Feasibility of Automatic Error Detect-and-Undo System in Human Intracortical Brain–Computer Interfaces

Nir Even-Chen, *Student Member, IEEE*, Sergey D. Stavisky, *Member, IEEE*, Chethan Pandarinath, *Member, IEEE*, Paul Nuyujukian, *Member, IEEE*, Christine H. Blabe, Leigh R. Hochberg, Jaimie M. Henderson, and Krishna V. Shenoy, *Senior Member, IEEE*

***Abstract*—*Objective:* Brain–computer interfaces (BCIs) aim to help people with impaired movement ability by directly translating their movement intentions into command signals for assistive technologies. Despite large performance improvements over the last two decades, BCI systems still make errors that need to be corrected manually by the user. This decreases system performance and is also frustrating for the user. The deleterious effects of errors could be mitigated if the system automatically detected when the user perceives that an error was made and automatically intervened with a corrective action; thus, sparing users from having to make the correction themselves. Our previous preclinical work with monkeys demonstrated that task-outcome correlates exist in motor cortical spiking activity and can be utilized to improve BCI performance. Here, we asked if these signals also exist in the human hand area of motor cortex, and whether they can be decoded with high accuracy. *Methods:* We analyzed posthoc the intracortical neural activity of two BrainGate2 clinical trial participants who were neurally controlling a computer cursor to perform a grid target selection task and a keyboard-typing task. *Results:* Our key findings are that: 1) there exists a putative outcome error signal reflected in both the action potentials and local field potentials of the human hand area of motor cortex, and 2) target selection outcomes can be classified with high accuracy (70–85%) of errors successfully detected with minimal (0–3%) misclassifications of success trials, based on neural activity alone. *Significance:* These offline results suggest that it will be possible to improve the performance of clinical intracortical BCIs by incorporating a real-time error detect-and-undo system alongside the decoding of movement intention.**

***Index Terms*—Brain-computer interface, clinical trial, error detection, motor cortex, task outcome.**

I. Introduction

BRAIN-COMPUTER interfaces (BCIs) are devices that estimate a user's movement intention from neural activity from the brain to guide an assistive device such as a prosthetic arm or a computer cursor. They aim to help people with motor impairment (e.g., due to amyotrophic lateral sclerosis, brainstem stroke, or cervical spinal cord injury) in the ability to communicate (e.g., controlling a cursor to type and use a computer) or through restored mobility. BCIs typically record neural activity through different modalities such as electroencephalography (EEG) [1]–[4], electrocorticography (ECoG) [5]–[9] or intracortical multielectrode arrays, which are typically chronically implanted in the motor cortex [10]–[24]. Intracortical BCIs (iBCI) have shown promising results in pilot clinical trials and are the highest-performing BCI systems to date, making them prime candidates for serving as an assistive technology for people with paralysis [25]–[28]. Although the performance of iBCI

Manuscript received July 26, 2017; revised October 2, 2017; accepted November 7, 2017. Date of publication November 21, 2017; date of current version July 17, 2018. This work was supported in part by Stanford Bio-X Fellowship, Stanford Office of Postdoctoral Affairs; in part by Craig H. Neilsen Foundation; in part by ALS Association Milton Safenowtiz Postdoctoral Fellowship, in part by Stanford BioX-NeuroVentures, in part by Stanford Institute for Neuro-Innovation and Translational Neuroscience; in part by Larry and Pamela Garlick; in part by Samuel and Betsy Reeves; NIH-NIDCD R01DC014034; in part by NIH-NINDS R01NS066311; in part by NIH-NIDCD R01DC009899; in part by Rehabilitation Research and Development Service, in part by Department of Veterans Affairs (B6453R and N9288C); in part by The Executive Committee on Research of Massachusetts General Hospital; and in part by MGH-Deane Institute for Integrated Research on Atrial Fibrillation and Stroke. *(Corresponding author: Krishna V. Shenoy).*

N. Even-Chen is with the Department of Electrical Engineering, Stanford University.

S. D. Stavisky is with the Department of Electrical Engineering and Neurosurgery, Stanford University.

C. Pandarinath is with the Department of Electrical Engineering and Neurosurgery, Stanford University, and also with the Department of Biomedical Engineering, Emory University and Georgia Tech.

P. Nuyujukian is with the Department of Bioengineering, Neurosurgery, and Electrical engineering, Stanford Neuroscience Institute, and Bio-X Programs, Stanford University.

C. H. Blabe is with the Department of Neurosurgery, Stanford University.

L. R. Hochberg is with Center for Neurorestoration and Neurotechnology, Rehabilitation R&D Service, VA Medical Center, School of Engineering and the Institute for Brain Science, Brown University, and also with the Center for Neurotechnology and Neurorecovery, Department of Neurology, Massachusetts General Hospital and Department of Neurology, Harvard Medical School, Synchron Med (Scientific Advisory Board), Stanford, CA 94305-5453 USA.

J. M. Henderson is with the Department of Neurosurgery and Stanford Neuroscience Institute, Stanford University.

K. V. Shenoy is with the Department of Electrical Engineering, Bioengineering and Neurobiology, Stanford Neuroscience Institute, and the Neurosciences and Bio-X Programs, Stanford, and also with the Howard Hughes Medical Institute, Stanford University (e-mail: shenoy@stanford.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBME.2017.2776204

systems has markedly improved in the last two decades, errors – such as selecting the wrong key during typing as a result of decoder or user error – still occur. The errors can be due to either user's mistake or BCI misinterpretation of user intention. Nevertheless, they decrease the performance.

While much work has and continues to be done to mitigate iBCI errors by improving the accuracy and reliability of movement intention decoders [29]–[31], here we explore a complimentary and less explored strategy: identifying when an error occurs so that the iBCI system can automatically correct for it. In our recent work with monkeys, we have successfully augmented the iBCI with a neurally driven error detection system [32]. The error detection system detected errors in real-time and intervened automatically with a corrective action to prevent or undo the errors. The new system improved the iBCI performance, especially during challenging tasks, when errors were more frequent. Similar strategies were also employed successfully in EEG-based BCIs with discrete decoding for trial-based typing [33]–[35], trial-based movements [36], and prosthetic device manipulation [37], [38]. This 'error detect-and-undo' strategy takes advantage of the closed-loop nature of a BCI: the user has constant visual feedback, and s/he is aware of when the BCI performs an unintended action (i.e., an error). The user's cerebral activity will reflect the detection of errors, though it is unknown if the same cortical areas being recorded for iBCI use in human will contain the salient neural activity reflecting error detection. If these neural correlates can be found and decoded, they could be used by the iBCI to execute an automatic corrective actions in real-time (e.g., automatically delete the last selected key), spare the user from deleting them manually and increase performance.

Prior studies have identified outcome error signals in the human brain, at the coarser resolution of EEG (reviewed in [33]), or more specifically in motor cortex using ECoG [39]. For high-performing iBCIs to perform error detection, an outcome error signal would ideally be available from the same multielectrode arrays used to decode movement intentions. Encouragingly, our prior study found that this is indeed the case in monkey motor cortical spiking activity [32]. However, the ultimate application of iBCI error detection is for people with paralysis, and thus it is crucial to test whether these signals also exist in human motor cortex, and more specifically in the hand area of the precentral gyrus, which is successfully used for iBCI. In this work, we asked two key questions that must be answered in the affirmative for automated error detection and mitigation to be a viable strategy for clinical iBCI systems: (1) do outcome error signals exist in the human hand area of motor cortex and (2) if so, can they be accurately and quickly decoded on single trials? We report here that both of these are indeed the case. To better predict the utility of exploiting this signal in an iBCI, we propose an error detect-and-undo system design and estimate the performance improvement that could be expected across a range of scenarios.

We extended the error decoding methods we previously developed in preclinical monkey experiments to the case of human motor cortical data from the BrainGate2 clinical trial. We offline analyzed intracortical recordings from the motor cortex of two participants performing a grid task (a cued target among a grid of selectable targets) and a typing task with a brain-controlled cursor to investigate the potential of error detection in iBCI.

II. Methods

Permission for these studies was granted by the US Food and Drug Administration (Investigational Device Exemption) and Institutional Review Boards of Stanford University (protocol #20804), Partners Healthcare / Massachusetts General Hospital (2011P001036), Providence VA Medical Center (2011-009), and Brown University (0809992560). The two participants in this study, T5 and T6, were enrolled in a pilot clinical trial of the BrainGate2 Neural Interface System (http://www.clinicaltrials.gov/ct2/show/NCT00912041). Informed consent, including consent to publish, was obtained from the participants prior to their enrollment in the study. Additional permission was obtained to publish participant photos and reproduce text typed by the participants.

A. Participants

Participant T6 is a right-handed woman, 51 years old at the time of the study whose data was used for [28], who was diagnosed with ALS and had a resultant motor impairment (functional rating scale (ALSFRS-R) measurement of 16). In December 2012, a 96-channel intracortical silicon microelectrode array (1.0 mm electrode length, Blackrock Microsystems, Salt Lake City, UT) was implanted in the hand area of dominant (left) motor cortex (Fig. 1).

Participant T5 is a right-handed man, 63 years old at the time of the study whose data was used for [28], who was diagnosed with a C2-3 ASIA C spinal cord injury approximately nine years prior to study enrollment. In August 2016, participant T5 had two 96-channel intracortical silicon microelectrode arrays (1.5 mm electrode length, Blackrock Microsystems, Salt Lake City, UT) implanted in the arm-hand area of dominant (left) motor cortex (Fig. 1).

B. BCI

For the present study, neural control and task cuing were controlled by custom software running on the Simulink/xPC real-time platform (The Mathwork, Natick, MA), enabling millisecond-timing precision for all computations. Neural data were collected by the NeuroPort System (Blackrock Microsystems, Salt Lake City, UT) and available to the real-time system with 5 ms latency.

