Science Translational Medicine
AAAS

www.sciencetranslationalmedicine.org/cgi/content/full/7/313/313ra179/DC1

Supplementary Materials for

Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface

Beata Jarosiewicz,* Anish A. Sarma, Daniel Bacher, Nicolas Y. Masse, John D. Simeral, Brittany Sorice, Erin M. Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, Sydney S. Cash, Emad N. Eskandar, Gerhard Friehs, Jaimie M. Henderson, Krishna V. Shenoy, John P. Donoghue, Leigh R. Hochberg

*Corresponding author. E-mail: beata@brown.edu

Published 11 November 2015, *Sci. Transl. Med.* **7**, 313ra179 (2015)
DOI: 10.1126/scitranslmed.aac7328

This PDF file includes:

Other Supplementary Material for this manuscript includes the following:
(available at www.sciencetranslationalmedicine.org/cgi/content/full/7/313/313ra179/DC1)

Movie S1 (.mov format). Self-paced typing session, participant T6's trial day 668.

Materials and Methods

Participants

The participants in this study (table S1) were enrolled in a pilot clinical trial of the BrainGate2 Neural Interface System (www.clinicaltrials.gov/ct2/show/NCT00912041; "CAUTION: Investigational device. Limited by federal law to investigational use"). They were implanted with a 1.0 mm (T6) or 1.5 mm (S3, T2, and T7) 96-channel intracortical silicon microelectrode array (Blackrock Microsystems) in the dominant hand/arm knob area of motor cortex (*44*), as previously described (*4*, *6*, *44*). Participant T7 was implanted with two arrays, both in the hand/arm area of motor cortex.

Participant S3 is a woman, 57 years old at the start of this study, and has tetraplegia and anarthria (inability to speak) resulting from a pontine stroke that occurred 9 years prior to array implantation. She retains eye movement, some head movement and facial expression, and breathes spontaneously. She has bilateral upper extremity flexor spasms that occur sporadically. Her array was implanted ~5 years prior to her participation in this research [see (*6*, *7*) for additional detail]. Participant T2 is a man, 66 years old at the start of the study, with tetraplegia and anarthria resulting from brainstem stroke. His array was implanted 4 months prior to his participation in this research. Participant T6 is a woman, 51 years old at the start of the study, and has ALS with a functional rating scale (ALSFRS-R) of 16. She retains speech, some residual limb movement, and dexterous movements of the fingers and wrist. Her array was implanted 10 months prior to her participation in this research. T7 is a man, 58 years old at the start of the study, and has ALS with an ALSFRS-R of 17. He retains speech and some limb movement, but very little hand movement. His array was implanted 6 months prior his participation in this research.

Because this is a multi-year, multi-participant study, there are differences in the details of the implementation of the investigational BrainGate system across sessions as a result of its evolution. This is also a multi-site study. Nevertheless, the implementation of the system across BrainGate clinical sites was carefully coordinated: the external, downstream system hardware was matched across sites as

closely as possible so that the same session software could be run at both sites, and personnel traveled between sites to ensure that similar methods were used across participants. Differences in methods across participants are pointed out in the relevant sections below, but it is important to note that these differences did not affect the within-session comparisons reported in this study.

Signal acquisition

Neural activity was detected by one or two 96-channel silicon microelectrode arrays (Blackrock Microsystems), and transmitted via a cable attached to a percutaneous connector during each 2-3 hour recording session. Signals were analog filtered (4th order Butterworth with corners at 0.3 Hz and 7.5 kHz) and digitized at 30 kHz by a 96-channel NeuroPort Neural Signal Processor (Blackrock Microsystems). These signals were fed to custom software written in Simulink (Mathworks) for further processing and decoding.

In participants T6 and T7, signals were downsampled to 15 kHz and then common-average referenced, which has previously been shown to improve spike signal-to-noise (*49*) (fig. S6). Namely, the mean signal from the 80 channels with the lowest noise distribution on each array, identified during a "reference block" at the start of each session (as described below), was computed and subtracted in real time from all channels' signals on that array.

In segments of 100 ms (S3) or 20 ms (T2, T6, and T7), signals were buffered for 4 ms to avoid edge effects, and then non-causally band-pass filtered for action potentials (spikes) using a 4^{th} order Butterworth filter with corners at 250 and 5000 Hz (*45*). The extracted signals were compared to an amplitude threshold set between -3 and -4.5 times the standard deviation of the filtered signals on each channel (*19*, *45*, *50*, *51*). The rate of threshold crossing events in each segment was saved to disk and, in CL blocks, decoded into intended movement direction for real-time neural control (see below for details). In participant T6, in whom there were few visible spikes exceeding the amplitude of

background multiunit activity, the power in the spike band (i.e. the root-mean square of the filtered signal) was used as an additional feature for decoding. For ease of interpretation, we will refer to both sets of neural features as "spike rates."

Standard decoder calibration task

For closed-loop neural control, intended cursor movements were decoded continuously from spiking activity using a steady-state Kalman filter (*5*, *46*, *47*), and in parallel using the same signals, an LDA classifier was used to decode the intention to click (*6*, *13*).

To initialize and calibrate the directional and click decoders, each session began with a standard decoder calibration task in which targets were cued one by one on a computer screen in a center-out-back pattern. The targets were displayed on an LCD monitor (30.5 cm high and 38 cm wide workspace) that was placed ~57 cm in front of the participant; at this screen distance, 1 cm corresponds to ~1 degree of visual angle. Targets appeared at the center, 13 cm (S3) or 15 cm (T2, T6, and T7) above or below the center, or 15 cm left or right of the center. For T2, T6, and the first T7 session, the target radius was 1.11 cm and the cursor radius was 0.4 cm. For the rest of the T7 sessions, the target radius was 1.08 cm and the cursor radius was 0.675 cm. For S3, the target radius was 1.2 cm and the cursor radius was 0.6 cm. The possible target locations were continuously visible on the computer screen, and targets became active one by one, cued by a color change. Each of the four peripheral targets was followed by the center target, and each set of peripheral targets became active in a random order before the set of targets was repeated.

First, a 1-minute OL "reference block" was run to determine the root-mean-square amplitude of each channel, in order to select a spike threshold for each channel and to select the channels with the lowest amplitude to use for common average referencing. Then, a single 2- to 3-minute OL decoder initialization block was run. In this block, the participant watched a cursor automatically move to targets

on the computer screen while imagining that s/he was controlling the cursor with her/his own hand. Participants were asked to imagine natural hand/arm movements whose directionality has an intuitive relationship with the direction of cursor movement (such as moving a mouse on a mousepad, sliding their pointer finger on a trackpad, or moving the cursor directly on the monitor as though it is on the end of their pointer finger). The particular movement imagery used by a participant on a given day was decided based on their preference, which computer input device that participant was most comfortable with, and/or which imagery modulated the recorded neural activity best in separate "cued movement" sessions. In participant T6, (imagined) movements of her thumb and forefinger elicited very robust responses in the recorded neurons, so in some sessions, she experimented with alternative directional imagery like controlling the x-dimension of the cursor with her pointer finger and the y-dimension with her thumb, using her right hand with her palm facing to the left (such that her finger and thumb movements still aligned directionally with cursor movement.) Participant T6 had considerable remaining hand movement; thus, she was permitted to move her arm/hand to follow the cursor movements if she wished, though in most sessions she chose not to move and to only imagine movement.

