Feedback-Controlled Parallel Point Process Filter for Estimation of Goal-Directed Movements From Neural Signals

Maryam M. Shanechi, *Member, IEEE*, Gregory W. Wornell, *Fellow, IEEE*, Ziv M. Williams, and Emery N. Brown, *Fellow, IEEE*

***Abstract*—Real-time brain–machine interfaces have estimated either the target of a movement, or its kinematics. However, both are encoded in the brain. Moreover, movements are often goal-directed and made to reach a target. Hence, modeling the goal-directed nature of movements and incorporating the target information in the kinematic decoder can increase its accuracy. Using an optimal feedback control design, we develop a recursive Bayesian kinematic decoder that models goal-directed movements and combines the target information with the neural spiking activity during movement. To do so, we build a prior goal-directed state-space model for the movement using an optimal feedback control model of the sensorimotor system that aims to emulate the processes underlying actual motor control and takes into account the sensory feedback. Most goal-directed models, however, depend on the movement duration, not known *a priori* to the decoder. This has prevented their real-time implementation. To resolve this duration uncertainty, the decoder discretizes the duration and consists of a bank of parallel point process filters, each combining the prior model of a discretized duration with the neural activity. The kinematics are computed by optimally combining these filter estimates. Using the feedback-controlled model and even a coarse discretization, the decoder significantly reduces the root mean square error in estimation of reaching movements performed by a monkey.**

***Index Terms*—Brain–machine interfaces (BMIs), motor control, neural decoding, optimal feedback control, point processes.**

Manuscript received October 15, 2011; revised April 23, 2012; accepted September 07, 2012. Date of publication October 02, 2012; date of current version January 04, 2013. This work was presented in part at the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP), Dallas, Texas, March, 2010.

M. M. Shanechi was with the Department of Electrical Engineering and Computer Science (EECS), Massachusetts Institute of Technology, Cambridge, MA 02139 USA. She is currently with the EECS Department, University of California, Berkeley, CA 94720 USA, and with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: shanechi@cornell.edu).

G. W. Wornell is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail:gww@mit.edu).

Z. M. Williams is with the Neurosurgery Department, Massachusetts General Hospital, Boston, MA 02114 USA (e-mail: zwilliams1@partners.org).

E. N. Brown is with the Department of Brain and Cognitive Sciences, the Institute of Medical Engineering and Sciences, and the Harvard-MIT Health Sciences and Technology Program, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is also with the Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital and Harvard Medical School, Boston, MA 02114 USA (e-mail: enb@neurostat.mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNSRE.2012.2221743

I. Introduction

Information about various states of an intended movement is encoded in the neural activity from the motor cortical areas. Peri-movement activity, i.e., the activity around the time of movement, in the primary motor cortex, posterior parietal cortex (PPC), and dorsal premotor cortex (PMd) is related to the movement kinematics such as direction, velocity, position, and acceleration (see e.g., [1]–[5]). Also neural activity from the PPC and the premotor cortex is related to the intended target of the movement prior to movement initiation (see e.g., [6]–[8]). Brain–machine interfaces (BMIs) work by recording these neural activities such as the ensemble spiking activity, mapping these activities to the intended behavior also known as decoding, and using the decoded signal to control a device such as a robotic arm or computer cursor [5], [9]–[26].

Many typical tasks—such as reaching for a book—involve performing goal-directed movements. Most work on real-time BMIs have focused on individually decoding either the movement's kinematics to reconstruct its trajectory [5], [10]–[22], or the movement's intended reach goal or target [23], [24]. However, designing the kinematic decoder to take into account the goal-directed nature of such movements could improve their estimation accuracy. Additionally, when both target and kinematic related neural activities are recorded, combining them in the decoding algorithm could result in more accurate estimation of goal-directed movements. In particular, it has been shown that information about the intended target location can be decoded reliably from neural activity prior to movement [23], [24]. Hence, the main question is how to design a kinematic decoder that can model the goal-directed nature of movements and moreover combine the target information with the peri-movement activity to improve the accuracy of trajectory estimation. One way to do so is to construct a prior goal-directed state-space model for the kinematics that can incorporate the target information. This prior model can in turn be used in a recursive Bayesian decoder to estimate the kinematics [18], [27]–[29].

Prior work have built this goal-directed model for reaching movements by conditioning a linear Gaussian state-space model, also known as a random-walk model, on being at the target at a *known* arrival time [28], [29] or using a linear feedforward controlled (i.e., not taking into account the sensory feedback) model again assuming a known arrival time [30]. Alternatively, goal-directed prior models have been built by using a training data set, for example fitting a linear Gaussian state-space model for a given target to empirical reaches to its location [27] or fitting a single model for arbitrary targets based on a data set of reaches to their locations [18]. Since the goal of the decoder is to estimate natural movement, a prior model that aims to closely emulate the sensorimotor processing underlying actual motor control could result in more accurate estimation of intended movement. Hence, we propose a prior model for goal-directed movements based on the optimal feedback control theory of the sensorimotor system [31], [32], used to interpret its function. This results in a feedback-controlled prior state-space model for the kinematics that not only exploits the information about the target location but also models the real-time sensory feedback. Also, unlike the models in [18], [27], this model does not rely on a training data set and can hence easily extend to different target locations without requiring a set of empirical reaches to them. Finally, by using the optimal feedback control formulation, it can generalize to tasks other than reaching movements, if desired, by simply quantifying the goals of such tasks as the cost function in this formulation. We derive this optimal feedback-controlled prior model as the first component to develop a recursive Bayesian decoder for goal-directed movements.

Regardless of the approach taken to incorporate the target information into the prior state-space model, most goal-directed state-space models are inherently dependent on the arrival time at the target or equivalently on the movement duration. For example there is much more constraint on the movement kinematics close to the arrival time compared to far from it, as in the former case the trajectory soon needs to reach the intended target. Also the assumption of known movement duration used in [28] and [30] prevents their implementation in real time since the duration is not known *a priori* to a real-time decoder observing only the neural signal. Hence, to be implementable in a BMI, in addition to using a goal-directed state-space model, a practical decoder needs to address this uncertainty in movement duration.

We introduced a framework to resolve this duration uncertainty in [33] by discretizing the movement duration and then optimally combining the kinematic estimates corresponding to the discretized durations based on the neural data. A similar approach using a discretized set of durations was subsequently used in [34] to resolve the duration uncertainty of the goal-directed prior model developed in [28] for estimation of simulated reaching trajectories. In this paper, as the second component of our decoder, we develop our framework to resolve the duration uncertainty for any goal-directed state-space model (including our feedback-controlled one) in more detail. Additionally, we examine the loss in estimation accuracy due to the unknown duration and the effect of the number of discretization points on this loss. Moreover, we discuss two possible approaches to treating a state-space model of a discretized duration after its arrival time is reached and their effect on decoder accuracy. To provide a more realistic validation, instead of testing the algorithm on simulated movement trajectories as done in, e.g., [28], [34], we test it on real reaching movements performed by a rhesus monkey.

Combining these two components, our decoder can be used in a real-time BMI to improve the estimation accuracy of goal-directed movements. Our focus in this paper is on the detailed derivation of the decoder and the characterization of its performance for estimation of goal-directed movements performed by a monkey from simulated spiking activity. The simulated neural data is obtained based on the monkey's trajectories using a rigorously validated model of neural activity in the primary motor cortex [2], [35] that relates it to the intended movement parameters. Using thousands of simulated trials, this analysis allows us to characterize the decoder performance under various conditions and give guidelines for its implementation and parameter selection. An implementation of this decoder for combining target and kinematic related activities in a real-time BMI that uses an optimal feedback control design is presented in our other work [9], [36], [37].

II. Methods

We denote the sequence of kinematic states by $\mathbf{x}_0, \ldots, \mathbf{x}_t$ and the spiking activity of the ensemble of C neurons by $\mathbf{N}_1, \ldots, \mathbf{N}_t$ where $\mathbf{N}_t = (\mathsf{N}_t^1, \ldots, \mathsf{N}_t^C)$ is the binary spike events of the C neurons at time t. We model the spiking activity of each neuron as a point process. Assuming that the neural spiking observations from the C neurons are conditionally independent given the state, the point process observation model is given by [35], [38]

$$p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{H}_t) = \prod_c \left(\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)\Delta\right)^{\mathsf{N}_t^c} e^{-\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)\Delta} \quad (1)$$

where $\mathbf{H}_t^c = \mathsf{N}_{1:t-1}^c, \mathbf{H}_t = \mathbf{N}_{1:t-1} = \mathbf{H}_t^{1:C}$ model the history dependence of the spiking activity, Δ is the time increment, and $\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)$ is the modeled instantaneous firing rate or equivalently the conditional intensity function of the cth neuron at time t. Note that we have explicitly indicated the encoding of the kinematic states, $\mathbf{x}_t$, in the conditional intensity function (the form of which depends on the kinematics tuning model and will be discussed further in Section III). If the observation model further assumes that the spiking activity is not history dependent, i.e., that the observation at each time step is conditionally independent of the observations in the previous time steps given the kinematic state, then the observation model simplifies to

$$p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{H}_t) = p(\mathbf{N}_t|\mathbf{x}_t) = \prod_c \left(\lambda_c(t|\mathbf{x}_t)\Delta\right)^{\mathsf{N}_t^c} e^{-\lambda_c(t|\mathbf{x}_t)\Delta}.$$

The goal of the decoder is to causally calculate the state posterior density, $p(\mathbf{x}_t|\mathbf{N}_{1:t})$, from the neural observations. We first develop the feedback-controlled prior state-space model. We then derive the decoder that combines this prior with the point process observation model and also resolves the duration uncertainty inherent to goal-directed state-space models.

