34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

A brain machine interface control algorithm designed from a feedback control perspective

Vikash Gilja[1,2*], Paul Nuyujukian[3,4*], Cindy A. Chestek[2,5], John P. Cunningham[5,8], Byron M. Yu[5,6,9,10], Joline M. Fan[3], Stephen I. Ryu[5,11] & Krishna V. Shenoy[2,3,5,6,7]

***Abstract*— We present a novel brain machine interface (BMI) control algorithm, the recalibrated feedback intention-trained Kalman filter (ReFIT-KF). The design of ReFIT-KF is motivated from a feedback control perspective applied to existing BMI control algorithms. The result is two design innovations that alter the modeling assumptions made by these algorithms and the methods by which these algorithms are trained. In online neural control experiments recording from a 96-electrode array implanted in M1 of a macaque monkey, the ReFIT-KF control algorithm demonstrates large performance improvements over the current state of the art velocity Kalman filter, reducing target acquisition time by a factor of two, while maintaining a 500 ms hold period, thereby increasing the clinical viability of BMI systems.**

I. INTRODUCTION

Many existing proof-of-concept brain machine interface (BMI) control algorithms are initially designed, tested, and fit offline using data collected without the BMI, or neural prosthesis, in loop (e.g., [1]–[5]). For example, at the beginning of the session, cursor movement is controlled by the native limb as illustrated in Fig. 1a. During this task, the arm movement kinematics (x_t) and neural activity (y_t) are recorded. These data are used to build a mathematical model used for neural control. The underlying assumption is that observations of neural signal outputs during arm control provide a good estimate of signal characteristics while under brain control (Fig. 1b). However, under brain control a new plant, defined by the dynamics of the neural prosthesis, is presented to the user. This change in the control loop likely alters the user's strategy and neural signal output characteristics.

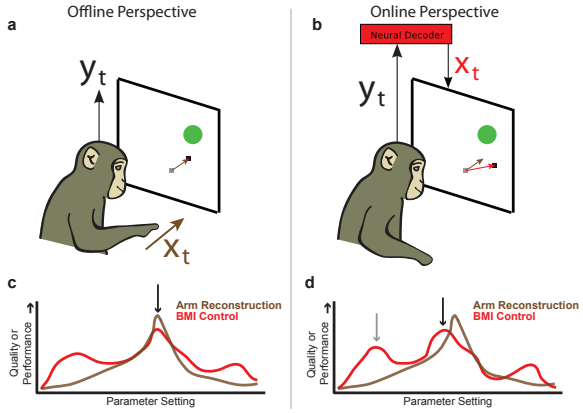

Fig. 1. **A comparison of the offline and online perspectives for BMI design.** (a) cursor control with native arm movements; neural data y_t are collected with arm kinematics x_t to fit the BMI parameters. (b) online neural control, x_t is now the BMI output. (c) hypothetical plot of parameter setting vs. quality/performance from the offline perspective. (d) same hypothetical plot from the online perspective.

Other studies have noted and addressed this change by holding model parameters constant and allowing performance to increase over days as the user learns [6] or by iteratively refining parameters during BMI experiments [7]–[10]. These approaches recognize that control strategies, and therefore model parameters, are best measured and understood during closed-loop BMI experiments. In this study, we adopt this philosophy to develop a new neural control algorithm, the recalibrated feedback intention-trained Kalman filter (ReFIT-KF), taking into account differences between offline arm movement reconstruction and online BMI control in both its algorithmic design and parameter fitting methodology. We test these algorithmic innovations and demonstrate BMI performance gains.

