# Real-Time Brain-Machine Interface Architectures: Neural Decoding from Plan to Movement



by

Maryam Modir Shanechi

B.A.Sc., Engineering Science
University of Toronto, 2004
S.M., Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2006

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011



Author .....
Department of Electrical Engineering and Computer Science
April 29, 2011

Certified by .....
Gregory W. Wornell
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by .....
Emery N. Brown
Professor of Computational Neuroscience and Health Sciences and Technology
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

**Real-Time Brain-Machine Interface Architectures:**

**Neural Decoding from Plan to Movement**

by

Maryam Modir Shanechi

Submitted to the Department of Electrical Engineering and Computer Science
on April 29, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

## Abstract

Brain-machine interfaces (BMI) aim to enable motor function in individuals with neurological injury or disease, by recording the neural activity, mapping or 'decoding' it into a motor command, and then controlling a device such as a computer interface or robotic arm. BMI research has largely focused on the problem of restoring the original motor function. The goal therefore has been to achieve a performance close to that of the healthy individual. There have been compelling proof of concept demonstrations of the utility of such BMIs in the past decade. However, performance of these systems needs to be significantly improved before they become clinically viable. Moreover, while developing high-performance BMIs with the goal of matching the original motor function is indeed valuable, a compelling goal is that of designing BMIs that can surpass original motor function. In this thesis, we first develop a novel real-time BMI for restoration of natural motor function. We then introduce a BMI architecture aimed at enhancing original motor function. We implement both our designs in rhesus monkeys.

To facilitate the restoration of lost motor function, BMIs have focused on either estimating the continuous movement trajectory or target intent. However, natural movement often incorporates both. Moreover, both target and trajectory information are encoded in the motor cortical areas. These suggest that BMIs should be designed to combine these principal aspects of movement. We develop a novel two-stage BMI to decode jointly the target and trajectory of a reaching movement. First, we decode the intended target from neural spiking activity before movement initiation. Second, we combine the decoded target with the spiking activity during movement to estimate the trajectory. To do so, we use an optimal feedback-control design that aims to emulate the sensorimotor processing underlying actual motor control and directly processes the spiking activity using point process modeling in real time. We show that the two-stage BMI performs more accurately than either stage alone. Correct target prediction can compensate for inaccurate trajectory estimation and vice versa. This BMI also performs significantly better than linear regression approaches demonstrating the advantage of a design that more closely mimics the sensorimotor system.

While restoring the original motor function is indeed important, a compelling goal is the development of a truly "intelligent" BMI that can transcend such function by considering the higher-level goal of the motor activity, and reformulating the motor plan accordingly. This would allow, for example, a task to be performed more quickly than possible by natural movement, or more efficiently than originally conceived. Since a typical motor activity consists of a sequence of planned movements, such a BMI must be capable of analyzing the complete sequence before action. As such its feasibility hinges fundamentally on whether all elements of the motor plan can be decoded concurrently from working memory. Here we demonstrate that such concurrent decoding is possible. In particular, we develop and implement a real-time BMI that accurately and simultaneously

decodes in advance a sequence of planned movements from neural activity in the premotor cortex. In our experiments, monkeys were trained to add to working memory, in order, two distinct target locations on a screen, then move a cursor to each, in sequence. We find that the two elements of the motor plan, corresponding to the two targets, are encoded concurrently during the working memory period. Additionally, and interestingly, our results reveal: that the elements of the plan are encoded by largely disjoint subpopulations of neurons; that surprisingly small subpopulations are sufficient for reliable decoding of the motor plan; and that the subpopulation dedicated to the first target and their responses are largely unchanged when the second target is added to working memory, so that the process of adding information does not compromise the integrity of existing information. The results have significant implications for the architecture and design of future generations of BMIs with enhanced motor function capabilities.

Thesis Supervisor: Gregory W. Wornell
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Emery N. Brown
Title: Professor of Computational Neuroscience and Health Sciences and Technology

*I dedicate this thesis to my parents:*
*Shahla Lotfi and Hasan Modir Shanechi,*
*for all their sacrifices.*

## Acknowledgments

It has been a long journey since the first day I started as a master's student at MIT with unexpected and exciting turns. I came here and started working with Prof. Gregory Wornell wanting to learn and apply the tools of information theory and inference to the problems in communication systems. When it came to choosing a PhD topic, I became interested in applying these tools to problems in neuroscience. I am extremely grateful to Greg for being very encouraging of me pursuing a PhD topic in an application other than what was traditionally done in our group. At that point, I got fortunate enough to hear about the work in Prof. Emery Brown's group. I found it fascinating how one could use the tools of inference to decode the neural code and interpret the brain function. I am extremely grateful to Emery for agreeing to co-advise me even though I had little background in neuroscience at the time, and for introducing me to different people in the field so that I could find the right problem. Emery and Greg have always amazed me with the broadness and depth of their knowledge, their ability to convey difficult technical details in a clear and simple manner, and their supportiveness throughout my PhD work. I have learned a great deal from both of them and they have been instrumental in shaping my career. I thank them both greatly.

I thank my committee member, Dr. Ziv Williams. Working with Ziv in the course of our experiments, not only did I learn a lot about experimental design, but also I gained valuable insights into algorithm design. I really appreciate the many hours he spent working with me on formulating what were the important questions to ask and providing me with great insights. I thank my committee member, Prof. Polina Golland. I got to know Polina when I worked as a teaching assistant with her at the start of my PhD and she was indeed one of the main figures who encouraged me to look at applications other than what I was used to. In the process, she provided me with great guidance whenever I needed it.

I thank all my colleagues in the SIA group, in particular, Uri Erez, Ying-zong Huang, Charles Swannack, Ashish Khisti, Urs Niesen, Vijay Divi, Da Wang, Quing He, Yuval Kochman, Gauri Joshi, James Krieger, and of course our great administrative assistant Tricia O'Donnell. I learned so much from all of them. I thank my friends for all the fun times and the support during the tough times, Farinaz Edalat, Rumi Chunara, Joyce Kwong, and Manish Bhardwaj. I specially thank my best friend and roommate of six years, Vivienne Sze for being there for me through the good and the bad, for being my family in Boston.

I thank my fiance, Behnam Jafarpour, for his love, support, and understanding throughout these years. I cannot wait to get married in July and live and share the rest of my life with you.

I thank my sister, Mercedeh, and my brother, Amirali, whom I can always count on, for all their love, support, and encouragements. And last but not least, I express my utmost gratitude to my parents, Hasan Shanechi and Shahla Lotfi, or as my fiance and sister call them, my '911'. I could not have dreamt of getting here without your love, support, and sacrifice. Growing up as a little girl, it was your constant encouragement that gave me the confidence to come all the way to MIT. I can never repay for the sacrifices you have made for me. But as the smallest token of my appreciation, I dedicate this thesis to you.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Brain-machine interfaces (BMI) also known as brain-computer interfaces (BCI) or neural prosthetic devices aim to enable motor function in individuals whose movement abilities have been impaired due to neurological injury or disease, such as spinal cord injury, stroke, or degenerative disorders (e.g., amyotrophic lateral sclerosis or ALS). In such cases, the injury or disease disconnects the otherwise healthy brain from the musculo-skeletal system leading to motor paralysis. Thus the goal of the BMI is to enable a direct interface to the healthy brain.

BMIs, therefore, consist of the following components. A recording device such as an intracortical electrode array, a decoding algorithm that interprets the recordings and maps them into movement commands, and a device that is in turn actuated or controlled by such commands such as a robotic arm, a computer cursor, or the native muscles. In the process, the BMIs can also provide some modality of sensory feedback about the movement to the individual. The most prominent source of this feedback so far has been visual [1–3].

The aim of the BMI research thus far has been to restore the original motor function to the paralyzed individual by attempting to obtain a performance close to that of the healthy individual. These designs have been tested in human and non-human primates [4–19] in the past decade and have provided a compelling proof of concept demonstration of the utility of such BMIs for the restoration of motor function. However, there is still much work to be done in order to improve the performance of BMIs and in turn enable their clinical viability.

While improving the performance of such BMIs to match the original motor functionality is indeed of great importance, BMIs also introduce new opportunities by creating a *direct* interface to the brain—namely, that of *enhancing* original motor functionality to achieve the motor goal of the individual more effectively than possible by natural movement. This possibility has not been traditionally explored in the BMI literature.

In the first part of this thesis (Chapters 2,3, and 4), we design and implement a novel BMI for more accurate estimation of natural movement. In the second part of this thesis (Chapters 5 and 6) we introduce the opportunity for BMI designs to enhance the original motor function and more significantly develop and implement a novel real-time BMI aimed at this goal.

## 1.1 Towards more accurate brain-machine interface designs for restoration of motor function

There are currently two major types of BMIs for restoration of motor function. The first type, which comprises the majority of these designs, attempts to estimate a continuous trajectory—for example the position of a computer cursor on the screen moving towards a visual target [4–17]. Recent efforts with this type of BMIs have demonstrated the ability to estimate continuous movement from motor cortical activity. However, the estimated movement parameters in this type have not closely matched the natural movement and hence the BMI performance has been far inferior to that. For example, the time required to reach the target on the screen in most these experiments is much longer than the subject's own movement. The other type of BMIs, which are faster than the former type, aim to predict a desired discrete target without estimating the corresponding desired trajectory towards it [18, 19]. While such BMIs are valuable for purposes such as typing on a keyboard, they do not provide trajectories that can enable natural and autonomous movement. One major factor that can improve the performance of BMIs is the design of more accurate decoding algorithms used to translate the neural signal into motor commands.

In the first part of this thesis (Chapters 2,3, and 4), we concentrate on developing more accurate real-time decoding algorithms for restoring natural motor function. We first develop the theoretical foundation for a novel real-time decoder that jointly estimates both the target and trajectory of a goal-directed movement and then demonstrate its implementation in a real-time BMI in two rhesus monkeys. Thus far BMIs have either decoded the target of single-targeted movements or continuous trajectories. However, information about both the high-level goal or target as well as the trajectory of movement is encoded in the motor cortical areas [5, 18, 20–40]. Target or goal information has been shown to be encoded far before movement execution [18, 25, 27, 30–40] and trajectory or kinematic information has been shown to be encoded around the time of movement [5, 20–29]. Moreover, most natural movements are performed to achieve a goal or target and hence the target and trajectory of movement are strongly correlated [41, 42]. These all suggest that a real-time BMI design taking advantage of both target and trajectory related information can result in a more accurate estimation of movement.

The main question that arises at this point is how to combine the target and trajectory related information. As target information is encoded prior to movement, one way to do so is to incorporate this information in the construction of a prior goal-directed state-space model for

movement [43,44], which can in turn be combined in a recursive Bayesian decoder with the peri-movement activity, i.e., the activity around the time of movement, to estimate the trajectory. However, any goal-directed state-space model (regardless of its particular design) is time-varying and dependent on the movement duration, which is not known a priori to a real-time decoder. This duration dependence is evident by considering that close to reaching a target there is much more constraint on the movement compared to far from it since in the former case the trajectory soon needs to converge to the target location. Hence prior work that have used a goal-directed state-space model to incorporate the target information in movement estimation have been simulation based and assumed that the movement duration is known [43,44]. However, this is not the case in real-time BMIs and hence goal-directed models have not been implemented in them. Therefore we first address the problem of resolving the duration uncertainty in *any* goal-directed state-space model for the purpose of movement estimation from ensemble spiking activity in Chapter 2. This approach in turn enables the incorporation of goal-directed prior models in real-time BMIs.

Having developed a framework to resolve the duration uncertainty in any goal-directed state-space model, we design one such model in Chapter 3. Namely, to build a goal-directed prior model, we take advantage of the optimal feedback control theory of the sensorimotor system [41,42] in order to emulate the sensorimotor processing underlying actual motor control in our decoder. The optimal feedback control model of the sensorimotor system has been used to interpret its function. In this framework each task is performed to accomplish a goal during which there is sensory feedback, such as vision or proprioception, about the state of the musculo-skeletal system. Based on the goal and this feedback, the brain acts as an optimal feedback controller and decides on the next plan of action to control its movement. We demonstrate the advantage of this model in estimating reaching movements in this chapter.

Having developed the theoretical foundation for incorporation of both target and trajectory related activity in estimation of movement, in Chapter 4 we develop a real-time BMI that, for the first time, jointly estimates the target of a movement and its continuous trajectory in two rhesus monkeys. Our BMI has a novel two-stage design. In the first stage it decodes the target of movement prior to its initiation and in the second stage it combines this target information with the peri-movement spiking activity. To do so, we use an optimal feedback control design based on the theoretical developments in Chapters 2 and 3 to mimic the sensorimotor processing underlying motor control and model the visual feedback present in the BMI. Another novel feature of this BMI is that, for the first time, it processes the ensemble spiking activity directly to estimate

the movement. Even though intra-cortical recordings capture the spiking activity of an ensemble of neurons, real-time BMIs thus far have not used this recorded activity directly to estimate the movement. Instead they have used a smoothed function of the spiking activity as input in real time to adapt it to the use of well-known algorithms such as a regression or a Kalman filter.

We demonstrate that the two stages of our BMI act in a complementary manner and result in a more accurate movement estimation than would be possible using either target or trajectory estimation alone. We also show that this BMI performs more accurately than a commonly used linear regression decoder.

## ■ 1.2 Towards brain-machine interface designs for enhanced motor function

In the second part of this thesis (Chapters 5 and 6) we shift our attention to introduce a unique opportunity presented by having a direct interface to the brain. So far in the BMI literature the focus has been on the design and improvement of BMIs that aim to restore the original motor functionality of the individual. Particularly, these designs have focused on achieving simple motor goals such as that associated with moving a cursor to a single target location on the screen. Such designs are indeed valuable and can enable the execution of many everyday tasks. However, by creating a direct interface to the brain, BMIs present an even more compelling opportunity. Namely, that of *surpassing* the natural motor functionality of an individual.

To illustrate this, note that in many natural tasks the goal is more complex than reaching a single target location and hence the motor plan for achieving it involves a sequence of such simple plan elements—for example the task of picking up a book and then putting it on a shelf. One way to execute such sequential plan elements is to use a BMI, for example our two-stage BMI in Chapter 4, to decode and execute the plan elements of a task one by one and in serial order. This in turn results in a *sequential* BMI architecture. However, an ultimately even more valuable BMI would be one that decodes the full sequence of motor plan elements simultaneously before execution, then analyzes the sequence to determine ways to perform the task more effectively. For example, the BMI might determine a way for the task to be accomplished more quickly. Or, based on additional sensor inputs, the BMI might determine that the planned sequence of movements would result in an accident and thus modify the execution of the task to avoid such an accident.

Much technological progress is needed for the implementation of such "intelligent" BMIs. But most importantly, as the foundation of such an evolution, our designs need to shift to what we

refer to as a *concurrent* BMI architecture. In a concurrent architecture, the elements of a planned motor task are decoded in parallel, in contrast to the serial process of a sequential BMI. Hence, the first requirement for the realization of such concurrent BMIs is the ability to decode the elements of a motor plan in parallel and before movement execution.

In Chapter 5 we turn to the design of BMIs for such sequential plans. We demonstrate that, indeed, it is possible to decode motor plan elements in parallel from neural activity of the premotor cortex in primates during working memory. Moreover we show that surprisingly small subpopulations are sufficient for such accurate decoding. More significantly, we design and implement a real-time BMI with such concurrent plan element decoding in two rhesus monkeys. In our experiments, monkeys are required to make a sequential movement to two target locations. Thus the two plan elements consist of the two targeted movements. We show that both targets can be decoded accurately and concurrently during working memory and before movement initiation. This in turn enables the BMI to execute both movements much faster than possible using natural sequential movement.

Having observed that such accurate and concurrent decoding of multiple motor targets is possible during working memory, in Chapter 6 we additionally explore the underlying neural mechanisms in the premotor cortex. Interestingly, we show that the two targets are encoded in the responses of largely disjoint subpopulations of premotor neurons. Together, these observations have significant implications for future BMI designs aimed at surpassing natural motor function.

Finally in Chapter 7 we summarize our contributions and present some future directions.

# Chapter 2
# Estimation of goal-directed movements of unknown duration

In this chapter we address the problem of movement duration uncertainty in prior goal-directed state-space models for estimation of movement using recursive Bayesian approaches. A recursive Bayesian decoder consists of two probabilistic models: the prior model on the time sequence of kinematic states, and the observation model relating the neural signal to these states. The prior model should in general incorporate any prior information available about the kinematics states. A widely used prior state model is the linear random-walk model (see, e.g., [45–47]), which mainly enforces a smoothness constraint on the kinematic states. However, most natural movements are made to achieve a goal or target location. For these goal-directed movements, the encoding of target information in the neural activity before movement [18,25,27,30–40] and successful attempts at its decoding [18,19] suggest that one should also incorporate this target information in the prior model to obtain more accurate trajectory estimations. For example one can condition the random-walk model on being at the target at a known arrival time [43, 48] or use a linear feedforward controlled (i.e., not taking into account the sensory feedback) state-space model [44] assuming a known arrival time.

Regardless of the approach taken to incorporate target information into the prior state-space model, goal-directed state-space models are inherently dependent on the arrival time at the target or equivalently the movement duration. For example there is much more constraint on the movement kinematics close to the arrival time compared to far from it as in the former case the trajectory soon needs to converge to the intended target. Also the assumption of known arrival time used in [43] and [44] is binding since the movement duration is not known a priori to an external observer, observing only the neural signal. Hence to be implementable in a real-time BMI, a practical decoder needs to address this uncertainty in movement duration in any goal-directed state-space model.

In this chapter we develop a recursive Bayesian decoder for goal-directed movements that can take advantage of *any* goal-directed state-space model but does not require prior knowledge of the arrival time. Hence our emphasis here is to design a decoder that can resolve the duration uncer-

tainty inherent to goal-directed prior state-space models regardless of their specific construction. Our decoder addresses this uncertainty in the arrival time by exploiting a parallel bank of point process filters that calculate not only causal estimates of the state at each time based on the neural observations, but also the likelihood of the arrival time based on these observations. Since these filters run in parallel, the time to generate the overall estimate is on the order of the run time of a single filter, a property which makes this approach appropriate for real-time implementation. Our approach can be applied to any goal-directed state model such as the ones in [43] and [44] to generalize them and relax their assumption of known movement duration. Hence it paves the way for the incorporation of goal-directed models in real-time BMIs, which in turn enable the combination of both target-related and kinematics-related neural activity to obtain more accurate trajectory estimation.

## ■ 2.1 Methodology

We denote the sequence of kinematic states by $\mathbf{x}_0, \cdots, \mathbf{x}_t$ and the neural point process observations of the ensemble of $C$ neurons by $\mathbf{N}_1, \cdots, \mathbf{N}_t$ where $\mathbf{N}_t = (N_t^1, \cdots, N_t^C)$ is the binary spike events of the $C$ neurons at time $t$. Assuming that the observations from the $C$ neurons are conditionally independent given the state, the point process observation model is given by [49,50]

$$p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{H}_t) = \prod_c (\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)\Delta)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)\Delta} \tag{2.1}$$

where $\mathbf{H}_t^c = N_{1:t-1}^c$, $\mathbf{H}_t = \mathbf{N}_{1:t-1} = \mathbf{H}_t^{1:C}$ model the history dependence of the spiking activity, $\Delta$ is the time increment, and $\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)$ is the modeled instantaneous firing rate or equivalently the conditional intensity function of the $c$th neuron at time $t$. Note that we have explicitly indicated the encoding of the kinematics states $\mathbf{x}_t$ in the conditional intensity function (the form of which depends on the kinematics tuning model and will be discussed further in Section 2.2). When the observation model further assumes that the spiking activity is not history dependent, i.e., that the observation at each time step is conditionally independent of the observation in the previous time steps given the kinematic state, then the observation model simplifies to

$$\begin{aligned} p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{H}_t) &= p(\mathbf{N}_t|\mathbf{x}_t) && (2.2) \\ &= \prod_c (\lambda_c(t|\mathbf{x}_t)\Delta)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t)\Delta} && (2.3) \end{aligned}$$

The goal of the decoder is to causally calculate the state posterior density, i.e., $p(\mathbf{x}_t|\mathbf{N}_{1:t})$ based on the neural observations. We derive the decoder for the general observation model with history dependence but the decoder recursions can be simplified easily for the case where such dependence does not exist.

### ■ 2.1.1 Prior state-space models for goal-directed movements

A Bayesian decoder for goal-directed movements should incorporate information about the target of the movement into the prior state-space model. Hence, these state-space models are inherently dependent on the arrival time at the target or equivalently the movement duration, which is not known a priori to the decoder observing only the neural signal. Since our emphasis in this chapter is not to design a goal-directed state-space model but rather to design a recursive Bayesian decoder that can use any such model without requiring prior knowledge of the arrival time, here we consider a general linear state-space model,

$$\mathbf{x}_{t+1} = \mathbf{G}(t, T, \mathbf{x}^*)\mathbf{x}_t + \mathbf{b}(t, T, \mathbf{x}^*) + \mathbf{w}_t, \tag{2.4}$$

where $\mathbf{x}^*$ is the target state, $T$ is the arrival time, $\mathbf{G}$ and $\mathbf{b}$ are the dynamics matrix and a constant term, respectively, that in general are time-varying and also dependent on $\mathbf{x}^*$ and $T$, and $\mathbf{w}_t$ is the zero-mean white Gaussian state noise with covariance matrix $\mathbf{W}_t$. The state-space models in both [43] and [44] can be written in this form and assume prior knowledge of $T$.

**Random-walk prior state-space model**

By setting $\mathbf{b} = \mathbf{0}$ and removing the dependence of $\mathbf{G}$ on $T$ and $\mathbf{x}^*$, the state-space model in (2.4) is reduced to a general form of the random-walk state-space model. In many cases, $\mathbf{G}$ is also assumed to be time-invariant for the case of random-walk models [45–47].

### ■ 2.1.2 Estimation of goal-directed kinematic states from neural observations

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

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{N}_{1:t}, T)\, p(T|\mathbf{N}_{1:t})\, dT \tag{2.5}$$

where $p(T|\mathbf{N}_{1:t})$ is the causal likelihood function for the arrival time based on the neural observations. However, in its continuous form, this likelihood is difficult to find and in general is not

Gaussian. Hence, to make the computations tractable, we discretize the arrival time and instead expand the posterior density as,

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_{j=1}^{J} p(\mathbf{x}_t|\mathbf{N}_{1:t}, T_j)\, p(T_j|\mathbf{N}_{1:t}) \tag{2.6}$$

where $T$ is discretized to $J$ possibilities and consequently a prior model is placed on it given by $p_T(T_j), j = 1, \cdots, J$. This prior model (including its support) can be selected based on empirical durations observed in a given task. We will later examine how the discretization refinement, or equivalently the number of discretization points used for a given support, affects the performance of the decoder and show in the Section 2.2 that even a coarse discretization is sufficient to obtain a small loss in performance compared to the case where movement duration is known.

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

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

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}, T) = \frac{p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1})\, p(\mathbf{x}_t|\mathbf{N}_{1:t-1}, T)}{p(\mathbf{N}_t|\mathbf{N}_{1:t-1}, T)} \tag{2.7}$$

The first term in the numerator comes from the observation model in (2.1) and the second term is the one-step prediction density. Note that we used the fact $p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1}, T) = p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1})$ based on the observation model in (2.1). We denote the partition function by

$$g(\mathbf{N}_t|T) = p(\mathbf{N}_t|\mathbf{N}_{1:t-1}, T) \tag{2.8}$$

since we will exploit it later to find the corresponding weights. To get the recursion for the posterior density, we write the prediction density as

$$p(\mathbf{x}_t|\mathbf{N}_{1:t-1}, T) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, T) p(\mathbf{x}_{t-1}|\mathbf{N}_{1:t-1}, T) d\mathbf{x}_{t-1} \tag{2.9}$$

using the conditional independence, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{N}_{1:t-1}, T) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, T)$, which comes from the state-space model in (2.4). Now the second term inside the integral is just the posterior density from the previous time step, hence substituting (2.9) into (2.7) generates the recursion. The exact expression in (2.7) is in general complicated. Hence here we make a Gaussian approximation to the

posterior density similar to [49] using the Laplace's method. Making this approximation and since the state-space model in (2.4) is also Gaussian, the prediction density in (2.9) will be Gaussian. Let's denote the minimum MSE (MMSE) estimator, i.e., $E(\mathbf{x}_t|\mathbf{N}_{1:t}, T)$ by $\mathbf{x}_{t|t,T}$ and its covariance matrix by $\mathbf{W}_{t|t,T}$. Similarly we denote the one step prediction mean by $\mathbf{x}_{t|t-1,T} = E(\mathbf{x}_t|\mathbf{N}_{1:t-1}, T)$ and its covariance matrix by $\mathbf{W}_{t|t-1,T}$. The Gaussian approximation to the posterior in (2.7) is made by expanding its logarithm in a Taylor series expansion around the prediction mean $\mathbf{x}_{t|t-1,T}$ and keeping terms of second order or smaller. The recursions for the MMSE estimator with this Gaussian approximation have been derived in [49]. The difference here is in the prediction step since the state-space model in (2.4) in general could have a constant term. The recursions of this point process filter in our case become

$$\mathbf{x}_{t|t-1,T} = \mathbf{G}(t, T, \mathbf{x}^*)\mathbf{x}_{t-1|t-1,T} + \mathbf{b}(t, T, x^*) \tag{2.10}$$

$$\mathbf{W}_{t|t-1,T} = \mathbf{G}(t, T, \mathbf{x}^*)\mathbf{W}_{t-1|t-1,T}\mathbf{G}'(t, T, \mathbf{x}^*) + \mathbf{W}_t \tag{2.11}$$

$$\begin{aligned}\mathbf{W}^{-1}_{t|t,T} = \mathbf{W}^{-1}_{t|t-1,T} + \sum_{c=1}^{C} \Bigg[ &\left(\frac{\partial \log \lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)}{\partial \mathbf{x}_t}\right)' \left(\frac{\partial \log \lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)}{\partial \mathbf{x}_t}\right) \lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)\Delta \\ &-(N^c_t - \lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)\Delta)\frac{\partial^2 \log \lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)}{\partial \mathbf{x}_t \partial \mathbf{x}'_t}\Bigg]_{\mathbf{x}_{t|t-1,T}}\end{aligned} \tag{2.12}$$

$$\mathbf{x}_{t|t,T} = \mathbf{x}_{t|t-1,T} + \mathbf{W}_{t|t,T} \sum_{c=1}^{C} \left[\left(\frac{\partial \log \lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)}{\partial \mathbf{x}_t}\right)' (N^c_t - \lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)\Delta)\right]_{\mathbf{x}_{t|t-1,T}} \tag{2.13}$$

where $[\cdot]_{\mathbf{x}_{t|t-1,T}}$ denotes the evaluation of the expression at $\mathbf{x}_{t|t-1,T}$. When the point process model for the spiking activity is assumed to have no history dependence, the recursions are obtained by simply replacing $\lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)$ with $\lambda_c(t|\mathbf{x}_t)$. This gives us the goal-directed point process filter with a known arrival time. Since $\mathbf{b} = \mathbf{0}$ in (2.4) corresponds to the random-walk model, (2.10)-(2.13) with $\mathbf{b} = \mathbf{0}$ recover the random-walk point process filter (RW-PPF).

So far we have found an approximation of the posterior assuming a *known* arrival time. To relax this assumption, we have to find the corresponding weights for each arrival time $p(T_j|\mathbf{N}_{1:t})$ and use (2.6) to find the posterior density. We can write

$$p(T_j|\mathbf{N}_{1:t}) = \frac{p(\mathbf{N}_{1:t}|T_j)p_T(T_j)}{p(\mathbf{N}_{1:t})}, \tag{2.14}$$

where $p(\mathbf{N}_{1:t})$ is independent of $T_j$ and hence treated as a constant and $p(\mathbf{N}_{1:t}|T_j)$ represents the likelihood of the observed neural data under a state-space model with the arrival time of

$T_j$. Hence, it is the partition function for the posterior $p(\mathbf{x}_{1:t}|\mathbf{N}_{1:t}, T_j)$ and its exact computation requires an integration, which is computationally prohibitive. However, using the Gaussian approximation to the posterior, we can find this without direct integration as follows. Using the chain rule we have

$$p(\mathbf{N}_{1:t}|\, T_j) = \prod_{i=1}^{t} P(\mathbf{N}_i|\mathbf{N}_{1:i-1}, T_j) = \prod_{i=1}^{t} g(\mathbf{N}_i|\, T_j) \tag{2.15}$$

where $g(\mathbf{N}_i|\, T_j)$ is defined in (2.8) and is the $i$th step partition function in the recursive filter. Now exploiting the Gaussian approximation of the posterior and hence the prediction densities in (2.7), they are completely characterized by their means and covariances given in (2.10)-(2.13) for a given $T_j$. We can hence explicitly evaluate (2.7) at $\mathbf{x}_{i|i,T_j}$ to get

$$\begin{aligned} g(\mathbf{N}_i|\, T_j) &= \sqrt{\frac{|\mathbf{W}_{i|i,T_j}|}{|\mathbf{W}_{i|i-1,T_j}|}}\, p(\mathbf{N}_i|\mathbf{x}_{i|i,T_j}, \mathbf{N}_{1:i-1}) \times \\ &\quad \exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})' \mathbf{W}^{-1}_{i|i-1,T_j} (\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})\right] \\ &= \sqrt{\frac{|\mathbf{W}_{i|i,T_j}|}{|\mathbf{W}_{i|i-1,T_j}|}} \prod_c \left(\lambda_c(i|\mathbf{x}_{i|i,T_j}, \mathbf{H}_i^c)\Delta\right)^{N_i^c} e^{-\lambda_c(i|\mathbf{x}_{i|i,T_j}, \mathbf{H}_i^c)\Delta} \times \\ &\quad \exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})' \mathbf{W}^{-1}_{i|i-1,T_j} (\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})\right] \end{aligned} \tag{2.16}$$

for $i = 1, \cdots, T_j$ and $j = 1, \cdots, J$ where all the quantities are known. Combining (2.6)-(2.16) gives the posterior. The MMSE estimate in the case of unknown arrival time is then given by,

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

We call the resulting filter the parallel point process filter (P-PPF) shown in Figure 2-1: it consists of $J$ parallel branches of point process filters, each not only calculating the MMSE estimate of $\mathbf{x}_t$ assuming a duration of $T_j$, but also the corresponding likelihood $p(\mathbf{N}_{1:t}|\, T_j)$. This approach can also be viewed as mixture modeling, a common framework in statistical inference that is used to estimate a desired density in different applications. For example, mixture modeling combined with sequential state estimation in dynamical systems, when the system could have different regimes, has been used as early as in [51]. See also the mixture Kalman filtering work in [52]

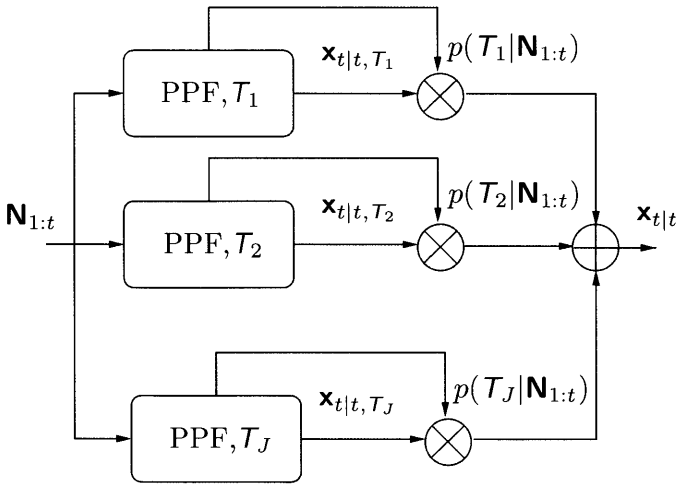


Figure 2-1: Parallel point process filter (P-PPF). P-PPF consists of $J$ parallel branches of point process filters, each not only calculating the MMSE estimate of $\mathbf{x}_t$ assuming a duration of $T_j$, but also the corresponding likelihood $p(\mathbf{N}_{1:t}|T_j)$.

and references therein. Mixture modeling has also been used in [53] to combine empirically fitted and time-invariant state models to multiple targets.

