




Rapid calibration of an intracortical brain–computer interface for people with tetraplegia

David M Brandman^{1,2,3,9,28} , Tommy Hosman⁴, Jad Saab^{4,8}, Michael C Burkhardt⁵, Benjamin E Shanahan², John G Ciancibello⁶, Anish A Sarma^{4,8,3}, Daniel J Milstein⁷, Carlos E Vargas-Irwin^{2,3}, Brian Franco¹⁰, Jessica Kelemen¹⁰, Christine Blabe¹², Brian A Murphy^{16,17,18}, Daniel R Young^{16,17,18}, Francis R Willett^{16,17,18} , Chethan Pandarinath^{12,13,25,26}, Sergey D Stavisky^{12,13}, Robert F Kirsch^{16,17,18,19}, Benjamin L Walter^{17,18,20}, A Bolu Ajiboye^{16,17,18}, Sydney S Cash^{10,21}, Emad N Eskandar¹¹, Jonathan P Miller^{17,18,19}, Jennifer A Sweet^{17,18,19}, Krishna V Shenoy^{13,14,15,22,23,27}, Jaimie M Henderson^{12,23,27}, Beata Jarosiewicz^{2,8,3,24}, Matthew T Harrison⁵, John D Simeral^{8,4,3} and Leigh R Hochberg^{8,3,4,10,21} 

¹ Neuroscience Graduate Program, Brown University, Providence, RI, United States of America

² Department of Neuroscience, Brown University, Providence, RI, United States of America

³ Brown Institute for Brain Science, Brown University, Providence, RI, United States of America

⁴ School of Engineering, Brown University, Providence, RI, United States of America

⁵ Division of Applied Mathematics, Brown University, Providence, RI, United States of America

⁶ Biomedical Engineering, Brown University, Providence, RI, United States of America

⁷ Department of Computer Science, Brown University, Providence, RI, United States of America

⁸ Department of Veterans Affairs Medical Center, Center for Neurorestoration and Neurotechnology, Rehabilitation R&D Service, Providence, RI, United States of America

⁹ Department of Surgery (Neurosurgery), Dalhousie University, Halifax, NS, Canada

¹⁰ Center for Neurotechnology and Neurorecovery, Neurology, Massachusetts General Hospital, Boston, MA, United States of America

¹¹ Neurosurgery, Massachusetts General Hospital, Harvard Medical School, Boston, MA, United States of America

¹² Department of Neurosurgery, Stanford University, Stanford, CA, United States of America

¹³ Department of Electrical Engineering, Stanford University, Stanford, CA, United States of America

¹⁴ Department of Bioengineering, Stanford University, Stanford, CA, United States of America

¹⁵ Department of Neurobiology, Stanford University, Stanford, CA, United States of America

¹⁶ Biomedical Engineering, Case Western Reserve University, Cleveland, OH, United States of America

¹⁷ Louis Stokes Cleveland Department of Veterans Affairs Medical Center, FES Center, Rehabilitation R&D Service, Cleveland, OH, United States of America

¹⁸ School of Medicine, Case Western Reserve University, Cleveland, OH, United States of America

¹⁹ Neurological Surgery, Case Western Reserve University School of Medicine, Cleveland, OH, United States of America

²⁰ Neurology, Case Western Reserve University School of Medicine, Cleveland, OH, United States of America

²⁴ Present affiliation: Department of Neurosurgery, Stanford University, Stanford, CA, United States of America.

²⁵ Present affiliation: Wallace H. Coulter Department of Biomedical Engineering, Emory University and Georgia Institute of Technology, Atlanta, GA, United States of America.

²⁶ Present affiliation: Department of Neurosurgery, Emory University, Atlanta, GA, United States of America.

²⁷ Contributed equally.

²⁸ Author to whom any correspondence should be addressed.

²¹ Neurology, Harvard Medical School, Boston, MA, United States of America

²² Howard Hughes Medical Institute, Stanford University, Stanford, CA, United States of America

²³ Stanford Neurosciences Institute, Stanford University, Stanford, CA, United States of America

E-mail: david_brandman@brown.edu

Received 4 March 2017, revised 16 November 2017

Accepted for publication 4 December 2017

Published 24 January 2018




Abstract

Objective. Brain–computer interfaces (BCIs) can enable individuals with tetraplegia to communicate and control external devices. Though much progress has been made in improving the speed and robustness of neural control provided by intracortical BCIs, little research has been devoted to minimizing the amount of time spent on decoder calibration.

Approach. We investigated the amount of time users needed to calibrate decoders and achieve performance saturation using two markedly different decoding algorithms: the steady-state Kalman filter, and a novel technique using Gaussian process regression (GP-DKF). **Main results.** Three people with tetraplegia gained rapid closed-loop neural cursor control and peak, plateaued decoder performance within 3 min of initializing calibration. We also show that a BCI-naïve user (T5) was able to rapidly attain closed-loop neural cursor control with the GP-DKF using self-selected movement imagery on his first-ever day of closed-loop BCI use, acquiring a target 37 s after initiating calibration. **Significance.** These results demonstrate the potential for an intracortical BCI to be used immediately after deployment by people with paralysis, without the need for user learning or extensive system calibration.

Keywords: brain–computer interfaces, neural decoding, Kalman filtering, Gaussian process regression, discriminative Kalman filter

 Supplementary material for this article is available [online](#)

(Some figures may appear in colour only in the online journal)

1. Introduction

Brain–computer interfaces (BCIs) use neural information recorded from the brain for the voluntary control of external devices [1–10]. Motor imagery or intention can be decoded using intracortical BCI (iBCI) systems, allowing people with paralysis to control computer cursors [11–18], robotic limbs [13, 19–22] and functional electrical stimulation systems [23, 24]. Though much progress has recently been made in further improving the speed and robustness of iBCIs in people with tetraplegia resulting from spinal cord injury (SCI), amyotrophic sclerosis (ALS), and stroke [16–18], little research has been devoted to minimizing the amount of time it takes a user to gain adequate neural control.

Two approaches have been described to calibrate decoders for motor imagery-based BCIs [25]. The first approach has been to have the users adapt their behavior to a fixed BCI decoder, based on feedback (often visual). While this approach leads to BCI-based control [26–35], the process of developing reliable control requires days to weeks. The second approach has been to adapt the decoder to the user, wherein decoder parameters are computed during an explicit calibration process [3, 11–17, 19, 21, 22, 36–42]. As part of this calibration process, a baseline statistical model provides the user with initial closed-loop control; we refer to this

as *decoder seeding*. Several methods of seeding have been described that are relevant to users with mobility impairments. First, decoder parameters can be seeded with random or arbitrary values [37, 43, 44]. Second, in EEG-BCI systems, decoders can be seeded based on large databases of exemplar signals collected from multiple individuals [45–48]. Third, and most commonly applied with human research participants, decoders can be seeded from open-loop imagery [11–13, 15–21, 43, 44, 49–52], where users imagine or attempt movements for several minutes, after which their intentions are inferred without real-time external feedback.

In order to provide users with closed-loop control, iBCI decoders are calibrated by modeling a relationship between neural features (e.g. neuronal firing rates) and motor intentions that are inferred from training data (e.g. vectors from the instantaneous cursor position to the target position) [41]. Often, decoder calibration relies on seeding decoders during an open-loop imagery task in which users are asked to attempt, or imagine, controlling a preprogrammed cursor that automatically moves to presented targets [11–18, 20, 21, 24]. The resulting mapping from neural data to movement intention is then used to provide users with closed-loop neural control. Since the tuning of neurons during open-loop imagery does not generalize perfectly to closed-loop contexts [15, 37, 53–55], parameters are typically re-computed using closed-loop data

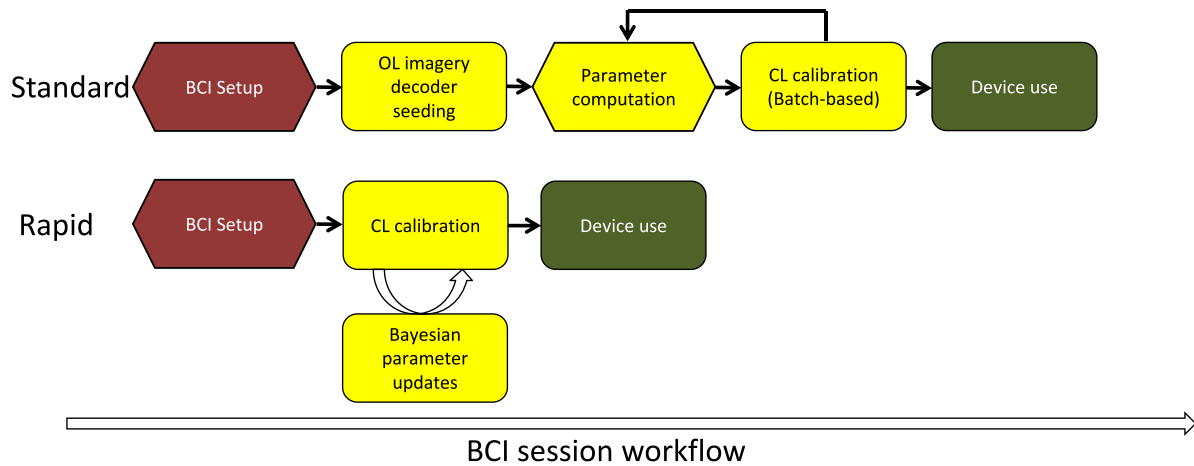


Figure 1. Schematic representation of a typical human iBCI calibration protocol versus the rapid calibration sequence. Each black arrow represents a step where a technician currently intervenes. Hexagonal and rounded steps refer to offline and online steps, respectively. The BCI user does not actively participate in offline steps. Red, yellow and green steps refer to the setup, calibration, and use of the BCI system, respectively. **Top.** Typical use of an intracortical BCI system has several steps. First, the user is connected to the BCI and the software is initialized. The user then performs open-loop imagery; decoders are seeded using this initial data; and then closed-loop calibration proceeds, and may be repeated several times depending on the protocol being used. **Bottom.** No explicit open-loop imagery step is required, and the decoder calibration steps occur without the need for technician oversight or intervention.

