

# Latent Inputs Improve Estimates of Neural Encoding in Motor Cortex

Steven M. Chase,<sup>1,2,3</sup> Andrew B. Schwartz,<sup>2,3</sup> and Robert E. Kass<sup>1,3</sup>

<sup>1</sup>Department of Statistics, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, <sup>2</sup>Department of Neurobiology, University of Pittsburgh, Pittsburgh, Pennsylvania 15213, and <sup>3</sup>Center for the Neural Basis of Cognition, University of Pittsburgh and Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

Typically, tuning curves in motor cortex are constructed by fitting the firing rate of a neuron as a function of some observed action, such as arm direction or movement speed. These tuning curves are then often interpreted causally as representing the firing rate as a function of the desired movement, or intent. This interpretation implicitly assumes that the motor command and the motor act are equivalent. However, any kind of perturbation, be it external, such as a visuomotor rotation, or internal, such as muscle fatigue, can create a difference between the motor intent and the action. How do we estimate the tuning curve under these conditions? Furthermore, it is well known that, during learning or adaptation, the relationship between neural firing and the observed movement can change. Does this change indicate a change in the inputs to the population, or a change in the way those inputs are processed?

In this work, we present a method to infer the latent, unobserved inputs into the population of recorded neurons. Using data from nonhuman primates performing brain–computer interface experiments, we show that tuning curves based on these latent directions fit better than tuning curves based on actual movements. Finally, using data from a brain–computer interface learning experiment in which half of the units were decoded incorrectly, we demonstrate how this method might differentiate various aspects of motor adaptation.

## Introduction

One of the most fundamental roles of the CNS is to generate behavior. Behavior is expressed as movement, and major advances have been made in elucidating neural correlates of movement execution (Evarts, 1968; Georgopoulos et al., 1982; Kakei et al., 1999; Cisek et al., 2003; Churchland et al., 2006). Movement execution, however, is the last in a series of cognitive steps that make up a motor act: for simple reaching, it is preceded by visual input, goal identification, and the formation of a movement command (Fig. 1). In this paper, we focus on the differentiation of the motor “action” from the motor “input.” Although the former is readily measured and parameterized, the latter is a latent property that cannot be observed directly. This input is distinct from the task goal: although the goal is typically to hit the target, the input is the signal that drives the neurons such that the goal may be accomplished.

Although the neural correlates of action have been studied extensively, the drivers of motor cortical activity are not well characterized. This is mostly related to the ease of collecting action-related parameters versus the difficulty of identifying the covert factors that drive movement. With the development of brain–computer interfaces (BCIs), this issue has become acute. These devices translate cortical activity into various

types of action, such as cursor (Taylor et al., 2002; Hochberg et al., 2006; Mulliken et al., 2008) or robotic arm (Chapin et al., 1999; Wessberg et al., 2000; Velliste et al., 2008) movement. The BCI paradigm allows us to study the input–action interface by observing populations of neurons directly responsible for the generated movement.

Perturbations create an inherent disconnect between intention and action and can help us determine the drivers of motor cortical activity. Several groups have shown that firing rates in motor cortex change during motor adaptation (Wise et al., 1998; Li et al., 2001; Paz and Vaadia, 2004; Arce et al., 2010). These firing rate changes have two potential sources. First, the input to the motor cortex could change as the subject attempts to produce a movement to counter the perturbation. We call this re-aiming. Second, the tuning curves of individual motor cortical cells could change through some kind of plastic mechanism that allows the subject to relearn the association between input and action under the perturbation state. We call this retuning. In this paper, we propose an algorithm that associates latent directions with the observed cursor movements and then fits tuning curves to those latent directions. Using data from several different BCI experiments, we demonstrate that tuning curves based on these latent directions have less residual error than tuning curves based on the actual cursor movements and are also more consistent across different experimental contexts. Finally, we reevaluate data taken during a BCI visuomotor adaptation task (Jarosiewicz et al., 2008) and demonstrate that their results reflect both compensatory re-aiming and retuning processes.

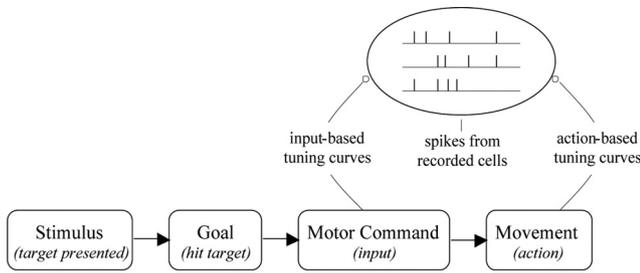
Received May 6, 2010; revised Aug. 6, 2010; accepted Aug. 13, 2010.

This project was supported by National Institutes of Health Grant R01EB005847 from the Collaborative Research in Computational Neuroscience program.

Correspondence should be addressed to Dr. Steven M. Chase, McGowan Institute, Room 245, 3025 East Carson Street, Pittsburgh, PA 15203. E-mail: schase@pitt.edu.

DOI:10.1523/JNEUROSCI.2325-10.2010

Copyright © 2010 the authors 0270-6474/10/3013873-10\$15.00/0



**Figure 1.** A schematic of the processing steps that occur during a reaching task. Here we differentiate the motor input, a latent signal that causes motor cortical cells to fire, from the movement itself. Tuning curves derived by regressing spike data against parameters related to the action are called action-based tuning curves. Tuning curves derived by regressing spike data against estimates of the input are called input-based tuning curves or latent tuning curves. In this study, we demonstrate how to derive these latent tuning curves and show that they provide better estimates of firing rates than the action-based tuning curves.

## Materials and Methods

**Data collection.** All procedures were performed in accordance with the guidelines of the Institutional Animal Care and Use Committee of the University of Pittsburgh. Data from three types of BCI experiments are used to validate the algorithm and demonstrate its usefulness in various brain-control contexts: standard two-dimensional (2D) center-out data, three-dimensional (3D) perturbation data, and 2D decoder comparison data. The 3D perturbation data are from Jarosiewicz et al. (2008), the 2D decoder comparison data are from Chase et al. (2009), and the standard 2D center-out data are new. All three sets of experiments use similar methods, described below. More complete methods can be found in the original studies.

A total of three male Rhesus monkeys (*Macaca mulatta*) were used in this work. Two were implanted with four or more 16-channel intracortical electrode arrays (50  $\mu\text{m}$  Teflon-coated tungsten wires, arranged in  $2 \times 8$  grids with 300  $\mu\text{m}$  spacing), whereas the other was implanted with a 96-channel Utah array (Cyberkinetics Neurotechnology Systems). All implantations were visually placed in the proximal arm area of primary motor and/or premotor cortex. Recordings were amplified, filtered, and sorted online with a 96-channel Plexon MAP system. In each case, some of the units recorded were well-isolated single cells, and others contained two or more cells that could not easily be isolated from one another but which were nevertheless tuned to intended movement direction as a group.

**Tuning curve fitting.** We consider the observed spike counts to be realizations of an underlying random process whose intensity function changes as a function of direction. Let  $F_i$  represent the random process responsible for spike generation in cell  $i$ , and  $f_{i,k}$  represent the observed spike count from cell  $i$  to target presentation  $k$ . For these analyses, we assumed that the underlying random process was Gaussian, and the tuning curves  $\lambda_i = E[F_i]$  were linear functions of direction:

$$\lambda_i = b_{0i} + m_i \vec{p}_i \cdot \vec{a}_k, \quad (1)$$

where  $b_{0i}$  is the baseline firing rate of the cell,  $m_i$  is its modulation depth (a scalar), and  $\vec{p}_i$  its preferred direction (PD) (a unit vector). Depending on the context, the unit vector  $\vec{a}_k$  could point in the direction of target  $k$ , point in the direction of the  $k$ th cursor movement, or represent the direction of an unobserved latent input; this will be described more fully below. In each case, we fit these tuning curves using ordinary least-squares regression. For notational convenience, we refer to the true tuning curve parameters as  $b_{0i}$ ,  $m_i$ , and  $\vec{p}_i$ , their estimated values as  $\hat{b}_{0i}$ ,  $\hat{m}_i$ , and  $\hat{\vec{p}}_i$ , and the corresponding parameters used in the decoder (discussed in the next section) as  $b_{0i}^D$ ,  $m_i^D$ , and  $\vec{p}_i^D$ .

