nature neuroscience

A high-performance neural prosthesis enabled by control algorithm design

Vikash Gilja[1,2,13], Paul Nuyujukian[3,4,13], Cindy A Chestek[2,5], John P Cunningham[5,6], Byron M Yu[5,7–10], Joline M Fan[3], Mark M Churchland[5,7], Matthew T Kaufman[7], Jonathan C Kao[5], Stephen I Ryu[5,11] & Krishna V Shenoy[2,3,5,7,12]

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 individuals with disabilities 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 intention–trained Kalman filter (ReFIT-KF) that incorporates assumptions about the nature of closed-loop neural prosthetic control. When tested in rhesus monkeys implanted with motor cortical electrode arrays, the ReFIT-KF algorithm outperformed existing neural prosthetic algorithms in all measured domains and halved target 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 in two monkeys, thereby increasing the clinical viability of neural prostheses.

Neural prostheses have recently shown considerable promise in 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 a closed loop (for example, refs. 1–3). When such a neural prosthesis is introduced to a monkey, performance can improve over 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 algorithm advances (for example, ref. 12), system performance on simple cursor-control tasks remains low relative to the performance of native arm control, presenting a critical barrier to clinical translation[14].

To improve the performance of neural prostheses, we focused 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. In addition, we chose signal-conditioning methods, which transform recorded neural signals into control algorithm input, to improve system stability and performance[15,16]. As demonstrated in closed-loop neural control experiments, these methods resulted 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 an algorithm, ReFIT-KF, that led to substantially higher-performance neural prosthetic control. **Figure 1** compares cursor movements for three different modalities: native arm control, ReFIT-KF and a velocity Kalman filter (Velocity-KF), which is state of the art for current neural prostheses (for example, refs. 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. In this center-out-and-back task, targets alternated between a central location and eight peripheral locations. During online neural control sessions, the monkey's contralateral arm was not restrained and movement continued. However, the physical movement was not stereotyped and would often attenuate or even stop

[1]Department of Computer Science, Stanford University, Stanford, California, USA. [2]Stanford Institute for Neuro-Innovation and Translational Neuroscience, Stanford University, Stanford, California, USA. [3]Department of Bioengineering, Stanford University, Stanford, California, USA. [4]School of Medicine, Stanford University, Stanford, California, USA. [5]Department of Electrical Engineering, Stanford University, Stanford, California, USA. [6]Department of Engineering, University of Cambridge, Cambridge, UK. [7]Neurosciences Program, Stanford University, Stanford, California, USA. [8]Gatsby Computational Neuroscience Unit, University College London, London, UK. [9]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. [10]Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. [11]Department of Neurosurgery, Palo Alto Medical Foundation, Palo Alto, California, USA. [12]Department of Neurobiology, Stanford University, Stanford, California, USA. [13]These authors contributed equally to this work. Correspondence should be addressed to V.G. (gilja@stanford.edu).

Received 23 May; accepted 18 October; published online 18 November 2012; doi:10.1038/nn.3265

