LONG-TERM MOVEMENT TRACKING FROM LOCAL FIELD POTENTIALS WITH AN ADAPTIVE OPEN-LOOP DECODER

Vijay Aditya Tadipatri †, *Ahmed H. Tewfik* †, *James Ashe*⋆

† The University of Texas, Austin, ⋆ University of Minnesota, Twin Cities

ABSTRACT

One of the challenges in using intra-cortical recordings like Local Field Potentials for Brain Computer Interface (BCI) is their inherent day-to-day variability and non-stationarity caused by subject motivation and learning. Practical Brain Computer Interfaces need to overcome these variations, as models trained on characteristic features from one day fail to represent new characteristics of another. This paper proposes a novel adaptive model that adjusts to signal variation by appending new features to the existing model and without knowledge of actual hand kinetics in an unsupervised way. With this adapting model we investigated the effects of learning and model adaptation on BCI performance. Using this new model we dramatically improve on all previously published long term decoding and show that target direction is accurately decoded in 95% of the trials over two weeks and in 85% of the trials in varying environments. Since the model needs no separate re-calibration, it can reduce user frustration and improve BCI experience.

Index Terms— Brain Computer Interface; Local Field Potentials; Adaptive Decoder

1. INTRODUCTION

Brain Computer Interfaces (BCI) decode behavioral signals like movement and imagination from neural data. A practical BCI requires an acquisition modality with high SNR; faithful reproduction of neural features, and optimal algorithms to translate the features to behavioral commands; and shorter training sessions [1, 2]. Advances in neural engineering and recording techniques extend the chronic recording envelope to multiple months. Recent studies recorded single unit activity (SUA) faithfully over 300 to 500 days in monkeys and local field potentials (LFP) over 1000 days in a tetraplegia patient [3, 4]. However, variations in subject motivation, behavior, and learning cause long-term non-stationarity and signal variability in these intra-cortical recordings [5]. While daily retraining and calibration can tackle the variability over time, such pauses increase user frustration and reduce the usability of BCI [1]. Other papers develop robust features to overcome the signal non-stationarity [6]; or look for patterns that recur consistently over time [7]. However these approaches fail to capture the long-term variability of the signals and as such do not adapt to any neuronal changes especially in varying environments. Recently, an adaptive closed-loop BCI successfully decoded arm movement over 1 year [8]. A closed-loop BCI allows subjects to learn and adapt to the decoding model and modulate the neural signals to suit the decoder [9, 10]. In this paper we propose a novel adaptive open-loop system that improves decoding; relieves the learning load from the user [11]; and advances the understanding of the learning mechanism. For this purpose, we decoded eight targets from neural data recorded from two monkeys on multiple days while they performed center-out reach tasks.

Kernel based methods are a class of supervised learning techniques that provide good approximations of targets by measuring the similarity between corresponding input vectors and a basis. Relevance Vector Machines (RVM) are such kernel based models trained in a Bayesian framework and provide probabilistic predictions of targets [12]. Their success arises from the sparse formulation to generalize training data as they work on the premise that only a few 'relevant' vectors describe most of the input space [12]. These methods require that the feature space is stationary and their performance deteriorates when the 'relevant' vectors can partially (or cannot) describe new instances of test input space. Such cases require model adaptation to include the new samples and their characteristics. While retraining the entire model is an option, it requires more computation and memory to store entire training data. This is not preferred in applications like Brain Computer Interfaces (BCI), where pausing for such an update increases user frustration [1]. This paper offers a solution in the form of an adaptive model that appends the existing model to generalize the recently observed samples.

