Inferring Hand Motion from Multi-Cell Recordings in Motor Cortex using a Kalman Filter

W. Wu[*] M. J. Black[†] Y. Gao[*] E. Bienenstock[* §] M. Serruya[§] J. P. Donoghue[§]

[*] Division of Applied Mathematics, Brown University, Providence, RI 02912
[†] Dept. of Computer Science, Brown University, Box 1910, Providence, RI 02912
[§] Dept. of Neuroscience, Brown University, Providence, RI 02912

weiwu@cfm.brown.edu, black@cs.brown.edu, gao@cfm.brown.edu, elie@dam.brown.edu,
Mijail_Serruya@brown.edu, john_donoghue@brown.edu

Abstract

This paper develops a control-theoretic approach to the problem of decoding neural activity in motor cortex. Our goal is to infer the position and velocity of a subject's hand from the neural spiking activity of 25 cells simultaneously recorded in primary motor cortex. We propose to model the encoding and decoding of the neural data using a Kalman filter. Towards that end we specify a measurement model that assumes the firing rate of a cell within 50ms is a stochastic linear function of position, velocity, and acceleration of the hand. This model is learned from training data along with a system model that encodes how the hand moves. Experimental results show that the reconstructed trajectories are superior to those obtained by linear filtering. Additionally, the Kalman filter provides insight into the neural encoding of hand motion. For example, analysis of the measurement model suggests that, while the neural firing is closely related to the position and velocity of the hand, the acceleration is redundant. Furthermore, the Kalman filter framework is exploited to recover the optimal lag time between hand movement and neural firing.

1. Introduction

This paper describes a new control-theoretic model for the encoding of hand movement in motor cortex and for inferring, or decoding, this movement from the firing rates of a population of cells. We argue that such an approach should (1) have a sound probabilistic foundation; (2) explicitly model noise in the data; (3) indicate the uncertainty in estimates of hand position; (4) make minimal assumptions about the data; (5) provide on-line estimates of hand position with short delay (less than 200ms); (6) provide insight into the neural coding of movement. The Kalman filtering method proposed here provides a rigorous and well understood framework that addresses these issues. Combined with advances in multi-electrode recording in awake, behaving, subjects, the method may be appropriate for the precise neural control of external devices (Isaacs et al., 2000, Wessberg et al., 2000, Serruya et al., 2002).

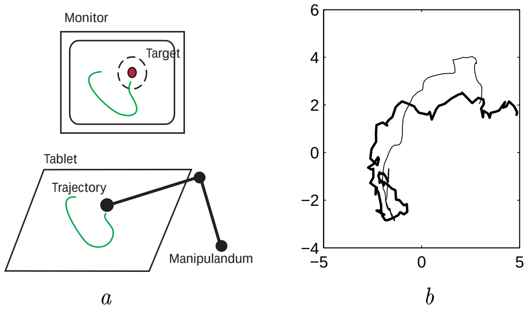

Figure 1: Reconstructing 2D hand motion. *(a)* **Training:** spiking activity is recorded while the subject tracks a target by moving a jointed manipulandum on a 2D plane. *(b)* **Decoding:** true target trajectory (thin) and reconstruction using the Kalman filter (thick).

Simultaneous recordings are acquired from an array consisting of 100 microelectrodes that is implanted in the primary motor cortex (MI) of a Macaque monkey. Using the experimental paradigm of (Paninski et al., 2001), the monkey viewed a computer monitor and gripped a two-link, negligible-friction manipulandum that was moved on a tablet parallel to the floor (Figure 1*a*). In each trial, the monkey's task was to manually follow a target that moved smoothly and randomly on the screen with visual feedback of its hand position presented on the screen. For the data analyzed here, there are 182 trials, each of which is approximately $8 - 10$ seconds long. The hand position, velocity, and acceleration were recorded every 50 ms along with the firing rate for each of 25 neurons within the previous 50ms.

Our primary goal is to reconstruct hand trajectory from the spiking activity (Figure 1*b*) with the ultimate goal of providing control of prosthetic devices for the severely disabled. We see this as a problem of *inferring* behavior from noisy measurements. Our approach develops a Kalman filter framework (Gelb, 1974) for modeling the relationship between firing rates in motor cortex and the position and velocity of the subject's hand. The method builds on previous work (Brown et al., 1998) by applying these techniques to infer smooth hand motion from motor cortical activity. In the Kalman framework, the hand movement (position, velocity and acceleration) is modeled as the system *state* and the firing rate is modeled as the *observation* (measurement). The approach specifies an explicit generative model that assumes the observation (firing rate in 50ms) is a linear function of the state (hand kinematics) plus Gaussian noise. Similarly, the hand state at time t is assumed to be a linear function of the hand state at the previous time instant plus Gaussian noise. The Kalman filter approach provides a recursive, and on-line, estimate of hand kinematics from the firing rate in non-overlapping 50ms bins.

In contrast to previous work, this probabilistic approach provides a measure of confidence in the resulting estimates. This can be extremely important when the output of the decoding method is to be used for later stages of analysis. The results of reconstructing hand trajectories from pre-recorded neural firing rates are compared with those obtained using more traditional linear filtering techniques (Serruya et al., 2002, Warland et al., 1997) using overlapping 550ms windows. The results indicate that the Kalman filter provides better results than the linear filter.

