Overcoming Long-Term Variability in Local Field Potentials Using an Adaptive Decoder

Vijay Aditya Tadipatri*, *Student Member, IEEE*, Ahmed H. Tewfik, *Fellow, IEEE*, Giuseppe Pellizzer, and James Ashe

***Abstract*—Long-term variability remains one of the major hurdles in using intracortical recordings like local field potentials for brain computer interfaces (BCI). Practical neural decoders need to overcome time instability of neural signals to estimate subject behavior accurately and faithfully over the long term. This paper presents a novel decoder that 1) characterizes each behavioral task (i.e., different movement directions under different force conditions) with multiple neural patterns and 2) adapts to the long-term variations in neural features by identifying the stable neural patterns. This adaptation can be performed in both an unsupervised and a semisupervised learning framework requiring minimal feedback from the user. To achieve generalization over time, the proposed decoder uses redundant sparse regression models that adapt to day-to-day variations in neural patterns. While this update requires no explicit feedback from the BCI user, any feedback (explicit or derived) to the BCI improves its performance. With this adaptive decoder, we investigated the effects of long-term neural modulation especially when subjects encountered new external forces against movement. The proposed decoder predicted eight hand-movement directions with an accuracy of 95% over two weeks (when there was no external forces); and 85% in later acquisition sessions spanning up to 42 days (when the monkeys countered external field forces). Since the decoder can operate with or without manual intervention, it could alleviate user frustration associated with BCI.**

***Index Terms*—Brain computer interface (BCI), local field potentials (LFPs), movement decoding.**

I. Introduction

BRAIN computer interfaces (BCI) translate neural data into behavioral signals, like commands for movement, to aid patients suffering from paralysis or spinal cord injury in performing daily tasks. A successful practical BCI requires 1) a reliable acquisition modality with high signal-to-noise ratio (SNR); 2) a decoding algorithm that translates the acquired neural data to behavioral signals like arm position or velocity; and 3) short training sessions to reduce learning fatigue on users [1], [2]. Intracortical recordings like local field potentials (LFPs) and single unit activity (SUA) are suitable to decode intended movement direction as they have superior spatial localization and better SNR than less invasive modalities like Electroencephalography and Electrocorticography [3]–[7]. Recent advances in neural engineering allow long-term recording of these invasive modalities over extended periods of time. For example, researchers recorded SUA over 500 days in monkeys and SUA along with LFP over 1000 days in a patient suffering from quadriplegia [8], [9]. Since recording from the same single units over multiple days might not be generally feasible, LFP could provide a more stable modality for long-term recordings [10], [11]. However, a major impediment in using LFP for BCI is their inherent nonstationarity and instability that occur due to variations in subject motivation, behavior, and learning; micromovements in electrode positions; and changes in electrode impedance over days [12]. These nonstationarities manifest in various forms including variations in signal power, and changes in neural spatial patterns even when performing the same behavioral task.

LFP variabilities could be addressed with a retraining session to calibrate BCI for new characteristics [13], or by rapid prototyping [14]. However, such pauses are inefficient and increase user frustration [1]. Invariant spatial patterns can be derived from training data to overcome signal variations [13]. This requires all variations of neural data to be captured during training and is generally not feasible due to the limited nature of the training dataset. An optimal linear filter, such as a Wiener filter could capture the correlations in the neural data [10]. These filters require the calculation of ensemble average of the cross-correlation and autocorrelation matrices. However, implementation of a Wiener filter requires the number of trials to be much larger than the number of data samples in order to avoid ill conditioning of the aforementioned matrices [15]. Prior research has focused on developing qualitative spatial patterns [16] and features lying on recurring subspaces [17] to overcome nonstationarity. While these approaches address the dynamic nature of LFP and its nonstationarity, they do not adapt to the day-to-day variations and fail to capture the long-term variability of the signals. Closed-loop BCI achieve long-term decoding by placing the learning burden on the user to adapt and learn the decoder performance [18], [19]. Day-specific decoders using state-of-the-art dynamical models were shown to perform at above 90%

Manuscript received June 7, 2015; revised November 11, 2015 and January 22, 2016; accepted April 12, 2016. Date of publication April 20, 2016; date of current version January 18, 2017. This work was supported in part by Merit Review Award CX-0004-37 from the U.S. Department of Veterans Affairs Clinical Sciences Research and Development Program. *Asterisk indicates corresponding author.*

*V. A. Tadipatri is with the Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712, USA (e-mail: vijay.tadipatri@utexas.edu).

A. H. Tewfik is with the Department of Electrical and Computer Engineering, The University of Texas.

G. Pellizzer and J. Ashe is with the Minneapolis VA Health Care System, and also with the Department of Neuroscience, University of Minnesota.

Digital Object Identifier 10.1109/TBME.2016.2557070

accuracy in estimating hand kinematics [20]. Further, adapting BCI in a closed-loop experiment showed improved performance in an SUA-based BCI [21]. In these experiments, reward to the BCI user on completing the task was used as direct feedback to update the BCI decoder. However it is estimated that 20–25% of users are unable to achieve such adaptation and control the BCI [22]. Hence, there is a need for a decoder that learns the user mental state in an open-loop fashion instead of constraining the user to learn a fixed decoder pattern [22]. In this paper, we propose a novel adaptive open-loop decoder that improves decoding and relieves the learning load from its user. We propose to take advantage of the man–machine learning dilemma and adapt the decoder (machine) continuously without interrupting the user [14]. The proposed method decoded the direction of eight targets accurately (>85%) from neural data recorded on two monkeys over multiple (4–6) weeks (28–42 sessions) as they performed center-out reaching tasks.

The proposed decoding model predicts the intended target direction by estimating the hand position in terms of its horizontal and vertical components from LFP. Regression is well suited for such analysis as it provides continuous control and generalizes to novel targets and environments [23]. Our first innovation proposes that *each behavior is best represented by multiple neural patterns instead of a single average pattern.* To identify these multiple representative patterns, this paper uses a kernel-based regression method called relevance vector machine (RVM) [24]. Using an RVM, arm movement decoders are trained in a Bayesian framework and provide a generalized sparse solution on the training data [24].They work on the premise that a few "relevant" training vectors describe the training feature space accurately, and thus, can generalize beyond the training data. However, these methods require relatively stationary feature space and fail to perform at the same level when the trained relevant vectors fail to describe the new features [25]. This situation arises in BCI, where feature nonstationarity is typical [18], [19]. While retraining the entire decoder to suit the necessities of a new feature set is an option, data collection to train the decoding model frustrates the user and diminishes the usability of BCI [1]. This paper offers a novel solution that updates the decoder online, in an unsupervised or semisupervised learning framework, without the necessity to collect calibration data. Adaptation proved an excellent boost to BCI in simulated neural data [26]. The second proposed innovation is *to adapt the decoder to track expected trajectories, which are modeled as a straight line from the center to the target.* Such a decoder update is a step closer to practical BCI and provides robust long-term performance even when the behavioral environment is altered. This paper provides results on data collected from both the dorsal premotor and primary motor cortices of two monkeys over multiple weeks.