The proposed model decodes target direction from LFP recorded from monkey brain by estimating the hand position and providing continuous decoding. Such an analysis is better suited than discrete target direction decoding, since it provides continuous control of more than one dimensions and generalizes well to novel targets [13]. Most regression techniques assume a single dimensional target vector and prescribe separate independent regressions to estimate each dimension in a multi-dimensional target. However, correlations (especially non-linear) in target dimensions result in spurious estimates of at least one dimension. To avoid this we propose to use the kernel dependency estimation (KDE) framework, which employs kernel functions to measure the similarity between target dimensions and a target basis, and encodes prior information about the targets in an elegant way [14]. Using the KDE framework the proposed model builds redundant RVM regressors to estimate multiple dimension target simultaneously and accurately.

The main contributions of this study are: 1. To imitate a practical BCI with minimal retraining; 2. Understand the long-term learning; and 3. Track changes in the motor cortex during learning. In our analysis we train baseline decoders on a single training session and adapt it to monitor the effects of learning. We overcome data variability by updating the decoder to track expected trajectories that trace a straight line from the center to the target and without prior knowledge of actual trajectories - mimicing a practical BCI. This paper proposes a novel decoder that learns and adapts to long-term variations of neural signals providing robust and consistent target decoding over two weeks and also in sessions with varying external forces against movement. Developing such an adaptive model also enables us to use the BCI system as a study tool to understand the dynamics of learning [15].

The rest of the paper is organized in the following way: Section 2 summarizes the neural and behavioral data; Section 3 discusses

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

the method and algorithms used for decoding; Section 4 discusses the results obtained by the method and Section 5 provides some concluding remarks and suggests related future work.

2. DATA

Two male rhesus monkey subjects (Macaca mulatta), H564 and H464, weighing 6.1 kg 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. LFP was recorded from two silicon-based electrode arrays (Cyberkinetics, Foxboro, MA) implanted in the contralateral arm areas of primary motor (M1) and dorsal premotor (PMd) cortices respectively. All the behavioral data (position, velocity, forces, and torques) and event markers were also collected. Once the monkey got familiar with the center-out tasks, external forces against the direction of movement were applied in some sessions viz. Stiff Clock-wise (SCW), Viscous Clock-wise (VCW) and Viscous Counterclockwise (VCCW). During our initial analysis, we observed that band-pass filtering the signal in the δ - band (0.4 - 4Hz) obtained the best decoding result. Hence, we calculate qualitative features in the form of instantaneous inter-channel power ranks on the data filtered in this band [16].

3. METHODS

The proposed algorithm uses redundant non-linear regression models to obtain hand trajectories and decode movement direction. The proposed adaptation needs little feedback on the actual movement as it uses a straight line to approximate the intended trajectory. This section briefly discusses the initial training and adaptation methods to obtain the trajectory estimates.

3.1. Relevance Vector Machine

Consider a training data $\{\mathbf{X_i}, \mathbf{t_i}\}$ where $\mathbf{X_i}$ is the neural feature vector and $\mathbf{t_i}$ is the corresponding target. As mentioned above, RVM is a set of general models of the form in eq 1

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

where $\mathbf{Y}_i$ are the chosen basis vectors and w_i are the corresponding weights and $\mathbf{\Phi}(.)$ is suitable a kernel function. Usually, the chosen basis vectors $\mathbf{Y}_i$ are a set of prototypical examples from the input training vectors. Maximum likelihood estimation of $\mathbf{w}$ without any constraints leads to over-fitting on the training set and to a generalized model requires a sparse $\mathbf{w}$ [12]. 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 the 'relevant' vectors leads to the best subset of input feature vectors that can represent the input space. 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 [17]. In this paper, we choose Gaussian radial basis functions with basis width σ of the form:

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

The basis width provides a handle on the support of a given basis vector. The model can be described by the kernel function parameters, basis vectors and the corresponding $\mathbf{w}$ as $\mathcal{M} := \{\mathbf{\Phi}(.), \mathbf{X}, \mathbf{w}\}$.

