



Published in final edited form as:

IEEE Trans Neural Syst Rehabil Eng. 2011 April ; 19(2): 193–203. doi:10.1109/TNSRE.2011.2107750.

Point-and-Click Cursor Control With an Intracortical Neural Interface System by Humans With Tetraplegia

Sung-Phil Kim,

Department of Computer Science, Brown University, Providence, RI 02912 USA

John D. Simeral,

Rehabilitation Research and Development Service, Department of Veterans Affairs, Providence, and the School of Engineering, Brown University, Providence, RI 02912 USA

Leigh R. Hochberg,

School of Engineering, Brown University and Rehabilitation Research and Development Service, Department of Veterans Affairs, Providence, RI 02912 USA and with the Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA

John P. Donoghue,

Rehabilitation Research and Development Service, Department of Veterans Affairs, Providence, and the Department of Neuroscience, Brown University, Providence, RI 02912 USA

Gerhard M. Friehs, and

Department of Clinical Neuroscience, Brown University, Providence, RI 02912 USA

Michael J. Black

Max-Planck Society, 72012 Tübingen, Germany, and also with the Department of Computer Science, Brown University, Providence, RI 02912 USA

Sung-Phil Kim: spkim@cs.brown.edu; John D. Simeral: john_simeral@brown.edu; Leigh R. Hochberg: leigh_hochberg@brown.edu; John P. Donoghue: john_donoghue@brown.edu; Gerhard M. Friehs: gfriehs@yahoo.com; Michael J. Black: black@cs.brown.edu

Abstract

We present a point-and-click intracortical neural interface system (NIS) that enables humans with tetraplegia to volitionally move a 2-D computer cursor in any desired direction on a computer screen, hold it still, and click on the area of interest. This direct brain–computer interface extracts both discrete (click) and continuous (cursor velocity) signals from a single small population of neurons in human motor cortex. A key component of this system is a multi-state probabilistic decoding algorithm that simultaneously decodes neural spiking activity of a small population of neurons and outputs either a click signal or the velocity of the cursor. The algorithm combines a linear classifier, which determines whether the user is intending to click or move the cursor, with a Kalman filter that translates the neural population activity into cursor velocity. We present a paradigm for training the multi-state decoding algorithm using neural activity observed during imagined actions. Two human participants with tetraplegia (paralysis of the four limbs) performed

© 2011 IEEE

S.-P. Kim is now with the Department of Brain and Cognitive Engineering, Korea University, Seoul, South Korea

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Disclosure: J. P. Donoghue was the Chief Scientific Officer and a Director of Cyberkinetics Neurotechnology Systems (CYKN). He held stock and received compensation. J. D. Simeral was a consultant for CYKN. L. R. Hochberg received research support from Massachusetts General and Spaulding Rehabilitation Hospitals, which in turn received clinical trial support from CYKN. The contents do not represent the views of the Department of Veterans Affairs or the United States Government.

a closed-loop radial target acquisition task using the point-and-click NIS over multiple sessions. We quantified point-and-click performance using various human-computer interaction measurements for pointing devices. We found that participants could control the cursor motion and click on specified targets with a small error rate (<3% in one participant). This study suggests that signals from a small ensemble of motor cortical neurons (~40) can be used for natural point-and-click 2-D cursor control of a personal computer.

Index Terms

Amyotrophic lateral sclerosis; human motor cortex; intracortical neural interface system; multi-state decoding; point-and-click control; quadriplegia; stroke

I. Introduction

Recent advances in neurotechnology have established the potential for creating direct connections between the human brain and external devices to restore communication or environmental control to people with paralysis. The development of these neural interface systems (NISs) has been facilitated by an understanding of the cortical control of movement, the emergence of new sensors in the form of implantable microelectrode arrays, and progress in mathematical modeling for decoding of neural population signals. An intracortical NIS is a form of brain-computer interface (BCI) that can provide new, neurotechnology-based, connections between the central nervous system and external devices such as computers, robotic actuators, and functional electrical stimulation systems. In the last decade, several research groups have demonstrated the feasibility of NIS in able-bodied nonhuman primates through experiments involving the closed-loop neural control of arm muscles, computer cursors, and robotic manipulators [1]–[7]. The majority of human BCIs have relied on the sensing of field potential signals recorded on the scalp using electroencephalography (EEG) [8]–[10] or electrocorticography (ECoG) signals directly from the cortical surface [11], [12].

Here our goal is to enable continuous point-and-click operation of a computer using the NIS equivalent of an ordinary computer mouse. In able-bodied humans, such control is achieved through the movement of the arm and hand. Arm actions such as these are generated by spiking potentials within the arm area of motor cortex (MI_{arm}). We have used a chronically implanted floating microelectrode array, developed for human applications, from which we record action potentials of multiple MI_{arm} neurons. The initial report of pilot clinical trial results with this device in two study participants demonstrated the feasibility of a human NIS for controlling prosthetic devices [13]. In particular, the data from these participants demonstrated that movement-related spiking activity of a neuronal ensemble could be detected in human motor cortex years after paralysis, modulated by imagined movements, and decoded for volitional control of prosthetic devices such as a simple robotic hand and a computer cursor.

The computer cursor control from the earlier study [13], however, did not reach the smoothness, movement accuracy, or selection precision that is readily achieved by able-bodied people using a standard computer mouse. A more recent study from our group addressed the problems of smoothness and movement accuracy, using a velocity-based Bayesian decoding method (Kalman filter) [14]. This work leveraged findings in previous nonhuman studies of 2-D directional tuning in MI_{arm} neurons [15]–[19] to decode neural signals by probabilistically integrating the likelihood of neural spike observations with a temporal prior over cursor movement [20]–[22]. It was shown that decoding cursor velocity

using the Kalman filter produced straighter and more accurate cursor movements than those obtained in the initial study.

The problem of full point-and-click control with an NIS, however, has remained unsolved. In the previous human NIS studies [13], [14], target selection was implemented by placing the cursor on top of a target for more than a fixed time period (e.g., 500 ms). The limitations of such an interface are obvious for practical computer applications where multiple other “targets” may be present at once and slow cursor motions may result in the inadvertent selection of targets. All practical pointing device interfaces (e.g., mouse, trackball, touch pad) employ an explicit and distinct action to effect selection or “clicking.”

A number of nonhuman primate studies have focused on decoding discrete states corresponding to a set of targets presented on the screen [4], [5]. Yu *et al.* [22] extended this discrete decoding approach using a Bayesian mixture model to decode both an intended reach direction and the continuous hand movement to that direction. Srinivasan *et al.* [23] also derived a state-space algorithm that modeled the relationship between target location and the path of the arm to reconstruct arm trajectory. In non-human closed-loop BCI studies, monkeys used their motor cortical signals to move a 2-D computer cursor and select a target [3] or to control a 3D robotic arm and a gripper to reach and grasp food [6].

To enable point-and-click control using human motor cortical signals, we built on the approach developed by Wood *et al.* [24] in which both continuous and discrete states were decoded simultaneously from MI activity of monkeys. They used a nonlinear particle filtering method [25] to infer discrete (performing/not-performing) and continuous (2-D cursor kinematics) states from a single neuronal population. In related work, Darmanjian *et al.* [26] discriminated movement versus nonmovement states from a monkey’s motor cortex and predicted the hand position of the monkey using linear filters. For real-time implementation, we modified the multi-state decoding model of Wood *et al.* [24] to avoid modeling the nonlinear joint probability of continuous and discrete state variables. As in Wood *et al.*, we used a linear classifier based on Fisher discriminant (FD) analysis [27] to classify the neural population activity into distinct states (click versus movement). A key difference, however, is that our model discriminated the neuronal firing patterns related to two actively imagined motions while the previous models only detected whether or not a firing pattern for a specific movement was elicited. Hence, the click signal decoded from our model was not simply a result of decreased movement intention but, rather, a result of voluntary click intention.