For participant T6, OL decoder calibration consisted of 2 minutes of center-out-back trajectories, each trajectory lasting 0.8 seconds with a 0.1-s hold and a 0.1-s delay between target onset and cursor movement (a total of ~60 center-out-back cursor movements). For participants S3 and T2, OL decoder calibration consisted of 12 center-out-back cursor movements, with each trajectory lasting 3 seconds with a 1-second target hold period and a 1-second delay between target onset and cursor movement (total OL calibration time ~2.5 minutes). For participant T7, OL calibration was 2 minutes long, and each trajectory lasted 1.2 seconds. The differences in these and other details of the methods across participants and over time within participants is a result of the evolution of the BrainGate system; these differences did not affect the within-session comparisons reported here.

After the OL decoder initialization block was run, a series of 4-8 standard CL calibration blocks were run, each 4-6 minutes long (*24*). In these blocks, the participant was asked to perform the 4-target

center-out-back task under CL neural control: the participant's intended movement direction was decoded from his/her recorded neural activity using the Kalman filter initialized in the OL block, and this movement intention was used to directly control the continuous movement of the cursor in real time. For a trial to be considered successful, the participant had 6-10 seconds (varying across participants and sessions) to move the cursor to the target and hold it there for at least 300 ms. The decoder was recalibrated after each CL block using all of the CL data acquired up until that point, using the same methods as for OL calibration, by assuming that the neural activity at each bin reflected the user's intention to move the cursor directly toward the active target (*24*, *25*) but without making any assumptions about the intended cursor speed (see *Kalman filter calibration*). The most recent CL filter was used for decoding in the following block.

Participant S3 had full 2D neural control in all CL calibration blocks. For T2, T6, and T7, in the CL calibration blocks, we algorithmically attenuated the component of the commanded velocity vector orthogonal to the instantaneous target direction using error attenuation (EA) (*7*, *24*, *52*). EA decreased the user's need to cognitively re-aim or otherwise modify their neural activity to compensate for imperfect decoding, which helped to ensure that their intended movement direction was directly toward the active target (an assumption of decoder calibration). As the decoder improved with the addition of more CL data, there was less need for error attenuation. In the first CL block, we started with 80% EA (meaning that the component of the decoded velocity vector orthogonal to the instantaneous target direction was decreased to 20% of its original magnitude), and we decreased EA by 20% each block until full 2D neural control was reached.

Click decoder calibration

In S3, T2, all but two of T6's sessions, and all but one of T7's sessions, a linear discriminant analysis (LDA) classifier running in parallel with the Kalman filter was used to decode neural "clicks" (*6*, *13*),

which the person used to select targets in the communication blocks. In T2, T6, and T7's sessions, clicks were used to select targets in the last 1 to 2 center-out blocks as well. For click calibration and decoding, spike rates were averaged over the preceding 200-400 ms (with the time window varying across participants and sessions to balance between sensitivity and robustness).

To calibrate the click decoder, the participant was instructed to imagine squeezing his or her hand closed (if using mouse or other movement imagery), or pressing down with his or her thumb on a trackpad (if using trackpad movement imagery) when cued to do so (as described next). Again, participant T6 was permitted to actually perform the movements selected for click imagery if she wanted to, but in most sessions she chose to only imagine the movements and not to perform them.

For S3, click decoder calibration was performed in a separate block from the Kalman filter calibration by running one additional 3-minute OL block after CL calibration was completed. For T2, T6, and T7, click decoder calibration was performed in the same blocks as Kalman filter calibration. In these blocks, periods during which the participant was asked to imagine clicking were cued after each return to the center target: the center target changed its color for 1 second while the cursor remained centered on it. In these blocks, successfully decoded clicks caused the cursor to change color but did not affect target acquisition (targets were still acquired by holding the cursor over them for 300 ms).

In the CL blocks with 0 error attenuation (and 0.2 error attenuation, for participants T6 and T7), the center target calibration task described above was turned off and a click-to-select task was turned on. Now, the participant was asked to acquire every target (including the center and all peripheral targets) using their neural click imagery; dwelling on the targets was no longer sufficient to acquire them. In these blocks, for purposes of labeling the training data for the LDA classifier, the participant was assumed to have been attempting to click whenever the cursor was on the active target, and was assumed not to be attempting to click during cursor movement periods of the CL block (as defined below for *Kalman filter calibration*).

The click decoder was calibrated, using linear discriminant analysis (LDA), to separate the mean-subtracted spike rates observed during intended click periods from those observed during intended cursor movements in the calibration data. For mean subtraction, each channel's mean spike rate was computed and subtracted from its ongoing rate separately for each block of data before blocks were concatenated for click decoder calibration. In the self-paced typing sessions (see *Adaptive feature mean tracking* below), each feature was also normalized by its whole-block variance before block concatenation for click decoder calibration. The top *N* channels, ranked by highest to lowest sensitivity index (*d'*), were included in the click decoder whose firing rates fell between 0.5 and 100 Hz and were not previously found to have excessive noise (as judged by root-mean-square signal amplitude). *N* varied across sessions and across blocks within sessions (range = 20–80), and was chosen based on previous sessions with good click control (T2 and S3) or to maximize 5-fold cross-validated offline click decoder performance using the calibration data (T6 and T7). Specifically, for T6 and T7, several LDA classifiers were calibrated, varying N from 20 to 80 in multiples of 10, and the best classifier was defined as the one with the highest mutual information between decoded and instructed click in the training data (*53*). A log-likelihood threshold was chosen for the LDA classifier that minimized false clicks (T2 and S3) or maximized the mutual information between decoded and instructed click in a hold-out set of data, which was chosen to be the last $1/6^{th}$ of the training data (T6 and T7). In the blocks in which neural clicks were decoded, spike rates were averaged over 200-400 ms, and a click was generated if the likelihood of the decoded click exceeded the selected threshold.

