

Mixture of Trajectory Models for Neural Decoding of Goal-Directed Movements

Byron M. Yu, Caleb Kemere, Gopal Santhanam, Afsheen Afshar, Stephen I. Ryu, Teresa H. Meng, Maneesh Sahani and Krishna V. Shenoy

J Neurophysiol 97:3763-3780, 2007. First published 28 February 2007;

doi: 10.1152/jn.00482.2006

You might find this additional info useful...

This article cites 51 articles, 28 of which you can access for free at:

<http://jn.physiology.org/content/97/5/3763.full#ref-list-1>

This article has been cited by 11 other HighWire-hosted articles:

<http://jn.physiology.org/content/97/5/3763#cited-by>

Updated information and services including high resolution figures, can be found at:

<http://jn.physiology.org/content/97/5/3763.full>

Additional material and information about *Journal of Neurophysiology* can be found at:

<http://www.the-aps.org/publications/jn>

This information is current as of August 11, 2012.

Mixture of Trajectory Models for Neural Decoding of Goal-Directed Movements

Byron M. Yu,¹ Caleb Kemere,¹ Gopal Santhanam,¹ Afsheen Afshar,^{1,2} Stephen I. Ryu,^{1,3} Teresa H. Meng,¹ Maneesh Sahani,^{5,*} and Krishna V. Shenoy^{1,4,*}

¹Department of Electrical Engineering, ²Medical Scientist Training Program, ³Department of Neurosurgery, and ⁴Neurosciences Program, Stanford University, Stanford, California; and ⁵Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom

Submitted 6 May 2006; accepted in final form 11 February 2007

Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SI, Meng TH, Sahani M, Shenoy KV. Mixture of trajectory models for neural decoding of goal-directed movements. *J Neurophysiol* 97: 3763–3780, 2007. First published February 28, 2007; doi:10.1152/jn.00482.2006. Probabilistic decoding techniques have been used successfully to infer time-evolving physical state, such as arm trajectory or the path of a foraging rat, from neural data. A vital element of such decoders is the trajectory model, expressing knowledge about the statistical regularities of the movements. Unfortunately, trajectory models that both 1) accurately describe the movement statistics and 2) admit decoders with relatively low computational demands can be hard to construct. Simple models are computationally inexpensive, but often inaccurate. More complex models may gain accuracy, but at the expense of higher computational cost, hindering their use for real-time decoding. Here, we present a new general approach to defining trajectory models that simultaneously meets both requirements. The core idea is to combine simple trajectory models, each accurate within a limited regime of movement, in a probabilistic mixture of trajectory models (MTM). We demonstrate the utility of the approach by using an MTM decoder to infer goal-directed reaching movements to multiple discrete goals from multi-electrode neural data recorded in monkey motor and premotor cortex. Compared with decoders using simpler trajectory models, the MTM decoder reduced the decoding error by 38 (48) percent in two monkeys using 98 (99) units, without a necessary increase in running time. When available, prior information about the identity of the upcoming reach goal can be incorporated in a principled way, further reducing the decoding error by 20 (11) percent. Taken together, these advances should allow prosthetic cursors or limbs to be moved more accurately toward intended reach goals.

INTRODUCTION

Neural activity from motor cortical areas has been shown in a number of studies to be related to various aspects of the corresponding arm reach, including movement direction (Ashe and Georgopoulos 1994; Georgopoulos et al. 1986; Moran and Schwartz 1999; Riehle and Requin 1989; Tanji and Evarts 1976), movement extent (Riehle and Requin 1989), position (Ashe and Georgopoulos 1994; Paninski et al. 2004), velocity (Ashe and Georgopoulos 1994; Paninski et al. 2004), acceleration (Ashe and Georgopoulos 1994), posture (Caminiti et al. 1991; Scott and Kalaska 1997), speed (Moran and Schwartz 1999), joint angular velocity (Reina et al. 2001), force (Evarts 1968; Sergio and Kalaska 1998), and intended reach goal

(Messier and Kalaska 2000; Shen and Alexander 1997). Although the coding scheme used by motor cortical areas is still incompletely understood (Fetz 1992; Moran and Schwartz 2000; Scott 2004), the regularities in the relationship between neural activity and aspects of the arm reach have enabled the development of direct brain-controlled prosthetic devices (Carmena et al. 2003; Chapin et al. 1999; Hochberg et al. 2006; Kennedy and Bakay 1998; Musallam et al. 2004; Santhanam et al. 2006; Serruya et al. 2002; Taylor et al. 2002). These devices are designed to allow disabled patients to regain motor function through the use of prosthetic limbs, or computer cursors, that are controlled by neural activity.

One of the key components of a neural prosthetic device is its decoding algorithm, which translates neural activity into desired movements. Examples of decoding algorithms that translate neural activity around the time of the movement (termed *peri-movement activity*) into continuous arm trajectories include population vectors (Taylor et al. 2002) and linear filters (Carmena et al. 2003; Hochberg et al. 2006; Serruya et al. 2002). Both of these decoding algorithms assume a linear relationship between the neural activity and arm state. In general, the arm state may include, but is not limited to, arm position, velocity, and acceleration.

Although these linear decoding algorithms are effective, recursive Bayesian decoders have been shown to provide more accurate trajectory estimates (Brockwell et al. 2004; Brown et al. 1998; Wu et al. 2004, 2006). Recursive Bayesian decoders are based on the specification of a probabilistic model comprising 1) a *trajectory model*, which describes how the arm state changes from one time step to the next, and 2) an *observation model*, which describes how the observed neural activity relates to the time-evolving arm state. If the modeling assumptions are satisfied, then Bayesian estimation makes optimal use of the observed data. Unlike the aforementioned linear decoding algorithms, recursive Bayesian decoders provide confidence regions for the arm state estimates and allow for nonlinear relationships between the neural activity and arm state.

The function of the trajectory model is to build into the recursive Bayesian decoder prior knowledge about the form of the reaches. The model may reflect 1) the hard, physical constraints of the limb (for example, the elbow cannot bend backward), 2) the soft, control constraints imposed by neural

* These authors contributed equally to this work.

Address for reprint requests and other correspondence: K. Shenoy, Stanford University, 330 Serra Mall, CISX 319, Stanford, CA 94305-4075 (E-mail: shenoy@stanford.edu).

The costs of publication of this article were defrayed in part by the payment of page charges. The article must therefore be hereby marked "advertisement" in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.

mechanisms (for example, the arm is more likely to move smoothly than in a jerky motion), and 3) the physical surroundings of the patient and his/her objectives in that environment. These statistical regularities and constraints can be learned by observing the reaching behavior and fitting the trajectory model to the actual reaches, which is the approach adopted in this paper. The degree to which the trajectory model reflects the kinematics of the actual reaches directly affects the accuracy with which trajectories can be decoded from neural data. A commonly used trajectory model is the random-walk model (Brockwell et al. 2004; Brown et al. 1998), which captures the fact that arm trajectories tend to be smooth. In other words, small changes in arm state from one time step to the next are more likely than large changes. The random-walk model is part of a family of trajectory models based on linear dynamics perturbed by Gaussian noise, which we refer to collectively as *linear-Gaussian models*. Linear-Gaussian models have been successfully applied to decoding the path of a foraging rat (Brown et al. 1998; Zhang et al. 1998), as well as arm trajectories in ellipse tracing (Brockwell et al. 2004), pursuit tracking (Shoham et al. 2005; Wu et al. 2004, 2006), and "pinball" tasks (Wu et al. 2004, 2006).

When selecting a trajectory model, one is typically faced with a trade-off between how accurately the trajectory model captures the movement statistics and the computational demands of its corresponding decoder. For example, for real-time applications, we may decide to use a relatively simple trajectory model because of its low computational cost, even if it fails to capture some of the salient properties of the observed movements. Even though we may be able to identify a more complex trajectory model that could yield more accurate decoded trajectories, the computational demands of the corresponding decoder may be prohibitive in a real-time setting. In this work, we present a general approach to constructing trajectory models that can exhibit rather complex dynamical behaviors, whose decoder can be implemented to have the same running time as simpler trajectory models. The core idea is to combine simple trajectory models, each accurate within a limited regime of movement, in a probabilistic mixture of trajectory models (MTM) (Kemere et al. 2003, 2004a,b).

We demonstrate the utility of this approach by developing a mixture of trajectory models suitable for goal-directed movements in settings with multiple goals. A common usage mode of real-time prosthetic systems involves guiding a computer cursor to acquire discrete goals in the subject's virtual workspace. This design approach was adopted in studies using multi-electrode neural recordings in monkeys (Carmena et al. 2003; Serruya et al. 2002; Taylor et al. 2002) and humans (Hochberg et al. 2006) and both electroencephalographic (EEG) (Wolpaw and McFarland 2004) and electrocorticographic (ECoG) (Leuthardt et al. 2004) recordings in humans. As shown by Donoghue and colleagues (Hochberg et al. 2006), the goal-directed cursor control design can allow a paralyzed patient to operate a computer interface controlling a variety of useful functions, including television and simulated email, thus illustrating its immediate clinical benefits. Because of the prevalence of goal-directed reaching in everyday life, this goal-directed design is likely to continue to be fruitful in systems involving prosthetic limbs, in addition to computer cursors, that are driven by the brain's activity. Indeed, many of the basic movements a paralyzed patient would desire are goal

directed, such as reaching for a cup, picking up the phone, and feeding oneself. The utility of prosthetic systems based on goal-directed movements has fueled the development of statistical models and decoding algorithms tailored for goal-directed movements (Cowan and Taylor 2005; Kemere and Meng 2005; Kemere et al. 2002, 2003, 2004a,b; Srinivasan and Brown 2006; Srinivasan et al. 2005, 2006; Yu et al. 2005).

Goal-directed movements can be characterized by the following three properties. First, each movement is typically directed toward one of a (possibly large) number of discrete goals available in the subject's workspace. These goals may be visual targets presented on a computer screen or physical objects located near the subject. Second, repeated movements to the same goal are not identical. For example, there may be variability in movement speed or curvature. Third, the trajectories generally start at rest, proceed out to the desired goal, and end at rest. Previous studies considered goal-directed movements toward a single stationary (Kemere and Meng 2005; Srinivasan et al. 2005, 2006) or dynamic (Srinivasan and Brown 2006) goal with a known arrival time or stereotyped movements to multiple goals (Kemere et al. 2002, 2004b). In this work, we develop a mixture of trajectory models that captures all three properties of goal-directed movements and show how it can be used to decode movements from neural activity.

While the peri-movement neural activity is informative of the moment-by-moment details of the desired movement, there may be additional information available about the identity of the desired reach goal well before the desired time of movement onset. For example, if the phone rings, there is a greater chance that the goal of the upcoming reach will be the phone rather than the light switch. Even without such an external clue, the upcoming goal identity can often be inferred from neural activity related to motor preparation (termed *delay activity* because motor preparation is typically studied using a delayed-reach behavioral task) (e.g., Churchland et al. 2006b,c; Kurata 1993; Messier and Kalaska 2000; Riehle and Requin 1989; Shen and Alexander 1997; Weinrich and Wise 1982). The type of information conveyed by delay activity is categorically different from that provided by peri-movement activity. Whereas peri-movement activity specifies the moment-by-moment details of the arm trajectory (e.g., Ashe and Georgopoulos 1994; Moran and Schwartz 1999; Paninski et al. 2004; Schwartz 1992), delay activity has been shown to indicate the upcoming reach goal (Hatsopoulos et al. 2004; Musallam et al. 2004; Santhanam et al. 2006; Shenoy et al. 2003; Yu et al. 2004). It should be possible to use this goal information, when available, to improve the accuracy of the decoded trajectory. Brown and colleagues (Srinivasan et al. 2005, 2006) showed how to constrain free movement trajectories, given goal information that takes the form of a continuous distribution around a single goal. For multiple goals, the information usually takes the form of a discrete distribution across the possible goals. As with decoders we previously proposed (Kemere et al. 2002, 2003, 2004a,b), the MTM framework can naturally incorporate this goal information across multiple goals to improve the accuracy of the decoded trajectory. In contrast to a decoder that selects among a set of canonical trajectories (Kemere et al. 2002, 2004b), the MTM decoder can take into account behavioral variability across reaches to the same goal. Furthermore, the MTM decoder does not require the use of a linear filter,

which was used in tandem with a mixture of hidden Markov models (Kemere et al. 2003) and a set of canonical trajectories with independent Gaussian noise at each time point (Kemere et al. 2004a).

We first present the MTM framework in its general form. Then, we construct an MTM that is appropriate for goal-directed reaches in settings with multiple goals and show how it can be used to decode arm trajectories from neural data. Next, we detail the behavioral task and neural recordings, along with how goal information can be extracted from delay activity. Finally, we compare the decoding accuracy of the MTM decoder with that using simpler trajectory models.

METHODS

Mixture of trajectory models framework

Recursive Bayesian decoders require the specification of a trajectory model that describes the statistics of arm trajectories we expect to observe. Ideally, we seek to construct a complete model of neural motor control that captures the hard physical constraints of the limb (Chan and Moran 2006), the soft control constraints imposed by neural mechanisms, and the physical surroundings and context. One way to approximate such a complete model is to probabilistically combine trajectory models each of which is accurate within a limited regime of movement (Kemere et al. 2002, 2003, 2004a,b). Examples of movement regimes include different parts of the workspace, different reach speeds, and different reach curvatures. For the particular implementation tested here, each movement regime will correspond to movements heading toward a particular reach goal. At the onset of a new movement, the movement regime is unknown, or imperfectly known, and so the full trajectory model is composed of a mixture of the individual, regime-specific trajectory models. Here, we develop a recursive Bayesian decoder based on a mixture of trajectory models.

The task of decoding a continuous arm trajectory involves finding the likely sequences of arm states corresponding to the observed neural activity. At each time step t , we seek to compute the distribution of the arm state \mathbf{x}_t given the peri-movement neural activity $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t$ (denoted by $\{\mathbf{y}\}_t^t$) observed up to that time. This distribution is written $P(\mathbf{x}_t | \{\mathbf{y}\}_t^t)$ and termed the *state posterior*. Here, \mathbf{y}_t is a vector of binned spike counts across the neural population at time step t , and $t = 1$ corresponds to the time at which we begin to decode movement. If the actual movement regime m^* is perfectly known before the reach begins, then we can compute the state posterior based only on the individual trajectory model corresponding to that regime. This distribution is written $P(\mathbf{x}_t | \{\mathbf{y}\}_t^t, m^*)$ and termed the *conditional state posterior*. However, in general, the actual movement regime is unknown or imperfectly known, so we need to compute $P(\mathbf{x}_t | \{\mathbf{y}\}_t^t, m)$ for each $m \in \{1, \dots, M\}$, where M is the number of movement regimes (also referred to as mixture components).

To combine the M conditional state posteriors, we can simply expand $P(\mathbf{x}_t | \{\mathbf{y}\}_t^t)$ by conditioning on the movement regime m

$$P(\mathbf{x}_t | \{\mathbf{y}\}_t^t) = \sum_{m=1}^M P(\mathbf{x}_t | \{\mathbf{y}\}_t^t, m) P(m | \{\mathbf{y}\}_t^t) \tag{1}$$

In other words, the state posterior is a weighted sum of the conditional state posteriors. The weights $P(m | \{\mathbf{y}\}_t^t)$ represent the probability that the actual movement regime is m , given the observed spike counts up to time t . Bayes' rule can then be applied to these weights in Eq. 1, yielding the key equation for the MTM framework

$$P(\mathbf{x}_t | \{\mathbf{y}\}_t^t) = \sum_{m=1}^M P(\mathbf{x}_t | \{\mathbf{y}\}_t^t, m) \frac{P(\{\mathbf{y}\}_t^t | m) P(m)}{P(\{\mathbf{y}\}_t^t)} \tag{2}$$

The conditional state posteriors $P(\mathbf{x}_t | \{\mathbf{y}\}_t^t, m)$ and likelihood terms $P(\{\mathbf{y}\}_t^t | m)$ in Eq. 2 can be computed or approximated using any of a number of different recursive Bayesian decoding techniques, including Bayes' filter (Brown et al. 1998), particle filters (Brockwell et al. 2004; Shoham et al. 2005), and Kalman filter variants (Wu et al. 2004, 2006). If available, prior information about the identity of the movement regime can be incorporated naturally into the MTM framework using $P(m)$ in Eq. 2. This information must be available before the reach begins and may differ from trial to trial. If no such information is available, the same $P(m)$ (e.g., a uniform distribution) can be used across all trials.