[15, 17, 43, 44, 49, 50, 56]. Once users are provided with initial closed-loop control, iBCI systems can continue to adapt based on closed-loop data. Adaptation during ongoing use optimizes the quality of control during extended use [9, 15, 16, 25, 43, 44, 49, 50, 57, 58] given signal nonstationarities [16, 59]. Strategies for updating decoders during ongoing use include re-computing model parameters continuously [44, 56–58] or in short batches on the timescale of minutes [43, 49, 50].

There are several reasons why the current approach to decoder calibration should be shortened and streamlined. First, some iBCI users will have diseases such as ALS or brainstem stroke, which may impair their ability to remain alert and engaged long enough to participate in prolonged calibration sequences. Second, by rapidly providing users with feedback that the device is working (i.e. by obviating the requirement for an explicit open-loop imagery task), one could anticipate greatly increased user engagement and decreased time required for developing adequate neural control [43, 44, 60, 61]. Third, in the current stage of research, where iBCI devices in humans require percutaneous connections, there is limited time available for data collection during research sessions. A reduction of calibration times on the order of minutes would dramatically streamline data collection, with further time savings gained across days. Finally, the immediate and intuitive calibration of a BCI decoder would make it possible in the future for a patient who has become acutely locked-in due to brainstem stroke to be provided with an immediately useful BCI.

Here, we demonstrate rapid calibration of iBCIs that allows users to achieve closed-loop neural control without an explicit open-loop imagery step [11–18, 20, 21, 24] (figure 1). Three iBCI users with paralysis developed accurate neural closed-loop cursor control with performance plateauing in under 3 min. We found that this approach could be used for two different decoding algorithms. Finally, a BCI-naïve individual with tetraplegia (participant T5) was able to gain continuous, unassisted, 2D closed-loop cursor control when using an iBCI system for the first time.

2. Methods

2.1. Permissions

The Institutional Review Boards of Brown University, Case Western Reserve University, Partners HealthCare/Massachusetts General Hospital, Providence VA Medical Center, and Stanford Medical Center, as well as the US Food and Drug Administration granted permission for this study (Investigational Device Exemption). The participants in this study were enrolled in a pilot clinical trial of the BrainGate Neural Interface System (ClinicalTrials.gov Identifier: NCT00912041). (Caution: Investigational device. Limited by federal law to investigational use.)

2.2. Participants

The three participants in the study were: T5, a 63 year-old right-handed male with C4 AIS-C SCI; T8, a 55 year-old right-handed male with C4 AIS-A SCI; and, T10, a 34 year-old right-handed male with C4 AIS-A SCI. All three participants underwent surgical placement of two 96-channel intracortical silicon microelectrode arrays [62] (1.5 mm electrode length, Blackrock Microsystems, Salt Lake City, UT) as previously described [11, 12]. In T5 and T8, both arrays were placed in the dominant precentral gyrus. In T10, one array was placed in the dominant precentral gyrus and a second was placed in the dominant caudal middle frontal gyrus. Data were used from trial (post-implant) days: 30 and 33 (T5); 660, 662, 665, 927, and 928 (T8); and 84, 112, 187, 188, 194, 195, 203, 215, 216, 236, 355, and 361 (T10).

2.3. Signal acquisition

Raw neural signals for each channel (electrode) were sampled at 30kHz using the NeuroPort System (Blackrock Microsystems,

Salt Lake City, UT). Further signal processing and neural decoding were performed using the xPC target real-time operating system (Mathworks, Natick, MA). Raw signals were downsampled to 15kHz for decoding, and de-noised by subtracting an instantaneous common average reference [16, 17] using 40 of the 96 channels on each array with the lowest root-mean-square value (selected based on their baseline activity during a 1 min reference block run at the start of each session). The de-noised signal was band-pass filtered between 250 Hz and 5000 Hz using an 8th order non-causal Butterworth filter [63]. Spike events were triggered by crossing a threshold set at $3.5 \times$ the root-mean-square amplitude of each channel, as determined by data from the reference block. The neural features used were: (1) the rate of threshold crossings (not spike sorted) on each channel, and (2) the total power in the band-pass filtered signal [14–16]. Neural features were binned in 20ms non-overlapping increments for decoding with T5 and T10. For T8, the 20 ms bins were causally smoothed over 60 ms.

For closed loop decoding, the neural features, z , were split between threshold crossings and the band power filtered signal [14–16]. We used the top 40 features ranked by signal-to-noise-ratio of neural modulation [64]. For all three participants, 35% ($\pm 3\%$) of the features selected were threshold crossings, and 65% ($\pm 3\%$) were band power. The number of features that were tuned for closed-loop neural cursor control was computed for each participant. A feature was considered tuned if there was a statistically significant relationship between the feature and the direction of movement grouped by octants (ANOVA, $p < 0.001$). The average fraction of features that were tuned for directional movement was 0.64 (T5, range: 0.53–0.70); 0.36 (T8, range: 0.17–0.51); and, 0.58 (T10, range: 0.23–0.84).

2.4. Calibration task

Task cueing was performed using custom built software running Matlab (Natick, MA). The participants used standard LCD monitors placed 55–60 cm distance from the participant, at a comfortable angle and orientation. Participants engaged in the Radial-8 Task as previously described [12, 16]. Briefly, targets (size = 2.4 cm, visual angle = 2.5°) were presented sequentially in a pseudo-random order, alternating between one of eight radially distributed targets and a center target (radial target distance from center = 12.1 cm, visual angle = 12.6°). Successful target acquisition required the user to place the cursor (size = 1.5 cm, visual angle = 1.6°) within the target's diameter for 300 ms, before a pre-determined timeout. Target timeouts resulted in the cursor moving directly to the intended target, with immediate presentation of the next target. During the first 60 s of calibration with T8, we attenuated 80% of the component of the decoded vector perpendicular to the vector between the cursor and the target [15, 39]. No error attenuation was used for T5 or T10.

2.5. Calibration procedure

2.5.1. Previous approaches to iBCI decoder calibration.

Prior to the procedure described in this report, decoder

calibration was a multi-step process requiring active intervention from a trained technician. We have previously reported two approaches. The first [17, 18] used: (i) initial BCI setup, (ii) 5 min of open-loop motor imagery, (iii) a technician supervised decoder calibration stage, (iv) 5 min of closed-loop control, and finally, (v) a second supervised decoder calibration (figure 1). Our team has also implemented protocols for initial decoder calibration wherein users performed 2 min of open-loop imagined movements, followed by a series of batch-based calibration sequences over 6 min [12] or 9 min [15, 16] of gradually decreasing computer assistance [15, 16, 39]. Thus, our standard approach to calibration has been to seed decoders with open-loop imagery, and then to perform batch-based decoder updates, taking approximately 8–11 min of actual calibration time (excluding the additional time required for manual intervention by the technician between calibration epochs—approximately 0.5–1 min between blocks).

We sought to accelerate calibration with three modifications to our standard approach. First, we removed the explicit open loop calibration step. Second, rather than performing batch-based calibration every 2–5 min, we updated decoder parameters every 2–5 s throughout the calibration phase. Third, we automated the remaining calibration steps normally performed by a technician.

We note that our new approach to calibration performs three processes simultaneously, which we briefly list here and then describe in detail in later sections. First, the decoding algorithm performs real-time predictions of the intended velocity based on the neural features, providing users with closed-loop neural control (section 2.6). Second, during calibration, data consisting of (y_t, z_t) pairs are collected, where y_t is the 2×1 unit vector pointing from the cursor's current position to the known target [14–16], and z_t is the corresponding 40×1 vector of neural features. Third, the (y_t, z_t) pairs are used to update the parameters used by the decoding algorithm every 2–5 s (section 2.7).

At the start of the session, the decoder parameters are initialized to values that leave the cursor stationary regardless of the neural features. Nevertheless, training data can be collected since the user has been instructed to attempt to move the cursor to the target on the screen. After the first parameter update (2–5 s after the start of the session), the decoder began using neural features to provide users with closed loop neural control, although the user had poor control, as expected. As the calibration task continued, the decoder parameters were re-computed (based on growing amounts of training data, every 2–5 s) and the quality of control improved. At the end of the calibration the parameters were locked before assessment with the Grid Task (see below).