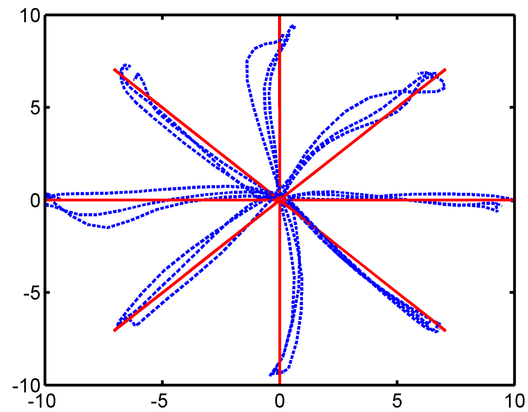

Fig. 1. Actual ($\mathbf{t}$, dashed blue), and the used intended movement ($\tilde{\mathbf{t}}$, solid red) of example trials overlaid on a 10cm $\times$ 10cm workspace.

3.2. Adaptation

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

$$\hat{y}_{new} = \sum \mathbf{w}_b^T \mathbf{\Phi}(\mathbf{X}_{new}, \mathbf{X}_b) \tag{3}$$

This initial model provides good generalization if the basis vectors can express the entire input space. Since the neural recordings and their features change due to learning and other environmental conditions, they will deviate from the constructed model. In a typical BCI, the new neural features tend to misalign with the existing basis vectors $\mathbf{X}_b$, warranting model updates to adapt to new data.

If the target y_{new} corresponding to $\mathbf{X}_{new}$ were available, adaptation only requires a corrective action that provides a good estimate for the residual trajectory:

$$\begin{gathered} e = y_{new} - \hat{y}_{new} \\ \mathbf{w}_u : \min ||e - \mathbf{w}_u^T \mathbf{\Phi}(\mathbf{X}_{new}, \mathbf{X}_{new})|| + \lambda ||\mathbf{\Phi}(\mathbf{X}_b, \mathbf{X}_{new})|| \\ \mathcal{M}_u := \{\mathbf{\Phi}(.), \mathbf{X_{new}}, \mathbf{w}_u\} \end{gathered} \tag{4}$$

The first part of equation 4 can be estimated using the RVM learning algorithm with similar constraints shown above in section 3.1. The constraint on the included basis ensures that the baseline model remains unaffected with the update ($\mathbf{w}_b$ needs no update). In the absence of such a constraint, the fit on the training data would suffer due to the addition of new basis.

Our key innovation is to mimic a practical BCI and without a prior knowledge of the actual hand trajectories y_{new}. Therefore, we approximate y_{new} by incorporating general principles of natural movements under the assumption that the monkey intends to reach the target in a straight line. Thus, we construct an intended trajectory as $\tilde{y} = \mathcal{F}(\hat{\theta})$ as shown in Figure 1. The overall model 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 model before the update. Since the current model structure is linear in the kernel function space we can obtain the updated model as follows:

$$\mathcal{M}_* = \{\mathbf{w}_0 \cup \mathbf{w}_u, \mathbf{\Phi}(.), \mathbf{X}_0 \cup \mathbf{X}_u\} \tag{5}$$

$$\hat{y}_*(\mathbf{X}_*) = \sum \mathbf{w}_0^T \mathbf{\Phi}(\mathbf{X}_*, \mathbf{X}_0) + \sum \mathbf{w}_u^T \mathbf{\Phi}(\mathbf{X}_*, \mathbf{X}_u) \tag{6}$$

This approach eliminates the need for a daily calibration session and can perfrom online updates without interrupting the user.