Kalman filter calibration

The methods for calibration of the Kalman filter are described in detail in (*24*). In brief, the Kalman filter is a recursive Bayesian estimation algorithm that infers the desired cursor state from the history of

spike rates. Its "observation model" assumes that the baseline-subtracted spike rates z are linearly related to the intended movement direction d at each time point *t*:

$$z(t) = Hd(t) + q(t), \tag{S1}$$

where H is the matrix relating spiking activity to movement direction and the error term, q, is drawn from a normal distribution with zero mean and covariance matrix Q. Its "state model" assumes that the intended movement direction at any time evolves from the movement direction in the previous time point

$$d(t) = Ad(t-1) + w(t), \tag{S2}$$

where A is the matrix relating movement directions at consecutive time points and the error term, *w*, is drawn from a normal distribution with zero mean and covariance matrix W. The Kalman filter is calibrated by finding the parameters H, Q, A, W that maximize the log probability of jointly observing the set of intended movement directions $D = \{d(1), d(2), \dots, d(N)\}$ and the set of spike rates $Z = \{z(1), z(2), \dots, z(N)\}$. As with LDA decoder calibration, each channel's mean spike rate was computed and subtracted from the ongoing rate separately for each block of data before the blocks were concatenated for Kalman filter calibration. In the self-paced typing sessions (see *Adaptive feature mean tracking* below), each feature was also normalized by its whole-block variance before block concatenation for decoder calibration. To compute *D*, we assumed that the neural activity at each bin reflected the user's intention to move the cursor directly toward the active target (*24*), without making any assumptions about the intended cursor speed; thus, for calibration, we set $d(t)$ to a unit vector pointing from the location of the cursor toward the location of the target. This causes the H matrix in the observation model to have units of Hz. In decoding, we converted the Kalman output back into velocity units using a gain factor tuned to each participant's preference, similar to adjusting the gain of a cursor's speed on the screen relative to the speed of a mouse movement.

We fixed A and W to provide a good trade-off between smoothness and responsiveness of cursor movement, rather than fitting them to the (somewhat arbitrary) cursor movement statistics of the calibration data (*7*, *24*). For S3, these were $A = 0.965I$ and $W = 0.03I$, where I is the identity matrix; for T2, they were $A = 0.9929I$ and $W = 0.012I$; for T6, they were $A = 0.9929I$ and $W = 0.04I$; and for T7, they were $A = 0.9929I$ and $W = 0.008I$ in one session, and $W = 0.04I$ in the remaining sessions. Thus, to calibrate the decoder, we only calculated the parameters H and Q that maximized the joint distribution.

To select channels to include in decoder calibration (and therefore to include in subsequent neural control), channels were first ranked by their normalized modulation index (NMI), a measure of the strength of their directional tuning (*24*). The NMI of unit i was defined as the norm of its PD divided by the standard deviation of the residuals in the fit:

$$NMI = \frac{||H_i||}{\sqrt{Q_{ii}}} \tag{S3}$$

The top *N* channels were included in the filter whose firing rates fell between 0.5 and 100 Hz and were not previously found to have excessive noise (as judged by root-mean-square signal amplitude). *N* varied across sessions and across blocks within sessions (range = 18–80), and was chosen to either exceed a certain NMI cutoff (T2 and S3), to maximize 5-fold cross-validated offline decoder performance using the calibration data (the first 4 T6 sessions), or set to a number based on previous sessions with good neural control (all other T6 sessions and all T7 sessions).

The data used for CL decoder calibration included only the first 2 seconds (T6 and the first T7 session) or the first 3 seconds (S3, T2, and all remaining T7 sessions) of each trial, beginning 100-400 ms after target presentation (chosen separately for each session and participant based on visual inspection of neural responses to target onset in previous sessions). This window was chosen to isolate the peak of neural responses in the initial, "ballistic" portion of each trial, helping to further decrease the

effect of any error correction on neural activity included in the calibration. For CL calibration, data from any part of the trial in which the edge of the cursor was within 1 cm of the edge of the target were also excluded from the decoder calibration because, as the cursor gets closer to the target, small cursor movements lead to larger angular changes in target direction, making intended movement direction more difficult to infer.

Neurally controlled virtual typing

Once the last standard decoder was calibrated using all available CL center-out data, the task was switched to a neurally controlled point-and-click QWERTY or radial communication interface (*14*). Using the most recent standard CL decoder for neural control, the participant was asked to type words and phrases such as "keyboard", "The quick brown fox jumps over the lazy dog", "Pack my box with five dozen liquor jugs", and/or "The quick fox is lazy now"; was asked to answer questions posed by the clinical technician; or was allowed to type words or sentences of his or her choice. In most sessions, some of the typed text was cued and some was generated freely by the participant; when there was cued text in a given session, the same text was cued both before and after RTI calibration, but the self-generated text was not constrained and could differ between pre- vs. post-RTI calibration. In the self-paced typing sessions, all of the text was generated freely by the participant.

After 2 to 4 blocks of typing, an RTI decoder was calibrated on the neural data acquired during typing (see below). Then the person was asked to type for several more blocks, using this RTI decoder for neural control. In most sessions, one or more additional RTI decoder calibrations occurred after each typing block, using up to 1 hour of typing data, and one more blocks were run using this new RTI decoder.

In participant S3, the QWERTY and radial keyboard blocks were interleaved during the same session. The standard decoder used for both keyboards was the same, but the RTI decoder used for the

QWERTY keyboard was calibrated using only the previous QWERTY keyboard blocks, and the RTI decoder used for the radial keyboard was calibrated using only the previous radial keyboard blocks. (This was an incidental experimental design choice for the purpose of testing an unrelated hypothesis; data from both keyboards could otherwise have been combined to calibrate a single RTI decoder.)

Retrospective target inference (RTI)–based decoder calibration

To calibrate the RTI decoder, at each moment along the eligible parts of the cursor's trajectory toward the next (retrospectively inferred) target, the person's intended movement direction was retrospectively inferred to have been directly toward the next selected target. Similar to closed-loop calibration with prescribed targets (but with additional assumptions about the time periods during which the person was intending to move toward the next target, as described below), these unit intended direction vectors were regressed against the corresponding baseline-subtracted neural activity to obtain H and Q. Again, only the N channels with the highest normalized modulation index were used for the decoder, where N varied across sessions as described above.

During practical BCI use, the user might spend time between selections with no particular movement intention or goal. In the case of typing, it might take time for the user to decide on the next word to type, search for the next letter, or check the word completion/prediction list for the desired word. Thus, for RTI decoder calibration, it is particularly important to ensure that only data related to movement intention are included in the calibration data. In this study, to select the time periods most likely to correspond to the user's intent to move the cursor, we employed a few simple heuristics: we only used data from the last 5 seconds preceding each target selection, and within those periods, we assumed that the user was actively trying to move the cursor toward that target only in those time bins in which the cursor moved closer to the next (retrospectively inferred) target than it was in the previous time bin (Fig. 3). Additionally, we excluded from Kalman filter calibration the time bins immediately

preceding the selection of the target, because the neural activity was likely influenced by the person preparing to generate a neural click, and because the angular estimate of the person's intended movement direction is less accurate when the cursor is closer to the target. Specifically, data points that occurred within 300 ms of target selection, and data points in which the center of the cursor was within 1.5 cm of the next selected target, were excluded from decoder calibration.

