MIT Open Access Articles

Efficient Decoding With Steady-State Kalman Filter in Neural Interface Systems

Efficient Decoding With Steady-State Kalman Filter in Neural Interface Systems

Wasim Q. Malik[Senior Member, IEEE],
Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with the School of Engineering and the Institute for Brain Research, Brown University, Providence, RI 02912 USA, and also with the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, and also with the Rehabilitation R&D Service, U.S. Department of Veterans Affairs, Providence, RI 02908 USA (wmalik@partners.org).

Wilson Truccolo,
Department of Neuroscience and the Institute for Brain Research, Brown University, Providence, RI 02912 USA, and also with the Department of Neurology, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA (wilson_truccolo@brown.edu).

Emery N. Brown[Fellow, IEEE], and
Department of Brain and Cognitive Sciences, and the MIT-Harvard Division of Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, and also with the Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA (enb@neurostat.mit.edu).

Leigh R. Hochberg
Center for Restorative and Regenerative Medicine, Rehabilitation R&D Service, U.S. Dept. of Veterans Affairs, Providence, RI 02908 USA, and also with the Department of Neurology, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and with the School of Engineering and the Institute for Brain Research, Brown University, Providence, RI 02912 USA (leigh_hochberg@brown.edu).

Abstract

The Kalman filter is commonly used in neural interface systems to decode neural activity and estimate the desired movement kinematics. We analyze a low-complexity Kalman filter implementation in which the filter gain is approximated by its steady-state form, computed offline before real-time decoding commences. We evaluate its performance using human motor cortical spike train data obtained from an intracortical recording array as part of an ongoing pilot clinical trial. We demonstrate that the standard Kalman filter gain converges to within 95% of the steady-state filter gain in 1.5 ± 0.5 s (mean $\pm$ s.d.). The difference in the intended movement velocity decoded by the two filters vanishes within 5 s, with a correlation coefficient of 0.99 between the two decoded velocities over the session length. We also find that the steady-state Kalman filter reduces the computational load (algorithm execution time) for decoding the firing rates of 25 ± 3 single units by a factor of 7.0 ± 0.9. We expect that the gain in computational efficiency will be much higher in systems with larger neural ensembles. The steady-state filter can thus provide substantial runtime efficiency at little cost in terms of estimation accuracy. This far more efficient neural decoding approach will facilitate the practical implementation of future large-dimensional, multisignal neural interface systems.

Keywords

Brain–computer interfaces (BCI); brain–machine interfaces (BMI); Kalman filter; motor cortex; neural decoding; paralysis; spinal cord injury; state-space models; tetraplegia

I. Introduction

NEURAL interfaces systems (NISs) have the potential to restore communication and control abilities for people with tetraplegia secondary to spinal cord injury, amyotrophic lateral sclerosis (ALS), or brainstem stroke [1]–[11]. Intracortical NISs use multielectrode arrays implanted in the cortex to acquire relevant signals from neuronal ensembles. NISs require accurate decoding models to decipher intended movements from neuromotor activity and generate appropriate command signals to control a computer cursor, prosthetic limb, functional electrical stimulation device, or other assistive devices.

At the beginning of the BrainGate[1] NIS pilot clinical trial, a linear decoder was used to predict cursor position from neural activity measured from a neuronal ensemble [6]. Subsequent analysis showed that a Kalman filter (KF), which modeled the movement kinematics as a random walk process, improved the decoding accuracy [12]–[14]. A recursive Bayesian decoder with guaranteed stability and robustness, the Kalman filter provides optimal state estimates along with the associated confidence regions for a linear Gaussian dynamical system [15]–[17]. While other decoding strategies have been proposed [18]–[23], the relative simplicity and good performance of the KF have made it a popular choice for neural decoding in NISs [10], [11], [24]–[29]. The KF used in these studies consisted of the most common implementation in which the observation and state-transition matrices and corresponding noise covariance matrices are assumed to be constant and estimated from training data, while the Kalman gain is adaptive and computed during each filter recursion.

With rapid advances in NIS technology, we anticipate the potential to harness large-dimensional signal sets, such as those as will eventually be obtained from multiple intracortical recoding arrays, as well as more complex signals with increased computational requirements, such as continuous-time multiunit activity (MUA) [30], [31] and local field potentials (LFP) [6], [32]. The KF may, however, be too computationally intensive for real-time decoding of such complex neurophysiological signal sets. Increasing interest in embedded wireless NISs also motivates improved algorithmic efficiency which would lower hardware requirements and extend battery life [33], [34]. We are, therefore, interested in determining whether we can exploit a decoding approach with lower complexity than the KF, bearing in mind that in the decoding accuracy-efficiency tradeoff, we are generally reluctant to sacrifice accuracy merely to decrease computational load or time.

The optimal Kalman gain depends only on the system matrices and initial conditions, and not on the incoming measurements [15]. It is therefore possible to precompute the gain offline and store it for subsequent decoding sessions to reduce real-time decoding complexity. However, the large memory requirement associated with the storage and processing of large Kalman gain matrices to decode complex, multidimensional signal sets renders this approach impracticable.

[1]The research is conducted under an Investigational Device Exemption and approval from the local Institutional Review Board (Caution: Investigational Device. Limited by Federal Law to Investigational Use).

Another alternative, with the potential to improve efficiency with little loss in accuracy, is the steady-state KF (SSKF) [15], [16]. In a time-invariant stochastic system, such as that assumed by the standard KF implementation commonly used in neural decoding [12], the optimal gain of the KF converges to a steady-state value after only a few recursions [16], which motivates the use of the SSKF. Equivalent to an infinite-length digital Wiener filter, the SSKF approximates the optimal time-varying KF gain with a precomputed constant matrix that represents the steady-state value of the filter gain.

Given a state space model with *s* states and *n* observations, the algorithmic complexity of optimal Kalman gain computation for a single decoding recursion is $O\left((\max\{s, n\})^3\right)$, as described in Table I [35]. The state space of a KF-based NIS decoder comprises the motion kinematics (usually position, velocity, and acceleration) in accordance with the encoding properties of motor cortical neurons. A multielectrode array typically spans 40–100 neuronal units (*n*). Since $n \gg s$ in this case, from Table I the KF complexity is effectively $O\left(n^3\right)$, while the SSKF complexity is only $O(n)$. The relative efficiency of the SSKF therefore varies as the square of the ensemble size, which can be consequential for a resource-constrained NIS with large-dimensional signal sets.

In this paper, we investigate the potential utility of SSKF in unique data sets generated by our clinical trial participants. We hypothesize that using neural spiking data, the filter gain could be replaced by its steady-state estimate, reducing the runtime complexity considerably with an acceptably small loss in estimation accuracy during the initial recursions. We analyze and compare the performance of KF and SSKF, investigating whether the latter can simplify practical NIS implementation.

II. Neural Data Acquisition

A. Clinical Trial

The neural data examined here are derived from two participants in the BrainGate pilot clinical trials. These trials were conducted with a U.S. Food and Drug Administration (FDA) Investigational Device Exemption (IDE) and with Institutional Review Board (IRB) approval from Rhode Island Hospital, New England IRB, Spaulding Rehabilitation Hospital and Massachusetts General Hospital.

B. Trial Participants

1) Participant S1—A 24-year-old man with complete tetraplegia due to C4 spinal cord injury from a knife wound sustained three years prior to trial recruitment. The data analyzed here were recorded on two separate days three months after array implantation.

2) Participant S3—A 55-year-old woman with tetraplegia and anarthria secondary to brainstem stroke. She had thrombosis of the basilar artery and extensive pontine infarction nine years before trial recruitment. The data analyzed here were recorded on two separate days 33 months after array implantation.

