

Coupling time decoding and trajectory decoding using a target-included model in the motor cortex

Vernon Lawhern^{a,*}, Nicholas G. Hatsopoulos^b, Wei Wu^a

^a Department of Statistics, Florida State University, Tallahassee, FL 32306-4330, USA

^b Department of Organismal Biology and Anatomy, Committees on Computational Neuroscience and Neurobiology, University of Chicago, Chicago, IL 60637, USA

ARTICLE INFO

Article history:

Received 11 January 2011

Received in revised form

24 May 2011

Accepted 9 October 2011

Communicated by D. Erdogmus

Available online 27 December 2011

Keywords:

Neural decoding

Motor cortex

Target-included model

Time decoding

Trajectory decoding

State-space model

ABSTRACT

Significant progress has been made within the last decade in motor cortical decoding that predicts movement behaviors from population neuronal activity in the motor cortex. A majority of these decoding methods have focused on estimating a subject's hand trajectory in a continuous movement. We recently proposed a time identification decoding approach and showed that if a stereotyped movement is well represented by a sequence of targets (or landmarks), then the main structure of the movement can be reconstructed by detecting the reaching times at those targets. Both trajectory decoding and landmark-time decoding have their particular advantages, whereas a coupling of these two different strategies has not been examined. In this article we propose a synergy that comes from combining these two approaches for a stereotyped movement under a linear state-space framework. We develop a new decoding procedure based on a forward-backward propagation where the target is used in the initial stage in the backward step. Experimental results show that the new method significantly improves decoding accuracy over the non-target-included models. Furthermore, the coupling based on the new target-included method effectively combines the time decoding and trajectory decoding and further improves the decoding accuracy.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Researchers have made significant contributions in the development of neural decoding methods in motor cortex over the last decade. These methods aim to reconstruct continuous arm movements of human or non-human primates using measurements of observed neural activity. Commonly used decoding methods include various linear Gaussian models [19,24,34,12] and GLM (generalized linear model)-based models [22,33,21,31]. Other approaches include neural networks [25,13], nonparametric models [32], general-purpose filters [29], *common-input* (or *hidden-state*) models [16,4,36,18], and approximate methods for state-space models [8,15,23].

These methods have largely focused on the decoding of the arm trajectory using neural activity, whereas they often lack an appropriate representation on the intended motor behaviors. For example, in many state-space frameworks, the hand movement is modeled as a first order autoregressive (i.e. AR(1)) model [5,33]. However, arm movements are, in general, directed to a target that is in close proximity of the subject. This knowledge will

(consequently) restrict the set of possible states that lead up to the target. Since the arm movements are essentially goal-directed, the inclusion of this prior information could be useful in the decoding process. Such information may be formulated in different ways. For example, we may know that the movement will stop when the target is reached (i.e. zero velocity at the end-point), or the target position is at a fixed point (i.e. position is known at the end-point).

Several recent studies have investigated the inclusion of the target information in the decoding framework. Kemere et al. [10] proposed a maximum likelihood approach to decode stereotyped reaching movements. This decoding is conducted in two steps: (1) decode the target or goal of the movement using neural activity in plan period; (2) decode the time evolution of the trajectory. The same research group also included the target in feed-forward-controlled linear systems [11], where the target was used in the measurement model. Yu et al. [37] described the movement to each target individually in a center-out reaching movement and combined all these simple trajectory paths in a probabilistic mixture of trajectory models. In a decoding study in posterior parietal cortex by Mulliken et al. [20], the static target position is incorporated into the hand kinematics. The neural activity and hand state are then modeled by a standard Kalman filter. The idea of incorporating target information in discrete locations was also examined by Cunningham et al. [7] where they

* Corresponding author.

E-mail addresses: vlawhern@stat.fsu.edu (V. Lawhern), nicho@uchicago.edu (N.G. Hatsopoulos), www@stat.fsu.edu (W. Wu).

investigated the optimal placement for targets to achieve maximum decoding accuracy.

Srinivasan et al. [30] recently developed a new type of target-included model where the target is characterized as the endpoint of the movement with a specified linear Gaussian equation. The hand kinematics are still modeled as a traditional linear Gaussian AR(1) model. The target representation in this model does not assume any stereotyped or canonical form on the movement and minimally constrains the kinematic state equation. Conditioned on the target, they showed that the kinematic terms are still linear Gaussian, yet with a drift term which characterizes the target information. The new target-included kinematic prior model can be directly used in the Bayesian decoding process with any likelihood representation on the neural activity. Their work has been extended to a more general case where the target position can be dynamic during the movement behavior [28]. More recently, Kulkarni and Paninski [17] derived a more efficient estimation procedure based on a continuous state transition using a first-order linear stochastic differential equation (SDE). This approach can directly obtain the target-included estimate for any time without a recursion (which is necessary in [30]).

In this article, we propose a new computational approach to conduct decoding in a target-included model. We adopt the same mathematical models on the hand state transition and target representation as that in [30]; that is, the hand state is an AR(1) model and the target is a linear Gaussian model to the endpoint of the hand state. However, in contrast to a new target-included kinematic transition, we estimate the hand state using a forward–backward propagation [17]. We will show that this new computational approach can greatly accelerate the decoding process. This efficiency is also comparable to that with continuous SDE on kinematics [17]. In addition, the new method has no constraint on the model parameters, which is often required in the SDE method for an efficient estimation. Furthermore, conditioned on the target information, the framework can be easily adapted for both filtering estimation (by past and current neural activity) and smoothing estimation (by past, current, and future neural activity).

We note that the new target-included algorithm, as well as that in [30,17], is based on an important assumption that the arrival time at the target is known beforehand. In practical neural decoding situations, such information is often not known which may limit the application of these methods. However, this assumption may not be necessary if the hand movement is stereotyped with certain landmarks. Our recent work indicates if the experimental paradigm involves movement to a sequence of fixed target positions, then the time of arrival to the targets can be directly estimated based on the same neural activity [6]. We referred to this method as *timing decoding* to differentiate from the common *trajectory decoding*. When the time to each landmark is detected, the main structure of the movement can be reconstructed by linearly connecting these landmark points. Note that the movement is continuous and the time decoding is different from the *state estimator* in an instructed-delay movement [27,1]. A similar decoding strategy has been explored in [2] where a cepstrum method is used to detect the time of occurrence of an eye saccade movement.

Trajectory decoding and time decoding provide reasonable and complementary reconstructions. The time decoding lacks the details between landmarks. In contrast, the trajectory decoding cannot identify the main structure of trajectory. To fully utilize the benefits from both decoding strategies, we examine an approach to couple them based on the proposed target-included decoding in the context of a landmark-defined movement. To achieve this goal, we plan to proceed in two steps: (1) detect the time to each landmark (target) by the recently developed time-

warping method [6]; and then (2) include the detected times to the new target-included decoding process. Note that here the time to the target provides essential information as the target positions are always known (the hand movement is stereotyped). We will show that this new method better addresses the characteristics of the stereotyped movement and significantly improves the decoding accuracy.

2. Methods

2.1. Experimental methods

Electrophysiological recording: The neural data used here were previously recorded and have been described elsewhere [6,35]. Briefly, silicon microelectrode arrays containing 100 platinum-tipped electrodes were implanted in the arm area of primary motor cortex (MI) in three juvenile male macaque monkeys (*Macaca mulatta*). Signals were filtered, amplified (gain, 5000) and recorded digitally (14-bit) at 30 kHz per channel using a Cerebus acquisition system (Blackrock Microsystems Inc.). Single units were manually extracted by the Contours and Templates methods using Offline Sorter (Plexon Inc.). Each monkey's arm rested on cushioned arm troughs secured to links of a two-joint exoskeletal arm (KINARM system, see [26]) underneath the projection surface. The shoulder joint was abducted 90° such that shoulder and elbow flexion and extension movements were made in the horizontal plane. The monkeys were trained to perform behavioral tasks by moving a cursor to targets via contralateral arm movements.

Random Target Pursuit (RTP) task—Monkeys 1 and 2: At any one time, a single target appeared at a random location in the workspace, and a monkey was required to reach it within 2 s. As soon as the cursor reached the target, the target disappeared and a new target appeared in a new, pseudo-random location (Fig. 1A). The size of the target was 1 cm × 1 cm. After reaching the seventh target, the monkey was rewarded with a drop of water or juice. A new set of seven random targets was presented on each trial. The majority of trials were 4–5 s in duration. One dataset was collected and analyzed for each of two monkeys where the first monkey successfully completed 550 trials with 124 distinct units simultaneously recorded and the second one completed 400 trials with 125 units recorded. The firing rates of these cells were computed by counting the number of spikes within the previous 50 ms time window. To match time scales, the hand positions were also sampled every 50 ms. We computed hand velocity using simple differencing on position. We also recorded the time instant when each target was reached.