Adaptive feature mean tracking

Although RTI decoder recalibration helps to mitigate nonstationarities in directional tuning, nonstationarities can also develop in the baseline threshold crossing rates of the recorded channels (Fig. 1; figs. S1 and S2). If nonstationarities in the rate of a given channel occur not from intention to direct the cursor, but from other sources, they can bias the cursor motion in the direction of (or the opposite direction of) that channel's contribution to the decoder (*18*). Nonstationarities in baseline rates were mitigated by maintaining an accurate estimate of the baseline rate for each channel, and subtracting that baseline rate from the ongoing rate before sending each channel's neural data to the decoding algorithm.

In sessions other than the self-paced typing sessions, feature means were updated in batch by using the raw average of the firing rates from the most recent block that was included in each decoder calibration, or from the baseline of the cosine tuning function fit to each neural feature (the two different estimates of baseline generally agreed closely). During real-time decoding, these estimated baseline rates were subtracted from each feature before decoding.

In the self-paced typing sessions, the intervals of time between recording blocks could become arbitrarily long, which sometimes allowed large nonstationarities to develop between typing blocks, causing the next block to begin with a strong bias in cursor motion. Thus, in the self-paced typing sessions, we iteratively updated our estimate of the baseline rates and normalized features between blocks of neural control using an exponentially weighted running mean and variance. Namely, the mean

and variance of each feature (threshold crossing rate or spike power) were computed in real time using a recursively-defined exponential estimate of each (*48*) with a 120- or 240-s time constant,

$$\mu_t = \left(\frac{\tau - 1}{\tau}\right)\mu_{t-1} + \frac{z_t}{\tau} \tag{S4}$$

$$\sigma_t^2 = \left(\frac{\tau - 1}{\tau}\right)\sigma_{t-1}^2 + \frac{(z_t - \mu_{t-1})^2}{\tau} \tag{S5}$$

where μ_t is the current estimate of the mean for a given feature, σ_t is the standard deviation (SD) and σ_t^2 is the variance, z_t is the current sample value, and τ is the time constant.

In all but the first self-paced typing session from T6 and all but the first two self-paced typing sessions from T7, we added a heuristic to this update rule that allows the features to adapt quickly to large, discrete baseline shift events. Namely, if the current sample exceeded the estimated mean by >10X the estimated SD, the mean and SD-tracking algorithm would enter a fast-adapting phase, as follows:

$$\mu_t = \left(\frac{t - t_0 - 1}{t - t_0}\right)\mu_{t-1} + \frac{z_t}{t - t_0} \tag{S6}$$

$$\sigma_t^2 = \left(\frac{t - t_0 - 1}{t - t_0}\right)\sigma_{t-1}^2 + \frac{(z_t - \mu_{t-1})^2}{t - t_0} \tag{S7}$$

where t_0 is the time at which the feature exceeded this threshold. The effect of this heuristic was to quickly adapt the estimate of the mean and variance with any large noise event, and then to equally, rather than exponentially, weight all time points from t_0 until $t - t_0 = \tau$ (at which point Equations S6 and S7 became equivalent to Equations S4 and S5, respectively, resuming the usual recursive feature tracking).

In a subsequent step, the mean was subtracted and the SD was divided from each feature to maintain a zero-mean and unit-SD feature input into the decoder. A negligible offset was added to the SD to protect

against division by zero. During typing, the mean and SD used for mean subtraction and variance-division were fixed to their estimates from the end of the immediately preceding interblock period.

Adaptive bias correction

Within blocks of neural control, we also iteratively estimated and subtracted out the velocity bias directly (Fig. 2). This innovation was used in all of the T2, T6, and T7 sessions (it was implemented after S3's sessions were completed). Specifically, the bias estimate was initialized to [0, 0] at the start of each block, and updated iteratively by computing an exponentially weighted running mean (with a 30-s time constant) of all decoded velocities whose speeds exceeded a predefined threshold (Fig. 2, A and B). The threshold was set to the 66th percentile of the decoded speeds estimated during the most recent filter calibration, which was empirically found to be high enough to include high-speed movements in the direction of the bias, and importantly, to exclude the low-velocity movements against the bias direction that would have caused the above-described problems for adaptive mean subtraction. This exponentially weighted running mean was subtracted from the decoded velocity signals to generate a bias-corrected velocity that commanded the cursor movements (see, e.g., Fig. 2C).

Self-paced typing

In participants T6 and T7, in addition to the typing tasks described above, we also tested in separate sessions whether adaptive feature mean tracking, bias correction, and RTI decoder calibration allow for stable neural decoding for prolonged periods of time of self-paced typing. These sessions began with the standard OL and CL calibration progression described above, and then the technician initiated the self-paced typing task, allowing participants to pace the rest of the session on their own. Participants were able to pause typing when they wanted by selecting the right arrow in the radial keyboard and then the wedge containing the function "Pause". Each pause initiated a file break and, if enough typing data had

been accumulated, it also initiated an RTI decoder calibration using the last *N* blocks of typing data (where *N* was selected to provide at least 20 to 60 minutes of typing data). This file write and calibration period lasted on the order of 1 to 3 minutes, and then neural control over the cursor was restored, allowing the user to resume typing when s/he was ready by selecting the right arrow and then the wedge containing the function "Unpause." Until this function-unpause sequence was selected, no clicks on any of the other wedges had any effect.

To verify that these self-calibration methods contributed to the stability of neural control, we also performed one "control" self-paced typing session with T6 and one with T7 in which all three self-calibration methods were turned off. Just as in the other self-paced typing sessions, we asked the participant to continue typing until session time ran out, or until s/he no longer had enough neural control to type, pause, or unpause before the session was over. The latter occurred in T7's control session, which allowed time to test whether turning the self-calibration methods back on helped to rescue neural control. First, bias correction and interblock mean tracking were reinstated, and then, after a block of typing (still self-paced), an RTI decoder was built using only the data collected in that last typing block. The participant typed using this RTI decoder, paused when s/he desired, and then the original decoder was reinstated (with bias correction and mean tracking still on) for one last block.

In a final series of sessions with participant T6 (S3, T2, and T7 were no longer in the trial), we tested whether RTI calibration, between-block feature tracking, and bias correction allows for stable neural decoding across multiple days of practical BCI use. In the first session, as before, the standard calibration task was used to initialize the decoder. This standard decoder was used for decoding in the first radial keyboard block. After that, an RTI decoder was calibrated during every self-timed pause, using the data acquired during the previous 20 to 60 minutes of free typing. Each session after the first was initialized with the previous session's last RTI decoder. During the collection of these data, we were still developing the methods to automate the transfer of all remaining relevant parameters (spike threshold settings, initial estimates of baselines and features variances) across the first four of these

sessions, so some technician intervention was allowed to adjust the speed gain of the cursor and the click decoding threshold once typing started. However, in the last two sessions (days 35 and 42), there was no technician intervention at all once self-paced typing started.

