Self-recalibrating classifiers for intracortical brain–computer interfaces

J. Neural Eng. 11 (2014) 026001 (18pp) doi:10.1088/1741-2560/11/2/026001

# Self-recalibrating classifiers for intracortical brain–computer interfaces

**William Bishop[1,2], Cynthia C Chestek[3,4,5], Vikash Gilja[4,6], Paul Nuyujukian[7,8], Justin D Foster[3], Stephen I Ryu[3,9], Krishna V Shenoy[3,4,7,10,11] and Byron M Yu[2,12,13]**

[1] Department of Machine Learning, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2] Center for the Neural Basis of Cognition, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3] Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA
[4] Stanford Institute for Neuro-Innovation and Translational Neurosciences, Stanford University, Stanford, CA 94305, USA
[5] Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[6] Department of Computer Science, Stanford University, Stanford, CA 94305, USA
[7] Department of Bioengineering, Stanford University, Stanford, CA 94305, USA
[8] School of Medicine, Stanford University, Stanford, CA 94305, USA
[9] Palo Alto Medical Foundation, Palo Alto, CA 94301, USA
[10] Department of Neurobiology, Stanford University, Stanford, CA 94305, USA
[11] Neurosciences Program, Stanford University, Stanford, CA 94305, USA
[12] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[13] Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

E-mail: byronyu@cmu.edu

Received 22 June 2013, revised 11 October 2013
Accepted for publication 11 December 2013
Published 6 February 2014

**Abstract**

*Objective.* Intracortical brain–computer interface (BCI) decoders are typically retrained daily to maintain stable performance. Self-recalibrating decoders aim to remove the burden this may present in the clinic by training themselves autonomously during normal use but have only been developed for continuous control. Here we address the problem for discrete decoding (classifiers). *Approach.* We recorded threshold crossings from 96-electrode arrays implanted in the motor cortex of two rhesus macaques performing center-out reaches in 7 directions over 41 and 36 separate days spanning 48 and 58 days in total for offline analysis. *Main results.* We show that for the purposes of developing a self-recalibrating classifier, tuning parameters can be considered as fixed within days and that parameters on the same electrode move up and down together between days. Further, drift is constrained across time, which is reflected in the performance of a standard classifier which does not progressively worsen if it is not retrained daily, though overall performance is reduced by more than 10% compared to a daily retrained classifier. Two novel self-recalibrating classifiers produce a ∼15% increase in classification accuracy over that achieved by the non-retrained classifier to nearly recover the performance of the daily retrained classifier. *Significance.* We believe that the development of classifiers that require no daily retraining will accelerate the clinical translation of BCI systems. Future work should test these results in a closed-loop setting.

Keywords: brain–computer interface, classification, stable decoding

S Online supplementary data available from stacks.iop.org/JNE/11/026001/mmedia

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

## 1. Introduction

Brain–computer interfaces (BCIs) aim to assist disabled patients by translating recorded neural activity into control signals for assistive devices. Intracortical BCI systems rely upon arrays of chronically implanted microelectrodes which penetrate into the cortex to record the activity of tens to hundreds of neurons. The last decade has witnessed a substantial investment of resources by the community into developing decoders for intracortical BCI which are more accurate, both in offline simulations with prerecorded data and in online, closed-loop settings. However, in almost all cases these decoders are retrained daily in a supervised manner to maintain their accuracy (e.g., Musallam *et al* 2004, Santhanam *et al* 2006, Velliste *et al* 2008, Li *et al* 2009, Hochberg *et al* 2012, Gilja *et al* 2012, Collinger *et al* 2012, Shanechi *et al* 2012). This retraining typically requires that the subject perform a series of trials where the goal of each trial, such as desired reach target, is known so labeled data for training can be collected.

While retraining procedures may only require minutes of a subject's time, future users may find daily repetition of a task which stands between them and device use burdensome. Indeed, imagine the annoyance modern cell phone users would experience if they had to calibrate the touch screens of their devices everyday before use. A few minutes of calibration would be a small price to pay for the connectivity these devices provide, but it is difficult to imagine the widespread adoption of modern phones under these conditions. In the same way, we suggest that modern BCI systems should aim to achieve predictable performance each day without undue burden to users.

An alternative to retraining in a supervised manner is to employ self-recalibrating neural decoders which can simultaneously estimate a user's intent along with changes in neural tuning parameters. These decoders, generally referred to as adaptive decoders, continuously train themselves during normal BCI use without requiring knowledge of a user's true intended actions. In this way, the retraining period at the beginning of each day can be eliminated. There have been relatively few studies on self-recalibrating decoders for intracortical BCI, and the work that does exist has focused on BCI decoders which decode a continuously valued control signal, such as the position of a computer cursor or prosthetic limb (Eden *et al* 2004a, 2004b, Srinivasan *et al* 2007, Li *et al* 2011).

While continuous decoders have many important applications, classifiers of discrete actions are an important part of existing real-world BCI systems (Ohnishi *et al* 2007). Potential clinical applications for BCI classifiers include making selections from a menu interface or virtual keyboard (e.g., Scherer *et al* 2004, Santhanam *et al* 2006, Brunner *et al* 2011), selecting among grasp types for a prosthetic hand (e.g., Pistohl *et al* 2012, Chestek *et al* 2013), classifying finger movements (e.g., Aggarwal *et al* 2008, Wang *et al* 2009), identifying periods of planning and movement (e.g., Shenoy *et al* 2003, Achtman *et al* 2007, Kemere *et al* 2008, Wang *et al* 2012) and providing discrete control signals for a wheelchair (e.g., Leeb *et al* 2007, Galán *et al* 2008). In some settings it may be possible to interface with a device using either a continuous or discrete control signal. For example, when interfacing with a menu, a cursor can be moved over items to be selected or a discrete decoder can be used to make a direct menu selection. In certain cases, the use of a classifier may provide a substantial performance improvement to an end user. Indeed, we have demonstrated a discrete decoder which transmits the user's intent at a rate of up to 6.5 bits s$^{-1}$ ($\sim$15 words min$^{-1}$), which is faster than continuously guiding a cursor to make discrete selections (Santhanam *et al* 2006).

Despite the potential clinical benefit that intracortical BCI classifiers may provide, the subject of self-recalibrating classifiers has received little attention in the intracortical community. In contrast, there is previous work in the EEG community which characterizes signal drift (e.g., Shenoy *et al* 2006, Pascual *et al* 2011), and proposes self-recalibrating classifiers. Example algorithms include approaches based on variational Kalman filtering (Sykacek and Roberts 2002, Sykacek *et al* 2004), expectation maximization (EM) (Blumberg *et al* 2007) and a variety of other methods to update the parameters of an LDA classifier (Vidaurre *et al* 2008, Vidaurre and Blankertz 2010). However, many of these algorithms consider only binary classification, and it is not clear the complexity of some (e.g., Sykacek and Roberts 2002, Sykacek *et al* 2004, Blumberg *et al* 2007) is required for BCI application.

In the present work, we seek to develop self-recalibrating classifiers for intracortical BCI. We note this differs from the previous work of Santhanam *et al* (2009). Santhanam *et al* develop classifiers with improved within-day accuracy by accounting for unobserved influences on neural activity, while we focus on achieving stable across-day performance through self-recalibration. We chose to forego spike sorting in favor of the use of threshold crossings. That is we mark any voltage event which crosses a set threshold on each electrode as a spike. In one subject, we also enforce that threshold events must pass a shape heuristic. Previous work has shown only minor reductions in performance when threshold crossings are used in place of spike sorting (Fraser *et al* 2009, Chestek *et al* 2011), and this removes the need to track and identify units across days.

Having made this design decision, we study how firing rates drift within and between recording sessions on different electrodes. This enables us to then develop self-recalibrating classifiers which take advantage of structure we find in this drift. Along the way, we also examine the performance of a standard BCI classifier that undergoes no daily retraining. In addition to being a novel result in itself, this provides a baseline when assessing the performance of the self-recalibrating classifiers presented in this work.

## 2. Methods

### 2.1. Reach task and neural recordings

Animal protocols were approved by the Stanford University Institutional Animal Care and Use Committee. We trained two adult male monkeys (*Macaca mulatta*, monkeys I and L) to perform center-out reaches for liquid rewards. As illustrated

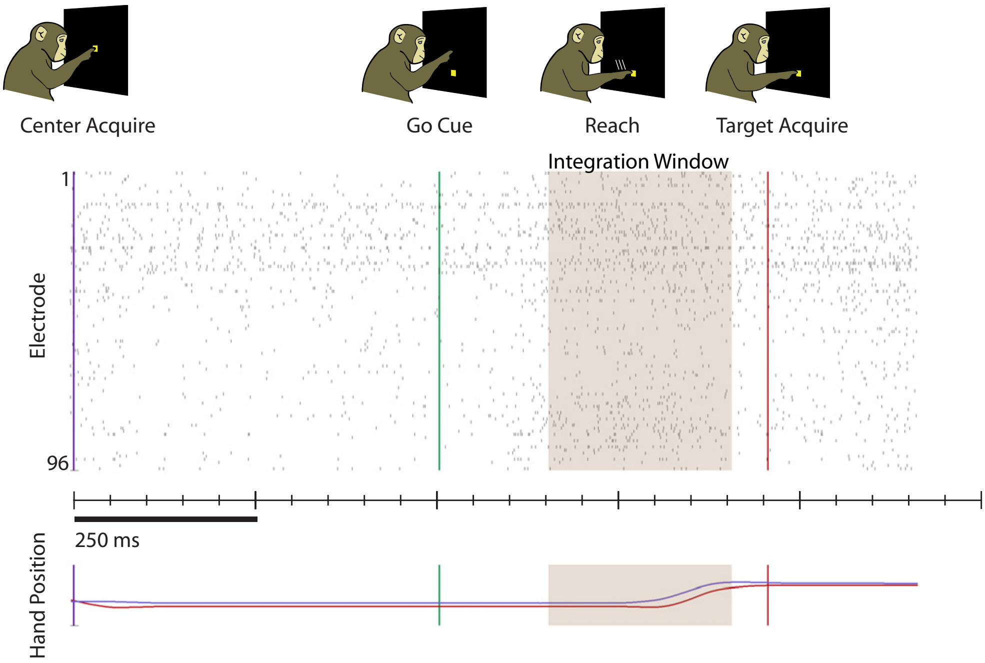

**Figure 1.** A representative trial of the center-out reaching task (Monkey L, trial ID: R,2008-12-26,mm, 109). The top row is the behavioral task. The middle row contains spike rasters across 96 electrodes and the bottom row presents the timecourse of the $x$ and $y$ coordinates of hand position. Full range of scale for the hand position is 15 cm. The shaded region represents the 250 ms integration window starting 150 ms after the go cue, which was used to count spikes in for each trial.

in figure 1, visual targets were back-projected onto a vertical screen in front of the monkey. A trial began when the monkey touched a central yellow square. Following a randomized (300–500 ms) hold period, a visual reach target appeared. After a brief (35 ms) delay, the animal was permitted to reach. Reach targets could appear at one of 28 locations. Each of the 28 targets was typically located at a distance of 4.5, 7.0, 9.5, or 12 cm from the central target and at an angle of 40°, 85°, 130°, 175°, 220°, 310°, or 355° from horizontal so that the targets were arranged in 4 concentric rings of 7 targets each. For all analyses in this paper, we collapse across reach distance and only consider the direction of reach on each trial. After the reach target was held for 200 ms for monkey L and 300 ms for monkey I, a liquid reward was delivered.

Monkeys sat in a custom chair (Crist Instruments, Hagerstown, MD) with the head braced and nonreaching arm comfortably restrained. A 96-channel silicon electrode array (Blackrock Microsystems, Salt Lake City, UT) was implanted in motor cortex contralateral to the reaching arm. When identifying neural spikes on each electrode, threshold crossings (Chestek *et al* 2009, Fraser *et al* 2009) were used in place of spike sorting. For monkey L, we also required voltage waveforms to pass a shape heuristic. Threshold levels for each electrode were set at the beginning of each recording day with an automated method and then, with possible rare exception, held fixed for the remainder of data collection. The automated method of setting threshold levels consisted of collecting 2 s of filtered voltage data sampled at 30 kHz from each electrode and calculating a root-mean-square like value from this data. Thresholds levels were then set at three times this value for each electrode.

Recordings were made on a total of 41 (monkey L) and 36 (monkey I) days spanning a period of 48 (monkey L) and 58 (monkey I) days in total. In all of the work that follows, we limit ourselves to only analyzing successful trials. In the data analyzed, there were (mean ± s.d.) 1737 ± 482 (monkey L) and 1688 ± 368 (monkey I) successful trials per day spanning 53 ± 16 (monkey L) and 88 ± 16 (monkey I) min per day.

### 2.2. Mathematical notation

We aim to develop BCI classifiers which maintain stable day-to-day performance without any daily supervised retraining. For clarity, we will speak of developing classifiers within the context of the datasets used in this study, but the techniques presented here should generalize to other intracortical BCI classification scenarios.

Throughout this work, we rely on probabilistic methods to infer the intended reach direction from neural data. For trial $t$ on day $d$, we form a feature vector, $\boldsymbol{x}_{d,t} \in \mathbb{N}^{E\times 1}$, from our neural data by counting the number of threshold crossings on each of $E$ electrodes of the recording array. The notation $x_{d,e,t}$ will refer to the specific spike count for electrode $e$[14]. Similarly, $y_{d,t} \in \{1, \ldots, J\}$ indicates which one of $J$ directions a reach is made in for a specific trial and day. Throughout this work we follow the convention of using bold script exclusively to denote vectors. The symbol $P$ will be used to indicate probability distributions.