II. THE REFIT-KF ALGORITHM

The ReFIT-KF is composed of two design innovations applied to a Kalman filter based neural control algorithm (Fig. 2a). The kinematic state vector, x_t, represents position and velocity of the cursor ($x_t = [p_t^{vert}, p_t^{horiz}, v_t^{vert}, v_t^{horiz}, 1]^T$), with a constant element to accommodate baseline firing rates. y_t is the measured neural

[1]Dept. of Computer Science, [2]Stanford Inst. for Neuro-Innovation and Translational Neuroscience, [3]Dept. of Bioengineering, [4]School of Medicine, [5]Dept. of Electrical Engineering, [6]Neurosciences Prog., [7]Dept. of Neurobiology, Stanford University, Stanford, CA; [8]Dept. of Engineering, University of Cambridge, Cambridge, United Kingdom; [9]Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom; [10]Dept. of Electrical and Computer Engineering and Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA; [11]Dept. of Neurosurgery, Palo Alto Medical Foundation, Palo Alto, CA

This work was supported by a Helen Hay Whitney postdoctoral fellowship (M.M.C.), Burroughs Welcome Fund Career Awards in the Biomedical Sciences (M.M.C., K.V.S.), the Christopher and Dana Reeve Paralysis Foundation (S.I.R., K.V.S.), Gatsby Charitable Foundation (B.M.Y.), Stanford Graduate Fellowship (C.A.C., J.P.C., J.M.F.), US National Defense Science and Engineering Graduate Fellowships (V.G.), National Science Foundation Graduate Research Fellowships (V.G., C.A.C., J.M.F., M.T.K.), Stanford Medical Scholars Program, HHMI Medical Research Fellows Program, Paul & Daisy Soros Fellowship, Stanford Medical Scientist Training Program (P.N.), Defense Advanced Research Projects Agency Revolutionizing Prosthetics 2009 N66001-06-C-8005, US National Institutes of Health (NIH), National Institute of Neurological Disorders and Stroke Collaborative Research in Computational Neuroscience grant R01-NS054283, and NIH Directors Pioneer Award 1DP1OD006409 (K.V.S.).

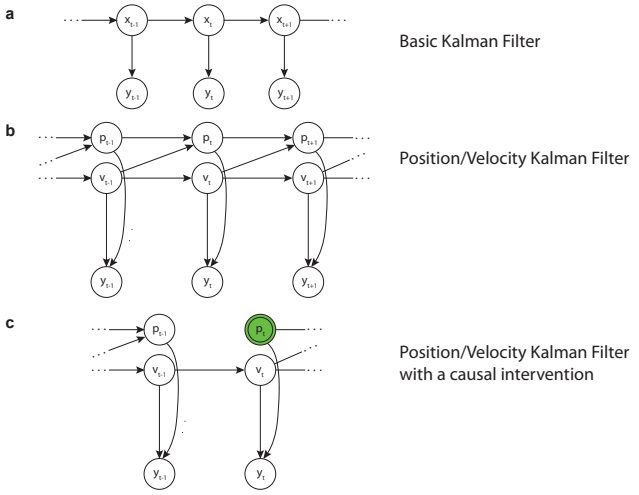

Fig. 2. **Kalman Filter Graphical Models.** (a) Basic model, x_t is kinematics and y_t is neural output at time t. (b) The position (p_t) and velocity (v_t) Kalman. (c) The position and velocity kalman filter modeling position feedback through causal intervention.

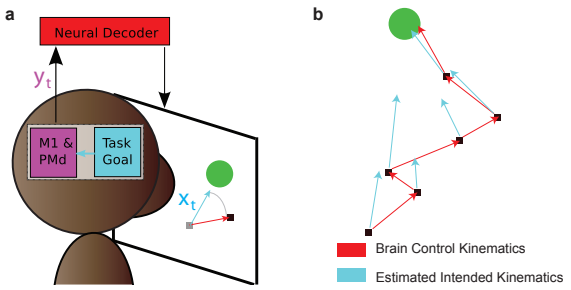

Fig. 3. **Generating an "intention-based" kinematic training set.** (a) the user is engaged in online control with a neural cursor. The neural decoder drives the cursor with a velocity (red vector). The assumed intention is to generate a velocity towards the target. Thus, in the training data velocity vectors are rotated, generating intended velocity estimates (blue vector), used to fit ReFIT-KF parameters. (b) is an example of this transformation applied to data from one trial.