**Establishing the BCI.** Establishing the BCI involves three steps: choosing an encoding model that describes how movement is represented in the firing rates, choosing a decoding algorithm for mapping those firing rates back into cursor movement, and performing a calibration to fit the parameters required by the decoding algorithm. As discussed above, we

assumed a linear encoding model for the BCI in which the tuning curves were functions of direction only.

All three of the datasets described here used the population vector algorithm (PVA) (Georgopoulos et al., 1986) for translating firing rates into cursor movements. In addition, the decoder comparison experiment (discussed below, *Behavioral tasks*) also used the optimal linear estimator (OLE). The OLE was implemented the same way as the PVA except that the preferred direction decoding parameters were different. For the PVA, the decoding parameters were set equal to the estimated tuning curve parameters, that is,  $b_{0i}^D = b_{0i}$ ,  $m_i^D = m_i$ ,  $\vec{p}_i^D = \vec{p}_i$ . For the OLE,  $b_{0i}^D$  and  $m_i^D$  were the same as for the PVA, and the decoding PDs were calculated as  $\vec{p}^D = (\hat{P}^T \hat{P})^{-1} \hat{P}^T$ , where  $\hat{P}$  is the matrix formed by gathering all of the PD estimates  $\hat{\vec{p}}_i$  together. For details, see Chase et al. (2009).

Once the decoding parameters were known, they were used to move the cursor as follows. Spike counts were binned into 33.3 ms intervals and converted to rates  $f_i(t)$  by dividing by the sampling interval. Normalized rates  $r_i(t)$  were computed through the equation

$$r_i(t) = \frac{f_i(t) - b_{0i}^D}{m_i^D} \quad (2)$$

and then smoothed with a five-point boxcar filter (i.e., by averaging the rates from the last five bins). These filtered, normalized rates were then converted to cursor velocity,  $\vec{v}(t)$ , as follows:

$$\vec{v}(t) = k_s \frac{n_D}{N} \sum_{i=1}^N r_i(t) \vec{p}_i^D, \quad (3)$$

where  $n_D$  represents the number of movement dimensions (either two or three), and  $N$  is the number of units used for decoding. The speed factor,  $k_s$ , is a parameter set by the experimenter that converts the magnitude of the population vector from a normalized range to a physical speed; in these experiments, chosen values ranged from 65 to 80 mm/s. Finally, cursor position,  $\vec{C}_p$ , was updated every sampling interval as

$$\vec{C}_p(t) = \vec{C}_p(t - \Delta t) + \Delta t \vec{v}(t). \quad (4)$$

Trajectories always started at the origin.

As expressed above, to perform decoding with these algorithms, the tuning functions of the recorded cells must be known. One possibility is to estimate these tuning curves from data recorded when the subject makes center-out movements with his own arm. We do not do this for two reasons. First, the clinical subjects for whom these BCI devices are intended cannot make volitional movements, and we desired a method that could work in a clinical setting. Second, the tuning functions recorded during actual arm movement are not necessarily the same as the tuning functions used during brain-controlled cursor movement (Taylor et al., 2002; Carmena et al., 2003). We instead calibrated the system in the following way. To initialize the system, the decoding parameters were chosen randomly. Targets for the center-out task were then presented, one at a time in random order, and left on the screen until a movement time-out period elapsed (typically, 1 s). Although the cursor did not move much during this time because of the randomized decoding parameters, firing rates modulated in response to target presentation. Once an entire cycle set consisting of one presentation of each of the targets was completed, the observed spike rates were regressed against target direction to compute the linear tuning function parameters  $\hat{b}_{0i}$ ,  $\hat{m}_i$ , and  $\hat{\vec{p}}_i$ , which were in turn used to calculate a new set of decoding parameters as discussed above. Once the new decoding parameters were calculated, another cycle set of targets was presented, and the process repeated until the monkey was able to complete the center-out task reliably. Typically, only three to five cycle sets of data ( $\sim 2$  min of data collection) are needed to achieve good control. Cells with modulation depths of  $< 4$  Hz were not used for control.

**Behavioral tasks.** The monkeys were trained to sit in a primate chair facing a mirror that reflected an image from a custom-designed stereo monitor (Dimension Technologies) mounted above. This monitor creates virtual 3D images by projecting different columns of pixels to each

eye. Each of the three sets of experiments has a slight variation on the center-out task.

In the standard 2D center-out task, the monkeys made 2D center-out brain-controlled cursor movements to 16 targets equally spaced around a circle of radius 85 mm, centered at the origin. The radii of both the cursor and target were 8 mm, and the movement timeout (the total time the monkey had to move the cursor to the target) was typically  $\sim 2$  s. This timeout period was generous; the median time between target presentation and target acquisition was 889 ms. The corresponding time it takes for a well trained monkey to complete this task with real arm movements is  $\sim 550$  ms (Georgopoulos et al., 1982; Schwartz et al., 1988; Moran and Schwartz, 1999). Thus, although not as fast as real arm movements, the BCI movement times are within approximately a factor of two of real movement times. After either success or failure, the cursor was reset to the origin after an intertrial period of  $\sim 1$  s. Each session typically consisted of between 80 and 160 successful movements; we analyzed 25 different sessions in total.

The decoder comparison experiment is described fully by Chase et al. (2009). Briefly, the monkey again made 2D center-out movements, this time to eight targets equally spaced around a circle of radius 85 mm. Target radii and movement times were the same as for the standard task described above. Movements were made using one of two decoders, the PVA or the OLE. After a number of successful movements (typically, 10–15) had been made to each target with one of the decoding algorithms, we switched the decoder to the other algorithm, without notice, and had the subject perform a similar number of movements with it.

In the 3D perturbation task (Jarosiewicz et al., 2008), the monkey made center-out movements to eight targets placed on the corners of a cube, such that each target was  $\sim 85$  mm from the origin. Cursor and target radii were typically 2.5 cm, larger than in 2D because of the increased task difficulty and because these experiments were performed shortly after implantation, when the monkey was not as proficient at brain control. Movement timeouts were typically  $\sim 3$  s. After the calibration session, a control period was run in which the monkey typically performed  $\sim 100$  successful center-out movements. After the control session, a subset of the units (either 25% or 50%) was randomly selected, and their decoding PDs ( $\vec{p}_i^j$ ) were rotated  $90^\circ$  about a common axis. Typically, upward of 150 successful center-out movements were performed in this perturbation session.

**The latent-target method.** We are faced with the problem of differentiating the subject's input from his actions, given recordings from a population of neurons whose tunings we do not know. We approached this problem by associating an unobserved latent variable with every target that represents the subject's input: his presumed "re-aiming" point or "intended" direction of movement when presented with that target. We then computed the latent tuning curves by regressing the observed firing rates as a function of these latent directions (Fig. 1).

To initialize the algorithm, we started with the action-based tuning curves derived by regressing the firing rates against the average cursor movement direction for each target. Once these initial tuning curves were computed, we used weighted least squares to calculate the most probable latent direction for each target based on the observed spike rates, assuming independent neurons. We then computed estimates of the latent tuning curves using the latent directions as the covariates  $d_k$  in Equation 1. With these new tuning curves, we re-estimated the latent directions again and continued iterating until the points converged.