One question that arises at this point is how to estimate the state once $t > T_j$ for some branch $T_j$. One way to do so is to perform the summation over all $j$ for which $T_j > t$. This means that after a given arrival time, $T_j$, the corresponding branch of the decoder exits the computations and is no longer used for state estimation. This approach is justified since the prior for a given branch is only defined up to its corresponding arrival time. Using this implementation and as time passes by, fewer branches are active and hence the computational load is lower. Another way to address this issue is to keep a branch in the decoder even after its arrival time is reached by changing its prior to one which corresponds to a still condition. This still condition will have its position fixed at the last position reached to at the arrival time and hence a zero velocity (and other higher order derivatives of the position) after the arrival time. This means that all the branches will be active at all times until the decoder is terminated at the largest arrival time of the branches.

We examine the effect of the two alternative implementations in detail in Section 2.2. However, we can argue that the first approach is more appropriate in particular in a real-time implementation for the following main reasons. First, the purpose in most real-time applications is to reach a target not to stand still in a given position. Once the target is reached, the decoder will usually reset or start reaching to a different target. Second, the computational complexity of the first im-

plementation is lower as there are fewer branches active at any given time compared to the second implementation. This is of particular importance in real-time applications.

Finally, note that our approach can also be applied to address the arrival time uncertainty in cases where the observations are taken to be the neural firing rates instead of the spiking activity and a Gaussian likelihood model is assumed on these firing rates as in [45]. In this case the Gaussian approximation to the posterior in our P-PPF becomes exact and the update recursions in the filter become those of the well-known Kalman filter. The calculation of the arrival time likelihoods, however, remain the same.

## ■ 2.2 Results

Here we show the application of P-PPF to two scenarios. First in decoding goal-directed reaching movements where the duration is unknown but the start time is known. Second in decoding reaching movements in which the start time is also unknown. In each case we compare the decoder with RW-PPF in which no target information is incorporated and show that the decoder improves the RW-PPF performance considerably, where performance is measured as the average root mean-square (RMS) error in the estimated trajectory.

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

### ■ 2.2.1 Estimation of reaching movements of unknown duration from neural signals

Here, we implement the P-PPF and adapt it for estimation of reaching movements that start at a known reference position at rest, acquire a target position at an *unknown* time, and come to a stop at that time. For illustration purposes we adapt the goal-directed state-space model developed in [43] in our decoder, i.e., we use this state-space model in (2.4). However, any other goal-directed state-space model can be alternatively used in our decoder.

This state-space model is derived by conditioning a simple random-walk state-space model,

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{z}_t \tag{2.17}$$

on being at the target state $\mathbf{x}^*$ at a given time $T$ where $\mathbf{A}$ is the random-walk dynamics matrix and $\mathbf{z}_t$s are i.i.d. zero-mean Gaussian noise vectors with covariance matrix $\mathbf{V}$. In general in [43] one can also put a Gaussian distribution $\mathcal{N}(\mathbf{x}^*, \mathbf{Q})$ on the target state for uncertain targets. The resulting goal-directed state-space model is in the form of (2.4) where the expressions for $\mathbf{G}(t, T, \mathbf{x}^*)$, $\mathbf{b}(t, T, \mathbf{x}^*)$, and $\mathbf{W}_t$ in this case are given in [43] and in the Appendix in (2.18), (2.19), and (2.20). Note that the random-walk state model is also in the form of (2.4) but with $\mathbf{b} = \mathbf{0}$, and $\mathbf{G}(t, T, \mathbf{x}^*) = \mathbf{A}$ and $\mathbf{W}_t = \mathbf{V}$ time-invariant and of course independent of target and arrival time.

For a simple two-dimensional reaching movement, we take the state in each dimension to be $\mathbf{x}_t = [x_t, v_t]'$, where the components represent position and velocity respectively. The parameter values used to draw sample trajectories from this state-space model are given in Table 2.1. Here, $\mathbf{A}$ is the dynamic matrix for a simple kinematic model and as is apparent from $\mathbf{V}$, the state noise is taken to be non-zero only in the velocity dimension. We sampled 30 trajectories with movement durations in the range $550 - 1000$ ms.

Table 2.1: Parameters used to sample trajectories.

| $\mathbf{A}$ | $\mathbf{V}$ | $\mathbf{Q}$ | $\mathbf{x}^*$ |
|---|---|---|---|
| $\begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} 0\ \text{cm}^2 & 0 \\ 0 & 10\ (\text{cm/s})^2 \end{bmatrix}$ | $\begin{bmatrix} 0.01\ \text{cm}^2 & 0 \\ 0 & 1\ (\text{cm/s})^2 \end{bmatrix}$ | $\begin{bmatrix} 25\ \text{cm} \\ 0\ \text{cm/s} \end{bmatrix}$ |

To get the corresponding neural activity for each sampled goal-directed trajectory, we used a modified version of the model of single cell activity in the motor cortex [20,23], also known as the cosine tuning model of the conditional intensity function. Using this model and for each trajectory, we simulated the spiking activity of an ensemble of $C = 20$ neurons, which is a typical ensemble size in real-time BMIs. The spiking activity for each neuron in the ensemble was independently simulated as a point process whose instantaneous firing rate or conditional intensity function was a function of speed and movement direction. More specifically, we modeled the conditional

intensity function of a neuron during a two-dimensional movement by [50],

$$\begin{aligned}\lambda_c(t|\mathbf{x}_t) &= \exp\left(\alpha_0 + \alpha_1|\mathbf{v}_t|\cos(\theta_t - \theta_p^c)\right) \\ &= \exp\left(\alpha_0 + \alpha_y^c v_y(t) + \alpha_x^c v_x(t)\right)\end{aligned}$$

where $\theta_p^c$ is the preferred angle of the $c$th neuron, $\theta_t$ is the movement angle at $t$, $v_y(t)$ and $v_x(t)$ are the velocities in the $y$ and $x$ directions, $\alpha_0$ and $\alpha_1$ are parameters of the model, and consequently, $\alpha_x^c = \alpha_1\cos(\theta_p^c)$ and $\alpha_y^c = \alpha_1\sin(\theta_p^c)$. Here, the preferred direction for each neuron is sampled randomly from $[-\pi, \pi]$, $\alpha_0 = 1.6$ is chosen to have a background firing rate of 5 Hz for each neuron and $\alpha_1 = 0.014$ (s/cm) is chosen to have a maximum firing rate of 25 Hz on average (over the trajectories) for each neuron. We simulated 100 realizations (trials) of the point process neural signal for each trajectory using the time-rescaling theorem [54]. Note that the window of uncertainty for the arrival time is 550-1000 ms.

**Performance measure**

We define the performance measure of the decoder as its average RMS error over the trajectories. To find this measure, we first find the RMS error for each time step along a sample trajectory by averaging the square estimation error at that time step over the 100 simulated neural realizations. This gives us an RMS error for each time step along a sample trajectory. To quantify an average RMS error for the entire trajectory, we then average the RMS error over its duration. Finally, we find the mean of the average RMS errors over all 30 sample trajectories as the measure of performance for a decoder.

**Performance loss due to unknown duration**

We first examine the loss incurred due to the lack of a priori knowledge of the movement duration by the decoder and the effect of the number of discretization points on this loss. To do so, we compare the performance of the P-PPF with various number of discretization points to its performance assuming exact a priori knowledge of movement duration. With this a priori knowledge, the P-PPF for each trajectory will only have a single branch corresponding to the known duration. The average RMS error in this case is hence used as a baseline as it provides a lower bound on the P-PPF error without the duration knowledge. To find the effect of the number of discretization points on the error, we discretize the duration of the uncertainty interval $[550, 1000]$ ms in steps of 50 ms, 90 ms, 150 ms, and 450 ms, corresponding to 10, 6, 4, and 2 discretization points, re-

spectively, and find the average RMS error of P-PPF for each case. We also find the average RMS error using a single branch at 1000 ms, which corresponds to a single discretization point at the largest possible duration. In each case we assume a uniform prior distribution on the discretization points. To make the comparison to the baseline case with known duration, we compute the average RMS error of the P-PPF for each trajectory up to the end of its movement as opposed to up to the end of the uncertainty window.

Figure 2-2 shows the average RMS error using different number of discretization points in P-PPF, using both implementations, and also the baseline average RMS error assuming a known movement duration. Using the first implementation, we can see that by increasing the number of discretization points the average RMS error has a steep drop to an asymptote value. Defining the average RMS error achieved using 10 discretization points as the asymptote value, we find that using only 4 discretization points, corresponding to a 150 ms discretization interval, the difference between the average RMS error and the asymptote is only 1% of the asymptote value. This shows how a relatively coarse discretization is sufficient to get to this asymptote. Also, increasing the number of discretization points from 1 to 4 reduces the difference between the average RMS of the decoder and the baseline (i.e., with a priori knowledge of the duration) by over 53%.

Similar results hold for the second implementation. We can see that the two implementations have very close average RMS error for the duration of movement. For example using 4 discretization points, the difference between the average RMS error of the two implementations is less than 1% of the larger value.

**Comparison to a random-walk point process filter**

We now compare the performance of the P-PPF to a simple RW-PPF that does not incorporate any target information (see prior model in (2.17)). In our comparison, we use 4 branches for the P-PPF as this choice is sufficient to get within 1% of the asymptote performance. We also use the first implementation of the decoder as the two have very close average RMS error, but the first implementation is computationally more efficient.

Figure 2-3a-2-3d show one of the 30 trajectories with a duration of $T = 600$ ms and 10 sample decoded trajectories in two scenarios: 1. When $T$ is known exactly to the decoder. 2. When $T$ is unknown to the decoder. For both cases we show the performance of P-PPF using the first implementation and RW-PPF. Note that when the decoder knows $T$, it uses a single parallel branch designed for that $T$. We can see that P-PPF performs better in both cases; For a known duration,

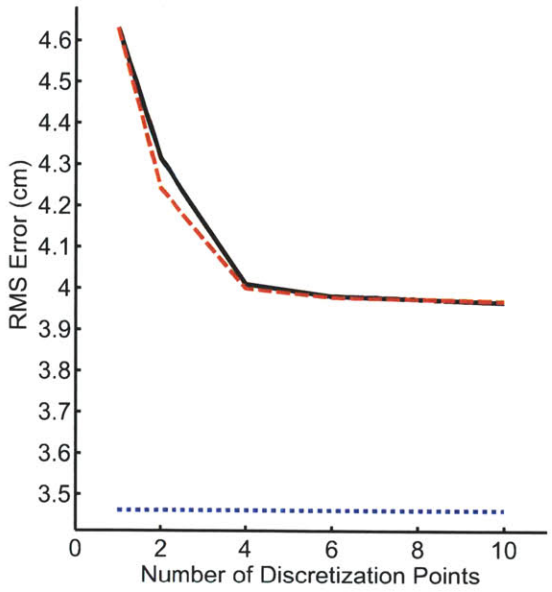


Figure 2-2: The effect of the number of duration discretization points on P-PPF average RMS error. Black curve shows the average RMS error during movement of the first implementation of P-PPF, dashed red curve shows that of the second implementation, and dotted blue line shows the lower bound on average RMS error corresponding to a known movement duration.

as expected, P-PPF performs better than RW-PPF since it places a better prior on the state. For an unknown duration, the P-PPF with only 4 parallel branches does better than RW-PPF for two reasons: 1. It places a better prior on the states. 2. It aims to detect the end of movement by combining the estimates of the parallel filters, each designed for a different arrival time, with time-varying optimal weights that are updated purely based on the neural observation.

Figure 2-3e-2-3h show the decoded velocities in the x and y directions for a single neural realization and the time evolution of the optimal combining weights in P-PPF for the given realization and also averaged over all 100 realization. Here the true arrival time is at $T = 600$ ms and as we can see the closest weight, i.e. the one corresponding to 550 ms, dominates up to 550 ms. For the remainder of the decoding interval, the closest remaining weight to $T$ dominates at each time. Also, comparing the decoded velocities for P-PPF and RW-PPF, we can see that P-PPF brings the estimated velocity close to zero at the end of movement as opposed to RW-PPF.

Table 2.2 shows the average RMS error in the estimate of the 30 sample trajectories during movement (i.e., averaging the RMS error for each trajectory until the end of movement as opposed to until the end of decoding time at 1000 ms). During movement, the average RMS error of RW-PPF is 67% larger than that of P-PPF with unknown arrival time. Also, the increase in P-PPF

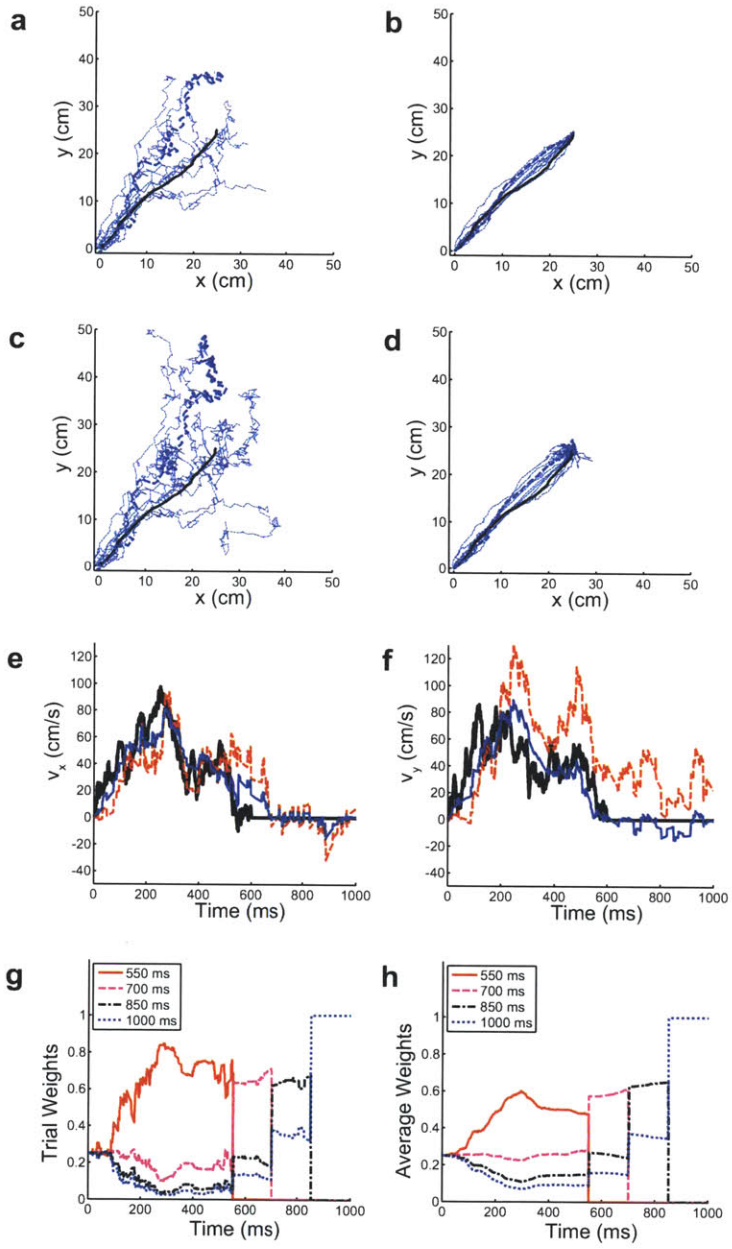


Figure 2-3: Comparison of RW-PPF and P-PPF estimated trajectories. (a-d) True trajectory (solid black) and 10 sample decoded ones (dashed) with known duration in (a) and (b) and unknown duration in (c) and (d). (a),(c) RW-PPF, (b),(d) P-PPF. (e-f) True velocities in the x and y directions (solid black) and sample decoded velocities for a single neural realization with P-PPF (solid blue) and RW-PPF (dashed red). (g-h) Combining weights vs. time for a single neural realization (g) and averaged over all 100 realizations (h). True duration is 600 ms.

average RMS error when $T$ is unknown compared to when it is known is small even though the discretization is very coarse. This is specially the case when considering the average RMS error

Table 2.2: Average RMS error (cm) in decoded trajectory of 30 sample reaching movements calculated over the duration of movements.

| RW-PPF | P-PPF (known $T$) | P-PPF (1) | P-PPF (2) |
|---|---|---|---|
| 6.69 | 3.46 | 4.01 | 4.00 |

Table 2.3: Average RMS error (cm) in decoded trajectory of 30 sample reaching movements calculated until the end of the uncertainty window or equivalently decoding interval.

| RW-PPF | P-PPF (1) | P-PPF (2) |
|---|---|---|
| 7.94 | 3.44 | 3.32 |

of the RW-PPF. Moreover, the two implementations of the P-PPF have very similar average RMS errors during movement.

To compare the P-PPF and RW-PPF over the entire decoding period, i.e., until the end of the uncertainty interval at 1000 ms, we found their average RMS error for this interval. Table 2.3 shows these results. Here RW-PPF has an average RMS error that is 2.3 times that of P-PPF. RW-PPF does worse when considering its RMS until the end of the decoding interval as it cannot detect the end of movement and hence its error accumulates. P-PPF, however, can detect the end of movement and hence bring the velocity close to zero after the movement ends. Note that when considering the performance until the end of the decoding interval, the second implementation of the P-PPF gains over the first implementation. This is because the second implementation keeps the branches in the filter even after their arrival time by putting the still prior model on them. It can hence bring the velocity closer to zero after the end of movement. Therefore in applications where the error after the end of movement and hence standing still at a position is important, for example when reaching a target and then holding it, this implementation could result in additional reduction of error. However, note that the first implementation is computationally more efficient as it does not keep all the branches in its calculations. Hence the first implementation is more appropriate in real-time applications in which the main purpose is to reach a target.

Figure 2-5a-2-5f in the Appendix show the estimated trajectories, velocities, and the average mixture weights for the second implementation of the P-PPF compared to the first implementation.

### ■ 2.2.2 Estimation of reaching movements of unknown duration and start time from neural signals

So far we looked at goal-directed movements with unknown arrival time at target but with a known start time. For instructed movements this could be a reasonable assumption as the start time will be around the instruction time. However, in some applications, the start time of a movement may not be known with enough precision relative to the range of movement duration.

One approach to this problem is to design a BMI that attempts to classify the neural activity as belonging to different epochs or states of a reaching task, e.g., baseline and execution [55]. For example a hidden Markov model was used in [55] to find the posterior probability of an epoch given the neural signal. The start of an epoch was then found by thresholding on its posterior probability. It was shown, however, that there is a trade-off between the correct classification rate and latency involved in detecting an epoch once it has started. More specifically, having a higher threshold increased the correct classification rate but also the latency. Hence in cases where the movement is short (for example in BMI applications interacting with a computer cursor), this latency could degrade the performance of the kinematics decoder since by the time the execution epoch is detected, the decoder will miss part of the relevant neural activity.

In such cases, an alternative approach is to add a single parallel branch to our P-PPF corresponding to the case of baseline or no movement, i.e., to the case with,

$$\mathbf{x}_t = \mathbf{0}, \qquad \forall t$$

We call this branch of the P-PPF the 'idle' branch. Having this branch, the decoder also allows for the possibility that the movement has not started yet and calculates its likelihood. The major difference between this approach and that in [55] is that here the decoder combines all the branches weighting them with their respective likelihood at all time and hence does not wait until a threshold is met on the execution epoch posterior to start the kinematics decoding. In other words, the decision on the start time here is a soft one as opposed to thresholding that is a hard decision and the neural signal at no time is discarded in the kinematics decoder.

To test the algorithm on this case, we used the same 30 trajectories as those in the previous section but zero-padded them in the beginning by 500 ms. Hence the true start time for these movements is now at 500 ms. For each trajectory, we then simulated 100 trials of the corresponding ensemble neural spiking activity as in the previous section . We again used the P-PPF to

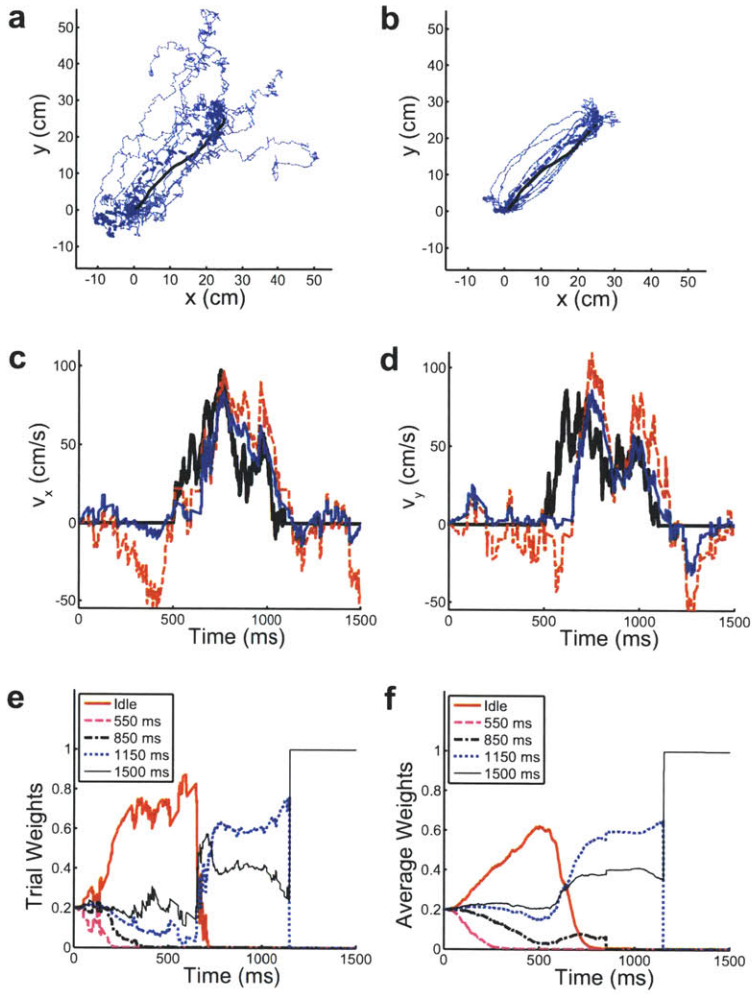


Figure 2-4: Comparison of RW-PPF and P-PPF estimated trajectories in the case of unknown start and arrival times. (a-b) True trajectory (solid black) and 10 sample decoded ones (dashed) using RW-PPF in (a) and P-PPF in (b). (c-d) True velocities in the x and y directions (solid black) and sample decoded velocities for a single neural realization with P-PPF (solid blue) and RW-PPF (dashed red). (e-f) Combining weights vs. time for a single neural realization (e) and averaged over all 100 realizations (f). True duration is 600 ms and true start time is 500 ms.

decode the trajectory, this time with 5 parallel branches corresponding to movement durations of $\{550, 850, 1150, 1500\}$ ms and an idle branch. Note again that time is discretized very coarsely. We also repeated the decoding using the RW-PPF.

Figure 2-4a-2-4b show one of the 30 trajectories with a duration of $T = 600$ ms and 10 sample decoded trajectories of RW-PPF and P-PPF with the first implementation. We can clearly see that P-PPF performs better than RW-PPF. Table 2.4 shows the RMS error in the decoded trajectory over the 30 sample reaching movements calculated until the end of movement and also until the end

Table 2.4: Average RMS error (cm) in decoded trajectory of 30 sample reaching movements with unknown duration and start time averaged until the end of movement for each trajectory and also until the end of the uncertainty window for all trajectories .

| | RW-PPF | P-PPF (1) | P-PPF (2) |
|---|---|---|---|
| Until end of Movement | 8.89 | 5.35 | 5.30 |
| Until end of Uncertainty Window | 9.93 | 5.04 | 4.86 |

of the decoding interval or uncertainty window at 1500 ms. The average RMS error of RW-PPF is 66% higher than that of the P-PPF until end of movement. If we also consider the error after the end of movement, the average RMS error of RW-PPF is 97% higher than that of P-PPF. Figure 2-4c-2-4f show the decoded velocities in the x and y directions for a sample neural realization and the time evolution of the optimal combining weights in P-PPF for the sample realization and also averaged over all 100 realizations. Here the true start time is at $T = 500$ ms and as we can see the idle component dominates in this range. After the movement starts, the idle weight quickly drops down and the remaining parallel branches of the filter dominate. Note also that the decoder quickly realizes that the movement does not end close to the first 2 weights, i.e., 550 ms and 850 ms and hence these weights are close to 0 very early on. Looking at Figure 2-4c-2-4d we can see that the P-PPF compared to RW-PPF keeps the velocity close to zero before movement initiation and also after movement has ended, since it can detect the end and start of movements as opposed to RW-PPF. This is the reason why the average RMS error in RW-PPF increases once considering the error after the end of movement.

It is important to note that in the case of unknown start time, this method works well when the window of uncertainty for the start time is on the order of possible movement durations. This is because of the way the prior model is built, i.e., the parallel branches other than the idle branch assume movement starts at the start of decoding. Hence this prior is a good approximation to the true one in the above case. This situation arises in short instructed tasks, for example those interacting with a computer cursor, in which the reaction time is unknown but will be within a short window of uncertainty, comparable to the movement durations. Note also that we picked the initial weight of the idle branch at time $t = 0$ equal to the weights of the other branches. This choice is appropriate when the start time is known to be comparable to the duration of the movement. However, depending on the specific task and prior information available about the uncertainty in start time, the decoder can pick a different initial weight for the idle branch.

## 2.3 Discussion

We have developed a recursive Bayesian decoder for goal-directed movements from ensemble neural spiking observations that can take advantage of *any* goal-directed state-space model without requiring prior information about the movement duration. To achieve this, the decoder calculates not only causal estimates of kinematic states at each time based on the ensemble spiking observations, but also the likelihood of the arrival time based on these observations. Hence our decoder can be implemented for real-time decoding of goal-directed movements in BMIs.

To make the decoder computationally tractable, we discretize the arrival time. Doing so our decoder reduces to a parallel point process filter, P-PPF, in which each parallel branch calculates the estimate of the current state assuming an arrival time corresponding to a discretization point and also the likelihood of that arrival time. The overall estimate is then generated by combining all the branch estimates with time-varying weights found from their corresponding likelihoods at each time. The observation model in our filter is the point process model of the ensemble spiking activity. Hence to reduce complexity, a Gaussian approximation is made to the posterior density of the kinematic states. This approximation makes calculating the partition function at each time step of the decoder tractable. The likelihood of the arrival time at each time is then found by appropriately combining the partition functions from all the previous time steps. Due to the parallel nature of the filter, the overall run-time is on the order of the run-time for a single branch, a property crucial for real-time implementation.

We examined the effect of the discretization refinement of the duration on the decoder average RMS error and showed that using only a coarse discretization is sufficient to achieve an asymptote average RMS error. This means that the added computational complexity involved to address the uncertainty in the arrival time is not large. As a baseline, we also found the average RMS error of P-PPF when the arrival time was assumed known. We observed that the increase in average RMS error using the P-PPF with a coarse discretization was relatively small compared to the baseline.

We further examined the effect of two possible filter implementations for P-PPF. In the first implementation the branches of the decoder exit the computation once their corresponding arrival time is reached. In the second implementation we keep all the branches in the decoder even after their arrival time is reached. To do so, we change the prior model of a branch after its arrival time is reached to a still model which assumes that the state has stopped at the location reached by the branch at its arrival time. We showed that the two implementations have very similar average

RMS error during movement. However, the second implementation is more successful in bringing the velocity close to zero after the movement ends. Hence in applications where standing still at a target position is important, for example for holding the target, it could have an advantage over the first implementation. However, in many real-time BMIs the purpose is mainly to reach a target. Hence the first implementation is more appropriate for such applications as it is computationally more efficient.

We also examined the advantages of using the P-PPF over a simple RW-PPF. We showed that using a very coarse discretization (and hence few branches in the P-PPF) the performance of our decoder in terms of RMS error is less than half that of RW-PPF. The reason for this improved performance is the fact that the P-PPF not only uses a better prior model on the kinematic states, but also correctly identifies the end of movement and hence brings the velocity close to zero at that point in contrast to RW-PPF which is not aware of movement termination.

We also demonstrated another application of P-PPF, namely to decode goal-directed movements in which the start time of the movement is also unknown to the decoder. We can address this case by adding a single branch to the P-PPF corresponding to the case where the movement has not been initiated, i.e., the 'idle' case. This way the P-PPF also calculates the likelihood that no movement has started at each time and hence keeps the kinematic states close to zero before movement initiation.

Prior work have indicated the advantage of using goal-directed state-space models for estimation of movements [43, 44] when movement duration is assumed to be known. The assumption of known movement duration, however, does not allow these models to be incorporated in a real-time BMI. Our framework can be used to incorporate any goal-directed state-space model in a real-time BMI to improve its accuracy in estimating goal-directed movements, especially when prior information about the target is available from the neural activity before movement initiation [18, 19]. This will in turn enable the BMIs to combine both the target and kinematic related activities in real-time. Combining information encoded about both the target and trajectory of movement can thus help refine the estimation accuracy for natural movements. We show the incorporation of this framework in a novel real-time BMI in Chapter 4.

## 2.4 Appendix

### 2.4.1 Goal-directed state-space model

The goal-directed state-space model of duration $T$ in [43] is given by:

$$\mathbf{G}(t, T, \mathbf{x}^*) = [\mathbf{I} - \mathbf{V}\boldsymbol{\Pi}^{-1}(t, T)]\mathbf{A} \tag{2.18}$$

$$\mathbf{b}(t, T, \mathbf{x}^*) = \mathbf{V}\boldsymbol{\Pi}^{-1}(t, T)\mathbf{A}^{t-T}\mathbf{x}^* \tag{2.19}$$

$$\mathbf{W}_t = \mathbf{V} - \mathbf{V}\boldsymbol{\Pi}^{-1}(t, T)\mathbf{V}' \tag{2.20}$$

where

$$\boldsymbol{\Pi}(t-1, T) = \mathbf{A}^{-1}\boldsymbol{\Pi}(t, T)\mathbf{A}^{-1'} + \mathbf{V} \quad [1]$$

$$\boldsymbol{\Pi}(T, T) = \mathbf{Q} + \mathbf{V}$$

### 2.4.2 Comparison of the two alternative implementations

Figure 2-5a-2-5f show the estimated trajectories, velocities, and the average mixture weights for the second implementation of the P-PPF compared to the first implementation. We can see that the second implementation brings the velocity closer to zero after the end of movement.

