Published in final edited form as:

Nat Neurosci. 2012 December ; 15(12): 1752–1757. doi:10.1038/nn.3265.

A High-Performance Neural Prosthesis Enabled by Control Algorithm Design

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], **Mark M. Churchland**[5,6], **Matthew T. Kaufman**[6], **Jonathan C. Kao**[5], **Stephen I. Ryu**[5,11], and **Krishna V. Shenoy**[2,3,5,6,7]

[1]Department of Computer Science, Stanford University, Stanford, CA

[2]Stanford Institute for Neuro-Innovation and Translational Neuroscience, Stanford University, Stanford, CA

[3]Department of Bioengineering, Stanford University, Stanford, CA

[4]School of Medicine, Stanford University, Stanford, CA

[5]Department of Electrical Engineering, Stanford University, Stanford, CA

[6]Neurosciences Program, Stanford University, Stanford, CA

[7]Department of Neurobiology, Stanford University, Stanford, CA

[8]Department of Engineering, University of Cambridge, Cambridge, United Kingdom

[9]Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom

[10]Department of Electrical and Computer Engineering and Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA

[11]Department of Neurosurgery, Palo Alto Medical Foundation, Palo Alto, CA

Abstract

Neural prostheses translate neural activity from the brain into control signals for guiding prosthetic devices, such as computer cursors and robotic limbs, and thus offer disabled patients greater interaction with the world. However, relatively low performance remains a critical barrier to successful clinical translation; current neural prostheses are considerably slower with less accurate control than the native arm. Here we present a new control algorithm, the recalibrated feedback

Correspondence and requests for materials should be addressed to V.G. (gilja@stanford.edu).

[*]These authors contributed equally to this work

Competing Interests

The authors declare that they have no competing financial interests.

Author contributions

V.G. and P.N. contributed equally to this work. V.G. and P.N. were responsible for infrastructure development, animal training, data collection and analysis. V.G. was responsible for algorithm design, supplementary modeling and writing of the paper. P.N. and B.M.Y. participated in algorithm design. C.A.C participated in animal training, infrastructure development, and data analysis. J.P.C. participated in algorithm design and infrastructure development. J.M.F. participated in data collection and analysis. M.M.C. and M.T.K. provided initial animal training for the obstacle avoidance task. J.C.K participated in data collection. S.I.R. was responsible for surgical implantation. K.V.S. was involved in all aspects of the study.

intention-trained Kalman filter (ReFIT-KF), that incorporates assumptions about the nature of closed loop neural prosthetic control. When tested with rhesus monkeys implanted with motor cortical electrode arrays, the ReFIT-KF algorithm outperforms existing neural prostheses in all measured domains and halves acquisition time. This control algorithm permits sustained uninterrupted use for hours and generalizes to more challenging tasks without retraining. Using this algorithm, we demonstrate repeatable high performance for years after implantation across two monkeys, thereby increasing the clinical viability of neural prostheses.

Introduction

Neural prostheses have recently demonstrated considerable promise through proof-of-concept animal experiments[1–9] and in human clinical trials[10–13] for partially restoring motor output in paralyzed individuals. Studies in this field primarily focus on adapting insights and methods from the basic neuroscience of cortical motor control to this engineering context. A critical example of this is the use of motor cortex tuning models, which describe the relationship between single unit firing rates and arm movement kinematics, to define a mapping for neural control of a computer cursor in closed loop (e.g.[1–3]). When such a neural prosthesis is introduced to a monkey, performance can increase across days through learning[3]. In addition to controlling computer cursors, these systems have successfully driven robotic end effectors[4]. Neural prosthesis studies have incorporated additional concepts from motor neuroscience, demonstrating the potential to augment system performance by modeling neural activity related to movement preparation[5] and proprioceptive feedback[8]. Recent work also suggests that when the recorded neural population and control algorithm are held constant, neural prosthetic performance increases over time as a stable neural output map is formed and multiple mappings, once learned, can be retained and retrieved across different control contexts[7]. Despite these new insights and additional algorithmic advances (e.g.[12]), system performance on simple cursor control tasks remains low relative to native arm control performance, presenting a critical barrier to clinical translation[14].

In an effort to improve the performance of neural prostheses, we chose to focus on a systems engineering approach. Building on existing methods in the field, we developed two key innovations that alter the modeling assumptions made by these algorithms and the methods by which these algorithms are trained. Additionally, signal conditioning methods, which transform recorded neural signals into control algorithm input, were chosen in an effort to improve system stability and performance[15,16]. As demonstrated in closed loop neural control experiments, these methods result in high performance across multiple cursor control tasks.

Results

Performance Overview

We trained monkeys to acquire targets with a cursor controlled by either native arm movement or neural activity. We developed a novel algorithm, ReFIT-KF, that led to substantially higher-performance neural prosthetic control. Figure 1a shows representative continuous, uninterrupted cursor movements for three different modalities: native arm control, ReFIT-KF, and a velocity Kalman filter (Velocity-KF), that is state-of-the-art for current neural prostheses (e.g.,[11–13]). Monkeys were required to move the computer cursor to a visual target and hold the cursor within a demand box for 500 ms to successfully complete a trial and receive a liquid reward. Targets alternated between central and eight peripheral locations. During online neural control sessions, the contralateral arm was not restrained, and movement continued. However, the physical movement was not stereotyped and would often attenuate or even stop during some neural control sessions, while retaining performance. In a set of additional control experiments, both arms were restrained, and little or no arm movement was observed with similar neural control performance (see Table 1 and Fig. 2).