A. Optimal Feedback-Controlled Prior State-Space Model

We develop an optimal feedback-controlled prior state-space model for the kinematics of goal-directed movements, which incorporates the target information and models the real-time sensory feedback. This model is inspired by the optimal feedback control theory of the sensorimotor system that is used to interpret its function [31], [32]. In this framework, each task is performed to accomplish a goal during which there is sensory feedback about the state of the system. Based on the desired goal,

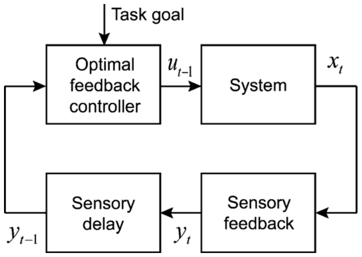


Fig. 1. Optimal feedback control framework. An optimal feedback control framework is used to build a state-space model for goal-directed movements. In this framework each task is performed to accomplish a goal during which there is real-time sensory feedback, $\mathbf{y}_t$, about the state of the system, $\mathbf{x}_t$. Based on the intended goal and the sensory feedback about the state of the system, the brain (controller) decides on a plan of action or control signal, $\mathbf{u}_t$, and controls the system.

the sensory feedback (for example vision and proprioception) about the current state of the system, and the intended time to accomplish the goal, the subject (controller) decides on the next control signal (for example muscle activation) to generate the movement. As a result, the subject can make real-time adjustments in the control signal based on the feedback to improve behavior. This framework is illustrated in Fig. 1. Once we specify an approximate kinematic model, quantify the task goals as cost functions, and model the sensory feedback, this framework can be used to predict the subject's next control signal. Therefore, we use this framework to develop the optimal feedback-controlled prior model to predict the next control signal in the decoder and consequently the next kinematic state.

We assume the sequence of kinematic states, $\mathbf{x}_0, \ldots, \mathbf{x}_t$, are generated according to the linear dynamical system

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t. \tag{2}$$

Here, $\mathbf{u}_t$ is the control signal at time t, which is decided by the controller (the subject), $\mathbf{w}_t$ is the zero-mean white Gaussian state noise with covariance matrix $\mathbf{W}$, and $\mathbf{A}$ and $\mathbf{B}$ are parameters of the kinematic model. We assume that the subject has perfect sensory feedback of the kinematic states (for example, proprioceptive or visual) and therefore the sensory feedback, $\mathbf{y}_t$, is noiseless, i.e., $\mathbf{y}_t = \mathbf{x}_t$. To find $\mathbf{u}_t$ in the control framework, we need to specify a cost function whose expected value will then be minimized by selecting the optimal $\mathbf{u}_t$. The cost function in a given task should quantify its goal. For the above linear Gaussian state-space model, if we pick the cost function as a quadratic function of the state and control signals given by

$$J = \sum_{t=1}^{\mathsf{T}-1} \left(\mathbf{x}_t'\mathbf{Q}_t\mathbf{x}_t + \mathbf{u}_t'\mathbf{R}\mathbf{u}_t\right) + \mathbf{x}_\mathsf{T}'\mathbf{Q}_\mathsf{T}\mathbf{x}_\mathsf{T} \tag{3}$$

where T is the movement duration, $\mathbf{Q}_t$ is positive semidefinite, and $\mathbf{R}$ is positive definite, then the optimal control signal at any time, $\mathbf{u}_t$, is simply a linear feedback of the state at that time given by [31], [39]

$$\mathbf{u}_t = -\mathbf{L}_t(\mathsf{T})\mathbf{x}_t \tag{4}$$

where $\mathbf{L}_t$ can be found recursively and offline using [39]

$$\mathbf{L}_t = (\mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{P}_{t+1}\mathbf{A}. \tag{5}$$

Here, $\mathbf{P}_t$ is in turn found recursively and backwards in time using

$$\begin{aligned}\mathbf{P}_t &= \mathbf{Q}_t \\ &+ \mathbf{A}'\left(\mathbf{P}_{t+1} - \mathbf{P}_{t+1}\mathbf{B}\left(\mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B}\right)^{-1}\mathbf{B}'\mathbf{P}_{t+1}\right)\mathbf{A}\end{aligned} \tag{6}$$

with the initial condition

$$\mathbf{P}_\mathsf{T} = \mathbf{Q}_\mathsf{T}. \tag{7}$$

This is the linear quadratic Gaussian (LQG) solution. Substituting (4) in (2) reduces this state-space model to the optimal feedback-controlled state-space model

$$\mathbf{x}_{t+1} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))\mathbf{x}_t + \mathbf{w}_t \tag{8}$$

which can now be used to predict the next kinematic state in the decoder.

Note that $\mathbf{Q}_t$, $\mathbf{Q}_\mathsf{T}$, and $\mathbf{R}$ should be appropriately designed for an application of interest. These matrices enforce the goal or performance criteria for a given task. For example, for a reaching task, the goal is to reach a certain target position at an arrival time T and then stop there. So the performance criteria are for the position to be close to the target position at the end of movement and for the velocity and higher-order derivatives of position to be zero then. Therefore, $\mathbf{Q}_\mathsf{T}$ that is multiplied by the kinematic state at arrival time, T, should enforce such criteria. Similarly $\mathbf{Q}_t$ can enforce performance criteria before the end of movement. Moreover, the cost function should have an effort (control) penalty term since one usually attempts to perform a task as accurately as possible while being energy efficient. That is what the term $\mathbf{u}_t'\mathbf{R}\mathbf{u}_t$ is enforcing. We show examples of these matrices for reaching movements below. It is important to also note that $\mathbf{L}_t(\mathsf{T})$ is time-varying and a function of the duration, T, and hence the state-space model is dependent on this duration and time-varying.

1) Optimal Feedback-Controlled Prior Model for a Reaching Movement: One of the most common goal-directed movements is a reaching movement. We can now specialize the feedback-controlled prior model to the case of reaching movements, which are used in many real-time BMI experiments. For a reaching movement, the goal is to reach a target position at a desired arrival time and stop there, while being energy efficient. Hence, the cost function in this case should enforce positional accuracy at the arrival time, stopping condition, and energetic efficiency [40], [41]. Therefore, denoting the desired final target position in the two dimensions by $\mathbf{d}^* = [\mathsf{d}_1^*, \mathsf{d}_2^*]'$, and the position, velocity, and force in the two dimensions by $\mathbf{d}_t = [\mathsf{d}_1(t), \mathsf{d}_2(t)]$, $\mathbf{v}_t = [\mathsf{v}_1(t), \mathsf{v}_2(t)]$, and $\mathbf{a}_t = [\mathsf{a}_1(t), \mathsf{a}_2(t)]$, respectively, similar to previous studies [40], [41] we take this cost function to be the weighted sum

$$J = \| \mathbf{d}_\mathsf{T} - \mathbf{d}^* \|^2 + w_v \| \mathbf{v}_\mathsf{T} \|^2 + w_a \| \mathbf{a}_\mathsf{T} \|^2 + w_r \sum_{t=1}^{\mathsf{T}-1} \| \mathbf{u}_t \|^2 \tag{9}$$

where the weights are chosen to penalize the terms in the cost function approximately equally on average. Taking the state to be $\mathbf{x}_t = [\mathsf{d}_1(t), \mathsf{v}_1(t), \mathsf{a}_1(t), \mathsf{d}_2(t), \mathsf{v}_2(t), \mathsf{a}_2(t)]'$ and adapting a first-order lowpass muscle-like filter in [41], we write the dynamical system in (2) in each dimension i as

$$\begin{bmatrix} \mathsf{d}_i(t+1)\\ \mathsf{v}_i(t+1)\\ \mathsf{a}_i(t+1)\end{bmatrix} = \begin{bmatrix}1 & \Delta & 0\\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m}\\ 0 & 0 & 1-\frac{\Delta}{\tau}\end{bmatrix}\begin{bmatrix}\mathsf{d}_i(t)\\ \mathsf{v}_i(t)\\ \mathsf{a}_i(t)\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{\Delta}{\tau}\end{bmatrix}\mathsf{u}_i(t) + \begin{bmatrix}0\\ 0\\ \mathsf{w}_i(t)\end{bmatrix} \qquad (10)$$

where Δ is the time increment, and parameters $b = 10$ Ns/m, $\tau = 0.05$ s, and $m = 1$ kg come from biomechanics [41].