2.6. Decoders: basic models and algorithms

We tested our calibration procedure with two different decoding algorithms: the steady-state Kalman filter, and a novel technique combining Gaussian process regression with the discriminative Kalman filter (GP-DKF), described below. Both decoders are Bayesian sequential filters. Let x_t denote

the hidden state at time step t , namely, the unobserved 2×1 vector of intended velocity of the cursor, and let z_t denote the observation at time step t , namely, the 40×1 vector of neural features (sections 2.3 and 2.5). In section 2.7 we discuss the distinction between x_t and y_t . The initial probability distribution function (PDF) of intended velocity at time 0 is denoted $p(x_0)$, the conditional PDF of intended velocity at time t given the intended velocity at time $t-1$ is denoted $p(x_t|x_{t-1})$, and the conditional PDF of neural features at time t given the intended velocity at time t is denoted $p(z_t|x_t)$. The conditional PDFs $p(x_t|x_{t-1})$ and $p(z_t|x_t)$ model, respectively, the temporal dynamics of intended velocity and the instantaneous mapping from intended velocity to neural features. Bayesian sequential filters use these PDFs and the history of observed neural features (z_1, \dots, z_t) in order to compute the conditional PDF of the intended velocity given the complete history of observed data:

$$p(x_t|z_1, \dots, z_t).$$

Both decoding algorithms used the mean of this conditional PDF as the prediction of the unknown intended velocity at time step t . This prediction was used to update the cursor position.

The two decoding algorithms are derived under differing assumptions for the observation model, $p(z_t|x_t)$, but they share the same model for intended velocity:

$$p(x_0) = \eta(x_0|0, W_0)$$

$$p(x_t|x_{t-1}) = \eta(x_t|Ax_{t-1}, W)$$

where $\eta(x|\mu, \Sigma)$ denotes the multivariate Gaussian PDF with mean vector μ and covariance matrix Σ evaluated at x . This is the model for the hidden states underlying the well-known Kalman filter [11, 65, 66]. W and W_0 are 2×2 covariance matrices and A is a 2×2 matrix. The parameters A , W , W_0 were fixed at values that have historically worked well for closed-loop neural control [14–16]. Both A and W noticeably affect the dynamics of cursor control. Loosely speaking, A controls the smoothness of cursor dynamics, whereas W controls the extent to which decoded neural features are used to drive cursor control.

2.6.1. Kalman filter. The first decoder uses the linear, Gaussian observation model

$$p(z_t|x_t) = \eta(z_t|Hx_t, Q),$$

where Q is a 40×40 covariance matrix and H is a 40×2 matrix relating intended velocity to brain activity. The parameters H and Q were learned from the training data collected during the calibration procedure (sections 2.5 and 2.7). The classical Kalman-update equations use the observed neural features as well as A , W , W_0 , H , and Q , in order to compute the mean of $p(x_t|z_1, \dots, z_t)$, used for closed-loop neural control [11, 65, 66]. The version of the algorithm that we used is based on the steady-state update Kalman equation [14–16] and uses real-time bias correction [16, 19]. During closed-loop calibration, the parameters H and Q were updated every 2–5 s. We do not go back to time step 0 and repeat the entire Kalman filtering algorithm with the updated parameters.

2.6.2. Discriminative Kalman filter (DKF). In contrast to the standard Kalman filter, the DKF places no constraints on the observation model: it may be nonlinear and non-Gaussian. Although relaxing this Gaussian assumption may better model the non-Gaussian features of neural data, it introduces two fundamental difficulties. First, the standard filtering algorithms become computationally intensive (unlike the fast Kalman-update equations). Second, learning a generic observation model from training data is much more complicated than learning the parameters H and Q in the Kalman model [67, 68].

We have recently shown how these two difficulties can be surmounted for our purposes using an approach that we call the discriminative Kalman filter (DKF) [67]. In brief, the DKF solves the problem of computationally intensive Bayesian filtering updates with judicious and theoretically motivated Gaussian approximations. Using these approximations does not require full knowledge of the observation model, but only requires knowing the conditional mean and covariance of the intended velocity given the current neural features. We denote the conditional mean and variance as the functions $m(z_t)$ and $S(z_t)$, respectively. The only requirement in using the DKF is to learn $m(z_t)$ and $S(z_t)$ during calibration (see sections 2.5 and 2.7). Even though $m(z)$ and $S(z)$ are arbitrary functions, we refer to them as parameters of the decoder.

Despite the highly non-linear nature of $m(z)$ and $S(z)$, the calculations needed to compute the predicted intended-velocity (given A , W , W_0) may be performed in real time systems [67]. When the parameters $m(z)$ and $S(z)$ are updated (every 2–5 s during the calibration phase), we simply modify their values in the DKF update equations for future updates. We do not go back to time step 0 and repeat the entire DKF filtering algorithm with the updated parameters. We emphasize that the DKF is distinct from the extended Kalman filter, the unscented Kalman filter, and related algorithms, which are based on combining Gaussian approximations with the conditional mean and covariance of the *neural features given the intended velocity*; instead, the DKF approximates the *intended velocity given the neural features*. Loosely speaking, the DKF can be viewed as a principled way to do model-based temporal smoothing of direct (nonlinear) regression predictions of intended velocity.

2.7. Decoders: parameter learning and updating

Each decoder has parameters that relate the intended velocity of the cursor to the neural features, namely, H and Q for the Kalman filter and $m(z)$ and $S(z)$ for the DKF. In principle, parameters could be learned from observed (x_t, z_t) pairs, however, the true intended velocity (x_t) is always unobservable. Instead, we learn the parameters using the (y_t, z_t) pairs that are collected throughout the calibration phase (section 2.5). Recall that y_t is the unit vector pointing from the cursor's current position to the known target. Our calibration algorithms simply use y_t as a direct surrogate for x_t , the rationale being based on the assumption that the user intended to move directly to the target at time t . At time step r , say, we horizontally concatenate y_1, \dots, y_r into a $2 \times r$ matrix Y and, similarly,

we concatenate z_1, \dots, z_r into a $40 \times r$ matrix Z . The matrices Y and Z comprise the training data at time step r . For certain values of r (every 2–5 s), these matrices are used to update the parameters of the decoding algorithm.

Given Y and Z , the calibration for the Kalman filter computes H and Q , and calibration for the DKF computes $m(z)$ and $S(z)$. In both cases, we use standard regression tools: linear regression for H and Q and nonlinear regression for $m(z)$ and $S(z)$. Since closed-loop parameter updates begin updating seconds after the start of calibration, regression methods were chosen that provided non-singular solutions given small amounts of data. In both cases, we used Bayesian regression methods. In a Bayesian approach, uncertainty about a parameter is represented by a probability distribution. The Bayesian prior refers to the probability distribution before data has been collected. As new information becomes available, the parameter's distribution is updated accordingly (referred to as the posterior distribution). Hence, Bayesian methods provide the experimenter a method for updating their belief of a parameter given evolving information. Note that the term Bayesian is used in two different contexts in this paper. Our decoding algorithms are based on Bayesian filtering, whereas our calibration algorithms incorporate new training data using Bayesian parameter updating. In principle, decoding and parameter learning could be combined into a unified Bayesian procedure, but this would necessitate more complicated decoding algorithms and would be too computationally demanding to perform real-time decoding.

2.71. Kalman filter with Bayesian linear regression. Given matrices Y and Z , the parameter H (relating motor intention to neural features) was updated via

$$H = ZY^T(YY^T + \alpha I)^{-1},$$

where I is the 2×2 identity matrix and α was a regularization parameter ($\alpha = 10^{-3}$). This particular method of updating H can be viewed equivalently as ridge regression or as Bayesian linear regression [68]. We will describe the Bayesian derivation. Let $\text{vec}(H)$ denote the vectorization of H , i.e. the 80×1 vector created by stacking the 40×2 columns of H on top of one another. We set the prior for $\text{vec}(H)$ (given Y and Q) to be a multivariate Gaussian with mean zero and covariance $\alpha^{-1}I \otimes Q$, where \otimes denotes the Kronecker product, and we use the Kalman filter observation model (replacing x_t with y_t) which specifies that $p(z_t|Y, H, Q) = \eta(z_t|H y_t, Q)$. Under this model (regardless of the prior on Q), the posterior mean of H is $ZY^T(YY^T + \alpha I)^{-1}$, and we use this as our estimate of H based on Y and Z . As more data were collected, the impact of the αI term decreased and our estimate for H approached the maximum likelihood estimate that is traditionally used for selecting parameters for the Kalman filter [66]. After H was updated, Q was updated using the covariance of the residuals $Z - HY$.

