

Mixing Decoded Cursor Velocity and Position from an Offline Kalman Filter Improves Cursor Control in People with Tetraplegia*

Mark L. Homer¹, Matthew T. Harrison², Michael J. Black³, János A. Perge⁴, Sydney S. Cash⁵, Gerhard Friehs⁶, and Leigh R. Hochberg⁷

Abstract—Kalman filtering is a common method to decode neural signals from the motor cortex. In clinical research investigating the use of intracortical brain computer interfaces (iBCIs), the technique enabled people with tetraplegia to control assistive devices such as a computer or robotic arm directly from their neural activity. For reaching movements, the Kalman filter typically estimates the instantaneous endpoint velocity of the control device. Here, we analyzed attempted arm/hand movements by people with tetraplegia to control a cursor on a computer screen to reach several circular targets. A standard velocity Kalman filter is enhanced to additionally decode for the cursor’s position. We then mix decoded velocity and position to generate cursor movement commands. We analyzed data, offline, from two participants across six sessions. Root mean squared error between the actual and estimated cursor trajectory improved by $12.2 \pm 10.5\%$ (pairwise t-test, $p < 0.05$) as compared to a standard velocity Kalman filter. The findings suggest that simultaneously decoding for intended velocity and position and using them both to generate movement commands can improve the performance of iBCIs.

*The research was supported by the Rehabilitation Research and Development Service, Office of Research and Development, Department of Veterans Affairs (Merit Review Award B6453R; Career Development Transition Award B6310N). Support was also provided by the National Institutes of Health: NIDCD (R01DC009899), NICHD-NCMRR (N01HD53403 and N01HD10018); DARPA REPAIR (N66001-10-C-2010); the Doris Duke Charitable Foundation; the MGH-Deane Institute for Integrated Research on Atrial Fibrillation and Stroke. The pilot clinical trial into which participant S3 was recruited was sponsored in part by Cyberkinetics Neurotechnology Systems (CKI). The contents do not represent the views of the Department of Veterans Affairs or the United States Government.

¹M. L. Homer is with Biomedical Engineering, Brown University, Providence, Rhode Island 02912 USA, (mark_homer@brown.edu).

²M. T. Harrison is with the Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912 USA, (matthew_harrison@brown.edu).

³M. J. Black is with the Max Planck Institute for Intelligent Systems, Tübingen, Germany and the Department of Computer Science, Brown University, Providence, Rhode Island 02912 USA, (black@tuebingen.mpg.de).

⁴J. A. Perge is with the Center for Neurorestoration and Neurotechnology, Rehabilitation Research & Development Service, Department of Veterans Affairs Medical Center, Providence, Rhode Island 02908 USA; the School of Engineering, Brown University, Providence, Rhode Island 02912 USA, (janos_perge@brown.edu).

⁵S. S. Cash is with the Department of Neurology, Massachusetts General Hospital and Harvard Medical School, Boston, MA 02114 USA, (scash@partners.org).

⁶G. Friehs is with the American Medical Center, Nicosia, Cyprus

⁷L. R. Hochberg is with the Center for Neurorestoration and Neurotechnology, Rehabilitation Research & Development Service, Department of Veterans Affairs Medical Center, Providence, Rhode Island 02908 USA; the School of Engineering, Brown University, Providence, Rhode Island 02912, USA; the Department of Neurology, Massachusetts General Hospital, Boston, Massachusetts 02114, USA, and Harvard Medical School, Boston, Massachusetts 02115, USA, (leigh.hochberg@brown.edu).

I. INTRODUCTION

In intracortical brain computer interfaces (iBCIs), a microelectrode array implanted in the motor cortex records neural activity. Computers then decode the neural activity into intended movements which drive external devices. With the technology, people with tetraplegia have controlled a computer cursor as well as robotic aids in clinical research [1], [2], [3]. iBCIs seek to enable those with paralysis or limb loss to carry out better activities of daily living.

Kalman filtering is a popular choice for decoding [4]. Here, we explore use of the Kalman filter to control a cursor to reach several targets on a computer screen. Two dimensional cursor velocity is decoded typically from time binned counts of extracellular action potentials, i.e. spikes [5], [6], [2]. However, other motor variables have been decoded in the context of goal directed reaching using iBCIs. One of the first successful closed-loop demonstrations decoded cursor position [7]. Since then, joint torques [8], angles [9] as well as grip action [10] have also been estimated.