In addition to providing a probabilistic inference framework that improves on the ad hoc linear filter, the Kalman filter also provides a new tool for gaining insight into the neural coding. Training the Kalman filter involves recovering an observation matrix that linearly relates hand motions with neural firing. By analyzing this matrix we see that both position and velocity of the hand are related to neural activity while acceleration is much less important. Moreover, the framework gives us a principled way of determining the optimal "lag" between hand motion and the neural activity.

1.1 Related Work

Many mathematical algorithms have been proposed to model the encoding of hand motion by neural firing activity and to decode this activity to recover the motion information from multi-cell recordings. For example, Georgopoulos and colleagues (Georgopoulos et al., 1986) have used a center-out task in which the subject moved the hand from a central location to one of eight radially located targets. They suggested that the movement direction may be encoded by the neural ensemble in the arm area of motor cortex (MI), and the ensemble activity of the cells was combined using a *population vector* algorithm.

Based on their work, Moran and Schwartz (Moran and Schwartz, 1999) encoded both the instantaneous speed and direction using the population vector. They showed that the cell's activity is modulated with speed when the subject moves the arm in the preferred direction. Also they suggested that spiking activity precedes, or lags, the corresponding movement and this lag may vary between cells. This population vector approach has been used for the real-time neural control of 2D and 3D cursor movement. The approach appears to work well but lacks a formal mathematical foundation and provides no estimate of uncertainty. These factors make it difficult to extend this approach to more the complex analysis of temporal movement patterns.

Traditional linear filtering has also been used for decoding (Paninski et al., 2001, Warland et al., 1997) and can be used to achieve real-time neural control of a 2D cursor (Serruya et al., 2002). This approach requires the use of data over a long time window (typically 500ms to 1s). Such a long window of temporal integration may not be appropriate for faster or more complex (higher frequency) motions. Other approaches based on Artificial Neural Networks (ANN) (Warland et al., 1997) and principal component analysis (PCA) (Chapin and Nicolelis, 1999, Isaacs et al., 2000) have similar limitations.

What is needed is a probabilistically grounded method that uses data in small time windows (e.g. 50ms) and integrates that information over time in a recursive fashion. The CONDENSATION algorithm has been recently introduced as a Bayesian decoding scheme (Gao et al., 2002), which provides a probabilistic framework for causal estimation and is shown superior to the performance of linear filtering when sufficient data is available (e.g. firing rates for several hundred cells). Note that the CONDENSATION method is more general than the Kalman filter in that it does not assume linear models and Gaussian noise. While this may be important for neural decoding as suggested in (Gao et al., 2002), current recording technology makes the method impractical for real-time control.

The Kalman filter has been widely used for estimation problems ranging from target tracking to vehicle control (Brown and Hwang, 1997, Gelb, 1974, Jacobs, 1993). Here we apply this well understood theory to the problem of decoding hand kinematics from neural activity in motor cortex. This builds on the work of (Brown et al., 1998) that uses a Kalman filter to estimate the position of a rat from the firing activity of hippocampal place cells.

In Section 2., we describe the mathematical framework and algorithm for the Kalman filter and apply it to our decoding problem. Section 3. presents the experimental results and discusses them while Section 4. concludes with a discussion and summary of future work.

2. Methods

Our goal is to estimate the *state* of the hand at the current instant in time, i.e. $\mathbf{x}_k = [x, y, v_x, v_y, a_x, a_y]_k^T$ representing x-position, y-position, x-velocity, y-velocity, x-acceleration, and y-acceleration at time $t_k = k\Delta t$ where $\Delta t = 50$ms in our experiments. The Kalman filter (Gelb, 1974, Welch and Bishop, 2001) model assumes the state is linearly related to the observations $\mathbf{z}_k \in \Re^C$ which here represents a $C \times 1$ vector containing the firing rates at time t_k for C observed neurons within 50ms. In our experiments, $C = 25$ cells.

This *generative model* of neural firing is formulated as

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{q}_k, \tag{1}$$

where $k = 1, 2, \cdots$ and $\mathbf{H} \in \Re^{C \times 6}$ is a matrix that linearly relates the hand state to the neural firing. We assume the noise in the observations is zero mean and normally distributed, i.e. $\mathbf{q}_k \sim N(0, \mathbf{Q}_k), \mathbf{Q}_k \in \Re^{C \times C}$. Below we will discuss how to estimate $\mathbf{H}_k$ and the covariance matrix $\mathbf{Q}_k$ from training data.

The states are assumed to propagate in time according to the system model

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k, \tag{2}$$

where $\mathbf{A}_k \in \Re^{6 \times 6}$ is the coefficient matrix and the noise term $\mathbf{w}_k \sim N(0, \mathbf{W}_k), \mathbf{W}_k \in \Re^{6 \times 6}$. This states that the hand kinematics (position, velocity, and acceleration) at time $k + 1$ is linearly related to the state at time k. Once again we assume these estimates are normally distributed and we will learn $\mathbf{A}_k$ and $\mathbf{W}_k$ from training data.

In practice, $\mathbf{A}_k, \mathbf{H}_k, \mathbf{W}_k, \mathbf{Q}_k$ might change with time step k. However, here we make the common simplifying assumption they are constant. Thus we can estimate them from training data using least squares estimation.

2.1 Learning (System Identification)

In this subsection, we elaborate how to learn the parameters in the model equations (1) and (2).