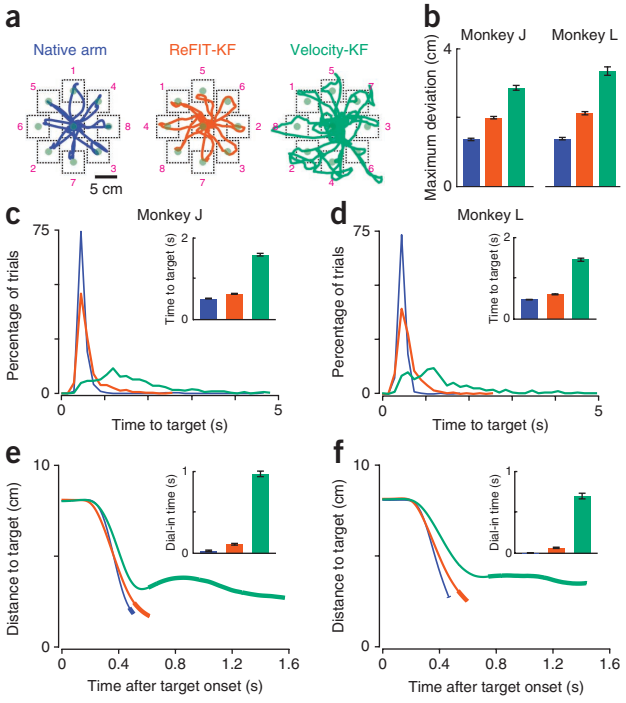

Figure 1 Cursor control with native arm, ReFIT-KF and Velocity-KF. (**a**) Representative traces of cursor path during center-out-and-back reaches by monkey J. Dotted lines (not visible to the monkey) are the demand boxes for the eight peripheral targets and the central target, shown as translucent green circles. Targets alternated between the center and the peripheral in the sequence indicated by the numbers shown. Traces were continuous for the duration of all 16 center-out-and-back movements, representing 15.27 s, 16.87 s and 32.23 s of native arm, ReFIT-KF and Velocity-KF reaching, respectively. (**b**) Maximum deviation from a straight-line path to the target on each successful trial (mean ± s.e.m.). (**c**,**d**) Time to target for successful trials for monkeys J and L. Insets show the time to target (mean ± s.e.m.). (**e**,**f**) Mean distance to the target as a function of time. Insets, 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. Thickened portions of line graphs also indicate dial-in time, beginning at the mean time of first target acquisition and ending at mean trial duration minus 500 ms. These data are from experiments (designated by monkey identifier letter, year, month and day) 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. Native arm control is shown in blue, ReFIT-KF control in orange and Velocity-KF control in green. All plots, except the cursor-path traces, were constructed from successful center-out trials from four experimental days for each monkey on which all three control methods were tested. They are computed from 644 native arm, 659 ReFIT-KF, and 619 Velocity-KF trials for monkey J and 632 native arm, 632 ReFIT-KF, and 545 Velocity-KF trials for monkey L.

during some neural control sessions while retaining performance. In additional control experiments, both arms of the monkey were restrained, and we observed little or no arm movement with similar neural control performance (**Table 1**).

The ReFIT-KF algorithm outperformed the Velocity-KF algorithm 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** and **2**). 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 control 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. Finally, 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 compared to native arm and ReFIT-KF control, but often significantly overshot the target, requiring additional time and multiple passes to stably acquire and hold the target. This overshoot-correction time dominated the overall time to successful target acquisition for Velocity-KF control and is captured by the metric 'dial-in time', which is the average time required to make the final target acquisition after having first reached the target (**Fig. 1c**–**f**). Both native arm and ReFIT-KF control allowed more precise stopping as compared to that with Velocity-KF control (**Fig. 1e**,**f**).

In all trials in eight experimental sessions with two monkeys, when given 4 s to acquire targets, ReFIT-KF achieved a success rate of >99%, whereas Velocity-KF had a success rate of 95%. We chose a task difficulty to achieve a high success rate for all three control modalities on the first experimental day (**Supplementary Table 1**). When we increased task difficulty, 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 those with native arm control (see below).

Experiments across days and years demonstrated consistent high performance of ReFIT-KF control (**Fig. 2**). Performance was stable as measured by throughput (**Supplementary Modeling**) on 280 individual experimental days. We collected these data over 29 months for monkey L and over 16 months for monkey J, spanning 0.4–4.4 years after 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] (Online Methods and **Supplementary Table 2**).

Generalization and 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 the algorithm with a pinball task in which targets could appear at any location in the two-dimensional 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 min

Table 1 Performance of ReFIT-KF–based control with observation-based algorithm training

Experiment	Target center distance (mm)	Window size (mm)	Acquisition time (s)	Index of difficulty (bits)	Throughput (bits s^{-1})	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

Throughput values, which normalize for task difficulty, are similar to values with ReFIT-KF trials for experimental sessions with arm-based algorithm training (**Fig. 2** and **Supplementary Modeling**), suggesting equivalent control performance.