On any given trial there is a probabilistic model relating observed spike counts to reach direction. We model this with the distribution $P(\boldsymbol{x}_{d,t}|y_{d,t}; \Theta_{d,t})$, where $\Theta_{d,t}$ is a set of parameters characterizing the distribution. In this work for electrode $e$, these parameters consist of the means $\mu_{e,j}$ and variances $\upsilon_{e,j}$ of spike counts when reaches are made

[14] Throughout this paper we will use the convention of listing subscripts in alphabetical order.

in each direction $j$. We will refer to this set of parameters as *tuning parameters*. While both means and variances can drift in principle, we focus on achieving stable classification through tracking changes in mean spike counts. We term the set of $\mu_{e,j}$ parameters for a fixed reach direction $j$ as class means, as they are mean counts conditioned on a class of movement. We can index class means across days and trials as $\mu_{d,e,j,t}$, where $d$ indexes day and $t$ indexes a trial within a day. In some of the work that follows, we will speak of class means across all trials of a day, which will we denote as $\mu_{d,e,j}$.

### 2.3. Tracking drifting tuning parameters across time

Movements of the recording array relative to the brain (Santhanam *et al* 2007), scar tissue buildup (Polikov *et al* 2005), behavioral changes (Chestek *et al* 2007) and neural plasticity (Taylor *et al* 2002, Jarosiewicz *et al* 2008, Ganguly and Carmena 2009, Chase *et al* 2012) may all cause tuning parameters to drift. In this section, we track drift in class means across days and time by convolving spike counts for all reach directions in a particular direction with a Gaussian kernel. This is a simple way of smoothing spike counts to estimate tuning parameters when true reach direction is known on each trial. This is an appropriate assumption for this analysis which aims to establish basic properties of tuning parameter drift. However, the techniques introduced in this section should not be confused with the methods that are introduced later to track drifting tuning parameters by our self-recalibrating classifiers when knowledge of true reach direction is not assumed known. Mathematically, estimates of tuning parameters are formed through convolution with a kernel as

$$\hat{\mu}_{d,e,j,t} = \frac{1}{C} \sum_{s:y_{d,s}=j} K(t-s;\, w) x_{d,e,s}, \qquad (1)$$

where $K(t-s;\, w)$ is Gaussian density function with 0 mean and standard deviation $w$ and $C = \sum_{s:y_{d,s}=j} K(t-s;\, w)$ is a normalizing constant. Mathematically, the function $K$ indexes over trial number, and therefore, $t-s$ in the argument is the distance in *number of trials* between two trials and not time. There remains the challenge of selecting the proper value of the width, $w$, of the Gaussian density function. For the purposes of this paper, we will present results for a single value of $w$ and then show in *Supplementary Results* (available at stacks.iop.org/JNE/11/026001/mmedia) that the qualitative results we obtain are robust as the value of $w$ is varied.

We are interested in comparing the amount of drift between and within days. If we smooth only within days, we enforce smoothing within but not across days and we may bias our results to show larger between than within-day drift. To account for this, we ignore day boundaries and smooth across all trials for a given electrode and reach direction as if they were on the same day. This may tend to 'over-smooth' between days, but to foreshadow what is to come, we would like to argue that within-day variation in tuning parameters is small relative to between-day variation. Therefore, this procedure is conservative for our purposes.

### 2.4. A standard BCI classifier

We desire to examine the impact of tuning parameter drift on a standard BCI classifier. We select a Gaussian, naïve Bayes decoder as our standard classifier, which has been used for BCI classification before (Santhanam *et al* 2006, Maynard *et al* 1999). Briefly, this classifier models spike counts on electrodes as independent Gaussian distributions conditioned on reach direction. The notation $\mu_{e,j}$ and $v_{e,j}$ refer to the mean and variance of the distribution for electrode $e$ and reach direction $j$. For notational convenience, we will group all the means and variances for a single electrode $e$ into the vectors $\boldsymbol{\mu}_e \in \mathbb{R}^{J\times 1}$ and $\boldsymbol{v}_e \in \mathbb{R}^{J\times 1}$, respectively. The joint probability of observing spike counts for all electrodes can be computed as the product of probabilities for individual electrodes. Formally, we write this as

$$P(\boldsymbol{x}_{d,t}|y_{d,t}) = \prod_{e=1}^{E} P(x_{d,e,t}|y_{d,t}). \qquad (2)$$

A uniform prior probability of reaching in any direction, $P(y_{d,t}) = 1/J$, is employed and Bayes' rule can then be used to find the posterior probability of reaching in direction $j$ given neural data, $P(y_{d,t} = j|\boldsymbol{x}_{d,t})$. This posterior is calculated for each reach direction, and the direction with the highest probability is selected as the decoded direction for each trial.

In this work we will examine the performance of two versions of this standard classifier, which differ only in the way they are trained. The first version, referred to as the standard retrained classifier, learns the $\boldsymbol{\mu}_e$ and $\boldsymbol{v}_e$ parameters fresh with trials from the beginning of each day on which classification will be performed. This emulates standard practice in the BCI field. The second version, referred to as the standard non-retrained classifier, is trained initially on all trials collected on a small number of days in which neural data and desired reach direction is known. After this training, the parameters of the standard non-retrained classifier are held fixed, and it is then used to predict reach direction for trials on subsequent days without any type of retraining or recalibration.

### 2.5. A probabilistic self-recalibrating classifier

We now describe the intuition and probabilistic model behind our two self-retraining classifiers. The first version, which we refer to as the *SR* classifier, takes a principled probabilistic approach which provides insight into the simplified *SR* ($SR_S$) classifier we subsequently present. Readers interested only in the $SR_S$ classifier may skip to section 2.6 after reading the part of this section outlining the probabilistic model common to both classifiers without loss of understanding.

The *SR* classifier can be considered an augmented standard classifier. It continues to assume tuning parameters are fixed within a day, but it no longer assumes class means are known at the beginning of each decoding day. Mathematically, it achieves this by introducing a distribution over class mean values that a decoding day begins with. As a decoding day progresses, the *SR* classifier produces refined distributions over class means while simultaneously estimating reach direction on a trial-by-trial basis.

A formal description of the probabilistic model for the *SR* classifier follows. When explicitly referring to the class mean for electrode $e$, reach direction $j$ on day $d$, we will use the notation $\mu_{d,e,j}$. In results we show that the $\mu_{d,e,j}$ parameters for the same electrode drift from day to day in a highly correlated manner. Intuitively, this means that the average spike counts observed for two different reach directions on the same electrode tend to move up and down together across days. We capture this intuition by referencing all class means for electrode $e$ to the same base value $b_{d,e}$ for day $d$, and we allow this base value to be drawn from a Gaussian distribution independently on each day. Mathematically,

$$b_{d,e} \sim \mathcal{N}(m_e, s_e), \tag{3}$$

where $m_e$ is the mean and $s_e$ is the variance of the distribution from which base values are drawn. For notational convenience we will group all the base values for all electrodes for day $d$ into a vector $\boldsymbol{b}_d \in \mathbb{R}^{E\times 1}$. We model the base values for different electrodes as independent, and the prior probability for the set of base values for all electrodes is therefore given by

$$P(\boldsymbol{b}_d) = \prod_{e=1}^{E} P(b_{d,e}). \tag{4}$$

We relate the class mean for reach direction $j$ for an electrode to its base value through an offset, $o_{e,j}$, so that

$$\mu_{d,e,j} = o_{e,j} + b_{d,e}. \tag{5}$$

The offset, $o_{e,j}$, is fixed across days. Class means for a single electrode can then vary from day to day but they must move up and down across days together. Having defined a probabilistic model for $b_{d,e}$, we specify an observation model for a single electrode by defining the probability of a spike count on electrode $e$ for trial $t$ on day $d$ given $b_{d,e}$ and reach direction $j$ as

$$x_{d,e,t}|b_{d,e}, y_{d,t} = j \sim \mathcal{N}(o_{e,j} + b_{d,e}, v_{e,j}). \tag{6}$$

This observation model is almost identical to that of the standard classifier with one important exception: spike counts now depend on the base value, which is a random variable that can vary from day to day. To specify the complete observation model, we again model electrodes as conditionally independent given reaching direction so that

$$P(\boldsymbol{x}_{d,t}|\boldsymbol{b}_d, y_{d,t}) = \prod_{e=1}^{E} P(x_{d,e,t}|b_{d,e}, y_{d,t}). \tag{7}$$

To complete the specification of the probabilistic model for the *SR* classifier, we define the prior probability of reaching in any direction as uniform, exactly as specified for the standard model.

*2.5.1. Decoding with the self-recalibrating classifier.* We now show how to decode reach direction while simultaneously refining knowledge of class means with the *SR* classifier. For each trial, we seek to find two distributions. The first is $P(y_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$, the posterior distribution over reach directions. In this work the notation $\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}$ will be used to indicate the set of trials from 1 through $t$ on day $d$. The most probable direction from $P(y_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$ is selected as the decoded reach direction on each trial. The second distribution is $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$, which embodies the decoder's knowledge of the base values for a day given the neural activity that it has seen.

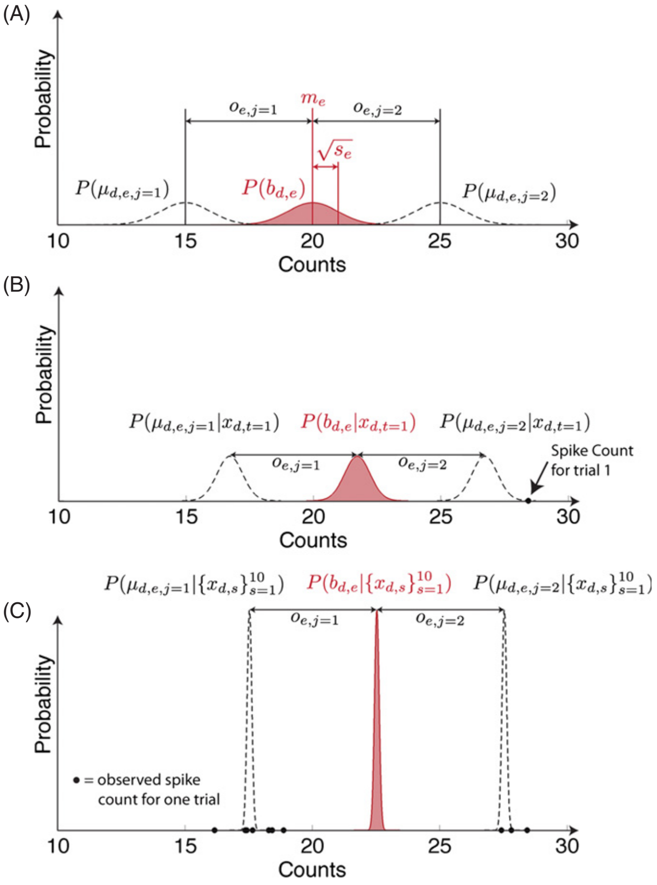

**Figure 2.** An illustration of the probabilistic distribution over tuning parameters for the self-recalibrating classifier at three different points in time during a decoding session on day $d$ for an example where spike counts recorded on one electrode are used to predict reaches in one of two directions. Panel (A) shows the distribution over the base value for the day, $b_{d,e}$, for the electrode at the start of the decoding day before any neural activity is available. This distribution is a Gaussian distribution with mean $m_e$ and variance $s_e$. In all panels, distributions over $b_{d,e}$ are shown in red to indicate that this is the distribution the self-recalibrating classifier explicitly represents. Panel (A) also shows the distributions over $\mu_{d,e,j=1}$ and $\mu_{d,e,j=2}$, which are the mean number of spike counts conditioned on reaching in directions 1 and 2. The classifier does not explicitly represent these distributions, but they can be recovered from the distribution over $b_{d,e}$ with the formula $\mu_{d,e,j} = b_{d,e} + o_{e,j}$. Panel (B) shows the distributions over $b_{d,e}$, $\mu_{d,e,j=1}$ and $\mu_{d,e,j=2}$ after a spike count for one trial has been observed. Importantly, note that the true reach direction associated with this spike count is not made available to the classifier. Panel (C) shows distributions over the same quantities after ten trials have been observed. The solid dots in panels (B) and (C) represent spike counts observed on individual trials. For this illustrative example, spike counts were drawn from Gaussian distributions and not rounded to the nearest whole number.

As a decoding day progresses, the decoder's knowledge of the true base values for a day, as represented in the distribution $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$, becomes more refined. This is possible even though the decoder is never provided with knowledge of the user's true reach directions. The key intuition is that spike counts will tend to cluster around class means for a day, and this structure can be exploited by our classifier, as illustrated for a one electrode and two reach direction decoding scenario in figure 2.

We now describe the formal decoding procedure for arriving at the distributions $P(y_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$ and $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$ for each trial. The algorithm is recursive and will use the results from the previous trial, the distributions $P(y_{d,t-1}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ and $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$, as its starting point in its calculations for trial $t$. When decoding the first trial, the prior distribution on reach directions, $P(y_{d,t})$, is used in place of $P(y_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ and the prior distribution on base values, $P(\boldsymbol{b}_d)$, is used in place of $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$.