The weighted least-squares calculation was performed by finding the latent direction,  $\vec{d}_j$ , that minimized the error between the observed spike rates for the  $j$ th target and the spike rates predicted by the tuning curves. That is, we solved

$$\vec{d}_j = \arg \min_{\vec{d}} \left( \sum_{i=1}^N \frac{(y_{i,j} - \lambda_i(\vec{d}))^2}{\sigma_i^2} \right), \quad (5)$$

where  $\lambda_i(d)$  is the tuning curve of neuron  $i$  evaluated at  $\vec{d}$  and  $\sigma_i^2$  is its variance.  $\vec{d}$  was constrained to be a unit vector in this calculation. Note that this is equivalent to a maximum likelihood estimate of the latent

direction under the assumptions of Gaussian noise in the firing rates and independent neurons.

To summarize, the algorithm proceeds as follows: (0) Assume the latent directions are the directions of cursor movement. (1) Fit tuning curves to the latent directions (Eq. 1). (2) Compute a new latent direction for each target (Eq. 5). (3) If convergence is not reached, repeat step 1. We considered the algorithm to have reached convergence when the average tuning curve error decreased by  $<1\%$  on subsequent iterations.

Note that there is a non-identifiability in the solution that emerges: we cannot distinguish a given solution from another in which the PDs of all cells are rotated uniformly in one direction while the latent directions are rotated uniformly in the other direction. The particular solution arrived at is determined by the initialization step. In our case, this means that we arrive at the solution that is, in a sense, closest to the one that assumes the cursor movement directions are the latent directions.

This algorithm generalizes well to alternate assumptions about the shapes of the tuning curves and the noise process underlying neural firing. We have repeated some of the analyses presented in this paper using log-linear tuning curves and a Poisson noise assumption; the results are presented in the supplemental data (available at [www.jneurosci.org](http://www.jneurosci.org) as supplemental material).

Why did we estimate a separate latent direction for each target? In experiments in which subjects learned to perform reaching movements in only one direction under a visuomotor rotation, there was limited generalization of the learned re-aiming to different directions of movement (Krakauer et al., 2000). These data suggest that the amount of re-aiming can vary as a function of direction.

## Results

### An example of why the latent-target method is necessary

Assuming that the target directions are the same as the input movement directions can lead to incorrectly estimated tuning curves. To demonstrate this, consider a case in which two neurons are used to drive a computer cursor through a BCI (Fig. 2). Even for perfectly cosine-tuned neurons, as in Figure 2a, a PVA decoder will not move the cursor in the intended movement direction unless the PDs of the recorded neurons are uniformly distributed (Salinas and Abbott, 1994; Kass et al., 2005; Chase et al., 2009). Assume instead that the two neurons have PDs of  $0^\circ$  and  $45^\circ$ , and consider a movement toward a target located at  $0^\circ$ . When aiming directly at the target, both neurons will be driven above their baseline rates. Therefore, both cells will contribute positive vectors to the population vector, resulting in a cursor movement that will be somewhere between  $0^\circ$  and  $45^\circ$  rather than toward the target. In fact, to make a straight-line movement to a target located at  $0^\circ$ , the subject would have to aim toward a spot located at  $-45^\circ$ , so that only the neuron whose PD is at  $0^\circ$  contributes to the population vector. Figure 2b shows the re-aiming points that would be needed to move to targets spaced every  $45^\circ$  around the origin.

Suppose that the subject learns to correct for this distortion to make straight-line center-out movements by re-aiming toward the points depicted in the outer circle of Figure 2b. The colored + symbols in Figure 2a show the firing rates of the neurons when moving to these targets. If one were to attempt to fit tuning curves to these neurons under the assumption that the target directions (or, equivalently, the actual cursor movement directions) were the inputs, the tuning curves would be distorted. In Figure 2c, the firing rates from Figure 2a are replotted as circles at the target directions. The best-fit cosine tuning functions to these points (solid lines) are quite different from the actual tuning curves (dotted lines).

To avoid this fitting problem, we can leverage the population response (in this case, both neurons) to infer the subject's intended movement (Eq. 5) and fit tuning curves with these new

latent directions, iterating until convergence. In Figure 2*d*, we show the tuning curves and latent directions estimated with the latent-target method. Our algorithm provides estimates that are much closer to the true tuning curves (shown as dotted lines for reference).

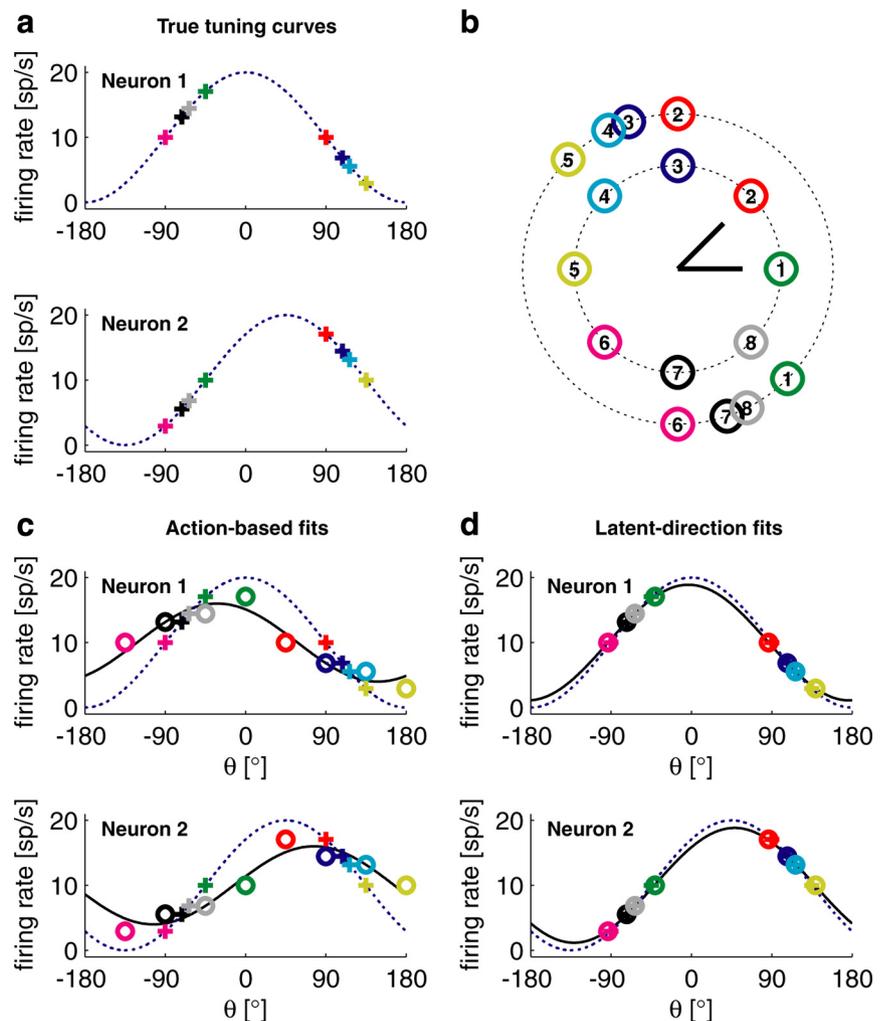
Admittedly, this is a rather extreme example of a highly biased decoder requiring a substantial amount of re-aiming, chosen for illustration purposes. In the next section, we apply the algorithm under more physiological conditions.

