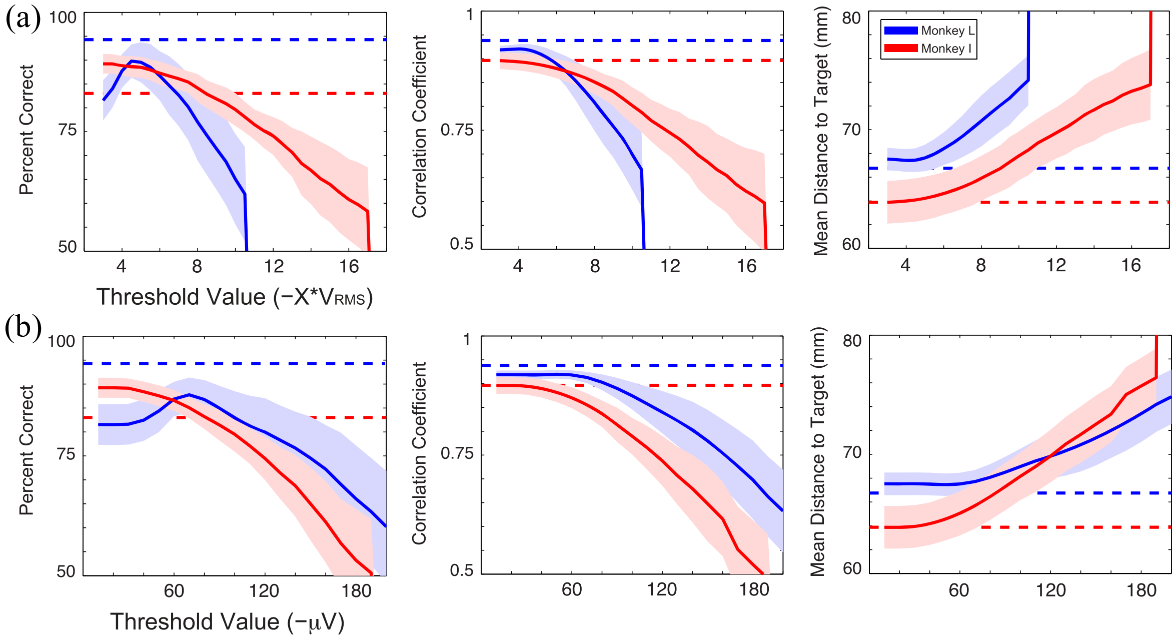

Figure 4. On all figures, Monkey L is shown in blue and Monkey I is shown in red. The shaded region represents standard deviation. The dashed lines depict the decoder performance of the spike sorted data, specifically waveforms with ratings 2–4. This subset of data was previously found to give the highest sorted performance while likely ensuring that noise contamination is removed. (a) The top row demonstrates the performances of three decoders at varying V_{rms} threshold levels. (b) The bottom row represents the performance at varying fixed voltage threshold levels.

Bayes accuracy within 1% of the performance seen when fully spike sorting for Monkey L.

When performing the partial spike sorting analysis, channels were ranked based on which gave the highest improvement in decoder performance on day 1 when its thresholded data was replaced with sorted data. In addition to the above analysis, we also analyzed the properties of the four best and worst channels to spike sort in both animals. As expected, the best channels to spike sort included multiple units with independent tuning properties. On the other hand, the worst channels to spike sort indicated that manual aspects of our spike sorting technique were occasionally flawed. For example, waveforms without a distinct bipolar shape were removed during spike sorting, yet they occasionally contained useful information. In addition, it is nearly impossible to visually distinguish between useful hash and noise. Two units would appear extremely similar, yet only one of them led to an improvement in decoder performance when spike sorted. Research by Wood and colleagues supports this claim, finding an average error rate of 23% false positive and 30% false negative regarding the manual sorting of synthetic data [28]. Finally, we found that the automated portion of our spike sorter occasionally broke a single good unit into multiple spikes, which may have made the signal noisier and less useful.

3.5. Wavelet-based spike sorting

To fairly represent the performance of other spike sorting algorithms, we performed an unsupervised, wavelet spike sorting algorithm [38]. Unlike PCA, wavelet coefficients offer better discrimination of the temporal features of action potentials due to the time scaling property of the wavelet decomposition. Moreover, SPC offers a more general clustering method, neither requiring non-overlapping clusters in the feature space, nor assuming a particular distribution (such as our original Gaussian expectation–maximization clustering method) [38]. We tested the wavelet-sorted algorithm on three data sets to ensure a good approximation. As figure 5(b) demonstrates, the differences in decoder performance between the wavelet-sorted data and our PCA-clustered data are very small. Therefore, we concluded that it was acceptable to perform PCA-based spike sorting methods in our analyses.

