# Control of a brain–computer interface without spike sorting



2009 J. Neural Eng. 6 055004

(http://iopscience.iop.org/1741-2552/6/5/055004)

J. Neural Eng. **6** (2009) 055004 (8pp) doi:10.1088/1741-2560/6/5/055004

# Control of a brain–computer interface without spike sorting

**George W Fraser**[1,2,5], **Steven M Chase**[1,2,3], **Andrew Whitford**[2,4] **and Andrew B Schwartz**[1,2,4]

[1] Department of Neurobiology, University of Pittsburgh, Pittsburgh, PA 15213, USA
[2] Center for the Neural Basis of Cognition, University of Pittsburgh, Pittsburgh, USA
[3] Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[4] Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA 15261, USA

E-mail: gwf2@pitt.edu

Received 16 January 2009
Accepted for publication 15 July 2009
Published 1 September 2009
Online at stacks.iop.org/JNE/6/055004

**Abstract**
Two rhesus monkeys were trained to move a cursor using neural activity recorded with silicon arrays of 96 microelectrodes implanted in the primary motor cortex. We have developed a method to extract movement information from the recorded single and multi-unit activity in the absence of spike sorting. By setting a single threshold across all channels and fitting the resultant events with a spline tuning function, a control signal was extracted from this population using a Bayesian particle-filter extraction algorithm. The animals achieved high-quality control comparable to the performance of decoding schemes based on sorted spikes. Our results suggest that even the simplest signal processing is sufficient for high-quality neuroprosthetic control.

(Some figures in this article are in colour only in the electronic version)

## Introduction

This paper demonstrates an alternate strategy for processing signals from intracortical electrodes for prosthetic control. Current brain–computer interfaces based on intracortical electrode arrays use extracellular action potentials processed with spike-sorting strategies to generate a control signal (Hochberg *et al* 2006, Musallam *et al* 2004, Santhanam *et al* 2006, Taylor *et al* 2002, Velliste *et al* 2008, Wessberg *et al* 2000). A threshold voltage is set for a particular electrode, and each time the voltage potential exceeds that threshold, a waveform snippet of the time-varying potential is recorded. These waveforms are divided into one or more categories, in the hope of identifying individual cells or separating out cell-related activity. During a typical neuroprosthetic control experiment, a human operator will view the ongoing waveforms for a brief period on each channel at the beginning of each day's session and attempt to identify distinct waveforms by setting the spike-sorting parameters of a digital signal processing device. The need for human intervention is an obstacle to bringing neural prosthetics from the lab to the clinic, as is the complex digital signal processing employed in current brain–computer interfaces..

A separate class of brain–computer interfaces (BCIs) use low-frequency signals from external electrodes (EEGs, for example Wolpaw *et al* 1991), surface macroelectrodes (ECoGs, for example Leuthardt *et al* 2004) or intracortical microelectrodes (LFPs, for example Mehring *et al* 2003, Pesaran *et al* 2002). These techniques and those based on spike activity can be ordered according to their spatial resolution, with EEGs at one end and spike activity at the other. The BCIs with the highest number of degrees of freedom and accuracy have been operated with spike activity (Taylor *et al* 2002, Velliste *et al* 2008, Wessberg *et al* 2000).

Recent papers have suggested other schemes for extracting a control signal from intracortical recordings. Stark and Abeles (2007) showed that multiunit activity, reflected in the power between 300 and 6000 Hz, is a good predictor of an upcoming hand movement in macaques. Ventura (2008), using simulated data, extracted movement intent from the mixtures of tuned units, with an accuracy comparable to traditional

[5] Author to whom any correspondence should be addressed.

 

spike-sorting approaches. Both of these papers suggest that there is information found in relatively unprocessed signals from microelectrodes in the motor cortex. In the current study, we report on the feasability of operating a brain–computer interface without spike sorting with recordings from two monkeys chronically implanted with microelectrode arrays.

## Methods

### *Chronic microelectrode implant*

Monkey A, a male rhesus macaque, was implanted in January 2007 with a single 96-channel array (Blackrock Microsystems; Maynard *et al* 1997) on the convexity of the motor cortex next to the central sulcus, with the lateral edge of the array ∼2 mm medial to the genu of the arcuate sulcus. The recording sessions reported here were done in June 2008. Monkey C, also a male rhesus macaque, was implanted in February 2009 in the same location. The reported sessions were recorded 10 weeks later. Monkey A's array gave as many as 130 distinguishable cells and multineuron combinations during the period of best recordings. At the time of this experiment, June 2008, there were 1–2 clear single units, 1–6 probable single units, ∼40 fair neuron/multineuron combinations and 15–30 poor multineuron traces on a typical day of recording. Monkey C's array gave ∼75 fair neuron/multineuron combinations and ∼5 probable single units on the day of this experiment.