### Application of the algorithm to center-out data

To demonstrate the reliability of this algorithm, we applied it to data taken during a standard 2D center-out experiment under brain-control. Twenty-five experiments were analyzed; trajectories from one of these experiments are shown in Figure 3. Spike rates were computed for each trial over the time window starting 150 ms after target presentation and ending when the cursor moved half the distance to the target. This window, typically lasting  $\sim 450$  ms, was chosen because it excludes most of the neural activity that occurs before target-related activity appears in M1 and also excludes most of the neural activity related to online corrective movements; in the supplemental data (available at [www.jneurosci.org](http://www.jneurosci.org) as supplemental material), we also consider shorter windows. We divided the spike rate data into two sets of equal size, a training set and a testing set. The training data were fed into the algorithm that computes latent directions and latent tuning curves. As a comparison, we also used the training data to fit the action-based tuning curves.

We assessed the goodness-of-fit through cross-validation, by computing the error between the firing rates predicted by the tuning curves (fit with the training data) and the firing rates observed in the testing data; the error was computed as the root mean square (RMS) difference between the two. In Figure 4*a*, we plot the error in the latent tuning curve of every neuron as a function of the corresponding error in the action-based tuning curve. In all, the latent tuning curves provided better fits than the action-based tuning curves for 437 of 663 neurons (66%,  $p < 1e-10$ , sign test), leading to an average improvement in fit of  $0.41 \pm 0.04$  Hz ( $p < 1e-10$ ,  $t$  test).

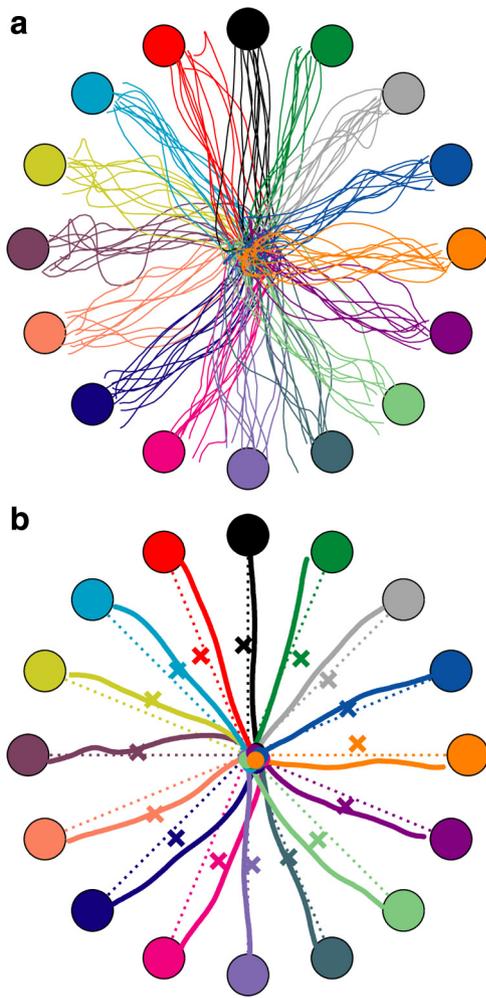
These results suggest that the subjects were not aiming in the direction of cursor movement. Could it be that they were aiming at the targets? Two lines of evidence suggest that they were not only aiming at the targets. Although the latent directions were, on average, closer to the directions of the targets than those of the cursor movements (the average absolute angular difference between the cursor movements and the target directions was  $8.4^\circ \pm 0.3^\circ$ , whereas the difference between the latent and target directions was only  $5.4^\circ \pm 0.2^\circ$ ), a bootstrap analysis reveals that the



**Figure 2.** Ignoring re-aiming can lead to incorrect estimates of tuning curves. *a*, Tuning curves of two hypothetical neurons, with PDs at  $0^\circ$  (top) and  $45^\circ$  (bottom). *b*, Inner circle, Eight targets spaced uniformly around the origin. Outer circle, Re-aiming points necessary to hit the targets in the inner circle using a PVA decoder and the two neurons shown in *a*. Black lines depict the PDs of the two neurons. *c*, If the subject re-aims appropriately, as in the outer circle of *b*, the firing rates of the two cells for each target will be as indicated by the correspondingly colored + symbols in *a*. If these firing rates are assumed to correspond to the target directions (or equivalently, the cursor movement directions) and not the re-aiming points, they will be incorrectly placed at the colored circles. Solid lines indicate the best cosine fits to these incorrectly placed firing rates; the dotted lines and + symbols indicate the true tuning curves and aiming points from *a*, for reference. *d*, Tuning curve fits and latent directions estimated by the latent-target method. Again, the dotted lines show the true tuning curves and aiming points.

difference between the latent directions and the target directions is still much larger than would be expected if the subjects were aiming at the targets ( $p < 1e-4$ ) (for a description of this analysis, see the supplemental data, available at [www.jneurosci.org](http://www.jneurosci.org) as supplemental material). Furthermore, we compared the latent tuning curves to the target-based tuning curves (i.e., the tuning curves fit by assuming the subject was aiming directly at the targets). Figure 4*b* shows the error in these latent tuning curves plotted as a function of the error in the target-based tuning curves. Although the improvement in the tuning curve fit is not as substantial as with the action-based tuning curves, the latent tuning curves still show less error than the target-based tuning curves in 363 of 663 neurons (55%,  $p = 0.01$ , sign test), with an average reduction in error of  $0.16 \pm 0.02$  Hz ( $p < 1e-9$ ,  $t$  test).

With the PVA, the directional distortion between the intended direction of movement and the decoded direction of movement is a function of the uniformity of the PD distribution. Because of limited sampling, the average  $\pm$  SD directional error



**Figure 3.** Typical cursor trajectories during a center-out task under brain control. *a*, All trajectories from one experiment. *b*, Average trajectories from the experiment shown in *a*. The  $\times$  symbols indicate the computed latent direction to each corresponding colored target.

one can expect with 26 cells (the median number used in these experiments) is  $6.3^\circ \pm 3.4^\circ$ , compatible with the  $8.4^\circ$  of error we observed. These data indicate that the latent tuning curves tend to fit better than the action-based tuning curves, even under these relatively mild distortion conditions.

### Assessing tuning curve stability

As another demonstration of the utility of the latent-target method, we can look at tuning curve stability. An open question in motor neurophysiology is whether, and how much, tuning curves drift over time. Some researchers have found that tuning functions of motor cortical neurons can change significantly over the course of 1 or 2 h (Rokni et al., 2007), whereas others have found that tuning curves can be quite stable over a similar time period (Chestek et al., 2007). In a BCI learning study, Ganguly and Carmena (2009a) found that, when they used the same decoding parameters for several days, the directional tuning curves of neurons engaged in a center-out task stabilized over time. However, when they performed a new calibration each day and got a different set of decoding parameters for the same set of neurons, they found that the directional tuning curves of neurons never stabilized.

To investigate these issues, we applied our analysis to data taken in a decoder switching experiment. Over the course of  $\sim 1$

h, we had a subject perform 2D center-out movements under brain control with an OLE decoder and then switched rapidly to a PVA decoder to show that subjects can compensate rapidly (within a few trials) for directional distortions between the two decoders (Chase et al., 2009; Koyama et al., 2009). To quantify the fit disparity between the two sessions, we fit tuning curves during each session and calculated how well they fit the firing rate data observed during the other session. That is, the disparity  $D$  for each unit was calculated as follows:

$$D = \sqrt{\langle (\hat{r}_{\text{PVA}}(\vec{d}_{\text{OLE},k}) - r_{\text{OLE},k})^2 + (\hat{r}_{\text{OLE}}(\vec{d}_{\text{PVA},k}) - r_{\text{PVA},k})^2 \rangle}, \quad (6)$$

where  $r_{\text{OLE},k}$  represents the observed firing rate to target  $k$  during the OLE session, and  $\hat{r}_{\text{PVA}}(\vec{d}_{\text{OLE},k})$  represents the predicted firing rate to that target using the tuning curve fit during the PVA session and the latent direction calculated during the OLE session. The angled brackets indicate an average across targets. To compute the action-based fit disparity, we assumed the latent directions were the same as the actual cursor movement directions. The larger the disparity, the larger the difference between the tuning curves fit in the OLE and PVA sessions.