[1] We have corrected this expression from the original in [43].

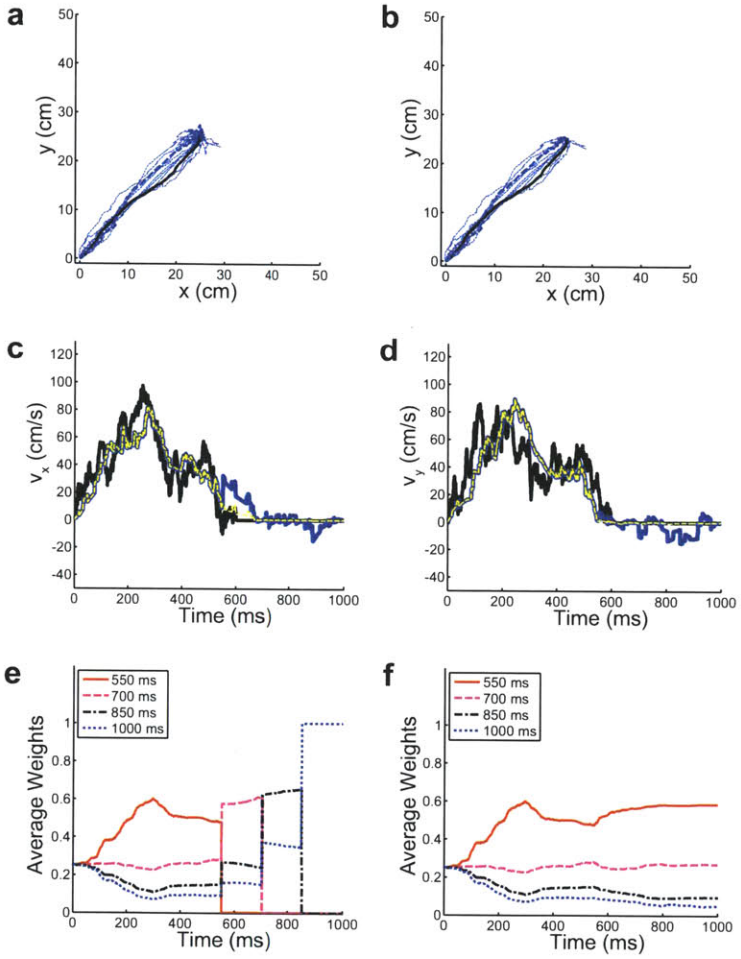


Figure 2-5: Comparison of P-PPF estimated trajectories using the two implementations. (a-b) True trajectory (solid black) and 10 sample decoded ones (dashed) with unknown duration using P-PPF with the first implementation in (a) and the second implementation in (b). (c-d) True velocities in the x and y directions (solid black) and P-PPF sample decoded velocities for a single neural realization with the first implementation (solid blue) and the second implementation (dashed yellow). (e-f) Combining weights vs. time averaged over all 100 realizations for P-PPF with the first implementation (e) and the second implementation (f). True duration is 600 ms.

# Chapter 3
# An optimal feedback-controlled goal-directed state-space model

In Chapter 2 we developed a parallel point process filter framework that enabled the use of any goal-directed state-space model for real-time decoding of goal-directed movements from neural signals. In this chapter, we propose one such goal-directed state-space model as the prior in our parallel point process filter. We use an optimal feedback control design to build this prior goal-directed state-space model for the kinematics. This design is inspired by the optimal feedback control theory of the sensorimotor system used to explain its function [41,42]. In this framework, each task is performed to accomplish a goal during which there is sensory feedback about the state of the musculo-skeletal system. Based on the intended goal, the sensory feedback (for example vision and proprioception) about the current state of the system, and the desired time to accomplish the goal, the subject (controller) decides on the next plan of action or control signal (for example muscle activation) and can hence make real-time adjustments based on the feedback to improve behavior. This framework is illustrated in Figure 3-1. Specifying an approximate kinematic model and quantifying the task goals as cost functions and also the sensory feedback, this framework can predict the next plan of action, and consequently the next kinematic state, in the presence of model uncertainty and sensory noise. Using this framework we develop a prior feedback-controlled state-space model based on which we can predict the subject's next plan of action or control signal in a kinematic decoder and consequently the next kinematic state. We illustrate the performance of this prior model in the parallel point process filter and for estimation of real reaching movements performed by a rhesus monkey and show that compared to a random-walk point process filter it significantly reduces the average RMS error.

## ■ 3.1 Optimal feedback-controlled prior state-space model

We now develop the feedback-controlled prior state-space model. Denoting the sequence of kinematic states by $\mathbf{x}_0, \cdots, \mathbf{x}_t$, we assume they are generated according to the linear dynamical system,

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t. \tag{3.1}$$

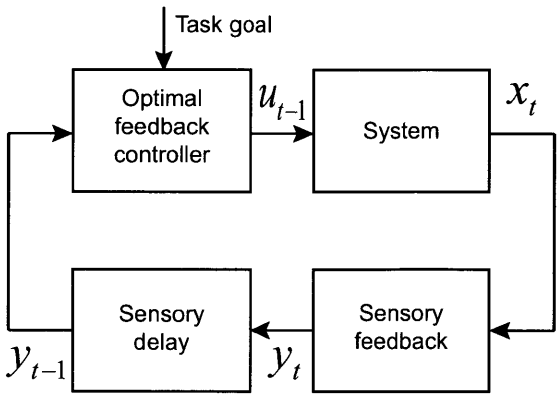


Figure 3-1: Optimal feedback control framework. An optimal feedback control framework is used to build a prior state-space model for goal-directed movements. In this framework each task is performed to accomplish a goal during which there is real-time sensory feedback, $\mathbf{y}_t$, about the state of the system, $\mathbf{x}_t$. Based on the intended goal and the sensory feedback about the state of the system, the brain (controller) decides on a plan of action or control signal $u_t$ and controls the system.

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

$$J = \sum_{t=1}^{T-1} \left(\mathbf{x}_t'\mathbf{Q}_t\mathbf{x}_t + \mathbf{u}_t'\mathbf{R}\mathbf{u}_t\right) + \mathbf{x}_T'\mathbf{Q}_T\mathbf{x}_T, \tag{3.2}$$

where $T$ is the movement duration, $\mathbf{Q}_t$ is positive semidefinite and $\mathbf{R}$ is positive definite, then the optimal control rule at any time, $\mathbf{u}_t$, is simply a linear feedback of the state at that time [56], i.e.,

$$\mathbf{u}_t = -\mathbf{L}_t(T)\mathbf{x}_t, \tag{3.3}$$

where $\mathbf{L}_t$ can be found recursively and offline given by [56]

$$\mathbf{L}_t = (\mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{P}_{t+1}\mathbf{A} \tag{3.4}$$

where $\mathbf{P}_t$ is found recursively and backwards in time using

$$\mathbf{P}_t = \mathbf{Q}_t + \mathbf{A}' \left( \mathbf{P}_{t+1} - \mathbf{P}_{t+1}\mathbf{B} \left( \mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B} \right)^{-1} \mathbf{B}'\mathbf{P}_{t+1} \right) \mathbf{A} \tag{3.5}$$

with the initial condition

$$\mathbf{P}_T = \mathbf{Q}_T. \tag{3.6}$$

This is the linear quadratic Gaussian (LQG) solution. Substituting (3.3) in (3.1) reduces this state-space model to the optimal feedback-controlled state-space model

$$\mathbf{x}_{t+1} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\mathbf{x}_t + \mathbf{w}_t, \tag{3.7}$$

which can now be used to make prediction on the kinematic states. Note that $\mathbf{Q}_t$ and $\mathbf{R}$ should be appropriately designed for an application of interest and $\mathbf{L}_t(T)$ is time-varying and a function of the duration $T$.

### ■ 3.1.1 Optimal feedback-controlled prior model for a reaching movement

One of the most common goal-directed movements is a reaching movement. We can now specialize the feedback-controlled prior model to the case of reaching movements, which are used in most real-time BMI experiments. For a reaching movement, the goal is to reach a target position at a desired arrival time and stop there, while being energy efficient. Hence the cost function in this case should enforce positional accuracy at the arrival time, stopping condition, and energetic efficiency [57, 58]. Denoting the desired final target position in the two dimensions by $\mathbf{d}^* = [d_1^*, d_2^*]'$, and the position, velocity, and force in the two dimensions with $\mathbf{d}_t = [d_1(t), d_2(t)]$, $\mathbf{v}_t = [v_1(t), v_2(t)]$, and $\mathbf{a}_t = [a_1(t), a_2(t)]$, respectively, similar to previous studies [57, 58] we take this cost function to be the weighted sum

$$J = \| \mathbf{d}_T - \mathbf{d}^* \|^2 + w_v \| \mathbf{v}_T \|^2 + w_a \| \mathbf{a}_T \|^2 + w_r \sum_{t=1}^{T-1} \| \mathbf{u}_t \|^2 \tag{3.8}$$

where the weights are chosen to penalize the terms in the cost function approximately equally on average [57, 58]. Taking the state to be $\mathbf{x}_t = [d_1(t), v_1(t), a_1(t), d_2(t), v_2(t), a_2(t)]'$, we adapt the following first order lowpass muscle-like system [58] for the dynamical system in (3.1) in each

dimension $i$

$$\begin{bmatrix} d_i(t+1) \\ v_i(t+1) \\ a_i(t+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta & 0 \\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} \\ 0 & 0 & 1-\frac{\Delta}{\tau} \end{bmatrix} \begin{bmatrix} d_i(t) \\ v_i(t) \\ a_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta}{\tau} \end{bmatrix} u_i(t) + \begin{bmatrix} 0 \\ 0 \\ w_i(t) \end{bmatrix} \qquad i = 1,2 \tag{3.9}$$

where $\Delta$ is the time increment, the parameters $b = 10$ Ns/m, $\tau = 0.05$ s, and $m = 1$ kg come from biomechanics [58] and the noise variance can be fitted to the behavioral trajectories using maximum likelihood parameter estimation.

Having specified the kinematic model in (3.9) and the cost function in (3.8) for the reaching movements, the feedback matrices $\mathbf{L}_t(T)$ can be computed from the recursive solution of LQG in (3.4). To do so, we augment the state to include the target position in the two dimensions [57,58], $\mathbf{d}^* = [d_1^*, d_2^*]'$, i.e., $\mathbf{x}_{\text{aug}}(t) = [d_1(t), v_1(t), a_1(t), d_1^*, d_2(t), v_2(t), a_2(t), d_2^*]'$. Hence the augmented state-space model in each dimension takes the form

$$\begin{bmatrix} d_i(t+1) \\ v_i(t+1) \\ a_i(t+1) \\ d_i^* \end{bmatrix} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} & 0 \\ 0 & 0 & 1-\frac{\Delta}{\tau} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_i(t) \\ v_i(t) \\ a_i(t) \\ d_i^* \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta}{\tau} \\ 0 \end{bmatrix} u_i(t) + \begin{bmatrix} 0 \\ 0 \\ w_i(t) \\ 0 \end{bmatrix} \qquad i = 1,2 \tag{3.10}$$

Now comparing (3.8) with (3.2) and denoting the two dimensional identity matrix with $\mathbf{I}$, we find $\mathbf{R} = w_r\mathbf{I}$, and $\mathbf{Q}_t = 0$ for $t < T$. To find $\mathbf{Q}_T$ note that $d_1(T) - d_1^* = p_1'\mathbf{x}_{\text{aug}}(T)$ with $p_1 = [1,0,0,-1,0,0,0,0]'$ and similarly for $p_2$, and that $v_1(T) = p_{v_1}'\mathbf{x}_{\text{aug}}(T)$ with $p_{v_1} = [0,1,0,0,\cdots,0]'$ and similarly for $p_{v_2}$, $p_{a_1}$, and $p_{a_2}$. Therefore we find $\mathbf{Q}_T = p_1p_1' + p_2p_2' + w_v(p_{v_1}p_{v_1}' + p_{v_2}p_{v_2}') + w_a(p_{a_1}p_{a_1}' + p_{a_2}p_{a_2}')$. Now using (3.4) with these choices we can obtain the feedback matrices, $\mathbf{L}_t(T)$, and consequently the feedback-controlled state-space model for the reaching movement from (3.7). Note that these feedback matrices can be precomputed offline and then stored for later real-time use in a BMI.

Having these matrices, we can predict the subject's next plan of action reflected in the control term, $\mathbf{u}_t$, using (3.3) and assuming perfect sensory feedback about the current state, $\mathbf{x}_t$.

### ■ 3.1.2 Sampled trajectories

Figure 3-2a-3-2d show 10 real reaching trajectories to a given target location produced by a rhesus monkey using a joystick (see Section 3.3) and 10 sampled trajectories from the feedback-controlled

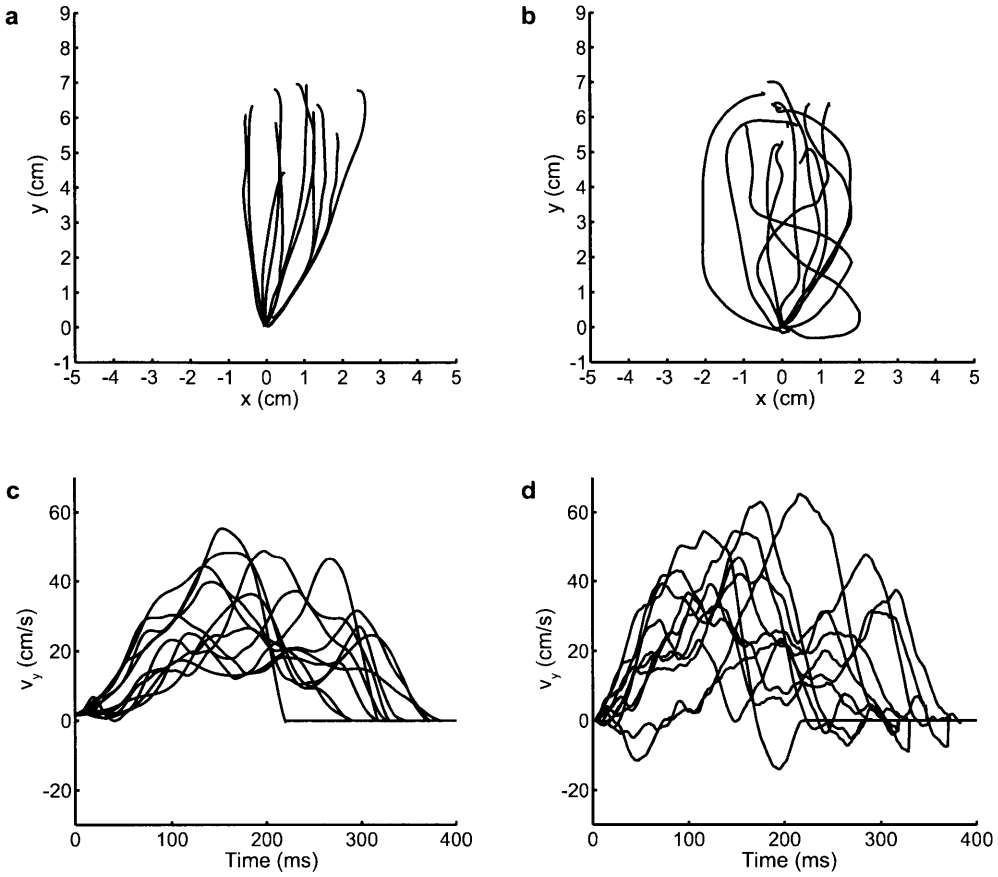


Figure 3-2: Sample reaching movements generated by the feedback-controlled state-space model. (a) and (c) show 10 sample trajectories of real reaching movements performed by a rhesus monkey to a given target location and the corresponding velocity profiles in the y direction. (b) and (d) show 10 sample trajectories generated by the feedback-controlled state-space model whose noise parameter is fitted to the monkey's trajectories.

prior state-space model. Here, we have fitted the state noise covariance $\mathbf{W}$ in (3.9) to the real reaching movements using maximum-likelihood parameter estimation. This noise covariance represents the variability in reaching to the same goal or target over different trials. We can see that the feedback-controlled state-space model samples trajectories that look similar to natural reaching movements. We will quantify the performance of this state-space model for estimation of such reaching movements in Section 3.3.

## ■ 3.2 Feedback-controlled parallel point process filter (FC-P-PPF)

Having derived the feedback-controlled prior model, we can now combine it with the point process observation model given in Chapter 2 in (2.1) to derive the feedback-controlled parallel point

process filter (FC-P-PPF) for estimation of reaching movements. Note that the feedback-controlled state-space model in (3.7), like any goal-directed state-space model, is dependent on the movement duration, $T$, not known a priori to the decoder. We hence use the same framework as in Chapter 2 to resolve this duration uncertainty. The feedback-controlled prior model in (3.7) for the augmented state is again in the form of (2.4) with $\mathbf{G}(t, T, \mathbf{x}^*) = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))$ and $\mathbf{b}(t, T, \mathbf{x}^*) = 0$. Hence the recursions of FC-P-PPF are simply given by (2.10)-(2.13) with these substitutions. Here we use the first implementation of the framework developed in Chapter 2 as it is computationally more efficient.

## ■ 3.3 Results

We measure the performance of the optimal feedback-controlled prior model in estimation of reaching movements. To do so, we compare the average RMS error of the FC-P-PPF in estimation of real reaching trajectories produced by a rhesus monkey to that of the random-walk point process filter (RW-PPF). For RW-PPF we use (3.1) as the prior model with $\mathbf{B} = \mathbf{0}$ and $\mathbf{A}$ as given in (3.9). Our data consists of 55 reaching trajectories performed by a rhesus monkey, based on which neural spiking activity is simulated (as will be explained). Using a joystick, the monkey moved a cursor from the center to one of four possible radial locations at the top, bottom, left, or right sides of a square computer screen 14 cm long in each dimension. Sample reaching trajectories and velocities to the top of the screen are shown in Figures 3-2a and 3-2c. The 55 trajectories had different durations in the range of 140-400 ms. Hence, here, the window of uncertainty for the arrival time is 140-400 ms. The only parameter of the state-space model in (3.9) fitted to reaching trajectories using maximum-likelihood parameter estimation was the state noise covariance $\mathbf{W}$. This noise covariance represents the variability in reaching to the same goal or target over different trials. Similar to the analysis in Chapter 2, we simulated the point process neural signal of $C = 20$ neurons for a given trajectory using the time-rescaling theorem and taking the cosine tuning model of the conditional intensity function for a two-dimensional movement [50],

$$\begin{aligned} \lambda_c(t|\mathbf{x}_t) &= \exp\Big(\alpha_0 + \alpha_1 |\mathbf{v}_t| \cos(\theta_t - \theta_p^c)\Big) \\ &= \exp\Big(\alpha_0 + \alpha_y^c v_2(t) + \alpha_x^c v_1(t)\Big) \end{aligned}$$

where $\theta_p^c$ is the preferred angle of the $c$th neuron, $\theta_t$ is the movement angle at $t$, $v_2(t)$ and $v_1(t)$ are the velocities in the $y$ and $x$ directions, $\alpha_0$ and $\alpha_1$ are parameters of the model, and consequently,

Table 3.1: Average RMS error (cm) during movement in decoded trajectories of 55 real reaching movements performed by a rhesus monkey. Here FC-P-PPF is assumed to know the movement duration.

| RW-PPF | FC-P-PPF (known $T$) |
|---|---|
| 1.40 | 0.87 |

$\alpha_x^c = \alpha_1 \cos(\theta_p^c)$ and $\alpha_y^c = \alpha_1 \sin(\theta_p^c)$. Here, the preferred direction for each neuron is sampled randomly from $[-\pi, \pi]$, $\alpha_0 = 1.6$ is chosen to have a background firing rate of 5 Hz for each neuron and $\alpha_1$ is chosen to have a maximum firing rate of 25 Hz on average (over the trajectories) for each neuron. We simulated 100 realizations (trials) of the point process neural signal for each of the 55 trajectories using the time-rescaling theorem [54].

To dissociate the effect of using the feedback-controlled prior model from that of using the parallel filters to resolve the duration uncertainty, we first compare the average RMS error of FC-P-PPF when the movement duration for each trajectory is assumed to be known to that of RW-PPF. This comparison illustrates the advantage of using a feedback-controlled prior model over a random-walk one. Note that when duration is known, FC-P-PPF uses a single branch corresponding to the true duration for each trajectory. The average RMS error of both decoders, which is calculated for each trajectory until the end of movement, is given in Table 3.1. The average RMS error of the RW-PPF is 61% higher than that of FC-P-PPF with known movement duration. This shows the benefit of using a more accurate prior model in the decoder.

Figure 3-3a and 3-3b show a sample reaching trajectory performed by the monkey with a duration of $T = 296$ ms and 10 sample decoded trajectories from RW-PPF and FC-P-PPF when $T$ is known exactly to the decoder. Figure 3-3c and 3-3d show the estimated velocity in the x and y direction for a sample trial using RW-PPF and FC-P-PPF. We can visually observe that using the feedback-controlled prior model improves the estimation accuracy.

We next consider the case where movement duration is not known a priori to the decoder, which is the case in real-time applications. We discretize the uncertainty interval starting at 150 ms, i.e. the [150, 400] ms interval, in steps of 25 ms, 50 ms, and 125 ms corresponding to 11, 6, and 3 parallel branches of the filter, respectively. We also find the performance of FC-P-PPF using a single branch at the longest arrival time at 400 ms. The average RMS error during movement as a function of the number of discretization points is shown in Figure 3-4 where we have also shown the average RMS error assuming that the decoder knows the exact arrival time for each

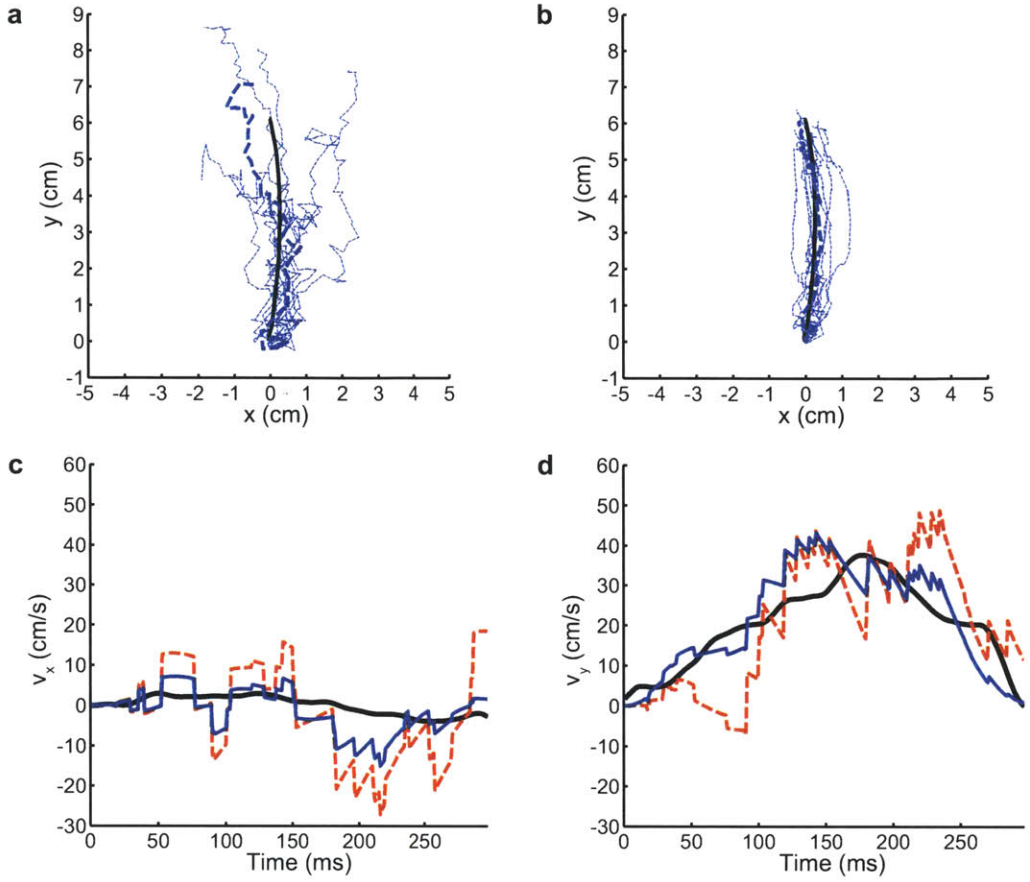


Figure 3-3: Comparison of the decoded trajectories using FC-P-PPF and RW-PPF when movement duration is known. (a) and (b) show a sample trajectory during a reaching movement performed by a rhesus monkey (solid black) and 10 sample decoded trajectories using RW-PPF in (a) and FC-P-PPF in (b) (dashed blue). (c) and (d) show the true velocity (black) in the x and y directions and the decoded velocities using RW-PPF (dashed red) and FC-P-PPF (solid blue) for a sample realization of the neural signal.

trajectory. Note that the average RMS error for known movement duration provides a lower bound on the average RMS error of FC-P-PPF. Similar to the results in Chapter 2, the average RMS error has a steep fall and reaches an asymptote average RMS value. More specifically, using only 3 discretization points corresponding to a discretization step of 125 ms, the difference between the average RMS error of FC-P-PPF and the asymptote value is less than 2% of the latter. Also, increasing the number of discretization points from 1 to 3 reduces the difference between the average RMS error of FC-P-PPF and the baseline (i.e., with a priori knowledge of the duration) by over 43%.

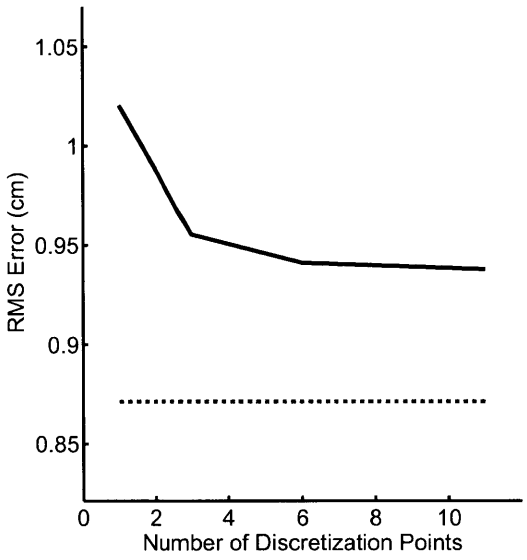


Figure 3-4: The effect of the number of duration discretization points on FC-P-PPF average RMS error. The black curve shows the average RMS error of FC-P-PPF as a function of the number of discretization points used. The dashed blue line shows the baseline average RMS error when FC-P-PPF has exact knowledge of the duration for each trajectory. This provides a lower bound on the average RMS error achievable by FC-P-PPF.

In the case where movement duration is not known, we therefore compare the performance of FC-P-PPF with 3 discretization points with that of RW-PPF . Table 3.2 shows the average RMS error of the two decoders during movement and also until the end of the uncertainty interval at 400 ms. We can see that RW-PPF has higher average RMS errors in both cases, which are higher than those of FC-P-PPF by 46% and 66%, respectively. Here FC-P-PPF does better than RW-PPF for two reasons: First, it puts a better prior model on the reaching movements. Second, it detects the end of movement by finding the likelihood of the discretized arrival times and reflecting that into its state estimate. This is done by combining the estimates of the parallel filters, each designed for a different arrival time, with time-varying optimal weights that are updated purely based on the neural observation. RW-PPF on the other hand cannot detect the movement termination. That is why when considering its performance until the end of the uncertainty window, it does worse.

Figure 3-5a and 3-5b show a sample trajectory with a duration of $T = 296$ ms (the same as in Figure 3-3) and 10 sample decoded trajectories using RW-PPF and FC-P-PPF when $T$ is unknown to the decoders and hence they decode until the end of the uncertainty window. We can see that FC-P-PPF generates more accurate trajectory estimates. Figure 3-5c-3-5f show the decoded

Table 3.2: RMS error (cm) in decoded trajectory of the 55 real reaching movements with unknown duration calculated until end of movement and until the end of the uncertainty window.

| | RW-PPF | FC-P-PPF |
|---|---|---|
| Until end of Movement | 1.40 | 0.96 |
| Until end of Uncertainty Window | 1.69 | 1.02 |

velocities in the x and y direction for a single realization of the neural signal and the time evolution of the optimal combining weights in FC-P-PPF for the given realization and also averaged over all 100 realizations. Here the true arrival time is at $T = 296$ ms and as we can see the closest weight, i.e., the one corresponding to 275 ms soon dominates up to its arrival time. Also, comparing the decoded velocities for FC-P-PPF and RW-PPF we can see that FC-P-PPF brings the estimated velocity close to zero at the end of movement as opposed to RW-PPF.

## ■ 3.4 Discussion

We have proposed a feedback-controlled prior state-space model for estimation of goal-directed movements and showed how to combine it with the neural spiking observations in a recursive Bayesian decoder. This state-space model is motivated by the optimal feedback control theory of the sensorimotor system used to explain its function [41, 42]. In this framework, each task is performed to accomplish a goal during which there is sensory feedback about the external state of the system. Specifying an approximate kinematic model, modeling the real-time sensory feedback about the state of the system, and quantifying the task goals as cost functions and the desired time to accomplish them, we can predict the next plan of action by finding the one that minimizes the cost function. Using this framework we derived the optimal feedback-controlled state-space model. However, similar to any other goal-directed state-space model, this model is dependent on the movement duration not known a priori to a decoder observing only the neural signal. We hence combined the feedback-controlled prior model with the framework developed in Chapter 2 that resolves the duration uncertainty inherent to any goal-directed state-space model using the neural activity. This resulted in FC-P-PPF for estimation of goal-directed movements.

