

# Strategies for the quick calibration of sensor-driven systems with low computational overhead

**Keywords:** BCI, Mobile, Ubiquitous Computing

**Abstract:** Physiological computing applications rely on time series data from sensors, which vary widely between users and over time. Application designers face a tradeoff between computational complexity, calibration time, and classification accuracy. We describe a novel quantization technique for EEG signals, which we find increases the computational speed associated with training a machine-learning based classifier without a significant detriment to system accuracy. We test this technique on a brain-computer interface (BCI), and find that an opportunistic calibration strategy can achieve acceptable accuracy in 86.6% of subjects with under five minutes of training. We discuss implications for the design of consumer-ready BCI and other physiological computing applications.

## 1 INTRODUCTION

Physiological data do not carry universal meanings. While the movement of a mouse can be straightforwardly mapped to the position of a cursor, the expression of biosignals vary widely between people, and often change within individuals over time. Brain-computer interface (BCI) serves as a dramatic example of this phenomenon: the nonstationary nature of neural signals necessitate regular calibration and recalibration within users as well. (Dornhege, 2007; McFarland and Wolpaw, 2011).

Supervised learning algorithms have enabled systems that adapt to users' physiological signals after a calibration period. In BCI, this approach has yielded proof-of-concept systems ranging from brain-controlled keyboards and wheelchairs to prosthetic arms and hands (Blankertz et al., 2007; Milln et al., 2010; D. Mattia, 2011; Hill et al., 2014; Campbell et al., 2010).

Compared to these lab-based prototypes, systems suitable for mobile, real-world use will have fewer sensors (due to ergonomic constraints on device size) and will process noisier signals (as data acquisition will occur while people are moving, walking, talking, and so on). As an added challenge, computational complexity (measured by both storage and processing power) may be limited by the mobile & wearable computing architectures on which these systems will most likely be deployed.

In this study, we use recordings from a single, dry electroencephalographic (EEG) sensor to simulate the calibration of a binary BCI, and investigate the effect of a novel signal extraction technique on

the system's computational performance and accuracy. We find that our signal extraction technique increases the computational speed of a classification-based BCI 450% without a significant detriment to accuracy. This technique allows us to calibrate 86.6% of users to a threshold of BCI control in under five minutes of training data, and to calibrate 100% of users in under 15 minutes.

## 2 RELATED WORK

### 2.1 Calibrating EEG-based BCI

BCI systems generally aim to recognize a user's mental gestures as one of a finite set of discrete symbols, a problem that can be thought of as a pattern recognition task (Lotte et al., 2007). The difficulty of this task stems primarily from the variable and non-stationary nature of neural signals: the "symbols" we wish to identify are expressed differently between individuals, and even vary within individuals from trial to trial (Vidaurre et al., 2006; Vidaurre et al., 2011b).

In order to compensate for variability in BCI signals, recent work has leveraged adaptive classification algorithms to distinguish between mental gestures. (Lotte et al., 2007; Vidaurre et al., 2011b) Automated calibration procedures have turned BCI novices into competent users over the course of hours instead of days or weeks, and without manual calibration by a researcher (Vidaurre et al., 2006; Vidaurre et al., 2011a; Vidaurre et al., 2011b). During calibration, users perform "labeled" (that is, known) mental gestures in order to produce samples for the classifier.

Meanwhile, the classifier attempts to determine which features of the data are most informative.

## 2.2 Statistical signal processing in EEG-based BCI

Calibrating a BCI requires an algorithm that can adapt to its inputs. Support vector machines (SVM) are a set of supervised machine learning methods that take labeled example data to create a model that can be used to predict the classes of unlabeled data. SVMs use a hyperplane (an  $n$ -dimensional construct in  $n+1$  dimensional space) to draw discriminatory boundaries between classes.

Past work has used linear SVMs in BCI applications with great success. (Garrett et al., 2003; Grierson and Kiefer, 2011) In contrast to linear discriminant analysis, which also have a long history of use in BCIs, SVMs select the hyperplane that maximizes distance from the nearest training points, which has been shown to increase the model's generalizability (Burges, 1998).