The main contribution of this study is the successful development of an adaptive RVM decoder that maintains eight direction decoding accuracy over multiple weeks without direct retraining. In particular, an RVM tracks daily changes in the measured motor cortex signals that may be due to modifications of electrode or neural characteristics. The novelty of the proposed method lies in training a robust baseline decoder on a single session and adapt it over future sessions. We hypothesize that some neural patterns identified during this baseline training remain stable over the next recordings. The proposed decoder adapts by identifying such stable neural patterns and achieves robust decoding even in novel behavioral environments. We present both an unsupervised and a semisupervised adaptation strategy for adaptation. The semisupervised adaptation strategy merely requires a confirmation of target reach, which could be accomplished in practice by detecting error-related potentials in neural signals [27]–[32]. We present a simulation of such error detector and its impact on the overall adaptation. In our retrospective analysis, the proposed method decoded target directions with an accuracy above 96% over two weeks (96% using unsupervised adaptation) and 89% in later sessions (76% using unsupervised adaptation) with varying external forces against movement. Developing such an adaptive decoder also enables the use of the BCI system as a study tool to understand the dynamics of learning [33]. The rest of this paper is organized as follows: Section II summarizes the neural and behavioral data, the method and algorithms used for decoding; Section III presents the results obtained using these techniques and Section IV provides a discussion. Section V provides concluding remarks and related future work.

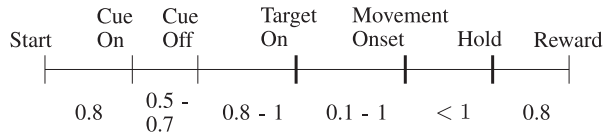

Fig. 1. Timeline (in seconds) of the neural data to be used in the analysis. Both monkeys were trained and required to reach the targets in this randomly variable timeline. The monkeys obtained a juice reward indicated in the figure after the successful reach and hold to the intended target.

II. Methods

A. Experimental Setup

Two male rhesus monkey subjects (Macaca mulatta), H564 and H464, weighing 6.1 and 4.5 kg, respectively, both left handed, were trained in an instructed-delay center-out task to perform a point-to-point movement to visually displayed targets using a manipulandum. Two silicon-based electrode arrays (Cyberkinetics, Foxboro, MA, USA) were implanted in the contralateral arm areas of the primary motor (M1) and dorsal premotor (PMd) cortices, respectively. Each array recorded signals from 64 electrodes (a total of 128) during every session. Under all task conditions, the subject executed 2-D horizontal reaching movements with the manipulandum from a central location to one of eight targets equally spaced around a circle of radius 9 cm. The center location and targets were displayed on a monitor placed at a distance of 55 cm in front of the subject. The monitor also provided the subject with real-time feedback of manipulandum position via a cursor. The subjects received a juice reward when they successfully performed the center-out-reach task within the time frame shown in Fig. 1. Behavioral data (position, velocity, forces, torques, and event markers) were also collected and stored. Only those trials that were performed correctly, determined online at the time of recording, were stored

for further analysis. LFP data were filtered between 0.3 and 220 Hz, and sampled at 500 Hz. Several sessions were conducted to record data over a period of four weeks (28 sessions) in H564 and six weeks (42 sessions) in H464. After the initial sessions (five sessions in H464 and three sessions in H564) external forces against the direction of movement were applied in some sessions, viz., stiff clockwise (SCW), viscous clockwise (VCW), and viscous counter clockwise (VCCW). However, the nature of forces against each direction in a particular session remained unchanged. We chose force constants to cause perceivable perturbations to the hand, while allowing the monkey to complete the hand reach. These perturbations represent changes in environmental conditions and compel the monkeys to modulate the neural signals to counter them. A successful decoder should overcome the nonstationarities introduced by these variations. Such a direction decoder is presented in the following section.

B. Feature Extraction

All LFP recordings were preprocessed to remove any noise due to electrode connectivity problems, baseline wanders and power-line noise resulting in $n = 86$ (H464) and 110 (H564) electrodes. To obtain robustness of decoders over multiple recording sessions and overcome signal dynamic range shifts over time, it is required to perform feature normalization [16]. To calculate the normalized features, instantaneous band power of each LFP channel in a specified time window is computed. The band power of all channels in this time window is sorted and the interchannel ranks are retained as the feature vector. These features are similar to the nonparametric statistical tests used for nonstandard data analysis [34], [35]. During initial analysis, we used the δ band (0.4–4 Hz) and calculate features in 250-ms time-windows with a 50% overlap. The rank features were calculated over the 1-s interval from movement onset to target reach [16], [36]. While the initial analysis focused on the δ band, using unfiltered data to extract features also provided statistically similar results ($p > 0.05$).

C. RVMs

The RVM uses a set of general models in the following form:

$$y(\mathbf{X}) = \sum w_i \Phi(\mathbf{X}, \mathbf{Y}_i) + w_0 \tag{1}$$

where $\mathbf{Y}_i$ are the learned neural features with corresponding weights w_i. $\Phi()$ is a kernel function that measures similarity between input neural features $\mathbf{X}$ and basis features. Training the decoder translates to estimating the weights $\mathbf{w}$ and their corresponding neural feature $\mathbf{Y}_i$. Maximum likelihood estimation of $\mathbf{w}$ without any constraints leads to overfitting on the training set [24]. The RVM framework obtains generalization via sparse formulation under the assumption that $\mathbf{w}$ is derived from a zero mean Gaussian distribution. The search for "relevant" vectors results in the best subset of input feature vectors that can represent a given input space. An RVM introduces a new set of hyperparameters α to set a Gaussian prior of the form

$$p(\mathbf{w}/\alpha) = \prod \mathcal{N}(w_i/0, \alpha_i^{-1})$$

$\mathbf{w}$ is estimated in an iterative fashion to optimize the marginal likelihood over α [37]. Usually, the chosen basis features $\mathbf{Y}_i$ are a set of prototypical examples from the input training vector set.

While a linear basis function also provided accurate decoding, we choose a Gaussian radial basis function to measure the nonlinear similarity between feature vectors and the trained relevant vectors, since a radial-basis support vector machine decoder provided successful decoding in a similar setting [38]. This is shown as

$$\Phi(\mathbf{X}, \mathbf{Y}) = \exp(-\frac{||\mathbf{X} - \mathbf{Y}||_2^2}{\sigma^2}) \tag{2}$$

where σ is the basis width that determines the spread of each basis featrure $\mathbf{Y}$ over the feature space.

D. Multioutput Regression

The RVM framework is formulated to provide a sparse solution for a single-hand-movement dimension. In this paper, we extend it to estimate multiple movement dimensions. One approach to estimate a multidimensional target is using an independent model for each dimension. Such models provide a good correlation of individual dimensions when they are independent but provide spurious results when dependence exists [39]. In our study, we trained decoders to independently track the x- and y-dimensions of hand movement. We observed that such independent estimations correlated well with the absolute position from the center $\sqrt{(x^2 + y^2)}$ (>0.92), but correlated poorly with the independent estimates of x- and y-positions (0.82). The technique proposed here is based on kernel dependence estimation (KDE) to leverage such dependence and obtain a better overall performance [40]. The use of KDE is partly motivated due its encapsulation of the multidimensional output dependence and partly due to its similarity to redundant classifiers used in multiclass classification, that tend to have better performance than separate binary classifiers [41].