We illustrated the performance of FC-P-PPF in estimation of reaching movements performed by a rhesus monkey and compared it to RW-PPF using simulated neural spiking activity. We showed that when movement duration is known, using a random-walk prior model instead of the feedback-controlled model increases the average RMS error by 61%. This shows the advantage of

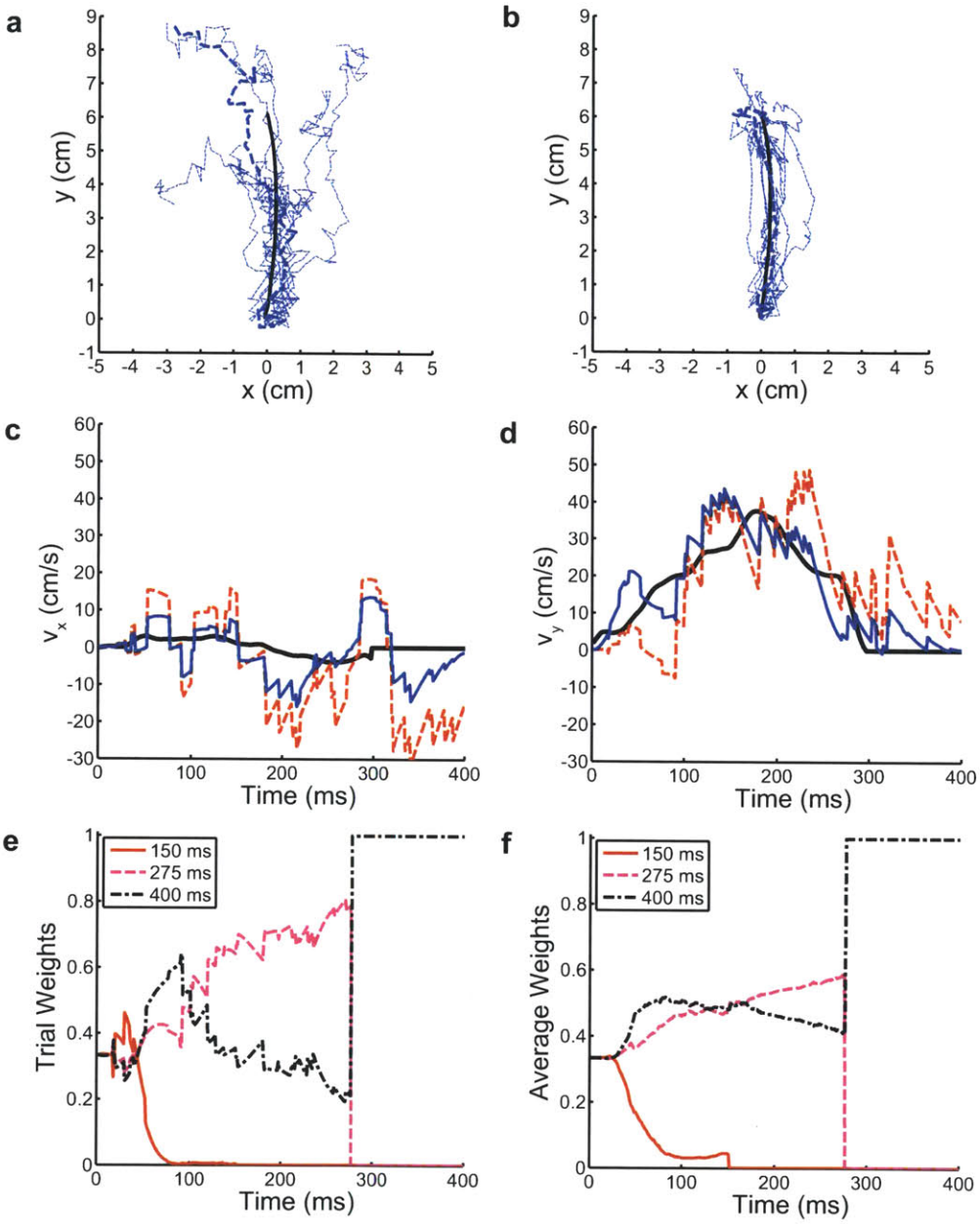


Figure 3-5: Comparison of the decoded trajectories using FC-P-PPF and RW-PPF when movement duration is unknown. (a) and (b) show a sample trajectory during a reaching movement performed by a rhesus monkey (solid black) and 10 sample decoded trajectories using RW-PPF in (a) and FC-P-PPF in (b) (dashed blue). The decoders are run until the end of the uncertainty interval. (c) and (d) show the true velocity (black) in the x and y directions and the decoded velocities using RW-PPF (dashed red) and FC-P-PPF (solid blue) for a sample realization of the neural signal. (e) and (f) show the evolution of the optimal combining weights for the branches of FC-P-PPF for the sample realization of the neural signal (e) and averaged over all 100 realizations (f).

using a more accurate prior model in the decoder. When movement duration is unknown, we showed that the average RMS error of the RW-PPF was 66% higher than that of FC-P-PPF with only 3 parallel branches. Similar to the results in Chapter 2 we showed that using very few parallel filters was sufficient to get to an asymptote average RMS value for FC-P-PPF, demonstrating that the added complexity to resolve the duration uncertainty is not large.

Here we have illustrated one example of the feedback-controlled prior state-space model in (3.7), namely for the specific case of reaching movement. However, this prior model is much more general and need not be confined to reaching movement. More specifically, this model could be specialized for any particular task by quantifying the goals of that task as the cost function in (3.2). Hence it provides a very general framework for construction of goal-directed prior models for various tasks that can in turn be used in the parallel point process framework to estimate the movement from the neural spiking observations.

# Chapter 4

# A real-time brain-machine interface combining target and trajectory information using an optimal feedback control design

In Chapters 2 and 3 we developed a recursive Bayesian decoder, FC-P-PPF, for estimation of goal-directed movements from ensemble spiking activity assuming that target location was known. We showed the effectiveness of FC-P-PPF using simulated neural activity. In this chapter and based on these developments, we implement a novel real-time BMI that, for the first time, jointly estimates the target and trajectory of movement in primates.

This approach is justified by two main reasons. First, both the kinematics and the target are encoded in the motor cortical areas [5, 18, 20–40]. Peri-movement activity, i.e., the activity around the time of movement, in the primary motor cortex, posterior parietal cortex (PPC), and dorsal premotor cortex (PMd) is related to the movement kinematics such as direction, velocity, position, and acceleration [5, 20–29]. In addition to peri-movement activity, neural activity in the PPC has been shown to encode the intended target [18,30–33] prior to movement initiation. Similar activity has also been observed in the premotor cortex including PMd [25,27,34–40]. Second, this approach more closely mirrors the natural way in which the sensorimotor system decides on a plan of action and executes its movement. In other words, the several components of the musculo-skeletal system are coordinated in order to reach a target and hence the target of a movement and the desired trajectory to reach it are strongly correlated [41, 42]. Based on these considerations, a more principled BMI design that aims to mirror the sensorimotor system by jointly decoding the movement target and the corresponding trajectory would allow for a potentially more accurate estimation of movement.

We have developed a real-time BMI that combines information about target and trajectory intent and implemented it in sensorimotor tasks performed by two rhesus monkeys. Unlike the developments in previous chapters which assumed knowledge of the target location, our real-time BMI decodes the target from the activity prior to movement. Our BMI hence employs a novel two-stage approach. In the first stage, it uses the neural spiking activity prior to movement

to predict the intended target of the movement. In the second stage, it combines this prediction with the peri-movement spiking activity to estimate the movement trajectory. To do so, we take advantage of the optimal feedback control theory of the sensorimotor system [41,42,58] by using the feedback controlled prior state-space model developed in Chapter 3. We also use the parallel filter framework developed in Chapter 2 to remove the duration uncertainty inherent to this prior model. Hence, the decoder used in the second stage of this BMI is the FC-P-PPF developed in Chapter 3. The target information in FC-P-PPF is in turn provided by the first stage of the BMI. Using the FC-P-PPF, the BMI models the real-time sensory feedback and incorporates it into its trajectory estimation. Also, using point process modeling, the two-stage decoder processes the spikes directly in real time for the first time, i.e., at the millisecond time-scale on which the neural spiking activity is recorded.

Here, we show that the two-stage BMI performs better than either stage alone and also significantly better than linear regression approaches commonly used in the literature [5–8,10]. We also find that the time to reach the target closely matches that normally achieved by the monkeys during natural movement. Also, using both target and trajectory information and combining them in a principled manner results in a robust BMI architecture that can achieve high performance in the presence of experimental variability.

## ■ 4.1 Experimental Setup

We measured the performance of our BMI in standard instructed-delay center-out directional tasks in which two monkeys used a joystick to move a cursor from the center of the screen to one of four targets displayed at its periphery while neurophysiologic recordings were made from their PMd and supplementary motor area (SMA). Unlike 'free-roaming' motor tasks in which the subject can freely move until reaching a target [6,7,9,10,14], this task required the monkey to reach the correct target *without* touching any of the incorrect targets under a limited time constraint.

The experiment consisted of two types of sessions, training sessions and BMI sessions. At the beginning of each day, the monkey first performed the standard task using a joystick during which statistical models for target and kinematics were constructed. These were termed the training sessions. The BMI sessions followed the training session during which the monkey performed the same task as before but this time cursor position was controlled by the neural activity recorded from the monkey as we will develop and discuss in Section 4.2. The BMI setup is illustrated in

Figure 4-1a. In the BMI sessions the monkey had visual feedback of the estimated cursor trajectory as it observed it on the screen.

We now explain the details of the behavioral task and the neurophysiologic recordings.

### ■ 4.1.1 Behavioral task

We used two adult male rhesus monkeys (*macaca mulatta*) in the study. During the tasks, the animals were seated in a primate chair (Crist Instrument Co Ltd, Damascus, MD). The primates' head were restrained using a head post, and a spout was placed in front of their mouth to deliver juice using an automated solenoid. A spring-loaded, two-degrees of freedom manipulandum was mounted anterior to chair on the side contralateral to recording. A computer monitor was placed in front of the animals at eye level which displayed the task. A NI DAQ card (National Instruments, TX) was used for the I/O behavioral interface, and the behavioral program was run in Matlab (MathWorks, MA) using custom made software (www.monkeylogic.net).

Primates performed a center-out visually-instructed motor directional task. The monkeys held the joystick contralateral to the site of recordings and could move their limb freely in the horizontal and vertical dimensions during the task. A computer monitor displayed the target locations and a cursor was used to represent the position of the joystick handle. Each individual trial began with the presentation of a central fixation point surrounded by four gray circular targets. Once the animals held the cursor within a central radius for a delay of 500 ms, one of the four randomly selected targets would turn green. After another 1000 ms, the fixation point would change color ("go" cue), at which time the monkeys could use the joystick to move the cursor from the center of the screen to the instructed target. Once the cursor reached the target, the animal received a drop of juice following another 250 ms delay if the correct target was selected. If during a trial, the animals moved prematurely, failed to reach any target during the allowed time or touched an incorrect target before reaching the correct target, the trial aborted. Once reward was delivered, another 1000 ms would lapse, the targets would erase, and the sequence would repeat again. The animals were required to return the spring-loaded joystick to the center fixation point before a new trial began.

### ■ 4.1.2 Neurophysiologic recordings and BMI setup

A titanium head post and recording electrodes were surgically implanted in each monkey contralateral to the side of joystick use. All procedures were performed in an IACUC-approved aseptic primate surgical facility. Prior to electrode implantation, craniotomies were performed

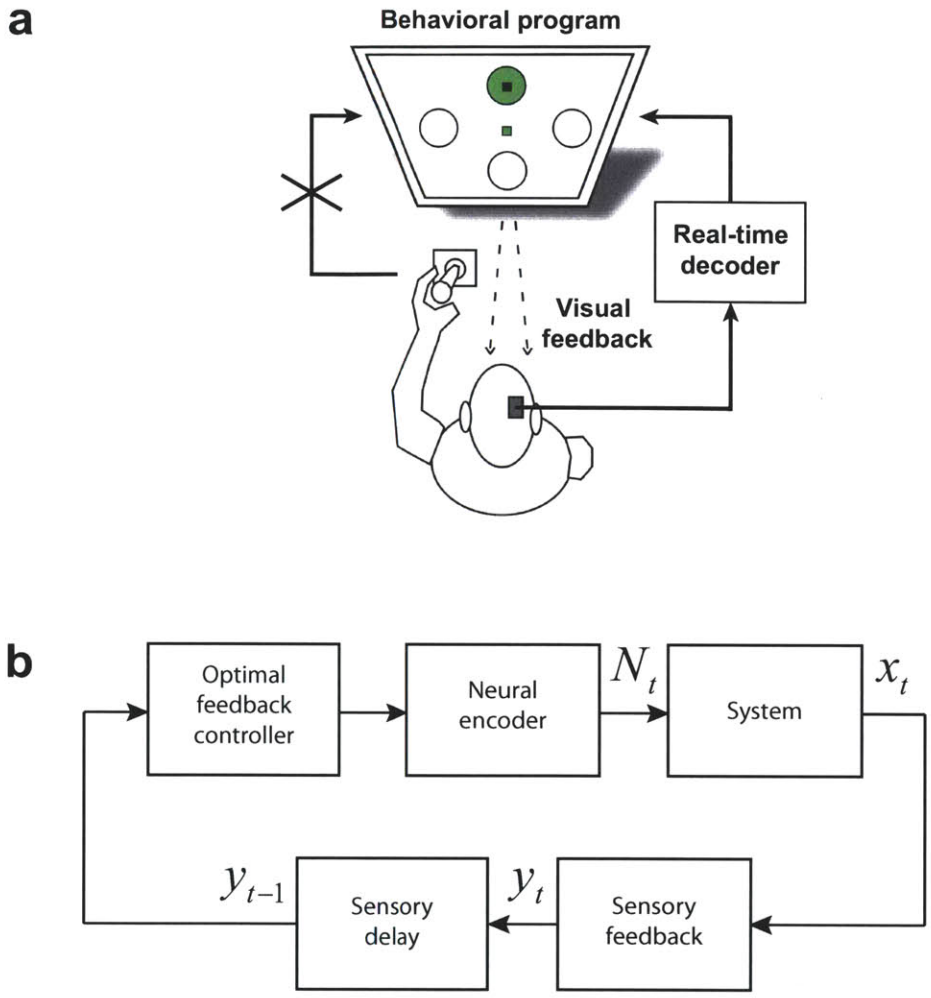


Figure 4-1: Experimental task and the optimal feedback control framework. (a) Experimental task. The experiment consisted of an instructed delay center-out task with four targets. To be rewarded, the monkey had to acquire the correct target *without* touching any of the incorrect ones first. After target presentation, there was 1 s of delay before the "go" cue, signaling that the monkey could begin moving the joystick. During the training sessions the monkey controlled the position of the cursor using a joystick. During the BMI sessions the joystick was disconnected and the real-time decoder controlled the cursor while the monkey had visual feedback of this process. (b) Optimal feedback control framework. An optimal feedback control framework is used to model the task. In this framework each task is performed to accomplish a goal during which there is real-time sensory feedback, $\mathbf{y}_t$ , about the state of the system, $\mathbf{x}_t$. Based on the intended goal and the sensory feedback about the state of the system, the brain (controller) decides on a plan of action, which is reflected in its neural activity, $\mathbf{N}_t$, and controls the system. In natural movement this is the animal's own musculo-skeletal system and in movement using a BMI this is the BMI.

over the sites of interest using standard stereotactic coordinates. Once the cortex was exposed and the sulcal anatomy identified, several silicone microelectrode arrays were placed in the cortex

(Neuronexus technologies, MI). The electrodes were secured into place using fibrin glue, silicone sealant, and methylmethacrolate. The distal leads were then attached to a female connector and secured to the skull with titanium miniplates and dental acrylic. Anatomic post-mortem confirmation of electrode positioning was performed in one monkey. The second monkey is still performing behavioral tasks.

Recordings began at two weeks following surgical recovery. A Plexon multichannel acquisition processor was used to amplify and band-pass filter the neuronal signals (150 Hz – 8 kHz; Plexon Inc., TX). Shielded cabling carried the signals from the electrode array to a set of six 16-channel amplifiers. Signals were then digitized at 40 kHz and processed to extract action potentials in real time by the Plexon workstation. Classification of the action potential waveforms were accomplished using dual-window discrimination and principle component analysis. Units with stable, identifiable waveform shapes and adequate refractory periods determined by autocorrelation were then used for the real-time experiments. Joystick position was sampled and recorded at 1 kHz. Neuronal data obtained from the Plexon workstation, in the form of action potential time stamps and channel, were then transmitted to a second PC computer running Matlab (Mathworks, MA) in real time. On decoder training sessions, the primates would use the joystick to move a cursor on the screen to one of four randomly selected targets over multiple trials. On decoder BMI sessions, the monkeys would still be allowed to use the joystick but the cursor image displayed on the screen would be supplied by the Matlab real-time decoder as illustrated in Figure 4-1a. The cursor was initially placed at the center fixation point at the beginning of each trial. Here, estimated cursor movements would be relayed through a DAQ I/O (National Instruments, TX) to a third PC computer running the behavioral task. The computer would then display the estimated cursor position.

## ■ 4.2 Methodology: The two-stage decoder

The BMI processing consists of two stages and decodes two aspects of movement. During the first stage, it uses a maximum-likelihood (ML) decoder based on a point process model of the neural spiking activity to predict the monkey's intended target of movement during the delay period after target presentation but before movement. In the second stage the BMI combines this decoded target with the peri-movement activity, using FC-P-PPF developed in Chapter 3 to estimate the trajectory.

We now explain the details of the decoder in the two stages.

### ■ 4.2.1 Target decoding

The BMI decodes the monkeys' intended target of movement by recording the ensemble spiking activity during the 800 ms delay interval prior to the "go" cue. Note that the delay between the start of target presentation and "go" cue is 1000 ms. We do not use the activity in the first 200 ms in the BMI in order to allow sufficient time for the visual target information to reach the PMd and SMA [19]. Using offline cross-validation analyses we observed that discarding this activity improves the prediction accuracy. Spiking activity of each neuron during this delay interval is modeled as a homogeneous Poisson process (a point process with constant rate) whose firing rate is a function of the intended target, fitted using the GLM framework. A maximum-likelihood (ML) decoder first calculates the likelihood probability of this ensemble activity for each possible target, $G$, and then selects the target with the highest likelihood as its prediction. Denoting the neural point process observations of the ensemble of $C$ neurons by $\mathbf{N}_1, \cdots, \mathbf{N}_t$ where $\mathbf{N}_t = (N_t^1, \cdots, N_t^C)$ is the binary spike events of the $C$ neurons at time $t$, and assuming that the neurons are conditionally independent given the target, the point process likelihood model for the ensemble is given by [50]

$$p(\mathbf{N}_t|G) = \prod_{c=1}^{C} (\lambda_c(G)\Delta)^{N_t^c} e^{-\lambda_c(G)\Delta} \tag{4.1}$$

where $\Delta = 5$ ms is the time increment used for binning the spikes and $\lambda_c(G)$ is the modeled firing rate of the $c$th neuron during the delay period for target $G$. The ML decoder then predicts the target as the one maximizing the ensemble likelihood,

$$\hat{G} = \underset{G=1,\cdots,4}{\arg\max}\, p(\mathbf{N}_1, \cdots, \mathbf{N}_{T_d/\Delta}|G) = \underset{G}{\arg\max} \prod_{c=1}^{C} \prod_{t=1}^{T_d/\Delta} (\lambda_c(G)\Delta)^{N_t^c} e^{-\lambda_c(G)\Delta} \tag{4.2}$$

Here $T_d = 800$ ms is the delay period.

### ■ 4.2.2 Kinematic decoding

In the second stage, the BMI combines the decoded target with the peri-movement ensemble spiking activity using FC-P-PPF. As we discussed in Chapter 2, the prior model in any recursive Bayesian decoder should in general incorporate any prior information available about the kinematic states, which for a goal-directed movement includes the intended target. Hence, we develop the goal-directed prior model of the decoder using the optimal feedback control design in Chapter 3, which is also used to model the visual feedback. To develop the observation model,

we use a point process model of the spiking activity whose instantaneous rate is a log-linear function of kinematics. The resulting decoder hence processes the spikes directly in real time and hence operates at the millisecond time scale of the spiking activity. We now develop the prior and observation models used in the decoder.

**Prior model: an optimal feedback control design**

An optimal feedback control design is used to build a prior goal-directed state-space model for the kinematics and to model the visual feedback generated by the BMI in computing the trajectory estimate. As described also in Chapter 3, this design is inspired by the optimal feedback control theory of the sensorimotor system used to explain its function [41, 42]. In this framework, each task is performed to accomplish a goal during which there is sensory feedback about the state of the musculo-skeletal system. Based on the intended goal, the sensory feedback (for example vision and proprioception) about the current state of the system, and the desired time to accomplish the goal, the subject (controller) decides on the next plan of action or control signal (for example muscle activation) and controls its movement. Specifying an approximate kinematic model and quantifying the task goals as cost functions and also the sensory feedback, this framework can predict the next plan of action in the presence of model uncertainty and sensory noise. The difference in applying this framework to natural movement [41, 42] and movement using a BMI is that in the former case the system to be controlled is the musculo-skeletal system and in the latter, it is the BMI. This is illustrated in Figure 4-1b. Based on the visual feedback of the cursor position and the intended target, the monkey decides on the next plan of action. Similar to natural movement, the next plan of action is in turn reflected in the neural activity but this time controls the system or BMI through the decoder as opposed to directly controlling the musculo-skeletal system. Hence we again use the prior feedback-controlled state-space model developed in Chapter 3 to predict the monkey's next plan of action or control signal in the decoder and consequently the next kinematic state.

The prior model is therefore given as in (3.7),

$$\mathbf{x}_{t+1} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\mathbf{x}_t + \mathbf{w}_t, \tag{4.3}$$

This prior model is in turn specialized to the particular case of the reaching movements in this experiment as developed in Section 3.1.1 by augmenting the state to include the target location in

the two dimensions. Note that in our BMI, the target location, $\mathbf{d}^*$, used in the cost function in (3.8) is determined from the decoded target in the first stage.

**Observation model**

We build the observation for each neuron as a point process whose instantaneous firing rate is a function of kinematics, i.e., [50]

$$p(N_t^c|\mathbf{x}_t) = (\lambda_c(t|\mathbf{x}_t)\Delta)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t)\Delta}$$

We used a modified version of the cosine tuning model [20, 23] for the instantaneous firing rate, modeling it as a log-linear function of position and velocity in the two dimensions, i.e.,

$$\lambda_c(t|\mathbf{x}_t) = \exp(\alpha_0 + \boldsymbol{\alpha}_c'\mathbf{x}_t), \tag{4.4}$$

where $\mathbf{x}_t$ denotes these kinematic states at time $t$ and $\alpha_0$ and $\boldsymbol{\alpha}_c$ are fitted using the GLM framework [50] on the peri-movement spiking activity. We assumed that the spiking activity of the neurons are conditionally independent given the kinematic states and hence the observation model for the ensemble is given by (cf. (2.3))

$$p(\mathbf{N}_t|\mathbf{x}_t) = \prod_{c=1}^{C} (\lambda_c(t|\mathbf{x}_t)\Delta)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t)\Delta} \tag{4.5}$$

**Uncertainty in the movement duration**

Having the prior and the observation models we can now develop the recursions for the Bayesian decoder. However, the prior model built in (3.7) is dependent on the movement duration, $T$, which is not known to the decoder. In other words, unlike natural movement in which the monkey (controller) decides on the movement duration, in movement using a BMI the decoder does not have a priori knowledge of this duration. We hence use the FC-P-PPF developed in Chapter 3 to resolve this duration uncertainty and estimate the kinematic state.

**Feedback-controlled parallel point process filter (FC-P-PPF)**

We use the FC-P-PPF to estimate the kinematic states. As described in Section 3.2, the recursions of this filter for a discretized movement duration are given by (2.10)-(2.13) with $\mathbf{G}(t, T, \mathbf{x}^*) = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))$ and $\mathbf{b}(t, T, \mathbf{x}^*) = 0$. For the rate model in (4.4) since $\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t} = \boldsymbol{\alpha}_c'$ and $\frac{\partial^2 \log \lambda_c}{\partial \mathbf{x}_t \partial \mathbf{x}_t'} = \mathbf{0}$

these recursions simplify to

$$\mathbf{x}_{t|t-1,T} = (\mathbf{A} - \mathbf{BL}_t(T))\mathbf{x}_{t-1|t-1,T} \tag{4.6}$$

$$\mathbf{W}_{t|t-1,T} = (\mathbf{A} - \mathbf{BL}_t(T))\mathbf{W}_{t-1|t-1,T}(\mathbf{A} - \mathbf{BL}_t(T))' + \mathbf{W}_t \tag{4.7}$$

$$\mathbf{W}^{-1}_{t|t,T} = \mathbf{W}^{-1}_{t|t-1,T} + \sum_{c=1}^{C} \boldsymbol{\alpha}_c \boldsymbol{\alpha}_c' \lambda_c(t|\mathbf{x}_{t|t-1,T})\Delta \tag{4.8}$$

$$\mathbf{x}_{t|t,T} = \mathbf{x}_{t|t-1,T} + \mathbf{W}_{t|t,T} \sum_{c=1}^{C} \boldsymbol{\alpha}_c (N_t^c - \lambda_c(t|\mathbf{x}_{t|t-1,T})\Delta) \tag{4.9}$$

To give some insight into these recursions, note that in the prediction step given in (4.6) the feedback controlled prior model is used to move the estimate forward. In the update step given in (4.9) the estimate is found by making a correction or update to this prediction. Here, $\lambda_c(t|\mathbf{x}_{t|t-1,T})\Delta$ is the predicted probability of having a spike in the time interval $\Delta$ and hence the correction is (1- predicted probability of a spike) if a spike occurs and (0- predicted probability of a spike) if no spike occurs. Hence if a spike occurs and the predicted probability of a spike is high this correction is small and vice versa. Therefore the estimate is a combination of the prediction and the correction terms. The more informative the spiking activity is about the state, i.e., the larger the absolute value of $\boldsymbol{\alpha}_c$ is, the more weight is placed on the correction term and vice versa. If the spiking activity is not informative at all, or $\boldsymbol{\alpha}_c = \mathbf{0}$ , then the estimate will just be the prediction, which is obtained only using the feedback-controlled state model and ignoring the observation model.

To remove the duration uncertainty, we coarsely discretize the duration of a trial. We also use the first implementation of the parallel filter framework developed in Chapter 2 as it is computationally more efficient.

### ■ 4.2.3 Possible Extension

Note that instead of making a decision on the intended target in the first stage of the BMI and then using only the prior model corresponding to that target, we can include branches in the FC-P-PPF for all the possible target locations and their corresponding discretized durations. Doing so, we can calculate the final kinematic estimate as the weighted average of the branch estimates corresponding to all possible target locations, as opposed to just the decoded target location. The calculation of the branch likelihoods, that in turn determine their weights in calculating the final kinematic estimate, remains exactly the same (see (2.15)-(2.16)). For this extension and to also in-

corporate the activity before movement initiation, at the beginning of kinematic decoding we can initialize the weight of the branches corresponding to each target proportional to its likelihood calculated from this activity as opposed to equiprobably. This extension is similar to the approach taken in [53] to combine time-invariant and empirically fitted trajectory models to different targets. In our case, this means including more parallel branches in the FC-P-PPF for the different targets (and their discretized durations). This will consequently increase the complexity of the decoder. We chose not to implement this extension since, using offline analysis, we observed that it resulted in little improvement at the price of four times the complexity. The absence of a significant improvement in this case was likely a result of the fact that in our experiments the target-related activity during the delay period was strongly tuned to the target and that overall the peri-movement activity was not as strongly tuned to the task as we will present in Section 4.3. However, in cases where such target-related activity is weak, this extension will potentially result in further improvement in the second stage as it allows for its higher weighting compared to the first stage.

## ■ 4.3 Results

We measured the performance of our BMI in standard instructed-delay center-out directional tasks in which two monkeys used a joystick to move a cursor from the center of the screen to one of four targets displayed at its periphery as described in detail in Section 4.1.1. It is important to emphasis here that unlike 'free-roaming' motor tasks in which the subject can freely move until reaching a target [6, 7, 9, 10, 14], this task required the monkey to reach the correct target *without* touching any of the incorrect targets under a limited time constraint. Hence the task required the monkey not only to reach the correct target but also to reach it using constrained paths that could at no point touch an incorrect target. The performance measure used in these experiments was the acquisition accuracy, which is the percentage of trials on which the task is successfully completed. Multi-electrode spiking activity was recorded from PMd and SMA from which $20 \pm 2$ neurons (mean $\pm$ s.d.) were isolated and used in the BMI. At the beginning of each day, the monkey first performed the standard task using a joystick (training session) during which target and kinematic models were constructed. The monkey then performed the same task as before but this time cursor position was controlled by the neural activity recorded from the monkey (BMI sessions). In the BMI sessions the monkey had visual feedback of the estimated cursor trajectory as it observed it on the screen.

### ■ 4.3.1 Model training and validation

Models for the BMI were trained on the neural spiking activity during the training session at the beginning of each day. Neural recordings were made during target presentation prior to the presentation of the "go" cue, which signaled that the monkey could move the joystick, as well as during movement itself after the "go" cue.

Each training session consisted of an average of 89 $\pm$ 2 trials. Point process models relating neural spiking activity to target location and movement kinematics were constructed based on the known target location and cursor position for each trial and the recorded multiple-neuronal activity using the generalized linear models (GLM) framework [50]. Models were then cross-validated (leave-one-out) on the same data by finding the corresponding target predictions and kinematic estimates. Target location was predicted using the ML decoder from the neural spiking activity in the 800 ms delay period prior to the "go" cue. Kinematics were estimated using either the two-stage decoder that combined the target predictions from the first stage with the peri-movement activity or its second stage, but not taking into account the target predicted from the first stage.

During the delay period, the ensemble spiking activity (20 $\pm$ 2 neurons) predicted the correct target with high accuracy in the training sessions (leave-one-out cross-validation). The prediction accuracy of the trained point process target models across sessions, measured as the percentage of trials on which the models correctly predicted the target using the delay neural activity, was 81 $\pm$ 3%. To examine further the contribution of the individual neurons to the target prediction accuracy, we performed a neuron dropping analysis in which the spiking activity of a single neuron during the delay period was used to decode the target. We found that across sessions, 48 $\pm$ 12% of the neurons had a target prediction accuracy significantly greater than chance ($P < 0.05$). As described in the Appendix, we further found that relatively few neurons (on average 3.3 $\pm$ 1.0 across sessions) were sufficient to obtain a target prediction accuracy that was higher than 90% that of the ensemble.

During movement, the premotor neurons were tuned to position and velocity. Fitting the point process models for the kinematics using the GLM framework [50], we found that across sessions 47 $\pm$ 16% of the premotor neurons were significantly tuned to either position or velocity at least in one dimension ($P < 0.05$; Bonferroni correction for multiple comparisons). Of these neurons, 57% were tuned to position only, 15% were tuned to velocity only, and 28% were tuned

to both position and velocity. In agreement with previous studies, these findings suggested that the recorded premotor neurons held significant information about both the target and kinematics of the movement.

### ■ 4.3.2 Offline model comparisons

We tested the performance of the two-stage decoder in an offline analysis of the training sessions data using leave-one-out cross-validation. We also compared to the performance of the second stage of the decoder alone by replacing the feedback-controlled state-space model in the FC-P-PPF with a random-walk (RW) model, which uses no prior target information and only enforces smoothness in the trajectory. The resulting filter is the RW-PPF (see Section 3.3). We also compared the performance of the two-stage decoder to that of the commonly used least-squares linear regression decoder [5–8,10]. For each decoder we updated the position estimate every 5 ms, which was also the bin width for the spiking activity. Note that the chance level acquisition accuracy in our task should be approximately 25% since there are four targets on the screen and hitting the wrong one results in an error. We confirmed this by estimating the trajectory from shuffled neural activity, which resulted in an accuracy of 23 $\pm$ 5% across sessions, as presented in the Appendix.

We found that the acquisition accuracy of the two-stage decoder across sessions was 83 $\pm$ 3% compared to 61 $\pm$ 7% for the RW-PPF, which used only the peri-movement activity. This is illustrated in Figure 4-2a. Hence, target predictions from the first stage of the decoder on average corrected 56% of the errors made by the inaccurate trajectory estimation of the second stage. Also, the trajectories estimated by the two-stage decoder were closer to the monkey's trajectory as illustrated in Figures 4-3a and 4-3b. To quantify this, we measured the average root mean-square (RMS) error across all trials as illustrated in Figure 4-2b.We found that the RMS error of the RW-PPF on average was 40% higher than the two-stage decoder. Also, incorporating the target and the feedback-controlled state-space model reduced the time it took for the two-stage decoder to reach the target. The two-stage decoder touched the correct target on average in 240 $\pm$ 11 ms across sessions compared to 317 $\pm$ 46 ms for the RW-PPF. Comparing the accuracy of the two-stage decoder to the target prediction accuracy of its first stage, we found that the second stage on average corrected 11% of the target prediction errors. However, as these were offline estimation of fast joystick movements, the more appropriate test for the correction of the incorrect target predictions by the peri-movement activity in the second stage is in real-time BMI sessions presented in Section 4.3.3.