II. Methods

A. Participants and Recordings

The two participants studied here were enrolled in clinical trials of the BrainGate NIS (Cyberkinetics Neurotechnology Systems, Inc. (CYKN), Foxborough, MA)¹. One participant (S3) was a 55-year-old female with tetraplegia caused by brain-stem stroke that occurred nine years prior to enrollment. The other participant (A1) was a 37-year-old male with paralysis resulting from amyotrophic lateral sclerosis (ALS, Lou Gehrig’s Disease). A 96-channel microelectrode array (BlackRock Microsystems, Salt Lake City, UT) was chronically implanted in a cortical arm area of participant S3 in November 2005 and A1 in February 2006. During the trial sessions, electrical neural signals were recorded using the microelectrode array. Recorded signals were digitized at 30 kHz and processed by commercially-available amplitude-thresholding and spike-sorting software [Cerebus Central,

¹Caution: Investigational Device. Limited by Federal Law to Investigational Use. Cyberkinetics ceased operations in 2009. The BrainGate pilot clinical trials are now directed by Massachusetts General Hospital (NCT00912041).

Blackrock Microsystems, Salt Lake City, UT] to discriminate individual waveforms. Then the spiking activity of these putative single or multiple neurons (referred to as “units” hereafter) were examined online by visual inspection by a technician who determined their inclusion in the session. We did not analyze the signals further to determine whether or not each unit contained single or multiple neurons. This process was performed at the beginning of every session to isolate the units used to build the decoder and control the computer cursor. More detailed information about the chronic neural signal recording and processing can be found in Hochberg *et al.* [13].

There is a possibility that the same units were recorded across multiple sessions. However, in this study, we did not examine the consistency of spike waveforms and other statistical properties of the recorded units across sessions. The spacing of the recording electrodes made it unlikely that multiple electrodes recorded the same neuron. The units were spatially distributed across the electrode array over multiple sessions with the exception of a few electrodes that did not show any spike waveforms. A few of these exhibited high impedance compared to other electrodes, suggesting some form of failure of those electrodes.

For this paper, we studied four point-and-click neural control recording sessions in participant S3 which were recorded on trial days 292 ($N = 37$ units), 301 (38), 303 (57), and 464 (28) after implantation of the array, and one recording session in participant A1, on trial day 231 (86). Here, the numbers in parentheses indicate the number of units used for training and testing the NIS on each day. Note that A1 participated in two other point-and-click study sessions on different trial days (day 232 with 99 units and day 239 with 90 units) but could not perform the full radial target acquisition task due to failure to obtain reasonable cursor control within the time limit of the session (1.5–2 h). A1 achieved good 2-D cursor control in our previous study [14], but the instructed tasks were different than those studied and compared here. All sessions with S3 that met our criteria (same tasks, algorithms, training paradigm, and imagined click action) were included in this study.

B. Training and Cursor Control Tasks

1) Decoder Training Procedure—The purpose of the decoder training procedure was to generate data used to estimate the parameters of the decoding algorithms. Decoder training involved mathematically modeling the relationship between desired cursor movement and the corresponding neural activity. Since physical arm/hand movement was precluded (A1), or minimal and of no functional relevance (S3), in these participants with tetraplegia, we generated desired cursor movements using a *training cursor* (TC) and instructed the participants to *imagine* controlling the TC [13], [28]. Both the TC movement signals and the neural signals were recorded and defined the training data. The overall training procedure was composed of two phases: continuous state and discrete state.

The training procedure began with the continuous-state training phase. This phase included multiple training “blocks,” where a block was a short training period (1.5 min). These blocks were categorized as open-loop (OL) blocks or closed-loop (CL) blocks. In an OL block, the participant was shown the TC on a computer monitor and instructed to imagine moving their arm or hand as if they were controlling the TC. After OL blocks, the synchronously recorded neural signals and the TC movement signals were used to build a continuous-state decoder. Then, a series of CL blocks followed in which a separate feedback cursor (FC) was shown together with the TC. The FC was moved by the participant’s neural activity using the current decoder. The FC provided visual feedback of controllability to the participants [14].

The TC was moved by a computer program to perform a center-out-and-back task [14]. Eight targets were radially distributed on the screen with angles, $\{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$ (where 0° refers to movement to the right and increases

counterclockwise). There was also a target at the center to represent a “home” position. The TC moved from the center target to one of eight radial targets and then traced its trajectory back to the center. The TC visited every radial target in a randomized order. The duration of the TC movement from target to target varied across blocks and sessions (1.5–4.5 s).

The second training phase was designed to train the discrete-state decoding algorithm. This phase was composed of multiple discrete state (DS) blocks. Each 1.5-min DS block began with an epoch (~ 20 s) of continuous CL cursor control. After this epoch, the screen became blank for 0.5 s. Then, an instructive word for a “click action” was displayed at the center of the monitor for 1.5 s. The click action was defined by an imagined arm/hand movement that could substitute for clicking. From these experiments, we chose the imagined motion of squeezing the right hand as the click action for S3; A1 preferred to imagine opening the right hand. Such an imagined click action appeared to modulate neural firing patterns that were well separated from those modulated during continuous cursor control. The on-screen text (“squeeze” for S3 or “hand open” for A1) provided a visual cue to the participant to imagine the click action. Then, a blank screen appeared again for 1 s followed by the continuous CL control epoch. This sequence of continuous and discrete state presentations was repeated four to five times in each DS block. Afterwards, we collected the neural data samples not only from two DS blocks but also from the previous two to four CL blocks and labeled each data sample as either the movement state or the click state. The neural signals recorded during the blank screen periods were not used for training any decoding algorithm. The total duration of the filter training data was approximately 11.6 min, which took approximately 20 min to acquire, including breaks between blocks and filter building time. See Fig. 1 for the overall flow of the multi-state training sequence.

2) Closed-Loop Cursor Control Task—After training was completed, the parameters of the decoding algorithms were fixed and the point-and-click cursor control performance was evaluated using a similar eight-target center-out-back task without the TC. On the screen, there were eight circular targets arranged radially around a circular center target as well as a circular neural cursor (NC) whose kinematics were decoded from the participant’s neural activity. As the 10-min evaluation period began, the NC appeared on top of the center target and all the targets appeared in gray. Then, one of the peripheral targets changed its color to pink to provide a visual cue indicating the target. To acquire the target, the participant moved the NC towards the target and clicked when the NC and target overlapped. The period of moving the NC from the center and clicking on a designated target was termed a “run” in this study.

A run could end with three different conditions: 1) the designated target was successfully acquired, turning its color to red; 2) one of the other seven radial targets (excluding the center target) was accidentally acquired, turning its color to blue; 3) no target was acquired before a time limit expired. The time limit varied across the sessions from 9 to 30 s. Clicks that were decoded on the background (i.e., not over any target) were recorded but did not otherwise affect the run.

After a run ended, the center target turned to pink and the participant moved the NC back to acquire the center target. The center target was acquired either by holding the NC on top of the target for more than 500 ms (three sessions of S3 and one session of A1) or by clicking on the target (one session of S3). There was no time limit for the center target acquisition. Once the participant acquired the center target, the next run began with a new target. This task continued for 10 min without interruption. There were a total of 212 runs from five recording sessions across both participants presented in this study (see Table I).

The distance from the participant to the center of the monitor was approximately 59 cm. The visual angle of the workspace on the screen was approximately 34.1°. The diameter of the target was 48 pixels (visual angle = 2.0°) where the screen resolution was 800 × 600 pixels (17-in monitor). The diameter of the TC, the FC and the NC were all 30 pixels (= 1.3°). The distance to target was 300 pixels (12.7°) from center to the horizontally located targets, 255 pixels (10.8°) to vertically located targets and 278 pixels (11.8°) to diagonally located targets; these were all center-to-center distances between targets.