signal, which is binned spike counts. Typical bin widths used in studies range from 10 ms to 100 ms. The system is modeled as linear-Gaussian:

$$x_t = Ax_{t-1} + w_t \quad (1)$$
$$y_t = Cx_t + q_t \quad (2)$$

where $A \in \mathbb{R}^{p \times p}$ and $C \in \mathbb{R}^{k \times p}$ represent kinematic and output model matrices, and w_t and q_t are additive Gaussian uncertainty and noise, defined as $w_t \sim \mathcal{N}(0, W)$ and $q_t \sim \mathcal{N}(0, Q)$. A models kinematic state transition for one time step and C models the mapping from kinematic state to neural output. In practice, we constrain the form of the A and W matrices so that integrated velocity perfectly explains position:

$$A = \begin{pmatrix} 1 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & dt & 0 \\ 0 & 0 & a_{v^{horiz},v^{horiz}} & a_{v^{horiz},v^{vert}} & 0 \\ 0 & 0 & a_{v^{vert},v^{horiz}} & a_{v^{vert},v^{vert}} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_{v^{horiz},v^{horiz}} & w_{v^{horiz},v^{vert}} & 0 \\ 0 & 0 & w_{v^{vert},v^{horiz}} & w_{v^{vert},v^{vert}} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4)$$

After fitting, $a_{v^{vert},v^{horiz}}$ and $a_{v^{horiz},v^{vert}}$ are typically close to 0 and $a_{v^{horiz},v^{horiz}}$ and $a_{v^{vert},v^{vert}}$ are less than 1. The resulting model introduces damped velocity dynamics. Therefore, given no neural measurements, we expect a cursor in motion to smoothly slow down. If we fit the full C matrix, then the neural output model incorporates both position and velocity. If we constrain the position terms to be 0, the model is a velocity kalman filter (Velocity-KF), used online in previous studies (e.g., [11], [12]).

Fig. 2b is a graphical representation of the position/velocity Kalman filter. x_t has been split into two components, p_t for position variables and v_t for velocity variables. As defined by the constraints on the A matrix, position does not have a direct influence on velocity.

A. Innovation 1: Parameter Fitting

Kalman filter parameters found to explain arm kinematics from neural outputs can be for used brain control. The hypothetical plot in Fig. 1c shows the relationship between parameter setting, offline reconstruction quality, and control performance suggested by this perspective. Suppose we were to systematically sweep one of the Kalman filter parameters and measure the filter's effectiveness. For arm kinematic reconstruction quality, this is a measure of correspondence between observed and reconstructed arm movements, which can be fully quantified and understood offline. For BMI control performance, we wish to measure the user's ability to complete task goals during online control. The offline perspective assumes that both applications have the same optimal parameters and so the offline and online measures share the same global maximum (black arrow).

It could be that these two maxima are not necessarily aligned, such as in the hypothetical plot in Fig. 1d. More concretely, a model designed for offline reconstructions may not necessarily translate to a good online controller. Thus, we pursued a different approach and fit model parameters from data collected during online control. One such strategy, which has been previously employed [9], [11], is to regress neural activity against neural cursor kinematics. Another strategy is to randomly seed decoder parameters and to provide assistive control during the training procedure [9]. In this assistive control scheme, the prosthetic output is driven by a mixture of decoder output and task relevant movements, such as precomputed trajectories directly to the target. At each iterative refinement the decoder's contribution is increased, until the prosthesis is fully driven by the

decoder. This scheme works well in practice, especially when easing monkeys into performing the task.

The initial decoder can be seeded with either random or learned parameters. Since prosthetic systems typically aim to record from arm related motor areas, it is possible that the global maximum is close to the parameter set fit by the offline perspective (the black arrow versus the gray arrow, local maxima, in Fig. 1d). Thus, instead of a random seed, the decoder can be seeded with this reasonable choice of parameters. Previous reports have employed this approach by having the prosthetic user observe movements to establish an initial model fit [9], [11], [13]. Then iterative training procedures fit the model with either the kinematics of an observed [11], [13] or controlled [9] cursor. The kinematics of the observed cursor are subject to the same limitations as arm kinematics: since the control algorithm is not in the feedback loop during this initial observation stage, the model fitting procedure is still fundamentally offline. Regressing against the kinematics of the controlled cursor is, therefore, perhaps a step in the right direction, since it regresses against measurements of the neural control signals during online control. However, this approach will tend to carry forward aspects of model misfit acquired during the initial seeding of decoder parameters. As a simple example, consider an initial decoder that rotates the user's desired velocity by 90 degrees. All measured movements of this cursor will retain this bias and when we refit the decoder parameters this bias will remain.

