J Neurophysiol 98: 2456–2475, 2007.
First published May 23, 2007; doi:10.1152/jn.01118.2006.

General-Purpose Filter Design for Neural Prosthetic Devices

Lakshminarayan Srinivasan,[1,2,4] Uri T. Eden,[5] Sanjoy K. Mitter,[2] and Emery N. Brown[3,4,6]

[1]Center for Nervous System Repair, Department of Neurosurgery, Massachusetts General Hospital, Boston; [2]Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science and [3]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge; [4]Harvard/MIT Division of Health Sciences and Technology, Cambridge; [5]Department of Mathematics and Statistics, Boston University, Boston; and [6]Neuroscience Statistics Research Laboratory, Department of Anesthesia and Critical Care, Massachusetts General Hospital, Charlestown, Massachusetts

Submitted 20 October 2006; accepted in final form 17 May 2007

Srinivasan L, Eden UT, Mitter SK, Brown EN. General-purpose filter design for neural prosthetic devices. *J Neurophysiol* 98: 2456–2475, 2007. First published May 23, 2007; doi:10.1152/jn.01118.2006. Brain-driven interfaces depend on estimation procedures to convert neural signals to inputs for prosthetic devices that can assist individuals with severe motor deficits. Previous estimation procedures were developed on an application-specific basis. Here we report a coherent estimation framework that unifies these procedures and motivates new applications of prosthetic devices driven by action potentials, local field potentials (LFPs), electrocorticography (ECoG), electroencephalography (EEG), electromyography (EMG), or optical methods. The brain-driven interface is described as a probabilistic relationship between neural activity and components of a prosthetic device that may take on discrete or continuous values. A new estimation procedure is developed for action potentials, and a corresponding procedure is described for field potentials and optical measurements. We test our framework against dominant approaches in an arm reaching task using simulated traces of ensemble spiking activity from primary motor cortex (MI) and a wheelchair navigation task using simulated traces of EEG-band power. Adaptive filtering is incorporated to demonstrate performance under neuron death and discovery. Finally, we characterize performance under model misspecification using physiologically realistic history dependence in MI spiking. These simulated results predict that the unified framework outperforms previous approaches under various conditions, in the control of position and velocity, based on trajectory and endpoint mean squared errors.

INTRODUCTION

Amyotrophic lateral sclerosis, spinal cord injury, brain stem infarcts, advanced-stage muscular dystrophies, and diseases of the neuromuscular junction profoundly disrupt voluntary muscle control. New technologies, variously called brain-computer interfaces (Leuthardt et al. 2004; Wolpaw and McFarland 2004), motor neural prostheses (Carmena et al. 2003; Hochberg et al. 2006; Serruya et al. 2002; Taylor et al. 2002), and cognitive prostheses (Musallam et al. 2004; Shenoy et al. 2003), represent a communication link that bypasses affected channels of motor output. Manually actuated devices, eye tracking, and other approaches (Frey et al. 1995) represent practical solutions for many patients but may not be feasible for individuals with profound motor deficits. Moreover, brain-driven interfaces, based on a variety of actuation technologies including functional electrical stimulation, active orthotics, exoskeletons, robotic arms, and wireless consumer electronics, have the potential to provide dexterous and natural control without muscle fatigue.

A brain-driven interface includes a method to monitor neural activity, an algorithm to map neural activity into intended device behavior, a device to be controlled, and a feedback mechanism from the device to the user (Andersen et al. 2004; Kubler et al. 2006; Lebedev and Nicolelis 2006; Schwartz et al. 2006; Wolpaw et al. 2002). This paper relates to the optimal mapping between preprocessed neural activity and estimates of the user's intention that determine control signals. The method presented here unifies four canonical approaches, demonstrates new applications, and suggests one path to further algorithm development.

In prosthetics literature, the optimal mapping is predominantly described as an estimation (filtering) problem followed by a control problem. First, an estimate of the user-intended prosthetic device state is calculated based on neural activity that serves as a noisy observation of that intention. Second, a control law determines inputs to the device that achieve this estimate of the user-intended device state. This optimization ignores feedback to the user but provides a practical approach that is accommodated within the existing framework of estimation theory or similarly, a tracking problem in stochastic control. The word "intention" is used to describe an objective behavior of the prosthetic device that the subject is driven to exhibit through mechanisms such as reward or gratification that may be intrinsic and extrinsic to the brain.

Previous approaches to the estimation problem include: manually adjusted linear combinations of power spectral band energies (Wolpaw and McFarland 1994), population vectors for automated but suboptimal linear mappings (Georgopoulos et al. 1986), linear regression for optimized linear mappings (Wessberg et al. 2000), support vector machines (Olson et al. 2005), and recursive Bayesian estimation procedures, including the Kalman filter (Wu et al. 2006), particle filter (Brockwell et al. 2004), and point process filter (Brown et al. 1998; Eden et al. 2004a).

Bayesian estimation allows better tracking than linear regression (including population vector based decoding) in offline data analyses in various neural decoding algorithm studies (Barbieri et al. 2005; Brockwell et al. 2004; Brown et al. 1998; Eden et al. 2004a), although this improvement may not be universally observed in practice (Kim et al. 2006). This approach describes the intended state of a prosthetic device and observed neural activity as a sequence of random variables indexed by time. The trajectory model defines a prior on the

Address for reprint requests and other correspondence: L. Srinivasan, Massachusetts General Hospital, Center for Nervous System Repair, 50 Blossom St., EDR-410, Boston, MA 02114 (E-mail: lsrinivasan@partners.org).

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

sequence of intended device states. The observation model defines the relationship between neural activity and intended device states. Actual device states are determined from neural activity based on these trajectory and observation models.

Neural prosthetic device algorithms are assessed a variety of ways, including simulated off-line decoding (Eden et al. 2004a,b; Kemere and Meng 2005; Kemere et al. 2004; Srinivasan and Brown 2007; Srinivasan et al. 2005, 2006), off-line decoding of actual neural activity recorded during movements in laboratory animals (Brockwell et al. 2004; Hatsopoulos et al. 2004; Wessberg et al. 2000), and on-line prototypes involving laboratory animals or humans (Carmena et al. 2003; Hochberg et al. 2006; Leuthardt et al. 2004; Musallam et al. 2004; Serruya et al. 2002; Shenoy et al. 2003; Taylor et al. 2002; Wahnoun et al. 2006; Wolpaw and McFarland 2004; Santhanam et al. 2006). One ultimate gold standard in comparing methods is a double-blinded longitudinal study of device performance in activities of daily living for a population of individuals that experience well-defined stages and pathological mechanisms of a target motor deficit. Among these various methods of assessment, simulation provides an important first step in clarifying essential design concepts, demonstrating mathematical approaches, providing intuition about performance characteristics, and preliminarily assessing relative performance of competing algorithms.

In the following sections, we present an estimation framework for brain-driven interfaces that explicitly allows the designer to span a full range of device capabilities by employing a hybrid state space composed of interacting discrete and continuous valued random processes. This method is shown to generalize previous Bayesian approaches to prosthesis design, including finite state machines (Shenoy et al. 2003), free-arm-movement models (Wu et al. 2006), reaching-movement-trajectory models (Cowan and Taylor 2005; Kemere and Meng 2005; Kemere et al. 2003; Srinivasan and Brown 2006; Srinivasan et al. 2006a; Wu et al. 2004), the switching observation model (Wu et al. 2004), and the mixture-of-trajectories model (Yu et al. 2007). One possible filtering procedure is derived for point process observations on the hybrid state space, and connections are drawn to existing literature on hybrid estimation for Gaussian observation processes (switching Kalman filters).

To demonstrate the versatility of this framework, two prosthetic device applications are described in simulation. The first application uses ensemble spiking activity from point process models of primary motor cortex (MI) to coordinate reaching movements to a target that may change within a discrete set of targets over the course of the movement. The second application uses power in EEG bands to navigate a wheelchair with definitive stopping. The reaching example is further expanded into a third application that demonstrates adaptive filtering under neuron death and discovery. In the final application, the effect of model misspecification is examined by incorporating physiologically realistic history dependence into simulated MI spiking activity.

METHODS

Elements of the hybrid framework

In the formulation of the neural prosthesis estimation problem, the user communicates the intended state of the prosthetic device via neural signals. The optimal brain-driven interface must convert these neural signals into an estimate of the intended device state that minimizes some distance metric (cost) to the intended sequence of device states. The cost is commonly assumed to be some form of mean squared error for continuous-valued device states and frequency of proper classification for discrete-valued states (Andersen et al. 2004; Kubler et al. 2006; Lebedev and Nicolelis 2006; Schwartz 2004; Wolpaw et al. 2002). Implicit in this formulation, a controller is subsequently expected to receive the estimate and drive the device to the corresponding state with the required precision and response time.

To maintain generality, we describe the user-intended device state at time step k by a vector of discrete random variables s_k and continuous random variables x_k. The user drives neural activity $n_k^{1:C}$ from C channels at time step k based on the desired device states s_k and x_k. Although we refer to n_k^c as the activity of the cth neuron in the specific context of multiple single-unit recording, n_k^c may correspond more generally to the cth signal of any measurement of activity at time step k, including single neuron spiking, multiunit activity, continuous electric field measurements, and even eye movements. Neural activity that leads or lags the desired device states can be incorporated by supplying the appropriate advanced or delayed neural activity in place of the current neural activity. As an example of discrete and continuous-valued states in the context of a classical center-out reaching task, delay period spiking activity in dorsal premotor cortex can be described as neural activity driven by the upcoming target [represented by a discrete set of states (Hatsopoulos et al. 2004; Shenoy et al. 2003; Srinivasan et al. 2006b)], and peri-movement spiking activity in primary motor cortex can be described as neural activity driven by arm velocity [represented by a continuous set of states (Georgopoulos et al. 1986; Hatsopoulos et al. 2004; Moran and Schwartz 1999; Serruya et al. 2002; Wessberg et al. 2000)]. However, neural observations of both discrete and continuous states are not necessary. The history of activity at this time step, $H_k = (n_1^{1:C}, n_2^{1:C}, \ldots, n_{k-1}^{1:C})$, may also affect $n_k^{1:C}$ due to recurrent neural connections and other sources of history dependence.

As an illustration of these variables, consider driving a car with your EEG instead of with your arms. Your intention to increment or decrement the gear as well as the current gear position can be captured by a discrete variable s_k, whereas the desired wheel or gas pedal angles can be further described by the continuous variable x_k that evolves depending on the resulting gear position recorded in s_{k+1}. The EEG amplitudes on C different channels, indicated by $n_k^{1:C}$, may depend on your discrete and continuous-valued intentions for the car but also the history of previous amplitudes H_k because EEG typically exhibits a power spectral density that is shaped rather than flat. Note that the intended device state need not correspond literally to parts of the car as with the intention to increment or decrement the gear. The choice of states can dramatically impact ease of use, just as with the design of an interface to a consumer electronic device such as the MP3 player.

We define the hybrid state space as a joint probability density on the entire temporal sequence (trajectory) of intended states and neural activity $p(x_1, s_1, n_1^{1:C}, x_2, s_2, n_2^{1:C}, \ldots)$. Graphical models on acyclic graphs are a pictorial description of this joint density (Jordan 2004). The graphical model allows us to constrain the form of the joint density with a simple and consistent prescription. Consider our specific graphical model of the hybrid state space (Fig. 1*A*) in which we have illustrated only one segment of the full graphical model that corresponds to the entire trajectory. The circles, called nodes, denote random variables corresponding to the intended states and neural activity. The arrows specify interdependencies between the random variables. A consistent prior distribution on the entire set of nodes is provided by specifying distributions for each node conditioned on its parents, which are all nodes at the base of arrows that point to that node. Nodes without parents require unconditioned priors. The graphical model imposes a Markov structure, where any node is independent of all other nodes when conditioned on its parents. The hanging

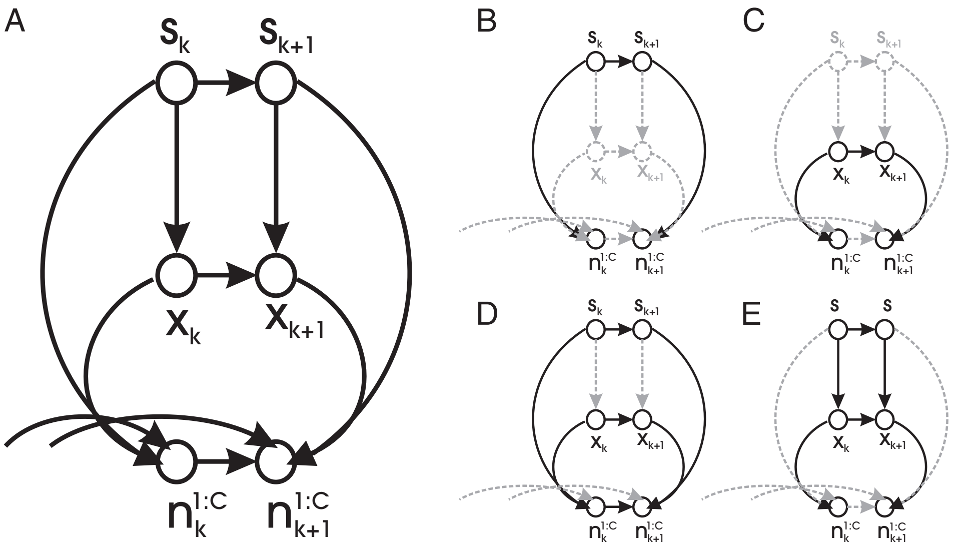

FIG. 1. Graphical models of the hybrid state space framework and 4 canonical approaches to estimation for prosthetic devices. Nodes represent random variables, and 1 probability density is specified for each node conditioned on its parents. Dashed edges are interactions that are possible in the hybrid framework but that are not represented in the canonical approach. *A*: hybrid framework represents a specific relationship among the sequence of neural signals ($n_k^{1:C}$), user-intended continuous states (x_k), and user-intended discrete states (s_k) that encompasses and extends previous approaches (*B–E*). *B*: finite state machine description of cognitive prostheses (Shenoy et al. 2003). *C*: free arm movement models (Wu et al. 2006) and reaching movement trajectory models (Cowan and Taylor 2005; Kemere and Meng 2005; Kemere et al. 2003; Srinivasan and Brown 2006; Srinivasan et al. 2005; Yu et al. 2005b). *D*: switching observation model (Wu et al. 2004) for poorly sorted spike trains. *E*: mixture-of-trajectories model (Yu et al. 2005a) for movements to stationary discrete targets.

arrows directed toward $n_k^{1:C}$ and $n_{k+1}^{1:C}$ represent history dependence in the neural activity.

The probability distribution $p(n_k^{1:C}|x_k, s_k, H_k)$ associated with this hybrid state space corresponds to the observation model because it relates the present measurement of neural activity to the present intention of the user and the history of neural activity. The probability distributions $p(x_{k+1}|x_k, s_{k+1})$ and $p(s_{k+1}|s_k)$ comprise the trajectory model; they describe the frequency and types of transitions in user intent for which the prosthetic device is designed.

The discrete probability distribution $p(s_{k+1}|s_k)$ in the trajectory model can be conveniently summarized as a discrete state transition matrix **M**

$$M_{i,j} = p(s_{k+1} = i \,|\, s_k = j) \qquad (1)$$

This notation means that the entry in the *i*th row and *j*th column of **M** corresponds to $p(s_{k+1} = i|s_k = j)$, the probability of desiring state *i* at time step $k + 1$ given that state *j* was desired at time step *k*.

In principle, the observation model should properly describe the relation between neural observations and user intent, and the trajectory model should accurately reflect the distribution of user intents. Model mismatch describes errors that accumulate from an incorrect model specification. Whereas continuous field potentials (LFP, ECoG, EEG) are typically described by Gaussian observation models (Tarvainen et al. 2004), spiking activity at millisecond resolution is better described by point process observation models (Brown 2005; Brown et al. 2002; Daley and Vere-Jones 2003; Snyder and Miller 1991). The continuous component $p(x_{k+1}|x_k, s_{k+1})$ of the trajectory model can often be reasonably approximated as Gaussian to anticipate smooth changes in the user's continuous state intent when conditioned on a particular discrete state. The discrete component of the trajectory model $p(s_{k+1}|s_k)$, also called the state transition density, is generally defined by specifying a probability between 0 and 1 for each possible pair of intentions (s_{k+1}, $\mathrm{s_k}$), although parameterization may be relevant to dealing with a large set of discrete intentions.

