CLUSTERING OF PREFERRED DIRECTIONS DURING BRAIN-COMPUTER INTERFACE USAGE

by

Jeffrey Chiou

A.B. in Biology and Philosophy-Neuroscience-Psychology, Washington University, 2012

Submitted to the Graduate Faculty of

The School of Medicine in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

School of Medicine

This thesis was presented

by

Jeffrey Chiou

It was defended on

September 2, 2016

and approved by

Steven M. Chase Ph.D., Assistant Professor, Biomedical Engineering, Center for the Neural Basis of Cognition

Brent Doiron Ph.D., Associate Professor, Mathematics

Neeraj Gandhi Ph.D., Associate Professor, Otolaryngology, Neuroscience, Bioengineering, Center for the Neural Basis of Cognition

Thesis Advisor: Aaron P. Batista Ph.D., Associate Professor, Bioengineering

CLUSTERING OF PREFERRED DIRECTIONS DURING BRAIN-COMPUTER INTERFACE USAGE

Jeffrey Chiou, M.S.

University of Pittsburgh, 2016

Brain-computer interfaces (BCIs) are proving to be viable clinical interventions for sufferers of amyotrophic lateral sclerosis, amputations, and spinal cord injuries. To improve the viability of BCIs, it will help to have a thorough understanding of how the brain controls them. Neural activity during usage of certain BCIs behaves in a surprising and seemingly counterintuitive manner – the preferred directions (PDs) of neurons cluster together. We trained monkeys to reach to targets in a center-out task either using their arm or a BCI. We found that neurons' PDs cluster similarly during training of the BCI decoder and usage of the BCI, but remain relatively unclustered when the monkeys use their arms. Modulation depths increase upon usage of the BCI, and narrowness of tuning tends to either increase or decrease rather than staying the same. In addition, the cluster direction can be predicted from per-target performance. A model where two neurons' PDs approach one another reveals how much modulation depths have to increase to maintain controllability. This thesis concludes with considerations of why this clustering might occur, and whether or not it benefits BCI control.

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

I am forever grateful to my mentor and advisor, Aaron P. Batista, for his unending support and encouragement. I would like to thank all the members of my lab for their feedback, assistance, and support. Patrick Sadtler has been an inspiration and an outstanding mentor. Berook Alemayehu and Nick Pavlovsky have been amazing colleagues to work with. In addition, Jessi Mischel has been a supportive friend. Jason Godlove and Kristin Quick have given me superb advice. Emily Oby, Alan Degenhart, Erinn Grigsby, and Nicole McClain have been great to work with. Eric Moe was an excellent undergrad. He and Matthew Ginsberg played a large role in getting this project to this point.

The Center for Neuroscience at the University of Pittsburgh (CNUP) and the Center for the Neural Basis of Cognition have given me a great education. I am grateful to all my professors - in particular, to Steven Chase and Byron Yu for their wonderful classes and academic support. I will not forget the annual Christmas parties and the things I learned in the SCABBY journal club.

I dedicate this work to my wife, Angela Chen, my parents, Yi-Wen Cheng and Shean-Yih Chiou, and my mentor, Aaron Batista. Without them I would not have been able to complete this project. This work was funded by the NSF Graduate Research Fellowship.

1.0 INTRODUCTION

In 1969, Eberhard Fetz successfully trained monkeys to increase the firing rate of single neurons, based on visual feedback (Fetz, 1969). In hindsight, we recognize that this was the first brain-computer interface (BCI), in that it provided the animals with direct feedback about the state of their own neural activity. Additional studies demonstrated accurate decoding of spike trains to estimate stimuli or motor output (Bialek et al., 1991; Humphrey et al., 1970). These studies and others anticipated the later development of BCIs (Fetz and Finocchio, 1971; Schmidt et al., 1977, 1978). BCIs translate neural activity into control signals that can be used to control computer cursors, robotic arms, or other prostheses.

The operation of a BCI involves two key steps. First, a decoder must be calibrated, during what we call a training period. This is done by relating the activity of a population of neurons to the intended or actual movements of the user. Second, once calibrated, the decoder is used in real-time to allow the subject to control an external device, such as a computer cursor or a robotic limb. The population vector algorithm (PVA) is one such decoding algorithm (Georgopoulos et al., 1986, 1988). The PVA exploits the classic observation that each individual neuron in the primary motor cortex (M1) has a preferred direction (PD). Preferred direction is a concept where the neuron, during certain types of motor tasks, fires at its highest rate for movements towards a specific direction. For directions away from the preferred direction, the neuron fires less. PDs can be estimated by fitting a tuning curve to the neural activity. For example, if the motor task requires

movements to targets along a circle, the PDs found from fitted tuning curves range from 0° to 360°. A fundamental aspect of the PVA is that the PDs of the constituent neurons are uniformly distributed; indeed, without that, the algorithm is biased and inaccurate (Scott et al., 2001).

We found that PDs cluster in our own data, which use a velocity Kalman filter for decoding. This was an unexpected discovery, and is rather remarkable, considering that PVA predicts exactly the opposite. The Kalman filter is a commonly used decoding algorithm for translating neural activity (Serruya et al., 2003; Wu et al., 2006). It is an optimal recursive filter that estimates the state (velocity), given measurements (neural activity) and a linear model of the state. The Kalman filter has advantages over other decoding algorithms, such as incorporating estimates of uncertainties and modeling kinematics. It is optimal under the conditions that the system can be described with a linear model and noise is white and Gaussian.

Intuitively, one might expect that PD clustering reduces controllability of a BCI, since the neurons would be providing overlapping information to drive the BCI (Tehovnik et al., 2013). However, PD clustering does occur, seemingly without reducing performance. The goal of this study is to characterize the phenomenon of PD clustering – the conditions under which it occurs and what it might be telling us about the neural strategies of BCI control. This work is part of a larger effort to understand what happens to neural signals during BCI usage, as compared to natural arm movements.

How can good BCI performance result from a population of clustered neurons? Preferred directions are not perfectly clustered such that all PDs point in exactly the same direction. A neural mechanism such as modulation depth change may provide an avenue towards BCI controllability through amplification of signals. Notably, while decoding under a linear regression model, modulation depth tends to increase over time (Ganguly and Carmena, 2009). Under a PVA

decoder, perturbations to a subset of neurons results in those neurons reducing their modulation depth (Jarosiewicz et al., 2008). Modulation depth changes in neurons with PDs away from the population mean PD may partially compensate for clustering.

Examination of other BCI studies reveal PD clustering during BCI usage (Carmena et al., 2003; Lebedev et al., 2005; O'Doherty et al., 2011). However, rarely is the phenomenon even mentioned (Green and Kalaska, 2011; Tehovnik et al., 2013). Note that these animal studies did not use a PVA. Instead, they used linear filters or Kalman filters. We aim not only to show that PD clustering occurs, but to investigate how this effect arises. By doing so, we can better understand how the brain controls BCIs.

2.0 METHODS

2.1 EXPERIMENTAL DETAILS

Rhesus macaques (*macaca mulatta*) were trained on a center-out task. During experiments, monkeys wore gloves with LED markers attached. A Phasespace Impulse system with six cameras tracked the markers and translated the marker location to a cursor on the screen. The screen itself is a Wheatstone stereoscope capable of 3-D. However, we only used 2-D tasks.

The center-out task occurred began with a start target appearing in the middle of the screen. Our analyses covered several datasets, so the timing of target presentation varied. For the most common task setup, however, after the monkey reached to the start target, the end target would appear almost immediately (25 ms). The monkey then had to reach to the end target, typically within 5 seconds. Upon reaching the end target, the monkey had to hold the cursor there for a short time (25 ms). If completed, the monkey would receive a water reward. Otherwise, if the monkey failed, a timeout penalty (a black screen) would be imposed for 4 to 9 seconds.

The monkeys also performed reaches using a BCI. 96-channel Blackrock microarrays were implanted in primary motor cortex of three monkeys. Neural signals were acquired through a TDT pipeline, and decoded into cursor movements using a decoding algorithm, such as a velocity Kalman filter, position-velocity Kalman filter, position-velocity-acceleration Kalman filter, and an Internal Model Estimation algorithm (Golub et al., 2013). We used threshold crossings instead

of sorted spikes for decoding. The in-house experimental software runs on Labview, and a Labview Realtime computer captures incoming non-neural signals.

Our implementation of the Kalman filter (KF) uses discrete samples of spike counts (threshold crossings) taken in 45ms bins. The KF's predictions rely on a combination of the present state and the neural activity. A velocity KF (VKF) predicts velocity, measures neural activity, compares the prediction and measurements, and adjusts the velocity using the comparison. Two equations are of interest during calibration, or training of the decoder.

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{q}_k$$

This equation is a generative model of the neurons' spike counts $\mathbf{z}_k \in \Re^C$ for every time bin k, with C being the number of neurons. The state vector $\mathbf{x}_k = \left[v_x, v_y\right]_k^T$ contains the velocity in the x and y directions, and $\mathbf{q}_k \sim N(0,\mathbf{Q})$, $\mathbf{Q} \in \Re^{C\times C}$ represents the normally distributed, zero-mean noise. $\mathbf{H} \in \Re^{C\times 2}$ linearly relates the spike counts to the velocity. The second equation is a linear model of the system:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k$$

$\mathbf{A} \in \Re^{2\times 2}$ contains coefficients that describe the evolution of velocity with every time bin, and $\mathbf{w}_k \sim N(0,\mathbf{W})$, $\mathbf{W} \in \Re^{2\times 2}$ represents the noise. During training of the decoder, parameters $\mathbf{H}$, $\mathbf{A}$, $\mathbf{Q}$, and $\mathbf{W}$ are calculated using these equations:

$$\mathbf{H} = \left(\sum_{k=1}^{K} \mathbf{z}_k \mathbf{x}_k^T\right)\left(\sum_{k=1}^{K} \mathbf{x}_k \mathbf{x}_k^T\right)^{-1}$$

$$\mathbf{A} = \left(\sum_{k=2}^{K} \mathbf{x}_k \mathbf{x}_{k-1}^T\right)\left(\sum_{k=2}^{K} \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T\right)^{-1}$$

$$\mathbf{Q} = \frac{1}{K}\sum_{k=1}^{K} (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)(\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)^T$$

$$\mathbf{W} = \frac{1}{K-1}\sum_{k=2}^{K}(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T$$

During decoding, two sets of equations are used. The time update equations are as follows:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1}^-$$

$$\hat{\mathbf{P}}_k^- = \mathbf{A}\hat{\mathbf{P}}_{k-1}\mathbf{A}^T + \mathbf{W}$$

The time update, or prediction, equations are *a priori* estimates of the state and noise. The measurement update equations are as follows:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-)$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\hat{\mathbf{P}}_k^-$$

Also known as the correction equations, they update the state and noise estimates. The Kalman gain $\mathbf{K}_k$ is given by:

$$\mathbf{K}_k = \hat{\mathbf{P}}_k^-\mathbf{H}^T\left(\mathbf{H}\hat{\mathbf{P}}_k^-\mathbf{H}^T + \mathbf{Q}\right)^{-1}$$

The Kalman gain is a weighting factor taking into account the error in the estimate and the error in the measurement. Initial values of $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{P}}_k$ are the sample mean and covariance of $\mathbf{x}_1$. To run the Kalman filter, initial values are fed into the measurement update equations. Then the time update and measurement update equations are alternately run until there is no more neural activity.

Each experimental session consisted of blocks of trials. The most common block types included brain control, hand control, and observation training. Observation training refers to trials where the monkey watches a cursor move to the correct target. The decoder is then trained on the neural activity recorded during this time. For brain control trials, the monkey's arms were restrained unless otherwise specified.

Data from monkey J came from prior experiments, including a visuomotor rotation task. In this task, the monkey performed center-out reaches using hand control or brain control. During certain blocks of trials, the workspace was rotated 45° or -45°. These blocks were excluded from analysis. However, the influence of the rotated block on non-rotated blocks may have affected the analyses performed for monkey J. Data from monkey N came from when monkey N was just learning to use a BCI. For monkeys L and J, training data came from observation training, where the monkey watched the cursor automatically move to the target. Training data from monkey N came from gradual training, where the monkey gradually gained control of the cursor over time. Due to the incongruence of task and situation for monkeys J and N, most of the analyses will focus on data from monkey L.