To address the limitations described above, we develop a new method for training BMI parameters. Initially, the BMI system is fit from neural and cursor data, where the cursor moves along with the native arm. Next, the monkey is placed in online brain control mode with this "offline perspective" control algorithm. Training data are collected during brain control and are transformed to estimate the user's intended control command. The details of this transformation are summarized in Fig. 3. The kinematics of the neurally driven cursor at each time-step may not be the best estimate of the user's intentions. The monkey generates intentions by applying knowledge of the task goal, in this case "acquire the green target," to the current state of the cursor. We make the simple assumption that the monkey intends to generate a velocity oriented towards the target at every time-step, since this is the most direct path to the goal and should lead to most rapid trial completion and reward. Thus, for model training purposes only, we rotate the velocity vector of the neural cursor (in red) to orient towards the goal, resulting in a new set of "intention-based" kinematics (in cyan). Additionally, when the cursor is on target, we assume that the user wishes to instruct zero velocity. We believe that this new set of kinematics are a better estimate of the user's intention than the original neural cursor kinematics, effectively cleaning up noisy training data. Importantly, after refitting the model in this way, the resulting decoder can be used with neural data alone and no knowledge of the target or task goal. A similar manipulation to training data was used in a rat study to adapt a one dimensional neural controller over time [8].

B. Innovation 2: Decoding Algorithm

Existing work typically decodes either position (e.g., [1], [4]) or velocity (e.g., [9]). In a comparison of position and velocity decoders, tetraplegic patients demonstrated higher performance control with velocity decoders than with position decoders [11]. However, we find that when position decoding is removed, decoded velocities tend to be less stable. Colloquially put, the cursor appears to get caught in "force fields" resulting in "orbiting" around the target and getting "stuck" in parts of the workspace. This is not surprising, given that firing rates in the recorded brain areas are correlated to cursor position.

One approach is to decode both position and velocity. However, the Kalman filter described in Sec. II (Fig. 2b) with a position and velocity output model describes the relationship between position and velocity in a manner that produces an undesired high frequency jitter in cursor position. Although the kinematic model is physically based, with the cursor effectively modeled as an object moving with damped velocity, these constraints are not preserved when the model is applied online.

At time t we have a previous estimate of the kinematic state, $\hat{x}_{t-1}$ and a new neural output, y_t. Next the filter applies the dynamics model to estimate x_t with all neural outputs up to time $t-1$. This is the *a priori* estimate of x_t:

$$\hat{x}_{t|t-1} \quad = \quad A\hat{x}_{t-1} \tag{5}$$

The model also estimates the *a priori* covariance (or uncertainty) of $\hat{x}_{t|t-1}$:

$$\Sigma_{t|t-1} \quad = \quad A\Sigma_{t-1}A^T + W \tag{6}$$

W is the uncertainty introduced by the trajectory model update. Even if W adds no uncertainty to position, such as in the constrained structure of equation 4, $A\Sigma_{t-1}A^T$ translates previous velocity uncertainty into current position uncertainty. This makes sense: if we do not know the previous velocity with certainty, we do not know the integrated velocity with certainty and so our position estimate may have error. Thus, in practice, there is uncertainty in the *a priori* estimate of every kinematic variable. This uncertainty in position translates to jitter in the decode, as noise in the neural outputs will now filter into position.