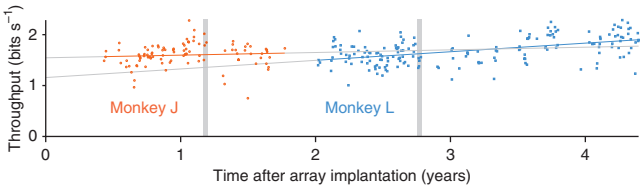

Figure 2 Performance of ReFIT-KF control across four years. Performance was measured by the Fitts's law metric (**Supplementary Modeling**). Data from monkey J and monkey L are shown as 98 orange circles and 182 cyan squares, respectively. Each point plots the performance of the ReFIT-KF algorithm trained on that experimental day. The eight filled data points (four for each monkey) within the time period indicated in gray are calculated from the same data sets used to generate **Figure 1**. Linear regression lines for data for monkey J (orange) and monkey L (cyan) are shown. For all data sets shown, the trial success rate was >90%. Additional details for these data are summarized in **Supplementary Table 2** and **Supplementary Figure 9**.

during two pinball-reaching sessions (**Fig. 3**), one with native arm control and one with the ReFIT-KF control (**Supplementary Videos 3** and **4**). Given 2 s to acquire a target in 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 that for native arm control (ReFIT-KF, 710 ms ± 317 ms and native arm, 519 ms ± 196 ms; mean ± s.d.). We observed comparable performance for monkey J (**Supplementary Video 5**). Performance with ReFIT-KF was not only high (over two-thirds as fast as the natural arm, with comparable acquisition-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 min.

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** and **Supplementary Video 6**; maze task[17–19]). The monkey reached from a central starting target to either a left or right peripheral target. In 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 in 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, 1,253 ms ± 588 ms and native arm, 932 ms ± 709 ms; mean ± s.d.). With Velocity-KF control, the monkey could not complete the task, and quickly became frustrated and disengaged. As in previous tasks, ReFIT-KF was fit with center-out-and-back movements and was used without modification for the maze task, demonstrating generalization across behavioral contexts.

ReFIT-KF: two innovations for closed-loop neural control

We achieved the described cursor-control performance by redesigning the Velocity-KF algorithm from a closed-loop control perspective (**Supplementary Modeling**). The prosthetic device constitutes a new physical plant with different dynamic 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, which affects neural signals in 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.

First innovation

The first design innovation was 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 to move the native arm and the prosthetic cursor. As these control strategies may be quite different, we, in the vein of past studies[2,4,6,9,12,20–22], 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. As we lack explicit access to the monkey's intentions, we hypothesized that the monkey wished to move directly toward the target; this resembles movements made by the native arm and is a good strategy for acquiring rewards.

We used a two-stage optimization procedure (**Fig. 5b**) 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 (recorded cursor kinematics), these estimates are calculated for model fitting using two transforms. First, the velocities are rotated toward the target to generate

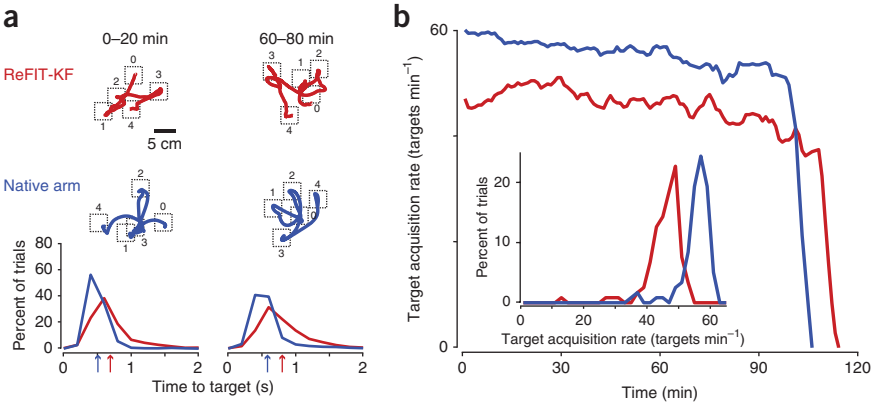

Figure 3 Performance comparison of native arm versus ReFIT-KF for the pinball task. (**a**) For two 20-min segments (columns), shown are randomly selected cursor traces from four consecutive target acquisitions (top; target-demand boxes are shown as dotted lines and target sequence is indicated from 0 to 4). In normalized histograms of time to target for successful trials (bottom), arrows below the plots indicate average time. (**b**) Target acquisition rate per minute throughout the sessions; the sharp rate drop indicates when the monkey lost interest in the task. Inset, acquisition rate across the sessions. The native arm and ReFIT-KF sessions (L-2010-04-01 and L-2010-04-12) were on two separate days, within 11 d of each other, when the monkey demonstrated a high degree of motivation. Native arm control is shown in blue and ReFIT-KF control in red. In this task, each target location was selected from a uniform distribution spanning the workspace.