2.2 ANALYSIS METHODS

2.2.1 Tuning Curves

For each block of trials, we fit cosine tuning curves and von Mises tuning curves to the neural activity. Firing rates were derived from spike counts between 200 ms and 550 ms after the go cue. Individual units were not spike sorted and spike counts were obtained from threshold crossings. Thus, units could be single units or multi-units. Only units with a firing rate greater than 1 spk/s were used. We fit cosine tuning curves of the form:

$$f(x) = r_0 + (r_{max} - \mathrm{r}_0)\cos(\theta - \theta_{pref})$$

Here, $f(x)$ represents the firing rate, r_0 is the baseline firing rate, $(r_{max} - r_0)$ is the modulation depth, and θ_{pref} is the preferred direction. We also fit von Mises tuning curves of the form:

$$f(x) = r_0 + (r_{max} - \mathrm{r}_0)\exp\left[\kappa\cos\left(\theta - \theta_{pref}\right)\right]$$

Here, κ is a dimensionless parameter that varies with narrowness of tuning. If $\kappa \ll 1$, the tuning function degenerates into the cosine tuning function. In general, the von Mises tuning curve has been found to provide better fits than the cosine tuning curve (Amirikian and Georgopulos, 2000; Swindale, 1998). A previous study derived equations for the y half-height (baseline) and x half-height-width (another measure of narrowness of tuning) of a von Mises tuning curve (Taubman et al., 2013). The y half-height is as follows:

$$y_{half} = r_0 + (r_{max} - r_0)\cosh(\kappa)$$

The x half-height-width is as follows:

$$2 * \cos^{-1}\left(\frac{\ln(\cosh(\kappa))}{\kappa}\right)$$

This is also known as the full width at half maximum. See Appendix A for the derivation of x half-height width.

2.2.2 Circular Statistics

The mean resultant vector is calculated from the distribution of preferred directions. If each PD is a vector of length one, the mean resultant vector will be the vector sum of all PD vectors. The magnitude of the mean resultant vector is defined as:

$$\bar{R} = \sqrt{\bar{X}^2 + \bar{Y}^2}$$

Where $\bar{X} = \frac{\sum_i^n \cos a_i}{n}$ and $\bar{Y} = \frac{\sum_i^n \sin a_i}{n}$, with n angles a.

To quantify clustering, we used circular variance and Rao's spacing test. Circular variance is defined as:

$$\bar{V} = 1 - \bar{R}$$

The circular variance ranges from zero to one, with lower values indicating greater clustering. Rao's spacing test tests the null hypothesis that preferred directions are uniformly distributed. The test compares a distribution where sample angles are evenly spaced (Rao, 1969, 1972, 1976). The Kuiper two-sample test compares two distributions and tests if they are different. We used the MATLAB toolbox CircStat to calculate circular statistics, perform Rao's spacing test and perform the Kuiper two-sample test (Berens and others, 2009).

2.2.3 Performance

The center-out task is relatively easy for the monkey to perform, so success rates were not used to assess performance. Instead, we used time to target and a couple measures of trajectory straightness. Time to target, or duration, is simply the time the go cue appeared subtracted from the time the monkey reached the end target. We used the straightness index (Batschelet, 1981) and angular dispersion (Estevez and Christman, 2006; Fisher, 1995; Miller et al., 2011) to measure straightness of trajectory. The straightness index is defined as the ideal straightest distance (ΔD) divided by the actual distance of the trajectory (L).

$$SI = \frac{\Delta D}{L}$$

Angular dispersion is given by

$$AD = \frac{\sqrt{C^2 + S^2}}{n}$$

Where $C = \sum\cos(\theta_i)$ and $S = \sum\sin(\theta_i)$. $\{\theta_i\}_{i=1}^{n}$is the set of successive turning angles for each position in the trajectory. Another way to interpret angular dispersion is as the mean resultant vector of the turning angles. Angular dispersion and straightness index vary from 0 to 1, with higher values indicating straighter trajectories.

2.2.4 Circular-Linear Regression

To find the relationship between cluster direction and target performance measures, we used circular-linear regression. Circular-linear regression fits a model:

$$\theta = \mu + 2 * \text{atan}(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)$$

where θ is the circular response variable, $x_1 \ldots x_n$ are the linear predictor variables, $\beta_1 \ldots \beta_n$ are the regression coefficients. Circular-linear regression assumes the circular response variable has a von Mises distribution with a mean μ (Fisher and Lee, 1992).

3.0 RESULTS

3.1 CLUSTERING OCCURS DURING TRAINING AND BRAIN CONTROL

Throughout this work, we will use tuning curve properties calculated from von Mises fits. Figure 1 shows the tuning curve fits and properties of a single unit. The von Mises R^2 is higher than the cosine R^2, suggesting that a von Mises tuning curve better describes this unit's behavior than a cosine tuning curve. Preferred direction is the peak of the tuning curve. The modulation depth and half-height width are calculated as specified in the Methods.

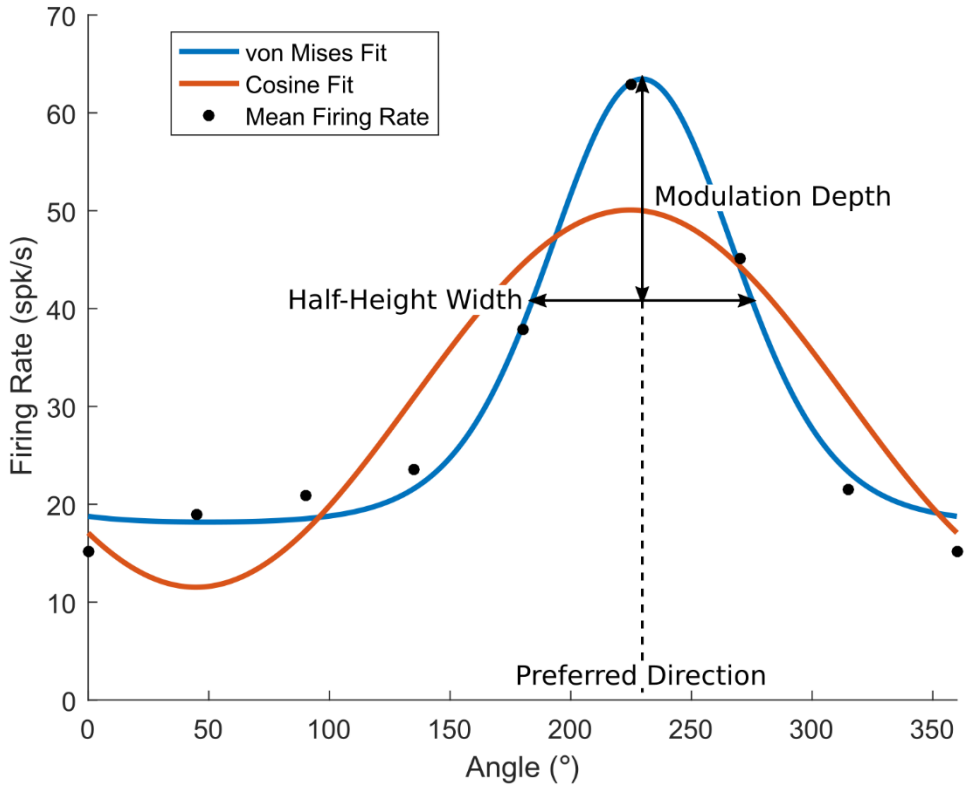

Figure 1. Tuning curve fits and properties of a single unit. Mean firing rates for each of 8 targets are shown as black dots. Von Mises (R^2=0.707) and cosine tuning curves (R^2=0.561) are fit to all data available using nonlinear least squares. Modulation depth (22.6 spk/s), full width at half maximum (half-height width, 45.5°), and preferred direction (229.6°) are indicated on the plot.

We found that, across three different monkeys (J, L, and N), with few exceptions, a high degree of clustering occurs during both brain control and training blocks, but not during natural reaching. Note that training refers to the BCI calibration period. We used circular variance to quantify the amount of clustering and Rao's spacing test to rule out the null hypothesis of uniformly distributed preferred directions. The percentage of datasets resulting in rejection of the null hypothesis is used to compare hand control, brain control, and training. Monkey N learned the BCI control task more slowly than did J and L, and the clustering during training was not as

apparent in early datasets. Monkey J did not show clustering during earlier datasets (mean circular variance = 0.801, SD = 0.090, Rao p-value <= 0.01 = 0%). However, the monkey's arm was unrestrained, and position-velocity and position-velocity-acceleration KFs were used. These data were excluded from primary analyses due to the difference in decoding algorithm and lack of arm restraint.

Figure 2 comes from an experimental session with monkey L. The angle distributions of individual blocks in Figure 2A show some notable trends. In brain control and training blocks, the neurons with preferred directions closest to the cluster center have lower modulation depths than the neurons further away from the cluster center. Furthermore, the angle distribution between brain control and training are similar. In this example dataset, as well as most others in monkeys J and L, training, brain control, and hand control all occurred during one experimental session.

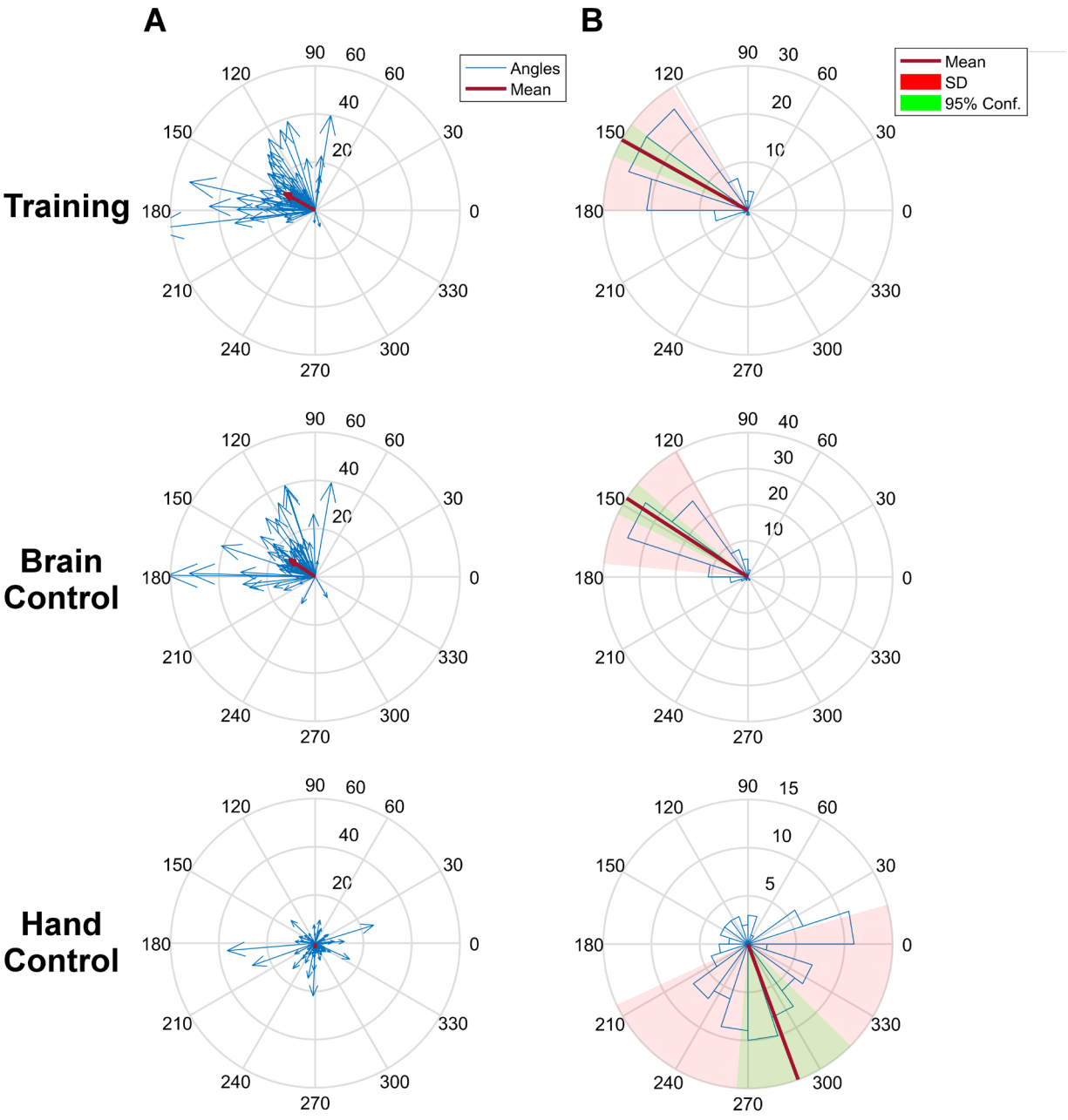

Figure 2. Preferred direction distributions show clustering during BCI calibration and control, but not during hand control. Data come from a single experimental session. **(A)** Angle distributions with each unit's preferred direction depicted by the blue arrow direction. The arrows are scaled in length by modulation depth. The red arrow shows the mean preferred direction, taking scaling into account. **(B)** Angle histograms with 20 bins each. The angle of the mean resultant vector is shown in red, 95% confidence intervals in green, and standard deviation in

light red. Preferred directions are determined from spike counts in a window between 200 ms and 550 ms after the go cue. Modulation depths are determined from von Mises fits.