*Calculating* $P(y_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$. For trial $t$, we calculate $P(y_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$ using Bayes' rule as

$$\begin{aligned} P\big(y_{d,t} = j|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}\big) &= \frac{P\big(\boldsymbol{x}_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}, y_{d,t} = j\big)P\big(y_{d,t} = j|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}\big)}{P\big(\boldsymbol{x}_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}\big)} \\ &= \frac{c_{t|j}P(y_{d,t})}{c_t}, \end{aligned} \tag{8}$$

where in moving to the second line we have defined $c_{t|j} = P(\boldsymbol{x}_{d,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}, y_{d,t} = j)$, made use of the fact that reach direction is independent of neural activity so that $P(y_{d,t} = j|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}) = P(y_{d,t})$, and recognized that the denominator of the first line is a normalizing constant calculated as $c_t = \sum_{j=1}^{J} c_{t|j}P(y_{d,t})$.

The scalar $c_{t|j}$ is the probability of the neural activity for trial $t$ if the subject reaches in direction $j$ given all past neural activity seen during the day. We will soon describe a means of calculating its value, but first note that it appears in the calculations for another distribution, $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}, y_{d,t} = j)$, which we will need shortly. This distribution gives the probability of base values conditioned on all observed neural activity if the subject were to reach in direction $j$ for the current trial. To calculate it, we again use Bayes' rule to write

$$\begin{aligned} P\big(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}, y_{d,t} = j\big) &= \frac{P(\boldsymbol{x}_{d,t}|\boldsymbol{b}_d, y_{d,t} = j)P\big(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}\big)}{c_{t|j}}, \end{aligned} \tag{9}$$

where $P(\boldsymbol{x}_{d,t}|\boldsymbol{b}_d, y_{d,t} = j)$ is given by equation (7) and $P(\boldsymbol{b}_d|\{\boldsymbol{x}_s\}_{s=1}^{t-1})$ is the posterior over base values obtained when decoding the previous trial. In forming the numerator of equation (9), we have used the fact that $\boldsymbol{x}_{d,t}$ conditioned on $\boldsymbol{b}_d$ is independent of all previous neural observations and that base value is independent of $y_{d,t}$ conditioned on the previous neural activity. We see that the numerator on the right-hand side of equation (9) is in the form of the product of two Gaussians. $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ is defined to be a Gaussian distribution with mean $\boldsymbol{m}_{t-1}$ and covariance matrix $\Sigma_{t-1}$. Viewed as a function of $\boldsymbol{b}_d$, $P(\boldsymbol{x}_{d,t}|\boldsymbol{b}_d, y_{d,t} = j)$ is Gaussian with mean $\boldsymbol{x}_{d,t} - \boldsymbol{o}_j$, where $\boldsymbol{o}_j \in \mathbb{R}^{E\times 1}$ is a vector of offsets for each electrode for reach direction $j$, and covariance matrix $V_j \in \mathbb{R}^{E\times E}$, where $V_j$ is a diagonal matrix with the values of $v_{e,j}$ along its diagonal. We can then calculate the mean, $\boldsymbol{m}_{t|j}$, and covariance matrix, $\Sigma_{t|j}$, of $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}, y_{d,t} = j)$ of the left-hand side of equation (7) as

$$\Sigma_{t|j} = \big(V_j^{-1} + \Sigma_{t-1}^{-1}\big)^{-1} \tag{10}$$

$$\boldsymbol{m}_{t|j} = \Sigma_{t|j}\big(V_j^{-1}(\boldsymbol{x}_{d,t} - \boldsymbol{o}_j) + \Sigma_{t-1}^{-1}\boldsymbol{m}_{t-1}\big), \tag{11}$$

where the notation $\cdot^{-1}$ indicates matrix inversion. The scalar $c_{t|j}$ is then the integral over the numerator of equation (9) and can be calculated as

$$\begin{aligned} c_{t|j} &= \int P(\boldsymbol{x}_{d,t}|\boldsymbol{b}_d, y_{d,t} = j)P\big(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}\big)\, \mathrm{d}\boldsymbol{x}_{d,t} \\ &= \sqrt{(2\pi)^E \det(\Sigma_{t|j})}P(\boldsymbol{x}_{d,t}|\boldsymbol{b}_d = \boldsymbol{m}_{t|j}, y_{t,d} = j) \\ &\quad \times P(\boldsymbol{b}_d = \boldsymbol{m}_{t|j}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}), \end{aligned} \tag{12}$$

where $P(\boldsymbol{x}_{d,t}|\boldsymbol{b}_d = \boldsymbol{m}_{t|j}, y_{t,d} = j)$ is obtained by evaluating equation (7) for the current observation at $\boldsymbol{b}_d = \boldsymbol{m}_{t|j}$ and $P(\boldsymbol{b}_d = \boldsymbol{m}_{t|j}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ is obtained by evaluating the posterior over base values from the last trial at the same point, $\boldsymbol{b}_d = \boldsymbol{m}_{t|j}$.

*Calculating* $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$. Finally, we compute $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$, the posterior distribution over the base values for all electrodes. An analytical expression for this can be written as

$$\begin{aligned} P\big(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}\big) &= \sum_{j=1}^{J} P\big(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}, y_{d,t} = j\big) \\ &\quad \times P\big(y_{d,t} = j|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}\big), \end{aligned} \tag{13}$$

where the two terms on the right-hand side of equation (13) are obtained from equations (8) and (9). The right-hand side of equation (13) is a mixture of multivariate Gaussians. This will be problematic as in general there will be an exponentially increasing number of components in the representation of $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$ as $t$ increases. Therefore, we approximate the right-hand side of equation (13) with mean $\boldsymbol{m}_t$ and covariance matrix $\Sigma_t$ using moment matching as has been done in previous hypothesis merging algorithms (Blom and Bar-Shalom 1988). We use the standard formulas for moment matching with two multivariate Gaussians to write

$$\boldsymbol{m}_t = \sum_{j=1}^{J} \boldsymbol{m}_{t|j}P\big(y_{d,t} = j|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}\big) \tag{14}$$

$$\begin{aligned} \Sigma_t &= \sum_{j=1}^{J} [\Sigma_{t|j} + (\boldsymbol{m}_{t|j} - \boldsymbol{m}_t)(\boldsymbol{m}_{t|j} - \boldsymbol{m}_t)^T] \\ &\quad \times P\big(y_{d,t} = j|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}\big), \end{aligned} \tag{15}$$

where $P(y_{d,t} = j|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$ is calculated from equation (8). This completes the decoding procedure for trial $t$. For each new trial, this set of calculations is repeated in an iterative manner.

### *2.5.2. Learning the parameters for the self-recalibrating classifier.*

While the *SR* classifier is designed to produce stable day-to-day classification without daily retraining, it must be initially trained once. To perform this initial training there must exist a small number of days for which neural activity, $\boldsymbol{x}_{d,t}$, has been recorded along with associated reach directions, $y_{d,t}$. We will use the notation $\mathcal{D}_{\text{Train}}$ to refer to this set of days. To avoid any possible confusion, we stress that after the classifier has been trained on this initial set of days, it requires no further supervised retraining and will be equipped to autonomously calibrate itself on new data that it encounters during normal decoding use.

The purpose of the initial training is to learn the mean, $m_e$, and variance, $s_e$, characterizing the distribution of base values for each electrode from equation (3) and the offset,

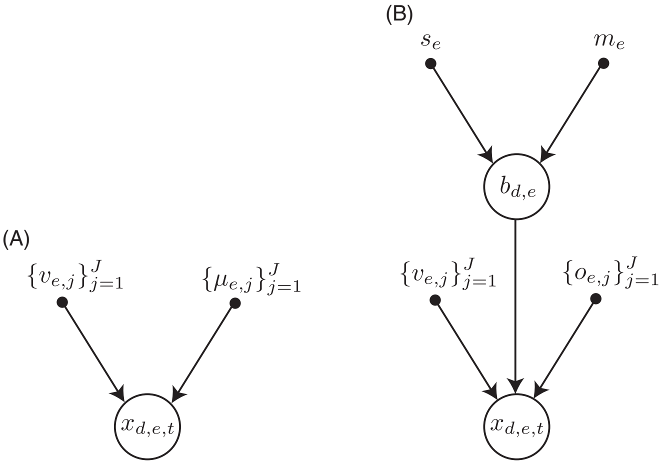

**Figure 3.** Graphical models for the generation of a single spike count on electrode $e$ for trial $t$ on day $d$ for the standard (A) and self-recalibrating (B) classifiers. Circles denote random variables and dots indicate model parameters. In the standard classifier, the electrode-class means, $\mu_{e,j}$, and variances, $v_{e,j}$, for all reach directions directly characterize the distribution over observed spike counts, $x_{d,e,t}$. For the self-recalibrating classifier, the base value for electrode $e$, $b_{d,e}$, is randomly generated each day from a Gaussian distribution with mean $m_e$ and variance $s_e$. For a given base value, spike counts are produced according to distributions with electrode-class means $\mu_{e,j} = b_{d,e} + o_{e,j}$ and variances $v_{e,j}$. In this figure, $\{v_{e,j}\}_{j=1}^{J}$, $\{\mu_{e,j}\}_{j=1}^{J}$ and $\{o_{e,j}\}_{j=1}^{J}$ indicate the set of variances, means and offsets, respectively, for all reach directions for electrode $e$.

$o_{e,j}$, and variance, $v_{e,j}$, characterizing the observation model in equation (6) for each electrode-class pair. Fitting the *SR* classifier is more difficult than fitting the standard classifier because the daily base values for each electrode, $b_{d,e}$, cannot be directly observed. Instead, these values are randomly drawn each day from the distribution specified by equation (3), and after the base values are drawn on each day, they specify the mean of the Gaussian distributions in equation (6) from which neural spike counts are drawn. Thus, the neural data we directly observe is essentially one step removed from the distribution that is characterized by the parameters we desire to estimate. Figure 3 illustrates the conceptual difference between the standard and *SR* classifier models.

The EM algorithm (Dempster *et al* 1977) is a general procedure for finding the maximum likelihood model parameters in the presence of unobserved variables, in this case $b_{d,e}$. The EM algorithm is an iterative procedure which is initialized with an initial guess for the model parameters. In this case, these are the $m_e$, $s_e$, $o_{e,j}$ and $v_{e,j}$ parameters. Each iteration of the algorithm produces new estimates for these parameters through a two step procedure. In the first step (referred to as the E-step) of each iteration, the latest parameter estimates and observed data are used to form a posterior distribution over the unobserved variables conditioned on the observed data. For the *SR* classifier, the unobserved variables are the daily base values for each electrode, $b_{d,e}$, and the observed data are the neural observations, $\boldsymbol{x}_{d,t}$, and reach directions, $y_{d,t}$, for all trials on all days in the training set. In the second step of each iteration (referred to as the M-step), parameter values are updated so that the expectation of the log likelihood of the observed and unobserved data, computed using the posterior distribution found in the first step of each iteration, is maximized. This algorithm is known to converge to a local optimum. See *Supplementary Methods* (available at stacks.iop.org/JNE/11/026001/mmedia) for specific details of the EM algorithm used in this work.

*2.5.3. Robust classification through outlier detection.* The classification procedure described above can be modified to improve robustness on real world data. Key to the probabilistic model underlying the *SR* classifier is the specification that firing rates can change between but not within days. While minor changes in firing rate throughout a day are tolerable, large changes, which may be observed in real data, can be problematic.

In particular, the *SR* classifier represents knowledge of the base values for a day obtained from classifying trials 1 through $t$ with the distribution $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$. This distribution is refined for each trial that is classified, and over time the uncertainty for the base values for a day, represented in the covariance matrix $\Sigma_t$, is reduced. This presents no problem in normal operation. However, if the firing rate of an electrode suddenly changes after the uncertainty in base values for a day is reduced, spike counts recorded on that electrode will strongly and erroneously affect classification.

The classification procedure can be modified by adding an additional step to be performed before any of the steps already described when classifying reach direction for trial $t$. In this step, electrodes with large firing rate changes are flagged, and the uncertainty in the estimate for the base value for those electrodes is increased to a predetermined value. This prevents those electrodes from dominating when determining reach direction probability and maintains classification accuracy. As classification continues, uncertainty around the base value estimates for the flagged electrodes is reduced starting from the predetermined value as the *SR* classifier refines the distribution $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$.

To flag erratically behaving electrodes, a distribution which predicts the probability of the spike count on trial $t$ for electrode $e$ given all previous neural data, $P(x_{d,e,t}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$, is formed for each electrode as shown in *Supplementary Methods* (available at stacks.iop.org/JNE/11/026001/mmedia). This can be used as a gauge for determining if the observed spike count, $x_{d,e,t}$, for a trial and an electrode is anomalous by setting upper and lower thresholds on the value of $x_{d,e,t}$ so that a normally behaving electrode will only exceed these values approximately 1% of the time. Whenever an electrode does exceed these values, it is flagged as behaving erratically. As with any anomaly detection procedure, thresholds have to be set to balance true and false positive rates, and a false positive rate of 1% should not significantly affect classification performance.