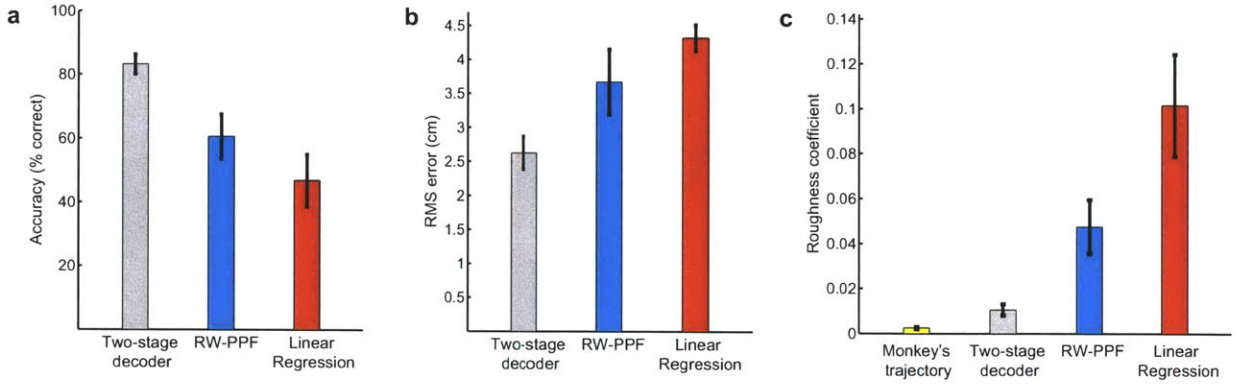


Figure 4-2: Offline model comparisons. The bars show mean quantities and the error bars show the standard deviation around the mean across sessions. All quantities are obtained from the training sessions using leave-one-out cross-validation. (a) Accuracy of the different models. (b) RMS error of the different models. (c) Roughness coefficient of the different models. The two-stage decoder (used in the real-time BMI) outperforms all other models in terms of accuracy, RMS error, and smoothness.

We also compared the performance of the two-stage decoder to that of a linear regression decoder as shown in Figure 4-3. The regression decoder reconstructed the position at each time as a linear function of the history of the ensemble firing rates, which were calculated every 5 ms in sliding bins of 100 ms. We included up to 800 ms—same duration as the delay period used for target prediction—of history coefficients in the regression decoder. Using an exhaustive search we used the number of history coefficients that maximized the accuracy of this decoder. The average accuracy of the regression decoder across sessions was $47 \pm 8\%$, which was 36% lower than the two-stage decoder as shown in Figure 4-2a. Also the average RMS error of the regression decoder was 65% higher than the two-stage decoder as shown in Figure 4-2b. Note that we allowed for a fair comparison to the linear regression decoder by adding history coefficients.

Finally the trajectory estimations in the two-stage decoder were smoother than either the RW-PPF or the regression decoder as can be seen from Figure 4-3. To quantify this, we calculated the roughness coefficient [59] for each of the decoders as shown in Figure 4-2c. The roughness coefficient measures the degree of smoothness in the estimated trajectory and is smaller for smoother estimates. We give the formal definition of this coefficient in the Appendix. We found that the roughness coefficients of the RW-PPF and the linear regression decoder were 4.5 and 9.6 times larger than that of the two-stage decoder, respectively.

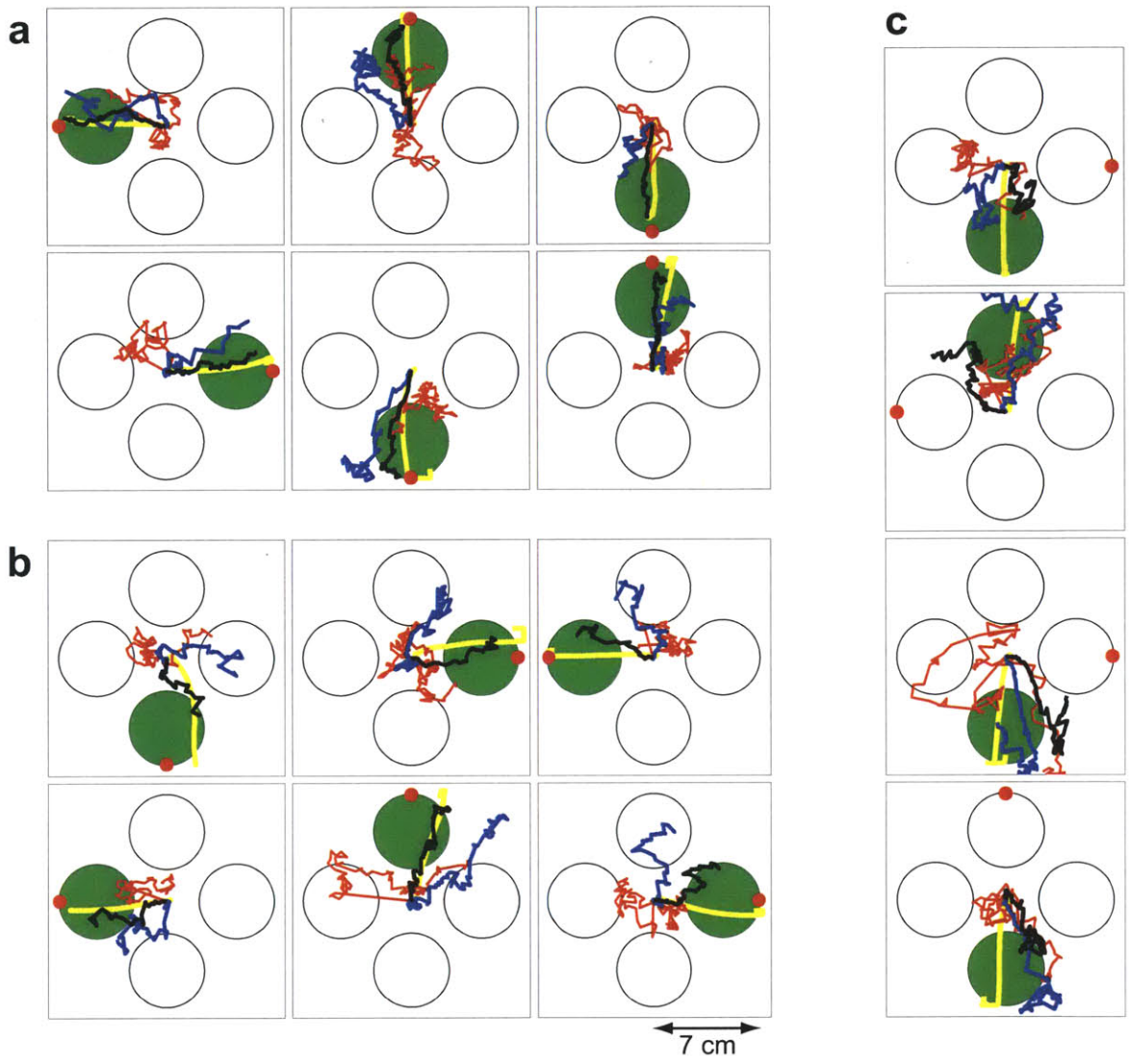


Figure 4-3: Comparison of the offline trajectory estimates. The green circle shows the instructed target and the yellow line shows the monkeys trajectory. The black line shows the trajectory estimate using the two-stage decoder, the red circle shows the predicted target from the first stage, the blue line shows the trajectory estimate of the RW-PPF, and the red line shows that of the linear regression decoder. (a) Sample trials in which both the two-stage decoder and the RW-PPF acquire the target correctly. (b) Sample trials in which the two-stage decoder acquires the target correctly but the RW-PPF does not. (c) Sample trials in which the two-stage decoder acquires the target correctly but the target is predicted incorrectly from the first stage.

### ■ 4.3.3 Combined target and trajectory decoding in a real-time BMI

To investigate whether kinematic and target related activity can be jointly used to obtain accurate motor performance in real time, the monkeys performed the same task as before but using a BMI.

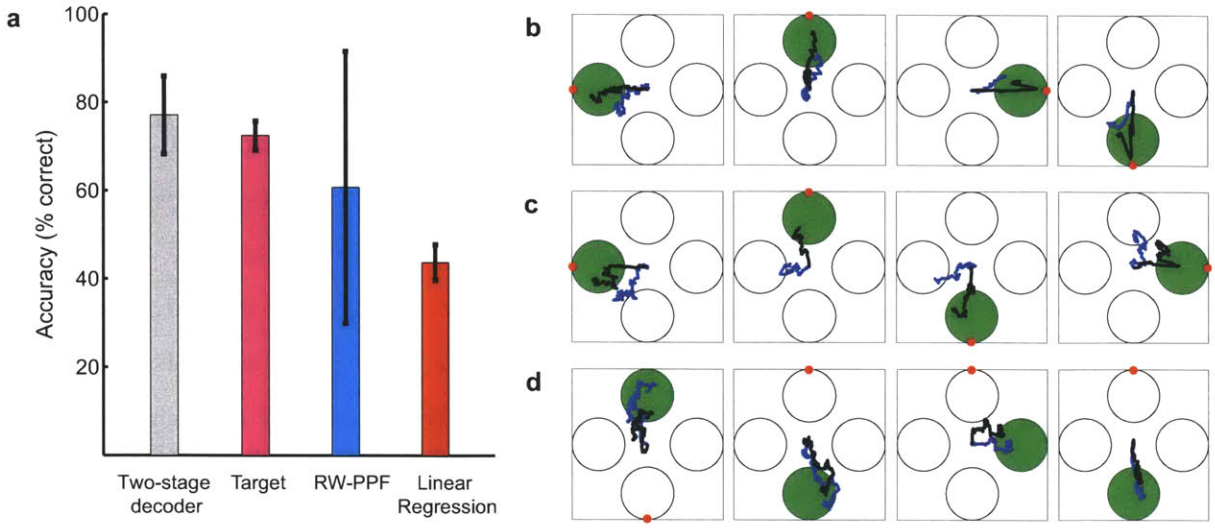


Figure 4-4: Real-time comparisons and the complementary property of the two-stage BMI. (a) Comparison of the real-time BMI accuracy with that of the RW-PPF and the linear regression decoder obtained using the same real-time data set, and also with the real-time target prediction accuracy. The bars show mean quantities and the error bars show the standard deviation around the mean across sessions. The real-time BMI outperforms all other decoders. (b-d) Comparison of the trajectory estimates. The green circle shows the instructed target, the red circle shows the predicted target from the first stage, the black line shows the trajectory estimate of the real-time BMI, and the blue line shows that of the RW-PPF using only the peri-movement activity. Sample trials where both the real-time BMI and the RW-PPF correctly acquire the target are shown in (b). Sample trials in which the correct target prediction of the first stage compensates for the inaccurate estimation of the kinematic decoder in the second stage are shown in (c). Sample trials in which the kinematics decoder in the second stage, using ongoing peri-movement activity, compensates for the incorrect target prediction of the first stage are shown in (d). These illustrate the complementary property of the two-stage BMI.

The real-time BMI used the two-stage decoder. During the 800 ms delay period prior to the "go" cue, the BMI predicted the target and after the "go" cue it combined this target information with the peri-movement activity using the FC-P-PPF to decode the kinematics. The BMI updated the estimated position of the cursor every 5 ms. We found that using the two-stage BMI, the monkeys could perform the task with an average accuracy of $77 \pm 9\%$ as shown in Figure 4-4. To assess the stability of the performance throughout the recordings per day, we compared it in the first and second half of sessions. We found that accuracy did not change significantly (Wilcoxon rank-sum test, $P > 0.5$) and remained stable.

We next compared the accuracy of the two-stage decoder to that for each of its stages alone and found that the two stages of the BMI performed in a complementary manner; the correct target predictions could compensate for the inaccurate performance of the kinematics decoder and the ongoing kinematic estimation could correct the incorrect target predictions. To compare the accuracy of the BMI with that of using only the peri-movement activity, i.e., using the second stage of the BMI alone without target information, we decoded the trajectory on the same real-time data set using the RW-PPF. Sample decoded trajectories are shown in Figure 4-4b-4-4d. The average acquisition accuracy of the RW-PPF was 61 $\pm$ 31% as shown in Figure 4-4a, which is considerably lower than the BMI. This suggests that the first stage of the BMI on average corrected 41% of the second stage errors. Sample trials with this correction are shown in Figure 4-4c. The average real-time target prediction accuracy of the first stage of the BMI was 72 $\pm$ 3%. This suggests that the second stage of the BMI (FC-P-PPF) on average corrected 18% of the incorrect target predictions of the first stage. Sample trials with this correction are shown in Figure 4-4d. Hence the joint performance of the BMI was higher than would have been possible using either kinematic or target related activity alone.

The acquisition time in the BMI sessions, i.e., time until the trial ended by rewarding the monkey, was close to the natural acquisition time in the training sessions. In our experiments we used a short 3 s time-out condition to make the task more challenging and the required acquisition time closer to that of monkey's own movement. The median acquisition time for the natural movement was 0.6 $\pm$ 0.3 s and for the two-stage BMI trials was 0.9 $\pm$ 0.5 s.

### ■ 4.3.4 Control comparisons

First, to relate the performance of the BMI to the frequently used linear regression decoder used in real time in the literature [5–8, 10], we compared the accuracy of our two-stage BMI with that of the regression decoder on the same real-time data set as shown in Figure 4-5. The linear regression decoder used up to 800 ms of history coefficients (same duration as the delay period used for target prediction) and the number of history coefficients in it was chosen by an exhaustive search and to maximize its accuracy. The average accuracy of the linear regression decoder was 44 $\pm$ 4%, which not only is significantly lower than the two-stage BMI, but also is lower than either stage of the two-stage BMI alone. This is illustrated in Figure 4-4a. Comparing the trajectories produced by the BMI to those of the linear regression decoder we observed that they were smoother. This was the case even in trials in which both decoders acquired the correct target (Figure 4-5a)

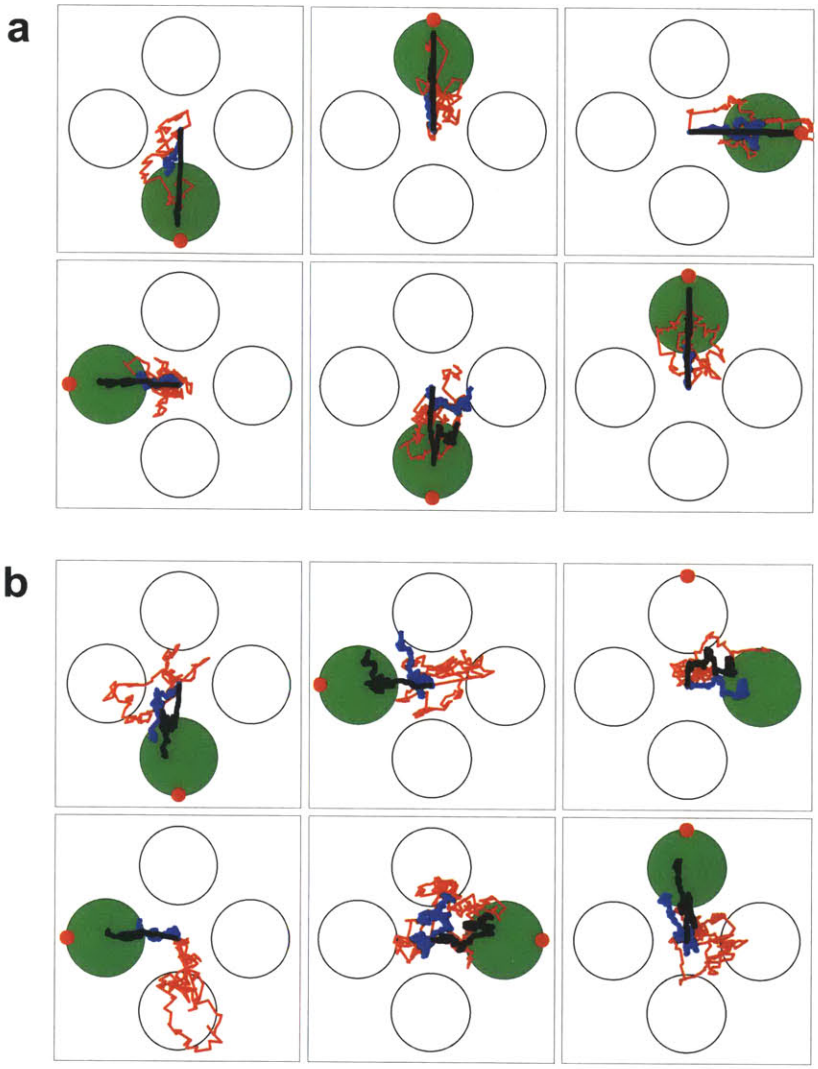


Figure 4-5: Comparison of the trajectory estimates. The black line shows the trajectory estimate of the real-time BMI. The blue line shows that of the RW-PPF using only the peri-movement activity, and the red line shows that of a linear regression decoder, both using the same real-time data set as the BMI. (a) Sample trials in which the linear regression decoder is correct. (b) Sample trials in which the linear regression decoder is incorrect. Even when both decoders acquire the correct target, the trajectory estimates of the two-stage BMI are smoother than the linear regression decoder.

where the roughness coefficient in the regression decoder was 3.3 times that of the two-stage BMI. These results show the potential benefit of using a more principled BMI that aims to mimic the sensorimotor processing underlying motor control to generate movement.

Second, to determine differences in BMI performance across monkeys and therefore the robustness of the two-stage BMI to individual variability, we examined differences in performances

between the two monkeys, A and B. For monkey A, the real-time BMI accuracy was 67 $\pm$ 4%. Comparing to the second stage alone using only the peri-movement activity, we found that the accuracy of the RW-PPF on the real-time data set for this monkey was only 29 $\pm$ 1%. Hence, even though the kinematic tuning in this monkey was weak, the real-time BMI had a relative high accuracy. This showed that the first stage of the BMI on average corrected 54% of the errors due to the inaccurate performance of the kinematics decoder. The real-time target prediction accuracy of the first stage using only the delay activity in this monkey was 71 $\pm$ 1%. Note that because of the weak kinematic tuning in the recorded neurons in this monkey, the correction happened only by the first stage. The accuracy of the linear regression decoder on the same real-time data set in this monkey was 45 $\pm$ 5%, significantly lower than that of the two-stage BMI using the same neural activity. For monkey B, the real-time BMI accuracy was 82 $\pm$ 5%. Comparing to the second stage alone using only the peri-movement activity, we found that the accuracy of the RW-PPF in this monkey was 76 $\pm$ 24% and hence the first stage of the BMI on average corrected 25% of the errors made due to the inaccurate trajectory estimation of the second stage. Comparing to the average real-time target prediction accuracy of the first stage, which was 73 $\pm$ 4%, suggested that the second stage of the BMI in monkey B on average corrected 33% of the target prediction errors made by the first stage. Therefore, in this monkey both stages exhibited the corrective behavior. The average accuracy of the linear regression decoder on the same real-time data set in this monkey was 43 $\pm$ 4%, significantly lower than the real-time BMI. Together, these findings suggest that although the performance of one stage in the two-stage BMI may not be equal across monkeys or recording areas, by combining both target and trajectory related information the two-stage approach provides a robust computational system that maintains good accuracy under variable experimental conditions.

## ■ 4.4 Discussion

Based on our understanding of the sensorimotor system [41, 42, 58], natural movement incorporates information about the intended target as well as the trajectory of the movement. We implemented a novel real-time BMI designed to mimic the sensorimotor system by a two-stage approach in which first the activity prior to movement initiation is used to predict the intended target, and second this prediction is combined with the peri-movement activity to estimate the trajectory using an optimal feedback control design. This is to our knowledge the first time that sensorimotor control principles and direct point process modeling of the spiking activity have

been used to implement a real-time BMI. This work hence differs from current real-time attempts at estimating movement using linear decoders by presenting a novel approach that aims to mimic the sensorimotor processing underlying motor control. In particular, compared to the performance of a standard linear regression decoder ($44 \pm 4\%$), not only did the two-stage BMI perform significantly more accurately ($77 \pm 9\%$), but also either of its stages alone had a higher accuracy compared to this decoder.

To combine the target and the peri-movement activity, we took advantage of the optimal feedback control theory of the sensorimotor system. In this view, each task is performed to accomplish a goal during which there is sensory feedback about the external state of the system. Specifying an approximate kinematic model, modeling the real-time sensory feedback about the state of the system, and quantifying the task goals as cost functions and the desired time to accomplish them, we can predict the next plan of action by finding the one that minimizes the cost function. For a BMI, the same framework can be applied to predict the next plan of action. The only difference is that the system being controlled is the BMI as opposed to an individual's own musculo-skeletal system. Hence the individual's next plan of action is reflected in the neural activity that, in turn, controls the BMI. Also, the BMI does not have knowledge of the desired time to accomplish the goal, which is decided by the controller (the individual). The present BMI hence resolved this movement duration uncertainty based on the neural spiking activity in real time. Also, by using an optimal feedback control design, our BMI modeled the sensory feedback in its trajectory estimation. Finally our BMI processed the spiking activity directly and hence operated at the millisecond time-scale of the spikes. In addition to its application to interpreting the sensorimotor function, optimal feedback control has also been deemed valuable for interpreting the neural basis of movement in the motor cortical areas [60]. This further motivates the use of optimal feedback control principles for the design of real-time BMIs.

We demonstrated that the two stages in the BMI functioned in a complementary manner. When the spiking activity for one stage was less informative, the other stage often provided sufficient information for the BMI to reach the correct target. As a result, the two-stage BMI performed better than either stage alone. Overall, the estimated trajectories using the two-stage approach were more accurate, had lower RMS error, and were smoother than the linear regression decoder or a random-walk point-process decoder. The BMI accuracy did not rely on multiple days of training or neuronal plasticity but was achieved after 15 minutes or less of training. Unlike 'free-roaming' motor tasks in which subjects can move freely until reaching a target, the present task

was demanding in that at no point the trajectories could touch the incorrect targets and then proceed to the correct target. This was considered an incorrect response. In addition the response time was constrained. Despite this, the monkeys were able to achieve a high accuracy using the two-stage BMI ($77 \pm 9\%$). The acquisition time in our BMI was also close to the monkeys' natural acquisition time.

The BMI achieved high accuracy by using relatively few neurons (11 on average) that were tuned to either target or trajectory. This was accomplished by modeling both the target and kinematic related spiking activity of these neurons directly as point processes. Recent work [61] has demonstrated that reducing the bin width used to calculate the firing rates of the spiking activity, which are in turn used as input in a BMI, improves its performance. These results could therefore suggest that moving to the time-scale on which the spiking activity is recorded, i.e., processing the spikes directly, could improve the performance of the BMIs.

In the two-stage BMI, one stage compensated for the inaccuracies of the other and vice versa. Since our model relies on neural activity to estimate the movement, we cannot test directly to what extent the brain uses these two aspects of motor control to execute a movement. It is interesting to speculate, however, that similar to findings made in these experiments, the premotor cortex may use information on intended target location to correct for discrepancies in ongoing movement. Similarly, it may use information about ongoing movement to fine-tune differences between initially intended target location and target location during movement itself. Hence, in addition to enabling the design of more accurate decoding algorithms, the present feedback-control design may provide insight into the control aspects of the motor function in natural settings. Therefore the two-stage BMI could provide an important and unique new step in developing neuroprosthetics that take advantage of the multiple types of movement information.

## ■ 4.5 Appendix

### ■ 4.5.1 Number of neurons required for accurate target prediction

To find the number of neurons that were sufficient to obtain an accurate target prediction during the delay period, we performed a single neuron analysis in which the spiking activity of a single neuron was used to decode the target. This is illustrated in Figure 4-6. We then sorted the neurons based on their single-neuron accuracies. From the sorted set, we selected different number of neurons and performed the decoding analysis for them. For example the decoding analysis for two neurons was done for the two neurons with the highest single neuron accuracies. Doing so,

we found the target prediction accuracy as a function of the number of top cells included in the decoding. We found that on average across sessions only 17 $\pm$ 6% of the neural ensemble or 3.3 $\pm$ 1.0 neurons were sufficient to obtain a prediction accuracy higher than 90% of the ensemble accuracy.

### ■ 4.5.2 Chance level accuracy

To find the chance level accuracy, we used the least-squares linear regression decoder to estimate the trajectory from shuffled neural activity. To do so we shuffled the calculated firing rates for each neuron across time and trials, keeping its average firing rate the same. We confirmed that across sessions the accuracy of this linear decoder on the shuffled data was 23 $\pm$ 5%.

### ■ 4.5.3 Roughness coefficient

The roughness coefficient for a sequence is an indicator of how smooth it is [59]. For a sequence $x_1, \cdots, x_T$, it is defined as

$$r = \frac{\sum_{t=2}^{T}(x_t - x_{t-1})^2}{\sum_{t=2}^{T}(x_t - \bar{x})^2}$$

where $\bar{x}$ denotes the mean of the sequence. It can also be generalized to the vector case by writing

$$r = \frac{\sum_{t=2}^{T} \| \mathbf{x}_t - \mathbf{x}_{t-1} \|^2}{\sum_{t=2}^{T} \| \mathbf{x}_t - \bar{\mathbf{x}} \|^2}$$

For a trajectory of duration $T$ this provides a measure of smoothness where smaller coefficients correspond to smoother trajectories. In this case $\mathbf{x}_t$ corresponds to the position vector at time $t$ .

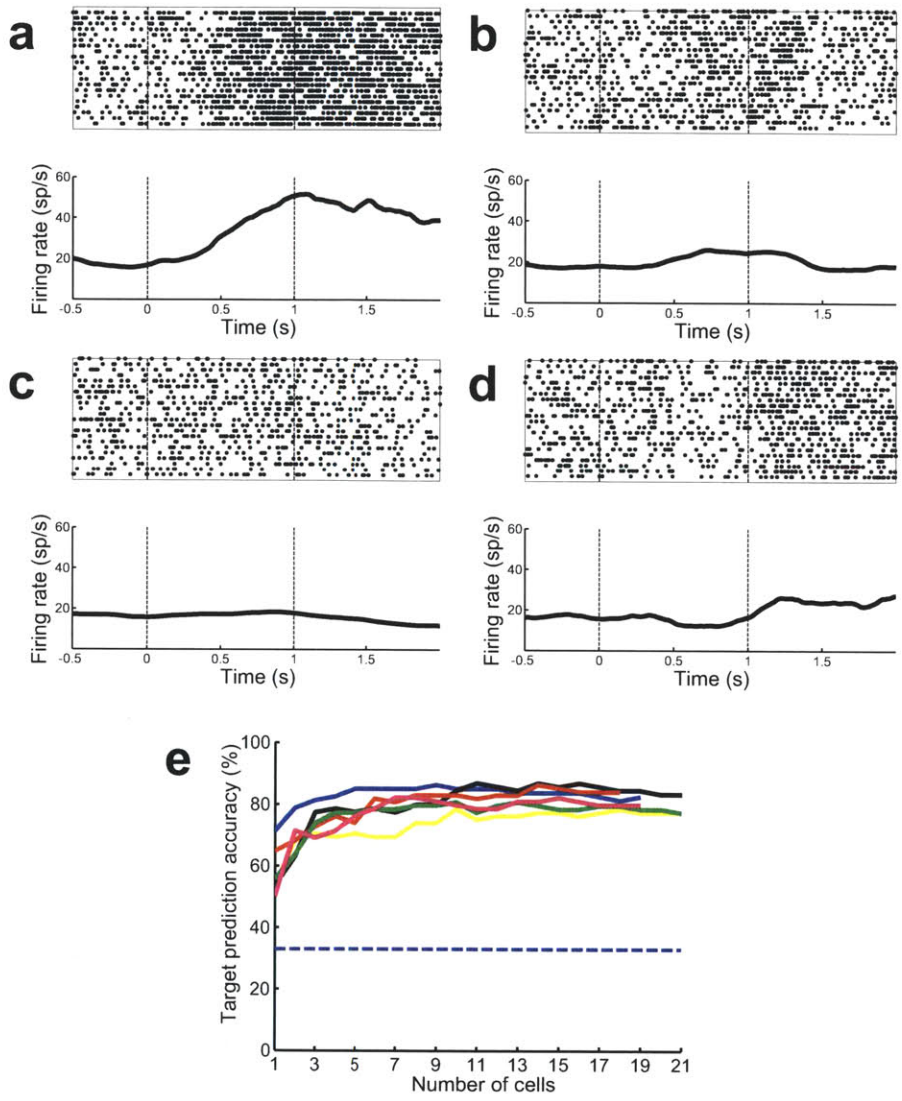


Figure 4-6: Neuron dropping analysis. (a-d) Activity of a single neuron under the four targets in one training session. In the top figure, each row corresponds to a different trial and the black dots indicate the spike times. The bottom figure shows the corresponding firing rate. Activity is aligned to the target presentation time and vertical dashed lines indicate the target presentation and "go" cue times. The target prediction accuracy (leave-one-out cross-validation) of this neuron is 65%. (e) Target prediction accuracy as a function of the number of top neurons. The solid lines show the target prediction accuracy of six training sessions as a function of the number of neurons included in the prediction. Neurons were sorted based on their single neuron accuracy. The dashed line shows the 95% upper confidence bound for the chance level accuracy.

# Chapter 5
# A concurrent brain-machine interface for enhanced motor function

So far we have concentrated on designing a more accurate real-time decoder for estimation of movement aimed at matching natural motor function in individuals with motor deficit. The restoration of lost motor function has been a traditional motivation for the design of BMIs [4–19]. Work to date has largely focused on achieving particularly simple motor goals, such as that associated with the task of moving a cursor on a display to a single target location (see Chapter 4). However, in many natural tasks, the goal is more complex and hence the motor plan for achieving it involves a sequence of such simple plan elements; for example picking up a key and then unlocking a door. Moreover, in addition to the ordering constraint of the plan elements, there may be timing and trajectory constraints.

Here, we shift our focus to the design of BMIs for these sequential plans. In designing a BMI for such plans, one obvious objective would be to decode and execute the intended plan elements one by one, where success is achieved only if all elements are executed, and in the correct order. We refer to this as a *sequential* BMI architecture. For example, one can use the two-stage BMI developed in Chapter 4 sequentially to perform such tasks. However, it is important to emphasize that this architecture corresponds to a narrow interpretation of the overall motor goal of the individual, i.e., one in which the only requirement is to reproduce the elements of the plan one by one and in serial order. As a result, restricting attention to a sequential BMI architecture inherently limits the potential of truly advanced prosthetics. Indeed, from a broader perspective, while a sequential BMI has the potential to restore or match original motor functionality, what is even more compelling is a BMI that can *surpass* such natural motor functionality.

