INVITED PAPER

Statistical Signal Processing and the Motor Cortex

Spike signals from neurons in the human brain may be decoded to control robotic hands, arms and other prosthetic devices.

By A. E. Brockwell, *Member IEEE*, Robert E. Kass, and A. B. Schwartz

ABSTRACT | Over the past few decades, developments in technology have significantly improved the ability to measure activity in the brain. This has spurred a great deal of research into brain function and its relation to external stimuli, and has important implications in medicine and other fields. As a result of improved understanding of brain function, it is now possible to build devices that provide direct interfaces between the brain and the external world. We describe some of the current understanding of function of the motor cortex region. We then discuss a typical likelihood-based state-space model and filtering based approach to address the problems associated with building a motor cortical-controlled cursor or robotic prosthetic device. As a variation on previous work using this approach, we introduce the idea of using Markov chain Monte Carlo methods for parameter estimation in this context. By doing this instead of performing maximum likelihood estimation, it is possible to expand the range of possible models that can be explored, at a cost in terms of computational load. We demonstrate results obtained applying this methodology to experimental data gathered from a monkey.

KEYWORDS | Brain–machine interface; cortex; decoding; Markov chain; Monte Carlo; neural; nonlinear filtering; sequential; state-space model

I. INTRODUCTION

The human brain, made up of something on the order of 100 billion neurons, is one of the most complex systems ever to be studied by researchers. It takes in enormous amounts of sensory information, processing it in ways that are only partially understood, and over time, alters its own internal state and produces complex motor control and other signals that alter the state of the body. Recent developments in technology have made it possible to collect information about the function of the brain under a wide range of different experimental conditions. Not only is this of great interest from the scientific point of view, but it also has the exciting potential to lead to the development of a range of new devices often referred to as BMIs ("brain–machine interfaces"), which would allow direct mental control of external devices. Work on such devices has proliferated in recent years, and mentally controlled computer cursors and other interfaces are being developed by a number of research groups (see, e.g. [1]–[8] and references therein).

Methods for recording neural activity include functional magnetic resonance imaging, use of surface electrodes placed on the scalp, use of subdural electrodes, as well as microwire electrodes and their silicon-machined variants [9], [10]. Of these, the microwire (or silicon-machined) electrodes, give the finest detail of measurement of neural activity, but are the most invasive. They are surgically implanted, and work by detecting "action potentials" or their extra-cellular signatures, known as "spikes," in individual neurons. These spikes are rapid changes in the voltage difference between the inside and outside of the cell, and are believed to be the primary mechanism by which neurons transmit information. Spikes last on the order of a millisecond, propagating through the neuron. Electrical signals are transferred between neurons through the action of neurotransmitters at synaptic junctions. By measuring spiking activity of neurons concurrently with associated external variables, researchers have been able to better explain the function of neurons in particular regions of the brain.

The purpose of this paper is to give a brief introduction to recent research into the function of one specific area of

Manuscript received November 15, 2006; revised February 2, 2007. This work was supported in part by the National Institutes of Health under Grant R01 MH064537-04AZ and Grant R01 EB005847-01.

A. E. Brockwell and **R. E. Kass** are with the Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: abrock@stat.cmu.edu; kass@stat.cmu.edu).

A. B. Schwartz is with the Department of Neurobiology, University of Pittsburgh, Pittsburgh, PA 15261 USA (e-mail: abs21@pitt.edu).

Digital Object Identifier: 10.1109/JPROC.2007.894703

the brain, the motor cortex,[1] and to illustrate how statistical modeling and signal processing methods can be used to extract ("decode") information from measurements of activity in individual neurons. In Section II, we describe experimental work that has been performed, and the various insights into motor cortex function that have arisen as a consequence of this work. In Section III, we illustrate a typical state-of-the-art likelihood-based framework for performing analysis of spike data collected from the motor cortex, and we describe the associated nonlinear filtering problem that can be used to "decode" signals for the sake of developing prosthetic devices. We use Markov chain Monte Carlo (MCMC) methods to fit models, and we use sequential Monte Carlo methods to perform decoding, although a wide range of alternative methods can be found in the literature. (MCMC methods, although slow, are relatively easy to implement and can be used with a wide range of possible models.) We compare results with those obtained using other methods. Finally, in Section IV, we discuss some of the additional issues associated with development of a properly functioning cortical prosthetic device.

II. FUNCTION AND ADAPTIVITY IN MOTOR CORTEX

A. Modularity and Distributed Processing

Early neurophysiologists were influenced by the Cartesian, mechanical (hydraulic) theory of the nervous system. As the importance of the brain became appreciated, a debate began as to whether function was a product of whole-brain activity or was, instead, localized in small regions of the brain (see, e.g., discussion in [11]). In the latter half of the 19th century localization became dominant, due especially to Broca's observation that a lesion in what is now called Broca's area caused serious damage to language production [12] and Fritsch and Hitzig's use of electrical stimulation in motor cortex to invoke muscle activity [13], which was replicated and improved by [14]. Much of the modern view is based on similar anatomical observations, often involving structurally defined components, the assumption being that distinct structures should have distinct functions. With the advent of computers, and the metaphor of brain as computer, it was convenient to describe brain function in terms of information flowing sequentially through discrete modules. However, new developments in both neurophysiology and psychology begun during the 1980s, have led many brain scientists to a more distributed concept of brain function.

Part of the appeal of localization comes from a natural inclination toward reduction. Just as electronic circuits are made from component parts, each having a specific purpose, there may be brain circuits with well-identified functional components. The applicability of such an analogy, however, is questionable. For one thing, while individual neuronal activity has been described with great success in terms of equivalent electrical circuits, at the level of neuronal networks it is unclear what "components" might be used, or what their properties might be: details of the vastly complicated and highly redundant interconnections remain largely unidentified. Sensory inputs traverse a variety of pathways from the periphery to arrive in an asynchronous, yet parallel fashion in multiple cortical areas, and they vary continuously in time. Cortical areas are generally connected reciprocally, so that if one area transmits information to another area, it also receives information from that area. It is not surprising then, that experimental investigations frequently report similar neural activity patterns from widely distributed regions, and reductions conceived as simple directional connections among local circuits are likely to miss important features of brain function.

These problems with modularity may be nicely illustrated with a historical example. In the late 1950s and early 1960s Vernon Mountcastle carried out a number of experiments in which he applied different degrees of mechanical pressure on the cat's foot while recording activity of a single neuron at different points along one of the major pathways to the sensory cortex [15]. He found a simple relation between applied pressure and neuronal firing rate throughout the neural axis. But more importantly, he showed that there appeared to be a straightforward organization to these responses in the sensory cortex. The cortex (neocortex, see footnote) has six layers of cells and he found that within a "column" of cortex, many of the cells were activated by the same type of peripheral receptor in the same part of the body. Within a column, cells are heavily interconnected. The radius of each column is 300–500 μm and they are formed early in the developmental process. Much of the input to the column arrives in layer IV, while output to cortical regions leaves via axons of layer II–III neurons and to structures outside of cortex (the thalamus, colliculus, brainstem, and spinalcord) via layer V and VI cells. When Mountcastle made vertical electrode penetrations through the cortex to record single cell activity, neurons tended to have the same "properties of place and modality," but when making horizontal penetrations the neurons were found to change properties abruptly at spatial intervals of 300–500 μm. Later studies by David Hubel and Torsten Wiesel in the visual cortex extended these columnar findings. In the primary visual cortex, Hubel and Wiesel found columns based on ocular dominance (cells that respond to visual stimulation to right or left eye) interleaved with orientation selectivity (differential response to visualized lines projected on a screen at varying angles).

This mapping concept motivated studies in the motor cortex. In the late 1960s and early 1970s, Hiroshi Asanuma

[1]The *cortex* is the "gray matter" of the brain, composed of neurons; it often refers to what is more properly called the *neocortex*, which is found in mammals, consisting of multiple layers of neurons.

developed a technique called intracortical microstimulation (ICMS, see [16], [17]). By applying repeated pulses of small current through a microelectrode tip placed in the motor cortex, he was able to activate individual muscles in different parts of the body. He found that this muscle organization was discrete: the same muscle would contract as the electrode moved in the same vertical penetration, but that different muscles would be successively activated when moving the electrode tangentially. This seemed to correspond to the column concepts coming from the work in sensory and visual systems and led him to propose the term "cortical efferent zone" for the type of muscle column he found. However, soon after this initial work, it was found that high frequency ICMS acutally activated axons passing by the electrode tip, so that most of the neurons affected with this technique were mediated indirectly [18]. The indirect excitation likely activated a large network of active elements, artificially focusing the stimulation onto a single muscle [19]. Recent studies employing anatomical tracers show that motor cortical neurons projecting to a single muscle arise from a wide region and overlap with neurons projecting to other muscles [20]. In other words, despite its initial promise, the idea that motor cortex is made up of single-purpose, localized circuits is unlikely to be a powerful tool for understanding motor function.

B. Directional Tuning

While it appears unlikely that motor cortex may be described as a collection of specialized local circuits, the idea that individual neurons control the movement of individual muscles or groups of muscles was, for many years, a dominant conception [21], [22]. A quite different notion was introduced as a result of a series of experiments conducted by Apostolos Geogopoulos [23], [24]. Georgopoulos designed an experiment in which monkeys started from a center start position and reached radially to targets in eight equally spaced directions. Activity was recorded from single neurons in motor cortex during the task. Many of these neurons had activity patterns that were clearly related to movement direction; the firing rates changed for each direction and could be summarized, approximately, as the cosine of the angle of movement after an appropriate phase adjustment, times some constant. The results for one neuron are shown in Fig. 1. The peak of the fitted cosine function corresponds to the direction with the highest discharge rate, called the "preferred direction." During reaching, many muscles are active simultaneously, implying that a direct relation between single cortical neurons and individual muscles would have a complex relation to movement direction. Yet, what was found was a simple, broad tuning of cells according to movement direction. This suggests a *functionally* oriented view of motor cortical activity: many motor cortical neurons fire in relation to direction of movement regardless of the particular muscles they drive;

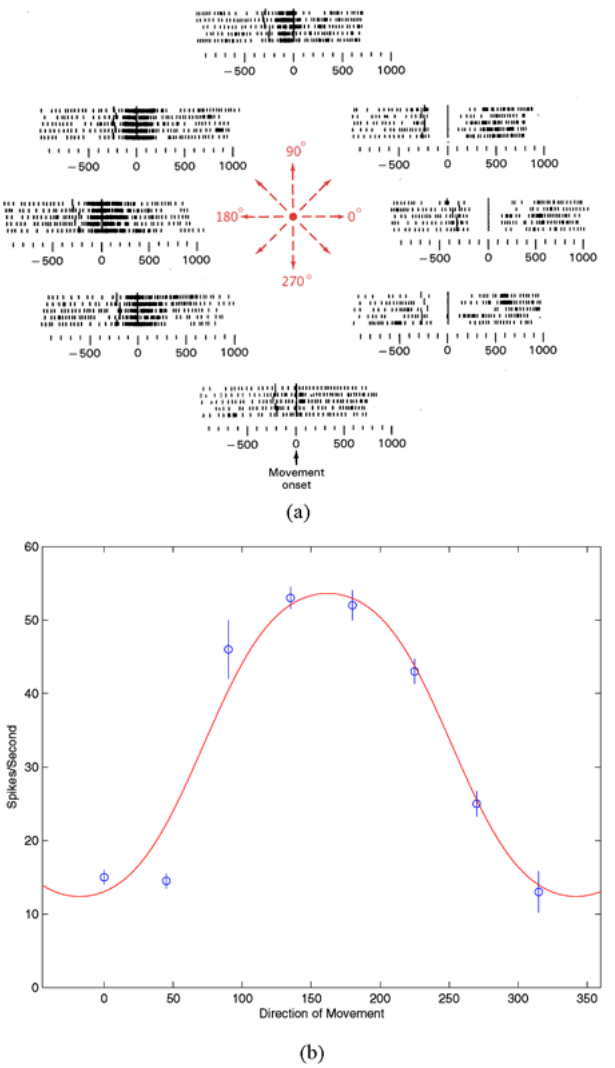

Fig. 1. ***Top: Spike trains (neuronal firing event times) from a single neuron in motor cortex recorded while a monkey performed a reaching task in each of eight directions, indicated here by counter-clockwise angle from right. Each of the eight "raster" plots displays five repetitions of the reach. Time 0 indicates initiation of movement. From the raster plots it is clear that this neuron fired much more intensely when the movement was roughly in the leftward direction than when it was in the rightward direction. Bottom: The mean firing rates (with standard errors) are plotted across the eight directions. The smooth curve is a cosine function. (Plots reconstructed from those of [23]).***

the downstream circuits may be more complicated than conceived under the localization paradigm, and one possibility is that they could translate kinematically coded signals into required particular combinations of muscle activation signals.