Two-dimensional continuous control of the cursor was enabled by the ReFIT Kalman Filter detailed in [17], [27]. Users could select a target by dwelling on it for 1 s or by a discrete "click" signal. Discrete selection ("click") was achieved using a Hidden Markov Model (HMM)-based state classifier, which was previously developed with non-human primates [20] and adapted for the current work. The users commanded a "click" by attempting to squeeze their left hand (i.e., the hand ipsilateral to the array(s)). While for T5 the control algorithms used only spiking (thresholded action potential) activity, for T6 both spikes

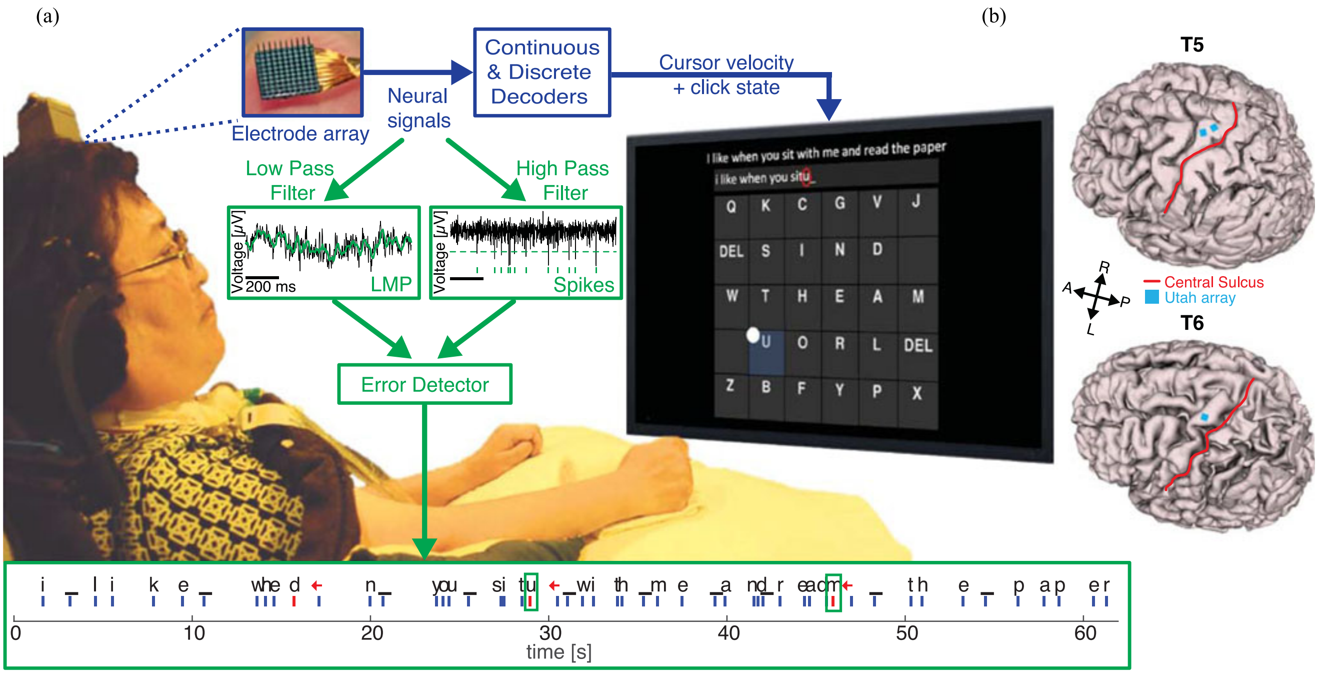

Fig. 1. Experiment layout and conceptual example of error detect-and-undo. (a) Image shows participant T6 copying a sentence during a typing task block. Schematic overview shows online iBCI control signal path in blue. Neural activity is recorded from the hand area of motor cortex using multi-electrode array(s) and mapped to cursor velocity with a continuous decoder. A parallel discrete decoder detects the intention to select a key (Methods). In an offline analysis (green-colored path), we extracted the spikes signal and the local motor potential (LMP) and fed them into a task outcome classifier that identifies when a key selection was erroneous. The bottom timeline shows an example of T6 typing a sentence using the iBCI ('i like when you sit with me and read the paper'). Key selection times are shown with vertical ticks (red ticks for errors), with the selected character shown above each tick ('_' denotes space, '←' denotes backspace). Selections that the outcome-error decoder flagged as erroneous are marked in green squares: here, the classifier was able to detect the two of the three errors in the example sentence ('u' and 'm', marked with green boxes). Note that since there was no error classifier during the online research session, the participant subsequently had to select the backspace key ('←') after both of these errors. Experimental layout figure modified from [27]. (b) Participant's fMRI imaging (processed with FreeSurfer) and arrays location in the hand area of the motor cortex (A = anterior, P = posterior, L = left, R = right).

and high frequency local field potentials (HF-LFP, representing spectral power in the 150–450 Hz frequency band) were used to compensate for a lower recorded spike signal quality.

For both the continuous cursor-positioning ReFIT-KF decoder and the discrete click-state HMM decoder, neural data were binned every 15 ms and sent through the decoders. Thus, for the ReFIT-KF decoder, updated cursor velocity estimates were provided every 15 ms for use in the rest of the iBCI system. This velocity was integrated to update the cursor position estimate every 1 ms, and therefore the most recent cursor position was sent to the display every 1 ms. The computer monitor was updated every 8.3 ms (i.e., at the 120 Hz frame rate of the monitor) with the most recent estimate of the desired cursor position.

C. Tasks

Typing task: The participants copied sentences by moving a computer cursor via decoded movement intentions to select letters on an on-screen OPTI-II keyboard (Fig. 1, [28]). Details of how the user commanded the cursor's velocity and made a "click" or "dwell" selection are explained in the "BCI" Methods section. During typing tasks, the participant saw the keyboard and cursor, as well as a field where entered text appeared with the prompted text above it.

Grid task: In this task, a grid spanning 1000 × 1000 pixels on the computer monitor was divided into a 6 × 6 or 9 × 9 grid of equally-sized gray squares [14], [28]. Each square was a selectable target, and on each trial, one square would randomly be prompted as the correct target by changing its color to green. The participant had to select the correct target (which resulted in a trial success) while avoiding selecting any of the other (incorrect) targets, which resulted in a trial failure. Correct versus incorrect selection was indicated by different audio tones immediately after the selection. After a selection had been made (correct or incorrect), a new target was immediately prompted. The overall workspace that the user could move the cursor in was 1078 × 1078 pixels, meaning that there was a small border around the grid with no selectable targets. This task is quite similar to the grid task performed in recent non-human primate (rhesus monkey) BCI experiments [32].

D. Offline Data Preprocessing

The NeuroPort data acquisition system samples each electrode at 30 kHz and applies an analog 0.3 Hz to 7.5 kHz band-pass filter to each electrode. To remove noise common to all electrodes, a common average reference was subtracted from each electrode [28]. While the online original Kalman filter used 15 ms bins, for the present offline analysis we chose to use slightly longer time bins (20 ms) to better smooth neural firing rates and field potential signals.

TABLE I
DATA DESCRIPTION

	# Days	# Trials	Success Rate	# Successes / # Failures
T5-grid	3	2579	98%	2516/63
T5-typing	2	536	93%	498/38
T6-grid	5	1057	93%	980/77
T6-typing	5	801	93%	747/54

Numbers of recorded days and analyzed trials for each task, and the total success rates (across all analyzed trials), for both participants. These same data appear in Pandarinath et al. 2017. T5-grid days are: 2016-10-12, 2016-10-13, and 2016-10-24; T5-typing days are: 2016-10-12 and 2016-10-24. T6-grid and T6-typing days are: 2014-06-30, 2014-07-02, 2014-07-07, 2014-07-18, and 2014-07-21.

Spikes: To extract neural spiking activity, a 250 Hz high-pass filter was applied to the data recorded by the NeuroPort system. A threshold detector was then applied every millisecond (in a causal 30 samples window) to detect the presence of a putative neural spike and binned in 20 ms non-overlapping bins. Choice of threshold was specific to each participant (T6: -50 μV; T5: -95 μV).

Local field potential (LFP): To extract the LFP signal a 250 Hz low-pass filter was applied to the recorded neural signal and was resampled in 1 KHz. To compute the LFP power for each frequency band, we computed the mean power of the bandpass filtered signal (3rd order Butterworth, see Sup. Fig 4(b) for magnitude frequency response).

Local motor potential (LMP): To compute the LMP, we binned the LFP signal in 20 ms bins, which is equivalent to applying a causal boxcar filter with 20 ms width (see Sup. Fig. 4(b) for magnitude frequency response).

Cursor Speed: The cursor speed was computed from the recorded online cursor position and binned in 20 ms non-overlapping bins.

Data description and average success rates for each task and participant are presented in Table I. In all analyses, the different experiment session (days) were pooled and analyzed as one data set. In both participants, erroneous target selections (failed trials) were rare but nonetheless provided enough data to evaluate whether these events could be detected from neural activity alone (see below). For both participants more than 95% of the errors were in the area of the targets that are adjacent to the cued targets. Only 11% (T5) and 13% (T6) of the trials were selected by dwelling on the target; those trials were removed from the offline analysis since their number was too low for statistical analysis, further work should be done to investigate the error signal when target is selected by dwelling.

E. Dimensionality Reduction

The recorded neural activity is composed of many processes that are related to kinematics, kinetics, and perhaps also to task outcome (the latter being the key question we investigated). Some of these processes are similar during both successful and failed trials, and thus even if they explain a large fraction of the overall neural variance, these are not dimensions of the data that we are interested in for the purpose of decoding trial outcome. Rather, here we wanted to perform targeted dimensionality reduction to reduce the number of neural features at each time bin from the full-dimensional space of spike counts and LFP power on each electrode down to a smaller number of features that still capture much of the variance in the neural signal component that is different between failed and successful trials (i.e., the putative error signal). While all of the putative task outcome information is in principle available in the full-dimensional neural data, dimensionality reduction is a widely used technique [40] to both de-noise single-trial data and to reduce the number of classifier parameters, which helps avoid overfitting limited training data.

