Neural Decoding of Hand Motion Using a Linear State-Space Model With Hidden States

Wei Wu, Jayant E. Kulkarni, Nicholas G. Hatsopoulos, and Liam Paninski

***Abstract*—The Kalman filter has been proposed as a model to decode neural activity measured from the motor cortex in order to obtain real-time estimates of hand motion in behavioral neurophysiological experiments. However, currently used linear state-space models underlying the Kalman filter do not take into account other behavioral states such as muscular activity or the subject's level of attention, which are often unobservable during experiments but may play important roles in characterizing neural controlled hand movement. To address this issue, we depict these unknown states as one multidimensional *hidden state* in the linear state-space framework. This new model assumes that the observed neural firing rate is directly related to this hidden state. The dynamics of the hand state are also allowed to impact the dynamics of the hidden state, and vice versa. The parameters in the model can be identified by a conventional expectation-maximization algorithm. Since this model still uses the linear Gaussian framework, hand-state decoding can be performed by the efficient Kalman filter algorithm. Experimental results show that this new model provides a more appropriate representation of the neural data and generates more accurate decoding. Furthermore, we have used recently developed computationally efficient methods by incorporating *a priori* information of the targets of the reaching movement. Our results show that the hidden-state model with target-conditioning further improves decoding accuracy.**

***Index Terms*—Hidden states, Kalman filter, motor cortex, neural decoding, state-space model.**

I. Introduction

NEURAL decoding, which converts brain signals into control commands that drive external devices such as computer cursors or robotic limbs, is a key component of fast-growing research efforts on brain-machine interfaces or neural motor prosthetics [1]–[6]. A number of mathematical algorithms have been developed to decode population neuronal activity in the motor or the premotor cortex. These methods focus on estimating either continuous motor trajectories or parameters of motor primitives such as hand movement direction. The methods for decoding motor primitives have largely focused on center-out type movements [7], with commonly-used methods including population vectors [7], maximum likelihood [8]–[10], and maximum *a posteriori* decoding [11]. There are significantly more decoding methods for continuous states which can be categorized based on linearity. Linear models include population vectors [12], multiple linear regressions [13], and the Kalman filter [14]. Though relatively simple, these models provide accurate estimation in practical motor cortical decoding. Moreover, due to their relatively low computational costs these methods have been successfully used in various closed-loop neural control experiments [1], [4], [5].

Recently a number of nonlinear methods have been developed which focus on accurately characterizing spiking activity. These methods include particle filters [15], [16], point process filters [17]–[19], mixture of trajectory models [20], nonlinear dynamic models [21], [22], neural networks [23], and hybrid filters [24]. Recent studies have also addressed the nonstationarity of neural signals in various frameworks. The population vector was modified by adding a supervised learning method for the parameters by Tillery *et al.* [25]. Gage *et al.* examined naive coadaptive cortical control using an adaptive Kalman filter where parameters in the model were updated over time [26]. Moreover, Eden *et al.* provided an adaptive point process filtering method to examine the dynamic representation of movement information [17].

Regardless of the structures of the above linear and nonlinear models, spiking activity of each neuron at each time is represented as a (stochastic) function of certain observed behavioral states, such as the hand position, and internal signals, such as the firing rates of all recorded neurons in a truncated history. However, neurally-controlled, muscle-executed hand movement is a complicated process. The spiking activity may be affected by various other factors such as a visually-presented target, sensory-motor transformations, muscular activity of the upper limb, and movement at joint angles of the shoulder and elbow. This spiking activity can also be affected by other conditions such as the ambience of the experimental environment, the comfort and anxiety of the subject, and even the subject's level of attention. While these states (internal or external) are often unobserved or even unobservable during experiments, they may play important roles in characterizing neurally-controlled movement.

As indicated in our recent study [19], there are many sources of variability that the standard state-space model is missing (and therefore the decoding performance of the standard model suffers). We developed a "common-input" model to include more of this excess variability in a specific, tractable model. Based on this idea, we propose to represent all these states as a multidimensional *hidden state* and add it to a state-space model. The system state in the new model includes two parts: one is the observable behavior state such as the hand state, and the

Manuscript received November 20, 2008; accepted April 13, 2009 First published June 02, 2009; current version published August 07, 2009. The work of W. Wu was supported by FSU Planning Grant. The work of J. E. Kulkarni was supported by Swartz Foundation Fellowship. The work of N. G. Hatsopoulos was supported by NIH-NINDS under Grant R01 NS45853. The work of L. Paninski was supported by an NSF CAREER award and a Sloan Research Fellowship.

W. Wu is with the Department of Statistics, Florida State University, Tallahassee, FL 32306 USA (e-mail: wwu@stat.fsu.edu).

J. E. Kulkarni is with the Center for Theoretical Neuroscience, Columbia University, New York, NY 10027 USA (e-mail: jk2619@columbia.edu).

N. G. Hatsopoulos is with the Department of Organismal Biology and Anatomy, University of Chicago, Chicago, IL 60637 USA (e-mail: nicho@uchicago.edu).

L. Paninski is with the Department of Statistics, Columbia University, New York, NY 10027 USA (e-mail: liam@stat.columbia.edu).

Digital Object Identifier 10.1109/TNSRE.2009.2023307

1534-4320/$26.00 © 2009 IEEE

other is the hidden state. The hidden state could, in principle, represent any kinematic, kinetic, or cognitive signal, but remains unobserved. The study is based on Bayesian state-space models which provide a coherent framework to characterize stochastic dynamic systems and have numerous applications in neuroscience. In this study, we use a Kalman filter which is the most classical state-space model. It adopts a linear Gaussian representation between spike trains and hand motion (the likelihood term), and a linear auto-regressive prior term for the hand state [14]. One important benefit of the linear model is that both parameter identification and system estimation of the model can be conducted using conventional, efficient methods. The new model assumes that the observed neural firing rate is linearly related to the hidden state. Furthermore, the dynamics of the hand state are allowed to impact the dynamics of the hidden state in a linear fashion, and vice versa. The parameters in the model can be identified by the conventional expectation-maximization (EM) algorithm. As the new model follows a linear Gaussian framework, the decoding can still be performed by the efficient Kalman filter algorithm. We test the method in two data sets recorded from two Macaque monkeys during a visuo-motor task.

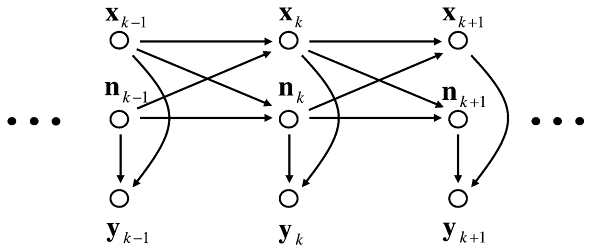

Fig. 1. Graphical model for the hidden-state-included Kalman filter. The neural firing rate, $\mathbf{y}_k$, is directly related to both the hand kinematics, $\mathbf{x}_k$, and the hidden state, $\mathbf{n}_k$. The dynamics of the hand kinematics and hidden state are both Markovian and impact each other over time.

II. Methods

A. Experimental Methods

Electrophysiological recording. The neural data used here were previously recorded and have been described elsewhere [27]. Briefly, silicon microelectrode arrays containing 100 platinized-tip electrodes (I2S Micro Implantable Systems, LLC, Salt Lake City, UT) were implanted in the arm area of primary motor cortex (MI) in two juvenile male macaque monkeys (Macaca mulatta). Signals were filtered, amplified (gain, 5000) and recorded digitally (14-bit) at 30 kHz per channel using a Cerebus acquisition system (I2S Micro Implantable Systems, LLC). Only waveforms (1.6 ms in duration) that crossed a threshold were stored and spike-sorted using Offline Sorter (Plexon Inc., Dallas, TX). Single units were manually extracted by the Contours and Templates methods. One data set was collected and analyzed from each monkey (124 and 125 distinct neural units recorded simultaneously, respectively). The firing rates of single cells were computed by counting the number of spikes within the previous 50 ms time window.