A second point emerged from the initial work of Georgopoulos and colleagues. At first glance, the approximate cosine tuning of each neuron might be considered to pose a conceptual problem: How can highly precise

movement direction be obtained from such broad tuning curves? The answer, of course, is that movement is a result of activity across an entire *population* of neurons. Georgopoulos and colleagues showed that by "decoding" the population activity across several hundred neurons, the movement itself may be predicted quite accurately.

Neurons with broad tuning related to movement direction have been found throughout the nervous system, from primary afferents to muscles themselves. This provides a powerful descriptive mechanism for extracting signals with which the relationship of behavior to cortical function may be examined. For example, [25] recorded populations of neurons from two different parts of motor cortex, primary motor cortex (M1) and ventral premotor cortex (PMv), as monkeys drew ellipses under two different experimental settings. In both settings, the monkeys received visual feedback of their hand position indirectly through a virtual reality display, but in the first, the 3-D coordinates of actual hand position were mapped directly to the virtual display, while in the second, coordinates were transformed linearly so as to alter the eccentricity of the ellipse. M1 is usually considered to be the origination of cortical output to muscle activation, while PMv projects to M1 and is associated with movement planning. In the recordings of [25], both structures contained populations of broadly tuned cells from which accurate trajectories could be decoded. However, M1 cell activity appeared to more accurately correspond to the actual trajectory of the arm, while PMv neural trajectories appeared to more accurately correspond to the perceived shape of the drawn objects. This is one example of the way functionality can be closely related, yet different across structures. The initial description of discrete cortical areas in terms of differing purposes remains useful, but it clearly must be supplemented with a distributed view of functionality. Even the idea that single motor cortical neurons encode single characteristics of movement is limiting: the activity of an individual neuron is likely to depend on many movement parameters, such as position, speed, direction, load, etc., including factors that are not routinely controlled in laboratory experiments. Some of these factors might have only a small effect on the firing of an individual neuron yet might be readily extracted from a neural population. In other words, the encoding space is likely to be highly multidimensional, important population effects may be subtle and, therefore, new signal processing approaches to the encoding/decoding problem are needed if we are to advance our understanding of motor cortical function.

C. Plasticity and Adaptivity

There is considerable evidence that motor cortical neuronal activity is not hard-wired, once and for all, after development, but instead is subject to rewiring as a result of either injury or purposeful use. Some of the work has focused on anatomical rewiring. For example, [26] transected the facial motor nerve that supplies the rat whisker musculature. This led to functional loss of the M1 whisker area, which was supplanted by representations of the adjacent forelimb or eye/eyelid regions. Working with monkeys, [27] showed that occlusion of an artery in the M1 hand area produced an inability to retrieve food pellets; without practice of the affected hand, the elbow and shoulder areas expanded into the remaining undamaged hand area (as a compensatory mechanism); with practice, the undamaged hand area expanded into the elbow and shoulder areas and behavior was recovered within 3–4 weeks. Similarly, [28] showed, using fMRI in humans, that practiced finger movements could increase the apparently relevant area of M1.

A different line of work has focused on functional rewiring. In an experiment reported in [29], two monkeys had to adapt their reaching movements to external forces; these authors found a sizable population of cells that changed their tuning properties during exposure to the force field. [30] examined neuronal activity while a monkey adapted to novel visuomotor transforms. Many neurons showed significant changes in their task-related activity, including changes in the magnitude of activity modulation during adaptation, and changes in preferred directions during rotation tasks. Using a virtual reality workspace, [31] trained monkeys to move a cursor under both hand control, via an optical tracker, and brain control, via M1 signal processing based on a version of the population vector algorithm (while the arms were restrained). This algorithm simply estimates intended cursor motion by computing, at each point in time, a linear combination of preferred direction vectors associated with each of a number of neurons, where the weights in the linear combination are proportional to time-localized estimate of the current firing rates of the respective neurons. Compared with hand control, under brain control neuronal preferred directions shifted substantially. [32] also performed experiments where human subjects moved a robotic arm through space, while receiving visual feedback on a projection screen. By performing visual feedback rotations (but not physical rotations) of 8-target reaching movements and using fMRI to measure brain-activation, they found changes in tuning caused by learning of new visuomotor transformations during movement preparation.

Potential mechanisms for such adaptive reconfiguration have been described [33], [34]. Taken together, these findings suggest a dynamic view of functionality: integrative cortical networks are able to adjust connections on relatively short time scales, allowing the role of each neuron to evolve according to behavioral needs.

III. ANALYSIS OF MOTOR CORTEX DATA

So far, we have discussed high-level features of the motor cortex and motivated the need for new signal processing

techniques. We now describe the use of statistical signal processing in the context of analysis of spike event times for individual neurons in the motor cortex. In particular, we illustrate with laboratory data how a likelihood-based technique can be used.

We discuss the likelihood-based approach, since it is optimal in certain senses when performed appropriately. In fact, to be precise, we will be interested in two different problems: model-fitting (or "encoding"), and decoding (or estimation of unobserved quantities). A full likelihood-based approach involves likelihood-based model-fitting as well as decoding that is matched to the selected model and involves analysis of certain conditional distributions to obtain optimal results. Likelihood-based methods have been in the literature for some time. Examples include, for instance, the early work of [35]–[40] as well as more recent work of [41]–[49]. A general review is also given by [50]. Generically, the approach consists of:

- fitting appropriate probabilistic models describing the behavior of covariates of interest as well as neuron spiking behavior, and relating these to each other as well as to any relevant additional measured covariates;
- performing goodness-of-fit testing of these models, potentially reformulating and refitting models if the tests fail;
- carrying out (likelihood-based) decoding based on the models.

The first step is simply to provide (good) statistical models that explain the relationship between these spike times and hand motion. The second step is important because likelihood-based decoding methods are only guaranteed to perform optimally when they are based on a "good" model. The third step, discussed below in Section III-D, is to use these models to perform filtering, thereby obtaining real-time estimates of intended hand motion, given the spike data. Such estimates could be used, for instance, to control a cursor or robotic prosthetic device. A key appeal of the approach is the following property. If the fitted models do indeed provide probabilistically accurate descriptions of the relationship between explanatory variables (covariates) and spiking behavior, and the decoding scheme is matched to the model in the sense that it determines conditional expected values of desired quantities given available information under the fitted model, then the decoded values are guaranteed to minimize the mean-squared error. Of course, if the model does not incorporate important explanatory variables or if it is simply probabilistically inaccurate, then this guarantee does not hold.

Our purpose here is primarily to illustrate how such an approach works, although in the process we do demonstrate the feasibility of the use of Markov chain Monte Carlo (MCMC) schemes to estimate parameters. This is useful because with these schemes, one can perform likelihood-based parameter estimation with a very general class of models, and in theory, this expands the range of models that can be considered. MCMC schemes are typically relatively simple to implement. However, they are known to be relatively inefficient in terms of computational requirements; at the moment this may limit their usefulness in devices that may require rapid updates of model parameters. On the other hand, given continuing improvement in number-crunching power of desktop computers, this consideration is certain to become less and less important in the future.

We adopt the approach of describing spiking behavior as it relates directly to kinematic properties of the hand. However, it should be noted that there is a substantial body of work (see, e.g., [51], [52] and references therein) that develops models relating spiking behavior to force, individual muscle activation levels, and other nonkinematic quantities. The debate in the literature continues over which of these types of models provides the most accurate description of motor cortical behavior; to date there is evidence supporting the validity of both viewpoints.

To illustrate the methodology, we consider data collected from a monkey trained to perform reaching motions starting with its hand positioned at a central position within a virtual cube. Its actual hand position was tracked using an optical tracking device and mapped to a cursor in a 3-D virtual display unit that the monkey was looking into. A succession of 57 "target" points was chosen, with each point chosen at random from one of eight corners of the cube. (In the experiment, there were between 5 and 11 replications of each of the eight targets.) On presentation of a target, the monkey was rewarded for moving the cursor (by moving its hand) to the target location. Once it reached the target (or failed to complete the reaching motion correctly), the next of the targets was presented, and so on. Over the sequence of reaching motions, spiking activity was recorded from 70 neurons simultaneously. Times of spiking events for neurons were recorded to the nearest millisecond, while hand kinematics were sampled every 10 ms. Thus we have spike measurements for all 70 neurons, as well as the measured location of the hand. Hand position and spiking activity are shown, for a portion of our data, in Fig. 2.

The task is carried out in 3-D virtual space, and the directional tuning described earlier in Section II-B still applies. (The generalization to three dimensions is explained in the following subsection.) To illustrate typical behavior, Fig. 3 shows spike trains recorded as the monkey reached into each of the eight corners of a virtual cube, for two particular neurons. One of these (top) is fairly active, but does not exhibit a strong preference for directions, while the other one (bottom) is not as active overall, but spike events clearly occur with a noticeably higher frequency when reaching is toward the ("preferred") front right corner.

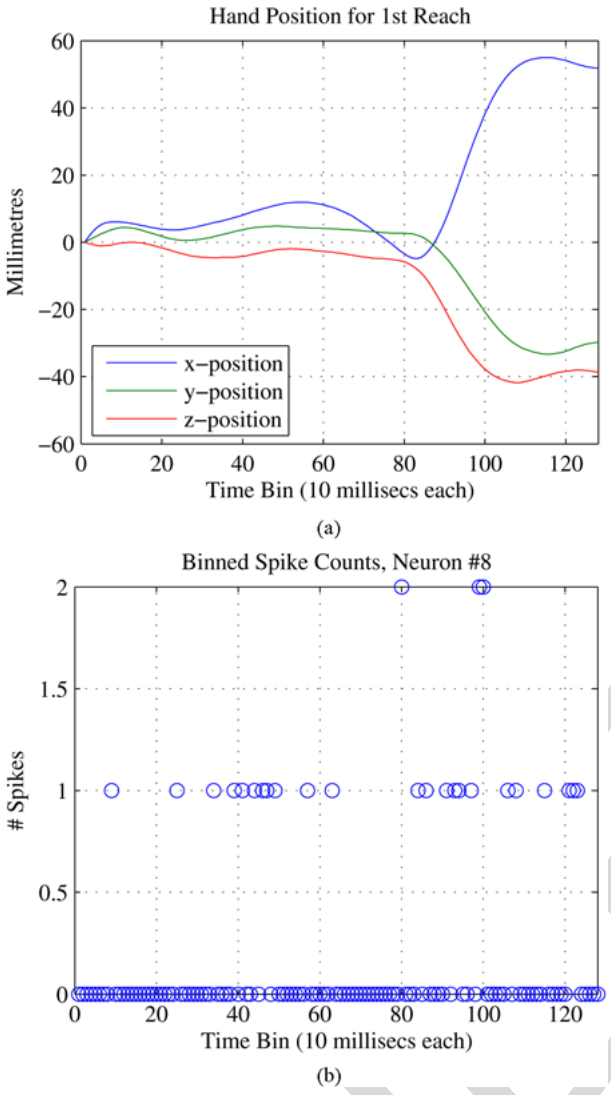

Fig. 2. *Top: trajectory (x, y, and z coordinates) of the monkey's hand during the first of 50 reaching motions, Bottom: counts of number of spikes in one neuron during motion. Counts here ranged from 0 to 2 in each 10 ms time bin.*

In the following four subsections, we lay out a typical state-space model and filtering approach to analysis of these data, stating algorithms and methods in a general form. In the subsequent subsection, we present results of our data analysis using the method, and give specific parameters and other details of our particular implementation of the algorithms.