Tuning curves measured with the latent-target method were more similar to each other across the two sessions than action-based tuning curves (Fig. 5). Of the 30 neurons used in this experiment, 26 showed less disparity in their latent tuning curves ( $p < 1e-4$ , sign test); across all neurons, the average  $\pm$  SE reduction in disparity with the latent-target method was  $1.02 \pm 0.21$  Hz ( $p < 1e-4$ ,  $t$  test). It is worth stressing that this reduction in fit disparity is not something the algorithm was programmed to do; fits in the two decoding sessions were derived completely independently. Rather, these results suggest that we can recover the underlying tuning curves better if we account for possible changes in re-aiming. As a corollary, differences in re-aiming, if not taken into account, can cause tuning curves to appear to change more than they actually do.

### Distinguishing learning-related changes

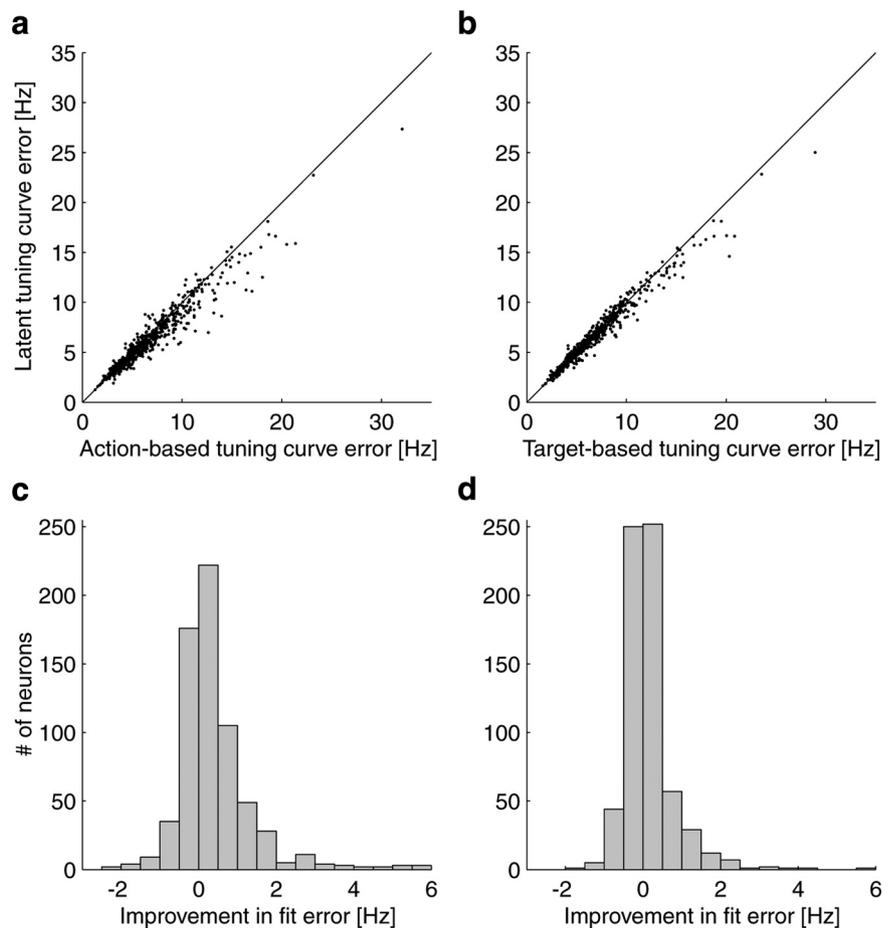
As a final demonstration of the latent-target method, we reanalyzed data from Jarosiewicz et al. (2008). In that work, subjects first completed a “control” session, in which a brain-controlled cursor was moved in a standard 3D center-out task. The subjects then completed a “perturbation” session, in which the decoding parameters of a subset of the neurons (either 25% or 50% of the population) were changed by rotating their decoding PDs ( $\vec{p}_i^D$ ) by  $90^\circ$  about a common axis. By comparing the tuning curves measured between these control and perturbation sessions, it was concluded that, although all of the neurons showed changes in PD in the direction of the global cursor perturbation, the rotated subset of neurons showed greater changes in PD than the nonrotated subset.

To fit the tuning curves in that work, firing rates were regressed against target direction. As we demonstrated in Figure 2, this is only appropriate when there is no re-aiming. To get around this issue, they assumed that the bulk rotation in PD across all cells corresponded to a rotation in intended direction. This assumption breaks down if the amount of re-aiming varies from target to target, as it does for the example in Figure 2. Depending on the particular re-aiming profile, cells can have different amounts of distortion in their measured tuning curves, leading to PDs that appear to shift more or less. In fact, the issue gets more complicated when one tries to imagine ideal re-aiming strategies that a subject might adopt in response to perturbations of the

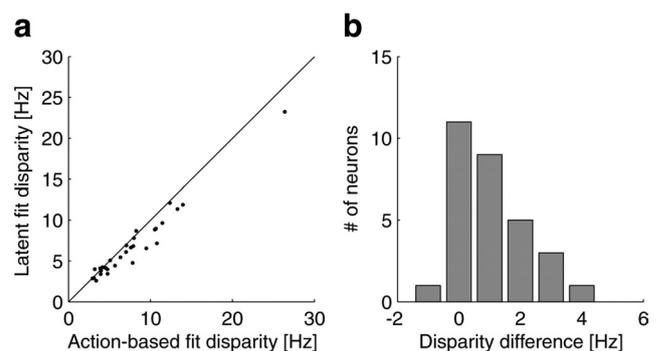
type Jarosiewicz and colleagues applied. Under certain re-aiming strategies, it is theoretically possible to get apparent differential changes in PDs between the rotated and nonrotated populations when in fact the only compensation is re-aiming (data not shown). To ensure that this is not the explanation for the results they reported, it is necessary to use a method that accounts for changes in input and changes in tuning simultaneously. Graphically, the problem is summarized in the neural network schematic of Figure 6. We seek to understand the network response to perturbations of the output weights ( $\tilde{p}_i^D$ ). To do this, we need to differentiate changes in  $\tilde{a}_{input}$ , which would affect all of the neurons, from changes to the individual tuning curves  $\tilde{p}_i$ .

We computed the latent directions and latent tuning curves in the control session as discussed in Materials and Methods, using the action-based tuning curves, based on the cursor movement directions, as the initial fits. In the perturbation session, we initialized the fits and latent directions with the estimated fits and latent directions from the control session. Figure 7*a* shows the shift in the latent directions between the control and perturbation sessions for the 12 experiments in which 50% of the cells were rotated. For ease of viewing, only four of the eight targets are shown. The blue ends of the lines represent the calculated latent directions for each target during the control session, and the red ends of the lines represent the calculated latent directions for the targets during the perturbation session. Although there is a fair amount of individual variation from target to target, the bulk rotation of the latent directions is in the direction that would counter the perturbation (counter to the rotation of the PDs). This is consistent with the idea that these latent directions represent re-aiming points.