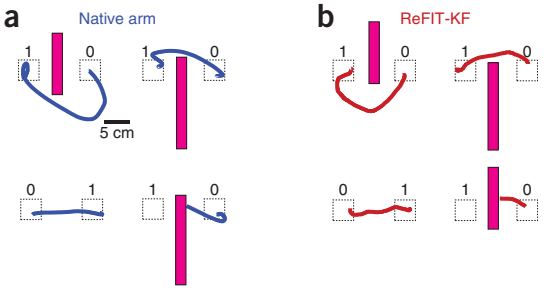

Figure 4 Performance comparison of native arm versus ReFIT-KF for the obstacle-avoidance task. (**a**,**b**) In this task the monkey had to move the cursor from the initial target (labeled 0) to the final target (labeled 1; demand box shown as dotted line) without hitting the magenta-colored barrier.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

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 assumed 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 improved online performance without changing the control algorithm (**Supplementary Modeling** and **Supplementary Fig. 5**). These estimated intentions and corresponding neural data were used to fit the ReFIT-KF control algorithm. Note that we applied the intention estimation only to training data: during online control the neural prosthesis has no knowledge of the task goal or placement of targets (unlike, for example, in refs. 5,23,24).

The aforementioned training protocol used 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 moved to targets. The initial model was fit using automated training cursor trajectories and simultaneously recorded neural activity, without using measured arm movement (Online Methods). **Table 1** summarizes ReFIT-KF performance for three experimental sessions from each monkey in which stage 1 of ReFIT-KF model training was based on observation data instead of arm movements. The performance, as measured by Fitts's law[25] (**Supplementary Modeling**), for these sessions was similar to that attained for the native arm control–initiated sessions described (**Figs. 1** and **2**).

Second innovation

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 clinical trial in humans has shown that neural prostheses modeling velocity have better performance than those modeling position[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 modeled 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 assumed 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 **Supplementary Modeling**).

DISCUSSION

Other studies have noted the potential change in plant-and-control strategy and have addressed it by iteratively refining parameters during experiments with neural prosthetics[2,4,6,9,12,20,21]. 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[2,4,21] may create a control algorithm that never attains the best 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 has been demonstrated with neural control mappings built to reconstruct arm kinematics, as well as with a neural control mapping in which neuron identities have been shuffled, so the decoder output was no longer predictive of native arm kinematics[7].

The focus of our study was to obtain high control performance in 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 (for example, ref. 26). 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 greater optimization. Previous studies[4,21] 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 had been mixed with the output of the neural prosthesis.

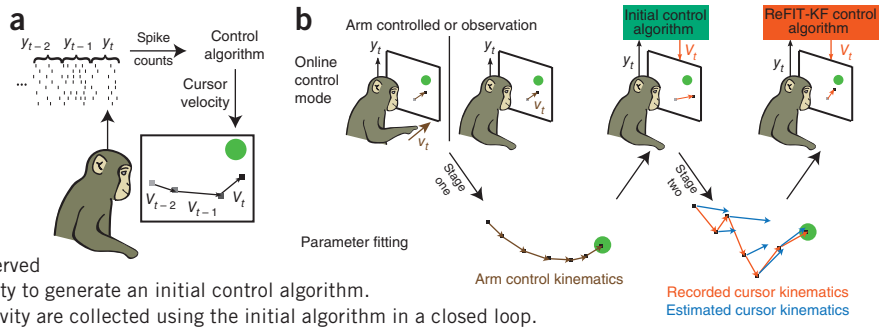

Figure 5 Illustrations of the online neural control paradigm and the ReFIT-KF training methodology. (**a**) The input to the control algorithm at time *i* is a vector of spike counts, y_t, from implanted electrodes. y_t is translated into a velocity output, v_t to drive the cursor. (**b**) ReFIT-KF is 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 kinematics 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 a closed loop. The kinematics collected during neural control (red vectors) are used to estimate intention by rotating the velocities toward the goal (blue vectors). This estimate of intended kinematics is regressed against neural activity to generate and run ReFIT-KF.