2.72. Gaussian process regression. Given matrices Y and Z , the nonlinear function $m(z)$ (relating neural features to motor intention) was computed via

$$m(z) = Y(K(Z, Z) + \alpha I)^{-1}K(z, Z)^T,$$

where α was a regularization parameter (set to $\alpha = 0.6$), I is the $r \times r$ identity matrix, $K(z, z')$ is the standard radial basis function kernel, $K(Z, Z)$ is an $r \times r$ matrix with $K_{ij}(Z, Z) = K(z_i, z_j)$, and $K(z, Z)$ is a $1 \times r$ vector with $K_j(z, Z) = K(z, z_j)$. Intuitively, the function $m(z)$ returns a 2×1 prediction of intended velocity by taking a weighted average of all of the intended velocities in Y (supplementary figure 1 (stacks.iop.org/JNE/15/026007/mmedia)). The weights were determined by comparing the current vector z of neural features to the neural training data in Z using the radial basis kernel (i.e. a Gaussian kernel in 40 dimensions) as a measure of similarity (supplementary figure 1). In this case the prior on the function $m(z)$ was a zero mean GP with covariance kernel K and we update using the posterior mean. The function $S(z) = S$ was taken to be constant and was estimated from the covariance of the residuals $m(z) - Y$. To emphasize that $m(z)$ is learned with GP regression, we call our second decoder the GP-DKF.

We found evaluating $m(z)$ was slow when the number of training points in Y and Z was too large. We used up to 60s (3000 datapoints) of data for decoding with T5 and T10. For T8, we increased this to 120s (6000 datapoints). The number of datapoints was chosen based on early validation of the GP-DKF method. Early validation of the GP-DKF also suggested higher quality cursor control using only band power for z_t , rather than both band power and threshold crossing counts. For the purposes of comparing performance to the Kalman decoder, we selected 40 features. Neural features were sorted into octants according to the direction of movement. Each octant contained a maximum number of datapoints (i.e. $3000/8 = 375$), with corresponding buffer allocations. If the number of datapoints exceeded the buffer size per octant, the oldest data in that octant were replaced.

2.8. Performance measurement

After calibration, we quantified performance using a Grid Task after locking decoder parameters [13, 18, 52]. This task consisted of a grid of N square targets arranged in a square grid ($N = 25, 36, 49, 64, 81$ or 100 , length of one side of square grid = 24.2cm , visual angle = 24.8°). One of N targets was presented at a time in a pseudo-random order. Targets were acquired when the cursor was within the area of the square for 1 s. Incorrect selections occurred if the cursor dwelled on a non-target square for an entire hold period [69]. When comparing the GP-DKF decoder to the previously described standard Kalman calibration scheme [14–16], each comparison block was 3 min in length. When performing rapid decoder calibration comparisons with T10, each comparison block was 2 min in length. Block lengths were selected based on the participant's preference.

The Grid Task is designed as a generalized version of a single-channel communication task, designed to measure target selection speed. The achieved bit rate (BR) measures the effective throughput of the system [52]:

$$BR = \log_2(N - 1) \max(S_c - S_i, 0) / t$$

where N is the number of possible selections, S_c and S_i are the number of correctly and incorrectly selected targets, respectively, and t is the elapsed time. The max function ensures that the bit rate remains non-negative.

2.8.1. Testing performance as a function of decoder and calibration times. We performed a series of comparisons between the GP-DKF and Kalman decoders with variable amounts of calibration time. Decoders were calibrated using the Radial-8 Task, parameters were locked, and the performance was measured using the Grid Task. For instance, when comparing performance of the Kalman decoder at 3 min versus 5 min, the Kalman decoder was first calibrated for 3 min and then immediately tested (representing condition A). Next, a new Kalman decoder was calibrated for 5 min before being tested with the Grid Task (B). Decoder/timing pairs were alternated, in an A–B–A–B format. Only data collected the same day were used for statistical comparisons.

2.8.2. Testing the new rapid calibration protocol versus the previous standard. To test the performance of the new calibration protocol versus the standard calibration protocol, we began by calibrating using the technician supervised, computer-assisted, batch-based 11 min calibration scheme (as previously used in [14–16]). Calibration began with 2 min of open loop imagery, which was used to seed a decoder. Next, three blocks (3 min each) of closed-loop neural cursor control were performed, while computer assistance was gradually removed. Performance was assessed using the Grid Task. Next, we performed 3 min of closed-loop decoder calibration using the GP-DKF decoder, locked decoder parameters, and then assessed performance with the Grid Task. We used block-based feature mean updates for the standard Kalman decoder [16]. Only data collected the same day were used for statistical comparisons.

2.8.3. Isolating the calibration protocol versus the previous batch-based protocol. To test the performance effect from the rapid calibration protocol, we compared it to the standard batch-based method while controlling for calibration length. We began by having the participant attempt open loop motor imagery for 1 min. The neural features and cursor kinematics were used to seed both the standard Kalman decoder (i.e. decoder parameters did not update for 2 min, condition A) and the rapidly updating Kalman decoder (i.e. decoder parameters updated every 3–5 s for 2 min, condition B). After locking parameters, we then tested performance using the Grid Task, balancing the comparisons by alternating whether A or B was tested first.

2.8.4. Mean angular error measurements. To investigate performance saturation, we performed offline simulations of decoder performance by training decoders with variable amounts of training data. We computed the angular error between the predicted decoder value without filtering (i.e. the K_z term of the Kalman filter) and the label modeled as the vector from cursor to target [14–16]. Data from a single experimental session were concatenated together. A decoder was trained using a random subsample without replacement and then used to predict the mean angular error for another

subsample of the same size. Decoder predictions were bootstrapped 100 times.

2.8.5. Additional metrics. In addition to bit-rate and angular error, we also report two additional metrics. The first is time to target, which is the time between target presentation and acquisition by the user. The second is orthogonal direction changes, which is a measure of how consistently the cursor went towards the target [12].

2.9. Additional methods for participant T5's first day of closed-loop neural control

Though participants often remain in the BrainGate and other BCI research for years [12, 18, 19, 41], there can only be one ‘first day’ of attempted neural control. There is also special interest in understanding how rapidly a BCI-naïve user with tetraplegia might gain useful neural control of a BCI. We had the opportunity to deploy GP-DKF on participant T5's first day of neural control.

Prior to T5's first attempt at closed-loop neural control, the following text was read verbatim:

‘As you know, we are recording from a part of the brain responsible for controlling movement. In this part of the brain, the nerve cells respond to you attempting to move part of your body. You'll be presented with a cursor and a target. We'd like you to think about moving your hand/arm/finger towards the target. As you try and do so, the system will be recording from the nerve cells. It will learn that [sic] the pattern of brain activity associated with wanting to go to a direction.

‘As you try to move the cursor to more targets, the system will learn a list of different responses. The cursor may not behave as you expect. For example, you may be trying to move the cursor to the right but the cursor moves to the left. This is OK. It's not your fault. But no matter what the cursor is doing, keep trying to move it towards the target. The system will rapidly learn your brain signals and may start to correct within a few seconds. It's important to stay consistent with what you're attempting to do. For example, rather than repeatedly trying to move your hand to the left, you have to attempt to perform a continuous left-moving motion.

‘We're going to try a few different imageries today. Can you think of some motions that would be intuitive to control a computer cursor on a screen?’

We explored a total of six motor imageries with T5. ‘Joystick’ refers to the control of an airplane using a joystick, where the dominant hand is resting comfortably on a surface and moving a joystick in a particular direction driving cursor movement. Left movement occurs through pronation, right through supination, upwards with ulnar deviation, and downwards with radial deviation. ‘Whole arm’ refers to attempting to control the cursor by pointing using the index finger, where the shoulder and elbow are free to move with fixed wrist and

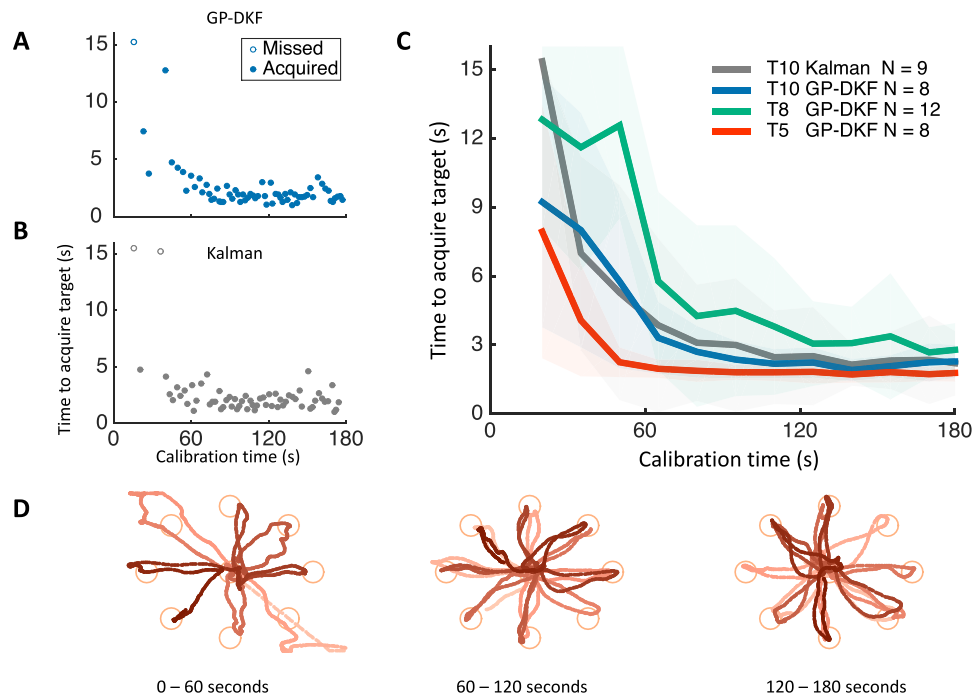


Figure 2. Rapid calibration during the Radial-8 Task. Participant T10 performed a 3 min calibration sequence with either the GP-DKF (a) or the Kalman (b) decoders. Targets were acquired when the cursor overlapped the target for 300 ms, with a 15 s timeout. (c). Multiple calibration sequences from participants T5 (red), T8 (green) and T10 (blue) were done using the GP-DKF decoder, and with the Kalman decoder in T10 (grey). The thin dark lines are the average amount of time to acquire a target across all blocks (shaded area is ± 1 standard deviation). Averages are computed by binning data in 15 s increments, with a 5 s offset from calibration start. (d) Example cursor trajectories during calibration using the GP-DKF decoder (participant T5). The brightness goes from light to dark as time elapses during the 60 s interval of closed-loop neural cursor control. Data were used from trial days: 30 and 33 (T5); 662, 665, (T8); and 112, 203, 215, 236 (T10).