The ReFIT-KF algorithm outperformed the Velocity-KF by several measures. First, cursor movements with ReFIT-KF control were straighter (Fig. 1a,b and Supplementary Fig. 1), producing less movement away from a straight line to the target. Cursor movements produced using the ReFIT-KF were qualitatively similar to native arm movements (Fig. 1a, Supplementary Figs. 1–2, and Supplementary Videos 1–6). Second, these movements were also completed faster. ReFIT-KF cursor control performance, as measured by the time to successfully acquire the target (Fig. 1c,d), was 75–85% of native arm control performance and at least twice Velocity-KF performance (Supplementary Modeling). In addition to lower mean time to target, the variance was substantially smaller, which is important as this signifies greater movement consistency and fewer potentially frustrating long trials. Critical to achieving this lower time to target, ReFIT-KF controlled cursor movements stopped better. The ability to stop is a critical differentiator between the three control modes. The Velocity-KF controlled cursor took only modestly longer to first acquire the target, but often significantly overshot the target, requiring additional time and multiple passes to stably acquire and hold the target. This overshoot-correction time dominates the overall time to successful target acquisition for Velocity-KF control (Fig. 1c,d) and is captured by “dial-in time” (Fig. 1e,f): the average time required to make the final target acquisition after having first reached the target. Both native arm and ReFIT-KF control allowed more precise stopping as compared to Velocity-KF (Fig. 1e,f). Across all trials in eight experimental sessions with two monkeys, when given five seconds to acquire targets, ReFIT-KF achieved a success rate of >99%, while Velocity-KF had a success rate of 95%. The task difficulty was chosen to achieve a high success rate for all three control modalities on the first experimental day (Supplementary Table 1). When the task difficulty is increased, the success rate with Velocity-KF can drop relative to the success rate with ReFIT-KF, and similarly ReFIT-KF success rates and performance can drop relative to native arm control (see generalization tasks described below).

Experiments across days and years demonstrated consistent high performance (Fig. 2). Performance was stable as measured by throughput (Supplementary Modeling) on 280 individual experimental days. These data were collected across 29 months for monkey L and 16 months monkey J, spanning the range of 0.4 to 4.4 years post-array implantation. To explore the possibility that performance changed with time, we computed least squares linear fits on these performance data for each monkey. The slopes of both regression lines are positive, suggesting that performance was stable over the time period of the study and providing evidence consistent with the hypothesis that intracortical microelectrode arrays may permit years of high performance neural control[15] (for more details see Online Methods: Quantifying Performance Across Months and Supplementary Table 2).

Generalization & Robustness

We also tested additional behavioral tasks to assess generalization of the ReFIT-KF control algorithm. We fit the ReFIT-KF algorithm with center-out-and-back reaches, as before, and then tested on a pinball task in which targets could appear at any location in the 2D workspace. Monkeys were again required to move the cursor to the target and hold it for 500 ms to successfully complete a trial. Monkey L continuously acquired targets for over 90 minutes during two pinball reaching sessions (Fig. 3), one with native arm control and one with the ReFIT-KF control. Given two seconds to acquire a target on each trial, both sessions had success rates >98%. Across the whole session, the mean time to target for ReFIT-KF control was 72% as fast as native arm control (ReFIT-KF: 710 ± 317 ms; native arm: 519 ± 196 ms; mean ± s.d.). Comparable performance was found for monkey J (Supplementary Video 5). Performance with ReFIT-KF was not only high (over two-thirds as fast as the natural arm; comparable acquire time distributions, Fig. 3a), but was also sustained without intervention (Fig. 3b). Sustained performance was typical of ReFIT-KF control sessions, whereas Velocity-KF control sessions with the same task parameters had much lower success rates (<40%), and the monkeys could not be motivated to acquire targets for more than 30 minutes.

To further test ReFIT-KF control, we trained monkey J to avoid visually defined obstacles that appeared in the direct path of the target (Fig. 4, maze task[17,18]). The monkey reached from a central starting target to either a left or right peripheral target. On some trials a barrier appeared along with the peripheral target. To successfully complete a trial, the monkey had to use the cursor to acquire and hold the peripheral target for 500 ms without hitting the barrier. This task was difficult, but the monkey successfully acquired and held the target on 77% of trials with his native arm (Fig. 4a) and on 75% of trials with ReFIT-KF control (Fig. 4b). Under ReFIT-KF control, mean time to target for this task was 74% as fast as with native arm control (ReFIT-KF: 1253 ± 588 ms; native arm: 932 ± 709 ms; mean ± s.d.). With Velocity-KF control, the monkey could not complete the task and quickly became frustrated and disengaged. As in the previous tasks, the ReFIT-KF was fit with center-out-and-back movements and used without modification for the maze task, demonstrating robustness across behavioral contexts.

ReFIT-KF: Two Innovations Designed for Closed Loop Neural Control

The described cursor control performance was achieved by redesigning the Velocity-KF algorithm from a closed loop control perspective (Supplementary Modeling). The prosthetic device constitutes a new physical plant with different dynamical properties than the native arm (Fig. 5a). The subject controls this new plant by modulating measured neural signals (y_t), which are then decoded into a velocity (v_t) by the control algorithm. This velocity is used to update the cursor on screen, affecting neural signals on subsequent time steps. This closed loop control perspective suggests two design innovations that both contribute to the described performance (Supplementary Figs. 3,4). The first innovation is a modification of neural prosthetic model fitting methodology. The second innovation is an alteration of the control algorithm. These ReFIT-KF innovations produce the neural prosthetic results described above.

Innovation 1—The first design innovation is to fit the neural prosthesis against estimates of intended velocity. Previous algorithms[1,3,5,10] implicitly assume that the subject uses the same control strategy for moving the native arm and the prosthetic cursor. Since these control strategies may be quite different, we, in the vein of past studies[2,4,6,9,12,19,20], evaluated methods that attempt to better capture the subject's strategy during prosthetic control. Ideally, the control algorithm would be fit to the subject's intended cursor velocity during closed loop neural control. Since we lack explicit access to the monkey's intentions, we hypothesized that the monkey wished to move directly towards the target; this resembles movements made by the native arm and is a good strategy for acquiring rewards.

A two-stage optimization procedure (Fig. 5b) is used to fit the neural prosthetic model to these estimates of intended velocity during online neural control. In stage 1, the monkey controls the cursor using his arm. An initial model is fit using arm trajectories and simultaneously recorded neural signals. The monkey then controls the neural prosthesis with this initial model. In stage 2, neurally controlled cursor kinematics and neural signals are recorded and used to fit a new model with an estimate of intended cursor velocity. By starting with cursor velocities collected during the previous online control session (shown in red), these estimates are calculated for model fitting using two transforms. First, the velocities are rotated towards the target (blue vectors) to generate the set of estimated intended velocities. Second, if the cursor is on target, the monkey's best strategy is to keep the cursor still to satisfy the hold time requirement. Thus, in the training set, we assume that the monkey's intention during these hold periods was to maintain the cursor position by commanding zero velocity. This zero velocity assumption applied to the fitting of model parameters improves online performance without changing the control algorithm (Supplementary Modeling and Supplementary Fig. 5). These estimated intentions and corresponding neural data are used to fit the ReFIT-KF control algorithm. It is important to note that the intention estimation is applied *only to training data:* during online control the neural prosthesis has no knowledge of the task goal or placement of targets (unlike e.g.[5,21,22]).