The relationship between any two conditions, such as successful and failed trials, can be represented with a common mode (their average, $\mathrm{Y_{cm}} = (\mathrm{Y_{suc}} + \mathrm{Y_{fail}})/2$) and a differential mode (their difference, $\mathrm{Y_{diff}} = \mathrm{Y_{suc}} - \mathrm{Y_{fail}}$). Here, the common mode contains activity presumably related to performing the task but unrelated to the specific outcome. To focus on the difference between outcomes while trying to filter out common processes, we performed principal component analysis (PCA) on the differential mode; i.e., the difference in the neural activity between the trial-averaged successful and failed trials [32] (see Fig. 2(a)). This identifies a neural subspace (i.e., a linear weighting across electrode features) characterized by highly time-varying patterns of neural activity that differ between successful and failed trials. Projecting neural activity into this subspace will tend to minimize the neural covariation patterns that are common between successful and failed trials, and will tend to emphasize outcome-dependent activity patterns. We note that although the dimensionality-reducing linear transformation matrix was identified, via PCA, from trial-averaged neural data (to better estimate the true outcome-dependent signal component), we subsequently also applied this transformation to individual time bins of neural data to measure how well we could classify task outcome on single trials. Doing so does not require knowing a priori whether the data being projected through the PCA matrix comes from a successful or failed trial. This outcome-naive single-trial evaluation is critical, since this is the regime in which a closed-loop error detect-and-undo system would operate in. It is worth noting that although the dimensionality-reducing PCA step was optimized on trial-averaged data, the classifier training operates on single-trial examples and therefore will take into account inter-trial variability when determining the class decision boundary. That said, we recognize that it may be possible to improve performance in future work by using dimensionality reduction techniques that explicitly consider single-trial variance, or by using statistical methods that jointly perform dimensionality reduction and classification.

The PCA input matrix was N × K (Fig. 2(a)), where N is the number of recording channels (features), and K is the number of time bins (samples). The resulting projection matrix was N × L, where L is the number of PCs.

F. Error Detection

To predict a trial's outcome, which is a binary classification as to whether the trial succeeded or failed, we used linear discriminant analysis (LDA). The data were composed of labeled trials (successful or failed), each with an associated vector with a length L × K, generated by concatenating the L PCA-derived features at each of the K time bins. For a hybrid classifier, which utilize both spike and LMP activity, the dimensionality reduction was done on each signal separately, and the features were combined (i.e., $\mathrm{L}_{\mathrm{hybrid}} = \mathrm{L}_{\mathrm{spikes}} + \mathrm{L}_{\mathrm{LMP}}$). In our previous non-human primate work [32] we used a support vector machine (SVM) classifier, but in preliminary analyses of these human data, we found that a simpler LDA method had comparable performance. Since LDA is faster to compute and simpler to interpret than SVM, we decided to use LDA in the present work.

Unless otherwise mentioned, classification accuracy is computed using a time window from 300 ms before until 500 ms after target selection time ($\mathrm{K} = 40\,\mathrm{bins}$). The number of PCs (L) used in the classifiers was optimized for each signal type (spikes or LMP) and each participant (see Sup. Fig. 2). We used 3 and 4 (T5), and 1 and 2 (T6) PCs for spikes and LMP, respectively.

In this work, we are reporting the performance of a classifier that seeks to detect errors in a task when the underlying success rate on that task is not known to the error classifier a priori. Thus, we present results in terms of separate classification accuracies of successful (true positive, 'TP') and failed (true negative, 'TN' rates) trials, rather than the combined success and fail trial outcome classification accuracy, which is more dependent on the underlying specific task performance and obscures potential differences in the error classifier's false positive and false negative rates. We used 100 random repetitions of Monte Carlo cross-validation (randomly splits the data set into training and testing data) to estimate trial-outcome classification accuracy. For each cross-validation split, we withheld a random 10% of the failed trials and the same number of random successful trials for validation. We used the rest of the trials for training the decoder. When comparing classification accuracy to chance level (e.g., Fig. 3(b)) we conducted a shuffle permutation test, in which we shuffled 1000 times the labels of the test set for each of the 100 repetitions. Then, we tested significance of the classification accuracy of the data compared to the distribution of the classification accuracy of the shuffle distribution.

When classifying data that had its dimensionality reduced via PCA, we first conducted PCA on only the training set to find the projection matrix and then used these PCA coefficients to subsequently reduce the dimensionality of the test data prior to classification. The 100 repetitions were used to compute the classification accuracy mean and standard deviation.

G. Statistical Testing

When comparing two different distributions, we used two-sided Wilcoxon rank-sum test with a confidence level of $\mathrm{p} = 0.05$ with Bonferroni correction (to account for the family-wise error rate) unless stated otherwise.

III. Results

Here we report the offline classification accuracy of task outcome using neural activity from motor cortex of two people controlling an iBCI. First, we investigated if the task outcome (successful or failed) of a BCI target selection task modulates spiking and local motor potential (LMP) neural activity. Second, to evaluate the potential for an online error detect-and-undo BCI capability, we report a post-hoc task outcome classification accuracy. We also tested if the error detector can be generalized from one task to another, which is an expected property of a task-independent signal such as an outcome error signal and would be useful for clinical BCI applications. Last, we propose a design for an online error detect-and-undo BCI and estimate the online performance improvement that can be expected for such a system.

A. Task Outcome Related Neural Modulation

To investigate whether task outcome is reflected in motor cortex neural activity, we first examined the trial-averaged neural activity of successful and failed trials. Fig. 1 shows the population peristimulus time histogram (PSTH) difference between successful and failed trials of spikes and the LMP signals during a Grid Task. To perform a neural population-level analysis and to better isolate the signals that correlate with task outcome, we reduced the dimensionality of the multielectrode activity using principal component analysis (PCA, Methods, Fig. 2(c)). From the PSTHs and the first PC projection, we can see that both spiking and LMP activity differs between successful and failed trials shortly after target selection. This analysis also reveals that the neural signal that correlates with task outcome is low-dimensional: most of the task outcome difference activity is captured by one to three PCs, depending on the participants and the signal type. In other words, the time-varying differences between successful and failed trials are well-described by changes of just a few patterns of activity across the neural ensemble. We will call the signal that is captured in this low-dimensional space the 'putative task outcome error signal.' The presence of such a signal in an area widely used for decoding movement intentions via a BCI encouraged us to test its classification accuracy and its potential use for an additional error detect-and-undo BCI capability.

B. Single-Trial Outcome Decoding

For this putative neural task outcome error signal to be useful for use online, a BCI would have to be able to decode it with high accuracy on single trials. To evaluate whether this was true, we built a single-trial linear discriminant analysis (LDA) classifier. To reduce the number of classifier parameters and prevent overfitting, we first projected the data to a lower dimension space using PCA, similarly to the dimensionality reduction used to visualize the data in Fig. 2 (see Methods). In designing the classifier, we had to choose which neural activity features to decode. As described in Methods, we used LMP and spikes signals, which are the low frequency and the high frequency signals of the neural activity, respectively.

The number of PCs used in the classifier was optimized for each signal type (spikes or LMP) and each participant (see Sup. Fig. 2). We found that the LDA decoded task outcome with high accuracy from spikes alone, LMP signals alone and from a 'hybrid' concatenation of spikes and LMP features (Fig. 3(a)).

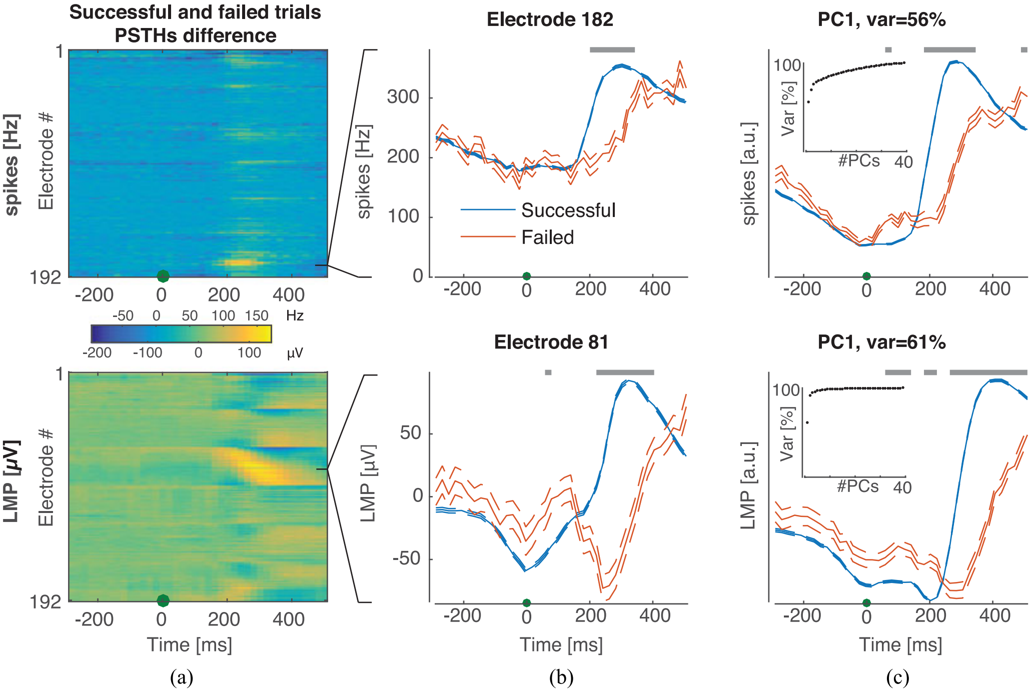

Fig. 2. Task outcome modulates neural activity in the motor cortex. Spikes and LMP of participant T5 during a grid task as a function of time (for T6 see Sup. Fig. 1), aligned to target selection time (green dot, t = 0). (a) The population peristimulus time histogram (PSTH) difference between successful and failed trials of all electrodes. (b) Selected electrode's PSTH (mean±s.e. of the firing rate) during failed (red) and successful (blue) trials. Gray bars indicate times with significant differences (two-sided Wilcoxon signed-rank test with Bonferroni correction, $p < 0.05$, Methods). (c) Trial-averaged projected neural activity into the first principal component (PC, mean±s.e.). (c-inset) Accumulated variance of the outcome difference neural activity as a function of the number of PCs. Most of the trial outcome-related variance is captured in just a few PCs.