In classification algorithms generally, larger feature vectors require that an exponential increase in the amount of data needed to describe classes, a property known as "the curse of dimensionality" (Jain et al., 2000; Raudys and Jain, 1991). Traditionally, BCI applications rely on dense, high-dimensional feature vectors produced by multi-electrode scanning caps with high temporal resolution, so dimensionality represents a major bottleneck in training classification algorithms. This bottleneck threatens the responsiveness of BCI from a user experience standpoint and places high requirements on end users' hardware.

## 2.3 Brain-computer interface "in the wild"

Many inexpensive, comfortable EEG devices have come to market, most of which use "dry" electrodes that do not require special gels. Compared to their lab-based counterparts, these devices have many fewer electrodes, thus limited spatial resolution, and produce significantly noisier signals. (De Vos and Debener, 2014) Regardless, past work has demonstrated several mobile-ready BCI systems that use these scanning devices, and the Neurosky MindSet in particular (the headset used in this study - a single, dry EEG electrode placed roughly at FP2, which connects wirelessly to phones and computers, and sells for roughly 100USD) has been used to successfully detect emotional states, event-related potentials (ERP), and to employ brain-based biometric authentication

(Crowley et al., 2010; Grierson and Kiefer, 2011; Chuang et al., 2013).

However, the use of consumer EEGs for the direct, real-time control of software interfaces has proven more difficult, as the number of electrodes on these headsets limit the spatial resolution required to discriminate between mental gestures. (Carrino et al., 2012; Larsen and Hokl, 2011). We expect significant improvements from consumer-grade EEG devices in the near future, with more sensors and better signal quality (e.g. Interaxon Muse, Melon headband, Emotiv Insight); however, we expect the signal from these devices will remain noisier than lab-based counterparts, as people will be wearing and using them while moving, and in uncontrolled environments with ambient electromagnetic signals interfering with endogenous biosignals. x

To transition BCI from the lab into naturalistic environments, we must squeeze more signal out of fewer, and less reliable, sensors. Furthermore, since BCIs are envisioned largely as always-available input devices, they will likely be deployed on mobile processors and perhaps even embedded processing systems; our computational resources may be more similar to that of a smartphone than of a desktop workstation, and it is feasible that we may need to do some processing "in the cloud" (i.e., on a more powerful server to which the client sends data over the network, similar to the way Apple's Siri processes voice data). For effective BCI to occur in these environments, we must extract signal in a maximally efficient way so as to limit our computational footprint, and perhaps even to minimize the size of data if we wish to ship it to an external server.

## 2.4 Online, co-adaptive calibration

# 3 METHOD

This section describes the signal extraction technique used in this study evaluates and discusses the supervised machine learning classifier used to build our binary BCI.

## 3.1 Signal extraction

### 3.1.1 Compressing power spectra in the temporal dimension

First, we compute an average of all the power spectra associated with a recording. We obtain a discrete probability density function (PDF) in which each bin

is the mean of its corresponding bins through time. At this stage, we have a discrete PDF of 1024 bins for the entire  $n$  second recording.

### 3.1.2 Logarithmic binning

Binning a probability density function (PDF) is a simple way to “quantize” the information contained in the full signal. By taking the mean of several adjacent points in the PDF, we are left with a single bin that compresses the information contained in its local area of frequencies. For instance, four contiguous frequencies (1-1.25, 1.25-1.5, 1.5-1.75, 1.75-2) of the values (4, 4.5, 5) could be combined into a single bin with the value 4.5.

Since EEG activity is associated with frequencies from 1-40Hz, we presume this range contains the majority of relevant signal. However, we do not rule out the possibility that useful signal exists in other frequency ranges (muscular activity, for example, might be correlated with mental gestures in some cases). In order to exploit the entire frequency spectrum while preserving our bias toward known sources of useful signal, we logarithmically space bins through the PDF, as shown in 1.

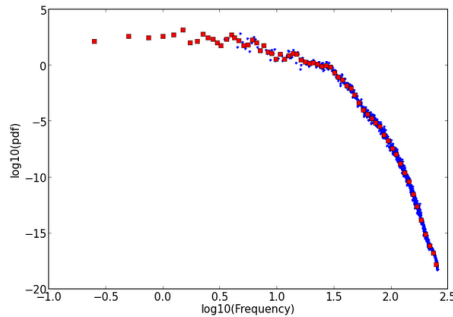


Figure 1: In double logarithmic scale, the original 1024 bins (blue dots) of the PDF obtained from averaging the  $n$  power spectra of one recording, and the resulting “quantized” PDF with a resolution of 100 log-bins. The quantized PDF preserves very well the structure of the original, 1024-point PDF.