Task. The monkeys were operantly trained to perform a random target-pursuit (RTP) task by moving a cursor to targets via contralateral arm movements. The cursor and a sequence of seven targets (target size: 1 cm × 1 cm) appeared on a horizontal projection surface. At any one time, a single target appeared at a random location in the workspace, and the monkey was required to reach it within 2 s. As soon as the cursor reached the target, the target disappeared and a new target appeared in a new, pseudo-random location. After reaching the seventh target, the monkey was rewarded with a drop of water or juice. A new set of seven random targets was presented on each trial. The majority of trials were approximately 4–5 s in duration. In data set one, the first monkey successfully completed 550 trials, and in data set two, the second monkey completed 400 trials. The monkeys' horizontal hand positions were calculated and recorded using the forward kinematics equations [28] at a sampling rate of 500 Hz. To match time scales, the hand position were down-sampled every 50 ms and from this we computed velocity and acceleration using simple differencing. Recent studies indicated that the averaged optimal latency between firing activity in MI and hand movement is around 100 ms [12], [13]. Therefore, in all our analysis we compared the neural activity in a 50 ms bin with the instantaneous kinematics (position, velocity, and acceleration) of the arm measured 100 ms later (i.e., a 2 time bin delay).

B. Statistical Methods

We have previously characterized the relationship between the hand kinematics and firing rates using a Kalman filter model [14]. To incorporate the effect of other behavioral states, here we add a hidden state to the Kalman filter. The new model assumes that the observed neural firing rate is directly related to the hidden state. Furthermore, the dynamics of the hand state are allowed to impact the dynamics of the hidden state, and vice versa (see graphical model in Fig. 1). Let $\mathbf{x}_k = [x, y, v_x, v_y, a_x, a_y]_k^T$ represent x-position, y-position, x-velocity, y-velocity, x-acceleration, and y-acceleration of the hand at time $t_k = k\Delta t$ ($\Delta t = 50$ ms in our experiments), and $\mathbf{y}_k \in \Re^C$ represent a $C \times 1$ vector containing the firing rates at time t_k for C observed neurons. Also, let $\mathbf{n}_k$ represent a d-dimensional hidden state at time t_k which is always unknown. Now, the likelihood term, prior term, and initial condition of the hidden state in the new model can be formulated in the following three equations:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{G}\mathbf{n}_k + \mathbf{q}_k \tag{1}$$

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{n}_{k+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{x}_{k} \\ \mathbf{n}_{k} \end{pmatrix} + \mathbf{w}_k \tag{2}$$

$$\mathbf{n}_1 \sim N(\mu, \Sigma) \tag{3}$$

where $\mathbf{H} \in \Re^{C\times 6}$, $\mathbf{G} \in \Re^{C\times d}$, and $\mathbf{A} \in \Re^{(6+d)\times(6+d)}$ are the linear coefficient matrixes. The noise terms $\mathbf{q}_k, \mathbf{w}_k$ are assumed normally distributed with zero mean, i.e., $\mathbf{q}_k \sim N(0, \mathbf{Q}), \mathbf{Q} \in \Re^{C\times C}$, and $\mathbf{w}_k \sim N(0, \mathbf{W}), \mathbf{W} \in \Re^{(6+d)\times(6+d)}$. $\mathbf{q}_k, \mathbf{w}_k$ are also assumed independent of each other. In the initial normal distribution, the mean and covariance are $\mu \in \Re^d$ and $\Sigma \in \Re^{d\times d}$, respectively.

1) Model Identification: In (1)–(3), the parameters are $\theta = (\mathbf{H}, \mathbf{G}, \mathbf{Q}, \mathbf{A}, \mathbf{W}, \mu, \Sigma)$. We can estimate them using a training set where both kinematics, $\{\mathbf{x}_k\}$, and firing rates, $\{\mathbf{y}_k\}$, are known. The estimation is to maximize the marginal log-likelihood

$$\log p(\{\mathbf{x}_k, \mathbf{y}_k\}; \theta)$$

with respect to θ. Equations (1)–(3) follow a general state-space framework, and we use the conventional EM algorithm to identify the model [29]. The EM algorithm was first introduced to leaning partially unknown linear dynamic systems [30] and nonlinear dynamic systems [31]. A point process version of the EM algorithm was recently developed in the analysis of spike trains [32] and other behavioral data [33], [34].

The E-step. The EM algorithm is an iterative method. Assuming all the parameters, denoted as θ_i, are identified at the ith iteration, the E-step is to estimate the first and second order statistics of $\mathbf{n}_k$ conditioned on $\{\mathbf{x}_k, \mathbf{y}_k\}$ and θ_i. To this end, we at first partition $\mathbf{A}$ and $\mathbf{w}_k$ in the prior (2) following the order of hand kinematics and hidden state; that is, we let

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \text{ and } \mathbf{w}_k = \begin{pmatrix} \mathbf{w}_{1k} \\ \mathbf{w}_{2k} \end{pmatrix}.$$

where $\mathbf{A}_{11} \in \Re^{6\times 6}$, $\mathbf{A}_{12} \in \Re^{6\times d}$, $\mathbf{A}_{21} \in \Re^{d\times 6}$, and $\mathbf{A}_{22} \in \Re^{d\times d}$ are four submatrixes, and $\mathbf{w}_1 \in \Re^6$ and $\mathbf{w}_2 \in \Re^d$ are two subvectors. Then the prior equation can be written as

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{n}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{pmatrix} + \begin{pmatrix} \mathbf{w}_{1k} \\ \mathbf{w}_{2k} \end{pmatrix}. \quad (4)$$

Therefore, (1) and (2) can be reorganized as

$$\begin{pmatrix} \mathbf{y}_k - \mathbf{H}\mathbf{x}_k \\ \mathbf{x}_{k+1} - \mathbf{A}_{11}\mathbf{x}_k \end{pmatrix} = \begin{pmatrix} \mathbf{G} \\ \mathbf{A}_{12} \end{pmatrix} \mathbf{n}_k + \begin{pmatrix} \mathbf{q}_k \\ \mathbf{w}_{1k} \end{pmatrix} \quad (5)$$

$$\mathbf{n}_{k+1} = \mathbf{A}_{22}\mathbf{n}_k + \mathbf{A}_{21}\mathbf{x}_k + \mathbf{w}_{2k} \quad (6)$$

To separate $\mathbf{w}_{1k}$ and $\mathbf{w}_{2k}$, we need to assume independence between them; that is, we have

$$\begin{pmatrix} \mathbf{w}_{1k} \\ \mathbf{w}_{2k} \end{pmatrix} \sim N(0, \mathbf{W}), \mathbf{W} = \begin{pmatrix} \mathbf{W}_{11} & 0 \\ 0 & \mathbf{W}_{22} \end{pmatrix}$$

where $\mathbf{W}_{11} \in \Re^{6\times 6}$ and $\mathbf{W}_{22} \in \Re^{d\times d}$ are two submatrixes. Then the two noise terms in (5) and (6) also follow a normal distribution with

$$\begin{pmatrix} \mathbf{q}_k \\ \mathbf{w}_{1k} \end{pmatrix} \sim N\left(0, \begin{pmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{W}_{11} \end{pmatrix}\right) \text{ and } \quad \mathbf{w}_{2k} \sim N(0, \mathbf{W}_{22}).$$

Hence, (5) and (6) constitute a new Kalman filter model where $\{\mathbf{n}_k\}$ is the only system state. $\mathbf{A}_{21}\mathbf{x}_k$ can be looked as a linear control term in the new prior model. Using the Kalman smoother, we can calculate the distribution of $\mathbf{n}_k$ conditioned on $\{\mathbf{x}_k, \mathbf{y}_k\}$. See Appendix A for mathematical details.