The aforementioned training protocol utilizes arm-controlled reaches as training data. In a paralyzed individual, it is not possible to record arm kinematics for this step. Instead, this training step could rely on the individual imagining a set of instructed movements. To test this possible strategy, we trained the initial algorithm based on visual cue observation[8,12], removing the requirement for arm control in step one. During these trials the monkey watched a computer controlled training cursor that automatically moves out to targets. The initial model was fit using automated training cursor trajectories and simultaneously recorded neural activity, without using measured arm movement (Online Methods: Observation Based Model Training). Table 1 summarizes ReFIT-KF performance for three experimental sessions from each monkey in which stage 1 of ReFIT-KF model training is based on observation data instead of arm movements. The performance, as measured by

Fitts's Law[23] (Supplementary Modeling), for these sessions is similar to that attained for the native arm control initiated sessions described (Figs. 1 and 2).

Innovation 2—The second design innovation builds on the observation that neural activity is correlated with both the velocity and the position of the cursor. Most existing neural prostheses model a relationship between neural activity and either velocity[2,4] or position[1,10]. A human clinical trial has shown that neural prostheses modeling velocity have higher performance[11,12]. However, if the control algorithm models only the velocity relationship, then position-based changes in firing will confound decoded velocities (Supplementary Modeling). To mitigate this effect, we explicitly model velocity as the user's intention and cursor position as an additional variable that affects neural output. This modification allows the user to control velocity with measured neural signals while accounting for the influence of cursor position. We explicitly assume that the current cursor position, determined by integrating the previous velocity output, is encoded in the neural activity along with the monkey's current intended velocity output. Thus, the expected contribution of position to neural activity is removed, enabling more accurate estimation of intended velocity (Fig. 1 and Supplementaty Modeling).

Discussion

Other studies have noted the potential change in plant and control strategy and have addressed it by iteratively refining parameters during neural prosthetic experiments[2,4,6,9,12,19,20]. This approach recognizes that control strategies, and therefore model parameters, are best measured and understood during closed-loop neural prosthetic experiments. However, randomizing initial parameters[2,4,20] may create a control algorithm that never attains the highest possible performance, just as optimization problems can easily become trapped in local optima (Supplementary Modeling). Although, if the recorded neural population and the neural control mapping are held constant, the consequences of the plant mismatch can be overcome through learning. Such learning was demonstrated with neural control mappings built to reconstruct arm kinematics, as well as with a neural control mapping in which neuron identities were shuffled, so the decoder output was no longer predictive of native arm kinematics[7].

The focus of the present study was to obtain high control performance within a single session by improving the neural control algorithm and optimizing its parameters. Although the neural prosthesis constitutes a new plant with different properties than the native arm, the motor cortices are involved in native arm control (e.g.,[24]). Therefore, we hypothesized that initializing a model with the relationship between neural activity and natural arm movement would allow the second stage of our training method to achieve a higher level of optimization. Previous studies[4,20] have relied on manipulating the control task to refine the neural decoder, such as by providing assisted control. In those studies, an automated correct answer was mixed with the output of the neural prosthesis. Over successive iterations the weight of automated control was decreased by experimenter intuition until control was driven only by neural activity. Our approach is different, as the control task remains constant throughout a neural prosthetic session and only the training data are manipulated between the first and second neural prosthetic sessions.

In a study with quadriplegic humans[11,12], the neural prosthesis was initially trained with visual cue observation, similar to the control experiments described above. The study also uses a second neural prosthetic training session to account for differences during online control. Unlike ReFIT-KF training, both the neural cursor and the automated training cursor were on screen during the second session. The neural cursor was presented to provide feedback so that the participant could attempt to alter their neural output to better follow the training cursor. After this second session, the neural prosthesis was fit with the training cursor kinematics. Thus, the underlying assumption is that the training cursor kinematics capture the intended kinematics during online control, while ReFIT-KF fitting assumes that intended kinematics are best inferred by the output of the neurally controlled cursor and knowledge of the task goals.

In studies with adaptive decoders[6,9], the kinematics of the neurally controlled cursor are continuously used to refine neural prosthetic parameters, also allowing them to account for differences when switching to online control. However, they take different approaches to estimating intended kinematics for retraining. One approach is to use decoder parameters non-causally, via smoothing, to estimate intended kinematics for retaining without task or target information[9]. Impressively, this method was shown to slow performance declines in one monkey over 29 days when using static spike sorting. However, unlike with ReFIT-KF model fitting, initial performance was not surpassed, perhaps because without incorporating task goals, the method is subject to inaccuracies present in the initial model fit. In another adaptive study[6], target information was incorporated in the kinematics used for retraining. Their algorithm was retrained with a weighted average of decoded trajectory positions and the target position for each trial as an estimate of intended position. In contrast, ReFIT-KF estimates intended velocities based on intuitive rules applied to cursor position, decoded velocity, and target position.

ReFIT-KF explicitly treats position and velocity differently. The resulting neural prosthesis assumes that the monkey is controlling velocity and not position, providing performance gains over a position/velocity Kalman filter that does not make this distinction (Supplementary Figs. 6–8 and Supplementary Modeling). We structured the model assuming that velocity intentions evolve smoothly and that the influence of position is based on the monkey’s internal model of the cursor. Furthermore, we assume that the control algorithm output and the monkey’s internal belief about cursor position agree. In reality, there is some mismatch between the control algorithm’s position estimation and the animal’s internal belief due to inaccuracies in assessing visual information. There are likely spatial and temporal components to this inaccuracy that are not modeled. The spatial aspect is an inexact assessment of the last seen location, and the temporal aspect is due to visual latency. The spatial aspect could be modeled as fixed position uncertainty. To fully account for the temporal aspect, one could attempt to algorithmically model the animal’s internal model of cursor dynamics since the last known position of the cursor. In this work, we chose to start with a simpler model, assuming that this estimation, which is local in time, is exact. It is possible that augmenting the algorithm to account for the mismatch between the temporally local forward model and our dynamics model could further increase control performance. Such work could also lead to improvements in the intention estimation methods used for model training. It is important to note that there may be alternative explanations for the presence of position information in neural output. For example, this information could be intended cursor position instead of an internal model estimate of current cursor position. In support of the internal model hypothesis, a recent study suggests that a forward model of cursor position is used during closed loop control[25]. However, further study of the role of position information in the neural activity during online control is necessary and could aid in the development of future control algorithms.