Alternate connections could have been used to describe the hybrid state space. The specific choice of connections made (Fig. 1*A*) draws on a standard form used in hybrid filtering on Gaussian observations (Murphy 1998; citeseer.ist.psu.edu/article/murphy98switching.html) but extends it to accommodate arbitrary history dependence. This imposed structure on the state space makes it easy to obtain estimates of the intended device state in a recursive fashion based on the latest set of neural activity. Moreover, the connections are sufficiently general as to accommodate a diverse set of applications.

Five previously described Bayesian approaches to neural prosthetics fall within this single framework (Fig. 1, *B–E*). A finite state machine description of the prosthesis (Shenoy et al. 2003) consists of a sequence of discrete user-intended states, rules for transitions between those states, and a relationship between states and neural signals (Fig. 1*B*). Free-arm-movement models (Wu et al. 2006) and reaching-movement-trajectory models (Cowan and Taylor 2005; Kemere and Meng 2005; Kemere et al. 2003; Srinivasan and Brown 2007; Srinivasan et al. 2005; Srinivasan et al.; Yu et al. 2005b), both describe continuous-valued arm-movement intentions that drive neural activity (Fig. 1*C*). The switching observation model (Wu et al. 2004) accommodates poorly sorted neural activity that might be better described by combinations of single-cell receptive fields (Fig. 1*D*). For example, multimodal multiunit activity can be described as a probabilistic combination of two or more unimodal tuning curves. The mixture-of-trajectories model (Yu et al. 2007) was designed for continuous-valued reaching movements to a stationary target drawn from a discrete set (Fig. 1*E*).

In subsequent discussion, we use some of these model names more generally to describe their use in situations for which they were not originally applied. We use "free movement" in reference to free-arm-movement models, but where the continuous state is not necessarily related to an arm, but rather some application-specific state, like wheelchair velocity. Similarly, we use "mixture of trajectories" more generally to denote any model with a fixed discrete state regardless of the specific application.

Because the hybrid framework is sufficiently general, these various applications can be implemented by appropriately defining states, the trajectory model, and the observation model. For example, the hybrid framework is equivalent to the free-movement model when defining only one possible discrete state, and applying a free-movement model to define the continuous component $p(x_{k+1}|x_k, s_{k+1})$ of the trajectory model. Similarly, the hybrid framework is equivalent to the mixture-of-trajectories model when the discrete state transition matrix **M** is assigned to be the identity matrix **I**. With $\mathbf{M} = \mathbf{I}$, the probability of transitioning to a different state from the current state is zero, capturing the notion of a static discrete state.

Although the hybrid state space depicted (Fig. 1) unifies these previous conceptions of neural prosthesis design, an estimation procedure (filter) must still be specified to generate probability densities of intended device states given neural activity from which average cost measures can be minimized. The resulting hybrid filter procedure can then be used to describe filtering in any of these previous conceptions. In the following sections, we develop a point process filter for spiking observations in the hybrid state space and review the corresponding Gaussian process filter for continuous field potentials.

Point process models of ensemble spiking activity

Neural activity in the form of action potentials is a sequence of transient spiking events. We first specify an observation model that captures the quality of temporally localized events as well as possible dependencies between neurons in an ensemble.

Signals of this nature are naturally described by point process observation models (Brown 2005; Brown et al. 2002; Daley and Vere-Jones 2003; Snyder and Miller 1991). The crux of the point process description of a single neuron is its conditional intensity function. This is the instantaneous probability of firing as a function of elapsed time t and generally conditioned on continuous-valued signals $x(t)$, discrete-valued signals $s(t)$, and spiking history $H(t)$, where $N(t)$ denotes the total number of spikes generated by the neuron since some arbitrary starting time

$$\lambda(t|x(t), s(t), H(t)) = \lim_{\Delta\to 0} \frac{Pr(N(t+\Delta) - N(t) = 1|x(t), s(t), H(t))}{\Delta} \tag{2}$$

We introduce additional notation to accommodate a population of neurons in a discrete time setting. For the kth discrete time step of length δ_k seconds, the conditional intensity of neuron c is represented as λ_k^c in units of spikes per second. Spiking activity at the kth time step is summarized by a vector $n_k^{1:C} = (n_k^1, n_k^2, \ldots, n_k^C)$ of binned spike counts. The cth element of $n_k^{1:C}$ contains the total number of spikes generated by the cth neuron in the respective δ_k-second interval. The observation model for the spiking activity $n_k^{1:C}$ of an ensemble of C neurons binned at δ_k second intervals is approximated (Eden et al. 2004a) as follows

$$p(n_k^{1:C}|x_k, s_k, H_k) \propto \prod_{c=1}^{C} \exp(n_k^c \log(\lambda_k^c\delta_k) - \lambda_k^c\delta_k) \tag{3}$$

where $H_k = (n_1^{1:C}, n_2^{1:C}, \ldots, n_{k-1}^{1:C})$ is the history of spiking activity at step k for the ensemble. This is an approximation in two regards. First, neurons are assumed to be independent conditioned on the history of population activity and current intended device state. This assumption still captures a causal notion of probabilistic dependence among neurons, that for example, the spiking history of one neuron might affect the present spiking probability of another neuron. Second, the discrete time observation model in 3 approximates the exact continuous-time observation model for a point process (Truccolo et al. 2005).

Filtering spikes with the hybrid framework

To develop an estimation procedure that maps spikes to hybrid device states, we looked to the switching Kalman filter (Bar-Shalom et al. 2001; Murphy 1998; citeseer.ist.psu.edu/article/murphy98switching.html) which maps Gaussian signals to hybrid device states. We could possibly bin the spike trains (lump them into sequential intervals of time) and then apply a standard switching Kalman filter. However, spike trains that have been binned (lumped into sequential intervals of time) may only satisfy the Gaussian assumption as the bin size grows. This results in a tradeoff between the user's control of when an action is supposed to happen versus how it is supposed to happen.

To avoid this tradeoff, we used the point process observation model (previous section) as a statistical description of spiking that is accurate on a millisecond-by-millisecond time scale (Brown et al. 1998; Truccolo et al. 2005). This necessitated the development of a point process filter for hybrid states. Just as there are several approaches to the switching Kalman filter that balance computational complexity and accuracy (Bar-Shalom et al. 2001), there are several possible ways to filter spikes for the hybrid framework.

Our point process filter is adapted from a mixture-of-Gaussians switching Kalman filter called the Interacting Multiple Model (IMM) (Bar-Shalom et al. 2001). The IMM has been a popular choice to balance complexity and accuracy for a wide array of Gaussian applications that track multiple targets with multiple sensors, such as camera-based human motion tracking (Farmer et al. 2002), and radar-based airplane tracking (Mazor et al. 1998). The specific constraints of an application dictates the proper balance between complexity and accuracy, which is commonly determined through empirical testing. In the prosthesis application, demands for hardware memory, energy consumption, and computation speed must be weighed against the accuracy needed for the user to achieve ease of use and reliability. Although the IMM cannot be applied directly to point process observations, it informs our approach to hybrid filtering for spikes because prosthesis applications share the same essential structure as previous multi-target multi-sensor applications. We summarize the point process hybrid filtering procedure in the following section.

By combining this observation model and the hybrid state space defined in the previous section, we derive a specific filter to estimate hybrid device states from ensemble activity that has been modeled as a point process (APPENDIX C). The exact, continuous-time solution to this filtering problem is a stochastic partial differential equation known as point process filtering with jump-Markov processes (McGarty 1974). The method in this section is one possible approximation.

Our point process filter derivation first manipulates probability densities without specifying their functional form and later introduces the functional form of the point process observation model given in *Eq. 3.* The resulting point process filter retains the same flavor as the IMM filter. Just as the IMM filter involves a bank of Kalman filters that run in parallel, our hybrid filter employs a bank of stochastic state point process filters (Eden et al. 2004a) that run in parallel, one for each possible value of the discrete state at a particular time step. A practical note on numerical issues for implementation is given in APPENDIX F.

Spike filtering with the hybrid framework in nine steps

Each iteration of the point process hybrid filter involves nine basic steps. The quantities $p(s_k|H_{k+1})$ and $p(x_k|s_k, H_{k+1})$ come from the previous iteration, where $p(s_1)$ and $p(x_1)$ are used instead for the first iteration. A Gaussian approximation to a probability density on the continuous state x_t is specified by a mean and covariance matrix. A probability mass function on the discrete state s_k is specified by a list of probabilities for each possible value of s_k. See APPENDIX F for a practical note on numerical issues.

Step 1

Compute $p(s_{k+1}|H_{k+1}) = \Sigma_{s_k} p(s_{k+1}|s_k)p(s_k|H_{k+1})$

Step 2

Compute $p(s_k|s_{k+1}, H_{k+1}) = \dfrac{p(s_{k+1}|s_k)p(s_k|H_{k+1})}{\Sigma_{s_k} p(s_{k+1}|s_k)p(s_k|H_{k+1})}$

Step 3

Approximate $p(x_k|s_{k+1}, H_{k+1}) = \Sigma_{s_k} p(s_k|s_{k+1}, H_{k+1})p(x_k|s_k, H_{k+1})$ with the Gaussian approximation to mixtures of Gaussians (see METHODS).

Step 4

Calculate the Gaussian approximation to $p(x_{k+1}|s_{k+1}, n^{1:C}_{k+1}, H_{k+1})$. Specifically, for each value that s_{k+1} can take on, send $p(x_k|s_{k+1}, H_{k+1})$ through one full iteration of a point process filter (see Appendix A) with observation equation $p(n^{1:C}_{k+1}|x_{k+1}, s_{k+1}, H_{k+1})$ and state equation $p(x_{k+1}|x_k, s_{k+1})$. Retain these densities (1 Gaussian for each possible value of s_{k+1}) for the next iteration.

Step 5

Calculate

$$p(n^{1:C}_{k+1}|s_{k+1}, H_{k+1}) \approx \frac{|W_{k+1|\mathrm{k+1},\, s_{\mathrm{k+1}}}|^{1/2}}{|_{k+1}|^{1/2}} \prod_{c=1}^{C} \exp(n^c_{k+1}\log(\lambda^c_{k+1}\delta_{k+1}) - \lambda^c_{k+1}\delta_{k+1})|_{\mathrm{x_{k+1}=x_{k+1|k,s_{k+1}}}}.$$

Note that $W_{k+1|k+1,s_{k+1}}$ and $W_{k+1|k,s_{k+1}}$ are posterior and prediction covariance terms from step 4 (see also APPENDIX A). This is the Laplace approximation (see APPENDIX E). Here, λ^c_{k+1} denotes the conditional intensity of neuron c at time step $k + 1$, which may be a function of s_{k+1}, x_{k+1}, and H_{k+1}.

Step 6

Calculate $p(s_{k+1}|n^{1:C}_{k+1}, H_{k+1}) = \dfrac{p(n^{1:C}_{k+1}|s_{k+1}, H_{k+1})p(s_{k+1}|H_{k+1})}{\sum_{s_{k+1}}p(n^{1:C}_{k+1}|s_{k+1}, H_{k+1})p(s_{k+1}|H_{k+1})}$. Retain this density for the next iteration.

Step 7

Calculate $p(x_{k+1}|n^{1:C}_{k+1}, H_{k+1}) = \sum_{s_{k+1}}p(x_{k+1}|s_{k+1}, n^{1:C}_{k+1}, H_{k+1})$ $p(s_{k+1}|n^{1:C}_{k+1}, H_{k+1})$ using the results from steps 4 and 6.

Step 8

Choose the discrete and continuous device states for step k +1 based on steps 6 and 7 and your cost function. For example, to approximately minimize average classification error, choose the value of s_{k+1} that maximizes step 6. To approximately minimize mean squared error, choose the average value of x_{k+1} under the density calculated in step 7.

Step 9

Return to step 1 for the next time step.

Filtering continuous field potentials with the hybrid framework

Continuous field potentials are also viable sources for the control of prosthetic devices, such as with EEG (Wolpaw and McFarland 2004), ECoG (Leuthardt et al. 2004), and LFP (Pesaran et al. 2006). An EEG-based device has the potential for wide application because it is completely noninvasive. The ECoG and LFP approaches may allow cheaper and more robust hardware solutions than spike-driven interfaces because skull screws and coarse electrodes may suffice for these signals where micromachined multiunit arrays are needed to record stable ensemble spiking activity.

The physiological basis of these continuous field potentials is varied and different from that of ensemble spiking activity. Additional research is needed to understand effective training paradigms and hardware design as they pertain to each of these signal sources. However, existing filtering procedures are sufficient to incorporate these signals into the hybrid framework (Bar-Shalom et al. 2001). This is because continuous field potentials have been extensively modeled as Gaussian observation processes, including autoregressive moving average (ARMA) models (Tarvainen et al. 2004). As a result, the many types of switching Kalman filter can be applied directly to accommodate these signals into the hybrid framework.

The interacting multiple model (IMM) (Bar-Shalom et al. 2001) is the switching Kalman filter that is analogous to the point process filter presented in the previous section. The IMM derivation can be written in almost the same fashion, except that the observation model 3 is now Gaussian. To specify the Gaussian observation model, represent the C continuous field potentials at time step k in a vector $n^{1:C}_k = [n^1_k, n^2_k, \ldots, n^C_k]'$. The entries in this vector could be field potential amplitudes, power in specific frequency bands, or some more complicated transformation of the raw signals. The Gaussian observation model relates $n^{1:C}_k$ to the intended device states, waveform history H_k, and zero-mean Gaussian observation noise ε_{k} as follows

$$n^{1:C}_k = D_k x_k + f_k(s_k, H_k) + \varepsilon_k \tag{4}$$

The covariance matrix of ε_k is denoted R_k. Here, x_k is a $J \times 1$ vector of continuous states, D_k is a $C \times J$ matrix that may depend on s_k, and $f_k(s_k, H_k)$ is a function that maps neural history to a $C \times 1$ vector, such as with ARMA models. The functional form of f_k and the remaining parameters of the observation model may in general depend on the discrete intended state s_k.

The IMM procedure is simply the point process hybrid filter procedure of the previous section, with modifications to steps 4 and 5. In step 4, Kalman filters are applied instead of point process filters. These are the Gaussian filter *Eqs. A6* and *A7* in APPENDIX A. In step 5, $p(n^{1:C}_{k+1}|s_{k+1}, H_{k+1})$ is calculated based on 4 and the prediction density $p(x_{k+1}|s_{k+1}, H_{k+1})$ from step 4. Specifically, $p(n^{1:C}_{k+1}|s_{k+1}, H_{k+1})$ is a separate Gaussian for each possible value of s_{k+1} with distribution given in *Eq. A8*.

These two modifications, together with the hybrid filter procedure described in the previous section, complete the description of the IMM filter. The IMM filter is illustrated in the following text (RESULTS, *Application 2*) with a wheelchair navigation task with definitive stopping based on EEG-band energy.

RESULTS

We now illustrate four applications of the hybrid filtering framework to brain-driven interfaces: *1*) reaching to discrete targets that switch during movement based on MI ensemble spiking activity; *2*) wheelchair navigation with definitive stopping based on EEG; *3*) adaptive filtering of MI ensemble spiking with ongoing neuron death and discovery; and *4*) filtering under model mis-specification due to physiologically realistic history dependence in MI ensemble spiking.

In each section, we illustrate the relative performance of three filters described in METHODS: free-movement estimation (Fig. 1*C*) (Wu et al. 2006) mixture of trajectory estimation (Fig. 1*E*) (Yu et al. 2007), and the hybrid filtering framework (Fig. 1*A*).

