Proceedings of the 26th Annual International Conference of the IEEE EMBS
San Francisco, CA, USA • September 1-5, 2004

Closed-Loop Neural Control of Cursor Motion using a Kalman Filter

W. Wu* A. Shaikhouni† J. P. Donoghue† M. J. Black‡
*Division of Applied Mathematics, Brown University, Providence, RI, USA
†Department of Neuroscience, Brown University, Providence, RI, USA
‡Department of Computer Science, Brown University, Providence, RI, USA

Abstract— **Recently, we proposed a Kalman filter method to model the probabilistic relationship between neural firing in motor cortex and hand kinematics. In this paper, we demonstrate on-line, closed-loop, neural control of cursor motion using the Kalman filter. In this task a monkey moves a cursor on a computer monitor using either a manipulandum or their neural activity recorded with a chronically implanted micro-electrode array. A number of advantages of the Kalman filter were explored during the on-line tasks and we found that the Kalman filter had superior performance to previously reported linear regression methods. While the results suggest the applicability of the Kalman filter for neural prosthesis applications, we observed the decoded cursor position was nosier under brain control as compared with manual control using the manipulandum. To smooth the cursor motion without decreasing accuracy we propose a method that smoothes the neural firing rates. This smoothing method is described and its validity is quantitatively evaluated with recorded data.**

Index Terms— **neural prosthesis, neural control, brain-machine interface, closed-loop control, primary motor cortex, Kalman filter.**

I. Introduction

Real-time neural decoding algorithms are necessary for the development of neural motor prostheses. A variety of algorithms have been proposed to convert neural activity into a voluntary control signal [1]–[4]. In our recent work, we proposed a Kalman filter to decode the neural firing rate obtained from multi-electrode recordings [5], [6] to produce an estimate of hand kinematics. We showed that the method has a number of benefits for neural prosthesis applications: 1) it requires only a small amount of "training" data; 2) the computational cost for the parameter estimation (learning) of the model is negligible; 3) it provides accurate and efficient state estimation (less than 10ms for each time step). Because of these advantages, we argued that the Kalman filter could be used for closed-loop neural control.

This paper is a continuation of our recent work; here, for the first time, we demonstrate the Kalman filter in a closed-loop neural control task. The experimental paradigm required a monkey to control the two-dimensional (2D) motion of a feedback cursor viewed on a video monitor. The animal's task was to move the cursor to hit targets that appeared at random locations on the screen. In the experiment, a single macaque monkey was implanted with an electrode array and the Kalman filter was used to decode the neural activity in real time to estimate the cursor position. By closed-loop we mean that the brain directly controls the cursor and the visual feedback to the animal about the cursor position closes the loop.

This work was supported by the DARPA BioInfoMicro Program, NIH-NINDS #NS25074, and NSF ITR #0113679.
Correspondence: W. Wu (weiwu@dam.brown.edu)

To evaluate the performance of the method we compared the animal's ability to hit targets with both the Kalman filter and a simple linear regression method which has been widely used for neural decoding. We also evaluated the accuracy of the method in off-line data analysis. While the advantages of the Kalman filter have been clearly described in our previous work [5], [6] here we show that the Kalman filter outperforms the linear regression method in the closed-loop neural control task.

Though accurate and efficient, the estimates produced by the Kalman filter are not as smooth as those observed under manual control using a manipulandum. While the significance of this is unclear from the viewpoint of neural control, we explored various methods to produce smooth neural decodings that more closely approximate the motion under manual control. In previous work the decoded hand position produced by the linear regression method was smoothed using a simple windowed average method [2]. In this paper, we propose an on-line smoothing approach that instead smoothes the neural firing rate data. This results in smooth reconstructions of hand motion while providing a good balance between bias (accuracy) and the variance (smoothness) [7].

II. Methods

A. Experimental paradigm

Simultaneous recordings of spikes are acquired from an array consisting of 100 micro-electrodes chronically implanted in the arm area of primary motor cortex (MI) in a Macaque monkey [8]. Multi-unit firing activity on each channel is detected using simple filtering and thresholding. We use the behavioral paradigm described in [2] for two tasks described below.

Manual-control task: The behavioral task for the monkey is referred to as a step-tracking task, or more intuitively, a pinball task which is designed to test direct neural control performance [2]. During the task, a target dot was shown on the screen in front of the monkey and the monkey moved a manipulandum on a 2D tablet that was parallel to the floor.

The position of the manipulandum (same as the hand) was shown as a feedback cursor on the same screen. The monkey was required to move this cursor to “hit” the target (within a pre-specified distance). When the target was acquired, it disappeared and then reappeared in a new random location. Each time the target appeared, the monkey moved to hit the new location.