The computational complexity of the MTM decoder is M times that of computing $P(\mathbf{x}_t | \{\mathbf{y}\}_t^t, m)$ and $P(\{\mathbf{y}\}_t^t | m)$ for a particular mixture component m . Because the computations for each mixture component can theoretically be carried out in parallel, it is possible to set up the MTM decoder so that its running time remains constant, regardless of the number of mixture components M . In other words, the MTM approach enables the use of more flexible—and potentially more accurate—trajectory models without a necessary penalty in decoder running time. Furthermore, the MTM decoder preserves the real-time properties of its constituent estimators and is thus suitable for real-time prosthetic applications.

MIXTURE OF TRAJECTORY MODELS FOR GOAL-DIRECTED REACHES. The particular probabilistic model explored in this work is

$$\mathbf{x}_t | \mathbf{x}_{t-1}, m \sim \mathcal{N}(A_m \mathbf{x}_{t-1} + \mathbf{b}_m, Q_m) \tag{3}$$

$$\mathbf{x}_1 | m \sim \mathcal{N}(\boldsymbol{\pi}_m, V_m) \tag{4}$$

$$s_{t-\text{lag}_i}^i | \mathbf{x}_t \sim \text{Poisson}[\exp(\mathbf{c}_i^T \mathbf{x}_t + d_i) \Delta] \tag{5}$$

where $m \in \{1, \dots, M\}$ indexes the reach goal and M is the number of reach goals. The dynamical arm state at time step $t \in \{1, \dots, T\}$ is $\mathbf{x}_t \in \mathbb{R}^{p \times 1}$, which includes position, velocity, and acceleration terms, as specified in the APPENDIX. The corresponding observation, $s_{t-\text{lag}_i}^i \in \{0, 1, 2, \dots\}$, is a peri-movement spike count for unit $i \in \{1, \dots, q\}$ taken in a time bin of width Δ , where lag_i is the time lag (in time steps) between the neural firing of the i th unit and the associated arm state. For notational convenience, the spike counts across the q simultaneously recorded units are assembled into a $q \times 1$ vector \mathbf{y}_t , whose i th element is $s_{t-\text{lag}_i}^i$. This is the \mathbf{y}_t that appears in Eqs. 1 and 2. The parameters $A_m \in \mathbb{R}^{p \times p}$, $\mathbf{b}_m \in \mathbb{R}^{p \times 1}$, $Q_m \in \mathbb{R}^{p \times p}$, $\boldsymbol{\pi}_m \in \mathbb{R}^{p \times 1}$, $V_m \in \mathbb{R}^{p \times p}$, $\text{lag}_i \in \mathbb{Z}$, $\mathbf{c}_i \in \mathbb{R}^{p \times 1}$, $d_i \in \mathbb{R}$ do not depend on time and are fit to training data, as subsequently described.

Equations 3 and 4 define the *trajectory model*, which describes how the arm state \mathbf{x}_t changes from one time step to the next. In this case, the full trajectory model is a mixture of linear-Gaussian trajectory models, each describing the trajectories toward a particular reach goal indexed by m . By this definition, each movement regime corresponds to movements heading toward a particular reach goal. Conditioned on the reach goal m , the trajectory model is a linear-Gaussian dynamical model.¹ Although the MTM framework will be illustrated in this work using Eqs. 3 and 4, mixtures of other trajectory models can also be used. For example, it is possible to define, for each reach goal, a linear-Gaussian model with a time-varying forcing term (Kemere and Meng 2005; Srinivasan et al. 2005, 2006) or a canonical trajectory (Kemere et al. 2002, 2004b).

Equation 5 defines the *observation model*, which describes how the observed peri-movement spike counts $s_{t-\text{lag}_i}^i$ relate to the arm state \mathbf{x}_t .

¹ The family of linear-Gaussian models includes the dynamic model defined by Eqs. 3 and 4 (conditioned on m) as well as numerous variants, including the random-walk model (Brockwell et al. 2004; Brown et al. 1998), those without a forcing term \mathbf{b}_m (Wu et al. 2004, 2006), those with a time-varying forcing term (Kemere and Meng 2005; Srinivasan et al. 2005, 2006), and those with higher-order Markov dependencies (Shoham et al. 2005). In the rest of this paper, we will refer to Eqs. 3 and 4 as a “linear-Gaussian model,” meaning that it is part of the linear-Gaussian family.

In Eq. 5, the linear mapping $\mathbf{c}_i^T \mathbf{x}_t + d_i$ is a cosine tuning model (Georgopoulos et al. 1982), where \mathbf{c}_i is the “preferred state vector.” This linear mapping is then passed through an exponential to ensure that the mean firing rate of the i th unit at time $t - \text{lag}_i$, $\exp(\mathbf{c}_i^T \mathbf{x}_t + d_i)$, is non-negative. Note that, whereas each mixture component indexed by m in the trajectory model (Eqs. 3 and 4) can have different parameters leading to different arm state dynamics, the observation model (Eq. 5) is the same for all m .

Although the neural activity is known to be physically driving the trajectories, the probabilistic model Eqs. 3–5 specifies that the neural activity \mathbf{y}_t is dependent on the arm state \mathbf{x}_t . This model incorrectly implies, for example, that noise arising from the mechanical properties of the muscles that corrupts the arm trajectory should also show up in the neural activity in motor cortical areas. Nevertheless, models with this “inverted” structure have been shown to effectively decode arm trajectories (Brockwell et al. 2004; Shoham et al. 2005; Wu et al. 2004, 2006). The primary motivation for adopting such a structure is that there are established techniques for efficiently estimating an unobserved time series with known dynamics (in this case, the arm trajectory) from noisy observations (in this case, the neural spike counts). These techniques are detailed in the next section.

RECURSIVE BAYESIAN DECODING. Arm trajectories can be decoded from neural activity by applying Bayes’ rule to the statistical relationships of Eqs. 3–5. Having observed the neural data, we seek the likely sequences of arm states that could have led to those neural observations. For each m and t , we need to compute the following two terms that appear in Eq. 2: the conditional state posteriors $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$ and the likelihood terms $P(\{\mathbf{y}\}_1^t | m)$.

The conditional state posteriors can be obtained by iterating the following two updates. First, the one-step prediction is found by applying Eq. 3 to the conditional state posterior at the previous time step

$$P(\mathbf{x}_t | \{\mathbf{y}\}_1^{t-1}, m) = \int P(\mathbf{x}_t | \mathbf{x}_{t-1}, m) P(\mathbf{x}_{t-1} | \{\mathbf{y}\}_1^{t-1}, m) d\mathbf{x}_{t-1} \quad (6)$$

Second, the conditional state posterior at the current time step is computed using Bayes’ rule

$$P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m) = \frac{P(\mathbf{y}_t | \mathbf{x}_t) P(\mathbf{x}_t | \{\mathbf{y}\}_1^{t-1}, m)}{P(\mathbf{y}_t | \{\mathbf{y}\}_1^t, m)} \quad (7)$$

Note that $P(\mathbf{y}_t | \mathbf{x}_t, \{\mathbf{y}\}_1^{t-1}, m)$ has been replaced by $P(\mathbf{y}_t | \mathbf{x}_t)$ to obtain Eq. 7 because, given the current arm state \mathbf{x}_t , the current observation \mathbf{y}_t does not depend on the previous observations $\{\mathbf{y}\}_1^{t-1}$ nor the reach goal m (cf. Eq. 5). The terms in the numerator of Eq. 7 are the observation model from Eq. 5 and the one-step prediction from Eq. 6. The denominator of Eq. 7 can be obtained by integrating the numerator over \mathbf{x}_t , as shown later in Eq. 9.

When the trajectory and observation models are both linear-Gaussian, all of the relevant distributions are Gaussian and the integral in Eq. 6 can be computed exactly. Taking the iterations defined by Eqs. 6 and 7 is identical to applying the standard Kalman filter.

However, the particular observation model here (Eq. 5) is not linear-Gaussian. This leads to distributions that are difficult to manipulate and the integral in Eq. 6 cannot be computed analytically. We instead use a modified Kalman filter that uses a Gaussian approximation during the measurement update step (Eq. 7). We approximate the conditional state posterior as a Gaussian matched to the location and curvature of the mode of $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$, as detailed in the APPENDIX. This Gaussian approximation then allows the integral in Eq. 6 to be computed analytically because each mixture component of the full trajectory model (Eq. 3) is linear-Gaussian. This yields a Gaussian one-step prediction, which is fed back into Eq. 7.

The likelihood terms $P(\{\mathbf{y}\}_1^t | m)$ can be expressed as

$$P(\{\mathbf{y}\}_1^t | m) = \prod_{\tau=1}^t P(\mathbf{y}_\tau | \{\mathbf{y}\}_1^{\tau-1}, m) \quad (8)$$

where

$$P(\mathbf{y}_t | \{\mathbf{y}\}_1^{t-1}, m) = \int P(\mathbf{y}_t | \mathbf{x}_t) P(\mathbf{x}_t | \{\mathbf{y}\}_1^{t-1}, m) d\mathbf{x}_t \quad (9)$$

The integral in Eq. 9, which cannot be computed analytically, is approximated using Laplace’s method (MacKay 2003). Note that this involves the same Gaussian approximation in \mathbf{x}_t (i.e., the same mean and covariance) as made above for $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$.

EVALUATING PERFORMANCE. The state posterior $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t)$ in Eq. 1 is a multimodal distribution. To compare the performance of different decoders and to control a prosthetic cursor or arm, we need to collapse this multimodal distribution into a single decoded trajectory. In other words, we need to summarize the belief embodied in the state posterior with a single value $\hat{\mathbf{x}}_t$ at each time point. This can be done by first defining a loss function L , which specifies the loss incurred by the summary $\hat{\mathbf{x}}_t$ for each possible value of \mathbf{x}_t . The single decoded trajectory is then the $\hat{\mathbf{x}}_t$ that minimizes the average loss under the state posterior

$$\text{Average loss}(\hat{\mathbf{x}}_t) = \int L(\mathbf{x}_t, \hat{\mathbf{x}}_t) P(\mathbf{x}_t | \{\mathbf{y}\}_1^t) d\mathbf{x}_t \quad (10)$$

Here, we choose to use the instantaneous sum of squared distance loss function

$$L(\mathbf{x}_t, \hat{\mathbf{x}}_t) = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 \quad (11)$$

in which case the $\hat{\mathbf{x}}_t$ that minimizes the average loss (Eq. 10) is the mean of the state posterior

$$\hat{\mathbf{x}}_t = E[\mathbf{x}_t | \{\mathbf{y}\}_1^t] \quad (12)$$

In particular, the mean of the state elements corresponding to arm position is taken to be the decoded position trajectory. To compare different decoders, we first compute the root-mean-square position error (E_{rms}) between the actual and decoded trajectories on a per-trajectory basis. This yields a distribution of E_{rms} values for a given decoder. The E_{rms} distribution of different decoders can then be compared and statistics of each distribution (such as mean and SE) can be computed.

The expectation in Eq. 12 can be expanded by conditioning on the reach goal m as in Eq. 1, yielding

$$\hat{\mathbf{x}}_t = \sum_{m=1}^M E[\mathbf{x}_t | \{\mathbf{y}\}_1^t, m] P(m | \{\mathbf{y}\}_1^t) \quad (13)$$

The interpretation of Eq. 13 is similar to that of Eq. 1. If the desired reach goal m^* is perfectly known before the reach begins, the decoded trajectory ($\hat{\mathbf{x}}_t$) is computed based only on the individual trajectory model (i.e., the mixture component) corresponding to that reach goal. The decoded trajectory, in this case, is simply the mean of the conditional state posterior corresponding to the known reach goal, written $E[\mathbf{x}_t | \{\mathbf{y}\}_1^t, m^*]$ and termed the *component trajectory estimate* for m^* . However, in general, the desired reach goal is unknown or imperfectly known, so we need to compute a component trajectory estimate $E[\mathbf{x}_t | \{\mathbf{y}\}_1^t, m]$ for each $m \in \{1, \dots, M\}$. The final decoded trajectory ($\hat{\mathbf{x}}_t$) is a weighted sum of these component trajectory estimates, as shown in Eq. 13. As in Eq. 1, the weights $P(m | \{\mathbf{y}\}_1^t)$ represent the probability that the desired reach goal is m , given the observed spike counts up to time t .

In this work, we compare the performance of four decoders. The first is a state-of-the-art decoder presented by Kass and colleagues

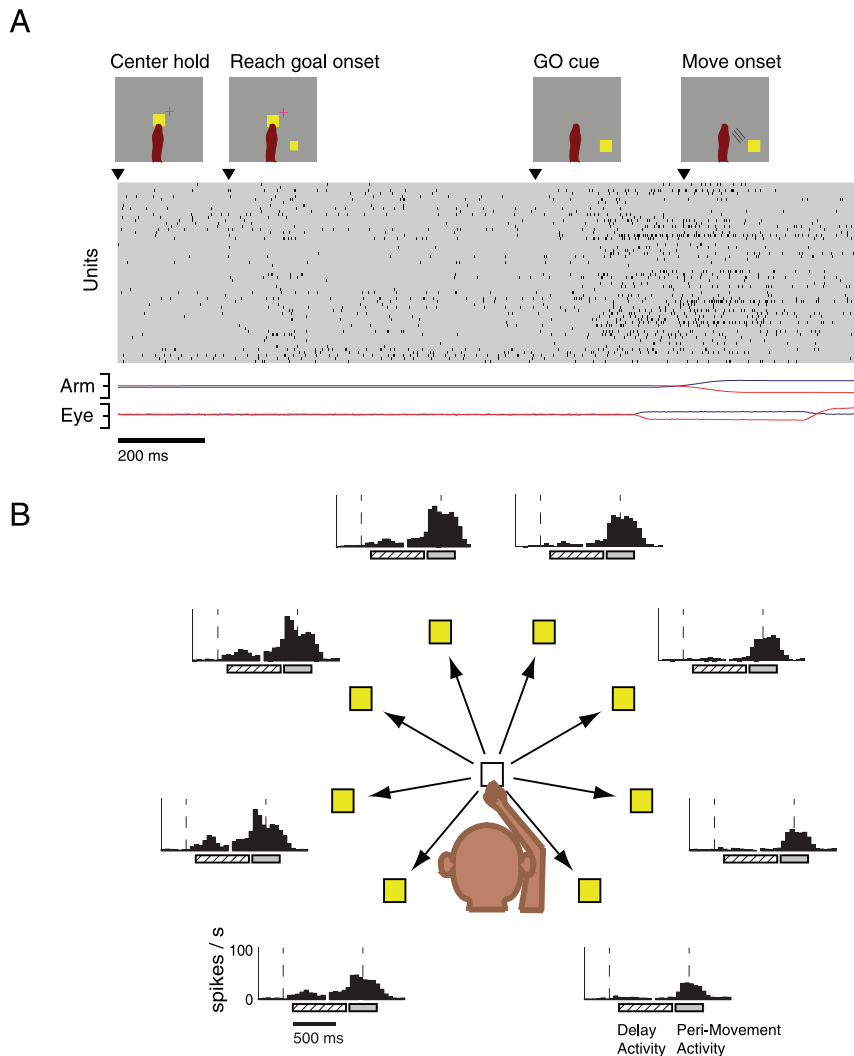


FIG. 1. Delayed reach task and neural recordings. *A*: task timeline (*top*), simultaneously recorded spike trains (*middle*), and arm and eye position traces (*bottom*) are shown for a single trial. Blue and red lines correspond to horizontal and vertical position, respectively. Full range of scale for the arm and eye position is ± 15 cm from the center target. Trial from experiment H20041106.1. *B*: spatial arrangement of the 8 reach goals and corresponding spike histograms for one representative unit (Unit H20041217.23.0). Bars below histograms indicate delay (hatched) and peri-movement (gray) activity. Dotted lines denote reach goal onset and movement onset.