C. Surgical Procedures

The BrainGate microelectrode array was implanted using a pneumatic technique in the precentral gyrus contralateral to the dominant hand in the region of the arm representation of each participant [6], [36]. The intracortical array consisted of 10×10 silicon microelectrodes that protruded 1 mm (S1) or 1.5 mm (S3) from a 4×4 mm platform. Neural data were recorded from this chronically implanted array.

D. Behavioral Tasks

We collected neural and motion kinematics data during 2-D behavioral pursuit-tracking and center-out tasks [13], [23].

The filter-building data was recorded under open-loop motor imagery, during which the participants observed a training cursor moving on a 19-in screen. The training cursor was controlled by a technician using a standard handheld mouse (random pursuit-tracking for S1) or computer (four-target center-out for S3). The participants imagined controlling the training cursor movement with their own dominant hand. In the pursuit-tracking task, the training cursor moved from a starting location towards a target randomly placed on the screen [Fig. 1(a)]. On acquisition, the target vanished from the screen and the next target appeared simultaneously. The cursor trajectory in this task spanned much of the screen area over the course of each data-acquisition block. In the center-out task, four peripheral targets (0°, 90°, 180°, and 270°) and one center target were displayed during the filter-building phase. One of the targets, selected pseudorandomly, was highlighted and the computer-controlled training cursor moved toward it with a Gaussian velocity profile [Fig. 1(b)]. On reaching the target, the cursor remained stationary for 0.5 s before retracing its path back to the center target.

During the subsequent closed-loop assessment phase for S3, there were eight radial targets and one center target in our center-out task. Successful target acquisition required reaching the target with the neurally driven cursor and simulating a click within the target area. The data recorded during the filter-building period were used to train the filter that was used for closed-loop assessment to drive a neural cursor volitionally by estimating its velocity from the participant's neural activity. The closed-loop assessment for S1, the first participant in our clinical trial, was performed before the implementation of the KF in our decoding platform, and those data are thus not analyzed here.

For the analysis in this paper, we organize the data recorded from the two participants into six sessions, as summarized in Table II. Of these, Sessions 1, 2, 3, and 5 consist of open-loop motor imagery while Sessions 4 and 6 involve closed-loop neural cursor control.

1) Session 1—Motor imagery data were collected from Participant S1 under an open-loop pursuit-tracking task with random target placement in four blocks of about 1 min length each. The data were recorded on Day 86 after array implantation. The participant observed a training cursor controlled by a technician and imagined controlling the cursor movement with his own hand. For our analysis, we divided this 4 min length of data into *N* trial runs lasting 10 s each, carefully taking into account the block discontinuities. Under an *N*-fold cross-validation scheme, we decoded each of the *N* trials sequentially by training the filter on the rest of the trials.

2) Session 2—This data set is similar to Session 1 but the research session occurred on Day 90.

3) Session 3—For filter building by participant S3, an open-loop center-out task was used. She imagined moving a computer-controlled cursor to four fixed-location radial targets. The data contain four blocks of about 90 s each, of which two blocks contain seven slow-speed trials of 12 s each and the other two blocks contain 11 high-speed trials of 8 s each. Each trial consists of the epoch culminating at target acquisition defined as reaching the target location. The remaining session data length consists of brief rest periods between consecutive trials. These data were recorded on Day 1002 after implantation.

4) Session 4—These closed-loop data were collected from Participant S3 also on Day 1002. The entire open-loop data set of Session 3 was used to train the Kalman filter that decoded the closed-loop neural activity during an 8-target center-out task in a single 10 min block.

5) Session 5—This open-loop data set is similar to Session 3 but the research session was conducted on Day 1003.

6) Session 6—This closed-loop data set is similar to Session 4 but the research session was conducted on Day 1003.

E. Data Preprocessing

In each session, we recorded neural signals from the 96-channel array with digitization at 30 kHz (analog bandpass 0.3–7500 Hz; digital highpass with cutoff at 250 Hz) and used real-time, amplitude-thresholding software for waveshape discrimination [37]. Putative single neurons and apparent multineuron activity with consistent waveforms were accepted or rejected for inclusion in the study at the beginning of each session based on visual inspection of the isolated waveforms. Following the identification of single-unit spikes, the spike times were recorded and the spikes were collected in nonover-lapping 100 ms time-bins to obtain the firing rates. The 2-D Cartesian coordinates of the cursor position were recorded periodically and their first-order derivative was used to obtain the cursor velocity used in our state space model. We used a firing rate threshold of 1 Hz, and units firing at a rate lower than this threshold were not used for decoding. In our data, 20–29 units were found to meet this criterion in each session, as described in Table II. Using these data sets for offline analysis, we first trained the KF and SSKF, and then decoded the cursor velocity from which we obtained a reconstruction of the cursor trajectory by integration.

III. Neural Decoding

A. State Space Model

The s kinematic states and the associated firing rates of n spike-sorted units can be related in terms of a time-invariant linear dynamical system [12]. We use the following discrete-time stationary Gauss-Markov stochastic system model:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{w}_k \tag{1}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

where $\mathbf{\Phi}$ and $\mathbf{H}$ denote the constant $s \times s$ state-transition matrix and $n \times s$ observation matrix, respectively. The $s \times 1$ vector $\mathbf{w}_k$ and $n \times 1$ vector $\mathbf{v}_k$ denote the process and measurement noise, respectively. With the states $\mathbf{x}_k$ and observations $\mathbf{y}_k$ representing movement velocity and single-unit firing rates respectively, this state space model implicitly assumes cosine tuning of motor neurons to velocity [12]. The filter is initialized with a multivariate Gaussian state estimate $\widehat{\mathbf{x}}_0 \sim \mathcal{N}(\mathbf{x}_0, \mathbf{P}_0)$. We choose $\mathbf{x}_0 = \mathbf{0}$ and $\mathbf{P}_0 = \mathbf{Q}$ for the decoding problem in this paper. Similar to [12] and [13], we assume that

$$\mathcal{E}\left\{ \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{w} \\ \mathbf{v} \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{w} \\ \mathbf{v} \\ 1 \end{bmatrix}' \right\} = \begin{bmatrix} \mathbf{P}_0 & 0 & 0 & 0 \\ 0 & \mathbf{Q}\delta_{ij} & 0 & 0 \\ 0 & 0 & \mathbf{R}\delta_{ij} & 0 \end{bmatrix} \tag{3}$$

where $(.)'$ and $\mathcal{E}\{.\}$ denote transposition and expectation, respectively. The firing rate of each unit is centered at zero, i.e., $\mathcal{E}\{\mathbf{y}_k\} = 0$.

B. Kalman Filter Recursions

The time update equations provide the *a priori* state estimate and its covariance at the *k*th ($k = 1, \ldots, N$) recursion

$$\widehat{\mathbf{x}_k^-} = \Phi \widehat{\mathbf{x}}_{k-1} \tag{4}$$

$$\mathbf{P}_k^- = \Phi \mathbf{P}_{k-1} \Phi' + \mathbf{Q} \tag{5}$$

where $\mathbf{P}_k^- = \mathcal{E}\left\{ \mathbf{e}_k^- {\mathbf{e}_k^-}' \right\}$ with $\mathbf{e}_k^- = \mathbf{x}_k - \widehat{\mathbf{x}_k^-}$, and (5) is the discrete-time algebraic Lyapunov equation. If we define

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}' \left(\mathbf{H} \mathbf{P}_k^- \mathbf{H}' + \mathbf{R} \right)^{-1} = \mathbf{P}_k \mathbf{H}' \mathbf{R}^{-1} \tag{6}$$

as the $s \times n$ filter gain matrix, the *a posteriori* state estimate and its covariance are given by the measurement update equations

$$\widehat{\mathbf{x}_k} = \widehat{\mathbf{x}_k^-} + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H} \widehat{\mathbf{x}_k^-} \right) \tag{7}$$

$$\mathbf{P}_k = \left(\mathbf{I}_s - \mathbf{K}_k \mathbf{H} \right) \mathbf{P}_k^- \tag{8}$$

where $\mathbf{P}_k = \mathcal{E}\left\{ \mathbf{e}_k \mathbf{e}_k' \right\}$, $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$, and $\mathbf{I}_s$ is the $s \times s$ identity matrix.