Having specified the kinematic model in (10) and the cost function in (9) for the reaching movements, the feedback matrices $\mathbf{L}_t(\mathsf{T})$ can be computed from the recursive solution of LQG in (5). To find the feedback matrices, first we write (9) and (10) in the form of (3) and (2) by augmenting the state to include the target position in the two dimensions [40], [41], $\mathbf{d}^* = [\mathsf{d}_1^*, \mathsf{d}_2^*]'$, i.e., $\mathbf{x}_{\mathrm{aug}}(t) = [\mathsf{d}_1(t), \mathsf{v}_1(t), \mathsf{a}_1(t), \mathsf{d}_1^*, \mathsf{d}_2(t), \mathsf{v}_2(t), \mathsf{a}_2(t), \mathsf{d}_2^*]'$. Hence, the augmented state-space model in each dimension takes the form

$$\begin{bmatrix} \mathsf{d}_i(t+1)\\ \mathsf{v}_i(t+1)\\ \mathsf{a}_i(t+1)\\ \mathsf{d}_i^*\end{bmatrix} = \begin{bmatrix}1 & \Delta & 0 & 0\\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} & 0\\ 0 & 0 & 1-\frac{\Delta}{\tau} & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\mathsf{d}_i(t)\\ \mathsf{v}_i(t)\\ \mathsf{a}_i(t)\\ \mathsf{d}_i^*\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{\Delta}{\tau}\\ 0\end{bmatrix}\mathsf{u}_i(t) + \begin{bmatrix}0\\ 0\\ \mathsf{w}_i(t)\\ 0\end{bmatrix} \qquad (11)$$

for $i = 1, 2$. The state-space model in (2) in this case is therefore given by

$$\mathbf{x}_{\mathrm{aug}}(t+1) = \begin{bmatrix}\mathbf{A}_{11} & \mathbf{A}_{12}\\ \mathbf{A}_{21} & \mathbf{A}_{22}\end{bmatrix}\mathbf{x}_{\mathrm{aug}}(t) + \begin{bmatrix}\mathbf{B}_{11} & \mathbf{B}_{12}\\ \mathbf{B}_{21} & \mathbf{B}_{22}\end{bmatrix}\mathbf{u}(t) + \mathbf{w}(t) \qquad (12)$$

where $\mathbf{A}_{12} = \mathbf{A}_{21} = \mathbf{0}$, $\mathbf{B}_{12} = \mathbf{B}_{21} = \mathbf{0}$, and $\mathbf{A}_{11} = \mathbf{A}_{22}$ and $\mathbf{B}_{11} = \mathbf{B}_{22}$ are given in (11) (following the form of the model in [41]). Here, $\mathbf{u}(t) = [\,\mathsf{u}_1(t) \quad \mathsf{u}_2(t)\,]'$. Now comparing (9) with (3) and denoting the two dimensional identity matrix with $\mathbf{I}$, we find $\mathbf{R} = w_r\mathbf{I}$, and $\mathbf{Q}_t = 0$ for $t < \mathsf{T}$. To find $\mathbf{Q}_\mathsf{T}$ note that $\mathsf{d}_1(\mathsf{T}) - \mathsf{d}_1^* = p_1'\mathbf{x}_{\mathrm{aug}}(\mathsf{T})$ with $p_1 = [1, 0, 0, -1, 0, 0, 0, 0]'$ and similarly for p_2, and that $\mathsf{v}_1(\mathsf{T}) = p_{v_1}'\mathbf{x}_{\mathrm{aug}}(\mathsf{T})$ with $p_{v_1} = [0, 1, 0, 0, \ldots, 0]'$ and similarly for p_{v_2}, p_{a_1}, and p_{a_2}. Therefore we find $\mathbf{Q}_\mathsf{T} = p_1p_1' + p_2p_2' + w_v(p_{v_1}p_{v_1}' + p_{v_2}p_{v_2}')$ $+ w_a(p_{a_1}p_{a_1}' + p_{a_2}p_{a_2}')$. Now using (5)–(7) with these choices and with $\mathbf{A}$ and $\mathbf{B}$ from the augmented state-space model in (12) we obtain the feedback matrices, $\mathbf{L}_t(\mathsf{T})$. Consequently, the optimal feedback-controlled state-space model for the reaching movement is found from (8). Note that the dynamic matrix in this model, $\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T})$, is multiplied by the augmented state that includes the target position. Therefore, the feedback-controlled state-space model evolves the kinematic states according to the target position.

It is important to note that the feedback matrices can be precomputed offline and then stored for later real-time use in a BMI. Also, the prior model can be generalized to any target location by just changing $\mathbf{d}^*$ accordingly in (11).

B. Estimation of Goal-Directed Kinematic States From Neural Observations

Having developed the prior model, we now derive the decoder. To do so, we combine the prior model with the ensemble spiking activity during movement and resolve the duration uncertainty inherent to the prior model using mixture modeling. Mixture modeling is a common framework in statistical inference that is used to estimate a desired density in different applications. For example, mixture modeling combined with sequential state estimation in dynamical systems, when the system could be operating in changing environments (leading to different regimes of operation), has been used in [42]. See also the mixture Kalman filtering work in [43] and references therein. For decoding the kinematics from neural activity, mixture modeling was used in [27] and successfully applied to combine empirically fitted and time-invariant state models for reaching movements to different targets. Here, we use mixture modeling to combine optimal feedback-controlled prior models of different durations and hence resolve the duration uncertainty inherent to this prior model (and other goal-directed models in general).

To decode the kinematics, we need to find the posterior density $p(\mathbf{x}_t|\mathbf{N}_{1:t})$. Conditioning on the arrival time, we can expand this density as

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{N}_{1:t}, \mathsf{T})\, p(\mathsf{T}|\mathbf{N}_{1:t})\, d\mathsf{T} \qquad (13)$$

where $p(\mathsf{T}|\mathbf{N}_{1:t})$ is the causal likelihood function for the arrival time based on the neural observations. However, in its continuous form, this likelihood is difficult to find and in general is not Gaussian. Hence, to make the computations tractable, we discretize the arrival time and instead, using the law of total probability, expand the posterior density as

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_{j=1}^{I} p(\mathbf{x}_t|\mathbf{N}_{1:t}, \mathsf{T}_j)\, p(\mathsf{T}_j|\mathbf{N}_{1:t}) \qquad (14)$$

where T is discretized to I possibilities and consequently a prior model is placed on it given by $p_\mathsf{T}(\mathsf{T}_j), j = 1, \ldots, I$. This prior model (including its support) can be selected based on empirical durations observed in a given task. We will later examine how the discretization refinement, or equivalently the number of discretization points used for a given support, affects the performance of the decoder. We show in the Results section that even a coarse discretization is sufficient to achieve an asymptote performance.

We now need to calculate two quantities: the posterior density $p(\mathbf{x}_t|\mathbf{N}_{1:t}, \mathsf{T})$ for a given *known* arrival time, T, and the corresponding weights $p(\mathsf{T}|\mathbf{N}_{1:t})$.

To find the posterior for a given arrival time we write it as

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}, \mathsf{T}) = \frac{p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1})\, p(\mathbf{x}_t|\mathbf{N}_{1:t-1}, \mathsf{T})}{p(\mathbf{N}_t|\mathbf{N}_{1:t-1}, \mathsf{T})}. \qquad (15)$$

The first term in the numerator comes from the observation model in (1) and the second term is the one-step prediction density. Note that we used the relation $p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1}, \mathsf{T}) = p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1})$ since the neural spiking activity is assumed to be only a function of the kinematic state and history of spiking activity and hence conditionally independent of the duration. We denote the normalization constant by

$$g(\mathbf{N}_t|\mathsf{T}) = p(\mathbf{N}_t|\mathbf{N}_{1:t-1}, \mathsf{T}) \tag{16}$$

since we exploit it later to find the corresponding weights. To get the posterior density recursions, we use the Chapman–Kolmogorov equation to write the prediction density as

$$p(\mathbf{x}_t|\mathbf{N}_{1:t-1}, \mathsf{T}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathsf{T}) p(\mathbf{x}_{t-1}|\mathbf{N}_{1:t-1}, \mathsf{T}) d\mathbf{x}_{t-1}. \tag{17}$$

Here, we have used the conditional independence $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{N}_{1:t-1}, \mathsf{T}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathsf{T})$, which comes from the state-space model in (8). Now the second term inside the integral is just the posterior density from the previous time step. Hence, substituting (17) into (15) generates the recursion. The exact expression in (15) is in general complicated. Hence, we make a Gaussian approximation to the posterior density similar to [38]. Making this approximation and since the state-space model in (8) is also Gaussian, the prediction density in (17) will be Gaussian. Let us denote the minimum mean-square error (MMSE) estimator, given by the posterior mean $E(\mathbf{x}_t|\mathbf{N}_{1:t}, \mathsf{T})$, by $\mathbf{x}_{t|t,\mathsf{T}}$ and its covariance matrix by $\mathbf{W}_{t|t,\mathsf{T}}$. Similarly, we denote the one step prediction mean by $\mathbf{x}_{t|t-1,\mathsf{T}} = E(\mathbf{x}_t|\mathbf{N}_{1:t-1}, \mathsf{T})$ and its covariance matrix by $\mathbf{W}_{t|t-1,\mathsf{T}}$. The Gaussian approximation to the posterior in (15) is made by expanding its logarithm in a Taylor series expansion around the prediction mean $\mathbf{x}_{t|t-1,\mathsf{T}}$ and keeping terms of second order or smaller. The recursions for the MMSE estimator with this Gaussian approximation have been derived in [38]. The difference here is in the prediction step as it uses the feedback-controlled state-space model in (8). The recursions of this point process filter in our case become

$$\mathbf{x}_{t|t-1,\mathsf{T}} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))\mathbf{x}_{t-1|t-1,\mathsf{T}} \tag{18}$$

$$\mathbf{W}_{t|t-1,\mathsf{T}} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))\mathbf{W}_{t-1|t-1,\mathsf{T}}(\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))' + \mathbf{W} \tag{19}$$

$$\mathbf{W}^{-1}_{t|t,\mathsf{T}} = \mathbf{W}^{-1}_{t|t-1,\mathsf{T}} + \sum_{c=1}^{C} \left[\left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t}\right)' \left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t}\right) \lambda_c \Delta - (\mathsf{N}_t^c - \lambda_c \Delta) \frac{\partial^2 \log \lambda_c}{\partial \mathbf{x}_t \partial \mathbf{x}_t'} \right]_{\mathbf{x}_{t|t-1,\mathsf{T}}} \tag{20}$$

$$\mathbf{x}_{t|t,\mathsf{T}} = \mathbf{x}_{t|t-1,\mathsf{T}} + \mathbf{W}_{t|t,\mathsf{T}} \sum_{c=1}^{C} \left[\left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t}\right)' (\mathsf{N}_t^c - \lambda_c \Delta) \right]_{\mathbf{x}_{t|t-1,\mathsf{T}}} \tag{21}$$

where $[\cdot]_{\mathbf{x}_{t|t-1,\mathsf{T}}}$ denotes the evaluation of the expression at $\mathbf{x}_{t|t-1,\mathsf{T}}$ and λ_c is used instead of $\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)$ for notational convenience. When the point process model for the spiking activity is assumed to have no history dependence, the recursions are obtained by simply replacing $\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)$ with $\lambda_c(t|\mathbf{x}_t)$. These recursions give the feedback-controlled point process filter (FC-PPF).