Assume there are M time steps in the training data (containing states $\{\mathbf{x}_k\}$ and the associated firing rates $\{\mathbf{z}_k\}$, $k = 1, \ldots, M$). Let $x_{i,k}$ be ith element of $\mathbf{x}_k$ at time t_k (i.e. x-position, y-position, x-velocity, y-velocity, x-acceleration, or y-acceleration) and $z_{j,k}$ be the neural firing rate in 50ms of the jth cell at time $t_k, i = 1, \ldots, 6, j = 1, \ldots, C, k = 1, \ldots, M$.

If $\mathbf{A}_k, \mathbf{H}_k, \mathbf{W}_k, \mathbf{Q}_k$ are independent of k, we can omit the subscript and denote them as $\mathbf{A}, \mathbf{H}, \mathbf{W}, \mathbf{Q}$. We estimate coefficient matrices $\mathbf{A}$ and $\mathbf{H}$ by least squares:

$$\begin{aligned} \mathbf{A} &= \underset{\mathbf{A}}{\text{argmin}} \sum_{k=1}^{M-1} ||\mathbf{x}_{k+1} - \mathbf{A}\mathbf{x}_k||^2, \\ \mathbf{H} &= \underset{\mathbf{H}}{\text{argmin}} \sum_{k=1}^{M} ||\mathbf{z}_k - \mathbf{H}\mathbf{x}_k||^2, \end{aligned}$$

where $||\cdot||$ is the conventional $\mathbf{L}^2$ norm.

The solutions of above equations are

$$\mathbf{A} = \mathbf{X}_2 \mathbf{X}_1^T (\mathbf{X}_1 \mathbf{X}_1^T)^{-1}; \;\; \mathbf{H} = \mathbf{Z}\mathbf{X}^T (\mathbf{X}\mathbf{X}^T)^{-1},$$

where

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,M} \\ \vdots & \ddots & \vdots \\ x_{6,1} & \cdots & x_{6,M} \end{pmatrix}, \mathbf{X}_1 = \begin{pmatrix} x_{1,1} & \cdots & x_{1,M-1} \\ \vdots & \ddots & \vdots \\ x_{6,1} & \cdots & x_{6,M-1} \end{pmatrix},$$

$$\mathbf{X}_2 = \begin{pmatrix} x_{1,2} & \cdots & x_{1,M} \\ \vdots & \ddots & \vdots \\ x_{6,2} & \cdots & x_{6,M} \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} z_{1,1} & \cdots & x_{1,M} \\ \vdots & \ddots & \vdots \\ z_{C,1} & \cdots & x_{C,M} \end{pmatrix}.$$

Furthermore, using the estimated $\mathbf{A}$ and $\mathbf{H}$, we can estimate $\mathbf{W}$ and $\mathbf{Q}$ by

$$\mathbf{W} = (\mathbf{X}_2 - \mathbf{A}\mathbf{X}_1)(\mathbf{X}_2 - \mathbf{A}\mathbf{X}_1)^T / (M - 1)$$

and

$$\mathbf{Q} = (\mathbf{Z} - \mathbf{H}\mathbf{X})(\mathbf{Z} - \mathbf{H}\mathbf{X})^T / M.$$

With the estimated $\mathbf{A}, \mathbf{H}, \mathbf{W}, \mathbf{Q}$, the firing rate and the hand motion are encoded by equations (1) and (2) respectively.

2.2 Estimation (Kalman Filter Algorithm)

Given the generative encoding model defined above, we turn to the problem of decoding; that is, reconstructing hand motion from the firing rates of the cells.

For each $\mathbf{x}_k$, reconstruction using the Kalman filter algorithm has two steps:

i): (*a priori* step) predict $\mathbf{x}_k$ from the state equation (2). This estimate is denoted by $\hat{\mathbf{x}}_k^-$;

ii): (*a posteriori* step) update $\hat{\mathbf{x}}_k^-$ by using the information of the firing rate at time t_k. The updated estimate is denoted by $\hat{\mathbf{x}}_k$.

Here we follow the conventional notation (see (Welch and Bishop, 2001) for a review).

To evaluate the performance of the estimation, we define the *a priori* and *a posteriori* errors as

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \text{ and } \mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k. \tag{3}$$

Assume $\mathbf{x}_k^-, \mathbf{x}_k \in \Re^6$ are unbiased estimates, then the error can be characterized by the covariance matrix (in the one dimensional case, the covariance matrix is just the square of Euclidean distance between the real and estimated values). We define the *a priori* and *a posteriori* estimate error covariance matrices by

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}],\ \mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T], \tag{4}$$

respectively.

The *a posteriori* estimator is our final estimation for the state. The accuracy of it can be evaluated under MSE (mean-square error), which is, here, the trace of matrix $\mathbf{P}_k$ for each k. To simplify the estimation process, we assume the estimators are linear. Thus, we can denote the *a posteriori* state estimate as a linear combination of an *a priori* estimate and a weighted difference between an actual measurement and a measurement prediction as shown below (See (Gelb, 1974) for details):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-). \tag{5}$$

In equation (5), the difference $(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$ is called the measurement innovation and the matrix $\mathbf{K}_k$ is called the *gain* matrix. The $\mathbf{K}_k$ which minimizes the MSE (i.e. $tr(\mathbf{P}_k)$) has the form (also see (Gelb, 1974)):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T(\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{Q})^{-1}. \tag{6}$$

Note that $\mathbf{Q}$ is the measurement error matrix. If the error is significant, the gain $\mathbf{K}_k$ weighs lightly whereas if the error is not, the gain $\mathbf{K}_k$ weighs heavily. Thus the effect of new measurements on the state depends on the reliability of the data.