A. State-Space Models for Kinematics and Spiking

We will begin by describing several formal state-space models that can be built to describe the intrinsic behavior of kinematics and to explain the connection between this behavior and spiking activity. Such models (including some which are posed in continuous-time instead of in discrete-time) have been used by a number of groups—see, for example [40], [48], [53]–[56].

To illustrate the approach, we introduce the following notation. Let $N_t^{(j)}$ denote the number of spiking events occurring for the j-th neuron during the t-th 10 ms time bin. Assume that we measure activity of P neurons. We write N_t to denote the vector $(N_t^{(1)}, \ldots, N_t^{(P)})^T$ consisting of spike counts for all P neurons. We also define X_t and V_t to be 3-D column vectors, denoting, respectively, hand position (mm) at the beginning of the t-th time bin, and average hand velocity (mm/sec) during the time bin. For convenience, we also define $K_t = (X_t^T, V_t^T)^T$ to be the 6-D column vector containing both position and velocity. We will also define, for the j-th neuron, a lag parameter lag_j, that represents the amount of time from kinematic activation to neural activity (so that negative lags correspond to neural activity preceding kinematic activation), measured in number of time bins.

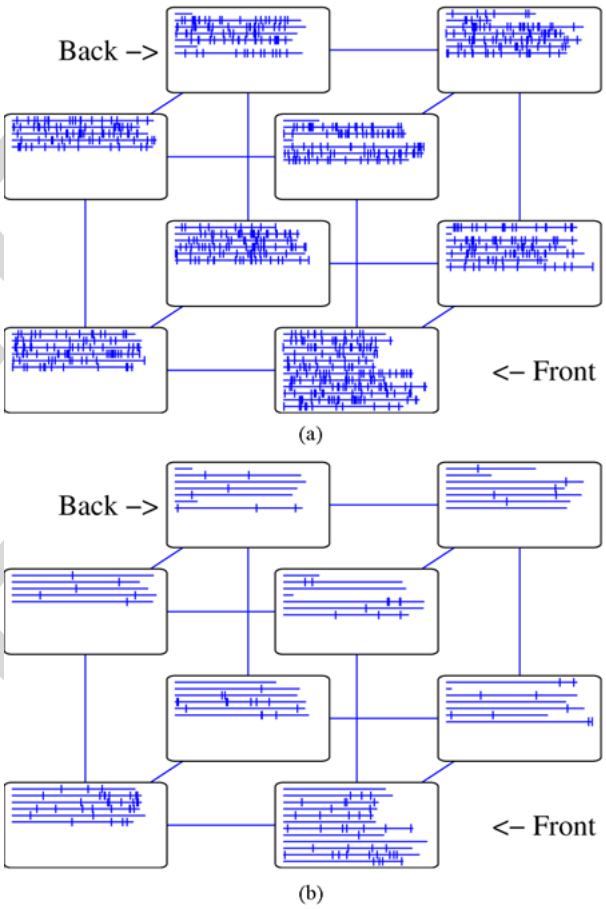

Fig. 3. *Spike trains associated with two neurons while 57 reaching motions were carried out to the eight corners of a cube. Top: a neuron has a high overall firing rate but is not very obviously tuned, Bottom: the neuron has a lower overall firing rate but is more strongly tuned, showing a preference (indicated by a higher firing rate) for motion to the front right.*

Early models used to capture directional preferences of neurons, for example, derived from results in [24], [57], although not always stated explicitly, were typically of the form

$$\sqrt{N_t^{(j)} + 3/8} = \beta_0^{(j)} + \beta^{(j)} \cdot V_{t-\text{lag}_j} + \epsilon_t^{(j)} \qquad (1)$$

where $\beta^{(j)} = (\beta_{x,j}, \beta_{y,j}, \beta_{z,j})^T$ and $\{\epsilon_t^{(j)}\}$ is a collection of independent identically distributed Gaussian random variables with mean zero and some specified variance. The square root transformation (with the constant 3/8) is a standard adjustment used when approximating Poisson-distributed counts using linear regressions models [58]. Directional tuning is therefore captured by virtue of the expansion of the dot product as

$$\beta^{(j)} \cdot V_{t-\text{lag}_j} = \left\|\beta^{(j)}\right\| \left\|V_{t-\text{lag}_j}\right\| \cos(\vartheta)$$

where ϑ is the angle between the vectors $\beta^{(j)}$ and $V_{t-\text{lag}_j}$. One such model can be fit to each neuron. This is relatively easy, since for a fixed value of lag_j, parameter estimation for the model (1) is trivial, as it is a simple linear regression. Typically, lag_j is chosen by fitting a linear regression to data from the j-th neuron at many possible lags (within a range of around -250 to $+250$ ms) and selecting the lag which yields the largest R^2 coefficient.

It is typically assumed that the P components of N_t are conditionally independent, given the process $\{K_t\}$. However, it is worth mentioning that a number of researchers have begun to consider (more realistic) models that do not make the simple assumption that the spiking behavior of individual neurons is conditionally independent given the kinematic (or other state) variable. They are working to develop models that allow for interaction between neurons, taking place for instance due to intrinsic neuronal network dynamic structure, that is not necessarily explained simply by the presence of a common explanatory variable. Such models are naturally more complex than those considered here, but offer the potential to yield further improvements in model goodness-of-fit and decoding, and are considered in, for example [48], [59], [60].

In the earlier literature, models like the one specified in (1) would be used, typically *without* additional specification of the probabilistic behavior of $\{K_t\}$ itself. More recently, it has been recognized that by specifying the (probabilistic) behavior of $\{K_t\}$ as well, one can exploit extra information about the nature of typical hand trajectories to perform better decoding (as discussed in Section III-D). The joint specification of dynamics for both hand motion and spiking activity is equivalent to the specification of a "generalized state-space" or "hidden Markov" model for hand motion and spiking activity. These models are well-studied in the engineering literature; many details of analysis and inference for the family of models can be found, for instance, in [61].

Although we would really like a model that accurately describes the distribution of hand movements over time, we have found that significant improvement in decoding performance can be obtained simply by using crude models of the form

$$K_t = \begin{bmatrix} X_t \\ V_t \end{bmatrix} = \begin{bmatrix} I_{3\times3} & \delta I_{3\times3} \\ 0 & \phi I_{3\times3} \end{bmatrix} K_{t-1} + \begin{bmatrix} 0 \\ \epsilon_t \end{bmatrix} \qquad (2)$$

where $\{\epsilon_t, t = 1, 2, \ldots\} \overset{iid}{\sim} N(0, \Sigma)$ for some 3×3 covariance matrix Σ, $\delta = 0.01$ (10 ms), and ϕ is some constant close in value to one (we use $\phi = 0.98$ in the analyses in this paper). Intuitively, (2) simply imposes the requirement that velocity changes smoothly over time. For the data we consider, with X_t measured in meters and V_t in meters per second, we take $\Sigma = \text{diag}(0.009, 0.009, 0.009)$, so that the standard deviation of the change in velocity in any particular coordinate axis over a single 10 ms time bin is 0.03 m/s. In other words, the model imposes smoothness by saying that in a time bin, mean acceleration is zero, standard deviation of acceleration over the 10 ms bin is taken to be 3 m/s^2.

The pair of (1), (2) makes up a state-space model that could be used to perform decoding. However, we can further refine the specification of the relationship between K_t and N_t. Since the observations are spike counts, it is clear that the Gaussian model (1) is not entirely appropriate. Further analysis and inspection of residuals from fitted models (see, for example, [62]) has led us to replace (1) by

$$N_t^{(j)} \sim \text{Poisson}\left(\lambda_j\left(X_{t-\text{lag}_j}, V_{t-\text{lag}_j}, \sigma_j \varepsilon_t^{(j)}\right)\right) \qquad (3)$$

where the neuron-specific "tuning function" $\lambda_j(\cdot)$ is given by

$$\lambda_j(x, v, e) = \exp\left(\beta_0^{(j)} + \beta^{(j)} \cdot v + \beta_1^{(j)} \|v\| + e\right) \qquad (4)$$

and $\{\varepsilon_t^{(j)}\}$ is a collection of independent standard normal (mean zero, unit variance) random variables. For simplicity, our functional form for λ here does not include any dependence on x. However, we note that if desired, one

could easily incorporate such dependence and carry through the remaining steps outlined in this section. We include a coefficient for $\|v\|$ because plots of spike counts suggest that some neurons exhibit a sensitivity to magnitude of velocity, regardless of direction. Note that the framework we describe below is capable of handling almost arbitrary parametric forms for the tuning function, so we are not restricted to the particular form of (4). Inclusion of the $\varepsilon_t^{(j)}$ term effectively says that there may be an unobserved additional source of noise driving the j-th neuron. A thorough discussion of a family of models with a more sophisticated version of this extension to the model can be found in the recent work of [49].

B. Parameter Estimation

Fitting the model (2)–(4) to our data requires estimation of the set of unknown parameters

$$\theta = (\theta_1, \ldots, \theta_P)$$

where

$$\theta_j = \left(\text{lag}_j, \beta_0^{(j)}, \beta_1^{(j)}, \beta^{(j)}, \log\left(\sigma_j^2\right)\right) \in \mathbb{R}^7$$

denotes the subset of parameters specific to the j-th neuron. (For convenience in the estimation algorithm described below, we reparameterize slightly, working with $\log(\sigma_j^2)$ instead of directly with σ_j^2 itself. This simply frees us from having to constrain the algorithm to only allow positive values of the parameter.) While the simple form of the model (1) allows for maximum likelihood estimation via linear regression (or by use of the Kalman filter when the model incorporates dynamics specified by (2)), in the more general case we are interested in, such simple methods cannot always be used, due to intractability of the likelihood function.

In particular, in (3), the presence of the terms $\varepsilon_t^{(j)}$ means that while it is easy to write down the conditional distribution $p(n_t^{(j)}|k_{t-\text{lag}_j}, \varepsilon_t^{(j)})$, the conditional distribution of $n_t^{(j)}$ given *only* $k_{t-\text{lag}_j}$ is more complicated. It could be computed by numerically integrating

$$p\left(n_t^{(j)}|k_{t-\text{lag}_j}\right) = \int p\left(n_t^{(j)}|k_{t-\text{lag}_j}, \varepsilon_t^{(j)}\right) p\left(\varepsilon_t^{(j)}\right) d\varepsilon_t^{(j)} \quad (5)$$

but this is generally computationally costly (although not in this case computationally prohibitive). In a natural extension where one allows the sequence $\{\varepsilon_t^{(j)}, t = 1, 2, \ldots\}$ to have serial dependence, it becomes even more difficult to deal with these terms by direct computation of the likelihood.