The maximum classification accuracies (Wilcoxon signed-rank test with Bonferroni correction, $p < 0.05$) for successful trials (true positive) are 99.7 ± 0.2% (T5) and 97.0 ± 0.6% (T6), and for failures (true negative) are 85 ± 1.4% (T5) 69.9 ± 1.9% (T6). The hybrid classifier achieved the maximum performance with T5 data but not with T6 data. We primarily attribute the difference between error detection performance to differences in the neural signals quality. T5 had two arrays which recorded spikes on many electrodes, whereas T6 had a single array with fewer channels recording spikes (details of T5's and T6's signal quality are available in [28], see Fig. 5 in particular). Consistent with this, T5 had higher overall iBCI communication rates (i.e., better velocity and click decoding performance), and his velocity and click decoders were driven solely by spikes, in contrast to T5's, which were driven mostly by high frequency LFP power in addition to a smaller contribution by spikes for velocity decoding [28]. The fact that the hybrid classifier did not improve performance for T6 likely reflects the poor quality of spiking signals on T6's single multielectrode array. That said, there could be other potential differences between participants (e.g., cognitive strategy, sensorimotor system reaction time) that contributed to our observed error detection performance differences, which are outside the scope of the present study.

The high true negative rate, combined with close to 100% true positive rate, suggests that an error detection system could detect and undo errors (for example, by deleting an erroneously selected character) with high accuracy while almost never misclassifying a successful selection as erroneous (for example, the system would rarely mistakenly delete a character that the BCI user did in fact mean to select).

Local field potentials are known to also contain information (e.g., kinematics) in other frequency bands [41]–[47]. To investigate if these frequency bands were modulated by task outcome, we compared the task outcome classification accuracy using LMP and different frequency bands' power (Methods, Sup. Fig. 4). For all LFP bands compared to LMP alone, the true positive rates were the same for T5 and higher by up to 3% for T6 (two-sided Wilcoxon signed-rank test), and the true negative rates were lower by at least 29% (T5) and 43% (T6). We can therefore infer that the power of those frequency bands contains some information about the task outcome but not as much as the LMP (which also contains the phase information), and thus we chose to use LMP as our LFP feature.

To better understand when the putative task outcome signal becomes evident in motor cortical activity, we decoded the task outcome using growing time windows. These windows all started 300 ms before selection and ended between 200 ms before until 500 ms after selection (Fig. 3(b)). The cross-validated combined classification accuracy increased above chance (50%, see Methods) 80 ms (T5) and 100 ms (T6) after target selection time (two-sided Wilcoxon signed-rank test with Bonferroni correction, $p < 0.05$) and saturated around 260 ms (T5) and 340 ms (T6) later. Thus, we can infer that error-related motor cortex activity increases after target selection in a span of few hundred milliseconds. This timing (coming >80 ms after selection) argues against the error detector working primarily

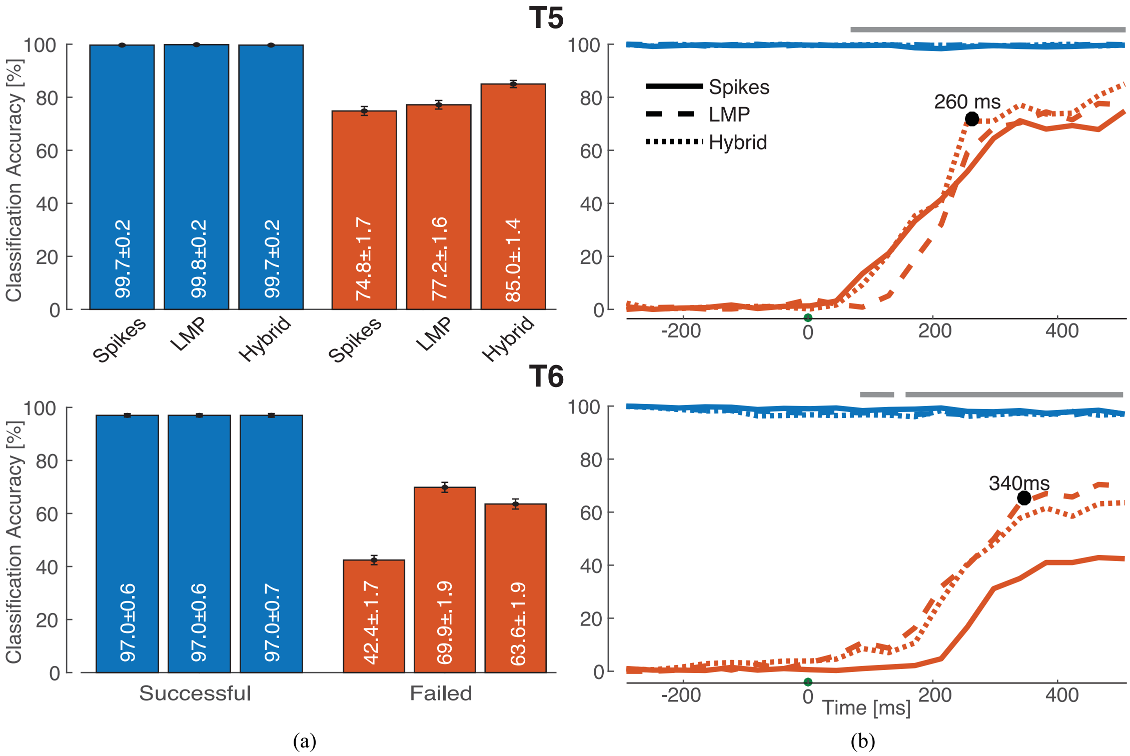

Fig. 3. Single-trial task outcome decoding using motor cortical neural activity. Classification accuracy of successful (true positive, blue) and failures (true negative, red) trials of the two participants (T5 and T6). (a) Classification accuracy is computed using a time window from 300 ms before until 500 ms after target selection time. Three different classifiers using spike signals, LMP signals, and both (Hybrid) were compared. (b) Classification accuracy as a function of the end of a growing decoded time window, which starts at 300 ms before selection and ended between 200 ms before until 500 ms after target selection. Green dot corresponds to t = 0, target selection time. Grey horizontal bars indicate when classification accuracy of the best method (hybrid for T5 and LFP for T6) is better than chance (50%, see Methods).

by detecting mis-classification by the HMM click decoder. Instead, this activity might be composed of several processes, such as error detection, correction planning, and corrective movement execution. We discuss interpretations in more detail in the Discussion.

We had examined outcome classification from a start time prior to selection in case participants had some inkling that the upcoming selection would be a failure (for example, if they sensed that their performance was poor on this trial or that they saw the cursor was about to drift off the target). In our data, however, outcome classification was only significantly above chance after target selection. We therefore also assessed whether starting the classified time window at 80 ms (T5) or 100 ms (T6) after target selection instead of 300 ms before selection would decrease overfitting and improve performance (by reducing the number of non-relevant features), but found that this did not affect classification accuracy ($p = 0.84$ for T5 and $p = 0.12$ for T6). For simplicity, we therefore started the time window at 80 ms for subsequent analyses.

For the iBCI auto-deletion application, we are interested in the practical utility of the neural activity to detect errors. As such, we are agnostic to whether the neural activity reflects the iBCI user's perception of errors versus his intention to correct. In a Grid task targets are randomly cued both after successful and failed trials. Thus, a stereotyped movement such as reaching to a backspace key is not intrinsically part of the task. However, other kinematic differences between successful and failed might exist. To verify that the decoding is not attributed to kinematic differences (e.g., stereotype movement), which might be expected from motor cortex activity, we have investigated the ability to predict errors through kinematics. We checked whether there were kinematic differences between successful and failed trials, and, if so, whether the neural activity being used by the task outcome decoder was more informative than just decoding these kinematic differences. We note that since this was a BCI task, the cursor's kinematics themselves reflect a specific projection of the recorded neural activity; thus, in fact, we are asking whether the putative trial-outcome error signal is distinct from the neural signals related to movement execution activity. Sup. Fig. 3 shows that there were indeed small kinematic differences between successful and failed trials. However, classification accuracy was lower when decoding trial outcome using cursor speed, direction, and X and Y velocity than when decoding the putative trial-outcome error signal (two-sided Wilcoxon signed-rank test, $p < 0.001$). This decoding performance difference shows that motor cortex neural activity contains information about the outcome that is not directly related to the BCI's kinematics and suggests that the trial-outcome error signal is distinct from movement intention. This distinction is arguably not critical from a neural engineering perspective, in that if subtle kinematic differences could be used to predict success versus failure, it would still be useful to exploit this. Nonetheless, our

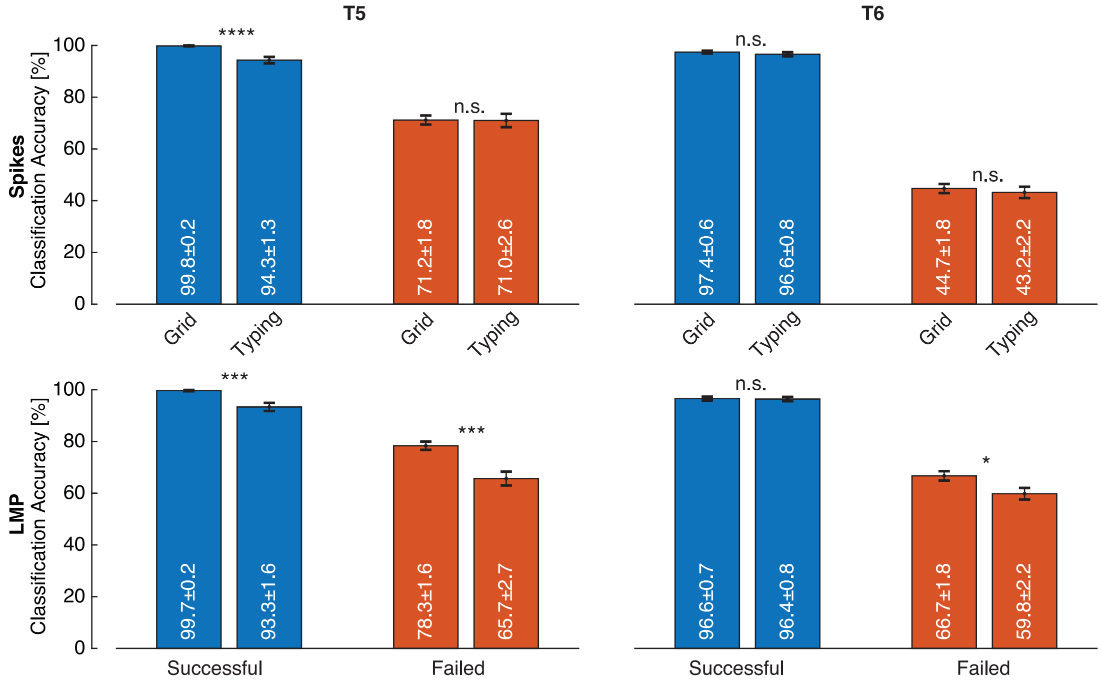

Fig. 4. Task outcome error signals generalize from the Grid Task to the Typing Task. Comparison of classification accuracy of Grid Task trial vs. Typing Task trials, using a classifier trained on Grid Task trials only. Classifiers were trained on time window from 80 ms until 500 ms after target selection time for spikes (top row) and LMP (bottom row).

results are encouraging because they show that the proposed error detect-and-undo strategy would utilize additional dimensions of the neural data in a way that kinematics-based heuristics (for example, requiring low speed to allow target selection) could not.