finger positions. ‘Index finger’ refers to imagery in which the wrist is resting comfortably on a surface, and the index finger is used to control a pointing stick mouse. ‘Stirring a pot’ refers to the imagery of holding a wooden spoon over a saucepan. The shoulder is fixed, and movement occurs with a combination of arm protraction and retraction, as well as elbow internal and external rotation. The wrist and fingers are fixed. ‘Pointing at a target’ refers to imagery wherein T5 is pointing at the screen where he wants the cursor to go with his arm fully extended, the elbow and wrist locked. Moving left, right, up and down refer to arm adduction, abduction, flexion and extension, respectively. ‘Mouse ball’ refers to the control of a trackball mouse, requiring a combination of elbow and wrist movement, with a fixed wrist/finger orientation. T5 described the mouse ball imagery as being continuous as opposed to repetitive; that is, moving the cursor in a direction did not require repeatedly resetting his hand position. Importantly, due to T5’s injury, he was unable to actually perform these tasks. During all of the imageries, his hand and arm were comfortably at rest at his side. T5 decided to use the ‘joystick’ imagery for the very first attempt at closed-loop control.

The research session began with a 1 min reference block for computing spike threshold values and choosing channels for common average referencing. Thereafter, T5 attempted neural cursor control with the GP-DKF decoder. On the first attempt, the block was stopped after approximately 1 min when T5 indicated that he did not understand he was supposed to have been attempting motor imagery (asking ‘when should I start?’).

After being reminded that he should be using the ‘joystick’ imagery, the calibration sequence was repeated and T5 gained closed-loop neural control of the cursor (see Results).

After the ‘joystick’ imagery, T5 then achieved closed-loop control with each of the imageries in the order listed above. T5 was then asked to select the top three imageries that felt most intuitive. He selected ‘joystick’, ‘index finger’, and ‘mouse ball’. We repeated calibration with each of these imageries. We presented him with the results of target acquisition as a function of calibration time (similar in format to figure 4), and provided him with the opportunity to select a single motor imagery that felt intuitive to use for the rest of the sessions. He selected ‘mouse ball’. Thereafter, he was asked to only use ‘mouse ball’ imagery for Radial-8 and Grid Task control.

3. Results

3.1. Rapid calibration with both GP-DKF and Kalman decoders

All three iBCI users with tetraplegia rapidly gained closed-loop cursor control during the initial calibration sequence using the GP-DKF decoder (figures 2(A)–(D), supplementary figures 2 and 3 and supplementary movies 1–3). At the start of calibration, neural control was poor as expected; as more data were collected, each user gained better control in all directions. To assess performance saturation, we fit the calibration performance with an exponential

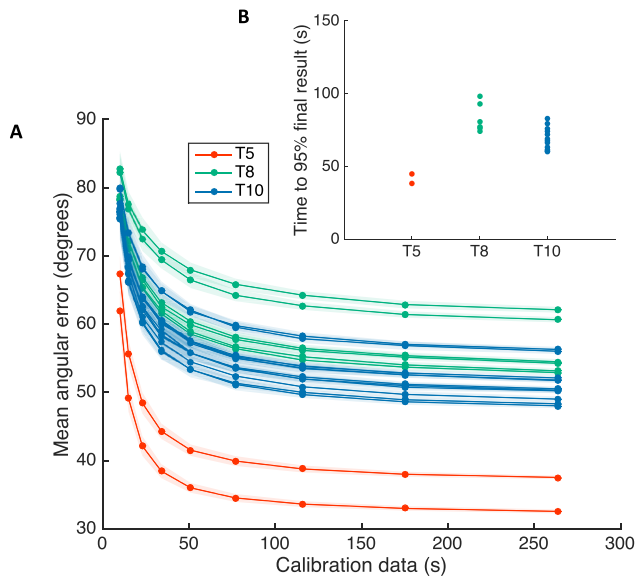


Figure 3. Bootstrapped angular error as a function of each participant's neural features used to simulate decoding. (A) For each experimental session, a decoder was trained using a random subsample, without replacement, and then used to mean angular error for another subsample of the same size. Decoder predictions were bootstrapped 100 times. Intuitively, the decoding performance would approach 90° as the amount of data approaches zero, since a poor quality decoder (e.g. one with limited training data) would decode a random angular error between 0° (perfect) and 180° (opposite) to the target. The average of a large number of random angles drawn between 0° and 180° would average to be 90° . (B) The angular error curves were fit using a decaying exponential, and the amount of data required to achieve 95% saturation of the peak angular error performance was computed. The bootstrapped mean angular error saturated in less than 3 min for all three users. (mean \pm standard deviation; participant T5: $41.6\text{ s} \pm 4.6\text{ s}$; T8: $83.3\text{ s} \pm 9.9\text{ s}$; T10: $69.3\text{ s} \pm 7.5\text{ s}$).

curve, computed the exponential time-constant, and then estimated the amount of calibration data required to achieve 95% maximal performance. We found the time to target performance saturated with the GP-DKF within 3 min for all three participants (participant T5: $43\text{ s} \pm 4\text{ s}$, T8: $136\text{ s} \pm 54\text{ s}$, T10: $100\text{ s} \pm 25\text{ s}$). Towards the end of calibration, the time to target acquisition was comparable to state-of-the-art Radial-8 task target acquisition times with fully calibrated decoders and locked parameters [17]. Note that GP-DKF calibration with T5 and T8 did not use computer assistance, whereas T8 used computer assistance for the first 60 s (see Methods [15]).

We further evaluated performance saturation for all three participants with offline simulations using a different performance metric. By bootstrapping neural features and cursor kinematics within a single experimental session, we computed the predicted the angular error between the simulated decoded direction and the known vector from cursor to target (see Methods). We found that offline predictions of decoder performance had an exponential behavior for all of the experimental sessions we tested (figure 3). We fit the angular error curves using a decaying exponential, computed the decay

half-lives, and then estimated the amount of neural data required to achieve 95% saturation of the minimum angular error. We found that offline simulations of angular error saturated in under 3 min of calibration time (mean \pm standard deviation; participant T5: $41.6\text{ s} \pm 4.6\text{ s}$; T8: $83.3\text{ s} \pm 9.9\text{ s}$; T10: $69.3\text{ s} \pm 7.5\text{ s}$).

Next, we assessed neural closed loop decoder performance at various durations of decoder calibration. We calibrated decoders for 1, 3, and 5 min, locked decoder parameters, and then quantified performance using the Grid Task (supplementary table 1, supplementary figures 4 and 5). Using the Kalman decoder, we found a statistically significant difference between bit rate at 1 min and 3 min, but did not find a statistically significant difference in bit rate between 3 min and 5 min (although there was a statistically significant difference in the number of orthogonal direction changes). Thus, we found that closed-loop neural control, as measured by communication bit-rate, saturated within 3 min of initializing calibration. For the GP-DKF decoder, we found a statistically significant difference between bit rate performance, time to target, and orthogonal direction changes at 1 min and 3 min. When comparing performance between the Kalman and GP-DKF decoders, we did not find a statistically significant difference between bit rate at 1 min and 3 min (although there was a statistically significant difference in the time to target and orthogonal direction changes).

3.2. Performance after rapid calibration

While closed-loop parameter updates provided users with closed-loop neural decoding that improved during calibration, we investigated whether there was any performance benefit that resulted from this rapid calibration protocol compared to the standard batch-based method (supplementary figure 5). Since the standard calibration method requires parameter initialization in order to provide users with closed-loop control, we initialized both decoding methods using 1 min of open loop imagery. We then calibrated both methods for 2 min without computer assistance, locked decoder parameters, and then tested each resulting decoder using the Grid Task. While neural cursor control improved during calibration, we found no statistically significant difference in bit rate, time to target, or orthogonal direction changes between calibration methods.