C. Decoding Model

Let \mathbf{z}_k be an $N \times 1$ vector of the firing rates of N units measured at discrete time index k , \mathbf{x}_k be a 2-D cursor velocity vector at k , and γ_k be a binary random variable representing whether the cursor was in a click or movement state (i.e., $\gamma_k \in \{\gamma^{(0)}, \gamma^{(1)}\} \equiv \{\text{click}, \text{movement}\}$). The firing rate of each unit was represented by the number of spikes within a non-overlapping time window (100 ms). From the firing rates of N units over L time windows, we created a vector $Z_k = [\mathbf{z}_k^T, \mathbf{z}_{k-1}^T, \dots, \mathbf{z}_{k-L+1}^T]^T$ that was a $(N \cdot L) \times 1$ short-time history vector of the firing rates. We empirically selected $L = 5$ (i.e., 0.5 s) which we found provided reasonable discrete-state classification performance.

To decode a discrete state, we first represented the *a posteriori* probability of γ_k conditioned on the history of the firing rates Z_k and the previous discrete state γ_{k-1} as in [24]

$$p(\gamma_k = \gamma^{(c)} | Z_k) = \frac{p(\gamma_k = \gamma^{(c)}) G(\mathbf{w}^T Z_k; \mu_c \sigma_c)}{\sum_c p(\gamma_k = \gamma^{(c)}) G(\mathbf{w}^T Z_k; \mu_c \sigma_c)}, \quad c \in \{0, 1\}. \quad (1)$$

$G(\alpha; \beta, \sigma)$ here denotes a univariate Gaussian distribution of a random variable α with a mean β and a standard deviation σ . The vector of weights, \mathbf{w} , was used to project Z_k onto a 1-D feature space. The parameters μ_c and σ_c are the mean and standard deviation of the Gaussian probability density function (pdf) of a 1-D variable ($= \mathbf{w}^T Z_k$) in the feature space for class c .

The output from the FD analysis was multiplied by the probability, $p(\gamma_k | \gamma_{k-1})$ that described the probability of discrete state transitions from time $k-1$ to k , which was empirically estimated from the training data. Finally, the state at k was decoded as “click” if

$$\frac{p(\gamma_k = \gamma^{(0)} | Z_k, \gamma_{k-1})}{p(\gamma_k = \gamma^{(1)} | Z_k, \gamma_{k-1})} > t \quad (2)$$

or “movement” otherwise, where $0 < t < \infty$ was the decision threshold. The threshold t was set to 1 in our study by assuming that each state was likely to occur with the same probability. We observed that overall the posterior probability ratio in (2) was well below 1 for the movement state and well above 1 for the click state. It is possible that optimizing this threshold could improve discrete-state decoding accuracy.

For continuous state decoding, cursor velocity ($\mathbf{x}_k = [v_k^x v_k^y]^T$) was predicted from neural ensemble firing rates (\mathbf{z}_k) using a Kalman filter [14], [21]. Here v_k^x and v_k^y denote the x- and y-coordinates of the velocity vector at time k . The Kalman filter modeled the firing rates (\mathbf{z}_k) as a linear function of velocity (\mathbf{x}_k) corrupted by Gaussian noise and the current velocity at time instant k (\mathbf{x}_k) as a linear function of the previous velocity at $k-1$ (\mathbf{x}_{k-1}) corrupted by Gaussian noise. The Kalman filter ran in real time to estimate velocity from the entire observation history of firing rates using recursive Bayesian inference and has been

successfully used in human neural cursor control [14]. The details of the Kalman filtering model can be found in [21].

Both the discrete and continuous state decoders were trained following the training procedure described in Section II-B1. The Kalman filter parameters were first trained with the OL block data using least squares (LS) estimation [21]. Then, we used the CL block data to update the Kalman filter parameters using LS to cope with the possible changes in neural activity resulting from closed-loop feedback. Finally, the optimal projection vector \mathbf{w} was estimated from all the DS block data using FD analysis. The generalization performance of these models was evaluated using the testing block data.

Since cursor velocity (\mathbf{x}_k) and the discrete state (γ_k) were independently inferred, we decoded both of them in parallel and combined the results. When $\gamma_k = \gamma^{(1)}$, the NC was moved with velocity \mathbf{x}_k estimated from the Kalman filter. When $\gamma_k = \gamma^{(0)}$, the NC was forced to stop (i.e., $\mathbf{x}_k = [0 \ 0]^T$) and a click state was generated. To reduce click errors due to noisy classification, we temporally integrated the click state estimates over a period of time (typically 500 ms). Specifically, if the click state was consistently decoded over the fixed time interval, a click signal was generated. This time interval was set to 500 ms corresponding to the length of the firing rate history for all the recording sessions except the one on day 464 of S3, where no integration interval was applied and the click signal was generated whenever the click state was decoded.

D. Analysis Methods

1) Tuning Analysis—Using the training data, we analyzed how each neuronal unit modulated its firing rate with respect to the discrete and the continuous states.

First, we evaluated neural tuning to the discrete state by testing the change of firing rates between discrete and continuous training phases. Specifically, we tested whether the firing rate distribution during DS blocks was significantly different from that during OL and CL blocks, using a Kruskal-Wallis (KW) test ($p < 0.01$). Units exhibiting a significant difference in the firing rate distribution were regarded as tuned to the discrete state.

Second, tuning of each unit to the continuous state was evaluated using a multivariate linear regression model

$$z_k^i = \beta_0 + \beta_1 v_k^x + \beta_2 v_k^y + \varepsilon_k \quad (3)$$

where β_j 's were linear coefficients and ε_k was noise. The parameters β_j were estimated from the training data using least squares estimation. We tested whether the coefficients β_1 and β_2 were significantly different from zero using the F-test ($p < 0.01$). If one or both of them was different from zero, we considered the unit as being tuned to velocity.

2) Control Performance Evaluation—We quantified the performance of neural point-and-click cursor control using two types of measures: gross measures and fine measures.

Gross measures evaluated speed and accuracy of the overall point-and-click performance for each session. Specifically, we adopted two measures defined in the ISO 9241-9 standard [29] for evaluating non-keyboard pointing devices. Speed was evaluated using movement time (MT), which measured how long on average it took to move the NC to a target and click on it. Note that MT was only measured for successful target acquisition runs; for targets that were not acquired, MT was undefined. Accuracy was evaluated using the error rate (ER), which measured the percentage of the runs where a target was missed. A target

could be missed because either a time limit expired or a false target was selected. We refer to the first type of error as “ER by timeout” (ER-to) and the second as “ER by false selection” (ER-fs), respectively.

Fine measures evaluated the details of individual cursor trajectories using several metrics proposed by MacKenzie *et al.* [30]. Consider an optimal straight path (termed a task axis) connecting the starting point of a NC movement to the center of a target. The actual NC movement path was evaluated against the task axis using four metrics: 1) orthogonal direction change (ODC) counts the changes in the direction orthogonal to the task axis, 2) movement direction change (MDC) counts the changes in the direction parallel to the task axis, 3) movement error (ME) measures the mean distance of the NC from the task axis, and 4) movement variability (MV) measures the standard deviation of the offset of the NC from the task axis. Overall, ODC represents how consistently the NC moves forward to the target, MDC represents how smooth the NC path is, ME represents how much the NC path deviates from the optimal straight path, and MV represents how straight the NC path is. See MacKenzie *et al.* [30] for details.

3) False Click Analysis

In addition to the above measures of continuous cursor control, we also evaluated the accuracy of discrete state decoding. First, we defined a false click rate (FCR), which measured how many times per run the click state was decoded on the background space before the NC reached the target. It is worth noting that the participants were not instructed to avoid clicking on the background. False clicks on the background space could be *unintentionally* generated due to decoding error or *intentionally* generated but in the wrong place. Intentional errors could occur, for example, when the cursor was close to, but not quite overlapping, the target. We analyzed these different possible causes of false clicks by assuming that if all the false clicks had been caused only by decoding error, false clicks would have occurred regardless of whether the NC was close to the target or not. Hence, we computed the Euclidean distance between the NC and the target when every click signal was generated (including all false and successful clicks). Then, we binned all those distance measurements with a bin size of 39 pixels (1.7°). The bin size was determined to be the sum of the target radius (24 pixels) plus the NC radius (15 pixels). This ensures that all successful clicks fell in the bin closest to the target and all other false clicks in the remaining bins (see Fig. 5).