One possible approach to the problem is the use of the EM ("expectation maximization") algorithm [63] to perform approximate maximum likelihood estimation when the model includes a latent Markov process. Indeed, this kind of approach is proposed and explored for a family of models that generalizes (2)–(4), in [49]. Here, we explore another method, often used for data analysis with models including latent variables. The idea is to perform a Bayesian analysis, using Markov chain Monte Carlo simulation to estimate parameters (for general discussion of these methods, see, e.g., [64], [65]). We perform the analysis, using realized observations $\{K_{1:q} = k_{1:q},\ N_{1:q} = n_{1:q}\}$ over some set of q training observations. We assume that the data points at times $1, \ldots, q$ cover a sufficiently rich range of motion that parameters can indeed be estimated with some degree of precision.

A prior distribution ν_j is assigned to each unknown parameter vector θ_j. This prior distribution should in principle reflect one's belief before seeing data about what values the parameter may take. Then for each neuron j, we are interested in the posterior distribution

$$p\left(\theta_j|k_{1:q}, n_{1:q}^{(j)}\right) \propto \nu_j(\theta_j) p_{\theta_j}\left(k_{1:q}, n_{1:q}^{(j)}\right)$$

where $p_{\theta_j}(k_{1:q}, n_{1:q}^{(j)})$ denotes the joint likelihood of the observations $\{x_{1:q}, v_{1:q}, n_{1:q}^{(j)}\}$ under the model (2)–(4) when parameters are equal to θ_j. As pointed out above, it is not trivial to compute $p_{\theta_j}(k_{1:q}, n_{1:q}^{(j)})$. It is, however, easy to compute $p_{\theta_j}(k_{1:q}, n_{1:q}^{(j)}, \varepsilon_{1:q}^{(j)})$. To take advantage of this, we use MCMC to sample from the density

$$\begin{aligned}\xi_j\left(\theta_j, \varepsilon_{1:q}^{(j)}\right) &= p\left(\theta_j, \varepsilon_{1:q}^{(j)}|k_{1:q}, n_{1:q}^{(j)}\right) \\ &\propto \nu_j(\theta_j) p\left(\varepsilon_{1:q}^{(j)}|\theta_j\right) p(k_{1:q}|\theta_j) \\ &\quad \times p\left(n_{1:q}^{(j)}|k_{1:q}, \varepsilon_{1:q}^{(j)}, \theta_j\right)\end{aligned}$$

In other words, we will sample from the joint posterior distribution of the parameters *as well as* the latent variables $\varepsilon_{1:q}^{(j)}$. Then we simply ignore the sampled values of $\varepsilon_{1:q}^{(j)}$, and the sampled values of θ_j represent approximate draws from the marginal posterior distribution of interest.

The following procedure is a simple Metropolis-Hastings Markov chain Monte Carlo algorithm that, for the model we are considering, yields a sequence of draws from distributions that converge to the posterior distribution as the number of iterations of the algorithm increases.

Algorithm 1 Metropolis-Hastings for Model (2)–(4)

1) (Initialization) Choose an arbitrary initial guess $\theta_j^{(0)}$ of the parameter vector θ_j. Draw initial values $\{e_t^{(j,0)} \overset{\text{iid}}{\sim} N(0,1), t = 1, \ldots, q\}$ representing "guesses" of $\{\varepsilon_t^{(j)}, t = 1, \ldots, q\}$. Set the iteration counter $i = 0$.
2) (Parameter Updates) For $k = 1, \ldots, 7$:
   - Construct a proposal θ_j^* by setting $\theta_j^* = \theta_j^{(i)}$, then adding a $N(0, \rho_k^2)$-distributed "step" to its k-th component.
   - Evaluate the acceptance probability

$$\alpha = \min\left(1, \xi_j\left(\theta_j^*, e_{1:q}^{(j,i)}\right) / \xi_j\left(\theta_j^{(i)}, e_{1:q}^{(j,i)}\right)\right).$$

   - With probability α, set $\theta_j^{(i+1)} = \theta_j^*$. Otherwise set $\theta_j^{(i+1)} = \theta_j^{(i)}$.
3) (Latent Variable Updates) For $t = 1, \ldots, q$:
   - Construct a proposal e_t^* by drawing from a $N(0,1)$ distribution.
   - Evaluate the acceptance probability

$$\alpha = \min\left(1, \frac{p\left(n_t^{(j)} | k_{1:q}, \varepsilon_t^{(j)} = e_t^*, \theta_j^{(i)}\right)}{p\left(n_t^{(j)} | k_{1:q}, \varepsilon_t^{(j)} = e_t^{(j,i)}, \theta_j^{(i)}\right)}\right).$$

   - With probability α, set $e_t^{(j,i+1)} = e_t^*$. Otherwise set $e_t^{(j,i+1)} = e_t^{(j,i)}$.
4) Replace i by $i + 1$ and go back to Step 2.

The posterior distribution summarizes information contained in the data about the parameters of interest. Since the MCMC procedure gives approximate draws from the posterior distribution, we can use, as parameter estimates,

$$\hat{\theta}_j = \frac{1}{(m - B)} \sum_{i=B+1}^{m} \theta_j^{(i)} \tag{6}$$

where B represents a number of initial iterations of the chain to be labelled as "burn-in" iterations and discarded. This is a standard technique used to account for the fact that MCMC simulation yields Markov chains whose limiting marginal distributions are the desired "target" posterior distributions, but whose marginal distributions at early iterations may not be particularly close to the target distribution. A full explanation of convergence rates and mixing properties of MCMC algorithms is beyond the scope of this paper, but more details can be found, for example, in [64], [65]. Note also that we chose to collapse the posterior distribution to its mean for the sake of coming up with fixed parameter estimates for each neuron. There are, however, two obvious alternatives. One would be to find the posterior mode, that is the parameter value at which the posterior density is maximal. The mode, although more difficult to compute (there is not a simple formula like (6)) can be more robust when the Markov chain does not behave ideally. It can also be thought of as a Bayesian analog of a maximum likelihood estimate. Another more sophisticated alternative would be to retain the entire posterior distribution, using it to represent uncertainty about parameters. We will not explore the possibility further in this paper, but we will note in passing that in theory, one can use this information on parameter uncertainty within the decoding process to get more "honest" confidence intervals for decoded quantities.

C. Goodness of Fit

There is a range of possible tools for exploration of goodness-of-fit of a particular model.

One standard approach that can be used when spikes are modeled in continuous time is based on the idea of time rescaling [41], [66], [67]. The idea is to rescale time depending on the fitted point process rate function so that the time-rescaled spike train becomes, if the model is "correct," a homogeneous Poisson process. We use the term "correct" to mean that the data were indeed generated by the specified model, or by a model which induces the same distribution for the data as the specified model. Then the interspike intervals will be independent and identically distributed exponential random variables. A range of tests can then be performed on these interarrival times to verify whether or not this is plausibly the case.

Since in this paper we are considering discrete-time models, the time rescaling approach is not applicable. As an alternative, it would be convenient to be able to resort to standard time series analysis techniques that rely on inspection of so-called "residuals," which behave in certain ways when the fitted model is "correct." Residuals in the traditional sense being defined (as, for instance in, [68]) as the (possibly scaled) differences between one step minimum mean-square predictors and the corresponding observed values, are not easily interpretable for the state-space models we describe in this paper. Standard properties of such residuals rely on linearity and/or Gaussianity of the model, neither of which is a property of the models we are considering. However, it is possible to construct a generalized kind of residuals which can indeed be examined for our models to check goodness-of-fit. Such residuals are described in [69], and can be computed as follows. For times $t = 1, \ldots, q$ (recall that q is

the length of the training data), and fitted model specified by (2)–(4) along with a particular value of θ, one can compute the one-step predictive conditional cumulative density functions

$$\begin{aligned}\overline{r}_t^{(j)} &= P\left(N_t^{(j)} \le n_t^{(j)} | N_{1:t-1}^{(j)} = n_{1:t-1}^{(j)}\right) \\ &= \int P\left(N_t^{(j)} \le n_t^{(j)} | K_{t-1} = k_{t-1}\right) \\ &\quad \cdot dP\left(k_{t-1} | N_{1:t-1}^{(j)} = n_{1:t-1}^{(j)}\right) \end{aligned} \tag{7}$$

and their left limits

$$\underline{r}_t^{(j)} = P\left(N_t^{(j)} < n_t^{(j)} | N_{1:t-1}^{(j)} = n_{1:t-1}^{(j)}\right). \tag{8}$$

Since the observations are integer-valued counts, in general, $\overline{r}_t^{(j)}$ is not the same as $\underline{r}_t^{(j)}$. We then construct residuals by drawing (independently)

$$R_t^{(j)} \sim \text{Unif}\left(\underline{r}_t^{(j)}, \overline{r}_t^{(j)}\right). \tag{9}$$

Under the assumption of model "correctness," for each j, $\{R_t^{(j)}, j = 1, \ldots, q\}$ will be a sequence of independent and identically distributed random variables uniformly distributed on the interval [0,1].

D. Filtering/Decoding

Neuroscientists have developed a range of techniques for performing real-time estimation of the latent hand position/velocity vector. These include population vector approaches (as developed by [57]), an improved method known as "optimal linear estimation" [37], and others. The estimates yielded by these methods, which depend only on observed spike counts, and not on actual hand position/velocity itself, can in principle be used, for instance to drive a robotic prosthetic arm to mimic or replace actual hand motion. Although these approaches are elegant in their simplicity, they have begun to give way to a range of decoding methods based on the formal specification of models. By computing appropriate conditional expectations under the formal models, one can obtain decoded values that are guaranteed (under the model correctness assumption) to minimize the mean-squared error. To begin with, the methods combined specification of a linear Gaussian model with the Kalman filter [70]. The approach derived a significant advantage over the earlier methods in part because they incorporate information about likely behavior of the underlying dynamics of interest (as for example, encapsulated in (2)), and in part because the decoding method is matched to the model. A natural next step is to move toward decoding methods that are matched to (possibly) nonlinear and/or non-Gaussian models. Some of the work making use of filtering approaches can be found in [40], [46], [48], [49], [53]–[56].

We now describe a standard formal statistical likelihood-based signal processing (filtering) decoding procedure for the general family of nonlinear and non-Gaussian state-space models. The basic tool we describe is the use of so-called *sequential Monte Carlo* (also known as *particle filtering*) methods, although it should be noted that there are other ways to perform filtering (see e.g. [48], [53]). These are simulation-intensive numerical schemes, two of the noteworthy early publications being [71] and [72]. In the context of neural decoding, the use of numerical methods for filtering has been described in, for example, [46], [73], among others.