Application 1: reaching to discrete targets that switch during movement

The first application employs the hybrid framework to track a reaching movement with a mid-flight change in the desired target. In the switching-target-reach task, the subject is required to reach to one of a discrete set of targets with a prosthetic arm driven by ensemble spiking activity from MI. Each reach must be completed within 2 s in a two-dimensional plane from the origin to one of eight targets arranged evenly on a circle of 0.25 m radius. In addition, the target changes once during the course of the movement, requiring the user to make a corrective maneuver to the new target location. The initial target and final target are chosen at random with equal probability over the target set. The switch time, unknown to the user, is drawn uniformly between 0.2- and 1.2-s postmovement onset. These parameters are chosen to explore reaching movements at a realistic spatial scale for humans while maintaining peak arm velocities that are comparable to those studied in related primate electrophysiology experiments of MI (Hatsopoulos et al. 2004; Li et al. 2001; Moran and Schwartz 1999).

How can the hybrid framework solve the switching-target-reach task? We begin by defining the continuous variable x_k as the arm state and the discrete variable s_k as the target identity from a set of R targets on a two-dimensional workspace

$$x_i = \begin{bmatrix} x \text{ position coordinate at timestep } k \\ y \text{ position coordinate at timestep } k \\ x \text{ velocity coordinate at timestep } k \\ y \text{ velocity coordinate at timestep } k \end{bmatrix}, \quad s_k \in \{1, 2, \ldots, R\} \tag{5}$$

For this example, consider neural observations n_k described as binned spikes of a point process (see *Point process models of ensemble spiking activity*). The essential structure of this hybrid state space is depicted by the mixture-of-trajectories model (Fig. 1*E*). To support switching targets, this static target diagram is altered by indexing the target s with time as in the switching observation model (Fig. 1*D*) (Wu et al. 2004).

Next we specify the conditional densities corresponding to each edge. Recall from METHODS that the density $p(s_{k+1}|s_k)$ is defined by a state transition matrix **M**

$$M_{i,j} = p(s_{k+1} = i \,|\, s_k = j) \tag{6}$$

This notation means that the entry in the *i*th row and *j*th column of **M** corresponds to $p(s_{k+1} = i|s_k = j)$.

The density $p(x_{k+1}|x_k, s_{k+1})$ constrains the path of a reaching movement for any given target s_{k+1}. This conditional density can be obtained from any of several reaching movement trajectory models (Kemere and Meng 2005; Kemere et al. 2003; Srinivasan et al. 2005; Srinivasan et al.; Yu et al. 2005b), directed specifically to the target location corresponding to s_{k+1}. In this example, we use the standard (unaugmented) reach state equation detailed in (Srinivasan et al. 2005; Srinivasan et al.) based on the following free-movement state equation

$$x_{k+1} = Ax_k + \varepsilon_k \tag{7}$$

The trajectory model described by the hybrid framework is not equivalent to the procedure used to simulate intended movements (described in the opening paragraph of this section). For example, whereas the hybrid framework assigns finite probability to the set of trajectories where there is more than one target switch, the reaching-movement simulation procedure allows only one switch that occurs randomly during the 0.2- to 1.2-s interval of the movement. Nevertheless, as demonstrated in the following text, the hybrid framework trajectory model of this problem will be a sufficiently appropriate description of the intended frequency and type of movement to allow neural observations to push the filter estimates along the simulated trajectory.

The point process observation model (*Eq. 3*) that describes $p(n_k^{1:C}|x_k)$ is specified by a discrete-time conditional intensity function λ_k^c for each neuron c. In this example, we choose a conditional intensity adapted from a model of primary motor cortex (Moran and Schwartz 1999)

$$\lambda_k^c = exp(\beta_0^c + \beta_1^c(v_x^2 + v_y^2)^{1/2} \cos(\theta - \beta_2^c)) \tag{8}$$

$$= \exp(\beta_0 + \alpha_1 v_x + \alpha_2 y_x) \tag{9}$$

where v_x and v_y are the velocity components of x_k in *Eq. 5*. Here we assume that any lag between neural activity and the user's intentions is known and has been corrected to allow for the zero-lag indexing used in the preceding text. In practice, this lag can be estimated as another model parameter.

The parameters of the observation model $p(n_k^{1:C}|x_k)$ can be tuned using point process adaptive filtering (Eden et al. 2004a) that also tracks changes due to neural plasticity. The parameters of the trajectory model in $p(x_{k+1}|x_k, s_{k+1})$ and $p(s_{k+1}|s_k)$ can be optimized a priori to reflect the types and frequency of behaviors that the neural prosthesis expects to support. Alternatively, adaptive methods will need to be developed to track the usage statistics of the device and adjust the trajectory model accordingly. In this example, we give our various competing filters an equal footing by providing them the actual trajectory and observation model parameters where applicable. A caveat is the state transition matrix **M**. In free-movement estimation, this parameter is nonexistent. In mixture-of-trajectories estimation, **M** is equal to the $|S| \times |S|$ identity matrix **I**, where $|S|$ is the number of possible discrete states. For the hybrid framework, this parameter can be tuned to the expected frequency of target switches.

In these applications, we parameterized **M** as follows

$$M = \begin{bmatrix} a & b & \cdots & b \\ b & a & \ddots & \vdots \\ \vdots & \ddots & \ddots & b \\ b & \cdots & b & a \end{bmatrix} \tag{10}$$

where $b = \dfrac{1-a}{|S|-1}$. This implies that transitions to states other than the current state are equiprobable. We found that performance was relatively insensitive to a range of choices between $a = 0.9$ and $a = 0.99$ but changed substantially for $a = 1$. (Although 0.99 is close to 1, this difference is geometrically magnified by successively multiplying 0.99 over multiple time steps.)

With the conditional densities specified, we can now use the hybrid point process filtering framework to drive the prosthetic device with ensemble spiking activity from motor cortex. We compare the performance of the hybrid framework against free-movement estimation and the mixture-of-trajectories model in a simulated analysis of the switching-target-reach task. The free-movement estimation procedure is implemented using a standard point process filter. This is mathematically equivalent to our hybrid framework where each target is given infinite uncertainty. The mixture-of-trajectories estimation method reported in Yu et al. (2007) is modified to implement the same reach state equation that our hybrid filter uses to provide equal grounds for comparison. This is mathematically equivalent to the hybrid filter with state transition matrix $\mathbf{M} = 1$.

We also examine the effect of premovement instructed delay period activity that may be available to the prosthetic device. Such activity is known to provide information about the desired target in posterior parietal cortex (Andersen and Buneo 2002), premotor cortex (Weinrich and Wise 1982), frontal cortex (Schall 2001), and other brain regions. Premovement target information is easily incorporated into the mixture-of-trajectories model and hybrid filter by specifying a nonuniform initial posterior density on the target states $p(s_1)$. We use one fixed moderately informative nonuniform posterior density (see parameter table) to simulate this premovement target information.

These filtering procedures were compared in a simulated version of the switching-target-reach task. The simulation comprised two stages. First, the subject's desired arm movement was generated based on the reach state equation (Srinivasan et al. 2005, 2006a) which is related to the stochastic optimal-control model (Bertsekas 2005; Kemere and Meng 2005; Todorov 2004). Optimal control has previously been described with relation to arm movement (Flash and Hogan 1985; Uno et al. 1989), including more recently, stochastic optimal control (Todorov and Jordan 2002). Second, the corresponding ensemble spiking activity from primary motor cortex (MI) was simulated based on *Eq. 8,* a velocity-tuned point process model of MI spiking activity (Moran and Schwartz 1999; Truccolo et al. 2005), using the time rescaling procedure that has previously been described in detail (Brown et al. 2002).

A summary of parameter choices is given in Table 1. The reach state equations require specification of the noise increment covariance and target uncertainty for each discrete target (Srinivasan et al. 2006). With states corresponding to position and velocity as described in *Eq. 5,* the original noise covariance of ε_k in *Eq. 7* was chosen to be nonzero in the entries corresponding to velocity increment variances

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 \\ 0 & 0 & 0 & q \end{pmatrix} \tag{11}$$

The uncertainty in each target state Π_T was also diagonal with

$$\Pi_T = \begin{pmatrix} r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix} \tag{12}$$

where *q, r,* and *p* are specified in Table 1.

TABLE 1. *Parameters*

Parameter	Value
A. Receptive field parameters of c^th M1 neuron	
$\beta_0{}^c$	2.28
$\beta_1{}^c$	4.67
$\beta_2{}^c$	Drawn uniformly from $[-\pi, \pi]$
B. Reach state equation parameters	
Parameter	Value
Reach distance, m	0.25
Target Positions, degrees	(45, 90, 135, 180, 225, 270, 315, 360)
Tune step, s	0.01
Noise covariance (q), m^2	1×10^{-4}
Reach duration, s	2
Target position uncertainty (r), m^2	1×10^{-6}
Target velocity uncertainty (p), m^2	1×10^{-6}
C. Other motor task parameters	
Parameter	Value
$(p(s_1 = 1), p(s_1 = 2), \ldots p(s_1 = 8))$	(.6, .15, .02, .02, .02, .02, .15)
Switch times, s	(.2, .4, .6, .8, 1, 1.2)
Ensemble sizes (No. of neurons)	(9, 16, 25, 36, 49, 64, 81)
Randomized trials per data point (Figs. 4 and 5)	100

Parameters used in application 1, the switched-target motor reaching task. *A*: receptive field parameters used to generate primary motor cortex (MI) spiking activity based on *Eq. 8*. *B*: parameters of the reach state equation applied to the hybrid framework. The parameters and implementation are identical to the description in the corresponding paper on the reach state equation (Srinivasan et al. 2007). *C*: other task parameters, including premovement target information provided by the probability density $p\ (s_1)$ that is represented as an array of values.

For this example and application 3, movements and spikes were simulated at 0.01-s resolution. A 0.001-s resolution is typically used to simulate spikes with time rescaling, as with application 4. The coarser time resolution results in some distortion of the K-S plots based on time rescaling such as when multiple spikes are collapsed into one spike within a 10-ms interval, but speeds the simulation 10-fold.

The subject's arm movement was governed by the same unaugmented reach state equation (Srinivasan et al. 2006) used in the preceding text to define $p(x_{k+1}|x_k, s_{k+1})$ in the hybrid framework. Arm movement at any given time step followed the reach state equation corresponding to the current target with low target uncertainty (see parameter table). The constants *q, r,* and *p* in that table refer to specific entries of the diagonal matrices for noise covariance and target uncertainty specified previously (Srinivasan et al. 2006). The target itself was allowed to switch once during the course of the movement. The target switch time was assigned at random, uniformly from a discrete set of possible times between 0.2 and 1.2 s post-movement onset, spaced at 0.2 s.

Because ensemble spiking is governed by conditional independence (see *Eq. 3*), the spiking activity of each cell could be generated separately. To generate the spike train of a given cell, the arm trajectory was first passed through the point process model in *Eq. 8.* The conditional intensity generated by the point process model of each neuron was then used to produce ensemble spiking activity based on the time rescaling theorem (Brown et al. 2002).

For each neuron *c*, model parameters β_0^c and β_1^c were chosen (see parameter table) to reflect typical background firing rate and depth of modulation for primate MI neurons during instructed-delay center-out reaching movements (Truccolo et al. 2005). The model parameter β_2^c was drawn randomly from a uniform distribution over $[-\pi, \pi]$ to ensure that preferred directions were represented over all angles over the course of multiple trials. Note, however, that preferred directions were not necessarily evenly distributed on any given trial because of the random variation in drawing from the uniform distribution just once. Neurons in this simulated MI ensemble exhibited background firing rates of 10 spike/s and firing rates of 24.9 spike/s at a speed of 0.2 m/s in the preferred direction.

In total, five filtering procedures were compared in the simulated switching-target-reach task: free-movement estimation, mixture of trajectory estimation, and hybrid filtering, the last two methods being evaluated with and without premovement target information. Figures 2–5 provide a comprehensive view of the ability of these filtering procedures to convert MI spiking activity into reaching movements to switching targets. Figures 2 and 3 show sample trajectories driven by ensemble spiking activity under the various estimation procedures for a population of 25 neurons with a target switch at 1-s postmovement-onset. Figures 4 and 5 characterize how filter performance scales with ensemble size and target switch time for each of these procedures.

Sample decoding results from one trial without premovement target information (Fig. 2) show that the hybrid framework combines the strengths of free-movement estimation and the mixture-of-trajectories model. By incorporating target information, the hybrid framework and mixture-of-trajectories estimates drive the prosthetic arm to rest at the desired target location, whereas free-movement estimation leaves the arm displaced from the target and still moving at the 2-s mark. However, this same target information also causes the mixture-of-trajectories estimate to pull toward each passing target late in the reach (Fig. 2*A*). This "gravity effect" is reflected in the target probabilities under the mixture-of-trajectories model (Fig. 1*C*). In the second half of the reach, the current heading causes the passing targets (black and red lines) to quickly become highly likely, drawing the trajectory estimate toward those corresponding target locations. The hybrid framework overcomes this problem because it anticipates that targets may switch. By choosing a = 0.99 in the state transition matrix, the target densities (Fig. 1*D*) decay with time, and additional supporting neural activity is required to drive the probability of any given target to dominate the others. This mollifies the gravity effect of the mixture-of-trajectories model.

The hybrid filter also handles premovement target information differently from the mixture-of-trajectories model (Fig. 3). With the premovement information, the first target's probability under the mixture-of-trajectories model (blue line, Fig. 3*C*) approaches certainty faster than before (Fig. 2*C*). However, single trial decoding results (Fig. 3, *A* and *B*) show that the mixture-of-trajectories estimate appears to persist to the original target location even when the desired trajectory has begun to reorient to the new target. This is also seen in the target probabilities (Fig. 3*C*) where the first target (blue) dominates

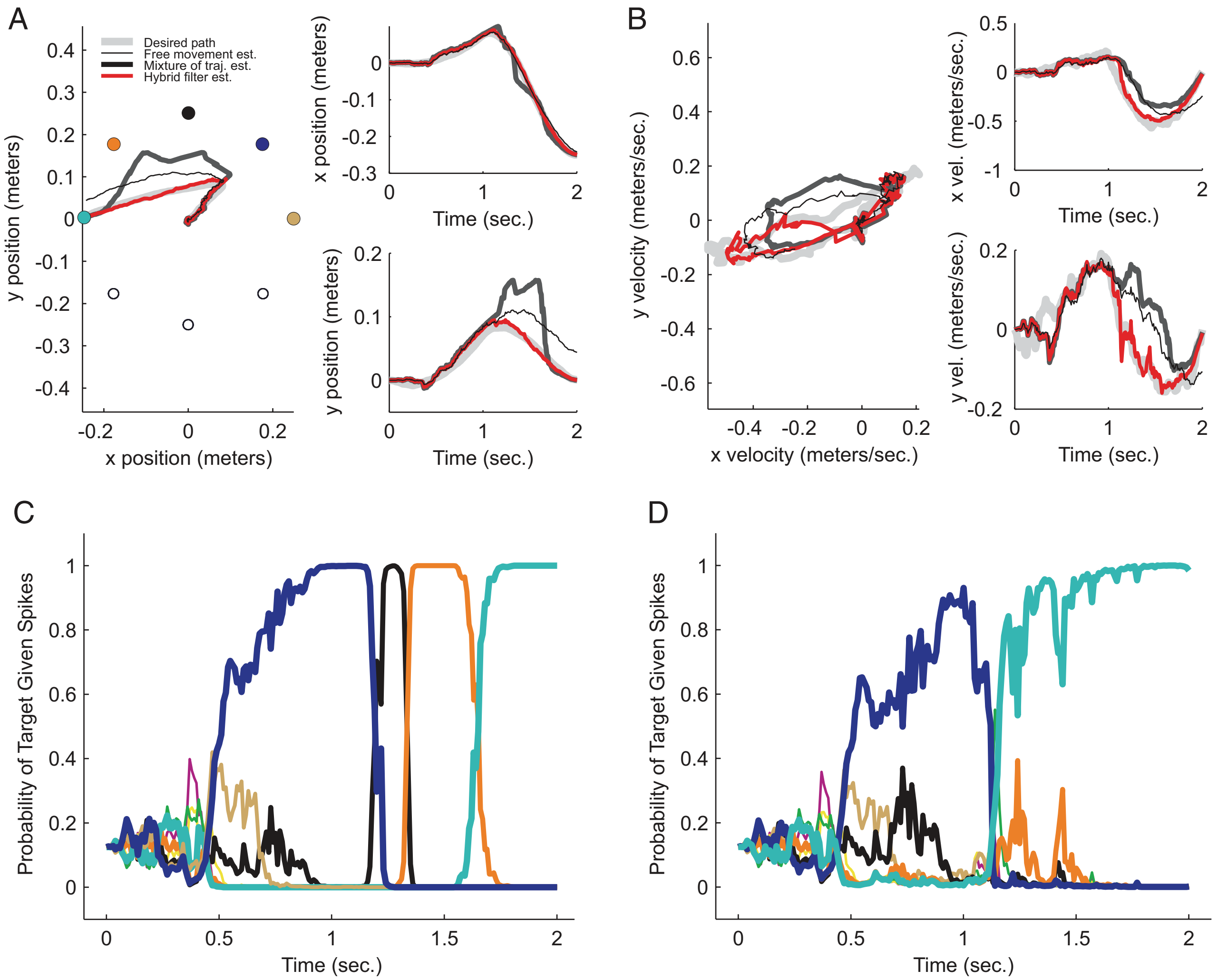

FIG. 2. Decoding results without premovement target information from 1 trial of the simulated switching target motor reaching task on a 2-dimensional plane using 25 motor cortical neurons. *A*: position trajectories, including the desired arm path (light gray), hybrid filter (red), mixture-of-trajectories model (dark gray), and free-movement estimation (thin black). *B*: velocity trajectories with the same color scheme. *C*: probability of each target given ensemble spiking as the reach progresses as determined by the mixture-of-trajectories model and (*D*) the hybrid filter. Colors in *C* and *D* correspond to these targets: the primary target at 45° (blue), final target at 180° (cyan), and the neighbors at 0° (gold), 90° (black), and 135° (orange).