Over successive iterations the weight of automated control had been decreased by experimenter's intuition until only neural activity drove the control. Our approach is different, as the control task remained constant throughout a neural prosthetic experiment session and only the training data were manipulated between the first and second sessions.

In a study with quadriplegic humans[11,12], the neural prosthesis initially had been trained with visual-cue observation, which is 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 in ReFIT-KF training, in that study 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, whereas 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 noncausally, via smoothing, to estimate intended kinematics for retaining without task or target information[9]. This method has been shown to slow performance declines in one monkey over 29 d when using static spike sorting. However, unlike with ReFIT-KF model fitting, initial performance had not been 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 had been 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 controls velocity and not position, providing performance gains over a position and 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 assumed 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 monkey's internal belief because of 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 monkey'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 improve 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 other 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[27]. However, additional 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 (for example, drop in target acquisition rate; **Fig. 3b**). This rapid drop-off is consistent with native arm control session performance and is presumably the analog 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 has been 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 before target selection. Additionally, in future work, it will be important to assess how multiday 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 multiday 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 on 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 use 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 more decodable kinematic 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[7,28,29]. 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)[15,21]. Second, the results reported here were acquired from arrays 19–53 months (monkey L) and 4–21 months (monkey J) after neurosurgical implantation[15,28]. 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 using these older array implants, which had relatively few clearly distinguishable single units, we confirmed that threshold crossing–based activity, together with the ReFIT-KF, provides high performance for months and years after array implantation (**Fig. 2**, **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 (for example, multiple 100-ms time bins[7] and multiple 50-ms time bins with history as far back as 1 s (refs. 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 owing to reduced closed-loop feedback time.

This study demonstrated 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 examine algorithm-design decisions (**Supplementary Modeling**). However, as with any simulation study, the applicability of the results is subject to both the limitations of the 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 because of neurological injury and disease. Although descending pathways are compromised, the motor cortex may be largely intact, enabling this class of technology[10–12,30]. In recent years, brain-interface technologies using a variety of signal sources, such as the intracortical arrays described here, electroencephalography[31] and electrocorticography[32], have been developed. The neural prostheses research community continues to create options for individuals with disabilities and to assess relative risk and benefit[33]. Here we investigated the principled design of closed-loop neural control algorithms, resulting in the development of the ReFIT-KF and demonstrations of a substantial 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 of prostheses.

METHODS

Methods and any associated references are available in the online version of the paper.

Note: Supplementary information is available in the online version of the paper.

ACKNOWLEDGMENTS

We thank M. Mazariegos, J. Aguayo, W. Kalkus, S. Kang, E. Morgan and C. Sherman for surgical assistance and veterinary care, D. Haven and B. Oskotsky for information technology support, S. Eisensee, B. Davis and E. Castaneda for administrative assistance, P. Ortega for mathematical insight, and S. Stavisky for data-collection assistance. This work was supported by a US National Defense Science and Engineering Graduate Fellowship (V.G.), National Science Foundation Graduate Research Fellowships (V.G., C.A.C., J.M.F., M.T.K. and J.C.K.), Stanford Medical Scholars Program, Howard Hughes Medical Institute Medical Research Fellows Program, Paul and Daisy Soros Fellowship, Stanford Medical Scientist Training Program (P.N.), Stanford Graduate Fellowship (C.A.C., J.P.C. and J.M.F.), Gatsby Charitable Foundation (B.M.Y.), a Helen Hay Whitney postdoctoral fellowship (M.M.C.), Burroughs Welcome Fund Career Awards in the Biomedical Sciences (M.M.C. and K.V.S.), the Christopher and Dana Reeve Paralysis Foundation (S.I.R. and K.V.S.), Defense Advanced Research Projects Agency Revolutionizing Prosthetics 2009 N66001-06-C-8005, Reorganization and Plasticity to Accelerate Injury Recovery N66001-10-C-2010, US National Institutes of Health, National Institute of Neurological Disorders and Stroke Collaborative Research in Computational Neuroscience grant R01-NS054283, and National Institutes of Health Directors Pioneer Award 1DP1OD006409 (K.V.S.).