Finally, we investigated whether the shortened calibration provided similar performance to our standard calibration method previously described. We calibrated a GP-DKF decoder for 3 min and compared the performance against a batch base protocol using 2 min of open loop imagery and 9 min of closed loop data with gradually decreasing computer assistance [12, 14–16]. For each of the three participants, the bit rate (figure 4), time to target acquisition, and number of orthogonal direction changes of the shortened calibration method was statistically comparable to the standard calibration method. When participant data was combined, there was no statistically significant difference between calibration

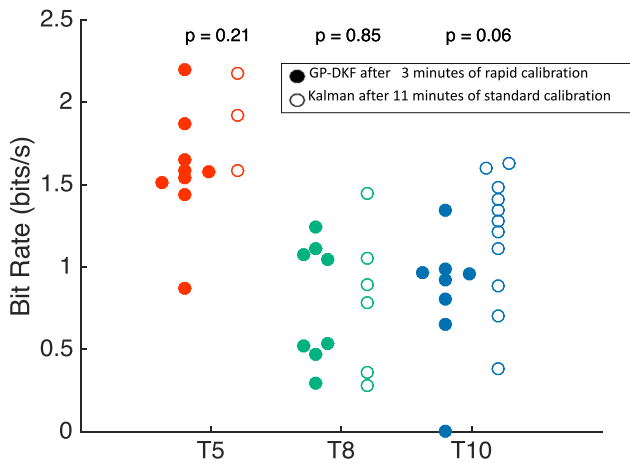


Figure 4. Bit rate comparisons between decoder methods. After calibrating the GP-DKF decoder for 3 (T5—red, T10—blue) or 4 (T8—green) minutes, the decoder parameters were locked, and the participants selected targets using the Grid Task. Bit-rates were computed for the GP-DKF decoder and compared to the performance using a Kalman filter using ~10 min of calibration data with an explicit open-loop imagery step [16]. Bit rates were not statistically different when comparing decoders within participants (Wilcoxon-rank test). Data were used from trial days: 30 and 33 (T5); 660, 662, 665 (T8); and 84, 112 (T10).

methods (median = 1.1 bit s⁻¹ for GP-DKF versus 1.3 bit s⁻¹ for standard Kalman, Wilcoxon Rank-test, $p = 0.58$).

3.3. Rapid calibration in a BCI-naïve user's first neural control session

We were interested in investigating the speed and accuracy with which a BCI-naïve user could acquire closed-loop cursor control. Prior to his first-ever BCI research session, we provided participant T5 with a high-level description of the BCI system (as had been discussed over months prior to enrollment) and detailed some principles of motor imagery (see Methods). T5 decided that he would use the imagery of his dominant hand controlling a joystick. On his first-ever BCI control session (Trial Day 30), after reporting that he understood the instructions and indicating that he was ready, T5 gained continuous, unassisted, 2D closed-loop cursor control using the GP-DKF. He acquired his first target approximately 37s after initializing calibration (figure 5, supplementary video 1). After using the system for 120s, cursor performance was comparable to the most recent state-of-the-art Radial-8 performance described in people [17].

Although T5 originally selected the ‘joystick’ imagery as being intuitive, we were also interested in exploring whether other motor imageries would be more intuitive, or result in higher quality closed-loop cursor control. Thus, following this initial calibration, we investigated six distinct motor imageries of both the proximal and distal dominant arm, allowing T5 to attempt to gain control using each of them one by one, restarting the calibration task over each time. Consistent with the heterogeneous representation of upper extremity movements at the level of individual neurons in the motor cortex [70–74], T5 was able to achieve closed-loop control using all six imageries within ~60s

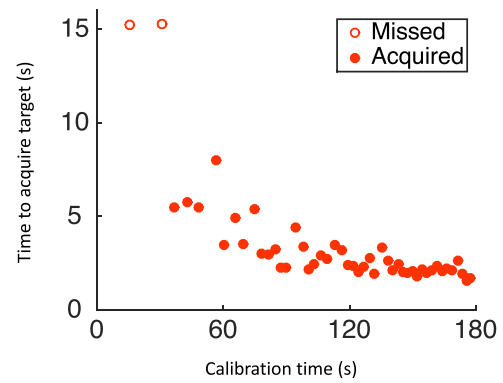


Figure 5. Time to target acquisition as a function of time on T5's first attempt at closed-loop neural control (trial day 30, block 3).

(supplementary figure 7). T5 then selected the imagery he found most intuitive: control of a mouse-ball with the dominant hand. After 3 min of calibration with the GP-DKF decoder using mouse ball imagery, the decoder parameters were locked, and performance was assessed in the Grid Task. T5 achieved a bit rate of 1.87 bits s⁻¹ on his very first attempt at the Grid Task.

4. Discussion

Three people with tetraplegia rapidly gained high-performance, 2D closed-loop neural control of a computer cursor using two markedly different approaches to neural decoding. Performance for all three participants saturated within 3 min of initiating calibration. Performance with each decoder was comparable to that obtained in previous studies using standard open-loop/closed-loop calibration routines that last ~10 min (see methods) [14–16]. In addition, a BCI-naïve man with high cervical SCI, gained closed-loop 2D neural control on his very first attempt within 2 min. This study thus provides additional progress and opportunities toward the goal of providing rapid and intuitive BCIs for people with paralysis.

The rapid calibration protocol reported here improves upon the traditional calibration sequence. Rather than explicitly using an open-loop imagery step, we provided the user with closed-loop control immediately after the first target was presented. This reduced the open-loop imagery phase from 2–5 min to ~3 s in length.

So as to focus entirely on the decoder calibration methods, the implementation of the Grid Task here relied on users dwelling over targets for a predetermined length of time to make correct (or incorrect) acquisitions. We have previously reported higher bit-rate performance by participants acquiring targets using an additional mouse-click imagery [18]; further research will combine the rapid 2D decoder calibration described here with additional state (e.g. click) detection approaches.

4.1. Future directions for iBCI systems

The ideal iBCI should provide immediate, intuitive restoration of communication and/or mobility. Doing so would make it possible to immediately restore communication for

someone who has become acutely locked-in due to brainstem stroke, where the current standard of care in intensive care units has been to rely upon patients' (often unreliable) eye blinks or vertical eye deviation for communication. Short of achieving this ideal, additional clinical and experimental imperatives include the following: minimizing the amount of user effort required to engage the system; enhancing user engagement during early use [43, 44, 60, 61]; eliminating calibration procedures that provide no feedback (i.e. an explicit open-loop imagery step) for users with impaired levels of alertness; early confirmation to users and caregivers that the BCI is working; and, a dramatic increase in research efficiency. Each of these imperatives is supported by a rapid calibration protocol.

All three participants in this study used intracortical electrode arrays [62] requiring percutaneous connections. As fully implantable iBCI systems [75] are developed that work 24 hours per day, the system should require less active intervention by trained technicians, relying instead on automated procedures to guide everyday use. For example, once the explicit target-based calibration procedure is complete, the user should be able to move on to general computer use, including communication tasks, where the cursor target is not explicitly known by the system.

To continue to calibrate the decoder during self-directed on-screen computer use, we have previously described retrospective target inference as a method of labeling neural data with the BCI users' intended movement directions based on their own selections, during self-directed on-screen keyboard use [16]. We did not investigate parameter updating during ongoing use; in future work, we hypothesize that combining retrospective target inference with the frequent decoder updates would continue to be an even more effective strategy for maintaining calibration over long periods of practical BCI use. For example, decoders could be updated more frequently—after every click or other action that denotes an intended target—rather than recalibrating only during self-timed pauses in BCI use [16]. We note that we did not explicitly examine long-term effects of learning on decoding performance. Future research would be able to disentangle performance improvements related to a user's ability to learn a neuromotor (BCI) skill versus the performance improvements that may result from adaptive decoder recalibration.

4.2. Future directions with Gaussian process regression

The described new strategy for Gaussian process regression neural decoding departs from standard linear methods described in the human iBCI literature [11–17, 19–21]. Instead of relying on an explicit function to estimate intended movement from neural data [76], decoding is based on comparing incoming data to activity patterns associated with different intended movement directions. This GP-DKF decoder is computationally tractable using standard BCI hardware, and would be reducible to a fully embedded system [35, 75].

As this is an early demonstration of GP-DKF, multiple modeling assumptions remain to be explored. For instance,

we focused on the popular radial basis kernel [77]. Additional gains may be sought by learning highly expressive kernel functions or incorporating a fully Bayesian feature selection using automatic relevance determination [77–79]. The highly non-linear nature (combined with multiple options for kernels) of the GP-DKF could provide a useful alternative to linear decoding algorithms for the control of end-effectors such as robotic arms [19–21] or functional electrical stimulation systems [23, 24].

5. Conclusions

Brain-computer interfaces have tremendous potential for improving the quality of life for individuals living with motor impairments. As the field looks towards developing BCI systems that can potentially work 24 hours per day, it will be critical to make the calibration process rapid and intuitive. The current study suggests that intracortical BCIs can provide 2D cursor control with performance plateauing within 3 min of starting calibration. Such results demonstrate an important step toward a neural prosthetic device that could be used by people with paralysis immediately upon deployment.