4. Discussion

We have demonstrated that threshold-crossing events can be used to analyze BMI data instead of spike sorting without causing a substantial decrease in performance. The Naïve Bayes percent correct changed by 5% on average and the correlation coefficient changed by 0.015. Based on the quality and performance of our neural data, it appears that it is representative of that commonly seen for BMIs. Our results demonstrate that thresholds between −3 and $-5 \times V_{rms}$ result in similar performances. While this study contains a limited number of arrays, we hypothesize that the best rms threshold to place for arrays with smaller noise is around $-3 \times V_{rms}$. For arrays with larger noise, the best threshold may be approximately $-4.5 \times V_{rms}$. It is not possible to determine a fixed voltage threshold that will work on every data set, because characteristics vary between subjects. A fixed voltage

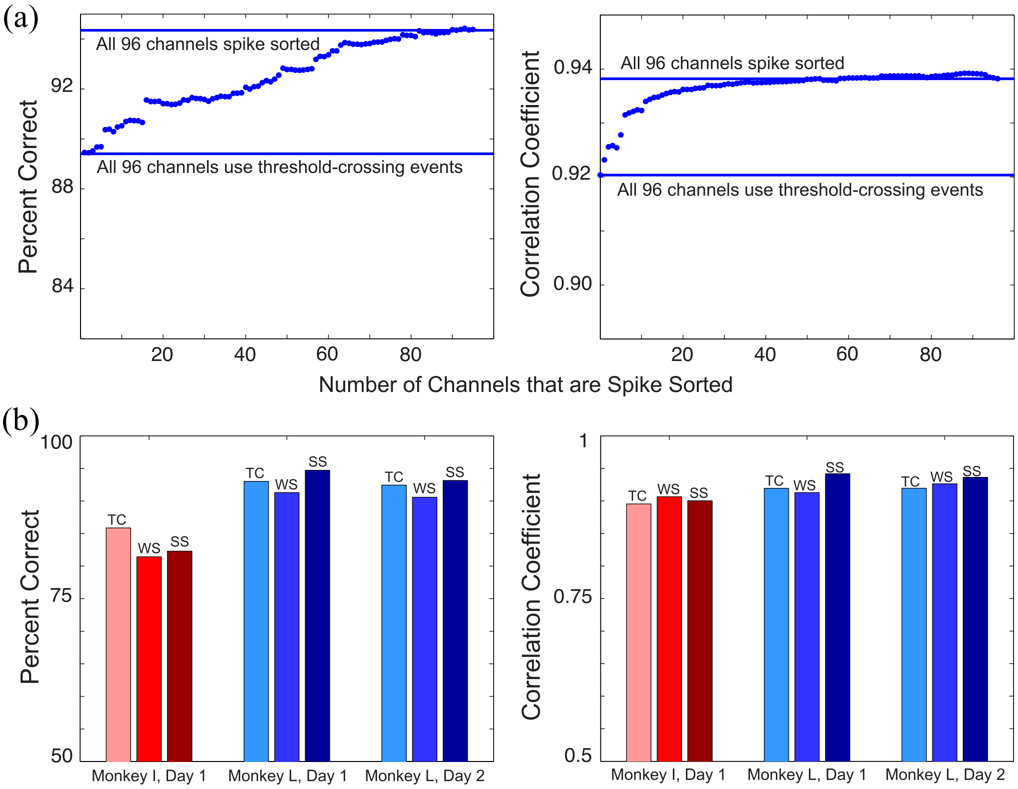

Figure 5. (a) Decoder performance when a subset of channels is spike sorted and the remainder is thresholded at that animal's optimal rms threshold. In each graph, one solid line represents the performance when all channels are thresholded and the other line represents spike sorting all channels on day 2 and beyond. The data points in between the lines represent the performance seen when sorting a particular number of channels. (b) Decoder performance for each method of data processing. TC represents data that use threshold-crossing events, WS symbolizes the wavelet sorted data, and SS characterizes the data sorted with our spike sorting algorithms.

threshold generally gives lower performance and also requires the user to manually tune the threshold.