### *Behavioral task*

Prior to the implant, the monkey was trained to do a center-out arm movement task in a virtual environment. It sat in front of a stereoscopic computer monitor (DTI, Rochester, NY) with one arm gently restrained, and the other free to move with an infrared marker taped slightly distal to the wrist so that its position could be monitored with a motion capture system (Northern Digital, Waterloo, ON, Canada). The position of the infrared marker drove the position of a green cursor displayed on the screen in a virtual environment. Blue spheres were displayed as targets at various points in the three-dimensional environment and the monkey was rewarded with a droplet of water for making and holding contact with the targets. In each case, the monkey first made contact with a central target and touched the target for a required time of 200–300 ms, selected randomly from a uniform distribution. A peripheral target was then randomly selected from a queue of remaining targets. In the experiments described in this paper, we used a set of 16 targets, their centers arranged evenly in a circle on the plane of the monitor with a radius of 85 mm for monkey A and 79 mm for monkey C. The monkey moved to the peripheral target within a limited time period—1600 ms in this study—and held contact for a required time of 0–200 ms. The cursor and the target had radii of 8 mm for monkey A and 9 mm for monkey C. The variable hold period was long enough that the monkey could not consistently succeed by moving straight through the target—it had to stop or drastically slow its movement as it approached the target. Once a target was hit successfully, it was dequeued from the remaining targets.

After the monkey was implanted and the neural signals were deemed large enough and stable enough to sort (about 3 weeks for both monkeys), the animal began to use the brain–computer interface. Both arms were restrained and the cursor was driven with neural activity. To decode intent, it is necessary to determine the tuning parameters of the neurons being recorded. In this study, we made a first estimate by running the task with null tuning parameters. In brain control, the cursor was placed on the central target at the beginning of each trial. Then the peripheral target was presented after the expiration of the central hold period. Initially, the monkey was unable to move the cursor with null tuning parameters and failed each trial. Nevertheless, the monkey modulated its neural activity consistently for each target, making it possible to estimate tuning parameters and begin real-time neural decoding. During this initial period where the control parameters were adapting, we enhanced the straightness of the trajectories by artificially reducing the deviation from the ideal, perfectly straight movement. At each timestep, the prediction of the decoding algorithm was decomposed as the sum of a vector straight toward the target and a vector orthogonal to it. The orthogonal vector was then multiplied by the *deviation gain*: between 0.1 and 0.5 in this study. Adjusting the deviation gain is a highly subjective process; we ramp up the deviation gain as the model parameters are re-fit with more data and the decoding becomes more consistent. Choosing how much deviation gain to apply is a matter of balancing the need for straight trajectories with the tendency of the monkey to modulate erratically if it realizes that the trajectories come out straight regardless of the consistency of its modulation. We need straight trajectories while the decoding parameters are being fit so that we can accurately compute the tuning functions of signals. At the same time, the monkeys have shown a tendency to produce inconsistent modulation if the deviation gain is too strong for too long. The experimenter attempts to balance these priorities. Typically the deviation gain is 0.1–0.25 for the first 16 targets, 0.5 for the next set and 0 thereafter. If the trajectories are still erratic, we will keep it on longer. This procedure was used only in these adaptive sessions. Deviation gains were used only for a short period at the beginning of a daily experimental session for calibration purposes. Subsequent control used no deviation gain.

### *Signal processing*

Signals were buffered, amplified, bandpass filtered at 250–8000 Hz and processed using a 96-channel Plexon Multichannel Acquisition Processor (Plexon Inc., Dallas, TX). The DSP system was configured so that it would register all events crossing a negative threshold in the downward direction and send their times to one of the task-management computers. In monkey A, a threshold was set across all channels at $-37.5\ \mu$V, except ten channels where the power of the time-varying voltage was so high that it constantly saturated the A/D converter. On those channels the gain was reduced until the clipping was under control, effectively putting the thresholds in the (−)40–80 mV range. In monkey C, the signals had higher amplitude and the threshold was set at –60 $\mu$V for all channels. In both animals the threshold was chosen to be as

low as possible without severely overloading the capacity of the DSP system on too many channels. There was a sufficient variation in the signals to cross the threshold on 95/96 channels in monkey A and 78/96 channels in monkey C. It should be noted that because of the difference in the age of the implanted arrays (10 weeks in monkey C versus 1.5 years in monkey A), the signals are very different. In the very old array of monkey A, the amplitude of the sortable action potentials is much closer to the amplitude of the background noise. In the very young array of monkey C, the action potentials stand out much more strongly from the background, but there are often more active neurons observable on a given channel. Hence the threshold had to be set higher in monkey C to avoid capturing so much activitiy that the capacity of the DSP system would be overloaded. This higher threshold resulted in 18 channels that observed no threshold crossings. For the same reason monkey C showed less of a difference in the baseline rate for sorted versus unsorted sessions, as evidenced in figure 5. It should be noted that in both monkeys the resulting signals were not comparable to single-unit recordings; the waveforms picked up by the threshold included large amounts of background activity and noise on nearly every channel where the signal crossed the threshold at all. This is evidenced by the fact that threshold crossings showed consistently higher baseline rates and modulation depths than units sorted on the same channels (figure 5).

