Auto-Deleting Brain Machine Interface: Error Detection Using Spiking Neural Activity in The Motor Cortex

Nir Even-Chen*, *IEEE Student Member*, Sergey D. Stavisky*, *IEEE Student Member*,
Jonathan C. Kao, *IEEE Student Member,* Stephen I. Ryu,
and Krishna V. Shenoy, *IEEE Senior Member*

***Abstract*— Brain machine interfaces (BMIs) aim to assist people with paralysis by increasing their independence and ability to communicate, e.g., by using a cursor-based virtual keyboard. Current BMI clinical trials are hampered by modest performance that causes selection of wrong characters (errors) and thus reduces achieved typing rate. If it were possible to detect these errors without explicit knowledge of the task goal, this could be used to automatically "undo" wrong selections or even prevent upcoming wrong selections. We decoded imminent or recent errors during closed-loop BMI control from intracortical spiking neural activity. In our experiment, a non-human primate controlled a neurally-driven BMI cursor to acquire targets on a grid, which simulates a virtual keyboard. In offline analyses of this closed-loop BMI control data, we identified motor cortical neural signals indicative of task error occurrence. We were able to detect task outcomes (97% accuracy) and even predict upcoming task outcomes (86% accuracy) using neural activity alone. This novel strategy may help increase the performance and clinical viability of BMIs.**

I. Introduction

Despite efforts invested in improving the quality of life of people with tetraplegia (e.g., ALS patients), there is much room for improvement over existing assistive technologies to provide patients with more independence. A major need is the ability to communicate, e.g., by using a cursor-based virtual keyboard [1]. Current devices such as eye trackers are not comfortable to use since they demand full attention and restrict the ability to freely look around while typing. Brain machine interfaces (BMIs) aim to provide a different and perhaps more easy-to-use communication channel by recording motor-intention related neural signals from the brain and converting them to control assistive devices [1]–[3]. To date, however, such BMIs have inconsistent performance and are prone to errors that result in the wrong character being typed. These mistakes then need to be deleted, which slows typing rate. In this work we consider the feasibility of detecting and correcting these errors by using the neural signals recorded during BMI control.

Current intracortical BMI decoders only look for neural correlates of movement intention. However, the BMI user has constant visual feedback about the ongoing task (Fig. 1, green path) and one can assume that somewhere in the brain there is information about recent task outcome. Non-invasive methods (e.g., EEG) have identified robust error signals during a variety of tasks [4], but it is unknown whether these signals are present in the motor cortical regions used for BMI control (though see [5], [6] for a discussion of BMIs driven by activity from other areas, and [7] for neural correlates of BMI kinematic errors in motor cortex). If these signals are present, and if they can be accurately decoded, intracortical error detection could be used during closed-loop BMI control to prevent incorrectly decoded actions, such as when a patient's BMI incorrectly types a letter that s/he did not intend. Such a system would detect the person's recognition of this error and would automatically delete the selection (Fig.1, red path).

To evaluate the feasibility of this approach, we conducted an experiment using a non-human primate that simulates a cursor-based virtual keyboard (Fig.1). We recorded neural signals as the monkey performed the BMI-controlled typing-like task and examined the neural data to investigate if we could detect the occurrence of wrong target selection solely from the neural data.

II. Methods

A. Behavioral Tasks

All procedures and experiments were approved by the Stanford University Institutional Animal Care and Use Committee. A rhesus macaque (monkey J) was trained to perform point-to-point virtual cursor movements in a 2D plane using either hand movements or BMI control. He was free to move his arm even during BMI control [8]–[10]. A keyboard-like task was modeled after the task described in [11] . The goal of the task (depicted in Fig. 1) was to acquire a cued target among 64 possible targets. The target was cued by changing its color from yellow to green. Only one target was randomly prompted on a given trial.

The workspace was 40×32 cm and had in its center a 24×24 cm grid uniformly divided into 8×8 contiguous, non-overlapping square target acquisition regions. Each target acquisition area contained, at its center, a circular visual

* Denotes equal contribution. Research supported by Stanford Electrical Engineering Departmental Fellowship (NEC), National Science Foundation Graduate Research Fellowship (SDS, JCK), NSF IGERT (SDS), NIH Pioneer Award 8DP1HD075623, NIH T-RO1 award NS076460, and DARPA REPAIR award N66001-10-C-2010 (KVS).