Squared-Path (SP) task—Monkey 3: In contrast to the RTP task, the targets in the SP task were all fixed at the four corners of a square and the movement was stereotyped (Fig. 1B). In this experiment, a landmark event was defined as the cursor reaching any of the corners of the square. This event also corresponded to the time at which the next target appeared. The size of the target is 1 cm × 1 cm. A movement sequence started at the time when the first corner was reached. A trial could start at any time after the end of the previous one. At the start of a trial, the target appeared at the upper right corner of the square and the monkey had to move the cursor to it. Once reached, the target jumped counterclockwise to the next corner and the monkey had to move the cursor again, and so on. A trial was registered as successful only if the monkey reached the target at all of the four corners within 5 s. In the data we collected, there were 240 successful trials. A total of 49 MI single units were recorded simultaneously. To use the time identification method in [6], the bin size was chosen to be 10 ms for both firing rates and hand kinematics.

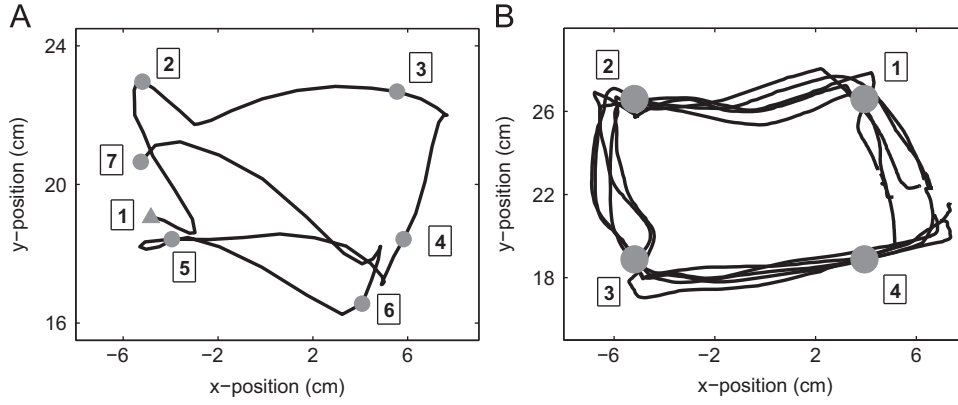


Fig. 1. Two behavioral tasks used in this study. *A.* Trajectory of hand movement in one sample trial of the RTP task. The hand started at the first target (small triangle), and then moved to reach each subsequent target (small dot). The trial was complete when the seventh target was reached. *B.* Trajectories of hand movement in five successful trials of the SP task. In each of the trials, the monkey’s hand passed the four corners of a square in counterclockwise order to reach a target that jumped from one corner to another, starting from the upper right corner. The time when the upper right corner was reached was defined as the onset of the movement sequence in a trial. The corners of the square are denoted by gray spots.

2.2. Statistical methods

The statistical methods include two parts: (1) develop a target-included decoding framework by assuming target information is known during the hand movement. This investigation will be for both RTP and SP tasks. (2) Identify the time when the target is reached for a stereotyped movement, and then couple the time decoding and trajectory decoding by incorporating the detected landmarks in the target-included decoding. This investigation will be for the SP task.

2.2.1. Basic setup

In recent state-space models, the hand kinematics are often represented as an AR(1) process; that is,

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t, \quad (1)$$

where $\mathbf{x}_t = [x, y, v_x, v_y]^T_t$ represents x -position, y -position, x -velocity, and y -velocity at time t (i.e. the t -th time bin $[(t-1)\Delta, t\Delta]$ where Δ is the bin size), $t=1,2,\dots,T$. $\mathbf{A} \in \mathbb{R}^{4 \times 4}$ is the linear coefficient matrix. The noise term \mathbf{w}_t is assumed zero mean and normally distributed, i.e. $\mathbf{w}_t \sim N(0, \mathbf{W})$, $\mathbf{W} \in \mathbf{S}_+^4$ (a 4×4 symmetric, positive semidefinite matrix. The notation is taken from [3]). Eq. (1) is a simple yet effective model, which basically provides a continuous constraint on the monkey’s hand movement.

Note that the subject’s hand movement was not freely moving in the plane, but was always directed by a target on the projection surface. In the target-directed model we assume some target information \mathbf{y}_T is known at time T . For example, we may know that at time T the hand will reach some pre-specified location, or the hand velocity will be zero. A simple model with noise is

$$\mathbf{y}_T = \mathbf{G}\mathbf{x}_T + \mathbf{v}, \quad (2)$$

where $\mathbf{G} \in \mathbb{R}^{r \times 4}$ is the transition matrix between kinematics at time T and target. The noise term \mathbf{v} is assumed zero mean and normally distributed, i.e. $\mathbf{v} \sim N(0, \mathbf{V})$, $\mathbf{V} \in \mathbf{S}_+^r$. For the datasets in both RTP and SP movements, the target position is around the end point of hand position (within the target size). Therefore, $r=2$ and

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (3)$$

We let \mathbf{V} be an identity matrix as the size of the target is $1 \text{ cm} \times 1 \text{ cm}$. The target-included model is graphically summarized in Fig. 2.

Two methods have been recently proposed to incorporate the target information to the kinematic model [30,17]. We refer the

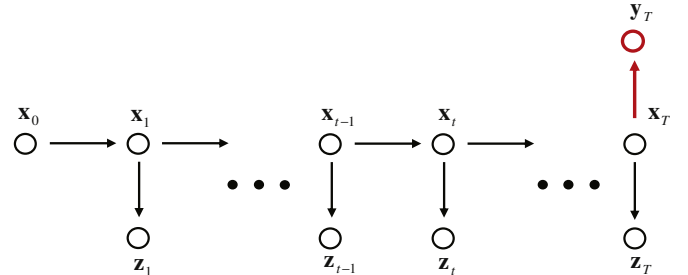


Fig. 2. Graphical model for the target-included model. The dynamics of the hand kinematics, \mathbf{x}_t , is assumed Markovian over time starting at the initial \mathbf{x}_0 (horizontal black lines). The target (red circle), \mathbf{y}_T , is assumed a function of the hand state at the end of movement (vertical red line). The neural firing rate at each time, \mathbf{z}_t , is a function of the current hand state (vertical black lines). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

first method [30] as Target-Included Method A (TIM-A), and the second method [17] as Target-Included Method B (TIM-B).

(i) *TIM-A*: Here the authors proposed to incorporate the target \mathbf{y}_T to Eq. (1) if \mathbf{y}_T is known in the decoding process. The transition probability for the kinematics at time t is

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_T) = p(\mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t | \mathbf{x}_{t-1}, \mathbf{y}_T).$$

Therefore, the hand state at time t can be written in the following form:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{u}_t + \boldsymbol{\epsilon}_t, \quad (4)$$

where $\mathbf{u}_t = E(\mathbf{w}_t | \mathbf{x}_{t-1}, \mathbf{y}_T)$, and $\boldsymbol{\epsilon}_t \sim N(0, \tilde{\mathbf{W}}_t)$, $\tilde{\mathbf{W}}_t = Cov(\mathbf{w}_t | \mathbf{x}_{t-1}, \mathbf{y}_T)$. \mathbf{u}_t and $\tilde{\mathbf{W}}_t$ can be computed via standard multivariate statistics formulae (see detailed procedure in [30]).

Based on Eq. (4), \mathbf{u}_t has the form

$$\mathbf{u}_t = \mathbf{M}_t(\mathbf{y}_T - \mathbf{G}\mathbf{A}^{T-t+1}\mathbf{x}_{t-1}),$$

where the matrix \mathbf{M}_t is a function of $\mathbf{G}, \mathbf{A}, \mathbf{W}$, and \mathbf{V} . Therefore,

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{u}_t + \boldsymbol{\epsilon}_t = (\mathbf{I} - \mathbf{M}_t\mathbf{G}\mathbf{A}^{T-t})\mathbf{A}\mathbf{x}_{t-1} + \mathbf{M}_t\mathbf{y}_T + \boldsymbol{\epsilon}_t. \quad (5)$$

Eq. (5) is the new system equation with the target included. It can be combined with any measurement equation (relation between neural activity and kinematics) to form a state-space model. It will be used (instead of Eq. (1)) in the decoding process which exploits the information from the target.