In summary, we build a probability density function of frequencies captured by the EEG scanning device from all the power spectra in a recording. We then use logarithmically-spaced bins to reduce the original 1024 frequency values to a smaller number of bins (e.g., 100 log-bins in 1). This method produces a statistical average of a time series, compressed into a single feature vector that it is easy to use in a classifier.

## 3.2 Classifying EEG signals

In this study, we build a binary BCI using a support vector machine (SVM) classifier, which we train individually on each subject’s recordings.

### 3.2.1 LibLinear

We use LinearSVC, (Fan et al., 2008) a wrapper for LibLinear exposed in Python through the ScikitLearn library. (Pedregosa et al., 2011) We chose LinearSVC because BCI classification problems are generally presumed to be linear (Garrett et al., 2003; Lotte et al., 2007), and because LibLinear’s underlying C implementation boasts among the fastest train- and test-time performance among state-of-the-art solutions. (Fan et al., 2008) For the SVM’s parameters, we use a squared hinge loss function, which maximizes the margin for greater generalizability, and a hyperparameter of 100, found through a “grid search”, or an exhaustive search through a randomly selected sample of our dataset.

### 3.2.2 Cross-validation

When training a classifier, it is important to make good estimates of the classifier’s performance. A common way to do this is through *cross-validation*, in which we train and test a classifier several times using different subsets of the data for training and testing. We use ScikitLearn’s built-in cross-validation toolkit, which performs seven cross-validation steps using different splits of data in each round.

## 4 DATASET

The data used in this experiment were taken from a previous study. (Chuang et al., 2013) The anonymized dataset consists of power spectrum time series data recorded by the software from the Neurosky MindSet headset from 15 subjects, students at UC Berkeley, performing mental tasks.

The seven mental tasks were: focusing on breathing; imagining moving one’s right index finger; imagining moving one’s body to repeatedly perform a sports-related movement of the subject’s choice; imagining singing a song or reciting a passage; listening to a tone with eyes closed; choosing a color (red; green; yellow or blue) and counting how many times one’s chosen color appears on a screen; choosing any thought to use as a “password”.

Participants performed each of the seven mental tasks, enumerated below, ten times. Each of the ten trials lasted ten seconds.

The Neurosky MindSet SDK delivered a power spectrum of its data every half second. The power spectra that the SDK delivers are computed with discrete bins of 1/4 Hz. Each bin represents the intensity of activation of a frequency range (e.g., between 1 and 1.25 Hz) in a half-second time window. There are therefore 1024 values reported for one power spectrum. Since our mental task recordings are 10 seconds long, each recording is represented by twenty power spectra on average.

The dataset was further cleaned by removing all readings marked as suboptimal signal quality by the Neurosky SDK. The SDK delivers a signal quality value that is greater than zero when signal quality is suboptimal. Factors causing this value to be greater than zero include lack of contact between electrode and skin, excessive non-EEG noise (EKG, EMG, EOG, electrostatic) and excessive motion.

## 5 EXPERIMENTS

### 5.1 Finding a compromise between computational speed and classifier accuracy

Generally, we seek to maximize our system's classification accuracy while minimizing its computational expense. One way to reduce the computational requirements of a SVM classifier is to reduce the size of the feature vectors on which it is trained and tested. Our signal quantization method allows us to directly adjust the size of feature vectors by changing the signal's resolution (see 3.1), though lowering the resolution of feature vectors could negatively effect the classifier's performance.

In this experiment, we examine the effect of resolution, operationalized by the number of bins used in the quantization step, on our BCI's performance, operationalized by the SVM's training time and by the SVM's estimated accuracy. We hypothesize that SVM accuracy will decrease with resolution, as will SVM training time.

#### 5.1.1 Protocol

For each subject, we generate every pair of two tasks and cross-validate our SVM seven times on recordings for those two tasks. We vary the resolution of the samples we feed to the SVM. For every task pair processed, we record mean classification accuracy across all rounds of cross-validation.