C. Filter Training

The state and observation matrices, Φ and $\mathbf{H}$, are estimated from training data using the ordinary least squares procedure

$$\Phi = \mathbf{X}_2 \mathbf{X}_1' \left(\mathbf{X}_1 \mathbf{X}_1' \right)^{-1} \tag{9}$$

$$\mathbf{H} = \mathbf{Y} \mathbf{X}' \left(\mathbf{X} \mathbf{X}' \right)^{-1} \tag{10}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$, $\mathbf{X}_1 = [\mathbf{x}_1, \ldots, \mathbf{x}_{M-1}]$, $\mathbf{X}_2 = [\mathbf{x}_2, \ldots, \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M]$, and M is the length of the training data. Then the minimum mean-square error estimates of the time-invariant noise covariance matrices are [12]

$$\mathbf{Q} = \frac{1}{M-1} (\mathbf{X}_2 - \mathbf{\Phi X}_1)(\mathbf{X}_2 - \mathbf{\Phi X}_1)' \tag{11}$$

$$\mathbf{R} = \frac{1}{M} (\mathbf{Y} - \mathbf{HX})(\mathbf{Y} - \mathbf{HX})'. \tag{12}$$

D. Kalman Filter Gain Convergence

From KF theory, if the eigenvalues of $\mathbf{\Phi}$ lie inside the unit circle, then the eigenvalues of $\mathbf{\Phi}(\mathbf{I}_s - \mathbf{KH})$ also lie within the unit circle and the linear dynamic system is said to be controllable and observable [16], [17]. Then, $\mathbf{P}_k \to \mathbf{P}$ as $k \to \infty$ with a geometric order of convergence, where $\mathbf{P}$ is a unique constant symmetric matrix. Due to (6), the optimal gain, $\mathbf{K}_k$, also converges exponentially to the steady-state gain, $\mathbf{K} = \lim_{k\to\infty}\{\mathbf{K}_k\}$, with rate r. At the kth filter recursion, the distance between the KF and SSKF gains is bounded as [16]

$$\Delta K_k = \text{tr}\left\{(\mathbf{K}_k - \mathbf{K})(\mathbf{K}_k - \mathbf{K})'\right\} = \sum_{i=1}^{s} \lambda_{i,k} \le cr^k \tag{13}$$

where $\lambda_{i,k}$ are the eigenvalues of $(\mathbf{K}_k - \mathbf{K})(\mathbf{K}_k - \mathbf{K})'$, $0 < r < 1$ and $c > 0$ are constants, and tr $\{.\}$ denotes matrix trace. Let $\hat{\mathbf{x}}_k$ and $\tilde{\mathbf{x}}_k$ denote the state estimates obtained with the KF and SSKF, respectively. Then, from (13), $\tilde{\mathbf{x}}_k$ is an asymptotially optimal estimate of $\hat{\mathbf{x}}_k$ with exponential convergence, and the Euclidean distance, Δx_k, between the two estimates is bounded as

$$(\Delta x_k)^2 = \|\hat{\mathbf{x}}_k - \tilde{\mathbf{x}}_k\|^2 \le cr^k \tag{14}$$

where $\|.\|$ denotes the ℓ^2−norm of a vector. From (13), we obtain the upper bounds on the constants c and r as

$$\widehat{c}_{\max} = \Delta K_0 = \text{tr}\left\{\mathbf{KK}'\right\} \tag{15}$$

$$\widehat{r}_{\max} = \exp\left[\frac{1}{k}\log\left(\frac{\Delta K_k}{\Delta K_0}\right)\right]. \tag{16}$$

E. Steady-State Gain Computation

We note from (6) that the expression for the KF gain, $\mathbf{K}_k$, does not involve the observed firing rates, $\mathbf{y}_k$, and depends only on the constant system matrices $\mathbf{\Phi}$, $\mathbf{H}$, $\mathbf{Q}$, and $\mathbf{R}$, and the initial state covariance matrix $\mathbf{P}_0$. It follows that $\mathbf{K}_k$ can be computed recursively *a priori* during the filter training phase and stored for use later during decoding in order to reduce runtime complexity. In the steady-state approach, $\mathbf{K}$ can be obtained from these recursions as $\mathbf{K} = \lim_{k-\infty} \mathbf{K}_k$. In this paper, however, we compute $\mathbf{K}$ with an efficient nonrecursive

procedure based on the solution of the discrete algebraic Riccati equation (DARE) using the MacFarlane–Potter–Fath eigenstructure method [15]. The steady-state value, $\mathbf{P}$, of the *a priori* estimate of the posterior error covariance, $\mathbf{P}_k$, is the solution to the steady-state DARE, given by

$$\mathbf{P} = \Phi \left\{ \mathbf{P} - \mathbf{P}\mathbf{H}' \left(\mathbf{H}\mathbf{P}\mathbf{H}' + \mathbf{R} \right)^{-1} \mathbf{H}\mathbf{P} \right\} \Phi' + \mathbf{Q}. \tag{17}$$

A stabilizing solution to (17) can be found in closed form by constructing a basis for the stable invariant subspace of the $2s \times 2s$ Hamiltonian matrix

$$\mathbf{S} = \begin{bmatrix} \left(\Phi^{-1} \right)' & \left(\Phi^{-1} \right)' \mathbf{H}' \mathbf{R}^{-1} \mathbf{H} \\ \mathbf{Q}\left(\Phi^{-1} \right)' & \Phi + \mathbf{Q}\left(\Phi^{-1} \right)' \mathbf{H}' \mathbf{R}^{-1} \mathbf{H} \end{bmatrix} \tag{18}$$

which is a symplectic matrix. Let represent $\{\lambda_i; i = 1, \ldots, 2s\}$ represent the eigenvalues of $\mathbf{S}$ such that $\det\{\mathbf{S}\} = 1$. Each $1/\lambda_i$ is also an eigenvalue of $\mathbf{S}$ with the same multiplicity as λ_i. Assuming $\lambda_i \neq 1$, there are s eigenvalues inside the unit circle and s outside. We can express $\mathbf{S}$ in its Jordan form as

$$\mathbf{S} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} = \mathbf{V} \begin{bmatrix} \Lambda^{-1} & 0 \\ 0 & \Lambda \end{bmatrix} \mathbf{V}^{-1} \tag{19}$$

where $\mathbf{D} = \text{diag}\{\lambda_i\}$ is a diagonal matrix containing the eigenvalues of $\mathbf{S}$, the ith column of $2s \times 2s$ matrix $\mathbf{V}$ contains the eigenvector corresponding to λ_i, and $\Lambda = \text{diag}\{\lambda_j\}$ such that $\lambda_j > 1$. We can partition $\mathbf{V}$ conformably into four $s \times s$ blocks, $\mathbf{V}_{ab}$, i.e.,

$$\mathbf{V} = \begin{bmatrix} \tilde{\mathbf{V}}_{11} & \tilde{\mathbf{V}}_{12} \\ \tilde{\mathbf{V}}_{21} & \tilde{\mathbf{V}}_{22} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{V}}_1 & \tilde{\mathbf{V}}_2 \end{bmatrix} \tag{20}$$

where $\tilde{\mathbf{V}}_1$ and $\tilde{\mathbf{V}}_2$ contain the eigenvectors corresponding to the stable and unstable eigenvalues of $\mathbf{S}$, respectively. As $\tilde{\mathbf{V}}_{12}$ is nonsingular, the unique and stable solution to (17) is obtained as the matrix fraction $\mathbf{P} = \tilde{\mathbf{V}}_{22}\tilde{\mathbf{V}}_{12}^{-1}$. Then from (6),

$$\mathbf{K} = \mathbf{P}\mathbf{H}' \left(\mathbf{H}\mathbf{P}\mathbf{H}' + \mathbf{R} \right)^{-1}. \tag{21}$$