Author contributions

DMB was responsible for the study design, development, validation, data analysis, and writing the manuscript. SDS, LRH, JDS, BJ, KVS, and JMH contributed to study design. DMB, TH, JS, AAS, BES, JGC, FRW, DRY, BJ, SDS, and CP were responsible for validation of the system. DMB, MCB, and MTH, were responsible for the development of the GP-DKF decoder. BF, JK, BAM, PN, and CB were responsible for data collection. DMB, MCB, BES, JGC, DJM, and CVI conducted offline analyses to inform algorithm design of the GP-DKF decoder. LRH is the sponsor-investigator of the multi-site pilot clinical trial. SSC assisted in planning T10 array placement and is a co-investigator of the pilot clinical trial. ENE placed the arrays in T10. JMH placed the arrays in T5. JPM and JAS placed arrays in T8. BLW provided clinical support and ABA, and RFK provided programmatic integration with participant T8. All authors reviewed and contributed to the manuscript.

Acknowledgements

The authors would like to thank participants T5, T8, T10, and their families; B Travers, and D Rosler for administrative support; S K Miller for manuscript editing; L Barefoot, C Grant, M Bowker, and S Mernoff for clinical assistance; L Ball for references; and M Vilela, P Nuyujukian, J P Donoghue for helpful scientific discussions. This work was supported by the National Institutes of Health: National Institute on Deafness and Other Communication Disorders—NIDCD (R01DC009899, R01DC014034), Eunice Kennedy Shriver National Institute of Child Health and Human Development/National Center for Medical Rehabilitation Research—NICHD/NCMRR (R01HD077220); Rehabilitation Research and Development

Service, Department of Veterans Affairs (B6453R, A6779I, N9228C, B4853C); National Science Foundation (1309004); Massachusetts General Hospital (MGH)—Deane Institute for Integrated Research on Atrial Fibrillation and Stroke; Joseph Martin Prize for Basic Research; The Executive Committee on Research (ECOR) of Massachusetts General Hospital; Stanford Institute for Neuro-Innovation and Translational Neuroscience; Stanford BioX-NeuroVentures; Stanford Office of Postdoctoral Affairs; ALSA Milton Safenowitz Postdoctoral Fellowship; Garlick Foundation; Katie Samson Foundation; Craig H Neilsen Foundation; Canadian Institute of Health Research (336092); Killam Trust Award Foundation; Brown Institute of Brain Science. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health, the Department of Veterans Affairs or the United States Government.

ORCID iDs

David M Brandman  <https://orcid.org/0000-0003-3224-7019>
Francis R Willett  <https://orcid.org/0000-0002-2652-8511>
Leigh R Hochberg  <https://orcid.org/0000-0003-0261-2273>

References

- [1] Donoghue J P 2002 Connecting cortex to machines: recent advances in brain interfaces *Nat. Neurosci.* **5** 1085–8
- [2] Donoghue J P, Nurmikko A, Black M and Hochberg L R 2007 Assistive technology and robotic control using motor cortex ensemble-based neural interface systems in humans with tetraplegia *J. Physiol.* **579** 603–11
- [3] Serruya M D, Hatsopoulos N G, Paninski L, Fellows M R and Donoghue J P 2002 Instant neural control of a movement signal *Nature* **416** 141–2
- [4] Wolpaw J R, Birbaumer N, McFarland D J, Pfurtscheller G and Vaughan T M 2002 Brain–computer interfaces for communication and control *Clin. Neurophysiol.* **113** 767–91
- [5] Lebedev M A and Nicolelis M A L 2006 Brain–machine interfaces: past, present and future *Trends Neurosci.* **29** 536–46
- [6] Schwartz A B, Cui X T, Weber D J and Moran D W 2006 Brain-controlled interfaces: movement restoration with neural prosthetics *Neuron* **52** 205–20
- [7] Fetz E E 2007 Volitional control of neural activity : implications for brain–computer interfaces *J. Physiol.* **579** 571–9
- [8] Gilja V, Chestek C A, Diester I, Henderson J M, Deisseroth K and Krishna S V 2011 Challenges and opportunities for next-generation intracortically based neural prostheses *IEEE Trans. Biomed. Eng.* **58** 1891–9
- [9] Carmena J M 2013 Advances in neuroprosthetic learning and control *PLoS Biol.* **11** 1–4
- [10] Chestek C A, Cunningham J P, Gilja V, Nuyujukian P, Ryu S I and Shenoy K V 2009 Neural prosthetic systems: current problems and future directions *2009 Annual Int. Conf. of the IEEE Engineering in Medical and Biology Society* pp 3369–75
- [11] Kim S-P, Simeral J D, Hochberg L R, Donoghue J P and Black M J 2008 Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia *J. Neural Eng.* **5** 455–76
- [12] Simeral J D, Kim S-P, Black M J, Donoghue J P and Hochberg L R 2011 Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array *J. Neural Eng.* **8** 25027
- [13] Hochberg L R, Serruya M D, Friehs G M, Mukand J A, Saleh M, Caplan A H, Branner A, Chen D, Penn R D and Donoghue J P 2006 Neuronal ensemble control of prosthetic devices by a human with tetraplegia *Nature* **442** 164–71
- [14] Bacher D, Jarosiewicz B, Masse N Y, Stavisky S D, Simeral J D, Newell K, Oakley E M, Cash S S, Friehs G and Hochberg L R 2015 Neural point-and-click communication by a person with incomplete locked-in syndrome *Neurorehabil. Neural Repair* **29** 462–71
- [15] Jarosiewicz B, Masse N Y, Bacher D, Cash S S, Eskandar E, Friehs G, Donoghue J P and Hochberg L R 2013 Advantages of closed-loop calibration in intracortical brain–computer interfaces for people with tetraplegia *J. Neural Eng.* **10** 46012
- [16] Jarosiewicz B et al 2015 Virtual typing by people with tetraplegia using a self-calibrating intracortical brain–computer interface *Sci. Transl. Med.* **7** 1–11
- [17] Gilja V et al 2015 Clinical translation of a high-performance neural prosthesis *Nat. Med.* **21** 1142–5
- [18] Pandarinath C, Nuyujukian P, Blabe C H, Sorice B L, Saab J, Willett F, Hochberg L R, Shenoy K V and Henderson J M 2017 High Performance communication by people with paralysis using an intracortical brain–computer interface *Elife* **21** 1–27
- [19] Hochberg L R et al 2012 Reach and grasp by people with tetraplegia using a neurally controlled robotic arm *Nature* **485** 372–5
- [20] Collinger J L, Wodlinger B, Downey J E, Wang W, Tyler-Kabara E C, Weber D J, McMorland A J C, Velliste M, Boninger M L and Schwartz A B 2013 High-performance neuroprosthetic control by an individual with tetraplegia *Lancet* **381** 557–64
- [21] Wodlinger B, Downey J E, Tyler-Kabara E C, Schwartz A B, Boninger M L and Collinger J L 2015 Ten-dimensional anthropomorphic arm control in a human brain machine interface: difficulties, solutions, and limitations *J. Neural Eng.* **12** 16011
- [22] Aflalo T et al 2015 Decoding motor imagery from the posterior parietal cortex of a tetraplegic human *Science* **348** 906–10
- [23] Bouton C E et al 2016 Restoring cortical control of functional movement in a human with quadriplegia *Nature* **533** 247–50
- [24] Ajiboye A B et al 2017 Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration *Lancet* **389** 1821–30
- [25] Shenoy K and Carmena J 2014 Combining decoder design and neural adaptation in brain–machine interfaces *Neuron* **84** 665–80
- [26] Birbaumer N, Ghanayim N, Hinterberger T, Iversen I, Kotchoubey B, Kübler A, Perelmouter J, Taub E and Flor H 1999 A spelling device for the paralysed *Nature* **398** 297–8
- [27] Rockstroh B, Birbaumer N, Elbert T and Lutzenberger W 1984 Operant control of EEG and event-related and slow brain potentials *Biofeedback Self-Regul.* **9** 139–60
- [28] Elbert T, Rockstroh B, Lutzenberger W and Birbaumer N 1980 Biofeedback of slow cortical potentials. I *Electroencephalogr. Clin. Neurophysiol.* **48** 293–301
- [29] Fetz E E 1969 Operant conditioning of cortical unit activity *Science* **163** 955–8
- [30] Ganguly K and Carmena J M 2009 Emergence of a stable cortical map for neuroprosthetic control *PLoS Biol.* **7** 1–13

