Reviews in the Neurosciences, 14, 107-119 (2003)

# Training in Cortical Control of Neuroprosthetic Devices Improves Signal Extraction from Small Neuronal Ensembles

S.I. Helms Tillery[1], D.M. Taylor[2] and A.B. Schwartz[3]

[1]*Harrington Department of Bioengineering, Arizona State University, Tempe, AZ, USA*
[2]*FES Center, Cleveland VA Medical Center, Cleveland, OH, USA*
[3]*Department of Neurobiology, University of Pittsburgh, Pittsburgh, PA, USA*

## SYNOPSIS

**We have recently developed a closed-loop environment in which we can test the ability of primates to control the motion of a virtual device using ensembles of simultaneously recorded neurons /29/. Here we use a maximum likelihood method to assess the information about task performance contained in the neuronal ensemble.**

**We trained two animals to control the motion of a computer cursor in three dimensions. Initially the animals controlled cursor motion using arm movements, but eventually they learned to drive the cursor directly from cortical activity. Using a population vector (PV) based upon the relation between cortical activity and arm motion, the animals were able to control the cursor directly from the brain in a closed-loop environment, but with difficulty. We added a supervised learning method that modified the parameters of the PV according to task performance (adaptive PV), and found that animals were able to exert much finer control over the cursor motion from brain signals.**

**Here we describe a maximum likelihood method (ML) to assess the information about target contained in neuronal ensemble activity. Using this method, we compared the information about target contained in the ensemble during arm control, during brain control early in the adaptive PV, and during brain control after the adaptive PV had settled and the animal could drive the cursor reliably and with fine gradations.**

**During the arm-control task, the ML was able to determine the target of the movement in as few as 10% of the trials, and as many as 75% of the trials, with an average of 65%. This average dropped when the animals used a population vector to control motion of the cursor. On average we could determine the target in around 35% of the trials. This low percentage was also reflected in poor control of the cursor, so that the animal was unable to reach the target in a large percentage of trials.**

**Supervised adjustment of the population vector parameters produced new weighting coefficients and directional tuning parameters for many neurons. This produced a much better performance of the brain-controlled cursor motion. It was also reflected in the maximum likelihood measure of cell activity, producing the correct target based only on neuronal activity in over 80% of the trials on average.**

**The changes in maximum likelihood estimates of target location based on ensemble firing show that an animal's ability to regulate the motion of a cortically controlled device is not crucially dependent on the experimenter's ability to estimate intention from neuronal activity.**

## KEY WORDS

arm, motor cortex, neuroprosthetic, maximum likelihood

Reprint address:
S.I. Helms Tillery
Harrington Dept. of Bioengineering, ECG 334
Arizona State University
Tempe, AZ 85287-9709, USA
e-mail: Steve.HelmsTillery@asu.edu

## INTRODUCTION

Effort in the field of cortical neuroprostheses over the last few years has focused on extracting increasing amounts of information from chronically recorded neurons /9,10,13,17,25,26,29,30/. The dominant thinking has been that if it is possible to read out a subject's intention, then it should be possible to implement that intention using alternative means.

Recently published results have shown that by providing feedback to an animal about the status of a neurally controlled device, an animal can use the device as a tool with as few as ten neurons /26,28, 29/. Since early reports showing that animals could control the one-dimensional motion of a cursor by changing the activity of single neurons /5,23,24/, neuroscientists have anticipated using this to control the motion of a more complex device, such as a robotic arm.

Initial reports showing subjects with direct cortical control over more complex devices have all relied on biofeedback /3,15,16,26,28/. In animal studies /3,26,28/, animals were trained to perform a natural movement that mimicked the device motion. Investigators recorded brain activity as the animals moved, and then developed a mapping that provided for a best-fit prediction of the animals' behavior from the activity of simultaneously recorded neurons. This prediction was then applied to the control of an external device, and in each of these cases, the animals were able to exert direct control over the device.

Finely graded generalizable control, however, waited on one further conceptual advance in the development of control algorithms /29/. Taylor *et al.* showed that device control could be remarkably improved by loosening the initial constraint which relies on the mapping between arm movement and neuronal activity. In that work, a device was controlled from brain signals using a modified population vector algorithm that was initialized by recording from the ensemble as the animal performed unconstrained point-to-point arm movements. As the animal directly controlled motion of a cursor from the same signals, however, the parameters of the population vector were modified using a supervised learning paradigm. Over the course of 20 minutes, the adaptive algorithm quickly retuned the parameters so that task performance went from about 50% up to 90% accurate target acquisition.

There are many possible reasons for the improvement (see /29/ for details). The population vector is known to be sensitive to both the number of well-tuned neurons and the distribution of tuning directions. The adaptive algorithm certainly led to improved tuning functions in many neurons, and changed their tuning directions by as much as 180 degrees. It is also the case that the population vector selects for neurons that are closely coupled to arm movements. It might be difficult for the animal to modulate the firing rates of these neurons outside the context of arm movements.

In this paper we look at the information about task performance that is contained in the ensemble discharge using a maximum likelihood method. We show that the ability to detect the target of a movement using only the neuronal firing rates is enhanced by modifications to the population vector.

## METHODS

### Behavioral tasks

We trained macaques to perform a standard three-dimensional center→out task in a virtual reality environment /22,28/. The animals viewed stereo images of a spherical cursor and targets projected onto a mirror placed obliquely a few centimeters from their face. The virtual images of the targets were projected into the animal's workspace, and the task was to move the cursor into the target spheres. An initial target sphere would appear in the center of the workspace. Once the animal had moved the cursor into this 'center' location, a peripherally located target would appear and the central target disappear. The animal would then have a limited period in which to move the cursor into the new target. The eight peripheral targets all lay about 9 cm from the center location on the corners of a cube.