### *Decoding algorithm*

We used a Bayesian–Monte Carlo estimation algorithm very similar to the one described by Brockwell and colleagues (2003). This algorithm, called a particle filter, solves the problem of decoding neural intent by offering a firing-rate model for each recorded unit, and then estimating the movement intent most consistent with this forward model and the recent history of movement. The particle filter can be intuitively understood in terms of a hypothesis space of movement intents that the monkey might have at the present timestep. The particle filter uses modulated control signals: a neuron whose firing rate increases for a particular movement direction, or in this study, a time-varying voltage signal which crosses a threshold more or less often depending on the current intended movement direction. A single modulating signal gives the particle filter information about which regions of the hypothesis space are consistent with its current level of activity. One signal alone will leave ambiguity: a high-firing neuron indicates movement in its preferred direction, or perhaps movement somewhat off its preferred direction at a higher speed. These signals are noisy, so one signal may simply be wrong at the present moment. The particle filter searches the hypothesis space at several hundred points—particles—for the region that is most likely given all the current activity levels. The distribution of particles in one timestep is generated from their position in the previous timestep, which enforces our assumption that movement in the current timestep is similar to movement in the previous timestep. This distribution is biased according to the likelihood of the hypothesis that each particle had found. Thus the particles form a cloud that follows the most likely region of the hypothesis space from one timestep to another. This process is diagrammed in figure 1. Using a particle filter as an extraction algorithm is simply a matter of identifying a modulated signal and specifying a tuning function for it. Brockwell and colleagues used an exp(cosine) firing-rate model where each sorted unit had a baseline rate, a single preferred direction and an adjustable-width tuning function. Our new approach is based on a firing-rate model that makes fewer assumptions about the shape of the tuning function:

$$\lambda_i = b_{i,0} + s \sum_{j=1}^{8} w_{i,j} \cdot f(\theta \cdot 4/\pi - j). \tag{1}$$

Here, $\lambda_i$ is the total rate of threshold crossing on channel $i$, $b_{i,0}$ is the baseline firing rate of that channel, $s$ is the speed of movement and $\theta$ is the angle of movement in the $X$–$Y$ plane. $f$ is a cubic spline basis function which is being shifted and stretched by the $\theta \cdot 4/\pi - i$ formula. It has the following mathematical definition:

$$f(x) = \left\langle \begin{array}{c} x < -2 | 2 < x \\ -2 \leqslant x < -1 | 1 < x \leqslant 2 \\ -1 \leqslant x \leqslant 1 \end{array} \right| \begin{array}{c} 0 \\ (2 - |x|)^3 \\ 1 + 3(1 - |x|) + 3(1 - |x|)^2 - 3(1 - |x|)^3 \end{array} \Bigg\rangle,$$

where $f$ is a bell-shaped function spanning the range $[-2, 2]$. It represents a cubic spline basis function; by shifting and spacing these basis functions at intervals of 1 and adding them up, we get a set of basis functions that span the entire circle of movement directions. $w_{i,j}$ is the weight of a particular spline basis function, fit by a regression model. The effect of the summation portion of the firing model is to fit a smoothing spline to the firing rate as a function of the intended movement direction in the $X$–$Y$ plane. This process is diagrammed in figure 2. Our model effectively states that the firing rate is equal to a baseline rate plus an 8-knot spline function in polar coordinates that expands and contracts according to the speed of movement. The $w_{i,j}$ parameters which determine the shape of the spline function are fit using linear regression. Our decoder is written in Matlab (Mathworks, Natick, MA) and uses its GLM library to fit the model with an identity link function.

The empirical firing rate of cell $i$ is taken to be Poisson distributed with mean $\lambda_i$. Thus, the probability of observing a specific bin count $n_i$ given a firing rate $\lambda_i$ from equation (1) is given by the following equation:

$$P(n_i|\lambda_i) = \text{poisspdf}(n_i, \lambda_i \cdot \Delta t),$$

where poisspdf gives the Poisson probability density function with mean $\lambda_i \cdot \Delta t$ evaluated at $n_i$ ($\Delta t$ is the width of the bin). The particle filter also incorporates an assumption about the way velocity changes over time. In both Brockwell *et al* and this paper, it is assumed that the velocity at one timepoint is related to the previous timepoint according to a Gaussian distribution. The standard deviation of this distribution depends on the length of the time step and the assumptions of the experimenter. In a brain-control experiment it should correspond to the distance to the target and the movement