The M-step. In the M-step, we update θ_i to get θ_{i+1} by maximizing

$$\mathrm{E}_{p(\{\mathbf{n}_k\} \mid \{\mathbf{x}_k, \mathbf{y}_k\}; \theta_i)} \log p(\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{n}_k\}; \theta)$$

with respect to the parameters θ. To simplify the notation, we use $p(\cdot \mid \cdots)$ to replace $p(\cdot \mid \{\mathbf{x}_k, \mathbf{y}_k\}; \theta_i)$. Then

$$\begin{aligned} &\mathrm{E}_{p(\{\mathbf{n}_k\} \mid \{\mathbf{x}_k, \mathbf{y}_k\}; \theta_i)} \log p(\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{n}_k\}; \theta) \\ &\quad = \int_{\{\mathbf{n}_k\}} p(\{\mathbf{n}_k\} \mid \{\mathbf{x}_k, \mathbf{y}_k\}; \theta_i) \\ &\qquad \times \log p(\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{n}_k\}; \theta) d\{\mathbf{n}_k\} \\ &\quad = \int_{\{\mathbf{n}_k\}} p(\{\mathbf{n}_k\} \mid \cdots) \\ &\qquad \times \log p(\{\mathbf{y}_k\} \mid \{\mathbf{x}_k, \mathbf{n}_k\}; \theta) d\{\mathbf{n}_k\} \\ &\qquad + \int_{\{\mathbf{n}_k\}} p(\{\mathbf{n}_k\} \mid \cdots) \log p(\{\mathbf{x}_k, \mathbf{n}_k\}; \theta) d\{\mathbf{n}_k\} \\ &\quad = \mathbf{E}_1 + \mathbf{E}_2 \end{aligned}$$

where $\mathbf{E}_1$ is a function of $\mathbf{H}, \mathbf{G}, \mathbf{Q}$, and $\mathbf{E}_2$ is a function of $\mathbf{A}, \mathbf{W}, \mu, \Sigma$. These parameters can be updated in closed-forms by maximizing $\mathbf{E}_1$ and $\mathbf{E}_2$, respectively. The details are shown in Appendix B.

2) Decoding: Decoding involves estimating the *state* of the hand at each time where only firing rates of neuronal ensemble are observed. The estimation is based on the new Kalman filter model with hidden states. One key benefit in the new model is that it keeps the desired linear Gaussian structure. Therefore, the efficient Kalman filter algorithm can be fully exploited [14], where both kinematics and hidden states can be jointly estimated.

The forward Kalman filter algorithm follows a recursive scheme. At first, let $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{n}}_k$ represent the estimated hand state and hidden state at each time t_k, respectively, and let $\mathbf{P}_k$ represent their error covariance. Initially, $\hat{\mathbf{x}}_1$ is set to the true hand state, and $\hat{\mathbf{n}}_1$ is set to μ. $\mathbf{P}_1$ is set to a block-diagonal matrix, where the first block is a 6×6 zero matrix, and the second block is Σ.

At each time t_k with $k \geq 2$, we use the prior model (2) to estimate the *a priori* state, $(\hat{\mathbf{x}}_k^-, \hat{\mathbf{n}}_k^-)$, and its error covariance matrix, $\mathbf{P}_k^-$, from the previous time t_{k-1}

$$\begin{pmatrix} \hat{\mathbf{x}}_k^- \\ \hat{\mathbf{n}}_k^- \end{pmatrix} = \mathbf{A} \begin{pmatrix} \hat{\mathbf{x}}_{k-1} \\ \hat{\mathbf{n}}_{k-1} \end{pmatrix} \quad (7)$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{W}. \quad (8)$$

Using the firing rate vector $\mathbf{y}_k$ and the likelihood model (1), we update the estimation by computing the *a posteriori* state and its error covariance matrix

$$\begin{pmatrix} \hat{\mathbf{x}}_k \\ \hat{\mathbf{n}}_k \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}_k^- \\ \hat{\mathbf{n}}_k^- \end{pmatrix} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k^- - \mathbf{G}\hat{\mathbf{n}}_k^-) \quad (9)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k(\mathbf{H} \;\; \mathbf{G}))\mathbf{P}_k^- \quad (10)$$

where $\mathbf{K}_k$ is the Kalman *gain* matrix, given by

$$\mathbf{K}_k = \mathbf{P}_k^-(\mathbf{H}\,\mathbf{G})^T((\mathbf{H}\,\mathbf{G})\mathbf{P}_k^-(\mathbf{H}\,\mathbf{G})^T + \mathbf{Q})^{-1}. \tag{11}$$

3) Target Conditioning: The accuracy of the hand-state estimates can be further improved by incorporating *a priori* information of the targets. The method is well described in [35], [36]. Briefly, let $\mathbf{z}_K$ denote a noise-contaminated observation of the final hand state $\mathbf{x}_K$; we are interested in computing $p(\mathbf{x}_k \,|\, \mathbf{x}_1, \mathbf{z}_K, \{\mathbf{y}_k\})$, where $\{\mathbf{y}_k\}$ denotes the neural firing-rate observations up to time k. Then

$$\begin{aligned} &p(\mathbf{x}_k \,|\, \mathbf{x}_1, \mathbf{z}_K, \{\mathbf{y}_k\}) \\ &\quad = (1/Z)p(\mathbf{x}_k, \{\mathbf{y}_k\}, \mathbf{z}_K \,|\, \mathbf{x}_1) \\ &\quad = (1/Z)p(\mathbf{x}_k, \{\mathbf{y}_k\} \,|\, \mathbf{x}_1)p(\mathbf{z}_K \,|\, \mathbf{x}_k, \mathbf{x}_1, \{\mathbf{y}_k\}) \\ &\quad = (1/Z)p(\mathbf{x}_k, \{\mathbf{y}_k\} \,|\, \mathbf{x}_1)\int p(\mathbf{z}_K \,|\, \mathbf{x}_K)p(\mathbf{x}_K \,|\, \mathbf{x}_k)d\mathbf{x}_K \end{aligned}$$

where Z is a normalizing constant. The key fact now is that all of the terms in the above equation may be computed quite efficiently. The term $p(\mathbf{x}_t, \{\mathbf{y}_k\} \,|\, \mathbf{x}_1)$ corresponds exactly to the forward sweep of the Kalman filter [14]. The remaining term $\int p(\mathbf{z}_K \,|\, \mathbf{x}_K)p(\mathbf{x}_K \,|\, \mathbf{x}_k)d\mathbf{x}_K$ turns out to be similarly easy to compute. Discrete-time recursions for this backwards probability and recursing backwards for $t = K-1, K-2, \ldots, 1)$ are well known; see [37] for the basic theory or [35], [36] for an implementation in the Gaussian model under consideration here.

III. Results

We examine the new model on two experimental data sets recorded from two Macaque monkeys. In each set, we use the first 100 trials of the RTP task as the training data. These training trials are about 7–8 min, which provides sufficient samples to estimate the parameters in the model. The remaining trials (450 in Monkey 1, and 300 in Monkey 2) are used as testing data to measure the effectiveness of the method in the decoding.