After one or more electrodes are flagged as behaving erratically for trial $t$, uncertainty around the estimates for their associated base values in the distribution $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ are increased. For ease of notation, we will momentarily switch from representing the set of base values in vector notation to set notation, so that each entry in the vector

$\boldsymbol{b_d}$ will be represented by a corresponding member in the set $\{b_{d,e}\}_{e=1}^{E}$. We will refer to the set of flagged electrodes for trial $t$ as $E_{\mathrm{t,flagged}}$ and the set of non-flagged electrodes as $E_{\mathrm{t,non\text{-}flagged}}$. The uncertainty for the base values of erratically behaving electrodes represented in the distribution for $P(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ is reset in a two step procedure. In the first step, the distribution over only non-flagged electrodes, $P(\{b_{d,e}\}_{e\in E_{\mathrm{t,non\text{-}flagged}}}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$, is formed. Formally, this is accomplished by forming the marginal distribution over the electrodes in $E_{\mathrm{t,non\text{-}flagged}}$ from $P(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ and is easily accomplished since $P(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ is a Gaussian distribution. In the second step, a new distribution over the base values for all electrodes, which we refer to as $P^*(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$, is formed as

$$P^*\big(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}\big) = P\big(\{b_{d,e}\}_{e\in E_{\mathrm{t,non\text{-}flagged}}}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1}\big)\prod_{e\in E_{\mathrm{t,flagged}}} P'(b_{d,e}), \tag{16}$$

where $P'(b_{d,e})$ are Gaussian distributions. The mean of the distribution $P'(b_{d,e})$ is set to, $m_{e,t-1}$, which is the element for electrode $e$ of the mean vector for the distribution $P(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$. This ensures that the mean of the final distribution, $P^*(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$, is the same as that of the original distribution $P(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$. The variance of the distribution $P'(b_{d,e})$ is set to $s_e$, the variance of the distribution that base values are modeled as being drawn from at the start of each decoding day. Computationally, these two steps can be carried out through a series of simple manipulations of the covariance matrix for $P(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ and increase the uncertainty around base values for suspect electrodes in $P^*(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$. After the distribution for $P^*(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ is formed, classification, as described by the algorithm in section 2.5.1, is performed where $P(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ is replaced with $P^*(\{b_{d,e}\}_{e=1}^{E}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t-1})$ as a starting point when decoding trial $t$.

### *2.6. A simplified self-recalibrating classifier*

The *SR* classifier addresses the problem of base values, $\boldsymbol{b}_d$, which can drift from day to day by starting with a prior distribution, $P(\boldsymbol{b}_d)$, over these values and then forming refined distributions, $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$, over base values based on neural activity seen throughout the decoding day. When predicting reach direction for a trial, the *SR* classifier then considers the probability of reach directions under all possibly values for $\boldsymbol{b}_d$ and weights its estimate according to how likely each value of $\boldsymbol{b}_d$ is, given by $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$.

Thus, the *SR* classifier never assumes any single value of $\boldsymbol{b}_d$ is correct, but decodes by considering an infinite number of $\boldsymbol{b}_d$ values, each weighted by some probability. This represents a fully probabilistic means of decoding. However, it is not the only possible means of accounting for drifting tuning parameters. In particular, we can forego modeling uncertainty in $\boldsymbol{b}_d$ and instead decode with a single point estimate for $\boldsymbol{b}_d$ formed for each trial. For the simplified classifier, $\boldsymbol{b}_d$ is defined so that $b_{d,e}$ is the average number of spike counts on electrode $e$ for any trial on day $d$. We can express this in mathematical notation as $b_{d,e} = \mathbb{E}[x_{d,e,t}]$, where the expectation is taken across trials.

An estimate for $b_{d,e}$ on trial $t$, which we refer to as $\hat{b}_{d,e,t}$, can be formed by taking the empirical average of all of the observed spike counts for electrode $e$ on day $d$ up to and including trial $t$. This estimate can be recursively formed as

$$n_{d,t} = n_{d,t-1} + 1, \tag{17}$$

$$\hat{b}_{d,e,t} = \frac{n_{d,t-1}\hat{b}_{d,e,t-1} + x_{d,e,t}}{n_{d,t}}. \tag{18}$$

We must select initial values for $n_{d,0}$ and $b_{d,e,0}$, where the hat has been dropped on $b_{d,e,0}$ to indicate this is a parameter of the model specifying a value with which to initialize equation (18). If we select $n_{d,0} = 0$, $\hat{b}_{d,e,t}$ will be the true empirical average for trials 1 through $t$. However, this has one disadvantage. Because of the very fact that equation (18) forms estimates as the empirical average of spike counts, at the start of the day there will be no previous trials with which to average spike counts for initial trials with and hence values for $\hat{b}_{d,e,t}$ will be strongly influenced by the spike counts of each trial. This will result in large fluctuations in $\hat{b}_{d,e,t}$ for the early trials of a decoding session. An alternative is to select $b_{d,e,0}$ so that it represents the expected average spike count on electrode $e$ for any given day and assign this some initial weight by setting $n_{d,0} > 0$. Intuitively, we can think of starting each day with an initial estimate of the average spike count for the electrode that was obtained as the empirical average of $n_{d,0}$ 'virtual samples'. This initial estimate is analogous to the prior distribution for the fully probabilistic classifier specified in equation (3). Importantly, as shown in *Supplementary Methods* (available at stacks.iop.org/JNE/11/026001/mmedia), as long as $n_{d,0}$ and $b_{d,e,0}$ are chosen to be finite, $\hat{b}_{d,e,t}$ approaches the true mean firing spike count for electrode $e$ as the number of trials observed during a day grows. While this is reassuring, there are still values of these parameters that will result in optimal decoding, and below we discuss how values for $n_{d,0}$ and $b_{d,e,0}$ can be learned from the training data.

After an estimate for $\hat{b}_{d,e,t}$ is formed for each trial, an estimate for the class means, $\mu_{d,e,j}$, can be obtained with equation (5), and a classification of reach direction can be made using the standard classifier described in section 2.4 with the updated class means. The $SR_S$ classifier can be conceptualized as consisting of two algorithms acting together. The first algorithm tracks base values, and hence tuning parameters, using an accumulated average of spike counts. This simple first algorithm is independent of and provides input in the form of updated tuning parameters to the second algorithm, which serves as the decoding core and is nothing more than the standard classifier. Figure 4 presents a schematic depiction of how the $SR_S$ classifier decodes a single trial and depicts the separation of the self-recalibrating and classification portions of the $SR_S$ decoding algorithm. See *Supplementary Methods* (available at stacks.iop.org/JNE/11/026001/mmedia) for a more formal description of the relationship between the *SR* and $SR_S$ classifiers. We note this method of tracking baseline firing rates is similar to the work of Li *et al* (2011), who in their work on adaptive decoders for continuous control signals experimented with tracking changes in baseline firing rates by measuring the empirical average of spike counts. However, since their algorithm runs in a batch mode and does not update

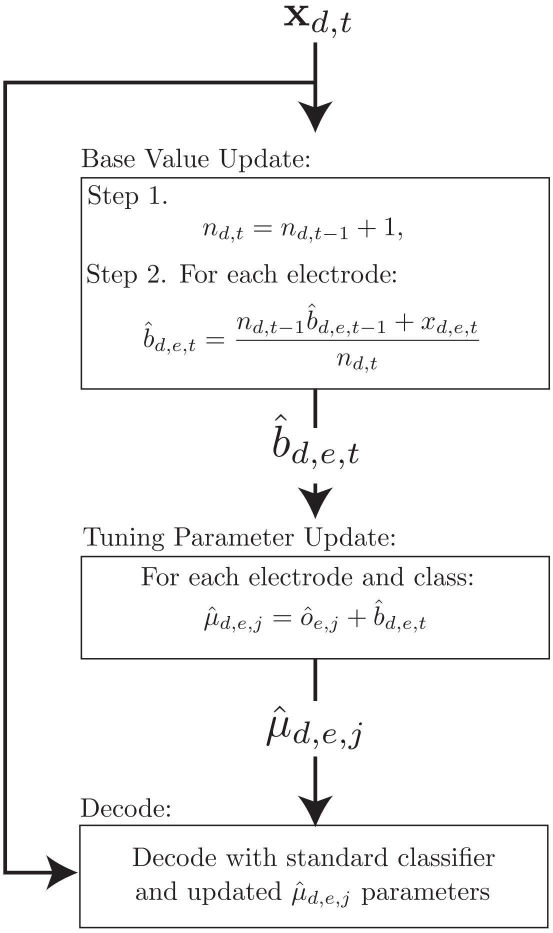
$\mathbf{x}_{d,t}$
Base Value Update:
Step 1.
$n_{d,t} = n_{d,t-1} + 1,$
Step 2. For each electrode:
$\hat{b}_{d,e,t} = \frac{n_{d,t-1}\hat{b}_{d,e,t-1} + x_{d,e,t}}{n_{d,t}}$
$\hat{b}_{d,e,t}$
Tuning Parameter Update:
For each electrode and class:
$\hat{\mu}_{d,e,j} = \hat{o}_{e,j} + \hat{b}_{d,e,t}$
$\hat{\mu}_{d,e,j}$
Decode:
Decode with standard classifier and updated $\hat{\mu}_{d,e,j}$ parameters

**Figure 4.** An illustration of how the $SR_S$ classifier predicts reach direction for a single trial. Starting from the top, spike counts, $\boldsymbol{x}_{d,t}$, for trial $t$ are recorded and the $SR_S$ decoder updates its estimate for the day's base values, $\hat{b}_{d,e,t}$, using a simple accumulated average. Updated estimates for tuning parameters, $\hat{\mu}_{d,e,j}$, are then calculated from the updated estimates of the base values. Finally, the standard classifier is used with the updated tuning parameters to predict reach direction.

parameters on a single trial basis as our does, they do not introduce the concept of selecting initial values and weights to begin the empirical average for a day with.

*2.6.1. Learning model parameters for the simplified self-recalibrating classifier.* As with the $SR$ classifier, the $SR_S$ classifier must also be trained once on an initial set of training days. The goal of this training is to learn the electrode-class variances, $v_{e,j}$, and the electrode-class offsets, $o_{e,j}$, for all electrodes and reach directions. In addition, values for $b_{d,e,0}$, must be learned for each electrode, and the initial weight, $n_0$, which will be assigned to these means must be determined.

Given a collection of days, $\mathcal{D}_{\text{Train}}$, we estimate values for $b_{d,e,0}$, $o_{e,j}$ and $v_{e,j}$ as follows. First, for each day in $\mathcal{D}_{\text{Train}}$ we estimate the average spike counts on each electrode over all reach directions, $\hat{\mu}_{d,e}$, and the average spike counts on each electrode when the subject made reaches in direction $j$, $\hat{\mu}_{d,e,j}$. We calculate

$$\hat{\mu}_{d,e} = \frac{\sum_{t=1}^{T_d} x_{d,e,t}}{T_d}, \tag{19}$$

and

$$\hat{\mu}_{d,e,j} = \frac{\sum_{t:y_{d,t}=j} x_{d,e,t}}{T_{d,j}}, \tag{20}$$

where $T_d$ is the number of trials on day $d$, and $T_{d,j}$ is the number of trials with reaches in direction $j$ on day $d$. Having calculated these intermediate values, an estimate of $b_{d,e,0}$ is then calculated as the average of the daily average spike counts as

$$\hat{b}_{d,e,0} = \frac{\sum_{d\in\mathcal{D}_{\text{Train}}} \hat{\mu}_{d,e}}{|\mathcal{D}_{\text{Train}}|}, \tag{21}$$

where $|\mathcal{D}_{\text{Train}}|$ is the number of days in $\mathcal{D}_{\text{Train}}$. Estimates for the class-electrode offsets, $o_{e,j}$, are calculated as the average of the difference between daily class-electrode mean spike counts and the daily average spike counts. We write this as

$$\hat{o}_{e,j} = \frac{\sum_{d\in\mathcal{D}_{\text{Train}}} \hat{\mu}_{d,e,j}}{|\mathcal{D}_{\text{Train}}|} - \hat{\mu}_{d,e}. \tag{22}$$

Finally, the electrode-class variances, $v_{e,j}$, are estimated as

$$\hat{v}_{e,j} = \frac{\sum_{d\in\mathcal{D}_{\text{Train}}} \sum_{t:y_{d,t}=j} (x_{d,e,t} - \hat{\mu}_{d,e,j})^2}{\left(\sum_{d\in\mathcal{D}_{\text{Train}}} T_{d,j}\right) - 1}, \tag{23}$$

which can be understood as a generalization of the standard formula for the variance of a sample generalized to account for values of $\mu_{d,e,j}$ which may change between days.

*Learning the value of $n_0$.* The equations above are used to estimate values for the $b_{d,e,0}$, $o_{e,j}$ and $v_{e,j}$ parameters of the model. We turn to cross-validation to learn a value for $n_0$, which is the amount of weight or number of virtual samples that should be assigned to the initial estimate of average electrode spike counts for a day, $b_{d,e,0}$.

In the cross-validation procedure, we seek the value of $n_0$ which will produce the highest classification accuracy on novel data. To do this, multiple rounds of training and validation are performed on the data that is present in $\mathcal{D}_{\text{Train}}$. In each round, one day in $\mathcal{D}_{\text{Train}}$ is set aside as a validation day. On the remaining days, the $b_{d,e,0}$, $o_{e,j}$ and $v_{e,j}$ parameters of the model are learned. Importantly, these values are learned on all days in $\mathcal{D}_{\text{Train}}$ except the validation day. This is critical because cross-validation uses the validation day to simulate classification on novel data, and it is therefore important that none of the trials in the validation day are used for learning any part of the model for a round of cross-validation. After learning these parameters, a value of $n_0$ is selected and the classifier is used to predict reach direction for each trial in the validation day. This is repeated for a range of $n_0$ values. For a single round of cross-validation, one or more values of $n_0$ will produce the highest classification accuracy on the validation day. To select the final value of $n_0$ for use in the trained classifier, a round of cross validation is performed for each day in $\mathcal{D}_{\text{Train}}$, and the value of $n_0$ which produces the highest average accuracy across all rounds is chosen.