Assessment metrics

One method by which typing rate was quantified was the number of correct characters typed per minute (CCPM). This is the measure of most practical value from a potential BCI user's perspective, as it determines the speed with which the person is able to communicate. However, it does not account for factors such as word prediction, the efficiency and useability of the keyboard, or the number of selections required per letter [for details about this and other metrics that can be used to compare keyboard types, see (*14*)]. Because both of the virtual keyboards used in this study have word prediction, each selection of a word in the radial keyboard requires the selection of the right arrow in addition to the word, words that are not in the dictionary require two wedge selections per letter, CCPM is not directly interpretable as the rate of intentional selections made by the user, and is therefore an imprecise measure of the quality of the person's neural control. Thus, we also computed the number of correct selections per minute (CSPM), which reflects the number of intentional selections (those that were not subsequently corrected with the back arrow or were declared as unintentional by the user), irrespective of the number of characters that resulted from those selections.

In the radial keyboard, in which all of the possible targets have the same size, this metric can also be translated into "extrapolated bitrate" (eBR), the number of bits of information conveyed per second (*17*, *32*), "extrapolated" to a virtual keyboard. This quantity is defined as $eBR = CSPM*\log_2(N\text{-}1)/60$ (where N is the number of targets; in the radial keyboard, $N = 8$). Because of the statistical structure inherent in natural language, the targets in the radial keyboard were not truly randomly distributed, so eBR is an

upper bound on true bitrate. Nevertheless, eBR provides a standard metric by which typing rates can be compared across different virtual keyboards and different studies.

SUPPLEMENTARY FIGURES

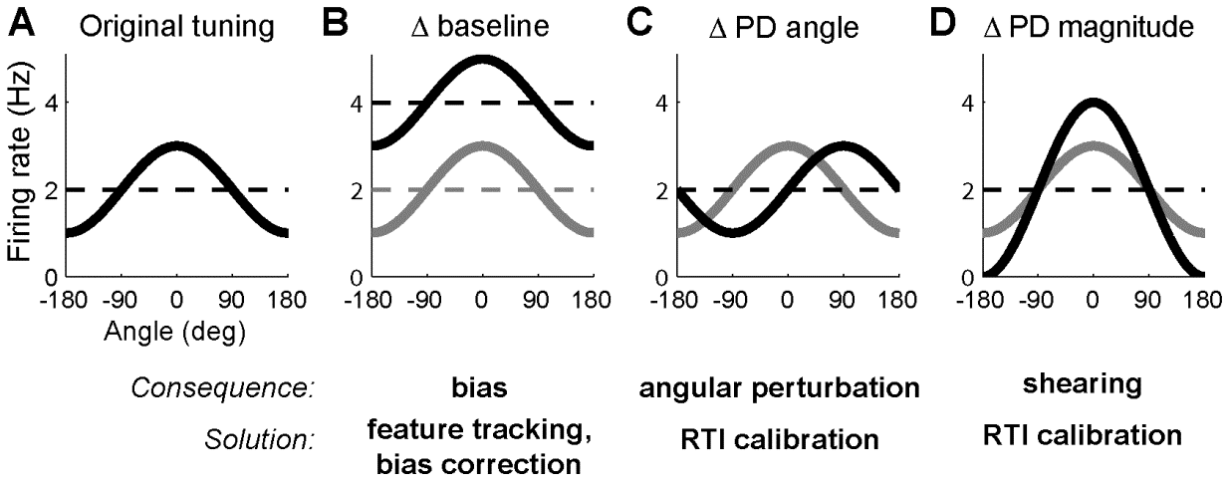


Figure S1. Schematic of all possible changes in cosine tuning. (**A**) Hypothetical cosine tuning curve, with PD at 0° (rightward), a baseline firing rate of 2 Hz, and a magnitude (difference in firing rate from baseline to peak) of 1 Hz. (**B**) Same tuning curve as in (A) with a hypothetical change in baseline rate from 2 to 4 Hz. The magnitude of this baseline shift is very small relative to typical blockwise shifts observed in real data (indeed, a 2 Hz change would be nearly invisible in Fig. 1B). Nevertheless, if this baseline shift is ignored, this hypothetical unit would appear to the decoder to be firing at or above its modeled peak rate all the time, thus biasing the cursor's motion in the direction of this unit's contribution to the decoder (*31*). (The magnitude of the bias would depend on the relative weight of this unit in the decoder). (**C**) Same tuning curve as (A), but with hypothetical change in PD angle from 0° (rightward) to 90° (upward). If ignored, this would cause a clockwise rotational perturbation in cursor motion: any time the person intended to move the cursor upward, this unit's contribution to the decoder would be rightward (*28*). (**D**) Same tuning curve as (A), but with a hypothetical change in PD magnitude from 1 to 2 Hz. If ignored, this would cause a "shearing" effect on cursor motion: cursor speeds toward and opposite of this unit's contribution to the decoder would become faster, while speeds orthogonal to that direction would be unaffected.