Note that a random-walk state model is also in the form of (8) but with $\mathbf{B} = \mathbf{0}$ and equivalently the dynamics matrix given by $\mathbf{A}$. Hence, the recursions in (18)–(21) with this choice recover the random-walk point process filter (RW-PPF).

So far we have found an approximation to the posterior assuming a *known* arrival time. To relax this assumption, we should find the corresponding weights for each arrival time, $p(\mathsf{T}_j|\mathbf{N}_{1:t})$, and use (14) to find the posterior density. We have

$$p(\mathsf{T}_j|\mathbf{N}_{1:t}) = \frac{p(\mathbf{N}_{1:t}|\mathsf{T}_j) p_{\mathsf{T}}(\mathsf{T}_j)}{p(\mathbf{N}_{1:t})} \tag{22}$$

where $p(\mathbf{N}_{1:t})$ is independent of T_j and treated as a constant and $p(\mathbf{N}_{1:t}|\mathsf{T}_j)$ is the likelihood of the observed neural data under a state-space model with the arrival time of T_j. Hence, the latter is the normalization constant for the posterior $p(\mathbf{x}_{1:t}|\mathbf{N}_{1:t}, \mathsf{T}_j)$ and its exact computation requires an integration, which is computationally prohibitive. However, using the Gaussian approximation to the posterior, we can find this without integration as follows. Using the chain rule we find

$$p(\mathbf{N}_{1:t}|\mathsf{T}_j) = \prod_{i=1}^{t} p(\mathbf{N}_i|\mathbf{N}_{1:i-1}, \mathsf{T}_j) = \prod_{i=1}^{t} g(\mathbf{N}_i|\mathsf{T}_j) \tag{23}$$

where $g(\mathbf{N}_i|\mathsf{T}_j)$ is defined in (16) and is the $i\mathrm{th}$ step normalization constant in the recursive filter. Now exploiting the Gaussian approximation of the posterior and hence the prediction densities in (15), they are completely characterized by their means and covariances given in (18)–(21) for any T_j. We can hence explicitly evaluate (15) at $\mathbf{x}_{i|i,\mathsf{T}_j}$ to get

$$\begin{aligned} g(\mathbf{N}_i|\mathsf{T}_j) &= \sqrt{\frac{|\mathbf{W}_{i|i,\mathsf{T}_j}|}{|\mathbf{W}_{i|i-1,\mathsf{T}_j}|}}\, p(\mathbf{N}_i|\mathbf{x}_{i|i,\mathsf{T}_j}, \mathbf{N}_{1:i-1}) \\ &\times \exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,\mathsf{T}_j} - \mathbf{x}_{i|i-1,\mathsf{T}_j})' \mathbf{W}^{-1}_{i|i-1,\mathsf{T}_j} (\mathbf{x}_{i|i,\mathsf{T}_j} - \mathbf{x}_{i|i-1,\mathsf{T}_j})\right] \\ &= \sqrt{\frac{|\mathbf{W}_{i|i,\mathsf{T}_j}|}{|\mathbf{W}_{i|i-1,\mathsf{T}_j}|}} \prod_c \left(\lambda_c(i|\mathbf{x}_{i|i,\mathsf{T}_j}, \mathbf{H}_i^c)\Delta\right)^{\mathsf{N}_i^c} e^{-\lambda_c(i|\mathbf{x}_{i|i,\mathsf{T}_j}, \mathbf{H}_i^c)\Delta} \\ &\times \exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,\mathsf{T}_j} - \mathbf{x}_{i|i-1,\mathsf{T}_j})' \mathbf{W}^{-1}_{i|i-1,\mathsf{T}_j} (\mathbf{x}_{i|i,\mathsf{T}_j} - \mathbf{x}_{i|i-1,\mathsf{T}_j})\right] \end{aligned} \tag{24}$$

for $j = 1, \ldots, I$ where all the quantities are known. Combining (14)–(24) gives the posterior. The MMSE estimate in the case of unknown arrival time is then given by

$$\mathbf{x}_{t|t} = E(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_j p(\mathsf{T}_j|\mathbf{N}_{1:t}) \mathbf{x}_{t|t,\mathsf{T}_j}.$$

We call the resulting filter the feedback-controlled parallel point process filter (FC-P-PPF) shown in Fig. 2; it consists of I parallel branches of feedback-controlled point process filters, each calculating not only the MMSE estimate of $\mathbf{x}_t$ assuming a duration of T_j, but also the corresponding likelihood $p(\mathbf{N}_{1:t}|\mathsf{T}_j)$.

One question that arises at this point is how to estimate the state once $t > \mathsf{T}_j$ for some branch T_j. One way to do so is to perform the summation over all j for which $\mathsf{T}_j > t$. This means

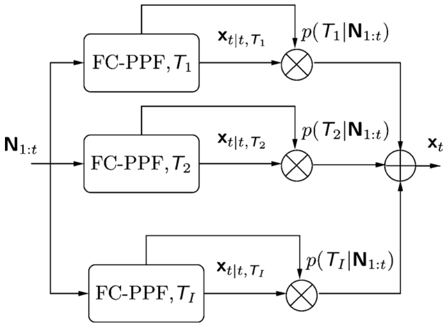


Fig. 2. Feedback-controlled parallel point process filter (FC-P-PPF). FC-P-PPF consists of I parallel branches of point process filters, each calculating not only the MMSE estimate of $\mathbf{x}_t$ assuming a duration of T_j, but also the corresponding likelihood $p(\mathbf{N}_{1:t}|\mathsf{T}_j)$.

that after a given arrival time, T_j, the corresponding branch of the decoder exits the computations and is no longer used for state estimation. This approach is justified since the prior for a given branch is only defined up to its corresponding arrival time. Using this implementation and as time passes by, fewer branches are active and hence the computational load is lower. Another way to address this issue is to keep a branch in the decoder even after its arrival time is reached by changing its prior to one that corresponds to a still condition. This still condition will have its position fixed at the final position reached to at the arrival time and hence a zero velocity (and other higher order derivatives of the position) after the arrival time. This means that all the branches will be active at all time until the decoder is terminated at the largest arrival time of the branches. We will examine the effect of the two alternative implementations in detail in Section III.

Finally, the parallel point process filter can be used to remove the duration uncertainty from any other goal-directed state-space model and not just the feedback-controlled one we have proposed here. This can be done simply by changing the prediction step in (18) and (19) using the desired state-space model. This is expanded on in the Appendix. Also, our approach to resolving the duration uncertainty can be applied in cases where the observations are taken to be the neural firing rates instead of the spiking activity and a Gaussian likelihood model is assumed on these firing rates as in [25]. In this case the Gaussian approximation to the posterior in our FC-P-PPF becomes exact and the update recursions in the filter become those of the well-known Kalman filter. The calculation of the arrival time likelihoods, however, remain the same.

III. Results

We show the application of FC-P-PPF to decoding of reaching movements performed by a rhesus monkey from ensemble spiking activity simulated based on a rigorously validated model of neural activity in the primary motor cortex [2], [35]. Our data consists of 55 reaching trajectories performed by a rhesus monkey,[1] based on which neural spiking activity is simulated. Using a joystick, the monkey moved a cursor from the center to one of four possible radial locations at the top, bottom, left, or right sides of a square computer screen 14 cm long in each dimension. The 55 trajectories had different durations in the range of 140–400 ms. Hence, the window of uncertainty for the arrival time is 140–400 ms. The state noise covariance, $\mathbf{W}$, in the state-space model in (10) was fitted to reaching trajectories using maximum-likelihood parameter estimation.

To dissociate the effect of using the feedback-controlled prior model from that of using the parallel filters to resolve the duration uncertainty, we first consider the scenario where the decoder knows the exact duration for each trajectory and then remove this assumption and consider the general case with unknown duration. Also in each case, we compare the decoder with RW-PPF, which does not model the goal-directed nature of movement and does not incorporate the target information. We show that the decoder improves the RW-PPF performance considerably, where performance is measured as the average root mean square (rms) error in the estimated trajectories.