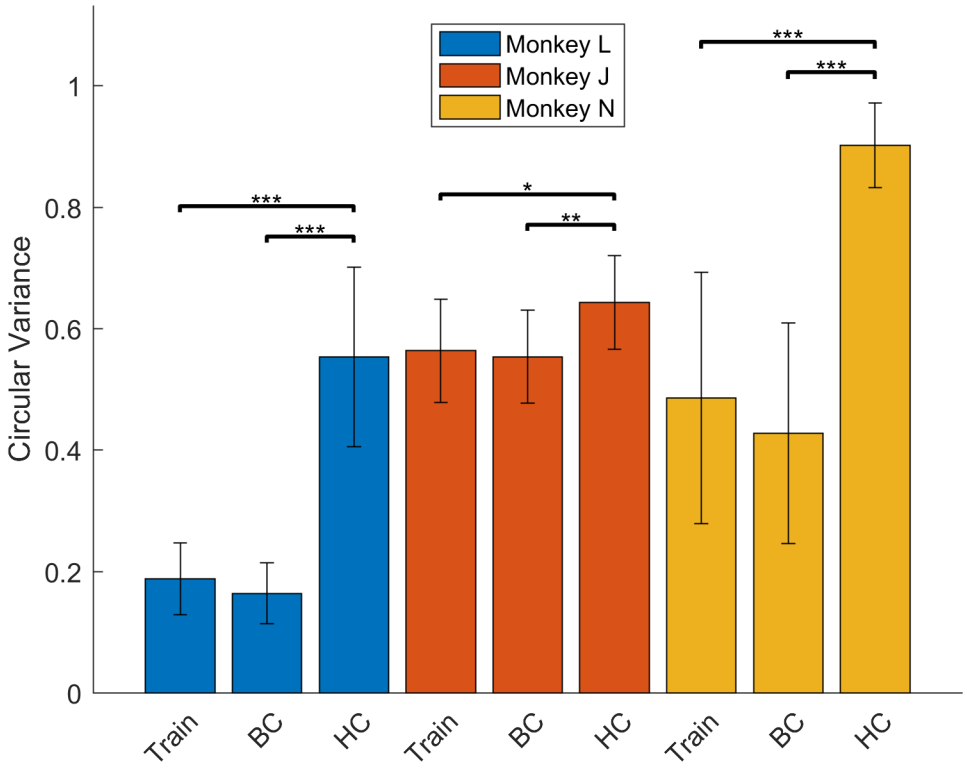

Figure 3. Circular variance is always lower during training and brain control, compared to hand control, across three monkeys. "Train" indicates the period during which the decoder was calibrated. "BC" represents brain control and "HC" represents hand control. Across all three monkeys, hand control had significantly higher circular variance than brain control and training. Error bars depict standard deviation. Number of stars indicates degree of significance as determined by the Mann-Whitney U test. One star represents $p <= 0.5$. Two stars represents $p <= 0.01$. Three stars represents $p <= 0.001$. Each bar represents several blocks of experiments. For Monkey L, n = 24 for training, n = 21 for brain control, and n = 23 for hand control. For Monkey J, n = 14 for training, n = 14 for brain control, and n = 14 for hand control. For monkey N, n = 14 for training, n = 12 for brain control, and n = 5 for hand control.

Figure 3 shows the mean circular variance across three different monkeys for different types of experimental blocks. For all monkeys, the training and brain control blocks were significantly lower than the hand control blocks, indicating increased clustering during training and brain control. In addition, the circular variance for training blocks was higher than that of brain control blocks, but not at a significant level.

Rao's spacing test revealed consistent results with circular variance (Figure 4). For all monkeys, the percentage of sessions with p-values <= 0.01 were lowest for hand control, indicating increased likelihood of a uniform PD spread. In contrast, training and brain control percentages were similarly high.

In addition to testing the uniformity of the distributions, we wanted to see whether there were significant differences between distributions under the different block types. Figure 5 compares the population mean PD, or cluster direction, between blocks. Brain control and training blocks had similar cluster directions (Figure 5A), while hand control cluster directions differed (Figure 5B, 5C). The Kuiper two-sample test tests the null hypothesis that two circular distributions are identical. We found that we could not reject the null hypothesis for the majority of brain control and training blocks at the 0.05 significance level (Figure 5D). Comparisons with hand control, however, lead us to reject the null hypothesis, indicating hand control distributions differ significantly from training and brain control distributions. Monkey N was excluded from this analysis because hand control blocks did not occur in the same session as brain control and training blocks.

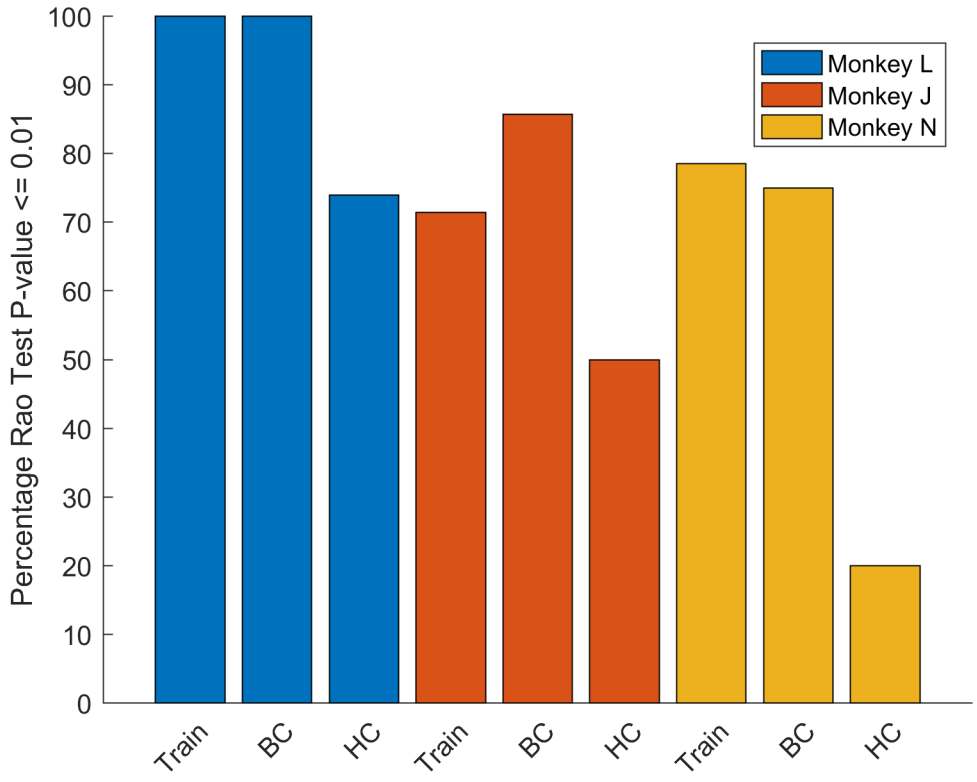

Figure 4. Rao's spacing test suggests hand control blocks have more uniform PD distributions than training or brain control blocks. The p-value threshold of 0.01 was chosen to reveal larger differences between blocks. "Train", "BC", and "HC" are the same as in Figure 3. Hand control has lower percentages than training and brain control, indicating that one cannot reject the null hypothesis of uniform distributions for a larger proportion of sessions. Sample sizes are the same as for Figure 3.

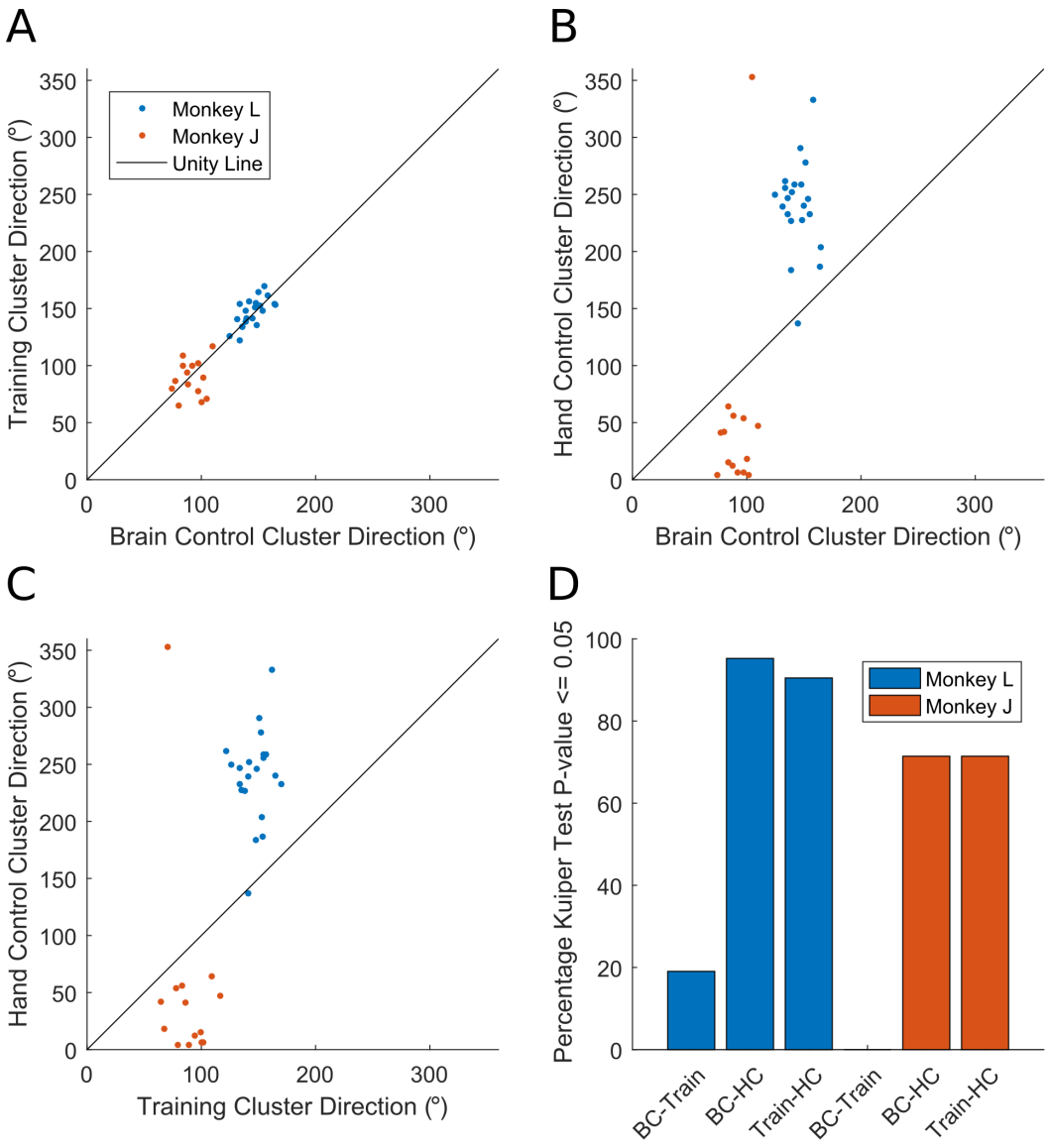

Figure 5. Cluster direction and PD distributions are nearly identical between brain control and training, but different for hand control. A) The cluster direction, or population mean PD, is similar for brain control and training, for both monkeys. Most of the points lie near the unity line. **B)** Brain control and hand control have different cluster directions. Most of the points lie away from the unity line. **C)** Training and hand control have different cluster directions. **D)** "Train", "BC", and "HC" are the same as in Figure 3. Only units with a von Mises fit R^2 >= 0.1 were included in the analysis. Hand control has lower percentages than training and brain control, indicating that one cannot reject the null hypothesis of uniform distributions for a larger proportion of sessions. Sample sizes are the same as for Figure 3.

3.2 MODULATION DEPTHS INCREASE FROM HAND TO BRAIN CONTROL

Given that PDs are clustered, this raises the important question of how the animal can still control the cursor. Something else must compensate. One candidate would be modulation depths: by activating neurons with very different PDs more than those near the center of the cluster, the animal might be able to preserve control despite the clustering. Visual inspection of Figure 2 suggests this is occurring; here we examine the effect more closely. See section 3.7 and Figure 16 for more details about the theory behind modulation depth compensation. Prior studies have looked at modulation depth changes before, and our results are consistent with their findings (Ganguly and Carmena, 2009; Zacksenhouse et al., 2007) in that modulation depths are increased during brain control compared to during hand control (Figure 6C). Between training and brain control, modulation depth shows a slight trend towards increasing (Figure 6A). Like brain control, modulation depths increase during training compared to hand control (Figure 6D). Only units with a von Mises fit R^2 value greater than 0.1 were plotted included in the analysis.

To examine whether neurons far from the cluster center have higher modulation depths, we plotted modulation depth by distance from the cluster direction (Figure 7). We found that modulation depths near the center of the cluster tended to be reduced compared to modulation depths of units approximately 45° away from the cluster direction. The peak in the -45° direction is broader and weaker, however. In hand control, modulation depths tended to be fairly uniform about the cluster direction.