(ii) *TIM-B*: In this method the decoding is looked as a forward and backward process, and the target is only used in the backward part. Assuming the neural observation up to time t is $\mathbf{z}_{0:t}$, the estimate at time t is

$$p(\mathbf{x}_t | \mathbf{x}_0, \mathbf{y}_T, \mathbf{z}_{0:t}) = (1/Z)p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0)p(\mathbf{y}_T | \mathbf{x}_t) \\ = (1/Z)p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0) \int p(\mathbf{y}_T | \mathbf{x}_T)p(\mathbf{x}_T | \mathbf{x}_t) d\mathbf{x}_T \quad (6)$$

where \mathbf{x}_0 is the initial and Z is a normalizing constant.

To get a direct computation without recursion, the authors used a continuous version of the system equation (Eq. (1)); that is,

$$\dot{\mathbf{x}}_t = \mathbf{R}\mathbf{x}_t + \rho + \mathbf{w}_t, \quad (7)$$

where $\mathbf{R} \in \mathbb{R}^{4 \times 4}$ corresponds to a transition matrix and $\rho \in \mathbb{R}^4$ denotes a drift term. Using the general solution to this linear stochastic differential equation (SDE)

$$\mathbf{x}_t = e^{\mathbf{R}(t-t_0)}\mathbf{x}_0 + \mathbf{R}^{-1}(e^{\mathbf{R}(t-t_0)} - 1)\rho + \int_{t_0}^t e^{\mathbf{R}(t-s)}\mathbf{w}(s) ds,$$

we can get the mean and covariance for the normal distribution $p(\mathbf{x}_T | \mathbf{x}_t)$. That is,

$$E(\mathbf{x}_T | \mathbf{x}_t) = e^{\mathbf{R}(T-t)}\mathbf{x}_t + \mathbf{R}^{-1}(e^{\mathbf{R}(T-t)} - 1)\rho,$$

$$Cov(\mathbf{x}_T | \mathbf{x}_t) = e^{\mathbf{R}T} \left[\int_t^T e^{-s\mathbf{R}} \mathbf{W} (e^{-s\mathbf{R}})^T ds \right] (e^{\mathbf{R}T})^T. \quad (8)$$

Based on this result, we can compute $p(\mathbf{y}_T | \mathbf{x}_t) = \int p(\mathbf{y}_T | \mathbf{x}_T) p(\mathbf{x}_T | \mathbf{x}_t) d\mathbf{x}_T$. This provides the backward probability on \mathbf{x}_t when \mathbf{y}_T is known. Combining this with the forward probability $p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0)$, the final estimate on \mathbf{x}_t can be obtained. *TIM-B* provides a direct way to estimate the state at any time t . However, for an efficient computation of the covariance $Cov(\mathbf{x}_T | \mathbf{x}_t)$, the matrix \mathbf{R} is assumed to be diagonalizable. That is, if there exists an invertible matrix \mathbf{B} and a diagonal matrix $\mathbf{D} = \text{diag}\{d_i\}$ such that $\mathbf{R} = \mathbf{B}\mathbf{D}\mathbf{B}^{-1}$, then

$$Cov(\mathbf{x}_T | \mathbf{x}_t) = \mathbf{B}\mathbf{U}_{(T-t)}\mathbf{B}',$$

where $\mathbf{U}_{(T-t)} = \int_t^T e^{(T-s)\mathbf{D}}\mathbf{B}^{-1}\mathbf{W}(\mathbf{B}^{-1})'(e^{(T-s)\mathbf{D}})' ds$. As \mathbf{D} is diagonal, each (ij) entry in $\mathbf{U}_{(T-t)}$ can be calculated as follows:

$$[\mathbf{U}_{(T-t)}]_{ij} = \int_t^T e^{d_i(T-s)}[\mathbf{B}^{-1}\mathbf{W}(\mathbf{B}^{-1})']_{ij}e^{d_j(T-s)} ds \\ = \frac{e^{(d_i+d_j)(T-t)} - 1}{d_i+d_j} [\mathbf{B}^{-1}\mathbf{W}(\mathbf{B}^{-1})']_{ij}, \quad (9)$$

where the subscript ij indicates the (ij) -th entry of the matrix.

Note that if \mathbf{D} is not diagonal, the computation of the covariance cannot be simplified to the efficient form in Eq. (9). This is true even for any Jordan block matrices; nonzero off-diagonal terms could make the matrix exponential terms difficult to compute and significantly complicate the integration process. Another (relatively minor) issue is that in addition to being diagonalizable, all the eigenvalues in \mathbf{R} are desired to be real. This is because complex values will introduce extra computational burden in the process. We will show this result in one of the datasets in Section 3.

2.2.2. Target-included method using discrete system equation with forward-backward procedure

Focusing on an efficient estimation, we adopt the same discrete system equation as in *TIM-A*, but use the forward-backward procedure as in *TIM-B* for the decoding computation. This new method is referred to as *TIM-C*. That is, the kinematic transition and target information in *TIM-C* are described by Eqs. (1) and (2), respectively. In this case, the covariance matrix $Cov(\mathbf{x}_T | \mathbf{x}_t)$ is recursively computed during the backward propagation, which does not change the computational order of the

process. In contrast to *TIM-B*, there is no restriction on the linear transition \mathbf{A} .

In addition to the system equation that describes the transition of state over time, a state-space model also includes a measurement equation which describes the relation between neural activity and kinematic state. To focus on kinematic representation with the target information, we adopt a Kalman filter, the most classical state-space model which assumes a linear Gaussian relationship between firing rates and hand motion (see Fig. 2). That is, the measurement is described using the following equation:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{q}_t, \quad (10)$$

where $\mathbf{z}_t \in \mathbb{R}^C$ represents a $C \times 1$ vector containing the firing rates at time t_k for C observed neurons. $\mathbf{H} \in \mathbb{R}^{C \times 4}$ is the linear coefficient matrix. The noise term \mathbf{q}_t is assumed zero mean and normally distributed, i.e. $\mathbf{q}_t \sim N(0, \mathbf{Q})$, $\mathbf{Q} \in \mathbf{S}_+^C$. The measurement in Eq. (10) and the system in Eq. (1) constitute a Kalman filter model. All parameters $\mathbf{A}, \mathbf{W}, \mathbf{H}, \mathbf{Q}$ in the model can be identified with closed-form solutions [34].

Filtering decoding in TIM-C: With parameters identified, the hand kinematics can be inferred from neural activity using a backward-forward procedure. Without the derivations in SDE, the target probability, $p(\mathbf{y}_T | \mathbf{x}_t)$, can be recursively computed.

Based on Eqs. (1) and (2), we have

$$\mathbf{y}_T = \mathbf{G}(\mathbf{A}\mathbf{x}_{T-1} + \mathbf{w}_T) + \mathbf{v} \\ = \mathbf{G}\mathbf{A}\mathbf{x}_{T-1} + \mathbf{G}\mathbf{w}_T + \mathbf{v} \\ = \mathbf{G}\mathbf{A}(\mathbf{A}\mathbf{x}_{T-2} + \mathbf{w}_{T-1}) + \mathbf{G}\mathbf{w}_T + \mathbf{v} \\ = \mathbf{G}\mathbf{A}^2\mathbf{x}_{T-2} + \mathbf{G}(\mathbf{A}\mathbf{w}_{T-1} + \mathbf{w}_T) + \mathbf{v} \\ = \dots \\ = \mathbf{G}\mathbf{A}^{T-t}\mathbf{x}_t + \mathbf{G} \sum_{i=t+1}^T \mathbf{A}^{T-i}\mathbf{w}_i + \mathbf{v}.$$

Therefore,

$$\mathbf{y}_T | \mathbf{x}_t \sim N \left(\mathbf{G}\mathbf{A}^{T-t}\mathbf{x}_t, \sum_{i=t+1}^T (\mathbf{G}\mathbf{A}^{T-i})\mathbf{W}(\mathbf{G}\mathbf{A}^{T-i})' + \mathbf{V} \right). \quad (11)$$

For any forward filtering estimation such as Kalman filtering, we can combine it with Eq. (11) and obtain the final estimate of \mathbf{x}_t . As both the filtering distribution and target distribution are normal, the combination $p(\mathbf{x}_t | \mathbf{x}_0, \mathbf{y}_T, \mathbf{z}_{0:t}) \propto p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0)p(\mathbf{y}_T | \mathbf{x}_t)$ has a closed-form which preserves the normality of the state \mathbf{x}_t .