In terms of the decoder implementation, we examine the effect of two factors on its performance. First, we examine the effect of the discretization refinement, i.e., the number of discretization points or parallel branches used in the decoder, and show that a relatively coarse discretization is sufficient for the decoder to achieve an asymptote average rms error. This guides the selection of the number of parallel branches in the decoder. Second we investigate the effect of the two possible implementations for treating a branch after its arrival time. In the first implementation the branches exit the decoder after their arrival time and in the second implementation they stay in the decoder in their still condition after their arrival time. We show that the average rms errors using the two implementations are comparable.

A. Estimation of Reaching Movements of Known Duration From Neural Signals

To assess the advantage of using the feedback-controlled prior model, we first implement the FC-P-PPF for estimation of reaching movements that start at a known reference position at rest, acquire a target position at a *known* time, and come to a stop at that time. Note that when duration is known, FC-P-PPF uses a single branch corresponding to the true duration for each trajectory.

To get the corresponding ensemble spiking activity for each real reaching trajectory, we used the model of neural activity in the motor cortex, also known as the cosine tuning model of the conditional intensity function, which has been extensively validated on primate motor cortical data [2], [35]. In this model, the firing rate of each neuron is related to the movement speed and direction. Using this model and for each trajectory, we simulated the spiking activity of an ensemble of $C = 20$ neurons, which is a typical ensemble size in real-time BMIs. The spiking activity for each neuron in the ensemble was independently simulated as a point process whose instantaneous firing rate or conditional intensity function during a 2-D movement was given by [35]

$$\begin{aligned}\lambda_c(t|\mathbf{x}_t) &= \exp\left(\beta_c + \alpha_c|\mathbf{v}_t|\cos(\theta_t - \theta_p^c)\right)\\ &= \exp\left(\beta_c + \alpha_y^c \mathsf{v}_2(t) + \alpha_x^c \mathsf{v}_1(t)\right) \qquad (25)\end{aligned}$$

[1] This study was performed in strict accordance with the recommendations in the Guide for the Care and Use of Laboratory Animals of the National Institutes of Health, and under regulation of the Subcommittee on Research Animal Care at Harvard Medical School.

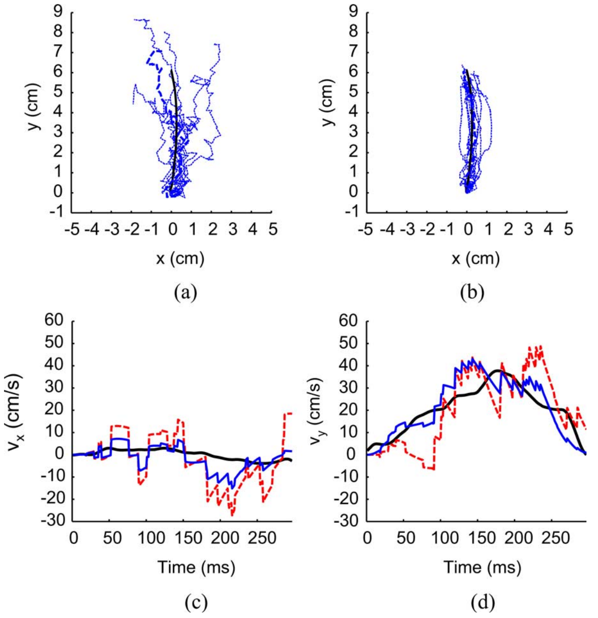


Fig. 3. Comparison of FC-P-PPF and RW-PPF when movement duration is known. (a) and (b) Sample trajectory during a reaching movement by the monkey (solid black) and 10 sample decoded trajectories using RW-PPF in (a) and FC-P-PPF in (b) (dashed blue in color and dashed grey in print version). (c) and (d) True velocity (solid black) in the two dimensions and the decoded velocities using RW-PPF (dashed) and FC-P-PPF (solid blue in color and solid grey in print version) for a sample neural realization.

where θ_p^c is the preferred angle of the cth neuron, θ_t is the movement angle at t, $\mathsf{v}_2(t)$ and $\mathsf{v}_1(t)$ are the velocities in the y and x directions, β_c and α_c are parameters of the model, and consequently, $\alpha_x^c = \alpha_c \cos(\theta_p^c)$ and $\alpha_y^c = \alpha_c \sin(\theta_p^c)$. Here, the preferred direction for each neuron is sampled randomly from $[-\pi, \pi]$, $\beta_c = 1.6$ is chosen to have a background firing rate of 5 Hz for each neuron and $\alpha_c = 0.04\ \mathrm{s/cm}$ is chosen to have a maximum possible firing rate of 25 Hz on average (over the trajectories) for each neuron. We simulated 100 realizations (trials) of the point process neural signal for each trajectory using the time-rescaling theorem described in [44].

1) Performance Measure: We define the performance measure of the decoder as its average rms error over the trajectories. To find this, we first find the rms error for each time step along a trajectory by averaging the square estimation error at that time step over the 100 simulated neural realizations. To quantify an average rms error for the entire trajectory, we then average the rms error over its duration. Finally, we find the mean of the average rms errors over all 55 trajectories as the measure of performance for a decoder.

2) Comparison to a Random-Walk Model: Assuming a known movement duration, we compare the average rms error of FC-P-PPF to that of RW-PPF. This comparison illustrates the advantage of using a feedback-controlled goal-directed prior model that incorporates the target information over a random-walk model. The average rms errors of RW-PPF and FC-P-PPF, which are calculated for each trajectory until the end of movement, are 1.40 cm and 0.87 cm, respectively. Hence, the RW-PPF average rms is 61% higher than that of FC-P-PPF. This shows the benefit of using a more accurate prior model in the decoder. Fig. 3(a) and (b) shows a sample reaching trajectory performed by the monkey with a duration of $\mathsf{T} = 296\ \mathrm{ms}$ and 10 sample decoded trajectories from RW-PPF and FC-P-PPF when T is known exactly to the decoder. Fig. 3(c) and (d) shows the estimated velocity in the x and y direction for a sample neural realization (trial) using RW-PPF and FC-P-PPF. We can also visually observe that using the feedback-controlled prior model improves the estimation accuracy. Note that the average rms error of FC-P-PPF assuming knowledge of movement duration provides a lower bound on its error in the general realistic case where this duration is not known, as considered in the next section.

B. Estimation of Reaching Movements of Unknown Duration From Neural Signals

We now remove the assumption of known movement duration from the problem and use FC-P-PPF to decode the same reaching movements. We examine the effects of the discretization refinement and two possible implementations on FC-P-PPF performance and also compare it to RW-PPF. Since the movement duration is unknown, we run both decoders until the end of the uncertainty window, i.e., until 400 ms.

1) Performance Loss Due to Unknown Duration: We first examine the loss incurred due to lack of *a priori* knowledge of the movement duration by FC-P-PPF and the effect of the number of discretization points on this loss. This will guide the selection of the number of branches needed in the decoder. To do so, we compare the average rms error of FC-P-PPF with various number of discretization points to its error assuming exact *a priori* knowledge of duration as considered in the previous section. The latter provides a lower bound on the former and is hence used as a baseline.

We discretize the uncertainty interval starting at 150 ms, i.e., the interval $[150, 400]\ \mathrm{ms}$, in steps of 250 ms, 125 ms, 83 ms, 50 ms, or 25 ms, corresponding to 2, 3, 4, 6, or 11 discretization points, respectively, and find the FC-P-PPF average rms error in each case. We also find the average rms error using a single branch at 400 ms, which corresponds to a single discretization point at the largest possible duration. In each case, we assume a uniform prior distribution on the discretization points. To make the comparison to the baseline case with known duration, we compute the average rms error of FC-P-PPF with different number of discretization points for each trajectory up to the end of movement as opposed to the end of the uncertainty window. In each case, we also examine the two possible implementations for treating a branch after its arrival time.

Fig. 4 shows the FC-P-PPF average rms error as a function of the number of discretization points for both implementations and also the baseline average rms error assuming movement duration is known. In the first implementation, the average rms error has a steep drop to an asymptote value by increasing the number of discretization points. Taking the average rms error achieved using 11 discretization points as the asymptote value, using only four discretization points results in the difference between the average rms error and the asymptote value to be only 1% of the latter. This shows that a relatively coarse discretization is sufficient to get to this asymptote. Also, increasing the

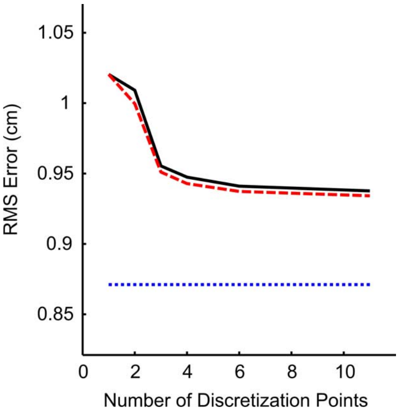


Fig. 4. The effect of the number of duration discretization points on FC-P-PPF average rms error. The solid curve shows the average rms error of the first implementation of FC-P-PPF during movement, the dashed curve shows that of the second implementation, and the dotted line shows the lower bound on average rms error of FC-P-PPF, which corresponds to a known movement duration.

number of discretization points from 1 to 4 reduces the difference between the average rms error of FC-P-PPF and the baseline (i.e., with *a priori* knowledge of the duration) by over 48%.

Similar results hold for the second implementation. The two implementations have very close average rms error during movement. For example using four discretization points, the difference between the average rms error of the two implementations is less than 1% of the smaller value.