Previously, position and velocity information have been shown to be encoded in neural activity at the same time during reach movements [11] and a Kalman filter has simultaneously decoded for both [4]. During decoding of an entire reach movement, factoring in the displayed location of the cursor reduced the time to reach targets with a neurally controlled cursor [12]. Here, similar to [4], a Kalman filter decodes simultaneously for cursor velocity and position. However, instead of controlling the cursor by integrating velocity [6] or linking it directly to the decoded position [4], we mix the velocity and position estimates from the Kalman filter to generate cursor movements. In people with tetraplegia making attempted reaching movements, the strategy results in improved offline reconstructions of movement intention. The boost in accuracy promises to improve iBCIs for the benefit of those who have lost motor control due to injury or disease.

II. NEURAL RECORDINGS AND SESSION TASK

A. Recordings

The details of the recordings have been described elsewhere [13], [2]. Briefly, two people with tetraplegia, referred to as S3 and T2 participated in the BrainGate2 clinical research sessions¹. S3 enrolled in the study when she was 54 years old; T2 enrolled when he was 65 years old. A

¹CAUTION: Investigational Device. Limited by Federal Law to Investigational Use.

96 microelectrode array (Blackrock Microsystems, Salt Lake City) was implanted in the hand-arm region [14] of the motor cortex. Each electrode measured 1.5 mm in length and they were evenly distributed in a 10x10 grid on a 4x4 mm platform with 400 μm spacing.

B. Signal Pre-Processing

A percutaneous connection sent the voltage signals onto various stages of pre-processing including amplification, 0.3 Hz to 7.5 kHz analog bandpass filtering, 30 kHz analog-to-digital conversion, and 250 Hz digital highpass filtering. For each digitized channel, either time-amplitude windows or threshold crossings were setup up to detect extracellular actions potentials, originating from a single cell or a group, i.e. unit spikes. This spiking activity was then time binned (100 msec or 20 msec) to supply a vector of spike counts to the Kalman filter at each time step. Detected spiking units with a mean and standard deviation > 1 Hz were selected for the analysis which resulted between 23 to 90 spiking units depending upon the session.

C. Session Task

We evaluated the decoders using three sessions with S3 and three with T2. Sessions with S3 (T2) were recorded from 986 and 1003 (67 and 82) days after implantation. We examined open loop blocks of trials. In the trials, a computer program moved a cursor through a series of preset motions on a computer screen. There were five circular targets, one at the screen's center and the rest in locations up, down, left, and right relative to the center target. During a trial, the cursor moved from one target to the next in a center-out-and-back fashion, meaning after the cursor completed a trial taking it to one of the peripheral targets, it would move back to the center in the subsequent trial (Figure 1a). During each block of trials, all peripheral targets were visited in a pseudo-random order. The cursor headed in a direct, straight line toward each target and its speed followed one of several possible profiles. All were truncated Gaussian curves (Figure 1b), but they ranged from 2.5 to 5 seconds.

During the trial blocks, the person was instructed to attempt hand and arm movements as if they were controlling a computer mouse to generate the same cursor motions as they saw on the screen. Although their paralysis prevented any functional arm and hand movements, their intended reaching movements were assumed to be a close match to that displayed on the screen. Supporting this assumption, during a session, the data generated from the blocks were all used to calibrate Kalman filter decoders that were run online in subsequent trials providing better than average neural cursor control.

III. DECODING & CURSOR CONTROL METHODS

A. Kalman Filter

A typical approach in iBCIs is to decode using the time invariant version of the Kalman filter to estimate kinematic variables at every time step, k . In the context of our analysis, the kinematic vector, \mathbf{x}_k , describes the intended motion the

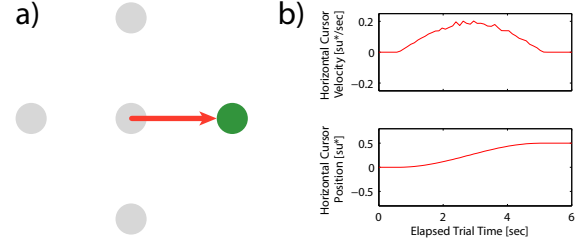


Fig. 1. Illustration of task and decoded variables. **a)** During a trial, i.e. reach, the trajectory of the cursor (red line) moved from a starting position (grey) and ended at a target (green). The motion was controlled by a computer and the person was instructed to attempt movements as if controlling a computer mouse to match the displayed motion. **b)** The cursor's velocity profile (horizontal component displayed) followed a truncated Gaussian curve. As a result, the target's position relative to the target (horizontal component displayed) changed in a smooth monotonic manner during the reach. *Distances are in screen units (su) where the screen is 1.2 su wide by 1.2 su tall.