In experimental sessions, ReFIT-KF performance was stable until the monkey appeared to lose interest in the task (e.g. drop in target acquisition rate, Fig. 3b). This rapid drop off is consistent with native arm control session performance and is presumably the analogue of when a hypothetical human user is finished using their neural prosthesis. It is expected that performance will drift over time[14], and methods for continuous adaptation of neural control algorithm parameters may be necessary. In a previous study[9], information from the output of the control algorithm was used with a Bayesian approach to adapt parameters throughout sessions to sustain performance. If task goals were known throughout neural prosthesis use, the intention estimates defined in this study could be used in conjunction with these parameter adaptation methods. It may be possible to estimate these task goals based on features of the neural prosthetic output. For example, if a click or target selection signal is simultaneously decoded[12], indicating user intended target selection, intended cursor velocities could be estimated for moments prior to target selection. Additionally, in future work, it will be important to assess how multi-day learning[7] affects the performance and robustness when control algorithm parameters are set as described in this work, based on estimated movement intention, versus existing methods for parameter initialization. Adapting the methods of this work to enable multi-day learning and plasticity, such as by providing a consistent controller day over day, may well lead to even better performance over time.

Long-duration, continuous, high-performance operation is central to successful translation of neural prostheses to human patients[14]. The above performance depended upon three specific design choices used by both Velocity-KF and ReFIT-KF, in addition to the two key innovations defining ReFIT-KF. First, we did not employ spike sorting. The goal of spike sorting is to separate single channels composed of action potentials from many neurons into multiple channels of spiking activity from individual neurons. This standard practice can yield high information per electrode, but requires tracking each sorted action potential shape over days, which has recently been shown to be extremely difficult for many electrode channels[7,26,27]. To reduce signal instabilities that can result from imperfect spike sorting and neuron tracking, we counted the number of threshold crossings per electrode instead of spike sorting (Online Methods: Signal Acquisition & Conditioning)[15,20]. Second, the results reported here were acquired from arrays 19–40 months (monkey L) and 4–21 months (monkey J) after neurosurgical implantation[15,26]. The number of highly distinguishable single neurons on an electrode array tends to decrease over time. Yet, remaining multiunit activity often has neural prosthesis relevant tuning. By employing these older array implants, which had relatively few clearly distinguishable single units, we confirm that threshold-crossing-based activity, together with the ReFIT-KF, provides high performance for months and years after array implantation (shown in Fig. 2, also see Supplementary Table 2, and Supplementary Fig. 9). Finally, we used a single, relatively short 50 ms neural integration time window with no additional temporal lag, unlike some neural prosthetic designs that explicitly incorporate neural data with longer histories and additional lags (e.g., multiple 100 ms time bin[7]; multiple 50 ms time bins with history as far back as 1 sec[8,10]). This choice was based on experiments with humans using an online prosthetic simulator[16] and on subsequent neural control experiments with monkeys. Both indicated that shorter time bins are preferable due to reduced closed-loop feedback time.

This study demonstrates the utility of an online control perspective for the development of neural control algorithms. Although performance advances must ultimately be verified online, this perspective can be applied in offline simulation studies to gain insight into algorithmic design decisions (Supplementary Modeling). However, as with any simulation study, the applicability of the results are subject to both the limitations of simulation platform and the design decisions made in developing the simulation[16].

The sustained performance and robustness of these ReFIT-KF neural prosthetic experiments demonstrate the potential to provide functional restoration for patients with a limited ability to move and act upon the world due to neurological injury and disease. Although descending pathways are compromised, motor cortex may be largely intact, enabling this class of technology[10–12,28]. In recent years, brain interface technologies employing a variety of signal sources, such as the intracortical arrays described here, electroencephalography (EEG)[29], and electrocorticography (ECoG)[30], have been developed. The neural prostheses research community continues to create options for disabled individuals and to assess relative risk and benefit[31]. In this report, we have investigated the principled design of closed loop neural control algorithms, resulting in the development of the ReFIT-KF and demonstrations of a significant advance in performance and robustness. This algorithm, closed loop control perspective, and system design methodology may be applied to other neural prosthetic domains with the potential to considerably increase benefit and the clinical viability prostheses.

Methods

Surgical Procedures and Behavioral Experiments

All procedures and experiments were approved by the Stanford University Institutional Animal Care and Use Committee (IACUC). Experiments were conducted with adult male rhesus macaques (L and J), implanted with 96 electrode Utah arrays (Blackrock Microsystems Inc., Salt Lake City, UT) using standard neurosurgical techniques[32]. Monkeys L and J were implanted 19–53 months and 4–21 months prior to the experiments. Electrode arrays were implanted in the dorsal aspect of premotor cortex (PMd) and primary motor cortex (M1), as estimated visually from local anatomical landmarks.

The monkeys were 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[16]. The virtual cursor and targets were presented in a 3D environment (MSMS, MDDF, USC, Los Angeles, CA). 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 less than 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) and were available to the behavioral control system within 5 ± 1 ms. Visual presentation was provided via two LCD monitors with refresh rates at 120 Hz, yielding frame updates within 7 ± 4 ms. Two mirrors visually fused the displays into a single 3D percept for the user, creating a Wheatstone stereograph (see Fig. 2 in [16]).

Central results were replicated multiple days in each monkey, employing a within-day A-B-A block structure trial design to highlight algorithmic impact and thereby quantify performance and robustness (Supplementary Figs. 3,4).

