

**New Frontiers in Population Recording**

by

George Williams Fraser

BS, Cognitive Science, Carnegie Mellon University, 2005

Submitted to the Graduate Faculty of  
University of Pittsburgh School of Medicine in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH

School of Medicine

This dissertation was presented by George Fraser

It was defended on

April 4, 2011

and approved by

Rob Kass, Adjunct Professor, Neuroscience

and Professor, Statistics, CMU

Tai Sing Lee, Adjunct Associate Professor, Neuroscience

and Associate Professor, Computer Science, CMU

Peter Strick, Professor, Psychiatry and Neurobiology

Doug Weber, Assistant Professor, Physical Medicine and Rehabilitation

Maneesh Sahani, Associate Professor, Theoretical Neuroscience and Machine Learning,

University College London

Dissertation Advisor: Andrew Schwartz, Professor, Neurobiology

Copyright © by George Fraser

2011

## **New Frontiers in Population Recording**

George Fraser

University of Pittsburgh, 2011

The advent of reliable simultaneous recording of the activity of many neurons has enabled the study of interactions between neurons at a large scale: the number of observed pairwise interactions is proportional to the square of the number of recorded neurons. The dominant phenomenon in these pairwise interactions is synchronization, reflecting a system where many observed variables have in common a smaller set of *latent variables*. This permits the possibility that the complex signals observed in the brain might be reducible to a simpler system. We used this insight to design a better signal processing scheme for neuroprosthetics; to identify the same neurons in many recording sessions from their pairwise interactions; to show that the tuning functions of neurons in motor and premotor cortex do not reflect simple coordinate frame models; and to identify error as a dominant signal during continuous movements.

## TABLE OF CONTENTS

<b>1.0</b>	<b>INTRODUCTION .....</b>	<b>13</b>
1.1	THE LATENT VARIABLE CONCEPT .....	13
1.2	NEUROPROSTHETICS AS ESTIMATORS OF LATENT VARIABLES.....	17
1.3	USING THE REDUNDANT CHARACTERISTICS OF FIRING RATES TO TRACK NEURONS CHRONICALLY .....	22
1.4	COMPLEXITY OF TUNING FUNCTIONS IN A RICH MOVEMENT TASK.....	24
1.5	USING LATENT VARIABLES TO CHARACTERIZE BRAIN FUNCTION .....	26
1.6	DEVELOPING A PLATFORM FOR WIDESPREAD SIMULTANEOUS RECORDING.....	29
<b>2.0</b>	<b>CONTROL OF A BRAIN-COMPUTER INTERFACE WITHOUT SPIKE SORTING.....</b>	<b>31</b>
2.1	ABSTRACT .....	31
2.2	INTRODUCTION .....	32
2.3	METHODS.....	33
2.3.1	Chronic microelectrode implant.....	33
2.3.2	Behavioral task.....	34
2.3.3	Signal Processing .....	36
2.3.4	Decoding algorithm .....	38

2.4	<b>RESULTS .....</b>	<b>44</b>
2.5	<b>DISCUSSION.....</b>	<b>52</b>
3.0	<b>RECORDING FROM THE SAME NEURONS CHRONICALLY IN MOTOR CORTEX..</b>	<b>55</b>
3.1	<b>ABSTRACT .....</b>	<b>56</b>
3.2	<b>INTRODUCTION .....</b>	<b>57</b>
3.3	<b>METHODS.....</b>	<b>59</b>
3.3.1	<b>Chronic microelectrode implant.....</b>	<b>59</b>
3.3.2	<b>Behavioral task.....</b>	<b>60</b>
3.3.3	<b>Tracking the same neurons .....</b>	<b>60</b>
3.3.4	<b>Synthetic data.....</b>	<b>67</b>
3.3.5	<b>Long-term accuracy .....</b>	<b>69</b>
3.3.6	<b>Preferred directions .....</b>	<b>70</b>
3.4	<b>RESULTS .....</b>	<b>71</b>
3.4.1	<b>Classification accuracy.....</b>	<b>71</b>
3.4.2	<b>Tuning Parameters.....</b>	<b>78</b>
3.5	<b>DISCUSSION.....</b>	<b>84</b>
3.6	<b>ACKNOWLEDGEMENTS.....</b>	<b>87</b>
4.0	<b>COMPLEXITY OF TUNING FUNCTIONS IN A RICH MOVEMENT TASK.....</b>	<b>88</b>
4.1	<b>INTRODUCTION .....</b>	<b>88</b>
4.2	<b>METHODS.....</b>	<b>89</b>
4.2.1	<b>Behavioral task.....</b>	<b>89</b>
4.2.2	<b>Neural recording .....</b>	<b>91</b>