We must distinguish online and offline use of the Kalman filter. In the online setting, the user is presented with the *a posteriori* estimate of cursor kinematics at every time-step. If we believe that the user sees and internalizes the presentation of the cursor on the screen at each time-step, then the way in which we model *a posteriori* covariance no longer makes sense, as the user accepts the presented position as the current position state. By presenting the decode to the user, we create a *causal intervention*, that explicitly sets the value of the kinematic variable. This operation is defined by probability theory and is well described by causal calculus [14] (see also [15], [16]).

As a first step to modify the filter to incorporate this feedback, we presume that the user internalizes the filter's estimate of cursor position, $\hat{p}_t$, *with complete certainty* at time t. Accordingly, p_t is explicitly set to $\hat{p}_t$, with no uncertainty. We are assuming that the user knows the previous cursor position via feedback and that his forward model is exact. This is shown graphically in Fig. 2c, where the intervened variable is in green. Note also that the arrows coming into p_t have been removed, to indicate that p_t has been externally set and uncertainty is not propagated.

The result of this intervention is to remove uncertainty in p_t. All parameter fitting methods described in previous sections remain unchanged. To implement this position feedback filter, we only have to alter estimation of the *a posteriori* covariance describing uncertainty in kinematics. Previously, we had:

$$\Sigma_{t|t-1} = A\Sigma_t A^T + W \;\; : \;\; \Sigma_{t|t-1} = \begin{bmatrix} \Sigma^{p,p}_{t|t-1} & \Sigma^{p,v}_{t|t-1} & 0 \\ \Sigma^{v,p}_{t|t-1} & \Sigma^{v,v}_{t|t-1} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7)$$

where each block of the matrix $\Sigma_{t|t-1}$ represents the uncertainty propagated from previous kinematic estimates (position to position, position to velocity, and so on). Each one of these sub-matricies of $\Sigma_{t|t-1}$ is 2 x 2, representing horizontal and vertical components. The bottom row and right column of zeros encodes the fact that the bias or constant offset term of x_t, the last element of the state vector, is known with certainty. Since we have intervened and set p_t with feedback, this matrix becomes:

$$\Sigma_{t|t-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & 0 \\ \mathbf{0} & \Sigma^{v,v}_{t|t-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (8)$$

We are zeroing out all *a priori* position uncertainty, as we are explicitly assuming that the monkey and the control algorithm have matching beliefs about the position of the cursor at time t. Otherwise, this filter is run in the same manner as the standard Kalman filter. This modified Kalman filter, together with innovation 1 described above, comprise the ReFIT-KF control algorithm.

III. Methods

All procedures and experiments were approved by the Stanford University Institutional Animal Care and Use Committee (IACUC). Experiments were conducted with an adult male rhesus macaque (L), implanted with a 96 electrode Utah array (Blackrock Microsystems Inc., Salt Lake City, UT) using standard neurosurgical techniques [17]. Monkey L was implanted 19-33 months prior to the experiments. The electrode array was implanted in the dorsal aspect of premotor cortex (PMd) and primary motor cortex (M1), as estimated visually from local anatomical landmarks.

The monkey was trained to make point-to-point reaches in a 2D plane with a virtual cursor controlled by the contralateral arm or by a neural decoder [18]. The virtual cursor and targets were presented in a 3D environment (MSMS, MDDF, USC, Los Angeles, CA). Visual presentation was provided via two LCD monitors with refresh rates at 120 Hz, yielding frame updates within 12 $\pm$ 4 ms. Two mirrors visually fused the displays into a single 3D percept, creating a Wheatstone stereograph (see Fig. 2 in [18]). Hand position data were measured with an infrared reflective bead tracking system (Polaris, Northern Digital, Ontario, Canada). Behavioral control and neural decode were run on separate PCs using the Simulink/xPC platform (Mathworks, Natick, MA) with communication latencies of 3 ms. This system enabled millisecond-timing precision for all computations. Neural data were initially processed by the Cerebus recording system (Blackrock Microsystems Inc., Salt Lake City, UT). An analog bandpass filter with a 0.3 Hz to 7.5 kHz passband was applied to each channel. Channels were sampled at 30 kSamples/second and were filtered with a 250 Hz to 7.5 kHz digital bandpass filter. A threshold detector was applied to each bandpassed channel. The threshold value was set automatically to -4.5 times the measured root mean squared value of the channel. When the signal value was less than threshold a spike event was registered for that channel and was received by the behavioral control system within 5 $\pm$ 1 ms. The number of spike events are counted in non-overlapping temporal bins (typically 50 ms). The counts for each channel over time are the inputs to the control algorithm.