As an additional performance audit, we measure the time needed to fit an SVM to the data for two

randomly selected taskpairs across all subjects. We repeat this process ten thousand times at different resolutions, collecting the minimum time observed in each series of attempts.

### 5.1.2 Results

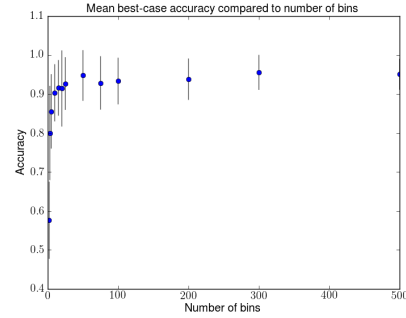


Figure 2: Mean best-case accuracy among all subjects compared to time needed to train the classifier. At resolutions of 100 points and greater, we find no evidence of an increase in classification accuracy.

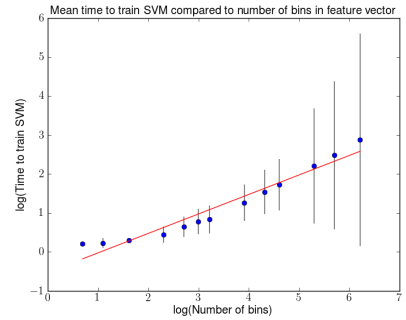


Figure 3: Log of mean classifier training time compared to log of data resolution. The time needed to train the classifier increases logarithmically with resolution.

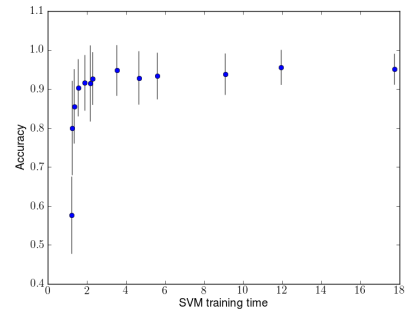


Figure 4: Best-case accuracy compared to the time needed to train the classifier. By decreasing the number of bins in the EEG data, we can decrease the time needed to train the support vector machine up to nine times without without significant detriment to classifier accuracy.

Resolution was positively correlated with classifier accuracy (slope = .0013 R-squared = .773,  $p < .001$ ). Resolution was also positively correlated with time to train classifier (slope = 0.5 R-squared = .947,  $p < .001$ ). We compare accuracy and SVM training time directly in 4.

We find support for our hypothesis that computational requirements (SVM training time) decreases along with resolution of the signal. We find partial support for the hypothesis that SVM accuracy decreases along with signal: this effect appears at resolutions lower than 100 points, but we find no significant increase in SVM accuracy at higher resolutions.

Overall, we find that relatively small feature vectors (75-100 bins) produced with our method yield classifiers as accurate as those that use much higher resolution samples, and that reducing vector size in this way can dramatically increase the computational speed of training an SVM.

## 5.2 Opportunistic strategy for calibrating a binary BCI

In the previous experiment, we found that our compression technique can speed up an SVM classifier without significant detriment to BCI accuracy. However, for the technique to be useful in real-world applications, it must also allow users to quickly calibrate the system to their personal physiological signals.

In this experiment, we evaluate an opportunistic strategy for user calibration. Using a resolution of 100 points identified as optimal in the previous experiment, we measure user calibration time (the time it takes a user to achieve a threshold accuracy with the BCI) and the classification accuracy each user achieves after calibration. We hypothesize that this technique will allow for faster calibration than an exhaustive search (720 seconds) while maintaining sufficient system accuracy across users.

### 5.2.1 Protocol

For each subject, we spliced all mental task recordings into 1/2-second chunks, each one representing a single power spectrum reading from our headset. By sampling from our dataset, we simulate recording a sixty second example of mental tasks from each user. We begin with examples of the three tasks most commonly associated with best-case performance (base, pass, color) for an initial user calibration time of 120 seconds. We then perform a seven-fold cross-validation on every permutation of two of these tasks (base versus task, pass versus task, pass versus color, etc).