The hand trajectory and the neural activity were recorded simultaneously. The spiking activity was detected via empirically determined threshold settings and a firing rate was computed using non-overlapping $70ms$ time bins [6]. The position, velocity, and acceleration of the hand were also computed every $70ms$.

Neural-control task: In the closed-loop neural control task the experiment remained the same except that the motion of the cursor was controlled by the decoded neural signals. This decoding was performed using a Kalman filter. To quantitatively compare with related work, the same procedure was repeated using a linear regression method for decoding. These two methods are described below. Recordings were made during experiments over several months and details of the experiments are described in Section III.

B. Statistical methods

1) Kalman filter: Here, we briefly describe the Kalman filter model and its decoding algorithm. The details can be found in our recent work [5], [6]. In general, decoding involves estimating 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 $\Delta t = 70ms$ in our experiments. The Kalman filter 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; the state itself is linearly related over time as well.

Such assumptions can be described in the following two equations:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{q}_k, \quad (1)$$

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_k, \quad (2)$$

where $k = 1, 2, \cdots, M$, M is the number of time steps in the trial, and $\mathbf{H} \in \Re^{C\times 6}$, $\mathbf{A}_k \in \Re^{6\times 6}$ are the linear coefficient matrices. The noise terms $\mathbf{q}_k, \mathbf{w}_k$ are assumed zero mean and normally distributed, i.e. $\mathbf{q}_k \sim N(0, \mathbf{Q}), \mathbf{Q} \in \Re^{C\times C}$, $\mathbf{w}_k \sim N(0, \mathbf{W}), \mathbf{W} \in \Re^{6\times 6}$. These equations define a linear Gaussian model from which the state and its uncertainty can be estimated recursively using the the Kalman filter algorithm [5], [6].

2) Linear regression method: The linear regression method (also referred to as a discrete Wiener filter [4]) has been used in the decoding of neural signals in motor cortex, and particularly in closed-loop neural control tasks [1], [2], [4]. We briefly describe it here as the baseline method for comparison.

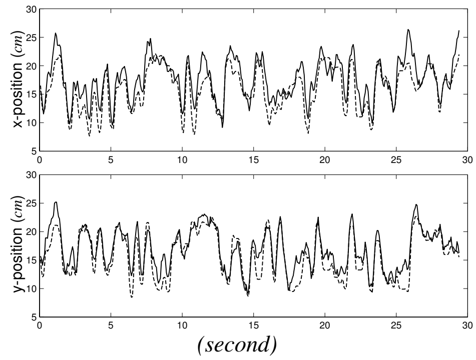

Fig. 1. Reconstruction for the x and y-position using the Kalman filter: first row: x-position; second row: y-position; dashed lines: true trajectories; solid lines: reconstructed trajectories.

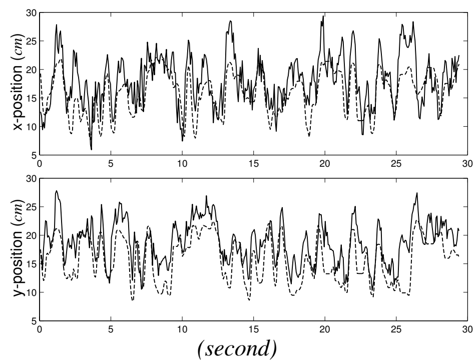

Fig. 2. Reconstruction for the x and y-position using the linear regression method: the notations are the same as Figure 1.

The linear regression method reconstructs hand position as a linear combination of the firing rates over some fixed time period; that is,

$$x_k = a + \sum_{i=1}^{C}\sum_{j=0}^{N} z_{k-j}^i f_j^i,$$

where x_k is the x-position (or, equivalently, the y-position) at time $t_k = k\Delta t$ ($\Delta t = 70ms$), $k = 1, \cdots, M$, where M is the number of time steps in a trial, a is a constant offset, z_{k-j}^i is the firing rate of neuron i at time t_{k-j}, and f_j^i are the filter coefficients. The coefficients can be learned from training data using a simple least squares technique. In our experiments here we take $N = 10$ which means that the hand position is determined from firing data over $0.7s$.