A suitable target kernel function, Ψ reflects the nonlinear dependence between hand-movement dimensions ($\mathbf{y}$) and the location on the hand-movement space ($\mathbf{p}_i$) as follows:

$$\mathbf{\Psi}(\mathbf{y}, \mathbf{p}_i) = \exp(-(\mathbf{y} - \mathbf{p}_i)\Sigma_y^{-1}(\mathbf{y} - \mathbf{p}_i)^T). \tag{3}$$

The aforementioned equation is a Gaussian kernel evaluated at each point on the hand-movement space, where $\mathbf{p}_i$ denotes the space in the form of its horizontal, vertical, and absolute positions :$\{p_x, p_y, p_r\}$, and $p_r = \sqrt{(p_x^2 + p_y^2)}$. This kernel assesses the similarity between the actual recorded hand position and all locations on the hand-movement space. The dimension of this output kernel, $\mathbf{\Psi}$ increases with the number of points on the hand-movement space included in the analysis. To reduce the dimensionality of the model, first we decompose this output kernel to its independent singular vectors. As shown in the following equation, modeling the singular vectors instead of the

output kernel reduces the dimensionality of the decoder:

$$\begin{aligned}&\mathbf{\Psi} = USV^T\\&\{w_i, \mathbf{X}_i\}^k : \min \sum_{\mathbf{X}} ||U^k - \sum_i w_i^k \Phi(\mathbf{X}, \mathbf{X}_i)||_2 + \lambda||w_i^k||_1\end{aligned} \tag{4}$$

where $(.)^T$ is the transpose operator, U^k is the kth column of U and represents the kth abstraction component of movement. This results in redundant approximations and provides high correlation in all the target dimensions simultaneously. Choosing only top K singular vectors, which represent more than 90% of the basis energy, reduces the number of regression models. During the training phase, we learn regression parameters for each component. During the test phase, the neural features are projected on the learned input basis feature using $\mathbf{\Phi}$, and each abstract component of the output basis, $\hat{\Psi}$, is estimated with the aid of independent regression models. This model can be described by the kernel function parameters, basis features and the corresponding $\mathbf{w}$ as $\mathcal{M} := \{\mathbf{\Phi}(.), \mathbf{X}_i, \mathbf{w}_i\}$.

To obtain the estimates for new test samples $\mathbf{X}_{\text{test}}$, decoder $\mathcal{M}$ can be applied as

$$\hat{U}^k_{\text{test}} = \sum_i w_i^k \mathbf{\Phi}(\mathbf{X}_{\text{test}}, \mathbf{X}_i^k)\psi_{\text{test}} = \hat{U}_{\text{test}} SV^T \tag{5}$$

where ψ_{test} is the score vector for each hand position $\mathbf{p}_i$ on the movement space. One of the challenges for KDE during the testing stage is preimage identification that refers to identifying the hand position for a corresponding target kernel [40]. In the current setting, this refers to identifying the hand position from the estimates of the output kernel. In general, finding the preimage of a Gaussian kernel is difficult and preimages might not necessarily exist [42]. To estimate the exact hand position for the neural features, the output kernel ψ_{test} is calculated as shown in (5). Note that this score is calculated for each point of the hand-position workspace. The hand position is obtained by identifying the 2-d hand position that has the maximum score. The decoded target is determined as the target closest to the endpoint angle made by the trajectory, measured as $\Theta = \tan^{-1}\frac{\hat{y}_y}{\hat{y}_x}$.

E. Adaptation

The previously described decoder provides good approximation and generalization if the basis features from training can express the entire input feature space. However, noisy and nonstationary time-series environments change the feature characteristics, causing an RVM to fail in capturing the input dynamics and provides unstable approximations [25]. It has been well documented that neural recordings and their features change due to learning, micromotions of the electrodes and other environmental conditions [12]. In a typical BCI, the new neural feature vectors acquired on a different day (or session) tend to align poorly with previously collected basis features $\mathbf{X}_b$ [11], [16]. Hence, updates are required to adapt the decoder to the new data.

Fig. 2 shows the estimated trajectories using the decoder. If the actual trajectories y_{test} corresponding to $\mathbf{X}_{\text{test}}$ were available, adaptation only requires a correction to improve the estimate on

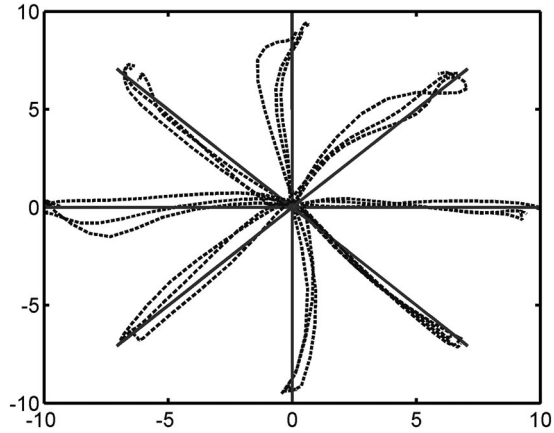

Fig. 2. Hand-movement workspace (10-cm radial grid) with the estimated hand movements ($\tilde{\mathbf{y}}_{\text{test}}$, dashed lines). Decoder adaptation chooses neural features that minimize the error between the straight line ideal movement ($\mathbf{y}_{\text{new}}$, solid lines) and the decoder estimates.

the residual trajectory

$$\begin{aligned}&e = y_{test} - \hat{y}_{test}\\&\{w_u, \mathbf{X}_u\} : \min \sum_{\mathbf{X}_{test}} ||e - \sum w_u \Phi(\mathbf{X}_{test}, \mathbf{X}_u)||_2\\&+ \lambda||\Phi(\mathbf{X}_u, \mathbf{X}_b)||_2\\&\mathcal{M}_u := \{\mathbf{\Phi}(.), \mathbf{X}_u, \mathbf{w}_u\}\end{aligned} \tag{6}$$

The first part of the aforementioned equation can be estimated using an RVM learning algorithm with similar constraints shown previously in Section II-C. The penalty on the already included basis features ensures that the baseline decoder remains unaffected with the update ($\mathbf{w}_b$ needs no update). In the absence of such a penalty, the fit on the training data would suffer due to the addition of new basis features.

While it is desirable for BCI to build the decoder update on actual (or intended) hand kinematics, they are unavailable in practical implementation. Our key innovation is to mimic a practical BCI even when no prior knowledge of the actual hand trajectories y_{new} exists. For this, we incorporate general principles of natural movements by assuming that the monkey *intends to reach the target in a straight line from the center to the target*. We propose to use the estimated target direction from the baseline decoder to obtain the intended path. Thus, we construct the intended linear trajectory from the center to the target as $\tilde{y} = \mathcal{F}(\hat{\theta})$ as shown in Fig. 2.