AUTHOR CONTRIBUTIONS

V.G. and P.N. were responsible for infrastructure development, animal training, and 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.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Published online at http://www.nature.com/doifinder/10.1038/nn.3265.
Reprints and permissions information is available online at http://www.nature.com/reprints/index.html.

1. Serruya, M.D., Hatsopoulos, N.G., Paninski, L., Fellows, M.R. & Donoghue, J.P. Instant neural control of a movement signal. *Nature* **416**, 141–142 (2002).
2. Taylor, D.M., Tillery, S.I. & Schwartz, A.B. Direct cortical control of 3D neuroprosthetic devices. *Science* **296**, 1829–1832 (2002).
3. Carmena, J.M. *et al.* Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol.* **1**, E42 (2003).
4. Velliste, M., Perel, S., Spalding, M.C., Whitford, A.S. & Schwartz, A.B. Cortical control of a prosthetic arm for self-feeding. *Nature* **453**, 1098–1101 (2008).
5. Mulliken, G.H., Musallam, S. & Andersen, R.A. Decoding trajectories from posterior parietal cortex ensembles. *J. Neurosci.* **28**, 12913–12926 (2008).
6. Shpigelman, L., Lalazar, H. & Vaadia, E. Kernel-arma for hand tracking and brain-machine interfacing during 3D motor control. *Adv. Neural Info. Proc. Sys.* **21**, 1489–1496 (2009).
7. Ganguly, K. & Carmena, J.M. Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol.* **7**, e1000153 (2009).
8. Suminski, A.J., Tkach, D.C., Fagg, A.H. & Hatsopoulos, N.G. Incorporating feedback from multiple sensory modalities enhances brain-machine interface control. *J. Neurosci.* **30**, 16777–16787 (2010).
9. Li, Z., O'Doherty, J.E., Lebedev, M.A. & Nicolelis, M.A. Adaptive decoding for brain-machine interfaces through bayesian parameter updates. *Neural Comput.* **23**, 3162–3204 (2011).
10. Hochberg, L.R. *et al.* Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* **442**, 164–171 (2006).
11. Kim, S.P., Simeral, J.D., Hochberg, L.R., Donoghue, J.P. & Black, M.J. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *J. Neural Eng.* **5**, 455–476 (2008).
12. Kim, S.P. *et al.* Point-and-click cursor control with an intracortical neural interface system in humans with tetraplegia. *IEEE Trans. Neural Syst. Rehabil. Eng.* **19**, 193–203 (2011).
13. Hochberg, L.R. *et al.* Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* **485**, 372–375 (2012).
14. Judy, J. Neural interfaces for upper-limb prosthesis control: opportunities to improve long-term reliability. *IEEE Pulse* **3**, 57–60 (2012).
15. Chestek, C.A. *et al.* Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *J. Neural Eng.* **8**, 045005 (2011).
16. Cunningham, J.P. *et al.* A closed-loop human simulator for investigating the role of feedback-control in brain-machine interfaces. *J. Neurophysiol.* **105**, 1932–1949 (2011).
17. Churchland, M.M., Cunningham, J.P., Kaufman, M.T., Ryu, S.I. & Shenoy, K.V. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron* **68**, 387–400 (2010).
18. Kaufman, M.T. *et al.* Roles of monkey premotor neuron classes in movement preparation and execution. *J. Neurophysiol.* **104**, 799–810 (2010).
19. Sadtler, P.T., Ryu, S.I., Yu, B.M., & Batista, A.P. High-performance neural prosthetic control along instructed paths. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2011**, 601–604 (2011).
20. Gage, G.J., Ludwig, K.A., Otto, K.J., Ionides, E.L. & Kipke, D.R. Naive coadaptive cortical control. *J. Neural Eng.* **2**, 52–63 (2005).
21. Fraser, G.W., Chase, S.M., Whitford, A. & Schwartz, A.B. Control of a brain-computer interface without spike sorting. *J. Neural Eng.* **6**, 055004 (2009).
22. Chase, S.M., Schwartz, A.B., & Kass, R.E. Bias, optimal linear estimation and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Networks* **22**, 1203–1213 (2009).
23. Srinivasan, L., Eden, U.T., Mitter, S.K. & Brown, E.N. General-purpose filter design for neural prosthetic devices. *J. Neurophysiol.* **98**, 2456–2475 (2007).
24. Yu, B.M. *et al.* Mixture of trajectory models for neural decoding of goal-directed movements. *J. Neurophysiol.* **97**, 3763–3780 (2007).
25. Card, S.K., English, W.K. & Burr, B.J. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics* **21**, 601–613 (1978).
26. Kalaska, J.F. From intention to action: motor cortex and the control of reaching movements. *Adv. Exp. Med. Biol.* **629**, 139–178 (2009).
27. Golub, M.D., Yu, B.M. & Chase, S.M. Internal models engaged by brain-computer interface control. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2012**, 1327–1330 (2012).
28. Chestek, C.A. *et al.* Neural prosthetic systems: current problems and future directions. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2009**, 3369–3375 (2009).
29. Santhanam, G. *et al.* HermesB: a continuous neural recording system for freely behaving primates. *IEEE Trans. Biomed. Eng.* **54**, 2037–2050 (2007).
30. Hochberg, L.R. Turning thought into action. *N. Engl. J. Med.* **359**, 1175–1177 (2008).
31. McFarland, D.J., Sarnacki, W.A. & Wolpaw, J.R. Electroencephalographic (EEG) control of three-dimensional movement. *J. Neural Eng.* **7**, 036007 (2010).
32. Schalk, G. *et al.* Two-dimensional movement control using electrocorticographic signals in humans. *J. Neural Eng.* **5**, 75–84 (2008).
33. Gilja, V. *et al.* Challenges and opportunities for next-generation intra-cortically based neural prostheses. *IEEE Trans. Biomed. Eng.* **58**, 1891–1899 (2011).