III. Results

A. Neural Tuning Analysis

We investigated how the spiking activity of a neuronal population was tuned with respect to the discrete state (click versus movement) and the continuous state (cursor velocity). We analyzed the discrete state tuning using the DS training blocks and the directional tuning using the final CL training blocks that were used to build the final Kalman filter for testing. We present the results for the tuning of a total of 160 units recorded during four sessions with S3 and 86 units from one session with A1.

We categorized the observed firing patterns into four groups. Group 1 units statistically changed their firing rates between the movement and click state and were not tuned to the TC velocity [Fig. 2(A)]. Group 2 varied their firing rate with respect to the TC velocity but did not show significant tuning to the discrete state [Fig. 2(B)]. Groups 3 and 4 exhibited firing activity correlated with both the TC velocity and the discrete state. While both groups exhibited tuning to the TC velocity, they differed in their tuning to the discrete state. Group 3 significantly increased the firing rate [Fig. 2(C)] while group 4 decreased the firing rate

[Fig. 2(D)] during the click state. We examined the firing patterns of these four groups during click training (Fig. 2, second row) and during the closed-loop point-and-click task (Fig. 2, third row), in which the cursor was under active control of the participant, with visual feedback from the TC. We observed similar firing patterns with respect to the click state in both cases. This suggests that the neural population firing activity was modulated by an abstract instruction of imagined motion (e.g., text on screen) without explicit visual movement cues, and that the participant utilized this imagined motion to volitionally modulate the same neural population to generate a click signal during active control.

We quantified the proportion of recorded units in each of the four groups using the statistical testing methods described above (see Section II-D). For S3, 68.1% (109/160 units) were tuned to the continuous state (i.e., TC velocity) (F-test: $p < 0.01$), 56.9% (91 units) were tuned to the discrete state (KW-test: $p < 0.01$) and 41.9% (67 units) were tuned to both (Fig. 3). This means that 83.1% (133 units) were tuned either to the continuous or discrete state. For A1, 65.1% (56/86 units) were tuned to the continuous state, 31.4% (27 units) were tuned to the discrete state, 23.3% (20 units) were tuned to both, and 73.3% (63 units) were tuned to at least one state (Fig. 3). It is noteworthy that here we evaluated the neural tuning with respect to cursor velocity only. Previous studies have shown that the neural populations of S3 and A1 were also tuned (to a lesser extent) to cursor position [14], [31]. Hence, the percentage of the neural populations tuned to any continuous state variable might have been larger if we had included other kinematic parameters.

Finally, it is worth noting that filter building time was fairly rapid. Filter building time taken over all blocks ranged from 10.5 to 13.5 min for the four S3 sessions (mean 11.6 ± 1.4 min). Building time for A1 was 17.5 min in the single session reported here. This excludes breaks between blocks.

B. Point-and-Click Neural Cursor Control

We illustrate continuous NC movements in a radial point-and-click task by showing the paths of the NC movement from the center area towards each of the eight targets. All the NC paths during a 10 or 15 min long control session, regardless of whether or not the target was successfully acquired, are illustrated in Fig. 4 for each of five recording sessions. In each case the NC was under continuous and uninterrupted control by the participant, although the illustration only shows the path out to the target from the center. Recall that each motion required that the NC be placed at the center target by the participant. We observed from these NC paths that the NC movements of A1 were less accurate than those of S3. In the sole data set from A1, the NC movements showed a bias towards the lower-left corner of the screen (see Section IV for further discussion). Our earlier study, which only considered cursor movement, also found that A1 had longer, less straight and more biased cursor motions than S3 [14]. These differences can also be seen in the average paths of the NC to each target (Fig. 4). To obtain the mean path for each target, we linearly interpolated individual NC trajectories to match the number of time samples. Then, we calculated the sample mean at every point. Despite having poorer 2-D control than S3, A1 was able to reach and click on targets using neural cursor control.

We assessed the point-and-click performance using standard pointing device evaluation measures (see Section II-D2) summarized in Table I. S3 successfully acquired 97.4% of the targets in 193 runs over four sessions. Mean trial duration was 7.2 ± 3.8 s to achieve continuous movements from the center to the target and then click on the target, where the mean distance to the target was 278 pixels (12 cm; visual angle = 11.8°). The 2.6% error in completing a trial for S3 was attributed fully to targets not clicked before the timeout period expired (ER-to = 2.6%); no false target was clicked (ER-fs = 0%). A1 acquired 52.6% of targets correctly. Again, all the errors were due to failing to complete a trial before the

timeout (ER-fs = 0%). The performance difference between the participants was larger in the continuous control measures including ODC, MDC, ME, and MV than in the discrete control measures including FCR.

The quantification above was obtained excluding the NC movements from the peripheral targets towards the center target because for the most sessions, the center target was selected by holding the NC on it for >500 ms instead of clicking on it. This indicates that the participants could select a target on the screen by either regulating the NC speed to hold it on the target or by generating a click. We examined the discrete decoding outputs during the periods when the center target was selected and observed that there was no significant increase of clicks when the participants attempted to hold the NC on the center target, showing that click generation was independent of the NC speed.

To investigate how much time was spent attempting to click on a target, we divided movement time (MT) into “pointing time” and “clicking time”: the former measured the average time taken to move the NC from a starting point to a target while the latter measured the average time taken to click on the target once the cursor reached it. The ratio of pointing time to clicking time was 4.7 s to 2.5 s for S3 and 7.4 s to 6.9 s for A1. Note that clicking time included 0.5 s for the temporal integration of consecutively decoded click states (in all but one session).

C. Discrete Cursor Control Performance

We found that a false click occurred on average 0.74 times per run in S3 and once per run in A1 (Table I). The participants were not explicitly instructed to avoid false clicks and there was no overt “penalty” for false clicks on the background.

For click histograms, we examined whether false clicks were related to the distance between the NC and the target, which would suggest anticipatory clicking when in the vicinity of a target of interest. First, we evaluated all the NC positions when either a correct or a false click signal was generated, from four sessions of S3 and one session of A1 [Fig. 5(A)]. To visualize the data from all eight targets in one plot, we rotated the task axis (a straight line between the starting point (e.g., center) and the target) for each target to the vertical axis and marked the NC positions relative to the rotated task axis. Next, we measured the distance between each NC position and the target and computed a histogram of these distances for each participant [Fig. 5(B)]. This histogram describes the occurrence of click as a function of distance to target. It shows that for S3, most false clicks occurred when the NC was near the target, suggesting that most false clicks were generated when target acquisition was anticipated but the NC did not exactly overlap the target. We also examined when the first click occurred in each run [Fig. 5(C)]; these also occurred near the target. This first-click analysis demonstrates that S3 successfully performed point-and-click target selection with no false clicks for more than 50% of runs while A1 did so for more than 25% of runs. Note that all false clicks corresponded to clicking on the “background” space on the monitor. There was no instance of clicking over incorrect targets during 212 runs in two participants (ER-fs = 0%).

We also analyzed whether false clicks were correlated with the direction of the NC movement. We hypothesized that the modulation of the units that were tuned to both the continuous state (direction) and the discrete state (click) could cause the decoding algorithm to misinterpret the movement intention as the intention to click, particularly when moving in the preferred directions (PDs) of those “multi-state tuned” units. We computed a histogram of the *intended* direction, which was defined as the instantaneous direction from the NC to target [31], measured at the time of every false click [Fig. 6(A)]. We then computed another histogram of the PDs of the multi-state tuned units [Fig. 6(B)]. If these two histograms

showed similar distributions, it would support the hypothesis that the multi-state units could lead to misclassification in the discrete-state decoding. We found, however, that the histogram of the intended directions was distinct from that of the PDs (Fig. 6). This suggests that false clicks might be more likely to occur in directions where a relatively small number of units modulated their firing rates, thus possibly increasing uncertainty about the movement state for the discrete-state decoding algorithm.