Center-Out-and-Back Task Configurations

Training sets for fitting the neural control algorithm were collected using the same center-out-and-back task shown in Figure 1a. Targets were either uniformly placed at an 8 cm radius or at a 12 cm radius. For some native arm control sessions, the top target was at 14 cm and the upper right and upper left targets were at 13 cm. Training sets were typically composed of about 500 (peripheral and central) target acquisitions. All of the test sets from Figure 1 were collected using a standardized target configuration, with eight peripheral targets uniformly placed at 8 cm from the central target with either 5 cm or 6 cm acceptance windows.

Signal Acquisition & Conditioning

Neural signals are acquired from an implanted 96-channel Utah Microelectrode Array (Blackrock Microsystem, Salt Lake City, UT) using the Cerebus Recording System (Blackrock Microsystems, Salt Lake City, UT). An analog bandpass filter with a 0.3 Hz to 7.5 kHz passband is applied to each channel. Channels are sampled at 30 k Samples/second and are filtered with a 250 Hz to 7.5 kHz digital bandpass filter. A threshold detector is applied to each bandpassed channel. The threshold value is set automatically as −4.5 times the measured root mean squared value of the channel. When the signal value is less than threshold a spike event is registered for that channel. The number of spike events are counted in non-overlapping temporal bins (typically 50 ms). The counts for each channel are the inputs to the control algorithm.

Quantifying Performance Across Months

The same center-out-and-back task was run on 280 sessions across monkeys L and J, spanning at least 16 months for each monkey. Although additional experiments (using different control algorithms and behavioral tasks) may have been run on these experimental days, at least 200 trials of center-out-and-back with the ReFIT-KF control algorithm were tested. On most experimental days, the task difficulty was greater than that shown in Figure 1 and Supplementary Table 1. For the experiments documented in Figure 1, the task difficulty was selected so that the monkey could successfully complete the task with the lower quality of control afforded by the Velocity-KF algorithm.

The Fitts' law calculation is used to provide a metric that normalizes across task difficulty. For reference, monkey L was implanted on 01/22/2008 and monkey J was implanted on 08/24/2009. Data for monkey L were collected on 182 sessions over 29 months (from 24 to 53 months post-implantation). Data for monkey J were collected on 98 sessions over 16 months (from 5 to 21 months post-implantation). Each open square and circle in Figure 2 corresponds to a single experimental day on which the index of difficulty was 1.32 (4 cm targets at 8 cm from center) and throughput was calculated from at least 40 trials of center-out to either a vertical or horizontal target. All experiments from the timespans indicated that match these criteria were included, except for days on which other experiments may have impacted animal behavior. Regression lines were fit for data from each monkey using least square regression and p-values were calculated using an ANOVA for linear regression models.

Observation Based Model Training

Since paralyzed users of neural prostheses cannot generate overt arm movements an observation based algorithm training methodology can be used, as in previous animal studies[8] and clinical trials[10]. We tested the ReFIT-KF algorithm with observation based training, replacing the native arm movement stage of algorithm training with an observation stage (Fig. 5b).

Observation-based decode models were built with both of the monkey's arms comfortably restrained along his side. A previously recorded arm-controlled experimental block of 500 center-out and back trials was shown to the monkey while in this posture. The kinematics of this recording were derived from a arm-controlled session from Monkey L. To help keep the monkey engaged in the task, he was rewarded when the computer-controlled cursor acquired and held the target for 500 ms.

Under this experimental context, the neural data recorded during these observation sessions and the previously recorded cursor kinematics served as the training data to build the initial decode model. This resulting model was then run online and used as training data to build the ReFIT-KF decoder. Little to no arm movement was visually noted during both observational blocks and decoding blocks.

Performance of ReFIT-KF based control during these sessions, as measured by the Fitts' law metric, was roughly equivalent to performance on sessions that initially trained from arm movement data.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank J. Aguayo, M. Risch, & S. Kang for surgical assistance & veterinary care, D. Haven & B. Oskotsky for IT support, S. Eisensee, B. Davis, & E. Castaneda for administrative assistance, P. Ortega for mathematical insight, and S. Stavisky for data collection assistance. 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., J.C.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, DARPA Reorganization and Plasticity to Accelerate Injury Recovery N66001-10-C-2010, 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.).

References

1. Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR, Donoghue JP. Instant neural control of a movement signal. Nature. 2002; 416:141–142. [PubMed: 11894084]
2. Taylor DM, Tillery SI, Schwartz AB. Direct cortical control of 3D neuroprosthetic devices. Science. 2002; 296:1829–1832. [PubMed: 12052948]
3. Carmena JM, et al. Learning to control a brain-machine interface for reaching and grasping by primates. PLoS Biol. 2003:1.
4. Velliste M, Perel S, Spalding MC, Whitford AS, Schwartz AB. Cortical control of a prosthetic arm for self-feeding. Nature. 2008; 453:1098–1101. [PubMed: 18509337]
5. Mulliken GH, Musallam S, Andersen RA. Decoding trajectories from posterior parietal cortex ensembles. J Neurosci. 2008; 28:12913–12926. [PubMed: 19036985]
6. Shpigelman, L.; Lalazar, H.; Vaadia, E. Kernel-arma for hand tracking and brain-machine interfacing during 3D motor control. In: Koller, D.; Schuurmans, D.; Bengio, Y.; Bottou, L., editors. Adv in Neural Info Proc Sys. Vol. 21. 2009. p. 1489-1496.
7. Ganguly K, Carmena JM. Emergence of a stable cortical map for neuroprosthetic control. PLoS Biol. 2009:7.
8. Suminski AJ, Tkach DC, Fagg AH, Hatsopoulos NG. Incorporating feedback from multiple sensory modalities enhances brain-machine interface control. J Neurosci. 2010; 30:16777–16787. [PubMed: 21159949]
9. Li Z, O'Doherty JE, Lebedev MA, Nicolelis MA. Adaptive decoding for brain-machine interfaces through bayesian parameter updates. Neural Comp. 2011; 23:3162–3204.
10. Hochberg LR, et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature. 2006; 442:164–171. [PubMed: 16838014]
11. Kim SP, Simeral JD, Hochberg LR, Donoghue JP, Black MJ. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. J Neural Eng. 2008; 5:455–476. [PubMed: 19015583]
12. Kim SP, et al. Point-and-click cursor control with an intracortical neural interface system in humans with tetraplegia. IEEE Trans Neural Syst Rehabil Eng. 2011; 19:193–203. [PubMed: 21278024]
13. Hochberg LR, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature. 2012; 485:372–375. [PubMed: 22596161]
14. Judy J. Neural interfaces for upper-limb prosthesis control: opportunities to improve long-term reliability. IEEE Pulse. 2012; 3:57–60. [PubMed: 22481748]
15. Chestek CA, et al. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. J Neural Eng. 2011; 8:045005. [PubMed: 21775782]
16. Cunningham JP, et al. A closed-loop human simulator for investigating the role of feedback-control in brain-machine interfaces. J Neurophysiol. 2011; 105:1932–1949. [PubMed: 20943945]
17. Churchland MM, Cunningham JP, Kaufman MT, Ryu SI, Shenoy KV. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? Neuron. 2010; 68:387–400. [PubMed: 21040842]
18. Kaufman MT, et al. Roles of monkey premotor neuron classes in movement preparation and execution. J Neurophysiol. 2010; 104:799–810. [PubMed: 20538784]
19. Gage GJ, Ludwig KA, Otto KJ, Ionides EL, Kipke DR. Naive coadaptive cortical control. J Neural Eng. 2005; 2:52–63. [PubMed: 15928412]