N. Even-Chen (nirec@stanford.edu) and J. C. Kao (jcykao@stanford.edu) are with the Department of Electrical Engineering at Stanford University, Stanford, CA 94305 USA.

S. D. Stavisky (sergey.stavisky@stanford.edu) is with the Neurosciences Graduate Program, Stanford University.

K. V. Shenoy (shenoy@stanford.edu) is with the Departments of Electrical Engineering, Bioengineering, and Neurobiology; the Neurosciences Graduate Program and the Bio-X Program; and the Stanford Neurosciences Institute, Stanford University.

S. I. Ryu (seoulman@stanford.edu) is with the Department of Neurosurgery at Palo Alto Medical Foundation, Palo Alto, CA, USA, and the Department of Electrical Engineering, Stanford University.

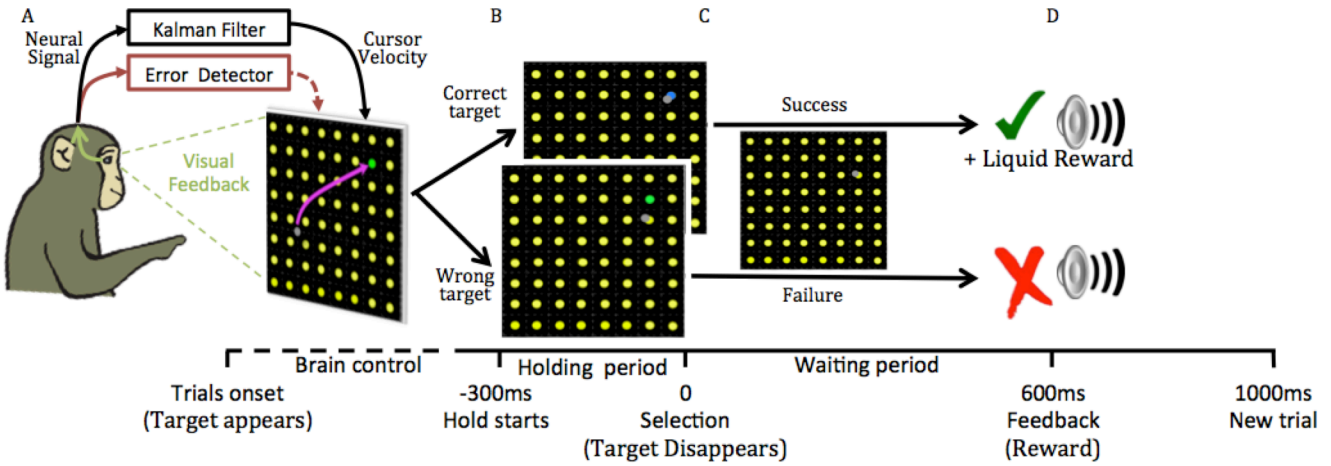

Fig. 1. Experiment layout and task timeline. (A) The monkey controls the cursor using a velocity Kalman filter decoder (black path). His goal is to move the cursor (gray dot) to the cued green target among 64 yellow potential targets. The monkey observes the screen and is putatively aware of his selection and the occurrence of task errors (green path). In offline analyses, we built an error decoder (red path) that in future work could be integrated for closed-loop error correction (dashed red line). (B) Holding the cursor over a target for 300 ms selects it; when the correct target is being held its color changes to blue. (C) After selection, the cued target disappears and the monkey waits 600 ms for auditory feedback. (D) After successful trials the monkey is rewarded. The monkey is not rewarded for failed trials. A new trial is started after an additional 400 ms.

representation of the target. Within the grid, the cursor was always in the acquisition region of a possible target. A target was selected by holding the cursor within its acquisition region for 300 ms. While the correct target was being held, its color changed to blue, whereas an incorrect target did not change color while the cursor was in its acquisition region. Once a target was selected, the color of the cued target reverted back to yellow signifying that there was no longer an active, cued target. Feedback for successful trials consisted of an auditory tone and a liquid reward 600 ms after selection. Incorrect target selection (failure) led to a different tone and no liquid reward, also 600 ms after selection. The next trial's target appeared after additional 400 ms. If no target was selected within the trial time limit of 5 s, the trial ended in failure.