C. Generalization Between Tasks

Although it would be plausible for the task outcome neural correlates to be task dependent (i.e., different tasks' outcomes are represented differently in motor cortical activity), we hoped that those correlates would be task independent (i.e., similar regardless of the specific task being used, such as our Grid and Typing tasks). To evaluate how well the success vs. failure signals that we recorded generalized between these tasks, we trained a classifier using Grid Task trials and compared how well it could classify held-out Grid Task trials versus Typing Task trials. In both participants, we could classify Typing Task trials with high accuracy using both spikes and LMP decoders trained using Grid Task data (Fig. 4). Although there was some decrease in performance, the decoder could still generalize well from one task to another. Although this property, which is expected from a task outcome error signal, is not requisite for developing a BCI error detect-and-undo system (if necessary, one could train separate decoders for different tasks), it is fortuitous and simplifying. Generalizability is likely to aid in the practical implementation of such a system by reducing training data requirements and improving robustness across different situations in which the BCI is used.

We did not test the reverse cross-task generalization (train-on-typing task, test-on-grid task) or a within-task (train-on-typing, test-on-typing), in the present study because of the small number of typing task trials available. A preliminary investigation found that even within-task typing classification was worse than train-on-grid, test-on-typing performance, consistent with there being insufficient data.

D. Closed-Loop Detect-and-Undo Design and Simulations

In this section, we put together what we have learned to propose how these neural correlates of task outcome error can be utilized to improve BCI performance. There are two main design choices that characterize a real-time error detector: 1) what kind of corrective action should it perform (e.g., prevent an action before it occurs or undo it afterward?) and 2) when should it perform this action (e.g., at the time of action selection, 100 ms later, or a second later?). In this work, we analyzed trials in which the selection was done using a click signal, which is the current state-of-the-art for motor BCI target selection [20], [28]. This contrasts with our previous study [32] in which monkeys dwelled over a target to select it. In the present human study, we were able to decode that a selection was erroneous only after target selection, which is consistent with the short latency of 30 ms of the 'clicking signal,' compared to a 500 ms dwelling time in monkeys. Given this latency, we therefore propose that when the neural correlates of the user perceiving an error are detected by the BCI, its corrective action should be undoing the system's last action. In a typing application, this would mean auto-deletion of the last typed character.

In a real-world application (e.g., typing), classifying whether a selection was erroneous or not after the target selection might delay the next movement. In the worst-case scenario, the BCI user will wait until the classification is done before continuing

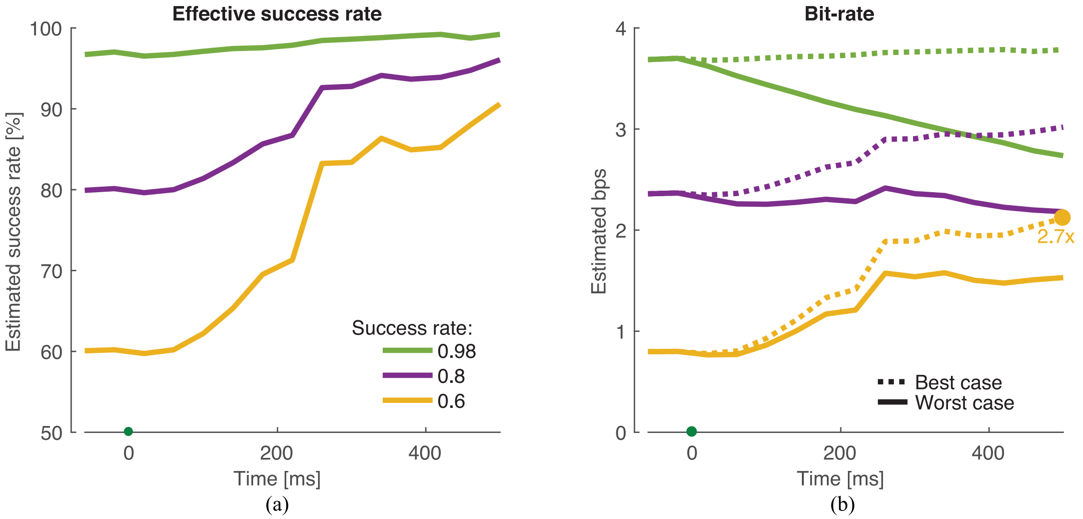

Fig. 5. Estimated success rate and bitrate as a function of added delay and task difficulty. The plots show the predicted success rate and bitrate if an error detect-and-undo system were used during online BCI typing. The three colors correspond to different base success rates (i.e., without parallel error decoding). When success rates are lower, there is more room for improvement using error detect-and-undo. The key parameter of task outcome classification accuracy was based on the empirically observed offline classification accuracy using participant T5's Grid Task data (which is shown in Fig. 3). (a) Effective Success rate (Eq. 1). (b) Estimated bitrate (Eq. 3) - solid and dashed lines are worst and best (dt is set to 0) case scenarios, respectively, based on whether the next reach is delayed or not as a result of the classifier latency.

to his next target (after all, why move to the backspace key or the next letter if the system might undo the previous selection). The best-case scenario is that error detection delay will not affect the user's cadence at all, because it is less than the user's preferred (or cognitive load-imposed) natural delay between movements (see more details in Discussion). To better understand the potential effect of detecting errors in real-time, we estimated the effective success rate and the estimated bitrate under different assumptions of underlying success rate, and delays between target selection and classification latency.

A real-time error detect-and-undo capability will potentially increase the 'effective success rate,' i.e., the success rate after error undoing. If 's' and 'f' are the numbers of success and failed trials, the success rate is defined as: s/(s+f). When incorporating an error detector on top of the standard BCI, some erroneous selections will be detected and automatically deleted, and the user would be able to continue to the correct target. Thus, those erroneous trials will not count as errors when computing the performance, since the user does not need to correct them manually. On the other hand, some correct selections could be misclassified as being erroneous, and result in an additional reach. The resulting success and failed trial numbers will be:

$$s' = s \cdot TP \; ; \; f' = f\,(1 - TN) \tag{1}$$

where TP and TN are the true positive and true negative rates. Thus, the effective success rate will be: s'/(s+f'). We estimated the effective success rate as a function of classification latency and the initial success rate (Fig. 5(a)) based on the empirically observed accuracy of our offline error detection decoder using T5 data (Fig. 3(b)). The effective success rate increased when the decoder was given more time after target selection, and this increase was greater for lower success rates. Thus, when using a BCI with an error detect-and-undo capability, hard tasks will become easier in the sense that the effective success rate will become higher.

However, waiting longer to make the determination of whether a selection was erroneous imposes a cost in terms of how many characters can be selected per minute, especially if this adds a latency between target selection and the next movement. While detect-and-act can be implemented in many application, we chose to simulate the its effect on typing. To measure information transfer rate during typing while accounting for both what fraction of trials are correct, and how long these trials take, recent BCI studies use the bitrate metric [19], [20], [28], [48], [49]. Bitrate is defined as the rate of corrected keys (weighted by how many bits of information they signal), while assuming conservatively that an incorrect selection had to be compensated by a corrected selection (similar to typing, when the user needs to delete his mistakes using a backspace key):

$$\text{bps} = \log_2 (N - 1) \frac{s - f}{T} \tag{2}$$

where T is the total length of the trials and N is the number of potential targets. When an error detect-and-undo mechanism is added to a BCI, its bitrate will depend on the effective success and failed trial count as well as and the added delay as a result of classification latency (dt):

$$\text{bps} = \log_2 (N - 1) \frac{s' - f'}{T + dt \cdot (s + f)}. \tag{3}$$

We estimated the bitrate improvement that could be expected using error detection under three different task difficulty scenarios (i.e., target selection success rates before error detect-and-undo): 60%, 75%, and 98%, where 98% was the empirical success rate in the offline T5 data analyzed for this feasibility study (Table I). For the estimation we used Eq. 3, the error detection accuracy of T5 (TP(dt) and TN(dt), Fig. 3(b)), and the average trial length and selection information content we observed when T5 performed the Grid Task (T = 1.3 sec, N = 36). The resulting system's performance should be somewhere between the solid (worst-case scenario) and the dashed (best-case scenario, dt is set to 0) lines that are shown in Fig. 5, depending on the initial success rate (task difficulty) and the effect of the classification latency on the user's latency to start the next movement. The performance change due to adding error detect-and-undo to the iBCI can range from improving communication rates more than two-fold in the optimistic scenario (2.7 times) under low success rate conditions (e.g., 60%), to decreasing performance when rates are already high in the worst case (e.g., 98%). This makes sense intuitively: automatically undoing most errors at a slight cost of time on every trial will be worth it if errors are frequent but less so (or not at all) if errors are rare. When implementing such a system, the optimal added delay can and should be estimated from real-time error detection experiments and be adapted online based on the prevalent error rates given the task difficulty and BCI neural control quality.