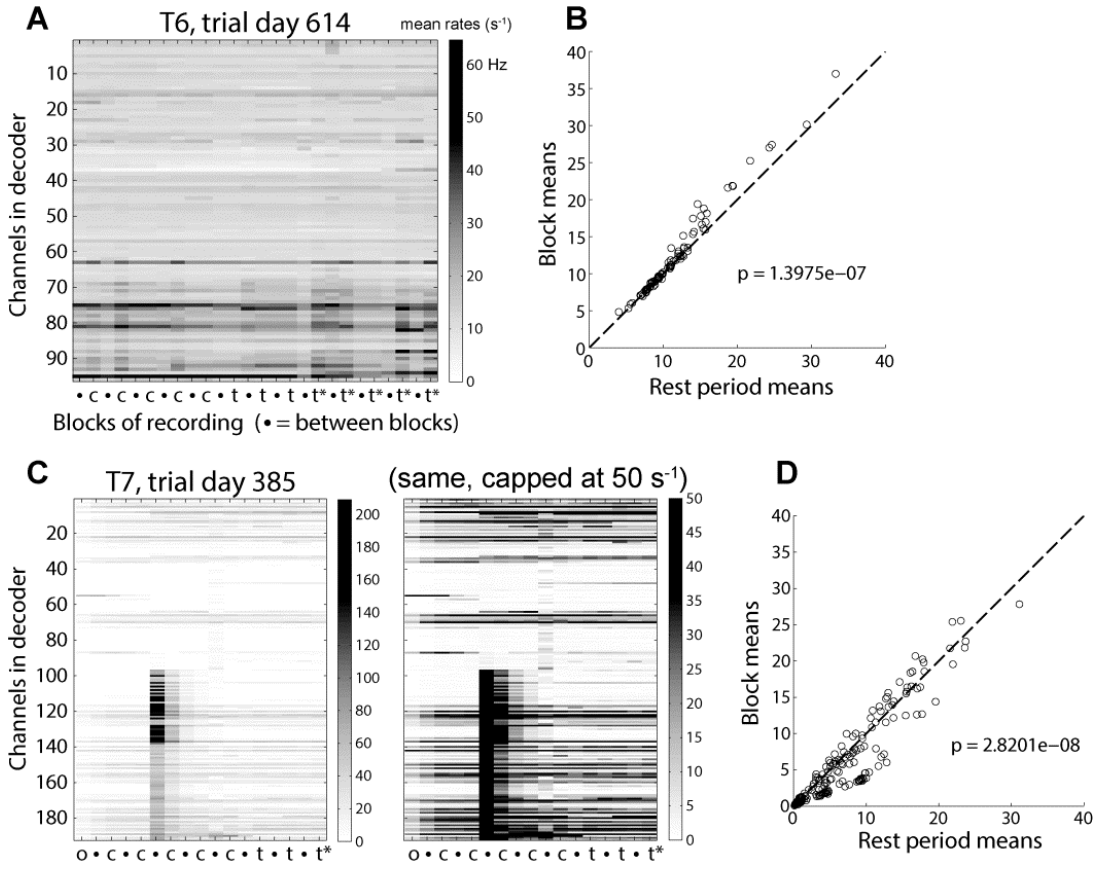


Figure S2. Nonstationarities in mean threshold crossing rates during and between blocks of neural control. (**A**) Mean rates during blocks of neural control (o, open-loop; c, closed-loop; t, typing; t*, typing using an RTI decoder) and rest periods (black dots) from an example self-paced typing session with participant T6 (corresponding to session marked with blue bars in Fig. 5A). (**B**) Mean threshold crossing rate of each channel averaged across all rest periods vs. averaged across all periods of neural control. (**C**) Left panel: similar to (A), mean rates during blocks of neural control from a self-paced typing session with participant T7 (corresponding to session denoted with blue bars in Fig. 5B). Right panel: same data as left panel, but with firing rates capped at 50 Hz for better visualization of dynamics in this lower firing rate range. (**D**) Same as in (B), for this T7 session. In this session, firing rates were significantly lower during neural control than during rest, but this might largely be attributable to the noise event that occurred on the 2nd array during the rest period after the 2nd closed-loop block and then gradually subsided (the channels on the first array were not affected, and demonstrated higher firing rates on average during blocks than during rest periods). The (likely largely non-physiological) fluctuations in rates over time, irrespective of rest vs. neural control, swamp the more subtle physiological interblock vs. block rate differences. Thus, despite known physiological differences in mean rates between rest and neural control, we found it to be useful (and sometimes essential) to update the estimate of the current mean rates from the preceding rest periods when beginning each block of neural control, particularly when rest periods are arbitrarily long, as they can be in self-paced BCI use.

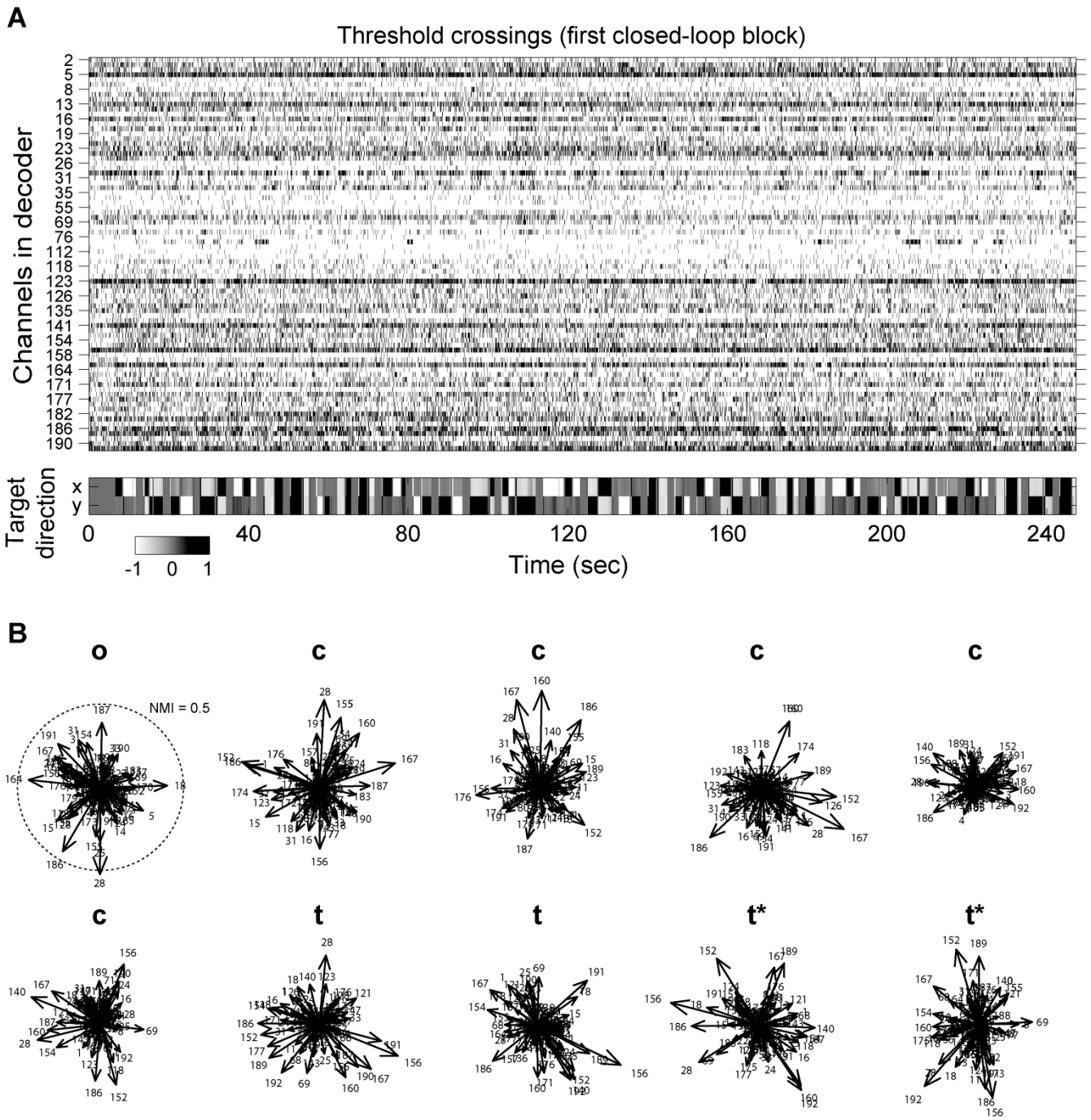


Figure S3. Directional tuning, example session. (**A**) (Top) threshold crossings over time from the first closed-loop block in T7's trial day 293 (same as shown in Fig. 1). In this session, thresholds were set to -4 x RMS. (Bottom) the corresponding instantaneous target direction (the grayscale colors indicate the x and y components of the normalized vector pointing from the current location of the cursor to the current location of the target). (**B**) Preferred directions (PD) from Fig. 1C, displayed in polar coordinates. Each PD vector is labeled with its channel ID. Its angle indicates the peak of the fitted cosine tuning curve, and its length represents its normalized modulation index (NMI; see Methods). Blocks are labeled as in Fig. 1: o, open-loop; c, closed-loop; t, typing; t*, typing using an RTI decoder. The dotted circle represents an NMI of 0.5, for scale.