## 3. Results

### 3.1. Tracking tuning parameter drift across time

Figures 5 and 6 show the results of the analysis of drifting tuning parameters for monkeys L and I, respectively.

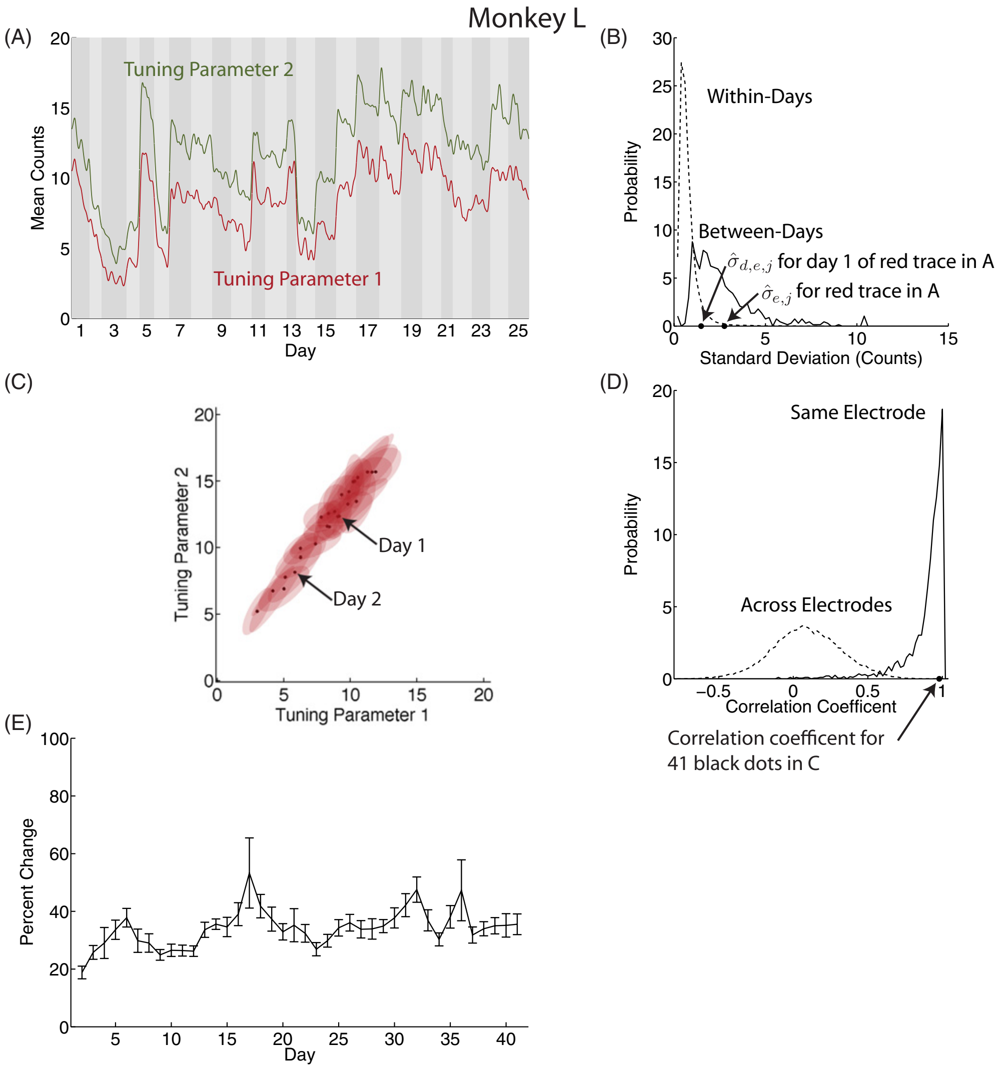

**Figure 5.** Statistical structure of changes in tuning parameters across days for monkey L. Panel (A): example traces of $\hat{\mu}_{d,e,j,t}$ for two representative tuning parameters on the same electrode across the first 25 days of data. Background shading represents day boundaries. Panel (B): distribution of the between ($\sigma_{e,j}$ for all $e$ and $j$) and within-day ($\sigma_{d,e,j}$ for all $d$, $e$ and $j$) standard deviations of tuning parameters. A small number (0.08%) of within-day standard deviations are outside the range of this plot. Panel (C): daily means (black dots) and associated ellipses which indicate regions in which tuning parameters for each day fall with approximately 95% probability for the same $\hat{\mu}_{d,e,j,t}$ parameters shown in panel (A) on each of the 41 days of recorded data. Panel (D): distribution of correlation coefficients computed across and within electrode pairs. To relate panel (C) to (D), the correlation of the cloud of 41 black points in (C) is *one* of the 96 × $\binom{7}{2}$ pairs of correlations that were used to construct the 'same electrode' distribution in panel (D). Panel (E): average values with 95% confidence intervals of unsigned percent change of mean tuning parameter values on days 2–41 from their corresponding value on day 1. Three electrodes were withheld form this analysis due to the small value of the tuning parameter for one or more reach directions on day 1. See text for details.

Throughout this work threshold crossings were collected in a single window per trial. In this analysis and in all of the work that follows we count spikes falling in a single 250 ms long window for each trial starting 150 ms after the go cue. As described in section 2.3, spike counts from each electrode were then convolved with a Gaussian kernel to produce smoothed estimates of tuning parameters across trials. The results presented here were obtained with a kernel width of 100 trials. Results obtained with kernel widths of 25, 50 and 200 trials were qualitatively similar (see *Supplementary Results*, available at stacks.iop.org/JNE/11/026001/mmedia).

Panel A in figure 5 shows the traces of two tuning parameters for one electrode for monkey L. Traces for monkey I were similar but are not shown. Three important properties of tuning parameter drift are apparent in this plot. First, tuning parameters change much more between than within days. Second, tuning parameters for the same electrode tend to move up and down together across days. Finally, tuning parameters do not tend to drift progressively further from their values on day 1 over time.

Figures 5(B) and 6(A) provide a quantitative measurement of the observation that tuning parameters appear to drift more between than within days. The drift of a tuning parameter within a single day is quantified as the standard deviation of its estimated values over all of the trials within a day, $\sigma_{d,e,j}$. This can be measured for all days, electrodes and reach directions.

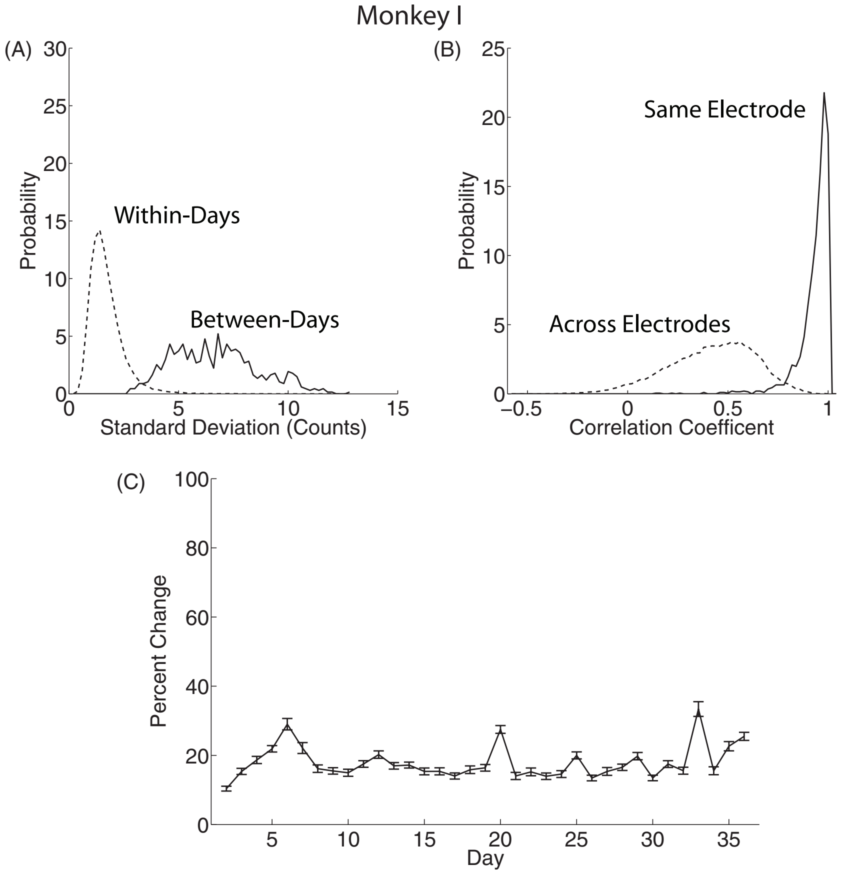

**Figure 6.** Summary of the statistical structure of changes in tuning parameters across days for monkey I. Panels (A), (B) and (C) correspond to panels (B), (D) and (E) in figure 5. No electrodes were withheld when calculating the average shown in panel (C). For space reasons, panels showing representative parameter traces and the spread of values for a pair of tuning parameters are not shown for monkey I.

The distribution of all $\sigma_{d,e,j}$ values is plotted as the dashed line in each figure. Similarly, drift between days is quantified as the standard deviation of mean tuning parameter values for each day, $\sigma_{e,j}$. The distribution of $\sigma_{e,j}$ parameters is plotted as the solid line in both figures. The mean of the between-day standard deviation values is significantly larger than that of the within-day standard deviation values (permutation test, $p < 0.01$). This finding motivates the decision to simplify the self-recalibrating classifiers so that they ignore within-day drift but compensate for between-day drift in tuning parameters.

Figures 5(D) and 6(B) provide evidence that tuning parameters on the same electrode drift in a correlated manner. Correlation coefficients for all pairs of daily mean tuning parameters (the black dots for the example pair of tuning parameters in figure 5(C)) on the same electrode were calculated for each electrode. We also examined correlation across electrodes by measuring the correlation of daily mean values for pairs of tuning parameters for the same reach direction across all pairs of electrodes. The distributions of these two sets of correlation coefficients are shown in figures 5(D) and 6(B). For both subjects, correlation coefficients for tuning parameters on the same electrode concentrate near 1. This insight has important practical implications for the design of self-recalibrating classifiers which will be detailed in section 4.

While tuning parameters drift in a correlated manner on the same electrode, it is important to point out that the distributions of correlation coefficients in figures 5(D) and 6(B) leave room for tuning parameters to move in a limited manner relative to one another while still moving up and down together. Indeed, this is visible in figure 5(A) where traces of the two tuning parameters move up and down together between days but are closer to one another on days 1–4 than they are for days 7–10, for example. We will show below that our decoders, which use a simplified design that models parameters on the same electrode as moving in perfect lockstep, achieve good performance, but future decoders could be developed to model additional aspects of tuning parameter drift. Interested readers may refer to *Supplementary Results* (available at stacks.iop.org/JNE/11/026001/mmedia), where we quantify changes in modulation depth across time.

Figures 5(E) and 6(C) measure the tendency of tuning parameters to progressively drift from their value on day 1. The percent change of the mean tuning parameter values from day 1, $\Delta_{d,e,j}$, for days 2 and on for all tuning parameters was calculated. The average of the absolute value of the percent change in tuning parameter values across all electrodes and reach directions for each day was then plotted in these figures. When performing this analysis, for numerical reasons we removed any electrodes which had a value of less than .001 spikes/window for any reach direction. The results are

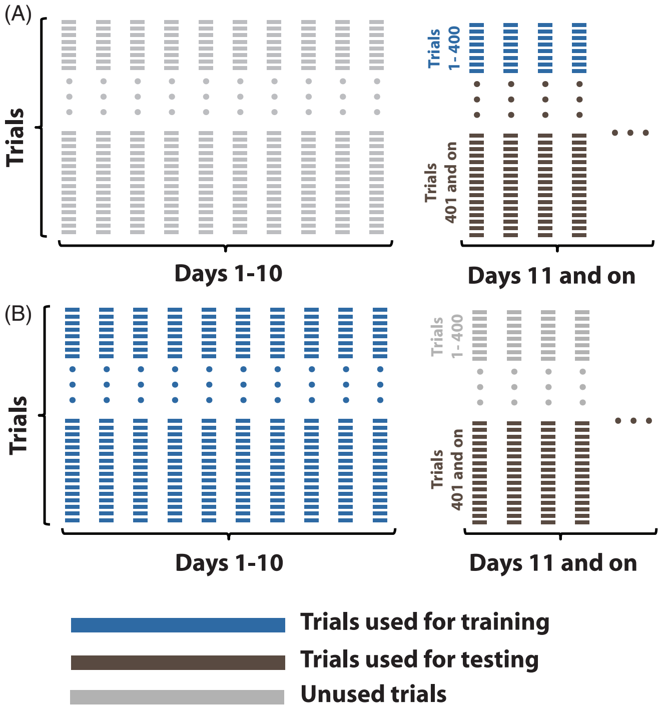

**Figure 7.** Partitioning of data when testing classifiers with (A) and without (B) daily retraining as described in section 2. Each column of rectangles represents an experimental day of collected data, and each rectangle represents a trial.

shown in figures 5(E) and 6(C). If tuning parameters drift in an unconstrained manner, the average of the unsigned $\Delta_{d,e,j}$ values should monotonically increase with time. However, average values do not appear to systematically increase with time, supporting the conclusion that tuning parameters can drift from day to day but do so in a constrained manner.