The goal of the algorithm is as follows. Once we have computed parameter estimates $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_P)$ (for example, using the MCMC approach described in Section III-B) we next need to find the conditional distributions

$$\pi_t(k_t) = p(k_t | n_1, n_2, \ldots, n_t) \tag{10}$$

for $t = 1, 2, \ldots$. We want to do this under the assumption that the data comes from the model (2)–(4) with $\theta = \hat{\theta}$. These distributions are often referred to as "filtering distributions," and under the assumption of model correctness, each corresponding expected value

$$\hat{K}_t = \mathbf{E}[K_t | N_1 = n_1, \ldots, N_t = n_t] = \int k_t \pi_t(dk_t) \tag{11}$$

is the minimum mean-squared error estimate of K_t based on information available up to time t. Note that we will be most interested in π_t for $t > q$ (recall that q was the length of our initial training data). However, in assessing model goodness-of-fit, the filtering distributions for $t \le q$ can also be used to (numerically) evaluate the integral appearing in (7).

In its most basic (but not necessarily most efficient) form, the full recursive filtering algorithm can be described for the filtering problem as follows. We will be approximating the conditional densities $\pi_t(k_t) = p(k_t | n_1, \ldots, n_t)$ by "particle approximations"

$$\hat{\pi}_t(dk) = \sum_{i=1}^{m} \delta_{k_t^{(i)}}(dk) \tag{12}$$

where m is some positive number of "particles," and $k_t^{(i)}$ represents an i-th draw (particle) from a distribution closely approximating π_t. Thus we can use a Monte Carlo approximation, replacing (11) by

$$\hat{K}_t \simeq \frac{1}{m}\sum_{i=1}^{m} k_t^{(i)}. \tag{13}$$

The algorithm prescribes a recursive method for obtaining the particles $\{k_t^{(i)}, i = 1, \ldots, m, t = 1, 2, \ldots\}$ in these approximations, and is a straightforward application of existing particle filtering methods to our model. In fact, it can be regarded as a special case of Algorithm 3 of the subsequent paper in this special issue [74], with the proposals the authors describe in their Section II-C.

To state the algorithm, it will be helpful to define the "lag-adjusted" spike count vector

$$\tilde{n}_t = \left(n_{t+\text{lag}_1}^{(1)}, n_{t+\text{lag}_2}^{(2)}, \ldots, n_{t+\text{lag}_P}^{(P)}\right). \tag{14}$$

Algorithm 2 Bootstrap Filter for Neural Decoding

1) (Initialization) Choose some positive number of particles m and obtain initial draws $\{k_1^{(i)}, i = 1, \ldots, m\}$ from a distribution approximating π_1. Set $t = 1$.
2) (Forward Simulation) For $i = 1, \ldots, m$:
   - Draw $\tilde{k}_{t+1}^{(i)}$ from $p(k_{t+1}|K_t = k_t^{(i)})$. (This distribution is specified by (2).)
   - Compute the weight $w_i = p(\tilde{n}_{t+1}|\tilde{k}_{t+1}^{(i)})$. (This weight is determined by (3) and (4).)
3) (Resampling) Draw a sample of size m from $\{\tilde{k}_{t+1}^{(i)}, i = 1, \ldots, m\}$, with replacement, with weights proportional to $\{w_i, i = 1, \ldots, m\}$. This sample becomes $\{k_{t+1}^{(i)}, i = 1, \ldots, m\}$.
4) Replace t by $t + 1$ and go back to Step 2.

An important computational feature of the algorithm is its recursive nature. In order to complete the t-th iteration, it is only necessary to keep track of the particles from the $(t-1)$st iteration, and use these in conjunction with the fitted model and the spike count (vector) observation n_t. Thus it can be implemented in real-time, yielding a moving cloud of particles, whose sample average we compute (c.f. (13)) in order to obtain optimal estimates of hand position/velocity.

In the algorithm, to assign weights, we (again) have to face the problem of computing the terms $p(n_t^{(j)}|k_{t-\text{lag}_j})$, as we have $p(\tilde{n}_{t+1}|\tilde{k}_{t+1}^{(i)}) = \prod_{j=1}^{p} p(n_{t+1+\text{lag}_j}^{(j)}|K_{t+1} = \tilde{k}_{t+1}^{(i)})$. To evaluate each term in the product in this expression, we can perform numerical integration of the expression in (5). Note also that use of the lag-adjusted spike count vector means that if we are to perform real-time decoding, we must introduce a delay of $(\max_j \text{lag}_j)$ time bins to gather all relevant spike counts before decoding can be performed. Of course, one can simply choose to incorporate only those neurons whose lags are negative (meaning neural activity occurs before motor activation), but this would lead to some loss of information. As a compromise, one might also tolerate some delay in decoding and simply ignore those neurons with a lag parameter larger than some specified value.

To implement the algorithm, the number of particles m must be chosen. Theoretical results guarantee that approximations become accurate as the number of particles increases, but in practice one must choose a finite number. We find that for models like (2)–(4), around 1000 particles provides a good balance between accuracy of approximation (of the filtering distributions π_t) and computational load. In general, as the model becomes more complex, for instance, if the state model (2) becomes nonlinear or the state vector becomes higher dimensional, it is necessary to increase the number of particles to maintain the same quality of approximation.

Although the algorithms are computationally demanding, a large amount of effort has been devoted to improving their efficiency, and detailed discussion can be found in the next paper in this special issue [74] (see in particular Sections II-D and II-E), or in other texts such as [75] or [61].

E. Data Analysis Results

For analysis of the data we introduced in Section III, we implemented the Markov chain Monte Carlo algorithm for estimation of parameters θ_j, $j = 1, \ldots, 70$, with prior distributions $\log(\sigma_j) \sim N(-3, 1.5^2)$, $\text{lag}_j \sim \text{Unif}(-25, -24, \ldots, 24, 25)$, $\beta_0^{(j)} \sim N(0, 10^{12})$, $\beta_1^{(j)} \sim N(0, 10^{12})$, $\beta^{(j)} \sim N([0, 0, 0], \text{diag}(10^{12}, 10^{12}, 10^{12}))$, for each neuron j, and took the prior joint density $\nu_j(\theta_j)$ to be the product of the individual marginal prior densities. Essentially, these choices say that we have almost no information a-priori about coefficient values, but we assume that lag is restricted to the range -25 to $+25$ (10 ms) time-bins, and that we believe the σ coefficients to be relatively small in magnitude. Within the algorithm, we used step size standard deviations $\rho_1^2 = \ldots = \rho_7^2 = 0.004$. (These were chosen so that $\sqrt{0.004}$ is approximately of the same order of magnitude as plausible parameter values.) To initialize the algorithm, we used standard maximum likelihood methods to estimate parameters for the generalized linear model that one would obtain by replacing the term $\varepsilon_t^{(j)}$ from the model (3) with zero. Then we used the estimated coefficients as initial values for the corresponding coefficients in (4), and initialized $\log(\sigma_j) = -3.0$.

We chose the number of iterations m to be 10 000, and discarded the first 2500 iterations as burnin (i.e. we chose

$B = 2500$ in (6)). Visual inspection of plots of the iterations of the Markov chain suggested that a stationary regime had been reached after a few hundred iterations, so we believe using the last 7500 out of 10 000 iterations was a conservatively cautious choice in this case. Of course, the total number of iterations was not constrained in our analysis since we performed a static analysis of an existing data set.

In the case of a real-time system, one would ideally choose m to be as large as is computationally feasible, given whatever timing constraints exist in the system. For example, a BMI device might require occasional updates to account for changes in brain function and/or electrode performance which would be reflected in changed model parameters. The update procedure would in many cases have to be scheduled in with normal operation of the device, and scheduling constraints would determine the maximal allowable time to be spent on parameter estimation. Our implementation of the algorithm on a single AMD Opteron 250 processor-based machine, running at 2.4 gigahertz, was able to get through approximately 75 iterations per second, with a training set size of $q = 8041$. Thus to perform around 500 iterations (which for this data would be arguably a minimal number) for 70 neurons, a typical modern quad-core CPU would take around 2 min, which is not unreasonable in many settings. (Arguably the performance of the decoding algorithm is more important for effective operation of a prosthetic device.)

In terms of goodness of fit, we assessed our models by evaluating residuals (assuming fixed parameter values) $R_t^{(j)}$ as described in Section III-C above. (To evaluate the conditional cumulative distribution functions required in equations (7) and (8), we used the bootstrap filter algorithm to obtain the one-step predictive distributions, then performed a straightforward numerical integration to evaluate the required functions.) The first 1000 (out of 8041) residuals for neuron #1 are shown in Fig. 4, for (1) the best linear state-space model of the form (1), (2), with parameters estimated by maximum likelihood estimation, and (2) the fitted model of the form (2)–(4), with parameters estimated by MCMC. Under the assumption of model correctness, the residuals should be independent realizations of random variables uniformly distributed on the interval [0,1]. The poor appearance of the residuals for the linear state-space model is in part explained by the fact that it is a Gaussian model being used to describe observations of counts, which are discrete. The residuals correctly show that the model is not explaining the discrete distribution very well. On the other hand, in spite of the poor match for a uniform distribution, the mean of the residuals is close to 0.5 as would be expected, suggesting that the model still might be reasonable in terms of capturing the mean or median of the data. The residuals for the more complex model, fit with the MCMC algorithm, visually appear fairly consistent with realizations of independent uniformly distributed random variables. Residuals for the other 69 neurons look qualitatively very similar to those for neuron #1.

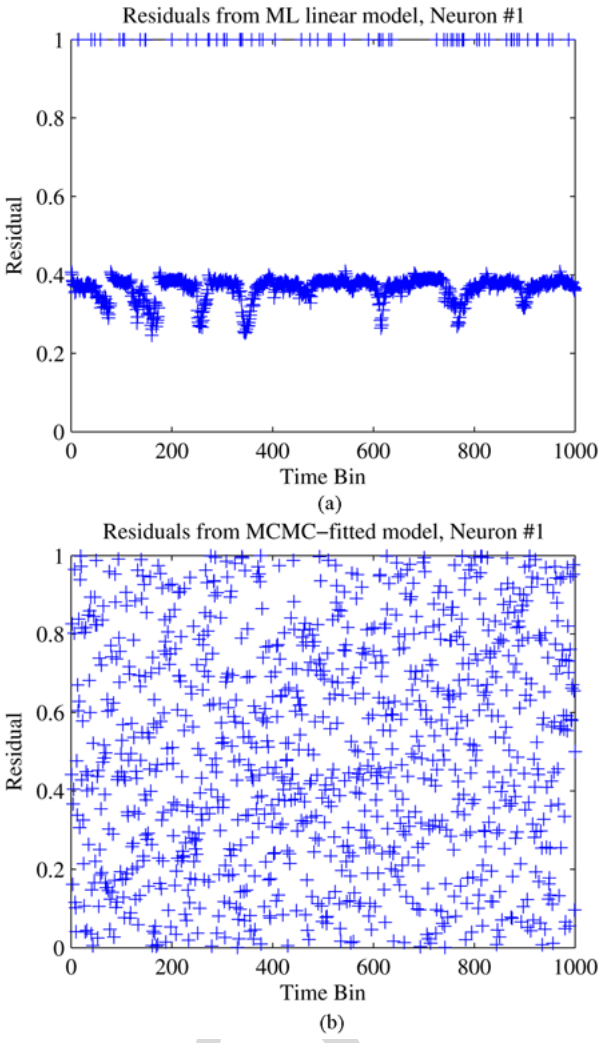

Fig. 4. *Residuals as defined in Section III-C, for (top) the linear Gaussian state-space model (1), (2) with parameters fit by maximum likelihood estimation, and (bottom) the model (2)–(4) with parameters fit by MCMC simulation. Under the assumption of model "correctness" these should be independent and uniformly distributed on the interval [0,1].*

To perform decoding, we implemented the bootstrap filter algorithm described in Section III-D. Beyond parameter estimates obtained already, the only choices we needed to make at this point were how to initialize our approximate draws from π_1, and the number of particles to be used. We chose π_1 to be the (trivial) distribution concentrated entirely on a single point—that is, we chose all our initial particles to be the initial position/velocity (bundled into the appropriate 6-D vector). In our context,

this makes sense because the initial position is determined by the experimental setup and the initial velocity is constrained to be zero. We used $m = 1500$ particles in decoding. (Increasing the number of particles to $m = 10\,000$ particles gave relatively little change in the sum of squared decoding errors—at most around 5%—so we judged 1500 to give a reasonable compromise between computational requirement and efficiency in terms of accurately determining the filtering distributions.) With this number of particles, a single step in the particle filter took around 100 ms on the aforementioned Opteron 250-based machine. With a more modern quad-CPU machine, one could carry out one iteration of this in less than 25 ms, which is around the level required for real-time control of a device. (The decoding algorithm is easily parallelizable, as the computational load is almost entirely in computing the sequence of weights, and these computations can be carried out independently of each other.) With more careful implementation and use of the various improvements as layed out in [61], [74], [75], we would expect to be able to improve performance significantly beyond this. One way to do this, for example, is to go beyond the bootstrap filter and perform better forward simulation within the algorithm. This can be done by simulating from what are referred to as "adapted proposals" and making corresponding adjustments to the weights.