A. Data Representation

The EM algorithm identifies the model by maximizing the likelihood in the training data. A model with more sophisticated structure often has a better likelihood score on the training data, but may over-fit (i.e., lead to a reduced likelihood on the test data, and therefore generalize poorly). To examine the goodness-of-fit, we calculate the likelihood of each model in both training and testing data. The comparison is based on the normalized log-likelihood ratio (NLLR) with respect to the classical Kalman filter. Let L_d represent the marginal likelihood of the observed data under the model with a d-dimensional hidden state, and L_0, the likelihood under the classical Kalman filter model. Then the NLLR is calculated as

$$\frac{1}{N}\log_2\frac{L_d}{L_0} \tag{12}$$

where N denotes the sample size. With base 2, the NLLR is measured in the units of bits, and is actually a normalized version of the standard likelihood ratio, L_0/L_d. The NLLR equals zero when the likelihood of the Kalman filter model is compared to itself in the same dataset. To indicate the difference between training and testing sets, the NLLR of the Kalman filter between them can be calculated as

$$\frac{1}{N_{\text{train}}}\log_2 L_0^{\text{train}} - \frac{1}{N_{\text{test}}}\log_2 L_0^{\text{test}} \tag{13}$$

where $N_{\text{train}}(N_{\text{test}})$ and $L_0^{\text{train}}(L_0^{\text{test}})$ are sample size and likelihood in the training (testing) data, respectively. We expect a positive difference as the model should fit better in the training set.

TABLE I
NLLRs of the Kalman Filter and the New Models With Different Dimensions of the Hidden State

	training set 1	testing set 1	training set 2	testing set 2
KF	0	0 [12.67]	0	0 [8.24]
d=1	0.27 ($3 \times 10^3*$)	0.27	0.26 ($4 \times 10^3*$)	0.24
d=2	0.46 ($5 \times 10^3*$)	0.42	0.46 ($7 \times 10^3*$)	0.43
d=3	1.16 ($1 \times 10^4*$)	1.10	0.69 ($1 \times 10^4*$)	0.62

The NLLRs of both data sets are estimated and summarized in Table I where the dimension of the hidden state, d, varies from 1 to 3. The NLLR displays an increasing trend as a function of d. This is true for training and testing sets in both monkeys' data. These results show that: 1) including the hidden state does not introduce over-fit, but rather improves the representation of the neural and kinematic signals; 2) the new models with higher-dimensional hidden states provide a better representation than the simple Kalman model with no hidden states. Note that the Kalman filter and the hidden state models are actually nested (the former one can be looked as a latter model where the dimension of the hidden state is zero). Thus, the significance of the improvement in the new models can also be demonstrated using a standard Chi-square test. The NLLR (in the units of bits) of each model with respect to the Kalman filter is shown in the two training set columns in Table I. Numbers in the square brackets indicate the NLLR of the baseline Kalman filter between training and testing sets (13). Numbers in the parentheses indicate the standard likelihood ratio statistics between the new model (with each hidden state dimension) and the classical Kalman filter. Stars denote the improvement is statistically significant (Chi-square test, p-value < 0.01).

Note that we also tested higher dimensional cases for the hidden state (when $d > 3$). We found that the parameters in the model cannot be identified in the EM process; that is, the likelihood of the model remains nearly constant over the iterations. The dimension of the hidden state is, therefore, limited to 1, 2, and 3 in this study. In general, this dimension could depend on number of observed neurons, dimension of the behavioral state, and amount of training data.

In addition to the likelihood, alternative approaches can be used to compare the representation of each model. For example, we can measure the estimated noise covariance matrix $\mathbf{Q}$ (1). As including the hidden state in the model should account for some of the "over-dispersion" in the observed neural data, $\mathbf{Q}$ is expected to be "smaller" when the hidden state is included than that in the classical Kalman filter. Similarly, we can compare the "over-dispersion" in the kinematics which is described by the first 6×6 sub-matrix in the main diagonal of the noise covariance matrix $\mathbf{W}$. In the d-dimensional hidden state model,

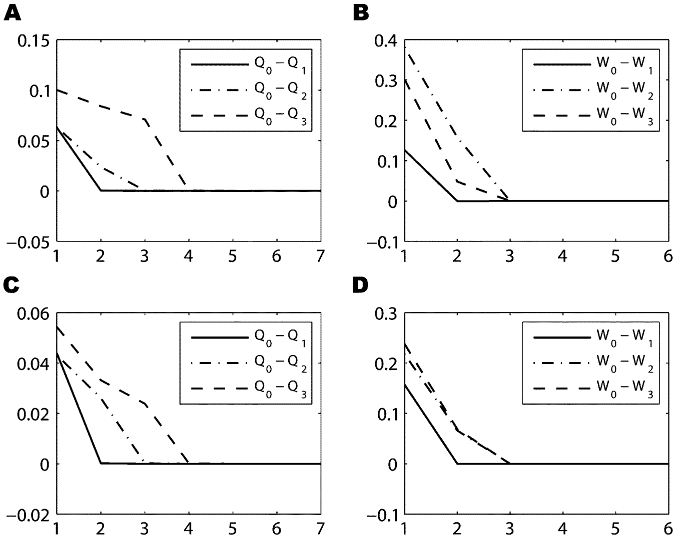

Fig. 2. (**A**) Eigenvalues of the matrixes $(\mathbf{Q}_0 - \mathbf{Q}_1)$, $(\mathbf{Q}_0 - \mathbf{Q}_2)$, and $(\mathbf{Q}_0 - \mathbf{Q}_3)$ in dataset 1. The eigenvalues are shown in descending order of the magnitude, and only the first 7 of the total 124 values are shown. (**B**) Eigenvalues of the matrixes $(\mathbf{W}_0 - \mathbf{W}_1)$, $(\mathbf{W}_0 - \mathbf{W}_2)$, and $(\mathbf{W}_0 - \mathbf{W}_3)$ in dataset 1. The eigenvalues are also shown in descending order of the magnitude. (**C**) and (**D**) same as (A) and (B) except for dataset 2.

TABLE II
DECODING ACCURACY OF THE CLASSICAL KALMAN FILTER AND THE NEW MODELS WITH DIFFERENT DIMENSIONS OF THE HIDDEN STATE

	Dataset 1	Dataset 2
Classical Kalman filter	7.6 (4.5)	8.2 (4.6)
1D hidden state	7.3* (3.8*)	7.8* (4.3*)
2D hidden state	6.7* (3.3*)	7.1* (3.8*)
3D hidden state	6.5* (3.3*)	6.9* (3.7*)

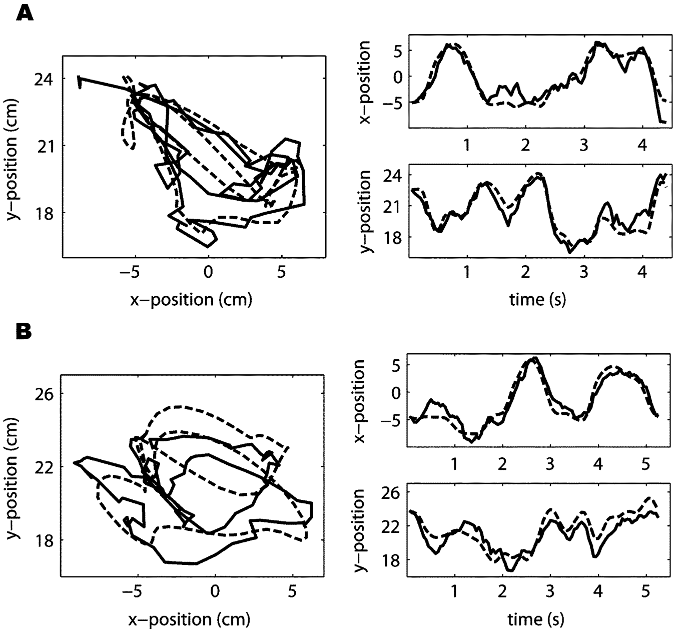

Fig. 3. (**A**) True hand trajectory (dashed) and reconstruction using the new Kalman filter model with a hidden state ($d = 3$) of an example trial from dataset 1. Left column: the trajectories in the 2-D working space. Right column: the trajectories by their x and y components. (**B**) Same as (A) except for an example trial from dataset 2.

we use $\mathbf{Q}_d$ and $\mathbf{W}_d$ to represent $\mathbf{Q}$ and the first 6×6 main-diagonal submatrix of $\mathbf{W}$, respectively. Likewise, Q_0 and W_0 are used for the classical Kalman filter. We find all the eigenvalues in both $(\mathbf{Q}_0 - \mathbf{Q}_d)$ and $(\mathbf{W}_0 - \mathbf{W}_d)$ are nonnegative for $d = 1, 2$, and 3 (Fig. 2); i.e., all the differences are semi-positive definite matrixes, and therefore the new models lower the variability of noise term in the likelihood equation. This variability study provides supporting evidence that the new models better represent the observed data.