The overall decoder used for the succeeding trials will be $\mathcal{M}_* = \mathcal{M}_0 \oplus \mathcal{M}_u$, where $\oplus$ is a suitable appending function and $\mathcal{M}_0 = \{\mathbf{\Phi}(.), \mathbf{X}_0, \mathbf{w}_0\}$ is baseline decoder before this update. Since the current decoder structure is linear in the kernel function space, the updated decoder is obtained as follows:

$$\mathcal{M}_* = \{\mathbf{w}_0||\mathbf{w}_u, \mathbf{\Phi}(.), \mathbf{X}_0||\mathbf{X}_u\} \tag{7}$$

$$\hat{y}_*(\mathbf{X}_*) = \sum \mathbf{w}_0^T \mathbf{\Phi}(\mathbf{X}_*, \mathbf{X}_0) + \sum \mathbf{w}_u^T \mathbf{\Phi}(\mathbf{X}_*, \mathbf{X}_u). \tag{8}$$

Since the decoder update involves only a few sample corrections, it takes less time to process the update and can be performed

TABLE I
CORRELATION COEFFICIENTS OF HAND POSITION PREDICTIONS AND MOVEMENT DIRECTION DECODING ACCURACY FOR THE FIRST FEW SESSIONS OF MONKEY H464 AND H564

Decoder Age (days)	# of Trials	ρ_x	ρ_y	Decoding Accuracy
H464				
8	263	0.94	0.91	93%
9	325	0.93	0.91	89%
13	348	0.88	0.87	82%
14	88	0.76	0.75	66%
H564				
8	206	0.89	0.85	80%
9	103	0.79	0.75	66%

During these sessions, no external field forces were introduced to the monkeys.

online as soon as a trial has finished. The update process can be further improved by performing the update only on successful trials, determined online by the BCI user. In our implementation, the user feedback involves only a binary decision on whether the desired target was reached. This is similar to the reward-based feedback approach used by DiGiovanna *et al.* [21].

III. RESULTS

To understand the long-term decoding and movement tracking capabilities of the decoder, we train an initial decoder on the first session and evaluate the decoder adaptation on successive sessions. The performance is evaluated in terms of decoding accuracy: fraction of trials that the decoder accurately predicted the intended direction of movement. We also use correlation coefficient between the original (recorded from the monkey hand) and the estimated trajectory (ρ_x, ρ_y). We present the correlation and decoding accuracy over time as a function of decoder age. Note that in this offline retrospective analysis the subjects performed the task in an open loop and could not actively learn the decoder. As discussed previously, there is no necessity of a calibration session and the decoder learns actively from the predicted hand positions in a reinforced manner.

Table I presents the decoding results obtained using the static version of the proposed model over the initial sessions, where no external field forces were applied. Note that the decoding accuracy decreases over decoder age (number of days between training and testing session) for both the monkeys. To overcome the variability of neural features over multiple days, we present an adapting decoder. The model is adapted to account for the varying signal characteristics after decoding 25 target reaches. Updating the decoder at the end of each trial results in an oversensitive update. While such an update presents a new model at every trial, it also requires an additional processing time (to update the decoder) at the end of every trial. Conversely, update after processing a large number of trials—passive model—might not track the fluctuations fast enough and will be ineffective for short term variabilities. Thus, the update process must choose an optimal number of trials to update the model. In the current analysis, the decoding performance varied little ($<2\%$) across the choice of different number of trials. Fig. 3 shows the performance of the adaptation algorithm in comparison with the static decoder over multiple blocks even if these blocks are spanned

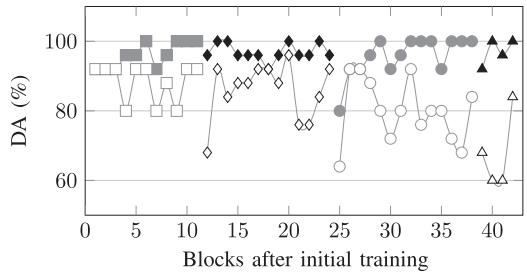

Fig. 3. Decoding accuracy (DA) with adaptation (filled) and without adaptation (unfilled) across multiple adaptation blocks of 25 trials each spread across four testing days. The gaps in the accuracy plots represent the end of day represented in different shapes.

TABLE II
DECODING ACCURACY (IN %) COMPARISON BETWEEN DIFFERENT ADAPTATION STRATEGIES

Decoder Age (in Days)	8	9	13	14
H464				
No update	93	89	82	66
Daily Decoder Reset	98	96	92	82
Continuous Update	98	97	96	97
H564				
No Update	80	66		
Daily Decoder Reset	81	70		
Continuous Update	81	80		

During daily decoder reset, the decoder is reset to $\mathcal{M}_0$ at the beginning of each day. For continuous adaptation, the decoder is continually adapted thus keeping memory of its previous learning.

over different days. The vertical axis shows the decoding accuracy measured as the fraction of all correct predictions up to the current instant over successive adaptation blocks (25 trials).

Table II compares two different adaptation strategies and their performance over the decoder age. All the strategies started with the decoder $\mathcal{M}_0$ calculated on day 0 and both adaptation strategies assume accurate user feedback to improve decoding accuracy. To observe the long-term effects of adaptation, we adapted one version of the decoder continuously over the two weeks of test data. Another version of the decoder adapted only the current test session and ignored any previous adaptation by resetting the decoder to $\mathcal{M}_0$ at the beginning of each session. To provide a baseline, we also present the decoding results with a static decoder.

Common spatial patterns (CSPs) are traditionally used to identify existing task-dependent spatial patterns [36], [43]. In our analysis, we observed direction-dependent spatial distribution of high-powered channels. We observed that while the dynamic range of the channels varied over different days, their relative power remained fairly consistent. To extract these patterns of qualitative information, we used interchannel instantaneous power ranks [16]. CSP and its variants perform a single-trial analysis and provide only the final target. The proposed method utilizes aspects of regression analysis similar to the Wiener filtering in [10].