III. Results

A. Off-line reconstruction

For off-line analysis we performed six experiments. For each experiment the recorded data was divided into separate training and testing datasets. Each dataset (training or testing) was approximately 1 to 2 minutes long. For each experiment we trained both the Kalman filter and

TABLE I

OFF-LINE RECONSTRUCTION

# of cells	Kalman filter		linear regression	
	CC	$MSE(cm^2)$	CC	$MSE(cm^2)$
23	(0.79,0.82)	13.0	(0.70,0.72)	18.9
30	(0.88,0.79)	10.6	(0.85,0.72)	12.2
36	(0.75,0.74)	19.0	(0.77,0.64)	19.2
26	(0.71,0.76)	20.1	(0.71,0.74)	19.3
69	(0.88,0.89)	9.7	(0.72,0.78)	28.1
69	(0.86,0.88)	10.6	(0.71,0.80)	15.9

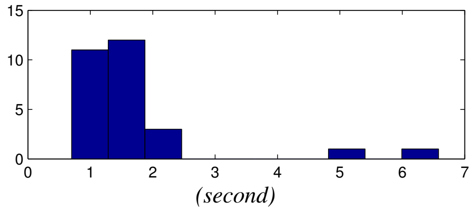

Fig. 3. Histogram of time to the target using Kalman filter decoding method in the fourth experiment. We see that most of time the target is promptly acquired, while there are a few outliers for which it took the animal 5-6s to acquire the target.

linear regression models using the training set, then reconstructed trajectories were computed for the corresponding test dataset. Some example reconstructions (for the 5th test dataset) are shown in Figures 1 and 2. As a summary, Table I shows the decoding results in all six test datasets where the correlation, CC, and the mean squared error, MSE, are used to quantitatively describe the accuracy. Note that the number of recorded cells differed from day to day and consequently we tested using the model trained on data from that same day. The significant increase in the number of cells for the last two experiments was due to the implantation of a new array.

Table I, shows that the Kalman filter has better decoding performance than the linear regression method for both criteria. This is consistent with our previous observations on different datasets [5], [6]. Note that the linear regression results are reported without the post hoc smoothing described below; the accuracy is consistently lower with smoothing.

B. Closed-loop neural control

Each closed-loop experiment had two phases. The training phase was the same as in the off-line experiments. The subject's hand movement and neural firing were recorded and used to train the two models.

After building the model, we switched to the second phase involving closed-loop neural control. In this stage, the motion of the feedback cursor was controlled by either the Kalman filter or the linear regression method. We performed four experiments using the Kalman filter and three using the linear regression method. The results are summarized in Table II.

Table II shows the total time for the experiment, the number of targets hit during this time, and the rate at which the animal hit the targets. For more detail we show

TABLE II

CLOSED-LOOP NEURAL CONTROL

# of cells	Kalman filter			linear regression		
	time	targets	rate	time	targets	rate
17	60sec	38	38/min			
30	105sec	55	31/min	58sec	24	25/min
36	57sec	28	29/min	42sec	15	21/min
69	45sec	28	37/min	60sec	22	22/min

a histogram of the time required to hit the targets using the Kalman filter for the fourth experiment in Figure 3. Most of time the targets were quickly acquired, while there a few outliers requiring much longer. To provide a criterion for the comparison of the Kalman filter and the linear regression method we take the number of targets hit per unit time (see the columns under "rate" in Table II). For these initial experiments we observe that the monkey appears to consistently acquire targets at a higher rate with the Kalman filter as compared with to the linear regression method.

IV. DISCUSSION

The results suggest that the Kalman filter decoding of motor cortical activity is appropriate for neural prosthesis applications and our preliminary results suggest that it is more accurate in this context than traditional linear regression methods. More experiments with more animals, however, are needed to confirm these observations.

We note here that the linear regression reconstruction is very erratic and requires post hoc smoothing to produce useful closed-loop cursor control (see [2]). We performed this smoothing using a moving average filter where the average is computed over 10 time bins (700ms). Figure 4 is an example of the reconstruction of the x hand position after smoothing. The figure illustrates that the averaging procedure introduces a time lag that contributes to a decrease in decoding accuracy in off-line experiments. Quantitatively, we obtained the decoding accuracy: $CC = (0.55, 0.58)$, $MSE = 33.4$. For comparison, the accuracy of the linear regression method without smoothing for this data was $CC = (0.72, 0.78)$ and $MSE = 28.1$.

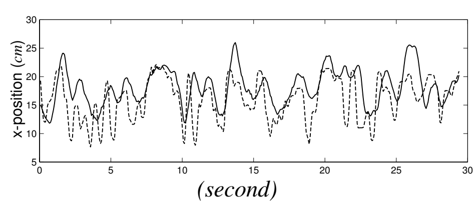

Fig. 4. Linear regression method with smoothing; reconstructed x-position. Dashed lines: true trajectory; solid lines: reconstructed trajectory.

The state equation in the Kalman filter links the estimate at neighboring time steps and this helps to smooth decoding results. While the Kalman filter was less jerky than the linear regression method without smoothing, the decoded cursor motion was still less smooth than the cursor motion observed during manual control. This can be seen in Figure 5 where the power spectra of the true and reconstructed trajectories are shown overlaid. The reconstructed trajectory has larger power in the high frequencies (above 4Hz) than does the true motion.