20. Fraser GW, Chase SM, Whitford A, Schwartz AB. Control of a brain-computer interface without spike sorting. J Neural Eng. 2009; 6:055004. [PubMed: 19721186]

21. Srinivasan L, Eden UT, Mitter SK, Brown EN. General-purpose filter design for neural prosthetic devices. J Neurophysiol. 2007; 98:2456–2475. [PubMed: 17522167]

22. Yu BM, et al. Mixture of trajectory models for neural decoding of goal-directed movements. J Neurophysiol. 2007; 97:3763–3780. [PubMed: 17329627]

23. Card SK, English WK, Burr BJ. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. Ergonomics. 1978; 21:601–613.

24. Kalaska JF. From intention to action: motor cortex and the control of reaching movements. Adv Exp Med Biol. 2009; 629:139–178. [PubMed: 19227499]

25. Golub MD, Yu BM, Chase SM. Internal Models Engaged by Brain-computer Interface Control. Conf Proc IEEE Eng Med Biol Soc. 2012; 2012:1327–1330. [PubMed: 23366143]

26. Chestek CA, et al. Neural prosthetic systems: current problems and future directions. Conf Proc IEEE Eng Med Biol Soc. 2009; 2009:3369–3375. [PubMed: 19963796]

27. Santhanam G, et al. HermesB: a continuous neural recording system for freely behaving primates. IEEE Trans Biomed Eng. 2007; 54:2037–2050. [PubMed: 18018699]

28. Hochberg LR. Turning thought into action. N Engl J Med. 2008; 359:1175–1177. [PubMed: 18784110]

29. McFarland DJ, Sarnacki WA, Wolpaw JR. Electroencephalographic (EEG) control of three-dimensional movement. J Neural Eng. 2010; 7:036007. [PubMed: 20460690]

30. Schalk G, et al. Two-dimensional movement control using electrocorticographic signals in humans. J Neural Eng. 2008; 5:75–84. [PubMed: 18310813]

31. Gilja V, et al. Challenges and opportunities for next-generation intra-cortically based neural prostheses. IEEE Trans Biomed Eng. 2011; 58:1891–1899. [PubMed: 21257365]

32. Santhanam G, Ryu SI, Yu BM, Afshar A, Shenoy KV. A high-performance brain-computer interface. Nature. 2006; 442:195–198. [PubMed: 16838020]

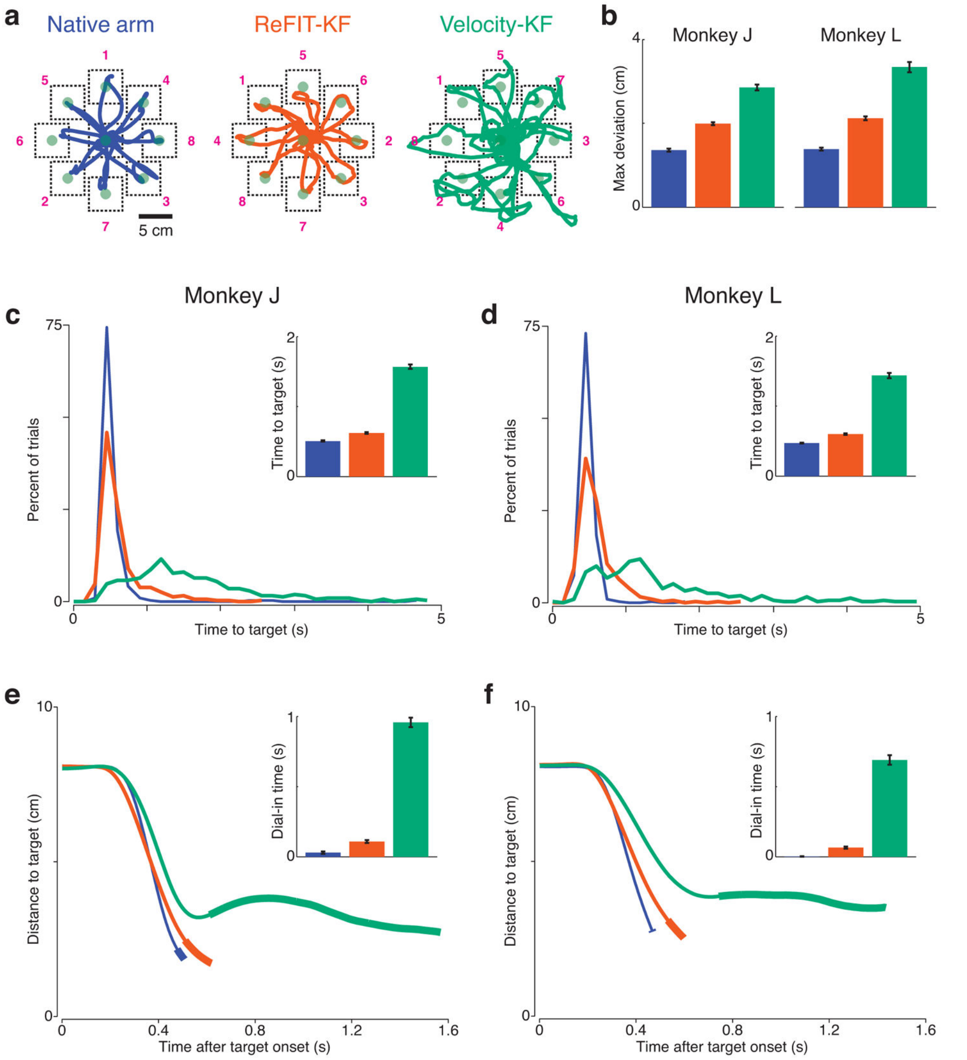

Figure 1. Performance comparison of native arm, ReFIT-KF, and Velocity-KF based cursor control