### *3.2. Evaluation of classifier stability with prerecorded data*

We compare the performance of the *SR*, $SR_S$, standard retrained and standard non-retrained classifiers on the two prerecorded datasets described in section 2.1. The large number of days in each makes these ideal datasets for evaluating classifier stability.

The standard retrained classifier is trained fresh each day using the first 400 trials of the day. After training, the classifier is used to predict reach direction on each of the remaining trials of the day. Importantly, while true reach direction is known for each trial of the datasets and is used for training, this information is not made available to the classifier when predicting reach direction (testing) and is only used after a prediction is made to evaluate classifier accuracy.

The *SR*, $SR_S$ and standard non-retrained classifiers must be initially trained on a small number of days where neural activity is recorded and the desired reach direction for trials is known. After this they require no further training when classifying reach direction on days when only neural activity is available. For this analysis, all trials on the first ten days of each dataset are used for training these classifiers. After this training, the classifiers are used to classify reach direction for trials on the remaining days of each dataset. To allow a fair head-to-head comparison with the standard retrained classifier, classification on each test day for these classifiers begins at trial 401 of each day. In this way, the set of trials that each classifier predicts reach direction for is the same on any given day. The breakup of data for training and testing each of the classifiers is shown in figure 7(B).

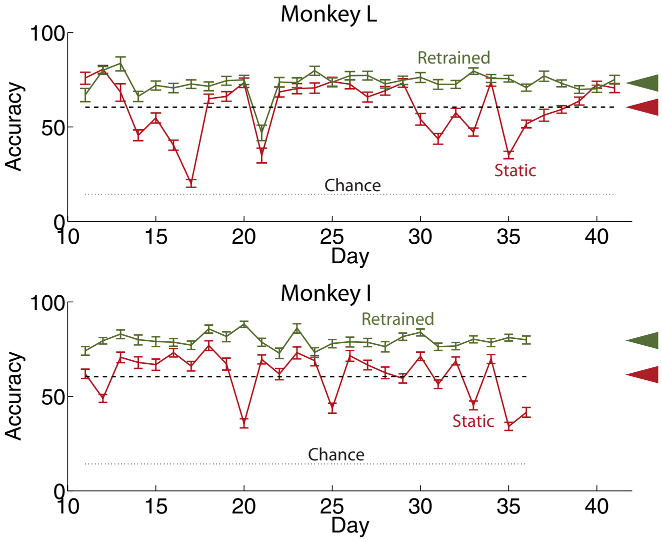

**Figure 8.** Daily accuracy values for the standard non-retrained and retrained classifiers for both subjects with 95% confidence intervals. The retrained classifier was retrained each day while the non-retrained classifier was trained on days preceding those decoded here and then held fixed. See figure 7 for a depiction of the partitioning of data when training and testing each classifier. The dashed line in each panel shows a linear fit of daily accuracy of the non-retrained classifier with test day. The slope of this fit is not significantly different from zero for either subject. Arrows on the right show overall accuracies of each classifier.

When training all classifiers, any electrode with an average of less than two spike counts per trial on the training data (the first 400 trials of each day for the standard retrained classifier and all trials on days 1–10 for the standard non-retrained, *SR* and $SR_S$ classifiers) is not used when subsequently decoding reach direction on test data. This step prevents these electrodes from causing problems if their average spike counts were to rise at a later point in time when performing classification. Critically, this can be done in clinical practice as the criterion for exclusion is based only on training data.

### *3.3. Performance of the standard classifier without daily retraining*

We now examine the implications of tuning parameter drift on the standard non-retrained classifier. The performance of this classifier indicates what might happen if we simply used the standard classifier without retraining and did nothing to account for drifting tuning parameters. Of course, this is an offline analysis and subjects may be able to adapt their neural signals through cognitive strategies in a closed-loop setting to improve BCI performance (Chase *et al* 2009), but the results presented here may still be considered as a type of lower bound on performance without retraining.

Examination of figure 8, which displays the average daily classification accuracy of the standard non-retrained

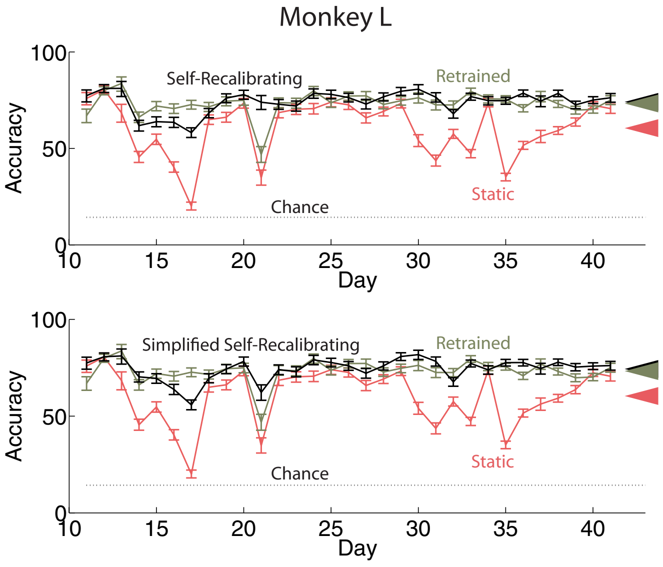

**Figure 9.** Daily performance of the self-recalibrating (top panel) and simplified self-recalibrating (bottom panel) classifiers with 95% confidence intervals for data from monkey L. Arrows on the right show overall accuracies of each classifier. Results for the standard classifier with and without daily retraining are reproduced from figure 8 for reference.

classifier for both subjects, reveals two important findings. First, performance of the standard non-retrained classifier is substantially lower than that of the daily retrained classifier on most days. This motivates the development of classifiers which account for drifting tuning parameters. However, classification accuracy of the standard non-retrained classifier does not significantly decline with time, which constitutes the second finding. To quantify this claim, the slope of a linear fit to daily classification accuracy is not significantly different than zero for both subjects. Specifically, the 95% confidence interval for the slope, measured in units of percent accuracy per day, is $(-0.006, 0.006)$ for monkey L and $(-0.012, 0.001)$ for monkey I, which contains zero for both monkeys. While this finding might at first seem surprising, it is consistent with the finding above that tuning parameters do not show a systematic tendency to progressively wander from their values on day 1 but seem to vary within a relatively limited range.

### *3.4 Performance of the self-recalibrating classifiers*

Figures 9 and 10 show the daily classification accuracies of the *SR* and $SR_S$ classifiers for each subject. We emphasize that after these classifiers were trained on days 1–10 for a dataset, they were not provided with any information about the true reach direction of trials with which to retrain themselves when predicting reach direction on the following days. Instead, using the techniques described in section 2, the *SR* and $SR_S$ classifiers simultaneously predicted reach direction and the daily value of tuning parameters from neural data alone. Therefore, the classification accuracies reported here are indicative of what might be achieved in the clinic without any further daily supervised retraining or recalibration after the system is initially trained.

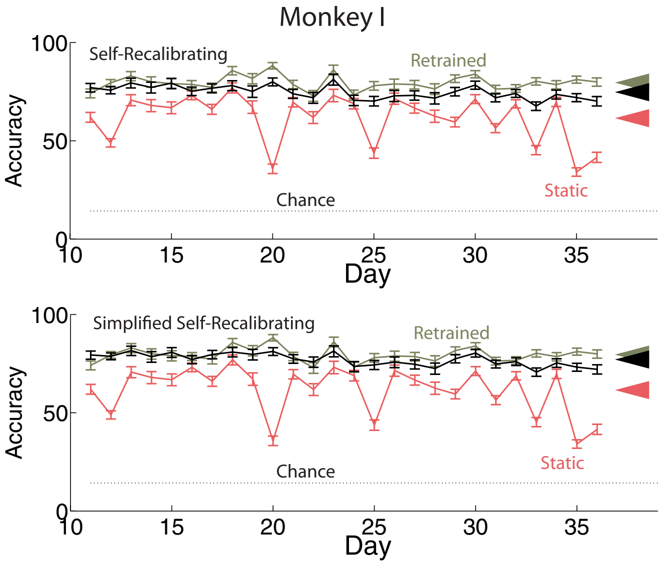

**Figure 10.** Daily performance of the self-recalibrating (top panel) and simplified self-recalibrating (bottom panel) classifiers with 95% confidence intervals for data from monkey I. Arrows on the right show overall accuracies of each classifier. Results for the standard classifier with and without daily retraining are reproduced from figure 8 for reference.

**Table 1.** Average daily accuracy with 95% confidence intervals for the non-retrained and retrained standard classifiers and both self-recalibrating classifiers for both subjects.

| Classifier | Monkey L | Monkey I |
|---|---|---|
| Non-retrained | 60% ± 5% | 62% ± 5% |
| Retrained | 73% ± 2% | 80% ± 2% |
| Self-recalibrating | 74% ± 2% | 75% ± 1% |
| Simplified self-recalibrating | 74% ± 2% | 77% ± 1% |

Table 1 compares the overall accuracies, calculated as the mean of the average daily accuracies, for all four classifiers considered in this work. The overall accuracies of the *SR* and $SR_S$ classifiers were both 14% higher than that of the standard non-retrained classifier for monkey L and 13% and 16%, respectively, higher for monkey I. These performance improvements were statistically significant for both self-recalibrating classifiers and both subjects (One Sided Wilcoxon Signed-Rank Test, $p < 10^{-6}$ for both classifiers for monkey L, $p < 10^{-5}$ for both classifiers for monkey I). Further, both self-recalibrating classifiers nearly recovered the accuracy achieved by the standard retrained classifier. The overall accuracy of the *SR* and the $SR_S$ classifiers were both 1% higher than that of the retrained classifier for monkey L and only 5% and 3% lower for monkey I. These differences were not statistically significant for either self-recalibrating classifier and monkey L (One Sided Wilcoxon Signed-Rank Test, $p > 0.1$ for *SR* classifier and $p > 0.05$ for the $SR_S$ classifier), while the small decrease in accuracies observed on the data for monkey I were (One Sided Wilcoxon Signed-Rank Test, $p < 10^{-4}$ for *SR* classifier and $p < 10^{-2}$ for the $SR_S$ classifier). To be clear, the self-recalibrating classifiers decoded the exact same trials, trials 401 and on for each test day, as the retrained classifier and achieved this performance without the benefit that the retrained classifier had of being

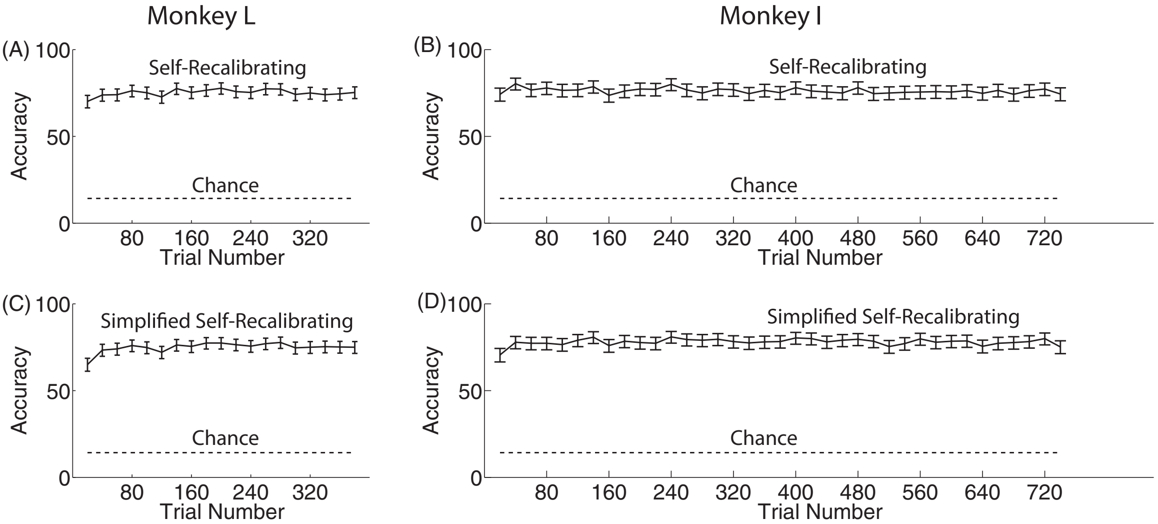

**Figure 11.** Average accuracy throughout a decoding session for the *SR* and $SR_S$ classifiers for both subjects with 95% confidence intervals. Accuracy values are calculated as the average classification accuracy in sequential bins of 20 trials over all test days for a subject. There were 386 or more trials for classification on each test day for monkey L and 752 or more on each test day for monkey I. Panels (A) and (C) show results for monkey L, while panels (B) and (D) show results for monkey I.

retrained on the fully labeled data of the first 400 trials of each day.

An important consideration when evaluating the performance of a self-recalibrating classifier is the number of trials required to learn tuning parameter values. A self-recalibrating classifier is of little use if its performance starts near chance each day and then takes hundreds of trials to reach acceptable prediction accuracy. Figure 11 shows the average classification accuracy achieved by each of the self-recalibrating classifiers throughout the course of a decoding day for both subjects. These accuracies were calculated by dividing all of the trials decoded during a test day into sequential bins of 20 trials each and then calculating the average prediction accuracy in each of these bins over all days of data for a subject. The number of trials in each day varied, and the number of bins that accuracies could be calculated over was limited by the day with the fewest number of trials for each subject. Monkey L had at least 386 trials for classification on each test day and monkey I had at least 752. Examination of the plots in figure 11 reveal that both classifiers start a decoding day well above chance levels of performance. Further, the *SR* classifier shows little evidence of a 'burn-in' period, where accuracy might be lower as tuning parameter values are being learned. It is likely that if there were more days of data present and smaller bin sizes could be used in the analysis, a burn-in period of less than 20 trials might become apparent, but in any case the *SR* classifier rapidly achieves steady state performance. The $SR_S$ classifier shows evidence of slightly decreased performance in the first bin of 20 trials for both subjects, and performance appears to be at asymptotic levels by trials 21–40.