With all the above terms, we can describe the Kalman filter algorithm to reconstruct the state from the given firing rate:

I. Discrete Kalman filter time update equations:

At each time t_k, we obtain the *a priori* estimate from the previous time t_{k-1}, then compute its error covariance matrix:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1}, \tag{7}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{W}. \tag{8}$$

II. Measurement update equations:

Using the estimate $\hat{\mathbf{x}}_k^-$ and firing rate $\mathbf{z}_k$, we update the estimate using equation (5), and compute its error covariance matrix. The process is described by:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T(\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{Q})^{-1}, \tag{9}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-), \tag{10}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- . \tag{11}$$

At each time instant, the Kalman filter iterates between the above two steps and provides an "on-line" estimate of hand kinematics every 50ms. Note that Equation (8), (9) and (11) are independent of the test data. Thus we can compute them "off-line" before the "on-line" estimation. Actually, this is a very nice property of Kalman filter which enables us to *a priori* estimate the performance of the reconstruction. Let $\mathbf{P} = \lim_{k\to\infty} \mathbf{P}_k$, then $tr(\mathbf{P})$ estimates the mean-squared error of the reconstruction.

3. Experimental Results

The experiments below use 182 pre-recorded trials (Paninski et al., 2001). Cross-validation is used in testing both encoding and decoding. The 182 trials are divided into seven sets of 26 trials. For each of the seven sets, we train the model ($\mathbf{A}$, $\mathbf{H}$, $\mathbf{W}$, $\mathbf{Q}$) with the remaining six data sets and test the reconstruction performance for the 26 trials in the excluded set. In this way we test the model on all 182 trials such that the test data is always excluded from the training data.

In each testing trial, we let the predicted initial condition equal the real initial condition and $\mathbf{P}_0^- = 0$. Then the update equations in Section 2. are applied. Some example reconstructed trajectories are shown in Figure 2. By inspection, the reconstructions suggest that the mean firing rates do encode information about the arm movement and that the Kalman filter algorithm is a reliable way to decode the movement.

Figure 3 shows the reconstruction of each component of the state variable for one example trial. We notice that the reconstruction of position and velocity is fairly successful, but the method fails to recover acceleration. This is discussed below.

Conventional r^2 squared error is used here to illustrate the accuracy of the reconstruction in both x- and y-position:

$$r^2 = \left(1 - \frac{\sum_k (x_k - \hat{x}_k)^2}{\sum_k (x_k - \bar{x})^2}, 1 - \frac{\sum_k (y_k - \hat{y}_k)^2}{\sum_k (y_k - \bar{y})^2}\right),$$

where x_k and y_k are the true values and $\bar{x}$ and $\bar{y}$ are the mean of these x and y values respectively.

In Figure 2 (e) and (f) the shapes of the reconstructed trajectories are similar to the true trajectories but they are spatially shifted resulting in a large r^2 error. The *correlation coefficient* provides a more appropriate measure of trajectory shape reconstruction: $cc =$

$$\left(\frac{\sum_k (x_k - \bar{x})(\hat{x}_k - \bar{\hat{x}})}{\sqrt{\sum_k (x_k - \bar{x})^2 \sum_k (\hat{x}_k - \bar{\hat{x}})^2}}, \frac{\sum_k (y_k - \bar{y})(\hat{y}_k - \bar{\hat{y}})}{\sqrt{\sum_k (y_k - \bar{y})^2 \sum_k (\hat{y}_k - \bar{\hat{y}})^2}}\right).$$

3.1 *Stability*

Equation (8), (9) and (11) define the evolution of the gain matrix $\mathbf{K}_k$ and error matrices $\mathbf{P}_k, \mathbf{P}_k^-$. For reliable estimates these matrices should be stable. Figure 4 illustrates that they stabilize (converge) very quickly and then remain constant.

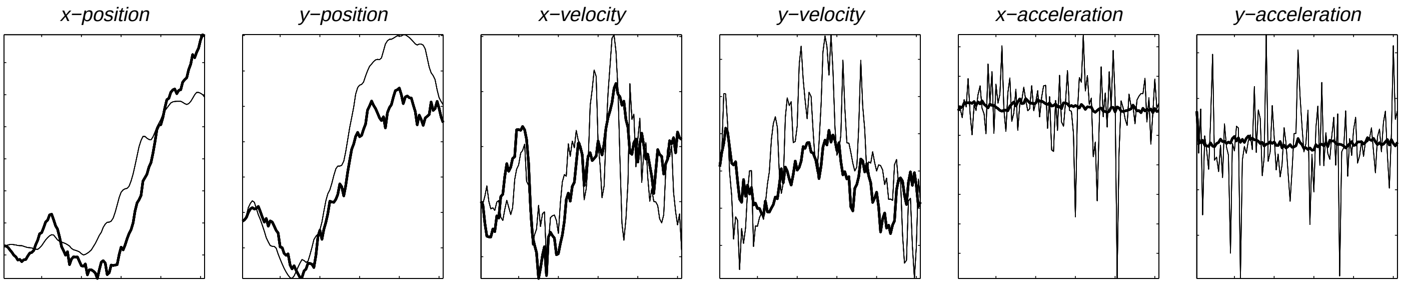

Figure 3: Reconstruction of each component of the system variable for one trial: true target trajectory (thin) and reconstruction using the Kalman filter (thick).