- [31] Ganguly K and Carmena J M 2010 Neural correlates of skill acquisition with a cortical brain-machine interface *J. Mot. Behav.* **42** 355–60
- [32] Sadtler P T, Quick K M, Golub M D, Chase S M, Ryu S I, Tyler-Kabara E C, Yu B M and Batista A P 2014 Neural constraints on learning *Nature* **512** 423–6
- [33] Jarosiewicz B, Chase S M, Fraser G W, Velliste M, Kass R E and Schwartz A B 2008 Functional network reorganization during learning in a brain-computer interface paradigm *Proc. Natl Acad. Sci. USA* **105** 19486–91
- [34] Kennedy P R and Bakay R A 1998 Restoration of neural output from a paralyzed patient by a direct brain connection *Neuroreport* **9** 1707–11
- [35] Vansteensel M J et al 2016 Fully implanted brain-computer interface in a locked-in patient with ALS *New Engl. J. Med.* **375** 2060–6
- [36] Santhanam G, Ryu S I, Yu B M, Afshar A and Shenoy K V 2006 A high-performance brain-computer interface *Nature* **442** 195–8
- [37] Taylor D M, Tillery S I H and Schwartz A B 2002 Direct cortical control of 3D neuroprosthetic devices *Science* **296** 1829–32
- [38] Wessberg J, Stambaugh C R, Kralik J D, Beck P D, Laubach M, Chapin J K, Kim J, Biggs S J, Srinivasan M A and Nicolelis M A 2000 Real-time prediction of hand trajectory by ensembles of cortical neurons in primates *Nature* **408** 361–5
- [39] Velliste M, Perel S, Spalding M C, Whitford A S and Schwartz A B 2008 Cortical control of a prosthetic arm for self-feeding *Nature* **453** 1098–101
- [40] Kao J C, Nuyujukian P, Ryu S I, Churchland M M, Cunningham J P and Shenoy K V 2015 Single-trial dynamics of motor cortex and their applications to brain-machine interfaces *Nat. Commun.* **6** 7759
- [41] Willett F R et al 2017 Feedback control policies employed by people using intracortical brain-computer interfaces *J. Neural Eng.* **14** 16001
- [42] Boninger M, Mitchell G, Tyler-Kabara E, Collinger J and Schwartz A B 2013 Neuroprosthetic control and tetraplegia—Authors' reply *Lancet* **381** 1900–1
- [43] Orsborn A L, Dangi S, Moorman H G and Carmena J M 2012 Closed-loop decoder adaptation on intermediate time-scales facilitates rapid BMI performance improvements independent of decoder initialization conditions *IEEE Trans. Neural Syst. Rehabil. Eng.* **20** 468–77
- [44] Shanechi M M, Orsborn A L and Carmena J M 2016 Robust brain-machine interface design using optimal feedback control modeling and adaptive point process filtering *PLoS Comput. Biol.* **12** 1–29
- [45] Fazli S, Popescu F, Danóczy M, Blankertz B, Müller K R and Grozea C 2009 Subject-independent mental state classification in single trials *Neural Netw.* **22** 1305–12
- [46] Vidaurre C, Sannelli C, Müller K-R and Blankertz B 2011 Co-adaptive calibration to improve BCI efficiency *J. Neural Eng.* **8** 025009–17
- [47] Ray A M et al 2015 A subject-independent pattern-based brain-computer interface *Frontiers Behav. Neurosci.* **9** 1–15
- [48] Arvaneh M, Guan C, Ang K K and Quek C 2012 Omitting the intra-session calibration in EEG-based brain computer interface used for stroke rehabilitation *Proc. Annual Int. Conf. IEEE Engineering in Medical and Biology Society EMBS* pp 4124–7
- [49] Orsborn A L, Moorman H G, Overduin S A, Shanechi M M, Dimitrov D F and Carmena J M 2014 Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control *Neuron* **82** 1380–93
- [50] Orsborn A L, Dangi S, Moorman H G and Carmena J M 2011 Exploring time-scales of closed-loop decoder adaptation in brain-machine interfaces *Proc. Annual Int. Conf. IEEE Engineering in Medical and Biology Society EMBS* pp 5436–9
- [51] Sussillo D, Churchland M M, Kaufman M T and Shenoy K V 2015 A neural network that finds a naturalistic solution for the production of muscle activity *Nat. Neurosci.* **18** 1025–33
- [52] Nuyujukian P, Fan J M, Kao J C, Ryu S I and Shenoy K V 2015 A high-performance keyboard neural prosthesis enabled by task optimization *IEEE Trans. Biomed. Eng.* **62** 21–9
- [53] Koyama S, Chase S M, Whitford A S, Velliste M, Schwartz A B and Kass R E 2010 Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control *J. Comput. Neurosci.* **29** 73–87
- [54] Chase S M, Schwartz A B and Kass R E 2009 Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms *Neural Netw.* **22** 1203–13
- [55] Carmena J M, Lebedev M A, Crist R E, O'Doherty J E, Santucci D M, Dimitrov D F, Patil P G, Henriquez C S and Nicolelis M A L 2003 Learning to control a brain-machine interface for reaching and grasping by primates *PLoS Biol.* **1** 193–208
- [56] Shanechi M M, Orsborn A, Moorman H, Gowda S and Carmena J M 2014 High-performance brain-machine interface enabled by an adaptive optimal feedback-controlled point process decoder *Conf. Proc. ... Annu. Int. Conf. IEEE Engineering in Medical Biology Society* vol 2014 pp 6493–6
- [57] Dangi S, Gowda S, Heliot R and Carmena J M 2011 Adaptive Kalman filtering for closed-loop brain-machine interface systems *2011 5th Int. IEEE/EMBS Conf. Neural Eng. NER 2011* pp 609–12
- [58] Shpigelman L, Lalazar H and Vaadia E 2008 Kernel-ARMA for hand tracking and brain-machine interfacing during 3D motor control *Neural Inf. Process. Syst.* 1489–96
- [59] Perge J A, Homer M L, Malik W Q, Cash S, Eskandar E, Friehs G, Donoghue J P and Hochberg L R 2013 Intra-day signal instabilities affect decoding performance in an intracortical neural interface system *J. Neural Eng.* **10** 36004
- [60] Barbero A and Grosse-Wentrup M 2010 Biased feedback in brain-computer interfaces *J. Neuroeng. Rehabil.* **7** 34
- [61] Acqualagna L, Botrel L, Vidaurre C, Kübler A and Blankertz B 2016 Large-scale assessment of a fully automatic co-adaptive motor imagery-based brain computer interface *PLoS One* **11** 1–19
- [62] Maynard E M, Nordhausen C T and Normann R A 1997 The Utah intracortical electrode array: a recording structure for potential brain-computer interfaces *Electroencephalogr. Clin. Neurophysiol.* **102** 228–39
- [63] Masse N Y et al 2015 Non-causal spike filtering improves decoding of movement intention for intracortical BCIs *J. Neurosci. Methods* **244** 94–103
- [64] Malik W Q, Hochberg L R, Donoghue J P, Brown E N, Member S, Hochberg L R, Donoghue J P and Brown E N 2015 Modulation depth estimation and variable selection in state-space models for neural interfaces *IEEE Trans. Biomed. Eng.* **62** 570–81
- [65] Wu W, Gao Y, Bienenstock E, Donoghue J P, Black M J, Bienenstock E, Donoghue J P and Black M J 2005 Bayesian population decoding of motor cortical activity using a Kalman filter *Neural Comput.* **18** 80–118
- [66] Wu W, Black M J, Gao Y, Bienenstock E, Serruya M, Shaikhouni A and Donoghue J P 2003 Neural decoding of cursor motion using a kalman filter *Advances in Neural*

- Information Processing Systems 15 Proc. 2002 Conf.* pp 133–40
- [67] Burkhart M C, Brandman D M, Vargas-Irwin C E and Harrison M T 2016 The discriminative Kalman filter for nonlinear and non-Gaussian sequential Bayesian filtering **1** 1–11 (arXiv:1608.06622)
- [68] Bishop C M 2006 *Pattern Recognition and Machine Learning* (Berlin: Springer)
- [69] Yuan P, Gao X, Allison B, Wang Y, Bin G and Gao S 2013 A study of the existing problems of estimating the information transfer rate in online brain–computer interfaces *J. Neural Eng.* **10** 26014
- [70] Vargas-Irwin C E, Brandman D M, Zimmermann J B, Donoghue J P and Black M J 2015 Spike train SIMilarity space (SSIMS): a framework for single neuron and ensemble data analysis *Neural Comput.* **27** 1–31
- [71] Shenoy K V., Sahani M and Churchland M M 2013 Cortical control of arm movements: a dynamical systems perspective *Annu. Rev. Neurosci.* **36** 337–59
- [72] Schwartz A B 1994 Direct cortical representation of drawing *Science* **265** 540–2
- [73] Paninski L, Fellows M R, Hatsopoulos N G and Donoghue J P 2004 Spatiotemporal tuning of motor cortical neurons for hand position and velocity spatiotemporal tuning of motor cortical neurons for hand position and velocity *J. Neurophysiol.* **91** 515–32
- [74] Pohlmeier E A, Solla S A, Perreault E J and Miller L E 2007 Prediction of upper limb muscle activity from motor cortical discharge during reaching *J. Neural Eng.* **4** 369–79
- [75] Yin M, Borton D A, Aceros J, Patterson W R and Nurmikko A V. 2013 A 100-channel hermetically sealed implantable device for chronic wireless neurosensing applications *IEEE Trans. Biomed. Circuits Syst.* **7** 115–28
- [76] Kass R E, Ventura V and Brown E N 2005 Statistical issues in the analysis of neuronal data *J. Neurophysiol.* **94** 8–25
- [77] Rasmussen C and Williams C 2006 *Gaussian Processes for Machine Learning* (London: The MIT Press)
- [78] Park I M, Seth S, Paiva A R C, Li L and Principe J C 2013 Kernel methods on spike train space for neuroscience: a tutorial *IEEE Signal Process. Mag.* **30** 149–60
- [79] Brockmeier A J, Choi J S, Kriminger E G, Francis J T and Principe J C 2014 Neural decoding with Kernel-based metric learning *Neural Comput.* **26** 1080–107