The decoder performances of Monkey L's data are indicative of the common notion that spike sorted data performance is slightly higher than that of thresholded data, although the Naïve Bayes accuracy is only 4% greater and the correlation coefficient is just 0.02 higher. While it still requires human intervention, our partial spike sorting analysis demonstrates that any performance lost due to thresholding can be regained after spike sorting a small subset of channels. The waveform characteristics of the best and worst channels to sort did not display any consistent trends, so it is not possible for us to say exactly which traits to look for when choosing a subset of channels to spike sort. Although spike sorting was favorable when there were multiple units with independent tuning properties, tuning properties cannot be determined upon inspection only. In addition, it is nearly impossible to visually distinguish between useful hash and noise. An algorithm by Ventura [44] would be useful in a partial spike sorting analysis, because it determines if units should be unsorted or sorted based on what leads to the highest decoder performance.

On the other hand, the spike sorted and thresholded data in Monkey I showed dissimilar results. The correlation coefficient was not significantly different. In addition, for Naïve Bayes classification, the thresholded data performance was actually higher than that of sorted data. This is possibly due to unnecessary unit splitting by the automated portion of the spike sorter.

Opposite results between two animals also occurred in a study performed by Fraser and colleagues, where spike sorted data outperformed unsorted data in one case but not the other [21]. Our results are consistent with the studies by Gilja *et al* and Stark and Abeles, in that we demonstrate that complex movements can be decoded without the detection of spikes [8, 24]. However, the recent study published by Todorova *et al* concluded that spike sorting adds value to the threshold-crossing methods employed in BMI decoding [27]. In their study, threshold levels were set in order to maximize the ability to perform spike sorting in Monkey A. Their figure 2 shows that many of the thresholds for this animal were higher in amplitude than the optimal levels we saw, which would tend to hinder the decoder performance of thresholded data (as seen in our figure 4). They demonstrated that the optimal linear estimator (OLE) amplitude-sorted data from Monkey A led to ~12% performance gain compared to OLE unsorted data. In our study, with Monkey L, a threshold of $-6.5 \times V_{\mathrm{rms}}$ can also cause the sorted data to outperform thresholded data by 12%. The data from the other animal in their study was thresholded at three standard deviations below the mean of the band-pass filtered voltage traces, which led to threshold levels more similar to ours. Subsequently, the OLE amplitude-sorted data only outperformed OLE unsorted data by ~6%, which was more consistent with Monkey L's results from our study. Our study concludes that if you sweep the threshold systematically, it is possible to find a range where threshold crossings perform essentially as well as spike sorting. The use of threshold crossings instead of spike sorting will make it easier to build implantable prosthetics processors and chips with threshold-crossing detectors [45].

It is important to note that our analysis was performed offline, and our sorting method was not fully automated. Other studies, such as that by Todorova *et al*, found similar results across more decoders. Our conclusions do not hold if there is a large amount of noise or motion artifacts, although it is rare for these to hinder threshold crossings while leaving units intact. In fact, noise artifacts tend to be lower in wireless systems that often have very short wires and good shielding [45]. To our knowledge, this is the first study to determine an optimal threshold level in addition to analyzing the contributions of different neuron subpopulations in a large number of data sets.

With current microelectrode array technology, our study demonstrates that thresholding and spike sorting can be used essentially interchangeably without seeing a significant decrease in performance. The utilization of threshold-crossing events may allow for an easier transition of BMI devices from research into the clinic due to simpler hardware and software. This also allows for long-term systems with lower power consumption and it can be applied to systems with a high number of electrode channels.

Acknowledgments

This work was supported by National Science Foundation graduate research fellowships (CAC, VG), NDSEG Fellowship (VG), Texas Instruments Stanford Graduate Fellowship (JF), Burroughs Wellcome Fund Career Awards in the Biomedical Sciences (KVS), the Christopher Reeve Paralysis Foundation (SIR, KVS), a Stanford University Graduate Fellowship (CAC), a Stanford NIH Medical Scientist Training Program grant and Soros Fellowship (PN) and the following awards to KVS: Stanford CIS, Sloan Foundation, NIH CRCNS R01-NS054283, McKnight Endowment Fund for Neuroscience, NIH Director's Pioneer Award DP1HD075623, DARPA Revolutionizing Prosthetics program contract N66001-06-C-8005, and DARPA REPAIR contract N66001-10-C-2010.