(Brockwell et al. 2004) based on a random-walk trajectory model (RWM) in acceleration. The trajectory (Eqs. A7 and A8) and observation (Eq. A9) models are defined in the APPENDIX. The second decoder is based on a single linear-Gaussian trajectory model (STM) shared across reaches to all goals. It is defined by Eqs. 3 and 4 for the special case of $M = 1$. The STM decoder uses the observation model shown in Eq. 5. The RWM and STM decoders provide points of comparison for the following two MTM decoders, both of which are based on Eqs. 3–5. Whereas the MTM_M decoder uses only peri-movement activity, the MTM_{DM} decoder uses both delay and peri-movement activity. In Eq. 2, the same $P(m)$ (in this case, a uniform distribution) is used across all trials for MTM_M . In contrast, a different $P(m)$ is used on each trial for MTM_{DM} based on the prior goal information extracted from delay activity.

Goal-directed reach task and neural recordings

Animal protocols were approved by the Stanford University Institutional Animal Care and Use Committee. We trained two adult male monkeys (*Macaca mulatta*, monkeys G and H) to perform delayed center-out reaches for juice rewards. As illustrated in Fig. 1A, visual targets were back-projected onto a frontoparallel screen 30 cm in front of the monkey. The monkey touched a central target and fixated his eyes on a crosshair at the upper right corner of the central target. After a center hold period of 500 (monkey G) or 400–600 ms (monkey H), selected randomly and uniformly in this range), a pseudorandomly

chosen reach goal was presented at one of eight possible radial locations ($30, 70, 110, 150, 190, 230, 310, 350^\circ$)² 10 cm away. After a pseudorandomly chosen instructed delay period of 200, 750, or 1,000 ms, the “go” cue (signaled by both the enlargement of the reach goal and the disappearance of the central target) was given and the monkey reached to the goal. After a hold time of 250 (monkey G) or 200 ms (monkey H) at the reach goal, the monkey received a liquid reward. The next trial started 200 (monkey G) or 100 ms (monkey H) later.

Eye fixation at the crosshair was enforced throughout the delay period. Reaction times (defined as the time between the “go” cue and movement onset) were enforced to be >80 ms and <600 (monkey G) or <400 ms (monkey H). The following are the statistics for the actual reaction times (mean \pm SD in milliseconds): 237 ± 23 for monkey G and 248 ± 22 for monkey H. The trials with 200-ms delay periods were used as catch trials to encourage the monkey to “plan” throughout the delay period. Without these 200-ms delays, the monkeys could learn that it is not necessary to plan during the first few hundred milliseconds of the delay period. These catch trials were not used in subsequent analyses.

During experiments, monkeys sat in a custom chair (Crist Instruments, Hagerstown, MD) with the head braced and the nonreaching

²Reach goals were not presented directly below (230 – 310°) the central target because they would be occluded by the monkey’s hand while he is touching the central target.

arm strapped to the chair. The presentation of the visual targets was controlled using the Tempo software package (Reflective Computing, St. Louis, MO). A custom photodetector recorded the timing of the video frames with 1-ms resolution. The position of the hand was measured in three dimensions using the Polaris optical tracking system (Northern Digital, Waterloo, Ontario, Canada; 60 Hz, 0.35-mm accuracy), whereby a passive marker taped to the monkey's fingertip reflected infrared light back to the position sensor. Eye position was tracked using an overhead infrared camera (Iscan, Burlington, MA; 240 Hz, estimated accuracy of 1°).

A 96-channel silicon electrode array (Cyberkinetics, Foxborough, MA) was implanted straddling dorsal premotor (PMd) and motor (M1) cortex (monkey G, right hemisphere; monkey H, left hemisphere), as estimated visually from local landmarks, contralateral to the reaching arm. Surgical procedures were described previously (Churchland et al. 2006c; Santhanam et al. 2006). Spike sorting was performed off-line using techniques described in detail elsewhere (Sahani 1999; Santhanam et al. 2004; Zumsteg et al. 2005). Briefly, neural signals were monitored on each channel during a 2-min period at the start of each recording session while the monkey performed the behavioral task. Data were high-pass filtered and a threshold level of three times the RMS voltage was established for each channel. The portions of the signals that did not exceed threshold were used to characterize the noise on each channel. During experiments, snippets of the voltage waveform containing threshold crossings (0.3 ms precrossing to 1.3 ms postcrossing) were saved with 30-kHz sampling. After each experiment, the snippets were clustered as follows. First, they were noise-whitened using the noise estimate made at the start of the experiment. Second, the snippets were trough-aligned and projected into a four-dimensional space using a modified principal components analysis. Next, unsupervised techniques determined the optimal number and locations of the clusters in the principal components space. We then visually inspected each cluster, along with the distribution of waveforms assigned to it, and assigned a score based on how well separated it was from the other clusters. This score determined whether a cluster was labeled a single-neuron unit or a multineuron unit.

Figure 1A shows the delayed reach task timeline, along with neural and behavioral data for a single trial with a lower-right reach goal. We later refer to the time between reach goal onset and the "go" cue as the delay period. Figure 1B illustrates the spatial arrangement of the eight reach goals, as well as the corresponding spike histograms for one representative unit across the eight reach goals. Each spike histogram was obtained by averaging the spike trains across multiple trials with the same reach goal. In broad terms, delay activity occurs during the delay period (always before movement onset), whereas peri-movement activity occurs around the time of the reach. The precise windows of delay and peri-movement activity used in this work are defined in later sections.

The monkeys were trained over several months and multiple data sets of the same behavioral task were collected. Each data set was collected in one day's recording session. For each monkey, we chose to analyze a data set with a large number of successful trials. The selected data sets consisted of 1,456 successful trials for monkey G (experiment G20040508) and 1,072 successful trials for monkey H (experiment H20041217), not including trials with 200-ms delay periods. The data set for monkey G (H) included 30 (56) single-neuron units and 68 (143) multineuron units, for a total of 98 (199) units.

The results reported in this work are cross-validated by randomly splitting the entire data set by trials into J roughly equal-sized parts. For each $j \in \{1, \dots, J\}$, the j th part served as test data for a model trained on the other $J - 1$ parts. We used $J = 9$ (11) for the data set for monkey G (H). To evaluate decoder performance at different numbers of neural units, we further randomly split each part by units into K equal subparts. Each subpart contained the same number of trials and identical behavioral data as its parent, but with only $1/K$ of

the neural data. To meaningfully compare the two data sets, we roughly equalized the number of units in each subpart. Unless otherwise specified, the results presented here assume $K = 1$ (98 units) for monkey G and $K = 2$ (99 units) for monkey H.

Incorporating goal information from delay activity

Up to this point, the neural activity discussed has been peri-movement activity, which takes place around the time of movement and specifies the moment-by-moment details of the arm trajectory. In the delayed-reach task, there is also neural activity present during an instructed delay period that directly precedes the "go" cue (termed *delay activity*). As shown in Crammond and Kalaska (2000) and Churchland et al. (2006c), neurons with delay activity are typically also active in the absence of an instructed delay during the reaction time period. Rather than specifying the moment-by-moment details of the trajectory, delay activity has been shown to reliably indicate the upcoming reach goal (Hatsopoulos et al. 2004; Musallam et al. 2004; Santhanam et al. 2006; Shenoy et al. 2003; Yu et al. 2004). The data sets for both monkeys G and H contain both delay and peri-movement activity on each trial. Furthermore, both types of activity may be emitted by the same unit on a single trial, as can be seen in Fig. 1.

The following describes how the reach goal can be decoded from delay activity by applying Bayes' rule. Let \mathbf{z} be a $q \times 1$ vector of spike counts across the q simultaneously recorded units in a prespecified time window during the delay period on a single trial. The distribution of spike counts (from training data) for each reach goal m can be fit to either a product of Gaussians (Maynard et al. 1999; Yu et al. 2004)

$$\mathbf{z}|m \sim \prod_{i=1}^q \mathcal{N}(z_i; \mu_{i,m}, \sigma_{i,m}^2) \tag{14}$$

or a product of Poissons (Hatsopoulos et al. 2004; Shenoy et al. 2003)

$$\mathbf{z}|m \sim \prod_{i=1}^q \text{Poisson}(z_i; \lambda_{i,m}) \tag{15}$$

where $\mu_{i,m}$, $\sigma_{i,m}^2$, and $\lambda_{i,m}$ are the parameters of the i th unit for the m th reach goal. The z_i notation in Eqs. 14 and 15 specifies that the distribution is describing the i th element of the vector \mathbf{z} . In both models, the units are assumed to be independent given the reach goal. It would be natural to introduce conditional dependencies between the units using a general multivariate Gaussian, but there are often difficulties in estimating an invertible covariance matrix for tens to hundreds of units with a limited number of training trials (Maynard et al. 1999).

For any test trial, the probability that the upcoming reach goal is m given the delay activity \mathbf{z} can be computed by applying Bayes' rule

$$P(m|\mathbf{z}) = \frac{P(\mathbf{z}|m)P(m)}{P(\mathbf{z})} = \frac{P(\mathbf{z}|m)}{\sum_{m'} P(\mathbf{z}|m')} \tag{16}$$

where $P(m)$ in Eq. 16 is assumed to be uniform. The most likely reach goal [i.e., the one with the largest $P(m|\mathbf{z})$] is usually taken to be the decoded reach goal (Hatsopoulos et al. 2004; Musallam et al. 2004; Santhanam et al. 2006; Shenoy et al. 2003; Yu et al. 2004).

The accuracy of the goal decoder (Eq. 16) varies with the duration and placement of the time window in which spikes are counted, as well as the precise spike count model $P(\mathbf{z}|m)$ that is used (Hatsopoulos et al. 2004; Santhanam et al. 2006). Optimizing the goal decoder is beyond the scope of this work and is treated in detail in the aforementioned references. Instead, we focus here on how to incorporate this goal information, if available, when decoding continuous arm trajectories. For this purpose, we choose to use the Gaussian model (Eq. 14) with a 200-ms spike count window starting 150 ms after the appearance of the reach goal.

The goal information from delay activity, $P(m | \mathbf{z})$, can be incorporated naturally in the MTM framework in the place of $P(m)$ in Eq. 2. The distribution $P(m)$ in Eq. 2 represents the prior knowledge (i.e., before movement onset) that the upcoming reach goal is m . Because the delay activity entirely precedes movement onset and provides information about the upcoming reach goal, it can be used to set $P(m)$ in Eq. 2 on a per-trial basis.

It is important to note that the most likely goal from Eq. 16 is not simply assumed here to be the goal of the upcoming reach. On a given trial, the delay activity may not definitively indicate the goal of the upcoming reach (e.g., two different reach goals may have significant probability) or it may indicate an incorrect goal for the upcoming reach. In this case, we would like to allow the subsequent peri-movement activity to determine the goal of the reach, or even correct the mistake, “in-flight.” Instead of making a hard goal decode based on delay activity, the entire distribution $P(m | \mathbf{z})$ is retained and passed to the MTM framework. For simplicity, we make the approximation that delay activity is informative only of the upcoming reach goal and is independent of the peri-movement activity; in other words, we assume that \mathbf{z} is not directly coupled with \mathbf{x} , or \mathbf{y}_r .

RESULTS

In this section, we evaluate and compare the performance of decoding goal-directed movements using the RWM, STM, MTM_M , and MTM_{DM} decoders. For all decoders, we first fit the model parameters to training data, as detailed in the APPENDIX. The test data for a single trial consisted of 1) the arm trajectory, taken from 50 ms before movement onset to 50 ms after movement end at $dt = 10$ -ms time steps; 2) the peri-movement spike counts, taken in nonoverlapping $\Delta = 10$ -ms bins and temporally offset from the arm trajectory by the optimal lag found for each unit; and 3) the delay-period spike counts, taken in a single 200-ms bin starting 150 ms after the appearance of the reach goal. Arm trajectories in the test phase were used to evaluate the accuracy of the trajectories estimated from neural data. Because neural data collection ended shortly after movement end, the arm trajectories were not padded as in the training phase.

Figure 2 details, for a particular test trial (monkey G, 98 units), how the MTM decoded trajectory was obtained and

compares the trajectory estimates produced by the different decoders. From Eq. 13, the MTM decoded trajectory is a weighted sum of component trajectory estimates $E[\mathbf{x}_t | \{\mathbf{y}\}_t^t, m]$, one for each reach goal indexed by $m \in \{1, \dots, 8\}$. In Fig. 2, *B* and *C*, the component trajectory estimates are plotted in the *top panels*, whereas the *middle panels* show how the corresponding weights $P(m | \{\mathbf{y}\}_t^t)$ evolved during the course of the trial.

The values of the weights at time zero ($t = 0$) represent the probability that the upcoming reach goal is m , before any peri-movement neural activity is observed. The distribution of weights at $t = 0$ is precisely $P(m)$ in Eq. 2. In Fig. 2*B*, we assumed that there was no information available about the identity of the upcoming reach goal before the reach began (i.e., no delay activity), so all eight goals were equiprobable [i.e., $P(m) = 1/8$ for $m \in \{1, \dots, 8\}$]. As time proceeded, these weights were updated as more and more peri-movement activity was observed. Recall that $P(m | \{\mathbf{y}\}_t^t)$ represents the probability that the actual reach goal is m , given the observed neural activity up to time t . During the first 200 ms, the actual reach goal (cyan) was more likely than the other seven reach goals at nearly every time step; however, there was some competition with the neighboring reach goals (blue and magenta). It was only after about 200 ms that the decoder became certain of the actual reach goal [i.e., $P(m | \{\mathbf{y}\}_t^t)$ approached one] and remained certain for the rest of the trial. A weighted sum of the eight component trajectory estimates (*top*) using these weights (*middle*) yields the MTM decoded trajectory (*top*, red; E_{rms} : 10.9 mm).