The first part of each day's task was the *arm-control* task, in which the animal directed motion of the cursor using unrestrained three-dimensional arm movements. We monitored the animal's wrist

location using an Optotrak motion tracking system (Northern Digital, Inc.), and used the movement as monitored by the Optotrak system to control the cursor motion in real time. The wrist motion required of the animal was the same as the required motion of the virtual cursor.

We recorded both the behavior and neuronal activity as the animal performed this phase of the task, and used these results to create a cortical control algorithm that mapped activity in the neuronal ensemble to motion of the virtual cursor. At this point, we restrained the animal's arms, and the animal was required to control motion of the cursor by modulating the activity of neurons. In the early stages of this *brain control*, cursor motion was determined entirely by a population vector based on the arm movements. After 8-10 brain-controlled movements to each of the targets, we began modifying the population vector based on performance using the adaptive algorithm /28,29/.

**Neuronal recording**

The animals were chronically implanted with fixed stainless steel and/or tungsten microwire arrays in motor and premotor cortical areas. Typically we implanted four arrays of 16 wires each, giving us ideally 64 channels. Surgery was performed under gas anesthesia and in aseptic conditions in accord with University IACUC and NIH guidelines.

Upon recovery, we recorded simultaneous activity on all the implanted electrodes using the Multichannel Acquisition Processor and associated RASPUTIN neuronal recording suite (Plexon Inc.) as the animals performed the behavioral tasks. Our recordings consisted of a mix of well-isolated single units and high frequency multi-unit hash.

**Cortical control algorithm**

In order to control the motion of the cursor from cortical signals, we created an initial mapping based upon a modified population vector algorithm /6,8,18,25,28,29/. The first step in construction of a population vector is to fit the firing rates of each of the neurons to a linear function of the direction of hand movement in space.

$$\text{Firing Rate} = a_0 + a_1X + a_2Y + a_3Z \quad (1)$$

For each recorded neuron, this produces a vector ($[a_1, a_2, a_3]$) that corresponds to the direction of movement for which the neuron has the highest firing rate. As the direction of movement deviates from this 'preferred' direction, the tuning function predicts that the firing rate will decrease with the cosine of the angle between the movement direction and the neuron's preferred direction. Thus, this tuning function is known as a cosine tuning function.

To estimate the direction of motion of the hand from the firing rates of neurons, one then takes a weighted sum of the preferred directions. In a standard population vector, that weighting depends on the firing rates of the neurons.

We modified the population vector in two crucial ways. First, we fitted two linear tuning functions to the firing rate of each neuron: one tuning function for changes above baseline firing rates, and another tuning function for changes below the baseline firing rate. This accommodated the flattening often observed in the cosine function by the floor at zero spikes/sec. In a cosine tuning function, the inflection point between accelerating and decelerating firing rate is a continuous change that occurs at the mean firing rate. In our tuning function, there is a discontinuity at the mean firing rate, with an abrupt change in the slope of the curve. When we constructed the population vector, we chose one of those two tuning functions according to whether changes in firing rate were increases or decreases from background rate.

Our second modification was to scale the contribution of each neuron to the final population vector according to the proportion of variance in the neuron's discharge that was captured by the fitting equation. In a standard population vector, each neuron with a statistically significant tuning function contributes to the output vector with equal weighting. This means that neurons with poor fits to cosine tuning functions contribute as much to the final output as neurons that fit the tuning profile exactly. We wanted to provide a means whereby neurons with better fit to the tuning functions would contribute more to the final vector. To this end we weighted the contribution from each cell by

$F(fit)^{0.8}$ where F(fit) is the F measuring the goodness of fit of the tuning function, and the power of 0.8 was chosen based on empirical experience.

Once we had established these parameters (tuning profiles for each neuron and weighting factors to be employed for each neuron), we established a mapping between neuronal activation and arm movement and applied that mapping to motion of the cursor in the virtual reality. With its arms restrained, the animal controlled the cursor by modulating its brain activity for several sets of trials using our modified population vector. After 5-10 sets of movements to all eight targets, we started the adaptive population vector algorithm (aPVA).

The aPVA is similar to the backpropagation algorithm used in neural networks /19/ in that it is a supervised learning algorithm. The aPVA uses the difference between the direction to the target and the actual direction of cursor motion to modify both the tuning parameters of the individual neurons ($a_i$ coefficients from Eqn 1) and the weight of their contribution to the population vector (see supplementary materials to /29/ for details).

The adaptive algorithm was designed to look for those cells that were being modulated during the attempts to control the motion of the cursor, and to increase their contribution to the control scheme, while minimizing the contribution of cells that were not being modulated. One might say that the population vector algorithm asks: "How does the brain control arm movement?", whereas the adaptive algorithm asks: "How can the brain best control the motion of the cursor?".

**Maximum likelihood method**

We chose maximum likelihood estimation (MLE) for analyzing the activity in the observed neurons /2,14,20,31,32/. Our goal in using this method was to see how faithfully the firing in the observed sample of neurons could be used to determine which of the eight target locations was the intended endpoint of the current movement. If we could record from every neuron participating in the behavior, we should have perfect information about the animal's behavior. As it stands, our sample is imperfect in the sense of both having only a very small number of neurons, and the fact that not all of the neurons we sample have close relations to our behavioral task. Thus, in part to compare the effectiveness of various extraction methods, we use MLE to assess the statistical relations between movement and the firing rate of a set of neurons.

The firing rates of $n$ neurons are given as an $n$-dimensional vector $r$. The likelihood of observing a particular firing rate in neuron i is given as $L(r_i)$. The MLE method requires finding a condition $d_j$ (e.g. movement direction) that maximizes the likelihood of observing a particular set of firing rates:

$$d = \arg\max_j \left[ L(r \mid d_j) \right] \quad (2)$$

Since $r$ is the firing rate of $n$ neurons, solving Eqn (1) requires finding the joint probabilities of the firing rates for each of the conditions sampled. Even if firing rates are described only as 'high' and 'low', this quickly becomes a difficult problem, with $2^n$ possible firing rate vectors to be sampled. With 20 neurons, there are over a million possible outcomes for $r$. At best we can acquire 1000 samples of that space - a very sparse sampling of a 20-dimensional space.