B. Decoding With and Without Target Location Information

In the decoding stage, only firing rates $\{\mathbf{y}_k\}$ are observed. We use these rates to estimate kinematics $\{\mathbf{x}_k\}$ and hidden states $\{\mathbf{n}_k\}$ by the Kalman filter algorithm [(7)–(11)]. We measure the decoding accuracy by the common mean squared error (MSE, in the units of cm^2). The results from the two test sets are summarized in Table II. Numbers outside of parentheses indicate the MSEs when target location was not included in the decoding. Stars denote the improvement is statistically significant (Wilcoxon signed-rank test, p-value$<$ 0.01). It is found that the decoding accuracy is significantly improved when a hidden state is included in the model. Also, the degree of improvement increases when the dimension of the hidden state varies from 1 to 3. Two decoding examples are shown in Fig. 3.

In addition, we examined whether including the target location would have any effect on our results, since in many neural prosthetic applications the location of a set of possible endpoint target locations are known *a priori*. We used standard Kalman forward–backward methods [35], [36] to condition on the endpoint location in each trial; as expected, including this endpoint information significantly decreased the overall decoding MSE. In Table II, the numbers inside of parentheses indicate the MSEs when target location was included in the decoding. Once again, we find that decoding accuracy is improved when a hidden state is included in the Kalman model, and the degree of improvement increases with the dimension of the hidden state.

C. Analysis of the Hidden State

As emphasized above, the classical Kalman filter approach represents neural activity using only hand kinematics, which lacks descriptions of other behavioral states such as kinetic terms or attentional level. We have added a hidden state to the model to address this issue, and demonstrated that this extended model provides better decoding. Next we investigate whether the inferred hidden state is correlated in some systematic fashion with the behavioral task variables.

First, we measured the correlation between the hidden state and kinematics in the test data directly, where both the hand kinematics and hidden state were estimated from the firing rates (7)–(11). The averaged correlations (over all test trials) were calculated for each of the new models with $d = 1, 2$, and 3. We found that in dataset 1 approximately 39% of the correlation values are significantly different from 0 (beyond the significance level with $\alpha = 0.05$). The percentage is approximately 52% in dataset 2. These results clearly indicate the dependency between the hidden state and hand kinematics.

To summarize the overall strength of these multivariate correlations, we can simply measure the conditional mutual information $I(\mathbf{n}_k; \mathbf{x}_k \mid \{\mathbf{y}_{k'}\}_{k' \leq k})$ between the hidden state $\mathbf{n}_k$

and the behavioral vector $\mathbf{x}_k$, given the observations up to time t_k. After the new model is identified by the EM algorithm, the decoding error at each time step can be recursively measured without using testing data (8), (10), and (11). These matrixes converge to constant values when k is large (in practice, when $k > 20$). Let $\mathbf{P} = \lim_{k\to\infty} \mathbf{P}_k$, which describes the errors for estimated kinematics and hidden states. For an n-dimensional random variable with distribution $N(m, V)$, its entropy is

$$\frac{1}{2}\log[(2\pi e)^n \det(V)].$$

As the joint distribution of kinematics and the hidden state is normally distributed, we can calculate the mutual information values between them using $\mathbf{P}$. In dataset 1, the values are 0.11, 0.07, and 0.18 (bits) for $d = 1, 2,$ and 3, respectively. In dataset 2, the values are 0.36, 0.30, and 0.35 (bits). [Note that the mutual information is meaningful as it is a scale-invariant quantity; the entropy of the hidden state, on the other hand, is harder to interpret because the units of the hidden state are not specified in (1) and (2).]

D. Reduced Models

In the hidden-state-included Kalman filter model, we assume that the hidden state is directly related to both neural activity (1) and hand kinematics (2). Here we examine whether we could limit the effect of the hidden state to only one of these variables. Such a limitation would result in a reduced representation. One reduced model is built by removing the hidden state from the likelihood (1); that is, the model is composed of (2) and the following equation:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{q}_k.$$

Likewise, the other reduced model is built by removing the dependence of the hidden state and kinematics from the prior (2); that is, we let $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ in (4) be zero matrixes, and the model is composed of (1) and the following equations:

$$\mathbf{x}_{k+1} = \mathbf{A}_{11}\mathbf{x}_k + \mathbf{w}_{1k}$$
$$\mathbf{n}_{k+1} = \mathbf{A}_{22}\mathbf{n}_k + \mathbf{w}_{2k}.$$

We tested the above two reduced models in the two experimental data sets. It was found that the EM algorithm cannot fit either model; that is, the likelihood in the algorithm remains nearly a constant over iterations. This indicates that only including the hidden state in one equation will not make a reasonable fit. The hidden state should be directly related to both neural activity and dynamics of the hand state.

E. Nonstationarity Check

Finally, it is known that neural activity in motor cortex may be strongly nonstationary over time [38]. For example, our recent study indicates that firing rates of approximately 50% of the neurons in these two data sets have a significant linear trend (ascending or descending) over time [27]. It is natural to ask whether the improvements in decoding we have seen here is simply due to the ability of the hidden state to represent the nonstationarity of neural signals.

To this end, we calculate the linear trend in the hidden state in each test set estimated from the neural activity. It is found that the slope of each component is statistically not different from zero. This indicates that each dimension of the hidden state has either very weak or no trend over time. As an example, the 3-D hidden states in both datasets are shown in Fig. 4. No apparent trend is observed in each dimensional component. This suggests that the hidden state does not appear to represent the nonstationarity of neural signals.

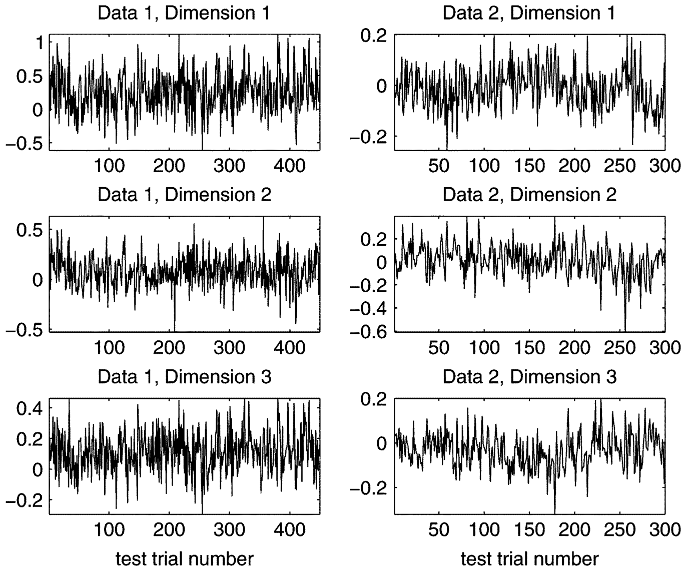

Fig. 4. Three components (three rows) of the estimated 3-D hidden state in two test data sets (two columns), where the hidden state is averaged as one value in each test trial.