How much re-aiming should we expect? The amount of overall cursor perturbation will depend on the specific PD distributions of the perturbed and unperturbed neurons. We can compute the cursor error that will result from the perturbation without any error correction by taking the average firing rates we observe in the control session and decoding them with parameters from the perturbation session. We call this the cursor perturbation direction. We summarize the average cursor perturbation as the average, across targets, of the angular difference between the cursor movement directions during the control session and the cursor perturbation directions in the plane of the perturbation. Figure 7*b* shows the average rotation in the latent directions between the control and perturbation sessions, in the plane of the perturbation, plotted as a function of the average cursor perturbation. The relationship between the two is approximately linear, with the amount of overall latent direction rotation approximately half of what would be required to fully compensate for the cursor perturbation. This is compatible with the results of



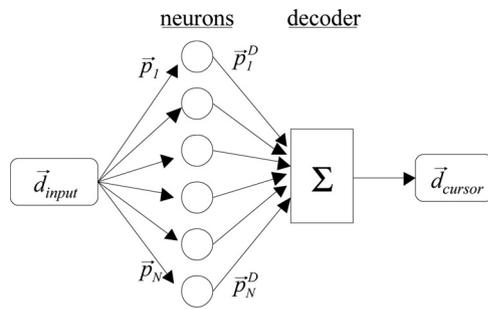
**Figure 4.** The latent-target method improves tuning curve fits. *a*, RMS errors of the latent tuning curves (fit with the latent-target method) plotted as a function of the RMS errors from the action-based tuning curves (fit with the cursor directions). *b*, RMS errors of the latent tuning curves plotted as a function of the RMS errors from the target-based tuning curves (fit with the target directions). *c*, Histogram of the improvements in fit error of the latent tuning curves relative to the action-based tuning curves. *d*, Same, but for the latent tuning curves relative to the target-based tuning curves.



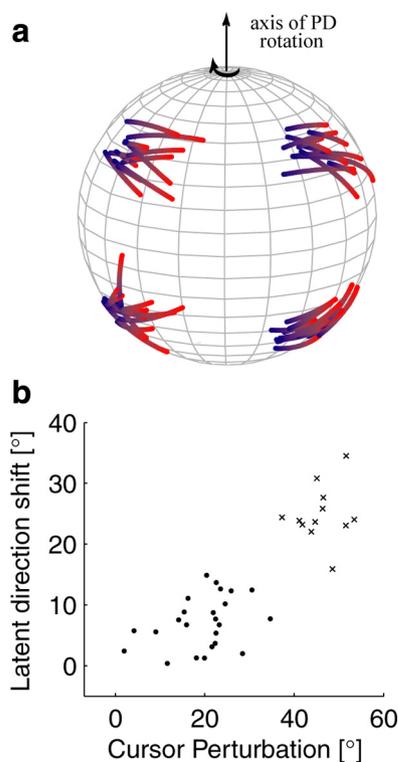
**Figure 5.** Tuning curve similarity under a decoder switching task. Latent tuning curves are more consistent than action-based tuning curves. *a*, Disparity in the latent tuning curves between the PVA and OLE sessions is plotted as a function of the disparity in the action-based tuning curve fits. The latent tuning disparity is almost always less than the action-based tuning disparity. *b*, A histogram of the difference in disparities (latent — action-based).

Jarosiewicz et al. (2008) (adaptation was not complete in their experiments, see their Figs. 1, 5).

Finally, we fit tuning curves with the latent-target method. Figure 8 shows the PD shifts of each neuron in the population for both the 25% and 50% rotation experiments (compare with Jarosiewicz et al., 2008, their Fig. 3). Note that the bulk rotation of



**Figure 6.** Network schematic of the learning experiments performed by Jarosiewicz et al. (2008). Changes to the decoding parameters  $\vec{p}_i^D$  of a subset of neurons cause the cursor movement  $\vec{d}_{\text{cursor}}$  to differ from the subject's desired movement. Cursor error could be corrected either by re-aiming (i.e., by changing  $\vec{d}_{\text{input}}$ ) or by changing the tuning curves of individual neurons,  $\vec{p}_i$ .



**Figure 7.** Latent directions shift to counter the applied perturbation. **a**, Change in the latent direction for each target from the control session (blue ends) to the perturbation session (red ends). Data are from the 50% rotation experiments. Note that the bulk rotation of the latent-direction shift is counter to the direction in which the decoding PDs were rotated. **b**, Average rotation in the latent directions, measured in the plane of the applied perturbation, as a function of the average cursor perturbation. Dots denote experiments in which 25% of the decoding PDs were rotated, and crosses denote experiments in which 50% were rotated.

the PDs is no longer evident in the latent tuning curves, indicating that this compensation has been effectively factored out of the tuning curves. However, there is still a difference in the average PD shift of the rotated and nonrotated subsets of cells (mean  $\pm$  SE; 25% experiments,  $\Delta\text{PD} = 4.4 \pm 1.3^\circ$ ,  $p < 0.001$ ; 50% experiments,  $\Delta\text{PD} = 4.8 \pm 1.6^\circ$ ,  $p = 0.002$ ;  $t$  test), indicating that the retuning compensation is still visible in the latent tuning curves. These results are similar to those of Jarosiewicz and colleagues, who reported  $\Delta\text{PD}$ s of 4.6 and  $5.6^\circ$  for the 25% and 50% experiments, respectively. We can conclude that the small target-specific re-aiming differences did not bias their results.

## Discussion

Any time we learn to use a new tool, perform a new task, or recover from injury, there is uncertainty in how our bodies will respond to a given motor command: our motor intent will often differ from the subsequent action. In fact, this difference between intention and action probably serves as the major drive for motor learning and adaptation. Motor intent can never be directly measured. However, we have presented a method by which unobserved inputs to a population of motor cortical cells can be estimated. We believe that these latent directions are consistent with intention or re-aiming: in a number of BCI experiments, these tuning curves provided better estimates of firing rates than action-based tuning curves (Fig. 4) and were more consistent across different experimental tasks (Fig. 5). Furthermore, in a visuomotor rotation task, the latent directions consistently pointed in the direction that would offset the perturbation (Fig. 7).

Using this technique, it is now possible to look at motor adaptation in detail. We have examined data from visuomotor perturbation experiments and shown that we can dissociate two types of compensation mechanisms in the neural responses: one related to changes in intent (Fig. 7) and one related to changes in neural tuning (Fig. 8). Methods like the one proposed in this paper may eventually help reveal the neurophysiological basis of motor learning and retention.

### Parametric models of motor cortical activity

One of the assumptions behind this work is that the motor input is explicitly represented within a population of motor cortical neurons in terms of the direction in an external coordinate frame where the cursor movement takes place. There is no doubt that motor cortical neurons are somehow related to upcoming movements: not only does motor cortical activity typically lead movement (Crutcher and Alexander, 1990), but also instructed-delay paradigms, which introduce a variable waiting time between when a reach can be planned and when it can be executed, indicate that many motor cortical neurons show activity that relates to movement planning (Riehle and Requin, 1989; Alexander and Crutcher, 1990; Crammond and Kalaska, 2000). However, there is by no means a consensus that the neurons explicitly represent motor intent (Mussa-Ivaldi, 1988; Todorov, 2000; Scott, 2003). It has been argued that, if the brain were operating like an optimal feedback controller (Todorov and Jordan, 2002), an explicit representation of motor commands might not even be necessary (Scott, 2004). Furthermore, there is no a priori reason to assume that a general recurrent neural network should be correlated with the output it produces (Churchland and Shenoy, 2007).

However, simple parametric models of motor cortical tuning curves based on external covariates, such as direction, are widely used because they are easily interpretable and allow one to generate testable hypotheses of how neurons should behave in given situations. The success of closed-loop BCI devices suggests that there is something to be gained by their use: when intention information is extracted under the assumption that it is explicitly represented in recorded neurons and presented back to the subject, the presentation is compatible enough to promote rapid learning (Taylor et al., 2002; Carmena et al., 2003; Jarosiewicz et al., 2008; Ganguly and Carmena, 2009a). One of the reasons cosine-tuning curves have been so successful is that they are fairly general. Velocity, force, and torque are all directional signals, and they are correlated in most simple reaching tasks (Mussa-Ivaldi, 1988; Reina et al., 2001). Thus, tuning curves fit to directions of movement tend to capture the patterns of variation regardless of the underlying encoding scheme. At the very least, the coding

of intended direction seems a good first-order approximation for representing the behavior of these neurons.