Instead, we used the multivariate normal probability density function (PDF). This function assumes only that the firing rates of each cell follow a Gaussian distribution. While this assumption is not perfect, it provides an adequate first approximation to the discharge profile of many neurons (see **Results** for more details).

To compute the multivariate normal probability density function requires the mean, $\mu$, of the elements of $r$, and the covariance, $\boldsymbol{C}$, among the elements of $r$. We can compute parameters for each target separately, and then express the multivariate normal PDF as a conditional likelihood function:

$$L(r|d_j) = \frac{1}{(2\pi)^{(n/2)} * |\boldsymbol{C}_j|^{(1/2)}} \exp\left\{ -0.5 * (r - \mu_j)^T (\boldsymbol{C}_j)^{-1} (r - \mu_j) \right\} \quad (3)$$

We compute these values for each of the conditions $d_j$ which we are testing. To find the ML for an individual trial, we find the ensemble firing rate, $r$, for that trial, compute $L(r|d_j)$ for each of the conditions, and then plug the results back into

Eqn (2) to find the condition which provides the maximum likelihood of observing $r$.

## RESULTS

### Data summary

The data used in this analysis come from a total of 21 recording days in two animals. In monkey 'M', the control scheme used between 63 and 75 neurons recorded from motor and premotor cortical areas in both hemispheres. In monkey 'O', there were between 30 and 46 neurons recorded in a single hemisphere. The animals were both well skilled at controlling three-dimensional cursor motion using only brain activity.

During the arm-controlled trials, the animals performed at essentially 100% accuracy. Every trial which had a point-to-point movement from the center would result in a capture of the target sphere. Occasionally the animals would allow a maximum reaction time following presentation of the target to elapse and thus miss a trial.

When the arms were restrained and the brain-controlled task begun, the animals initially performed at around 50% success rates. In only half of the trials were the animals able to use our modified population vector to direct the cursor into a target. However, during the last trials of any given day, following parameter modification with the adaptive PVA, the animals were able to acquire greater than 90% of the targets.

Within the initial span of 3-5 minutes under the aPVA, cortically controlled cursor motion would undergo dramatic improvement. In the first sets of target presentations, the motion of the cursor would typically be dominated by biases in the fitting parameters, and would thus tend to drift continuously in a single direction or along a single axis. Forty to sixty target presentations later, just a few minutes after starting the adaptive PVA, the animal would be able to drive the cursor in arbitrary directions. Our 'early brain control' trials come from this epoch of behavior.

After initial adaptation, the tuning parameters would continue to change, albeit slowly. The animal would work for another 30-90 minutes, showing the ability to continuously grade its control, to reverse directions, and to capture all of the presented targets with a high degree of reliability (see /29/ for more details of task performance).

### Firing rate distributions

To gain some indication of the deviation of firing rate distributions from normal we performed a Kolmogorov-Smirnov test /12,21/. In the upper part of Figure 1 we show the distribution of the Kolmogorov statistic with respect to normal distribution among the data used in this paper. Each of the normal distributions was generated to have the same mean and standard deviation as the firing rate distribution to which it was being compared. All of our firing rate distributions were created using over 1000 samples. With an $n$ of 1000, the Kolmogorov statistic with Lilliefors' correction is significant to a level of 0.99 whenever it exceeds 0.033. Therefore, virtually all of our firing rate distributions showed significant deviation from normality. In most cases, this reflected a skew to the positive, with more firing rates below the mean than expected for a Gaussian distribution. This is illustrated in the cumulative density functions (cdf) shown at the bottom of Figure 1 for three representative neurons, ordered in increasing deviation from normal distributions from left to right. In each of these plots, the data are illustrated as a solid line, and the cdfs corresponding to normal curves with the same mean and standard deviation are shown by dashed lines.

In the leftmost neuron, the only real difference between the two distributions lies in a significant number of samples in which no spikes were recorded. This is reflected as a difference between the two distributions only at very low firing rates. In both of the other cases, there is a larger separation between the two distributions, and in both cases it is because the firing rates are distributed more heavily to low firing rates than predicted by a Gaussian tuning function. Despite these deviations, the mass of the firing rate distributions fall under a normal distribution, and we use the multivariate normal method with this caveat in mind.