We delayed the task outcome feedback in order to separate the putative error signal reflecting the monkey's recognition of his error from the neural signal related to receiving the explicit task feedback. Thus, the only overt immediate difference between successful and failed trials was the color of the cued target (blue if correctly selected and green if not selected) during the holding period. It is unlikely that this color difference causes a neural difference in the motor cortex, but we plan to address this potential confound in future work.

B. Neural Recording and Signal Processing

Monkey J was implanted with two 96 electrode Utah arrays (Blackrock Microsystems, Inc.) using standard neurosurgical techniques [12] 61-62 months prior to this study. The arrays were implanted into the left hemisphere of the brain; one into primary motor cortex (M1) and the other into dorsal premotor cortex (PMd), as estimated visually from local anatomical landmarks. Voltage signals from each of the 192 electrodes were band-pass filtered from 250 to 7500 Hz. A spike was then detected whenever the voltage crossed below a threshold set at the beginning of each day (at -4.5 × rms voltage). Contralateral hand position (for decoder training) was measured with an infrared reflective bead tracking system (Polaris, Northern Digital) polling at 60 Hz.

C. BMI Cursor Control

At the start of each experiment we collected a training dataset of 500 arm-controlled trials of a planar random target reaching task according to the protocol described in [12]. This data were used to train a Feedback Intention Trained Kalman filter (FIT-KF) decoder [12], which operates at every time step on the observed firing rate vector $y_t \in \mathbb{R}^N$ ($N = 192$ electrodes). The decoder outputs a velocity command every 25 ms from input consisting of binned spikes counts from the previous 25 ms. Briefly, FIT-KF is a streamlined version of the ReFIT-KF [10] and improves upon a standard KF by adjusting kinematics of the training data to better match the subject's presumed movement intention. Note that for this experiment we did not zero training set velocities during the hold epoch, as was done in [12].

The velocity Kalman filter (VKF) converges quickly to a steady state: $v_t = M_1 v_{t-1} + M_2 y_t$, where $v_t \in \mathbb{R}^2$ is the velocity of the cursor [13]. We call the first term $(M_1 v_{t-1})$ the momentum and second term $(M_2 y_t)$ the neural push.

D. Dimensionality reduction via principal component analysis (PCA)

For all offline analyses, multiunit threshold crossing spike counts from each electrode were binned every 10 ms ($y_k \in \mathbb{R}^N$) and each trial ($Y_i \in \mathbb{R}^{N \times K}$, where K is the number of time sample in the trial) was aligned to target selection time ($t = 0$, Fig. 1).

The recorded neural activity is composed from several components including kinematics (v_t^{kin}), trials outcome (v_t^{OC}) and noise (v_t^{noise}). A simplified model might be: $y_t = v_t^{kin} + v_t^{OC} + v_t^{noise}$. We wanted to direct our dimensionality reduction of the neural data to focus on the variance between outcomes (successes and failures) rather than overall variance (which includes kinematics and noise). Thus, we performed PCA on the difference in neural activity between the outcome-averaged successful and failed trials ($\Delta y_k^{\mu} = \frac{1}{N_S}\sum_{i \in Suc} y_k^i - \frac{1}{N_F}\sum_{i \in Fail} y_k^i, \Delta y_k^{\mu} \in \mathbb{R}^N$).
To eliminate day-to-day baseline firing rate variance when

combining data from multiple days, we subtracted from each channel's instantaneous firing rate its average rate (for that day) before performing PCA.

E. Classification via support vector machine

The data were composed of trials $Z_i \in \mathbb{R}^{D\times M}$ (which is a submatrix and/or linear transformation of Y_i), where the D is either the number of channels (N) or number of principal components, and M is number of time samples in the chosen time window. Each trial was labeled as one of two outcomes: a successful trial where the monkey selected the correct target, or a failed trial where the monkey selected an incorrect target.