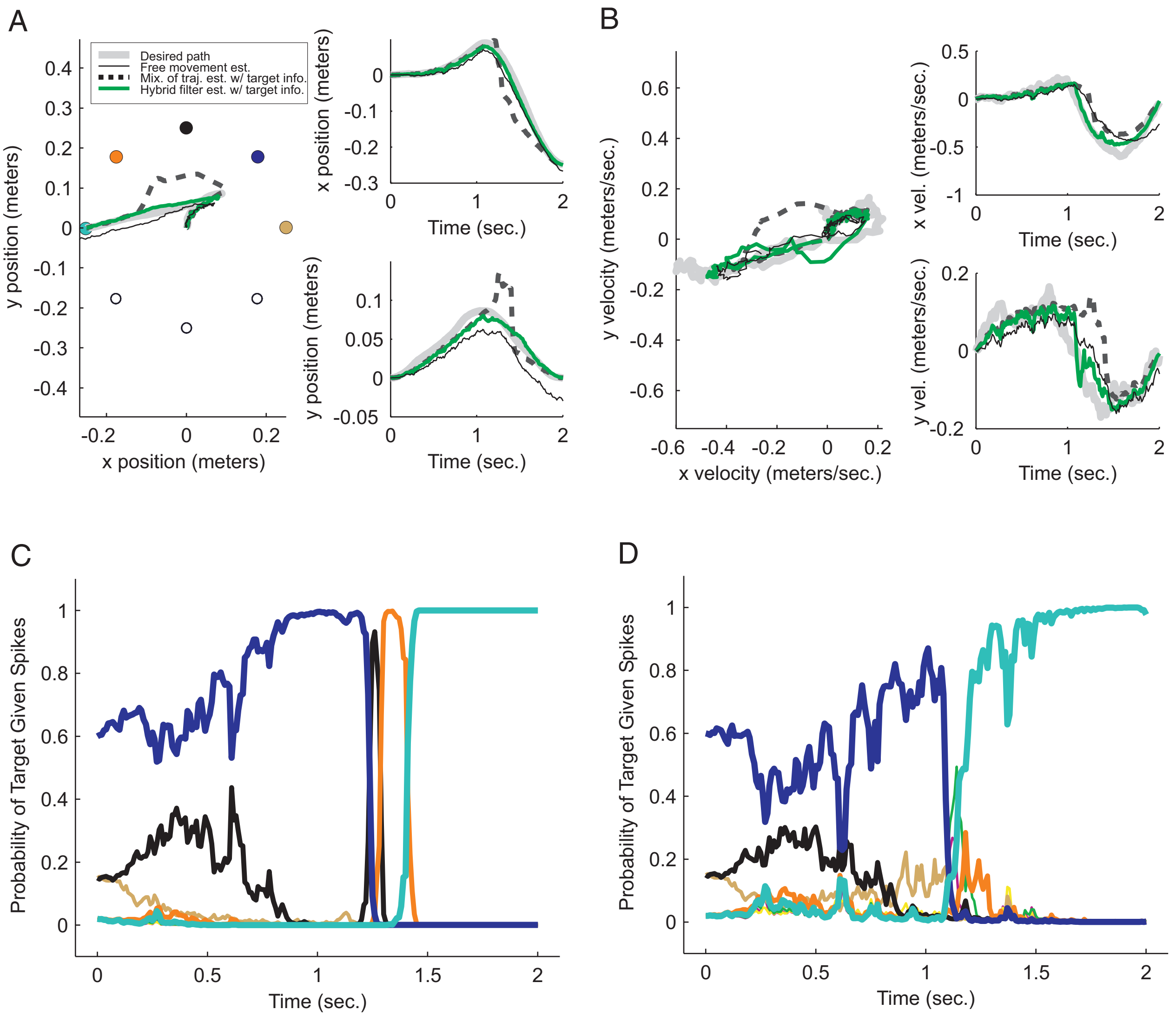

FIG. 3. Decoding results with premovement target information from 1 trial of the simulated switching target motor reaching task on a 2-dimensional plane using 25 motor cortical neurons. *A*: position trajectories including the desired arm path (light gray), hybrid filter (green), mixture-of-trajectories model (dark gray), and free-movement estimation (thin black). *B*: velocity trajectories with the same color scheme. *C*: probability of each target given ensemble spiking as the reach progresses, as determined by the mixture-of-trajectories model and (*D*) the hybrid filter. Colors in *C* and *D* follow Fig. 2.

200 ms beyond the time of the target switch. In contrast, the hybrid framework incorporates the target information early in the reach but progressively "forgets" or downweights its influence because it anticipates the possibility of a target switch, again by using $a = 0.99$. The free-movement estimate does not incorporate premovement information.

We next examined filter performance over a wide range of ensemble sizes, ranging from 15 to 80 neurons. Root mean squared (RMS) error was evaluated in two ways: averaged over the entire trajectory (Fig. 4, *A* and *B*) and over the endpoint at the 2-s mark (Fig. 4, *C* and *D*). Additionally, we examined the fidelity of position tracking (Fig. 4, *A* and *C*) and velocity tracking (*B* and *D*) separately. RMS error decreases for all five methods with larger ensemble sizes.

Trajectory RMS errors are typically smaller than endpoint RMS errors because trajectories begin with the accurate initial condition and accumulate error with time. All methods appear to perform equally well in endpoint error except free-movement estimation which does not incorporate the discrete target locations. Moreover, endpoint error appears to level out faster than trajectory error. This is likely due to the fact that just a few MI neurons are needed to make an accurate target classification, and once the accurate classification is made, the mixture-of-trajectories and hybrid framework methods will drive the prosthetic arm to rest at that target.

Premovement target information appears to provide a slight or insignificant improvement, but this is largely due to the moderate information provided by our choice of initial target prior. Higher fidelity premovement target information will likely make overshooting more pronounced in mixture-of-trajectories estimation and decrease RMS error in the first half of the reach generated by hybrid estimation.

Earlier target switches are easier to track for all methods than later target switches (Fig. 5) for a population of 25 neurons. Later switch times require faster velocity corrections, causing trajectory RMS error to rise across all meth-

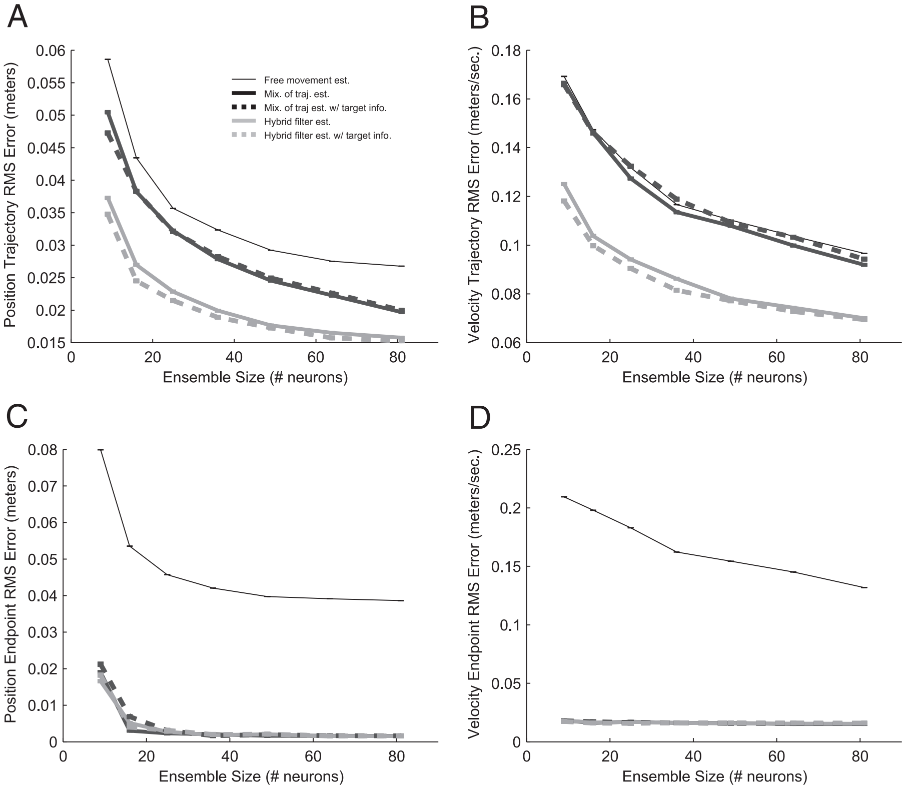

FIG. 4. Various root mean squared (RMS) error performance metrics vs. ensemble size for free-movement estimation (thin black), mixture-of-trajectories model (solid dark gray), hybrid filter (solid light gray), and versions of the latter 2 filters with premovement target information (dashed lines). Switch times are drawn uniformly from the set {0.2, 0.4, 0.6, 0.8, 1, 1.2} in units of seconds. Error bars represent SE across 100 randomized trials for each mean. *A*: RMS error averaged over the entire position trajectory. *B*: RMS error averaged over the entire velocity trajectory. *C*: RMS error averaged only over the endpoint position. *D*: RMS error averaged only over the endpoint velocity.

ods (Figs. 5, *A* and *B*). Trajectory RMS error accumulates rapidly with later switch time for the mixture-of-trajectories model that lags in reorienting the arm movement unlike hybrid estimation, which anticipates switching and reorients quickly.

Endpoint errors (Fig. 5, *C* and *D*) under mixture-of-trajectories and hybrid estimation are largely insensitive to switch time in contrast to free-movement estimation. For mixture-of-trajectories and hybrid estimation, neural observations after the switch are sufficient to classify the target correctly, and be-

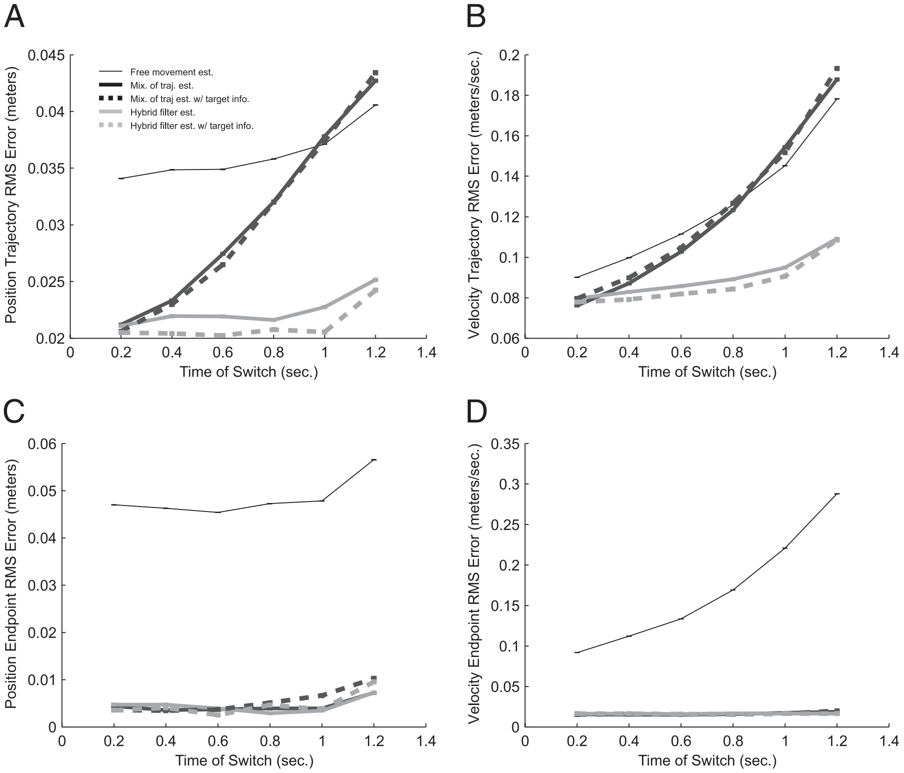

FIG. 5. Various RMS error performance metrics vs. the time postmovement onset at which the target switches, for free-movement estimation (thin black), mixture-of-trajectories model (solid dark gray), hybrid filter (solid light gray), and versions of the latter 2 filters with premovement target information (dashed lines). Ensemble size is fixed at 25 neurons. Error bars represent SE across 100 randomized trials for each mean. *A*: RMS error averaged over the entire position trajectory. *B*: RMS error averaged over the entire velocity trajectory. *C*: RMS error averaged only over the endpoint position. *D*: RMS error averaged only over the endpoint velocity.

cause these latter methods incorporate the set of target locations, the prosthesis movement can reliably converge to the target.

Receiving information from the premovement activity that results in an incorrect maximum likelihood target classification is comparable to a zero second switch time because in both cases, premovement activity initially pushes path estimates toward the wrong final target. This represents the easiest case for tracking switching movements because subsequent neural activity over the full interval of reach time is available to correct estimates toward the final target. In later switches, shorter intervals of neural activity are available to redirect the arm movement. This is the regime where hybrid estimation shows marked improvement over the mixture-of-trajectories approach.

The simulation predicts that performance breaks down for all methods under moderate ensemble sizes for very late switches, where the target can no longer be reliably identified and high-velocity corrective movements must be tracked. A more subtle trend (Fig. 5, *A* and *B*) shows that free-movement estimation performs substantially worse than the mixture-of-trajectories model for early switch times but slightly better in very late switch times. These very late switch times make the overshoot and gravity effects of the mixture-of-trajectories model so pronounced that resulting trajectory estimates accumulate more RMS error than even the free-movement model.

Application 2: EEG-based wheelchair navigation with definitive stopping

Motorized wheelchairs allow users with severe motor deficits to navigate within and outside the home. EEG has previously been suggested as one mechanism by which users could specify movements of their wheelchairs (Wolpaw et al. 2002). Although recordings from scalp-based EEG leads are susceptible to task-independent neural activity and a variety of artifacts, basic two dimensional cursor control was recently demonstrated using EEG in an on-screen center-out reaching task (Wolpaw and McFarland 2004). In the wheelchair application, the ability to reliably stop and start movements is critical to safety and functionality. Definitive stopping (and starting) is essential in a variety of practical settings, such as crossing streets, boarding a subway train, and navigating a grocery store aisle.

In this application, we simulated 10 user-intended consecutive point-to-point movements of a wheelchair in a 10 × 10-m workspace, punctuated by periods of rest (Fig. 6 *B–G*). The intended movement kinematics were simulated based on a velocity-based linear quadratic control model used previously in the neural prosthetic device literature (Bertsekas 2005; Kemere and Meng 2005). Endpoints were drawn uniformly over the workspace, rest periods were drawn uniformly between 0 and 5 s, and travel times for each movement were calculated based on average velocities drawn uniformly between 0.5 and 2 m/s. Kinematics were simulated in discrete time increments of 100 ms.