participant attempts to impart on a computer cursor. The filter estimates \mathbf{x}_k from observations of the spike counts at every time step, \mathbf{z}_k . The change in \mathbf{x}_k over time and the relationship between \mathbf{x}_k and \mathbf{z}_k are modeled as

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + \mathbf{w}_k : \quad \mathbf{w}_k \sim \mathcal{N}(0, W) \quad (1)$$

$$\mathbf{z}_k = H\mathbf{x}_k + \theta + \mathbf{q}_k : \quad \mathbf{q}_k \sim \mathcal{N}(0, Q), \quad (2)$$

where $\mathbf{x}_0 =$ all zeros and $\mathcal{N}(\mu, \Sigma)$ denotes a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ . Equation (1) is often called the state dynamic model and (2) called the tuning model. Model parameters reside in A , W , H , θ , and Q and are estimated via linear regression methods. The calibration procedure uses data from iBCI sessions where the time history of both \mathbf{x}_k and \mathbf{z}_k are assumed known [4].

Given (1) and (2), the maximum a posteriori probability estimate for velocity at the present time step, n , can be found using the recursive equations that define a time invariant Kalman filter [15], [16],

$$\begin{aligned} P_{k-1}^+ &= AP_{k-1}^- A^T + W \\ K_k &= P_{k-1}^+ H^T (HP_{k-1}^+ H^T + Q)^{-1} \\ P_k^- &= (I - K_k H) P_{k-1}^+ \\ \hat{\mathbf{x}}_k &= A\hat{\mathbf{x}}_{k-1} + K_k (\mathbf{z}_k - \theta - HA\hat{\mathbf{x}}_{k-1}), \end{aligned} \quad (3)$$

for $k \in 1, 2, \dots, n$ with chosen initializations $\hat{\mathbf{x}}_0 = \mathbf{x}_0$ and $P_0^- =$ the zero matrix.

B. Velocity Based Cursor Control

A standard approach is to solely decode for the two dimensional velocity vector of the cursor, $\mathbf{x}_k \triangleq \mathbf{v}_k$ [5], [6], [2]. The cursor's commanded location, $\hat{\mathbf{c}}_k$, is then controlled by integrating the estimated velocity, $\hat{\mathbf{v}}_k$,

$$\hat{\mathbf{c}}_k = \hat{\mathbf{c}}_{k-1} + \delta \hat{\mathbf{v}}_k, \quad (4)$$

where δ is the interval between time steps (100 or 20 msec for our data) and $\hat{\mathbf{v}}_k = \hat{\mathbf{x}}_k$.

C. Position Based Cursor Control

Alternatively, we can use the four-dimensional kinematic vector $\mathbf{x}_k \triangleq [\mathbf{v}_k^T, \mathbf{r}_k^T]^T$, where \mathbf{v}_k is velocity as before and \mathbf{r}_k is the two-dimensional position vector of the cursor. It is not obvious how to best use $\hat{\mathbf{x}}_k$ to drive $\hat{\mathbf{c}}_k$. One possibility is to simply use the cursor position component of $\hat{\mathbf{x}}_k$ as in [4], namely,

$$\hat{\mathbf{c}}_k = \hat{\mathbf{r}}_k, \quad (5)$$

which we call position based control. Note, as in Section III-B, one can also only integrate velocity, $\hat{\mathbf{c}}_k = \hat{\mathbf{c}}_{k-1} + \delta\hat{\mathbf{v}}_k$, which turns out to work slightly better than the more standard velocity based control in III-B (the difference being the inclusion of position in the kinematic variables).

D. Mixed Velocity & Position Based Cursor Control

Our innovation here is to mix $\hat{\mathbf{v}}_k$ and $\hat{\mathbf{r}}_k$, both simultaneously estimated from a Kalman filter, to obtain an improved method of commanding $\hat{\mathbf{c}}_k$ via

$$\hat{\mathbf{c}}_k = \hat{\mathbf{c}}_{k-1} + \alpha\delta\hat{\mathbf{v}}_k + (1 - \alpha)\delta\frac{\|\hat{\mathbf{v}}_k\|}{\|\Delta\hat{\mathbf{r}}_k\|}\Delta\hat{\mathbf{r}}_k, \quad (6)$$

where $\alpha \in [0, 1]$ is the mixing coefficient, $\|\cdot\|$ denotes magnitude, and $\Delta\hat{\mathbf{r}}_k = \hat{\mathbf{r}}_k - \hat{\mathbf{c}}_{k-1}$. The change in $\hat{\mathbf{c}}_k$ is now a mixture of two vectors, one is $\delta\hat{\mathbf{v}}_k$ and the other points in the direction of $\hat{\mathbf{r}}_k$.