If delay activity is available, it can be used to set a nonuniform $P(m)$ in Eq. 2 on a per-trial basis, as previously discussed. The only difference between *B* and *C* in Fig. 2 is that the MTM decoder used delay activity in the latter, but not the former. In Fig. 2*C* (*middle*), the weights at $t = 0$ represent the probabilities of each reach goal based only on delay activity, before any peri-movement activity was observed. In this case, the delay activity indicated that the actual reach goal (cyan) was more probable than the other goals. This prior knowledge of the identity of the upcoming reach goal was then taken into

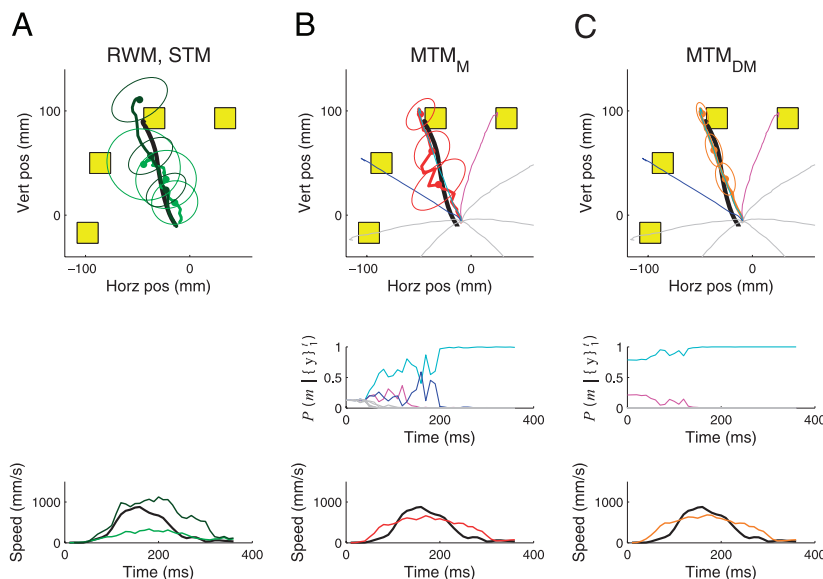


FIG. 2. A representative test trial in which the use of delay activity improved the mixture of trajectory models (MTM) decoded trajectory. *Top panels* compare the actual trajectory (all panels, black) with the decoded trajectories for the (A) random-walk trajectory model (RWM, dark green) and single linear-Gaussian trajectory model (STM, light green); and a mixture of linear-Gaussian trajectory models, (B) one that uses only perimovement activity (MTM_M , red), and (C) one that uses both delay and perimovement activity (MTM_{DM} , orange). Ellipses denote 95% confidence intervals at 3 different time steps. Yellow squares represent the visual reach goals presented to the monkey in actual dimensions. *Top panels* in *B* and *C* also show the 8 component trajectory estimates for the MTM (cyan, blue, and magenta for the 3 components with the largest weights; gray for the other 5 components). Their corresponding weights, as they evolve during the trial, are plotted in the *middle panels*. *Bottom panels* compare the actual and estimated single-trial speed profiles using the same color conventions as in the *top panels*. Time zero corresponds to 60 ms before movement onset (i.e., one time step before we begin to decode movement). Note that the red and orange traces in the *top panels* are overlaid with the cyan trace. For this trial, root-mean-square position error (E_{rms}) was 11.8, 26.9, 10.9, and 10.5 mm for the RWM, STM, MTM_M , and MTM_{DM} , respectively. Monkey G, 98 units (Experiment G20040508, trial ID 474).

account when updating the weights $P(m | \{y\}_t^t)$ during the course of the trial as more and more peri-movement activity was observed. Note that using delay activity affected only $P(m)$ in Eq. 2; the conditional state posteriors $P(\mathbf{x}_t | \{y\}_t^t, m)$ and the likelihood terms $P(\{y\}_t^t | m)$ remained unchanged. As the means of the conditional state posteriors, the component trajectory estimates therefore also remained unchanged, as can be verified by comparing Fig. 2, *B* and *C* (*top*). For the trial shown in Fig. 2, the use of delay activity reduced the competition between the actual reach goal (cyan) and the neighboring goals (blue and magenta). Compared with Fig. 2*B* (*middle*), the weight for the actual reach goal (cyan) in Fig. 2*C* (*middle*) was higher at every time point, the clearest effect seen during the first 200 ms. In other words, by using delay activity, the decoder was more certain of the actual reach goal throughout the trial. In Fig. 2*C*, a weighted sum of the eight component trajectory estimates (*top*) using these weights (*middle*) yields the MTM decoded trajectory (*top*, orange; E_{rms} : 10.5 mm).

By comparing the MTM decoded trajectories with the actual trajectory in Fig. 2, *B* and *C* (*top*), we see that the use of delay activity decreased the decoding error and tightened the confidence ellipses for this trial. The derivation of the MTM confidence intervals are given in the APPENDIX. Both MTM decoded trajectories had lower decoding error than the RWM (E_{rms} : 11.8 mm) and STM (E_{rms} : 26.9 mm), whose decoded trajectories are plotted in Fig. 2*A* (*top*). On this trial, the RWM decoder produced a reasonably accurate decoded trajectory, whereas the STM decoded trajectory proceeded slowly outward with wide confidence intervals.

These decoders can also be used to estimate the bell-shaped speed profile of the actual reach (Fig. 2, *bottom*). For the STM and MTM, we computed speed using its exact nonlinear relationship with the velocity elements in the state vector, rather than directly taking the speed element in the state vector, which involves a linear approximation. Compared with the speed profiles estimated by the RWM (dark green) and STM (light green) decoders, those estimated by the MTM decoders (red and orange) seem to better track the actual bell-shaped speed profile (black); this observation is shown even more clearly in the subsequent example trials.

In contrast to Fig. 2, Fig. 3 shows a trial (monkey G, 98 units) where the peri-movement activity alone was able to quickly determine the actual reach goal without much competition from neighboring goals. This can be seen in Fig. 3*B* (*middle*), where the weight corresponding to the actual reach goal (cyan) rose to unity after about 100 ms and stayed there for the remainder of the trial. As a result, the resulting MTM decoded trajectory (*top*, red; E_{rms} : 11.2 mm) was quite accurate. As in Fig. 2, we can incorporate delay activity if available; however, in this case, the dominant weight at $t = 0$ (blue) did not correspond to the actual reach goal (cyan), as seen in Fig. 3*C* (*middle*). In other words, the delay activity incorrectly indicated the identity of the upcoming reach goal. However, as these weights were updated by the observation of peri-movement activity, this “error” was soon corrected (within about 100 ms). From that point on, the weight corresponding to the actual reach goal dominated. Despite this error at the beginning of the trial, the MTM decoded trajectory in Fig. 3*C* (*top*, orange; E_{rms} : 13.0 mm) still headed to the correct goal and provided a reasonably accurate estimate of the arm trajectory. The larger confidence ellipses for MTM_{DM} compared with MTM_M reflect the competition between the actual (cyan) and neighboring (blue) reach goals. The decoded trajectories for the RWM (dark green) and STM (light green) are shown in Fig. 3*A* for comparison. As in Fig. 2, both MTM decoded trajectories yielded lower decoding error than the RWM (E_{rms} : 27.7 mm) and STM (E_{rms} : 24.2 mm). Furthermore, as shown in the *bottom panels*, the speed profiles estimated by the MTM decoders (red and orange) tracked the actual bell-shaped speed profile (black) more closely than those estimated by the RWM (dark green) and STM (light green) decoders.

Figures 2 and 3 together illustrate the benefits of the joint use of peri-movement and delay activity. When one type of activity is unable to definitively identify (or incorrectly identifies) the actual reach goal, the MTM framework allows the other type of activity to strengthen (or overturn) the goal identification in a probabilistic manner. In Fig. 2, the peri-movement activity alone was unable to definitively identify the actual reach goal during the first 200 ms because there was competition with a neighboring goal. When prior goal information from delay

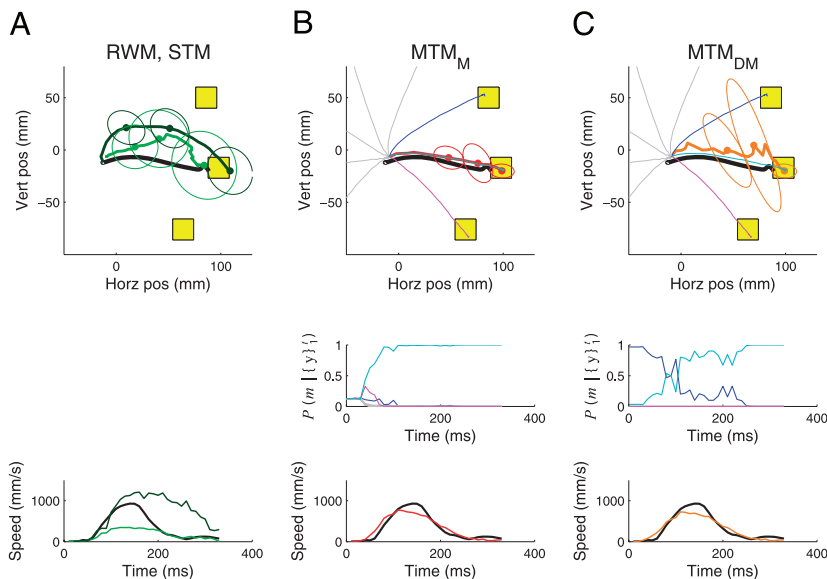


FIG. 3. Representative test trial in which the peri-movement activity corrected an incorrect goal identification from delay activity. Figure conventions are identical to those in Fig. 2. For this trial, E_{rms} was 27.7, 24.2, 11.2, and 13.0 mm for RWM, STM, MTM_M, and MTM_{DM}, respectively. Monkey G, 98 units (Experiment G20040508, trial ID 676).

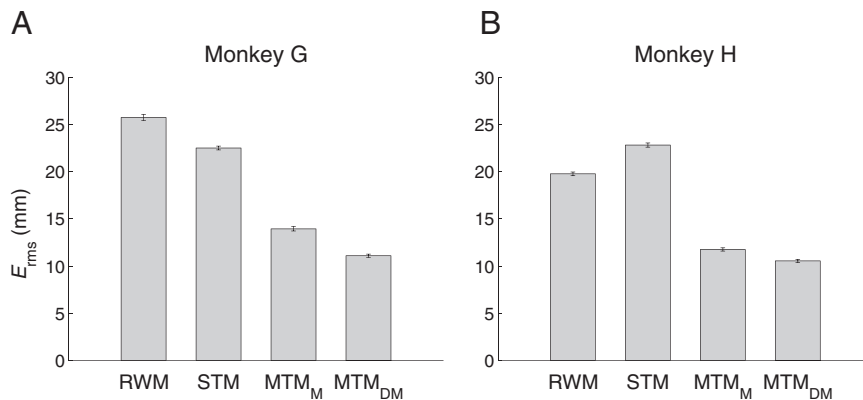


FIG. 4. E_{rms} (mean \pm SE) comparison for the RWM, STM, MTM_M , and MTM_{DM} decoders. A: monkey G (98 units). B: monkey H (99 units).

activity was incorporated, the decoder was more certain of the actual reach goal throughout the trial. In Fig. 3, the delay activity incorrectly indicated the identity of the upcoming reach goal. However, the peri-movement activity overturned this incorrect goal identification early on and rescued the decoder from incurring a large E_{rms} on this trial.

Having demonstrated how the MTM framework produces trajectory estimates on individual trials, we can quantify and compare the average performance of the various decoders (RWM, STM, MTM_M , MTM_{DM}) across entire data sets. Figure 4 illustrates the following two main results, which hold true across both monkeys. First, a mixture of linear-Gaussian trajectory models (MTM_M) provides lower decoding error than either of the nonmixture trajectory models (RWM and STM) (Wilcoxon paired-sample test, $P < 0.01$). Compared with the STM decoder, the MTM_M decoder reduced E_{rms} from 22.5 to 13.9 mm (22.8 to 11.8 mm) in monkey G (H). Second, the use of prior goal information $P(m)$ in the MTM framework (MTM_{DM}) can further decrease decoding error (Wilcoxon paired-sample test, $P < 0.01$). Compared with the MTM_M decoder, the MTM_{DM} decoder reduced E_{rms} from 13.9 to 11.1 mm (11.8 to 10.5 mm) in monkey G (H). Because the MTM decoder is inherently parallelizable (as described in METHODS), these performance gains can be obtained *without* an associated increase in decoder running time. The superior performance of the MTM_M compared with the RWM and STM can be explained by the fact that the MTM better captures the kinematics of goal-directed reaches. This can be seen in both the generative (prior) speed profiles (Fig. A1, *bottom panels*), as well as the decoded (posterior) speed profiles (Figs. 2 and 3, *bottom panels*). If delay activity is available, this additional source of

information can be naturally incorporated in the MTM framework to further improve decoding performance (MTM_{DM}). The relative performance of the RWM and STM decoders is subsequently addressed in the context of Fig. 8.

To compare decoders on a *trial-by-trial* basis, we constructed two-dimensional histograms of E_{rms} differences between pairs of decoders, shown in Fig. 5. The MTM_M performed better than the STM for any trial lying to the left of the vertical zero axis, whereas the MTM_{DM} performed better than the STM for any trial lying below the horizontal zero axis. We can also directly compare the MTM_M and MTM_{DM} using this two-dimensional histogram. By construction of the histogram, the MTM_{DM} performed better than the MTM_M for any trial lying below the diagonal axis. For both monkeys, all three mean differences (dotted lines) differ from zero (Wilcoxon paired-sample test, $P < 0.01$). The values of these means show that, on average, the MTM_M performed better than the STM, the MTM_{DM} performed better than the STM, and the MTM_{DM} performed better than the MTM_M . The same mean differences can be obtained by taking pairwise differences in bar heights in Fig. 4.

The letters **a** and **b** in Fig. 5A indicate where the trials shown in Figs. 2 and 3 lie on the histogram. Both trials are taken from the dominant central region of the histogram and are thus considered to be representative trials. However, there are also outlying trials for which the STM performed better than the MTM_M and/or the MTM_{DM} . We consider two of these trials (labeled **c** and **d** in Fig. 5A) in detail in Figs. 6 and 7.

Figure 6 shows an outlying test trial (monkey G, 98 units) for which the MTM_M performed worse than the STM. This occurred for 15.9% (11.7%) of the trials for monkey G (H).

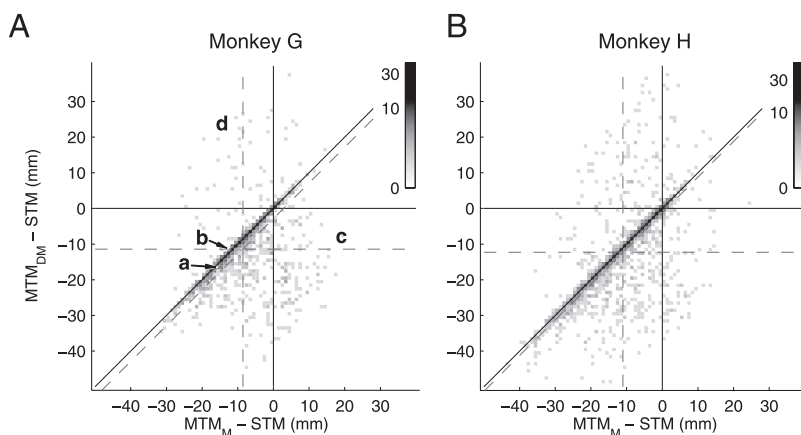


FIG. 5. Two-dimensional histogram of E_{rms} differences between pairs of decoders for (A) monkey G (98 units) and (B) monkey H (99 units). Horizontal axis: E_{rms} difference between MTM_M and STM; vertical axis: E_{rms} difference between MTM_{DM} and STM; diagonal axis: E_{rms} difference between MTM_{DM} and MTM_M . Grayscale intensity (log scale) indicates the number of trials lying in each bin. Dotted lines represent the means of the E_{rms} differences along each axis. Letters **a**, **b**, **c**, and **d** show where the trials in Figs. 2, 3, 6, and 7 lie on the histogram, respectively.