3.3. Multiple-output regression

The approach in this paper decodes the target direction by estimating the hand trajectory and then computing the angle of the trajectory. It estimates multiple trajectory parameters like x-position, y-position and absolute movement towards target. We observed that the estimated absolute position (direction independent) correlated well with the actual value (>0.92) than the independent estimates of x- and y- positions. In general, the model $\mathcal{M}$ learned in the RVM framework predicts a single dimensional output. One approach recommends a different independent model for each dimension of target vector. While such models provide a good correlation of individual dimensions when they are independent, provide spurious results when dependence exists. The technique proposed here is based on Kernel Dependence Estimation (KDE) to take advantage of such dependence and obtain a better overall performance [14]. The technique uses a kernel function to reflect the non-linear dependence of the target dimensions. Each basis vector of this kernel represents a unique point in the target space as shown by eq (7) below:

$$\mathbf{\Psi}(\mathbf{y}, \mathbf{y}_i) = \exp(-(\mathbf{y} - \mathbf{y}_i)\Sigma_y^{-1}(\mathbf{y} - \mathbf{y}_i)^T) \tag{7}$$

The above equation is a Gaussian kernel with a spatial covariance Σ_y evaluated at each point on the target space, where $\mathbf{y}_i$ denotes the hand movement space in the form of its horizontal, vertical and absolute positions :$\{y_x, y_y, y_r\}$, and $y_r = \sqrt{(y_x^2 + y_y^2)}$. Such a representation allows decomposing the obtained basis into its independent singular vectors, and approximating them individually in an RVM framework as shown in the algorithm. This results in multiple redundant approximations and provides high correlation in all the target dimensions. The next section discusses the results obtained during our analysis.

Algorithm

Training Stage: Learn $\mathcal{M}$ from Data:$\{\mathbf{X}_i, \mathbf{t}_i\}_{i=1}^N$
Build Input Kernel : $\mathbf{\Phi}(\mathbf{X_i}, \mathbf{X_i})$ using eq (2)
Build Output Kernel : $\mathbf{\Psi}(\mathbf{t_i}, \mathbf{y_j})$ using eq (7)
$\mathbf{\Psi} = USV^T$
for each column k of U **do**
    Estimate $\mathbf{w}_k$: $U_k = \sum_{i=1}^N w_{ki}\Phi(X) + w_{k0}$
end for
Basis Vectors : $\mathbf{D} \leftarrow \mathbf{X}$
Store Model: $\mathcal{M}_0 \leftarrow \{S, V, \mathbf{D}, \mathbf{w}\}$

Initialize $\mathcal{M}^* \leftarrow \mathcal{M}_0$
Testing Stage: Estimate $\hat{\mathbf{t}}_i$ from Data:$\{\mathbf{X}_i^*\}_{i=1}^N, \mathcal{M}^*$
Build Input Kernel : $\mathbf{\Phi}(\mathbf{X_i^*}, \mathbf{D_j})$ using eq (2)
for each column k of U **do**
    Calculate $\hat{U_k} = \sum_{j=1}^{ND} w_{kj}\Phi_j(X) + w_{k0}$
end for
$\hat{\mathbf{\Psi}} = \hat{U}SV^T$
$\hat{\mathbf{t}}_i = \max_i \hat{\mathbf{\Psi}}_i$, $\hat{\theta}_i = \arctan\frac{t_y}{t_x}$

Adaptation Stage: Update Model $\mathcal{M}^*$ after L trials using $\{\mathbf{X}_i^*, \hat{\theta}_i^*\}_{i=1}^L$
$\tilde{\mathbf{t}} = \mathcal{F}(\hat{\theta}_i)$
$\tilde{\Psi} = \mathbf{\Psi}(\hat{\mathbf{t}}_\mathbf{i}, \mathbf{y_j})$ using eq (7)
$\tilde{U} = \tilde{\mathbf{\Psi}}VS^{-1}$
for each column k of U **do**
    Update $\mathbf{w}_k^u$: $\tilde{U}_k - \hat{U}_k = \sum_{i=1}^N w_{ki}^u\mathbf{\Phi}(\mathbf{X}_i^*) + w_{k0}^u$ using eq (4)
end for
Basis Vectors : $\mathbf{D}^u \leftarrow \mathbf{X}^*$
$\mathcal{M}^* \leftarrow \{S, V, \mathbf{D}|\mathbf{D}^u, \mathbf{w}|\mathbf{w}^u\}$