2) Comparison to a Random-Walk Point Process Filter: We now compare the performance of FC-P-PPF to RW-PPF that does not exploit the target information. In our comparison, we use four branches for FC-P-PPF as this choice is sufficient to get within 1% of the asymptote average rms error.

Fig. 5(a) and (b) shows a sample trajectory with a duration of $\mathsf{T} = 296$ ms (the same as in Fig. 3) and 10 sample decoded trajectories using RW-PPF and FC-P-PPF when T is unknown to the decoders and hence they decode the neural activity until the end of the uncertainty window. We have shown the FC-P-PPF results using the first implementation. As the figure demonstrates, FC-P-PPF generates more accurate trajectory estimates compared to RW-PPF (as will be quantified in Table I). Fig. 5(c)–(f) shows the decoded velocities in the x and y directions for a single realization of the neural signal and the time evolution of the optimal combining weights in FC-P-PPF for the given realization and also averaged over all 100 realizations for the given trajectory. Here, the true arrival time is at $\mathsf{T} = 296$ ms and the closest weight, i.e., the one corresponding to 316 ms, soon dominates up to its arrival time. Also, FC-P-PPF brings the estimated velocity close to zero at the end of movement as opposed to RW-PPF.

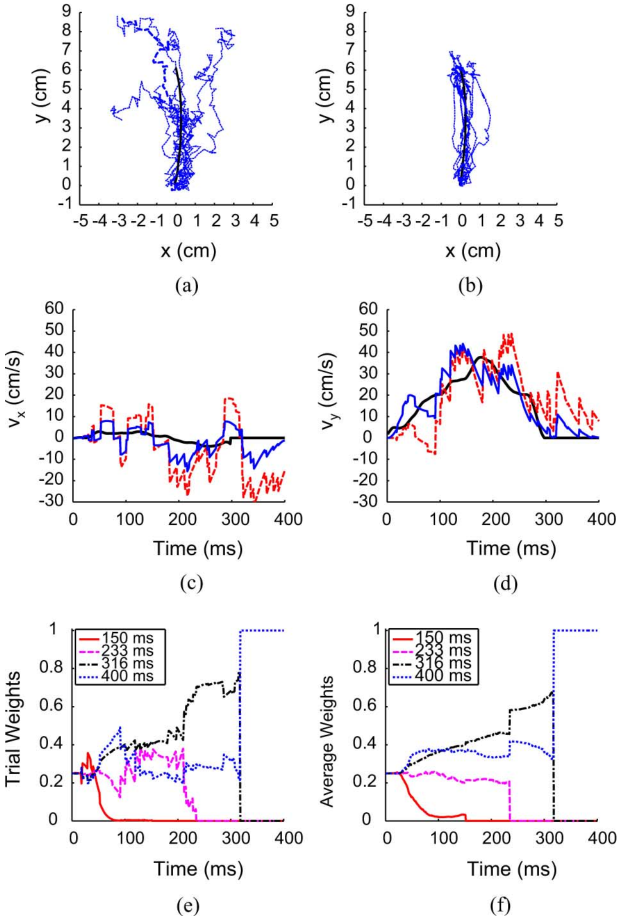


Fig. 5. Comparison of FC-P-PPF and RW-PPF when movement duration is unknown. (a) and (b) Sample trajectory during a reaching movement by the monkey (solid black) and 10 sample decoded trajectories using RW-PPF in (a) and the first implementation of FC-P-PPF in (b) (dashed blue in color and dashed grey in print version). The decoders are run until the end of the uncertainty interval. (c) and (d) True velocity (solid black) in the two dimensions and the decoded velocities using RW-PPF (dashed) and FC-P-PPF (dashed blue in color and dashed grey in print version) for a sample realization of the neural signal. (e) and (f) Evolution of the optimal combining weights for the branches of FC-P-PPF for the sample realization of the neural signal in (e) and averaged over all 100 realizations in (f).

TABLE I
AVERAGE rms ERROR (CM) IN DECODED TRAJECTORY OF THE 55 REAL REACHING MOVEMENTS WITH UNKNOWN DURATION CALCULATED UNTIL THE END OF MOVEMENT AND UNTIL THE END OF THE UNCERTAINTY WINDOW

	RW-PPF	FC-P-PPF (1)	FC-P-PPF (2)
Until the end of movement	1.40	0.95	0.94
Until the end of uncertainty window	1.69	1.01	0.99

To quantify these observations, Table I shows the average rms error in the estimate of the 55 trajectories during movement and during the entire decoding period. These are found by averaging the rms error for each trajectory until the end of its corresponding movement and until the end of the uncertainty window at 400 ms, respectively. During movement, the average rms error of RW-PPF is 47% larger than that of FC-P-PPF. Moreover, the two implementations of FC-P-PPF have very similar average rms errors, as is also observed in Fig. 4. When also considering the error after the end of movement, RW-PPF average rms error is 67% higher than FC-P-PPF. In this case, FC-P-PPF does better than RW-PPF for two reasons: First, it puts a better prior model on the reaching movements. Second, it detects the end of movement and hence brings the velocity close to zero after the movement ends. To do so, FC-P-PPF finds the likelihoods of the discretized arrival times and reflects them into its estimate. This is done by combining the estimates of the parallel filters, each designed for a different arrival time, with time-varying optimal weights that are updated purely based on

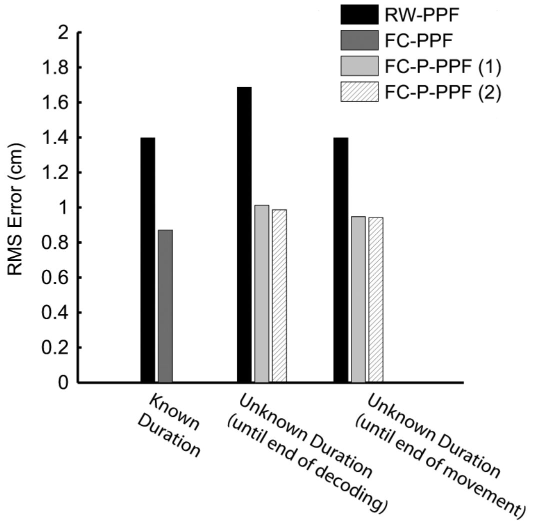


Fig. 6. Comparison of average rms error (cm) in decoded trajectory of the 55 real reaching movements for RW-PPF and FC-P-PPF with the two implementations. rms errors for known duration are shown on the left and for unknown duration (calculated until the end of decoding or until the end of movement) are shown on the right. See also Table I.

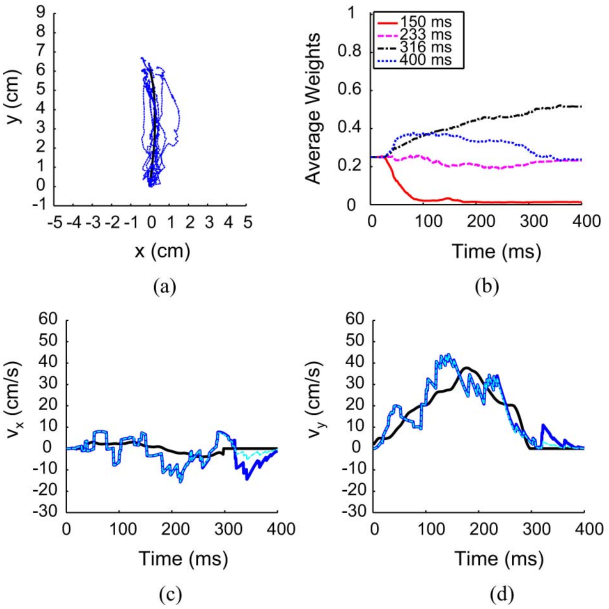


Fig. 7. Comparison of FC-P-PPF implementations. (a) True trajectory (solid black, same as in Fig. 5) and 10 sample decoded ones (dashed blue in color and dashed grey in print version) using the second implementation of FC-P-PPF with unknown duration [compare with Fig. 5(b)]. (b) Combining weights versus time averaged over all 100 neural realizations for the second implementation of FC-P-PPF [compare with Fig. 5(f)]. (c)–(d) True velocities in the two dimensions (solid black) and decoded velocities for a sample neural realization with the first (solid blue in color and solid grey in print version) and second (dashed cyan in color and dashed black in print version) implementations of FC-P-PPF. True duration is 296 ms.

the neural observation. In contrast, RW-PPF cannot detect the movement termination. Therefore, when considering its error until the end of the uncertainty window, it does worse. Note that the two implementations of FC-P-PPF again have a close average rms error. All these rms comparisons have been shown in the bar graph in Fig. 6.

Fig. 7(a)–(d) shows sample estimated trajectories, velocities, and average combining weights for the second implementation of FC-P-PPF. Keeping all the branches in the filter even after their arrival time, by changing their prior model to a still model, allows the second implementation to bring the velocity closer to zero after the movement ends and results in a lower average rms error after the end of movement compared to the first implementation (1.08 cm versus 1.16 cm, calculated from the end of movements until the end of uncertainty window at 400 ms). Therefore, in applications where the error after the end of movement and hence standing still at a position is important, for example when reaching a target and then holding it, this implementation could result in additional reduction of error. However, the first implementation is computationally more efficient as it does not keep all the branches in its calculations until the end of decoding. Hence, the first implementation is more appropriate in real-time applications in which the main purpose is only to reach a target.