Table III compares the proposed method with state-of-the-art CSP method and its variant [38], [44] along with the Wiener filter [10]. To provide a fair comparison, the decoding model uses

TABLE III
COMPARISON OF DECODING ALGORITHMS DURING INITIAL SESSIONS FOR H464

Algorithm	Before Introducing Field Forces (two weeks) Decoding Power (ρ_x, ρ_y)	During Varying Field Forces (four weeks) Decoding Power (ρ_x, ρ_y)	All Sessions (six weeks) Decoding Power (ρ_x, ρ_y)
CSP	34	13	15
Rank CSP	47	18	21
Wiener Filter	46	28	30
	(0.42,0.49)	(0.24,0.12)	(0.26,0.14)
Proposed Method	81	31	36
	(0.88,0.86)	(0.33,0.25)	(0.36,0.29)
Adaptive CSP	40	13	16
Adaptive Rank CSP	77	49	52
Adaptive Wiener Filter	74	50	53
	(0.81,0.79)	(0.63,0.62)	(0.64,0.59)
Proposed Method With Adaptation	96	89	90
	(0.95,0.91)	(0.78,0.74)	(0.77,0.74)

This table presents decoding accuracy of CSP, Rank-CSP, Wiener Filter, and proposed decoder, including their versions of adaptation. For the Wiener Filter and the proposed methods, the correlation coefficients (ρ_x and ρ_y) are also presented. Since CSP is a discrete direction decoder correlation coefficients are not indicative of performance.

multiple redundant linear classifiers and the final output delivered through an error correcting output code [41]. We optimized the lag parameter of the autoregression in the Wiener filter to provide the best estimate of hand position. The end-trial angle was tracked from the hand position using similar strategies described in Section II-D. For comparison, we applied a similar adaptation strategy on all decoders, adapting them after every 25 trials. Adaptation of CSP was achieved by tracking the trial average covariance matrices and updating them during testing with trials that were identified correctly. Similarly, adaptation of Wiener filters was performed by updating the autoregression matrices. As shown in Table III, the proposed algorithm outperforms the traditional methods, especially over increasing decoder age. Since continuous tracking is possible with a Wiener filter and the proposed RVM-based method, the correlations between the actual hand movements and estimated movements in x- and y-dimensions (ρ_x, ρ_y) are also presented.

Finally, to observe the effects of long-term learning, we applied the static decoder learnt on day 0 to all recordings that spanned over six weeks (H464) and four weeks (H564) and compared the results with a continually adapting decoder. In this simulation, the decoder adapted across sessions conducted over multiple days and in varying environmental conditions. Figs. 4 and 5 show the decoding performance of such an adaptation over all the sessions in comparison to the same algorithm without adaptation. Further, the correlation coefficients (ρ_x, ρ_y) are also presented. Note that after two weeks in H464 (1 week in H564), various field forces were applied against the movement to the target (indicated by different shapes in the figures). After both monkeys learned to reach the targets against this type of curl force, the VCCW force became the new normal. The initial decoder $\mathcal{M}_0$ learnt for both monkeys was on sessions with no field forces recorded on day 0. Even when new field forces were encountered the adaptation strategy remained the same. We observed that this adaptive decoder outperforms the static decoder even in novel field force conditions and stabilizes its performance over time.

IV. DISCUSSION

This paper presents a neural arm movement decoder that predicts eight radially spaced targets. LFP recorded from both premotor and primary motor area were processed by filtering in the delta band to calculate the instantaneous power of individual channels. The decoder used qualitative interchannel rank patterns to overcome the dynamic variability of raw power-based features across multiple recording sessions. LFP possesses a $\frac{1}{f}$ frequency response, and hence, most of the LFP power is contained in the lower frequency bands like the δ-band. Since the decoder only uses the relative ranking of the channels, we expect similar performance for band-limited or broadband LFP features. Indeed, we observed that the decoding accuracy was not statistically different ($p > 0.05$), when the features were derived from either δ-band or the raw LFP.

As shown in Fig. 3, the adaptation algorithm maintains the decoding consistently across multiple blocks and over multiple days, while the performance of the baseline algorithm slowly decays over successive blocks. As hypothesized, we observed that adaptation provides both short-term improvements within sessions and also long-term improvements across sessions.

This paper presented two strategies in terms of user feedback to update the decoder. The improvement in performance due to decoder update for both strategies depends on the quality of the initial decoder; if the initial decoder performs poorly, the error is propagated to the future sessions. The semisupervised approach to decoder update provided improved performance for all decoders. For example, when all decoder predictions were considered (irrespective of their accuracy) for update, the CSP algorithm's decoding performance deteriorated to random levels of classification within two sessions. However, the performance of the proposed method remained at an average of 78% over six weeks. To provide a comparative analysis, Table III presents results from the semisupervised adaptation strategy for all the decoders. In this strategy, the decoder mitigates error propagation by retraining only on the few trials that had successful reaches. We observed that the performance of CSP drops to

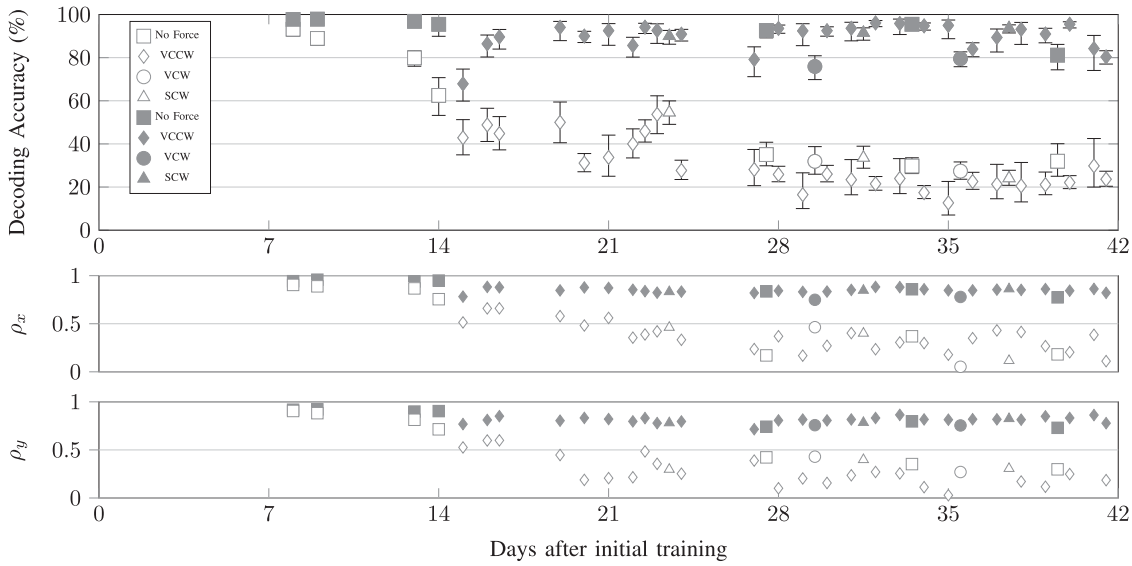

Fig. 4. Decoding accuracy in all recorded sessions from monkey H464. The filled icons represent accuracy of adapting decoder and the unfilled icons represent the accuracy of the baseline static decoder. The error bars in the top panel represent the 95% confidence limits. For ease of reading, different field forces have been represented with different shapes. The bottom two panels represent the correlation coefficients (ρ_x and ρ_y) in the respective sessions.