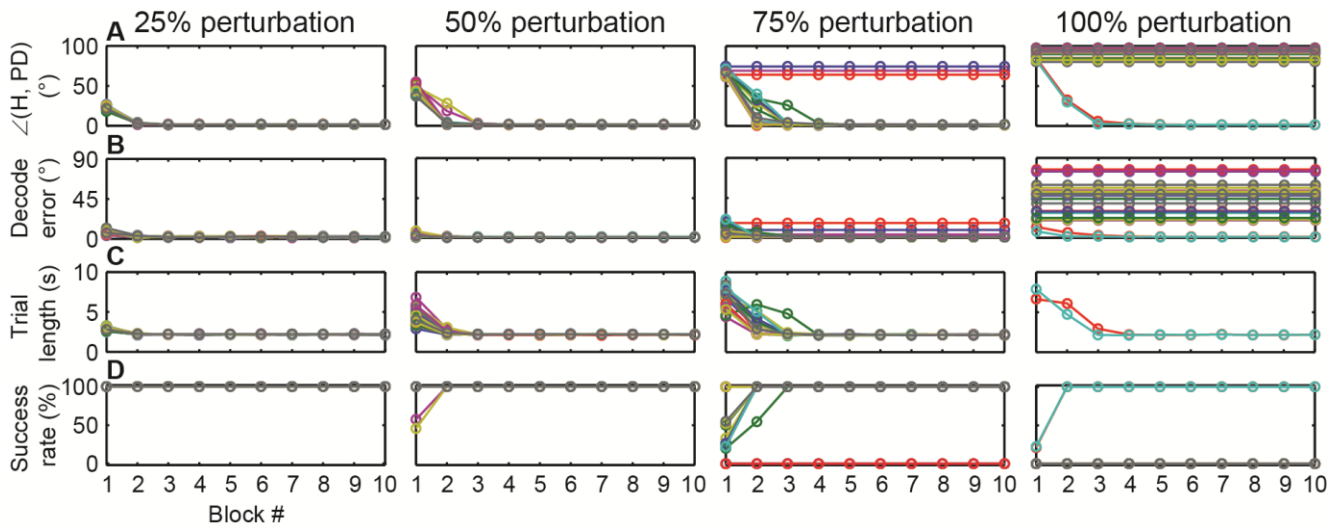
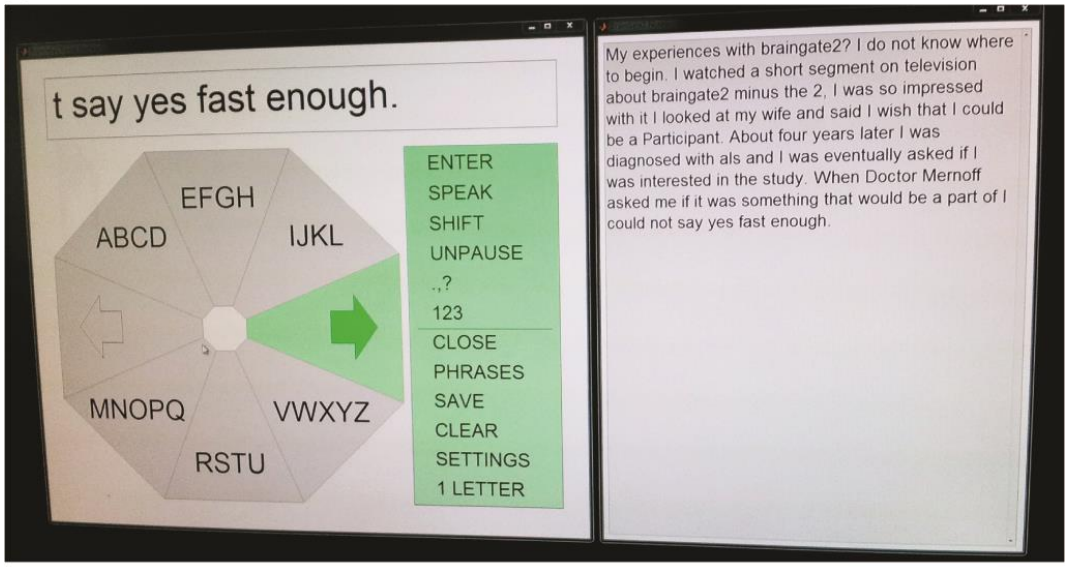


Figure S4. Simulation showing that RTI calibration can accommodate known shifts in PDs. A set of 80 simulated neurons were initialized with a set of PDs from a typical participant session, and a decoder was created whose observation model (*H*) matched those initial PDs. Then, we perturbed the PDs of random subsets (25, 50, 75, or 100%) of the simulated neurons by random angles in random directions (leaving their modulation depths the same), and we tested under what conditions RTI calibration was able to match the observation model to the changed PDs and rescue neural control in a simulated center-out-back task. Each perturbation was run 20 times; each run for each perturbation size is shown in a different color. Each simulated block was 3 minutes long, and contained approximately 17 trials when the trial timed out (the timeout was 10 sec), or approximately 80 trials when the cursor moved directly toward each target at its fastest speed. Each RTI calibration used only the last block of simulated data. The same RTI calibration function was used in this simulation as in the participant sessions, with the same heuristics for selecting data points for calibration. (**A**) The mean angular error between the decoder observation model (H_i for neuron i) and the actual PD of each simulated neuron (including both perturbed and unperturbed PDs). Because each perturbation was random, the mean angular perturbation varied across different runs of the simulation. (**B**) Decode error, defined as the angular error between a straight path to each of the 8 targets and the direction in which the cursor went when the simulator was "aiming" directly toward that target (where "aiming" is defined as modulating each simulated neuron's firing rate given that target direction and its assigned PDs). (**C**) The average time to acquire each target (averaged across all successful trials) in the block that used the decoding model described in panels (A and B). (**D**) The percentage of targets successfully acquired in that block.

A



B

Length (min)	CCPM	CSPM	Typed text
18.9	7.0	11.2	My experiences with braingate2? I do not know where to begin. I watched a short segment on television about braingate2 minus the 2,
12.5	6.5	11.4	I was so impressed with it I looked at my wife and said I wish that I could be a
15.6	6.8	9.4	Participant. About four years later I was diagnosed with als and I was eventually asked if I was interest
18.6	6.6	10.1	ed in the study. When Doctor Mernoff asked me if it was something that would be a part of I could not say yes fast enough.