References

[1] Wolpaw J R, Birbaumer N, McFarland D J, Pfurtscheller G and Vaughan T M 2002 'Brain–computer interfaces for communication and control *Clin. Neurophysiol.* **113** 767–91

[2] Schwartz A B 2004 Cortical neural prosthetics *Annu. Rev. Neurosci.* **27** 487–507

[3] Serruya M D, Hatsopoulos N G, Paninski L, Fellows M R and Donoghue J P 2002 Brain–machine interface: instant neural control of a movement signal *Nature* **416** 141–2

[4] Taylor D M 2002 Direct cortical control of 3D neuroprosthetic devices *Science* **296** 1829–32

[5] Carmena J M, Lebedev M A, Crist R E, O'Doherty J E, Santucci D M, Dimitrov D F, Patil P G, Henriquez C S and Nicolelis M A L 2003 Learning to control a brain–machine interface for reaching and grasping by primates *PLoS Biol.* **1** e2

[6] Musallam S 2004 Cognitive control signals for neural prosthetics *Science* **305** 258–62

[7] Santhanam G, Ryu S I, Yu B M, Afshar A and Shenoy K V 2006 A high-performance brain–computer interface *Nature* **442** 195–8

[8] Gilja V *et al* 2012 A high-performance neural prosthesis enabled by control algorithm design *Nat. Neurosci.* **15** 1752–7

[9] Hochberg L R, Serruya M D, Friehs G M, Mukand J A, Saleh M, Caplan A H, Branner A, Chen D, Penn R D and Donoghue J P 2006 Neuronal ensemble control of prosthetic devices by a human with tetraplegia *Nature* **442** 164–71

[10] Kennedy P R, Bakay R A, Moore M M, Adams K and Goldwaithe J 2000 Direct control of a computer from the human central nervous system *IEEE Trans. Rehabil. Eng.* **8** 198–202

[11] Leuthardt E C, Schalk G, Wolpaw J R, Ojemann J G and Moran D W 2004 A brain–computer interface using electrocorticographic signals in humans *J. Neural Eng.* **1** 63–71

[12] Patil P G, Carmena J M, Nicolelis M A L and Turner D A 2004 Ensemble recordings of human subcortical neurons as a source of motor control signals for a brain–machine interface *Neurosurgery* **55** 27–35 discussion 35–38

[13] Velliste M, Perel S, Spalding M C, Whitford A S and Schwartz A B 2008 Cortical control of a prosthetic arm for self-feeding *Nature* **453** 1098–101

[14] Collinger J L, Wodlinger B, Downey J E, Wang W, Tyler-Kabara E C, Weber D J, McMorland A J, Velliste M, Boninger M L and Schwartz A B 2013 High-performance neuroprosthetic control by an individual with tetraplegia *Lancet* **381** 557–64

[15] Hochberg L R *et al* 2012 Reach and grasp by people with tetraplegia using a neurally controlled robotic arm *Nature* **485** 372–5

[16] Georgopoulos A P, Kalaska J F, Caminiti R and Massey J T 1982 On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex *J. Neurosci.* **2** 1527–37

[17] Evarts E V 1966 Pyramidal tract activity associated with a conditioned hand movement in the monkey *J. Neurophysiol.* **29** 1011–27

[18] Lewicki M S 2009 A review of methods for spike sorting: the detection and classification of neural action potentials *Network: Comp. Neural Syst.* **9** R53–78

[19] Santhanam G, Sahani M, Ryu S I and Shenoy K V 2004 An extensible infrastructure for fully automated spike sorting during online experiments *26th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society* vol 2, pp 4380–4

[20] Chestek C A *et al* 2011 Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex *J. Neural Eng.* **8** 045005

[21] Fraser G W, Chase S M, Whitford A and Schwartz A B 2009 Control of a brain–computer interface without spike sorting *J. Neural Eng.* **6** 055004

[22] Zumsteg Z S, Kemere C, O'Driscoll S, Santhanam G, Ahmed R E, Shenoy K V and Meng T H 2005 Power feasibility of implantable digital spike sorting circuits for neural prosthetic systems *IEEE Trans. Neural Syst. Rehabil. Eng.* **13** 272–9

[23] Dethier J, Nuyujukian P, Ryu S I, Shenoy K V and Boahen K 2013 Design and validation of a real-time spiking-neural-

network decoder for brain–machine interfaces *J. Neural Eng.* **10** 036008

[24] Stark E and Abeles M 2007 Predicting movement from multiunit activity *J. Neurosci.* **27** 8387–94