4. RESULTS AND DISCUSSION

To understand the long term decoding and movement tracking capabilities of the decoder, we train an initial model on the first session and evaluate the model adaptation on successive sessions. The performance is evaluated in terms of decoding accuracy: fraction of trials that the model accurately predicted the intended direction of movement. Note that the subjects performed the task in an open loop and could not actively learn the model. The model is adapted to account for the varying signal characteristics after decoding 25 trials and selecting only successfully predicted trials. One could also update the model after every successful trial - oversensitive update. While such an update presents a new model at every trial, it also requires an additional processing time (to update the model) at the end of every trial. Conversely, update after a large number of trials - passive model - might not track the fluctuations fast enough. Thus, the update process must choose an optimal number of trials to update the model. In the current data, the decoding performance varied little (<2%) with different number of adaptation trials. Figure 2 shows the stable performance of the adaptation algorithm over multiple blocks and over 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).

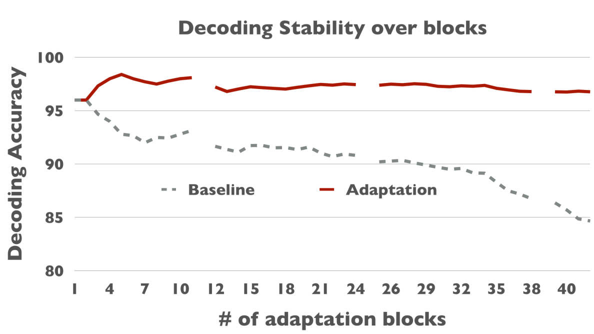

Fig. 2. Cumulative decoding accuracy with and without adaptation across multiple adaptation blocks of 25 trials each. The gaps in the curves represent the end of day.

The update process involves obtaining new basis vectors that fit the errors from the original model increasing the number of basis vectors in the updated model. This process includes all estimates despite their proximity to the intended trajectory. However, we can improve the computational performance (reduce number of basis vectors) by selecting only those trajectories, where estimated value ($\hat{t}$) deviates more than a threshold from the expected trajectory ($\tilde{t}$). By allowing a deviation of 1 cm between the expected and intended trajectories the number of basis vectors drastically reduces without impacting the decoding accuracy.

Table 1 compares the result of adaptation to the decoding performance over the decoder age. To observe the long-term effects of adaptation, we adapted one model continuously over the two weeks of test data. Another model adapted only the current test session and