Decoded velocities, along with 95% pointwise confidence bands are shown in Fig. 5. The corresponding positions along with 95% confidence bands are shown in Fig. 6. For comparison, decoded velocities obtained using the maximum likelihood-fitted linear Gaussian model (1), (2) along with the Kalman filter, are shown in Fig. 7.

Finally, in Table 1, we give summary results of the two different model/decoding methods in simple terms of the total sum-of-squared decoding errors in reaching motions 51 to 57. For additional comparison, we also give the sum-of-squares of decoding errors under a cruder model. The cruder model is simply the linear regression model (1), but *without* incorporating any specification of state dynamics as in (2). Under the linear regression model, it is simple to carry out decoding by directly maximizing the likelihood of the (lag-adjusted) spike counts with respect to the unobserved position/velocity vector K_t. Interestingly, the primary reduction in error in this case appears to come from the incorporation of the state dynamics (2). Apart from this, the different specifications of the distribution of the counts given the state make only a modest difference in terms of sums-of-squared errors.

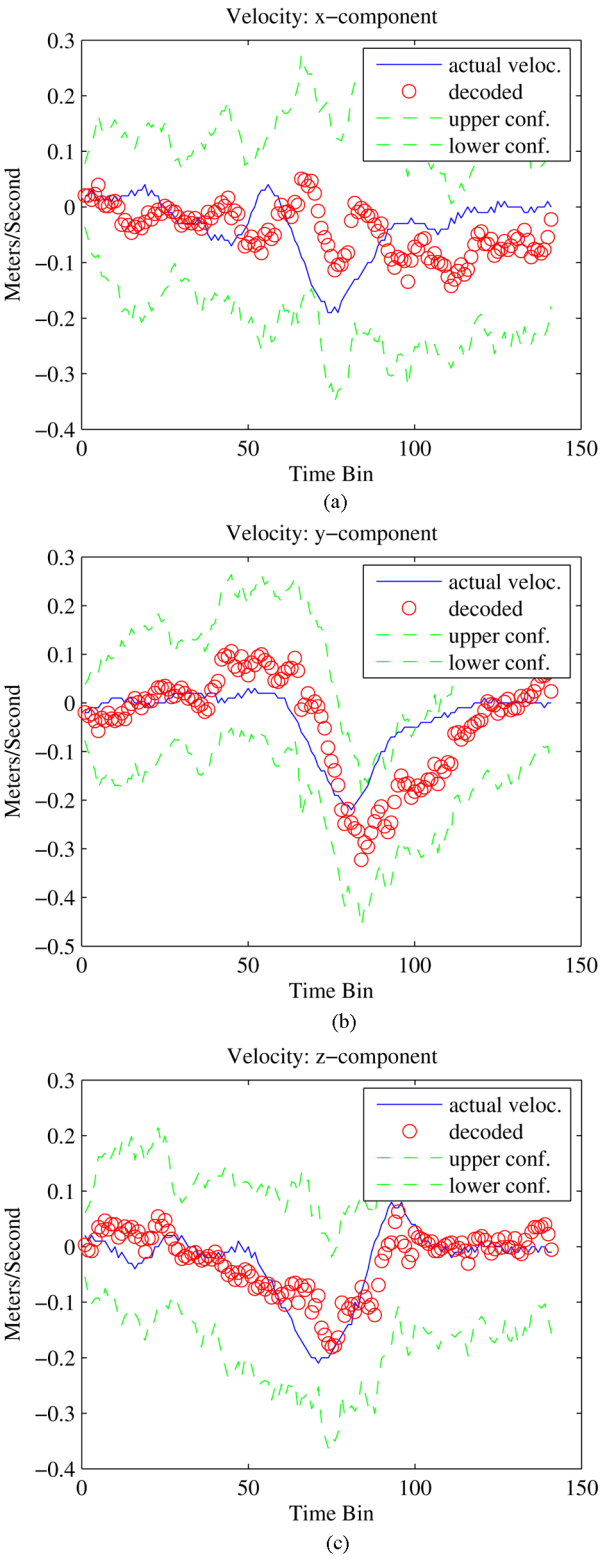

Fig. 5. ***Decoded and actual velocities for the center-out experimental data, based on the MCMC-fitted model (2)–(4), for the 51st reaching motion. The three components of velocity are shown (meters/sec), with the dashed lines indicating 95% pointwise confidence bands.***

IV. PROSTHETIC DEVICES

To build a brain-controlled prosthetic device, we would like to be able to use the procedures described above to fit state-space models to capture intended hand motion, then use these models for decoding to obtain control signals to

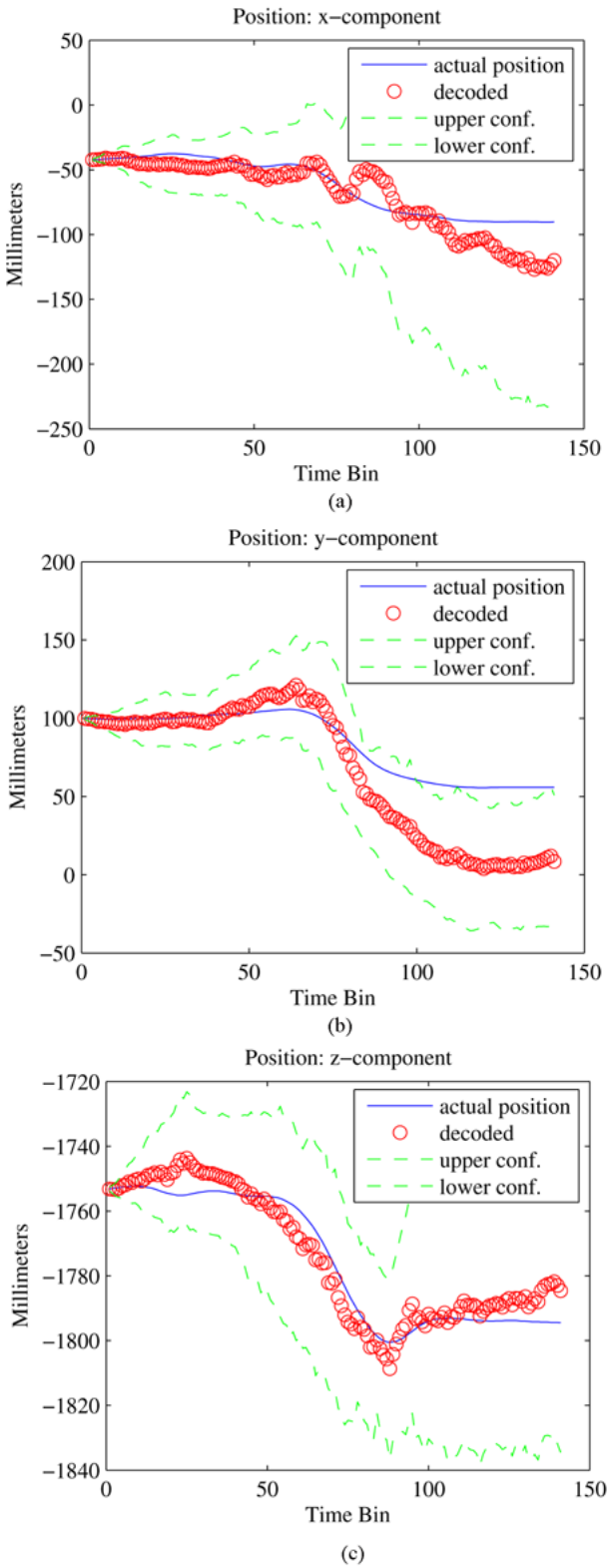

Fig. 6. *Decoded and actual positions for the center-out experimental data, based on the MCMC-fitted model (2)–(4), for the 51st reaching motion. The three components of position are shown (m/s), with the dashed lines indicating 95% pointwise confidence bands. Note that, as expected, uncertainty increases as errors in velocity are accumulated.*

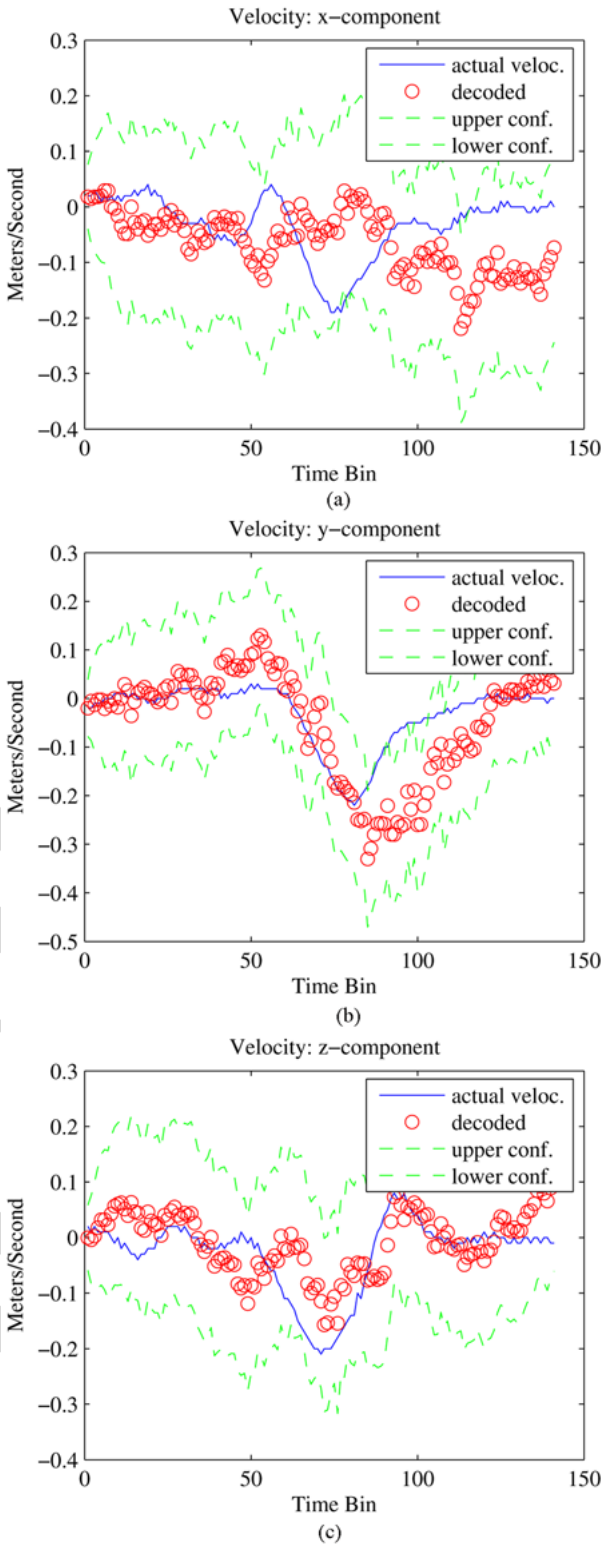

Fig. 7. *Decoded and actual velocities for the center-out experimental data, based on the linear state-space model (1), (2), with parameters estimated by maximizing likelihood, for the 51st reaching motion. The three components of velocity are shown (meters/sec), with the dashed lines indicating 95% pointwise confidence bands.*

Table 1 Sum-of-Squared Errors in Velocities When Decoding Reaching Motions #51 to #57, Using Three Different Model/Decoding Methods. (Reaching Motion #55 Was Canceled During the Course of the Experiment, so There Are no Results for It.)