Smoothing decoding in TIM-C: Eq. (11) separates target information from filtering estimation and provides an efficient combination. This is because $\mathbf{z}_{0:t}$ and \mathbf{y}_T are independent when conditioned on the knowledge that \mathbf{x}_t is known. However, for a smoothing estimation, the overall firing rates, $\mathbf{z}_{0:T}$, and the target position, \mathbf{y}_T , are not conditionally independent given \mathbf{x}_t . In this case, the goal is to estimate the state at time t with neural observations up to time T ; that is, we are interested in the distribution $p(\mathbf{x}_t | \mathbf{x}_0, \mathbf{y}_T, \mathbf{z}_{0:T})$.

As the measurement model is linear Gaussian, the target model at time T can be combined in the measurement

$$\begin{pmatrix} \mathbf{z}_T \\ \mathbf{y}_T \end{pmatrix} = \begin{pmatrix} \mathbf{H} \\ \mathbf{G} \end{pmatrix} \mathbf{x}_T + \begin{pmatrix} \mathbf{q}_T \\ \mathbf{v} \end{pmatrix}. \quad (12)$$

Here the new measurement is the same as the non-target case, except that at time T the observation also includes the target. The smoothing probability can be estimated as

$$p(\mathbf{x}_t | \mathbf{x}_0, \mathbf{y}_T, \mathbf{z}_{0:T}) = (1/Z)p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0)p(\mathbf{y}_T, \mathbf{z}_{t+1:T} | \mathbf{x}_t), \quad (13)$$

where Z denotes a constant. The forward probability $p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0)$ can be obtained using a standard Kalman filtering procedure. $p(\mathbf{y}_T, \mathbf{z}_{t+1:T} | \mathbf{x}_t)$ denotes the target-included backward probability. The computations for the smoother follow the same procedure as

a standard Kalman smoother except that the backward propagation starts at the target-included distribution

$$p(\mathbf{x}_T | \mathbf{z}_{0:T}, \mathbf{x}_0, \mathbf{y}_T) \propto p(\mathbf{x}_T | \mathbf{z}_{0:T}, \mathbf{x}_0) p(\mathbf{y}_T | \mathbf{x}_T).$$

Assume the forward probability at time T , $p(\mathbf{x}_T | \mathbf{z}_{0:T}, \mathbf{x}_0)$, is normally distributed with mean \mathbf{x}_T^f and covariance \mathbf{P}_T^f . Using Eq. (2), $p(\mathbf{x}_T | \mathbf{z}_{0:T}, \mathbf{x}_0, \mathbf{y}_T)$ is also normally distributed with variance

$$\mathbf{P}_T^s = (\mathbf{P}_T^f + \mathbf{G}^T \mathbf{V}^{-1} \mathbf{G})^{-1}$$

and mean

$$\mathbf{x}_T^s = \mathbf{P}_T^s ((\mathbf{P}_T^f)^{-1} \mathbf{x}_T^f + \mathbf{G}^T \mathbf{V}^{-1} \mathbf{y}_T).$$

The detailed smoothing steps can be found in [9]. Note that the target information is only used in the initial step for the backward propagation. Therefore, the recursive computation based on Eq. (11) (in the filtering estimation) would not be needed in the smoothing procedure. In practice, we have found that for the same target-included model, the computation in the smoothing estimation is actually more efficient than that in the filtering estimation.

Multiple Target Decoding: We have introduced a new approach to include the target information in the neural decoding process. One important extension of this study is the decoding of a sequence of targets during the movement (e.g. the RTP and SP tasks). Assume the times to a sequence of K targets are T_1, \dots, T_K , respectively. For the filtering estimation, we compute the posterior

$$p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0, \{\mathbf{y}_{T_k}\}). \quad (14)$$

This problem has been well addressed in [17] where a recursive estimation is performed based on the normal assumption at each target. We further simplified the estimation by assuming that given the current target, the hand movement at each time t does not depend on any other targets. Therefore, the estimation of the filtering posterior in Eq. (14) can be represented in the following form:

$$\begin{cases} p(\mathbf{x}_t | \mathbf{z}_{0:t}, \mathbf{x}_0, \mathbf{y}_{T_1}) & \text{if } t \leq T_1, \\ p(\mathbf{x}_t | \mathbf{z}_{T_{k-1}:t}, \hat{\mathbf{x}}_{T_{k-1}}, \mathbf{y}_{T_k}) & \text{if } T_{k-1} < t \leq T_k, \end{cases}$$

where $\hat{\mathbf{x}}_{T_{k-1}}$ in the second case is the estimate when $t = T_{k-1}$. It is used as the initial condition for the time period $(T_{k-1}, T_k]$. Similarly, the smoothing estimate $p(\mathbf{x}_t | \mathbf{z}_{0:T}, \mathbf{x}_0, \{\mathbf{y}_{T_k}\})$ can be simplified in the following form:

$$\begin{cases} p(\mathbf{x}_t | \mathbf{z}_{0:T_1}, \mathbf{x}_0, \mathbf{y}_{T_1}) & \text{if } t \leq T_1, \\ p(\mathbf{x}_t | \mathbf{z}_{T_{k-1}:T_k}, \hat{\mathbf{x}}_{T_{k-1}}, \mathbf{y}_{T_k}) & \text{if } T_{k-1} < t \leq T_k, \end{cases}$$

2.2.3. Coupling the time decoding and trajectory decoding

For a target-directed movement, we have proposed a new approach, TIM-C, to include the target in the decoding process. The inclusion is based on the assumption that the arrival time at the target is known before the decoding. However, this precondition may not be necessary in certain experimental paradigms. If the target only appears at certain sequential fixed positions (e.g. in the SP task), our recently developed method can identify the time when each target is reached using the same neural signals [6]. Using this idea, we will explore the coupling of time decoding and trajectory decoding in the context of a landmark-defined movement. To this end, we plan to proceed in two steps: (1) detect landmark times by the identification method; (2) include the detected target to the backward propagation in TIM-C.

Landmark points detection: Here we briefly describe the identification method. The details can be found in [6]. Suppose the spiking activity of the recorded C neurons is associated with a sequence of K landmark events during the course of a certain type

of behavior such as the SP task. When the bin size is sufficiently small, the spiking activity in each neuron can be approximately represented by a Bernoulli process $x_c(1), x_c(2), x_c(3), \dots$, where $x_c(t)$ is 1 if the neuron generates spikes in the t -th time bin $[(t-1)\Delta, t\Delta]$, or 0 otherwise, $c=1, \dots, C$. For $i=1, \dots, K$, the i -th event may be defined as the reaching of the i -th target. The $K-1$ intervals, denoted as I_1, I_2, \dots, I_{K-1} , between the times of consecutive events during an occurrence of the event sequence will be referred to as inter-event intervals (IEIs). For $i=1, \dots, K$, $c=1, \dots, C$, and $j=-B, \dots, B$, let $p_{i,c}(j)$ be the probability that the c -th neuron generates a spike in the j -th time bin away from the i -th event (B is a positive integer). This probability can be estimated using a peri-stimulus time histogram method. For $i=1, \dots, K-1$, we use $q_i(t, \theta_i)$ to model the probability density (e.g. using a Gamma distribution) of the duration of the i -th IEI. The parameter θ_i can be estimated based on the sample as well.

To identify the occurrences of the whole event sequence, the first step is to evaluate the likelihood of the occurrence of a single event in each time bin. For the i -th event and the c -th unit, we introduce $\Phi_{i,c}(j)$, such that for $j=-B, \dots, B$, it is evaluated as

$$\Phi_{i,c}(j) = \log \frac{p_{i,c}(-j)}{1 - p_{i,c}(-j)}.$$

It follows that

log Pr{the i -th event occurs in the t -th time bin}

$$= \sum_{c=1}^C \Phi_{i,c} * x_c(t) + \text{constant}, \quad (15)$$

where $*$ denotes the mathematical convolution. We therefore use

$$F_i(t) = \sum_{c=1}^C \Phi_{i,c} * x_c(t),$$

as the similarity score between the spiking activity starting in the t -th time bin and the pattern associated with the i -th event. By convolving $\Phi_{i,c}$ with the entire binary sequence x_c , the score is obtained for all time bins.