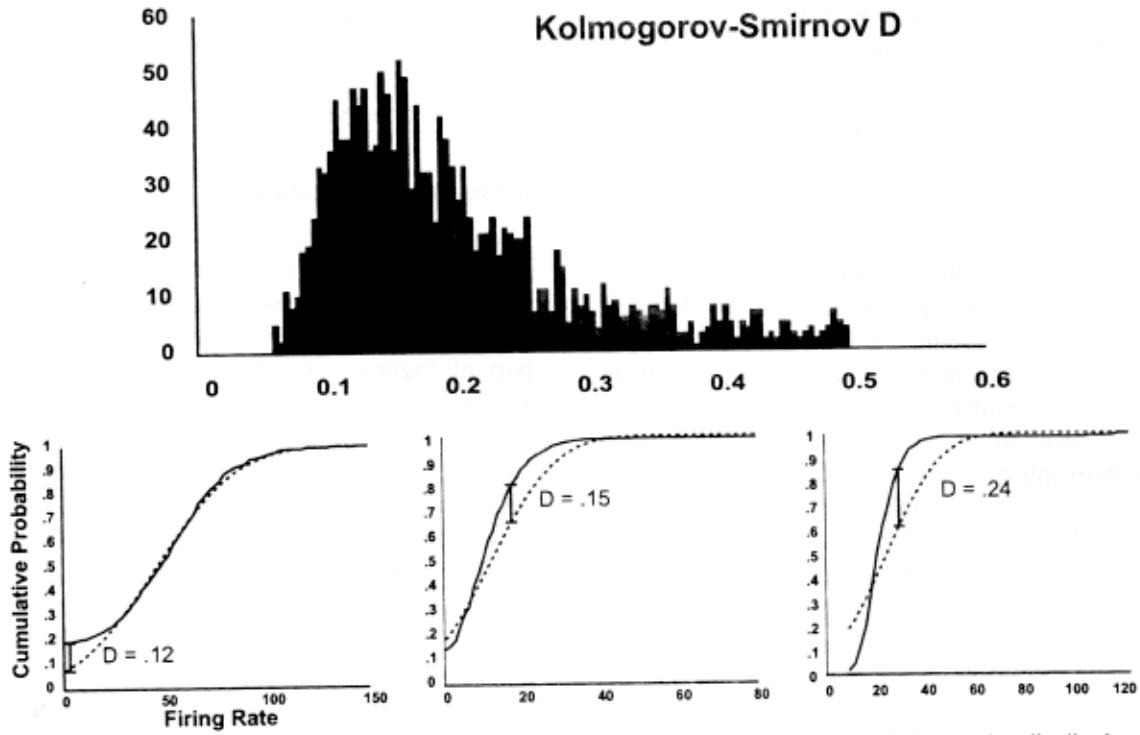


**Fig. 1: Goodness of fit between firing rates and normal distributions.** The upper panel shows the distribution of the Kolmogorov statistic between the firing rate distributions for the data used in this paper and normal distributions. In the lower panel we show the cumulative distribution functions (cdf) for three neurons with varying degrees of fit to a normal distribution. Solid lines show the cdfs for the data, dashed lines show cdfs for corresponding normal distributions.

### Maximum likelihood

The maximum likelihood method begins by computing mean firing rates for all of the cells for movements to each of the eight targets ($\mu_j$ from Eqn (3)), and the covariance matrix ($\boldsymbol{C}_j$) among the neurons for the same sets of movements. We computed the firing rate initially as simply the number of spikes that occurred between the target onset and target capture divided by the elapsed time.

In the cases that we describe here, the parameters for Eqn 3 were computed from the first set of five movements to each of the eight targets in the epoch being analyzed. We use three data epochs in the analysis described here. The first set of data comes from trials in which the cursor motion was driven by arm movements. The second and third data sets come from trials of direct cortical control of cursor motion. The early brain control trials are from the first 10 minutes of brain control. During this phase of the experiment, the mapping from brain activity to cursor motion is under adaptation, but the animal's performance, measured as the number of successful trials, is quite poor. The final data set is the late brain control trials, in which the population vector is fully adapted, and the animal is able to drive the cursor into the target reliably and continuously for extended periods.

Figure 2 shows average firing rates for two opposite movement directions for one day's worth of arm movements in monkey 'M'. We were recording 65 cells on this day. Shown as points and bars are the mean and standard deviation of the firing rates observed as the animal moved the cursor from the center position to the target located down, lateral, and towards the monkey. The dashed line overlaid on the plot shows the mean firing rates of the same cells for cursor movements in the opposite directions.

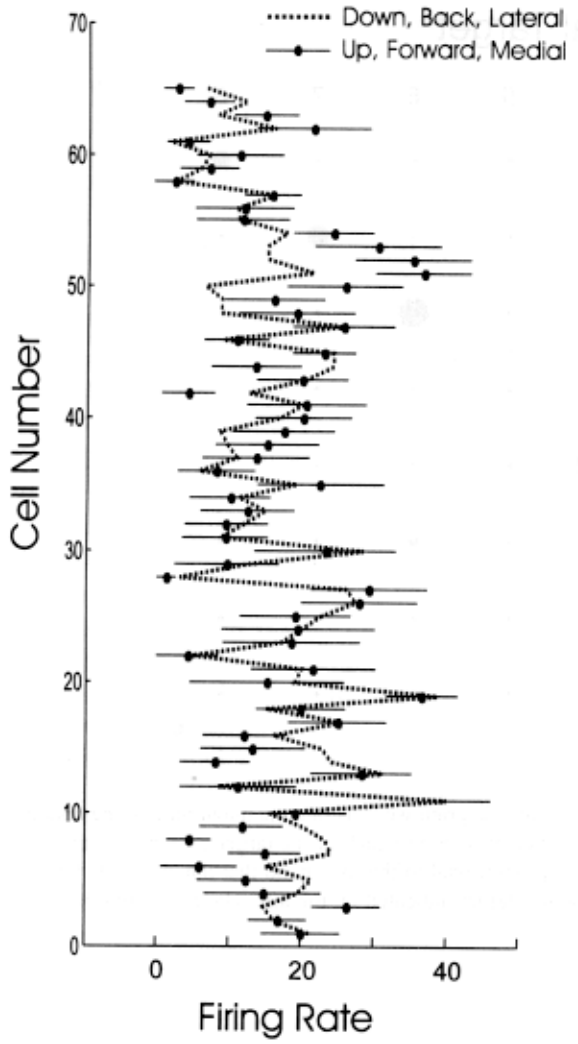


**Fig. 2: Firing rates across a neuronal ensemble.** The mean firing rates for 65 recorded neurons are shown for movements to two oppositely oriented targets. The black dots show the rates for movements to target 4, and the dashed line for movements to target 1. The bars on the target 4 movements are standard deviations of the firing rates for those movements.

Clearly not all the cells modulated their firing rates as a function of movement direction. For this particular day's data, the variance in firing rate was significantly modulated by movement direction in 28 of the neurons ($p < 0.01$ as measured by ANOVA). This is reflected in Figure 2 by the separation of the mean firing rates between the two conditions. The greater the degree of separation between the mean firing rates for each condition and the smaller the covariance between any two neurons, the more that neuron's firing rate is going to contribute to determining which target the animal is moving towards.