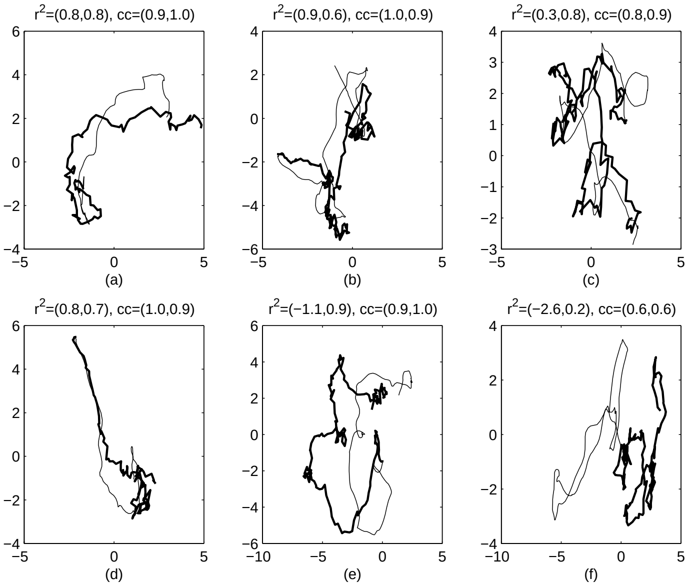

Figure 2: Reconstruction on some test trials: true target trajectory (thin) and reconstruction using the Kalman filter (thick).

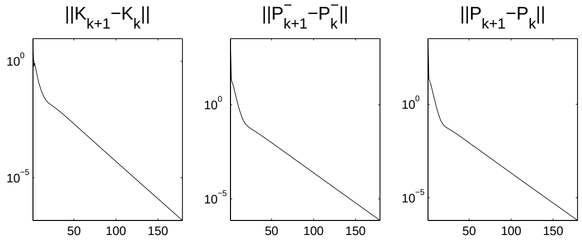

Figure 4: The $\mathbf{L}^2$ norm of the difference of the consecutive matrices for $\mathbf{P}_k, \mathbf{P}_k^-$ and $\mathbf{K}_k$. As k increases, the norm decreases exponentially (since its logarithm decreases linearly).

3.2 Comparison with linear filtering

Linear filtering is a popular method for relating hand position and firing activity (Georgopoulos et al., 1986, Serruya et al., 2002, Warland et al., 1997). Let x_k be the x-position at time $t_k = k\Delta t$ ($\Delta t = 50$ms), $k = 0, \cdots, M-1$, where M is the number of time steps in a trial. Assume $\mathbf{x}_k$ is a linear combination of firing rates of all obtained neurons at time $t_{k-N}, t_{k-N+1}, \cdots, t_k$ with a constant offset, i.e.

$$\mathbf{x}_k = a + \sum_v \sum_{j=0}^{N} r_{k-j}^v f_j^v,$$

where a is the constant offset, r_{k-j}^v is the firing rate of neuron v at time t_{k-j}. The $\{f_j^v\}$ are the filter coefficients.

Let $\mathbf{x} = (x_0, \cdots, x_{M-1})$ is the real x-position (we can do the same for y-position). We define response matrix R as (Warland et al., 1997)

$$\begin{bmatrix} 1 & r_0^1 & \cdots & r_{-N+1}^1 & \cdots & r_0^C & \cdots & r_{-N+1}^C \\ 1 & r_1^1 & \cdots & r_{-N+2}^1 & \cdots & r_1^C & \cdots & r_{-N+2}^C \\ \cdot & \cdot & \cdots & \cdot & \cdots & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdots & \cdot & \cdots & \cdot \\ 1 & r_{M-1}^1 & \cdots & r_{M-N}^1 & \cdots & r_{M-1}^C & \cdots & r_{M-N}^C \end{bmatrix}.$$

The reconstruction $\hat{\mathbf{x}}$ is obtained as

$$\hat{\mathbf{x}} = R\hat{\mathbf{f}},$$

where the filter $\hat{\mathbf{f}} = [\hat{a}, \hat{f}_0^1, \cdots, \hat{f}_{N-1}^1, \cdots, \hat{f}_0^C, \cdots, \hat{f}_{N-1}^C]^T$ is estimated by minimizing the square difference between $\hat{\mathbf{x}}$ and $\mathbf{x}$, i.e.

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\text{argmin}} \; \|R\mathbf{f} - \mathbf{x}\| = (R^T R)^{-1} R^T \mathbf{x}.$$

Here we take $N = 10$; that is we are linearly relating hand position and neural firing over 550ms. The linear filter reconstruction is shown in Figure 5. Compared with Figure 2, we see that the Kalman filter reconstruction is smoother and more similar to the real trajectory, while that of linear filter appears chaotic. In Figure 3 we see that the Kalman filter roughly reconstructs velocity in addition to position. This suggests that velocity is correlated with firing and that the comparison with linear filtering on position alone may be unfair (though the linear filter uses more data at a given time instant).