IV. Conclusion and Discussion

We have developed an intracortical NIS that enabled persons with tetraplegia to move a computer cursor to an arbitrary position on the screen, stop the cursor, click on the area of interest and move to another position, without any interruption, automated recentering of the cursor or other external intervention used in previous BCIs. One participant (S3) was able to smoothly control cursor velocity and switch between cursor movement and clicking using different imagined movements. The other participant (A1) was able to move the cursor and click on targets, but cursor velocity control was relatively poor in the single session appropriate for analysis. A key advance of this study is the demonstration of online decoding of both continuous and discrete motor signals from a small population of motor cortical neurons in the dominant arm-hand area of humans with tetraplegia. The decoding model combines two methods: Fisher discriminant analysis and the Kalman filter. The simplicity of these two methods makes real-time implementation practical and decoder training easy. We modified previous filter building methods [13], [14] for users with paralysis by incorporating a new discrete state training method into the paradigm. We found that some particular imagined motions (e.g., squeezing the hand for S3 or opening the hand for A1) modulated motor cortical activity in a way that could be readily discriminated from continuous cursor movements and, therefore, could be used to generate a click signal. With well-established measurements for the effectiveness of non-keyboard pointing devices [29], [30], we have quantified the point-and-click performance of two participants (including one over multiple recording sessions). This assessment approach provided more comprehensive and practical measures of cortical cursor control than conventional measures such as mean squared errors or correlation coefficients. We believe that these measures will be useful for future BCI research, for which there is no currently accepted set of standard performance metrics [32].

We found motor cortical units that exhibited distinct tuning to both cursor direction and “click” movements. We cannot rule out the possibility, however, that these units were, in fact, multi-units of two distinct neurons, each being tuned to a different movement feature, because mixtures of neurons can occur with these fixed electrodes. Importantly, other work has found that hand and arm information is intermingled within the MI arm region, as sampled by this sensor [13], [33]. From the perspective of neural prosthetic development, however, the important observation is that such a classification is possible using only this small population of cells.

It would be of practical interest to know how many neurons are needed to achieve reasonable control of both pointing and clicking. One possible way is to address this point is performing the neuron dropping analysis [3] in which we remove one or more neurons at a time from the decoder and evaluate the effect on accuracy during closed-loop control. As we observed diverse groups of neurons tuned to velocity, clicking or both, the neuron dropping analysis may be done for each tuning group independently. Of course, the better-tuned units are likely to contribute more to generating correct control signals. If there is a linear relationship between the tuning depth and the contribution to control, we might be able to predict the level of cursor control from the degree of the tuning of all recorded units even before executing a closed-loop experiment. This could be useful for determining what level

of control may be possible and thus allow a technician to change properties of the interface to suit the conditions (e.g., making the targets larger or smaller, or changing the integration window for click decoding).

2-D cursor control performance with the intracortical NIS shown in this study differs from previous BCI studies [8], [9]. After our initial report on point-and-click cursor control in a human [34], McFarland *et al.* demonstrated 2-D cursor control with target selection using EEG signals in humans [9]. Besides fundamental differences in cortical signals, an important difference is the way the cursor was controlled. First, we simultaneously decoded both continuous cursor kinematics and click signals from the same neural ensemble while McFarland *et al.* operated cursor control in two separate modes—continuous cursor movement and discrete target selection—and the switch between modes was conducted based on non-neural signals (i.e., whether or not a target was reached). Second, in our study, the cursor was always present and had to be held on the target by the participant until the target was selected. In McFarland *et al.*, the cursor movement was frozen once it came in contact with a known target location. Hence, the user could focus only on the target selection with no need to control the cursor movement. Third, we did not use an automatic recentering of the cursor in the workspace while most previous studies have relied on this. Fourth, McFarland *et al.* updated the decoder parameters using information about the target after acquisition. However, in a natural cursor control task, the true target is unknown, making such updating infeasible in practice. Consequently, we trained the decoder parameters and then held them fixed during testing.

A1 participated in three point-and-click sessions through the entire pilot trial and did not achieve point-and-click control in two of them. In these two sessions, the neuronal ensemble did not modulate well with cursor direction or click state during training, so we ended the session without evaluation. In the single session reported here, A1 achieved some level of point-and-click control. A1's click performance was only slightly worse than S3 (1 versus 0.74 of FCR), showing that A1 could volitionally generate clicks when intended. A1's velocity control was quantitatively worse than S3's, resulting in an overall 47.4% error rate; this error rate was due entirely to runs ending as a result of the timeout. This error rate was, however, roughly comparable to previous studies of 2-D velocity control in which A1 had a 31.8% error rate on average with a simpler four-target task [14]. With a more complicated cursor control task (smaller targets, click selection and all targets displayed) as in this study, the roughly 17% increase in error rate is not surprising. However, it is an obvious problem that the neural cursor tended to move in a particular direction (i.e., lower-left corner of the screen). The cause of the movement bias to the lower-left portion of the screen is not known (see [14] for a discussion of related issues). Current work is exploring the possible sources of performance variability. It is noteworthy, however, that control was achieved in participants with both ALS, a progressive neurodegenerative disease, and after long-standing stroke.

Although the cursor was controlled reasonably well by motor cortical activity, there is still significant room to improve control, which might be accomplished by improving the encoding and decoding models. For instance, the maximum speed of the neural cursor was markedly slower than cursor speed generated by able-bodied users. We hypothesize that this slower cursor movement might be in part caused by the linearity assumption in our decoding model. Future work should explore nonlinear models as well as acceleration decoding. To better discriminate movement and click states, we are currently considering more accurate classification methods such as a logistic regression, a Bayes classifier with reduced input dimensions, or a support vector machine with a linear kernel. A more sophisticated temporal model of changes in discrete state, such as a hidden Markov model, might improve the FCR.

We also aim to determine the optimal classification threshold for our current classifier from training data using cross-validation.

In our study, a click state meant that the cursor came to a stop. However, it is possible to define the discrete state (γ_k) in different ways for different computer applications. For instance, if a given application requires dragging an item from one location to another on the screen, we should decode both the velocity and the click at the same time. In that case, we can simply change the outputs of the discrete state such that $\gamma^{(1)}$ keeps the cursor moving and $\gamma^{(0)}$ simply generates a click without stopping the cursor. As the imagined motions for the continuous and discrete states are made independently, we can adjust the discrete state outputs flexibly to the user applications.

The demonstration of reliable and accurate point-and-click cursor control provides further evidence that even the use of a single implanted microelectrode array might be useful for neural cursor control applications. The development of a natural, reliable, and fast point-and-click interface for people with tetraplegia would be extremely valuable. A reliable control signal could be used to operate most computer software as well as commercial assistive technology. This study, together with [14], presents the first demonstration of an intracortical point-and-click cursor neural interface used by humans with tetraplegia. The results provide a proof of concept that a direct neural interface system can provide people with paralysis the ability to achieve continuous control and make goal selections using a small ensemble of MI neurons even when paralyzed for years.

Acknowledgments

The authors would like to thank M. Fritz, A. Caplan, and E. Peterson for clinical trial maintenance and assistance. The authors would also like to thank K. Severinson-Eklundh for advice about the evaluation of pointing devices.