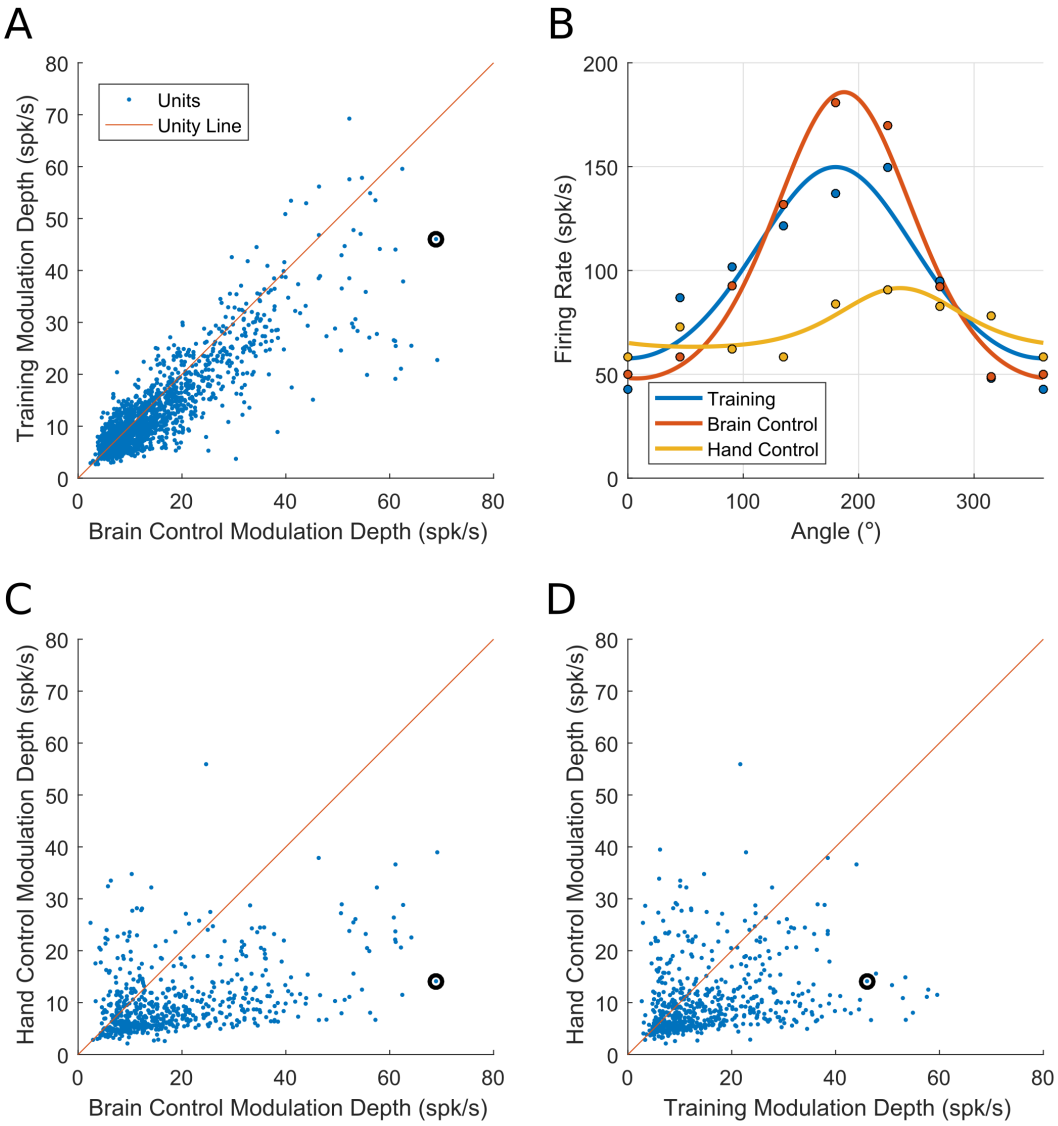

Figure 6. Modulation depths increase during brain control compared to hand control in monkey L. A) Each point represents a unit. Von Mises fits were used to determine modulation depths. Only units with R^2 values greater than 0.1 were plotted (n = 1669). 64.5% of points lie below the unity line, indicating a slight trend towards increasing modulation depths for brain control. **B)** The tuning curves of the unit enclosed by a circle in A, C, and D are shown, comparing tuning during training, brain control and hand control. Points indicate mean firing rates for the target angle. **C)** The majority (80.9%) of points (n = 592) lie below the unity line, indicating a trend towards increased modulation depths during brain control. **D)** The majority (73.8%) of points (n = 579) lie below the unity line, indicating a trend towards increased modulation depths during training.

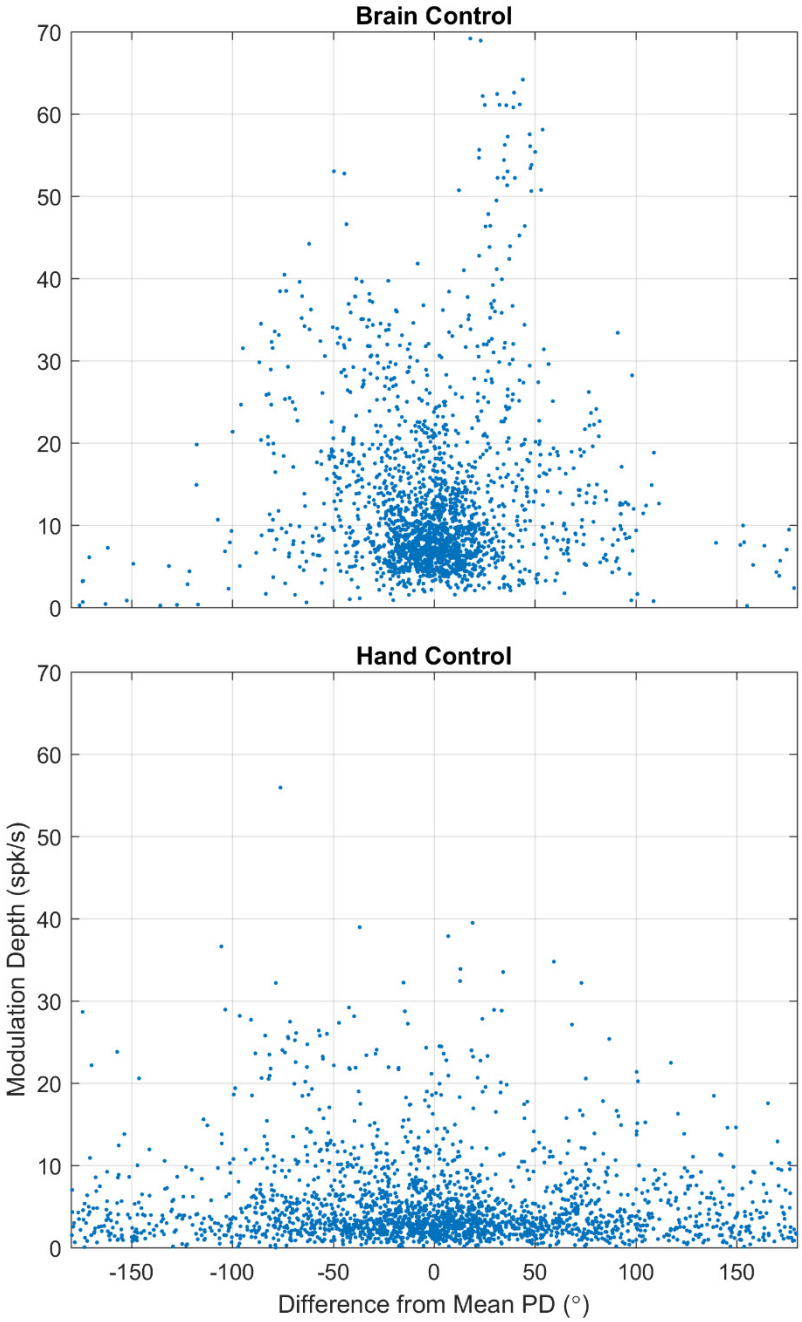

Figure 7. Modulation depth as a function of difference from population mean PD (cluster direction) in monkey L. For each session, each unit's modulation depth was acquired and the PD distance from the cluster direction was calculated. Note the "peaks" in brain control around -45° and 45°. In contrast, units with PDs close to the cluster direction have lower modulation depths. In hand control, most modulation depths are low and the points near the cluster direction are more spread out than in brain control.

We also wanted to investigate whether sessions with more clustered preferred directions had greater increases in modulation depths. To find out, we plotted the average proportional increase in modulation depth from hand control to brain control as a function of circular variance (Figure 8). Indeed, modulation depth shows greater increases as circular variance decreases.

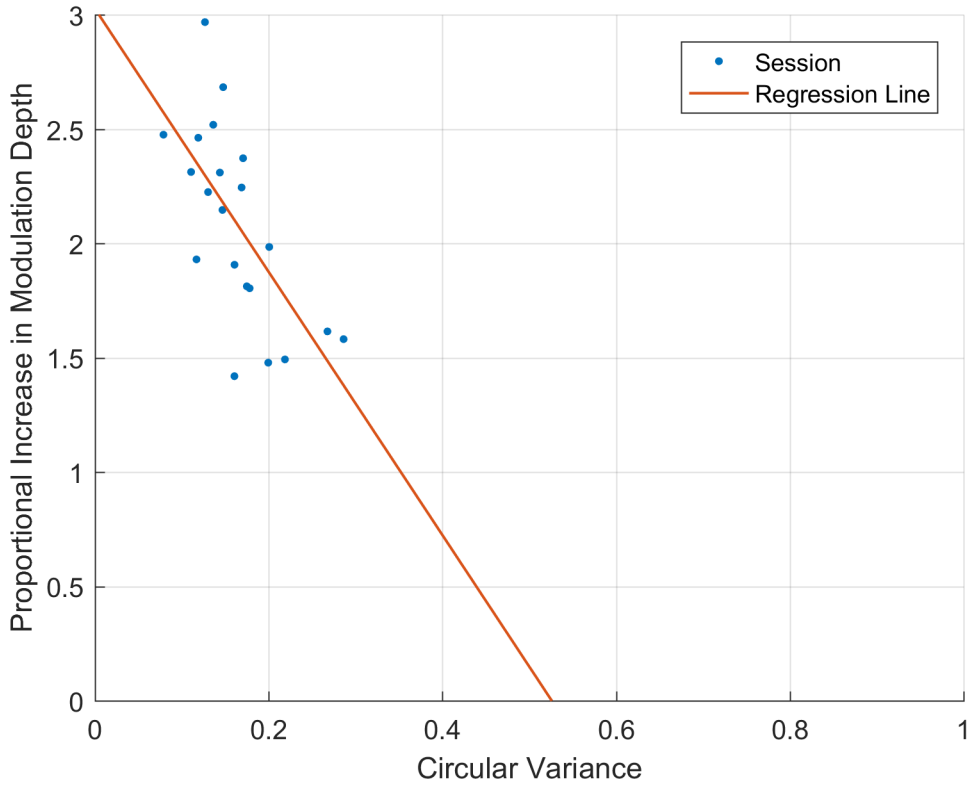

Figure 8. Modulation depths increase as preferred directions become more clustered in monkey L. For each session, each unit's modulation depth during brain control and hand control was acquired. Brain control modulation depths were divided by hand control modulation depths for each unit. The mean proportion was calculated and plotted for each session. The plot shows a trend of increasing modulation depths as circular variance decreases. The regression line has $R^2 = 0.449$ and $p<0.001$, and is defined by $\boldsymbol{y = -5.75x + 3.03}$.

3.3 NARROWNESS OF TUNING INCREASES OR DECREASES BETWEEN BRAIN AND HAND CONTROL

Narrowness of tuning may also influence controllability of the decoder. A narrowly tuned neuron will be more of an all-or-none responder, firing at baseline most of the time until a movement towards the preferred direction is performed. However, most neurons in motor cortex have relatively broad tuning curves (Taubman et al., 2013). Broadly tuned neurons contribute to a wider range of movements than narrowly tuned neurons. Whether the narrower or broader tuning curves contribute more to decoding is difficult to tell (Pouget et al., 1999), but more broadly tuned neurons may be better for higher-dimensional tasks (Zhang and Sejnowski, 1999). Hand control is one such high-dimensional task, so we expected to see broadening of tuning curves from brain control to hand control.

We calculated the full width half max for each unit and plotted the difference between blocks in Figure 9. There appears to be a difference between the comparison of observation training and brain control (Figure 9A) and the comparison of brain control and hand control (Figure 9C). From brain control to hand control, the narrowness of tuning appears to modulate to a greater degree than from brain control to observation training. Again, only units with a von Mises tuning curve fit R^2 value greater than 0.1 were plotted. The cluster of points slightly above the unity line in the top right of Figure 9C suggest a subset of units become more broadly tuned when switching from brain to hand control. The same cluster exists in Figure 9D. However, the number of points above the unity line is approximately equal to the number of points below the unity line, indicating that several units also sharpened their tuning curves.