Power in EEG-bands from 20 electrodes were simulated (Fig. 6*A*) at each time step *k* based on the user-intended wheelchair kinematics

$$\begin{bmatrix} y^1 \\ \vdots \\ y^{20} \end{bmatrix}_k = \begin{bmatrix} 0 & 0 & a^1 & b^1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a^{20} & b^{20} \end{bmatrix} \begin{bmatrix} x \text{ position} \\ y \text{ position} \\ x \text{ velocity} \\ y \text{ velocity} \end{bmatrix}_k + \varepsilon_k \qquad (13)$$

This equation represents EEG-band energy as a linear function of user-intended wheelchair velocity, corrupted by zero-mean additive Gaussian noise ε_k. A very similar relationship was described in recent EEG (Wolpaw and McFarland 2004) and ECoG (Leuthardt et al. 2004) implementations. Although drawn independently at each time step, the additive noise is correlated across leads, as given by the following 20 × 20 noise covariance matrix

$$\text{cov}(\varepsilon_k) = \begin{bmatrix} 5 \times 10^{-2} & 10^{-4} & \cdots & 10^{-4} \\ 10^{-4} & 5 \times 10^{-2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 10^{-4} \\ 10^{-4} & \cdots & 10^{-4} & 5 \times 10^{-2} \end{bmatrix} \qquad (14)$$

Three filters discussed in METHODS were designed to reconstruct intended wheelchair movement from EEG-band energy: free-movement estimation (Fig. 1*C*) (Wu et al. 2006), mixture of trajectory estimation (Fig. 1*E*), and the hybrid filter (Fig. 1*A*). With continuous-valued observations, free-movement estimation is equivalent to the standard Kalman filter (see Kailath et al. 2000 or APPENDIX A), whereas mixture of trajectory estimation and hybrid filtering are akin to the interacting multiple model (IMM) framework (see Bar-Shalom et al. 2001) (see also METHODS). All three filters were provided exact parameters corresponding to the EEG model (*Eq. 13*) as equal grounds for comparison, although these parameters can be learned in practice with a training set based on standard approaches such as multiple linear regression (Kailath et al. 2000). Mixture of trajectory estimation and the hybrid filter require that we define discrete state variables, continuous state variables, and their interdependence at each time step *k*. The starting and stopping behaviors are naturally described as discrete states: $s_k \in \{\text{stopped, moving}\}$. The intended two dimensional position and velocity of the wheelchair corresponds to a continuous state x_k, expressed as the kinematic vector on the right hand side of *Eq. 13* in the preceding text. In both mixture of trajectory and hybrid filter implementations of the wheelchair system, the intended kinematics x_k evolves depending on the discrete movement state s_k

$$\begin{cases} x_{k+1} = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + v_k \text{ if } s_k = \text{moving} \\ x_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_k \text{ if } s_k = \text{stopped} \end{cases} \qquad (15)$$

Here, each v_k is a zero mean Gaussian random variable with the following covariance

$$\text{cov}\,(v_k) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \qquad (16)$$

The probability of being in either discrete state given the past discrete state, $p(s_{k+1}|s_k)$, evolves according to the following equation

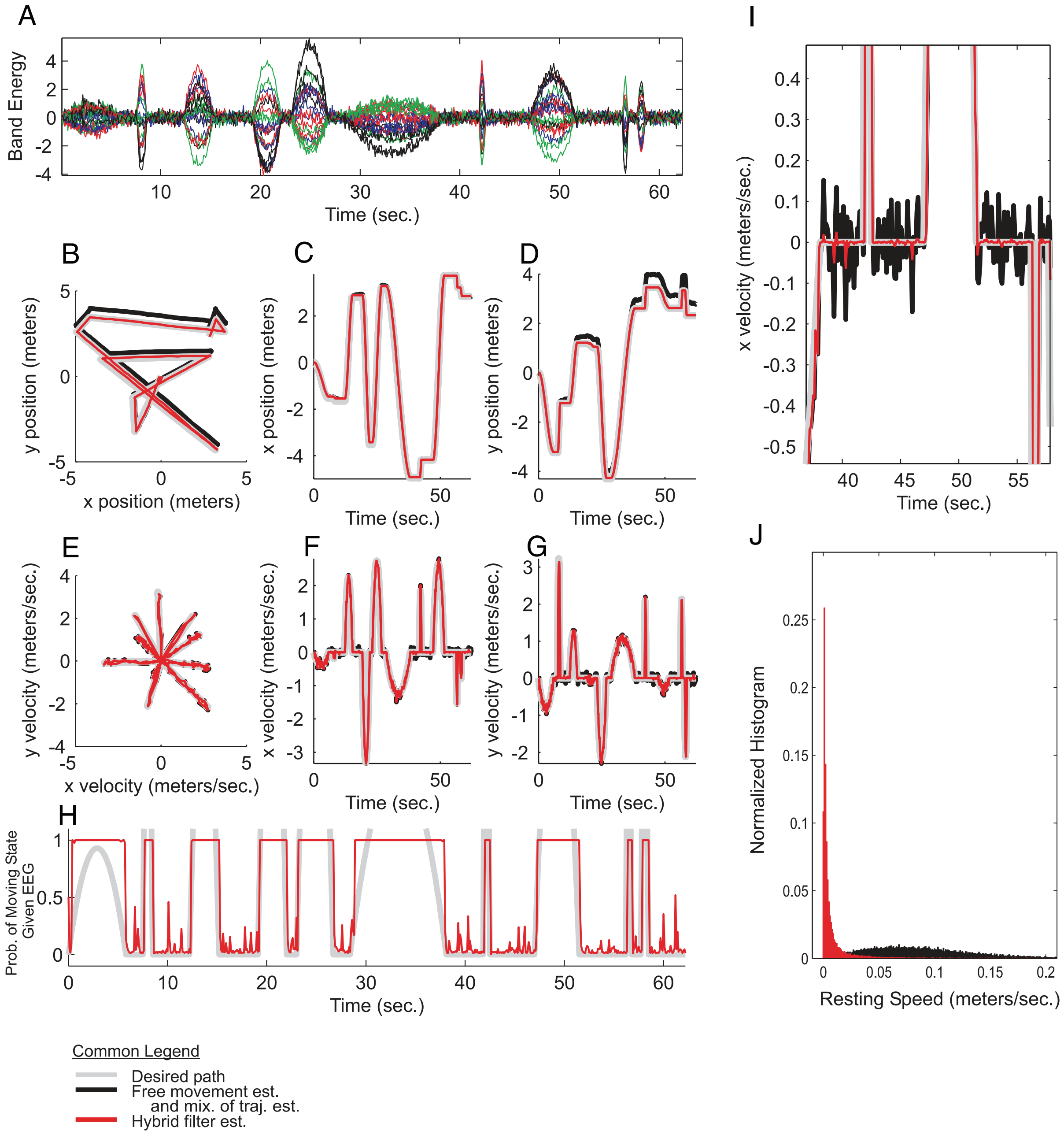

FIG. 6. Decoding performance under repeated starting and stopping in a simulated wheelchair navigation task controlled with 20 electroencephalographic (EEG)-band channels. *A*: traces of power in 20 EEG rhythm frequency bands across multiple leads under velocity control similar to the Wolpaw and McFarland (2004) observation model, simulated at 0.1-s resolution. *B–G*: intended point-to-point wheelchair movements (gray) on a 10 × 10-m workspace based on linear quadratic optimal control, plotted with decoding results from the hybrid filter (red) and mixture-of-trajectories model (black). Free-movement estimation produces virtually identical results to the mixture-of-trajectories model in this case (see explanation in text). Trajectories are plotted with *x* vs. *y* position (*B*), *x* position vs. time (*C*), *y* position vs. time (*D*), and similarly for velocity (*E–G*). *H*: wheelchair speed (gray) and posterior density of movement state under the hybrid filter (red) vs. time. *I*: expanded view of wheelchair starting/stopping from *x* velocity profile in *F* with hybrid filter (red), mixture-of-trajectories model (black), and intended *x* velocity (gray). *J*: fraction of all stop period time samples ($dt = 0.1$ s) over 500 trials vs. resting speed (meters/s) achieved by the hybrid filter (red) and mixture-of-trajectories model (black), with histogram binned at 0.001 m/s nonoverlapping intervals.

$$\begin{bmatrix} \Pr(\text{moving}) \\ \Pr(\text{stopped}) \end{bmatrix}_{k+1} = M \times \begin{bmatrix} \Pr(\text{moving}) \\ \Pr(\text{stopped}) \end{bmatrix}_{k} \qquad (17)$$

In the mixture-of-trajectories model (Fig. 1*E*), the discrete state is static, corresponding to a state transition matrix $M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. In contrast, the hybrid filter accommodates nonzero probability of switching between moving and stopped states. Here, we use $M = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$, although the exact choice of **M** is in general based on the frequency with which states are expected to alternate. The probabilities of each discrete state are initialized uniformly at 0.5 in both filters.

Filters based on these design choices were implemented as described in METHODS. Filter reconstructions were graphed against the actual trajectory in position and velocity (Fig. 6, *B–G*). The free-movement and mixture-of-trajectory reconstructions almost completely overlap with each other (black) because the posterior probability of discrete states in the mixture-of-trajectories filter approaches unit probability in the s_k = *moving* state within the first second of movement. Consequently, the free-movement and mixture-of-trajectory filters become virtually identical. For the hybrid filter, the probability of s_k = *moving* and s_k = *stopped* alternate in alignment with periods of intended movement and rest (Fig. 6*H*). The posterior density still shows oscillations during the resting periods because the EEG-band signals during zero-intended velocity are consistent with both the movement (i.e., movement with 0 velocity) and resting dynamics (*Eq. 15*).

Although very little error can be seen at the meter scale (Fig. 6, *B–G*), by magnifying the *x* velocity about four rest periods (Fig. 6*I*), we see that the free-movement and mixture-of-trajectory implementations exhibit a "resting tremor" of ~10 cm/s amplitude, whereas the hybrid filter x-velocity remains closer to 1 cm/s. A normalized histogram of reconstructed speed during the simulated rest periods (Fig. 6*J*), shows that the hybrid filter is better able to hold still compared with the free-movement and mixture of trajectory filters. Importantly, definitive stopping does not compromise the ability to start movements from rest, or track movements during s_k = *moving* periods (Fig. 6*I*).

Application 3: adaptive spike filtering under ongoing neuron death and discovery

In the current brain-driven interface prototypes, parameters of the observation model that relate neural activity to the user's intentions are learned by the algorithm during a training session consisting of real, imagined, or viewed actions (Carmena et al. 2003; Leuthardt et al. 2004; Musallam et al. 2004; Serruya et al. 2002; Shenoy et al. 2003; Taylor et al. 2002; Wahnoun et al. 2006; Wolpaw and McFarland 2004). In practice, this facilitates rapid (also termed "instant") improvements in device performance because the device learns to match the user, whereas the user can maintain their operating assumptions about the device. Moreover, these parameters may need to be adjusted based on ongoing changes in brain function such as neuron death. Finally, parameters may need to be learned anew without additional training, such as when new neurons appear on recording electrodes. This can be achieved by employing observations of neural activity from existing neurons to learn parameters of the new neurons (Eden et al. 2004b). Neuron discovery arises in the use of adjustable-depth MEMS electrodes (Muthuswamy et al. 2005) and automated electrode positioning algorithms (Cham et al. 2005).

The concept of adaptive estimation is central to all of these parameter learning problems in neural prosthetic device applications. Although neural activity parameters are commonly fit in prosthesis applications using separate intervals of training data (Carmena et al. 2003; Hochberg et al. 2006; Leuthardt et al. 2004; Musallam et al. 2004; Santhanam et al. 2006; Serruya et al. 2002; Taylor et al. 2002; Wahnoun et al. 2006; Wolpaw and McFarland 2004), these parameters can also be adjusted on the fly by simultaneously estimating the user's intentions and neural activity parameters (Eden et al. 2004a; Eden et al. 2004b). In a more modular "lock-step" implementation, the user's intentions are estimated, followed by refinements of the neural activity parameter estimates, but these actions are alternated at every time step. This recursive procedure can be applied to neural activity in training periods, as well as afterward, in the face of neuronal death and discovery.

The hybrid framework, and our implementations of the free-movement and mixture-of-trajectories filters, are all compatible with recursive adaptive estimation procedures, such as the stochastic state point process filter designed to process spiking activity (Eden et al. 2004a) that was previously demonstrated in MI decoding of arm kinematics during neuron death and discovery (Eden et al. 2004b).

In application 3, we use MI-based coordination of reaching movements (application 1) to demonstrate that the hybrid framework is still consistent with this adaptive filtering framework under the "lockstep" procedure. This simply alters the existing implementation of each filter by adding a new point process filter (APPENDIX A) (see also Eden et al. 2004a) that generates new estimates of MI tuning curve parameters at each time step. This filter uses current estimates of kinematics as if these estimates were exact [also called certainty equivalence (Bertsekas 2005)].

We now simulated a sequence of 300 intended reaching movements with target switching and repeated this simulation 50 times to construct percentiles on this performance (Fig. 7). Similarly, intended reaching movements were reconstructed with a population of 25 neurons. However, in this application, neurons were sequentially removed and replaced every 30 trials (once per minute of simulated movement time) with a new, randomly generated tuning curve unknown to the decoding algorithm. In response to death and discovery, the algorithm simply re-initializes its uncertainty (estimate covariance) of the new neuron's parameters to some large value. The initial estimates of the new neuron's parameters are assigned to the average antipreferred direction and average background firing rate of the estimated population tuning curves, which has been found to be an appropriate initialization in a previous study of adaptive filtering with MI neurons (Eden et al. 2004b). Following this initialization, the filter continues the lockstep learning process.

In total, 10 neurons were replaced over each run of 300 simulated movements. These conditions represent, in some ways, a more difficult scenario than encountered in practice: death and discovery are not expected to occur simultaneously, and this rate of death and discovery (1/min) far exceeds the anticipated rate. However, the tuning curve parameters were initialized to their exact values at the beginning of the simulation. This allowed us to dissociate performance degradation from initial parameter uncertainty versus ongoing neuronal death and discovery. In practice, these initial parameter estimates can be obtained (independent of the filter implementation) from training data to arbitrary precision (by using larger data sets) using standard parameter optimization procedures such as Matlab's glmfit routine if the model class is chosen to result in a convex parameter optimization (Truccolo et al. 2005). This is the case with the MI and EEG-band neural activity models described in applications 1 and 2, respectively.