For all sessions reported in this paper, the monkey acquired targets in a center-out-and-back task in which a uniform ring of 8 targets was 8 cm from the center target. Target acceptance windows were square boxes with sides 4-6 cm in length. Successful acquisition requires a 500 ms hold period, during which the cursor must remain within the acceptance window. Acquisition times do not include this hold period.

IV. Results

Taken together, the innovations described in Sections II-A and II-B increased performance relative to a velocity Kalman filter (Velocity-KF), that is state-of-the-art for current BMIs (e.g., [11], [12]). Fig. 4 shows representative cursor traces for the different neural control modes and native arm based control. Fig. 5 shows the relative contributions to performance made by each innovation; Fig. 5a shows performance with Velocity-KF (green) compared to the Velocity-KF with only the first innovation (yellow) and Fig. 5b shows the Velocity-KF with the first innovation compared against the ReFIT-KF (both innovations).

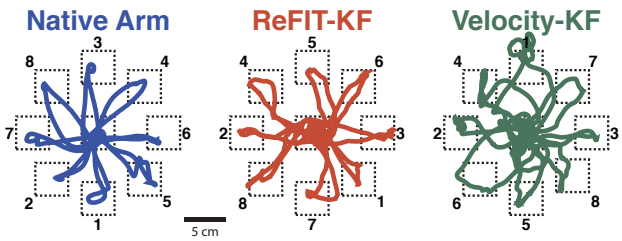

Fig. 4. Continuous center-out-and-back traces for 3 cursor control modes. The order in which the radial targets appeared is indicated by the adjacent numbers.

We tested algorithms in succession, switching between them on the same day against identical task conditions. The acceptance window was 5 cm for Fig. 5a and 4 cm for Fig. 5b and the monkey had 3 s to acquire the target. The acceptance window was larger in Fig. 5a because acquisition of smaller targets with Veloicty-KF control was incredibly difficult and the monkey would lose interest in the task.

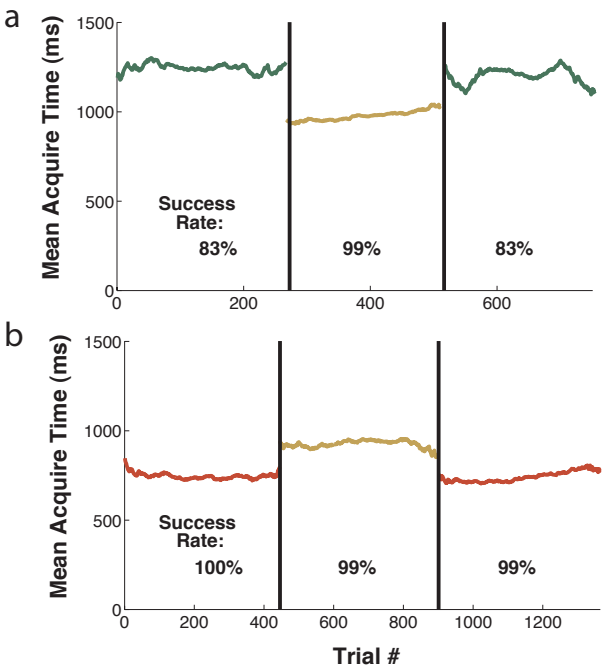

Fig. 5. (a) Velocity-KF (green) vs Kalman filter with innovation 1 only (yellow) (L-2010-01-13) (b) Kalman filter with innovation 1 only (yellow) vs ReFIT Kalman filter (red) (L-2010-08-19)

Table 1 shows average acquisition times for the three control modes during four different sessions. During all of these sessions, the task parameters were the same as those specified above, except the acceptance box was expanded to 6 cm to allow for high success rates across all three control modes (>95%) permitting direct comparison of acquisition times. Note that ReFIT-KF reduces acquisition times by a factor of 2.