Even with reasonable separations, however, the maximum likelihood method is able to readily classify the targets of movement based on the firing rates observed in the ensemble. In Figure 3 we show the outcome of computing the multivariate normal PDF for a set of individual movements to each of the eight targets using the same day's data shown in Figure 2. Each column in Figure 3 corresponds to a movement to one of the targets (target locations are listed in Table 1). Each row in the figure corresponds to one of the eight conditions tested using the multivariate normal distribution. The diameter of the circles shown at each point is proportional to the likelihood estimate given by Eqn 3. Thus, the third column is for a movement to the target located down, anterior, and medial from the center location. Each circle in the column, starting at the bottom, is drawn with a size proportional to the computed likelihood of the observed ensemble discharge pattern for movements to the indicated test target. Clearly, the condition satisfying Eqn 2 is target 3. Therefore the maximum likelihood estimate in this case chose the correct target. In fact, in each of the eight trials illustrated in Figure 3, the MLE was able to correctly identify the target of movement from the firing rates observed in the set of neurons.

**TABLE I**

Target directions

| Target | Target Direction |
|---|---|
| 1 | down, back, lateral |
| 2 | up, back, lateral |
| 3 | down, forward, medial |
| 4 | up, forward, medial |
| 5 | up, back, medial |
| 6 | down, back, medial |
| 7 | up, forward, lateral |
| 8 | down, forward, medial |

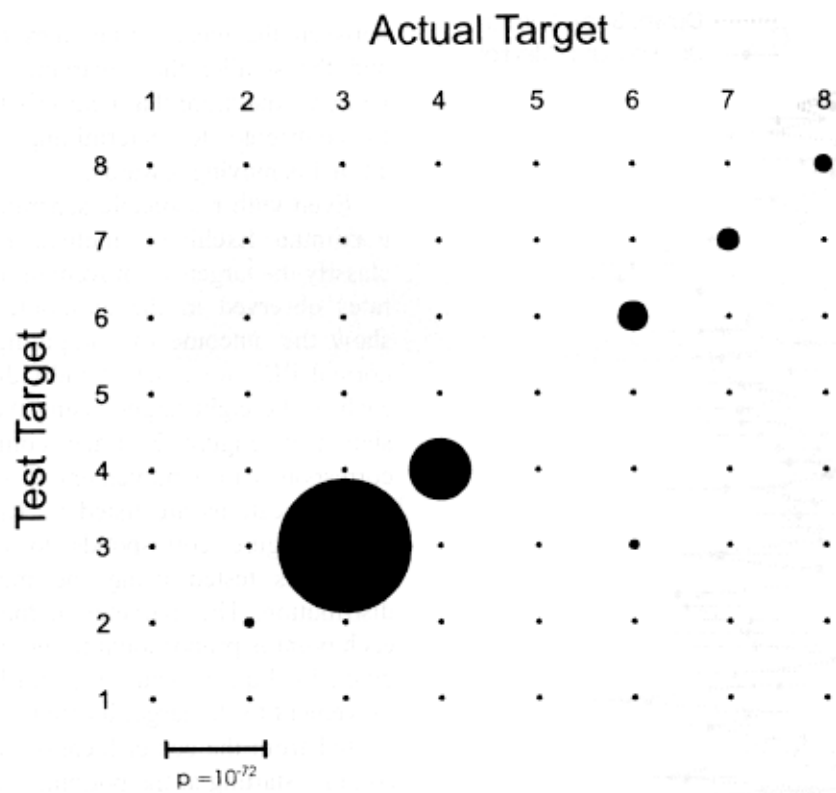


**Fig. 3: Conditional likelihood estimates for observed firing rates.** In this diagram we illustrate the magnitude of the conditional likelihood estimates (Eqn 3) using data recorded from eight movements, one to each of the targets. Each column is from a movement to a particular target. The diameter of each circle is proportional to the likelihood that the ensemble firing rates observed in that movement occurred during movements to the test target indicated on the left. The calibration bar shows a probability of $10^{-72}$.

If this sample contained only one neuron, then the likelihood would correspond to the value of the Gaussian for that cell at the observed firing rate. With multiple uncorrelated neurons, the covariance matrix would only have variances along the diagonal, and computing the joint PDF would correspond to multiplying the probabilities for each of the individual cells together in one large product. In general, however, the firing rates between neurons will exhibit some degree of correlation. In our case, the covariance matrix was never simply diagonal. The average covariances between neurons were typically within an order of magnitude of the variances for the individual neurons. In the case of the data shown in Figures 2 and 3, the average variance ($\sigma_j^2$) in the firing rates for the neurons is 47.2, and the average covariance between neurons ($\sigma_i\ \sigma_j$) was 3.45. The correlation between neurons acted as a weighting factor in Eqn 3. The more highly two neurons were correlated, the more the contribution to the final likelihood of two neurons would be combined. If two neurons are perfectly correlated, then the two together would only contribute one probability to the final likelihood.

The result of multiplying together all these individual probabilities to produce the probability of observing a particular set of firing rates across the entire ensemble is predictable. The final resulting number is typically vanishingly small. The scale bar at the bottom of Figure 2 indicates a mean probability of $10^{-72}$. The largest value we observed on computing Eqn 3 in this data set was roughly $2\text{x}10^{-20}$.

**Changes in neural code for neuroprosthetic control**

As the adaptive algorithm runs, it modifies the weighting coefficients for the population vector, but it also modifies the tuning parameters of the individual neurons. In Figure 4 we show the differences between the arm-control directional parameters and the final brain control parameters. Each line traces a path along the surface of a unit sphere from the preferred direction under arm control to the preferred direction under brain control (circle ends). In animal 'M', we recorded cells from implants in both hemispheres, thus producing neurons both ipsi- and contralateral to the arm used in the arm-control task. Open circles are for contralateral neurons. All of the neurons shown in the figure were significantly tuned in both tasks.