Sum-of-Squared Errors in Decoded Velocities			
Model Fitting Alg.	Linear Reg. MLE	Linear State-Sp. MLE	Model (2-4) MCMC
Decoding Alg.	MLE	Kalman Filter	Bootstrap Filter
#51	5.46	2.54	1.68
#52	4.42	1.76	1.90
#53	4.89	2.92	1.87
#54	4.97	2.95	2.77
#56	6.34	3.05	2.14
#57	6.79	2.49	1.89

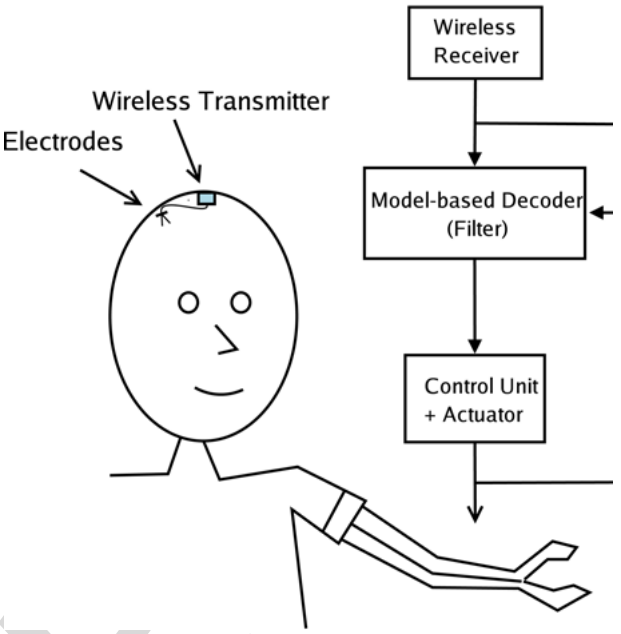

Fig. 8. *Block diagram of a potential robotic prosthetic device controlled directly from the motor cortex.*

apply to the prosthetic device. However, there is one obvious and significant difficulty. Patients who are to receive such a device generally will not have limb motion to begin with, hence it is not clear what training data might be used to estimate model parameters to begin with. [31] devised an effective solution to this problem based essentially on intermittent updates of a fitted model. A generalized version of their approach is illustrated in Fig. 8.

The prosthetic device works as follows. To begin with, a model-based decoder is designed to take spike activity input and translate it to "intended hand motion." This could be a particle filter-type decoder as described in Section III-D. It is based on an arbitrary guess at the true model relating intended hand motion to spiking activity. The output of this decoder is used to drive a control unit operating the prosthetic arm. The device is then operated for some amount of time, and spiking data are recorded along with prosthetic data are recorded. So far, because the decoder was designed based on a poor quality model, the device will not function very well. At this point, the "intent estimator" module can review recorded data and estimate actual intended motion over the time period. For a monkey, this can be done simply by assuming that the monkey was "trying" to move the prosthetic arm toward a target. For a human, more elaborate methods could be used to elicit intent. Next, the "model update" module can use recorded estimated intent and spike trains to build a new model relating the two quantities. This could be done, for instance, using the MCMC approach described in Section III-B. As soon as this is done, the decoder can take advantage of the new model, and the whole process can repeat itself. Over successive iterations, we expect performance of the device to improve.

Approaches like this have been implemented by several groups, see, e.g. [76]–[79]. In the Motorlab at the University of Pittsburgh, such a scheme has been implemented, with a monkey trained to use a robotic arm to reach for a piece of fruit. They used relatively simple linear regression models, along with population vector decoding schemes. Although, arguably, these techniques are not as sophisticated as the likelihood-based schemes we have discussed, the monkey was able to learn to use the device fairly rapidly. A digitally recorded movie of the result is available in the file `Direct3DRobot-Control.mpg`, which is available for download on the internet at http://www.motorlab.neurobio.pitt.edu/videos.

Two important aspects of these schemes are the dual closed-loop feedback nature of the experiment, and the brain's inherent adaptivity and plasticity (as referred to briefly in Section II-C). In experiments, at the same time that the model is updated in an attempt to better capture the monkey's intent, the monkey *is also* adapting to the device. Furthermore, this adaptation appears to take place rapidly—on the order of minutes. Such adaptation has also been noted and explicitly discussed in [1], [80]. This raises an interesting question—namely, to what degree is it important to accurately estimate intent for a BMI device to function effectively? Given the ability of the brain to adapt to behavior of a particular BMI device, it may be possible to obtain good performance with relatively simple decoding algorithms, and indeed this has been demonstrated in a number of the papers that we have referred to. Further understanding of the relative importance of high-quality decoding methods and brain adaptation for a BMI remains an important topic of research. ■

Acknowledgment

The authors are grateful to several anonymous reviewers whose insightful comments and suggestions helped improve this document.

REFERENCES

[1] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain–machine interface for reaching and grasping by primates," *PLoS Biol.*, vol. 1, pp. 193-208, 2003.

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

[3] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proc. Nat. Acad. Sci. USA*, vol. 101, pp. 17 849–17 854, 2004.

[4] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, pp. 258–262, 2004.

[5] M. D. Serruya, A. H. Caplan, M. Saleh, D. S. Morris, and J. P. Donoghue, "The BrainGate (TM) pilot trial: Building and testing a novel direct neural output for patients with severe motor impairment," in *Abstract Viewer/ Itinerary Planner*. Washington, DC: Society for Neuroscience, 2004.

[6] A. B. Schwartz, "Cortical neural prosthetics," *Annu. Rev. Neurosci.*, vol. 27, pp. 487–507, 2004.

[7] G. Santhanam, S. I. Ryu, B. M. Yua, A. Afshar, and K. V. Shenoy, "A high performance brain-computer interface," *Nature*, vol. 442, pp. 195–198, 2006.

[8] L. R. Hochberg, M. D. Serruya, G. H. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, pp. 164–171, 2006.

[9] P. J. Rousche and R. A. Normann, "Chronic recording capability of the Utah intracortical electrode array in cat sensory cortex," *J. Neurosci. Methods*, vol. 82, pp. 1–15, 1998.

[10] D. R. Kipke, R. J. Vetter, J. C. Williams, and J. F. Hetke, "Silicon-substrate intracortical microelectrode arrays for long-term recording of neuronal spike activity in cerebral cortex," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 151–155, Jun. 2003.

[11] S. Finger, *The Origins of Neuroscience: A History of Explorations Into Brain Function*. Oxford, U.K.: Oxford Univ. Press, 1994.

[12] P. P. Broca, "Loss of speech, chronic softening and partial destruction of the anterior left lobe of the brain," *Bulletin de la Socit Anthropologique*, vol. 2, pp. 235–238, 1861.

[13] G. Fritsch and E. Hitzig, "On the electrical excitability of the cerebrum (1870)," in *Some Papers on the Cerebral Cortex*, (G. von Bonin). Springfield, IL: Thomas, 1960.

[14] D. Ferrier, *The Functions of the Brain*, 2nd ed. London, U.K.: Smith, Elder, 1876.

[15] V. Mountcastle, "An organizing principle for cerebral function: The unit model and the distributed system," in *The Mindful Brain*. Cambridge, MA: MIT Press, 1978.

[16] H. Asanuma and H. Sakata, "Functional organization of a cortical efferent system examined with focal depth stimulation in cats," *J. Neurophysiol.*, vol. 30, 1967.

[17] H. Asanuma and I. Rosen, "Topographical organization of cortical efferent zones projecting to distal forelimb muscles in the monkey," *Exp. Brain Res.*, vol. 14, 1972.

[18] E. Jankowska, Y. Padel, and R. Tanaka, "Projections of pyramidal tract cells to alpha motoneurons innervating hind-limb muscles in the monkey," *J. Physiol.*, vol. 249, 1975.

[19] E. Jankowska, Y. Padel, and R. Tanaka, "The mode of activation of pyramidal tract cells by intracortical stimuli," *J. Physiol.*, vol. 249, no. 3, pp. 617–636, 1975.

[20] J. A. Rathelot and P. L. Strick, "Muscle representation in the macaque motor cortex: An anatomical perspective," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 21, pp. 8257–8262, 2006.

[21] E. V. Evarts, "Relation of pyramidal tract activity to force exerted during voluntary movement," *J. Neurophys.*, vol. 31, pp. 14–27, 1968.

[22] E. E. Fetz and P. D. Cheney, "Postspike facilitation of forelimb muscle activity by primate corticomotoneuronal cells," *J. Neurophys.*, vol. 44, pp. 751–771, 1980.

[23] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex," *J. Neurosci.*, vol. 2, pp. 1527–1537, 1982.

[24] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, pp. 1416–1419, 1986.

[25] A. B. Schwartz, D. W. Moran, and G. A. Reina, "Differential representation of perception and action in the frontal cortex," *Science*, vol. 303, pp. 380–383, 2003.

[26] J. N. Sanes, S. Suner, and J. P. Donoghue, "Dynamic organization of primary motor cortex output to target muscles in adult rats. I. Long-term patterns of reorganization following motor or mixed peripheral nerve lesions," *Exp. Brain Res.*, vol. 79, pp. 479–491, 1990.

[27] R. J. Nudo, B. M. Wise, F. SiFuentes, and G. W. Milliken, "Neural substrates for the effects of rehabilitative training on motor recovery after ischemic infarct," *Science*, vol. 272, pp. 1791–1794, 1996.

[28] A. Karni, G. Meyer, P. Jessard, M. M. Adams, R. Turner, and L. G. Ungerleider, "Functional mri evidence for adult motor cortex plasticity during motor skill learning," *Nature*, vol. 377, pp. 155–158, 1998.

[29] F. Gandalfo, C. S. R. Li, B. J. Benda, C. P. Schioppa, and E. Bizzi, "Cortical correlates of learning in monkeys adapting to a new dynamical environment," *Proc. Nat. Acad. Sci. USA*, vol. 97, pp. 2259–2263, 2000.

[30] S. P. Wise, S. L. Moody, K. J. Blomstrom, and A. R. Mitz, "Changes in motor cortical activity during visuomotor adaptation," *Exp. Brain Res.*, vol. 121, pp. 285–299, 1998.

[31] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, pp. 1829–1832, 2002.

[32] R. Paz, T. Boraud, C. Natan, H. Bergman, and E. Vaadia, "Preparatory activity in motor cortex reflects learning of local visuomotor skills," *Nat. Neurosci.*, vol. 6, pp. 882–890, 2003.