IV. RESULTS

Figure 2 provides example trajectories driven by the three different cursor control strategies using hold-out data. In the trial, position based control appears less accurate than velocity based control. With $\alpha = 0.7$, mixed velocity and position control closely tracks velocity control, but appears more accurate. The better tracking results from the mixing, where position estimates have some predictive value but are not completely correlated with those developed by integrating estimated velocity.

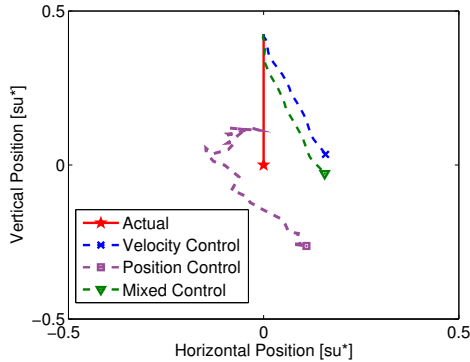


Fig. 2. Example trial showing the true cursor trajectory during a return from the peripheral target to the center and estimated versions by the three control strategies. Markers such as "x" show the ends of the trajectories. For mixed control, $\alpha = 0.7$. *Screen units

We performed five fold cross-validation on each of the six sessions, resulting in 30 tests. To quantify the performance during a trial, we calculated the rms error (RMSE) in euclidean distance between estimated and actual cursor trajectories [4]. Average RMSE over trials was then calculated for each of the 30 tests and cursor control strategies (Figure 3). Mixed velocity and position control improved RMSE by $12.2 \pm 10.5\%$ and $37.8 \pm 10.7\%$ over velocity based and position based control respectively (pairwise t-test, $p < 0.05$). Compared to velocity control, mixed control did better in 28 out of the 30 tests whereas against position control, mixed control was superior in all trials.

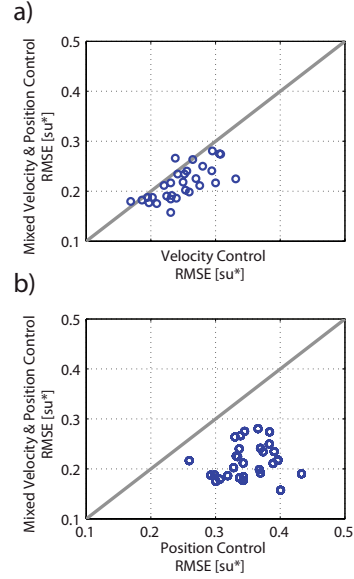


Fig. 3. Mixing velocity and position estimates improves offline decoding performance. **a)** Mixed control's performance (RMSE) compared to that from velocity based control and **b)** position based control. Each point is a test (30 total) and the grey line denotes where the RMSE values are the same, meaning all points below the line indicate a lower error with mixed control. *Screen units

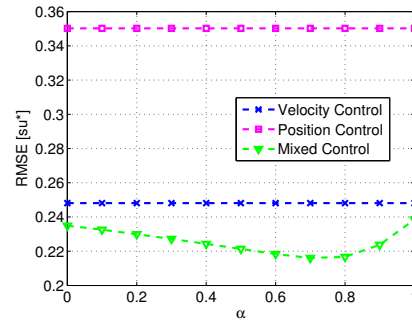


Fig. 4. Mixed cursor control performance (RMSE) as a function of the mixing coefficient, α . An α close to one is being driven mostly by decoded velocity, while α close to zero is influenced mostly by decoded position. RMSE values for velocity based and position based control shown for reference (they are not functions of α). Note, at $\alpha = 0$ or 1, mixed cursor control is still better than either velocity or position control, since mixed control simultaneously decodes for position and velocity. In contrast, velocity control solely decodes for velocity and position control solely for position. *Screen units

While $\alpha = 0.7$ yielded the lowest RMSE values on average, mixed velocity and position control did well under any setting for α (Figure 4). The data show an over reliance on either position or velocity fails to do better than a blending between them. Also, the curve’s approximate convexity and single minima suggest a clear trend in positive gains achieved by mixing.

V. DISCUSSION

The results show that mixing velocity and position improves cursor control over present methods in an offline analysis. Specifically, average RMSE of the trial cursor trajectories was better than those from prior approaches [4], [6]. Between the velocity based and position based control, the former was better, consistent with prior findings [6]. While $\alpha = 0.7$ gave the best results, the findings are not overly sensitive to this tuning parameter.