As a basic illustration of the opportunity, consider the task of picking up a key and unlocking a door. While a BMI that reproduces such natural movement would certainly be useful, an ultimately even more valuable BMI would be one that decodes the full sequence of elements of the motor plan simultaneously, then analyzes the sequence to determine ways to perform the task more effectively. For example, the BMI might determine a way for the task to be accomplished more quickly, or more efficiently (to within any physical constraints that might exist). Or, based

on additional sensor inputs, the BMI might determine that the planned sequence of movements would result in an accident, such as the person's hand knocking over an object while reaching for the door, and thus will modify the execution of the task to avoid such an accident.

The realization of such "intelligent" BMIs, which are based on a higher-level interpretation of the motor goal, will obviously require significant technological innovation on a variety of fronts. But most importantly, as the foundation of such an evolution it requires shifting to what we refer to as a *concurrent* BMI architecture. In a concurrent architecture, the elements of a planned motor task are decoded in parallel (i.e., simultaneously), in contrast to the serial process of a sequential BMI. Hence, the feasibility of such intelligent BMIs hinges on the degree to which the elements of a motor plan can, in fact, be decoded in parallel.

In this chapter, we demonstrate that, indeed, it is possible to decode motor plan elements in parallel from neural activity in PMd and SMA of primates. More significantly, we design and implement a real-time BMI with such concurrent plan element decoding, and demonstrate its effectiveness in a series of experiments involving two rhesus monkeys. In our experiments, the monkeys serially add two target locations to working memory and thus the two plan elements consist of reaching movements to the two targets. We show that the real-time BMI can accurately decode both target locations before movement from a single working memory period. Moreover, we show that surprisingly small subpopulations of neurons are sufficient for accurate decoding. This is particularly important for a real-time BMI as the number of neural units captured falls with the age of the implants. These observations have significant implications for future BMI designs aimed at surpassing natural motor function.

## ■ 5.1 Background and main questions

Before describing our experiments and findings in more detail, we briefly summarize the prior work that constitutes the needed background for our investigation. As we have already discussed, individual neurons in the premotor cortex of primates display selective responses to planned single-target movements before their initiation, and such responses often remain sustained during working memory until movement execution [25, 27, 33–37, 39, 40]. Such responses have been successfully exploited in the design of BMIs for simple single-target tasks [18, 19]. We also used these responses in our two-stage BMI in Chapter 4. However, the encoding of more complex tasks requiring a sequence of planned movements is less well understood. Moreover, the design of BMIs for such sequential motor plans has been largely unexplored. While it is known that a

given neuron can have a response that is selective to one or more elements of a sequential motor plan [62–66], it has not been known whether such plans can be accurately decoded from neural populations, in a concurrent manner. Additionally, it is important to understand the structure of the encoding: how the encoding of the motor plan sequence is distributed across neurons, and how it evolves as elements of the plan are added, in sequence, to working memory. Finally, it is important to assess robustness—whether the limitation of a BMI to recordings from a relatively small number of neurons adversely affects decoding accuracy. In this chapter, we investigate the population encoding of multiple motor targets during working memory and show that indeed the premotor population concurrently and accurately encodes multiple targets. Based on this observation, we implement a concurrent BMI architecture for simultaneous decoding of multiple motor targets. In the next chapter, we address the question of how such an accurate encoding is achieved in the premotor areas.

## ■ 5.2 Experimental setup

To investigate whether multiple motor targets may accurately be encoded in the premotor cortex and to implement a concurrent BMI architecture, we conducted a set of experiments in which two rhesus monkeys performed a sequential delayed motor task while neurophysiologic recordings were made from their PMd and SMA. The task required the monkeys to remember two sequentially presented target locations on the computer screen during working memory and then move a cursor to them in the correct order using a joystick. Hence the goal in this task was achieved through two plan elements consisting of the reaching movements to the two remembered targets.

The experiment consisted of two types of sessions, standard recording sessions and BMI sessions. In the standard sessions, the monkey performed the task using the joystick while recordings were made from its PMd and SMA. In the BMI sessions, the monkey first performed the task using the joystick as in the standard sessions. During this training phase, statistical models were constructed for the neural activity. The monkey then performed the same task as before but this time cursor position was controlled by the neural activity recorded during the working memory period as we develop and discuss in Section 5.3.4.

We now explain the details of the behavioral task and neurophysiologic recordings.

### ■ 5.2.1 Task sequence in the standard recording sessions

During the standard recording sessions, two rhesus monkeys (Macaca Mulatta) were trained to perform a sequential delayed motor task in which they were first presented with two distinct

targets and then had to move a cursor to the two remembered targets using a joystick as illustrated in Figure 5-1a. This task was termed the dual-target task. After initial presentation of a blank screen, two targets were sequentially presented each of which was located in one of four possible spatial locations; 'up' (U), 'down' (D), 'right' (R) or 'left' (L). To ensure that the target locations over the two target presentations were distinct, the motor sequence was chosen at random from a total of 12 possible sequences, i.e., all possible combinations of the two targets excluding the ones with repeated target locations. Targets were displayed for 500 ms each and were interleaved by a 300 ms interval during which a blank screen was shown. Following the second target presentation, there was an additional variable delay of 1000-1300 ms following which a "go" cue signal appeared indicating that the monkeys could move.

After presentation of the first "go" cue, the monkeys were required to move a cursor from the center of the screen to the first remembered target. After reaching the target, they were required to return the joystick to the center and then wait for a second "go" cue to appear after an additional 500 ms delay interval. Once the second "go" cue appeared, they were allowed to move the cursor from the center of the screen to the second remembered target. The monkeys received a juice reward if they correctly moved to the two instructed targets.

### ■ 5.2.2 Task sequence in the BMI sessions

The behavioral task for the BMI sessions was the same as the dual-target task except that the "go" cue always appeared 1000 ms after the start of second target presentation (i.e., without the added variable delay).

### ■ 5.2.3 Multi-electrode array implantation

All procedures were performed under IACUC-approved guidelines. Prior to recordings, multiple recording silicone multi-electrode arrays (NeuroNexus Technologies Inc., MI) were surgically implanted in each monkey. A craniotomy was placed along the expected location of premotor cortex under stereotactic guidance (David Kopf Instruments, CA). The multi-electrode arrays were implanted into the SMA and PMd. The electrode lead of each array was secured to the skull and attached to female connectors with the aid titanium miniscrews and dental acrylic. After implantation, confirmation of electrode positions was done in both monkeys by direct visual inspection of the sulci and gyral pattern. Additional post-mortem confirmation of electrode positions was made in one monkey (the second monkey is still performing experiments).

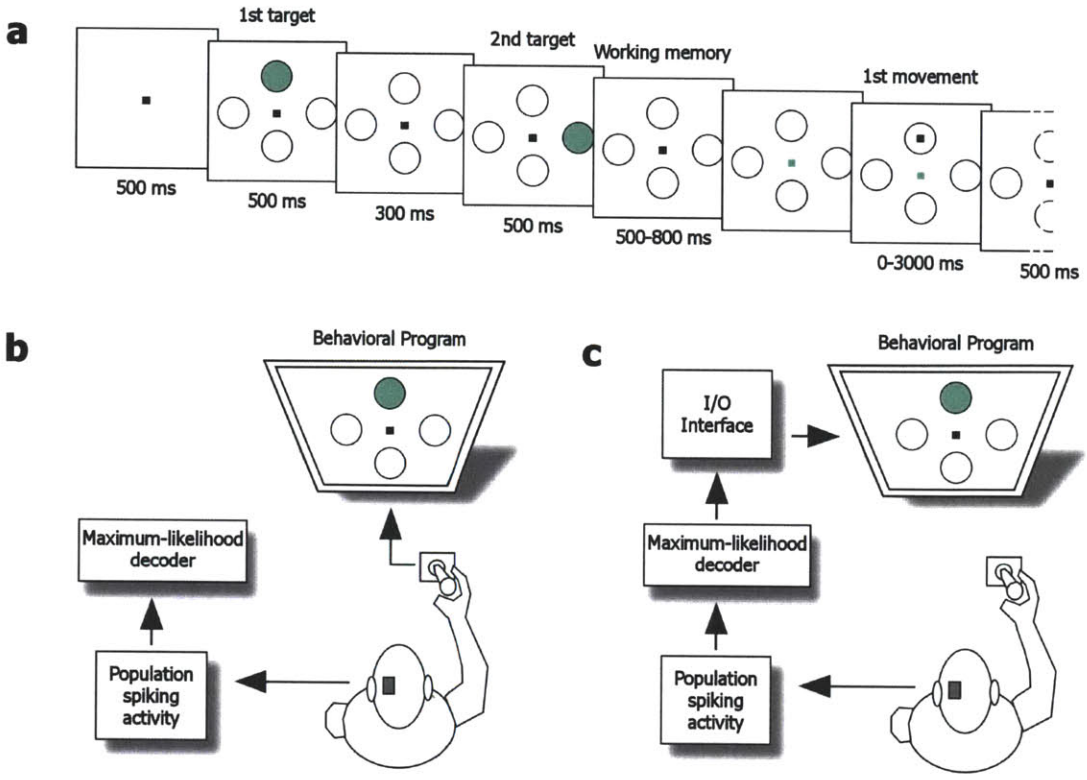


Figure 5-1: Task design and experimental setup. (a) Schematic illustration of a standard dual-target task over a single trial. Task periods and their timings are displayed over a single trial from left to right. The right end of the panel in which the second movement is made is truncated for space. (b) Schematic illustration of the experimental setup for standard sessions and training sessions. (c) Schematic illustration of the brain-machine interface sessions.

### ■ 5.2.4 Single-unit isolation and recordings

Recordings began two weeks following surgical recovery. A Plexon multichannel acquisition processor was used to amplify and band-pass filter the neuronal signals (150 Hz – 8 kHz; 1 pole low-cut and 3 pole high-cut with 1000x gain; Plexon Inc., TX). Shielded cabling carried the signals from the electrode array to a set of six 16-channel amplifiers. Signals were then digitized at 40 kHz and processed to extract action potentials in real time by the Plexon workstation. Classification of the action potential waveforms was performed using template matching and principle component analysis based on waveform parameters. Only single-, well-isolated units with identifiable waveform shapes and adequate refractory periods (less than 1% of spikes within a 1 millisecond interval) were used for the online experiments and offline analysis. No multiunit activity was used.

## 5.3 Methodology

Our goal in these experiments is to quantify the accuracy by which the full sequence of targets can be decoded from the population neural activity in any given period in time during the task. To do so, we first build statistical point process models for the ensemble spiking activity and then use a maximum-likelihood decoder to decode the full sequence based on this activity, as we now develop.

### 5.3.1 Building the encoding models for the standard recording sessions

We model the activity of each neuron under any given sequence during the standard recording sessions as an inhomogeneous Poisson process whose likelihood function is given by [50, 67]

$$p(N_{1:T}^c|S_i) = \prod_{t=1}^{T}(\lambda_c(t|S_i)\Delta)^{N_t^c}\exp(-\lambda_c(t|S_i)\Delta) \qquad i = 1,\cdots,12 \tag{5.1}$$

where $\Delta$ is the time increment taken to be small enough to contains at most one spike, $N_t^c$ is the binary spike event of the $c$th neuron in the time interval $[(t-1)\Delta, t\Delta]$, $\lambda_c(t|S_i)$ is its instantaneous firing rate in that interval, $S_i$ is the $i$th sequence, and $T$ is the total number of bins in a duration $T\Delta$. We take $\Delta = 5$ ms as the bin width of the spikes. By building the neuronal models under each sequence separately in the dual-target task we avoid making any a priori assumptions about whether the cells encode individual targets or combined sequences.

For each sequence and neuron, we need to estimate the firing rate, $\lambda_c(t|S_i)$, using the neuronal data observed. One way to do so is to bin the data into non-overlapping windows of fixed length during which the firing rate is assumed to be constant and estimated using maximum likelihood techniques. This method is equivalent to finding the peristimulus time histogram (PSTH) that simply averages the number of spikes over any given window. The main drawback of this technique is that unless there are a large number of training trials under each sequence, to get a good estimate one has to pick relatively large windows. This in turn masks the fine-scaled evolution of the firing rate. Also, there is no principled way for selecting a window size, which the analysis is dependent on. One way to avoid these problems in estimating the spike rate function is to use a state-space approach [68, 69]. This approach is used in many applications to estimate an unobservable state process and consists of two models: A prior or state model that in general enforces any prior information available about the unobservable states—such as a simple continuity condition—and an observation model that relates the neuronal observations to these states.

In the case of estimating the spike rate function, and since it is a non-negative quantity, similar to previous work [68, 69] we take the state at time increment $t$ , $x_t$, to be the logarithm of the firing rate, i.e., $x_t = \log(\lambda_c(t|S_i))$, or equivalently

$$\lambda_c(t|S_i) = \exp(x_t), \tag{5.2}$$

and enforce a continuity condition on it by assuming that it evolves according to a linear first-order Gaussian model [68, 69]

$$x_t = x_{t-1} + \epsilon_t,$$

where $\epsilon_t$ is the zero-mean white Gaussian noise with variance $\sigma_\epsilon^2$. The observation model is in turn given by substituting (5.2) in (5.1). Here, $\theta = \sigma_\epsilon^2$ is an unknown parameter of the model and should be estimated jointly with the state. Hence we use the expectation-maximization (EM) iterative algorithm to find the maximum likelihood estimate of $\theta$ and in turn estimate the firing rate [68–70]. Denoting the estimate of $\theta$ in the $i$th iteration by $\theta^{(i)}$, its estimate in the $i+1$th iteration after the maximization step is given by

$$\theta^{(i+1)} = \frac{1}{T}\left(\sum_{t=1}^{T} \mathbf{W}_t + \mathbf{W}_{t-1} - 2\mathbf{W}_{t,t-1}\right) \tag{5.3}$$

where $\mathbf{W}_t = E\left[x_t^2 | N_{1:T}^c; \theta^{(i)}\right]$ and $\mathbf{W}_{t,t-1} = E\left[x_{t-1}x_t | N_{1:T}^c; \theta^{(i)}\right]$ are found from the forward filter, fixed-interval smoothing, and covariance recursive algorithms in the expectation step as follows. Assuming that there are $J$ total trials and denoting the causal filter state estimate by $x_{t|t} = E\left[x_t | N_{1:t}^c; \theta^{(i)}\right]$ and its variance by $w_{t|t}$ , and the smoothed state estimate by $x_{t|T} = E\left[x_t | N_{1:T}^c; \theta^{(i)}\right]$ and its variance by $w_{t|T}$ , the recursions in the E-step are given by the forward filter recursions [49, 68, 69]

$$\begin{aligned}
w_{t|t-1} &= w_{t-1|t-1} + \theta^{(i)} \\
x_{t|t-1} &= x_{t-1|t-1} \\
w_{t|t} &= \left(w_{t|t-1}^{-1} + J\exp(x_{t|t-1})\Delta\right)^{-1} \\
x_{t|t} &= x_{t|t-1} + w_{t|t}\sum_{j=1}^{J}\left(N_t^c(j) - \exp(x_{t|t-1})\Delta\right)
\end{aligned}$$

for $t = 1, \cdots, T$ where $N_t^c(j)$ is the spike event in trial $j$, and by the fixed interval smoothing recursions [46,68,69]

$$\begin{aligned} A_t &= w_{t|t} w_{t+1|t}^{-1} \\ x_{t|T} &= x_{t|t} + A_t \left( x_{t+1|T} - x_{t+1|t} \right) \\ w_{t|T} &= w_{t|t} + A_t^2 \left( w_{t+1|T} - w_{t+1|t} \right) \end{aligned}$$

for $t = T-1, \cdots, 0$ and with initial condition $x_{T|T}$ and $w_{T|T}$ from the filter recursions. We pick the initial conditions for the forward filter at each iteration of the EM algorithm as $x_{0|0}^{(i+1)} = x_{0|T}^{(i)}$ and $w_{0|0}^{(i+1)} = w_{0|T}^{(i)}$. Finally the state-space covariance algorithm gives all the terms needed for the M-step to find $\theta^{(i+1)}$ in (5.3) using these recursions [68,69,71]

$$\mathbf{W}_t = w_{t|T} + x_{t|T}^2$$

for $t = 0, \cdots, T$ and

$$\mathbf{W}_{t+1,t} = A_t w_{t+1|T} + x_{t|T} x_{t+1|T}$$

for $t = 0, \cdots, T-1$. The iterations of the EM algorithm are run until convergence. The estimated firing rate at any time bin $t = 1, \cdots, T$ is in turn the smoothed estimate, $\hat{\lambda}_c(t|S_i) = \exp(x_{t|T})$ evaluated at the estimate of $\theta$ in the final iteration.

Repeating this procedure for all neurons under each sequence and fitting the inhomogeneous Poisson models results in a continuous smoothed estimate of the rate function for each neuron under any given sequence and over the entire length of a trial. Our implementation of the EM algorithm is similar to prior work [68,69]. After fitting the models, we validated them using the $\chi^2$ goodness-of-fit test on the data [69] and confirmed that they fitted the data well ($P > 0.7$ for all cells in all sessions).

### ■ 5.3.2 Building the encoding models for the BMI sessions

In each BMI session, the monkeys first performed the BMI task using a joystick as shown in Figure 5-1b (training session) during which models were constructed for the neuronal activity during the 800 ms time window prior to presentation of the "go" cue. As we will discuss in Section 5.4, this window length was chosen because it achieved higher than 95% of the maximum accuracy

possible over the second target to working memory period. We modeled the activity of each neuron in this 800 ms window under any sequence as a homogeneous Poisson process, instead of an inhomogeneous one used in the standard dual-target sessions, to make the model construction faster for the BMI experiments. Hence here the likelihood function for the spiking activity of neuron $c$ under any of the sequences, $S_i$ , was modeled as

$$p(N^c_{1:T}|S_i) = \prod_{t=1}^{T}(\lambda_c(S_i)\Delta)^{N^c_t}\exp(-\lambda_c(S_i)\Delta) \tag{5.4}$$

where $\lambda_c(S_i)$ denotes the fitted firing rate of that neuron in the 800 ms delay period for sequence $S_i$ and $T = 800/\Delta$ is the total number of bins in this period with bin width $\Delta = 5$ ms. The firing rate for each neuron and under each sequence was fitted using maximum likelihood parameter estimation. These trained Poisson models were later used in the real-time BMI sessions to predict the sequence using the maximum-likelihood decoder as we will develop next.

Each training session consisted of $26 \pm 2$ trials per sequence on average. Here, the task involved either four (both monkeys) or eight sequences (monkey P). The four sequence task consisted of either U-R, U-L, D-R, D-L or L-U, L-D, R-U, R-D. The eight sequence task consisted of the union of the sequences in the two four sequence tasks. To assess the accuracy of the trained models, we performed a leave-one-out cross-validation analysis on the training data using the maximum-likelihood decoder developed in Section 5.3.3.

### ■ 5.3.3 Simultaneous decoding of the sequence of targets: A maximum-likelihood decoder

Once models are fitted, a maximum-likelihood decoder is used to decode the intended sequence based on the neural activity in any period of interest. A maximum-likelihood decoder is the optimal decoder in the sense of maximizing accuracy, i.e., the percentage of trials in which the combined sequence is decoded correctly, when the sequences are equally likely to be presented as is the case in our experiments. The decoder finds the likelihood of observing the population neuronal data under each sequence and selects the sequence with the highest likelihood as its prediction. Using the likelihood model in (5.1) and assuming that neurons are conditionally independent given the sequence, the population likelihood under any sequence is given by

$$p(N^{1:C}_{1:T}|S_i) = \prod_{c=1}^{C}\prod_{t=1}^{T}(\lambda_c(t|S_i)\Delta)^{N^c_t}\exp(-\lambda_c(t|S_i)\Delta) \qquad i = 1, \cdots, 12 \tag{5.5}$$

where $T$ is the total number of bins in any period of interest during the trial, $C$ is the total number of neurons, and $\lambda_c(t|S_i)$ for $t = 1, \cdots, T$ and $c = 1, \cdots, C$ is the smoothed estimate of the firing rate of the cells in that period. The predicted sequence, $\hat{S}$, is thus given by

$$\hat{S} = \arg\max_{S_i} p(N_{1:T}^{1:C}|S_i)$$

To find the decoding accuracy of a single cell or a subset of cells, the maximum-likelihood decoder uses only that cell or subset's spiking activity to decode the sequence as illustrated in Figure 5-2.

The decoder also outputs the posterior probability of each sequence, which is the probability that it is the correct one after the neuronal observations, i.e.,

$$p(S_i|N_{1:T}^{1:C}) = \frac{p(N_{1:T}^{1:C}|S_i)}{\sum_i p(N_{1:T}^{1:C}|S_i)}. \qquad i = 1, \cdots, 12$$

To dissociate the decoding accuracy of the first and second targets, denoted by $G_1$ and $G_2$, the decoder also outputs their predictions based on the neuronal activity. To do so, the decoder finds their posterior probabilities, i.e., $p(G_1 = l_1|N_{1:T}^{1:C})$ and $p(G_2 = l_2|N_{1:T}^{1:C})$, for all possible spatial locations, $l_1$ and $l_2$, by summing over the posterior probability of the sequences that have these spatial locations as their first or second targets. The decoder then picks the spatial location with the highest first target (second target) posterior, or equivalently likelihood in our design, as its first (second) target prediction.

### ■ 5.3.4 Concurrent prediction of two movements and their execution in the BMI

During the BMI experiments and after the training session, the monkeys performed the same task as before. However, this time, cursor position was controlled by target predictions made by the maximum-likelihood decoder rather than the joystick. During the real-time BMI experiments, individual spike timings of all cells within the population were continuously recorded at a 40 kHz resolution by the Plexon multi-channel acquisition processor. Each recorded spike was then transmitted via an Ethernet port to a separate computer running a Matlab routine in real time. For each real-time trial, the Matlab routine then used the maximum-likelihood decoder to calculate the likelihood of the population spiking activity during the 800 ms time window prior to presentation of the "go" cue, i.e., $N_{1:T}^{1:C}$, under each sequence, $S_i$, assuming neurons were independent conditioned on the sequence. This likelihood was calculated based on the trained Poisson models, i.e., the fitted firing rates from the training sessions, $\lambda_c(S_i)$. Hence the population likelihood for

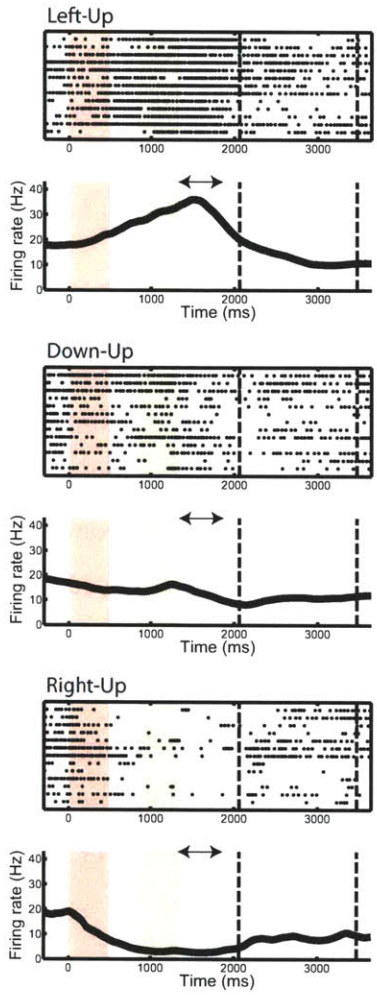


Figure 5-2: Maximum-likelihood decoder. Spiking activity of a sample cell under three sample sequences with a common second target but different first targets. In the top panels each row corresponds to a different trial and the black dots indicate the spike times. The bottom panels show the corresponding firing rate estimates using the EM procedure. Time at zero is aligned to the start of first target presentation. The red and green vertical bars indicate the periods during which the first and second targets were presented, respectively. The first and second dashed black lines indicate the mean times on which the first and second "go" cues were presented, respectively. The arrow indicates the working memory period. The maximum-likelihood decoder computes the likelihood of the spiking activity of a cell during the working memory period for each trial and under each of the 12 possible sequences using the cell's rate estimate for that sequence (bottom panels). It then selects the sequence with the highest likelihood as its single cell prediction for that trial. As is evident from the variations in the neuronal activity between the three sample sequences for the represented cell, it is selective to the first target. It has a first target accuracy of 65% which is significantly higher than chance.

each sequence was found as

$$p(N_{1:T}^{1:C}|S_i) = \prod_{c=1}^{C}\prod_{t=1}^{T}(\lambda_c(S_i)\Delta)^{N_t^c}\exp(-\lambda_c(S_i)\Delta)$$

where $T = 800/\Delta$ is the total number of bins during the 800 ms time window prior to presentation of the "go" cue. The maximum likelihood decoder then outputted the sequence under which the population likelihood was maximized as the predicted sequence, i.e.,

$$\hat{S} = \arg\max_{S_i} p(N_{1:T}^{1:C} | S_i)$$

where $\hat{S}$ denotes the predicted sequence.

Based on the sequence decoded from the recorded population, a second Matlab routine running on the same computer then activated an analog output channel on the NI DAQ I/O interface to go from 0V to either +5V or -5V for 500 ms. The voltage line was, in turn, connected to a second NI DAQ I/O input channel located on a third computer running the behavioral program. Depending on the voltage received, the cursor displayed in the middle of the screen moved in a straight line to one of the four possible target locations (e.g. +5V corresponded to a cursor location within the top target). This process then repeated for the second decoded target after another artificially introduced time delay. Here, the time delays in the two generated movements were selected similar to those that the monkeys normally experienced if they were performing the standard task using a joystick. As described below, however, the I/O card could in principle generate the two movements as fast as 1 millisecond apart.

### ■ 5.3.5 Target correlations and chance level accuracy

To investigate the statistical significance of the decoding accuracy in the recorded neural data for the sequence or either target, we first need to calculate their chance level accuracy. For the sequence, the chance level accuracy is simply $1/S$, where $S$ is the number of sequences used. For targets, however, one has to take into account the correlation between the first and second targets when calculating the chance level accuracy. This is because depending on the number of sequences used in the decoding analysis, the first and second decoded targets may not be independent. In the case of 12 sequences, for example, since both targets cannot be at the same location, information about one also implies some information about the other. These correlation effects must therefore be taken into account when calculating the chance level accuracy of the targets.

We define indicator functions for the first and second targets, denoted by $I_1$ and $I_2$ , that are 1 if the corresponding targets are decoded correctly and 0 otherwise. We show this analysis for the case when 12 sequences are used. In all other cases they can be found similarly. For 12 sequences, using the total law of probability, the probability that the second target is decoded correctly is

given by,

$$p(I_2 = 1) = p(I_2 = 1|I_1 = 1)p(I_1 = 1) + p(I_2 = 1|I_1 = 0)p(I_1 = 0)$$

Now if the first target is correctly decoded, the second target could be at one of three possible locations as the two targets cannot be at the same location. Hence the chance level accuracy in this case is given by $p(I_2 = 1|I_1 = 1) = 1/3$ . By a similar argument, if the first target is decoded incorrectly, the chance level accuracy of the second target is $p(I_2 = 1|I_1 = 0) = 2/9$ . Hence the chance level accuracy of the second target is given by

$$p_2^{\text{chance}} = \frac{2}{9} + \frac{1}{9}p(I_1 = 1) \tag{5.6}$$

and vice versa for the first target as the two targets are selected symmetrically in the choice of sequences. For example, in a session where we observe a first target accuracy of $p^*$ in our decoding analysis, to assess the significance of the second target accuracy we should compare to $p_2^{\text{chance}} = 2/9 + p^*/9$ as opposed to simply $1/4$, using a binomial test or a one-sided $z$-test for a Bernoulli process. Note that if $p^* = 1/4$, i.e., at chance level, then $p_2^{\text{chance}} = 1/4$ and also at chance level as expected.

## ■ 5.4 Results

We started this chapter with the goal of designing a *concurrent* BMI architecture that enables the simultaneous decoding of multiple plan elements required to achieve a motor goal. Hence, the feasibility of such a BMI hinges on the degree to which the elements of a motor plan can, in fact, be decoded in parallel.

Here we present the analysis results for both the standard recording sessions and the BMI sessions demonstrating that the two plan elements of the sequential movement in our experiments can indeed be accurately decoded from a single working memory window before movement initiation.

### ■ 5.4.1 Simultaneous decoding of the full sequence from working memory

In the standard sessions, we defined the working memory period as the 500 ms blank-screen delay interval following the second target presentation. Multiple-unit responses were recorded from the premotor cortex as the primates performed the dual-target task. Statistical models were

fitted (trained) to each neuron's spiking activity as developed in Section 5.3.1 and a maximum-likelihood decoder was used to quantify the probabilities that an ensemble of neurons could correctly identify the first and second targets on a trial-by-trial basis during the working memory period as developed in Section 5.3.3. We used leave-one-out cross-validation for all the decoding analysis in the standard sessions. We measured the amount of information held by an ensemble of neurons about the first or second target in terms of the percentage of trials in which the maximum-likelihood decoder predicted that target correctly from their activity; i.e. the decoding accuracy. For the full sequence of targets, we defined the sequence decoding accuracy as the percentage of trials in which both targets were decoded correctly.

We find that neural population activity within the premotor cortex accurately encoded both targets during the working memory period. We recorded 281 neurons from the premotor cortex (SMA and PMd) over 11 sessions, for an average of $26 \pm 6$ cells per recording session. During the working memory period and across all sessions, the population correctly encoded the first and second targets on $76 \pm 11\%$ and $56 \pm 17\%$ of trials respectively, both of which were significantly above chance (one-sided z-test, $P < 10^{-15}$). When considering all 12 possible target combinations, the population encoded both targets correctly on average on $45 \pm 12\%$ of the trials, which was also far higher than chance at $1/12 \approx 8\%$ ($P < 10^{-15}$). The time evolution of the average decoding accuracy for the sequence and targets across all sessions is shown in Figure 5-3a. For the best session tested, accuracies were as high as 85% for the first target, 82% for the second target, and 72% for both targets as illustrated in Figure 5-3b.

We further found the decoding accuracy across subsets of the full 12 sequences, each consisting of 4 or 8 sequences. For 4 sequences, the average sequence decoding accuracy was as high as 93% and for 8 sequences, it was as high as 80% in the best session. The corresponding first and second target accuracies in the best session for the 4 sequence set were 100% and 93% and for the 8 sequence set were 84% and 90%.

These results illustrate that population activity during the 500 ms working memory period can simultaneously predict the full sequence of targets with high accuracy. They thus indicate that a concurrent BMI architecture based on parallel decoding of plan elements can indeed be implemented.