Note that the linear filter can implicitly capture information about the relationship between position and

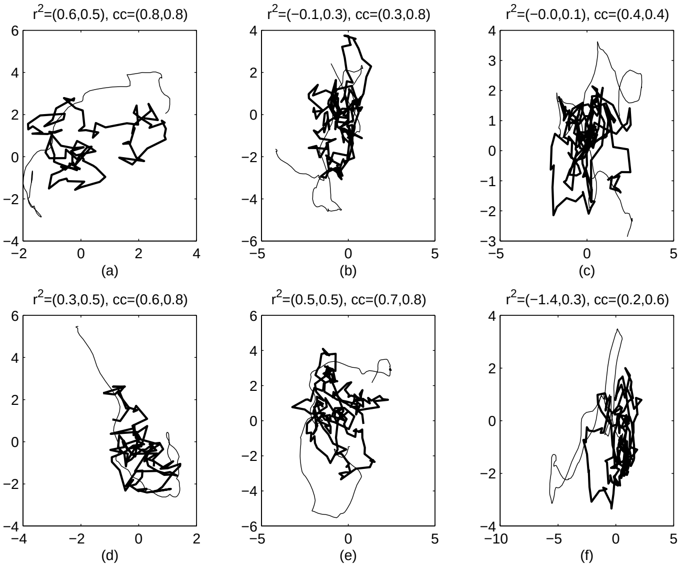

Figure 5: Reconstruction on the test trials using the linear filter: true target trajectory (thin) and reconstruction using the linear filter (thick).

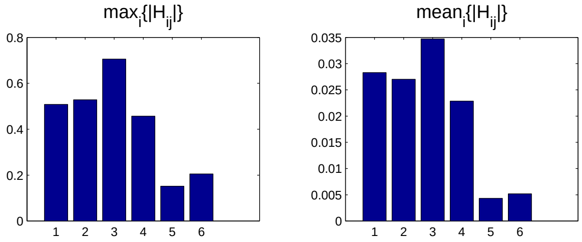

Figure 6: The maximum and mean of each column of $\tilde{\mathbf{H}}$. The position (1&2) and velocity (3&4) have much stronger effect on the model than acceleration (5&6).

velocity because it exploits data over multiple time instants. In the experiments here however, position and velocity are nearly conditionally independent by design (Paninski et al., 2001). This gives an advantage to the Kalman filter which explicitly models velocity as part of the system state.

In cross-validation, each trial is chosen as test data once and only once, and the r^2 error and *correlation coefficient* of its reconstruction (by both the linear and Kalman filters) are calculated. While the r^2 error of the Kalman reconstruction was better than the linear filter reconstruction about half the time, the correlation coefficient was better 91% of the time for the x-position and 80% of the time for y-position.

While linear filtering is extremely simple, it lacks many of the desirable properties of the Kalman filter. The method requires long windows in which to collect data. For rapid motions, this long time window will be inappropriate yet smaller time windows lead to very inaccurate results. Additionally, the linear filter does not make the system dynamics and noise models explicit. In contrast, the Kalman filter provides an explicit generative model, a clear probabilistic interpretation, an incremental estimate of the state that improves over time, and an estimate of the uncertainty in the state. Computationally, the Kalman filter is simple to train and the real-time implementation of tracking is trivial.

3.3 Analysis

The firing rate was described above as a linear stochastic function of position, velocity and acceleration. We *a posteriori* consider the redundancy of the model. In Figure 3, the reconstruction of position and velocity are reasonable while acceleration is not well recovered. Heuristically, it appears that acceleration is redundant. There are two reasons to support this. First, acceleration is a second order difference of position, thus measurements of acceleration tend to be very noisy in real data.

Second, by examining the linear coefficient matrix $\mathbf{H}$ we can evaluate the significance of acceleration in the estimation. Recall that $\mathbf{H}$ encodes the relationship between the kinematics and the firing rate. Each column of $\mathbf{H}$ contains the C coefficients for a particular system variable. The normalized magnitude of these coefficients is related to how much each state variable contributes to the model. Each column is normalized by the effective range of the state variable to create a new matrix $\tilde{\mathbf{H}}$ in which the absolute value of the coefficients are all approximately scaled to the same range. The maximum and mean of each column of $\tilde{\mathbf{H}}$ provide ad hoc measures of the coefficient significance and are plotted in Figure 6. By inspection of $\tilde{\mathbf{H}}$ it appears that the acceleration has a weak effect on the model relative to position and velocity.

We now explore the use of only position and velocity to model the firing rate and repeat the cross-validation experiments above with this reduced state space. The system state becomes $\mathbf{x}_k = [x, y, v_x, v_y]^T_k$ and $\mathbf{A}$, $\mathbf{H}$, $\mathbf{W}$, $\mathbf{Q}$ are updated accordingly.

Figure 7 shows the Kalman filter reconstruction on a few test trials. Comparing Figure 7 with Figure 2, we see that the simplified model and the original model give a visually similar reconstruction. This further supports the conjecture that the acceleration is redundant.

3.4 Optimal lags

The physical relationship between neural firing and arm movement means there exists a positive time lag between them (Moran and Schwartz, 1999, Paninski et al., 2001). If an "optimal lag" can be found, it should improve the model encoding and should improve the accuracy of the decoding.