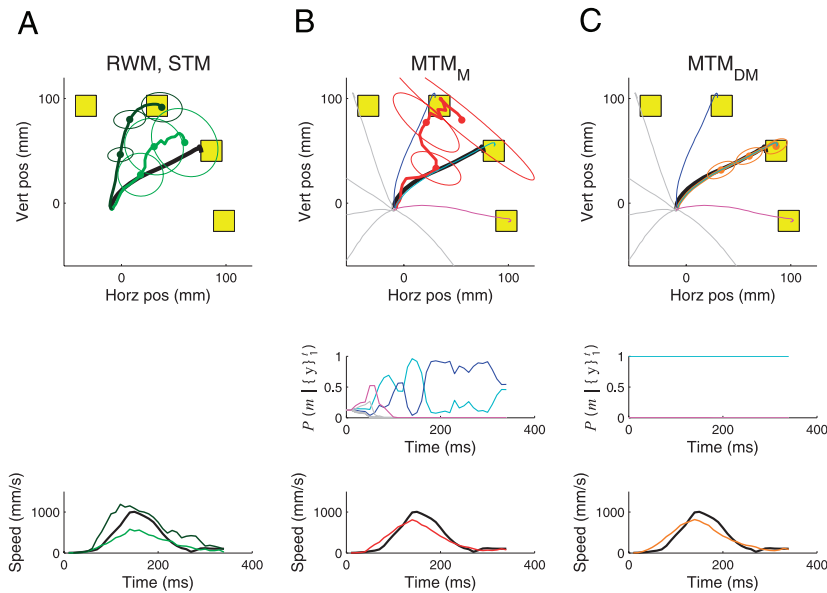


FIG. 6. Outlying test trial in which the MTM_M decoded trajectory exhibited a snap-to-component effect. Figure conventions are identical to those in Fig. 2. For this trial, E_{rms} was 45.3, 18.0, 37.1, and 10.4 mm for RWM, STM, MTM_M , and MTM_{DM} , respectively. Monkey G, 98 units (Experiment G20040508, trial ID 1921).

Although the MTM framework allows for soft weighting between the mixture components, the MTM decoded trajectories often transitioned rather abruptly from one component trajectory estimate to another (referred to as the *snap-to-component* effect). This effect is seen in Fig. 6B (top), where the MTM decoded trajectory (red; E_{rms} : 37.1 mm) moved back and forth between the cyan and blue component trajectory estimates, rather than taking an in-between path as did the STM in Fig. 6A (top, light green; E_{rms} : 18.0 mm). From the perspective of weights, the snap-to-component effect corresponds to rapid weight changes with only a single dominant weight at most time points, as seen in Fig. 6B (middle). At a given time point, the presence of a single dominant weight is related to the variability of the neural responses specified by the fitted model. The effect tends to arise if the neural variability across multiple reaches to a given goal (the “within-class scatter”) is small relative to the differences in mean neural responses across goals (the “between-class scatter”).³ When delay activity was incorporated in the MTM decoder, the competition between the two neighboring reach goals (cyan and blue) was suppressed and the weight corresponding to the actual reach goal (cyan) dominated throughout the reach, as shown in Fig. 6C (middle). Notice that the delay activity strongly favored the actual reach goal (cyan), as indicated by the distribution of weights at $t = 0$. Thus the incorporation of delay activity biased the choice of models toward the correct goal sufficiently strongly to avoid the “snap” to the competing component trajectory. The resulting MTM decoded trajectory (orange; E_{rms} : 10.4 mm) is shown in Fig. 6C (top). It is interesting to note that the RWM and STM decoded trajectories are both pulled by the neural observations toward the same neighboring goal as the MTM_M decoded trajectory.

³ The idea can be simply illustrated by considering a mixture of two Gaussians in one dimension. Let the mixture of Gaussians be defined by $P(y | m=1) = \mathcal{N}(\mu_1, \sigma^2)$ and $P(y | m=2) = \mathcal{N}(\mu_2, \sigma^2)$ with equal priors. We are interested in the posterior $P(m | y_{new})$ for a new data point y_{new} . The smaller σ^2 (the “within-class scatter”) is relative to $|\mu_1 - \mu_2|$ (the “between-class scatter”), the more strongly one of the mixture components will dominate the other in the posterior for all values of y_{new} , except $y_{new} = (\mu_1 + \mu_2)/2$.

Figure 7 shows an outlying test trial (monkey G, 98 units) for which the MTM_{DM} performed worse than the STM. This occurred for 10.6% (12.2%) of the trials for monkey G (H). Without delay activity, the weight for the actual reach goal (cyan) rapidly rose from 1/8 to unity and remained there for the rest of the trial, as seen in Fig. 7B (middle). This led to a fairly accurate MTM decoded trajectory (top, red; E_{rms} : 10.8 mm). As in Fig. 3, the delay activity incorrectly indicated the identity of the upcoming reach goal, as shown in Fig. 7C (middle). The dominant weight (blue) at $t = 0$ did not correspond to the actual reach goal (cyan). However, unlike the trial shown in Fig. 3, the observed peri-movement activity was not able to correct the error in this case and the resulting decoded trajectory (top, orange; E_{rms} : 49.2 mm) headed to a neighboring goal.

The weights represent a probabilistic compromise between the reach goal indicated by the peri-movement activity and that indicated by the delay activity. This can be seen by comparing Eqs. 1 and 2, where the weights $P(m | \{y\}'_t)$ are computed by multiplying a term $P(\{y\}'_t | m)$ that depends only on peri-movement activity with a term $P(m)$ that depends only on delay activity (if delay activity is available). The relative influence of the two types of neural activity is dependent not only on the observed neural data, but also on the particular forms of parametric models used (Eqs. 3–5 and 14). Figure 7 suggests that, for this particular trial, the relative influence of the delay activity was too strong relative to that of the peri-movement activity.

Because the number of units available on an implant generally decreases over time as a result of biological processes (Polikov et al. 2005), we are interested in how the different decoders perform as the number of units varies, shown in Fig. 8. The following two main results from Fig. 4 were preserved across the range of unit counts tested for both monkeys. First, a mixture of linear-Gaussian trajectory models (MTM_M) yielded lower decoding error than either of the non-mixture trajectory models (RWM and STM). Second, the use of prior goal information $P(m)$ in the MTM framework (MTM_{DM}) further decreased decoding error. Except in one case (MTM_M vs. MTM_{DM} for monkey H at 198 units, where so much neural

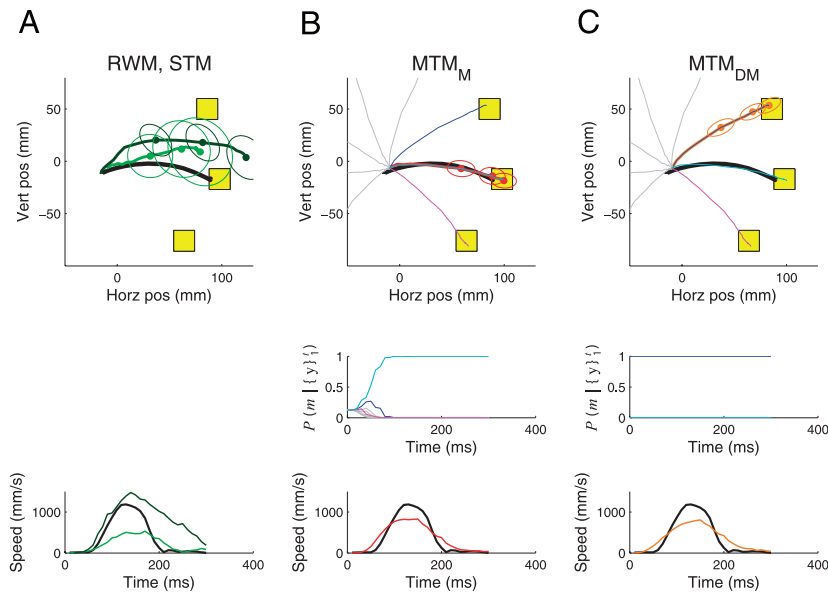


FIG. 7. Outlying test trial in which the peri-movement activity was not able to correct an incorrect reach goal identified by the delay activity. Figure conventions are identical to those in Fig. 2. For this trial, E_{rms} was 27.9, 25.1, 10.8, and 49.2 mm for RWM, STM, MTM_M , and MTM_{DM} , respectively. Monkey G, 98 units (Experiment G20040508, trial ID 1608).

information was available that both decoders performed well), all pairwise comparisons between decoders for a particular monkey and unit count were statistically significant (Wilcoxon paired-sample test, $P < 0.01$). As expected, in all cases, the error decreased as more units were used. Although directly comparing the performance of the RWM and STM decoders is beyond the scope of this work, Fig. 8 explains why the STM decoder outperformed the RWM decoder for monkey G, but the opposite was true for monkey H. Because the RWM decoder was more robust to a loss of units than the STM decoder, there was a crossover point at which the relative performance ordering of the two decoders switched. For each monkey, this crossover point occurred at a different unit count. Because the unit count used in Fig. 4 lay to the right of the crossover point for monkey G and to the left of the crossover point for monkey H, the relative ordering of the RWM and STM differed for the two monkeys in Fig. 4.

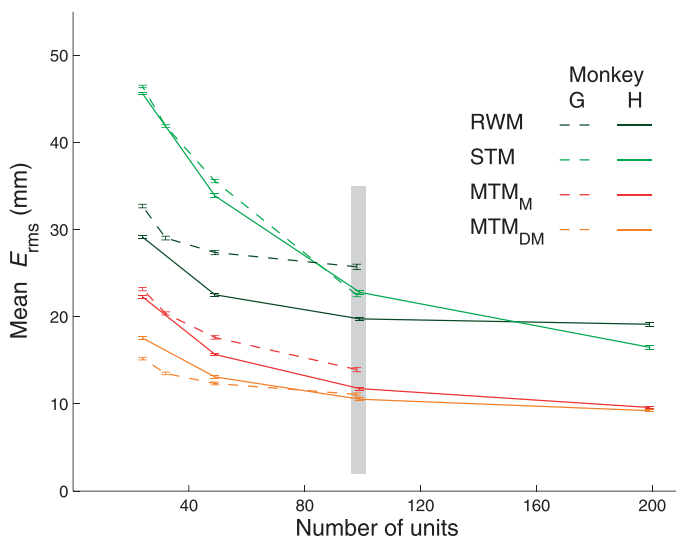


FIG. 8. E_{rms} (mean \pm SE) comparison of RWM (dark green), STM (light green), MTM_M (red), and MTM_{DM} (orange) decoders at different numbers of units. Dashed curves: monkey G, solid curves: monkey H. Vertical gray bar indicates the number of units used for the performance reported in Fig. 4.

We have previously demonstrated two effective decoding strategies for acquiring discrete goals in the subject's workspace. The first involves estimating only the goal identity and simply placing a computer cursor on the decoded goal (Musallam et al. 2004; Santhanam et al. 2006; Shenoy et al. 2003). Although this strategy allows for rapid goal selections on a computer display, controlling a physical prosthetic arm requires knowing more than the identity of the intended reach goal—it requires the specification of a path to the goal. The MTM decoder is an extension of this cursor positioning system, whereby an estimated path is produced that incorporates the same goal information. The second decoding strategy involves defining a canonical trajectory to each goal and selecting among them based on neural activity (Kemere et al. 2002, 2004b). This can be seen as a special case of the MTM, where the trajectory model is a mixture of canonical trajectories. A limitation of this approach is that, if the subject attempts to perform multiple reaches to a particular goal, the decoded trajectory will be identical each time. This poses difficulties if there are obstacles in the workspace (e.g., Hochberg et al. 2006) whose locations may not be fixed. Furthermore, natural reaching movements can exhibit significant variability (for example, in reach speed or curvature) across reaches to the same goal (cf. Fig. A1A), even in highly trained subjects (Churchland et al. 2006a,b,c). Recent evidence has shown that much of this behavioral variability arises from variability in motor planning, which is manifested in delay activity (Churchland et al. 2006b). Because the planned (or “intended”) movement is not identical each time, the use of a canonical trajectory could lead the subject to attempt to bring the canonical trajectory toward the intended trajectory, which could compromise the decoder's effectiveness.

In contrast, the MTM decoder is capable of producing different trajectory estimates to the same goal and capturing trial-by-trial behavioral variability. For example, if the reach speed is faster than usual on a particular trial, this fact should also be reflected in the decoded trajectory. To verify that trial-by-trial behavioral variability was captured by the MTM decoder, we shuffled the decoded trajectories across trials with

TABLE 1. E_{rms} comparison of unshuffled and shuffled trajectories

	STM	MTM _M	MTM _{DM}
Monkey G			
Unshuffled	22.5 ± 0.20	13.9 ± 0.23	11.1 ± 0.17
Shuffled	22.9 ± 0.21	14.7 ± 0.24	11.9 ± 0.18
Monkey H			
Unshuffled	22.8 ± 0.22	11.8 ± 0.18	10.5 ± 0.16
Shuffled	23.0 ± 0.22	12.5 ± 0.18	11.3 ± 0.17

Values are means ± SE (in millimeters). The E_{rms} values for the unshuffled case are identical to those appearing in Fig. 4.

the same reach goal. If the decoded trajectories reflect the trial-by-trial variability of the actual reaches, then we expect the E_{rms} of the shuffled trajectories to be higher than that of the unshuffled trajectories. In cases where the duration of the actual and decoded trajectories differed because of shuffling, E_{rms} was computed by either truncating or padding the decoded trajectory. Table 1 compares the E_{rms} of the unshuffled and shuffled trajectories. For the MTM_M and MTM_{DM} in both monkeys, the shuffled trajectories yielded higher E_{rms} than that of the unshuffled trajectories (Wilcoxon paired-sample test, $P < 0.01$). The effect of shuffling for the STM was largely washed out by the higher overall E_{rms} for both monkeys. These results show that the MTM decoder indeed captured trial-by-trial behavioral variability. The absolute differences in means between the unshuffled and shuffled cases were rather modest because of the stereotypy of the actual reaches in the present data sets (cf. Fig. A1A). In general, repeated reaches to the same goal may exhibit greater variability, leading to a larger absolute E_{rms} difference between the unshuffled and shuffled cases.

Although beyond the scope of the present report, we have also begun to explore how the MTM framework performs for larger numbers of reach goals. A data set with 16 reach goals was collected from monkey H. The goals were arranged in two rings of eight goals at radii of 70 and 120 mm. A total of 189 single- and multineuron units were isolated and 63 trials per reach goal were analyzed. To use roughly the same number of units as in Fig. 4, we randomly split the data set into two halves by units, as described in METHODS. Using 94 units, the mean E_{rms} values for the RWM, STM, MTM_M, and MTM_{DM} decoders were 22.4, 22.4, 20.6, and 17.8 mm, respectively. The two main results from Fig. 4 for eight reach goals were also true for 16 reach goals. First, the mixture of trajectory models (MTM_M) gave lower decoding error than either of the nonmixture trajectory models (RWM and STM) (Wilcoxon paired-sample test, $P < 0.01$). Second, the use of prior goal information $P(m)$ in the MTM framework (MTM_{DM}) further decreased decoding error (Wilcoxon paired-sample test, $P < 0.01$).

DISCUSSION

We have presented a general approach to constructing trajectory models that can exhibit rather complex dynamical behaviors, whose decoder can be implemented to have the same running time as that of simpler trajectory models. The core idea is to combine simple trajectory models, each accurate within a limited regime of movement, in a probabilistic mixture of trajectory models. We showed how trajectories can be decoded from neural activity using the MTM framework and

how prior information about the identity of the upcoming movement regime can be incorporated in a principled way.

The following are two considerations that should guide the construction of a mixture of trajectory models. First, each mixture component, when considered individually, should adequately capture the kinematics within a particular regime of movement. Second, the number of mixture components (i.e., the number of defined movement regimes) should be large enough so that each mixture component is relatively simple and can be efficiently decoded. In this work, each mixture component was a linear-Gaussian model, whose corresponding decoder was a modified Kalman filter. In general, the component-specific decoder may be one of a number of state-of-the-art probabilistic decoders, including the Bayes filter (Brown et al. 1998), particle filter (Brockwell et al. 2004; Shoham et al. 2005), and Kalman filter variants (Wu et al. 2004, 2006).