IV. Discussion

We have developed a recursive Bayesian decoder for estimation of goal-directed movements from neural spiking activity. Our decoder employs two main components. First, it uses the sensorimotor optimal feedback control principles to build an optimal feedback-controlled prior model for goal-directed movements. This model not only incorporates the target information but also models the real-time sensory feedback. Second, the decoder resolves the duration uncertainty inherent to this model (and other goal-directed prior models) based on the neural activity. The result is the feedback-controlled parallel point process filter (FC-P-PPF) that can be used in a real-time BMI for estimation of goal-directed movements.

A. Summary of Contributions

We used a theory of motor control, namely the optimal feedback control theory of the sensorimotor system to build the prior model. This allows us to formulate the BMI or neural decoding problem in a framework based on the theories of motor control. Our prior model incorporates the target information and models the real-time sensory feedback. It can generalize to different target locations without requiring a training data set of reaches to these locations. Even though we have illustrated one example of the optimal feedback-controlled prior model in (8) for the specific task of reaching a target, this model could be specialized to other tasks by quantifying their goals as the cost function in (3), thus providing a possible general framework for construction of goal-directed prior models.

Despite the advantage of goal-directed prior models for estimation of movement, one key factor that has prevented their implementation in real-time BMIs is their dependence on the movement duration, which is not known *a priori* to a real-time decoder. Previous work using duration-dependent goal-directed models have therefore assumed that the decoder has knowledge of this duration [28], [30], which is not the case in a real-time BMI. We thus developed a parallel point process filter framework to resolve this duration uncertainty in *any* goal-directed prior model. The framework works by discretizing the duration, causally estimating the kinematics for each discretized duration based on the neural activity, and optimally combining these estimates according to the duration likelihoods, which are again calculated from the neural activity. Due to the parallel nature of our decoder, its overall run-time is on the order of the run-time for a single branch, a property important for real-time implementation.

We additionally examined the effect of the duration discretization refinement on the average rms error and showed that using only a coarse discretization is sufficient to achieve an asymptote average rms error. This guides the selection of the number of branches in the parallel filter and shows that the added computational complexity involved to address the duration uncertainty is not large.

We further examined the effect of two possible filter implementations for FC-P-PPF, which differed in treating the branches of the filter after their arrival time. In the first implementation, any branch exits the computation after its arrival time. In the second implementation, we keep all the branches in the filter after their arrival time, but change their prior model to a still model. The still model for a branch assumes that the kinematic state has stopped at the location estimated at its arrival time. We showed that the two implementations have similar average rms errors. However, the second implementation brings the velocity closer to zero after the movement ends (and therefore may have an advantage for applications such as holding a target), whereas the first implementation is computationally more efficient.

We also examined the advantages of using our decoder compared to RW-PPF that uses a random-walk prior model. We showed that using a very coarse discretization (and hence few branches in FC-P-PPF) the average rms error of our decoder is significantly lower than that of RW-PPF. The reason for this improved performance is that FC-P-PPF not only uses a better prior model for goal-directed movements, but also attempts to identify the end of movement and hence bring the velocity close to zero at that point. This is in contrast to RW-PPF, which cannot detect the end of movement.

B. Concluding Remarks and Future Directions

We have evaluated the feedback-controlled parallel point process filter in this work in decoding of real reaching movements performed by a rhesus monkey based on simulated neural activity. Simulation-based validation methods provide a practical way to develop and test new decoding algorithms before investment in the invasive multielectrode recording experiments with extensive resource and time requirements. Hence, these simulation-based methods have been used for algorithm development in previous work, e.g., [28], [29], [45], [46]. There are different levels of simulation-based validation. To have a more realistic validation method than that used in previous work on goal-directed decoding, e.g., in [28], [29], we test the decoder on real reaching movements performed by a rhesus monkey. This is in contrast to these previous works that in addition to using simulated neural data, also simulate the movement trajectory based on the same state model they develop and use in the decoder. We obtain the simulated neural data corresponding to the monkey's trajectories using a rigorously validated model of neural activity in the primary motor cortex [2], [35] that relates it to the intended movement parameters. Using thousands of simulated trials, our validation method allows us to characterize the decoder performance under various conditions and give guidelines for its implementation and parameter selection. The next level of validation for our decoder is to implement and test it in extensive goal-directed experiments with multielectrode recordings both offline and in a real-time BMI, and is the focus of our other work that develops a BMI using optimal feedback control theory [9], [36], [37].

In this work we assume that the upper-bound on the movement duration is known. For the decoder to be universal, it should also be applicable in scenarios in which such an upper-bound is not known. One possible approach to address this is to start the decoder using an initial guess of the upper-bound (e.g., based on previous task performances) and then refine this guess as neural observations are made. The development of this universal decoder will be the topic of future investigation.

We used the optimal feedback control theory of the sensorimotor system to build the goal-directed prior model. This allows us to formulate the BMI or neural decoding problem in a framework based on this theory of motor control, which has been shown to explain the behavioral observations in experiments involving motor tasks [31], [32], [40], [41]. In a BMI, our decoder takes into account the subject's task goal as well as the sensory feedback during BMI control to estimate the subject's intended kinematic state that is reflected in the neural activity. Even though we have evaluated one example of the optimal feedback-controlled decoder for the specific task of moving towards a target location, it is possible to extend the decoder to estimate movements in other goal-directed tasks by quantifying their goals as the cost function in (3). Recent work have discussed the importance of sensory feedback in the performance of real-time BMIs. For example [47] develops a closed-loop simulator as a way to incorporate the feedback effects in simulation-based validation methods for testing of decoding algorithms. In addition to modeling the task goals, the optimal feedback-controlled decoder here provides one approach to model the real-time sensory feedback in BMIs.

Motivated by the observation that target location can be accurately decoded from motor cortical areas before movement initiation [23], [24], in this work we assumed that target location is known and developed a decoder that combines it with the peri-movement activity to estimate the intended kinematics. Note, however, that we can easily extend this decoder for cases where target location is not known prior to movement by using an approach similar to the work in [27], which combines time-invariant and empirically fitted trajectory models to different targets. To do so, we can include branches in FC-P-PPF for all possible target locations and their corresponding discretized durations. We can then calculate the overall kinematic estimate as the weighted average of the branch estimates corresponding to all possible target locations, as opposed to just the correct target location. The calculation of the branch likelihoods, that in turn determine their weights in calculating the overall kinematic estimate, remains exactly the same [see (22)–(24)]. In this case, if target related activity before movement initiation is recorded, we can incorporate it by initializing the weight of the branches corresponding to each target proportional to its likelihood calculated from this activity [27], as opposed to equiprobably.

Real-time BMIs have largely decoded individually either the target of the movement or its corresponding trajectory without modeling its goal-directed nature. Here, we developed a decoder that aims to closely model a goal-directed movement and the real-time sensory feedback using a theory of motor control, and enables joint decoding of both target and trajectory from neural activity. This can in turn result in more accurate estimation of intended movement in a real-time BMI [9], [36], [37].

Appendix
Resolving the Duration Uncertainty From a General Linear Goal-Directed State-Space Model

A general linear goal-directed state-space model depends on the target and movement duration and can be written as

$$\mathbf{x}_{t+1} = \mathbf{G}(t, \mathsf{T}, \mathbf{x}^*)\mathbf{x}_t + \mathbf{b}(t, \mathsf{T}, \mathbf{x}^*) + \mathbf{w}_t \qquad (26)$$

where $\mathbf{x}^*$ is the target state, $\mathbf{G}$ and $\mathbf{b}$ are the dynamics matrix and a constant term, respectively, that in general are time-varying and also dependent on $\mathbf{x}^*$ and T, and $\mathbf{w}_t$ is the zero-mean white Gaussian state noise with covariance matrix $\mathbf{W}_t$. Note that the state-space models in both [28] and [30] can be written in this form and assume prior knowledge of T. Hence, the exact same filter recursions in (18)–(21) can be applied to a general model by just changing the filter prediction steps to

$$\begin{aligned}\mathbf{x}_{t|t-1,\mathsf{T}} &= \mathbf{G}(t, \mathsf{T}, \mathbf{x}^*)\mathbf{x}_{t-1|t-1,\mathsf{T}} + \mathbf{b}(t, \mathsf{T}, \mathbf{x}^*)\\ \mathbf{W}_{t|t-1,\mathsf{T}} &= \mathbf{G}(t, \mathsf{T}, \mathbf{x}^*)\mathbf{W}_{t-1|t-1,\mathsf{T}}\mathbf{G}'(t, \mathsf{T}, \mathbf{x}^*) + \mathbf{W}_t.\end{aligned}$$

All the other expressions including those for calculating the duration likelihoods remain the same.

References

[1] A. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movement and cell discharge in primate motor cortex," *J. Neurosci.*, vol. 2, no. 11, pp. 1527–1537, Nov. 1982.

[2] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching," *J. Neurophysiol.*, vol. 82, pp. 2676–2692, 1999.

[3] R. Caminiti, P. B. Johnson, C. Galli, S. Ferraina, and Y. Burnod, "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.*, vol. 11, no. 5, pp. 1182–1197, May 1991.

[4] G. H. Mulliken, S. Musallam, and R. A. Andersen, "Forward estimation of movement state in posterior parietal cortex," *Proc. Nat. Acad. Sci. USA*, vol. 105, pp. 8170–8177, 2008.

[5] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivsan, and M. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, Nov. 2000.