To classify the trials we used a 10-fold cross-validated linear support vector machine (SVM). Given a set of examples $Z_i \in \mathbb{R}^{D\times M}$ (each labeled as successful or failed) the SVM training algorithm builds a model that can be then used to assign new examples into one of these two categories. The classification result (detection of failed trials) yields three measures of the classifier performance:

- Miss rate (M) – ratio of the number of failed trials that were classified mistakenly as successful (#Miss) to the total number of failed trials ($\#\ Failed$). $\mathrm{M} = \frac{\#\mathrm{Miss}}{\#\ Failed}$
- False alarm rate (F) – ratio of the number of successful trials that were classified mistakenly as failed ($\#FA$) to the total successful trials ($\#\ Successes$). $F = \frac{\#FA}{\#\ Successes}$
- Misclassification error rate (Er) – the ratio of wrongly classified trials to the total number of trials

  $Er = \frac{\#\ Failed}{\#\ all\ trials}\cdot M + \frac{\#\ Successes}{\#\ all\ trials}\cdot F.$

  Note that M and F are independent of the ratio between the task outcomes (success rate of the monkey), while Er is dependent on this ratio.

III. Results

A. Behavioral data

We conducted two days of experiments; on the first day the monkey performed 1,576 trials with 77% success rate; on the second, 1,141 trials with 76% success rate. We intentionally calibrated the task difficulty to achieve a 75%-80% success rate so as to have as many failure trials as possible without frustrating the monkey. In previous hand controlled experiments the monkey performed the task with near perfect success rate, indicating that errors in the present data were likely due to the imperfect BMI cursor control.

Fewer than 3% of trials exceeded the 5 s maximum trial limit; these were not analyzed. In 98% of failure trials, the target that was (mistakenly) selected was one of those adjacent to the cued target; this suggests that most mistakes were due to an inability to precisely hold the cursor over the cued target.

B. Error detection from neural data

First we classified trial outcome (success or failure) using neural activity spanning a wide time window from 200 ms before the target was successfully held (-500 ms) until feedback appearance (600 ms). This yielded a misclassification rate of 4 ± 2 % (mean ± std over cross-validation folds). As a control analysis asking whether the classifier was using an aspect of the data that is not related simply to the kinematics (which could potentially distinguish between successful and failed trials), we compared this "full" neural activity decoder to ones working with the cursor position, velocity and neural push (immediate neural activity commanded, see Methods) in different coordinate systems. The classification accuracy using the full neural information was over 7 times better than using kinematics, as shown in Table 1.

All Neural	Kinematic Controls	Position	Velocity	Neural Push
4 ± 2%	Cartesian	36 ± 6%	37 ± 7%	39 ± 4%
	Polar	32 ± 4%	29 ± 4 %	35 ± 5 %

Table 1: Misclassification rate comparison of several different error classifiers, which operate on either the full neural data or kinematics.

Classification accuracy as a function of trial time

To better understand when neural activity can be used to classify between the two trial outcomes, we analyzed the evolution of Er, F and M rates over time. First, we aligned to the time of target selection and classified the neural activity in sliding 100 ms non-overlapping windows ($Z_i \in \mathbb{R}^{192\times 10}$). Fig. 2A shows that the misclassification rate decreases throughout the target hold time and is minimized approximately 100 ms after target selection. This pattern is similar for F and M rates, and suggests that the putative error signal carrying the monkey's recognition of whether he is about to (or did) select the correct target is strongest immediately after a target is selected.

We also classified the trials using an accumulating epoch (growing window) in 100 ms steps ($Z_i \in \mathbb{R}^{192\times 10\cdot j}$, where j is increasing). This formulation of the problem most closely matches how data will become available to an error decoder during closed-loop error detection/correction. In Fig. 2B we see that the misclassification rate continues to decrease after selection and converges at 4 ± 2% approximately 500 ms after target selection. Similarly, M and F rates converge to 13 ± 6% and 0.7 ± 0.6%, respectively.