Task	Freq Bestcase
Color	10
Base	5
Pass	4
Sport	3
Finger	2
Song	2
Eye	2

Table 1: An exhaustive search of SVM accuracy on taskpairs identified each subject’s best-performing taskpair. We recorded the frequency of each task’s occurrence in a best-case taskpair, shown here. We used these datato inform the order in which our opportunistic calibration strategy would prompt the user to record tasks.

The taskpair with the highest mean score across cross-validation rounds is selected for an additional testing session, in which the remaining 80 seconds of recordings for both tasks are used to generate an estimate of the classifier’s accuracy on new EEG signals.

If the score on this additional testing round is below 75% (a threshold for BCI literacy) (Vidaurre and Blankertz, 2010), the user records sixty seconds of the taskpair next most correlated with bestcase accuracy across users. We repeat the above process on unexplored taskpairs repeated until a taskpair achieves over 75% accuracy on post-calibration data, or until all taskpairs have been evaluated.

### 5.2.2 Results

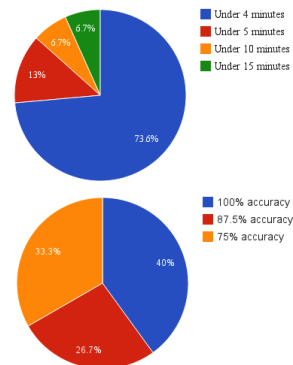


Figure 5: Calibration time across subjects (top) and classifier accuracy (bottom). The vast majority of subjects achieve acceptable accuracy in under five minutes of training, and all subjects achieve BCI literacy in under 15 minutes.

We find that our strategy calibrates users to BCI control more quickly than an exhaustive search.

## 6 DISCUSSION

In this study, we investigated the effect of a signal quantization technique on the performance of a binary BCI that used a low-cost, single-channel EEG headset as input. We found that our technique allowed for a computationally efficient BCI that achieves decent simulated accuracy for all users in our dataset and boasts quick user calibration times overall.

Specifically, we find that our quantization method decreased the computational expense of EEG-based calibration (18 ms to 2ms for SVM traintime) without a significant detriment to accuracy and, using quantized data, our opportunistic user calibration strategy achieved an average of 88.3% accuracy across all subjects. All subjects required under fifteen minutes of calibration time, and 86.6% of these subjects required five minutes or fewer.

## 7 CONCLUSIONS & FUTURE WORK

The conclusions to be drawn from this study are limited in a few regards. First, calibration and classification were performed offline, so factors involving the user interface (such as feedback) are not taken into account. We cannot be sure, for instance, that our findings with short splices of ten-recordings data will persist when a system solicits recordings of only a second or under. Furthermore, a few of our tasks (e.g. the color task) relied on exogenous stimuli, which may be impractical in naturalistic settings for ergonomic reasons.

Our study implies that practical BCI can be achieved with as few as one, inexpensive EEG sensor, minimal processing power and a only a few minutes of user calibration. Future work could build usable, online BCI systems to test this claim more rigorously, especially in mobile and out-of-lab environments.

Since many types of physiological data can be represented as power spectra (electrocardiography, electromyography), future work could test our quantization technique in other classification-based physiological applications (heart sensing, gesture recognition, or systems with heterogeneous sensors).

Since logarithmic binning dramatically decreases the size of physiological data in memory, this technique could allow developers to more easily ship biosignals to remote servers. Future work could explore the design space associated with the storage and transmission of physiological signals. BCI calibration, for example, could occur in the cloud - the client would quantize power spectra data and quantize them

using our method, then ship these compressed data to a more powerful server.

Alternatively, the small size of quantized feature vectors could enable long-term, pervasive recording of mental states. The data would be small enough to ship and store on a centralized server, or to store locally in a decentralized fashion. Monitoring mental (or other biophysical) activity continuously in everyday settings could yield observations about human activity & physiology that would be difficult to observe in controlled, laboratory environments.