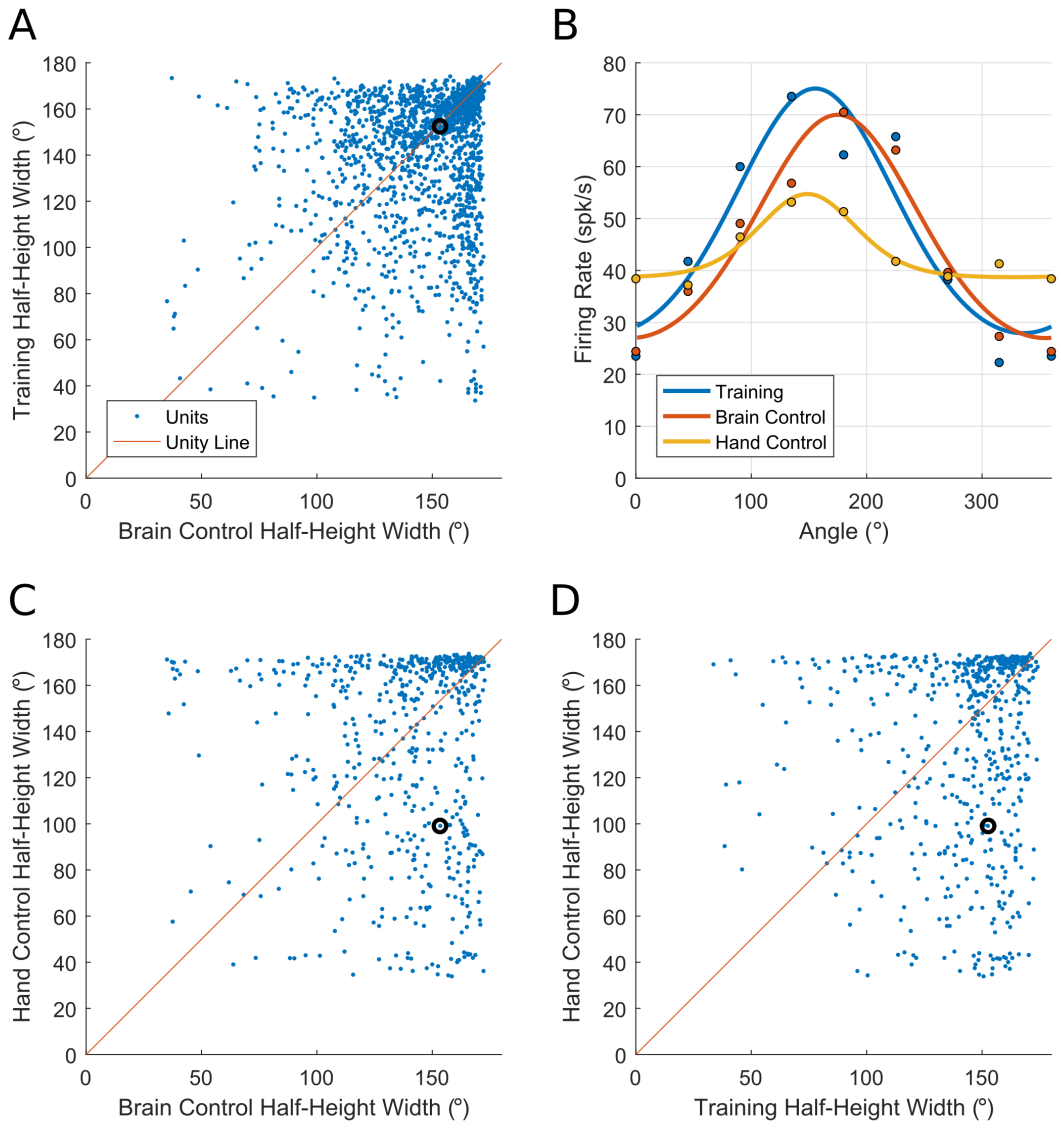

Figure 9. Narrowness of tuning increases or decreases between brain and hand control. Narrowness of tuning is defined as the width of the tuning curve at half height, or the half-height width, as determined by von Mises fits. Units far from the unity line have greater shifts in their half-height widths. **A)** 52.9% of 1669 points lie below the unity line. Note the cluster of points along the unity line in the top right corner, indicating several units did not change their tuning. **B)** The tuning curves of the unit enclosed by a circle in A, C, and D are shown, comparing tuning during training, brain control, and hand control. Points indicate mean firing rates for the target angle. **C)** 45.6% of 592 points lie below the unity line. **D)** 49.2% of 579 points lie below the unity line.

3.4 PREFERRED DIRECTION SHIFT

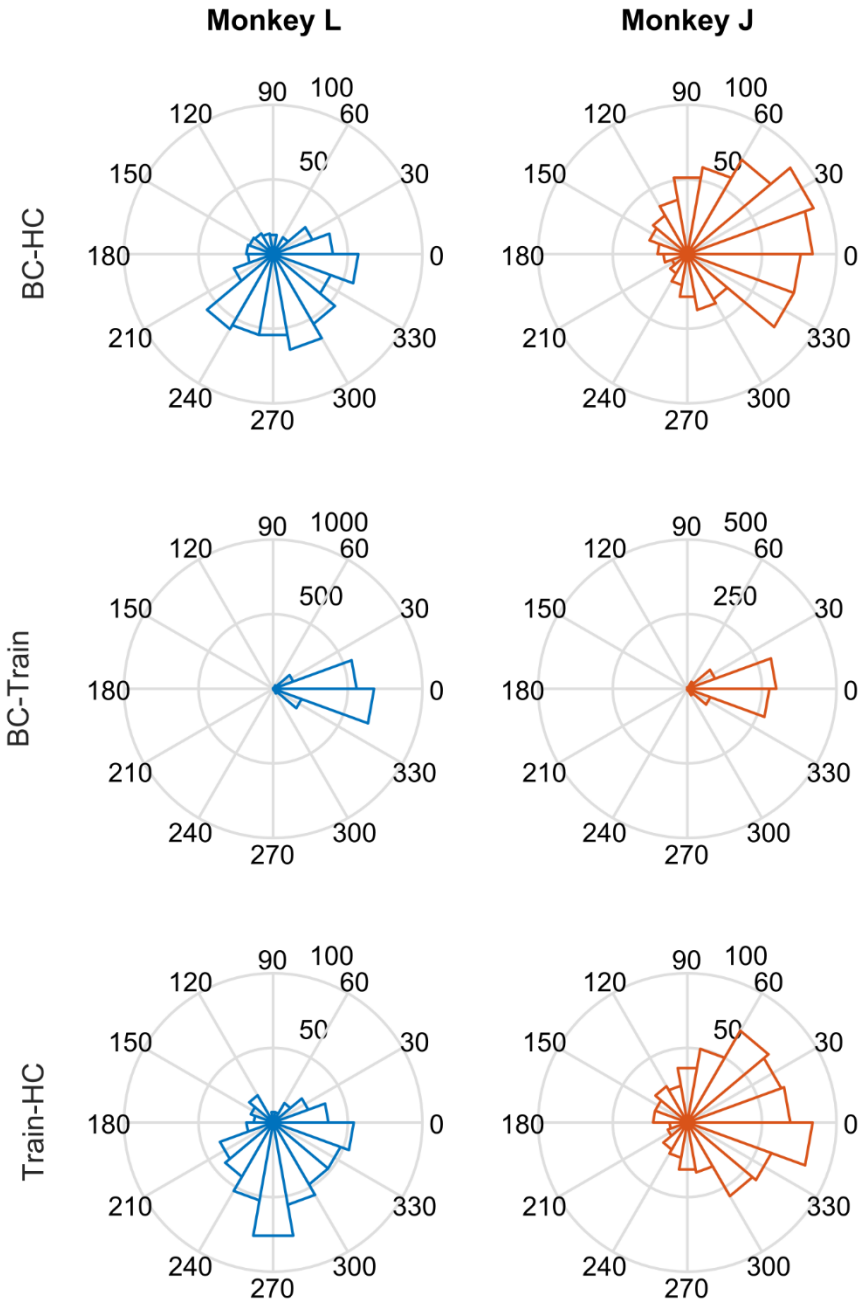

Figure 10. Angle histograms showing the shift in preferred direction for monkeys L and J. "Train", "BC", and "HC" are the same as in Figure 3. Preferred directions for each unit during a block type were subtracted from preferred directions of the same units during another block type. Only units with a von Mises fit $R^2 > 0.1$ were used.

Preferred directions shift little between brain control and training. Note the similarity between BC-HC and Train-HC histograms.

We found that PDs change dramatically between brain control and hand control (Figure 1). To examine whether there was any consistency to that change, we looked at PD shifts. PD shifts were calculated by subtracting the PD of a unit during one block type from the PD of the same unit in a different block type (Figure 10). Only sessions where training, brain control, and hand control were conducted on the same day were included. Monkey N did not have results because hand control and brain control were conducted in separate experimental sessions. The PD shift between brain control and hand control was similar to the PD shift between training and hand control. Meanwhile, the distribution of PD shifts between brain control and training lay tightly centered around 0, suggesting minimal changes in PD.

3.5 TIMESCALE OF CHANGES

Figure 11 shows the timeline of experimental sessions and the overall mean PD, or the cluster direction, for each day and block type. For monkeys L and J, the cluster direction remains near the same value over time, suggesting a stable cortical representation (Ganguly and Carmena, 2009). On the other hand, hand control cluster direction appears to vary more greatly. For monkey N, who was learning how to use a BCI, training, brain control, and hand control all varied to a larger extent than in the other monkeys. It is worth noting that for hand control, hundreds of thousands of neurons contribute to behavior, so the neural code for hand movements might be relatively insensitive to the stability of tuning of individual neurons. For BCI control, only the 96 neural

channels being recorded control behavior, so there is less robustness inherent in the system. This might help explain why tuning appears more stable during BCI control than during hand control.

Figure 12 shows various measures regarding the tuning of individual units over the course of a single block of trials. The preferred direction of a unit remains nearly constant. Modulation depth, narrowness of tuning, and baseline firing rate change minimally over time. The units chosen had the largest von Mises fit R^2 values in the population. The results from Figure 12 suggest that the shift in neural tuning measures occurs abruptly from one block type to the next. This matches results from the literature (Zacksenhouse et al., 2007). The high baseline firing rate of unit 1 in Figure 12 suggests it may be a multi-unit. Even so, its tuning properties remain relatively stable.

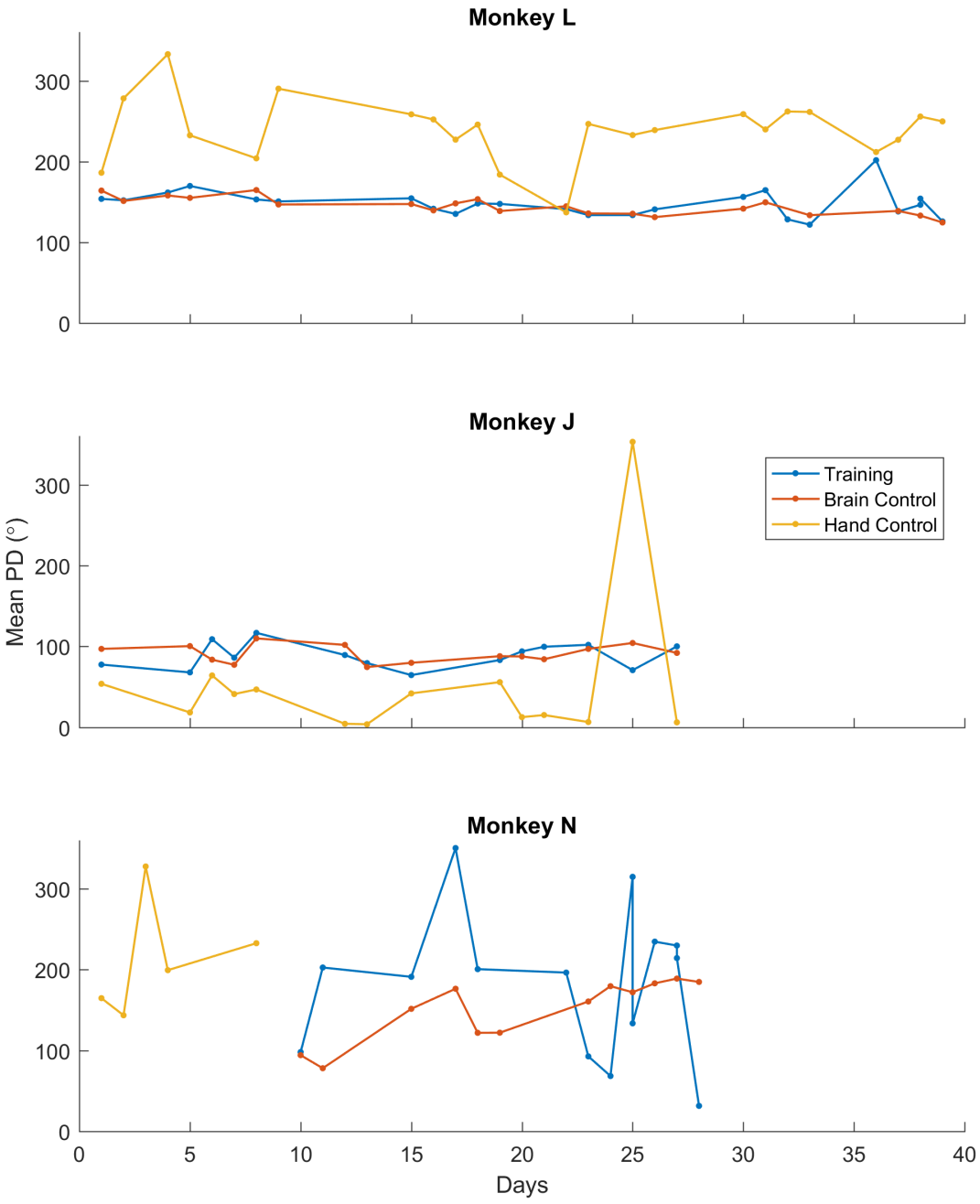

Figure 11. Timelines of experimental sessions and the mean preferred direction for each block type. For each day and block type, the mean PD, or cluster direction, was calculated. For monkey J, the large peak in hand control on day 25 is actually a small difference due to the representation of angles chosen.