IV. Discussion

Various internal and external states may play important roles during brain-controlled, muscle-executed hand movement. Since these states are often unobserved or even unobservable during recordings, they have not been used in characterizing spiking activity in the motor cortex. To overcome this problem, we propose to describe these important yet unknown states as one multidimensional hidden state. This hidden state could, in principle, represent any kinematic, kinetic, or cognitive signal. In this manuscript, we examine the hidden state in a linear state-space model which is a natural extension of the classical Kalman filter [14]. We propose a different approach by adding a hidden state in both likelihood and prior equations.

One important benefit of the linear model is that both parameter identification and system estimation of the model can be conducted using conventional, efficient methods such as the EM algorithm and Kalman filter algorithm. Our results indicate that the new method provides a better characterization of the firing rate data than does the classical Kalman filter. More importantly, it provides more accurate decoding (around 15% improvement) while preserving real-time efficiency. The standard Kalman filter has been successfully used in closed-loop neural control for animals [39] and human subjects [40]. We expect the improvement in this investigation could significantly impact the performance in the use of practical neural control.

The results in the new model suggest the hidden state may play a "common input" role [19] and the decoding improvement may result from the fact that the observation noise is reduced (the excess variability is being accounted for by the hidden state). Indeed, since the Kalman filter is a linear time-invariant (LTI) filter, one simple interpretation of our results is that it is useful to search over a larger class of LTI filters than that spanned by the classical Kalman filter, without searching over the class of *all* possible LTI filters (since in practice this unrestricted search results in overfitting; this is why direct linear regression approaches perform suboptimally).

Note that our major goal in this manuscript is to demonstrate that hidden noise terms can significantly improve the basic Kalman filter decoding method. We do not argue that linear Kalman filter methods are better than recently developed generalized linear model (GLM)-based methods [18], [19], and therefore we do not make comparisons between the linear and GLM-based methods here. Though accurate and efficient, the Kalman filter model assumes a simple linear, Gaussian model on the integer-valued spiking process. To better characterize the activity, a number of powerful models have been developed [41], [18], [19], [24]. These models often focus on the nonlinearity of the representation or the nonstationarity (dynamic variation) of the observed data. They have been shown to appropriately represent the neural activity and to accurately perform decoding in simulated and/or experimental data. These more powerful models often involve more complicated operations, and various approximations (such as linearization, Gaussian approximation, or Laplace approximation) are needed in order to make these methods efficient in practical use. Note that the proposed hidden state is entirely compatible with these nonlinear methods. We plan to characterize the spiking activity as a multivariate point process in a generalized linear model (GLM) where the conditional density for a neuron is a function of the hand state, spike history, and the hidden state [18], [19]. The identification and estimation in these models, however, will be significantly more challenging.

In summary, we expect that the proposed linear state-space model introduced here could potentially be a useful tool in prosthetic applications, and can be easily incorporated in current real-time, online experimental decoding experiments. In the future we plan to include hidden states in the more physiologically-accurate point-process filters [19], [42]. It would be greatly interesting to compare the computational complexity and accuracy of all these linear and nonlinear methods.

Appendix

A. The E-Step

To simplify the notation, we let $p(\cdot \mid \cdots) = p(\cdot \mid \{\mathbf{x}_k, \mathbf{y}_k\}; \theta_i)$ where θ_i represents the parameters at the ith iteration. Assuming the total number of time steps is K in training data, we can calculate

$$\alpha_1 = \sum_{k=1}^{K} \left[\int_{\mathbf{n}_k} \mathbf{n}_k p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k \right] \mathbf{x}_k^T$$

$$\alpha_2 = \sum_{k=1}^{K} \left[\int_{\mathbf{n}_k} \mathbf{n}_k p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k \right] \mathbf{y}_k^T$$

$$\alpha_3 = \sum_{k=1}^{K} \int_{\mathbf{n}_k} \mathbf{n}_k \mathbf{n}_k^T p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k$$

$$\alpha_4 = \sum_{k=1}^{K-1} \left[\int_{\mathbf{n}_k} \mathbf{n}_k p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k \right] \mathbf{x}_{k+1}^T$$

$$\alpha_5 = \sum_{k=1}^{K-1} \left[\int_{\mathbf{n}_{k+1}} \mathbf{n}_{k+1} p(\mathbf{n}_{k+1} \mid \cdots) d\mathbf{n}_{k+1} \right] \mathbf{x}_k^T$$

$$\alpha_6 = \sum_{k=1}^{K-1} \left[\int_{\mathbf{n}_k} \mathbf{n}_k p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k \right] \mathbf{x}_k^T$$

$$\alpha_7 = \sum_{k=1}^{K-1} \int_{\mathbf{n}_k+1} \int_{\mathbf{n}_k} \mathbf{n}_{k+1} \mathbf{n}_k^T p(\mathbf{n}_k, \mathbf{n}_{k+1} \mid \cdots) d\mathbf{n}_k d\mathbf{n}_{k+1}$$

$$\alpha_8 = \sum_{k=1}^{K-1} \int_{\mathbf{n}_k} \mathbf{n}_k \mathbf{n}_k^T p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k$$

$$\alpha_9 = \sum_{k=2}^{K} \int_{\mathbf{n}_k} \mathbf{n}_k \mathbf{n}_k^T p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k$$

$$\alpha_{10} = \sum_{k=1}^{K-1} \left[\int_{\mathbf{n}_{k+1}} \mathbf{n}_{k+1} p(\mathbf{n}_{k+1} \mid \cdots) d\mathbf{n}_{k+1} \right] \mathbf{x}_{k+1}^T$$

and

$$\begin{aligned}\beta_1 &= \sum_{k=1}^{K-1} \int_{\mathbf{n}_{k+1}} \int_{\mathbf{n}_k} \begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{n}_{k+1} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{pmatrix}^T \\ &\quad \times p(\mathbf{n}_k, \mathbf{n}_{k+1} \mid \cdots) d\mathbf{n}_k d\mathbf{n}_{k+1} \\ &= \begin{pmatrix} \sum_{k=1}^{K-1} \mathbf{x}_{k+1}\mathbf{x}_k^T & \alpha_4^T \\ \alpha_5 & \alpha_7 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\beta_2 &= \sum_{k=1}^{K-1} \int_{\mathbf{n}_k} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{pmatrix}^T p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k \\ &= \begin{pmatrix} \sum_{k=1}^{K-1} \mathbf{x}_k\mathbf{x}_k^T & \alpha_6^T \\ \alpha_6 & \alpha_8 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\beta_3 &= \sum_{k=1}^{K-1} \int_{\mathbf{n}_{k+1}} \begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{n}_{k+1} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{n}_{k+1} \end{pmatrix}^T \\ &\quad \times p(\mathbf{n}_{k+1} \mid \cdots) d\mathbf{n}_{k+1} \\ &= \begin{pmatrix} \sum_{k=2}^{K} \mathbf{x}_k\mathbf{x}_k^T & \alpha_{10}^T \\ \alpha_{10} & \alpha_9 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\beta_4 &= \sum_{k=1}^{K-1} \int_{\mathbf{n}_{k+1}} \int_{\mathbf{n}_k} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{n}_{k+1} \end{pmatrix}^T \\ &\quad \times p(\mathbf{n}_k, \mathbf{n}_{k+1} \mid \cdots) d\mathbf{n}_k d\mathbf{n}_{k+1} \\ &= \beta_1^T.\end{aligned}$$