4.2.3	Sampling.....	92
4.2.4	Tuning parameters.....	92
4.2.5	Latent variable analysis.....	93
4.2.6	Information theoretic analysis.....	94
4.3	RESULTS .....	95
4.4	DISCUSSION.....	103
5.0	LATENT VARIABLES REVEAL AN ERROR SIGNAL IN PREMOTOR AND MOTOR CORTEX	106
5.1	ABSTRACT .....	107
5.2	INTRODUCTION .....	107
5.3	METHODS.....	110
5.3.1	Behavioral task.....	110
5.3.2	Neural recording .....	111
5.3.3	Sampling.....	112
5.3.4	Illusion response scores.....	113
5.3.5	Latent variable analysis.....	114
5.3.6	Computing the kinematic error signal in drawing .....	117
5.4	RESULTS .....	118
5.5	DISCUSSION.....	123
6.0	A PLATFORM FOR WIDESPREAD SIMULTANEOUS RECORDING .....	127
6.1	A LEGGED TITANIUM RECORDING CHAMBER.....	127
6.2	THE 256-ELECTRODE ARRAY .....	133

**BIBLIOGRAPHY ..... 140**

## LIST OF TABLES

Table 1. ....	46
Table 2. ....	75
Table 3. ....	76

## LIST OF FIGURES

Figure 1. ....	14
Figure 2. ....	15
Figure 3. ....	28
Figure 4. ....	40
Figure 5. ....	41
Figure 6. ....	47
Figure 7. ....	49
Figure 8. ....	50
Figure 9. ....	62
Figure 10. ....	65
Figure 11. ....	66
Figure 12. ....	77
Figure 13. ....	78
Figure 14. ....	80
Figure 15. ....	81
Figure 16. ....	83

Figure 17. ....	90
Figure 18. ....	91
Figure 19. ....	96
Figure 20. ....	97
Figure 21. ....	99
Figure 22. ....	101
Figure 23. ....	102
Figure 24. ....	111
Figure 25. ....	112
Figure 26. ....	116
Figure 27. ....	117
Figure 28. ....	119
Figure 29. ....	121
Figure 30. ....	122
Figure 31: .....	123
Figure 32: .....	128
Figure 33. ....	130
Figure 34. ....	131
Figure 35: .....	133
Figure 36. ....	135
Figure 37. ....	137
Figure 38. ....	138