The constant gain, $\mathbf{K}$, can be used to calculate the SSKF-based *a posteriori* state estimates $\tilde{\mathbf{x}}_k$ from (7) without recomputing the gain and covariance matrices at each recursion.

Note that if Φ is singular or ill-conditioned, or $\mathbf{S}$ has multiple or nearly multiple eigenvalues, then other methods should be used to obtain the solution to the DARE. For example, the gain can be obtained from (17) via an iterative Newton–Raphson procedure (which may be prone to error accumulation) or a nonrecursive inverse-free generalized Schur method (which is a deflating subspace technique with good numerical integrity) [17], [38]. However, the need to use these alternative approaches does not usually arise if the structure of the dynamical system is modeled correctly.

IV. Experimental Results

We evaluated the decoding accuracy and complexity of the SSKF algorithm and compared them to those of the standard KF in [12] using data from our clinical trial participants.

A. Convergence of the Kalman Gain

We first analyzed the temporal evolution of the KF gain in order to investigate its convergence properties. We observed that the difference between the KF and SSKF gains, ΔK_k, vanishes rapidly in all trials for all of our session data (Fig. 2). Throughout this analysis, we found the system matrices to be well-conditioned and consequently the eigenstructure method to be robust for SSKF gain computation. For our data, $\mathbf{K}_k$ converged to within 95% of $\mathbf{K}$ in 0.7–2.1 s (see Table III) and to within 99% in 3.3 s. We also confirmed that the convergence of ΔK_k was exponential in k. From (16), the regression line of $\log \Delta K_k$ on k gives us the convergence rate upper bound, $\hat{r}_{\max}$. The values of $\hat{r}_{\max}$ thus obtained ranged between 0.6 and 0.9, as listed in Table III. We note that even the slowest observed convergence rate of 0.9 translates into a 95% convergence time of only about 2 s. Since in a decoding setting the filter gain is not reset during a session, the initial transient duration of 2 s is insignificant in comparison to that of a typical decoding session that typically lasts several minutes (such as Sessions 4 and 6).

B. State Estimation Accuracy

We examined the effect of the steady-state gain approximation on the accuracy of state estimation. We decoded neuronal firing rate data with both KF and SSKF, and evaluated the Euclidean distance, Δx_k, between the velocity estimates provided by the two filters at each time-step. From (14), we expect Δx_k to decay asymptotically to zero with rate $\sqrt{r}$ which is slower than the convergence rate, r, of ΔK_k. Our results in Fig. 3 confirm the expected exponential decay, with Δx_k falling below 0.5°/s within 5 s under both open-loop and closed-loop conditions. Note that $\Delta x_0 = 0$ for all sessions since we initialize the filters with $\hat{\mathbf{x}}_0 = \tilde{\mathbf{x}}_0 = \mathbf{x}_0 = \mathbf{0}$. Averaged over the first 7 s, $\varepsilon \{\Delta x_k\} < 2°/\text{s}$, and is typically 0°/s afterwards.

For Sessions 1 and 2, Δx_k has a sharp peak reaching 3.8°/s (Session 2) at 0.5 s, followed by rapid decay, while for Sessions 3–6, Δx_k generally does not exceed 1°/s. The data in Sessions 1 and 2 consisted of multiple target-acquisition epochs with different durations, but we divided these data into 10 s trials for our offline decoding analysis. We initialized the decoder with $\hat{\mathbf{x}}_0 = \mathbf{0}$ in accordance with our general scheme, instead of the actual nonzero velocity. The KF and SSKF treat this initialization error differently due to large difference in their gains at small k, resulting in considerable different state estimates during the first few recursions. Even though the incorrect initialization renders Δx_k relatively large initially, the distance nevertheless decays as rapidly as in the other sessions. Thus the SSKF approach is robust to inaccurate state initialization and leads to the same estimates as the KF after a short period irrespective of the initial state estimates provided.

The comparison of KF and SSKF state estimates, obtained offline for the closed-loop data from Session 6, confirms the convergence of the state component estimates within 5 s (Fig. 4). We also note that the KF posterior error covariance estimates converge to the SSKF covariance estimate within the same duration. Our results show that once the estimates have converged, they remain identical for the remaining 10 min decoding session. As a result, the states decoded by the KF and SSKF in a session have an average correlation coefficient of 0.99 (Table III). This also can be observed in terms of the corresponding cursor trajectory reconstruction over 2-D space, obtained from the state (velocity) estimates by numerical integration, shown in Fig. 5 for the entire 10 min duration of the closed-loop center-out task in Session 6. The filters perform identically except for a minor over-excursion by the SSKF

in reaching the very first target at the bottom-center of the screen. On the basis of these observations, we conclude that the SSKF state estimation accuracy is essentially identical to that of the KF.

C. Gain in Computational Efficiency

Having verified the decoding accuracy of the SSKF, we next evaluated the advantage in computational efficiency achieved with the SSKF. To extend the above theoretical discussion on algorithmic complexity, we conducted runtime performance tests based on the time taken for actual program execution. Our computational platform for this purpose consisted of a laptop computer with an Intel Core 2 Duo CPU with T7700 chipset and 2.4-GHz clock running Matlab 2010a on 32-bit Windows 7 operating system. The exact runtimes will depend on the computing hardware and software, but we expect the runtime reduction factor to remain approximately constant across platforms. For each of the sessions, we performed the decoding 25 times to obtain a reliable estimate of the execution time.

We found that the average execution time per recursion to decode the neural data in Session 6 is 0.13 ms for the KF and 0.015 ms for the SSKF; the former is almost an order of magnitude larger. From Fig. 6, we observe that the time to decode each session's neural data with the KF is several times larger than that of the SSKF. For instance, the KF and SSKF mean decoding times for Session 6 are 0.84 and 0.10 s, respectively. Comparing the times in Fig. 6 against session details in Table II, we find that the execution time is proportional to the length of the decoding session. The reason is that the complexity for K recursions of the decoder is K times the complexity for a single recursion. This is the basis of the execution time difference between Sessions 3 and 4, as these two sessions differ in the decoded data length and thus the number of filter recursions.

To assess the reduction in complexity per recursion, we define the runtime efficiency factor as the ratio of the KF and SSKF execution times to decode the session data. This quantity is independent of the number of filter recursions in a decoding session, since the execution time is linearly related to the number of recursions which is identical for both filters. The efficiency factors, listed in Table III, range from 5.8 to 8.3 for our data. We see that the efficiency due to SSKF has a clear relation with the number of units selected for decoding, and is higher in sessions with more units. This empirical observation confirms the earlier theoretical discussion (Table I) according to which the SSKF efficiency improvement increases as the square of the observation vector dimension, n. Therefore we can expect that when neural data from just 50 units is decoded, the relative efficiency of the SSKF will approach an order of magnitude, and will be even larger for larger ensembles.