This work was supported in part by NIH-NINDS R01 NS 50867-01 as part of the NSF/NIH Collaborative Research in Computational Neuroscience Program, by NIH grant, R01DC009899, N01HD53403 and RC1HD063931, and by the Office of Naval Research (award N0014-04-1-082). This work was also supported by the European Neurobotics Program FP6-IST-001917. This work was based upon work supported in part by the Office of Research and Development, Rehabilitation R&D Service, Department of Veterans Affairs, and support from the Doris Duke Charitable Foundation and the MGH-Deane Institute for Integrated Research on Atrial Fibrillation and Stroke. This work was also in part supported by WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-10008). The pilot clinical trial from which these data were derived was sponsored by Cyberkinetics Neurotechnology Systems, Inc. A preliminary version of this work appeared in the Proceedings of the 3rd IEEE International Conference on Neural Engineering. The current manuscript is significantly expanded, containing neuronal tuning analysis, additional experiments and results with a second participant.

Biographies



Sung-Phil Kim received the B.S. degree in nuclear engineering from Seoul National University, Seoul, Korea, in 1994, and the M.S. and Ph.D. degrees in electrical and computer engineering from University of Florida, Gainesville, in 2000 and 2005, respectively.

He is an Assistant Professor at the Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea. He was a Postdoctoral Research Associate in Computer Science at Brown University, Providence, RI, until 2009. His current research is focused on neural decoding for brain–computer interfaces and statistical signal processing of neural activity.



John D. Simeral received the B.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1985, the M.S. degree in electrical and computer engineering from the University of Texas at Austin, in 1989, and the Ph.D. degree in neuroscience from Wake Forest University School of Medicine, Winston-Salem, NC, in 2003.

He is an Assistant Professor (Research) in the School of Engineering at Brown University, Providence, RI, and a Research Biomedical Engineer for the Department of Veterans Affairs Rehabilitation Research and Development Service. He designed high-performance VLSI microelectronic devices and subsystems for massively parallel computer systems at NCR, AT&T, and Teradata. His current research in the Laboratory for Restorative Neurotechnology integrates engineering, computer science, and neuroscience to translate discoveries regarding how the brain controls movement into practical neural prosthetic systems with the potential to enhance communication and independence for individuals with severe motor disability.



Leigh R. Hochberg is Associate Professor of Engineering, Brown University; Investigator, Center for Restorative and Regenerative Medicine, Rehabilitation R&D Service, Providence VA Medical Center; and Visiting Associate Professor of Neurology, Harvard Medical School. He maintains clinical duties in Stroke and Neurocritical Care at Massachusetts General Hospital and Brigham and Women's Hospital, and is on the consultation staff at Spaulding Rehabilitation Hospital. He directs the Laboratory for Restorative Neurotechnology at Brown and MGH, as well as the pilot clinical trials of the BrainGate2 Neural Interface System. His research is focused on developing and testing implanted neural interfaces to help people with paralysis and other neurologic disorders.

John P. Donoghue received the Ph.D. degree in neuroscience from Brown University, Providence, RI.

He is Henry Merritt Wriston Professor in of the Department of Neuroscience and Director of the Brain Science Institute at Brown University. His research is aimed at understanding neural computations used to turn thought into movements and to produce brain machine interfaces that restore lost neurological functions.

Prof. Donoghue has received a Javits Award (National Institutes of Health) and the 2007 K. J. Zülch Prize (Max Planck/Reemstma Foundation).



Michael J. Black received the B.Sc. degree from the University of British Columbia, in 1985, the M.S. degree from Stanford University, Stanford, CA, in 1989, and the Ph.D. degree from Yale University, New Haven, CT, in 1992.

After postdoctoral research at the University of Toronto, Toronto, ON, Canada, he worked at Xerox PARC as a member of research staff and as an area manager. From 2000 to 2010 he was a Professor in the Department of Computer Science at Brown University, where he is now Adjunct Professor (Research). He is presently a founding director of a new Max Planck Institute, Tübingen, Germany, that studies intelligent systems from molecules to machines.

Dr. Black has won several awards including the IEEE Computer Society Outstanding Paper Award, in 1991, Honorable Mention for the Marr Prize, in 1999 and 2005, and the 2010 Koenderink Prize for Fundamental Contributions in Computer Vision.

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 SIH, 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; 1:192–208.
4. Musallam S, Corneil BD, Greger B, Scherberger H, Andersen RA. Cognitive control signals for neural prosthetics. *Science*. 2004; 305:258–262. [PubMed: 15247483]
5. Santhanam G, Ryu SI, Yu BM, Afshar A, Shenoy KV. A high-performance brain-computer interface. *Nature*. 2006; 442:195–198. [PubMed: 16838020]
6. 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]
7. Moritz CT, Perlmutter SI, Fetz EE. Direct control of paralysed muscles by cortical neurons. *Nature*. 2008; 456:639–642. [PubMed: 18923392]
8. Wolpaw JR, McFarland DJ. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proc Nat Acad Sci USA*. 2004; 101:17849–17854. [PubMed: 15585584]
9. McFarland DJ, Krusienski DJ, Sarnacki WA, Wolpaw JR. Emulation of computer mouse control with a noninvasive brain-computer interface. *J Neural Eng*. 2008; 5:101–110. [PubMed: 18367779]
10. Trejo LJ, Rosipal R, Matthews B. Brain-computer interfaces for 1-D and 2-D cursor control: Designs using volitional control of the EEG spectrum or steady-state visual evoked potentials. *IEEE Trans Neural Syst Rehabil Eng*. Jun; 2006 14(2):225–229. [PubMed: 16792300]
11. Leuthardt EC, Schalk G, Wolpaw JR, Ojemann JG, Moran DW. A brain-computer interface using electrocorticographic signals in humans. *J Neural Eng*. 2004; 1:63–71. [PubMed: 15876624]
12. Schalk G, et al. Two-dimensional movement control using electrocorticographic signals in humans. *J Neural Eng*. 2008; 5:75–84. [PubMed: 18310813]

13. Hochberg LR, et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*. 2006; 442:164–171. [PubMed: 16838014]
14. 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]
15. Georgopoulos AP, Kalaska JF, Caminiti R, Massey JT. On the relations between the direction of two-dimensional arm movements and cell discharge in primary motor cortex. *J Neurosci*. 1982; 2:1527–1537. [PubMed: 7143039]
16. Schwartz AB. Direct cortical representation of drawing. *Science*. 1994; 265:540–542. [PubMed: 8036499]
17. Moran DW, Schwartz AB. Motor cortical representation of speed and direction during reaching. *J Neurophysiol*. 1999; 82:2676–2692. [PubMed: 10561437]
18. Amirkian B, Georgopoulos AP. Directional tuning profiles of motor cortical cells. *Neurosci Res*. 2000; 36:73–79. [PubMed: 10678534]
19. Paninski L, Fellows MR, Hatsopoulos NG, Donoghue JP. Spatial tuning of motor cortical neurons for hand position and velocity. *J Neurophysiol*. 2004; 91:515–532. [PubMed: 13679402]
20. Brockwell AE, Rojas AL, Kass RE. Recursive Bayesian decoding of motor cortical signals by particle filtering. *J Neurophysiol*. 2004; 91:1899–1907.
21. Wu W, Gao Y, Bienenstock E, Donoghue JP, Black MJ. Bayesian population coding of motor cortical activity using a Kalman filter. *Neural Comput*. 2006; 18:80–118. [PubMed: 16354382]
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. Srinivasan L, Eden UT, Willsky AS, Brown EN. A state-space analysis for reconstruction of goal-directed movements using neural signals. *Neural Comput*. 2006; 18:2465–2494. [PubMed: 16907633]
24. Wood, F.; Prabhat; Donoghue, JP.; Black, MJ. Inferring attentional state and kinematics from motor cortical firing rates. *Proc. IEEE-EMBS 27th Ann. Int. Conf*; 2005. p. 149-152.
25. Doucet, A.; deFreitas, N.; Gordon, N. *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag; 2001.
26. Darmanjian, S., et al. Bimodal brain-machine interfaces for motor control of robotic prosthetics. *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*; 2003. p. 3612-3617.
27. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*. New York: Springer-Verlag; 2001.
28. Tkach D, Reimer J, Hatsopoulos NG. Congruent activity during action and action observation in motor cortex. *J Neurosci*. 2007; 27:13241–13250. [PubMed: 18045918]
29. Douglas, SA.; Kirkpatrick, AE.; MacKenzie, SI. Testing pointing device performance and user assessment with the ISO 9241, part 9 standard. *Proc. ACM SIGCHI Conf. Human Factors Comput. Syst. (CHI'99)*; 1999. p. 215-222.
30. Mackenzie, SI.; Kauppinen, T.; Silfverberg, M. Accuracy measures for evaluating computer pointing devices. *Proc. ACM SIGCHI Conf. Human Factors Comput. Syst. (CHI'01)*; 2001. p. 9-16.
31. Truccolo W, Friehs GM, Donoghue JP, Hochberg LR. Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia. *J Neurosci*. 2008; 28:1163–1178. [PubMed: 18234894]
32. Felton EA, Radwin RG, Wilson JA, Williams JC. Evaluation of a modified Fitts law brain-computer interface target acquisition task in able and motor disabled individuals. *J Neural Eng*. 2009; 6:056002. [PubMed: 19700814]
33. Vargas-Irwin CE, Shakhnarovich G, Yadollahpour P, Mislow JM, Black MJ, Donoghue JP. Decoding complete reach and grasp actions from local primary motor cortex populations. *J Neurosci*. 2010; 30:9659–9669. [PubMed: 20660249]
34. Kim, SP.; Simeral, JD.; Hochberg, LR.; Donoghue, JP.; Friehs, GM.; Black, MJ. Multi-state decoding of point-and-click control signals from motor cortical activity in a human with tetraplegia. *Proc. 3rd IEEE-EMBS Conf. Neural Eng*; 2007. p. 486-489.