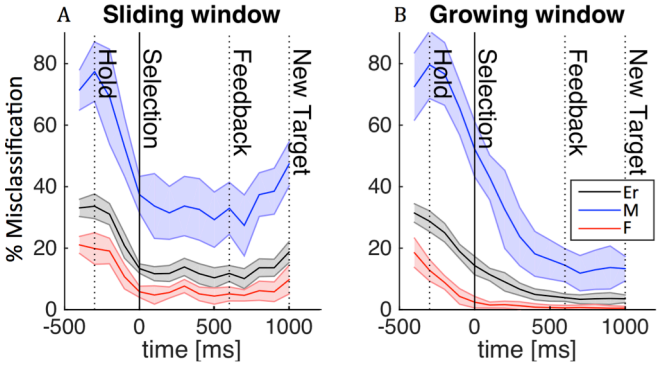

Fig. 2. Classification accuracy over time. (A) Classification of a sliding 100 ms time window $Z_i \in \mathbb{R}^{192\times 10}$. (B) Classification of a growing time window $Z_i \in R^{192\times 10\cdot j}$. The abscissa coordinate corresponds to the end of the window. Shading denotes standard deviation. For context: a naive classifier, which classifies all trials as successful ones, will have more than 20% misclassification rate (equal to the failure selection rate). Our decoder initially performs worse than chance (due to overfitting when minimal useful information is available) but as the trial time proceeds, rapidly performs significantly better than chance after t = -100 ms ($p < 10^{-3}$).

	Er	M	F
Before Selection	14 ± 2%	21 ± 5%	12 ± 2%
Before Feedback	3 ± 1%	6 ± 3%	2 ± 1 %

Table 2: Error, miss, and false alarm rate of classification using the activity of the top 5 principal components in the entire epoch specified.

Dimensionality reduction improves error decoding

There is high redundancy in neural recordings during reaching tasks; the underlying dimensionality of the neural computations is believed to be lower than the number of recorded neurons [14]. This led us to try projecting the measurements into a lower dimensional space in the hopes of improving classifier performance and reducing over-fitting.

We optimized the number of PCs used for decoding by computing misclassification rate in two time periods as a function of the neural dimensionality. These time periods both began 200 ms before hold target starts (-500 ms); one time period ended before selection (-10 ms, $Z_i \in \mathbb{R}^{5\times 49}$), while the other one ended before task outcome feedback (590 ms, $Z_i \in \mathbb{R}^{5\times 109}$). We found that the decoder performed best using 5 PCs in both datasets; Table 2 shows that using this parameter, we could correctly detect trial outcome with 97% accuracy before the monkey was given trial outcome feedback, and could even predict trial outcome with 86% accuracy.

C. Neural signals underling error detection

To better understand the classification results and to try to begin to understand their neurophysiological source, we computed the population PSTH of the difference in neural activity between task outcomes. Fig. 3A shows that across both arrays, but especially in PMd (channels 97 to 192), firing rates are higher immediately following correct target selection; later (around 300 ms after selection) this pattern reverses: the average firing rates across the PMd channels through the 300 ms after selection for success and fail trials were 18 ± 3 Hz and 15 ± 5 Hz, respectively; in the subsequent 300 ms (from 300 ms to 600 ms after selection) they were 18 ± 4 Hz and 24 ± 9 Hz, respectively. In both epochs, these means were significantly different ($p < 10^{-3}$, *t*-test without assuming equal variances).

We also observed that during the epochs when activity

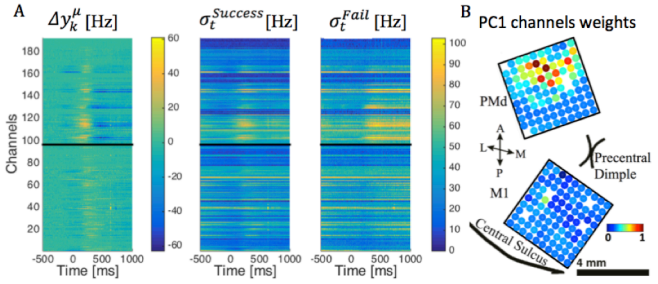

Fig. 3. Neural correlates of successful versus failed target selection. (A) The population PSTH of the difference between task outcomes (Δy_k^{μ}) and the standard deviation of firing rates for successful ($\sigma_t^{Success}$) or failure (σ_t^{Fail}) trials. The black line separates the channels by arrays: M1 (1-96) and PMd (97-192) (B) Leading PC channel weights are shown on the multi-electrode array.

corresponding to the two task outcomes differed most, the neural variance was higher for failure trials. Using a bootstrap test (1000 resamples), the average standard deviation of the firing rates across the PMd channels through the 600 ms after selection for success and fail trials were 4.5 ± 0.2 Hz and 7.5 ± 0.3 Hz, respectively ($p < 10^{-3}$).