[6] L. H. Snyder, A. P. Batista, and R. A. Andersen, "Coding of intention in the posterior parietal cortex," *Nature*, vol. 386, pp. 167–170, Mar. 1997.

[7] A. P. Batista, C. A. Buneo, L. H. Snyder, and R. A. Andersen, "Reach plans in eye-centered coordinates," *Sci.*, vol. 285, pp. 257–260, Jul. 1999.

[8] J. Messier and J. F. Kalaska, "Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task," *J. Neurophysiol.*, vol. 84, pp. 152–165, 2000.

[9] M. M. Shanechi, "Real-time brain-machine interface architectures: Neural decoding from plan to movement," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 2011.

[10] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nat. Neurosci.*, vol. 2, no. 7, pp. 664–670, Jul. 1999.

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

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

[13] 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. 193–208, 2003.

[14] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 51, pp. 17849–17854, Dec. 2004.

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

[16] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, pp. 1098–1101, Jun. 2008.

[17] C. T. Moritz, S. I. Perlmutter, and E. E. Fetz, "Direct control of paralysed muscles by cortical neurons," *Nature*, vol. 456, pp. 639–643, Dec. 2008.

[18] G. H. Mulliken, S. Musallam, and R. A. Andersen, "Decoding trajectories from posterior parietal cortex ensembles," *J. Neurosci.*, vol. 28, no. 48, pp. 12913–12926, Nov. 2008.

[19] S.-P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, and M. J. Black, "Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia," *J. Neural Eng.*, vol. 5, pp. 455–476, 2008.

[20] K. Ganguly and J. M. Carmena, "Emergence of a stable cortical map for neuroprosthetic control," *PLoS Biol.*, vol. 7, no. 7, pp. 1–13, Jul. 2009.

[21] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. L. Nicolelis, "Unscented Kalman filter for brain-machine interfaces," *PLoS ONE*, vol. 4, no. 7, pp. 1–18, Jul. 2009.

[22] S. M. Chase, A. B. Schwartz, and R. E. Kass, "Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms," *Neural Netw.*, vol. 22, pp. 1203–1213, 2009.

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

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

[25] 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 Comput.*, vol. 19, pp. 80–118, 2006.

[26] M. M. Shanechi, R. C. Hu, M. Powers, G. W. Wornell, E. N. Brown, and Z. M. Williams, "Neural population partitioning and a concurrent brain-machine interface for sequential motor function," *Nat. Neurosci.*, vol. 15, no. 12, pp. 1715–1722, Dec. 2012.

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

[28] 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 Comput.*, vol. 18, pp. 2465–2494, 2006.

[29] J. E. Kulkarni and L. Paninski, "State-space decoding of goal-directed movements," *IEEE Signal Process. Mag.*, vol. 25, no. 4, pp. 78–86, Jan. 2008.

[30] C. Kemere and T. Meng, "Optimal estimation of feed-forward-controlled linear systems," *Proc. ICASSP*, pp. 353–356, Mar. 2005.

[31] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nat. Neurosci.*, vol. 5, no. 11, pp. 1226–1235, Nov. 2002.

[32] E. Todorov, "Optimality principles in sensorimotor control," *Nat. Neurosci.*, pp. 907–915, Sep. 2004.

[33] M. M. Shanechi, G. W. Wornell, Z. M. Williams, and E. N. Brown, "A parallel point-process filter for estimation of goal-directed movements from neural signals," *Proc. ICASSP*, Mar. 2010.

[34] L. Srinivasan and M. da Silva, "Breaking the fixed-arrival-time restriction in reaching movements of neural prosthetic devices," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 6, pp. 1555–1564, Jun. 2011.

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

[36] M. M. Shanechi, Z. M. Williams, G. W. Wornell, and E. N. Brown, "A brain-machine interface combining target and trajectory information using optimal feedback control," presented at the Computat. Syst. Neurosci. (COSYNE) Meeting, Salt Lake City, UT, Feb. 2011.

[37] M. M. Shanechi, Z. M. Williams, G. W. Wornell, R. Hu, M. Powers, and E. N. Brown, "A real-time brain-machine interface combining motor target and trajectory intent using an optimal feedback control design," *PLOS One*, submitted for publication.

[38] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Comput.*, vol. 16, pp. 971–998, 2004.

[39] D. Bertsekas*, Dynamic Programming and Optimal Control*. Nashua, NH: Athena Scientific, 2005.

[40] E. Todorov, "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system," *Neural Comput.*, vol. 17, pp. 1084–1108, 2005.

[41] D. Liu and E. Todorov, "Evidence for the flexible sensorimotor strategies predicted by optimal feedback control," *J. Neurosci.*, pp. 9354–9368, Aug. 2007.

[42] G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Trans. Autom. Control*, vol. 15, no. 1, pp. 10–17, Feb. 1970.

[43] R. Chen and J. S. Liu, "Mixture Kalman filters," *J. Roy. Stat. Soc. B*, vol. 62, pp. 493–508, 2000.

[44] E. N. Brown, R. Barbieri, V. Ventura, R. Kass, and L. Frank, "The time-rescaling theorem and its application to neural spike train data analysis," *Neural Comput.*, vol. 14, pp. 325–346, 2001.

[45] V. Ventura, "Spike train decoding without spike sorting," *Neural Comput.*, vol. 20, pp. 923–963, 2008.

[46] 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.

[47] J. P. Cunningham, P. Nuyujukian, V. Gilja, C. A. Chestek, S. I. Ryu, and K. V. Shenoy, "A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces," *J. Neurophysiol.*, vol. 105, pp. 1932–1949, 2011.

Maryam M. Shanechi (M'12) received the B.A.Sc. degree with honors in engineering science from the University of Toronto, Toronto, ON, Canada, in 2004, and the S.M. and Ph.D. degrees in electrical engineering and computer science (EECS) from the Massachusetts Institute of Technology (MIT), Cambridge, in 2006 and 2011, respectively.

She has held a postdoctoral fellowship at Harvard Medical School and Massachusetts General Hospital, and is currently completing a postdoctoral fellowship at the University of California, Berkeley. She currently holds a Visiting Assistant Professor position in the School of Electrical and Computer Engineering at Cornell University, Ithaca, NY, where she will join as an Assistant Professor in 2014. Her research focuses on applying the principles of information and control theories and statistical signal processing to develop effective solutions for basic and clinical neuroscience problems, including brain–machine interface design for motor function and for closed-loop control of anesthesia. Her work combines methodology development with *in vivo* implementation and testing.

Dr. Shanechi has received various awards for academic achievement including the Professional Engineers of Ontario gold medal, the W. S. Wilson medal, and the Natural Sciences and Engineering Research Council of Canada doctoral fellowship.

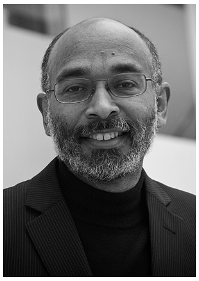
Gregory W. Wornell (S'83–M'91–SM'00–F'04) received the B.A.Sc. degree from the University of British Columbia, Vancouver, BC, Canada, and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, all in electrical engineering and computer science, in 1985, 1987, and 1991, respectively.

Since 1991 he has been on the faculty at MIT, where he is Professor of Electrical Engineering and Computer Science, and Chair of Graduate Area I (Systems, Communication, Control, and Signal Processing) within the department's doctoral program. He has held visiting appointments at the former AT&T Bell Laboratories, Murray Hill, NJ, the University of California, Berkeley, and Hewlett-Packard Laboratories, Palo Alto, CA. His research interests and publications span the areas of signal processing, statistical inference, digital communication, and information theory, and include algorithms and architectures for wireless and sensor networks, multimedia applications, imaging systems, and aspects of computational biology and neuroscience.

Dr. Wornell has won a number of awards for both his research and teaching. He has been involved in the Signal Processing and Information Theory societies of the IEEE in a variety of capacities, and maintains a number of close industrial relationships and activities.

Ziv M. Williams is an Assistant Professor of neurosurgery at Massachusetts General Hospital, Harvard Medical School. His interests lie in neural network communication and neural prosthesis development.

Prof. Williams is a recipient of the Presidential Early Career Award for Scientists and Engineers.



Emery N. Brown (M'01–SM'06–F'08) received the B.A. degree from Harvard College, Cambridge, MA, the M.D. degree from Harvard Medical School, Boston, MA, and the A.M. and Ph.D. degrees in statistics from Harvard University, Cambridge, MA.

He is currently a Professor of computational neuroscience and health sciences and technology in the Department of Brain and Cognitive Sciences and in the Institute of Medical Engineering and Sciences, Massachusetts Institute of Technology, the Warren M. Zapol Professor of Anaesthesia at Harvard Medical School, an Anesthesiologist in the Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, Associate Director of the Institute for Medical Engineering and Science and Co-Director of the Harvard-MIT Health Sciences and Technology Program. His research interests include signal processing algorithms for the study of neural systems, functional neuroimaging, and electrophysiological approaches to study general anesthesia in humans.

Dr. Brown is a member of the Association of University of Anesthesiologists. He is also a Fellow of the American Statistical Association, a Fellow of the American Association for the Advancement of Science, a member of the Institute of Medicine, and a Fellow of the American Academy of Arts and Sciences. He was a recipient of a 2007 NIH Director's Pioneer Award, the 2011 recipient of the National Institute of Statistical Sciences Sacks Award for Outstanding Cross-Disciplinary Research and a 2012 NIH Director's Transformative Research Award.