On the basis of the above discussion, if we use 50 ms time-bins to calculate the firing rates for our data, the number of filter recursions will increase, and so will the execution times of both filters. The difference in the KF and SSKF execution times would then be even larger. To confirm the effect of higher sample rate signals, we conducted the execution time analysis with Session 6 data using 50 ms time-bins. In this case, the decoding times for this session with KF and SSKF were found to be 1.52 and 0.19 s, respectively, and the efficiency factor was 8.2. Although the total execution times were longer than those obtained with 100 ms bins due to a greater number of recursions, the efficiency factor was consistent.

V. Discussion

Our analysis of the motor cortical activity decoded from two pilot clinical trial participants with tetraplegia demonstrates that the SSKF could be used for efficient decoding in neural interface systems. This finding has implications not just for our current relatively small data sets but also for larger and more complex data obtained with neural interface systems.

A. Implications for Large-Dimensional Systems

The gain in computational efficiency from using SSKF would be massive when the neuronal ensemble used for decoding is large, as may be considered for some future clinical NISs (see Table I). It is not uncommon for an intracortical microelectrode array to record neural data simultaneously from a population of over 100 units. In that case, with larger n, as might also occur with more than one array, we would expect the SSKF to exhibit increased relative efficiency. For a dual-state Kalman filter and a 100-neuron population, state estimation at each recursion will involve over 10^6 floating point operations with the KF, but only 204 operations with the SSKF.

B. Decoding Complex Signal Sets

The execution time for decoding neural data from a session is directly related to the session length, due to which the SSKF becomes increasingly advantageous as more time-samples are processed. This has important implications for decoding with neural signals besides a firing rate signal with 10 Hz sampling. For the same duration, a neural signal with a higher sampling rate will involve a larger number of recursions, and so the relative computational cost of the SSKF will be correspondingly lower. One example of such a neural signal is multiunit activity (MUA) sampled at 500 Hz [30]. Decoding the MUA signal for a given length of time will therefore involve 50 times as many recursions as the firing rate signal sampled at 10 Hz, and the comparative reduction in execution time with the SSKF will be more pronounced.

A fundamental requirement for real-time decoding is that the computations involved in one decoder recursion must complete before the next data sample becomes available for processing. Thus for the MUA signal sampled at 500 Hz, we are constrained to complete a single recursion within 2 ms. Decoding such high-rate signals becomes particularly challenging for a large-dimensional system with, say, 100 MUA channels. In such scenarios, the SSKF is especially useful in enabling real-time implementation.

C. Using Precomputed Optimal Kalman Gain

One possibility for avoiding real-time KF gain computation in each recursion is to precompute and store the time-series of adaptive KF gain matrices, $\mathbf{K}_k$, $k = 1, \ldots, N$, before decoding commences, as discussed in Section III-E. For a NIS decoding 9 kinematic states (3-D position, velocity, and acceleration) from 100 MUA channels, the N time-samples of the $s \times n$ Kalman gain matrix for one 24-h decoding session on a 64-bit computer would consume 300 GB of storage space. Storing, transmitting and processing such large volumes of data would challenge the capabilities of current hardware, software and communications architectures used in practical NIS computing platforms. This simple example demonstrates the impracticability of using precomputed adaptive Kalman gain matrices for high-rate neural signals and confirms that the SSKF approach is critical for both online and offline neural decoding in such a setting.

D. Effect of Signal Non-Stationarity

The training paradigms of KF and SSKF are identical, in which the open-loop training data are used to compute the constant matrices $\mathbf{\Phi}, \mathbf{H}, \mathbf{Q}$, and $\mathbf{R}$ that are then used for closed-loop decoding. This training approach is based on the assumption of neural signal stationarity within a session. It is worth emphasizing that the SSKF does not impose any stationarity conditions beyond those assumed by the KF. Considerable temporal variability in motor cortical neural signals has, however, been reported [39], [40]. This variability may arise from routine anatomical and physiological processes in the brain as well as electromechanical imperfections in the interface system. The impact of this nonstationarity

on decoding accuracy has been addressed and strategies to cope with it have been proposed [10], [26]. Most of these approaches assume pseudo-stationarity over short periods and use a block estimation paradigm with periodic filter updates. Within that period, the state and observation matrices and noise covariances are assumed to remain constant, as is the case with the standard KF [12] and SSKF, so the underlying state space model and filter structure are essentially the same. The SSKF can be easily incorporated into the block-estimation adaptive framework, since the calculation of the steady-state Kalman gain (see Section III-E) is computationally inexpensive and can be repeated along with filter training as often as needed.

E. Performance Under Feedback

In our offline analysis, we have found that the SSKF estimates differ from the KF estimates during the initial recursions, which leads to a slight error in cursor trajectory estimation during that period (Fig. 5). In an online closed-loop task with visual feedback, however, the participant is able to compensate for such effects by altering the neural activity accordingly. Thus, if the SSKF is used in closed-loop, we expect that the participant would compensate for the initial difference and prevent the overshoot observed in Fig. 5. Therefore, we expect that in a practical decoding setting with closed-loop control, the SSKF and KF accuracy levels would have no perceptible difference.

VI. Conclusion

Our analysis establishes the utility of the steady-state Kalman filter for neural decoding. The steady-state Kalman filter significantly increases the computational efficiency for even relatively simple neural spiking data sets from a human NIS. From our analysis, the SSKF converges to within 95% of the KF in about 2 s. Once convergence is attained, the estimates from the two filters are identical. The decoding complexity is reduced dramatically by the SSKF, resulting in approximately seven-fold reduction in the execution time for decoding a typical neuronal firing rate signal. This improved efficiency is important for online neuroprosthetic control applications and offline performance analyses. We anticipate that the accurate and low-complexity decoding performance obtained with SSKF will make this approach useful for practical implementation of future neural interface systems, including fully embedded systems exploiting complex signal sets, where computational efficiency will be particularly beneficial.

Acknowledgments

Biography

Wasim Q. Malik (S'97–M'00–SM'08) received the Ph.D. degree in electrical engineering from Oxford University, Oxford, U.K.

He is presently Instructor in Anesthesia at Harvard Medical School and Massachusetts General Hospital. He holds visiting academic appointments at Massachusetts Institute of Technology and Brown University, and is also affiliated with the United States Department of Veterans Affairs, Providence, RI. His research focuses on statistical signal processing for brain-machine interfaces, two-photon neuroimaging and wideband wireless communications.

Dr. Malik is a recipient of the ACM Recognition of Service Award 2000, Best Paper Award in the ARMMS RF & Microwave Conference 2006, ESU Lindemann Fellowship 2007, and CIMIT Career Development Award 2011.

Wilson Truccolo received the Ph.D. degree in complex systems from Florida Atlantic University, Boca Raton, FL, and postdoctoral training from the Department for Neuroscience, Brown University, Providence, RI.

He is currently an Assistant Professor (Research) at the Department of Neuroscience, Brown University, where he is also with the Brown Institute for Brain Science, and a Research Fellow at the Department of Neurology, Massachusetts General Hospital, Boston. His research interests include stochastic models of cortical networks dynamics and computation, statistical analysis of neural systems, and brain–machine interfaces.