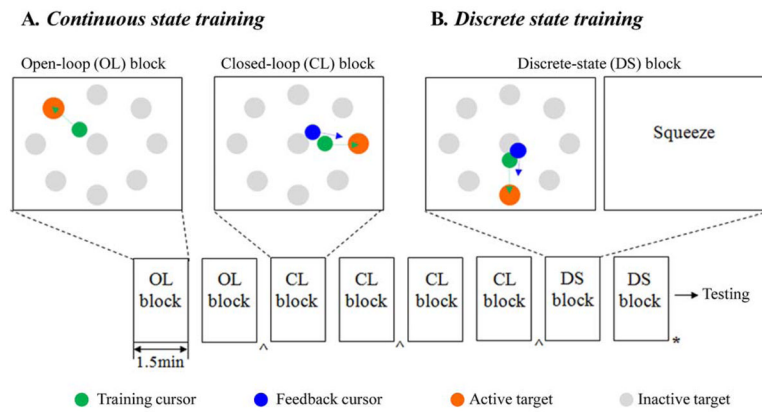


Fig. 1.

Illustration of training paradigm. (A) Continuous state (CS) training was divided into OL and CL training blocks. See the text for details of OL and CL blocks. (B) DS training involved alternating continuous movement and “click action” within a block. See the text for details of DS blocks. (C) A typical sequence of training blocks is illustrated. The CS training phase began with two OL blocks (each for 1.5 min) followed by four CL blocks (each 1.5 min). Then, the DS training phase used two DS blocks (each 1.5 min). The symbol \wedge indicates training/updating the CS decoder and * indicates training the DS decoder. The time between blocks was < 1 min.

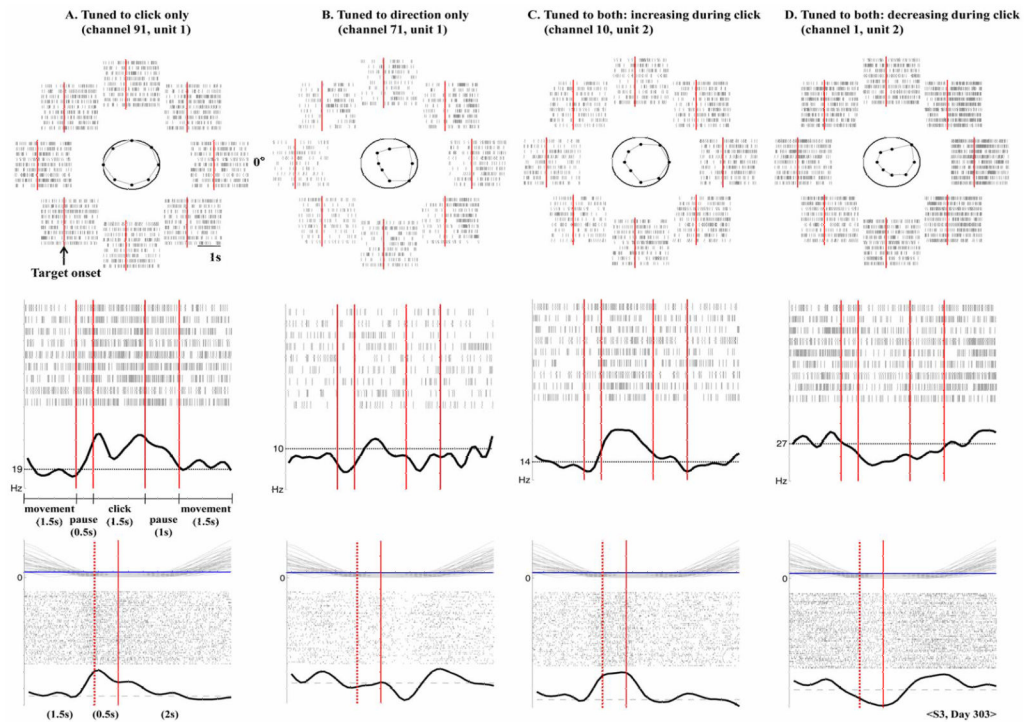


Fig. 2.

Neuronal firing patterns related to cursor movement and click. Raster plots of spiking activity of four units (recorded on day 303, S3) represent four different types of neuronal firing behaviors: (A) firing rate varied with click training only; (B) firing rate varied with TC direction only; (C) and (D) firing rate varied with both click motion and the TC direction by increasing the rate (C) or decreasing the rate (D) during click. (top) For each TC movement direction, 10 spike trains are shown beginning 1 s before target onset (vertical bar) to 1.5 s after onset. The octagons in the center circle illustrate the mean firing rates for each direction during 1.5 s after target onset. (middle) The raster plots show 10 spike trains recorded during click training in which the participant (S3) imagined squeezing the hand. The sequence of click training is described in the text (see Section II-B). Below each raster plot, the smoothed version of the peri-stimulus time histogram (PSTH) is shown. Black horizontal dashed lines indicate the mean firing rate estimated from the training data. (bottom) The spiking activity during the closed-loop point-and-click target acquisition task is shown for 48 successful target acquisition runs for all eight directions. The top plot shows the temporal variation of the distance from the NC to the target for each run. The horizontal line indicates a distance within which the NC overlapped a target. The middle spike raster plots are aligned from 2 s before to 2 s after target acquisition (the second vertical bar). The target was acquired when the click was continuously decoded over a 0.5 s interval during which the NC overlapped the target. The dashed bar indicates the start of decoding the click state 0.5 s before generating the actual click signal. The bottom plot shows the smoothed PSTH. Black horizontal dashed lines indicate the mean firing rate observed in the training data.

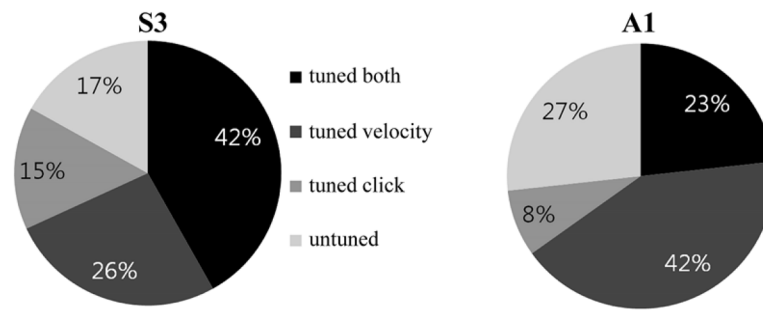


Fig. 3. Percentage of neuronal units with different tuning properties. The variability in tuning properties of neuronal units, recorded over four sessions in S3 and one session in A1, is shown. These include: tuned to both cursor velocity and click; tuned to velocity only; tuned to click only; or tuned to neither velocity nor click.