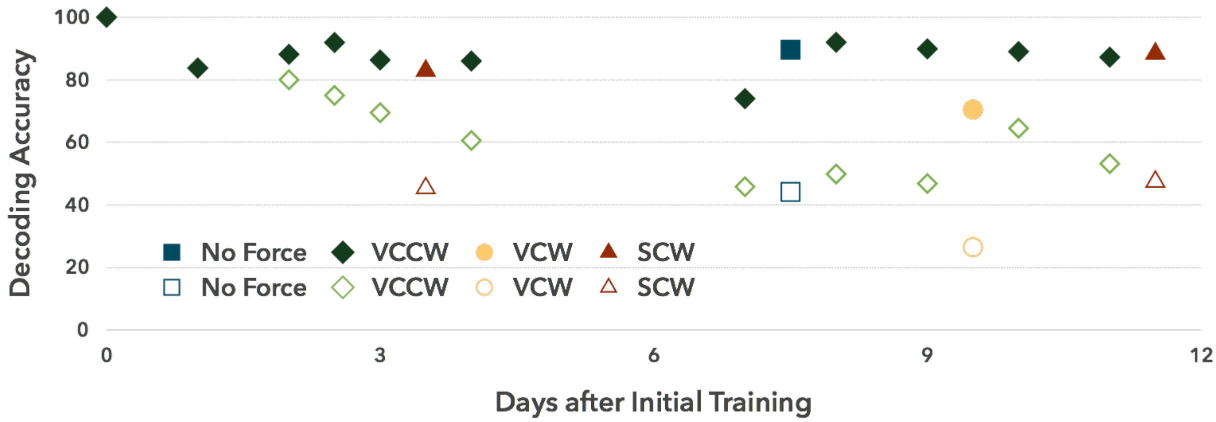

Fig. 3. Decoding Accuracy in sessions with varying field forces. The filled icons represent the accuracy with adapting model and the unfilled icons represent the accuracy of the baseline model. For ease of reading, different field forces are represented with different shapes.

Table 1. Decoding Accuracy (in %) across decoder age.

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

ignored any previous adaptation by resetting the model to $\mathcal{M}_0$ at the beginning of each day. We observe that adaptation improves the target decoding accuracy over the two weeks. When the model was not adapted, the accuracy drastically fell around day 14, but adaptation stabilized the accuracy over 95%. While model 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. These results show that learning modulates the neural activity continuously (rather than daily) and decoders benefit from the adaptation to variations introduced by this learning.

Table 2 presents results from (while not exhaustive) a representative literature that decode movement direction from LFP. A direct comparison of results is difficult as these papers use different recording protocols and use varying classification methods. However, these studies used cross-validation to obtain the decoding power and ignore any non-stationarity between training and testing samples. A closer comparison can be made with results from [8], which shows long-term decoding in an online setting with an accuracy of 80%.

To investigate the effect of adaptation on changing environments, we applied a similar strategy on sessions where external field forces against movement were applied. Even for this experiment, the initial model was initially trained on a session (where a field force VCCW was applied) and the updated on successive sessions. The model here needs to tackle both variations due to learning over time and due to varying external field forces. The results in figure 3 show that the model is robust to both and achieves an average decoding of 85% on sessions even with different external field forces. Prominent

Table 2. A comparison of literature using decoding accuracy.

Algorithm	Decoding Accuracy
Bayesian Decoding, SVM [4]	40%
Directional Tuning [18]	50%
Bayesian Classification [19]	81%
Proposed Method	95%

results occur when 1. there is latency between sessions (between days 4 and 7) and 2. novel field force (like VCW on day 10) are applied. In these sessions the decoding is boosted with adaptation (especially on day 10). These results motivate the use of LFP for practical BCI even under varying environment conditions.

5. CONCLUSIONS

This paper proposes an adaptive modeling technique that updates the existing model to account for the variation in data characteristics. We show that such an adaptation of open-loop neural decoders improves their performance. Straight line estimates of hand kinetics enabled efficient update of the decoder without knowledge of actual kinetics. In this paper, we used a kernel dependency estimation framework to obtain redundant approximations of multi-dimensional hand kinetics from intra-cortical LFP and decoded the hand movement direction. We showed that an adaptive open-loop algorithm sustains the decoding accuracy above 95% over two weeks while an algorithm of the same framework without adaptation resulted in only 66%. These results also show that learning is a continuous process and prototypical examples from the learning stage aid in the accurate decoding of target direction.

The proposed algorithm appends new relevant feature vectors to the existing model increasing model size with each update. We expect that over time some vectors in the model might have little to no impact on the new observations. Pruning the model to remove such vectors might result in better computational performance. A closed-loop BCI would also benefit from adopting this algorithm. Especially, if the user produces consistent neural features, it reduces the need to update the model and also sustain consistent performance.

6. REFERENCES

[1] A. Nijholt and D. Tan, "Brain-computer interfacing for intelligent systems," *Intelligent Systems, IEEE*, vol. 23, no. 3, pp. 72–79, 2008.

[2] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan, "Braincomputer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, June 2002.