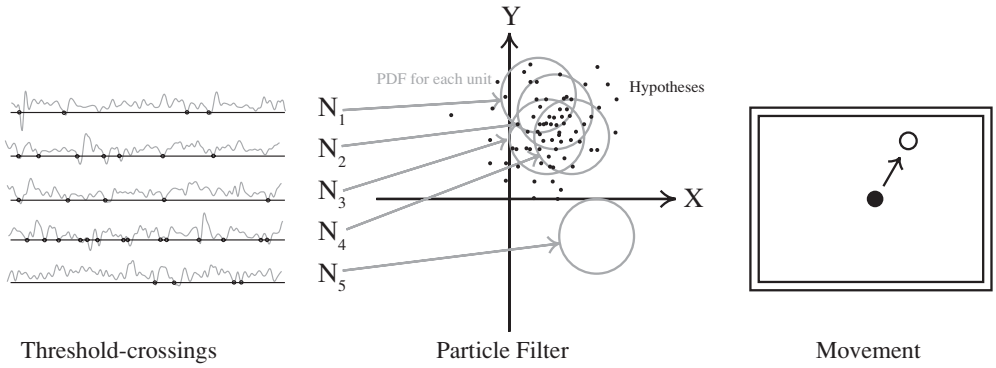


**Figure 1.** Extracellular activity-related signals are recorded from motor cortex activity ($N_1$–$N_5$). The number of threshold crossings is counted on each channel for each 33 ms interval (left panel) and fed into a particle filter extraction algorithm that compares bin counts to the known tuning functions of those channels. Each bin count points to a probable region of the hypothesis space of movement trajectories (middle panel). The particle filter maintains a set of hypotheses about movement (middle panel, small dots). At each timepoint these particles are moved randomly by a distance drawn from a Gaussian distribution, then resampled according to the probabilities indicated by the counts $N_1$–$N_5$. In this manner the particle cloud is dragged around by the probable regions from the $N_1$–$N_5$ counts. An outlier bin count which disagrees with the rest of the population (above, $N_5$) will have little influence over the particle cloud. To generate an estimate of the global movement intent we simply take the mean of all the particles.

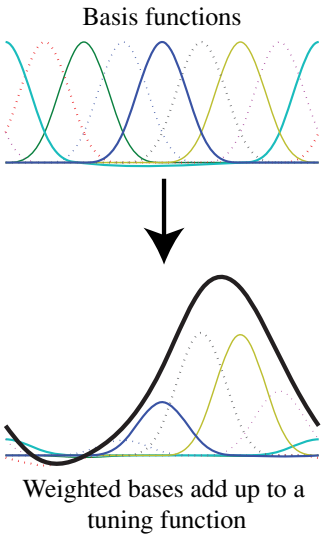


**Figure 2.** A tuning function is fit as a weighted sum of spline basis functions. The tuning function of one channel is shown as the thick black line. It is fitted as a sum of shifted bell-shaped basis functions (top), each of which spans a $\pi/4$ radian section of movement directions. The basis functions are scaled (bottom) and added to produce the fitted function.

duration. Here, we assume that velocity changes with a standard deviation of 67 mm $s^{-1}$ over 1 s. The prior probability of the $x$-component velocity at the current timestep, $\bar{x}$, given the $x$-component of velocity at the previous timestep, $x$, is a function of the length of the timestep $\Delta t$ and the standard deviation of movement per second $\sigma$:

$$P(\bar{x}|x) = \text{normpdf}(\bar{x}, x, \sqrt{\Delta t} \cdot \sigma), \tag{2}$$

where normpdf describes the normal distribution probability density function for the distribution with mean $x$ and standard deviation $\sqrt{\Delta t} \cdot \sigma$, evaluated at $\bar{x}$. The problem of decoding an intended movement direction from the threshold-crossing activity is then a matter of maximizing the probability distribution defined by equations (1) and (2):

$$P(\bar{x}, \bar{y}|x, y, n_1, \ldots, n_m) \sim P(\bar{x}|x) \cdot P(\bar{y}|y) \cdot P(n_1|\bar{x}, \bar{y}) \cdot \cdots \cdot P(n_m|\bar{x}, \bar{y}), \tag{3}$$

where $x$ and $y$ are set to 0 at the beginning of each trial (during the central hold period); $n_1, \ldots, n_m$ are the observed bin counts for the $m$ channels being used. The absolute probability of the left-hand side of equation (3) is scaled by additional terms, but since we are only interested in the relative maximum we can leave them out. The particular filter uses a set of particles—points in the $X$–$Y$ velocity coordinate space—to estimate the maximum of this distribution. Each time a new set of bin counts arrives, the particles are moved probabilistically. First, each particle makes a random movement whose destination is chosen from a normal distribution with mean at the previous location of the particle and standard deviation of $\sqrt{\Delta t} \cdot \sigma$. This corresponds to the first two terms on the right-hand side of equation (3). Second, for each particle, a probability is calculated equal to the remaining terms of equation (3). The entire population of particles is then resampled from itself, with replacement, according to these probabilities. This means that high-probability particles are more likely to reappear after resampling, and that some particles will end up being represented multiple times in the new population.