Two elements of our results can be seen in the figure. First, there was a wide degree of variation in how the tuning of the neurons changed as the adaptive algorithm adjusted parameters. Most of the neurons exhibited changes in their preferred direction between the arm movement task and the cortically controlled cursor under the adaptive population vector. Preferred directions of some neurons changed very little, whereas other neurons changed preferred directions by nearly 180 degrees. In the data used in this analysis, the average change in preferred directions was between 55° and 95° across the recorded ensembles. Second, we did not observe any systematic changes in the preferred directions. There was no clear tendency for the preferred directions to rotate in any predetermined fashion.

Nonetheless, the adaptive algorithm did produce remarkably improved performance in the brain-control task /29/. In the arm-control task, we could easily see when the animal was not working at the task. It was not so easy in the cortical control tasks as the animal was fully restrained and the cursor continued to move. Therefore assigning an overall success rate to the brain-control tasks involves a certain degree of guesswork. However, using the starting population vector, the animals could drive the cursor into the target in as few as 50% of the trials. Following adjustment of tuning parameters, the animals could acquire the targets with 100% success during task performance spanning 5-10 minute blocks.

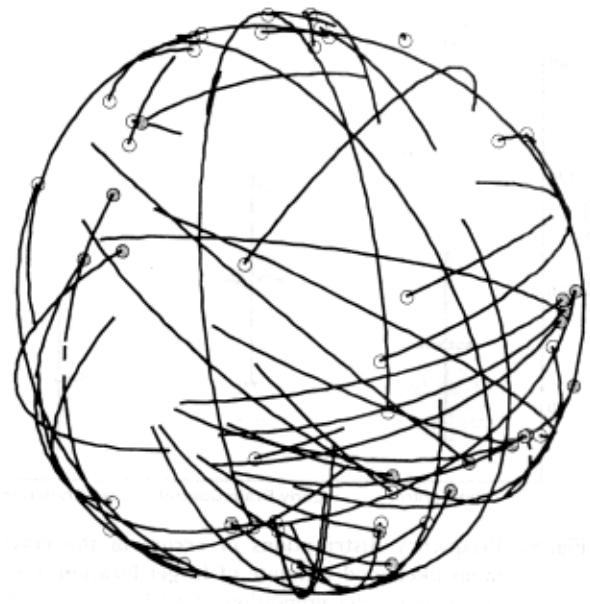

**Fig. 4: Changes in preferred direction between arm control and final brain control.** Each line is the projection onto a unit sphere of the change in preferred direction of an individual neuron between the arm-control task (blank end of line) and the adapted brain-control parameters (indicated by circles). Open circles are for neurons contralateral to the arm used in the arm control task, and filled circles are for ipsilateral neurons. All neurons shown had significant directional tuning in both the arm-control and the brain-control tasks.

In Figure 5 we show the average results of the ML analysis performed on data broken into three separate blocks: arm control, early brain control (the first ten complete brain-controlled movements to the set of eight targets), and late brain control (final ten complete brain-controlled movements to the set of eight targets). For each of those blocks of trial, we used the ML method to estimate, from the ensemble brain activity alone, which target was the target of the movement.

In the arm-control trials, we were able to determine the target of movement using the ML method, on average, in 65% of the trials. This was not because of inaccuracy in performance. The animals performed, effectively, at 100% accuracy

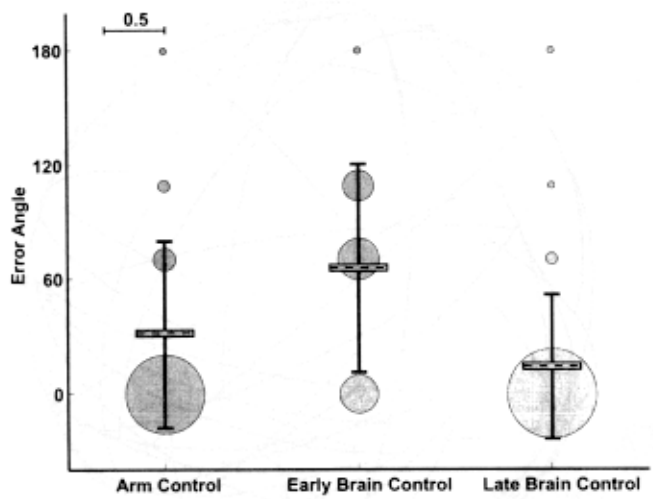


**Fig. 5: Frequency distributions of errors in the maximum likelihood estimate of target location.** Each circle shows the proportion of trials in which the maximum likelihood method estimated that the target of a trial fell the indicated distance from the actual target. In the 3D center→out task, targets can be separated by 70°, 110°, or 180°. In both the arm control and late brain control, the errors were well behaved in the sense that the ML method tended to estimate a nearer, rather than a more distant, target.

in the brain-control trials. Instead, this reflects a lack of information contained in the sampled ensemble of neurons. The activity in the small ensembles did not contain complete information about the target.

It is important to note that this statement does not assume anything about the specific relations between the firing rate of individual neurons and the direction of movement. The only requirement for tuning profiles for our ML analysis is that some proportion of the neurons show firing rates that change between the eight targets.

When we analyzed the early brain-control trials with the ML method, we were able to predict the target of the movement with only 35% accuracy on average. The animal's performance on this task dropped to 45-50% correct trials. On average, however, the animal's performance remained substantially better than the performance of the ML method, which could identify the correct target of movements in successful trials only 35% of the time.

In the late brain-control epoch, the animals could acquire the correct target in virtually all of the trials, and the ML method was able to correctly identify the target of the movement in 80% of the successful trials.