Figure 12 displays tuning parameter estimates on two electrodes produced by both self-recalibrating classifiers on the first five days of test data for monkey L. Results for monkey I were similar. For this analysis, estimates for true tuning parameters were obtained by convolved spikes with a Gaussian kernel with a standard deviation of 100 trials as described in section 2.3 with one exception: days were processed separately and not concatenated together. The *SR* classifier produces a distribution over tuning parameter values on each trial (shown as the distributions over $\mu_{d,e,j=1}$ and $\mu_{d,e,j=2}$ in figure 2), and the mean of these distributions, $\mathbb{E}[\mu_{d,e,j}|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t}]$, was used to plot the traces in figure 12. Even though there are seven tuning parameters per electrode, traces for only two tuning parameters per electrode are shown for visual clarity. Characteristics of the way both classifiers predict tuning parameters become apparent with inspection of the plots in figure 12. First, on all days estimates for tuning parameters on the same electrode move up and down together. This immediately follows from the modeling assumption that tuning parameters on a given electrode are yoked to a single base value, $b_{d,e}$, in equation (5). Second, tuning parameter estimates vary more strongly at the beginning than at the end of a decoding session. This can be observed in the steep change in tuning parameter values at the beginning of certain days, such as day 14 for electrode 17 in figure 12. Intuitively, the reason for the reduced variability in tuning parameter estimates for both decoders is that confidence in estimates of base values increases as more neural activity is observed. Mathematically, this corresponds to less uncertainty in the distribution of base values, $P(\boldsymbol{b}_d|\{\boldsymbol{x}_{d,s}\}_{s=1}^{t})$, for the *SR* classifier and an increasing value of $n_{d,t}$ for the $SR_S$ classifier.

The 'spikes' in the parameter estimates for an electrode produced by the *SR* classifier visible in both figures occur when that electrode is flagged as behaving erratically by the mechanism for detecting anomalous spike counts. When this occurs, the uncertainty around the base value for that electrode is increased. This allows spike counts collected immediately after an electrode is flagged to strongly influence parameter estimates, and just as at the beginning of a day, multiple trials have to pass before the uncertainty for the base value of a flagged electrode is reduced and parameter estimates vary less. The reader may notice that the plotted electrodes are flagged

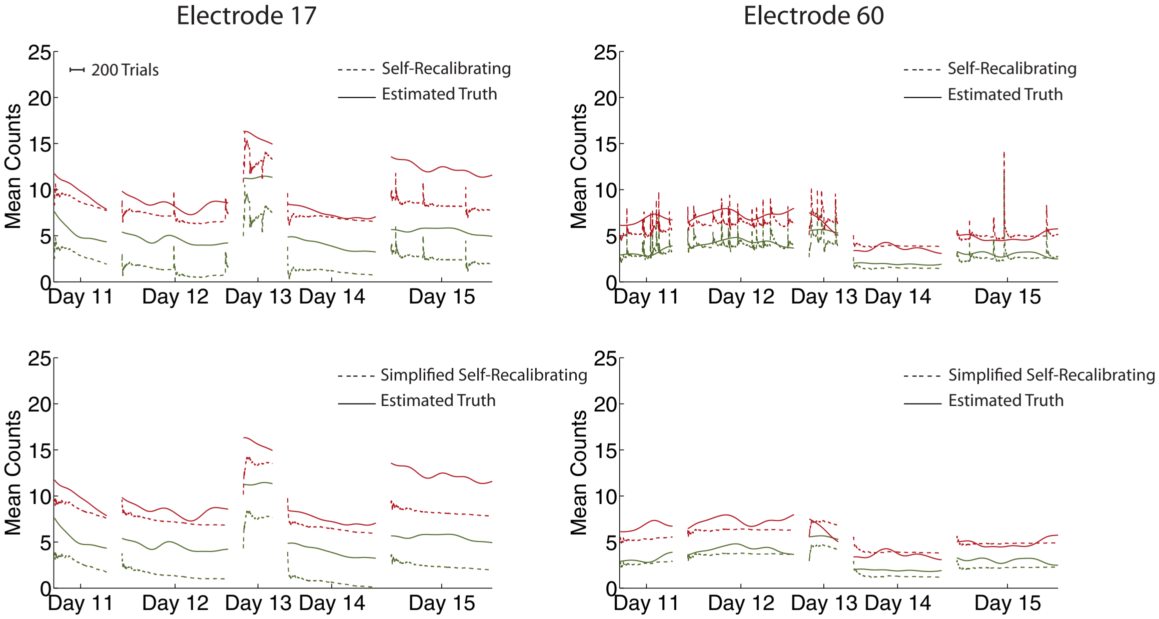

**Figure 12.** Estimates of tuning parameters on two representative electrodes (left and right columns) produced by the *SR* (top) and $SR_S$ (bottom) classifiers on the first five days of test data for monkey L compared to smoothed estimates obtained by convolving spikes with a Gaussian kernel. The 'spikes' in the tuning parameter estimates of the *SR* classifier occur when the mechanism for detecting anomalous spike counts is triggered and the uncertainty around the parameter estimates for an electrode is reset. The false positive rate for detecting anomalous spikes was set to approximately 1%, and the rate of spikes seen in this plot should not be understood as the true rate of anomalous events.

as behaving erratically multiple times within the same day. As explained in section 2.5.3, the false positive rate of the flagging mechanism was set to 1%; therefore, the frequency of parameter resets seen in figure 12 should not be understood to indicate the true frequency of anomalous events. While we do not show results for this, in practice the *SR* classifier appears to be reasonably robust without this mechanism to detect and compensate for electrodes with anomalous spike counts. In fact, classification accuracy of the *SR* classifier was markedly improved on only three days for monkey L and on no days for monkey I by its inclusion.

## 4. Discussion

The development of self-recalibrating classifiers for intracortical BCI has received little previous attention. In this work, we first characterized properties of tuning parameter drift which can be leveraged for the design of self-recalibrating classifiers. We provided evidence that between-day tuning parameter drift is substantially larger than within-day drift and that tuning parameters on the same electrode drift in a highly correlated manner. Further, tuning parameter drift appears constrained across time. This is consistent with the performance of a non-retrained classifier, which is unstable from day-to-day but does not tend to decline with time. We developed two novel self-recalibrating classifiers which leverage the insights gained into tuning parameter drift to maintain stable performance over periods of 31 and 26 days of prerecorded data. This results in an approximate 15% improvement in overall accuracy compared to the standard classifier without daily retraining. This improvement was large enough that both self-recalibrating classifiers nearly recovered the accuracy achieved by the standard classifier when it was retrained daily.

The self-recalibrating classifiers presented in this work train themselves daily without knowledge of true reach directions. This is the first report of such classifiers for intracortical BCI to the authors' knowledge. While these results must be verified in a closed-loop setting, they may represent a significant step toward the clinical viability of intracortical BCI systems. Indeed, future systems patterned off of the ideas in this paper would be ready for use almost immediately after being powered on each day as threshold levels can be set via an automated method and the classifiers update their parameters autonomously in the background during normal BCI use.

The classifiers presented here require foreknowledge of the window in time during which a user is making a decision. Previous work has demonstrated the utility of BCI systems subject to this constraint (Vaughan *et al* 2006). Further, self-paced BCI systems can be developed by combining the self-recalibrating classifiers presented here with previously published algorithms to determine at each point in time whether the user is intending to make a decision or not (Achtman *et al* 2007, Kemere *et al* 2008).

The two self-recalibrating classifiers performed very similarly. Is one to be preferred over the other for clinical application? The $SR_S$ classifier adds little computational cost to the standard BCI classifier. This has important implications for resource constrained systems, such as implantable medical devices. Computational hardware which supports the standard classifier will also likely support an upgrade to the $SR_S$ classifier. However, the *SR* classifier may still be preferred in two scenarios. First, as shown in figure 11, the *SR* classifier requires fewer trials to reach stable levels of performance at the beginning of a decoding day. Second, the $SR_S$ classifier implicitly assumes that a user makes reaches in all directions with equal frequency. If this assumption is broken, which may occur if a user reaches predominantly in one direction, tuning parameter estimation will suffer. The *SR* classifier bypasses this shortcoming by calculating the probability of reaching in each direction while estimating tuning parameters for each trial. In addition to these considerations, the *SR* classifier makes modeling assumptions explicit. Changes to these assumptions can be carried out in a straight forward manner with the *SR* classifier, and it not obvious how this might be done with the $SR_S$ classifier.

Tuning parameter estimates produced by both self-recalibrating classifiers could differ significantly from those estimated as ground truth. However, this did not prevent either classifier from nearly reproducing the decoding accuracy of the daily retrained standard classifier. While this may seem surprising, it is likely classification accuracy is robust to small deviations in estimated tuning parameters as long as their relative ordering is maintained. Our models ensure this by referencing all tuning parameters on an electrode to a single base value. Preliminary analyses with more complicated models which did not force tuning parameters on the same electrode to move in perfect lockstep and also modeled drift in tuning parameters within days did not reveal markedly improved decoding performance (data not shown). Therefore, as the ultimate goal of this work is stable classification and not tuning parameter estimation, we believe the simpler models are to be preferred.

Our decoders also do not carry information about tuning parameters from one decoding day to the next. However, this does not significantly hinder decoding performance. This may seem surprising at first as figure 5(A) indicates that on some pairs of days the previous day's tuning parameter values may be predictive of the next day's values. However, we note that this is not the case for all days. In figure 5(A), for example, there is a large jump in the value of parameters between days 4 and 5. Thus, on same days using information about the decoding parameters from the previous day would likely help the decoder learn the correct tuning parameters for the present day more quickly, and on some others it could hinder learning. Additionally, as shown in figure 11, both decoders learn tuning parameter values quickly enough to allow average decoding performance to reach asymptotic levels in tens of trials. Thus, the improvement in decoding accuracy that could be realized by learning tuning parameters more quickly is fairly minimal. Finally, formally modeling a time series of base values between days would make our EM algorithm for the *SR* classifier more complicated as the distribution of base values conditioned on observed data would no longer factor as a product of the distribution of base values on individual days (see equation S1 of the Supplementary Methods, available at stacks.iop.org/JNE/11/026001/mmedia), and additional parameters for a model of transitions in base values between days would need to be learned. Though these reasons motivate the simplification to not carry tuning parameter information between days, we acknowledge this may be interesting future work.

An important aim of this work was to identify properties of the drift in intracortical BCI signals that could be exploited for the development of simplified classifiers. The finding that tuning parameters on the same electrode drift in a highly correlated manner reduces the dimensionality of the space that tuning parameters must be tracked in as algorithms can track a single base value instead of all individual parameters on an electrode. High correlations in the drift of tuning parameters on the same electrode may be consistent with changes in baseline firing rates. Changes in baseline could be driven by changes in recorded action potential amplitude (Chestek *et al* 2011), micromovements of the recording array relative to cortical tissue (Santhanam *et al* 2007) and daily fluctuations in the value of the threshold that was used for spike detection. Whatever the mechanisms which underlie this behavior, robust classification was achieved by the relatively simple $SR_S$ classifier. The success of this simple approach is tantalizing because it suggests that any BCI algorithm which can be parameterized by variables which are referenced to the average spike counts of electrodes could be made self-recalibrating by following a recipe similar to that presented in section 2.6 for the $SR_S$ classifier. While we caution below results that pertain to adaptive classifiers should not be trivially generalized to that of regressors, it could be productive to apply this same modification to standard algorithms for regression, such as the Kalman filter (Wu *et al* 2006, Wu and Hatsopoulos 2008) or the population vector algorithm (Georgopoulos *et al* 1986, 1988) to render them self-recalibrating.

Another important finding was that tuning parameters drift much more between than within days. This suggests that self-recalibrating classifiers can be simplified to capture only between-day changes, as was done in this work. Recently Perge *et al* (2013) investigated intracortical BCI signal stability within the context of decoding continuous control signals (i.e., regression) in closed-loop human experiments. While they do not explicitly compare within-day changes to between-day changes in BCI signals, they do report reduced decoding accuracy due to within-day changes in neural spike rates. The differing conclusions may be explained by the relative sensitivity of regressors and classifiers to changes in BCI signals. Even small signal changes in BCI signals will likely be reflected in the decoded output of a regressor, whereas the performance of a classifier will only be affected if the relative ordering of the probabilities assigned to each class (reach direction in this work) are altered. This should lend caution when applying our findings on tuning parameter drift to the continuous decoding scenario. While tuning parameters do move up and down together between days on the same electrode, other aspects of tuning, such as modulation depth, as shown in *Supplementary Results* (available at stacks.iop.org/JNE/11/026001/mmedia), can also change. For the purposes of developing self-recalibrating classifiers for our datasets, we were able to simplify our classifiers and ignore these other aspects of drift. However, it may be that these simplifications would be more detrimental in the context of regression. This is not to degrade the importance of the present study. As discussed in the introduction, BCI classifiers offer unique and important clinical promise, and insights that allow for the unique simplification of BCI classifiers may facilitate their use in non-laboratory settings.