### Other sources of error in tuning curve fits

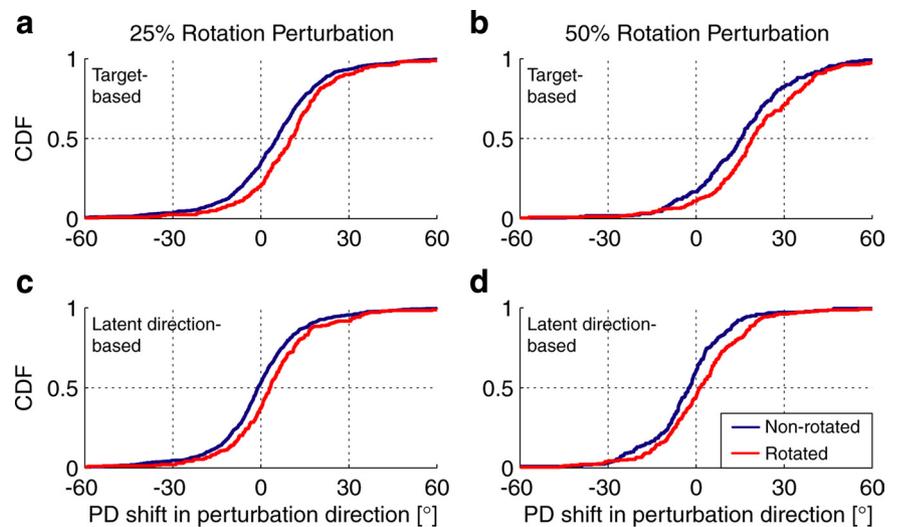
We demonstrated that the algorithm reduces the error in tuning curve fits when it is applied to data from center-out movements performed with a PVA decoder (Fig. 4). How do we know that the observed reduction in error is real? To ensure that we were not overfitting, we used cross-validation to compute the error reduction; the data used to fit the model was separate from the data used to test it. This ensures that the improvement in fit does not come about from an increased ability to fit the noise; rather, the latent-target fits are capturing some systematic aspect of the data better than fits based on actual cursor movements.

Of course, there are a number of other factors that can create systematic differences between cosine-tuning predictions of firing rates and observed firing rates. One example is nonlinearity in the tuning curve itself. Amirikian and Georgopoulos (2000) have observed that most directionally tuned neurons in motor cortex have tuning curves that are narrower than cosine tuning would predict. Furthermore, often neurons will exhibit rectification in their tuning curves, in which the increase in firing rate to movements in the PD of the neuron is larger than the decrease in firing rate to movements in the opposite direction (Taylor et al., 2002). Log-linear tuning curves, of the form

$$\log(\lambda_i) = b_{0i} + m_i \vec{p}_i \cdot \vec{d}_k, \quad (7)$$

can capture both of these effects and have been used to investigate tuning curve fits in motor cortex (Truccolo et al., 2008). As an additional test to see that our results do not just reflect an ability to better capture systematic deviations from linearity, in the supplemental data (available at [www.jneurosci.org](http://www.jneurosci.org) as supplemental material), we repeat our analysis using log-linear tuning curves. All of the results reported here with linear tuning curves also hold true for log-linear tuning curves.

Another factor that can cause systematic errors in tuning curve fits is the presence of unfitted covariates. If the recorded neurons are responsive not only to the direction of movement but also to other variables that correlate with direction, it would be possible to get systematic errors in tuning curve fits. Such a finding would not be unexpected in motor cortex, in which neural activity has been found to correlate with intrinsic variables (Evarts, 1968; Crutcher and Alexander, 1990; Ajemian et al., 2008), extrinsic variables (Crutcher and Alexander, 1990; Ashe and Georgopoulos, 1994; Moran and Schwartz, 1999), as well as mixtures of the two (Kakei et al., 2003) (for review, see Scott, 2003). It is impossible to refute that our neural population may be tuning for things other than direction; we cannot know precisely how the subject volitionally controls the cursor. It is also possible that the some of the reduction in tuning curve error we see when we apply our algorithm does not represent re-aiming but rather some kind interaction between the unfitted covariates



**Figure 8.** PD changes in the 3D perturbation task between the perturbation and control sessions. *a, b*, Empirical cumulative distribution functions (CDF) of the shift in the PD measured during the perturbation session relative to the control session, for all units in experiments in which 25% (*a*) or 50% (*b*) of the units were rotated. Shifts are measured in the perturbation direction, for tuning curves measured assuming the target directions are the aiming points (for details, see Jarosiewicz et al., 2008). *c, d*, Same as *a* and *b*, for tuning curves measured with the latent-target method. Note that the bulk shift in PD observed in the target-based fits of *a* and *b* goes away when the latent-target method is used.

and the movement direction that shows some consistency across the population. However, in the visuomotor rotation perturbations applied by Jarosiewicz et al. (2008), the re-aiming points we computed were all in the direction that would offset the perturbation (Fig. 7); it is unlikely that this effect is entirely attributable to the action of unfitted covariates. One of the major open questions in motor neurophysiology is the nature of the volitionally controllable inputs to motor cortex. Our method represents a first step toward elucidating the latent factors that subservise motor control.

### Steady-state versus dynamic inference

We have tested this algorithm mainly under steady-state conditions, when the intended movements may be considered fairly constant from trial to trial. A major effort remains to extend these ideas to situations in which intention is dynamically evolving, such as when a subject is initially learning a task or new condition. The BCI calibration session is an especially pertinent example, as inaccurate assumptions about intent during this session could limit device performance. Although on one hand it seems clear that longer calibration sessions should be better because they allow more time to collect data on tuning functions, at the same time a long calibration allows the subject the chance to adapt and change his strategy. How the exact form of the calibration procedure impacts BCI performance is not known; it is clear, however, that the correct procedure will not be found until the process that transforms errors into movement command updates is taken into account.

### Tuning curve stability and comparison

To build better prosthetic devices and gain insight into the cognitive processes that occur when using a BCI, a number of researchers have compared various features of tuning curves measured during the operation of a BCI device with the same features measured during a reaching task (Taylor et al., 2002; Carmena et al., 2003; Zacksenhouse et al., 2007; Ganguly and Carmena, 2009a,b; Wahnoun et al., 2009; Whitford et al., 2009).

Any BCI must rely on an inference of intention. These are typically based on some initial estimate of target-based tuning functions of the cells. Although procedures for establishing these initial tuning curve estimates vary widely, they are all prone to estimation error through limited data, incorrect encoding model assumptions, recording noise, etc. These errors will cause the intent to be misestimated, which will likely lead to a change in the control strategy. To gain more insight into the factors underlying firing rate changes between the two paradigms, one should use a method that can differentiate changes in intention (re-aiming) from changes in intention-based tuning (retuning).

## Conclusions

The methods we have used to describe intention representation emphasize the analysis of neuronal population activity. Although individual neurons are well modulated by direction, their tuning functions are broad, noisy, and ambiguous; single neurons provide only a small amount of directional information. However, latent factors that have a small but consistent effect on individual neurons can be recognized by looking at the activity of many simultaneously recorded neurons.

It remains to be seen whether learning in a BCI context is the same as motor learning in more natural movement contexts. One of the reasons that BCI learning is such an interesting area to study is that it builds a defined link between a change in neural firing rate and a definite behavior (cursor movement), so that changes in neural tuning can be understood at the behavioral level. The algorithm we present in this paper is one of the tools that we can use to start understanding the links between learning, neural tuning, and behavior.