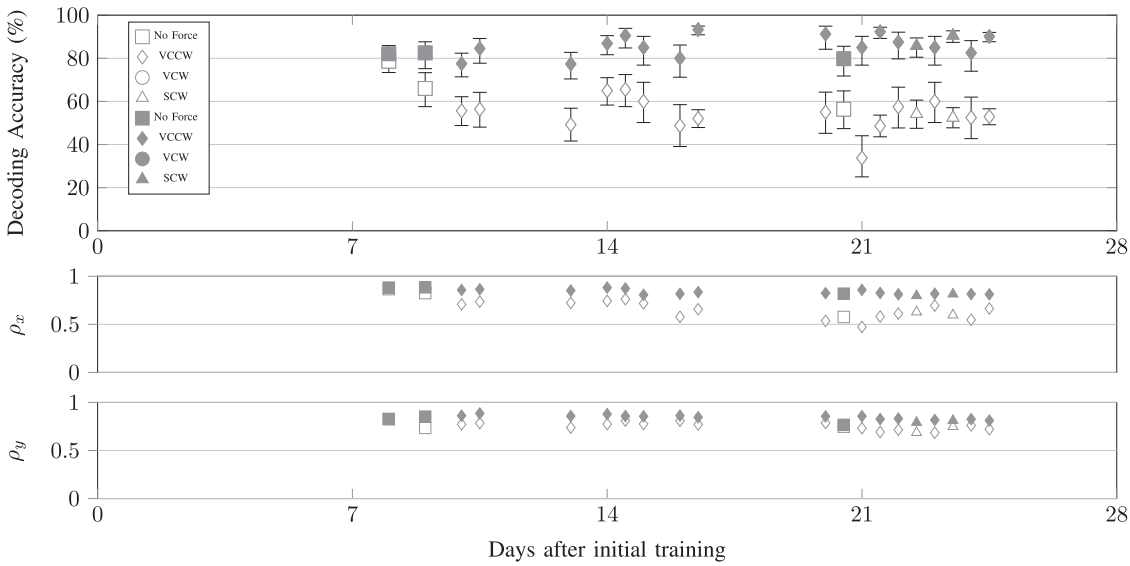

Fig. 5. Decoding accuracy in all recorded sessions from monkey H564. The filled icons represent accuracy with adapting decoder and the unfilled icons represent the accuracy of the baseline static decoder. The error bars in the top panel represent the 95% confidence limits. For ease of reading, different field forces have been represented with different shapes. The bottom two panels represent the correlation coefficients (ρ_x and ρ_y) in the respective sessions.

almost random classification level by two weeks even in such an adaptation strategy. CSP fails to incorporate the signal variability over time, and thus, reports lower performance, while CSP applied on the rank features provides a stable performance by virtue of using robust features. However, the proposed method outperforms both versions of CSP and provides robust performance over time and external forces. Table III also shows the performance of the algorithms when they are adapted using techniques described in II-E.

First, we note that adaptation improved the performance significantly (at $p < 0.05$) of all algorithms over their static versions in the first few sessions. However, owing to their poor performance with a static decoder, we observed that the advantage of adaptation quickly deteriorated in the traditional decoders. In fact in the initial sessions without field forces, the static version of the proposed method outperforms the adaptive versions of traditional methods. Since the traditional methods use an ensemble average of all neural patterns, they fail to

capture variability in the neural features. In the sessions where field forces were applied, the modulated features were different from the training features and results in the poor performance of the static decoder. These results show that robust performance requires a robust static decoder for adaptation. The algorithm presented in this paper decodes the intended movement successfully without adaptation and its performance is enhanced with adaptation. The success of the decoder is due to its encoding of behavior in multiple neural patterns. While the comparative Wiener filter provided better decoding over CSP algorithms, it assumes a linear association between the neural features and the hand-position metrics [10], [15]. Further, it also depends on ensemble averaging across trials from multiple days and carries the same disadvantages of CSP and fails to represent the entire feature space [15]. The proposed decoder uses a nonlinear kernel and uses multiple relevant features (instead of the ensemble average) to encapsulate this relationship and hence is able to provide improved decoding.

From Table II, we observe that adaptation improves the target decoding accuracy over the two weeks for different adaptation strategies. For the static decoder, the accuracy drastically fell around day 14, but adaptation stabilized the accuracy over 95%. In this analysis, following two strategies were explored: 1) adapting the decoder on only the current session and 2) adapting the decoder continuously over all sessions. While decoder adaptation only on the current day improves the decoding accuracy, its performance gradually decreases with the decoder age due to evolution of new neural patterns. From these results, we can infer that learning modulates the neural activity continuously (rather than daily) and decoders benefit from the adaptation to variations introduced by this learning. On the other hand, even in the absence of the most recently updated decoder, resuming the adaptation of the available version of decoder still improves the decoding performance than by using a static decoder.

From Figs. 4 and 5, we observed that the adaptive decoder provides stable decoding over time and across varying field forces. Note that the adapting strategy did not change with the field forces and remained the same. We observed that decoding performance fell when a novel field force was introduced to the subject. This is observed during the first VCCW session in both the monkeys (H464—day 15, H564—day 10), and when other field forces (VCW and SCW) were introduced in later sessions. This drop in performance was anticipated because the field forces were introduced to the monkeys for the first time. In fact, the monkeys were unable to immediately cope with this abrupt introduction of force against the manipulandum. They needed two sessions to adjust to this change and reach the targets with the same level of success and efficiency. Once the monkeys became accustomed to these forces, they reached the targets efficiently, and we hypothesize that they generated stable neural patterns. From the results in the figures, we can observe that the decoding accuracy stabilizes in sessions following the field forces as the adaptive decoder learns these stable patterns. Interestingly, the decoding performance dipped when the applied field force was opposite to the anticipated direction (H464 day 30, 36 with VCW forces).

As mentioned in Section II-E, the semisupervised decoder adaptation requires user feedback on the accuracy of the intended target reach. The feedback mechanism used in this paper is similar to the simultaneous reward feedback to both subject and the closed-loop BCI used for coadaptation by DiGiovanna *etal.* [21]. While in our retrospective analysis, the subjects did not have an opportunity to learn the updated decoder, we note that reward from successful target reaches could be effective in decoder adaptation. Such feedback to confirm the intended target reach might be provided by vocal and tactile features [29], [31], [45]. Such user feedbacks have also been studied during other simulation studies that propose the neural decoder to obtain feedback from another error detector as shown in Fig. 6 [26], [45]. In fact, such automatic feedback has been used to boost BCI bit-rate, when error trials were detected with an accuracy of 75% [46]. As mentioned previously, the proposed neural decoder in this paper only requires a binary feedback on its target reach accuracy. The adaptation strategy then uses only those neural patterns that resulted in accurate target reaches. Please note that while such adaptive decoders close the loop between the overall BCI and the user, the learning burden is placed entirely on the neural decoder and allows the user to generate new patterns.

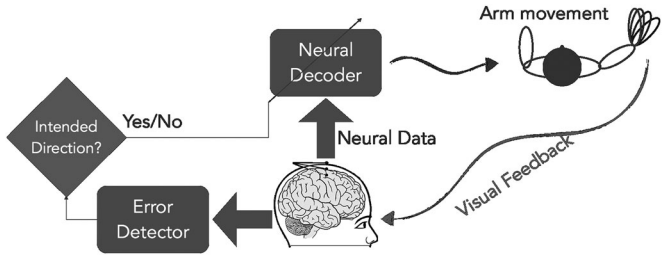

Fig. 6. Recommended schematic of a composite BCI including an Error Detector to provide feedback on target reach. The outcome of the Error Detector could modulate the adaptation of the Neural Detector.