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

He is presently Professor of Computational Neuroscience and Health Sciences and Technology at the Massachusetts Institute of Technology (MIT) and the Warren M. Zapol Professor of Anaesthesia at Harvard Medical School and Massachusetts General Hospital. His research interests are in the study of mechanisms of general anesthesia in humans and in the use point process and state-space methods to develop algorithms for neural signal processing.

Dr. Brown is a fellow of the American Statistical Association, a fellow of the American Association for the Advancement of Science, a member of Institute of Medicine of the National Academies and a recipient of a 2007 NIH Director's Pioneer Award.

Leigh R. Hochberg received the M.D. and Ph.D. degrees in neuroscience from Emory University, Atlanta, GA.

He is Associate Professor of Engineering, Brown University; Investigator at the Center for Restorative and Regenerative Medicine, Rehabilitation R&D Service, Providence VA Medical Center; and Visiting Associate Professor of Neurology at Harvard Medical School. In addition, he maintains clinical duties on the Stroke and Neurocritical Care Services at the MGH and Brigham & Women's Hospital, and is on the consultation neurology staff at Spaulding Rehabilitation Hospital. He directs the Laboratory for Restorative Neurotechnology at Brown and MGH, and his research focuses on the development and testing of implanted neural interfaces to help people with paralysis and other neurologic disorders.

References

1. Dornheg, G.; Millan, J. d. R.; Hinterberger, T.; McFarland, DJ.; Muller, K. Toward Brain-Computer Interfacing. MIT Press; Cambridge, MA: 2007.

2. Principe, J.; Sanchez, JC.; Enderle, J. Brain-Machine Interface Engineering. Morgan & Claypool; San Rafael, CA: 2007.

3. Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR, Donoghue JP. Instant neural control of a movement signal. Nature Mar.;2002 416(6877):141–142. [PubMed: 11894084]

4. Mussa-Ivaldi FA, Miller LE. Brain-machine interfaces: Computational demands and clinical needs meet basic neuroscience. Trends Neurosci Jun.;2003 26(6):329–334. [PubMed: 12798603]

5. Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, Dimitrov DF, Patil PG, Henriquez CS, Nicolelis MAL. Learning to control a brain-machine interface for reaching and grasping by primates. PLoS Biol Nov.;2003 1(2):193–208.

6. Hochberg LR, Serruya MD, Friehs GM, Mukand J, Saleh M, Caplan AH, Branner A, Chen D, Penn RD, Donoghue JP. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature Jul.;2006 442(7099):164–171. [PubMed: 16838014]

7. Santhanam G, Ryu SI, Yu BM, Afshar A, Shenoy KV. A high-performance brain-computer interface. Nature Jul.;2006 442(7099):195–198. [PubMed: 16838020]

8. Aggarwal V, Acharya S, Tenore F, Etienne-Cummings R, Schieber MH, Thakor NV. Asynchronous decoding of dexterous finger movements using M1 neurons. IEEE Trans Neural Syst. Rehabil. Eng Feb.;2008 16(1):3–14. [PubMed: 18303800]

9. Velliste M, Perel S, Spalding MC, Whitford AS, Schwartz AB. Cortical control of a prosthetic arm for self-feeding. Nature Jun.;2008 453(7198):1098–1101. [PubMed: 18509337]

10. Wu W, Hatsopoulos NG. Real-time decoding of nonstationary neural activity in motor cortex. IEEE Trans. Neural Syst. Rehabil. Eng Jun.;2008 16(3):213–222. [PubMed: 18586600]

11. Mulliken GH, Musallam S, Andersen RA. Decoding trajectories from posterior parietal cortex ensembles. J. Neurosci Nov.;2008 28(48):12913–12926. [PubMed: 19036985]

12. Wu W, Gao Y, Bienenstock E, Donoghue JP, Black MJ. Bayesian population decoding of motor cortical activity using a Kalman filter. Neural Comput Jan.;2006 18(1):80–118. [PubMed: 16354382]

13. Kim S-P, Simeral JD, Hochberg LR, Donoghue JP, Black MJ. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. J. Neural. Eng Dec.;2008 5(4):455–476. [PubMed: 19015583]

14. Wu, W.; Shaikhouni, A.; Donoghue, JP.; Black, MJ. Closed-loop neural control of cursor motion using a Kalman filter. Proc. IEEE Eng. Med. Biol. Conf.; San Francisco, CA. Sep.. 2004 p. 4126-4129.

15. Simon, D. Optimal State Estimation. Wiley; New York: 2006.

16. Chui, CK.; Chen, G. Kalman Filtering. 4th ed.. Springer; New York: 2009.

17. Mendel, JM. Lessons in Digital Estimation Theory. Prentice-Hall; Upper Saddle River, NJ: 1986.

18. Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SIR, Meng TH, Sahani M, Shenoy KV. Mixture of trajectory models for neural decoding of goal-directed movements. J. Neurophysiol Feb.;2007 97(5):3763–3780. [PubMed: 17329627]

19. Yu, BM.; Cunningham, JP.; Shenoy, KV.; Sahani, M. Neural Information Processing. Springer; New York: 2008. Neural decoding of movements: From linear to nonlinear trajectory models.

20. Brockwell AE, Rojas L, Kass RE. Recursive Bayesian decoding of motor cortical signals by particle filtering. J. Neurophysiol Apr.;2004 91(4):1899–1907. [PubMed: 15010499]

21. Shoham S, Paninski LM, Fellows MR, Hatsopoulos NG, Donoghue JP, Normann RA. Statistical encoding model for a primary motor cortical brain-machine interface. IEEE Trans. Biomed. Eng Jul.;2005 52(7):1312–1322. [PubMed: 16041995]

22. Barbieri R, Wilson MA, Frank LM, Brown EN. An analysis of hippocampal spatio-temporal representations using a Bayesian algorithm for neural spike train decoding. IEEE Trans. Neural Syst. Rehabil. Eng Jun.;2005 13(2):131–136. [PubMed: 16003890]

23. Truccolo W, Friehs GM, Donoghue JP, Hochberg LR. Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia. J. Neurosci Jan.;2008 28(5):1163–1178. [PubMed: 18234894]

24. Wu W, Black MJ, Mumford D, Gao Y, Bienenstock E, Donoghue JP. Modeling and decoding motor cortical activity using a switching Kalman filter. IEEE Trans. Biomed. Eng Jun.;2004 51(6):933–942. [PubMed: 15188861]

25. Pistohl T, Ball T, Schluze-Bonhage A, Aersten A, Mehring C. Prediction of arm movement trajectories from ECoG-recordings in humans. J. Neurosci. Methods 2008;167(1):105–114. [PubMed: 18022247]

26. Gage GJ, Ludwig KA, Otto KJ, Ionides EL, Kipke DR. Naive coadaptive cortical control. J. Neural. Eng Jun.;2005 2(2):52–63. [PubMed: 15928412]

27. Wang, Y.; Principe, JC. Tracking the non-stationary neuron tuning by dual kalman filter for brain machine interfaces decoding. Proc. IEEE Conf. Eng. Med. Biol.; Vancouver, BC, Canada. Aug.. 2008 p. 1720

28. Wu W, Kulkarni JE, Hatsopoulos NG, Paninski L. Neural decoding of hand motion using a linear state-space model with hidden states. IEEE Trans. Neural Syst. Rehab. Eng Aug.;2009 17(4):370–378.