One element that might be expected of a reasonable method for decoding neuronal signals is a degree of orderliness in estimation errors. When the ML method makes an error in determining which is the current movement target, are those errors randomly distributed among the other seven targets, or is there some degree of orderliness that biases the errors to neighboring targets?

In the 3D center→out task that we used in this experiment, targets can have three degrees of separation. Targets on neighboring vertices of the cube (e.g. 1 and 2, 3 and 4) are separated by 70°. Targets on the next level of separation (e.g. 1 and 3, 2 and 4) are separated by 110°, and of course opposite targets (e.g. 1 and 4, 2 and 3) are separated by 180°.

In Figure 5 we plot frequency distributions of the errors made by the ML method for each of the three conditions of the task: arm control, early brain control, and late brain control. The circles indicate by diameter how many times the ML method classed trials into targets with a particular separation from the actual target. The dashed line and error bars give the average angular error between target and ML-estimated target for each condition.

In the arm control phase of the task, even though we could only accurately identify the target in 65% of the successful trials, the remainder of the distribution is well behaved. In nearly 60% of the remaining trials, the ML method assigned one of the nearest neighbor targets, and the average angular error was 35°.

In the early trials of brain control, the animals' performance was poor. The ML method could only correctly categorize the target of each trial in 33% of cases. More interestingly, the errors in the ML method under this condition were far more widely dispersed. Of the remaining trials, 50% were assigned to the nearest neighbor target, and almost 40% to the targets once removed. The average angular error was 62°.

In the late brain control trials, the animals performed much better, as did the ML method. In this case, 83% of trials were correctly identified. The remaining 17% of trials were assigned 60% to the nearest neighbor, and 29% and 11% to targets of 110° and 180° of separation.

## DISCUSSION

We trained two animals to control the motion of a cursor in three dimensions directly from the brain /29/. There were two key elements to this training. First, we created a closed-loop environment in which the animals were able to observe the effect of brain activity on the motion of the cursor. This allowed the animals to learn and practice the changes in neuronal firing that provided for controlled motion of the cursor into the targets.

Second, and at least as important, our method for generating a mapping between brain activity and cursor motion did not rely upon providing a best-possible estimate for what the animal intended to do with its arm. Instead, the supervised learning algorithm described by Taylor and colleagues /29/ allows the specific form of the brain activity-cursor motion mapping to be determined by the changes in brain activity that are observed during attempts to control the cursor.

This is reflected in the results of the maximum likelihood analysis of the firing patterns observed in each of the components of this task. In computing the multi-variate normal, we relied on the assumption that the firing rate observed multiple times in a single task condition for an individual neuron is normally distributed /1/. Importantly, the multi-variate normal distribution takes into account the correlations among neurons in the sample. This is a crucial element of any such analysis as it prevents neurons with high degrees of correlation from contributing independently to the estimate of the probability.

The probabilities computed by the multi-variate normal distribution thus form an otherwise assumption free estimate of how the neurons respond during this task. For neurons that do not change their firing rates in any systematic way with movements to the eight targets, the means measured across the various task conditions tend to be the same. On the other hand, using a simple measure of the mean firing rates during movements to each of the eight targets does not constrain the tuning profiles of the neurons into any particular form such as a cosine tuning function. Therefore, this method of computing a maximum likelihood estimate should produce a truer representation of the information contained within the ensemble discharge than methods which rely on such assumptions /27/.

Our ML method has shown that the information about the target in an ensemble of neurons does not need to be that high before it is possible to create a neuroprosthetic system that provides fine control of a device.

There are many possible reasons why using an arm movement generated mapping between firing rates and device motion does not provide for the best possible control. Two are worth brief mention here.

First, it is well known that a method such as the population vector requires input from approximately 100 well-tuned neurons, and those neurons must have preferred directions that span the movement space with a high degree of uniformity /7,8/. As typical in chronic implants, we were able to obtain well-tuned neurons on no more than 80% of our microwires. More typically, we found well-tuned neurons on 30-50% of the recorded channels. For implants with 64 wires, that could be as few as 20 well-tuned neurons. Reconstruction of arm movements was even further complicated by the lack of uniformity in the distribution tuning directions.

The adaptive PV allowed preferred directions and tuning quality to change, and in fact we observed substantial improvements in the goodness-of-fit of many neurons in the sample (see /29/ for details of these results). The changes in preferred directions may have also improved the distribution of directions coded across the movement space. These two aspects together (improvements in tuning, and improvements in the uniformity of tuning distributions) would have improved the behavior of the population vector.

Alternatively, the PV based upon arm movements selects directly for neurons that are most systematically modulated in relation to arm move-

ments. This maximizes the quality of the population vector estimate of direction of arm movement. However, it may also have two deleterious effects in designing a control scheme. It could prevent a control algorithm from making use of neurons that the animal can reliably control outside the context of arm movements. It could also force the animal to modulate the activity of neurons that the animal has difficulty accessing outside the same context.

The adaptive PV described by Taylor *et al.* /29/ loosens these constraints, and allows a subject to control the prosthetic device using modulation patterns of his choice among recorded neurons of his choice. We feel that this is a crucial distinction. Several laboratories have described efforts to establish better mappings between the ensemble firing rate of motor cortical neurons and arm movement /9,10,13,17,25,26,29,30/. These efforts fall under two headings. Traditionally, these analytical methods are used to provide an improved understanding of the functions of the motor cortex in driving arm movements. In the more recent context, the methods are being applied to the field of neuroprosthetics with an underlying assumption that the more information an algorithm extracts from a set of neurons, the better a user will be able to control a cortically controlled prosthetic system.

Extracting increasing information from a recorded ensemble of course comes at an increased computational price. We set out to find a measure of the information contained within an ensemble so that we could gain a specific estimate of the information that is available to be extracted. This would allow us to make a quantitative statement about the relative merits of different extraction techniques, and thus be in a better position to assess the price being paid for that information.