These steps are constructed so that the probability distribution of the position of a single particle is exactly equal to the distribution being estimated. The entire population of particles acts as a proxy for the distribution being estimated. We generate a single estimate of movement for real-time brain control by taking the mean of the entire population of particles. The accuracy of decoding increases with the number of particles, but so does the computational complexity. We used 400 particles, which we found to be the highest number where the computation could be completed reliably in the time between bin counts. With a simulated population we found that the quality of control did not become noticeably worse until the number of particles dropped below 50.

The described decoding algorithm has a number of parameters (the $b_{i,0}$ and $w_{i,j}$ terms in the model) that must be fit from the actual tuning of the neurons. In the reported unsorted neural control session for monkey A, we used previous day's parameters as the initial parameters for decoding. At the beginning of the day we ran an adaptive session where we re-fit the $w_{i,j}$ and $b_{i,0}$ terms. We gave 4 sets of 16 targets, re-fitting the model to the cumulative set of data for that day after each set. For the purposes of linear regression, the intended movement was taken as the idealized vector from the center of the workspace to the peripheral target. For monkey C we set the initial parameters to 0, which means that for the first 16 targets the cursor did not move. Nevertheless, the monkey looked at the monitor and modulated its neural activity sufficiently to get a set of parameters to move during the remainder of the adaption period.

The sorted session for monkey C was done in exactly the same way as the unsorted session, with parameters initially set to 0. The sorted session for monkey A was done somewhat differently; parameters were initially 0 but the extraction algorithm being used was a variation on the Bayesian inference decoder where Laplace's method of integration is used in place of the Monte Carlo particles of the particle filter. We have found that the accuracy of the two methods is similar; the main difference is that the Laplacian integration method is computationally more efficient.

## Results

Four datasets are reported here, unsorted and sorted for monkeys A and C. In all cases cursor movement was confined to the two-dimensional plane of the monitor, though the VR system displayed objects in three dimensions. Monkey C did the unsorted control session first, followed by an 80 min break, and then the sorted control session. It had less than half its daily water quota in the first session so its motivation level was still high during the second session, assessed by the fact that it engaged in the task continuously for the entirity of both sessions. It used exactly the same decoding algorithm for both sessions, except that the extraction algorithm was being fed counts of threshold crossings in the first session and counts of sorted spikes in the second. Monkey A did the unsorted session 2 weeks after the sorted session. The sorted session contained two breaks where the decoding algorithm was seamlessly switched to the population vector algorithm (Taylor *et al* 2002) for 80 and 112 trials. These trials are excluded. This particular sorted session was chosen for comparison because it was also done in the middle of the week when the monkey's motivation level tends to be highest, it used a Bayesian decoding algorithm very similar to the particle filter, and it was contemporaneous with the other session.

There is an inherent variation in the quality of brain-control decoding from one session to the next, even when they are performed on the same day. When the parameters are initially set to zero and then fit from a limited set of movements during the adaptation period, the accuracy of the fit depends on the way that the animal modulates its activity during those particular trials. Therefore, it is not possible to make an exacting comparison between the quality of control in the sorted versus unsorted session. We can only evaluate whether the unsorted scheme works approximately as well as a decoder based on sorted units. The primary metrics we have for the quality of control are our subjective impression of figure 3; the success rate of the animal; the speed of movement; and the straightness of the trajectories.

| | Monkey A | | Monkey C | |
|---|---|---|---|---|
| | Unsorted | Sorted | Unsorted | Sorted |
| Success rate | 78% | 93% | 96% | 84% |
| Duration | 815 ms | 932 ms | 707 ms | 630 ms |
| Straightness | 1.12 | 1.10 | 1.11 | 1.17 |

Success rate is simply the proportion of trials where the animal succeeds according to the central hold, movement time and peripheral hold criteria defined in the methods section. Speed is the time between the peripheral target being presented and the cursor contacting it. Straightness is the total path length of the cursor from when it left contact with the no longer visible central target to when it made contact with the peripheral target, divided by the length of a perfectly straight line between the endpoints of the same path. The above table gives the median speed and straightness for all successful trials.

We can treat the threshold crossings on a particular channel as though it were a neuron and define a tuning function for it—an estimate of what the threshold-crossing rate will be when the monkey moves in a particular direction at a particular speed. The $w_{i,j}$ parameters of the particle filter decoder describe such a tuning function; figure 4 compares the assumptions of the decoder with the empirical per-target firing rates observed in the control session. The thick black lines and the black circles in figure 4 show the comparison between the model's spline fit and the later activity of the channels. The channels shown are selected from the better-tuned half of what we recorded, but are representative in terms of the model's fit and the lack of multimodality in the tuning functions. The particular examples shown in figure 4 are indicated on the scatterplots in figure 5.