So far, $\mathbf{z}_k$ is the vector of C cells' firing rates at time t_k but positive and negative time lags may be consid-

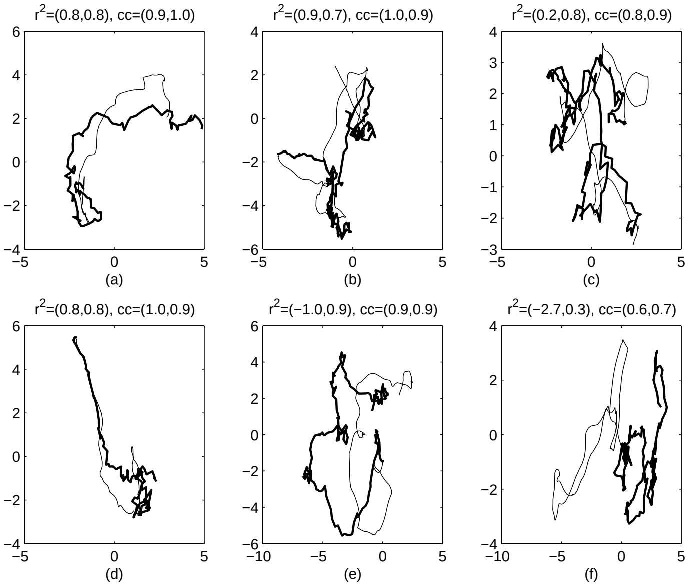

Figure 7: Reconstruction using just position and velocity: true target trajectory (thin) and reconstruction using the Kalman filter (thick).

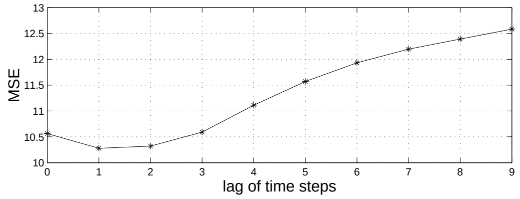

Figure 8: Assuming all $C = 25$ cells have the uniform time lag. This plot is the mean square error (MSE) when the lag is $0, 1, \cdots$ or 9 time steps.

ered instead. In this subsection, we discuss the optimal time lag both uniformly (i.e. the same for all cells) and non-uniformly. Different lags produce different data for the $\mathbf{z}_k$. $\{\mathbf{P}_k\}$'s are the *a posteriori* estimate error covariances defined in equation (4). We exploit this Kalman framework to find the optimal lag; that is, the one that results in the lowest estimate error. Since $\{\mathbf{P}_k\}$ converges very fast, the optimal lag is obtained when the trace of matrix $\mathbf{P}_k$ (for k large enough) is smallest over all possible lags.

Uniform time lag: For each $j \in \{0, 1, 2, \cdots, 9\}$, define $\mathbf{z}_k$ as the vector of C cells' firing rates at time t_{k-j}, then fit the model with training data. $\{\mathbf{P}_k\}$ are computed "off-line". We plot the mean-square error $tr(\mathbf{P}_k)$ for a large enough k in Figure 8, where $j = 0, 1, \cdots, 9$.

Figure 8 shows that the smallest estimation error is achieved when the neural cells have lag for one or two time steps (50-100ms). For longer lags, the error increases monotonically with the lag time.

Non-uniform time lag: More practically, the neural cells may not have uniform spiking activity during the experiment. Some of them act very fast, whereas others slowly. From the analysis of the uniform lag, it seems that the optimal time lag of the cells should not be too long. To simplify our data analysis, we assume the optimal lag for all cells is less than 4 time steps (200 ms).

Due to the more subtle situation here, we reorganize the notation: let $l_i \in \{0, 1, 2, 3, 4\}$ be the lag steps of ith cell, $i = 1, 2, \cdots, C$; The kth firing rate vector is $\mathbf{z}_k = (z_{1,k}, z_{2,k}, \cdots, z_{C,k})$, in which each $z_{i,k}$ is the firing rate of cell i at time step $k - l_i$. For each different choice of $\{l_i\}$ in {0,1,2,3,4}, train the Kalman filter. The Kalman filtering algorithm generates the error covariance matrix $\mathbf{P}_k$ (for k large enough). Letting $mse_{\{l_i, i=1,2,\cdots,C\}} = tr(\mathbf{P}_k)$, our goal is to find the optimal $\{l_i\}$, i.e.

$$\{l_i\} = \text{argmin}_{\{l_i, i=1,2,\cdots,C\}} mse_{\{l_i, i=1,2,\cdots,C\}}$$

A brute force search of all possibilities would require computing the Kalman filter result for 5^{25} possibilities for our data set. This is impractical so, instead, we assume that the correlation of the firing rate among the cells is weak, and we can obtain the suboptimal time lag from the following greedy algorithm:

1. Choose the initial lag l_i in {0,1,2,3,4}, $i = 1, 2, \cdots, C$.

2. For i=1 to C

   Fix $\{l_j, j \neq i\}$. Update l_i by the equation:

   $$l_i = \text{argmin}_{l_i \in \{0,1,2,3,4\}} mse_{\{l_i, i=1,2,\cdots,C\}}.$$

3. Return $\{l_i, i = 1, 2, \cdots, C\}$.

This algorithm requires that the Kalman filtering algorithm be applied to the training data only $5 \times 25 = 125$ times. We tried two initial conditions: one with a uniform lag of 50ms (1 time step), the other with random lag. The greedy result is shown in Figure 9. The different initials conditions result in similar lags, which confirms the assumption that the firing of all cells have weak correlation. Moreover, these two suboptimal time lag solutions have the mean-square error 9.88 and 9.88, which is much less than the that of uniform time lag in Figure 8 (where the minimum is 10.28 at a lag of 50ms). This suggests that a non-uniform time lag is superior to a uniform one.