[33] G. Q. Bi and M. M. Poo, "Synaptic modifications by correlated activity: Hebb's postulate revisited," *Ann. Rev. Neurosci.*, vol. 24, pp. 139–166, 2001.

[34] M. Stopfer, X. Chen, Y. Tai, G. S. Huang, and T. J. Carew, "Site specificity of short-term and long-term habituation in the tail-elicited siphon withdrawal reflex of aplysia," *J. Neurosci.*, vol. 16, pp. 4923–4932, 1996.

[35] D. Brillinger, "Maximum likelihood analysis of spike trains of interacting nerve cells," *Biol. Cybern.*, vol. 59, pp. 189–200, 1998.

[36] H. S. Seung and H. Sompolinsky, "Simple model for reading neuronal population codes," *Proc. Nat. Acad. Sci. USA*, vol. 90, pp. 10 749–10 753, 1993.

[37] E. Salinas and L. F. Abbott, "Vector reconstruction from firing rates," *J. Comput. Neurosci.*, vol. 1, pp. 89–107, 1994.

[38] T. D. Sanger, "Probability density estimation for the interpretation of neural population codes," *J. Neurophysiol.*, vol. 76, no. 4, pp. 2790–2793, 1996.

[39] K. Zhang, I. Ginzburg, B. L. McNaughton, and T. J. Sejnowski, "Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells," *J. Neurophysiol.*, vol. 79, pp. 1017–1044, 1998.

[40] E. N. Brown, D. P. Nguyen, L. M. Frank, M. A. Wilson, and V. Solo, "An analysis of neural receptive field plasticity by point process adaptive filtering," *Proc. Nat. Acad. Sci.*, vol. 98, pp. 12 261–12 266, 2001.

[41] R. Barbieri, M. C. Quirk, L. M. Frank, M. A. Wilson, and E. N. Brown, "Construction and analysis of non-Poisson stimulus response models of neural spike train activity," *J. Neurosci. Methods*, vol. 105, pp. 25–37, 2001.

[42] R. Kass and V. Ventura, "A spike-train probability model," *Neural Comput.*, vol. 13, pp. 1713–1720, 2001.

[43] E. N. Brown, R. Barbieri, U. T. Eden, and L. M. Frank, *Likelihood Methods for Neural Spike Train Data Analysis*. Boca Raton, FL: CRC, 2003, ch. 9, pp. 253–286.

[44] Y. Gao, M. J. Black, E. Bienenstock, S. Shoham, and J. P. Donoghue, "Probabilistic inference of hand motion from neural activity in motor cortex," in *Advances in Neural Information Processing Systems*. Cambridge, MA: The MIT Press, 2002, vol. 14.

[45] C. Kemere, K. V. Shenoy, and T. H. Meng, "Model-based neural decoding of reaching movements: A maximum likelihood approach," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 925–932, Jun. 2003.

[46] A. Brockwell, A. Rojas, and R. Kass, "Recursive Bayesian decoding of motor cortical signals by particle filtering," *J. Neurophysiol.*, vol. 91, pp. 1899–1907, 2004.

[47] L. Paninski, "Maximum likelihood estimation of cascade point-process neural encoding models," *Netw. Comput. Neural Syst.*, vol. 15, pp. 243–262, 2004.

[48] W. Truccolo, U. T. Eden, M. R. Fellos, J. P. Donoghue, and E. N. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble and extrinsic covariate effects," *J. Neurophys.* vol. 93, pp. 1074–1089, 2005.

[49] J. E. Kulkarni and L. Paninski, "Common-input models for multiple neural spike-train data," *Netw. Comput. Neural Syst.*, 2007.

[50] R. E. Kass, V. Ventura, and E. N. Brown, "Statistical issues in the analysis of neuronal data," *J. Neurophysiol.*, vol. 94, pp. 8–25, 2005.

[51] S. H. Scott, "Optimal feedback control and the neural basis of volitional motor control," *Nat. Rev. Neurosci.*, vol. 5, pp. 534–546, 2004.

[52] E. Todorov, "Direct cortical control of muscle activation in voluntary arm movements; A model," *Nature Neurosci.*, vol. 3, no. 4, pp. 391–399, 2000.

[53] R. Barbieri, L. M. Frank, D. P. Nguyen, M. C. Quirk, V. Solo, M. A. Wilson, and E. N. Brown, "Dynamic analyses of information encoding by neural ensembles," *Neural Comput.*, vol. 16, no. 2, pp. 277–307, 2004.

[54] A. C. Smith and E. N. Brown, "Estimating a state-space model from point process observations," *Neural Comput.*, vol. 15, pp. 965–991, 2003.

[55] S. Shoham, L. M. Paninski, M. R. Fellows, N. G. Hatsopoulos, J. P. Donoghue, and R. A. Normann, "Statistical encoding model for a primary motor cortical brain–machine interface," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 7, pp. 1312–1322, Jul. 2005.

[56] W. Wu, M. Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhouni, and J. Donoghue, "Neural decoding of cursor motion using a Kalman filter," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, vol. 15.

[57] A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz, "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," *Neuroscience*, vol. 8, pp. 2928–2937, 1988.

[58] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed. London, U.K.: Chapman & Hall, 1989.

[59] M. Okatan, M. A. Wilson, and E. N. Brown, "Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity," *Neural Comput.*, vol. 17, pp. 1927–1961, 2005.

[60] F. Rigat, M. de Gunst, and J. van Pelt, "Bayesian modelling and analysis of spatio-temporal neuronal networks," *Bayesian Anal.*, vol. 1, pp. 733–764, 2006.

[61] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. New York: Springer, 2005.

[62] A. E. Brockwell, "Likelihood based analysis of a class of generalized long memory models," *J. Time Series Anal.*, 2007.

[63] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Statistical Society*, Ser. B, vol. 39, pp. 1–38, 1977.

[64] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London, U.K.: Chapman & Hall/CRC, 1996.

[65] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer, 1999.

[66] E. N. Brown, R. Barbieri, V. Ventura, R. Kass, and L. M. Frank, "A note on the time-rescaling theorem and its implications for neural data analysis," *Neural Comput.*, vol. 14, no. 2, pp. 325–346, 2002.

[67] M. C. Wiener, "An adjustment to the time-rescaling method for application to short-trial spike train data," *Neural Comput.*, vol. 15, no. 11, pp. 2565–2576, 2003.

[68] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*, 2nd ed. New York: Springer, 1991.

[69] A. E. Brockwell, "Universal residuals: A multivariate transformation," *Stat. Probab. Lett.*, 2007, to appear.

[70] R. E. Kalman, "A new approach to linear prediction and filtering problems," *J. Basic Eng. (ASME)*, vol. 82D, pp. 35–45, 1960.

[71] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F*, vol. 140, pp. 107–113, 1993.

[72] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. Comput. Graph. Stat.*, vol. 5, no. 1, pp. 1–25, 1996.

[73] U. T. Eden, L. M. Frank, V. Solo, and E. N. Brown, "Dynamic analyses of neural encoding by point process adaptive filtering," *Neural Comput.*, vol. 16, no. 5, pp. 971–998, 2004.

[74] O. Cappé, S. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proc. IEEE*, vol. 95, no. 5, pp. xxx–xxx, May 2007.

[75] A. Doucet, N. de Freitas, and N. Gordon, Eds. *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[76] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, "Realtime control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neurosci.*, vol. 2, pp. 664–670, 1999.

[77] P. G. Patil, J. M. Carmena, M. A. L. Nicolelis, and D. A. Turner, "Ensembles of subcortical neurons as a source of motor control signals for a brainmachine interface in humans," *Neurosurgery*, vol. 55, no. 1, pp. 27–35, 2004.

[78] M. J. Black, E. Bienenstock, J. P. Donoghue, M. Serruya, W. Wu, and Y. Gao, "Connecting brains with machines: The neural control of 2d cursor movement," in *Proc. 1st Int. IEEE/EMBS Conf. Neural Engineering*, 2003, pp. 580–583.

[79] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141–142, 2002.

[80] W. Wu, A. Shaikhouni, J. P. Donoghue, and M. J. Black, "Closed-loop neural control of cursor motion using a kalman filter," in *Proc. IEEE Engineering in Medicine and Biology Society*, 2004, pp. 4126–4129.

ABOUT THE AUTHORS

A. E. Brockwell (Member, IEEE) received the Ph.D. degree in statistics at the University of Melbourne, Australia, in 1998.

He took a faculty position in the Department of Statistics at Carnegie Mellon University, Pittsburgh, PA, where he is currently Associate Professor. He also holds affiliated faculty positions with the Department of Machine Learning and the Parallel Data Lab at Carnegie Mellon University. He is serving as Associate Editor of the *Annals of Applied Statistics*. His research interests lie in analysis of dynamical systems, including theoretical, methodological and applications problems and associated computational techniques. Much of his work concentrates specifically on methods for analysis of generalized state-space models (also known as hidden Markov models).

Prof. Brockell is a member of the ASA and the IMS.

Robert E. (Rob) Kass received the Ph.D. degree in statistics from the University of Chicago, Chicago, IL, in 1980.

He has been been on the faculty of the Department of Statistics at Carnegie Mellon University, Pittsburgh, PA, since 1981 and served as Department Head from 1995 to 2004; he joined the Center for the Neural Basis of Cognition in 1997. His early work formed the basis for his book *Geometrical Foundations of Asymptotic Inference*, coauthored with Paul Vos, and published by John Wiley and Sons. Most of Kass's other statistical research has been in Bayesian inference. His major applied interests are in neuroscience. He has served as Chair of the Section for Bayesian Statistical Science of the American Statistical Association, Chair of the Statistics Section of the American Association for the Advancement of Science, Executive Editor of the international review journal *Statistical Science*, and founding Editor-in-Chief of the journal *Bayesian Analysis*.

He is an elected Fellow of the American Statistical Association, the Institute of Mathematical Statistics, and the American Association of the Advancement of Science. He has been recognized by the Institute for Scientific Information as one of the 10 most highly cited researchers, 1995–2005, in the category of mathematics.

A. B. Schwartz received the Ph.D. degree from the University of Minnesota, Minneapolis, in 1984 with a thesis entitled "Activity in the deep cerebellar nuclei during normal and perturbed locomotion."

He went on to a postdoctoral fellowship at the Johns Hopkins School of Medicine, Baltimore, MD, where he worked with Dr. Apostolos Georgopoulos, who was developing the concept of directional tuning and population-based movement representation in the motor cortex. While there, he was instrumental in developing the basis for 3-D trajectory representation in the motor cortex. In 1988, he began his independent research career at the Barrow Neurological Institute, Phoenix, AZ. There, he developed a paradigm to explore the continuous cortical signals generated throughout volitional arm movements. This was done using monkeys trained to draw shapes while recording single-cell activity from their motor cortices. After developing the ability to capture a high fidelity representation of movement intention from the motor cortex, he teamed up with engineering colleagues at Arizona State University to develop cortical neural prosthetics. The work has progressed to the point that monkeys can now use these recorded signals to control motorized arm prostheses to reach out grasp a piece of food and return it to the mouth. He moved from the Barrow Neurological Institute to the Neurosciences Institute, San Diego, CA, in 1995 and then to the University of Pittsburgh, Pittsburgh, PA, in 2002. In addition to the prosthetics work, he has continued to utilize the neural trajectory representation to better understand the transformation from intended to actual movement using motor illusions in a virtual reality environment.

AUTHOR QUERY

No query.