For the i -th IEI, $i=1, \dots, K-1$, and $t \geq 1$, let $G_i(t) = -\log q_i(t, \theta_i)$. Then the logarithm of the probability that the t -th time bin contains the onset of an occurrence of the event sequence with IEIs I_1, \dots, I_{K-1} can be expressed in the following form:

$$l(t, I_1, \dots, I_{K-1}) = \sum_{i=1}^K F_i(t + I_1 + \dots + I_{i-1}) - \sum_{i=1}^{K-1} G_i(I_i) + \text{constant}. \quad (16)$$

We therefore define the score assigned to the time bin t as

$$L(t) = \max_{I_1, \dots, I_{K-1}} l(t, I_1, \dots, I_{K-1}). \quad (17)$$

Up to an additive constant, $L(t)$ is the maximum likelihood that an occurrence of the event sequence starts in the t -th time bin.

The score function $L(t)$ is computed for all time bins using a dynamic time warping [6]. We can then use it to detect occurrences of the event sequence. The detection is based on the assumption that the score at the onset of an occurrence of the event sequence (i.e. the time of the first event) is higher than the scores elsewhere within the neighborhood of the onset. The local maximum points associated with $L(t)$ are therefore identified as potential onsets (times when the first targets are hit) of occurrences of the event sequence. The times when all the subsequential targets are hit are a direct result within the dynamic programming process.

Add the detected landmarks to the trajectory decoding model: Let the positions of the K landmarks in a certain type of behavior be $\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_K$. Assume the occurrence times of all landmarks have been identified and denoted as $\hat{t}_1, \hat{t}_2, \dots, \hat{t}_K$. When the last target

L_K in a trial is reached, the new target will be the first one, L_1 again, for the next trial. Therefore the target at each time t is calculated as,

$$\hat{\mathbf{y}}_{t_k} = \begin{cases} L_1 & \text{if } t < \hat{t}_1 \text{ or } t \geq \hat{t}_K, \\ L_{k+1} & \text{if } \hat{t}_k \leq t < \hat{t}_{k+1}, \quad k = 1, \dots, K-1. \end{cases} \quad (18)$$

Using the above identified target, the new target model is

$$\hat{\mathbf{y}}_{t_k} = \mathbf{G}\mathbf{x}_{t_k} + \mathbf{v}_k, \quad (19)$$

where \mathbf{G} is given in Eq. (3) and $\mathbf{v}_k \sim N(0, \mathbf{W}_k)$. The value of \mathbf{W}_k will be specified based on the size of the target. We let it be an identity matrix for the data in this study as the size of the target is $1 \text{ cm} \times 1 \text{ cm}$. Eq. (19) is the same as Eq. (2) except that the target at each time is estimated from the identification method. Eqs. (1), (10), (19) constitute a target-included state-space model. The decoding can be performed using a forward-backward procedure for both filtering and smoothing estimations.

3. Results

3.1. TIM-C decoding in the RTP task

The total number of trials in the two datasets from Monkeys 1 and 2 where the RTP task was performed are 550 and 400, respectively. Here we take the first 100 trials in each set as the training data to identify the model. Each training part is around 8 min long and provides sufficient samples to identify the parameters [35]. Then we conduct decoding of the neural firing rates to reconstruct the hand trajectories in the remaining testing part (450 and 300 trials, respectively). The decoding software was written in Matlab (Mathworks Inc., Natick, MA), and the experiments were run on a computer with 8 Intel(R) Xeon(R) 2.00 GHz CPU processors and 12.0 GB of RAM.

In the RTP task, the monkey needs to reach a sequence of seven targets in each trial. The trial starts when the first target is hit. Therefore, the *actual* number of targets is six during the movement. We investigate the effect of targets on the performance of neural decoding, where the number of targets varies from 1, to 2, to 3, and then to 6. The one target case is on target 7 (last target); the two target case is on targets 4 and 7; the three target case is on targets 3, 5, and 7; and the six target case is on targets 2–7.

Decoding performance: We chose the commonly used mean squared error (MSE in the units of cm^2) and the correlation coefficient (CC, no units) to measure decoding accuracy in each testing trial. The decoding results in Dataset 1 are summarized in Table 1. When no target is included, the decoding results (in the row starting with ‘None’) are estimated by the classical Kalman filter and Kalman smoother, respectively. It is apparent that including target information significantly improves the performance of decoding. The MSE decreases as a function of number of targets included, and the largest decrease (when six targets are used) is more than 50% in both filtering and smoothing results. Likewise, the CC (for x - and y -position, respectively) increases

Table 1
Averaged decoding accuracy in Dataset 1 (RTP task).

Targets included	Filtering		Smoothing	
	MSE	(CC _x , CC _y)	MSE	(CC _x , CC _y)
None	7.53	(0.87, 0.84)	6.36	(0.92, 0.87)
Target 7	7.02	(0.88, 0.84)	5.72	(0.93, 0.88)
Targets 4, 7	6.17	(0.89, 0.85)	5.02	(0.93, 0.89)
Targets 3, 5, 7	5.35	(0.90, 0.86)	4.42	(0.94, 0.89)
Targets 2–7	3.40	(0.93, 0.90)	2.75	(0.95, 0.91)

Table 2
Averaged decoding accuracy in Dataset 2 (RTP task).

Targets included	Filtering		Smoothing	
	MSE	(CC _x , CC _y)	MSE	(CC _x , CC _y)
None	7.76	(0.87, 0.85)	6.46	(0.90, 0.87)
Target 7	7.41	(0.88, 0.86)	5.84	(0.91, 0.88)
Targets 4, 7	6.17	(0.89, 0.87)	4.96	(0.92, 0.89)
Targets 3, 5, 7	5.19	(0.91, 0.88)	4.07	(0.93, 0.90)
Targets 2–7	3.21	(0.94, 0.91)	2.56	(0.96, 0.92)

with respect to the number of targets included. The largest improvement is about 7% in the filtering and 4% in the smoothing.

Table 2 shows the decoding results in Dataset 2. We also see that the target-included methods significantly improve the decoding accuracy over the regular Kalman filter and Kalman smoother. Some examples of reconstructed trajectories under all conditions (number of targets included, filtering or smoothing) are shown in Fig. 3. It is apparent that the decoding gets improved with target information included (most improvement around the target position).

Comparison with other target-included methods: TIM-C adopts the mathematical models of TIM-A on hand kinematics, target position, and neural activity, and therefore, the decoding accuracy of these two methods are nearly identical (minor difference at initial conditions). The result when all six targets are used is shown in Table 3. For a direct comparison, we also show the result in TIM-C (repeat the filtering estimates from Tables 1 and 2 when all targets are used). The main difference between these two methods is computational efficiency. Note that TIM-A incorporates the target information into the system equation with a multivariate estimation for mean and covariance. The transformation generates a new drift term ($\mathbf{M}_t \mathbf{y}_t$ in Eq. (5) which results in more computational steps than the standard forward-backward procedure in TIM-C. More importantly, the new state equation depends on a specific target position, and therefore the error covariances during the decoding procedure cannot be computed beforehand with the identified model. In contrast, this is not an issue in the TIM-C method as all error covariances can be computed before the decoding estimation on the testing data [34]. This off-line pre-computation could significantly reduce the computational burden. Indeed, we find that the averaged time costs (when all six targets are used) in Dataset 1 are 69 ± 1.5 s for TIM-A and 6.0 ± 0.26 s for TIM-C, respectively. In Dataset 2 the costs are 54 ± 3.8 s for TIM-A and 4.8 ± 0.23 s for TIM-C, respectively.

The TIM-B method is based on a continuous system equation (Eq. (7)). The decoding can be obtained at any time t in an efficient, non-recursive manner. Using a classical backward Euler method

$$\dot{\mathbf{x}}_t \approx (\mathbf{x}_t - \mathbf{x}_{t-1})/\Delta t,$$

we can identify \mathbf{R} in both datasets via simple AR models. We find that the eigenvalues of \mathbf{R} are -1.69 , -3.31 , -8.13 , and -6.67 in Dataset 1, and -1.07 , -11.87 , $-3.49 + 2.37i$, and $-3.49 - 2.37i$ in Dataset 2. Thanks to these distinct eigenvalues, the matrix \mathbf{R} in these two datasets are both diagonalizable though the diagonalization in Dataset 2 can only be in the complex domain (not diagonalizable in the real domain as two eigenvalues are complex). As the TIM-C model is only a discrete version of TIM-B, their decoding results are also nearly identical (see averaged decoding accuracy in Table 3). To compare the efficiency, we also compute the averaged time cost of TIM-B in both datasets. We found that in Dataset 1 the cost is 6.1 ± 0.26 s, nearly the same efficiency as that in TIM-C. In Dataset 2 the cost is 5.7 ± 0.02 s, which is about