ONLINE 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 Microelectrode arrays (Blackrock Microsystems) using standard neurosurgical techniques[34]. Data were collected 19–53 months and 4–21 months after implantation for monkeys L and J, respectively. 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 two-dimensional 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 three-dimensional environment (MusculoSkeletal Modeling Software, Medical Device Development Facility, University of Southern California). Hand-position data were measured at 60 Hz with an infrared reflective bead–tracking system (Polaris, Northern Digital). Behavioral control and neural decode were run on separate PCs using the Simulink/xPC platform (Mathworks) 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) and were available to the behavioral control system within 5 ms ± 1 ms. Visual presentation was provided via two LCD monitors with refresh rates at 120 Hz, yielding frame updates of 7 ms ± 4 ms. Two mirrors visually fused the displays into a single three-dimensional percept for the user, creating a Wheatstone stereograph (see figure 2 in ref. 16).

Central results were replicated on multiple days in each monkey, using 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 from center. Training sets were typically composed of about 500 (peripheral and central) target acquisitions. All of the test sets shown in **Figure 1** were collected using a standardized target configuration, with eight peripheral targets uniformly placed 8 cm away from the central target with either 5-cm or 6-cm acceptance windows.

Signal acquisition and conditioning. Neural signals were acquired from an implanted 96-channel Utah Microelectrode Array (Blackrock Microsystem) using the Cerebus Recording System (Blackrock Microsystems). An analog band-pass filter with a 0.3 Hz to 7.5 kHz pass-band wass applied to each channel. Channels were sampled at 30,000 samples s^{-1} and are filtered with a 250 Hz to 7.5 kHz digital band-pass filter. A threshold detector was applied to each band-passed channel. The threshold value was set automatically as −4.5 times the measured root mean squared value of the channel. When the signal value was less than threshold, a spike event registered for that channel. The number of spike events was counted in nonoverlapping temporal bins (typically 50 ms). The counts for each channel were 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 was used to provide a metric that normalizes across task difficulty. For reference, monkey L was implanted on 22 January 2008 and monkey J was implanted on 24 August 2009. Data for monkey L were collected on 182 sessions over 29 months (from 24 months to 53 months after implantation). Data for monkey J were collected on 98 sessions over 16 months (from 5 months to 21 months after 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 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 time spans 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. As 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 an arm-controlled session for 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 in sessions that initially trained from arm-movement data.

34. Santhanam, G., Ryu, S.I., Yu, B.M., Afshar, A. & Shenoy, K.V. A high-performance brain-computer interface. *Nature* **442**, 195–198 (2006).

doi:10.1038/nn.3265