Recently Flint *et al* (2013) examined BCI performance of a non-retrained continuous decoder of arm movements in a closed-loop setting when either local field potentials or threshold crossings were used for control signals. We focus on their results for threshold crossings. Two findings of their work are of direct relevance to the present study. First, in an offline analysis of hand-control data, they report highly variable day-to-day performance but little change in average performance with time as decoders age after the initial day they were trained on. This is consistent with our similar finding for a discrete decoder. Second, when they measure correlation coefficients between a variety of measures of closed-loop decoder performance and time, they show non-degrading performance over time for one subject and a mild degradation of performance for a second. Beyond these overall trends, day-to-day variability in performance is still apparent. While this may be partly due to variability in subject motivation, it is also quite possible that the use of self-recalibrating decoders could have provided reduced day-to-day variability in performance in this closed-loop setting.

There are at least two possible reasons tuning parameter drift is constrained across time, as we observed in this work. First, it may be the case that tuning parameters are fixed across days but our estimates varied from day to day due to fluctuations in threshold settings based on a finite sample (2 s) of data. However, this likely does not entirely explain the changes in tuning parameters that we observe because the relationship between multiunit activity and arm kinematics can change even when threshold settings are held fixed across days (Flint *et al* 2013). A second tantalizing possibility is that micromovements of the recording array might underlie such tuning parameter changes so that electrodes are in a random position within a relatively stable neural environment. If this is the case, the development of recording arrays that maintain a more fixed position relative to brain tissue may hold promise for increasing the fundamental stability of signals for intracortical BCI use.

## Acknowledgments

The authors would like to thank the numerous autonomous reviewers for their helpful evaluations and critiques of the different versions of this manuscript.

This work was supported by National Science Foundation graduate research fellowships (CCC, VG), NDSEG fellowships (VG, WB), Burroughs Wellcome Fund Career Awards in the Biomedical Sciences (KVS), the Christopher Reeve Paralysis Foundation (KVS), a Stanford University Graduate Fellowship (CCC), a Texas Instruments Stanford Graduate Fellowship (JDF), a Stanford NIH Medical Scientist Training Program grant and Soros Fellowship (PN), NIH grant T90 DA022762 (WB), NIH grant R90 DA023426-06 (WB) and the following awards to KVS: Stanford CIS, Sloan Foundation, NIH CRCNS R01-NS054283, McKnight Endowment Fund for Neuroscience, NIH Director's Pioneer Award 1DP1OD006409, DARPA Revolutionizing Prosthetics program contract N66001-06-C-8005, and DARPA REPAIR contract N66001-10-C-2010.

## References

Achtman N, Afshar A, Santhanam G, Byron M Y, Ryu S I and Shenoy K V 2007 Free-paced high-performance brain–computer interfaces *J. Neural Eng.* **4** 336–47

Aggarwal V, Acharya S, Tenore F, Shin H-C, Etienne-Cummings R, Schieber M H and Thakor N V 2008 Asynchronous decoding of dexterous finger movements using M1 neurons *IEEE Trans. Neural Syst. Rehabil. Eng.* **16** 3–14

Blom H and Bar-Shalom Y 1988 The interacting multiple model algorithm for systems with Markovian switching coefficients *IEEE Trans. Autom. Control* **33** 780–3

Blumberg J, Rickert J, Waldert S, Schulze-Bonhage A, Aertsen A and Mehring C 2007 Adaptive classification for brain computer interfaces *Proc. 29th Annual Int. Conf. IEEE EMBS (Lyon, France)* pp 2536–9

Brunner P, Ritaccio A, Emrich J, Bischof H and Schalk G 2011 Rapid communication with a 'p300' matrix speller using electrocorticographic signals (ecog) *Front. Neurosci.* **5** 5

Chase S M, Kass R E and Schwartz A B 2012 Behavioral and neural correlates of visuomotor adaptation observed through a brain-computer interface in primary motor cortex *J. Neurophysiol.* **108** 624–44

Chase S M, Schwartz A B and Kass R E 2009 Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain–computer interface algorithms *Neural Netw.* **22** 1203–13

Chestek C, Batista A, Santhanam G, Yu B, Afshar A, Cunningham J, Gilja V, Ryu S, Churchland M and Shenoy K 2007 Single-neuron stability during repeated reaching in macaque premotor cortex *J. Neurosci.* **27** 10742–50

Chestek C *et al* 2011 Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex *J. Neural Eng.* **8** 045005

Chestek C A, Cunningham J P, Gilja V, Nuyujukian P, Ryu S I and Shenoy K V 2009 Neural prosthetic systems: current problems and future directions *Proc. 31st Annual Int. Conf. IEEE EMBS (Minneapolis, MN)* pp 3369–75

Chestek C A, Gilja V, Blabe C H, Foster B L, Shenoy K V, Parvizi J and Henderson J M 2013 Hand posture classification using electrocorticography signals in the gamma band over human sensorimotor brain areas *J. Neural Eng.* **10** 026002

Collinger J L, Wodlinger B, Downey J E, Wang W, Tyler-Kabara E C, Weber D J, McMorland A J, Velliste M, Boninger M L and Schwartz A B 2012 High-performance neuroprosthetic control by an individual with tetraplegia *Lancet* **381** 557–64

Dempster A P, Laird N M and Rubin D B 1977 Maximum likelihood from incomplete data via the em algorithm *J. R. Stat. Soc.* B **39** 1–38

Eden U, Truccolo W, Fellows M, Donoghue J and Brown E 2004a Reconstruction of hand movement trajectories from a dynamic ensemble of spiking motor cortical neurons *Proc. 26th Annual Int. Conf. IEEE EMBS (San Francisco, CA)* pp 4017–20

Eden U T, Frank L M, Barbieri R, Solo V and Brown E N 2004b Dynamic analysis of neural encoding by point process adaptive filtering *Neural Comput.* **16** 971–98

Flint R D, Wright Z A, Scheid M R and Slutzky M W 2013 Long term, stable brain machine interface performance using local field potentials and multiunit spikes *J. Neural Eng.* **10** 056005

Fraser G, Chase S, Whitford A and Schwartz A 2009 Control of a brain–computer interface without spike sorting *J. Neural Eng.* **6** 055004

Galán F, Nuttin M, Lew E, Ferrez P W, Vanacker G, Philips J and Millán J d R 2008 A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots *Clin. Neurophysiol.* **119** 2159–69

Ganguly K and Carmena J M 2009 Emergence of a stable cortical map for neuroprosthetic control *PLoS Biol.* **7** e1000153

Georgopoulos A P, Kettner R E and Schwartz A B 1988 Primate motor cortex and free arm movements to visual targets in three-dimensional space: ii. Coding of the direction of movement by a neuronal population *J. Neurosci.* **8** 2928–37

Georgopoulos A P *et al* 1986 Neuronal population coding of movement direction *Science* **233** 1416–9

Gilja V *et al* 2012 A high-performance neural prosthesis enabled by control algorithm design *Nature Neurosci.* **15** 1752–7

Hochberg L *et al* 2012 Reach and grasp by people with tetraplegia using a neurally controlled robotic arm *Nature* **485** 372–5

Jarosiewicz B, Chase S M, Fraser G W, Velliste M, Kass R E and Schwartz A B 2008 Functional network reorganization during learning in a brain–computer interface paradigm *Proc. Natl Acad. Sci.* **105** 19486–91

Kemere C, Santhanam G, Yu B M, Afshar A, Ryu S, Meng T and Shenoy K 2008 Detecting neural-state transitions using hidden Markov models for motor cortical prostheses *J. Neurophysiol.* **100** 2441–52

Leeb R, Friedman D, Müller-Putz G, Scherer R, Slater M and Pfurtscheller G 2007 Self-paced (asynchronous) bci control of a wheelchair in virtual environments: a case study with a tetraplegic *Comput. Intell. Neurosci.* **2007** 79642

Li Z, O'Doherty J E, Hanson T L, Lebedev M A, Henriquez C S and Nicolelis M A 2009 Unscented Kalman filter for brain–machine interfaces *PLoS One* **4** e6243

Li Z, O'Doherty J, Lebedev M and Nicolelis M 2011 Adaptive decoding for brain-machine interfaces through Bayesian parameter updates *Neural Comput.* **23** 3162–204

Maynard E, Hatsopoulos N, Ojakangas C, Acuna B, Sanes J, Normann R and Donoghue J 1999 Neuronal interactions improve cortical population coding of movement direction *J. Neurosci.* **19** 8083–93

Musallam S, Corneil B, Greger B, Scherberger H and Andersen R 2004 Cognitive control signals for neural prosthetics *Science* **305** 258–62

Ohnishi K, Weir R and Kuiken T 2007 Neural machine interfaces for controlling multifunctional powered upper-limb prostheses *Expert Rev. Med. Devices* **4** 43–53

Pascual J, Vidaurre C and Kawanabe M 2011 Investigating eeg non-stationarities with robust pca and its application to improve BCI performance *Int. J. Bioelectromagnetism* **13** 50–51

Perge J A, Homer M L, Malik W Q, Cash S, Eskandar E, Friehs G, Donoghue J P and Hochberg L R 2013 Intra-day signal instabilities affect decoding performance in an intracortical neural interface system *J. Neural Eng.* **10** 036004

Pistohl T, Schulze-Bonhage A, Aertsen A, Mehring C and Ball T 2012 Decoding natural grasp types from human ecog *Neuroimage* **59** 248–60

Polikov V, Tresco P and Reichert W 2005 Response of brain tissue to chronically implanted neural electrodes *J. Neurosci. Methods* **148** 1–18

Santhanam G, Linderman M, Gilja V, Afshar A, Ryu S, Meng T and Shenoy K 2007 HermesB: a continuous neural recording system for freely behaving primates *IEEE Trans. Biomed. Eng.* **54** 2037–50

Santhanam G, Ryu S I, Yu B M, Afshar A and Shenoy K V 2006 A high-performance brain–computer interface *Nature* **442** 195–8

Santhanam G, Yu B, Gilja V, Ryu S, Afshar A, Sahani M and Shenoy K 2009 Factor-analysis methods for higher-performance neural prostheses *J. Neurophysiol.* **102** 1315–30

Scherer R, Muller G, Neuper C, Graimann B and Pfurtscheller G 2004 An asynchronously controlled eeg-based virtual keyboard: improvement of the spelling rate *IEEE Trans. Biomed. Eng.* **51** 979–84

Shanechi M M, Hu R C, Powers M, Wornell G W, Brown E N and Williams Z M 2012 Neural population partitioning and a concurrent brain–machine interface for sequential motor function *Nature Neurosci.* **15** 1715–22

Shenoy K V, Meeker D, Cao S, Kureshi S A, Pesaran B, Buneo C A, Batista A P, Mitra P P, Burdick J W and Andersen R A 2003 Neural prosthetic control signals from plan activity *Neuroreport* **14** 591–6

Shenoy P, Krauledat M, Blankertz B, Rao R and Müller K 2006 Towards adaptive classification for BCI *J. Neural Eng.* **3** R13–23

Srinivasan L, Eden U, Mitter S and Brown E 2007 General-purpose filter design for neural prosthetic devices *J. Neurophysiol.* **98** 2456–75

Sykacek P and Roberts S 2003 Adaptive classification by variational Kalman filtering *Advances in Neural Information Processing Systems (NIPS)* vol 15 ed S Thrun, S Becker and K Obermayer (Cambridge, MA: MIT Press) pp 737–44

Sykacek P, Roberts S and Stokes M 2004 Adaptive BCI based on variational Bayesian Kalman filtering: an empirical evaluation *IEEE Trans. Biomed. Eng.* **51** 719–27

Taylor D M, Tillery S I H and Schwartz A B 2002 Direct cortical control of 3d neuroprosthetic devices *Science* **296** 1829–32

Vaughan T M, McFarland D J, Schalk G, Sarnacki W A, Krusienski D J, Sellers E W and Wolpaw J R 2006 The wadsworth BCI research and development program: at home with BCI *IEEE Trans. Neural Syst. Rehabil. Eng.* **14** 229–33

Velliste M, Perel S, Spalding M C, Whitford A S and Schwartz A B 2008 Cortical control of a prosthetic arm for self-feeding *Nature* **453** 1098–101

Vidaurre C and Blankertz B 2010 Towards a cure for BCI illiteracy *Brain Topography* **23** 194–8

Vidaurre C, Schlögl A, Blankertz B, Kawanabe M and Müller K 2008 Unsupervised adaptation of the lda classifier for brain–computer interfaces *Proc. 4th Int. Brain–Computer Interface Workshop and Training Course (Graz, Austria)* vol 2008 pp 122–7

Wang W *et al* 2009 Human motor cortical activity recorded with micro-ecog electrodes, during individual finger movements *Proc. 31st Annual Int. Conf. IEEE EMBS (Minneapolis, MN)* pp 586–9

Wang Z, Gunduz A, Brunner P, Ritaccio A, Ji Q and Schalk G 2012 Decoding onset and direction of movements using electrocorticographic (ecog) signals in humans *Front. Neuroeng.* **5** 15

Wu W, Gao Y, Bienenstock E, Donoghue J P and Black M J 2006 Bayesian population decoding of motor cortical activity using a Kalman filter *Neural Comput.* **18** 80–118

Wu W and Hatsopoulos N 2008 Real-time decoding of nonstationary neural activity in motor cortex *IEEE Trans. Neural Syst. Rehabil. Eng.* **16** 213–22