Although the primary aim of this paper was to lay out the MTM methodology, its application to goal-directed reach trajectories and neural data recorded in PMd and M1 illustrate several key properties of the MTM approach. First, probabilistically mixing simple trajectory models is a powerful approach to create relatively complex dynamic behaviors. As shown in Fig. A1, the salient properties of goal-directed reaches produced under neural motor control can be captured exceedingly well by mixing a set of basic linear-Gaussian models. In particular, the generative trajectories of the MTM (Fig. A1D) are each directed toward one of the eight goals, their across-trial variability is realistic, and their single-trial speed profiles are bell-shaped. Second, the MTM framework provides a natural way to combine delay and peri-movement activity in settings with multiple goals. The *middle panels* in Figs. 2 and 3 illustrate how prior goal information extracted from delay activity is updated as more and more peri-movement activity is observed over time. These two representative trials demonstrate how one type of activity can compensate if the other type of activity provides ambiguous or incorrect information about the current reach. Overall, we found that the MTM decoder yielded more accurate trajectory estimates than did decoders that do not take into account the goal-directed nature of the reaches. The E_{rms} for the RWM, STM, MTM_M, and MTM_{DM} decoders were, respectively, 25.7, 22.5, 13.9, and 11.1 mm (19.8, 22.8, 11.8, and 10.5 mm) for monkey G (H). These results suggest that the MTM framework can provide substantial performance benefits for prosthetic systems that involve guiding a computer cursor or prosthetic arm to acquire discrete goals in the subject's workspace (Carmena et al. 2003; Hochberg et al. 2006; Serruya et al. 2002; Taylor et al. 2002; Wolpaw and McFarland 2004).

For goal-directed reaches, the observed neural activity provides two categorically different types of information about the arm trajectory to be estimated. One type is informative of the moment-by-moment details of the arm trajectory (dynamic), whereas the other is informative of the identity of the upcoming reach goal (static). The former is typically extracted from neural activity from motor cortical areas, such as M1 and PMd, during movement (e.g., Hatsopoulos et al. 2004; Moran and Schwartz 1999). The latter may be obtained from several possible sources. In the present work, the goal information was extracted from "planning" activity present in motor and pre-motor cortical areas preceding the reach. The posterior parietal cortex was also previously shown to encode reach goals and

could serve as a source of goal information (Batista et al. 1999; Shenoy et al. 2003). In addition, there may be events in a patient's surroundings that could be indicative of the upcoming reach goal. For example, if the phone rings, the upcoming reach goal is likely to be the phone.

If the moment-by-moment details of the arm trajectory can be decoded perfectly using only neural activity present during movement, then there would be no need for goal information. However, the moment-by-moment details of the arm trajectory and the goal identity are each decoded with varying levels of uncertainty. When both types of information are available, it is desirable to combine them in a way that takes into account their relative uncertainty and yields a coherent arm trajectory estimate (Kemere et al. 2003, 2004a). Previous approaches either assumed that there was no across-trial variability in the moment-by-moment details of reaches to a given goal (Kemere et al. 2002, 2004b), used a switching scheme between the two types of information (Tkach et al. 2005), or considered the case of a single goal with known arrival time (Kemere and Meng 2005; Srinivasan et al. 2005, 2006). The MTM framework presented here unifies our previous work (Kemere et al. 2002, 2003, 2004a,b) and provides a principled way to combine the two types of information in settings with multiple goals.

To date, the field of cortical prosthetics has largely been split based on which of the two types of information is being used (Pesaran et al. 2006). Whereas *motor* prosthetics attempt to decode the moment-by-moment details of a trajectory (Carmena et al. 2003; Serruya et al. 2002; Taylor et al. 2002), *communication* (or *cognitive*) prosthetics seek to decode the intended reach goal (Musallam et al. 2004; Santhanam et al. 2006; Shenoy et al. 2003). By combining the two types of information, the MTM decoder can be viewed as a way to bridge differences in the design approach of cortical prosthetics.

Based on previous studies, both types of information can likely be extracted from neural activity present in paralyzed patients. First, motor cortical units can be activated (i.e., emit peri-movement activity) without physical movement and be used to control prosthetic cursors or limbs (Carmena et al. 2003; Serruya et al. 2002; Taylor et al. 2002). Recently, motor cortical recordings in tetraplegic patients were used to control a prosthetic cursor (Hochberg et al. 2006). In all of these studies, moment-by-moment details of the trajectory were estimated from the available neural activity. Second, it was shown in PMd (Hatsopoulos et al. 2004; Musallam et al. 2004; Santhanam et al. 2006) and parietal areas (Musallam et al. 2004; Shenoy et al. 2003) that goal information can be reliably decoded from neural activity without physical movement. In addition, functional magnetic resonance imaging studies revealed that motor cortical areas activate similarly in tetraplegics and in healthy humans (Glidden et al. 2006; Shoham et al. 2001). In this work, we extracted both types of information from the same cortical areas—PMd and M1. The type of information being decoded depends on *when* the neural activity occurs relative to the reach, which we assumed to be known. In settings where the subject is free to decide when to reach, it will be necessary to implement a state machine (Afshar et al. 2005; Kemere et al. 2006; Shenoy et al. 2003) that determines the type of information being conveyed by the neural activity at each time point.

Although activity in M1 and PMd generally precedes or coincides with movement, a minority of units show activity *trailing* the associated arm movement (e.g., Paninski et al. 2004). The optimal lags of 36.7 (41.4) percent of the 98 (99) units for monkey G (H) were indeed negative (i.e., neural activity trails movement). These acausal units cannot be used for real-time prosthetic applications without incurring a decoding delay. If their activity is related to proprioception, the activity may altogether be unavailable in disabled patients. We thus excluded the units with acausal lags from our analyses and found the same trends as in Fig. 4 across both monkeys. For monkey G (62 causal units), the mean E_{rms} values for the RWM, STM, MTM_M, and MTM_{DM} decoders were 26.1, 26.0, 14.9, and 10.9 mm, respectively. For monkey H (58 causal units), the mean E_{rms} values for the corresponding decoders were 21.3, 25.6, 12.9, and 11.5 mm. These results provide further support for the suitability of the MTM framework for real-time prosthetic applications.

The MTM framework is more general than indicated by its application to the specific data sets shown in this work. We recognize that numerous additional experiments will be necessary to experimentally verify all aspects and benefits of the MTM framework. Of particular interest is the ability to decode trajectories to novel goals and trajectories that are less stereotyped than those in the present data sets. First, accurately decoding trajectories to novel goals (i.e., those that do not appear in the training set) will require a denser arrangement of goals than the relatively sparse arrangement of eight goals in the present data sets. For example, reaches to a 10×10 grid of goals can be collected for the training set. Then, if an off-grid goal is desired, a relatively accurate trajectory estimate can be formed by weighting the component trajectory estimates corresponding to neighboring goals. With a sparse goal arrangement, there tends to be a single dominant weight at most time points, resulting in the snap-to-component effect described in RESULTS. Second, the flexibility of the MTM framework was not fully used by the present data sets because of the stereotypy of the trajectories. We envision the MTM decoder being applied in settings where repeated reaches to the same goal may exhibit significant variability in, for example, reach curvature or reach speed. This may arise in settings where the subject must avoid obstacles along the path to the goal (e.g., Hochberg et al. 2006). In contrast to a decoder that selects among a set of canonical trajectories (Kemere et al. 2002, 2004b), the MTM framework can be used to capture the trial-by-trial behavioral variability and reconstruct the desired trajectory for each trial individually.

As the number of goals is increased, we expect the MTM decoder to continue to outperform the STM decoder. The reason is that the MTM will continue to better capture the kinematics of goal-directed reaches, in particular the bell-shaped speed profile. Our preliminary results based on 16 reach goals, as described in RESULTS, are encouraging. Our work with 16 reach goals also suggests that, when larger numbers of goals are used, more sophisticated firing rate models may need to be developed to capture the firing rate profiles (cf. Fig. A2) across an increased number of reach goals. The MTM framework can ultimately be extended from M discrete reach goals to a continuum of goal locations.

Although we have focused on goal-directed movements in this work, the MTM framework can potentially be applied in

settings where movements are not goal directed, such as foraging (Brown et al. 1998), ellipse tracing (Brockwell et al. 2004), and pursuit tracking (Shoham et al. 2005; Wu et al. 2004, 2006). As in the goal-directed case, the MTM framework can be used to probabilistically mix simple trajectory models to create relatively complex dynamic behaviors. For example, a rat may be observed to forage more rapidly at the beginning than at the end of an experimental session as the result of changes in motivation. In this case, a single random-walk covariance may not be sufficient to model both rapid and sluggish movements. The decoded path based on a single random-walk covariance may have difficulty either keeping up with rapid movements or holding still during sluggish movements (Santhanam and Shenoy 2003). It may be desirable to use a mixture of random-walk models, where each mixture component has a different random-walk covariance (and possibly a different drift). Similar ideas could apply to arm movements that are not goal directed, whereby different modes of movement could be modeled separately by simple dynamic models then probabilistically mixed. It may even be possible to augment the particular MTM decoder presented in this work with a random-walk mixture component so that it is able to decode both goal-directed and non-goal-directed movements.

APPENDIX

Modal Gaussian approximation for measurement update

We first show that the conditional state posterior $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$ is strictly log-concave given a Gaussian one-step prediction $P(\mathbf{x}_t | \{\mathbf{y}\}_1^{t-1}, m)$. Then, we describe how to find a modal Gaussian approximation of $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$ during the measurement update step (Eq. 7).

Assuming that the one-step prediction $P(\mathbf{x}_t | \{\mathbf{y}\}_1^{t-1}, m)$ is Gaussian with mean \mathbf{x}_t^{t-1} and covariance V_t^{t-1}

$$\begin{aligned}
 L(\mathbf{x}_t) &= \log P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m) \\
 &= \log P(\mathbf{y}_t | \mathbf{x}_t) + \log P(\mathbf{x}_t | \{\mathbf{y}\}_1^{t-1}, m) + \dots \\
 &= \sum_{i=1}^q [-\Delta \cdot \exp(\mathbf{c}_i^t \mathbf{x}_t + d_i) + s_{t-\text{lag}_i}^i (\mathbf{c}_i^t \mathbf{x}_t + d_i)] \\
 &\quad - \frac{1}{2} (\mathbf{x}_t - \mathbf{x}_t^{t-1})' (V_t^{t-1})^{-1} (\mathbf{x}_t - \mathbf{x}_t^{t-1}) + \dots \tag{A1}
 \end{aligned}$$

where the ellipses denote all terms that do not involve \mathbf{x}_t . Taking the gradient and Hessian with respect to \mathbf{x}_t ,

$$\begin{aligned}
 \nabla L(\mathbf{x}_t) &= \sum_{i=1}^q [-\Delta \cdot \exp(\mathbf{c}_i^t \mathbf{x}_t + d_i) + s_{t-\text{lag}_i}^i] \cdot \mathbf{c}_i - (V_t^{t-1})^{-1} (\mathbf{x}_t - \mathbf{x}_t^{t-1}) \tag{A2} \\
 \nabla^2 L(\mathbf{x}_t) &= \sum_{i=1}^q [-\Delta \cdot \exp(\mathbf{c}_i^t \mathbf{x}_t + d_i)] \cdot \mathbf{c}_i \mathbf{c}_i' - (V_t^{t-1})^{-1} \tag{A3}
 \end{aligned}$$

Because $\nabla^2 L(\mathbf{x}_t)$ is negative definite for all \mathbf{x}_t , $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$ is strictly log-concave.

During the measurement update step, we approximate the conditional state posterior as a Gaussian matched to the location and curvature of the mode of $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$, as in Laplace's method (MacKay 2003). Because $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$ is strictly log-concave, its unique mode $\mathbf{x}_{t,m}^*$ can easily be found by Newton's method. The modal Gaussian approximation is thus

$$P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m) \approx \mathcal{N}(\mathbf{x}_{t,m}^*, -\nabla^2 L(\mathbf{x}_{t,m}^*)^{-1}) \tag{A4}$$

In other words, we approximate the mean and covariance of the conditional state posterior as

$$E[\mathbf{x}_t | \{\mathbf{y}\}_1^t, m] \approx \mathbf{x}_{t,m}^* \tag{A5}$$

$$\text{cov}(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m) \approx -\nabla^2 L(\mathbf{x}_{t,m}^*)^{-1} \tag{A6}$$

This approximation works best when $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$ is unimodal, which we know to be the case here because $P(\mathbf{x}_t | \{\mathbf{y}\}_1^t, m)$ is strictly log-concave.

Random-walk trajectory model

To compare the proposed decoders to a state-of-the-art decoder in the field, we also implemented the random-walk trajectory model with Poisson observations presented by Kass and colleagues (Brockwell et al. 2004)

$$\mathbf{v}_t - \mathbf{v}_{t-1} = \mathbf{v}_{t-1} - \mathbf{v}_{t-2} + \boldsymbol{\varepsilon}_t \tag{A7}$$

$$\begin{bmatrix} \mathbf{v}_2 \\ \mathbf{v}_1 \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\pi}, V) \tag{A8}$$

$$s_{t-\text{lag}_i}^i | \mathbf{v}_t \sim \text{Poisson}[\exp(\mathbf{c}_i^t \tilde{\mathbf{v}}_t + d_i) \Delta] \tag{A9}$$

where $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, Q)$ in Eq. A7, $\mathbf{v}_t \in \mathbb{R}^{p \times 1}$ is the arm velocity at time t , $\tilde{\mathbf{v}}_t$ is defined to be $[\mathbf{v}_t' | \|\mathbf{v}_t\|]'$ in Eq. A9, and $\|\mathbf{v}_t\|$ is the arm speed at time t . As in Eq. 5, $s_{t-\text{lag}_i}^i$ is the peri-movement spike count of the i th unit at time $t - \text{lag}_i$, where lag_i is the time lag between the neural firing of unit $i \in \{1, \dots, q\}$ and the associated arm velocity. Spike counts are taken in time bins of width Δ . The parameters $Q \in \mathbb{R}^{p \times p}$, $\boldsymbol{\pi} \in \mathbb{R}^{2p \times 1}$, $V \in \mathbb{R}^{2p \times 2p}$, $\text{lag}_i \in \mathbb{Z}$, $\mathbf{c}_i \in \mathbb{R}^{(p+1) \times 1}$, $d_i \in \mathbb{R}$ are fit to training data, as subsequently described. Note that the random-walk trajectory model is a special case of the linear-Gaussian trajectory model with appropriately chosen parameters in Eqs. 3 and 4.

Equations A7 and A8 define the random-walk trajectory model that imposes smoothness in acceleration; Eq. A9 defines the Poisson observation model. To decode arm trajectories using this probabilistic model, we followed Kass and colleagues (Brockwell et al. 2004) and implemented particle filtering with 2,500 particles at each time step. This yielded a velocity estimate at each time step. To obtain a single decoded position trajectory, the means of these velocity estimates were integrated over time. Because the arm state does not include positional variables in this model, we assumed the actual initial arm position was known. Thus the decoder based on the random-walk trajectory model was given a slight advantage over the other decoders.