[25] Scherberger H, Jarvis M R and Andersen R A 2005 Cortical local field potential encodes movement intentions in the posterior parietal cortex *Neuron* **46** 347–54

[26] Wang D, Zhang Q, Li Y, Wang Y, Zhu J, Zhang S and Zheng X 2014 Long-term decoding stability of local field potentials from silicon arrays in primate motor cortex during a 2D center out task *J. Neural Eng.* **11** 036009

[27] Todorova S, Sadtler P, Batista A, Chase S and Ventura V 2014 To sort or not to sort: the impact of spike-sorting on neural decoding performance *J. Neural Eng.* **11** 056005

[28] Wood F, Black M J, Vargas-Irwin C, Fellows M and Donoghue J P 2004 On the variability of manual spike sorting *IEEE Trans. Biomed. Eng.* **51** 912–8

[29] Markowitz D A, Wong Y T, Gray C M and Pesaran B 2011 Optimizing the decoding of movement goals from local field potentials in macaque cortex *J. Neurosci.* **31** 18412–22

[30] Bansal A K, Truccolo W, Vargas-Irwin C E and Donoghue J P 2012 Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials *J. Neurophysiol.* **107** 1337–55

[31] Flint R D, Lindberg E W, Jordan L R, Miller L E and Slutzky M W 2012 Accurate decoding of reaching movements from field potentials in the absence of spikes *J. Neural Eng.* **9** 046006

[32] Flint R D, Ethier C, Oby E R, Miller L E and Slutzky M W 2012 Local field potentials allow accurate decoding of muscle activity *J. Neurophysiol.* **108** 18–24

[33] Flint R D, Wright Z A, Scheid M R and Slutzky M W 2013 Long term, stable brain–machine interface performance using local field potentials and multiunit spikes *J. Neural Eng.* **10** 056005

[34] Mehring C, Rickert J, Vaadia E, de Oliveira S C, Aertsen A and Rotter S 2003 Inference of hand movements from local field potentials in monkey motor cortex *Nat. Neurosci.* **6** 1253–4

[35] Zhuang J, Truccolo W, Vargas-Irwin C and Donoghue J P 2010 Decoding 3D reach and grasp kinematics from high-frequency local field potentials in primate primary motor cortex *IEEE Trans. Biomed. Eng.* **57** 1774–84

[36] Hwang E J and Andersen R A 2013 The utility of multichannel local field potentials for brain–machine interfaces *J. Neural Eng.* **10** 046005

[37] Maneesh S 1999 Latent variable models for neural data analysis *Dissertation* California Institute of Technology

[38] Quiroga R, Nadasdy Z and Ben-Shaul Y 2004 Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering *Neural Comput.* **16** 1661–87

[39] Rao Y N, Kim S-P, Sanchez J C, Erdogmus D, Principe J C, Carmena J M, Lebedev M A and Nicolelis M A 2005 Learning mappings in brain–machine interfaces with echo state networks *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'05)* vol 5 pp 233

[40] Cunningham J P, Nuyujukian P, Gilja V, Chestek C A, Ryu S I and Shenoy K V 2011 A closed-loop human simulator for investigating the role of feedback control in brain–machine interfaces *J. Neurophysiol.* **105** 1932–49

[41] Gilja V, Pandarinath C, Blabe C H, Hochberg L R, Shenoy K V and Henderson J M 2013 Design and application of a high performance intracortical brain computer interface for a person with amyotrophic lateral sclerosis *Neuroscience Meeting Planner Program No. 80.06. 2013 (San Diego, CA)*

[42] Pandarinath C, Gilja V, Blabe C, Jarosiewicz B, Perge J A, Hochberg L R, Shenoy K V and Henderson J M 2014 High-performance communication using neuronal ensemble recordings from the motor cortex of a person with ALS *American Society for Stereotactic and Functional Neurosurgery (Washington, DC)*

[43] Fan J M, Nuyujukian P, Kao J C, Chestek C A, Ryu S I and Shenoy K V 2014 Intention estimation in brain–machine interfaces *J. Neural Eng.* **11** 016004

[44] Ventura V 2008 Spike train decoding without spike sorting *Neural Comput.* **20** 923–63

[45] Chestek C, Gilja V, Nuyujukian P, Kier R J, Solzbacher F, Ryu S, Harrison R R and Shenoy K V 2009 HermesC: low-power wireless neural recording system for freely moving primates *IEEE Trans. Neural Syst. Rehabil. Eng.* **17** 330–8