Employing such an error detector, with its own error characteristics, would automate the feedback to the neural arm movement decoder. To simulate the presence of such feedback, we incorporated a random error generator and used it to provide feedback on the target reach. Table IV presents the average decoding of ten such simulations. We make two interesting observations. First, providing any form of feedback improves the performance in comparison to without feedback. As expected the best performance is observed with a 100% accurate feedback. We compare the results with the performance of a static decoder and a decoder that works in an unsupervised manner, assuming that its predictions are always accurate. The impact of accurate feedback is evident in both monkeys. Even with an unsupervised update (including both successful reaches and unsuccessful reaches), the strategy improved accuracy significantly (from 36% to 78% in H464 and 56% to 72% in H564 as shown in Table IV). Second, this strategy reduces the impact of error propagation. In such updates, the decoder awarded higher weight to successful target reaches while imparting lower weights to unsuccessful target reaches.

Table V presents other studies using LFP to decode similar behavioral tasks. These studies varied in behavioral and experimental settings and used different features to decode movement. Thus, there is no direct way of comparing the results but are

TABLE IV
IMPACT OF FEEDBACK ACCURACY ON ADAPTIVE DECODER PERFORMANCE

Session	Before Introducing Field Forces (1–2 weeks)	During Varying Field Forces (3–4 weeks)	All Sessions (4-6 weeks)
H464			
Static Model	81	31	36
Unsupervised feedback	96*	76*	78*
75% Accurate feedback	96*	86*†	87*†
80% Accurate feedback	97*	87*†	88*†
90% Accurate feedback	96*	88*†	89*†
100% Accurate feedback	96*	89*†	90*†
H564			
Static Model	72	54	56
Unsupervised feedback	70	72*	72*
75% Accurate feedback	77	81*†	81*†
80% Accurate feedback	78	84*†	83*†
90% Accurate feedback	79*	86*†	85*†
100% Accurate feedback	81*†	87*†	86*†

Performance of the algorithm during different recording phases is compared. For this comparison, the decoder was trained on neural data from day 0 and its performance evaluated on future sessions. * represents significant improvement from the static model and † represents significant improvement (at $p < 0.05$) over the unsupervised feedback.

TABLE V
DECODING POWER (DP) OF OTHER LITERATURE IN COMPARISON WITH THE PROPOSED METHOD

Algorithm	Decoding Accuracy
Directional Tuning [7]	50%
Bayesian Classification [5]	81%
Wiener Cascade Filter [10]	80%
RE-FIT Kalman Filter [20]	>90%
Reinforcement Learning [21]	68%–74%
Proposed Method	86% –90%

presented here for contextual reference. The results in this paper are in line with recent publications that show long-term decoding in an online setting or via coadaptation, where the user adjusts to the decoder (static or adaptive) and stabilizes neural patterns [11], [18], [20], [21]. Similarly, other studies have shown advantages of an adaptive decoder in simulated neural data [26], [45]. Results in this paper show the feasibility of LFP in long-term decoding applications even in an open-loop setting. Further, we show that an adaptive decoder overcomes variabilities inherent to LFP including changes in behavioral environments, recording limitations, and variabilities due to learning. The actual source of the nonstationarity and its impact will require additional studies. These results suggest that a similar adaptive or coadaptive decoder can perform better in a closed-loop setting, as both the user and the decoder adapt continuously and in synergy, relieving BCI user fatigue and "illiteracy."

V. CONCLUSION

This paper introduces a novel adaptive algorithm to decode the intended movement direction from LFP, overcoming signal nonstationarities and variability. A significant element of this algorithm is a robust baseline decoder that models a movement to each direction with multiple representative neural patterns. Since the same target could be reached in different ways, multiple neural patterns might define a target reach better than an average pattern. We show that such a decoder can be updated without prior knowledge of the actual hand trajectories using only a straight line estimate. The performance of the algorithm to trace a challenging trajectory (like a curve or in the presence of obstacles) is to be explored in the future. The adaptation allows decoding of eight radial targets at above 85% in recordings 4–6 weeks from initial decoder training, including sessions where behavioral environments were altered.

This study did not focus on selecting or optimizing the recorded electrodes and further analysis might be required for such optimization. The significant advantage of using LFP over SUA is their long-term recording stability. However, LFP might also be susceptible to loss in recordings resulting due to electrode connectivity and other noisy conditions. While, the current study assumes the same set of electrodes to provide recordings, future studies could investigate the robustness of decoders in the absence of some recordings. Further, the algorithm proposed in this paper is specifically tuned to decode movement direction and is limited to reaching a predetermined set of discrete targets. Future studies shall explore how to scale the algorithm to accomplish high level tasks, especially those involving movement of a cursor to a random location. Another area of research could explore if new targets initially unknown to the decoder might also be reached during adaptation.

In our simulations, we observed that the decoder that adapted continuously over all sessions performed better than a decoder that ignores adaptation over previous days. From these results (see Table II), we infer that decoder learning is a continuous process and there is advantage in adapting the decoder over multiple sessions rather than just the current session. However, in a situation where a series of errors are made the decoder, the supervised adaptation might not be successful and error propagation might occur.

Finally, closed-loop BCI might also benefits from such an adaptive model, where the users modulate neural patterns in response to BCI adaptation [21]. In such a scenario, the adaptation burden is shared by BCI and its user, making the man–machine learning dilemma an advantage by reinforcing neural patterns that elicit accurate responses. These benefits can ultimately reduce BCI frustration and fatigue.

REFERENCES

[1] A. Nijholt and D. Tan, "Brain-computer interfacing for intelligent systems," *IEEE Intell. Syst.*, vol. 23, no. 3, pp. 72–79, May/Jun. 2008.
[2] J. R. Wolpaw *et al.*, "Brain computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–791, Jun. 2002. [Online]. Available: http://dx.doi.org/10.1016/S1388-2457(02)00057-3
[3] A. Georgopoulos *et al.*, "Neuronal population coding of movement direction," *Sci.*, vol. 233, pp. 1416–1419, 1986.
[4] I. Asher *et al.*, "Comparison of direction and object selectivity of local field potentials and single units in macaque posterior parietal cortex during prehension," *J. Neurophysiol.*, vol. 97, no. 5, pp. 3684–3695, 2007.
[5] 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. [Online]. Available: http://dx.doi.org/10.1016/j.neuron.2005.03.004
[6] 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. [Online]. Available: http://dx.doi.org/10.1523/JNEUROSCI.0816

[7] 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. [Online]. Available: http://dx.doi.org/10.1152/jn.00106.2006

[8] A. S. Dickey *et al.*, "Single-unit stability using chronically implanted multielectrode arrays," *J. Neurophysiol.*, vol. 102, no. 2, pp. 1331–1339, Aug. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2724357/

[9] 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, p. 025027, Apr. 2011. [Online]. Available: http://iopscience.iop.org/1741-2552/8/2/025027

[10] 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, p. 046006. [Online]. Available: http://stacks.iop.org/1741-2552/9/i=4/a=046006

[11] R. D. Flint *et al.*, "Long term, stable brain machine interface performance using local field potentials and multiunit spikes," *J. Neural Eng.*, vol. 10, no. 5, p. 056005, 2013. [Online]. Available: http://stacks.iop.org/1741-2552/10/i=5/a=056005

[12] W. Jensen and P. Rousche, "On variability and use of rat primary motor cortex responses in behavioral task discrimination," *J. Neuro Eng.*, vol. 3, pp. L7–L13, 2006.