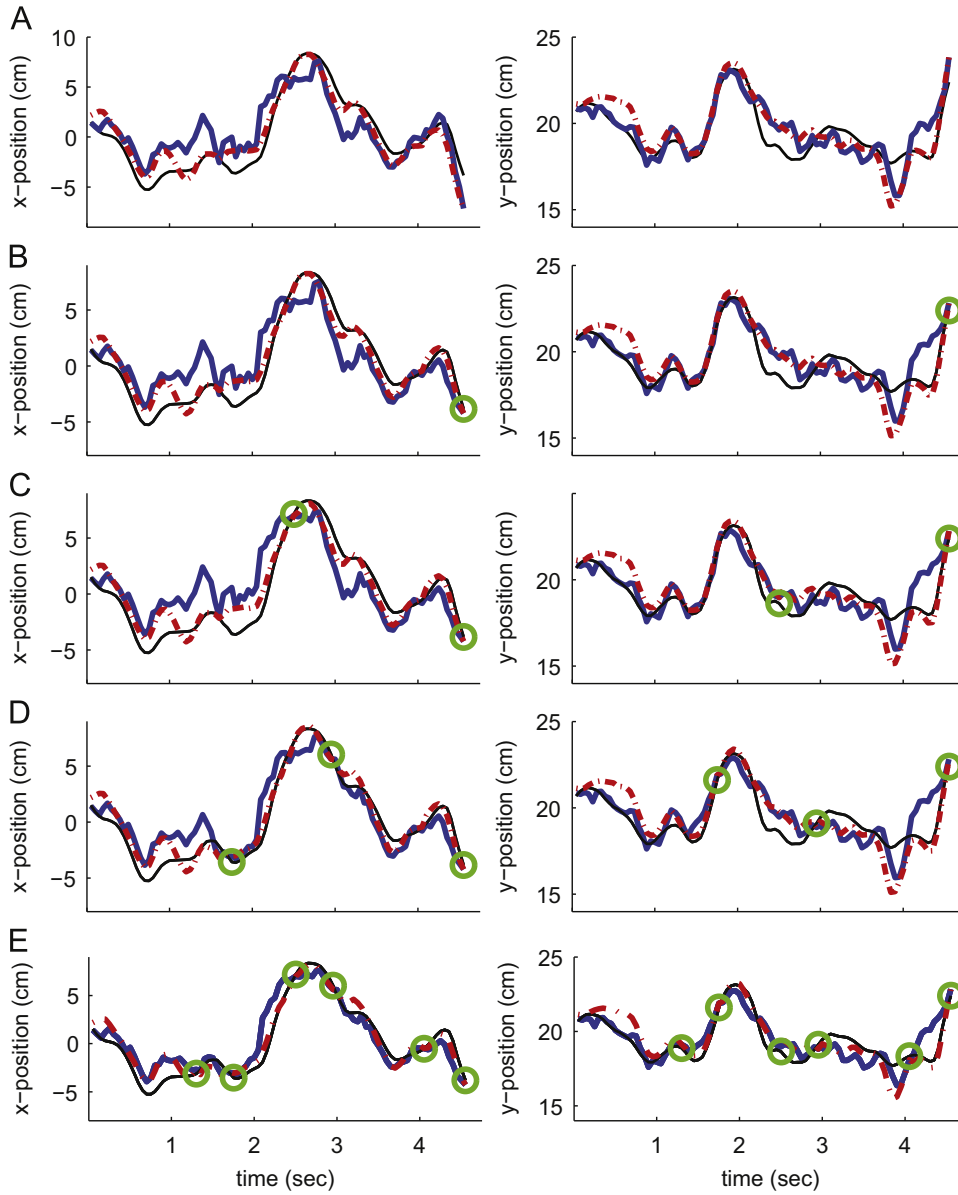


Fig. 3. Reconstruction examples of the RTP task where the left panels are for x -position and the right panels are for y -position. A. True x - and y -position (thin black) of one test trial and its reconstructions using the classical Kalman filter (solid blue) and Kalman smoother (dash-dot red). B. Same as A except that target 7 is included in the decoding process. The green circle on each true position line indicates the time when the target is reached. C. Same as B except that targets 4 and 7 are included in the decoding process. D. Same as B except that targets 3, 5, and 7 are included in the decoding process. E. Same as B except that targets 2–7 are included in the decoding process. We see that the decoding gets improved with target information included (better reconstruction around green circles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Averaged decoding accuracy comparison.

Method	Dataset 1			Dataset 2		
	MSE	(CC_x, CC_y)	Time cost	MSE	(CC_x, CC_y)	Time cost
TIM-C	3.40	(0.93, 0.90)	6.0 ± 0.26 s	3.21	(0.94, 0.91)	4.8 ± 0.23 s
TIM-A	3.42	(0.94, 0.91)	69 ± 1.5 s	3.19	(0.94, 0.91)	54 ± 3.8 s
TIM-B	3.44	(0.94, 0.89)	6.1 ± 0.26 s	3.26	(0.94, 0.91)	5.7 ± 0.02 s
Regression	5.74	(0.90, 0.87)	–	5.27	(0.91, 0.88)	–

20% slower than that in TIM-C due to the complex-valued computation.

In [20], the target is incorporated into the hand kinematic state, that is, the new hand state $\tilde{\mathbf{x}}_t = [x, y, v_x, v_y, p_x, p_y]^T_t$ represents

x -position, y -position, x -velocity, y -velocity, target x -position, and target y -position at time t . The new state and neural activity are then represented via a standard Kalman filter. This target-included regression model can be identified with least squares on a training set where the hand state, neural activity, and all six targets in each trial are used. Once the model is identified, the decoding will be conducted with a Kalman filter algorithm. The decoding result of this method on both datasets is also shown in Table 3. We find that the TIM-C method has significantly better decoding accuracy than the regression method for both filtering and smoothing estimations; the MSEs in TIM-C are about 40–50% lower than those in the regression method.

Other target-included methods such as [10,37] represent the target as the endpoint of the hand where the behavioral task is often a stereotyped center-out movement and the trajectory is reasonably consistent for each given target position. The decoding

in these methods are conducted in two steps: (1) identify the target position (often using neural activity in delayed period before the reaching movement); (2) estimate trajectory with the target-known likelihood model. In general, the movement

behavior in these methods is largely known when the target position is known. In contrast, in TIM-C we only assume that the hand will finally reach that position which has no restriction on the trajectory form and imposes minimal restriction on the motor behavior. We have shown that this type of target inclusion achieves significantly better accuracy than a regression-type target inclusion [20]. Note that the models in [10,37] cannot be directly used in the RTP behavior. We therefore will not have any direct comparison with them on the given two datasets.

3.2. Coupling decoding in the SP task

Here we test the decoding performance in the SP task when the target information is included. One key difference from that in

Table 4
Averaged decoding accuracy in Dataset 3 (SP task).

Method	Filtering		Smoothing	
	MSE	(CC _x , CC _y)	MSE	(CC _x , CC _y)
No targets	7.23	(0.93, 0.91)	6.17	(0.94, 0.93)
True targets	4.16	(0.97, 0.92)	4.37	(0.97, 0.92)
Estimated targets	4.70	(0.96, 0.92)	4.83	(0.96, 0.92)