B. The M-Step

From Section II-B1, recall

$$\begin{aligned}\mathbf{E}_1 &= \int_{\{\mathbf{n}_k\}} p(\{\mathbf{n}_k\} \mid \cdots) \log \\ &\quad \times p(\{\mathbf{y}_k\} \mid \{\mathbf{x}_k, \mathbf{n}_k\}; \theta) d\{\mathbf{n}_k\} \\ &= \sum_{k=1}^{K} \int_{\mathbf{n}_k} \log p(\mathbf{y}_k \mid \mathbf{x}_k, \mathbf{n}_k; \theta) p(\mathbf{n}_k \mid \cdots) d\mathbf{n}_k.\end{aligned}$$

Using the likelihood (1)

$$\begin{aligned}&\log p(\mathbf{y}_k \mid \mathbf{x}_k, \mathbf{n}_k; \theta)\\ &\quad= -\frac{1}{2}((\mathbf{y}_k - \mathbf{H}\mathbf{x}_k - \mathbf{G}\mathbf{n}_k)^T \mathbf{Q}^{-1}(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k - \mathbf{G}\mathbf{n}_k)\\ &\qquad + \log\det \mathbf{Q}) + \text{constant}.\end{aligned}$$

This indicates that $\mathbf{E}_1$ only contains parameters $\mathbf{H}, \mathbf{G}$, and $\mathbf{Q}$. Let partial derivatives

$$\frac{\partial \mathbf{E}_1}{\partial \mathbf{H}} = \frac{\partial \mathbf{E}_1}{\partial \mathbf{G}} = \frac{\partial \mathbf{E}_1}{\partial \mathbf{Q}} = 0$$

we have

$$\begin{aligned}(\mathbf{H}\quad \mathbf{G}) &= \left(\textstyle\sum_{k=1}^K \mathbf{y}_k\mathbf{x}_k^T \quad \alpha_2^T\right)\begin{pmatrix}\sum_{k=1}^K \mathbf{x}_k\mathbf{x}_k^T & \alpha_1^T\\ \alpha_1 & \alpha_3\end{pmatrix}^{-1}\\ \mathbf{Q} &= \frac{1}{K}\left(\sum_{k=1}^K \mathbf{y}_k\mathbf{y}_k^T - \mathbf{H}\sum_{k=1}^K \mathbf{x}_k\mathbf{y}_k^T - \mathbf{G}\alpha_2\right).\end{aligned}$$

Similarly

$$\begin{aligned}\mathbf{E}_2 &= \int_{\{\mathbf{n}_k\}} p(\{\mathbf{n}_k\} \mid \cdots)\log p(\{\mathbf{x}_k, \mathbf{n}_k\}; \theta)d\{\mathbf{n}_k\}\\ &= \int_{\mathbf{n}_1} \log p(\mathbf{x}_1, \mathbf{n}_1; \theta)p(\mathbf{n}_1 \mid \cdots))dn_1\\ &\quad + \sum_{k=1}^{K-1}\int_{\mathbf{n}_{k+1}}\int_{\mathbf{n}_k} \log p(\mathbf{x}_{k+1}, \mathbf{n}_{k+1} \mid \mathbf{x}_k, \mathbf{n}_k; \theta)\\ &\quad \times p(\mathbf{n}_k, \mathbf{n}_{k+1} \mid \cdots)d\mathbf{n}_k d\mathbf{n}_{k+1}.\end{aligned}$$

Using the prior equation and initial condition (2) and (3)

$$\begin{aligned}&\log p(\mathbf{x}_1, \mathbf{n}_1; \theta)\\ &\quad= \log p(\mathbf{n}_1; \theta) + \text{constant}\\ &\quad= -\frac{1}{2}(\log\det\Sigma\\ &\qquad + (\mathbf{n}_1 - \mu)^T\Sigma^{-1}(\mathbf{n}_1 - \mu)) + \text{constant}.\end{aligned}$$

and

$$\begin{aligned}&\log p(\mathbf{x}_{k+1}, \mathbf{n}_{k+1} \mid \mathbf{x}_k, \mathbf{n}_k; \theta)\\ &\quad= -\frac{1}{2}\left(\log\det \mathbf{W}\right.\\ &\qquad + \left(\begin{pmatrix}\mathbf{x}_{k+1}\\ \mathbf{n}_{k+1}\end{pmatrix} - \mathbf{A}\begin{pmatrix}\mathbf{x}_k\\ \mathbf{n}_k\end{pmatrix}\right)^T \mathbf{W}^{-1}\\ &\qquad \times \left.\left(\begin{pmatrix}\mathbf{x}_{k+1}\\ \mathbf{n}_{k+1}\end{pmatrix} - \mathbf{A}\begin{pmatrix}\mathbf{x}_k\\ \mathbf{n}_k\end{pmatrix}\right)\right) + \text{constant}.\end{aligned}$$

This indicates that $\mathbf{E}_2$ only contains parameters $\mathbf{A}, \mathbf{W}, \mu$, and Σ. Let partial derivatives

$$\frac{\partial \mathbf{E}_2}{\partial \mathbf{A}} = \frac{\partial \mathbf{E}_2}{\partial \mathbf{W}} = \frac{\partial \mathbf{E}_2}{\partial \mu} = \frac{\partial \mathbf{E}_2}{\partial \Sigma} = 0$$

we have

$$\begin{aligned}\mathbf{A} &= \beta_1\beta_2^{-1}\\ \mathbf{W} &= \frac{1}{K-1}(\beta_3 - \mathbf{A}\beta_4)\\ \mu &= \int_{\mathbf{n}_1} \mathbf{n}_1 p(\mathbf{n}_1 \mid \cdots)d\mathbf{n}_1\\ \Sigma &= \int_{\mathbf{n}_1} \mathbf{n}_1\mathbf{n}_1^T p(\mathbf{n}_1 \mid \cdots)d\mathbf{n}_1\\ &\quad - \int_{\mathbf{n}_1} \mathbf{n}_1 p(\mathbf{n}_1 \mid \cdots)d\mathbf{n}_1 \int_{\mathbf{n}_1} \mathbf{n}_1^T p(\mathbf{n}_1 \mid \cdots)d\mathbf{n}_1.\end{aligned}$$

Note that μ and Σ only depend on values at one step. Hence, their estimation may not be robust. In practical data analysis, they can simply be replaced with zero matrixes. Such change has little effect on the decoding performance [14].

ACKNOWLEDGMENT

The authors would like to thank S. Francis, D. Paulsen, and J. Reimer for training the monkeys and collecting the data.

REFERENCES

[1] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS, Biol.*, vol. 1, no. 2, pp. 001–016, 2003.

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

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

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

[5] D. Taylor, S. H. Tillery, and A. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.

[6] J. Wessberg, C. Stambaugh, J. Kralik, L. M. Beck, P. J. Chapin, J. Kim, S. Biggs, M. Srinivasan, and M. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, 2000.

[7] A. Georgopoulos, A. Schwartz, and R. Kettner, "Neural population coding of movement direction," *Science*, vol. 233, pp. 1416–1419, 1986.

[8] N. G. Hatsopoulos, J. Joshi, and J. G. O'Leary, "Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles," *J. Neurophysiol.*, vol. 92, pp. 1165–1174, 2004.

[9] C. Kemere, K. V. Shenoy, and T. H. Meng, "Model-based neural decoding of reaching movements: A maximum likelihood approach," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 925–932, Jun. 2004.

[10] K. V. Shenoy, D. Meeker, S. Cao, S. A. Kureshi, B. Pesaran, C. A. Buneo, A. P. Batista, P. P. Mitra, J. W. Burdick, and R. A. Andersen, "Neural prosthetic control signals from plan activity," *NeuroReport*, vol. 14, no. 4, pp. 591–597, Mar. 2003.

[11] J. Hu, J. Si, B. P. Olson, and J. He, "Feature detection in motor cortical spikes by principal component analysis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 256–262, Sep. 2005.