We would also like to make a quantitative comparison of the tuning characteristics of channels and neurons between unsorted and sorted sessions. In the unsorted session, we fit a cosine function to the 16-target mean threshold-crossing rates. In the sorted session, we fit a cosine function in the same manner, except that we used the combined activity of the sorted neurons in place of the threshold-crossing event. Monkey A had 60 channels with sorted units where such a comparison could be made; monkey C had 62. Monkey A had 10 channels with two sorted units and monkey C had 20; in these cases the activities of the two units were simply merged together for the purpose of fitting a cosine. In this small dataset, the mean difference in preferred directions for two units on the same channel was 80°. The model for the cosine fit is

$$n_{i,j} = \beta_{i,0} + \beta_{i,x} x_j + \beta_{i,y} y_j,$$

where $n_{i,j}$ is the mean firing rate for unit $i$ and target $j$; $\beta_{i,0}$ is the baseline firing rate for neuron $i$; $\langle x_j, y_j \rangle$ is the

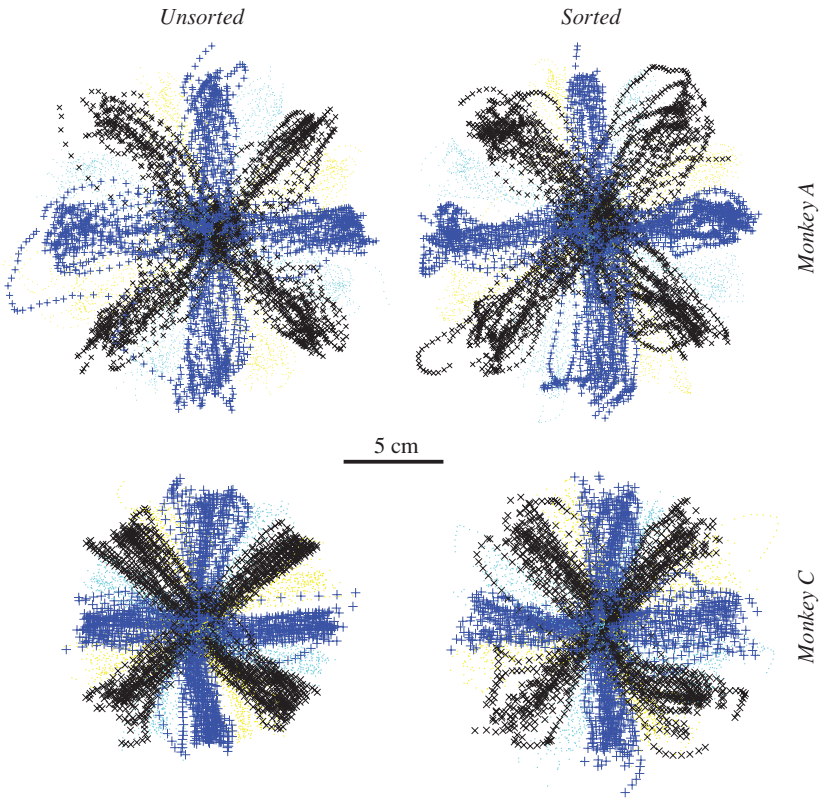


**Figure 3.** Superimposed movement trajectories as the monkey does the center-out task for 16 targets using unsorted threshold-crossing signals (left) or sorted activity (right). This plot shows position samples at 30 Hz and incorporates all successful trials on a single day between the start of the central hold period and the end of the peripheral hold period. The symbols are varied according to the target direction in the current trial. The upper-left plot contains 384 trials, the upper right shows 233, lower left 480, lower right 512. The trials for each condition were done in one contiguous block, except that in the monkey A/sorted condition there were two breaks in the trials where the decoder was switched to the population vector algorithm (Taylor *et al* 2002) for 80 and 112 trials (these trials are excluded). Success rates were 78%, 93%, 96% and 84% for A/unsorted, A/sorted, C/unsorted, C/sorted. Because of the vagaries of monkey motivation and the fact that no two adaptation sessions are the same, it is not possible to make an exacting comparison of performance here. We can only observe that both algorithms produce qualitatively good control.

vector to target $j$; the angle of $\langle \beta_{i,x}, \beta_{i,y} \rangle$ is the preferred direction of unit $i$ and the length of $\langle \beta_{i,x}, \beta_{i,y} \rangle$ is the modulation depth of unit $i$. The baseline rates, modulation depths and preferred directions are compared between sorted and unsorted conditions in figure 5. In both animals the baseline rates and modulation depths were higher in the unsorted condition, and the preferred directions were similar between unsorted threshold crossings and the combined sorted units recorded later on the same channels.