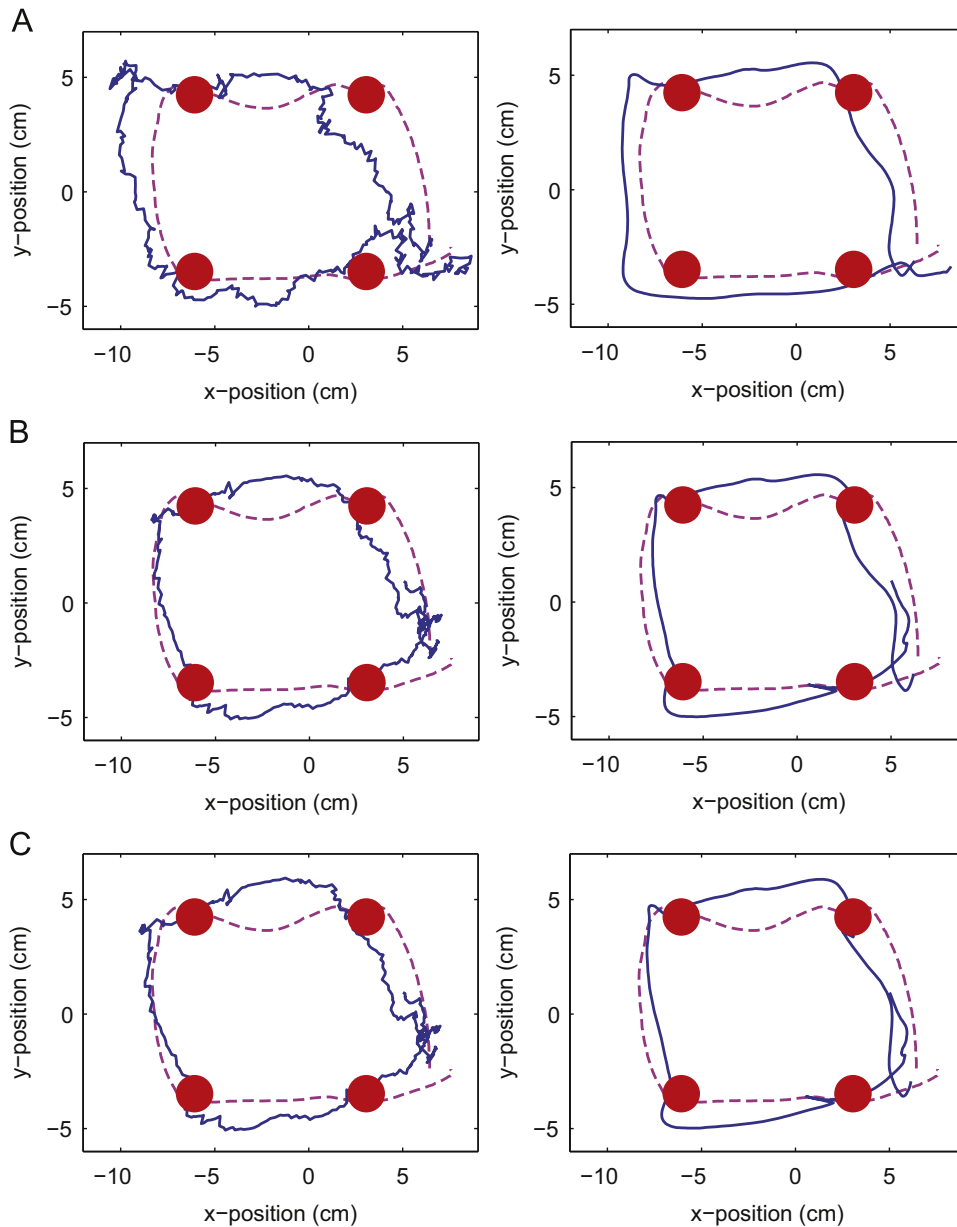


Fig. 4. Reconstruction examples of the SP task where the left panels are for filtering estimation and the right panels are for smoothing estimation. A. True 2-dimensional position trajectory (dashed magenta) of one test trial and its reconstructions using the Kalman filter (solid blue, left panel) and Kalman smoother (solid blue, right panel). The four red spots denote the four corners in the SP movement. B. Same test trial as in A and its reconstructions using the TIM-C method with true reaching time at four targets. C. Same test trial as in A and its reconstructions using the TIM-C method with estimated reaching time at four targets. It is apparent that the decoded trajectories are closer to the true trajectory when the target information is used (better reconstruction around the four corners). Moreover, the smoothing procedures produce much smoother trajectories which are more similar to the real hand movement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the RTP task is that the time when each target is reached can be identified using the same neural signals. This is because the SP task is a stereotyped movement (passing through four fixed corners on a square). Over the 240 trials, we take the first 200 trials as the training set (for parameter estimation), and the remaining 40 as the testing set (for neural decoding).

Coupling the time decoding and trajectory decoding: Using the identification method in [6], we found that over the 40 test trials, 29 of them can be accurately detected where the averaged error was only 0.223 ± 0.200 s (over four corners). As the target only appeared at the fixed four corners of the square, we obtained the target position at each time using Eq. (18). Then the detected target was incorporated into the linear Gaussian model on the target (Eq. (19)). Here we compare the decoding performance of the TIM-C method with that of the classical Kalman filter and Kalman smoother for the 29 trials where the target times are not used. Experimental results are summarized in Table 4. When the true target information is used, the decoding error (MSE) is reduced by about 42% in the filtering procedure, and about 29% in the smoothing procedure. These improvements are consistent to those in the RTP task. More importantly, when the estimated target information is used, we also observe significant improvements—about 35% in the filtering procedure, and about 22% in the smoothing procedure. This indicates that the estimated target times are very accurate and decodings based on them can perform nearly as well as those based on the true target times.

The improved performance of the target-included methods is illustrated by a few typical examples (see Fig. 4). We see that when target information is used, the decoded trajectories will be able to pass all the targets, significantly improving the decoding over the non-target-included methods. Note that a smoothing procedure produces a much smoother trajectory similar to the true hand movement. However, we note that the better shape does not necessarily mean the decoding is more accurate (see Table 4, MSE is lower in filtering procedure for both true and estimated target cases). This is somewhat surprising as smoothing estimation is often more accurate than filtering estimation. Further work will be conducted to investigate this issue.

4. Discussion

In this article, we propose a new computational procedure for performing target-included decoding in the motor cortex. The new approach, TIM-C, is mainly based on two previous methods: TIM-A in [30] and TIM-B in [17]. TIM-C adopts the same model as in TIM-A but uses the forward–backward propagation in TIM-B for computation. We show that the TIM-C decoding can avoid certain limitations in the TIM-B method, where the matrix \mathbf{R} needs to be diagonalizable with real eigenvalues for an efficient estimation. In TIM-C, the covariance at each time $Cov(\mathbf{y}_T | \mathbf{x}_t)$ in Eq. (11) can be recursively obtained for any parameter values. In a decoding procedure where recursion is necessary in the forward filtering such as a Kalman filter algorithm, the backward propagation in TIM-C adds only linearly computational burden. As compared to TIM-A, the target information in TIM-C is only used during the backward propagation. It avoids the relatively more complicated computations for the drift term in the target-included state transition [30]. Moreover, as the error covariances in the forward propagation only depend on the fitted model, they can be computed before the analysis of the testing data. This pre-computation could significantly accelerate the estimation process which might be useful for on-line neural control applications.

Finally, we note that the TIM-C is based on the assumption that the target location and arrival time need to be known before the decoding process. However, these limitations are not as strongly imposed in previous methods where the targets can be dynamic over the time process [28].

We examined the target-included kinematic decoding and show that the target is actually used only in the backward step. The forward step is a filtering estimation with any measurement model. To simplify the procedure, we use a linear Gaussian framework to describe the relationship between neural activity and hand motion. With this linear state-space assumption, the classical Kalman filter algorithm can be employed in the forward step. A recent comparison of neural decoding methods showed that the Kalman filter has comparable performance with various GLM-based methods [14]. Indeed, the implementation of the Kalman filter is also much simpler which leads to efficient decoding desirable in on-line neural control as well as for practical prosthetic applications [20,12]. Our future work will explore other Bayesian formulations where the likelihood model can be any non-linear equation such as GLM-based Poisson or non-Poisson models [22,33,30] or mixture models [29]. A non-linear likelihood and the target-included prior constitute a more general state-space model [23]. These types of models are more theoretically appropriate as the neural data used is in the form of binned spike counts, which are not continuous variables as assumed in a linear Gaussian framework.

Our recent work showed that for a landmark-defined movement, an identification on the time to each target provides an alternative way to reconstruct the hand motion [6]. Time decoding and trajectory decoding each have advantages. A time decoding can reconstruct the main structure, while it lacks the details between landmarks. In contrast, a trajectory decoding can reconstruct each point in the movement, while it is difficult to identify the main structure of a landmark-defined trajectory. For stereotyped movements such as the SP task we combine these two different decoding methods to fully exploit the benefits from each approach. In this case, the target information does not need to be known beforehand and can be estimated from the same neural signals. We then incorporate the identified target into the TIM-C and demonstrate that such coupling results in more effective decoding. Experimental results show that the combination is effective as more accurate decoding is obtained based on the same neural activity. However, we note that the current time decoding method is only based on stereotyped movements, and that it is an off-line method as all neural signals are needed to identify the targets. We will explore an on-line identification method where only the past and current neural activity is used. Moreover, the coupling method was only used in 29 trials over the total 40 testing trials. This is because time decoding method was only successful in these 29 trials [6]. In our future work, we will investigate ways to improve the time decoding performance and make the coupling method more generally applicable. Finally, we will explore including the target information into the likelihood model, such as using linear or non-linear regression [20]. Such an approach would provide more feasible control in neural motor prostheses.