4. Conclusions

In summary, we described the discrete linear Kalman filter and applied it to model hand movement and neural spiking activity. It is a rigorous probabilistic approach with a well understood theory. Experimental results show its superiority to linear filtering. Moreover, the recursive estimation in 50ms non-overlapping time bins provides a computationally efficient filtering algorithm. In addition to decoding, the approach is useful for analysis. For example, examination of the measurement

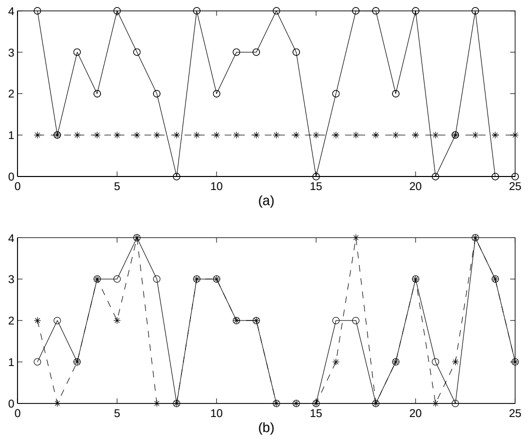

Figure 9: (a) Initial lag: uniform (dashed line with star) and random (solid line with circle); (b)Optimal lag: from uniform initial (dashed line with star) and random initial (solid line with circle).

matrix gives heuristic insight into the coding problem (acceleration appears to not be encoded). Additionally, the framework allows the analysis of optimal lag times which result in improved state estimates. By making its assumptions explicit and by providing an estimate of uncertainty, the Kalman filter offers significant advantages over previous methods.

However, there are some limitations in our model and further work is needed. The Gaussian assumption for the measurement noise might not be appropriate and we are exploring a Poisson model of the neural firing (cf. (Brown et al., 1998)). The linear dynamic model is a strong assumption for the relationship between movement and firing rate; this might be relaxed with a nonlinear (extended) Kalman filter. Additionally, previous work has suggested that a representation of velocity in terms of direction and speed may be more appropriate than x and y velocity. This results, however, in a nonlinear state update equation.

Finally, our current work is exploring the real-time implementation of this method for neural control of 2D cursor movement and further comparisons with population vector methods (Moran and Schwartz, 1999) and particle filtering techniques (Gao et al., 2002).

Acknowledgments. This work was supported by the Keck Foundation, National Institutes of Health (#R01 NS25074 and #N01-NS-9-2322), DARPA (#MDA972-00-1-0026), and the National Science Foundation (ITR Program award #0113679). We are very grateful to M. Fellows, L. Paninski, and N. Hatsopoulos who provided the neural data and valuable insight. We also thank David Mumford for his helpful comments.

References

Brown, E., Frank, L., Tang, D., Quirk, M., and Wilson, M. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. of Neuroscience*, 18(18):7411–7425.

Brown, R. G. and Hwang, P. Y. C., (Eds.) (1997). *Introduction to Random Signals and Applied Kalman Filtering, Third Edition.* John Wiley & Sons, Inc.

Chapin, J. K. and Nicolelis, M. A. (1999). Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations. *J. of Neuroscience Methods*, 94:121–140.

Gao, Y., Black, M. J., Bienenstock, E., Shoham, S., and Donoghue, J. P. (2002). Probabilistic inference of hand motion from neural activity in motor cortex. *Advances in Neural Information Processing Systems 14, The MIT Press.*

Gelb, A., (Ed.) (1974). *Applied Optimal Estimation.* MIT Press.

Georgopoulos, A., Schwartz, A., and Kettner, R. (1986). Neural population coding of movement direction. *Science*, 233:1416–1419.

Isaacs, R. E., Weber, D. J., and Schwartz, A. B. (2000). Work toward real-time control of a cortical neural prosthesis. *IEEE Trans. on Rehabilitation Engineering*, 8(2):196–198.

Jacobs, O. L. R., (Ed.) (1993). *Introductin to Control Theory, Second Edition.* Oxford University Press.

Moran, D. and Schwartz, B. (1999). Motor cortical representation of speed and direction during reaching. *J. of Neurophysiology*, 82(5):2676–2692.

Paninski, L., Fellows, M., Hatsopoulos, N., and Donoghue, J. P. (2001). Temporal tuning properties for hand position and velocity in motor cortical neurons. *submitted, J. of Neurophysiology.*

Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., and Donoghue, J. P. (2002). Brain-machine interface: Instant neural control of a movement signal. *Nature*, (416):141–142.

Warland, D., Reinagel, P., and Meister, M. (1997). Decoding visual information from a population of retinal ganglion cells. *J. of Neurophysiology*, 78(5):2336–2350.

Welch, G. and Bishop, G. (2001). An introduction to the kalman filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, Chapel Hill,NC 27599-3175.

Wessberg, J., Stambaugh, C., Kralik, J., Beck, P., Laubach, M., Chapin, J., Kim, J., Biggs, S., Srinivasan, M., and Nicolelis, M. (2000). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408:361–365.