Model fitting

TRAJECTORY MODEL. This section describes how to fit the following three trajectory models: a random-walk model (RWM, Eqs. A7 and A8) in acceleration, a single linear-Gaussian trajectory model (STM, Eqs. 3 and 4 for special case of $M = 1$), and a mixture of linear-Gaussian trajectory models (MTM, Eqs. 3 and 4). Arm position data were taken from 50 ms before movement onset to the end of the trial. The data were then padded with the final arm position up to 1,000 ms after movement end to emphasize the importance of stopping at the reach goal. In effect, this penalized trajectory models whose trajectories simply pass through, rather than come to rest at, the reach goals. Each of the trajectory models was fitted to the padded arm data with a time step of $dt = 10$ ms. Although arm position was tracked in three dimensions, we only analyzed movement in the plane of the fronto-parallel screen because there was relatively little movement perpendicular to the screen. Arm velocity and acceleration were obtained by taking first and second differences of the arm position.

For the STM and MTM, the following physical quantities were included in the arm state vector \mathbf{x}_t : position, velocity, acceleration,

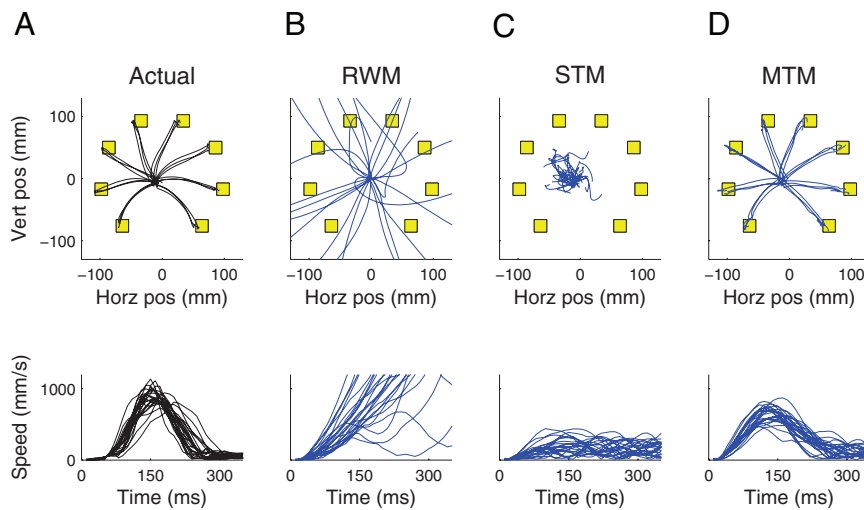


FIG. A1. Position trajectories (top panels) and speed profiles (bottom panels). A: collected empirically, B: generated by the RWM, C: generated by the STM, and D: generated by the MTM. Only 24 reaches (3 to each reach goal) are shown in each column for clarity.

position magnitude, and velocity magnitude. As shown in Eq. A10, this eight-dimensional state vector included two dimensions each for position, velocity, and acceleration; and one dimension each for position magnitude and velocity magnitude. Sample trajectories generated from the trajectory model were qualitatively similar, regardless of whether the magnitude terms were included in the state vector. However, the magnitude terms were critical for fitting the observation model (Eq. 5) to the neural data, as described later.

The parameters of all three trajectory models were fit using maximum likelihood. For the RWM, the fitted parameters were $\{Q, \pi, V\}$, where Q was constrained to be diagonal (Brockwell et al. 2004). For the STM and MTM, the fitted parameters were $\{A_m, \mathbf{b}_m, Q_m, \pi_m, V_m\}$ (STM: $m = 1$; MTM: $m \in \{1, \dots, 8\}$). For the STM, a single linear-Gaussian trajectory model was shared across all goal locations. The STM is similar to the trajectory model used by Donoghue and colleagues (Wu et al. 2004 2006), where it was applied to pursuit-tracking and “pinball” tasks. In contrast, for the MTM, a separate linear-Gaussian trajectory model was trained for each reach goal, based only on reaches to that goal.

For the STM and MTM, the fitted transition matrices A_m and additive constants \mathbf{b}_m took on the form shown in Eq. A10, where \star denotes a nonzero entry and $dt = 10$ ms. The elements of the state vector \mathbf{x}_t are included for visual reference

$$A_m = \begin{bmatrix} 1 & 0 & dt & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & dt & 0 & 0 & 0 \\ \star & & & & & & \star & \\ \vdots & & & & & & \vdots & \\ \star & & & & & & \star & \end{bmatrix}, \mathbf{b}_m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \star \\ \vdots \\ \star \end{bmatrix}, \mathbf{x}_t = \begin{bmatrix} \text{horz pos} \\ \text{vert pos} \\ \text{horz vel} \\ \text{vert vel} \\ \text{horz acc} \\ \text{vert acc} \\ \text{mag pos} \\ \text{mag vel} \end{bmatrix} \quad (A10)$$

Although not explicitly constrained as such in the fitting procedure, the fitted A_m and \mathbf{b}_m took on this form as a result of the physical relationships of the state vector elements.⁴

Figure A1A shows position trajectories to each reach goal collected empirically, along with the corresponding speed profiles. Three properties of goal-directed reaches are seen in Fig. A1A. First, the trajectories lead to discrete reach goals rather than taking on arbitrary

⁴ Although the position magnitude has an exact nonlinear relationship with the horizontal and vertical positions, this model assumes an approximate linear relationship between the position magnitude and all state elements. As a result, the position magnitude will not necessarily be consistent with the horizontal and vertical positions in the generative and decoded trajectories. This is also the case for velocity.

paths in the workspace. Second, multiple reaches to the same goal are not all identical. There is variability in both the position traces and speed profiles. Third, the trajectories start at rest, proceed out to the reach goal, and end at rest. The degree to which the trajectory model captures the kinematics of the empirically collected reaches directly affects the accuracy with which new trajectories can be decoded from neural data. We therefore seek a trajectory model that can capture all three properties of goal-directed reaches.

We can qualitatively assess the fitness of different trajectory models by generating sample trajectories from the fitted models and comparing them with the empirically collected trajectories. Decoders based on the different trajectory models are quantitatively compared in RESULTS. Generative trajectories of the fitted RWM, STM, and MTM are shown in Fig. A1, B–D. Note that these are sample trajectories of the trajectory models and are not decoded trajectories; generating these trajectories did not involve neural data. The RWM (Eq. A7) provides smoothness in acceleration, where the degree of smoothness is determined by the random-walk covariance Q . We generated sample velocity trajectories according to Eqs. A7 and A8 using a Q matrix fitted to training data, then integrated the velocities over time to obtain sample position trajectories (Fig. A1B). On the other hand, the STM favors certain characteristic trajectory patterns in arm state space. One characteristic pattern that looks similar to Fig. A1A has trajectories emanating radially outward from the origin (ignoring the non-position terms in the arm state vector for now). Such trajectories (not shown) extend outward indefinitely and cannot stop at the reach goals. To minimize the average mismatch between the trajectory model and the empirically collected trajectories (Fig. A1A) over the entire duration of the padded arm data, the STM fitted to the training data has sample position trajectories (Fig. A1C) that proceed outward very slowly. Other features seen in the sample trajectories in Fig. A1C can be explained by the presence of non-position terms in the arm state vector and the noise covariance Q_m in Eqs. 3 and 4.

Although the sample trajectories of the RWM and STM each reflect some aspects of arm kinematics, they are not flexible enough to capture the goal-directed nature of the actual reaches. The corresponding speed profiles also do not match those of the actual reaches very well. In contrast, as shown in Fig. A1D, the sample trajectories of the MTM exhibit the three properties of goal-directed reaches: 1) the trajectories are directed toward the eight discrete reach goals, 2) there is variability among trajectories to the same goal, and 3) the trajectories start and end roughly at rest. Furthermore, these sample trajectories are similar to the empirically collected trajectories in Fig. A1A in terms of their bell-shaped speed profiles and the across-trial variability seen in the position traces and speed profiles. In essence, compared with the RWM and STM, the MTM better captures the kinematics of goal-directed reaches.

The following is the intuition behind how a model as simple as a mixture of linear-Gaussian models can capture the essential properties of goal-directed reaches. For each m , the fitted transition matrix A_m (Eq. 3) defines a convergent linear-Gaussian model. In other words, in the noiseless case, its sample trajectories converge to a point in state space. If $\mathbf{b}_m = \mathbf{0}$, this stable equilibrium point is the origin of the state space. For a nonzero \mathbf{b}_m , the stable equilibrium point (in particular, those elements corresponding to arm position) can be shifted away from the origin and, in this case, lie at the m th reach goal. Regardless of where the sample trajectories start, they are directed by the m th mixture component toward the m th reach goal, where they come to rest. These trajectories are further constrained by the fitted $\boldsymbol{\pi}_m$ and V_m (Eq. 4) to start near the center of the workspace with nearly zero velocity. Thus one can imagine a point mass, initially at rest at the center of the workspace, that is released and directed toward the m th reach goal, where it comes to rest.

This behavior can be confirmed by analyzing the fitted model parameters in the noiseless case. First, we verified that the absolute values of the eigenvalues of the fitted A_m are all < 1 . This ensures that any equilibrium point that is found is stable. Second, based on Eq. 3, the equilibrium point location can be expressed as $(I - A_m)^{-1}\mathbf{b}_m$. For each goal, we verified that this point corresponds not only to the goal position, but also to zero velocity and acceleration. The position and velocity magnitudes are roughly 10 cm and zero, respectively.

The trajectory model can be viewed, in the space of all possible trajectories, as a specification of which trajectories are more likely than others and by how much. This information is encoded in the parametric form of the trajectory model (e.g., linear-Gaussian), as well as in the fitted values of the model parameters. For the trajectory models considered in this work, there is a nonzero probability of generating any arbitrary trajectory in Fig. A1, B–D. However, for the MTM fitted to the training data shown in Fig. A1A, trajectories that do not head toward one of the eight reach goals or those that do not have a bell-shaped speed profile are far less likely than those that do. Although it is technically possible to generate a trajectory in Fig. A1D that looks very different from those shown, the chances are negligibly small. Thus the MTM can be viewed as imposing a soft constraint on what trajectories are possible; how steeply the soft constraint drops off depends on how tightly the training trajectories are clustered in Fig. A1A.

OBSERVATION MODEL. For each observation model (Eqs. 5 and A9), we sought the optimal lag for each unit and the parameters $\{c_i, d_i\}$, where i indexes unit. The optimal lag refers to the temporal relationship between the activity of a neural unit and the arm trajectory (Moran and Schwartz 1999). For example, if a unit is causally related to motor execution, the unit's firing would be expected to lead the arm movement in time. Donoghue and colleagues (Wu et al. 2006) used a greedy algorithm to find the set of lags that minimized the uncertainty of the position estimates. In contrast, Kass and colleagues (Brockwell et al. 2004) chose the best-fitting lag for each unit by comparing model deviances. The optimal lag could be found for each unit separately because the units were modeled to be independent given the arm state (cf. Eq. A9). We adopted the latter approach.

We considered a fixed window of peri-movement neural activity starting 200 ms before movement onset (t_1) and ending 150 ms after movement end (t_2). Spike counts were taken in $\Delta = 10$ -ms bins. This was aligned to segments of arm trajectory data of the same duration, but offset by 31 possible lags ranging from 150 to -150 ms at 10-ms intervals.⁵ The convention here is that positive lags are causal (neural activity leads arm movement), whereas negative lags are acausal.

⁵ Based on the task design, we allowed as large a range of possible lags as possible without violating behavioral epoch boundaries. For example, if a causal lag is too large, the window of peri-movement activity would overlap with the delay period. If an acausal lag is too large, the window of peri-movement activity would overrun the end of the trial.

Acausal lags in the context of prosthetic applications were addressed earlier in the DISCUSSION.

The following generalized linear model (GLM) fitting procedure was performed for each of the q units. For notational simplicity, the unit index i is omitted. Let $\{\mathbf{x}\}$ denote \mathbf{x}_t at all times, $\{s\}_{t_1}^{t_2}$ denote the spike counts from time t_1 to t_2 , and the observation model parameters $\theta = \{c, d\}$. We seek the parameters θ and lag that maximize the likelihood $P(\{s\}_{t_1}^{t_2} | \theta, \text{lag}, \{\mathbf{x}\})$. First, we used the built-in glmfit function in Matlab (The MathWorks, Natick, MA) to find the maximum-likelihood parameters θ^* for each possible lag. Next, the likelihood was evaluated at $\theta = \theta^*$ for each lag. The maximum-likelihood lag and its corresponding parameters θ^* were then used in Eq. 5. The same fitting procedure was used for Eq. A9. The optimal lag should be interpreted as the best-fitting temporal alignment between the neural activity and arm trajectories for the particular parametric observation model used. In general, different observation models yield different optimal lags. Thus the optimal lag is model dependent and only roughly reflects how the unit is temporally related to motor execution.

We included magnitude terms in the arm state vector for the STM and MTM for the same reason that the velocity magnitude (i.e., the arm speed) appears in Eq. A9 for the RWM; that is, to allow the associated firing rate models to capture nondirectional firing rate modulations. The firing rate models are the exponentials in Eqs. 5 and A9, where each dimension of \mathbf{x}_t in Eq. 5 and $\tilde{\mathbf{v}}_t$ in Eq. A9 is an explanatory variable. The importance of including arm speed as an explanatory variable for firing rate modulations was first recognized by Schwartz and colleagues (Schwartz 1992). Although the focus of this work is not to compare different parametric firing rate models, we demonstrate this point by comparing the firing rate profiles that result from including and excluding the magnitude terms in the arm state vector \mathbf{x}_t in Eq. 5 for one illustrative unit (Fig. A2). Using the methods described earlier, we found the optimal lag and fitted $\{c_i, d_i\}$ based on the training data. Then, actual arm trajectories (from test data) were mapped to mean firing rates using the firing rate model in Eq. 5. These predicted mean firing rates were aligned on movement onset and averaged across test trials. Figure A2 shows the resulting firing rate profiles for this unit when including (blue) and excluding (red) the magnitude terms in the firing rate model. These firing rate profiles can be compared with the empirical firing rate histograms (gray) for the same test trials. In this case, the magnitude terms allowed firing rate peaks to be present in all reach directions, considerably improving the model fit for the lower reach goals. Nondirectional firing rate modulations, like those shown in Fig. A2, were common across the population of units recorded in both monkeys and were better captured by including magnitude terms as explanatory variables.