[12] D. Moran and A. Schwartz, "Motor cortical representation of speed and direction during reaching," *J. Neurophysiol.*, vol. 82, no. 5, pp. 2676–2692, 1999.

[13] L. Paninski, M. Fellows, N. Hatsopoulos, and J. P. Donoghue, "Spatiotemporal tuning of motor cortical neurons for hand position and velocity," *J. Neurophysiol.*, vol. 91, pp. 515–532, 2004.

[14] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a Kalman filter," *Neural Computat.*, vol. 18, no. 1, pp. 80–118, 2006.

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

[16] Y. Gao, M. J. Black, E. Bienenstock, S. Shoham, and J. P. Donoghue, , T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., "Probabilistic inference of hand motion from neural activity in motor cortex," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 213–220.

[17] U. Eden, W. Truccolo, M. R. Fellows, J. P. Donoghue, and E. M. Brown, "Reconstruction of hand movement trajectories from a dynamic ensemble of spiking motor cortical neurons," in *Proc. IEEE Eng. Med. Biol. Soc.*, Sep. 2004, pp. 4017–4020.

[18] W. Truccolo, U. Eden, M. Fellows, J. Donoghue, and E. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble and extrinsic covariate effects," *J. Neurophysiol.*, vol. 93, pp. 1074–1089, 2005.

[19] J. E. Kulkarni and L. Paninski, "Common-input models for multiple neural spike-train data," *Network: Computat. Neural Syst.*, vol. 18, pp. 375–407, 2007.

[20] B. M. Yu, C. Kemere, G. Santhanam, A. Afshar, S. I. Ryu, T. H. Meng, M. Sahani, and K. V. Shenoy, "Mixture of trajectory models for neural decoding of goal-directed movements," *J. Neurophysiol.*, vol. 97, pp. 3763–3780, 2007.

[21] B. M. Yu, K. V. Shenoy, and M. Sahani, "Expectation propagation for inference in non-linear dynamical models with poisson observations," in *Proc. IEEE Nonlinear Stat. Signal Process. Workshop*, Sep. 2006, pp. 83–86.

[22] B. M. Yu, J. P. Cunningham, K. V. Shenoy, and M. Sahani, "Neural decoding of movements: From linear to nonlinear trajectory models," in *Proc. Int. Conf. Neural Inf. Process.*, 2007, pp. 586–595.

[23] J. C. Sanchez, D. Erdogmus, J. C. Principe, J. Wessberg, and M. A. L. Nicolelis, "Interpreting spatial and temporal neural activity through a recurrent neural network brain machine interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 2, pp. 213–219, Jun. 2005.

[24] L. Srinivasan, U. T. Eden, S. K. Mitter, and E. N. Brown, "General-purpose filter design for neural prosthetic devices," *J. Neurophysiol.*, vol. 98, pp. 2456–2475, 2007.

[25] S. I. H. Tillery, D. M. Taylor, and A. B. Schwartz, "Training in cortical control of neuroprosthetic devices improves signal extraction from small neuronal ensembles," *Rev. Neurosci.*, vol. 14, pp. 107–119, 2003.

[26] G. J. Gage, K. A. Ludwig, K. J. Otto, E. L. Ionides, and D. R. Kipke, "Naive coadaptive cortical control," *J. Neural Eng.*, vol. 2, pp. 52–63, 2005.

[27] W. Wu and N. Hatsopoulos, "Real-time decoding of non-stationary neural activity in motor cortex," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 3, pp. 213–222, Jun. 2008.

[28] S. H. Scott, "Apparatus for measuring and perturbing shoulder and elbow joint positions and torques during reaching," *J. Neurosci. Methods*, vol. 89, pp. 119–127, 1999.

[29] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Series B*, vol. 39, pp. 1–38, 1977.

[30] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *J. Time Series Anal.*, vol. 3, pp. 253–264, 1982.

[31] S. Haykin, *Kalman Filtering and Neural Networks*. New York: Wiley, 2001.

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

[33] A. C. Smith, L. M. Frank, S. Wirth, M. Yanike, D. Hu, Y. Kubota, A. M. Graybiel, W. Suzuki, and E. N. Brown, "Dynamic analysis of learning in behavioral experiments," *J. Neurosci.*, vol. 24, pp. 447–461, 2004.

[34] A. C. Smith, M. R. Stefani, B. Moghaddam, and E. N. Brown, "Analysis and design of behavioral experiments to characterize population learning," *J. Neurophysiol.*, vol. 93, pp. 1776–1792, 2005.

[35] L. Srinivasan, U. T. Eden, A. S. Willsky, and E. N. Brown, "A state-space analysis for reconstruction of goal-directed movements using neural signals," *Neural Computat.*, vol. 18, pp. 2465–2494, 2006.

[36] J. E. Kulkarni and L. Paninski, "State-space decoding of goal directed movement," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 78–86, Jan. 2007.

[37] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[38] S. P. Kim, F. Wood, M. Fellows, J. P. Donoghue, and M. J. Black, "Statistical analysis of the non-stationarity of neural population codes," in *1st IEEE/RAS-EMBS Int. Conf. Biomed. Robotics Biomechatron.*, Feb. 2006, pp. 295–299.

[39] W. Wu, A. Shaikhouni, J. P. Donoghue, and M. J. Black, "Closed-loop neural control of cursor motion using a Kalman filter," in *Proc. IEEE Eng. Med. Biol. Soc.*, Sep. 2004, pp. 4126–4129.

[40] S. P. Kim, J. Simeral, L. Hochberg, J. P. Donoghue, G. Friehs, and M. J. Black, "Multi-state decoding of point-and-click control signals from motor cortical activity in a human with tetraplegia," in *3rd IEEE EMBS Conf. Neural Eng.*, May 2007, pp. 486–489.

[41] U. Eden, L. Frank, R. Barbieri, V. Solo, and E. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Computat.*, vol. 16, pp. 971–988, 2004.

[42] B. M. Yu, A. Afshar, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, "Extracting dynamical structure embedded in neural activity," in *Adv. Neural Inf. Process. Syst.*, 2006, pp. 1545–1552.

Wei Wu received the B.S. degree in applied mathematics from the University of Science and Technology of China, Hefei, China, in 1998, and the Ph.D. degree in applied mathematics from Brown University, Providence, RI, in 2004.

He joined the faculty at Florida State University in 2006, where he is an Assistant Professor in the Department of Statistics and an Associate Faculty Member in the Program of Neuroscience. His research explores statistical methods for computational neuroscience and neural engineering.

Jayant E. Kulkarni received the B.Tech. degree from the Indian Institute of Technology, Madras, in 1999 and the doctoral degree from Cornell University, Ithaca, NY, in 2005.

His research interests include robust control of periodic systems, optimal control, and computational neuroscience.

Nicholas G. Hatsopoulos received the B.A. degree in physics from Williams College, Williamstown, MA, in 1984, and the M.S. degree in psychology and the Ph.D. degree in cognitive science from Brown University, Providence, RI, in 1991 and 1992, respectively.

He joined the faculty at the University of Chicago in 2002 and is currently an Associate Professor in the Department of Organismal Biology and Anatomy and is Chairman of the Committee on Computational Neuroscience. His research focuses on the neural coding of motor behavior in large cortical ensembles and on the development of brain–machine interfaces (BMIs). In 2001, he co-founded a company, Cyberkinetics Neurotechnology Systems, Inc. which developed BMI technology to assist people with severe motor disabilities.

Liam Paninski received the Ph.D. degree in neural science from New York University, in 2003.

He is currently an Associate Professor in the Department of Statistics at Columbia University. His group is interested in statistical modeling and analysis of neural data.