## Discussion

We have demonstrated that good neural control can be achieved without conventional spike sorting or careful setting of thresholds. There is intrinsic variability in the quality of control from one session to the next, so it is not possible to make an exact quantitative comparison between sessions in these data. An experimental paradigm that better controls for the quality of adaptation data and the motivation level of the animal is clearly an avenue for future research. This initial finding has demonstrated that unsorted signals can be substituted for conventional ones without a dramatic, obvious drop in the quality of control.

We chose an 8-knot spline to model the tuning function because we expected mixed neuron signals to create complicated, sometimes multimodal tuning functions. We were surprised to find that on virtually all channels, the tuning function of the unsorted threshold crossings was roughly unimodal, in spite of the fact that where two units were recorded on the same channel their preferred directions did not tend to be similar. Since the tuning functions of the unsorted signals are not multimodal, one may reasonably ask why the population vector algorithm does not work well with these signals. We did attempt to use the population vector algorithm on these signals, but found the quality of control so poor that we could not collect enough data to report. We speculate that the challenging aspect of unsorted signals is not multimodality but the background noise that is introduced. Statistical extraction algorithms like the particle filter have the advantage of being able to recognize when a channel is an outlier in the present timestep, and to effectively reduce its contribution to the inference of intent. The particle filter examines the space of hypothetical velocities and asks the question: what is the probability of observing these firing rates at various points in the hypothesis space of velocities? If a channel is momentarily inundated with background noise, it will point to a region of

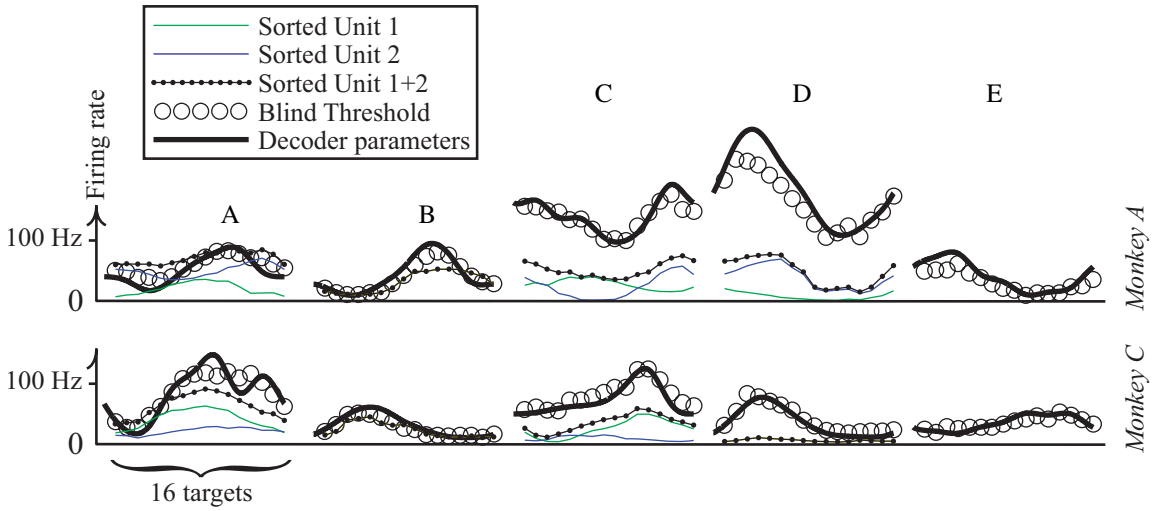


**Figure 4.** Five examples of 16-target tuning functions for neurons, sums of neurons and unsorted threshold crossings. Each of these five examples shows various tuning functions from a single channel. The *Y*-axis shows the firing rate of a pseudo-unit during the movement period of the task. The *X*-axis gives the angle from the start point in the center of the monitor to the peripheral target. The multiple plotted lines illustrate how one or two identifiable neurons and background activity add up into the signal we observe with the blind thresholding we used. The thin lines are the tuning functions of sorted units. The connected-circles line is the sum of these units. The open black circles show the tuning function of the blind-threshold pseudo-unit used in unsorted control. The thick black line shows the fitted spline function that the particle filter decoder is using, which is fit from a separate dataset at the beginning of the recording session. In these five example channels, we can see the different relationships between the identifiable units on a channel and the signal you get when you set a blind threshold. Unsorted activity is sometimes well-explained as the sum of units on that channel (A), (B); most often it has the same shape as the sum of units but a higher baseline rate and modulation depth (C), (D); and sometimes a channel without sortable activity will show strong modulation in the hash (E).