In order to better visualize the sources of the neural activity that enabled us to distinguish between the two trial outcomes, we plotted the channel weights of the leading PC ($u_1 \in \mathbb{R}^{192}$) during the Waiting Period. The leading PC captures 75% of the variance of the difference in recorded neural activity between trial outcomes. Fig. 3B shows that most of this process is concentrated on the PMd array, whereas M1 channels do not appear to contribute as much to this process.

IV. Discussion

We were pleasantly surprised to find that we were able to decode task errors (i.e., whether a trial was successful versus failed) from neural activity in the motor cortex with high accuracy. We were further encouraged to find that the neural signals enabling this decoding were different from those related directly to the kinematics. There are theoretical reasons to expect motor error signals to reach motor cortex in order to enable motor learning (see e.g., [15]); it is thus possible that the neural signals underlying this decode performance relate to these error signals. We observed that our decoder preferentially used neural activity from PMd as opposed to M1; this is consistent with a greater role of PMd in higher-level aspects of motor control [16], possibly including processing errors. We believe that an exciting avenue for future work is to better characterize the source and nature of the discriminated signal.

A more immediate application of these results is to use the error signal to detect and correct mistakes during BMI use. Although other studies have begun to consider and exploit the closed-loop nature of BMI control [10], [17]–[19], to the best of our knowledge this is the first time that a BMI user's recognition of task outcome has been decoded directly from the same arrays that enable cursor control. This study provides a proof of concept that it may be possible to use such a decoder during closed-loop BMI use (Fig. 1A, dashed red line) to improve performance. When the user's BMI mistakenly selects the wrong target, this parallel decoder would detect the error and would undo the incorrect selection ('auto delete'). Furthermore, if the error decoder predicts, during target hold, that the upcoming selection is incorrect, the hold timer can be reset to zero to prevent incorrect selection. The classification's computational time is negligible. Thus, in a real-time implementation, the auto-delete lag will be determined by the amount of information we choose to accumulate for classification, which dictates an accuracy versus latency tradeoff, as shown in Fig.2.

We note that in these data, the detection and prediction rates for successful trials were more accurate than for failure trials (i.e., $M \gg F$, even when the number of trials in each condition was equal). This may be due to the higher variance that characterized failure trials (Fig. 3). We expect that most users would prefer to keep false alarm rate low (even at the

cost of increased miss rate), and we note that changing the decision boundary of the SVM classifier provides an easy knob by which to adjust this balance. Once properly adjusted for user comfort and maximizing communication throughput [11], this BMI auto-delete feature could substantially increase typing rate and thus the clinical viability of BMIs.

ACKNOWLEDGMENT

We thank M. Mazariegos, M. Wechsler, L. Yates, & S. Smith for expert surgical assistance & veterinary care, and B. Davis, & E. Casteneda for administrative assistance.

REFERENCES

[1] D. Bacher, B. Jarosiewicz, N. Y. Masse, S. D. Stavisky, J. D. Simeral, K. Newell, E. M. Oakley, S. S. Cash, G. Friehs, and L. R. Hochberg, "Neural Point-and-Click Communication by a Person With Incomplete Locked-In Syndrome.," *Neurorehabil. Neural Repair*, Nov. 2014.

[2] M. L. Homer, A. V Nurmikko, J. P. Donoghue, and L. R. Hochberg, "Sensors and decoding for intracortical brain computer interfaces.," *Annu. Rev. Biomed. Eng.*, vol. 15, pp. 383–405, Jan. 2013.

[3] J. C. Kao, S. D. Stavisky, D. Sussillo, P. Nuyujukian, and K. V. Shenoy, "Information Systems Opportunities in Brain–Machine Interface Decoders," *Proc. IEEE*, vol. 102, no. 5, pp. 666–682, May 2014.