Lockstep adaptive versions of all three filters were able to learn new neuron parameters and produce estimates of intended reaching kinematics throughout the simulation (Fig. 7). Estimated parameter values approximate actual values for the first replaced MI neuron in one 300-run sequence (Fig. 7,

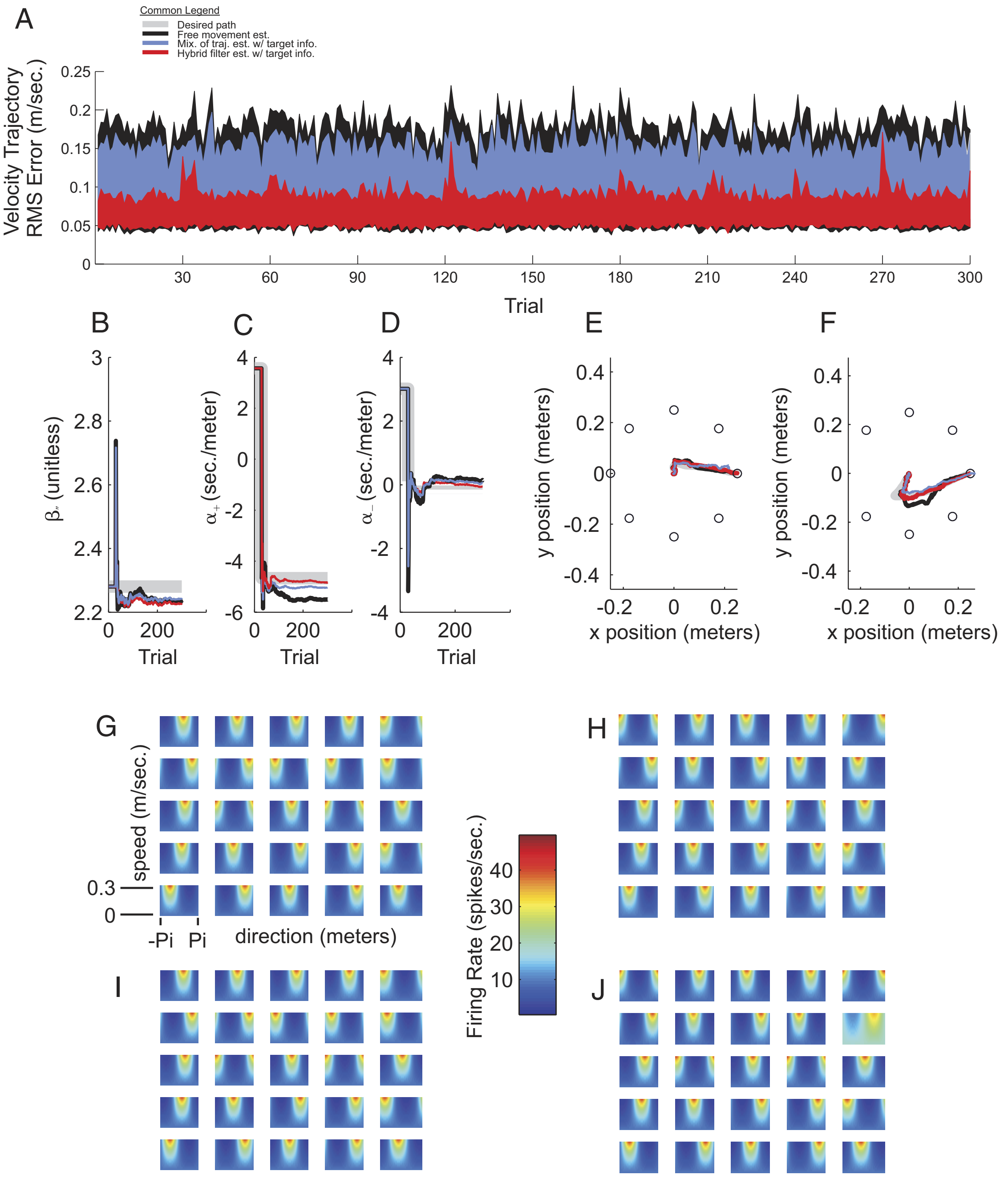

FIG. 7. Decoding performance under neuron death and discovery for center-out reaching with switching targets using 25 simulated neurons to compare adaptive versions of estimation with free-movement (purple), mixture-of-trajectories (black), and hybrid (red) models. *A*: RMS error in reconstructed velocity trajectory against trial number for a regimen of 300 simulated reaching movements. The shaded regions are bounded by 5th and 95th percentiles, from 50 repetitions of this regimen. *B–D*: adaptive parameter estimation of a 3-parameter cosine-velocity-tuned primary motor cortical cell. Each panel depicts a different parameter vs. trial number, including the actual parameter value (gray) and estimates from the adaptive filters. *E* and *F*: *x-y* trajectories of the actual movement and estimates from the 3 filter types (same color scheme as *B–D*), during trial 1 (*E*) and trial 300 (*F*). *G–J*: cosine-velocity tuning curves for each simulated neuron, during trial 1 (*G* and *I*) and trial 300 (*H* and *J*), including actual tuning curves (*G* and *H*) and estimates generated by the adaptive hybrid filter (*I* and *J*).

B–D), and the lockstep adaptive hybrid filter appears to converge faster and more reliably for α_1 (Fig. 7*C*) and α_2 (Fig. 7*D*) parameters in this example. Actual tuning curves over the entire population are shown for one run sequence at trial 1 (Fig. 7*G*) and trial 300 (Fig. 7*H*), where the first 10 neurons have been replaced with different, randomly drawn tuning curves. At trial 1, the tuning curve parameter estimates generated by the lockstep adaptive hybrid filter are initialized to equal the true simulated parameters (Fig. 7*I*). These tuning curve estimates still approximate the actual tuning curve at trial 300 (Fig. 7*J*), despite ongoing death and discovery of neurons in the recorded ensemble. The estimated tuning curve for neuron 10 at trial 300 (Fig. 7*J*) looks washed out. It has just been replaced on this trial, and its parameter estimates have been initialized to correspond to the anti-preferred direction of the estimated ensemble tuning curves, at the average background firing rate.

Sample trajectory decodes from the first (Fig. 7*E*) and last (*F*) trials of a 300-run sequence show qualitatively similar performance. However, the 90% confidence intervals in velocity trajectory RMS error (Fig. 7*A,* constructed with 5th and 95th percentiles over the 50 trials) were substantially improved for the hybrid filter versus the other two methods, on the order of 10 cm/s reduction in the 95th percentile. Moreover, the mixture-of-trajectories approach fared slightly worse in this measure than the free-movement approach, perhaps because of its pull toward passing targets (Fig. 7*F*) and as discussed under application 1 (Figs. 2 and 3). The filtering methods, especially the hybrid filter, show transient spikes in the RMS error at every 30 trials, corresponding to the time point at which a neuron has been replaced. These transient effects could potentially be removed by incorporating parameter uncertainty into the trajectory estimate, a consideration that was ignored by using certainty equivalence in the lockstep filtering procedure. Together, these results emphasize that adaptive estimation is compatible with all three filter types, and that hybrid filtering can facilitate more accurate on-line parameter estimation.

Application 4: filtering under model mis-specification due to physiologically realistic history dependence in MI ensemble spiking

Inevitably, the mathematical models we use to describe the relationship between neural activity and the underlying intention will be incorrect [as from George Box (1979), "all models are wrong, but some are useful"]. Although application 3 showed that adaptive estimation could compensate unknown parameter assignments, we now quantify performance degradation in the face of model mis-specification, again in the context of MI spiking activity and reaching movements to switching targets.

Experimentally recorded spiking activity from intact primates can be filtered in an off-line fashion to examine filtering under model mis-specification in an attempt to predict closed-loop performance in a human patient population with representative movement disorders. In this application, we instead simulate spikes from empirical statistical models that have been fit to experimentally recorded data from intact primates. These history-dependent point process generalized linear models have recently applied to capture unprecedented realism in MI spike timing based on the Akaike Information Criterion, a model quality index that balances goodness-of-fit against model complexity to avoid overfitting (Truccolo et al. 2005). As a result, the spiking activity generated by these models is qualitatively and quantitatively similar to the experimentally recorded primate MI neurons from which they were derived. The advantage of applying empirical statistical models is that we can systematically evaluate the effect of including or removing structure in MI spiking (such as history dependence) on the performance degradation of competing algorithms that are not perfectly matched to this structure.

In this application, the model form used to generate spiking activity is different from the model form used to decode these spikes. Specifically, MI spiking activity is now simulated with physiologically realistic history dependence, in addition to modulation based on arm velocity. The strength of this history dependence is described in Fig. 8*A,* where the parameter values have been approximately extracted from Fig. 5*B* of Truccolo et al. (2005), which describes MI spiking activity recorded from a monkey performing a visuomotor pursuit-tracking task with a manipulandum that controls an on-screen cursor.

Simulated traces of kinematics (Fig. 8*B, 1* and *2* from the top) and neural activity (Fig. 8*B, 3–6* from the top) show that spiking activity that results from the same movement is qualitatively different under the physiologically realistic history-dependent model (Truccolo et al. 2005) (Fig. 8*B,* 5 and 6 from the top) versus the velocity modulated model (*Eq. 9*) used in applications 1 and 3 (Fig. 8*B,* 3 and 4 from the top). Note that the conditional intensity, or instantaneous firing rate, of the history-dependent model, well exceeds 200 spike/s, but these deflections are momentary, so that they generate physiologically observed bursts rather than producing an unrealistic, sustained elevation in mean firing rate.

How well do each of the three filters perform when they assume the basic velocity modulated model (*Eq. 9*) while they decode spiking activity that was actually generated with physiologically realistic history dependence (Truccolo et al. 2005)? We simulated 1,000 intended reaching movements to switching targets and reconstructed the intended movements with the three filter types to generate empirical probability densities on four performance metrics: position trajectory RMS error (Fig. 8*C*), position endpoint RMS error (Fig. 8*D*), velocity trajectory RMS error (Fig. 8*D*), and velocity endpoint RMS error (Fig. 8*E*). Each of these figures (8, *C* and *D*) includes panels for (from top to bottom) free-movement, mixture-of-trajectories, and hybrid filters. Each panel displays the distribution of performance when spikes are simulated with physiologically realistic history dependence (red) and with the original velocity modulated model (black). Position and velocity trajectory RMS errors degrade the least for the hybrid filter, followed by the mixture-of-trajectories, and free-movement filters. Both hybrid and mixture-of-trajectories filters maintain low endpoint RMS errors, while the free-movement filter shows relatively strong degradation in this performance measure.

Why does the hybrid filter show the least degradation under model mis-specification, followed by the mixture-of-trajectories and free-movement filters? The model mis-specification tends to push trajectory estimates away from their true values, whereas priors on the movement (embodied by the trajectory model) help restore estimates closer to their true values. Because the hybrid filter includes a switching discrete state, it is better able to describe the typical set of movements than the mixture-of-trajectories filter. Similarly, both the hybrid filter and mixture-of-trajectories filter constrain their priors on trajectory endpoint based on a known set of possible target locations. This not only improves endpoint RMS error but mitigates degradation under model mismatch that otherwise significantly affects the free-movement filter.

DISCUSSION

We have introduced a unified approach for the design of filters for prosthetic devices. By using this technique, we can map spikes and continuous field potentials to estimates of the user's intention for a wide array of neural prosthetic device applications. The technique draws on Bayesian filter theory to generalize the dominant approaches to filter design in neural prosthetic devices (Brockwell et al. 2004; Carmena et al. 2003; Cowan and Taylor 2005; Eden et al. 2004b; Hochberg et al. 2006; Kemere and Meng 2005; Kemere et al. 2003; Musallam

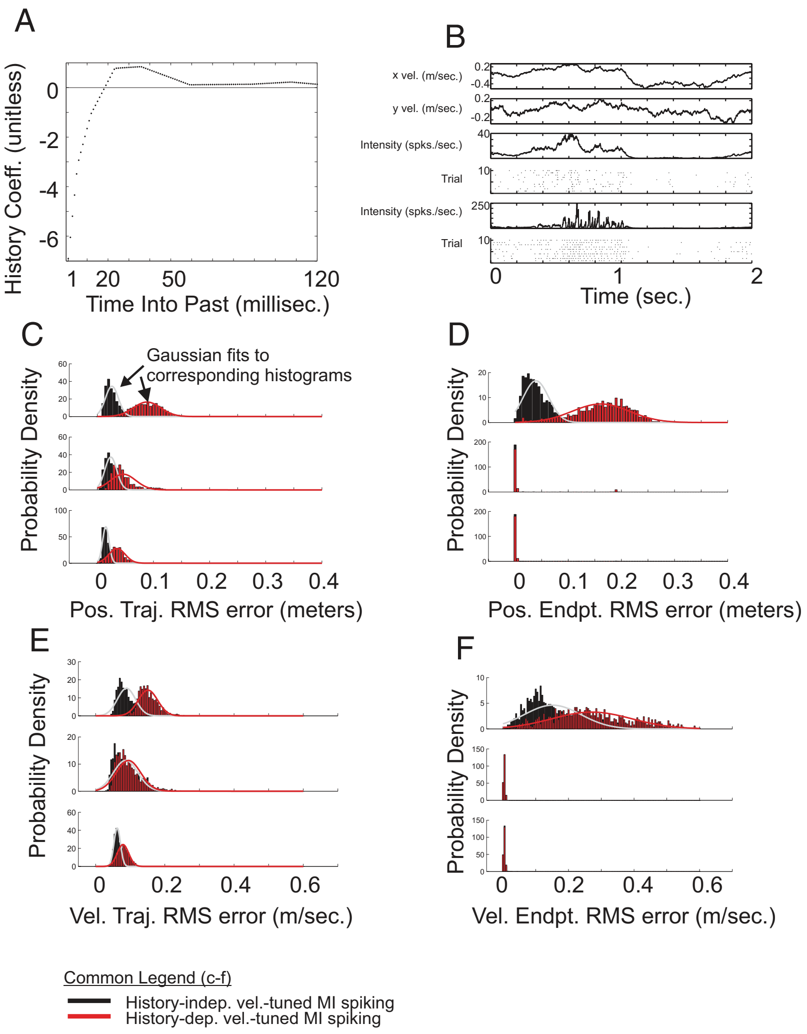

FIG. 8. Decoding performance under model-misspecification for center-out reaching with switching targets using 25 simulated neurons over 1,000 simulated reaching movements. *A*: physiologically realistic history dependence for the average simulated primary motor cortical cell, based on Truccolo et al. (2005). *B*: simulated traces vs. time (s) from 1 trial, including (subpanels, from top to bottom) *x* velocity (m/s) (*1*); *y* velocity (*2;* m/s); spiking intensity (*3;* spike/s) from 1 neuron without history dependence, 10 sample rasters of spike times from that 1 neuron across multiple trials simulated with that same spiking intensity (*4*), spiking intensity (spike/s) on 1 trial from that same neuron, now including history dependence (*5;* with history dependence, spiking intensity will vary from trial to trial), and 10 corresponding rasters of spike times from that history-dependent spiking intensity (*6*). *C–F*: empirical probability density functions vs. various RMS errors using spikes simulated from standard cosine-velocity tuning curve (black) or physiologically realistic history dependence (red) where the decoding filter assumes only the standard cosine-velocity tuning curve. RMS errors include position trajectory (*C*), position endpoint (*D*), velocity trajectory (*E*), and velocity endpoint (*F*). Each panel includes 3 subpanels, corresponding to estimation with (from top to bottom): free-movement (*1*), mixture-of-trajectories (*2*), and the hybrid filter (*3*).

et al. 2004; Santhanam et al. 2006; Serruya et al. 2002; Shenoy et al. 2003; Srinivasan and Brown 2007; Srinivasan et al. 2006a; Taylor et al. 2002; Wessberg et al. 2000; Wolpaw and McFarland 2004; Wu et al. 2004, 2006; Yu et al. 2005, 2007).

The hybrid framework and filtering technique captures these previous methods completely but also allows for more flexibility. For this reason, device performance with this technique is expected to be equal or better than these previous methods. The versatility of this technique is illustrated in four applications using simulated ensemble spiking activity or multi-channel EEG: reaching to discrete targets that switch during movement, wheelchair navigation with definitive stopping, adaptive spike filtering under ongoing neuron death and discovery, and filtering under model mis-specification.

For both the hybrid point process filter and the IMM Switching Kalman Filter for Gaussian observation models, the number of operations at time step k scales with $|S_k|$, the number of values that a discrete state variable can take on. This is because the posterior density on the discrete state is nonparametric, and the posterior density on the continuous state is represented as a mixture of $|S_k|$ Gaussians. The particle filter (Brockwell et al. 2004; Doucet et al. 2001; Ergun et al. 2007), a Monte Carlo approach, would increase the fidelity the posterior density at the expense of increased computational cost. Ultimately, the way in which the posterior density is represented will depend on the cost of computation versus device performance in any specific application.

As shown in the previous section, the hybrid framework accommodates multiple discrete random processes by condensing them into one. Unfortunately, n discrete random variables, each with p possible values at step k, results in a condensed random variable with $|S| = n^p$. Fortunately, filtering on the hybrid framework can be parallelized fairly directly. This means that even with large $|S_k|$, the device can be controlled in real time if the hardware supports parallel computations. Parallelizing a digital hardware implementation may not necessarily save energy but could allow a slower clock speed. Parallelized algorithms do not necessitate the development of new types of processors [although power considerations are motivating new hardware architecture (Sarpeshkar et al. 2005)]; multithreading emulates parallelization on a single CPU, and even the prototyping language Matlab can now be parallelized with standard desktop computers (Kepner 2007). Formal analysis of hardware implementation is a key topic for subsequent investigation.

This real-time implementation is also possible because our algorithms are the same order of complexity as hybrid filtering based on the extended Kalman filter, which has been used extensively in mission-critical real-time military applications (Bar-Shalom et al. 2001). The algorithms are recursive rather than batch-processed, meaning that estimates of the user's intention can be updated using the latest neural observations without re-analyzing previously recorded activity.