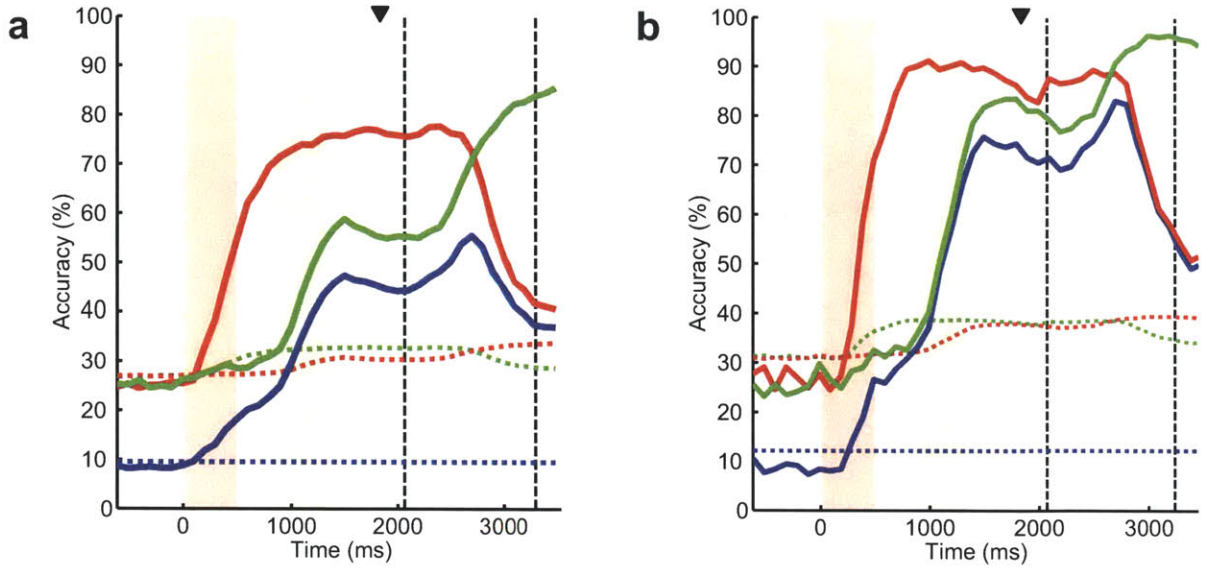


Figure 5-3: Population decoding accuracies. Population decoding accuracy for the first target (red), second target (green) and the full sequence (blue) over time averaged over all recorded sessions in (a) and for the best session in (b). Each point on the curves indicates the decoding accuracy using the neuronal activity in the preceding 500 ms window. Time at zero is aligned to the start of first target presentation. The red and green vertical bars indicate the times during which the first and second targets were presented, respectively. The first and second dashed black lines indicate the mean times on which the first and second "go" cues were presented, respectively. The arrow indicates the time point corresponding to the decoding accuracy of the preceding working memory period (0-500 ms from the end of second image presentation). Horizontal dotted lines indicate the 99% chance upper confidence bounds for the first target, second target and sequence (out of 12 possibilities) with the same color scheme as above.

### ■ 5.4.2 Number of cells required for accurate decoding of the full sequence

To implement a concurrent BMI with robust performance over time, in addition to obtaining a high population decoding accuracy, we need this high accuracy to be achievable using relatively few cells. This is because the number of units captured in a BMI is limited. Furthermore, fewer units get captured as the implants age. Therefore, having found that the population activity could accurately predict the full sequence during working memory, we further investigated the number of cells that were required for such an accurate decoding.

We found the number of cells required to achieve a given percentage of the population accuracy by first sorting them in each session based on their single neuron sequence accuracy. To find the single neuron sequence accuracy of each cell, we used only its activity in the maximum-

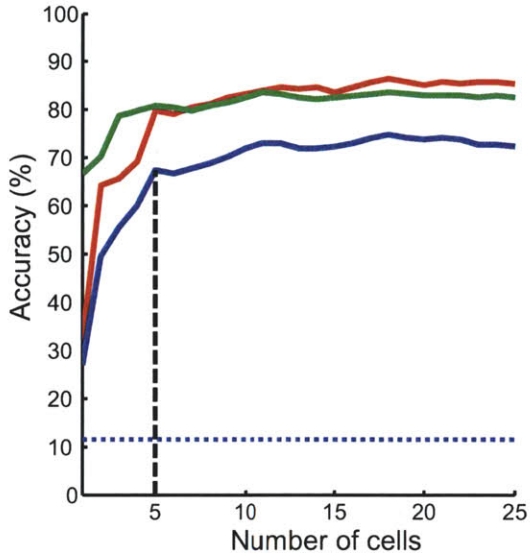


Figure 5-4: Number of cells needed to reach the sequence decoding accuracy asymptote. First target (red), second target (green) and sequence (blue) accuracies as a function of the number of cells in descending order of single cell accuracy in one sample session. The number of cells necessary to reach over 90% of the population accuracy in this session is indicated by the vertical dashed line. This procedure was repeated for all sessions to find the average number of cells necessary to reach the sequence accuracy asymptote across the sessions.

likelihood decoder to decode the sequence. We then performed the decoding analysis in each session for different number of cells in descending order of single neuron accuracy. We hence found the sequence accuracy as a function of the number of cells in descending order included in the decoding, for example for the top cell, top two cells, etc.

We found that only a small number of cells were needed to predict the full sequence of targets with high accuracy. When considering all 12 possible target combinations, on average across sessions only 29% of the population (7.5 cells) was sufficient to achieve higher than 90% of the population sequence accuracy. This is shown for one session in Figure 5-4. We can see that as the number of cells increases, the sequence decoding accuracy has a steep rise and reaches an asymptote value. When performing the decoding analysis over subsets of all trials, which employed only 4 or 8 target combinations, population sequence accuracies in the best session were as high as 93% and 80%, respectively. In these cases, decoding from only 2 and 4 cells, respectively, was sufficient to achieve higher than 90% of these sequence accuracies.

These results illustrate that relatively few neurons could result in an accurate prediction of the full sequence, further motivating the design of a concurrent BMI for simultaneous decoding of both targets.

### ■ 5.4.3 A real-time concurrent BMI

Having observed that a relatively small number of neurons in the premotor cortex encoded sufficient information to concurrently and accurately predict both targets from the working memory period motivated the design of a real-time concurrent BMI to predict both targets simultaneously.

In an additional set of real-time BMI experiments, we recorded a mean of 21 $\pm$ 2 cells per session over 4 sessions from the premotor of the same monkeys. Here, statistical models were first trained on the population activity as the primates rehearsed a subset of target combinations that included either four or eight possible sequences over an average of 26 $\pm$ 2 training trials per sequence as described in Section 5.3.2 and illustrated in Figure 5-1b. In these experiments we used the activity in the 800 ms window prior to the "go" cue to predict both targets; in the standard dual-target sessions, this window size was sufficient to achieve better than 95% of the (maximum) sequence accuracy possible when using the entire window starting from second target presentation until the earliest "go" cue. This is illustrated in Figure 5-5a. Using the maximum-likelihood decoder, the leave-one-out cross-validation sequence decoding accuracies for the set of four and eight sequences were 79 $\pm$ 2% and 80 $\pm$ 3% (mean $\pm$ s.e.m., $P < 10^{-15}$, $P < 10^{-15}$), respectively, during the training sessions. Following training, the primates performed the same task as before, but with the cursor now being sequentially positioned by the BMI on the targets decoded from the recorded neuronal activity during the preceding working memory period as illustrated in Figure 5-1c and discussed in detail in Section 5.3.4. Here, the delays in the BMI-generated cursor movement were selected to approximate the same time delays that the monkeys normally experienced when moving the cursor themselves following the "go" cue (although, in practice, cursor movement could be generated without the added delay as discussed below).

Both monkeys performed a total of 459 trials on the four-sequence set, and one monkey performed 110 trials on the eight-sequence set using the real-time BMI. Sequence accuracies for the set of four and eight sequences were 72 $\pm$ 2% and 71 $\pm$ 4%, respectively, both of which were significantly above chance (mean $\pm$ s.e.m., $P < 10^{-15}$, $P < 10^{-15}$). Sequence accuracies using the BMI were also close to the cross-validated sequence accuracies during the training sessions when taking into account the primates' natural error rates during the standard task. In fact the 95% con-

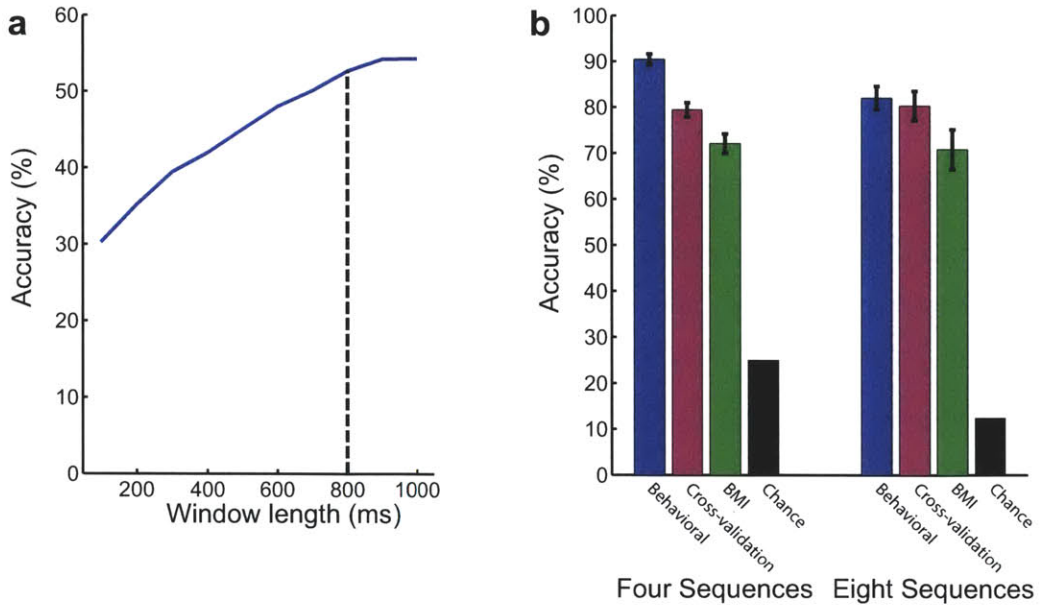


Figure 5-5: Performance on brain-machine interface trials. (a) Average sequence decoding accuracy over all the standard recording sessions (out of 12 possibilities) as a function of the time window preceding the earliest "go" cue used in the decoding. Using an 800 ms window the sequence decoding accuracy exceeds 95% of the maximum possible when using the neuronal activity from the start of second target presentation until the earliest "go" cue (1000 ms period). (b) Bar graph indicating the monkeys' behavioral performance (blue), maximum-likelihood cross-validation performance on the training data (magenta), real-time BMI performance (green), and chance performance (black) and also the corresponding s.e.m error bars for four-sequence (left) and eight sequence (right) sessions.

fidence bounds for the two accuracies were overlapping (72 $\pm$ 4% vs. 73 $\pm$ 4% and 71 $\pm$ 8% vs. 66 $\pm$ 6%, for the sets of four and eight sequences, respectively, found as explained in the Appendix). Results are summarized in Figure 5-5b.

We therefore demonstrated the first realization of a concurrent BMI architecture in these experiments. Our results illustrate that the two plan elements for the dual-target task, i.e., the two intended sequential targets of movement, could be simultaneously predicted and executed in real-time with high accuracy.

Decoding all the elements of the motor plan prior to movement allows the BMI to analyze the full plan and hence find more effective ways to execute the movements required to achieve the overall goal. For example, it can perform the two movements much faster than possible by natural sequential movement. While we purposefully introduced time delays in the BMI-generated movements to make it appear to the monkeys that they were performing the standard task, having

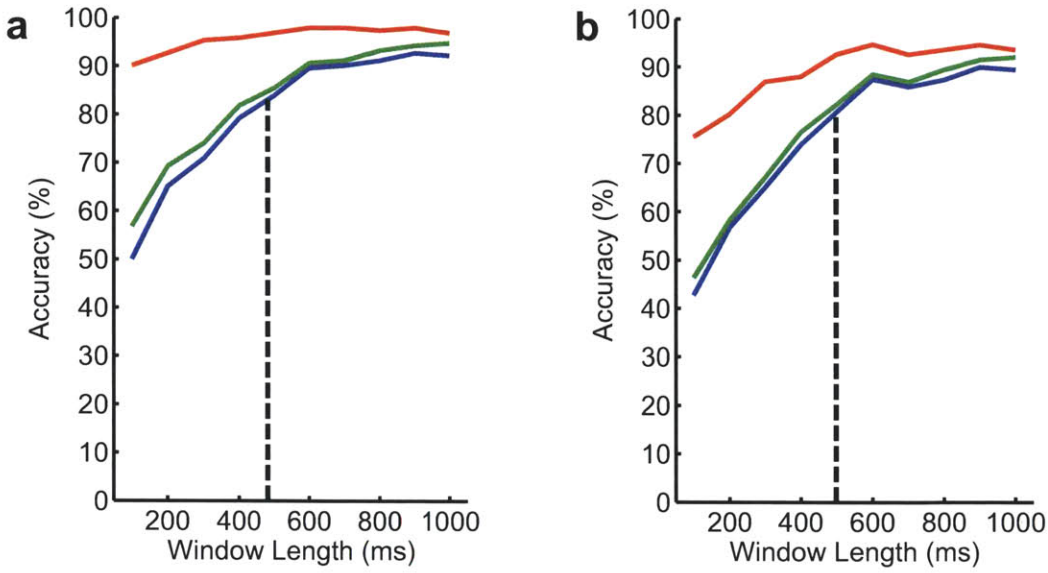


Figure 5-6: Fast acquisition of the two targets. Average decoding accuracy for the 4 sequence sets (a) and the 8 sequence set (b) in a sample session for the first target (red), second target (green) and the full sequence (blue) as a function of the time window used for decoding, following the second target presentation. The dashed vertical line shows the window size for which the sequence decoding accuracy reaches 90% of the maximum accuracy possible using the entire 1000 ms second target presentation to "go" cue interval. Hence the full sequence can be decoded as early as the time indicated by the dashed lines and immediately executed.

concurrent information about the full sequence allows the BMI to acquire both intended targets much faster than possible through natural sequential movement. Figure 5-6a-5-6b show the time required to achieve higher than 90% of the maximum sequence accuracy—obtained using the activity in the entire period from the second target presentation to the "go" cue—for the sets of 4 and 8 sequences in one of the standard sessions. On average across sessions, the full sequence in the 4-sequence and 8-sequence sets could be decoded with an accuracy higher than 90% of the maximum as early as 474 ms and 557 ms after the start of the second target presentation, respectively, and far before either of the two planned movements could be physically executed. Once this accurate decoding is obtained, the BMI could move the cursor almost instantaneously to the two targets.

These results therefore illustrate the possibility of concurrently decoding multiple plan elements to achieve an overall motor goal, which in turn enables the development of more intelligent BMIs aimed at enhancing natural motor function.

## 5.5 Discussion

Achieving a complex motor goal often requires the execution of a sequence of plan elements, such as picking up a key and unlocking the door. In this chapter, we proposed a concurrent BMI architecture for execution of such sequential movements or plan elements. While an obvious choice to execute such sequential plans is to match the natural sequential function by decoding and executing them one by one and in serial order, decoding these elements concurrently enables a BMI to *surpass* such natural function. This can be achieved by analyzing the entire plan and finding ways to execute the movements more effectively.

The development of such concurrent BMIs therefore depends on the ability to decode in parallel multiple plan elements from the motor cortical areas during working memory. We therefore investigated whether such accurate concurrent decoding could be achieved in a set of delayed sequential tasks. In these tasks monkeys were required to keep two motor targets in working memory and then reach for them in the correct order. The plan elements in these tasks therefore consisted of the two reaching movements to the two targets.

We demonstrated that indeed an accurate and concurrent prediction of both motor targets was possible during working memory and before movement execution. Further we found that a relatively few cells were sufficient to achieve a high accuracy for the full sequence. These observations, which are both necessary for the development of robust BMIs, motivated the design of a real-time concurrent BMI that predicted both targets simultaneously from a single working memory period and before movement execution. We implemented the BMI in two rhesus monkeys and showed that it could accurately decode the full sequence in real time.

Having access to the entire plan for achieving an overall goal before movement initiation enables a BMI to analyze all its elements and hence find the most effective way to execute them. As a simple but illustrative example of the enhanced motor function such concurrent decoding enables, in the context of our experiments it would be possible for the BMI to move the cursor through the sequence of two targets much faster than possible by natural movement—indeed, effectively instantaneously. More specifically, the full sequence could be accurately decoded within approximately 500 ms after presentation of the second target—well before either of the two planned movements could be physically executed.

These results thus motivate a shift towards *concurrent* BMI architectures. These architectures will result in more "intelligent" BMIs, which are based on a higher-level interpretation of the motor goal and aim to enhance the natural motor function.

## 5.6 Appendix

### 5.6.1 Comparison of the BMI sequence accuracies with cross-validated sequence accuracies during the training sessions

Since the primates do not perform the dual-target task with 100% behavioral accuracy, to compare the BMI accuracy with the accuracies during training sessions we need to take into account the primates' natural error rates. In other words, during the cross-validation analysis we calculate the sequence accuracy by comparing the decoded sequence with the sequence the monkeys actually select after the "go" cue, which in some cases is different from the instructed sequence due to monkeys' behavioral errors. For the BMI sessions, however, we compare the decoded sequence to the instructed sequence to find the accuracy. Hence some of the errors in the BMI are due to behavioral errors and not decoder errors. Denoting the behavioral accuracy of the monkeys by $P_b$ and the decoder accuracy found from the training session by $P_t$ we can calculate what the accuracy of selecting the instructed sequence would be after taking into account the behavioral errors. Denoting the resulting accuracy by $P_f$ we have that

$$P_f = P_b P_t + (1 - P_b)(1 - P_t) \times \frac{1}{S - 1}$$

In other words, when the monkey and the decoder are both correct, the instructed sequence is selected. However, if the monkey is incorrect and the decoder is also incorrect in decoding the monkey's intended sequence, the probability of the decoder selecting the correct instructed sequence by random chance is $1/(S-1)$ . We can find the s.e.m. of $P_f$ from those of $P_b$ and $P_t$ using Monte-Carlo simulations and then compare it with the BMI accuracy.

# Chapter 6
# Neural partitioning during working memory

In Chapter 5 we found that multiple plan elements are concurrently encoded in PMd and SMA of primates and implemented a concurrent BMI that predicted two targets of a sequential movement in parallel during working memory and before movement execution. However, we did not explore how the PMd and SMA population could simultaneously encode both targets accurately during working memory. For example, it is possible that each cell in the population jointly encodes information about the two targets at some level of accuracy or alternatively that it only encodes one target.

In this chapter we investigate the neural mechanism by which the accurate encoding of the full sequence of targets is achieved during working memory. More specifically, we investigate how neurons within the premotor cortex are able to add new information about the second target to working memory without compromising the integrity of information about the first target that is already stored. In particular, and interestingly, we show that the plan elements, or equivalently the two targets, are encoded in the responses of largely disjoint subpopulations of neurons. Moreover, these subpopulations remain dedicated to encoding their corresponding target throughout the recording session in any given day. We further show that when adding the second target information to working memory, the subpopulation dedicated to encoding the first target and their responses remain largely unchanged. We thus show that the population employs a partitioning mechanism to encode the two targets, i.e., it partitions mainly into two subpopulations each dedicated to encoding of one target.

We also perform some control analyses to illustrate that the activity in the premotor population during working memory is not a simple response to movement or visual target images. We also discuss the effect of the number of sequences used in the training and of the behavioral context on the encoding strategy.

## ■ 6.1 Experimental setup

The experimental setup and recording procedures used here are exactly the same as those used in Chapter 5 described in Section 5.2.1 and 5.2.3. However, to examine the effect of adding a new target to working memory, we further examined a subset of sessions in which in addition to

the standard dual-target task, the monkeys also performed a single-target task in randomly interleaved trials. This was necessary to disambiguate the process of holding information in working memory from that of adding information to it. On dual-target trials, described in Section 5.2.1, primates were given two sequentially presented targets that were followed by a blank screen delay. The time delay from the end of the first target presentation to the first "go" cue was therefore 1300-1600 ms. On single-target trials, in comparison, the primates were presented by only the first target and the second target presentation period was replaced with a blank screen period of the same duration. This was followed by an intervening delay, which was of the same length as that in dual-target trials (cf. Figure 5-1a). The monkeys therefore had to keep the first target in working memory for the same 1300-1600 ms time duration as in dual-target trials. Because of the random interleaving and the variable intervening delay, the monkeys could not directly anticipate whether a single target or two targets would appear, or the precise time at which the "go" cue would present.

## ■ 6.2 Methodology

The model fitting and decoding procedures are as described in Chapter 5 in Section 5.3.1 and 5.3.3.

Here, in addition to the population analysis, we also perform the decoding for each neuron to find its single unit decoding accuracies for each target and the sequence. We then test the statistical significance of these accuracies to examine whether the neuron encodes information about one or both targets. Note that as discussed in Section 5.3.1, by building the neuronal models under each sequence separately in the dual-target task, we avoid making any a priori assumptions about whether the cells encode individual targets or combined sequences. More specifically, the significant encoding of the targets will be deduced from the decoding analysis. We also compare the population and single neuron first target decoding accuracies in single-target and dual-target tasks to examine the effect of adding information to working memory. We now develop the details of these analyses.

### ■ 6.2.1 Testing the significance of first and second target encoding for each neuron

To test whether a given neuron encoded significant information about only the first target, only the second target, or both targets during the working memory period, we examined its decoding accuracy in this period for each target. Here, a neuron was considered to encode significant information about a target if its accuracy for the target was above the upper confidence bound for the chance level accuracy. Since this involved two comparisons, one for the first and one for the

second target, we used a Bonferroni correction for multiple comparisons reflected in calculating the upper confidence bound.

### ■ 6.2.2 Comparison of the first target decoding accuracies in the single-target and dual-target tasks

To find the first target decoding accuracy of the single-target task we modeled the activity of each neuron under any given first target as an inhomogeneous Poisson process, which was fitted using the EM procedure. We then performed the maximum-likelihood decoding analysis using leave-one-out cross-validation in the single-target trials. To make the comparison, for the dual-target task we constructed two models one for the first target and one for the second target and then performed the decoding analysis for each target separately. Note that in this analysis for the dual-target task we fit the models to targets as opposed to sequences and subsequently decode them separately. This removes the correlation between the two target accuracies in the dual-target task and hence results in a fair comparison with the single-target task as the second target decoding will not help the decoding of the first target.

## ■ 6.3 Results

Here we present our results on the neural encoding mechanism underlying the accurate and concurrent representation of the full sequence of the motor targets during working memory.

### ■ 6.3.1 A neural partitioning mechanism during working memory

We find that most cells encoded significant information about only one of the two targets during working memory. From the 281 neurons recorded in all sessions, 46% (129 cells) had a target accuracy significantly higher than chance for at least one of the two targets during the working memory period (one-sided z-test, $P < 0.01$). Figure 6-1a shows the scatter plots of the first and second target accuracies of these cells[1]. Of these 129 cells, 68% encoded significant information about only the first target, and 23% encoded significant information about only the second added target (one-sided z-test; $P < 0.01$). The percentage of cells that encoded significant information about both targets was only 9% in comparison and, even of these, most had target accuracies much closer to one of the two subpopulations of cells that significantly encoded only one target. More specifically, once we tested whether the target accuracies of each cell were significantly above

[1]We have corrected these for the correlation effects between the targets by subtracting the chance level contribution of one target from the other as described in the Appendix.

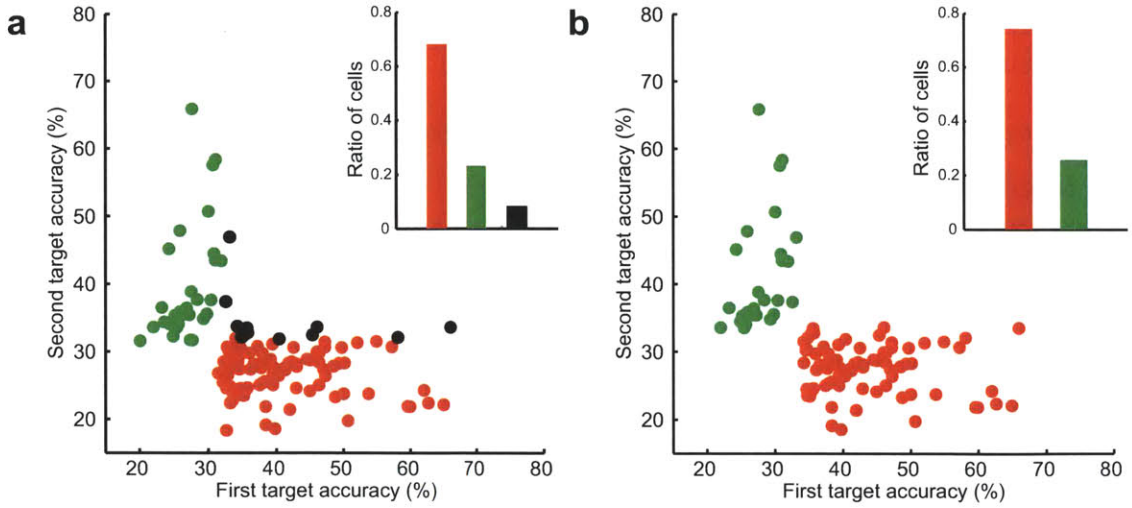


Figure 6-1: Partitioning of the population during working memory. Scatter plot of the first and second target accuracies of the cells that encoded significant information about at least one target during the working memory period. Statistical significance of the target accuracies was tested at the 0.01 level in (a) and the 0.001 level in (b). Red points indicate cells that encoded significant information about only the first target, green points indicate those that encoded significant information about only the second target, and black points indicate those that encoded significant information about both targets. The inset indicates the proportion of cells that encoded significant information about only the first, only the second or both targets during the working memory period with the same coloring schemes from left to right.

chance at a more conservative level ($P < 0.001$), all cells fell in one of two entirely distinct groups, each dedicated for encoding of one target but not the other. This is illustrated in Figure 6-1b.[2]

### ■ 6.3.2 Stability of the target selectivity over time

Since most cells encoded significant information about only one of the two targets, we were interested in examining whether this selectivity was 'hard-coded' in individual cells or whether cells altered their encoding selectivity over time. To do so, we divided all sessions into halves and performed the model construction, leave-one-out cross-validation, and single neuron decoding analyses independently for each half.

We found that, even though multiple sequences were presented over many hundreds of trials in any given session, most neurons remained dedicated to encoding either the first or the second target. Among the two subpopulations of cells that encoded significant information about only a single target, most (89%) did not change the accuracy of the target they encoded between the first

[2]In the Appendix, we show that even without considering a statistical classification, the pattern of the scatter plot observed in Figure 6-1 is highly unlikely ($P < 10^{-15}$) to appear by chance.

and the second halves of the recording session ($\chi^2$ test, $P > 0.05$). Also, the population sequence decoding accuracy across all 12 sequences did not change over time between the first and second halves of the sessions (Wilcoxon's signed-rank test $P = 0.37$).

These results indicated that neurons were mostly dedicated to encoding of a single target during the entire session.

### ■ 6.3.3 Effect of adding information to working memory on the neuronal encoding

Having observed the partitioning mechanism during working memory, we further investigated how the addition of the second target to working memory modified the neural encoding of the first target. To do so, we dissociated the process of holding information about the first target in working memory from that of adding information to it, by comparing the first target decoding accuracy of the same cells during the same period between the single-target and dual-target tasks as explained in Section 6.2.2. Note that the timing of the two tasks are exactly the same. However, in single-target tasks, the second target is not presented to the monkeys and hence they only need to hold the first target in working memory until the "go" cue.

We find that adding information about the second target location to working memory did not incur significant loss of information about the first target. Of cells that encoded significant information about the first target during working memory on single-target trials, most (74%) did not change their first target accuracy during this period on dual-target trials, despite the addition of a second target ($\chi^2$ test, $P > 0.05$). For the population, there was also no significant difference in the first target accuracy during the working memory period when comparing dual-target and single-target trials across sessions (Wilcoxon's signed-rank test, $P = 0.69$). Figure 6-2 shows the population accuracies in a sample session for the single-target and dual-target tasks. These results indicate that the addition of the second target to working memory mostly does not affect the single cell or population encoding of the first target.

### ■ 6.3.4 Predicting the upcoming behavioral performance

We further investigated whether the activity of the partitioned subpopulations was predictive of the upcoming correct/incorrect behavioral performance for the two targets. Namely, we were interested to find whether the partitioned subpopulations could predict whether the animal would execute the two movements correctly after the "go" cue in any given trial.

We found that the activity of each of the partitioned subpopulations of neurons was highly predictive of whether the primates were going to move to the correct target for only one of the

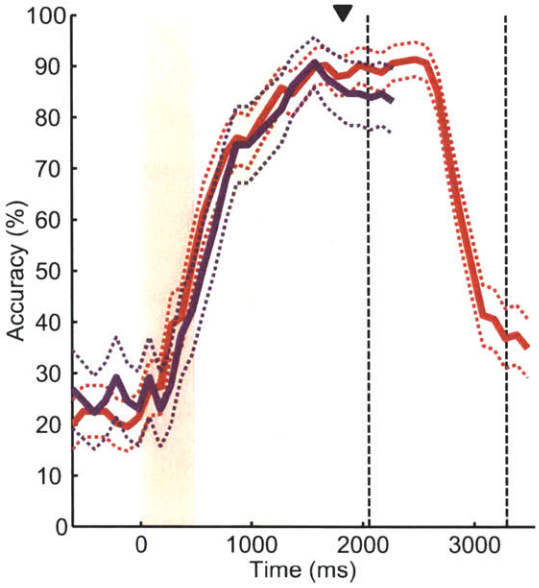


Figure 6-2: Effect of adding information to working memory. On interleaved dual-target trials for a selected session, decoding accuracy for the first target is shown in red. Decoding accuracy for the target given on single-target trials in the same session is shown in magenta. Dotted lines indicate the 95% confidence bounds for each accuracy curve (rather than chance level). The red and green vertical bars indicate the times during which the first and second targets were presented, respectively. The first and second dashed black lines indicate the mean times on which the first and second "go" cues were presented, respectively. The arrow indicates the time point corresponding to the decoding accuracy of the preceding working memory period (0-500 ms from the end of second image presentation).

two distinct targets. Cells that selectively encoded the first target location during the working memory period correctly predicted whether the primates were going to perform the correct first movement 73% of the time ($P < 10^{-15}$). These neurons, however, were poor in predicting whether the primates would perform the second movement correctly (accuracy of 39% compared with 99% upper confidence bound for chance of 38%). This was similarly true for neurons that selectively encoded the second target. These neurons predicted whether the primates would perform the second movement correctly with an accuracy of 60% ($P < 10^{-15}$), but were poorly predictive of the primates' performance on the first movement (accuracy of 42% compared with the 99% upper confidence bound for chance of 40%). This suggested that the neural activity of the partitioned subpopulations of neurons was selectively predictive of upcoming behavior. This further suggested that information encoded by the partitioned subpopulations of neurons within the premotor cortex was relevant to the correct execution of upcoming motor behavior.

### ■ 6.3.5 Control analyses

We have demonstrated that the concurrent and accurate encoding of the targets was achieved through a neural partitioning mechanism during working memory. However, we have not yet investigated the potential contribution of a simple visual response to the target images, or that of a purely movement related response, to the activity observed during the working memory period. Here, we perform control analyses to demonstrate that the activity observed during the working memory period cannot be attributed to visual or movement related responses. We further show that the number of sequences used in the experiments does not affect the encoding of the two targets.