However, applying the maximum likelihood method to the firing rates observed when animals were learning to control a prosthetic device from cortical signals illustrated for us a key principal for neuroprosthetic system design: producing an optimal estimate, or read-out, of the intention of an animal is not a necessary prerequisite for creating a well-controlled device. Conversely, tying a neuroprosthetic control algorithm too closely to firing rate-arm movement relationships observed during normal arm movements can limit the fine controllability of a device. The key goal in design of a neuroprosthetic system is to create a system that a subject can control.

## ACKNOWLEDGEMENTS

The results reported here have been presented in preliminary form /11/. This work was supported by the Whitaker Foundation, and by PHS contracts #N01-NS-6-2347 and #N01-NS-9-2321.

## REFERENCES

1. Amemori KI, Ishii S. Gaussian process approach to spiking neurons for inhomogeneous Poisson inputs. Neural Comput 2001; 13: 2763-2797.
2. Brown EN, Frank LM, et al. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. J Neurosci 1998; 18: 7411-7425.
3. Chapin JK, Moxon KA, et al. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. Nat Neurosci 1999; 2: 664-670.
4. Fetz EE, Finocchio DV. Operant conditioning of specific patterns of neural and muscular activity. Science 1971; 174: 431-435.
5. Fetz EE, Finocchio DV. Operant conditioning of isolated activity in specific muscles and precentral cells. Brain Res 1972; 40: 19-23.
6. Georgopoulos AP, Caminiti R, et al. Spatial coding of movement: a hypothesis concerning the coding of movement direction by motor cortical populations. Exp Brain Res 1983; 7 (Suppl): 327-336.
7. Georgopoulos AP, Kettner RE, et al. Primate motor cortex and free arm movements to visual targets in three-dimensional space II Coding of the direction of movement by a neuronal population. J Neurosci 1988; 8: 2928-2937.
8. Georgopoulos AP, Schwartz AB, et al. Neuronal population coding of movement direction. Science 1986; 233: 1416-1419.
9. Hatsopoulos NG, Harrison MT, et al. Representations based on neuronal interactions in motor cortex. Prog Brain Res 2001; 130: 233-244.
10. Hatsopoulos NG, Ojakangas CL, et al. Information about movement direction obtained from synchronous activity of motor cortical neurons. Proc Natl Acad Sci USA 1998; 95: 15706-15711.
11. Helms Tillery SI, Taylor DM, Schwartz AB. A maximum likelihood measure applied to signals used in controlling a neural prosthesis. Soc Neurosci Abstr 2002; 357.9.

12. Hollander M, Wolfe DA. Nonparametric Statistical Methods. New York-Chichester: Wiley, 1999.
13. Isaacs RE, Weber DJ, et al. Work toward real-time control of a cortical neural prosthesis. IEEE Trans Rehabil Eng 2000; 8: 196-198.
14. Jensen O, Lisman JE. Position reconstruction from an ensemble of hippocampal place cells: contribution of theta phase coding. J Neurophysiol 2000; 83: 2602-2609.
15. Kennedy PR, Bakay RA. Restoration of neural output from a paralyzed patient by a direct brain connection. NeuroReport 1998; 9: 1707-1711.
16. Kennedy PR, Bakay RA. Direct control of a computer from the human central nervous system. IEEE Trans Rehabil Eng 2000; 8: 198-202.
17. Maynard EM, Hatsopoulos NG, et al. Neuronal interactions improve cortical population coding of movement direction. J Neurosci 1999; 19: 8083-8093.
18. Moran DW, Schwartz AB. Motor cortical representation of speed and direction during reaching. J Neurophysiol 1999; 82: 2676-2692.
19. Plaut DC, Nowlan SJ, et al. Experiments on learning by back propagation. Comp Sci Tech Report CMU-CS-86-126, 1986.
20. Pouget A, Zhang K, et al. Statistically efficient estimation using population coding. Neural Comput 1998; 10: 373-401.
21. Press WH, Flannery BP, et al. Numerical Recipes in C: The Art of Scientific Computing. Cambridge: Cambridge University Press, 1988.
22. Reina GA, Moran DW, et al. On the relationship between joint angular velocity and motor cortical discharge during reaching. J Neurophysiol 2001; 85: 2576-2589.
23. Schmidt EM, Bak MJ, et al. Operant conditioning of firing patterns in monkey cortical neurons. Exp Neurol 1977; 54: 467-477.
24. Schmidt EM, McIntosh JS, et al. Fine control of operantly conditioned firing patterns of cortical neurons. Exp Neurol 1978; 61: 349-369.
25. Schwartz AB, Taylor DM, et al. Extraction algorithms for cortical control of arm prosthetics. Curr Opin Neurobiol 2001; 11: 701-707.
26. Serruya MD, Hatsopoulos NG, et al. Brain-machine interface: instant neural control of a movement signal. Nature 2002; 416: 141-142.
27. Seung HS, Sompolinsky H. Simple models for reading neuronal population codes. Proc Natl Acad Sci USA 1993; 90: 10749-10753.
28. Taylor DM, Schwartz AB. Using virtual reality to test the feasibility of controlling an upper limb FES-system directly from multi-unit activity in motor cortex. Proceedings 6th Annual International Functional Electrical Stimulation Society, Cleveland, Ohio, 2001.
29. Taylor DM, Tillery SI, et al. Direct cortical control of 3D neuroprosthetic devices. Science 2002; 296: 1829-1832.
30. Wessberg J, Stambaugh CR, et al. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. Nature 2000; 408: 361-365.
31. Wu S, Nakahara H, et al. Population coding with correlation and an unfaithful model. Neural Comput 2001; 13: 775-797.
32. Zhang K, Ginzburg I, et al. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. J Neurophysiol 1998; 79: 1017-1044.