Deriving confidence intervals for MTM estimates

We seek to express the uncertainty of the overall MTM estimate, $\text{cov}(\mathbf{x}_t | \{\mathbf{y}\}_t^t)$, in terms of the conditional state posteriors $P(\mathbf{x}_t | \{\mathbf{y}\}_t^t, m)$ and weights $P(m | \{\mathbf{y}\}_t^t)$. By the definition of covariance

$$\text{cov}(\mathbf{x}_t | \{\mathbf{y}\}_t^t) = E[\mathbf{x}_t \mathbf{x}_t' | \{\mathbf{y}\}_t^t] - E[\mathbf{x}_t | \{\mathbf{y}\}_t^t] E[\mathbf{x}_t | \{\mathbf{y}\}_t^t]' \tag{A11}$$

As shown in Eq. 13, the term $E[\mathbf{x}_t | \{\mathbf{y}\}_t^t]$ can be expanded by conditioning on m

$$E[\mathbf{x}_t | \{\mathbf{y}\}_t^t] = \sum_{m=1}^M E[\mathbf{x}_t | \{\mathbf{y}\}_t^t, m] P(m | \{\mathbf{y}\}_t^t) \tag{A12}$$

Similarly

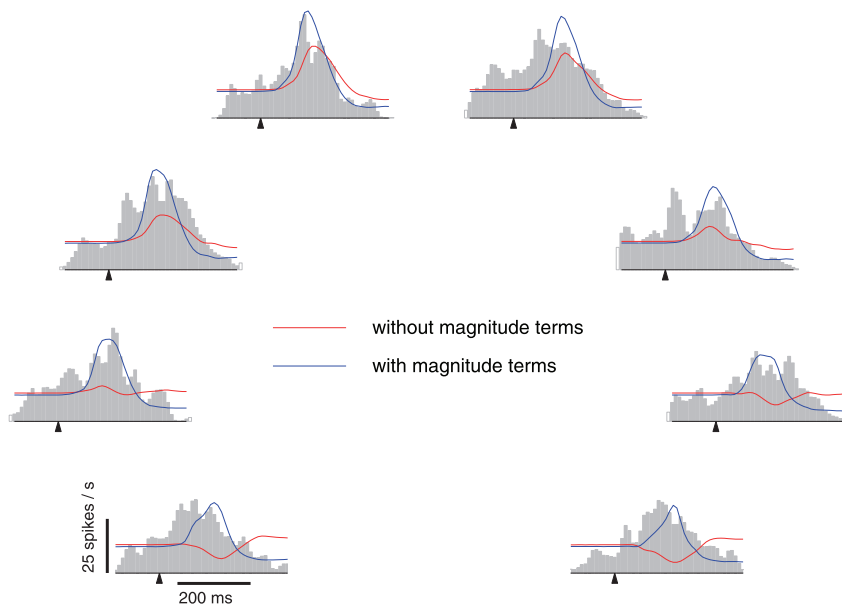


FIG. A2. Comparison of empirical firing rate histograms (gray) with firing rate profiles predicted by firing rate models with (blue) and without (red) magnitude terms (in this case, position and velocity magnitudes) as explanatory variables. Vertical arrows denote movement onset. Each panel corresponds to one reach goal (Unit G20040508.38.1).

$$\begin{aligned}
 E[\mathbf{x}_t | \{\mathbf{y}'_1\}] &= \sum_{m=1}^M E[\mathbf{x}_t | \{\mathbf{y}'_1\}, m] P(m | \{\mathbf{y}'_1\}) \\
 &= \sum_{m=1}^M (\text{cov}(\mathbf{x}_t | \{\mathbf{y}'_1\}, m) + E[\mathbf{x}_t | \{\mathbf{y}'_1\}, m] E[\mathbf{x}_t | \{\mathbf{y}'_1\}, m]') P(m | \{\mathbf{y}'_1\})
 \end{aligned}
 \tag{A13}$$

Thus the uncertainty of the overall MTM estimate can be expressed analytically in terms of the mean $E[\mathbf{x}_t | \{\mathbf{y}'_1\}, m]$ and covariance $\text{cov}(\mathbf{x}_t | \{\mathbf{y}'_1\}, m)$ of the conditional state posteriors and the weights $P(m | \{\mathbf{y}'_1\})$.

ACKNOWLEDGMENTS

We thank M. Howard for expert animal care, S. Eisensee for administrative assistance, and Dr. Mark Churchland and J. Cunningham for helpful discussions. We also thank the three anonymous reviewers whose comments helped to considerably improve the presentation of this work.

GRANTS

This work was supported by National Defense Science and Engineering Graduate Fellowships, National Science Foundation (NSF) Graduate Research Fellowships, National Institute of Neurological Disorders and Stroke/Collaborative Research in Computational Neuroscience Grant 5-R01-NS054283-02, Focus Center Research Program Center for Circuit and System Solutions under Contract 2003-CT-888, Medical Scientist Training Program, Christopher Reeve Paralysis Foundation, Gatsby Charitable Foundation, Burroughs Wellcome Fund Career Award in the Biomedical Sciences, Stanford Center for Integrated Systems, NSF Center for Neuromorphic Systems Engineering at Caltech, Office of Naval Research, Sloan Foundation, and Whitaker Foundation.

REFERENCES

Afshar A, Achtman N, Santhanam G, Ryu SI, Yu BM, Shenoy KV. Free-paced target estimation in a delayed-reach task. *Soc Neurosci Abstr* 401.13, 2005.
 Ashe J, Georgopoulos AP. Movement parameters and neural activity in motor cortex and area 5. *Cereb Cortex* 4: 590–600, 1994.
 Batista AP, Buneo CA, Snyder LH, Andersen RA. Reach plans in eye-centered coordinates. *Science* 285: 257–260, 1999.
 Brockwell AE, Rojas AL, Kass RE. Recursive Bayesian decoding of motor cortical signals by particle filtering. *J Neurophysiol* 91: 1899–1907, 2004.
 Brown EN, Frank LM, Tang D, Quirk MC, Wilson MA. A statistical paradigm for neural spike train decoding applied to position prediction from

the ensemble firing patterns of rat hippocampal place cells. *J Neurosci* 18: 7411–7425, 1998.
 Caminiti R, Johnson PB, Galli C, Ferraina S, Burnod Y. Making arm movements within different parts of space: the premotor and motor cortical representation of a coordinate system for reaching to visual targets. *J Neurosci* 11: 1182–1197, 1991.
 Carmena JM, Lebedev MA, Crist RE, O’Doherty JE, Santucci DM, Dimitrov DF, Patil PG, Henriquez CS, Nicolelis MAL. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol* 1: 193–208, 2003.
 Chan SS, Moran DW. Computational model of a primate arm: from hand position to joint angles, joint torques and muscle forces. *J Neural Eng* 3: 327–337, 2006.
 Chapin JK, Moxon KA, Markowitz RS, Nicolelis MAL. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nat Neurosci* 2: 664–670, 1999.
 Churchland MM, Afshar A, Shenoy KV. A central source of movement variability. *Neuron* 52: 1085–1096, 2006a.
 Churchland MM, Santhanam G, Shenoy KV. Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach. *J Neurophysiol* 96: 3130–3146, 2006b.
 Churchland MM, Yu BM, Ryu SI, Santhanam G, Shenoy KV. Neural variability in premotor cortex provides a signature of motor preparation. *J Neurosci* 26: 3697–3712, 2006c.
 Cowan TM, Taylor DM. Predicting reach goal in a continuous workspace for command of a brain-controlled upper-limb neuroprosthesis. *Proc 2nd IEEE EMBS Neural Eng*, 2005, p. 74–77.
 Crammond DJ, Kalaska JF. Prior information in motor and premotor cortex: activity during the delay period and effect on pre-movement activity. *J Neurophysiol* 84: 986–1005, 2000.
 Evarts EV. Relation of pyramidal tract activity to force exerted during voluntary movement. *J Neurophysiol* 31: 14–27, 1968.
 Fetz EE. Are movement parameters recognizably coded in the activity of single neurons? *Behav Brain Sci* 15: 679–690, 1992.
 Georgopoulos AP, Kalaska JF, Caminiti R, Massey JT. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J Neurosci* 2: 1527–1537, 1982.
 Georgopoulos AP, Schwartz AB, Kettner RE. Neuronal population coding of movement direction. *Science* 233: 1416–1419, 1986.
 Glidden HK, Yozbatiran N, Rizzuto DS, Cramer SC, Andersen RA. fMRI during goal-directed movement planning in normal and spinal cord-injured subjects. *Soc Neurosci Abstr* 13.3, 2006.
 Hatsopoulos N, Joshi J, O’Leary JG. Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles. *J Neurophysiol* 92: 1165–1174, 2004.
 Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, Caplan AH, Branner A, Chen D, Penn RD, Donoghue JP. Neuronal ensemble

- control of prosthetic devices by a human with tetraplegia. *Nature* 442: 164–171, 2006.
- Kemere C, Meng T.** Optimal estimation of feed-forward-controlled linear systems. *Proc IEEE ICASSP*, 2005, p. 353–356.
- Kemere C, Sahani M, Meng TH.** Robust neural decoding of reaching movements for prosthetic systems. *Proc 25th Annu Conf IEEE EMBS*, 2003, p. 2079–2082.
- Kemere C, Santhanam G, Yu BM, Ryu SI, Meng TH, Shenoy KV.** Model-based decoding of reaching movement for prosthetic systems. *Proc 26th Annu Conf IEEE EMBS*, 2004a, p. 4524–4528.
- Kemere C, Shenoy KV, Meng TH.** Model-based neural decoding of reaching movements: a maximum likelihood approach. *IEEE Trans Biomed Eng* 51: 925–932, 2004b.
- Kemere C, Yu BM, Santhanam G, Ryu SI, Afshar A, Meng TH, Shenoy KV.** Hidden Markov models for spatial and temporal estimation for prosthetic control. *Soc Neurosci Abstr* 256.17, 2006.
- Kemere CT, Santhanam G, Yu BM, Shenoy KV, Meng TH.** Decoding of plan and peri-movement neural signals in prosthetic systems. In: *IEEE Workshop on Signal Processing Systems*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2002, p. 276–283.
- Kennedy PR, Bakay RAE.** Restoration of neural output from a paralyzed patient by a direct brain connection. *Neuroreport* 9: 1707–1711, 1998.
- Kurata K.** Premotor cortex of monkeys: set- and movement-related activity reflecting amplitude and direction of wrist movements. *J Neurophysiol* 69: 187–200, 1993.
- Leuthardt EC, Schalk G, Wolpaw JR, Ojemann JG, Moran DW.** A brain-computer interface using electrocorticographic signals in humans. *J Neural Eng* 1: 63–71, 2004.
- MacKay DJC.** *Information Theory Inference and Learning Algorithms*. Cambridge, UK: Cambridge Univ. Press, 2003.
- Maynard EM, Hatsopoulos NG, Ojakangas CL, Acuna BD, Sanes JN, Normann RA, Donoghue JP.** Neuronal interactions improve cortical population coding of movement direction. *J Neurosci* 19: 8083–8093, 1999.
- Messier J, Kalaska JF.** Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task. *J Neurophysiol* 84: 152–165, 2000.
- Moran DW, Schwartz AB.** Motor cortical representation of speed and direction during reaching. *J Neurophysiol* 82: 2676–2692, 1999.
- Moran DW, Schwartz AB.** One motor cortex, two different views (Letter). *Nat Neurosci* 3: 963, 2000.
- Musallam S, Corneil BD, Greger B, Scherberger H, Andersen RA.** Cognitive control signals for neural prosthetics. *Science* 305: 258–262, 2004.
- Paninski L, Fellows MR, Hatsopoulos NG, Donoghue JP.** Spatiotemporal tuning of motor cortical neurons for hand position and velocity. *J Neurophysiol* 91: 515–532, 2004.
- Pesaran B, Musallam S, Andersen RA.** Cognitive neural prosthetics. *Curr Biol* 16: 77–80, 2006.
- Polikov VS, Tresco PA, Reichert WM.** Response of brain tissue to chronically implanted neural electrodes. *J Neurosci Methods* 148: 1–18, 2005.
- Reina GA, Moran DW, Schwartz AB.** On the relationship between joint angular velocity and motor cortical discharge during reaching. *J Neurophysiol* 85: 2576–2589, 2001.
- Riehle A, Requin J.** 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, 1989.
- Sahani M.** *Latent Variable Models for Neural Data Analysis* (PhD thesis). Pasadena, CA: California Institute of Technology, 1999.
- Santhanam G, Ryu SI, Yu BM, Afshar A, Shenoy KV.** A high-performance brain-computer interface. *Nature* 442: 195–198, 2006.
- Santhanam G, Sahani M, Ryu SI, Shenoy KV.** An extensible infrastructure for fully automated spike sorting during online experiments. *Proc 26th Annu Conf IEEE EMBS*, 2004, p. 4380–4384.
- Santhanam G, Shenoy KV.** Methods for estimating neural step sequences in neural prosthetic applications. *Proc 1st IEEE EMBS Neural Eng*, 2003, p. 344–347.
- Schwartz AB.** Motor cortical activity during drawing movements: single-unit activity during sinusoid tracing. *J Neurophysiol* 68: 528–541, 1992.
- Scott SH.** Optimal feedback control and the neural basis of volitional motor control. *Nat Rev Neurosci* 5: 532–546, 2004.
- Scott SH, Kalaska JF.** Reaching movements with similar hand paths but different arm orientations. I. Activity of individual cells in motor cortex. *J Neurophysiol* 77: 826–852, 1997.
- Sergio LE, Kalaska JF.** Changes in the temporal pattern of primary motor cortex activity in a directional isometric force versus limb movement task. *J Neurophysiol* 80: 1577–1583, 1998.
- Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR, Donoghue JP.** Instant neural control of a movement signal. *Nature* 416: 141–142, 2002.
- Shen L, Alexander GE.** Preferential representation of instructed target location versus limb trajectory in dorsal premotor area. *J Neurophysiol* 77: 1195–1212, 1997.
- Shenoy KV, Meeker D, Cao S, Kureshi SA, Pesaran B, Mitra P, Buneo CA, Batista AP, Burdick JW, Andersen RA.** Neural prosthetic control signals from plan activity. *Neuroreport* 14: 591–596, 2003.
- Shoham S, Halgren E, Maynard EM, Normann RA.** Motor-cortical activity in tetraplegics (Brief Communication). *Nature* 413: 793, 2001.
- Shoham S, Paninski LM, Fellows MR, Hatsopoulos NG, Donoghue JP, Normann RA.** Statistical encoding model for a primary motor cortical brain-machine interface. *IEEE Trans Biomed Eng* 52: 1313–1322, 2005.
- Srinivasan L, Brown EN.** Dynamic-goal state equations for tracking reaching movements using neural signals. *Proc 1st IEEE/RAS-EMBS Conf Biomed Robotics Biomechanics*, 2006, p. 758–762.
- Srinivasan L, Eden UT, Willsky AS, Brown EN.** Goal-directed state equation for tracking reaching movements using neural signals. *Proc 2nd IEEE EMBS Neural Eng*, 2005, p. 352–355.
- Srinivasan L, Eden UT, Willsky AS, Brown EN.** A state-space analysis for reconstruction of goal-directed movements using neural signals. *Neural Comput* 18: 2465–2494, 2006.
- Tanji J, Evarts EV.** Anticipatory activity of motor cortex neurons in relation to direction of an intended movement. *J Neurophysiol* 39: 1062–1068, 1976.
- Taylor DM, Tillery SIH, Schwartz AB.** Direct cortical control of 3D neuroprosthetic devices. *Science* 296: 1829–1832, 2002.
- Tkach DC, Reimer J, Hatsopoulos NG.** A hybrid neuromotor brain-machine interface using trajectory and goal state control modes. *Soc Neurosci Abstr* 707.11, 2005.
- Weinrich M, Wise SP.** The premotor cortex of the monkey. *J Neurophysiol* 2: 1329–1345, 1982.
- Wolpaw JR, McFarland DJ.** Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proc Natl Acad Sci USA* 101: 17849–17854, 2004.
- Wu W, Black MJ, Mumford D, Gao Y, Bienenstock E, Donoghue JP.** Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Trans Biomed Eng* 51: 933–942, 2004.
- Wu W, Gao Y, Bienenstock E, Donoghue JP, Black MJ.** Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Comput* 18: 80–118, 2006.
- Yu BM, Ryu SI, Santhanam G, Churchland MM, Shenoy KV.** Improving neural prosthetic system performance by combining plan and peri-movement activity. *Proc 26th Annu Conf IEEE EMBS*, 2004, p. 4516–4519.
- Yu BM, Santhanam G, Ryu SI, Shenoy KV.** Feedback-directed state transition for recursive Bayesian estimation of goal-directed trajectories. In: *Computational and Systems Neuroscience (COSYNE) Meeting*. Available online at <http://www.cosyne.org/program05/291.html>. 2005.
- Zhang K, Ginzburg I, McNaughton BL, Sejnowski TJ.** Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *J Neurophysiol* 79: 1017–1044, 1998.
- Zumsteg ZS, Kemere C, O'Driscoll S, Santhanam G, Ahmed RE, Shenoy KV, Meng TH.** Power feasibility of implantable digital spike sorting circuits for neural prosthetic systems. *IEEE Trans Neural Syst Rehabil Eng* 13: 272–279, 2005.