IV. Discussion

In this study, we translated our previous preclinical (monkey) research on augmenting intracortical movement BMIs with a parallel task outcome decoder [32] to a human clinical trial. Though these pre-clinical tests were encouraging, three critical questions remained: are task outcome neural correlates present in the hand area of human motor cortex, how well can task outcome can be decoded?, and what should be the design of an error detector.

A. Putative Outcome Error Signal in Human Motor Cortex

An important result of this study was finding that task outcome error signals were present in neural activity at the hand area of human motor cortical, which is the brain area that movement iBCIs have achieved the highest performance to date in preclinical [20] and clinical [28] studies. Previous studies showed that task outcome correlates exist in other brain areas such as anterior cingulate cortex (ACC), basal ganglia and supplementary motor area [50]–[53]. Encouragingly, few studies postulated the existence of different types of error signals in motor cortex [39], [54]–[56]. However, clear evidence for task outcome error signals in human motor cortex, and more specifically in the precentral gyrus, with intracortical recordings has not been presented. The existence of task outcome error signals in that brain area suggests that iBCI error detect-and-undo capability can be implemented "for free" in terms of not requiring additional sensors. While we were optimistic that we would find this signal given our previous monkey results [32] and previous human ECoG study [57], this was not guaranteed. In addition to the standard caveat that the homology between monkeys and humans is imperfect, there were important differences in the behavioral tasks performed by our monkey and human BCI users. Several outcome correlates present in monkey experiments – such as expecting a liquid reward and planning to lick for it – are potential confounds because they are distinct from the outcome-error itself. These do not exist in human clinical research sessions. Also, whereas the monkeys were typically directly rewarded for successfully selecting a target with a drop of liquid, the "reward" for the humans was an internal desire to communicate a certain character (in the Typing Task) or succeed in the Grid Task game. Additionally, although the Grid Tasks were otherwise similar between the two species, the additional cognitive burden of the human-only Typing Task was quite different from what can be tested in monkeys. It is therefore highly encouraging that human motor cortex also shows correlates of successful versus erroneous target selection in this more real-world task. Reproducing the previous monkey results in humans also strengthens our prior claim that a putative outcome error signals exist in motor cortex.

An interesting question for future study is whether neural correlates differ between different types of selection errors (for example, unintentionally clicking while traveling towards the target, versus intentionally clicking when just slightly off-target). This additional information could be used to further improve performance, for example by updating the click detection threshold if many false-clicks happen or by decoding what the intended target would have been in a near-target erroneous selection. Unfortunately, the limited numbers of failure trials in the present study and their proximity to the cued target preclude a thorough examination of fine-grained error type differences. In the future, cursor movement perturbations/errors [58], false positive clicks, or false negative clicks could be intentionally introduced during iBCI use to systematically examine neural responses across different types of velocity-and-selection control errors.

In this study, we have taken a largely practical view of the putative outcome error signal we observed, asking whether, regardless of the subtleties of what this signal means or does, we can use it for iBCI detect-and-undo. A deeper understanding of the putative outcome error signal can provide both scientific insight about the role of outcome feedback in motor cortex, as well as help us anticipate whether the iBCI utility of decoding this signal is likely to generalize to different tasks. The putative outcome error signal could be elicited by internal processes (e.g., the user understands he made a mistake based on self-monitoring his performance leading up to and at the time of selection) or extrinsic feedback (e.g., it reflects neural correlates of the user perceiving auditory or visual feedback indicating that he made a mistake). The differences between the Grid task and Typing tasks analyzed here provide some degree of insight in to this question. Whereas the Grid task provides explicit feedback in the form of different auditory tones immediately after correct and incorrect selections, in the Typing task there was no explicit outcome feedback except for the selected letter appearing in the typing bar (as shown in Fig. 1(a)). This visual feedback did not directly communicate "correct" vs "incorrect", but rather the participant had to compare it to the cued letter. Nonetheless, the error signal we observed was present in both tasks, suggesting that it is related to the user's internal feedback (or at least can be determined without a simple explicit feedback cue). On the other hand, the observation that we were unable to decode task outcome error until shortly after the selection suggests that, at least in these tasks, forward modeling [59] of task performance did not appear to generate a task outcome prediction in motor cortex. This result contrasts with our previous monkey task outcome decoding study [32], in which neural correlates did anticipate the upcoming dwell selection outcome. This difference could be due to task differences (in particular, dwell versus click selection), the degree of the iBCI user's familiarity with the task (the monkeys were over-trained and perhaps were better at anticipating outcomes), or human versus monkey differences. Future experiments can more comprehensively investigate these question by, for example, presenting false explicit feedback in order to assess whether the error-related neural correlates reflect self-monitoring outcome estimates, explicit feedback, or both. The fact that we only saw the error signal after target selection, and that in the Grid task making an error did not subsequently require any correction (such as the delete key), argues against this signal reflecting the user planning or attempting to correct the action that led to the incorrect target selection.

Together, the results imply that task outcome is made available to human motor cortex. While our study cannot speak to why this outcome error is in motor cortex, or what other brain regions it arrives from, one possibility is that it is a teaching signal for reinforcement learning [60]. In terms of future iBCI error detection applications, the invariance of the signal to simple vs indirect feedback increases our optimism that this approach will work in more complex tasks. Nonetheless, it remains to be seen whether this is the case, especially in open-ended tasks such as free-typing or drawing where there is no value feedback at all provided after each action, meaning that outcome error can only be provided by the user's internal self-assessment.

B. Error Detection: Detect-and-Undo

Our finding that errors (task outcome) can be detected with high accuracy from human motor cortex neural activity (Fig. 3) is consistent with the high accuracy of decoding monkey motor cortical task outcome error signals [32]. This bodes well for the future implementation of an online error detect-and-undo system. Whereas these previously reported monkey task outcome error decoders operated on threshold crossing spiking activity, here we also found that high decoder performance was achieved using the LMP feature of the local field potentials. It has been suggested that LFP may be recordable using chronic multielectrode arrays for longer than spikes [47], [61], which would be of high value for extending the useful lifespan of a clinical BCI system. Additionally, the ability to train the decoder based on different days (Table I) suggests that the signal is stable across days; however, future work is needed to characterize it better. The signal stability and task generalization (Fig. 4) are encouraging properties since it means that an error classifier can be trained once with pre-collected data and be used without additional training in a real-time BCI. This can save training time and recalibration of the error-decoder, which is often needed for the kinematic decoder in a human BCI use [21], [62].

The error detect-and-undo capability provides a novel opportunity and specific direction for improving human BCI systems. Some work in this area has been previously done using EEG [33]–[35], [37], [63] and ECoG [39]. Most recent iBCI research, both in preclinical animal experiments [29]–[31], [49] and human clinical trials [21], [23], [24], [27], [28], [64]–[66] has focused on designing better kinematic or kinetic decoders to more accurately infer the user's movement intention. Error detection with corrective action will improve iBCI performance via a very different approach: a parallel, independent decoder that extracts more information from the recorded neural activity [32], [33], [39] and reduces the cost of errors when they occur. Specifically, it provides BCI systems with the capability to automatically undo the last action taken when the system detects that the user perceives this action as erroneous, sparing them from having to undo it manually. Such a detect-and-undo system could be used for various BCI applications. During typing [14], [21], [27], [28], [67], this could be used for immediate character auto-deletion or for error tracking to improve upon word prediction algorithms (by assigning a probability to the "correctness" of each typed letter and thereby focusing the options for word completion/correction). Detect-and-undo systems can also be utilize for returning to the previous menu during tablet use, and returning to a previous position when using a robotic arm [68], [69]. Though encouraging, whether or not similar task outcome error signals exist in more complex tasks such as prosthetic limb control remains an open question for future research.

Error detect-and-undo is more impactful in a low success rate task. Thus, it can be most effective for rescuing performance when the BCI performance degrades (for example, as a result of neural signal degradation), or increasing performance in more difficult tasks. Chronic intracortical electrode signals degrade with time, which reduces BCI performance and success rates [18], [70]–[72]. Recent years have seen considerable work towards rescuing BCI performance by designing new kinematic decoders [41], [46], [47], [73]–[77]. Here, we suggest an alternative approach: use error detection to increase effective success rates and the user experience, thus rescuing BCI performance. Even in the absence of performance degradation, error detect-and-undo can help improve performance by increasing the tolerable task difficulty. For example, increasing the number of available keys on a keyboard will increase the transmitted information rate from each selection; however, it will decrease the target size and as a result the success rate. By using the error detector, the system could increase the number of keys while still keeping the success rate high. To find the optimal keyboard density, a mapping of the (error-corrected) success rate as a function of layout needs to be found [19].

C. Online Detect-and-Undo Design

The design of an online error-decoder hinges on two parameters: the classification latency and the type of corrective action (e.g., auto-deletion). The resulting performance improvement also depends on a number of factors: the system's classification accuracy (Fig. 3(b)); the user's baseline success rate (which depends on task difficulty); average length of a discrete action (such as a key selection movement), and the cost of making errors (e.g., whether the error requires a delete key selection). As discussed in the Results section, there is a tradeoff between longer classification latency and accuracy, which have opposite effects on overall system information transfer. Importantly, the effect of classification latency on the trial length is yet unknown and needs to be investigated in online experiments. In the worst-case scenario, the BCI user might develop a strategy of waiting until the error detect-and-undo classification is made before initiating the next action; this will delay his next action by a maximum of the classification latency after every selection. On the other hand, the person could adopt a strategy of initiating the next action immediately and adjusting it later only if necessary (i.e., if the selection was erroneous and the error detect-and-undo does not work). Such a strategy will reduce the effect of the classification latency and would be the optimal action after successful trials and after erroneous selections when detect-and-undo works (which our results suggest would be most of the time given the high true positive and low false positive rates measured offline). The average cost of such strategy is low and is incurred when errors are not detected or when a false positive selection undo happens. When that happens, the participant will need to update his/her movement after the decoder latency (dt), which will prolong the trial length by approximately the same latency. However, since selection errors are rare to begin with (e.g., 20%) and the detect-and-undo FN is low (15% (T5)), only about 3% of the trials (20% × 15%) will become longer by the decoder latency (dt) as a result of FN. Similarly, detect-and-undo FPs are very rare (<1% (T5)) and presumably each cost the user approximately dt of time moving towards what now becomes the wrong key. Thus, the effect of both FP and FN on trial length using this suggested strategy will on average be approximately dt × 0.04. In our data, the trial length average was $\mathrm{T} = 1.3$ sec, but the delay required for high error detection accuracy will be only about $\mathrm{dt} = 300$ ms, which is less than T/4. Thus, the total effect is less than 1% $(\mathrm{T} + \mathrm{dt} \times 0.04 = \mathrm{T} \times 1.01)$. Thus, the added trial time using this suggested strategy is very low and in practice close to the best-case limit we present in Fig. 5.