Figure S5. Self-paced typing session, participant T7's trial day 293. In this session (pink bars in Fig. 5B), an RTI decoder was calibrated during the self-timed pauses after the first 2 blocks of self-paced typing, using all typing data acquired up to that point in that session. (**A**) Photograph of the typed text and radial keyboard interface. (**B**) The length of each block of typing, the number of correct characters per minute (CCPM) and correct selections per minute (CSPM) in that block, and the text entered.

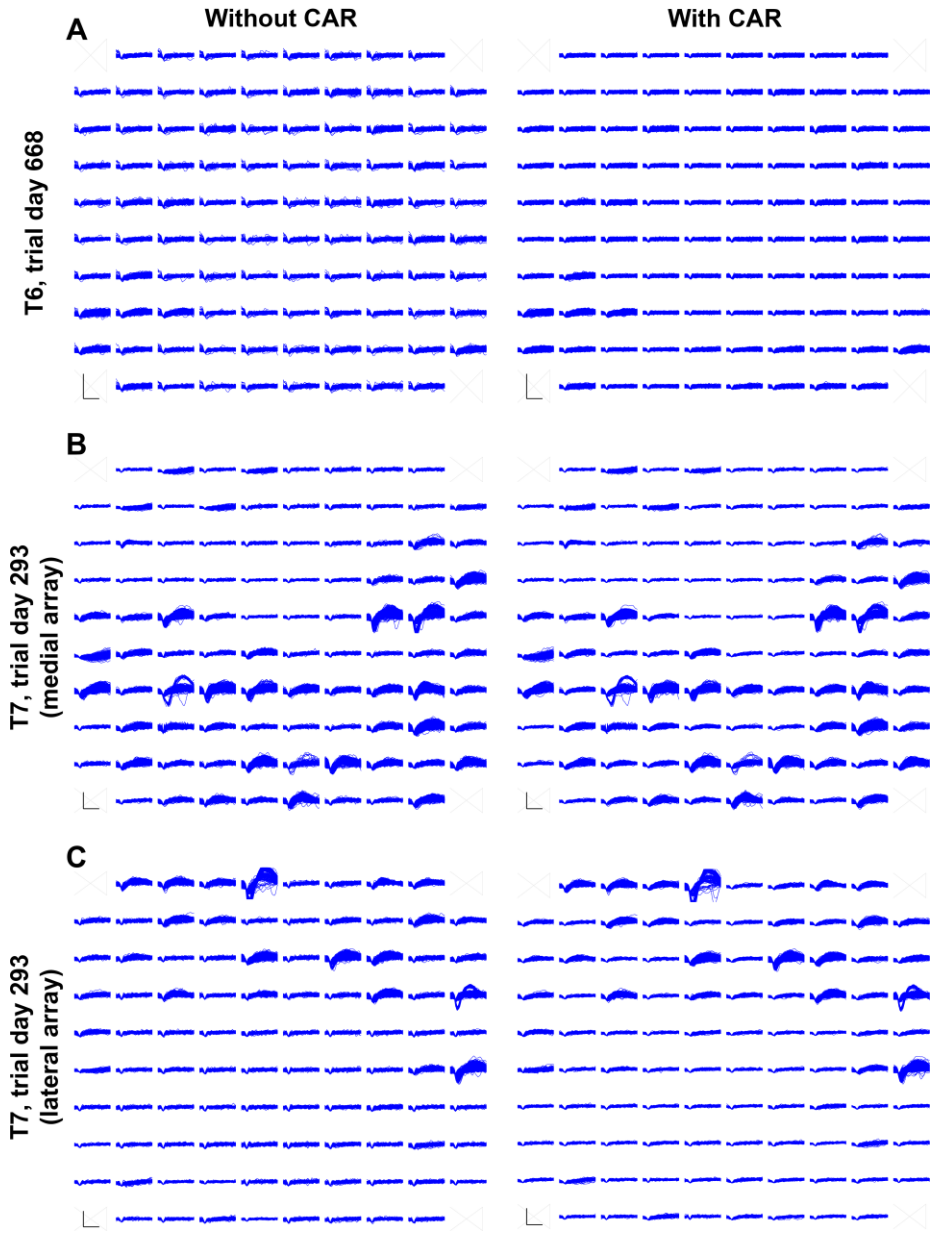


Figure S6. Spike panels from participants T6 and T7 without versus with common-average referencing. CAR helps to reduce noise in the recorded spikes (*49*). (**A**) Participant T6's spike panels on trial day 668 without and with CAR, showing 30 seconds of threshold crossings beginning 30 seconds into the first closed-loop block. Spike thresholds were set at -3.25 x RMS. Scale bars, 0.5 ms (horizontal) and 150 µV (vertical). Participant T6 had no discernable single-unit activity in her spike panel, and yet, her neural control (in terms of CCPM) was the highest of all participants in the study. This is consistent with other studies that suggest, perhaps counterintuitively, that there is not a strong relationship between neural control and the number of discernable single-unit spikes (*19*, *50*, *51*).
(**B** and **C**) Participant T7's medial array (B) and lateral array (C) on trial day 293, with and without CAR. Spike thresholds were set at -4.0 x RMS.

SUPPLEMENTARY TABLES

Table S1. Summary of participants.

Participant	Gender	Age	Etiology	Electrode length (mm)	*n* arrays	Time since implant
S3	F	57	Stroke	1.5	1	5 years
T2	M	66	Stroke	1.5	1	4 months
T6	F	51	Amyotrophic lateral sclerosis (ALS)	1	1	10 months
T7	M	58	ALS	1.5	2	6 months

Table S2. Sessions contributed by each participant for each experiment.

Participant	Post-implant day
RTI vs. standard decoder comparison sessions (Fig. 3, B and C)	
T2	137
S3	1918*, 1895*
T6	298, 301, 305, 308, 313, 512, 543, 614, 668
T7	167, 258, 280, 314, 385
Self-paced RTI sessions (Fig. 5)	
T6	543, 614, 661 (control), 668
T7	314, 385, 434 (control)
Multi-day RTI sessions (Fig. 6)	
T6	759, 761, 763, 773, 794, 801

*These sessions generated 1 data point for RAK and 1 data point for QWERTY.

SUPPLEMENTARY MOVIE

Movie S1. Self-paced typing session, participant T6's trial day 668. A real-time video of participant T6 using the radial keyboard, corresponding to the end of the last typing block (text shown in Fig. 4, performance shown by the green bars in Fig. 5A). In this session, the radial keyboard was in "cursor fly-back" mode, such that the cursor automatically moved back to the center of the keyboard after every selection. (The blurred words were redacted at the request of the participant.)