Acknowledgments

We thank S. Francis, Z. Haga, D. Paulsen, and J. Reimer for training the monkeys and collecting the data. WW is supported by an NSF grant IIS-0916154, and NGH is supported by an NIH-NINDS grant R01 NS45853.

References

- [1] N. Achtman, A. Afshar, G. Santhanam, B.M. Yu, S.I. Ryu, K.V. Shenoy, Free-paced high-performance brain–computer interfaces, *J. Neural Eng.* 4 (2007) 336–347.
- [2] H.S. Bokil, B. Pesaran, R.A. Andersen, P.P. Mitra, A method for detection and classification of events in neural activity, *IEEE Trans. Biomed. Eng.* 53 (2006) 1678–1687.
- [3] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [4] A.E. Brockwell, R.E. Kass, A.B. Schwartz, Statistical signal processing and the motor cortex, *Proc. IEEE* 95 (2007) 1–18.
- [5] A.E. Brockwell, A.L. Rojas, R.E. Kass, Recursive Bayesian decoding of motor cortical signals by particle filtering, *J. Neurophysiol.* 91 (2004) 1899–1907.
- [6] Z. Chi, W. Wu, Z. Haga, N. Hatsopoulos, D. Margoliash, Template-based spike pattern identification with linear convolution and dynamic time warping, *J. Neurophysiol.* 97 (2007) 1221–1235.
- [7] J.P. Cunningham, B.M. Yu, V. Gilja, S.I. Ryu, K.V. Shenoy, Toward optimal target placement for neural prosthetic devices, *J. Neurophysiol.* 100 (2008) 3445–3457.
- [8] U. Eden, L. Frank, R. Barbieri, V. Solo, E. Brown, Dynamic analysis of neural encoding by point process adaptive filtering, *Neural Comput.* 16 (2004) 971–988.
- [9] S. Haykin, *Kalman Filtering and Neural Networks*, John Wiley & Sons, Inc., 2001.
- [10] C. Kemere, K.P. Shenoy, T.H. Meng, Model-based neural decoding of reaching movements: a maximum likelihood approach, *IEEE Trans. Biomed. Eng.* 5 (2005) 353–356.
- [11] C. Kemere, T.H. Meng, Optimal estimation of feed-forward-controlled linear systems. in: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, 2005, pp. 353–356.
- [12] S. Kim, J.D. Simeral, L.R. Hochberg, J.P. Donoghue, M.J. Black, Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia, *J. Neural Eng.* 5 (2008) 455.
- [13] S.P. Kim, J.C. Sanchez, J.C. Principe, Real time input subset selection for linear time-variant MIMO systems, *Optim. Methods Software* 22 (1) (2007) 83–98.
- [14] S. Koyama, S.M. Chase, A.S. Whitford, M. Velliste, A.B. Schwartz, R.E. Kass, Comparison of brain–computer interface decoding algorithms in open-loop and closed-loop control, *J. Comput. Neurosci.* 29 (2010) 73–87.
- [15] S. Koyama, L.C. Pérez-Bolde, C.R. Shalizi, R. Kass, Approximate methods for state-space models, *J. Am. Stat. Assoc.* 105 (489) (2010) 170–180.
- [16] J.E. Kulkarni, L. Paninski, Common-input models for multiple neural spike-train data, *Network: Comput. Neural Syst.* 18 (2007) 375–407.
- [17] J.E. Kulkarni, L. Paninski, State-space decoding of goal directed movement, *IEEE Signal Process. Mag.* 25 (1) (2008) 78–86.
- [18] V. Lawhern, W. Wu, N.G. Hatsopoulos, L. Paninski, Population decoding of motor cortical activity using a generalized linear model with hidden states, *J. Neurosci. Methods* 2 (2010) 267–280.
- [19] D. Moran, A. Schwartz, Motor cortical representation of speed and direction during reaching, *J. Neurophysiol.* 82 (5) (1999) 2676–2692.
- [20] G.H. Mulliken, S. Musallam, R.A. Andersen, Decoding trajectories from posterior parietal cortex ensembles, *J. Neurosci.* 28 (48) (2008) 12913–12926.
- [21] D. Nykamp, A mathematical framework for inferring connectivity in probabilistic neuronal networks, *Math. Biosci.* 205 (2007) 204–251.
- [22] L. Paninski, Maximum likelihood estimation of cascade point-process neural encoding models, *Network: Comput. Neural Syst.* 15 (2004) 243–262.
- [23] L. Paninski, Y. Ahmadian, D. Ferreira, S. Koyama, K. Rahnema Rad, M. Vidne, J. Vogelstein, W. Wu, A new look at state-space models for neural data, *J. Comput. Neurosci.* 29 (2010) 107–126.
- [24] L. Paninski, M. Fellows, N. Hatsopoulos, J.P. Donoghue, Spatiotemporal tuning of motor cortical neurons for hand position and velocity, *J. Neurophysiol.* 91 (2004) 515–532.
- [25] J.C. Sanchez, D. Erdogmus, J.C. Principe, J. Wessberg, M.A.L. Nicolelis, Interpreting spatial and temporal neural activity through a recurrent neural network brain machine interface, *IEEE Trans. Neural Syst. Rehabil. Eng.* 13 (2005) 213–219.
- [26] S.H. Scott, Apparatus for measuring and perturbing shoulder and elbow joint positions and torques during reaching, *J. Neurosci. Methods* 89 (1999) 119–127.
- [27] K.V. Shenoy, D. Meeker, S. Cao, S.A. Kureshi, B. Pesaran, C.A. Buneo, A.P. Batista, P.P. Mitra, J.W. Burdick, R.A. Andersen, Neural prosthetic control signals from plan activity, *NeuroReport* 14 (4) (2003) 591–597.
- [28] L. Srinivasan, E.N. Brown, A state-space analysis for reconstruction of goal-directed movements using neural signals, *IEEE Trans. Biomed. Eng.* 54 (3) (2007) 526–535.
- [29] L. Srinivasan, U.T. Eden, S.K. Mitter, E.N. Brown, General-purpose filter design for neural prosthetic devices, *J. Neurophysiol.* 98 (2007) 2456–2475.
- [30] L. Srinivasan, U.T. Eden, A.S. Willsky, E.N. Brown, A state-space analysis for reconstruction of goal-directed movements using neural signals, *Neural Comput.* 18 (2006) 2465–2494.
- [31] I.H. Stevenson, J.M. Rebesco, N.G. Hatsopoulos, Z. Haga, L.E. Miller, K.P. Kording, Bayesian inference of functional connectivity and network structure from spikes, *IEEE Trans. Neural Syst. Rehabil. Eng.* 17 (2009) 203–213.
- [32] W. Truccolo, J.P. Donoghue, Nonparametric modeling of neural point processes via stochastic gradient boosting regression, *Neural Comput.* 19 (2007) 672–705.
- [33] W. Truccolo, U.T. Eden, M.R. Fellows, J.P. Donoghue, E.N. Brown, A point process framework for relating neural spiking activity to spiking history, neural ensemble and extrinsic covariate effects, *J. Neurophysiol.* 93 (2005) 1074–1089.
- [34] W. Wu, Y. Gao, E. Bienenstock, J.P. Donoghue, M.J. Black, Bayesian population decoding of motor cortical activity using a Kalman filter, *Neural Comput.* 18 (1) (2006) 80–118.
- [35] W. Wu, N. Hatsopoulos, Real-time decoding of non-stationary neural activity in motor cortex, *IEEE Trans. Neural Syst. Rehabil. Eng.* 16 (3) (2008) 213–222.
- [36] W. Wu, J.E. Kulkarni, N.G. Hatsopoulos, L. Paninski, Neural decoding of hand motion using a linear state-space model with hidden states, *IEEE Trans. Neural Syst. Rehabil. Eng.* 17 (4) (2009) 370–378.
- [37] B.M. Yu, C. Kemere, G. Santhanam, A. Afshar, S.I. Ryu, T.H. Meng, M. Sahani, K.V. Shenoy, Mixture of trajectory models for neural decoding of goal-directed movements, *J. Neurophysiol.* 97 (2007) 3763–3780.



Vernon Lawhern received the B.S. degree in mathematics from the University of West Florida, Pensacola, FL in 2005 and the M.S. and Ph.D. degrees in Statistics from the Florida State University, Tallahassee, FL in 2008 and 2011, respectively.

He is currently a postdoctoral researcher in the Department of Computer Science at the University of Texas, San Antonio. His research interests include statistical modeling and analysis of neural activity.



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

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



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

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