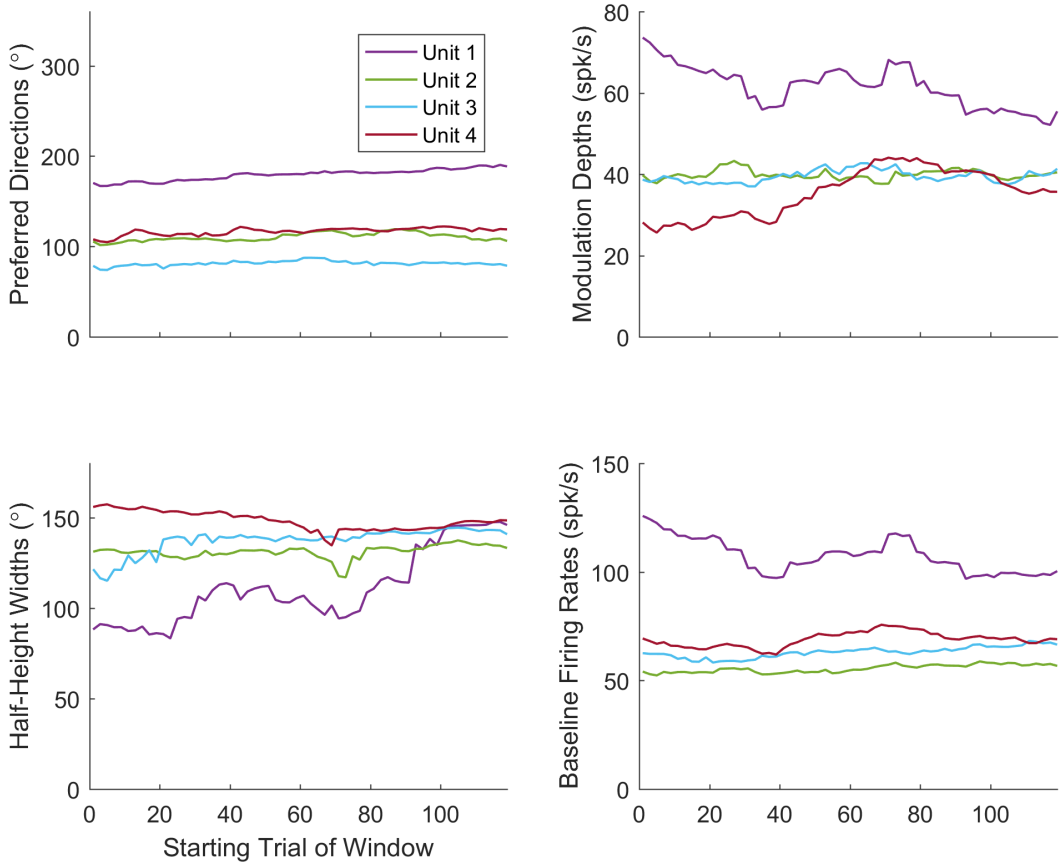

Figure 12. Tuning properties over the course of a brain control block are relatively stable. The preferred direction, modulation depth, half-height width, and baseline firing rate for 4 units are shown over time. A moving window of 40 trials was used. Von Mises fits and tuning properties were calculated for each window location. Tuning properties are relatively stable over a single block of trials.

3.6 PERFORMANCE AND CLUSTERING

To quantify performance, we used the measures of reach duration, angular dispersion, and straightness index. For angular dispersion and straightness index, higher values indicate straighter trajectories, with 1 being a perfectly straight trajectory. Figure 13 shows two example trajectories.

The gray trajectory has a higher straightness index and angular dispersion, indicating a straighter trajectory overall.

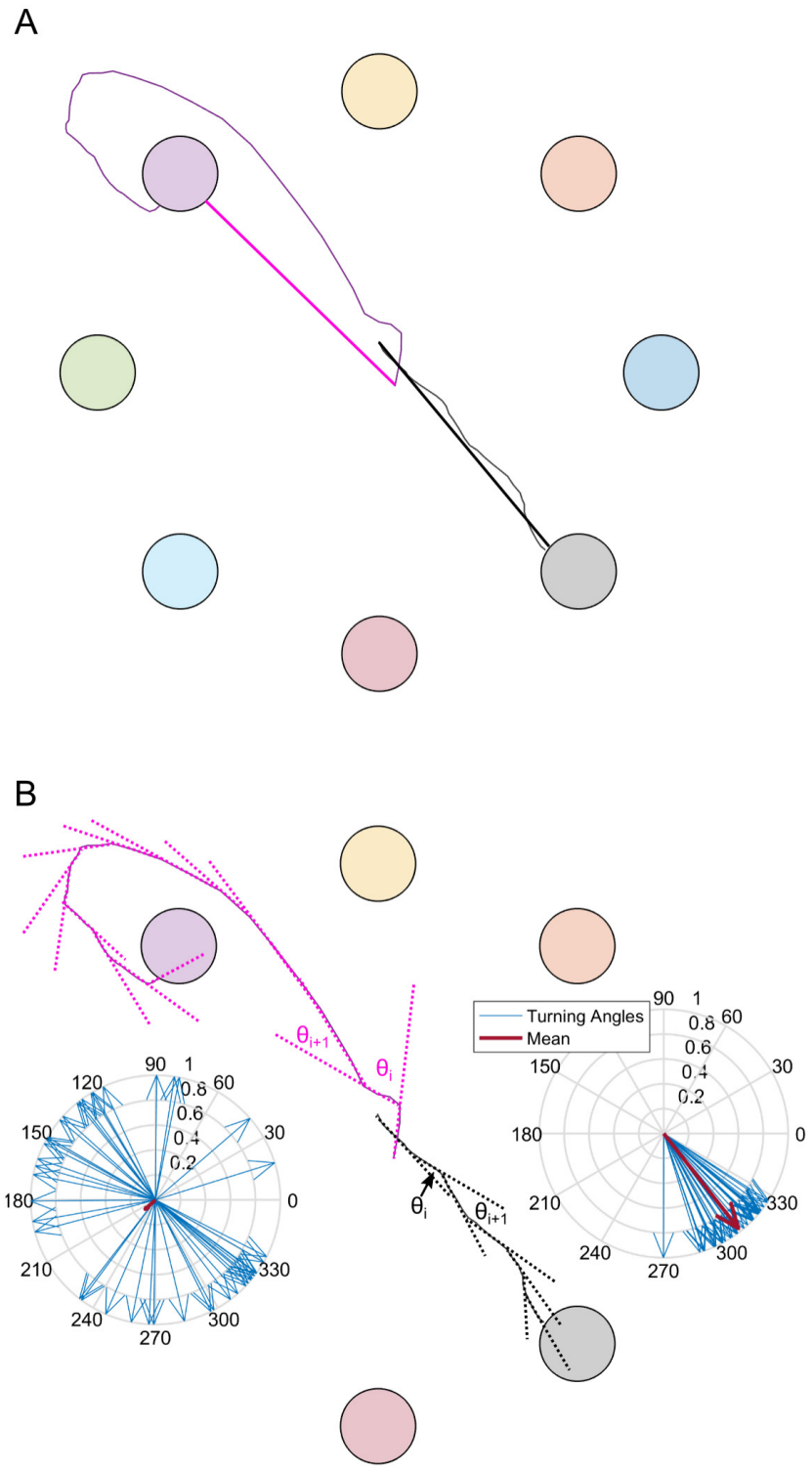

Figure 13. Calculation of straightness index and angular dispersion for two example trajectories from a brain control session in monkey L. Colored circles represent targets in the workspace. The gray trajectory took 1.440 s to complete, while the purple trajectory took 1.888 s to complete. The gray trajectory had a straightness index of 0.922 and an angular dispersion of 0.972. The purple trajectory had a straightness index of 0.372 and an angular dispersion of 0.111. **A)** Straightness index is the ideal length (magenta and black) divided by the actual length (purple and gray). **B)** Angular dispersion is the length of the mean resultant vector (red, in compass plots) of the turning angles ($\boldsymbol{\theta_i, \theta_{i+1} \ldots \theta_n}$). Illustrative turning angles in magenta and black are not actual turning angles. Real turning angles are shown as blue arrows in the compass plot.

Figure 14 displays the results for a single brain control session in monkey L. Monkey L performs well at targets around 90°. However, across sessions, the relationship between performance and cluster direction is not readily apparent. To find if there was a relationship, we performed circular-linear regression. A few clues suggest a relationship exists. From observation during experiments, we know that some targets are more difficult to reach than others. In addition, when the monkey gives up, the cursor drifts off the screen. This drift direction is typically opposite of the cluster direction.

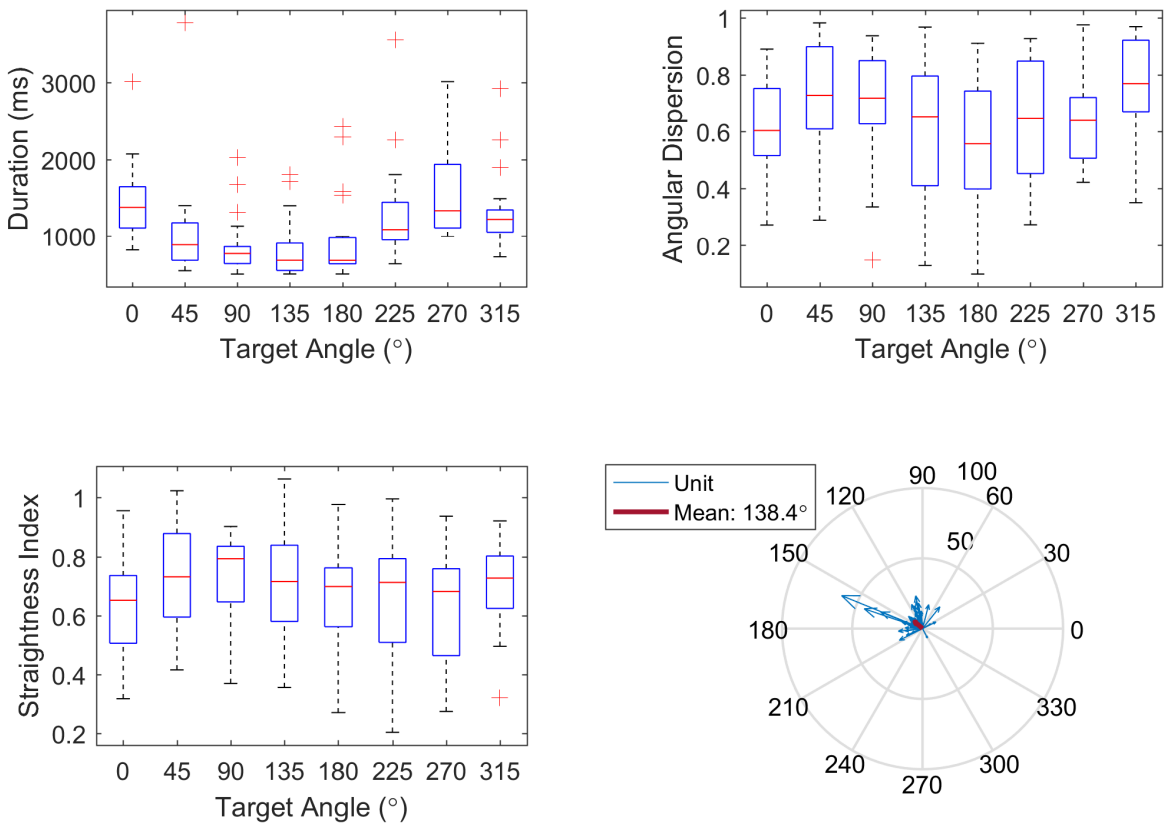

Figure 14. Performance relative to cluster direction for a monkey L brain control block. Performance measures are plotted for each of 8 targets. Boxplots show the median, interquartile range (IQR), and outliers. Whisker lengths show the lowest and highest data within 1.5 times the IQR of the lower and upper quartile, respectively. Angle distribution shows unit PDs as arrows scaled by modulation depth.