[3] Adam S. Dickey, Aaron Suminski, Yali Amit, and Nicholas G. Hatsopoulos, "Single-unit stability using chronically implanted multielectrode arrays," *Journal of Neurophysiology*, vol. 102, no. 2, pp. 1331–1339, Aug. 2009, PMID: 19535480 PMCID: PMC2724357.

[4] Carsten Mehring, Jörn Rickert, Eilon Vaadia, Simone Cardosa de Oliveira, Ad Aertsen, and Stefan Rotter, "Inference of hand movements from local field potentials in monkey motor cortex.," *Nature neuroscience*, vol. 6, no. 12, pp. 1253–1254, Dec. 2003.

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

[6] Vijay Aditya Tadipatri, Ahmed H. Tewfik, Vikrham B. Gowreesunker, J. Ashe, G. Pellizer, and R. Gupta, "Time robust movement direction decoding in local field potentials using channel ranking," in *Engineering in Medicine and Biology Magazine*. IEEE, 2010, vol. 29.

[7] B. Gowreesunker, A. Tewfik, Vijay Aditya Tadipatri, J. Ashe, G. Pellizzer, and R. Gupta, "A subspace approach to learning recurrent features from brain activity," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. PP, no. 99, pp. 1, 2011.

[8] Robert D Flint, Zachary A Wright, Michael R Scheid, and Marc W Slutzky, "Long term, stable brain machine interface performance using local field potentials and multiunit spikes," *Journal of Neural Engineering*, vol. 10, no. 5, pp. 056005, 2013.

[9] A.L. Orsborn, S. Dangi, H.G. Moorman, and J.M. Carmena, "Closed-loop decoder adaptation on intermediate time-scales facilitates rapid bmi performance improvements independent of decoder initialization conditions," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 20, no. 4, pp. 468–477, 2012.

[10] Beata Jarosiewicz, Nicolas Y Masse, Daniel Bacher, Sydney S Cash, Emad Eskandar, Gerhard Friehs, John P Donoghue, and Leigh R Hochberg, "Advantages of closed-loop calibration in intracortical braincomputer interfaces for people with tetraplegia," *Journal of Neural Engineering*, vol. 10, no. 4, pp. 046012, 2013.

[11] Benjamin Blankertz, Motoaki Kawanabe Ryota Tomioka, Friederike U. Hohlefeld, Vadim Nikulin, and Klaus-robert Müller, "Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing," in *In Ad. in NIPS 20*, 2008, vol. 20.

[12] Michael E. Tipping, "Sparse bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sept. 2001.

[13] D.J. McFarland and J.R. Wolpaw, "Sensorimotor rhythm-based brain-computer interface (bci): feature selection by regression improves performance," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13, no. 3, pp. 372–379, 2005.

[14] Jason Weston, Olivier Chapelle, Andr Elisseeff, Bernhard Schlkopf, and Vladimir Vapnik, "Kernel dependency estimation.," in *NIPS*, Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, Eds. 2002, pp. 873–880, MIT Press.

[15] Tim Blakely, Kai Miller, Jeffrey Ojemann, and Rajesh Rao, "Exploring the cortical dynamics of learning by leveraging bci paradigms," in *Brain-Computer Interface Research*, Christoph Guger, Brendan Z. Allison, and Gnter Edlinger, Eds., SpringerBriefs in Electrical and Computer Engineering, pp. 53–60. Springer Berlin Heidelberg, 2013.

[16] V.A. Tadipatri, A.H. Tewfik, J. Ashe, and G. Pellizzer, "Robust movement direction decoders from local field potentials using spatio-temporal qualitative patterns," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, 2012, pp. 4623–4626.

[17] 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.

[18] John G. O'Leary and Nicholas 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, Sept. 2006.

[19] Hansjörg Scherberger, Murray R. Jarvis, and Richard A. Andersen, "Cortical local field potential encodes movement intentions in the posterior parietal cortex.," *Neuron*, vol. 46, no. 2, pp. 347–354, Apr. 2005.