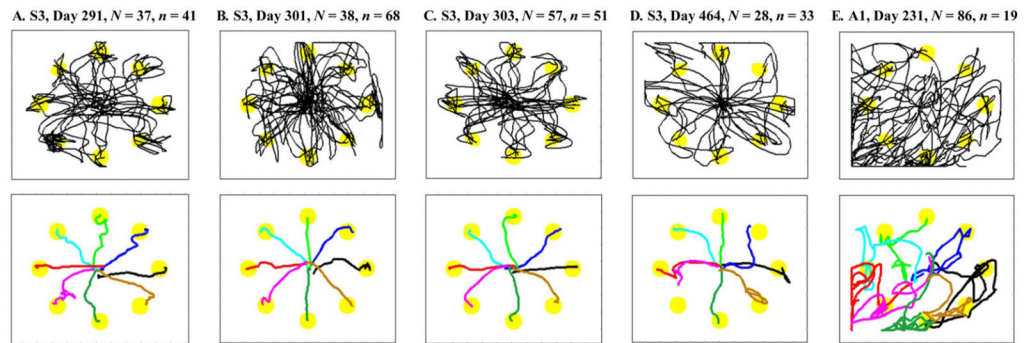
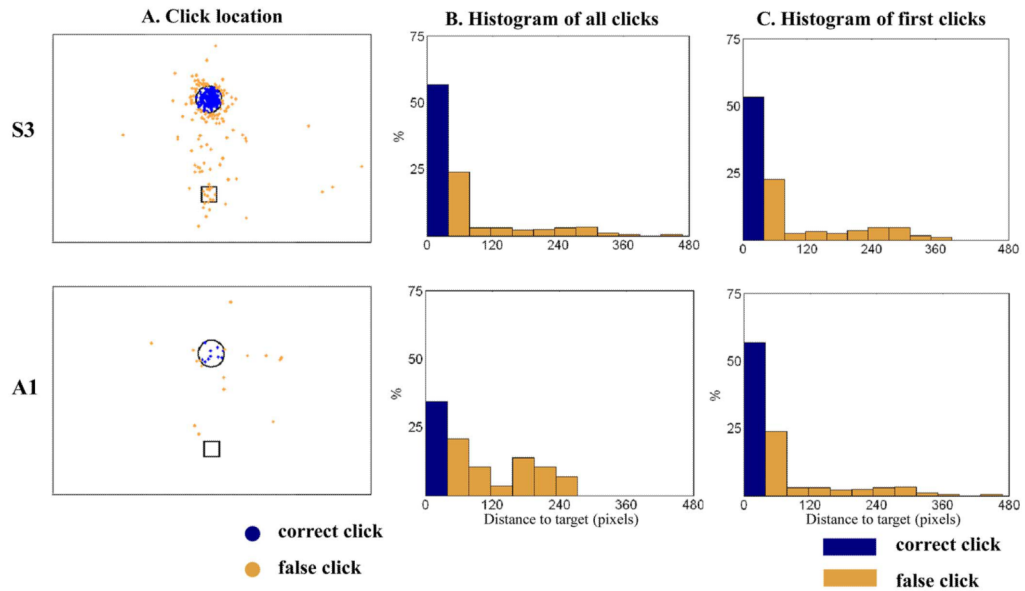


Fig. 4.

The NC movement paths. Individual NC movement paths made from the onset of target appearance to target acquisition (successful or failed) are illustrated by black lines. Circles approximately represent the target location and size (visual angle 2.6°). Below each plot of the NC paths are the mean NC paths to each target. Different lines denote the mean path for each of eight targets. The number of neuronal units (N) used for controlling the NC and the number of target acquisition runs (n) are marked.

**Fig. 5.**

Click generation as a function of distance to the target. The NC location when a click was generated is compared with its distance to the target. (A) A scatter plot of the NC locations with respect to the target is shown for all successful clicks (blue dots) and false clicks (orange dots). 193 click location samples are shown for S3 and 16 samples for A1. The circle represents the area around the target within which the NC overlapped the target (i.e., the target was selectable). The square represents the center of the screen. (B) The normalized histogram of the distance to the target for all successful (blue) and false (orange) clicks (the bin width was 39 pixels). (C) The normalized histogram of the distance at which the first click occurred in each target acquisition run.

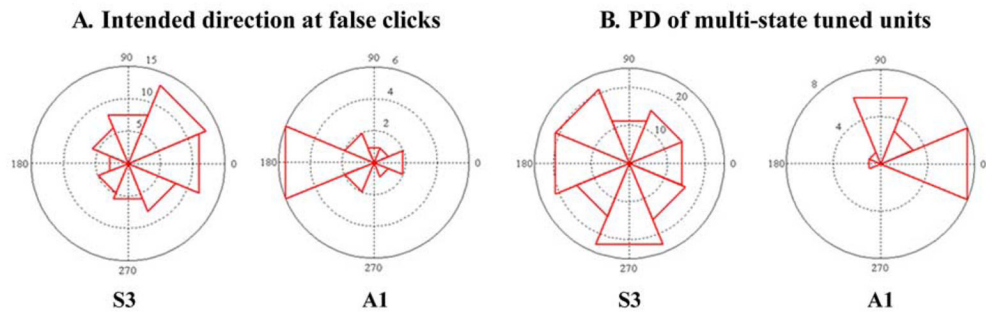


Fig. 6. Histograms of the preferred direction and the false-click-related movement direction. (A) Polar histograms of the intended direction of the NC movement at the point when false clicks were generated are shown for eight angular bins centered at $\{0^\circ, 45^\circ, \dots, 315^\circ\}$. 138 false click events were analyzed for S3 and 19 for A1. (B) Polar histograms of the preferred direction (PD) of the “multi-state tuned” units that were tuned to both click and direction. The histogram for 61 multi-state tuned units is shown for S3 (15 units for A1).

TABLE I

Point-and-Click Cursor Control Performance Measures

Day	N^a	Time out (s)	n^b	ER-to ^c (%)	ER-fs ^d (%)	MT ^e (s)	ODC ^g	MDC ^h	ME ⁱ (mm)	MV ^j (mm)	FCR ^k
S3											
292	37	30	41	0	0	8.73	3.9 (4.5)	6.7 (4.1)	15.5 (7.5)	15.2 (6.8)	0.78
301	38	30	68	0	0	6.77	1.8 (2.0)	4.5 (2.9)	15.7 (11.1)	14.3 (8.8)	0.94
303	57	9	51	3.9	0	6.18	1.8 (1.6)	4.1 (2.3)	13.5 (8.0)	13.1 (7.4)	0.29
464	28	15	33	9.1	0	7.74	2.4 (1.6)	4.6 (2.9)	23.7 (28.8)	19.9 (16.3)	0.97
S3 avg	40		48	2.6	0	7.20	2.3 (2.7)	4.9 (3.2)	16.4 (14.9)	15.1 (10.0)	0.74
A1											
231	86	27	19	47.4	0	14.29	13.1 (7.3)	15.9 (8.1)	46.6 (27.6)	41.2 (20.9)	1.00

^a N denotes the number of neuronal units identified in each session.

^b n describes the number of target acquisition runs.

^c ER-to: Error Rate by timeout.

^d ER-fs: Error Rate by false selection.

^e MT: Movement Time.

^f The numbers represent the mean (standard deviation).

^g ODC: Orthogonal Direction Change.

^h MDC: Movement Direction Change.

ⁱ ME: Movement Error.

^j MV: Movement Variability.

^k FCR: False Click Rate.