In addition, in typing and other forms of real-world BCI use, the user must decide on a next action after completing the previous action (in contrast to the Grid Task data here, where the experimental system automatically cued the next target). In this case, thinking about the next action (e.g., identifying the next letter to type) might impose a natural delay before starting the action such that the added delay due to the task outcome classification will be reduced. A task with such a cognitive load or longer trial lengths (meaning that the relative cost becomes smaller) will further reduce the effect of the delay.

Here, we presented the upper and the lower limits of the error decoder performance improvement for a particular set of parameters (Fig. 5). While zero delay (best case) is optimistic, we believe that in a task with high cognitive load and with the right user strategy, detect-and-undo performance improvement can approach close to this limit. When calibrating online systems, a similar analysis should be done for every task and participant, and these parameters should potentially be adapted online based on recent success rates, action completion times, and inferred false positive / false negative rates.

Most iBCI studies are to some degree application-specific and try to maximize overall system performance, meaning that not only the iBCI decoder but also its interface (e.g., key size) are optimized to minimize costly errors (e.g., [19]). The data we analyzed here had been collected as part of a study striving for high communication rates (without error detect-and-undo), and consequently the success rate in these experiment sessions was very high (98% (T5) and 93% (T6), see Table I). Our simulation predicts that very high success rate scenarios such as these will not benefit from a detect-and-act system (Fig. 5(b)). However, to facilitate iBCI users' independence across a wider range of activities, we envision them using standard (non-dedicated) interfaces, e.g., a cursor to control an off-the-shelf tablet computer [78]. Because optimal target sizes and layouts will not always be present in real-world use, errors are likely to occur more often than when working with a lab system optimized for that specific user's information throughput. Furthermore, having an iBCI with error detect-and-undo capability itself changes the calculus of what task success rates will result in high performance. Thus, we simulated a range of task difficulties to show the potential of detect-and-act across a spectrum of target selection error rates.

Our classifier was biased towards high accuracy when identifying success trials (true positives) as a result of the empirical distribution bias (i.e., high success rate, Table I). A high true positive rate is a desired property of an error detector since misclassifying success trials will frustrate the user. However, to get an optimal decoder that will balance the two, the weight of each outcome can be modeled and used to modify the classifier. For instance, in a typing application, classifying a successful trial as a failure will result in a penalty of re-selecting the key, but since the cursor is in the vicinity of the key this would require just a short movement (or even just a click). However, classifying a failure as a success (i.e., not detecting an error) will result a higher penalty, since the user would need to manually delete the last key and select the original key (two movements). Such cost differences can be modeled from online experiments and used to re-balance the classifier to achieve even higher bitrates.

During the present research sessions, the participants were not instructed to do anything special after successful or failed trials except to carry on with the task. However, in online use of a BCI with error detect-and-undo, the user can be informed about the additional decoder and could potentially be trained to emphasize the task outcome error signal. This would improve error classification accuracy and improve the bitrate.

V. Conclusion

In our previous work [32] we demonstrated the potential for incorporating error detection into a motor cortically-driven BCI using a pre-clinical monkey animal model. Here, we continued this line of work as part of a pilot clinical trial with two participants by demonstrating that a putative task outcome signal also exists in hand area of human motor cortex, and that this signal can be decoded with high accuracy. This work suggests a new neural signal present in human motor cortex activity and sets the stage for future work incorporating error detector in an online iBCI to increase its performance.

Acknowledgment

The authors would like to thank participants T5, T6, and their families and caregivers; B. Davis, J. Ventura, P. Rezaii, B. Pedrick, E. Casteneda, M. Coburn, S. Patnaik, B. Travers, and D. Rosler for administrative support; J. Menon; S.I. Ryu for surgical assistance.

Conflict of Interest: L. R. Hochberg has a financial interest in Synchron Med, Inc., a company developing a minimally invasive implantable brain device that could help paralyzed patients achieve direct brain control of assistive technologies. L. R. Hochberg's interests were reviewed and are managed by the Massachusetts General Hospital, Partners HealthCare, and Brown University in accordance with their conflict of interest policies.

Author Contributions: N. Even-Chen and S. D. Stavisky designed the error detect-and-undo algorithm, analyzed the data, and prepared the manuscript. C. Pandarinath, P. Nuyujukian, and C. H. Blabe were responsible for the original online BCI data collection and helped prepare the manuscript. L. R. Hochberg is the sponsor-investigator of the multi-site pilot clinical trial. J. M. Henderson surgically implanted the microelectrode recording arrays in both study participants. K. V. Shenoy and J. M. Henderson were involved in all aspects of the study. All authors reviewed and edited the manuscript.

Disclaimer: The content 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.

Caution: Investigational Device. Limited by Federal Law to Investigational Use.

REFERENCES

[1] J. R. Wolpaw *et al.*, "EEG-based communication: Improved accuracy by response verification," *IEEE Trans. Rehabil. Eng.*, vol. 6, no. 3, pp. 326–333, Sep. 1998.

[2] H.-J. Hwang *et al.*, "Development of an SSVEP-based BCI spelling system adopting a QWERTY-style LED keyboard," *J. Neurosci. Methods*, vol. 208, no. 1, pp. 59–65, Jun. 2012.

[3] M. Spüler *et al.*, "Online adaptation of a c-VEP Brain-computer Interface(BCI) based on error-related potentials and unsupervised learning," *PLoS One*, vol. 7, no. 12, Dec. 2012, Art. no. e51077.

[4] G. Bin *et al.*, "An online multi-channel SSVEP-based brain-computer interface using a canonical correlation analysis method," *J. Neural Eng.*, vol. 6, no. 4, Aug. 2009, Art. no. 046002.

[5] E. C. Leuthardt *et al.*, "A brain-computer interface using electrocorticographic signals in humans," *J. Neural Eng.*, vol. 1, no. 2, pp. 63–71, Jun. 2004.

[6] G. Schalk *et al.*, "Two-dimensional movement control using electrocorticographic signals in humans," *J. Neural Eng.*, vol. 5, no. 1, pp. 75–84, Mar. 2008.

[7] D. Moran, "Evolution of brain-computer interface: action potentials, local field potentials and electrocorticograms," *Current Opinion Neurobiol.*, vol. 20, no. 6, pp. 741–745, Dec. 2010.

[8] P. Brunner *et al.*, "Rapid communication with a "p300" matrix speller using electrocorticographic signals (ECoG)," *Front. Neurosci.*, vol. 5, p. 5, Feb. 2011.

[9] W. Wang *et al.*, "An electrocorticographic brain interface in an individual with tetraplegia," *PLoS One*, vol. 8, no. 2, Feb. 2013, Art. no. e55344.

[10] M. D. Serruya *et al.*, "Brain-machine interface: Instant neural control of a movement signal," *Nature*, vol. 416, no. 6877, pp. 141–142, Mar. 2002.

[11] D. M. Taylor *et al.*, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, Jun. 2002.

[12] J. M. Carmena *et al.*, "Learning to control a brain–machine interface for reaching and grasping by primates," *PLoS Biol.*, vol. 1, no. 2, Oct. 2003, Art. no. e42.

[13] G. Santhanam *et al.*, "A high-performance brain-computer interface," *Nature*, vol. 442, no. 7099, pp. 195–198, Jul. 2006.

[14] L. R. Hochberg *et al.*, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, Jul. 2006.

[15] K. Ganguly *et al.*, "Reversible large-scale modification of cortical networks during neuroprosthetic control," *Nature Neurosci.*, vol. 14, no. 5, pp. 662–667, Apr. 2011.

[16] J. E. O'Doherty *et al.*, "Active tactile exploration using a brain-machine-brain interface," *Nature*, vol. 479, no. 7372, pp. 228–231, Oct. 2011.

[17] V. Gilja *et al.*, "A high-performance neural prosthesis enabled by control algorithm design," *Nature Neurosci.*, vol. 15, no. 12, pp. 1752–1757, Dec. 2012.

[18] J. D. Simeral *et al.*, "Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array," *J. Neural Eng.*, vol. 8, no. 2, Apr. 2011, Art. no. 025027.

[19] P. Nuyujukian *et al.*, "A high-performance keyboard neural prosthesis enabled by task optimization," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 1, pp. 21–29, Jan. 2015.

[20] J. C. Kao *et al.*, "A high-performance neural prosthesis incorporating discrete state selection with hidden Markov models," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 4, pp. 935–945, Apr. 2017.

[21] B. Jarosiewicz *et al.*, "Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface," *Sci. Transl. Med.*, vol. 7, no. 313, Nov. 2015, Art. no. 313ra179.

[22] C. E. Bouton *et al.*, "Restoring cortical control of functional movement in a human with quadriplegia," *Nature*, vol. 533, no. 7602, pp. 247–250, May 2016.

[23] B. Wodlinger *et al.*, "Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations," *J. Neural Eng.*, vol. 12, no. 1, Feb. 2015, Art. no. 016011.

[24] D. Bacher *et al.*, "Neural point-and-click communication by a person with incomplete locked-in syndrome," *Neurorehabil. Neural Repair*, vol. 29, no. 5, pp. 462–471, Jun. 2015.

[25] C. H. Blabe *et al.*, "Assessment of brain–machine interfaces from the perspective of people with paralysis," *J. Neural Eng.*, vol. 12, no. 4, 2015, Art. no. 043002.

[26] J. L. Collinger *et al.*, "Functional priorities, assistive technology, and brain-computer interfaces after spinal cord injury," *J. Rehabil. Res. Dev.*, vol. 50, no. 2, pp. 145–160, 2013.