29. Li Z, O'Doherty JE, Hanson TL, Lebedev MA, Henriquez CS, Nicolelis MAL. Unscented Kalman filter for brain-machine interfaces. PLoS ONE Jul.;2009 4(7):E6243. [PubMed: 19603074]

30. Stark E, Abeles M. Predicting movement from multiunit activity. J. Neurosci Aug.;2007 27(31):8387–8394. [PubMed: 17670985]

31. Humphrey DR. Systems, methods and devices for controlling external devices by signals derived directly from the nervous system. U.S. 6,171,239. 2001

32. Hwang EJ, Andersen RA. Brain control of movement execution onset using local field potentials in posterior parietal cortex. J. Neurosci Nov.;2009 29(45):14363–14370. [PubMed: 19906983]

33. Nurmikko AV, Donoghue JP, Hochberg LR, Patterson WR, Song Y-K, Bull CW, Borton DA, Laiwalla F, Park S, Ming Y, Aceros J. Listening to brain microcircuits for interfacing with external world—progress in wireless implantable microelectronic neuroengineering devices. Proc. IEEE Mar.;2010 98:375–388.

34. Miranda H, Gilja V, Chestek C, Shenoy KV, Meng TH. Hermesd: A high-rate long-range wireless transmission system for simultaneous multichannel neural recording applications. IEEE Trans. Biomed. Circuits Syst. 2010

35. Mendel JM. Computational requirements for a discrete Kalman filter. IEEE Trans Autom. Control Dec.;1970 16(6):748–758.

36. Yousry TA, Schmid UD, Alkadhi H, Schmidt D, Peraud A, Buettner A, Winkler P. Localization of the motor hand area to a knob on the precentral gyrus—A new landmark. Brain Jan.;1997 120(1): 141–157. [PubMed: 9055804]

37. Suner S, Fellows MR, Vargas-Irwin C, Nakata GK, Donoghue JP. Reliability of signals from chronically implanted, silicon-based electrode array in non-human primate primary motor cortex. IEEE Trans. Neural Syst. Rehabil. Eng Dec.;2005 13(4):524–541. [PubMed: 16425835]

38. Datta, BN. Numerical Methods for Linear Control Systems. Elsevier; New York: 2004.

39. Kim, S-P.; Wood, F.; Fellows, M.; Donoghue, JP.; Black, MJ. Statistical analysis of the non-stationarity of neural population codes. Proc. IEEE Conf. Biomed. Robot. Biomech.; Feb.. 2006 p. 811-816.

40. Carmena JM, Lebedev MA, Henriquez CS, Nicolelis MAL. Stable ensemble performance with single-neuron variability during reaching movements in primates. J. Neurosci Nov.;2005 25(46): 10712–10716. [PubMed: 16291944]
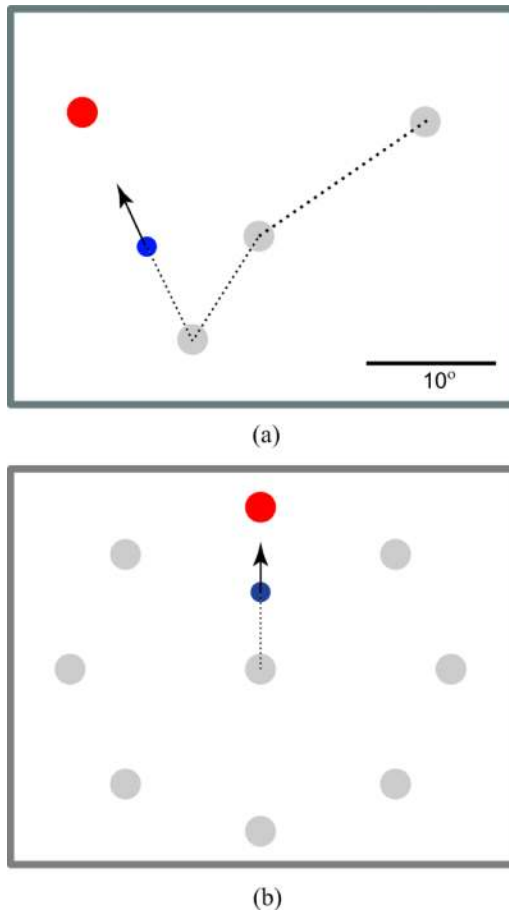
Fig. 1.
Behavioral tasks. (a) In the pursuit-tracking task, the training cursor (blue circle) acquires the selected target (red circle) placed pseudorandomly on the screen. Previous targets (gray circles) vanish from the screen upon acquisition. The nominal cursor trajectory is also shown (dotted line) for clarity. (b) In the center-out task, there are four (during training) or eight (during assessment) radial targets and one center target in this center-out task (gray circles), one of which is selected for acquisition (red circle). The velocity of the cursor (blue circle) is computer-controlled in the open-loop filter-building phase and decoded from the participant's neural activity in the closed-loop assessment phase. (a) Pursuit-tracking. (b) Center-out.
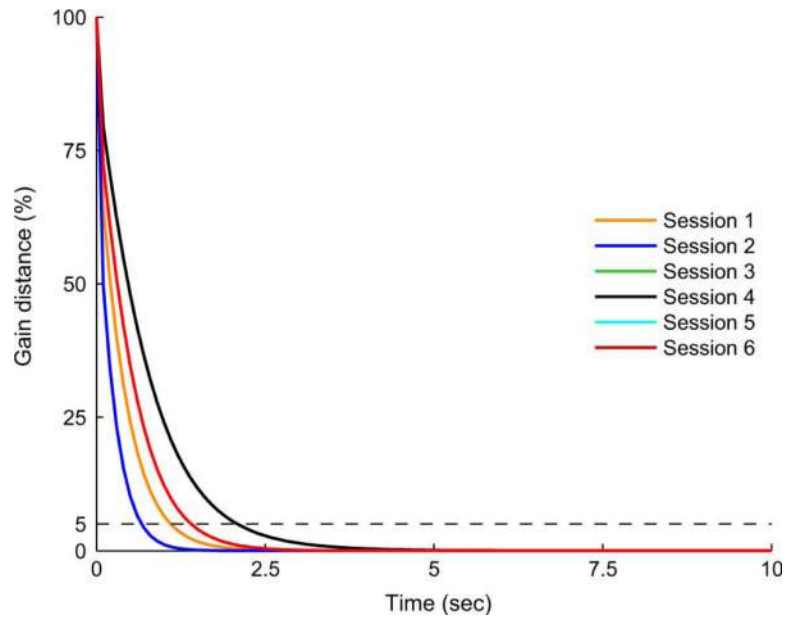
Fig. 2.
Filter convergence. Distance, ΔK_k, between the Kalman and steady-state Kalman filter gains, normalized to ΔK_0. The ΔK_k curves for Sessions 3 and 4 overlap since they use the same training data; the same is the case for Sessions 5 and 6. The curves for Sessions 1, 2, 3, and 5 are obtained from multiple trials and show the mean ± sem across trials, but the sem's are too small to be discernible.
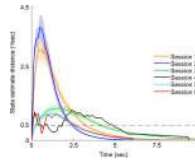
Fig. 3.
State estimate convergence. Euclidean distance, Δx_k, between the Kalman filter and steady-state Kalman filter state (velocity) estimates. Solid lines and shaded regions show the mean ± sem, respectively, across trials. The sem's are not obtained for closed-loop Sessions 4 and 6 since they consist of a single decoding trial each.
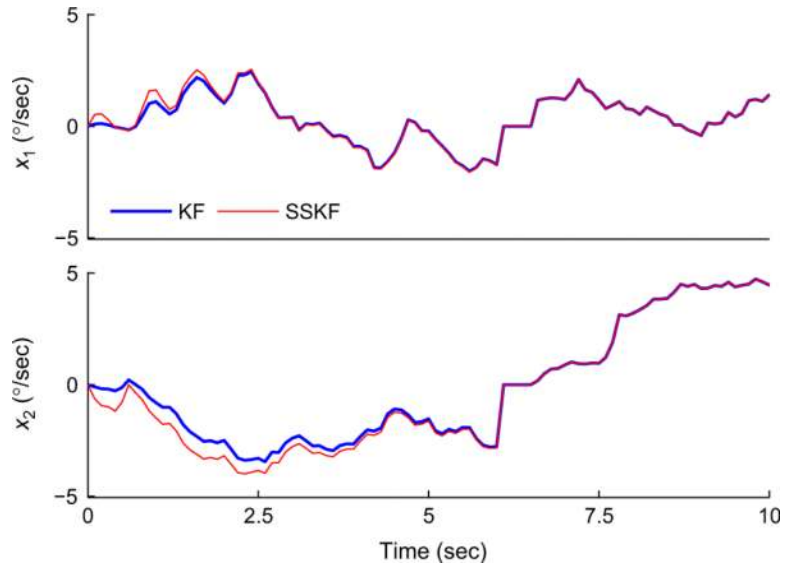
Fig. 4.
A posteriori estimates of the system states (neural cursor velocity) obtained with KF and SSKF for Session 6 (closed loop). State vector components, horizontal ($x_{1,k}$) and vertical ($x_{2,k}$) velocity, are shown in units of visual angle per unit time for the first 10 s out of the 10 min session duration. For each dimension, the KF and SSKF estimates of both the mean state estimates and their 95% confidence intervals (not shown) converge within 5 s.
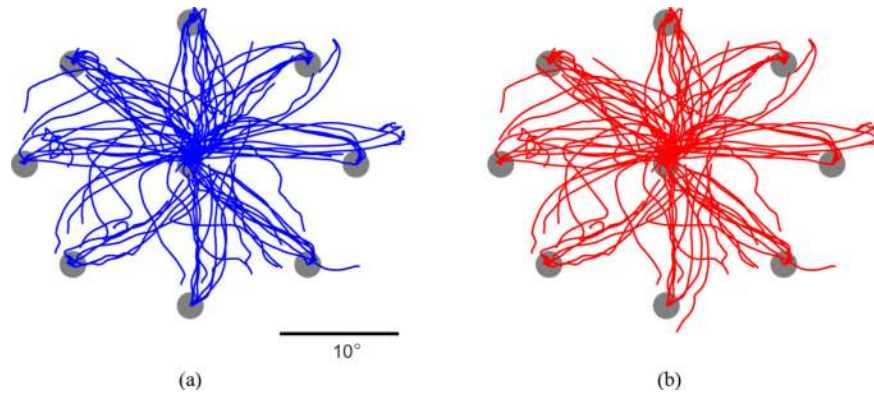
Fig. 5.
Reconstructed 2-D neural cursor trajectory estimation for Session 6 (closed loop) obtained from the decoded velocities for the entire 10 min duration. The radial targets in this center-out task are shown as gray circles. The only salient difference among the trajectories is an overshoot in (b) toward the bottom target when using the SSKF, which reflects the fact that the first excursion of the neural cursor was from the center to the bottom target. (a) Kalman filter. (b) Steady-state Kalman filter.
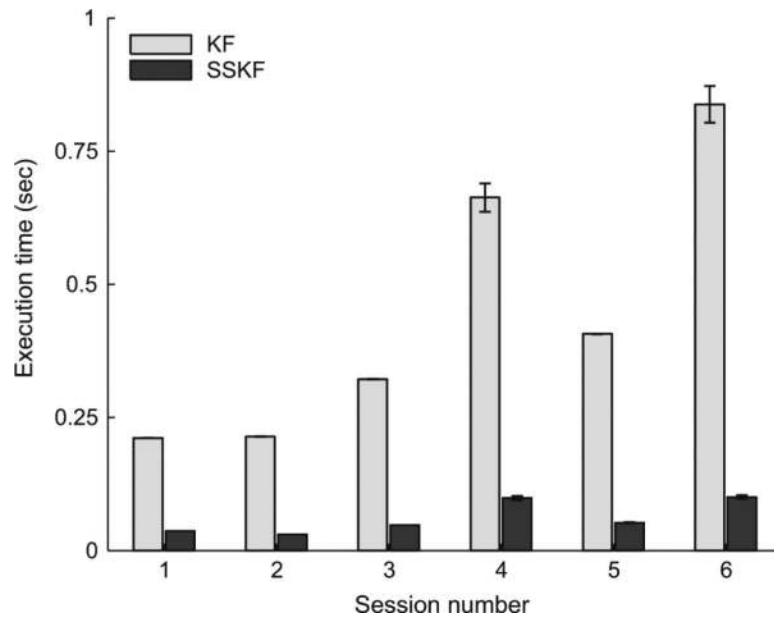
Fig. 6.
Program execution time for neural decoding using the KF and SSKF filters on a standard computing platform (mean ± sem). These execution times are for offline decoding of the entire length of a session's firing rate signals with 100 ms time-bins, e.g., 0.21 s to decode 240 s of session 1-s firing rate data with the KF, etc.