Dataset	Native Arm	**ReFIT-KF**	Velocity-KF
2010-10-27	487	581	1736
2010-10-28	441	650	1653
2010-10-29	453	605	1329
2010-11-02	496	549	1089
Mean	469	596	1452

TABLE I
AVERAGE ACQUISITION TIMES FOR 3 CURSOR CONTROL MODES.

V. CONCLUSION

We present two control algorithm innovations developed from a feedback control perspective. Each of these innovations results in an increase in BMI control performance, as measured by success rate and/or acquisition time. As demonstrated across multiple datasets, the ReFIT-KF control algorithm results in a large reduction in acquisition time relative to Velocity-KF control, approaching native arm control performance for the tested cursor control task.

ACKNOWLEDGEMENT

We thank M. Risch & S. Kang for surgical assistance & veterinary care, D. Haven for IT support, S. Eisensee for administrative assistance, Mark M. Churchland & Matthew T. Kaufman for insightful comments & suggestions, and P. Ortega for mathematical insight.

REFERENCES

[1] 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, pp. 141–142, Mar 2002.

[2] 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 Biol*, vol. 1, Nov 2003.

[3] 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 Comput*, vol. 18, pp. 80–118, Jan 2006.

[4] 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, pp. 164–171, Jul 2006.

[5] G. H. Mulliken, S. Musallam, and R. A. Andersen, "Decoding trajectories from posterior parietal cortex ensembles," *J Neurosci*, vol. 28, pp. 12913–12926, Nov 2008.

[6] K. Ganguly and J. M. Carmena, "Emergence of a stable cortical map for neuroprosthetic control," *PLoS Biol*, vol. 7, Jul 2009.

[7] D. M. Taylor, S. I. Tillery, and A. B. Schwartz, "Direct cortical control of 3d neuroprosthetic devices," *Science*, vol. 296, pp. 1829–1832, Jun 2002.

[8] G. J. Gage, K. A. Ludwig, K. J. Otto, E. L. Ionides, and D. R. Kipke, "Naive coadaptive cortical control," *Journal of Neural Engineering*, vol. 2, no. 2, p. 52, 2005.

[9] 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, pp. 1098–1101, Jun 2008.

[10] G. W. Fraser, S. M. Chase, A. Whitford, and A. B. Schwartz, "Control of a brain-computer interface without spike sorting," *J Neural Eng*, vol. 6, pp. 055004–055004, Oct 2009.

[11] 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," *J Neural Eng*, vol. 5, pp. 455–476, Dec 2008.

[12] S. P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, G. M. Friehs, and M. J. Black, "Point-and-click cursor control with an intracortical neural interface system in humans with tetraplegia," *IEEE Trans Neural Syst Rehabil Eng*, Jan 2011.

[13] A. J. Suminski, D. C. Tkach, A. H. Fagg, and N. G. Hatsopoulos, "Incorporating feedback from multiple sensory modalities enhances brain-machine interface control," *J Neurosci*, vol. 30, pp. 16777–16787, Dec 2010.

[14] J. Pearl, *Causality*. Cambridge, UK: Cambridge University Press, 2000.

[15] P. A. Ortega and D. A. Braun, "A bayesian rule for adaptive control based on causal interventions," *Proceedings of the Third Conference on Artificial General Intelligence*, pp. 121–126, 2010.

[16] D. Koller and N. Friedman, *Probabilistic Graphical Models*. Cambridge, UK: MIT Press, 2009.

[17] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain-computer interface," *Nature*, vol. 442, pp. 195–198, Jul 2006.

[18] J. P. Cunningham, P. Nuyujukian, V. Gilja, C. A. Chestek, S. I. Ryu, and K. V. Shenoy, "A closed-loop human simulator for investigating the role of feedback-control in brain-machine interfaces," *J Neurophysiol*, vol. 105, pp. 1932–1949, Oct 2011.