We posit that the jerkiness in the reconstruction is due to the finite approximation of the neural firing rate derived from our $70ms$ time bins. With more cells, this is less likely to be a problem but, for the near term, the population size for neural prosthesis applications is unlikely to be significantly larger than that used here.

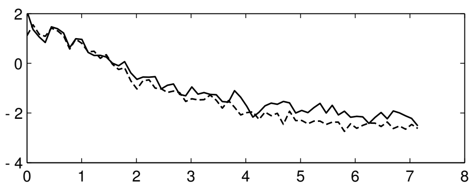

Fig. 5. Logarithm of power spectra of true x-position and reconstructed x-position in the 5th test dataset using the Kalman filter: dashed line: true; solid line: reconstructed.

To produce a smoother control signal, we propose a weighted low-pass filter of the firing rate signals. In order to minimize the time lag introduced by smoothing the firing rate we weight current measurements more heavily. Assume for the i^{th} cell, the observed firing rate at the k^{th} time step is: z_k^i; then the smoothed firing rate at the same time step is:

$$w_1 z_{k-N+1}^i + w_2 z_{k-N+2}^i + \cdots + w_{N-1} z_{k-1}^i + w_N z_k^i,$$

where N is the length of the smoothing window, $\{w_1, \cdots, w_N\}$ are the weights of firing rates. Note for neural control the smoothing window cannot look forward in time without introducing a time lag.

Figure 6 shows the power spectra for the true and reconstructed trajectories where the reconstruction uses the smoothed firing rates in both training and test datasets. Here we took $N = 5$ and the linear weights to be $w_i = \frac{i}{15}, i = 1, \cdots, 5$. The reconstructed trajectory has similar power in the high frequencies to the true one suggesting that smoothing the firing rate also smoothes the estimated reconstruction. In contrast to the smoothed linear regression results, we do not observe as large a decrease in accuracy. Quantitatively, we obtain the decoding accuracy of $CC = (0.84, 0.85)$, $MSE = 12.0$ for the smoothed case. For the unsmoothed firing rates the accuracy was $CC = (0.88, 0.89)$ and $MSE = 9.7$.

V. Conclusion

Through off-line and on-line experiments we demonstrated that the Kalman filter decoding method can be successfully exploited for closed-loop 2D neural motor control tasks. Furthermore, we compared the Kalman filter with a linear regression method with respect to their closed-loop performance. For this limited dataset the results showed that the Kalman filter was superior (in terms of number of targets

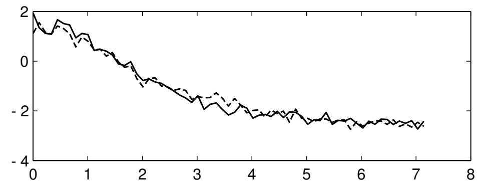

Fig. 6. Logarithm of power spectra of true x-position and reconstructed x-position in the 5th test dataset using the Kalman filter with smoothed firing rates: dashed line: true; solid line: reconstructed.

hit in unit time). We also proposed an on-line approach for smoothing the firing rate and showed that it produced smoother off-line hand reconstructions without significant loss of decoding accuracy.

Future work will focus on determining the source of jerkiness in the Kalman reconstruction and will explore additional smoothing approaches. In the work here we used a very simple linear system model to propagate the hand state over time. In the future we will explore new dynamic state models and test their performance in closed-loop tasks. These experiments will be repeated in additional animals and with more complex tasks including both motion and static hold periods. The use of multiple arrays will allow us to explore neural control using activity from multiple brain areas (parietal in addition to motor cortex).

Acknowledgment

We are very grateful to B. Travers for her significant help with the closed-loop experiments. We also thank D. Mumford for his helpful comments.

References

[1] J. Wessberg, C. Stambaugh, J. Kralik, L. M. Beck, P., J. Chapin, J. Kim, S. Biggs, M. Srinivasan, and M. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, 2000.

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

[3] D. Taylor, S. Helms Tillery, and A. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.

[4] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS, Biology*, vol. 1, no. 2, pp. 001–016, 2003.

[5] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, and J. P. Donoghue, "Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter," in *SAB'02-Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices*, August 2002, pp. 66–73.

[6] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhouni, and J. P. Donoghue, "Neural decoding of cursor motion using a Kalman filter," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 133–140.

[7] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[8] J. P. Donoghue, J. N. Sanes, N. G. Hatsopoulos, and G. Gaal, "Neural discharge and local field potential oscillations in primate motor cortex during voluntary movements," *Journal of Neurophysiology*, vol. 79, pp. 159–173, 1998.