Native arm control shown in blue, ReFIT-KF in red, and Velocity-KF in green. All plots, except the cursor path traces, are constructed from 4 experimental days for each monkey on which all 3 control methods were tested. For each monkey and control method there are 545 to 659 center-out-and-back movements. (a) Representative traces of cursor path during center-out-and-back reaches. Dashed lines (not visible to the monkey) are the demand boxes for the eight peripheral targets and the central target, shown as translucent green circles. Targets alternated between the center and the peripheral in the sequence indicated by the numbers shown. Traces are continuous for the duration of all sixteen center-out-and-back movements, representing 15.27, 16.87, and 32.23 seconds of native arm, ReFIT-KF, and Velocity-KF reaching, respectively. (b) Bar graphs plotting maximum deviation from a straight-line path to the target on each successful trial (mean ± s.e.m). (c,d) Histograms of time to target for successful trials are shown as line graphs for monkeys J and L. The inset bar graphs plot the time to target (mean ± s.e.m). (e,f) Line graphs plotting the mean distance to the target as a function of time. The inset bar graph plots the mean ± s.e.m of the dial-in time, or the time required to finally settle on the demand box, after first acquired, to successfully hold for 500 ms. Hold time is not included in the dial-in time. The thickened portion of the line graphs also indicate dial-in time, beginning at the mean time of first target acquire, and ending at mean trial duration minus 500ms. These data are from experiments J-2010-10-27, J-2010-10-28, J-2010-10-29, J-2010-11-02, L-2010-10-27, L-2010-10-28, L-2010-10-29, and L-2010-11-02.

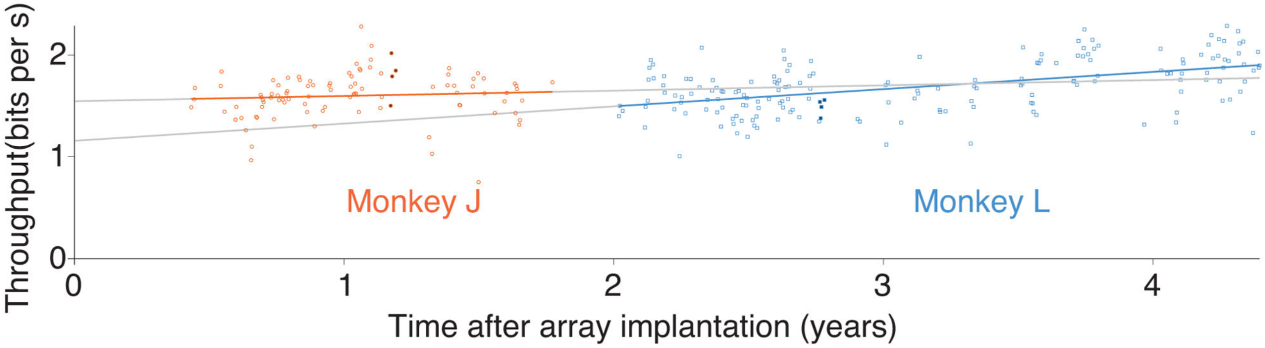

Figure 2. Performance of ReFIT-KF control across 4 years

Performance is measured by the Fitts's law metric (Supplementary Modeling). Data from monkey J and monkey L are shown as 98 orange circles and 182 cyan squares, respectively. Each point plots the performance of the ReFIT-KF algorithm trained on that experimental day. The eight filled data points (four for each monkey) are calculated from the same datasets used to generate Figure 1. Linear regression lines for data from monkey J (orange) and monkey L (cyan) are shown. For all datasets shown, the trial success rate was >90%. Additional details on these data are summarized in Supplementary Table 2.

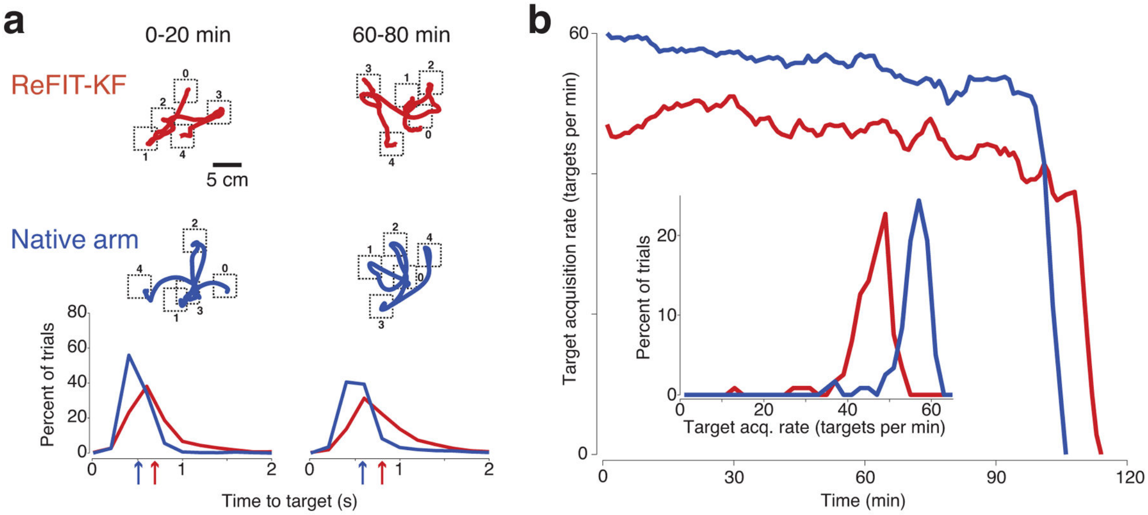

Figure 3. Performance comparison of native arm vs. ReFIT-KF for the pinball task

Native arm control is in blue, ReFIT-KF in red. In this task, each target location is selected from a uniform distribution spanning the workspace. (a) Each column shows data from 20-minute segments. The top rows are randomly selected cursor traces for 4 subsequent target acquisitions. Target demand boxes are shown as dashed lines and target sequence is indicated from 0 to 4. The bottom row shows normalized histograms of time to target for successful trials. Arrows below the plot indicate average time. (b) Target acquisition rate per minute throughout the sessions is shown. The sharp rate drop indicates when the monkey lost interest in the task. A histogram of acquisition rate across the sessions is inset. The native arm and ReFIT-KF sessions (L-2010-04-01 and L-2010-04-12) were on two separate days, within 11 days of each other, when the monkey demonstrated a high degree of motivation.