[27] V. Gilja *et al.*, "Clinical translation of a high-performance neural prosthesis," *Nature Med.*, vol. 21, no. 10, pp. 1142–1145, Oct. 2015.

[28] C. Pandarinath *et al.*, "High performance communication by people with paralysis using an intracortical brain-computer interface," *Elife*, vol. 6, Feb. 2017, Art. no. e18554.

[29] J. C. Kao *et al.*, "Information systems opportunities in brain-machine interface decoders," *Proc. IEEE*, vol. 102, no. 5, pp. 666–682, May 2014.

[30] M. M. Shanechi, "Brain-machine interface control algorithms," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1725–1734, Oct. 2017.

[31] J. I. Glaser *et al.*, "Machine learning for neural decoding," arXiv:1708.00909, Aug. 2017.

[32] N. Even-Chen *et al.*, "Augmenting intracortical brain-machine interface with neurally driven error detectors," *J. Neural Eng.*, vol. 14, no. 6, Nov. 2017, Art. no. 066007.

[33] R. Chavarriaga *et al.*, "Errare machinale EST: The use of error-related potentials in brain-machine interfaces," *Front. Neurosci.*, vol. 8, Jul. 2014, Art. no. 208.

[34] M. Spüler *et al.*, "Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a P300-BCI," *Clin. Neurophysiol.*, vol. 123, no. 7, pp. 1328–1337, Jul. 2012.

[35] N. M. Schmidt *et al.*, "Online detection of error-related potentials boosts the performance of mental typewriters," *BMC Neurosci.*, vol. 13, p. 19, 2012.

[36] P. W. Ferrez and J. del R Millan, "Error-related EEG potentials generated during simulated brain-computer interaction," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 3, pp. 923–929, Mar. 2008.

[37] I. Iturrate *et al.*, "Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control," *Sci. Rep.*, vol. 5, Sep. 2015, Art. no. 13893.

[38] R. Chavarriaga *et al.*, "Robust, accurate spelling based on error-related potentials," in *Proc. 6th Int. Brain-Computer Interface Meeting, Asilomar, CA, USA*, 2016, pp. 1328–1337.

[39] T. Milekovic *et al.*, "Detection of error related neuronal responses recorded by electrocorticography in humans during continuous movements," *PLoS One*, vol. 8, no. 2, Feb. 2013, Art. no. e55235.

[40] J. P. Cunningham and B. M. Yu, “Dimensionality reduction for large-scale neural recordings,” *Nature Neurosci.*, vol. 17, no. 11, pp. 1500–1509, Nov. 2014.

[41] S. D. Stavisky *et al.*, “A high performing brain–machine interface driven by low-frequency local field potentials alone and together with spikes,” *J. Neural Eng.*, vol. 12, no. 3, May 2015, Art. no. 036009.

[42] A. K. Bansal *et al.*, “Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: Spikes, multiunit activity, and local field potentials,” *J. Neurophysiol.*, vol. 107, no. 5, pp. 1337–1355, Mar. 2012.

[43] J. G. O’Leary and N. G. Hatsopoulos, “Early visuomotor representations revealed from evoked local field potentials in motor and premotor cortical areas,” *J. Neurophysiol.*, vol. 96, no. 3, pp. 1492–1506, Sep. 2006.

[44] J. Rickert *et al.*, “Encoding of movement direction in different frequency ranges of motor cortical local field potentials,” *J. Neurosci.*, vol. 25, no. 39, pp. 8815–8824, Sep. 2005.

[45] H. Scherberger *et al.*, “Cortical local field potential encodes movement intentions in the posterior parietal cortex,” *Neuron*, vol. 46, no. 2, pp. 347–354, Apr. 2005.

[46] K. So *et al.*, “Subject-specific modulation of local field potential spectral power during brain–machine interface control in primates,” *J. Neural Eng.*, vol. 11, no. 2, 2014, Art. no. 026002.

[47] R. D. Flint *et al.*, “Accurate decoding of reaching movements from field potentials in the absence of spikes,” *J. Neural Eng.*, vol. 9, no. 4, Aug. 2012, Art. no. 046006.

[48] P. Nuyujukian *et al.*, “Performance sustaining intracortical neural prostheses,” *J. Neural Eng.*, vol. 11, no. 6, Dec. 2014, Art. no. 066003.

[49] J. C. Kao *et al.*, “Single-trial dynamics of motor cortex and their applications to brain-machine interfaces,” *Nature Commun.*, vol. 6, Jul. 2015, Art. no. 7759.

[50] O. E. Krigolson *et al.*, “Electroencephalographic correlates of target and outcome errors,” *Exp. Brain Res.*, vol. 190, no. 4, pp. 401–411, Oct. 2008.

[51] O. E. Krigolson and C. B. Holroyd, “Evidence for hierarchical error processing in the human brain,” *Neuroscience*, vol. 137, no. 1, pp. 13–17, 2006.

[52] O. E. Krigolson and C. B. Holroyd, “Predictive information and error processing: the role of medial-frontal cortex during motor control,” *Psychophysiology*, vol. 44, no. 4, pp. 586–595, Jul. 2007.

[53] F. Bonini *et al.*, “Action monitoring and medial frontal cortex: Leading role of supplementary motor area,” *Science*, vol. 343, no. 6173, pp. 888–891, Feb. 2014.

[54] H. T. van Schie *et al.*, “Modulation of activity in medial frontal and motor cortices during error observation,” *Nature Neurosci.*, vol. 7, no. 5, pp. 549–554, May 2004.

[55] M. Inoue *et al.*, “Error signals in motor cortices drive adaptation in reaching,” *Neuron*, vol. 90, no. 5, pp. 1114–1126, Jun. 2016.

[56] S. D. Stavisky *et al.*, “System identification of brain-machine interface control using a cursor jump perturbation,” in *Proc. 2015 7th Int. IEEE/EMBS Conf. Neural Eng.*, 2015, pp. 643–647.

[57] T. Milekovic *et al.*, “Error-related electrocorticographic activity in humans during continuous movements,” *J. Neural Eng.*, vol. 9, no. 2, Apr. 2012, Art. no. 026007.

[58] S. D. Stavisky *et al.*, “Motor cortical visuomotor feedback activity is initially isolated from downstream targets in output-null neural state space dimensions,” *Neuron*, vol. 95, no. 1, pp. 195.e9–208.e9, Jul. 2017.

[59] R. C. Miall and D. M. Wolpert, “Forward models for physiological motor control,” *Neural Netw.*, vol. 9, no. 8, pp. 1265–1279, Nov. 1996.

[60] C. B. Holroyd and M. G. H. Coles, “The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity,” *Psychol. Rev.*, vol. 109, no. 4, pp. 679–709, Oct. 2002.

[61] J. A. Perge *et al.*, “Reliability of directional information in unsorted spikes and local field potentials recorded in human motor cortex,” *J. Neural Eng.*, vol. 11, no. 4, Aug. 2014, Art. no. 046007.

[62] J. A. Perge *et al.*, “Intra-day signal instabilities affect decoding performance in an intracortical neural interface system,” *J. Neural Eng.*, vol. 10, no. 3, Jun. 2013, Art. no. 036004.

[63] P. W. Ferrez and J. del R. Millán, “Simultaneous real-time detection of motor imagery and error-related potentials for improved BCI accuracy,” in *Proc. 4th Int. Brain–Comput. Interface Workshop Training Course*, 2008, pp. 197–202.

[64] T. Aflalo *et al.*, “Neurophysiology. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human,” *Science*, vol. 348, no. 6237, pp. 906–910, May 2015.

[65] C. Ethier and L. E. Miller, “Brain-controlled muscle stimulation for the restoration of motor function,” *Neurobiol. Dis.*, vol. 83, pp. 180–190, Nov. 2015.

[66] A. B. Ajiboye *et al.*, “Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration,” *Lancet*, vol. 389, no. 10081, pp. 1821–1830, May 2017.

[67] P. Nuyujukian *et al.*, “A non-human primate brain computer typing interface,” *Proc. IEEE*, vol. 105, no. 1, pp. 66–72, Jan. 2017.

[68] L. R. Hochberg *et al.*, “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm,” *Nature*, vol. 485, no. 7398, pp. 372–375, May 2012.

[69] J. L. Collinger *et al.*, “High-performance neuroprosthetic control by an individual with tetraplegia,” *Lancet*, vol. 381, no. 9866, pp. 557–564, Feb. 2013.

[70] A. R. C. Donati *et al.*, “Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients,” *Sci. Rep.*, vol. 6, 2016, Art. no. 30383.

[71] J. C. Barrese *et al.*, “Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates,” *J. Neural Eng.*, vol. 10, no. 6, Dec. 2013, Art. no. 066014.

[72] J. Krüger *et al.*, “Seven years of recording from monkey cortex with a chronically implanted multiple microelectrode,” *Front. Neuroeng.*, vol. 3, p. 6, May 2010.

[73] D. Sussillo *et al.*, “Making brain-machine interfaces robust to future neural variability,” *Nature Commun.*, vol. 7, Dec. 2016, Art. no. 13749.

[74] E. A. Pohlmeyer *et al.*, “Using reinforcement learning to provide stable brain-machine interface control despite neural input reorganization,” *PLoS One*, vol. 9, no. 1, Jan. 2014, Art. no. e87253.

[75] J. C. Kao *et al.*, “Leveraging historical knowledge of neural dynamics to rescue decoder performance as neural channels are lost: Decoder hysteresis,” in *Proc. 2015 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2015, pp. 1061–1066.

[76] A. L. Orsborn *et al.*, “Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control,” *Neuron*, vol. 82, no. 6, pp. 1380–1393, Jun. 2014.

[77] Z. Li *et al.*, “Adaptive decoding for brain-machine interfaces through Bayesian parameter updates,” *Neural Comput.*, vol. 23, no. 12, pp. 3162–3204, Dec. 2011.

[78] P. Nuyujukian *et al.*, “A bluetooth wireless brain-machine interface for general purpose computer use,” presented at the Society for Neuroscience, Chicago, IL, USA, 2015, Program No. 748.01.