In many applications, however, the number of discrete states can be kept small by using context. Context means that the space of device states is restricted at any given step in a way that still allows the user to eventually reach the desired device state. Consider how you organize files on your computer. By arranging your files in a sequence of subdirectories, you make it easy to scan through the list of files at each step. By placing all your files on the desktop, you are forced to select your file from a very large list, even though the file is just one mouse click away.

Looking forward, we expect to draw extensively on the rich field of dynamic Bayesian networks to address future applications. Prototyping is needed to determine the best computation/accuracy tradeoff for specific prosthetic devices. Learning and real-time sensory feedback (visual, somatosensory, auditory) must also be considered in developing algorithms that define the prosthetic interface. Associated technologies like computer vision and robotic control can be integrated with the hybrid framework to enhance real-world performance measures.

Ultimately, these approaches must be compared in a longitudinal study of device performance in activities of daily living for a population of individuals that experience well-defined stages and pathological mechanisms of a target motor deficit. Off- and on-line decoding studies in the laboratory will be important intermediate steps between simulated algorithm performance and end-stage clinical evaluation.

Finally, estimation with a minimum average cost criterion is not the only approach to formally describing the prosthetics problem. Future work will explore stochastic control, hierarchical design architectures, robustness, power consumption, hardware architecture, monetary cost, and other themes in systems design to achieve the type of performance in practical tasks that is necessary to benefit the full spectrum of limited motor function, from locked-in syndrome to single-arm amputation.

APPENDIX

Appendix A: Approximate point process filter for Gauss-Markov process (discrete time)

The Gaussian approximation to the posterior density with a Taylor series expansion about the prediction mean is employed in the following filter equations (Eden et al. 2004a). Consider a Gauss-Markov trajectory model

$$p(x_{k+1}|x_k) \sim \quad N(F_k x_k \; + \; b_k, Q_k) \tag{A1}$$

A point process observation model is specified for an ensemble of C neurons. The conditional intensity function of the cth neuron, denoted λ_k^c, may depend on H_k and x_k. For the kth time step and cth neuron, n_k^c spikes arrive in a δ_k time interval.

The prediction density mean $x_{k+1|k}$ and covariance $W_{k+1|k}$ are

$$x_{k+1|k} = F_k x_{k|k} + b_k \tag{A2}$$

$$W_{k+1|k} = F_k W_{k|k} F_k' + Q_k \tag{A3}$$

The posterior density covariance $W_{k+1|k+1}$ and mean $x_{k+1|k+1}$ are

$$\begin{aligned}(W_{k+1|k+1})^{-1} &= (W_{k+1|k})^{-1} \\ &+ \sum_{c=1}^{C} \left[\left(\frac{\partial \log \lambda_k^c}{\partial x_k} \right)' [\lambda_k^c \delta_k] \left(\frac{\partial \log \lambda_k^c}{\partial x_k} \right) \; - \; \left(n_k^c - \lambda_k^c \delta_k \right) \frac{\partial^2 \log \lambda_k^c}{\partial x_k \partial x_k'} \right]_{x_{k+1|k}}\end{aligned} \tag{A4}$$

$$x_{k+1|k+1} = x_{k+1|k} + W_{k+1|k+1} \sum_{c=1}^{C} \left[\left(\frac{\partial \log \lambda_k^c}{\partial x_k} \right)' (n_k^c - \lambda_k^c \delta_k) \right]_{x_{k+1|k}} \tag{A5}$$

Consider instead, an array of c neural signals $n_k^{1:C} = [\, n_k^1, n_k^2, \ldots, n_k^C]'$ described by a Gaussian observation model (such as EEG) with mean $D_k x_k \; + \; f_k(\mathrm{s}_k, \; H_k)$ and variance R_k. Here x_k is a $J \times 1$ vector of continuous states, D_k is a $C \times J$ matrix that may depend on s_k, and $f_k(s_k, H_k)$ is a function that maps neural history to a $C \times 1$ vector, such as with ARMA models. The posterior density covariance and mean are then given by the standard Kalman filter equations (Kailath et al. 2000)

$$W_{k+1|k+1} = W_{k+1|k} - W_{k-1|k} D_{k+1}' (D_{k+1} W_{k+1|k} D_{k+1}' + R_{k+1})^{-1} D_{k+1} W_{k+1|k} \tag{A6}$$

$$\begin{aligned} x_{k+1|k+1} \; &= \; x_{k+1|k} + W_{k+1|k} D_{k+1}' (D_{k+1} W_{k+1|k} D_{k+1}' \\ &\qquad + R_{k+1})^{-1} (n_{k+1}^{1:C} - D_{k+1} x_{k+1|k} - f_{k+1}(s_{k+1}, H_{k+1}))\end{aligned} \tag{A7}$$

The probability density in step 5 of the point process hybrid filter (see preceding text) is then replaced by

$$\begin{aligned} n_{k+1}^{1:C} | s_{k+1}, H_{k+1} \; &\sim N(D_{k+1} x_{k+1|k} \; + \; f_{k+1}(s_{\,k+1}, H_{k+1}), \\ &\qquad D_{k+1} W_{k+1|k} D_{k+1}' + R_{k+1}) \end{aligned} \tag{A8}$$

to correspond to the Interacting Multiple Model (IMM) approach to switching Kalman filters (Bar-Shalom et al. 2001).

Appendix B: Gaussian approximation to mixture of Gaussians

Consider a distribution composed of the weighted average of R multidimensional Gaussians

$$p(x) = \sum_{i=1}^{R} d_i N(x; \, \mu_i, \, W_i) \tag{B1}$$

with weights d_i, and where $N(x; \, \mu_i, \, W_i)$ denotes the Gaussian probability density function with mean μ_i and covariance W_i. The following standard approximation (Bar-Shalom et al. 2001) is obtained by moment matching [calculating the mean and covariance of $p(x)$]

$$p(x) \approx N(x; \, m, \, K) \tag{B2}$$

where

$$m = \sum_{i=1}^{R} d_i \mu_i \tag{B3}$$

$$K = \sum_{i=1}^{R} d_i \times [W_i + (\mu_i - m)(\mu_i - m)'] \tag{B4}$$

Appendix C: Derivation of a point process hybrid filter to map spikes to hybrid prosthetic device states

For the kth discrete time step, define the user-intended continuous state x_k, discrete state s_k, and the ensemble spiking activity of all C neurons $n_k^{1:C}$. The history of ensemble spiking at time step k is given by $H_k = (n_1^{1:C}, \, n_2^{1:C}, \ldots, \, n_{k-1}^{1:C})$. Define the observation model $p(n_{k+1}^{1:C}|x_{k+1}, s_{k+1}, H_{k+1})$ that represents the relationship between user intentions and spiking activity. Define the trajectory model $p(x_{k+1}|x_k, \, s_{k+1})$ and discrete state transition density $p(x_{k+1}|x_k, \, s_{k+1})$ that reflect the distribution of intentions that the user is expected to request over time. In this section, we seek a recursive method to obtain $p(x_{k+1}, s_{k+1}|n_{k+1}^{1:C}, H_{k+1})$ from $p(x_k, s_k|n_k^{1:C}, H_k)$ and $n_{k+1}^{1:C}$. This constitutes the point process hybrid filtering procedure.For our specific hybrid state space in Fig. 1*A*,

$$p(x_{k+1}, s_{k+1}|n_{k+1}^{1:C}, H_{k+1}) = p(x_{k+1}|s_{k+1}, n_{k+1}^{1:C}, H_{k+1})p(s_{k+1}|n_{k+1}^{1:C}, H_{k+1}) \quad (C1)$$

This implies that our problem is equivalent to obtaining $p(x_{k+1}|s_{k+1}, n_{k+1}^{1:C}, H_{k+1})$ and $p(s_{k+1}|n_{k+1}^{1:C}, H_{k+1})$ from $p(x_k|s_k, n_k^{1:C}, H_k)$, $p(s_k|n_k^{1:C}, H_k)$, and $n_{k+1}^{1:C}$.Note that

$$p(x_{k+1}|n_{k+1}^{1:C}, H_{k+1}) = \sum_{s_{k+1}} p(x_{k+1}|s_{k+1}, n_{k+1}^{1:C}, H_{k+1})p(s_{k+1}|n_{k+1}^{1:C}, H_{k+1}) \quad (C2)$$

We now calculate $p(x_{k+1}|s_{k+1}n_{k+1}^{1:C}, H_{k+1})$ using *Eqs. C3–C8* and calculate $p(s_{k+1}|n_{k+1}^{1:C}, H_k)$ using *Eq. C9*. Observe that

$$p(x_{k+1}|s_{k+1}, n_{k+1}^{1:C}, H_{k+1}) = \frac{p(n_{k+1}^{1:C}|x_{k+1}, s_{k+1}, H_{k+1})p(x_{k+1}|s_{k+1}, H_{k+1})}{p(n_{k+1}^{1:C}|s_{k+1}, H_{k+1})} \quad (C3)$$

where $p(x_{k+1}|s_{k+1}, H_{k+1})$ is the prediction density given by the Chapman-Kolmogorov equation

$$p(x_{k+1}|s_{k+1}, H_{k+1}) = \int_{x_k} p(x_{k+1}|x_k, s_{k+1}, H_{k+1})p(x_k|s_{k+1}, H_{k+1})\mathrm{d}x_k \quad (C4)$$

Equations C3 and *C4* comprise one step of a filter on $p(x_k|s_{k+1}, H_{k+1})$ with the observation model $p(n_{k+1}^{1:C}|x_{k+1}, s_{k+1}, H_{k+1})$ and trajectory model $p(x_{k+1}|x_k, s_{k+1}, H_{k+1}) = p(x_{k+1}|x_k, s_{k+1})$. For computational simplicity, we approximate both the trajectory model and posterior density $p(x_{k+1}|s_{k+1}, n_{k+1}^{1:C}, H_{k+1})$ to be Gaussian. Such a filter (reproduced under APPENDIX A) is developed in (Eden et al. 2004a) for point processes using a Taylor expansion about the prediction density mean rather than the posterior density mean employed in (Brown et al. 1998). The density $p(x_k|s_{k+1}, H_{k+1})$ is obtained by

$$p(x_k|s_{k+1}, H_{k+1}) = \sum_{s_k} p(s_k|s_{k+1}, H_{k+1})p(x_k|s_k, s_{k+1}, H_{k+1}) \quad (C5)$$

This density is a mixture of Gaussians that is approximated by one Gaussian density using a standard moment-matching formula given in APPENDIX B. The first density in the summation (*Eq. C5*) is calculated as follows

$$p(s_k|s_{k+1}, H_{k+1}) = \frac{p(s_{k+1}|s_k, H_{k+1})p(s_k|H_{k+1})}{p(s_{k+1}|H_{k+1})} \quad (C6)$$

where

$$p(s_{k+1}|H_{k+1}) = \sum_{s_k} p(s_{k+1}|s_k)p(s_k|H_{k+1}) \quad (C7)$$

Here $p(s_{k+1}|s_k)$ is the discrete state transition density, and $p(s_k|H_{k+1})$ is the posterior density on the discrete state, given in the previous iteration. The second density in the summation (*Eq. C5*) is given by a quantity retained from the previous step

$$p(x_k|s_k, s_{k+1}, H_{k+1}) = p(x_k|s_k, H_{k+1}) \quad (C8)$$

This statement is verified in APPENDIX D. We now calculate $p(s_{k+1}|n_{k+1}^{1:C}, H_{k+1})$ in *Eq. C2* using the following relation

$$p(s_{k+1}|n_{k+1}^{1:C}, H_{k+1}) = \frac{p(n_{k+1}^{1:C}|s_{k+1}, H_{k+1})p(s_{k+1}|H_{k+1})}{p(n_{k+1}^{1:C}|H_{k+1})} \quad (C9)$$

Equation C7 calculates $p(s_{k+1}|H_{k+1})$. The density $p(n_{k+1}^{1:C}|s_{k+1}, H_{k+1})$ is given by the following integral.

$$p(n_{k+1}^{1:C}|s_{k+1}, H_{k+1}) = \int_{x_{k+1}} p(n_{k+1}^{1:C}|s_{k+1}, x_{k+1}, H_{k+1})p(x_{k+1}|s_{k+1}, H_{k+1})\mathrm{d}x_{k+1} \quad (C10)$$

An approximation to this integral for point process observations is given by Laplace approximation as detailed in APPENDIX E. Finally, $p(n_{k+1}^{1:C}|H_{k+1})$ is a normalizing factor obtained by summing the numerator over all possible values of s_{k+1}.

Appendix D: Corollary

Verify *Eq. C8* that $p(x_k|s_k, s_{k+1}, H_{k+1}) = p(x_k|s_k, H_{k+1})$

$$\begin{aligned} p(x_k|s_k, s_{k+1}, H_{k+1}) &= \frac{p(x_k, s_{k+1}|s_k, H_{k+1})}{p(s_{k+1}|s_k, H_{k+1})} \\ &= \frac{p(s_{k+1}|x_k, s_k, H_{k+1})p(x_k|s_k, H_{k+1})}{p(s_{k+1}|s_k, H_{k+1})} \end{aligned} \quad (D1)$$

From Fig. 1A, observe that

$$p(s_{k+1}|x_k, s_k, H_{k+1}) = p(s_{k+1}|s_k, H_{k+1}) \quad (D2)$$

Thus *Eqs. D1* and *D2* imply that

$$p(x_k|\mathrm{s}_k, \mathrm{s}_{k+1}, H_{k+1}) = p(x_k|s_k, H_{k+1}) \quad (D3)$$

Appendix E: Laplace approximation of $p(n_{k+1}^{1:C}|s_{k+1}, H_{k+1})$

This section derives the Laplace approximation of *Eq. C10*, repeated in the following text for convenience

$$p(n_{k+1}^{1:C}|s_{k+1}, H_{k+1}) = \int_{x_{k+1}} p(n_{k+1}^{1:C}|s_{k+1}, x_{k+1}, H_{k+1})p(x_{k+1}|s_{k+1}, H_{k+1})dx_{k+1} \quad (E1)$$

Define

$$h(x_{k+1}, n_{k+1}^{1:C}) = \log\,[p(n_{k+1}^{1:C}|s_{k+1}, x_{k+1}, H_{k+1})p(x_{k+1}|s_{k+1}, H_{k+1})] \quad (E2)$$

The Laplace approximation to *Eq. E1* is given by

$$\begin{aligned} p(n_{k+1}^{1:C}|s_{k+1}, H_k) &\approx (2\pi)^{m/2}|-\nabla_{x_{k+1}}^2 h(x_{k+1}, n_{k+1}^{1:C})|^{-1/2} \\ &\times p(n_{k+1}^{1:C}, x_{k+1}|s_{k+1}, H_{k+1})\Big|_{x_{k+1}=x_0} = (2\pi)^{m/2}|-\nabla_{x_{k+1}}^2 h(x_{k+1}, n_{k+1}^{1:C})|^{-1/2} \\ &\times p(n_{k+1}^{1:C}|x_{k+1}, s_{k+1}, H_{k+1})\mathrm{p}(x_{k+1}|s_{k+1}, H_{k+1})\Big|_{x_{k+1}=x_0} \end{aligned} \quad (E3)$$

where the mode x_0 maximizes $p(n_{k+1}^{1:C}|s_{k+1}, x_{k+1}, H_{k+1})$ $p(x_{k+1}|s_{k+1}, H_{k+1})$ for a given $n_{k+1}^{1:C}$.

Approximate the mode as in (Eden et al. 2004a) using a prediction density, in this case given by

$$x_0 \approx x_{k+1|k,\, s_{k+1}} \quad (E4)$$

Under this approximation, the following equalities hold

$$-\nabla_{x_{k+1}}^2 h(x_{k+1}, n_{k+1}^{1:C})\Big|_{\mathrm{x}_{k+1}=x_{k+1|k,\mathrm{s}_{k+1}}} = W_{k+1|k+1,\, s_{k+1}}^{-1} \quad (E5)$$

$$p(x_{k+1}\Big| s_{k+1}, H_{k+1})|_{x_{k+1=x_{k+1|k,s_{k+1}}},} = \frac{1}{(2\pi)^{m/2}|W_{k+1|k,\, s_{k+1}}|^{1/2}} \quad (E6)$$

where $W_{k+1|k+1,\, s_{k+1}}$ is precisely the variance of the Gaussian approximation to the posterior density given in (Eden et al. 2004a) and APPENDIX A.