To measure the relationship between performance and mean cluster direction, we performed circular linear regression using mean performance per target as the predictors and mean cluster direction as the response variable. In other words, we asked whether the mean cluster direction was predictable from the performance measures. Significant relationships were discovered in monkey L (Figure 15), suggesting that clustering has some influence on performance.

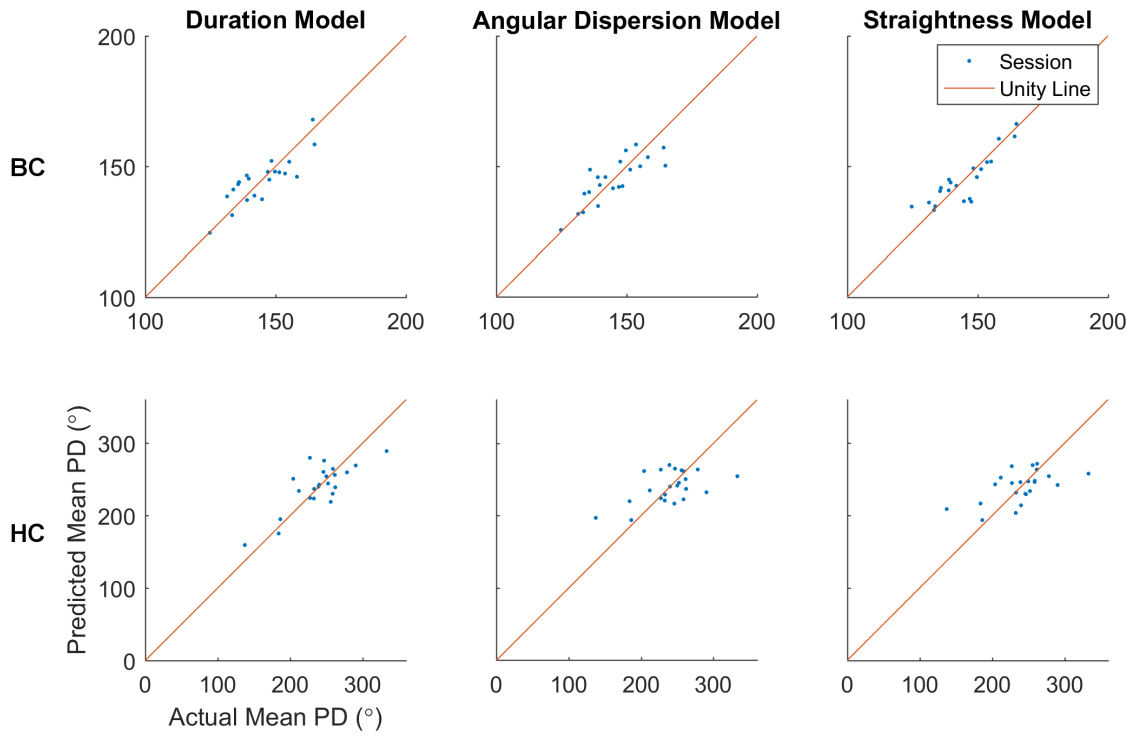

Figure 15. Actual vs. predicted mean PDs from circular-linear regression in monkey L. Top row is brain control and bottom row is hand control. From top-to-bottom, left-to-right, the p-values of the regressions are 0.022, 0.043, 0.008, 0.040, 0.695, and 0.584. All brain control regressions are significant to the 0.05 level, while only the duration model is significant for hand control. The x- and y-axes for brain control are reduced to show the relationship more clearly.

3.7 MODULATION DEPTH INCREASE REQUIRED

Neurons further away from the cluster direction typically have greater modulation depth. How might this be related to controllability? To examine this, we looked at the generative model of neuron's spike counts:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{q}_k$$

Ignoring noise, the equation can become:

$$\mathbf{H}^{+}\mathbf{z} = \mathbf{x}$$

Each neuron's spike count can be multiplied by numbers in the pseudoinverted $\mathbf{H}^{+}$ matrix to obtain its contribution to velocity. The sum of the contributions of all neurons produces a final velocity. Making a few assumptions, one can find the firing rates necessary for two neurons with different PDs to control a BCI in a center-out task. First, assume the neural contributions as represented by $\mathbf{H}^{+}$ are in the same direction as the neuron's PD. Also assume all contributions have a vector length of 1. For example, a neuron with a 0° PD contributes 1 to velocity in the x direction and 0 to velocity in the y direction. Next, assume eight target velocities lie on the unit circle, i.e. the $\mathbf{x}$ vector may become $[1,0]^T$, $\left[\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}\right]^T$, $[0,1]^T$…etc. Finally, assume the baseline (most controllable) neural PDs are at 0° and 90° (found by testing).

With these assumptions in place, we can manipulate $\mathbf{H}^{+}$and $\mathbf{x}$ to find the maximum modulation depths required as one neuron changes its preferred direction to cluster more closely with the other neuron. The results are shown in Figure 16.

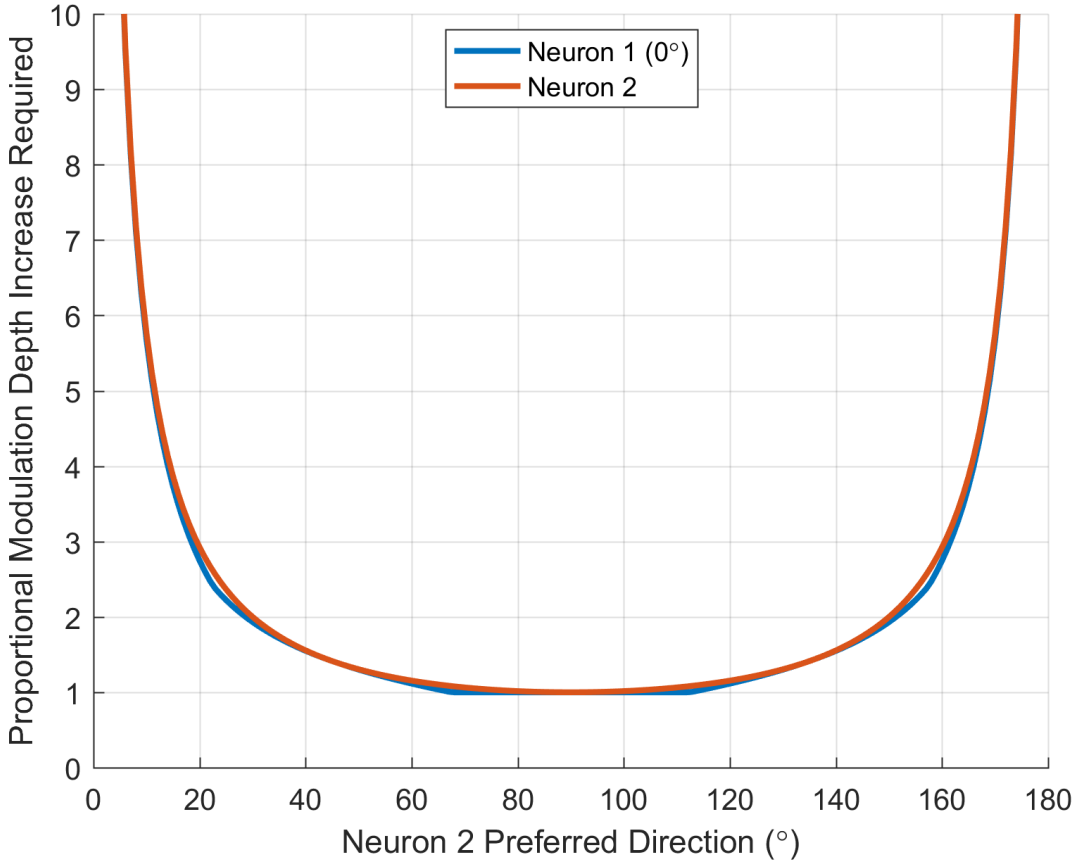

Figure 16. As 2 neurons' preferred directions approach one another, the minimum firing rate required to reach all 8 targets increases. Neuron 1's preferred direction is fixed at 0°, while neuron 2's preferred direction is swept from 0° to 180°. Proportional modulation depth increase required is calculated using the modulation depth required at 90° as a baseline.

There are at least two strategies for good control when you have clustering. One is to use the neurons near the cluster direction. Those would need to have very large modulation depths (Figure 16). The other is to use neurons whose PDs are far from the cluster direction. The proportional firing rate increase required is lowest when neuron 2 is at 90°. In Figure 7, the modulation depths around -45° and 45°of the cluster direction show possible peaks. Most of the units with PDs close to the cluster direction have lower modulation depths. This seems to suggest

usage of the second strategy. The overall clustering effect is visible in Figure 7 when comparing brain control and hand control.

4.0 DISCUSSION

Preferred directions cluster during BCI usage. Intuitively, one would expect that less clustering results in better control. In a BCI, if recordings were only available from one neuron, the user would be able to move along that neuron's tuning axis. Having another neuron tuned to an orthogonal direction would enable movement in a 2-D plane. For 2-D velocity decoding, additional neurons may provide extra signal for the velocity to be decoded more accurately. If the neural PDs were unclustered, the brain would only have to modulate a few neurons to move the cursor in any direction. Instead, the neural PDs do cluster during BCI control.

Conflicting results regarding clustering have been found in the literature. Inferring from the figures, one human study did not show PD clustering (Truccolo et al., 2008). Kim et al. found that for one human subject, most of the PDs remained the same between training and brain control while for another, most of the PDs changed (Kim et al., 2008). The modulation depths of the subject whose PDs changed were lower, however. Overall, the modulation depths of the human subjects increased between training and brain control. In posterior parietal cortex, PDs became less clustered and modulation depth increased as the animal learned across sessions (Mulliken et al., 2008). They used a ridge regression decoding algorithm.

On the other hand, figures from other studies are consistent with our results. Several show PD clustering during brain control (Carmena et al., 2003; Lebedev et al., 2005; O'Doherty et al., 2011). Others found that mean preferred direction differences between brain control and hand

control increased across days (Taylor et al., 2002). Going from brain control with hand movements to brain control without hand movements, one study found that modulation increased abruptly (Zacksenhouse et al., 2007). Finally, others found that modulation depths either increased or remained the same, but did not decrease (Ganguly and Carmena, 2009).

Methods to calibrate a BCI decoder differ between labs. A decoder can be trained on the neural activity that occurs as a monkey actively reaches to the target. Alternatively, the monkey may simply observe the cursor moving to a target. Clustering of preferred directions may occur due to decoder training and monkey behavior idiosyncrasies. Jarosiewicz et al. showed that when a subset of neurons has their preferred directions perturbed, the entire population compensates, with the subset changing more than the non-perturbed neurons (Jarosiewicz et al., 2008). It is conceivable that some neurons will be misinterpreted during training of a decoder, and the resulting compensation may result in clustering. Our results show clustering occurring during, but this may be due to the monkey having already learned how to use a BCI.

One can propose hypotheses about PD clustering from a couple categories: why the clustering can occur (proximate) and why the clustering does occur (ultimate). From the proximate category, one hypothesis states that only a few neurons without clustered PDs are important for the decoding, and the tuning of the remaining neurons matters little. In this study, we found that many units increased their modulation depths abruptly going from hand control to brain control. One could test if the units with the greatest modulation depth changes are weighted highly in the decoding algorithm. These units tended to be further away from the cluster direction. The contribution of each unit to the decoder is in the rows of the $\mathbf{H}$ matrix. To test the hypothesis experimentally, one could remove these units' contribution to decoding, simply by setting the decode weight to 0 for that unit. If the hypothesis is correct, either performance will decrease or

another subset of units will shift their tuning curves to an unclustered state. Alternatively, if the hypothesis is incorrect, the performance will stay the same and/or any tuning curve shifts do not show a clear trend towards an unclustered state. In addition, to test the hypothesis post-hoc without performing additional experiments, one could remove a sample of neurons with no clear cluster direction and observe the change in offline-decoded trajectories.