Figure 39. .... 139

## 1.0 INTRODUCTION

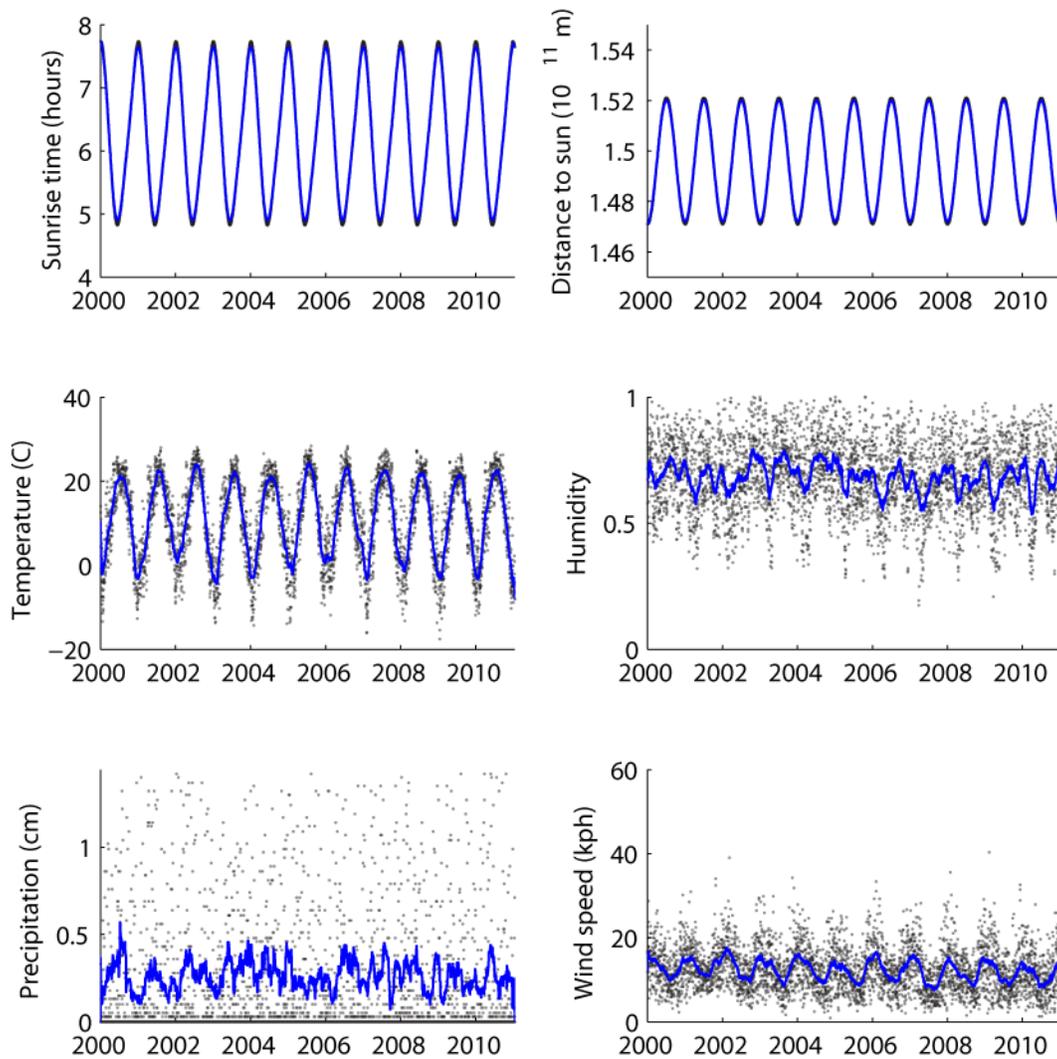
### 1.1 THE LATENT VARIABLE CONCEPT

Latent variables occur in any situation where some observed phenomena reflect a simpler, more abstract set of unobserved latent variables. For example, we could say that precipitation, temperature, and the time of sunrise all reflect the latent variable of the season. The term *latent variable model* generally refers to models of the form:

$$Y = CX$$

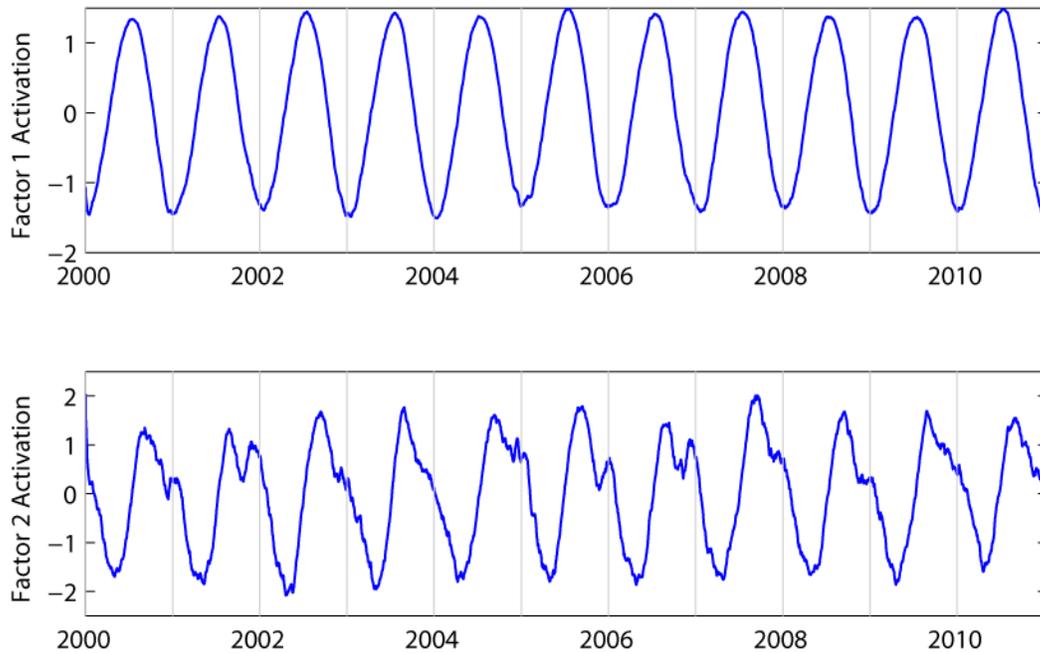
#### Equation 1

where  $Y$  is a  $q \times T$  matrix of  $q$  observed variables, over  $T$  observations;  $X$  is a  $p \times T$  matrix of unobserved latent variables; and  $C$  is a  $q \times p$  matrix defining a linear relationship between  $X$  and  $Y$ . It is understood that the true relationship may be nonlinear: in our example of season as a latent variable, there is not a linear relationship between season and snowfall. Nonetheless a linear model can be a good approximation that will allow us to discover the latent variables in a set of data. Let us consider a simple example. Figure 1 shows some seasonally varying data.



**Figure 1.** Several astronomical and weather variables for Pittsburgh, PA. Blue line is 60-day moving average.

We suspect that all these variables are associated with the latent variable *season*. We can use a standard technique, factor analysis, to try to identify this variable from the data. Factor analysis fits the maximum-likelihood  $C$  for the model in Equation 1 under the assumptions that everything is linear and the observed variables in  $Y$  are each contaminated by noise. Factor analysis requires that we specify how many latent variables are present; we are going to specify two. Figure 2 shows the resulting factors.



**Figure 2.** Latent variables for Figure 1 data, computed using factor analysis. Vertical gray lines show January 1 of each year.

We have identified two sinusoidal signals which are roughly  $90^\circ$  out of phase. This makes sense when we consider that season is a cyclic variable, so we need a sine and a cosine to fully specify it (we shall return to this concept much later in this dissertation). In this example, we know that the discovered latent variable corresponds to the earth orbiting the sun. In general, it is not necessarily the case that a latent variable will correspond to a single physical process. If we were to study sales of 6' tall trees, wrapping paper, and hardcover books we might identify *Christmas* as a latent variable, which does not correspond to a specific physical event.

This type of model does not apply to every situation. A good example of a process for which these methods are ill-suited is the tides. From the point of view of a person standing

on the shore looking down, there is a single observed variable, the height of the water, which is driven by two unobserved variables: the positions of the sun and the moon. For this person to understand the tidal process they need to realize that the single complex signal they observe (the height of the water) can be explained by two simpler cyclic signals and a nonlinear relationship. The type of model in Equation 1 is not useful here, because there are more hidden variables (sun and moon) than observed variables (water height), and the relationships involved are not well-approximated by linear models. Latent variable models make sense for neural data when we have many simultaneously recorded neurons and there is considerable correlation in the population. Broad-timescale correlation has long been observed in cross-correlations between neurons (Figure 26). This type of correlation is often considered to reflect common inputs, rather than specific anatomical connectivity between the recorded neurons (Moore et al. 1970). Broad-timescale cross-correlograms like those in Figure 26 have received less attention in the literature because they are difficult to interpret. In a technically daunting effort, Matsumura et al. (1996) recorded intracellularly in neurons in the motor cortex of both anesthetized and awake monkeys. Intracellular recordings are much more sensitive to the presence of monosynaptic connections than extracellular spike-triggered averages. Nonetheless, those authors showed that broad-timescale synchronization was the dominant phenomenon, which in the case of intracellular recording manifests itself as a broad depolarization of the intracellularly recorded neuron both before and after another neuron fires an action potential.