Using *Eqs. E5* and *E6*, the Laplace approximation (*Eq. E3*) simplifies to

$$p(n_{k+1}^{1:C}|s_{k+1}, H_{k+1}) \approx p(n_{k+1}^{1:C}|x_{k+1}, s_{k+1}, H_{k+1})|_{x_{k+1}=x_{k+1}|k,s_{k+1}} \quad (E7)$$

Express $p(n_{k+1}^{1:C}|x_{k+1}, s_{k+1}, H_{k+1})$ using a discrete-time approximation for point processes (Eden et al. 2004a)

$$p(n^{1:C}_{k+1}|x_{k+1}, s_{k+1}, H_{k+1}) \propto \prod_{c=1}^{C} \exp(n^{c}_{k+1}\log(\lambda^{c}_{k+1}\delta_{k+1}) - \lambda^{c}_{k+1}\delta_{k+1}) \qquad (E8)$$

where λ^c_{k+1} denotes the conditional intensity of neuron c at time step $k + 1$, which may be a function of s_{k+1}, x_{k+1}, and H_{k+1}.

Substituting this approximation into *Eq. E7,* we have the final approximate equation for $p(n^{1:C}_{k+1}|s_{k+1}, H_{k+1})$

$$p(n^{1:C}_{k+1}|s_{k+1}, H_{k+1}) \approx \frac{|W_{k+1|k+1,s_{k+1}}|^{1/2}}{|W_{k+1|k,s_{k+1}}|^{1/2}} \times \prod_{c=1}^{C} \exp(n^{c}_{k+1}\log(\lambda^{c}_{k+1}\delta_{k+1}) - \lambda^{c}_{k+1}\delta_{k+1})\Bigg|_{x_{k+1}=x_{k+1|k,s_{k+1}}} \qquad (E9)$$

Appendix F: Spike filtering with the hybrid framework: practical note on numerical issues

This section documents four points to consider when implementing the hybrid filter.

First, the spike filtering (hybrid point process) filter described in this paper uses a bank of stochastic state point process filters (SSPF), described in Eden et al. (2004a) and APPENDIX A. As with the SSPF, the prediction or posterior covariance may become singular because of numerical implementation or badly conditioned if the values in certain matrix elements are dramatically smaller than others. In a practical implementation, it is useful to check that a covariance matrix is well-conditioned or invertible before taking the inverse operation required by the SSPF. If the posterior covariance is not invertible, perform a Fisher's scoring step instead of executing the posterior covariance equation, by removing the $-(n^c_k - \lambda^c_{k,s_{k+1}}\Delta\delta_k)\frac{\partial^2 \log \lambda^c_{k,s_{k+1}}}{\partial x_k}$ term of the posterior covariance equation for just that time step. If the prediction covariance is badly conditioned, retain the prediction covariance as the posterior covariance.

Second, you may encounter divide-by-zero or floating-point errors if you incorrectly implement the nine step spike filtering procedure. Check that you are not dividing by a discrete state probability that has approached zero.

Third, to generate smoother continuous state trajectories, such as in application 1 of RESULTS, augment your state space to include acceleration terms, and introduce the nonzero diagonal term of increment covariance only in the acceleration dimensions.

Fourth, note that application 1 of RESULTS is a discrete-target version of problem of reaching to drifting targets that evolve over a continuum of positions. The discrete nature of the targets in application 1 necessitates the hybrid framework. Similarly, look for parallels between your application and discrete or continuous versions of it.

GRANTS

This research was supported by the National Institutes of Health Medical Scientist Training Program Ruth Kirschstein National Research Service Award T32 GM-07753-28 to L. Srinivasan and Grant R01 DA-015644 to E. N. Brown and National Science Foundation Grant CCR-0325774 to S. K. Mitter.

REFERENCES

Andersen RA, Buneo CA. Intentional maps in posterior parietal cortex. *Annu Rev Neurosci* 25: 189–220, 2002.

Andersen RA, Burdick JW, Musallam S, Pesaran B, Cham JG. Cognitive neural prosthetics. *Trends Cogn Sci* 8: 486–493, 2004.

Barbieri R, Wilson MA, Frank LM, Brown EN. An analysis of hippocampal spatio-temporal representations using a Bayesian algorithm for neural spike train decoding. *IEEE Trans Neural Syst Rehabil Eng* 13: 131–136, 2005.

Bar-Shalom Y, Li X-R, Kirubarajan T. *Estimation With Applications to Tracking and Navigation : Theory, Algorithms and Software.* New York: Wiley, 2001.

Bertsekas DP. *Dynamic Programming and Optimal Control.* Belmont, MA: Athena Scientific, 2005.

Box GEP. *Robustness in the Strategy of Scientific Model Building,* edited by Launer RL, Wilkinson GN. New York: Academic, 1979, p. 201–236.

Brockwell AE, Rojas AL, Kass RE. Recursive bayesian decoding of motor cortical signals by particle filtering. *J Neurophysiol* 91: 1899–1907, 2004.

Brown EN. Theory of point processes for neural systems. In: *Methods and Models in Neurophysics*, edited by Chow CC, Gutkin B, Hansel D, Meunier C, Dalibard J. Paris: Elsevier, 2005, p. 691–726.

Brown EN, Barbieri R, Ventura V, Kass RE, Frank LM. The time-rescaling theorem and its application to neural spike train data analysis. *Neural Comput* 14: 325–346, 2002.

Brown EN, Frank LM, Tang D, Quirk MC, Wilson MA. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J Neurosci* 18: 7411–7425, 1998.

Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, Dimitrov DF, Patil PG, Henriquez CS, Nicolelis MA. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol* 1: E42, 2003.

Cham JG, Branchaud EA, Nenadic Z, Greger B, Andersen RA, Burdick JW. Semi-chronic motorized microdrive and control algorithm for autonomously isolating and maintaining optimal extracellular action potentials. *J Neurophysiol* 93: 570–579, 2005.

Cowan TM, Taylor DM. Predicting reach goal in a continuous workspace for command of a brain-controlled upper-limb neuroprosthesis. *Proc 2nd Internat IEEE EMBS Conf on Neural Engineering* Arlington, VA, p. 74, 2005.

Daley DJ, Vere-Jones D. *An Introduction to the Theory of Point Processes.* New York: Springer, 2003.

Doucet A, de Freitas N, Gordon G. *Sequential Monte Carlo Methods in Practice.* New York: Springer-Verlag, 2001.

Eden UT, Frank LM, Barbieri R, Solo V, Brown EN. Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Comput* 16: 971–998, 2004a.

Eden UT, Truccolo W, Fellows MR, Donoghue JP, Brown EN. Reconstruction of hand movement trajectories from a dynamic ensemble of spiking motor cortical neurons. *Proc IEEE Eng Med Biol Soc Annu Conf* 2: 4017–4020, 2004b.

Ergun A, Barbieri R, Eden UT, Wilson MA, Brown EN. Construction of point process adaptive filter algorithms for neural systems using sequential Monte Carlo methods. *IEEE Trans Biomed Eng* 54: 419–428, 2007.

Farmer ME, Hsu R-L, Jain AK. Interacting multiple model (IMM) Kalman filters for robust high speed human motion tracking. *Proc Int Conf Pattern Recogn* 2: 20–23, 2002.

Flash T, Hogan N. The coordination of arm movements: an experimentally confirmed mathematical model. *J Neurosci* 5: 1688–1703, 1985.

Frey DD, Carlson LE, Ramaswamy V. Voluntary-opening prehensors with adjustable grip force. *J Prosthetics Orthotics* 7: 124–131, 1995.

Georgopoulos AP, Schwartz AB, Kettner RE. Neuronal population coding of movement direction. *Science* 233: 1416–1419, 1986.

Hatsopoulos N, Joshi J, O'Leary JG. Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles. *J Neurophysiol* 92: 1165–1174, 2004.

Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, Caplan AH, Branner A, Chen D, Penn RD, Donoghue JP. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* 442: 164–171, 2006.

Jordan MI. Graphical models. *Stat Sci* 19: 140–155, 2004.

Kailath T, Sayed AH, Hassibi B. *Linear Estimation.* Upper Saddle River, NJ: Prentice Hall, 2000.

Kemere C, Meng TH. Optimal estimation of feed-forward-controlled linear systems. *Proc IEEE Int Conf Acoustics Speech Signal Process* 5: 353–356, 2005.

Kemere C, Sahani M, Meng TH. Robust neural decoding of reaching movements for prosthetic systems. *Proc Annu Mtg IEEE EMBS* 3: 2079–2082, 2003.

Kemere C, Shenoy KV, Meng TH. Model-based neural decoding of reaching movements: a maximum likelihood approach. *IEEE Trans Biomed Eng* 51: 925–932, 2004.

Kepner J. Parallel programming with MatlabMPI. In: *http://www.ll.mit.edu/MatlabMPI/*, 2007.

Kim SP, Sanchez JC, Rao YN, Erdogmus D, Carmena JM, Lebedev MA, Nicolelis MA, Principe JC. A comparison of optimal MIMO linear and nonlinear models for brain-machine interfaces. *J Neural Eng* 3: 145–161, 2006.

Kubler A, Mushahwar VK, Hochberg LR, Donoghue JP. BCI meeting 2005—workshop on clinical issues and applications. *IEEE Trans Biomed Eng* 14: 131–134, 2006.

Lebedev MA, Nicolelis MA. Brain machine interfaces: past, present, future. *Trends Neurosci* 29: 536–546, 2006.

Leuthardt EC, Schalk G, Wolpaw JR, Ojemann JG, Moran DW. A brain-computer interface using electrocorticographic signals in humans. *J Neural Eng* 1: 63–71, 2004.

Li CS, Padoa-Schioppa C, Bizzi E. Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field. *Neuron* 30: 593–607, 2001.

Mazor E, Averbuch A, Bar-Shalom Y, Dayan J. Interacting multiple model methods in target tracking: a survey. *IEEE Trans Aerospace Electronic Syst* 34: 103–123, 1998.

McGarty TP. *Stochastic Systems and State Estimation.* New York: Wiley, 1974.

Moran DW, Schwartz AB. Motor cortical representation of speed and direction during reaching. *J Neurophysiol* 82: 2676–2692, 1999.

Murphy K. Switching Kalman Filters. *UC Berkeley Technical Report*, 1998, citeseer.ist.psu.edu/article/murphy98switching.html.

Musallam S, Corneil BD, Greger B, Scherberger H, Andersen RA. Cognitive control signals for neural prosthetics. *Science* 305: 258–262, 2004.

Muthuswamy J, Okandan M, Gilletti A, Baker MS, Jain T. An array of microactuated microelectrodes for monitoring single-neuronal activity in rodents. *IEEE Trans Biomed Eng* 52: 1470–1477, 2005.

Olson BP, Si J, Hu J, He J. Closed-loop cortical control of direction using support vector machines. *IEEE Trans Neural Syst Rehabil Eng* 13: 72–80, 2005.

Pesaran B, Musallam S, Andersen RA. Cognitive neural prosthetics. *Curr Biol* 16: R77–80, 2006.

Santhanam G, Ryu SI, Yu BM, Afshar A, Shenoy KV. A high-performance brain-computer interface. *Nature* 442: 195–198, 2006.

Sarpeshkar R, Salthouse C, Sit JJ, Baker MW, Zhak SM, Lu TK, Turicchia L, Balster S. An ultra-low-power programmable analog bionic ear processor. *IEEE Trans Biomed Eng* 52: 711–727, 2005.

Schall JD. Neural basis of deciding, choosing and acting. *Nat Rev Neurosci* 2: 33–42, 2001.

Schwartz AB. Cortical neural prosthetics. *Annu Rev Neurosci* 27: 487–507, 2004.

Schwartz AB, Cui XT, Weber DJ, Moran DW. Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron* 52: 205–220, 2006.

Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR, Donoghue JP. Instant neural control of a movement signal. *Nature* 416: 141–142, 2002.

Shenoy KV, Meeker D, Cao S, Kureshi SA, Pesaran B, Buneo CA, Batista AP, Mitra PP, Burdick JW, Andersen RA. Neural prosthetic control signals from plan activity. *Neuroreport* 14: 591–596, 2003.

Snyder DL, Miller MI. *Random Point Processes in Time and Space.* New York: Springer-Verlag, 1991.

Srinivasan L, Brown EN. Dynamic-goal state equations for tracking reaching movements using neural signals. *1st IEEE RAS-EMBS Internat Conf on Biomed Robotics and Biomechatronics (BioRob '06)*. Pisa, Italy, 2006.

Srinivasan L, Brown EN. A state space framework for movement control to dynamic goals through brain-driven interfaces. *IEEE Trans Biomed Eng* 54: 526–535, 2007.

Srinivasan L, Eden UT, Willsky AS, Brown EN. Goal-directed state equation for tracking reaching movements using neural signals. *Proc 2nd Internat IEEE EMBS Conf on Neural Engineering* 352–355, 2005.

Srinivasan L, Eden UT, Willsky AS, Brown EN. A state-space analysis for reconstruction of goal-directed movements using neural signals. *Neural Comput* 18: 2465–2494, 2006a.

Srinivasan L, Mitter SK, Brown EN, Hatsopoulos N. Structure of delay period target representation in premotor dorsal cortex (PMd). *Soc Neurosci Abstr* 307.10, 2006b.

Tarvainen MP, Hiltunen JK, Ranta-aho PO, Karjalainen PA. Estimation of nonstationary EEG with Kalman smoother approach: an application to event-related synchronization (ERS). *IEEE Trans Biomed Eng* 51: 516–524, 2004.

Taylor DM, Tillery SI, Schwartz AB. Direct cortical control of 3D neuroprosthetic devices. *Science* 296: 1829–1832, 2002.

Todorov E. Optimality principles in sensorimotor control. *Nat Neurosci* 7: 907–915, 2004.

Todorov E, Jordan MI. Optimal feedback control as a theory of motor coordination. *Nat Neurosci* 5: 1226–1235, 2002.

Truccolo W, Eden UT, Fellows MR, Donoghue JP, Brown EN. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J Neurophysiol* 93: 1074–1089, 2005.

Uno Y, Kawato M, Suzuki R. Formation and control of optimal trajectory in human multijoint arm movement. Minimum torque-change model. *Biol Cybern* 61: 89–101, 1989.

Wahnoun R, He J, Helms Tillery SI. Selection and parameterization of cortical neurons for neuroprosthetic control. *J Neural Eng* 3: 162–171, 2006.

Weinrich M, Wise SP. The premotor cortex of the monkey. *J Neurosci* 2: 1329–1345, 1982.

Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, Nicolelis MA. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* 408: 361–365, 2000.

Wolpaw JR, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM. Brain-computer interfaces for communication and control. *Clin Neurophysiol* 113: 767–791, 2002.

Wolpaw JR, McFarland DJ. Multichannel EEG-based brain-computer communication. *Electroencephalogr Clin Neurophysiol* 90: 444–449, 1994.

Wolpaw JR, McFarland DJ. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proc Natl Acad Sci USA* 101: 17849–17854, 2004.

Wu W, Black MJ, Mumford D, Gao Y, Bienenstock E, Donoghue JP. Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Trans Biomed Eng* 51: 933–942, 2004.

Wu W, Gao Y, Bienenstock E, Donoghue JP, Black MJ. Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Comput* 18: 80–118, 2006.

Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SI, Meng TH, Sahani M, Shenoy KV. Mixture of trajectory models for neural decoding of goal-directed movements. *Soc Neurosci Abstr* 52018, 2005a.

Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SI, Meng TH, Sahani M, Shenoy KV. Mixture of trajectory models for neural decoding of goal-directed movements. *J Neurophysiol* 97: 3763–3780, 2007.

Yu BM, Santhanam G, Ryu SI, Shenoy KV. Feedback-directed state transition for recursive Bayesian estimation of goal-directed trajectories. *Computational and Systems Neuroscience (COSYNE) Meeting Abstract, Salt Lake City, UT*, 2005b.