## REFERENCES

- Blankertz, B., Krauledat, M., Dornhege, G., Williamson, J., Murray-Smith, R., and Mller, K.-R. (2007). A note on brain actuated spelling with the berlin brain-computer interface. In Stephanidis, C., editor, *Universal Access in Human-Computer Interaction. Ambient Interaction*, number 4555 in Lecture Notes in Computer Science, pages 759–768. Springer Berlin Heidelberg.
- Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.
- Campbell, A., Choudhury, T., Hu, S., Lu, H., Mukerjee, M. K., Rabbi, M., and Raizada, R. D. (2010). NeuroPhone: brain-mobile phone interface using a wireless EEG headset. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, page 38. ACM.
- Carrino, F., Dumoulin, J., Mugellini, E., Khaled, O., and Ingold, R. (2012). A self-paced BCI system to control an electric wheelchair: Evaluation of a commercial, low-cost EEG device. In *Biosignals and Biorobotics Conference (BRC), 2012 ISSNIP*, pages 1–6.
- Chuang, J., Nguyen, H., Wang, C., and Johnson, B. (2013). I think, therefore i am: Usability and security of authentication using brainwaves. In Adams, A., Brenner, M., and Smith, M., editors, *Financial Cryptography and Data Security*, volume 7862 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg.
- Crowley, K., Sliney, A., Pitt, I., and Murphy, D. (2010). Evaluating a brain-computer interface to categorise human emotional response. In *ICALT*, page 276278.
- D. Mattia, F. Pichiorri, M. M. R. R. (2011). Brain computer interface for hand motor function restoration and rehabilitation. In *Towards Practical Brain Computer Interfaces*. Springer, Biological and Medical Physics, Biomedical Engineering.
- De Vos, M. and Debener, S. (2014). Mobile eeg: towards brain activity monitoring during natural action and cognition. *International journal of psychophysiology: official journal of the International Organization of Psychophysiology*, 91(1):1–2.
- Dornhege, G. (2007). *Toward Brain-computer Interfacing*. MIT Press.
- Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: a library for large linear classification. *J. Mach. Learn. Res.*, 9:18711874.
- Garrett, D., Peterson, D., Anderson, C., and Thaut, M. (2003). Comparison of linear, nonlinear, and feature selection methods for eeg signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):141–144.
- Grierson, M. and Kiefer, C. (2011). Better brain interfacing for the masses: progress in event-related potential detection using commercial brain computer interfaces. page 1681. ACM Press.
- Hill, N. J., Ricci, E., Haider, S., McCane, L. M., Heckman, S., Wolpaw, J. R., and Vaughan, T. M. (2014). A practical, intuitive braincomputer interface for communicating yes or no by listening. *Journal of Neural Engineering*, 11(3):035003.
- Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):437.
- Larsen, E. A. and Hokl, C.-s. J. (2011). *Classification of EEG Signals in a Brain- Computer Interface System*.
- Lotte, F., Congedo, M., Lcuyer, A., Lamarche, F., Arnaldi, B., et al. (2007). A review of classification algorithms for EEG-based braincomputer interfaces. *Journal of neural engineering*, 4.
- McFarland, D. J. and Wolpaw, J. R. (2011). Brain-computer interfaces for communication and control. *Commun ACM*, 54(5):60–66.
- Milln, J. D. R., Rupp, R., Mller-Putz, G. R., Murray-Smith, R., Giugliemma, C., Tangermann, M., Vidaurre, C., Cincotti, F., Kbler, A., Leeb, R., Neuper, C., Mller, K.-R., and Mattia, D. (2010). Combining brain-computer interfaces and assistive technologies: State-of-the-art and challenges. *Front Neurosci*, 4.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, . (2011). Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:28252830.
- Raudys, S. J. and Jain, A. K. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, 13(3):252264.
- Vidaurre, C. and Blankertz, B. (2010). Towards a cure for BCI illiteracy. *Brain topography*, 23(2):194198.
- Vidaurre, C., Sannelli, C., Mller, K.-R., and Blankertz, B. (2011a). Co-adaptive calibration to improve BCI efficiency. *Journal of Neural Engineering*, 8(2):025009.
- Vidaurre, C., Sannelli, C., Mller, K.-R., and Blankertz, B. (2011b). Machine-learning-based coadaptive calibration for brain-computer interfaces. *Neural Computation*, 23(3):791–816.
- Vidaurre, C., Schloogl, A., Cabeza, R., Scherer, R., and Pfurtscheller, G. (2006). A fully on-line adaptive BCI. *IEEE Transactions on Biomedical Engineering*, 53(6):1214–1219.