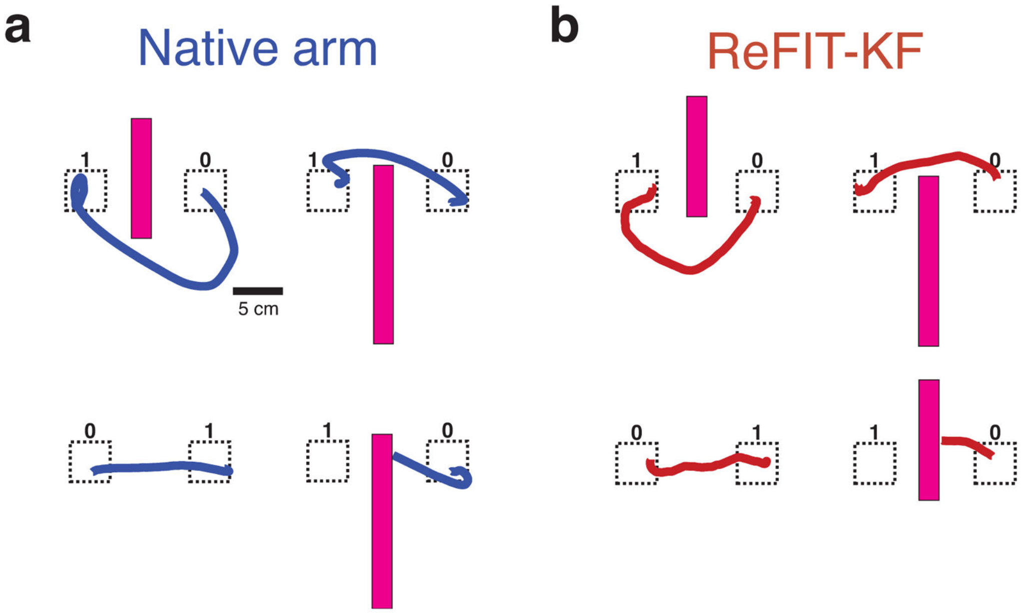

Figure 4. Performance comparison of native arm vs. ReFIT-KF for the obstacle avoidance task
Native arm control shown in blue, ReFIT-KF in red. In this task the monkey had to move the cursor from the initial target (labeled 0) to the final target (labeled 1, demand box shown as dashed line) without hitting the magenta-colored barrier. One representative cursor trace is shown from each of the four principle observed movement types: curve under, curve over, straight (no barrier), and collision into barrier. These data are from experiment J-2010-03-09.

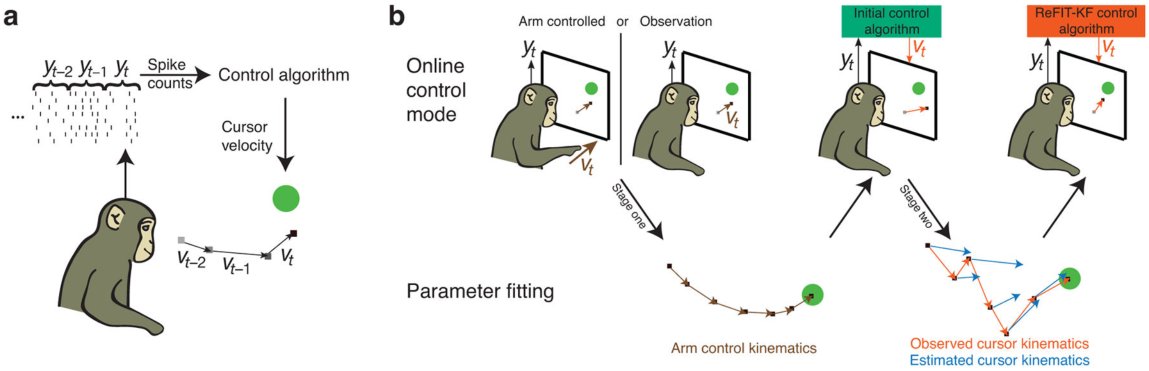

Figure 5. Illustrations of the online neural control paradigm and the ReFIT-KF training methodology

(a) The input to the control algorithm at time *i* is a vector of spike counts, y_t, from implanted electrodes. Y_t is translated into a velocity output, v_t, to drive the cursor. (b) ReFIT-KF is trained in two-stages. Initially, cursor kinematics and neural activity are collected during arm control or during an observation phase in which cursor movement is automated. These arm movement or observed cursor kinematics are regressed against neural activity to generate an initial control algorithm. Then, a new set of cursor kinematics and neural activity are collected using the initial algorithm in closed loop. The kinematics collected during neural control (red vectors) are used to estimate intention by rotating the velocities towards the goal (blue vectors). This estimate of intended kinematics is regressed against neural activity to generate and run ReFIT-KF.

Table 1

Performance of ReFIT-KF based control with observation based algorithm training.

Experiment	Target Center Distance (mm)	Window Size (mm)	Acquire Time (s)	Index of Difficulty (bits)	Throughput (bits/s)	Success Rate
L-2010-08-10	80	40	0.89	1.32	1.49	94%
L-2010-08-11	80	40	0.89	1.32	1.48	95%
L-2010-08-12	80	40	0.82	1.32	1.59	94%
J-2010-08-10	80	40	0.76	1.32	1.74	97%
J-2010-08-16	80	40	0.82	1.32	1.60	97%
J-2010-08-17	80	40	0.76	1.32	1.73	98%