One untested sensitivity is dependence of the results on the type of task. During the sessions, the person was instructed to attempt to move the cursor to match that displayed on the screen, where the cursor moved in a smooth fashion over a period of seconds. Thus, we view the task as a variant of pursuit tracking. Differences in the best Kalman filtering approach have been noted between pursuit and step tracking [4]. The most appropriate cursor control strategy may be dependent on the task as well.

A major limitation of the results concerns the offline nature of the decoding. Performance gaps between novel decoders and more standard approaches can narrow when people use the algorithms online within an iBCI to neurally control a cursor [17]. In the online, closed-loop setting, the person has constant feedback about the decoder’s ability to estimate their intended movements and thus can attempt to compensate for any decoding errors. However, complete information about the intended movements are also not known, e.g. desired cursor speed. Thus, offline studies often have better information on the actual intended movements of the person, leading to a more direct analysis of decoder inaccuracies.

Nevertheless, a next step is to run the mixed velocity and position based cursor control online in closed-loop trials. Doing so will ascertain whether the decoder can improve task performance. A complementary effort will be to better capture the interplay between intended velocity and position in driving neural activity. Possible future advancements include introducing nonlinearities into the tuning model [18] in order to improve accuracy even further.

ACKNOWLEDGMENT

We thank participants S3 and T2 for their dedication to this research. We thank E. Eskandar, E. Gallivan, E. Berhanu, L. Barefoot, K. Centrella, and B. Sorice for their work in the study.

REFERENCES

[1] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia,” *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.

[2] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm,” *Nature*, vol. 485, no. 7398, pp. 372–375, 2012.

[3] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. C. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, “High-performance neuroprosthetic control by an individual with tetraplegia,” *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.

[4] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, “Bayesian population decoding of motor cortical activity using a kalman filter,” *Neural computation*, vol. 18, no. 1, pp. 80–118, 2006.

[5] 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. Nicolelis, “Learning to control a brain-machine interface for reaching and grasping by primates,” *PLoS Biology*, vol. 1, no. 2, pp. 193–208, 2003.

[6] S. P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, and M. J. Black, “Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia,” *Journal of Neural Engineering*, vol. 5, no. 4, pp. 455–476, 2008.

[7] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, “Instant neural control of a movement signal,” *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.

[8] A. H. Fagg, G. W. Ojakangas, L. E. Miller, and N. G. Hatsopoulos, “Kinetic trajectory decoding using motor cortical ensembles,” *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 5, pp. 487–496, 2009.

[9] C. E. Vargas-Irwin, G. Shakhnarovich, P. Yadollahpour, J. M. Mislou, M. J. Black, and J. P. Donoghue, “Decoding complete reach and grasp actions from local primary motor cortex populations,” *The Journal of Neuroscience*, vol. 30, no. 29, pp. 9659–9669, 2010.

[10] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, “Cortical control of a prosthetic arm for self-feeding,” *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.

[11] N. G. Hatsopoulos, Q. Xu, and Y. Amit, “Encoding of movement fragments in the motor cortex,” *The Journal of neuroscience*, vol. 27, no. 19, pp. 5105–5114, 2007.

[12] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and S. K. V., “A high-performance neural prosthesis enabled by control algorithm design,” *Nature Neuroscience*, vol. 15, no. 12, pp. 1752–1757, 2012.

[13] J. D. Simeral, S. P. Kim, M. J. Black, J. P. Donoghue, and L. R. Hochberg, “Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array,” *Journal of Neural Engineering*, vol. 8, no. 2, p. 025027, 2011.

[14] T. A. Yousry, U. D. Schmid, H. Alkadhi, D. Schmidt, A. Peraud, A. Buettner, and P. Winkler, “Localization of the motor hand area to a knob on the precentral gyrus. a new landmark,” *Brain*, vol. 120, no. 1, pp. 141–157, 1997.

[15] A. Gelb, *Applied Optimal Estimation*. Cambridge, US: MIT Press, 1974.

[16] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York: Wiley-Interscience, 2001.

[17] S. M. Chase, A. B. Schwartz, and R. E. Kass, “Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms,” *Neural Networks*, vol. 22, no. 9, pp. 1203–1213, 2009.

[18] W. Wang, S. S. Chan, D. A. Heldman, and D. W. Moran, “Motor cortical representation of position and velocity during reaching,” *Journal of neurophysiology*, vol. 97, no. 6, pp. 4258–4270, 2007.