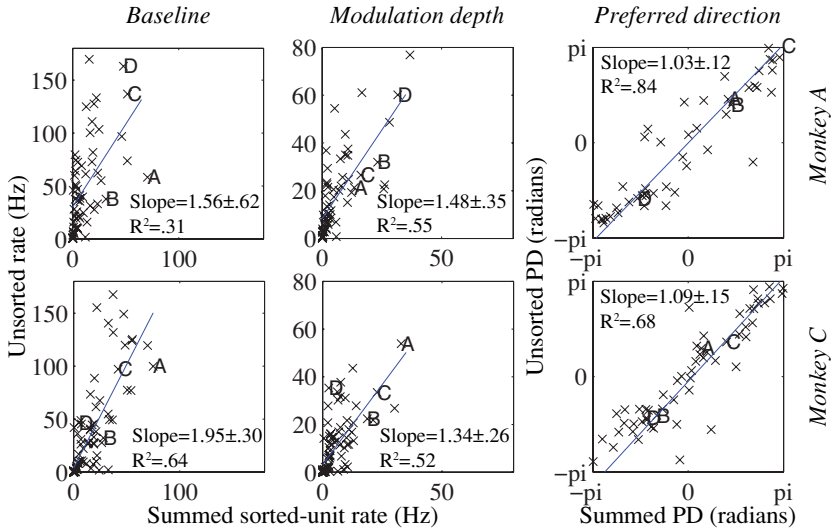


**Figure 5.** The summed activity of all the units on each channel was fit as a linear model of the *X*- and *Y*-components of target direction plus a constant. The same was done with the unsorted threshold crossings. This fit is equivalent to a cosine-tuning function with a baseline rate, a modulation depth and a preferred direction. Baseline rate is the mean firing rate across all targets. Modulation depth is the height of the cosine-tuning function, approximately the difference between the most-active target and the least-active target. Preferred direction is the movement direction which, according to the tuning function, would elicit maximum firing from the unit. The above plots show the relationship between each parameter of this model when it is fit with the unsorted threshold-crossing data, on the *Y*-axis, versus the sum of the sorted units recorded on the same channels, on the *X*-axis. Each channel gives a single point on the plot; the channels from figure 4 are indicated with letters. A linear fit is detailed on each plot. Unsorted control shows higher baseline rates, higher modulation depths and nearly the same preferred directions.

the hypothesis space that is inconsistent with the rest of the channels, and it will have little weight toward the decoding of intent. The population vector algorithm gives each signal equal weight in the computation of intent, even if that signal is highly inconsistent with the majority of the population. We have demonstrated that with a good choice of extraction algorithm, a simple global threshold specification can be used in place of the spike-sorting schemes of conventional brain–computer interfaces. These results have an immediate relevance for designs of self-contained spike processing circuits in the

next generation of neural prosthetic devices. Without the need to set parameters of signal processing, it is possible to make an effective neural prosthetic system without operator intervention.

Perhaps more importantly, the kind of extraction algorithm demonstrated in this paper is arguably better suited to the indistinct patterns of multiunit activity that are typical of long-term chronic multielectrode recordings. These signals are composed of summed mixtures of signals from a number of sources rather than action potentials of individual neurons. They are subject to high levels of baseline noise and their tuning functions are harder to predict. We have shown that a well-chosen extraction algorithm can contend with these issues and provide a good control signal. While we used the same signal processing equipment that has been employed for years in neural prosthetics, we used it in a way that simulated a much simpler system. Our results show that a probe with fixed thresholds and one-way telemetry could be used for effective prosthetic control. It is easier to imagine a turn-key clinical system for neural prosthetics that is based on the threshold-crossing counter used here. The elimination of elaborate signal processing regimes is an important step towards putting neural prosthetics into the real world.

## References


Brockwell A E, Rojas A L and Kass R E 2003 Recursive Bayesian decoding of motor cortical signals by particle filtering *J. Neurophysiol.* **91** 1899–907

Hochberge 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

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

Maynard E M, Nordhausen C T and Normann R A 1997 The Utah intracortical electrode array: a recording structure for potential brain–computer interfaces *Electroencephalogr. Clin. Neurophysiol.* **102** 228–39

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

Musallam S, Corneil B D, Greger B, Sherverger H and Andersen R A 2004 Cognitive control signals for neural prosthetics *Science* **305** 258–62

Pesaran B, Pezaris J, Sahani M, Mitra P M and Andersen R A 2002 Temporal structure in neuronal activity during working memory in macaque parietal cortex *Nat. Neurosci.* **5** 805–11

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

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

Taylor D M, Tillery S I H and Schwartz A B 2002 Direct cortical control of 3D neuroprosthetic devices *Science* **296** 1829–32

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** 1098–1101

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

Wessberg J, Stambaugh C R, Kralik J D, Beck P D, Laubach M, Chapin J K, Kim J, Biggs S J, Srinivasan M A and Nicolelis M A L 2000 Real-time prediction of hand trajectory by ensembles of cortical neurons in primates *Nature* **408** 361–5

Wolpaw J R, McFarland D J, Neat G W and Forneris C A 1991 An EEG-based brain–computer interface for cursor control *Electroencephalogr. Clin. Neurophysiol.* **78** 252–9