**Controls for visual related responses**

All of the main analyses focused on the working memory period during which no visual cues were presented. Nonetheless, we were also interested in examining whether some of the activity recorded during the first and second target presentation periods themselves could be attributable to a simple visual response to the target images. To investigate this question directly, for each cell we compared the first and second target accuracies during their corresponding presentation periods. Since all four spatial locations (U,D,L,R) were used for both the first and second targets, the visual images shown during the two presentation periods were identical. Therefore, if neuronal responses were simply due to a visual response to the images, then there would be no significant difference between the two target accuracies in their presentation periods, i.e., based on their order in the sequence. We found that out of 281 cells recorded, 112 cells had a significantly higher than chance first or second target accuracy during their presentation periods ($P < 0.01$). Of these cells, however, the vast majority (79%) had a significant target accuracy during its presentation for only one of the two visual targets but not the other ($P < 0.01$). This indicated that the activity during the target presentation periods was not due to a visual response.

**Controls for potential movement related responses during working memory**

Here we show that differences in the neuronal response across sequences during the working memory period were not associated with movement related activity. Monkeys were trained to move the joystick only after presentation of the "go" cue and any premature movement terminated the trial. Nonetheless, it was still possible that small unconstrained movements made by the animals during the working memory period may have contributed to differences in neuronal activity observed. To examine this possibility directly we performed three control analyses.

As the first and most direct control, we examined the electromyography (EMG) activity recorded during the working memory period itself. EMG recordings were obtained from the contralateral arm (covering the deltoid, triceps and biceps muscles) using tin surface electrodes. EMG signals were amplified and recorded on the Plexon work-station (0.7Hz $-$ 300 Hz; 1 pole low-cut and 1 pole high-cut with 50x gain), and sampled at 1 kHz. EMG activity was set above a threshold 3 standard deviations above baseline, template-matched and clustered using principal component analysis. EMG activity was then used to decode the targets using the same maximum-likelihood approach as used for decoding the spiking activity. As expected, we found that EMG activity during the first movement period, which we defined as the 500 ms interval after the "go" cue, predicted the direction of limb movement for the first target with an accuracy significantly higher than chance ($P < 0.02$). In contrast, EMG activity recorded during the immediately preceding working memory period carried no significant information about the first target ($P > 0.1$). These findings indicated that no target-predictive movements were made by the primates during the working memory period.

As the second control analysis, we compared the activity of the cells during the working memory period and the first movement period. More specifically, if the neural activity during the working memory period was purely related to muscle movement, then cells that had a first target selective activity during the first reaching movement would also have it during the preceding working memory period. We hence examined what proportion of cells encoding the first target during the first movement period itself also encoded it during the working memory period. We found that 127 cells had a first target accuracy significantly higher than chance ($P < 0.01$) during the first movement period. Of these, however, a considerable proportion (37%, 47 cells) did not have a significant first target accuracy during the working memory period ($P < 0.01$), indicating that the neural activity during the working memory period was not movement related.

Even though the above controls strongly suggest that the activity during the working memory period was not movement-related, perhaps the most compelling argument comes from the main findings themselves. Hence, as the third control analysis, we examined the implications of the neural partitioning mechanism during working memory. As described in Section 6.3.1, most cells across the population encoded significant information about only one of the two targets. Therefore, if cells responded purely to movement during the working memory period—for example the first upcoming movement—then no cell would encode the second target during this period and vise versa. In other words, if neuronal responses during the working memory period were a

direct reflection of the muscle movements made by the animals (whether first or second), then the population as a whole would only reflect that single movement. We find, however, that the population encoded significant information about both targets in the single window in time during the working memory period.

**Effect of the training set size on the neuronal activity**

Here we explore whether the number of sequences in the training set altered the neuronal encoding of the sequences themselves. To do so, we conducted an experiment where the monkey performed a task in which only 4 sequences were presented (R-U, R-D, L-U, L-D). We then compared the sequence decoding accuracy in this task with that calculated over a subset of trials in the 12 sequence dual-target task in which the same 4 sequences were presented. This means that over this subset of trials we were decoding 1 out of 4 possible sequences. We used leave-one-out cross-validation for all these analyses. We found that the population decoding accuracy of the sequence or either target did not significantly change even when the monkey was behaviorally performing a few select sequences vs. the full set ($\chi^2$ test $P = 0.18$, $P = 0.06$, $P = 0.29$ for sequence, first target, and second target, respectively). This was similarly true for single cells. When considering the decoding accuracy of the individual cells, the vast majority (85%) demonstrated no significant change in sequence decoding accuracy across the select 4 sequences when performing 4 vs. 12 sequence sets ($\chi^2$ test, $P > 0.05$).

In an additional set of controls, we were further interested in examining whether performing only single-target movements in a session versus performing single-target movements interleaved with dual-target movements affected the neuronal encoding of the single target. In other words, it was possible that expecting an additional sequential target to appear in some trials in the interleaved sessions influenced the encoding of the target in the single-target trials. We found, however, that there was no significant difference between the target encoding accuracy of the population between control sessions in which only single targets were presented compared to sessions in which single-target trials were interleaved with dual-target trials ($\chi^2$ test; $P > 0.15$). This comparison between a session with only single-target trials and one with interleaved trials during the same day is shown in Figure 6-3.

### ■ 6.3.6 Neural encoding of targets over time

While the main focus of our work is to examine how new information is encoded by premotor populations at a given instant in time during working memory, these findings do not necessarily

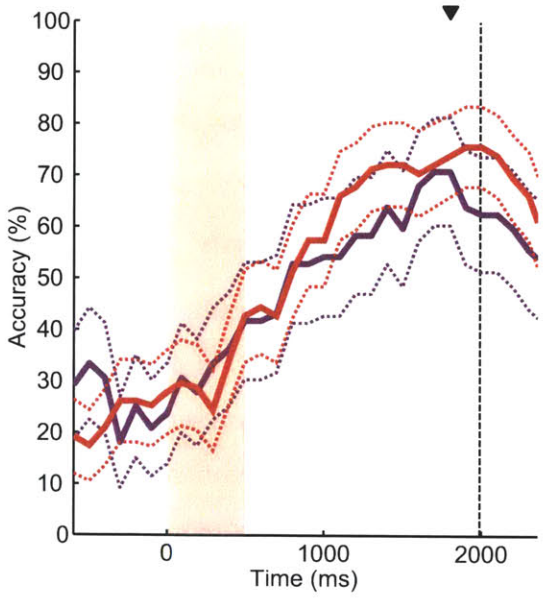


Figure 6-3: Decoding accuracies for interleaved and non-interleaved single-target sessions. Single-target decoding accuracy in the interleaved session is shown in red and in the session with only single-target trials is shown in magenta. Dotted lines indicate the 95% confidence bounds for each accuracy curve (rather than chance level). The red vertical bar indicates the time during which the target was presented. The vertical dashed black line indicates the mean time on which the "go" cue was presented. The arrow indicates the time point corresponding to the decoding accuracy of the preceding working memory period.

reflect how information is encoded by cells over time. The time-evolution of the first and second target accuracies of sample cells are shown in Figure 6-4. Here, we found that neurons significantly altered the degree to which they encoded information about the two targets across different time-points during the trial. Some cells, for example, encoded no information about the second target during the working memory period, but then encoded significant information about the second target during the second movement as shown in an example cell in Figure 6-4b. Therefore, observing that a cell predominantly encodes information about the first target during the working memory period does not necessarily indicate that it does not encode the second target at any other time during the task. What the present findings, however, do suggest is that when both targets are held in working memory at a given instant in time, the vast majority of cells mostly encode information about only one of the two targets.

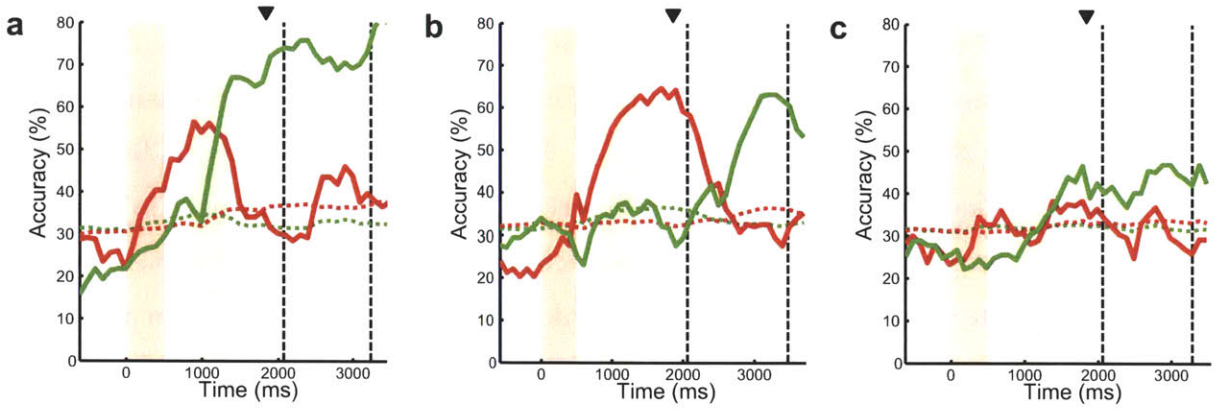


Figure 6-4: Decoding accuracy of sample cells that encode only the second (a), only the first (b), or both targets (c) during the working memory period. Red curve shows the mean decoding accuracy of the first target and green curve shows that of the second target. Each point on the curves indicates the decoding accuracy using the neuronal activity in the preceding 500 ms window. Time at zero is aligned to the start of first target presentation. The red and green vertical bars indicate the times during which the first and second targets were presented, respectively. The first and second dashed black lines indicate the mean times on which the first and second "go" cues were presented, respectively. The arrow indicates the time point corresponding to the decoding accuracy of the preceding working memory period (0-500 ms from the end of second image presentation). Horizontal dotted lines indicate the 99% chance upper confidence bounds for the first target and second target with the same color scheme as above.

### ■ 6.3.7 Comparison to previous studies on sequential planning

Previous studies have implicated the prefrontal and premotor areas in the planning of sequential behavior [62–65, 72–75]. Our observation of the existence of some cells that hold significant information about both targets is consistent with prior work in sequential planning that have found that a small proportion of neurons in these areas demonstrate enhanced activity for particular sequences [62–65]. In comparison to the present study, however, most experiments have focused on the recollection of well rehearsed sequences rather than the maintenance of target-related information in working memory per se [63–65].

While we employed sequential movements in the present task, here the main purpose of doing so was to examine how information is simultaneously encoded and maintained in working memory when new information is added to already held information. Here, we employed three methodological approaches that allowed us to examine this question. First, we designed the task so that we could examine potential differences between maintaining target-related information in working memory and adding such information to it. This was done by comparing single-target

and dual-target tasks. Second, we employed a maximum-likelihood approach to (i) define an accuracy measure for the amount of information that is simultaneously encoded by the neuronal activity about the two targets, and (ii) find and differentiate the amount of information encoded by individual cells about each of the two targets. This approach is significantly different from declaring a significant change in firing rate in observing a target or sequence used in most previous work. Third, we simultaneously recorded the activity of multiple cells within the population, which allowed us to examine the distribution of information across the population at any given point in time.

## 6.4 Discussion

We have illustrated that accurate and concurrent encoding of two motor targets by the premotor population was achieved through a partitioning mechanism. More specifically, the premotor population partitioned into two subpopulations each dedicated to encoding of one target but not the other throughout a day. Moreover, activity of each of the two subpopulations could accurately predict whether the primates would select the correct spatial location after the "go" cue for the target the subpopulation encoded. This further suggests that information encoded by the partitioned subpopulations of neurons within the premotor cortex was also relevant to the correct execution of upcoming motor behavior.

To examine the effect of adding information to working memory on the encoding of information already stored, we compared the dual-target task with a single-target task with the exact same task timing but in which no second target was presented. We showed that, interestingly, the subpopulation that encoded the target during the single-target trials and their responses were largely unchanged during the dual-target trials with the addition of the second target. This allowed the second target information to be added to working memory without degrading the integrity of information already stored about the first target.

While we found that the concurrent encoding of targets was achieved through a neural partitioning strategy in the dual-target task, the current findings may not be applicable to all other behavioral contexts. In the present task, only two targets were tested. In some situations, however, certain behavioral contexts may require up to three or more targets to be simultaneously held in working memory prior to reaching to them. Therefore, adding a third target to two concurrently held targets may result in a different strategy for encoding information. An interesting finding made in the present study is that the added target recruited a far smaller number of cells

than the number of cells encoding the first concurrently held target. It is interesting to speculate whether the need to recruit smaller subpopulations of neurons may also provide one limitation on how many items of information may be held by individuals during working memory. Future experiments that employ more targets may provide insight into this question.

## 6.5 Appendix

### 6.5.1 Correcting for the correlation effect between the two targets in the scatter plots of target accuracies

In the scatter plots, we correct for the correlation effect between the first and second target accuracy values, which is the byproduct of the choice of sequences used in the experimental design. Since the first and second target locations cannot be the same within the set of 12 sequences, the accuracy of one target has a contribution (even though fairly small) to the accuracy of the other even if the neuronal activity is not encoding that target per se. Hence, we also need to remove this effect to see the true representation of a target by the neuronal activity, just as we took it into account in calculating the chance level accuracies in Section 5.3.5. This means that for the first target accuracy value of each cell, we subtract the chance level contribution of the second target accuracy of that cell, or $p_1^{\text{chance}} - 1/4$ , from it and vice versa.

### 6.5.2 Testing the statistical significance of the partitioning mechanism observed

To establish that a significant divergence exists between the amount of information held by the premotor neurons about the two targets, we need to show that the absolute difference between the two target accuracies of each cell averaged across the population is significantly larger than a population with the same target accuracy values but with no structured relationship between each cell's target accuracies. To find the distribution of this average absolute difference in such a population with no structure, we keep the accuracy values the same but randomly permute them 100,000 times within the population. This removes any possible structure between the target accuracies of each cell and hence creates a null hypothesis distribution. We can then establish the significance of this divergence by comparing the average absolute difference of the target accuracies of the premotor neurons against this null distribution and calculate a p-value.

To do so, we first correct for the correlation effect between the first and second target accuracy values, which is the byproduct of the choice of sequences used in the experimental design, as explained above. We then randomly permute these first target accuracy values among the cells

many times while keeping their second target accuracy values the same. This generates a new population each time with the same accuracy values but no pair-wise structure between the first and second target accuracies of each cell. For each new population, we compute the average difference and after repeating this many times, find its distribution. We then use this distribution to find whether the difference of the first and second target accuracies of the cells averaged over the population is significantly different from a population with no structure.

# Chapter 7

# Conclusion

Brain-machine interfaces have the potential to enable motor function in individuals with neurological injury or disease. One approach to the problem of BMI design is to develop the BMI to match the original motor functionality of the individual. Such designs are indeed valuable for performing many natural tasks and hence they have been the focus in the BMI research thus far. However, a compelling goal is that of designing BMIs that can enhance the original motor functionality to achieve the motor goal of the individual more effectively. Here, we have developed novel BMIs on both fronts. On the first front, we have developed a BMI that aims to estimate natural movement more accurately by emulating the sensorimotor processing underlying actual motor control. On the second front, we have introduced a concurrent BMI architecture for enhancing original motor function and implemented it for execution of a sequential movement. We have implemented both novel designs in rhesus monkeys.

## ■ 7.1 Contributions of this thesis

To restore natural motor function, we developed a BMI that jointly decodes the target and trajectory of movement in contrast to current BMIs that decode only one or the other. There are two major motivations for such a design. First, both target and trajectory information are encoded in the motor cortical areas. Target information is encoded before movement initiation and trajectory or kinematic information is encoded around the time of movement. Second, based on our understanding of the sensorimotor system, any natural movement is performed to achieve a goal or target and hence the kinematics and the target of the movement are strongly correlated. Therefore, decoding information about one can improve the decoding accuracy of the other.

To combine the encoded target and trajectory information of movement, we use the optimal feedback control theory of the sensorimotor system. In this view, any task is performed to accomplish a goal during which there is sensory feedback about the state of the system. Based on the goal and this feedback, the brain decides on the next plan of action and generates the movement. Once we decode the target from the activity prior to movement, we can then use it in the optimal feedback control framework to build a goal-directed prior state-space model for the movement. We can then use this prior model in a recursive Bayesian decoder to combine the target informa-

tion with the peri-movement spiking activity and estimate the trajectory. Our BMI hence has a two-stage design. In the first stage, it decodes the target of the movement based on the spiking activity before movement initiation. In the second stage, it combines the decoded target with the peri-movement spiking activity to decode the kinematics.

Moreover, unlike current BMI designs, our BMI processes the spiking activity directly on the millisecond time scale that it is recorded. Since the spiking activity is a sequence of zeros and ones, we use a point process observation model for this activity. In other words, while current BMIs adapt the recorded spiking activity to the use of well-known decoders (e.g., linear regression, Kalman filter) by converting it to continuous inputs, we design an algorithm that is adapted to the originally recorded spiking signal.

We show that by combining both the target and kinematic related spiking activities, the BMI performs more accurately, has a lower RMS error, and generates smoother estimates. The combination of the two types of information also results in a more robust decoder, which can result in high performance in the presence of experimental variations.

While BMIs that aim to restore the original motor functionality are valuable in many natural settings, a compelling goal is the design of more intelligent BMIs that aim to surpass such original functionality. To do so, we introduced a concurrent BMI architecture that simultaneously decodes multiple elements of a plan before movement execution, when achieving the overall goal of the task involves a sequence of such elements—for example picking up a key and then unlocking the door or picking up a book and then putting it on a shelf. While one way to achieve such complex goals is to decode and execute the plan elements one by one using for example our two-stage BMI, an alternative way to do so is to decode all the plan elements concurrently or in parallel before movement execution. Having access to the entire plan enables the BMI to analyze the whole sequence before movement and determine more effective ways to perform the task.

The development of such intelligent BMIs is therefore dependent on the ability to decode multiple plan elements concurrently and accurately during working memory. We have demonstrated that such concurrent decoding is indeed possible and it can be done accurately and robustly. Based on this observation, we have implemented a concurrent BMI in rhesus monkeys that decodes the plan elements of a two-step sequential movement, i.e., the two targets of the movement, in parallel and before movement initiation. This provides the BMI with the flexibility to more effectively perform the sequential movement. For example, it can execute both movements much faster than possible by natural sequential movement itself. This design thus demonstrates the potential in

such concurrent BMIs. More generally, in the future, a wide variety of possible higher-level motor goals could be achieved, in principle, by intelligent BMIs exploiting such concurrent decoding.

## ■ 7.2 Future directions and challenges

We demonstrated the application of two novel BMIs for restoration and enhancement of natural movement. However, there is still much work to be done before these can be used in clinical applications.

For a BMI to be truly autonomous, it should be able to detect different types of activity. For example, the BMI should be able to determine whether the user is inactive, is planning a movement, or wants to execute the movement. In real-time BMI experiments, because of the experimental task design, the timing of these epochs are known. For example the target presentation time indicates the start of the planning phase or the "go" cue indicates the start of the execution phase. However, in the context of natural behavior, such distinctions must be made purely based on the neural activity. We discussed this problem and some possible approaches to it in Chapter 2. A necessary extension for a BMI is to design the decoding algorithm to jointly estimate the epoch of activity and the intended movement.

One limitation of current BMI decoders is their requirement for a separate training session in which models are built at the beginning of each day. This means that the user needs to perform a training session every day before being able to use the BMI to perform a task. A possible extension is thus to design the BMI to adaptively update the model parameters from day to day and avoid repeating the training session. This requires the design of adaptive BMI decoders for ensemble spiking activity, which jointly update the kinematic states and the model parameters.

One topic of significant importance in the design of high-performance BMIs is how learning or neural adaptation can help with this performance. Prior studies have indicated the ability of the brain to adapt to various decoding algorithms such as regression decoders (e.g., [8–10, 13]) over the period of days or hours, which in turn results in improved performance. This has motivated an approach towards the BMI problem that relies mostly on the ability of the brain to learn to use a decoder, as opposed to more careful modeling of the processes underlying motor control and decoder design. It has actually been suggested that using *simple* decoders could be beneficial as it may be easier for the brain to learn them. However, it is important to note that a *computationally* simple decoder is not necessarily simpler for the brain to learn. As a counter example, a hypothetical and sophisticated decoder that can perfectly decode the intended movement would

require no learning. Therefore, an alternative approach is to take a more systematic view towards the BMI problem to first build a more accurate decoder and then add learning at the end to further refine estimation accuracy. For example, in this work we built a decoder that attempted to carefully model the sensorimotor processes underlying actual motor control and the properties of the recorded spiking activity. Since these systematic designs are traditionally not tested in real-time BMIs, it is not established how they would compare to more standard decoders, such as the regression decoder, in the presence of learning and adaptation over long periods. Hence, careful study of the learning effects on more systematic decoders and their meaningful comparison with more standard decoders is of great importance to the field.

To plan and execute a natural movement, the brain uses multiple modalities of sensory feedback, such as visual, proprioceptive, and tactile. However, in BMIs, typically the only source of feedback available is visual. Recent results [76] have indicated that adding proprioceptive feedback to visual feedback in a BMI by passively moving the arm of a monkey to follow the real-time cursor position estimate improves the BMI performance. For paralyzed patients, however, additional sensory modalities need to somehow be delivered to the relevant regions of the brain to induce such sensation. Electrical microelectrode stimulation of the somatosensory cortex has been indicated as a potential means to deliver the proprioceptive feedback to the brain [77]. This motivates the design of bi-directional BMIs in which movement information is decoded from the brain while various modalities of sensory feedback are delivered to the brain to improve real-time performance.

In the case of the BMI design for enhanced motor function, we demonstrated the ability to decode two motor targets or plan elements concurrently before the initiation of a sequential movement. It would be interesting to further investigate whether accurate and simultaneous decoding can be achieved with more than two motor plans and in different behavioral contexts.

Finally, an essential problem that needs to be resolved for clinical viability of BMI systems using intra-cortical recordings is that of obtaining stable recording over the period of years. The loss of the neural signal captured by the electrode arrays over the course of months limits the robustness and performance of the BMIs over time and hence their clinical viability. This loss of recorded signal could be a result of scar tissue formation around the array, failure of the array, or motion of the electrodes, also known as micro-motion. Hence an important challenge for the clinical applicability of intra-cortical BMIs is the design of electrode arrays that can capture stable recordings over the time course of years.

Future advances on all these fronts will facilitate the development of fully autonomous, accurate, robust, and bi-directional BMIs and in turn enable their widespread clinical use for helping disabled patients.

# Bibliography


[1] M. A. Lebedev and M. Nicolelis, "Brain-machine interfaces: past, present and future," *Trends in Neurosciences*, vol. 29, no. 9, pp. 536–546, 2006.

[2] M. D. Linderman *et al.*, "Signal processing challenges for neural prostheses," *IEEE Signal Process. Mag.*, pp. 18–28, Jan. 2008.

[3] N. G. Hatsopoulos and J. P. Donoghue, "The science of neural interface systems," *Annu. Rev. Neurosci.*, vol. 32, pp. 249–266, Jul. 2009.

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

[5] J. Wessberg *et al.*, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, Nov. 2000.

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

[7] L. R. Hochberg *et al.*, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, pp. 164–171, Jul. 2006.

[8] J. M. Carmena *et al.*, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biol.*, vol. 1, no. 2, pp. 193–208, 2003.

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

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

[11] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 51, pp. 17 849–17 854, Dec. 2004.

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

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

[14] G. H. Mulliken, S. Musallam, and R. A. Andersen, "Decoding trajectories from posterior parietal cortex ensembles," *J. Neurosci.*, vol. 28, no. 48, pp. 12 913–12 926, Nov. 2008.

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

[16] Z. Li *et al.*, "Unscented Kalman filter for brain-machine interfaces," *PLoS ONE*, vol. 4, no. 7, pp. 1–18, Jul. 2009.

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

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

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

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

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

[22] J. Ashe and A. P. Georgopoulos, "Movement parameters and neural activity in motor cortex and area 5," *Cereb. Cortex*, vol. 6, pp. 590–600, Nov. 1994.

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

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

[25] D. J. Crammond and J. F. Kalaska, "Differential relation of discharge in primary motor cortex and premotor cortex to movements versus actively maintained postures during a reaching task," *Exp. Brain Res.*, vol. 108, pp. 45–61, 1996.

[26] R. Caminiti, P. B. Johnson, C. Galli, S. Ferraina, and Y. Burnod, "Making arm movements within different parts of space: The premotor and motor cortical representation of a coordinate system for reaching to visual targets," *J. Neurosci.*, vol. 11, no. 5, pp. 1182–1197, May 1991.

[27] K. Kurata, "Premotor cortex of monkeys: Set- and movement-related activity reflecting amplitude and direction of wrist movements," *J. Neurophysiol.*, vol. 69, pp. 187–200, 1993.

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

[29] B. B. Averbeck, M. V. Chafee, D. A. Crowe, and A. P. Georgopoulos, "Parietal representation of hand velocity in a copy task," *J. Neurophysiol.*, vol. 93, pp. 508–518, 2005.

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

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

[32] C. A. Buneo, M. R. Jarvis, A. P. Batista, and R. A. Andersen, "Direct visuomotor transformations for reaching," *Nature*, vol. 416, pp. 632–636, Apr. 2002.

[33] R. A. Andersen, J. W. Burdick, S. Musallam, B. Pesaran, and J. G. Cham, "Cognitive neural prosthetics," *Trends in Cognitive Sciences*, vol. 8, no. 11, pp. 486–493, Nov. 2004.

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

[35] D. J. Crammond and J. F. Kalaska, "Modulation of preparatory neuronal activity in dorsal premotor cortex due to stimulus-response compatibility," *J. Neurophysiol.*, vol. 71, no. 3, pp. 1281–1284, Mar. 1994.

[36] D. Boussaoud and F. Bremmer, "Gaze effects in the cerebral cortex: reference frames for space coding and action," *Exp. Brain Res.*, vol. 128, pp. 170–180, 1999.

[37] S. Kakei, D. S. Hoffman, and P. L. Strick, "Sensorimotor transformations in cortical motor areas," *Neurosci. Res.*, vol. 46, pp. 1–10, 2003.

[38] M. M. Churchland, G. Santhanam, and K. V. Shenoy, "Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach," *J. Neurophysiol.*, vol. 96, pp. 3130–3146, 2006.

[39] D. J. Crammond and J. F. Kalaska, "Prior information in motor and premotor cortex: Activity during the delay period and effect on pre-movement activity," *J. Neurophysiol.*, vol. 84, pp. 986–1005, 2000.

[40] M. D. Crutcher, G. S. Russo, S. Ye, and D. A. Backus, "Target-, limb-, and context-dependent neural activity in the cingulate and supplementary motor areas of the monkey," *Exp. Brain Res.*, vol. 158, pp. 278–288, 2004.

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

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

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

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

[45] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a Kalman filter," *Neural Comput.*, vol. 19, pp. 80–118, 2006.

[46] E. N. Brown, L. M. Frank, D. Tang, M. C. Quirk, and M. A. Wilson, "A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells," *J. Neurosci.*, vol. 18, no. 18, pp. 7411–7425, Sep. 1998.

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

[48] J. E. Kulkarni and L. Paninski, "State-space decoding of goal-directed movements," *IEEE Signal Process. Mag.*, vol. 25, pp. 78–86, Jan. 2008.

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

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

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

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

[53] B. M. Yu *et al.*, "Mixture of trajectory models for neural decoding of goal-directed movements," *J. Neurophysiol.*, vol. 97, pp. 3763–3780, 2007.

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

[55] C. Kemere *et al.*, "Detecting neural-state transitions using hidden Markov models for motor cortical prostheses," *J. Neurophysiol.*, vol. 100, pp. 2441–2452, Jul. 2008.

[56] D. Bertsekas, *Dynamic Programming and Optimal Control.* Athena Scientific, 2005.

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

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

[59] P. Bloomfield, *Fourier analysis of time series.* Wiley, 2000.

[60] S. H. Scott, "Optimal feedback control and the neural basis of volitional motor control," *Nat. Rev. Neurosci.*, vol. 5, pp. 532–546, Jul. 2004.

[61] J. P. Cunningham *et al.*, "A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces," *J. Neurophysiol.*, vol. 105, pp. 1932–1949, 2011.

[62] Y. Ninokura, H. Mushiake, and J. Tanji, "Representation of the temporal order of visual objects in the primate lateral prefrontal cortex," *J. Neurophysiol.*, vol. 89, pp. 2868–2873, 2003.

[63] J. Tanji and K. Shima, "Role for supplementary motor area cells in planning several movements ahead," *Nature*, vol. 371, pp. 413–416, Sep. 1994.

[64] K. Shima, M. Isoda, H. Mushiake, and J. Tanji, "Categorization of behavioural sequences in the prefrontal cortex," *Nature*, vol. 445, pp. 315–318, Jan. 2007.

[65] K. Shima and J. Tanji, "Neuronal activity in the supplementary and presupplementary motor areas for temporal organization of multiple movements," *J. Neurophysiol.*, vol. 84, pp. 2148–2160, 2000.

[66] D. Baldauf, H. Cui, and R. A. Andersen, "The posterior parietal cortex encodes in parallel both goals for double-reach sequences," *J. Neurosci.*, vol. 28, no. 40, pp. 10 081–10 089, Oct. 2008.

[67] E. N. Brown, R. Barbieri, U. T. Eden, and L. M. Frank, "Likelihood methods for neural data analysis," *In: Computational Neuroscience: A Comprehensive Approach, edited by Feng J. London: CRC*, 2003.

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

[69] A. C. Smith *et al.*, "State-space algorithms for estimating spike rate functions," *Computational Intelligence and Neuroscience*, vol. 2010, 2010.

[70] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[71] P. D. Jong and M. J. Mackinnon, "Covariances for smoothed estimates in state space models," *Biometrika*, vol. 75, no. 3, pp. 601–602, 1988.

[72] H. Mushiake, N. Saito, K. Sakamoto, Y. Itoyama, and J. Tanji, "Activity in the lateral prefrontal cortex reflects multiple steps of future events in action plans," *Neuron*, vol. 50, pp. 631–641, May 2006.

[73] B. B. Averbeck, J.-W. Sohn, and D. Lee, "Activity in prefrontal cortex during dynamic selection of action sequences," *Nat. Neurosci.*, vol. 9, no. 2, pp. 276–282, Feb. 2006.

[74] M. Ohbayashi, K. Ohki, and Y. Miyashita, "Conversion of working memory to motor sequence in the monkey premotor cortex," *Science*, vol. 301, pp. 233–236, Jul. 2003.

[75] K. Shima and J. Tanji, "Both supplementary and presupplementary motor areas are crucial for the temporal organization of multiple movements," *J. Neurophysiol.*, vol. 80, pp. 3247–3260, 1998.

[76] A. J. Suminski, D. C. Tkach, A. H. Fagg, and N. G. Hatsopoulos, "Incorporating feedback from multiple sensory modalities enhances brain-machine interface control," *J. Neurosci.*, vol. 30, no. 50, pp. 16 777–16 787, Dec. 2010.

[77] B. M. London, L. R. Jordan, C. R. Jackson, and L. E. Miller, "Electrical stimulation of the proprioceptive cortex (area 3a) used to instruct a behaving monkey," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 1, pp. 32–36, Feb. 2008.