[13] B. Blankertz *et al.*, "Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing," in *Advances in Neural Information Processing Systems*, vol. 20. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.5159

[14] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," in *Proc. IEEE*, vol. 89, no. 7, pp. 1123–1134, Jul. 2001. [Online]. Available: http://dx.doi.org/10.1109/5.939829

[15] Z. Wang *et al.*, "Single-trial evoked potential estimation using wavelets," *Comput. Biol. Med.*, vol. 37, no. 4, pp. 463–473, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.compbiomed.2006.08.011

[16] V. A. Tadipatri *et al.*, "Time robust movement direction decoding in local field potentials using channel ranking," in *IEEE Eng. Med. Biol. Mag.*, vol. 29, 2010.

[17] B. Gowreesunker *et al.*, "A subspace approach to learning recurrent features from brain activity," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 3, pp. 240–248, Jun. 2011.

[18] A. Orsborn *et al.*, "Closed-loop decoder adaptation on intermediate timescales facilitates rapid BMI performance improvements independent of decoder initialization conditions," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 4, pp. 468–477, Jul. 2012.

[19] B. Jarosiewicz *et al.*, "Advantages of closed-loop calibration in intracortical braincomputer interfaces for people with tetraplegia," *J. Neural Eng.*, vol. 10, no. 4, p. 046012. [Online]. Available: http://stacks.iop.org/1741-2552/10/i=4/a=046012

[20] V. Gilja *et al.*, "A high-performance neural prosthesis enabled by control algorithm design," *Nat. Neurosci.*, vol. 15, no. 12, pp. 1752–1757, Dec. 2012. [Online]. Available: http://dx.doi.org/10.1038/nn.3265

[21] J. DiGiovanna *et al.*, "Coadaptive brain machine interface via reinforcement learning," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 1, pp. 54–64, Jan. 2009.

[22] C. Vidaurre and B. Blankertz, "Towards a cure for BCI illiteracy," *Brain Topography*, vol. 23, no. 2, pp. 194–198, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1007/s10548-009-0121-6

[23] D. McFarland and J. Wolpaw, "Sensorimotor rhythm-based brain-computer interface (BCI): Feature selection by regression improves performance," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 372–379, Sep. 2005.

[24] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001. [Online]. Available: http://dx.doi.org/10.1162/15324430152748236

[25] N. Nikolaev and P. Tino, "Sequential relevance vector machine learning from time series," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, 2005, pp. 1308–1313.

[26] T. Gürel and C. Mehring, "Unsupervised adaptation of brain machine interface decoders," *CoRR*, vol. abs/1206.3666, 2012. [Online]. Available: http://arxiv.org/abs/1206.3666

[27] M. J. Frank *et al.*, "Error-related negativity predicts reinforcement learning and conflict biases," *Neuron*, vol. 47, no. 4, pp. 495–501, Aug. 2005.

[28] P. W. Ferrez and J. D. R. Milln, "You are wrong!automatic detection of interaction errors from brain waves," presented at the 19th Int. Joint Conf. Artificial Intelligence, Edinburgh, U.K., 2005.

[29] G. Schalk *et al.*, "EEG-based communication: presence of an error potential," *Clin. Neurophysiol.*, vol. 111, no. 12, pp. 2138–2144, Dec. 2000. [Online]. Available: http://www.clinph-journal.com/article/S1388-2457(00)00457-0/abstract

[30] M. Falkenstein *et al.*, "ERP components on reaction errors and their functional significance: A tutorial," *Biol. Psychol.*, vol. 51, no. 2–3, pp. 87–107, Jan. 2000.

[31] B. Cesqui *et al.*, "EMG-based pattern recognition approach in post stroke robot-aided rehabilitation: A feasibility study," *J. Neuro-Eng. Rehabil.*, vol. 10, no. 1, p. 75, Jul. 2013. [Online]. Available: http://www.jneuroengrehab.com/content/10/1/75/abstract

[32] S. F. Taylor *et al.*, "Neural systems for error monitoring: Recent findings and theoretical perspectives," *Neuroscientist*, vol. 13, no. 2, pp. 160–172, Apr. 2007.

[33] T. Blakely *et al.*, "Exploring the cortical dynamics of learning by leveraging BCI paradigms," in *Brain-Computer Interface Research* (Springer Briefs in Electrical and Computer Engineering), C. Guger, B. Z. Allison, and G. Edlinger, Eds. Berlin, Germany: Springer, 2013, pp. 53–60.

[34] R. L. White, "Object classification in astronomical images," in *Statistical Challenges in Modern Astronomy II*, G. J. Babu and E. D. Feigelson, Eds. New York, NY, USA: Springer, Jan. 1997, pp. 135–151.

[35] A. Bluman, *Elementary Statistics: A Step By Step Approach with Data CD and Formula Card*, 8th ed. New York, NY, USA: McGraw-Hill, Jan. 2011.

[36] N. F. Ince *et al.*, "High accuracy decoding of movement target direction in non-human primates based on common spatial patterns of local field potentials," *PLoS ONE*, vol. 5, no. 12, p. e14384, Dec. 2010. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0014384

[37] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximization for sparse Bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, C. M. Bishop and B. J. Frey, Eds., 2003. [Online]. Available: http://www.miketipping.com/papers/met-fastsbl.pdf

[38] V. Tadipatri *et al.*, "Robust movement direction decoders from local field potentials using spatio-temporal qualitative patterns," in *Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2012, pp. 4623–4626.

[39] G. BakIr, "Extension to kernel dependency estimation with applications to robotics," Max-Planck-Gesellschaft, Nov. 2005.

[40] J. Weston *et al.*, "Kernel dependency estimation," in *Advances Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA, USA: MIT Press, 2002, pp. 873–880. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2002.html

[41] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Int. Res.*, vol. 2, pp. 263–286, Jan. 1995. [Online]. Available: http://portal.acm.org/citation.cfm?id=1622826.1622834

[42] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[43] B. Blankertz *et al.*, "Classifying single trial EEG: Towards brain computer interfacing," in *Advances in Neural Information Processing Systems*, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.8038

[44] B. Blankertz *et al.*, "Optimizing spatial filters for robust EEG single-trial analysis," in *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 41–56, 2008.

[45] 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, p. e87253, Jan. 2014. [Online]. Available: http://dx.doi.org/10.13712Fjournal.pone.0087253

[46] P. 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.

Authors' photographs and biographies not available at the time of publication.