[4] R. Chavarriaga, A. Sobolewski, and J. D. R. Millán, "Errare machinale est: the use of error-related potentials in brain-machine interfaces.," *Front. Neurosci.*, vol. 8, p. 208, Jan. 2014.

[5] G. H. Mulliken, S. Musallam, and R. a Andersen, "Decoding trajectories from posterior parietal cortex ensembles.," *J. Neurosci.*, vol. 28, no. 48, pp. 12913–26, Nov. 2008.

[6] R. a Andersen, S. Musallam, and B. Pesaran, "Selecting the signals for a brain-machine interface," *Curr. Opin. Neurobiol.*, vol. 14, no. 6, pp. 720–6, Dec. 2004.

[7] S. D. Stavisky, J. C. Kao, S. M. Jordan, S. I. Ryu, and K. V Shenoy, "System Identification of Brain-Machine Interface Control Using a Cursor Jump Perturbation," in *7th Annual International IEEE EMBS Conference on Neural Engineering*, 2015.

[8] P. Nuyujukian, J. M. Fan, V. Gilja, P. S. Kalanithi, C. A. Chestek, and K. V Shenoy, "Monkey models for brain-machine interfaces: The need for maintaining diversity," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2011, pp. 1301–1305.

[9] P. Nuyujukian, J. M. Fan, J. C. Kao, S. I. Ryu, K. V. Shenoy, S. S. Member, S. I. Ryu, K. V. Shenoy, and S. S. Member, "A High-Performance Keyboard Neural Prosthesis Enabled by Task Optimization," *IEEE Trans. Biomed. Eng.*, vol. 62, no. c, pp. 21–29, 2015.

[10] 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 K. V Shenoy, "A high-performance neural prosthesis enabled by control algorithm design.," *Nat. Neurosci.*, vol. 15, no. 12, pp. 1752–7, 2012.

[11] P. Nuyujukian, J. M. Fan, J. C. Kao, S. Member, S. I. Ryu, K. V Shenoy, and S. Member, "A high-performance keyboard neural prosthesis enabled by task optimization," *IEEE Trans. Biomed. Eng.*, pp. 1–9, 2014.

[12] J. M. Fan, P. Nuyujukian, J. C. Kao, C. A. Chestek, S. I. Ryu, and K. V Shenoy, "Intention estimation in brain–machine interfaces," *J. Neural Eng.*, vol. 11, no. 1, p. 016004, 2014.

[13] W. Q. Malik, W. Truccolo, E. N. Brown, and L. R. Hochberg, "Efficient decoding with steady-state Kalman filter in neural interface systems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 1, pp. 25–34, Feb. 2011.

[14] K. V Shenoy, M. Sahani, and M. M. Churchland, "Cortical control of arm movements: a dynamical systems perspective.," *Annu. Rev. Neurosci.*, vol. 36, pp. 337–359, Jul. 2013.

[15] R. Shadmehr, M. A. Smith, and J. W. Krakauer, "Error correction, sensory prediction, and adaptation in motor control.," *Annu. Rev. Neurosci.*, vol. 33, pp. 89–108, Jan. 2010.

[16] S. P. Wise, D. Boussaoud, P. B. Johnson, and R. Caminiti, "Premotor and parietal cortex: corticocortical connectivity and combinatorial computations.," *Annu. Rev. Neurosci.*, vol. 20, pp. 25–42, 1997.

[17] M. D. Golub, B. M. Yu, and S. M. Chase, "Internal models engaged by brain-computer interface control," in *34th Annual International Conference of the IEEE EMBS*, 2012, vol. 2012, pp. 1327–30.

[18] A. L. Orsborn, H. G. Moorman, S. A. Overduin, M. M. Shanechi, D. F. Dimitrov, and J. M. Carmena, "Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control," *Neuron*, vol. 82, no. 6, pp. 1380–1393, Jun. 2014.

[19] F. R. Willett, A. J. Suminski, A. H. Fagg, and N. G. Hatsopoulos, "Improving brain-machine interface performance by decoding intended future movements," *J. Neural Eng.*, vol. 10, no. 2, p. 026011, Apr. 2013.