An additional proximate hypothesis about why clustering can occur is that it depends on our choice of BCI decoding algorithm. It might be that the PVA specifically prevents clustering, but that other algorithms, such as the velocity Kalman filter we used, permit it. Other algorithms that might permit clustering include the Weiner filter (Carmena et al., 2003) and the unscented Kalman filter algorithm (O'Doherty et al., 2011). This can be tested by switching between several different decoding algorithms with the same animals, and observing whether clustering changes. If clustering does depend on the algorithm choice, the question is still open whether clustering is something the brain wants to do but is prevented from doing by some algorithms, or whether clustering somehow actually improves BCI performance under some decoding algorithms. As seen in Figure 16, it is hard to see how this could be true, but there could be some aspect of the decoding algorithm's function we have not thought of yet. If the brain's natural state is to cluster PDs during BCI control, then that interesting phenomenon will warrant explanation. That leads to our hypotheses about why clustering ultimately occurs.

We propose three hypotheses about why clustering ultimately occurs. Hypothesis one states that the neural PDs cluster due to a tendency of the recorded population to modulate together. If this is true, the clustering represents a balance between controllability and ease of modulation. To test the hypothesis, consider hand control. Our results show that neural PDs during hand control are relatively unclustered. In hand control, there are many more intermediary dimensions to control

than two, which may include limb degrees of freedom or even muscles themselves (Kakei et al., 1999). The additional dimensions may lead to a more uniform distribution of preferred directions. One could restrain the arm such that fewer muscles and/or degrees of freedom are required to perform the same task. For example, wrist movements could be used to control a cursor in a 2-D center-out task. This could be compared with using the whole arm to perform the same task. If the hypothesis is true, one would expect more clustering in the wrist control regime, due to having less dimensions to control.

Hypothesis two states that clustering reflects a strategy the animal learned early. Monkeys take time learning to control a BCI, and may develop idiosyncrasies. During hand control, our setup is such that the monkey has to lift his arm upwards from the chair's armrest to the central position. During brain control, the arm is restrained to the chair's armrest. Perhaps our monkeys are straining against their arm restraints during brain control, attempting to reach upwards. This could result in clustered preferred directions, as each reach would be influenced by the upwards strain. One way to test this hypothesis is to perform electromyography of the monkey's arm muscles, especially early on. In this new experiment, successful trials require minimal activation of the muscles. The monkey would be trained not to strain, and if the hypothesis is correct, the clustering seen during brain control would disappear. Another test of this hypothesis would be to specifically require the animal to touch a different onscreen target during BCI control. Touch targets could be up, left, and right of the BCI stimulus array. This might cause the PDs of the neurons to change under BCI control.

The third ultimate hypothesis states that clustering reflects the native tuning of motor cortex. Perhaps upstream areas such as dorsal premotor cortex drive motor cortex in a simple manner, and the clustering we see comes from the lack of proprioceptive and somatosensory

feedback. One way to test this hypothesis is to train the monkey to perform hand control while a nerve block stops proprioceptive and somatosensory feedback. If the hypothesis is true, increased clustering should be seen. Studies have found that by reducing somatosensory feedback, motor cortex becomes disinhibited (Brasil-Neto et al., 1993; Kaelin-Lang et al., 2002). This may be related to the modulation depth increases we see during brain control. A second test of this hypothesis would be to record in PMd during BCI control from M1. Similarly clustered PDs in PMd during brain control and hand control would support the hypothesis.

Our study confirms the appearance of PD clustering during brain control and training with a Kalman filter decoding algorithm. Although this effect is present in the publications of others, to our knowledge, it has rarely been noted (Green and Kalaska, 2011; Tehovnik et al., 2013), and never explained in the literature. We found that tuning changes occur abruptly and modulation depths increase from hand control to brain control. Narrowness of tuning can either increase or decrease, with a cluster of units becoming broader from brain control to hand control. Preferred directions shift little between training and brain control, but similar shifts occur when performing hand control. Overall, the characteristics of training and brain control are similar. This supports the view that the brain is in a comparable functional state during BCI calibration and use, and that those two states are very different from how the brain responds during arm movement control. This flexibility of M1 function is an important motif that is emerging from BCI studies. Next, we found that the mean cluster direction can be predicted from trajectory statistics, indicating some influence of clustering on performance. Finally, we found that the units with the highest modulation depths lie approximately 90° apart, which also happens to be the separation with the minimum modulation depth increase required for controllability.

APPENDIX A

DERIVATION OF HALF-HEIGHT WIDTH

$x_{\mathrm{h}alfwidth\mathrm{h}} = x_{\mathrm{h}alf} - x_{max}$

$\mathrm{f(x)} = \mathrm{y_{half}}$

$$\mathrm{b + me^{\kappa\cos(x-\mu)} = b + m\cosh(\kappa)}$$

$$\mathrm{e^{\kappa\cos(x-\mu)} = \cosh(\kappa)}$$

$$\mathrm{\kappa\cos(x-\mu) = \ln(\cosh(\kappa))}$$

$$\mathrm{\cos(x-\mu) = \frac{\ln(\cosh(\kappa))}{k}}$$

$$\mathrm{x-\mu = acos\left(\frac{\ln(\cosh(\kappa))}{k}\right)}$$

$$\mathrm{x_{half} = acos\left(\frac{\ln(\cosh(\kappa))}{k}\right) + \mu}$$

$\mathrm{f(x)} = \mathrm{y_{max}}$

$$\mathrm{b + me^{\kappa\cos(x-\mu)} = b + me^{\kappa}}$$

$$\mathrm{e^{\kappa\cos(x-\mu)} = e^{\kappa}}$$

$$\mathrm{\kappa\cos(x-\mu) = \kappa}$$

$$\mathrm{\cos(x-\mu) = 1}$$

$$x - \mu = \operatorname{acos}(1) = 0$$

$$x_{max} = \mu$$

$$x_{halfwidth} = x_{half} - x_{max} = \operatorname{acos}\left(\frac{\ln(\cosh(\kappa))}{k}\right) + \mu - \mu$$

$$x_{halfwidth} = \operatorname{acos}\left(\frac{\ln(\cosh(\kappa))}{k}\right)$$

BIBLIOGRAPHY

Amirikian, B., and Georgopulos, A.P. (2000). Directional tuning profiles of motor cortical cells. Neurosci. Res. *36*, 73–79.

Batschelet, E. (1981). Circular statistics in biology. Acad. PRESS 111 FIFTH AVE N. Y. NY 10003 1981 388.

Berens, P., and others (2009). CircStat: a MATLAB toolbox for circular statistics. J Stat Softw *31*, 1–21.

Bialek, W., Rieke, F., Steveninck, R. de R. van, and Warland, D. (1991). Reading a neural code. Science *252*, 1854–1857.

Brasil-Neto, J.P., Valls-Solè, J., Pascual-Leone, A., Cammarota, A., Amassian, V.E., Cracco, R., Maccabee, P., Cracco, J., Hallett, M., and Cohen, L.G. (1993). Rapid modulation of human cortical motor outputs following ischaemic nerve block. Brain *116*, 511–525.

Carmena, J.M., Lebedev, M.A., Crist, R.E., O'Doherty, J.E., Santucci, D.M., Dimitrov, D.F., Patil, P.G., Henriquez, C.S., and Nicolelis, M.A.L. (2003). Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates. PLOS Biol *1*, e42.

Estevez, I., and Christman, M.C. (2006). Analysis of the movement and use of space of animals in confinement: the effect of sampling effort. Appl. Anim. Behav. Sci. *97*, 221–240.

Fetz, E.E. (1969). Operant Conditioning of Cortical Unit Activity. Science *163*, 955–958.

Fetz, E.E., and Finocchio, D.V. (1971). Operant conditioning of specific patterns of neural and muscular activity. Science *174*, 431–435.

Fisher, N.I. (1995). Statistical analysis of circular data (Cambridge University Press).

Fisher, N.I., and Lee, A.J. (1992). Regression models for an angular response. Biometrics 665–677.

Ganguly, K., and Carmena, J.M. (2009). Emergence of a Stable Cortical Map for Neuroprosthetic Control. PLOS Biol *7*, e1000153.

Georgopoulos, A.P., Schwartz, A.B., and Kettner, R.E. (1986). Neuronal population coding of movement direction. Science *233*, 1416–1419.

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*, 2928–2937.

Golub, M., Chase, S., and Yu, B. (2013). Learning an Internal Dynamics Model from Control Demonstration. In ICML (1), pp. 606–614.

Green, A.M., and Kalaska, J.F. (2011). Learning to move machines with the mind. Trends Neurosci. *34*, 61–75.

Humphrey, D.R., Schmidt, E.M., and Thompson, W.D. (1970). Predicting measures of motor performance from multiple cortical spike trains. Science *170*, 758–762.

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*, 19486–19491.

Kaelin-Lang, A., Luft, A.R., Sawaki, L., Burstein, A.H., Sohn, Y.H., and Cohen, L.G. (2002). Modulation of human corticomotor excitability by somatosensory input. J. Physiol. *540*, 623–633.

Kakei, S., Hoffman, D.S., and Strick, P.L. (1999). Muscle and movement representations in the primary motor cortex. Science *285*, 2136–2139.

Kim, S.-P., Simeral, J.D., Hochberg, L.R., Donoghue, J.P., and Black, M.J. (2008). Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. J. Neural Eng. *5*, 455.

Lebedev, M.A., Carmena, J.M., O'Doherty, J.E., Zacksenhouse, M., Henriquez, C.S., Principe, J.C., and Nicolelis, M.A. (2005). Cortical Ensemble Adaptation to Represent Velocity of an Artificial Actuator Controlled by a Brain-Machine Interface. J. Neurosci. *25*, 4681–4693.

Miller, C., Christman, M.C., and Estevez, I. (2011). Movement in a confined space: Estimating path tortuosity. Appl. Anim. Behav. Sci. *135*, 13–23.

Mulliken, G.H., Musallam, S., and Andersen, R.A. (2008). Decoding Trajectories from Posterior Parietal Cortex Ensembles. J. Neurosci. *28*, 12913–12926.

O'Doherty, J.E., Lebedev, M.A., Ifft, P.J., Zhuang, K.Z., Shokur, S., Bleuler, H., and Nicolelis, M.A. (2011). Active tactile exploration using a brain–machine–brain interface.

Pouget, A., Deneve, S., Ducom, J.-C., and Latham, P.E. (1999). Narrow versus wide tuning curves: What's best for a population code? Neural Comput. *11*, 85–90.

Rao, J.S. (1969). Some contributions to the analysis of circular data.

Rao, J.S. (1972). Some variants of chi-square for testing uniformity on the circle. Probab. Theory Relat. Fields *22*, 33–44.

Rao, J.S. (1976). Some tests based on arc-lengths for the circle. Sankhyā Indian J. Stat. Ser. B 329–338.

Schmidt, E.M., Bak, M.J., McIntosh, J.S., and Thomas, J.S. (1977). Operant conditioning of firing patterns in monkey cortical neurons. Exp. Neurol. *54*, 467–477.

Schmidt, E.M., McIntosh, J.S., Durelli, L., and Bak, M.J. (1978). Fine control of operantly conditioned firing patterns of cortical neurons. Exp. Neurol. *61*, 349–369.

Scott, S.H., Gribble, P.L., Graham, K.M., and Cabel, D.W. (2001). Dissociation between hand motion and population vectors from neural activity in motor cortex. Nature *413*, 161–165.

Serruya, M., Shaikhouni, A., and Donoghue, J.P. (2003). Neural decoding of cursor motion using a Kalman filter. In Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference, (MIT Press), p. 133.

Swindale, N.V. (1998). Orientation tuning curves: empirical description and estimation of parameters. Biol. Cybern. *78*, 45–56.

Taubman, H., Vaadia, E., Paz, R., and Chechik, G. (2013). A Bayesian approach for characterizing direction tuning curves in the supplementary motor area of behaving monkeys. J. Neurophysiol. *109*, 2842–2851.

Taylor, D.M., Tillery, S.I.H., and Schwartz, A.B. (2002). Direct Cortical Control of 3D Neuroprosthetic Devices. Science *296*, 1829–1832.

Tehovnik, E.J., Woods, L.C., and Slocum, W.M. (2013). Transfer of information by BMI. Neuroscience *255*, 134–146.

Truccolo, W., Friehs, G.M., Donoghue, J.P., and Hochberg, L.R. (2008). Primary Motor Cortex Tuning to Intended Movement Kinematics in Humans with Tetraplegia. J. Neurosci. *28*, 1163–1178.

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.

Zacksenhouse, M., Lebedev, M.A., Carmena, J.M., O'Doherty, J.E., Henriquez, C., and Nicolelis, M.A.L. (2007). Cortical Modulations Increase in Early Sessions with Brain-Machine Interface. PLOS ONE *2*, e619.

Zhang, K., and Sejnowski, T.J. (1999). Neuronal tuning: To sharpen or broaden? Neural Comput. *11*, 75–84.