## References

- Ajemian R, Green A, Bullock D, Sergio L, Kalaska J, Grossberg S (2008) Assessing the function of motor cortex: single-neuron models of how neural response is modulated by limb biomechanics. *Neuron* 58:414–428.
- Alexander GE, Crutcher MD (1990) Preparation for movement: neural representations of intended direction in three motor areas of the monkey. *J Neurophysiol* 64:133–150.
- Amirikian B, Georgopoulos AP (2000) Directional tuning profiles of motor cortical cells. *Neurosci Res* [Erratum (2000) 37:83; Georgopoulos AP (corrected to Georgopoulos AP)] 36:73–79.
- Arce F, Novick I, Mandelblat-Cerf Y, Israel Z, Ghez C, Vaadia E (2010) Combined adaptiveness of specific motor cortical ensembles underlies learning. *J Neurosci* 30:5415–5425.
- Ashe J, Georgopoulos AP (1994) Movement parameters and neural activity in motor cortex and area 5. *Cereb Cortex* 4:590–600.
- Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, Dimitrov DF, Patil PG, Henriquez CS, Nicolelis MA (2003) Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol* 1:E42.
- Chapin JK, Moxon KA, Markowitz RS, Nicolelis MA (1999) Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nat Neurosci* 2:664–670.
- Chase SM, Schwartz AB, Kass RE (2009) Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Netw* 22:1203–1213.
- Chestek CA, Batista AP, Santhanam G, Yu BM, Afshar A, Cunningham JP, Gilja V, Ryu SI, Churchland MM, Shenoy KV (2007) Single-neuron stability during repeated reaching in macaque premotor cortex. *J Neurosci* 27:10742–10750.
- Churchland MM, Shenoy KV (2007) Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *J Neurophysiol* 97:4235–4257.
- Churchland MM, Santhanam G, Shenoy KV (2006) Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach. *J Neurophysiol* 96:3130–3146.
- Cisek P, Crammond DJ, Kalaska JF (2003) Neural activity in primary motor and dorsal premotor cortex in reaching tasks with the contralateral versus ipsilateral arm. *J Neurophysiol* 89:922–942.
- Crammond DJ, Kalaska JF (2000) Prior information in motor and premotor cortex: activity during the delay period and effect on pre-movement activity. *J Neurophysiol* 84:986–1005.
- Crutcher MD, Alexander GE (1990) Movement-related neuronal activity selectively coding either direction or muscle pattern in three motor areas of the monkey. *J Neurophysiol* 64:151–163.
- Evarts EV (1968) Relation of pyramidal tract activity to force exerted during voluntary movement. *J Neurophysiol* 31:14–27.
- Ganguly K, Carmena JM (2009a) Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol* 7:e1000153.
- Ganguly K, Carmena JM (2009b) Neural mechanisms underlying the changes in directional tuning in brain-machine interfaces. *Soc Neurosci Abstr* 35:14.12.
- Georgopoulos AP, Kalaska JF, Caminiti R, Massey JT (1982) On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J Neurosci* 2:1527–1537.
- Georgopoulos AP, Schwartz AB, Kettner RE (1986) Neuronal population coding of movement direction. *Science* 233:1416–1419.
- Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, Caplan AH, Branner A, Chen D, Penn RD, Donoghue JP (2006) Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* 442:164–171.
- Jarosiewicz B, Chase SM, Fraser GW, Velliste M, Kass RE, Schwartz AB (2008) Functional network reorganization during learning in a brain-computer interface paradigm. *Proc Natl Acad Sci U S A* 105:19486–19491.
- Takei S, Hoffman DS, Strick PL (1999) Muscle and movement representations in the primary motor cortex. *Science* 285:2136–2139.
- Takei S, Hoffman DS, Strick PL (2003) Sensorimotor transformations in cortical motor areas. *Neurosci Res* 46:1–10.
- Kass RE, Ventura V, Brown EN (2005) Statistical issues in the analysis of neuronal data. *J Neurophysiol* 94:8–25.
- Koyama S, Chase SM, Whitford AS, Velliste M, Schwartz AB, Kass RE (2009) Comparison of decoding algorithms in open-loop and closed-loop performance. *J Comput Neurosci*. Advance online publication. Retrieved September 11, 2010. doi:10.1007/s10827-009-0196-9.
- Krakauer JW, Pine ZM, Ghilardi MF, Ghez C (2000) Learning of visuomotor transformations for vectorial planning of reaching trajectories. *J Neurosci* 20:8916–8924.
- Li CS, Padoa-Schioppa C, Bizzi E (2001) Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field. *Neuron* 30:593–607.
- Moran DW, Schwartz AB (1999) Motor cortical representation of speed and direction during reaching. *J Neurophysiol* 82:2676–2692.
- Mulliken GH, Musallam S, Andersen RA (2008) Decoding trajectories from posterior parietal cortex ensembles. *J Neurosci* 28:12913–12926.
- Mussa-Ivaldi FA (1988) Do neurons in the motor cortex encode movement direction? An alternative hypothesis. *Neurosci Lett* 91:106–111.
- Paz R, Vaadia E (2004) Specificity of sensorimotor learning and the neural code: neuronal representations in the primary motor cortex. *J Physiol Paris* 98:331–348.
- Reina GA, Moran DW, Schwartz AB (2001) On the relationship between joint angular velocity and motor cortical discharge during reaching. *J Neurophysiol* 85:2576–2589.
- Riehle A, Requin J (1989) Monkey primary motor and premotor cortex: single-cell activity related to prior information about direction and extent of an intended movement. *J Neurophysiol* 61:534–549.
- Rokni U, Richardson AG, Bizzi E, Seung HS (2007) Motor learning with unstable neural representations. *Neuron* 54:653–666.
- Salinas E, Abbott LF (1994) Vector reconstruction from firing rates. *J Comput Neurosci* 1:89–107.
- Schwartz AB, Kettner RE, Georgopoulos AP (1988) Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement. *J Neurosci* 8:2913–2927.
- Scott SH (2003) The role of primary motor cortex in goal-directed movements: insights from neurophysiological studies on non-human primates. *Curr Opin Neurobiol* 13:671–677.

- Scott SH (2004) Optimal feedback control and the neural basis of volitional motor control. *Nat Rev Neurosci* 5:532–546.
- Taylor DM, Tillery SI, Schwartz AB (2002) Direct cortical control of 3D neuroprosthetic devices. *Science* 296:1829–1832.
- Todorov E (2000) Direct cortical control of muscle activation in voluntary arm movements: a model. *Nat Neurosci* 3:391–398.
- Todorov E, Jordan MI (2002) Optimal feedback control as a theory of motor coordination. *Nat Neurosci* 5:1226–1235.
- Truccolo W, Friehs GM, Donoghue JP, Hochberg LR (2008) Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia. *J Neurosci* 28:1163–1178.
- Velliste M, Perel S, Spalding MC, Whitford AS, Schwartz AB (2008) Cortical control of a prosthetic arm for self-feeding. *Nature* 453:1098–1101.
- Wahnoun R, Helms Tillery SI, He J (2009) Task specific encoding of cortical motor units; targeted adaptation to the neuroprosthesis? *Soc Neurosci Abstr* 35:181.24.
- Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, Nicolelis MA (2000) Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* 408:361–365.
- Whitford AS, Chase SM, Schwartz AB (2009) Task-specific tuning properties of motor cortical units used to control a brain-computer interface. *Soc Neurosci Abstr* 35:770.5.
- Wise SP, Moody SL, Blomstrom KJ, Mitz AR (1998) Changes in motor cortical activity during visuomotor adaptation. *Exp Brain Res* 121:285–299.
- Zacksenhouse M, Lebedev MA, Carmena JM, O'Doherty JE, Henriquez C, Nicolelis MA (2007) Cortical modulations increase in early sessions with brain-machine interface. *PLoS One* 2:e619.