TABLE I

Theoretical Algorithmic Complexityof a Single Recursion of the Kalman Filter for Decoding *s*-Dimensional State Vector With *n*-Dimensional Observation Vector

Estimate	KF	SSKF
A priori state	$\mathrm{O}(s^2)$	$\mathrm{O}(s^2)$
A priori covariance	$\mathrm{O}(s^3)$	–
A posteriori state	$\mathrm{O}(sn)$	$\mathrm{O}(sn)$
A posteriori covariance	$\mathrm{O}(s^2 + s^2 n)$	–
Kalman filter gain	$\mathrm{O}(s^2 n + sn^2 + n^3)$	–
Full recursion	$\mathrm{O}(s^3 + s^2 n + sn^2 + n^3)$	$\mathrm{O}(s^2 + sn)$

TABLE II

Experimental Configuration

Session no.	1	2	3	4	5	6
Participant	S1	S1	S3	S3	S3	S3
Session type	Open-loop	Open-loop	Open-loop	Closed-loop	Open-loop	Closed-loop
Task type	Pursuit-tracking	Pursuit-tracking	Center-out	Center-out	Center-out	Center-out
Days since array implantation	86	90	1002	1002	1003	1003
Distance of screen from participant (m)	0.59	0.59	0.54	0.54	0.57	0.57
No. of units sorted	30	29	39	39	39	39
No. of units selected for filter building	20	25	24	24	29	29
Recording session duration (sec)	249	235	344	596	344	595
Duration of decoded activity (sec)	240	200	342	596	342	595

TABLE III

Summary of Results

Session no.	1	2	3	4	5	6
Convergence rate upper bound, $\hat{f}_{max}$	0.77	0.64	0.87	0.87	0.79	0.79
Time for convergence of $\mathbf{K}_k$ within 95% of $\mathbf{K}$ (sec)	1.1	0.7	2.1	2.1	1.4	1.4
Time for convergence of $\hat{\mathbf{x}}$ within 0.5°/sec of $\bar{\mathbf{x}}$ (sec)	3.3	2.5	4.2	4.9	2.7	2.7
Maximum absolute deviation (°/sec)	3.1	3.8	1.1	1.0	0.9	0.9
Mean absolute deviation in first 7 sec (°/sec)	1.2	2.1	0.1	0.5	0.1	0.4
State estimate correlation coefficient	0.99	0.98	0.98	0.99	0.99	0.99
Runtime efficiency factor	5.8	7.0	6.7	6.7	7.8	8.3