The most obvious way to study broad-timescale correlations is to examine cross-correlograms between pairs of simultaneously recorded neurons. With more than two

neurons being recorded at once there will be multiple combinations that can be considered:  $n \cdot (n-1)/2$  possible pairs where  $n$  is the number of neurons. With arrays of more than a few electrodes, it is no longer practical to examine cross-correlograms individually in search of a general principle. The availability of so many cross-correlograms is nonetheless an advantage, but we will need to use more sophisticated approaches to understand the data, such as the latent variable model in Equation 1. In this situation the  $Y$  matrix corresponds to firing rates of individual neurons, while the  $X$  matrix corresponds to signals that many neurons have in common.

In this dissertation, we are going to look at explicit latent variable models of neural data, but also at various technical issues that are illuminated by thinking about neural populations in terms of latent variables. We will leverage the insight we gain to build a better neuroprosthetic (1.2, Chapter 2.0) and to track neurons over multiple days (1.3, Chapter 3.0). We will use the large amounts of data we collected to show how simple models fail to provide a complete description of the firing properties of single neurons in a complex task (1.4, Chapter 4.0). Finally, we will use machine learning techniques to identify an error-related signal that has previously been shown to exist, but never viewed directly due to its complex embedding in the population (1.5, Chapter 5.0).

## **1.2 NEUROPROSTHETICS AS ESTIMATORS OF LATENT VARIABLES**

The observation that many neurons in motor cortex are broadly tuned to the intended direction of arm movement (Georgopoulos et al. 1982) led to a scheme for extracting

intention from many separately recorded neurons: the population vector (Georgopoulos et al. 1988). The population vector is a vector-based algorithm in which the normalized firing rate of each neuron drives a cursor in that neuron's preferred direction (PD). Normalization consists of subtracting a baseline rate and dividing by a modulation depth for each neuron. The formula for the basic population vector is:

$$\vec{v} = \sum_{i=1}^q \vec{p}_i \cdot (r_i - b_i) / m_i$$

**Equation 2**

where  $\vec{v}$  is the intended movement vector,  $\vec{p}_i$  is a unit-length vector representing the PD of neuron  $i$ ,  $r_i$  is the firing rate of neuron  $i$ ,  $b_i$  is the baseline firing rate of neuron  $i$  and  $m_i$  is the modulation depth of neuron  $i$ . The  $p$ ,  $b$ , and  $m$  parameters are specified according to the following model:

$$r_i = b_i + m_i (\vec{p}_i \cdot \vec{d})$$

**Equation 3**

where  $\vec{d}$  is the direction of intended movement. This model can be fit with linear regression using neural firing rates that occur when the monkey's intent is known. Fitting this model represents something of a chicken-and-egg problem, since we can't decode the monkey's intent until we fit the model, and the tuning characteristics of neurons are not always the same between normal arm movement and prosthetic control (Taylor et al. 2002). The procedure for solving this problem has evolved over the years. Our current practice in neuroprosthetic cursor control is to begin each session by presenting several peripheral targets with a stationary cursor at the center of the monitor (Jarosiewicz et al. 2008). We

assume the monkey attempts to move the cursor in the direction of the target and fit the model of Equation 3 accordingly. We periodically re-fit that model during the first few minutes of control using the accumulated data. During this time, we artificially straighten the monkey's movement using a procedure called deviation gain, in which the movement of the cursor is decomposed into a vector pointing directly at the target plus a vector pointing orthogonal to that direction (Velliste et al. 2008, Supplementary Methods). The orthogonal vector is multiplied by the *deviation gain* which is between 0 and 1. This has the effect of making the monkey's movements more accurate without actually pushing the cursor toward the target. We only use the deviation gain during the adaptation period, which tends to last around 30 trials, after which the parameters of Equation 3 are fixed.

The population vector and the associated adaptation procedures have always been based on the idea that individual neurons are approximately cosine-tuned to the direction of intended movement. When we do online spike sorting we often include "units" which are small and noisy and probably include the activity of multiple neurons. Nonetheless the extraction algorithm (the adaptation and population vector procedures) is based on the assumption that individual cosine-tuned neurons drive these signals. However, in the context of pure neuroprosthetics we are not actually interested in the neurons in and of themselves, only in the monkey's intent. The population vector can be considered an estimator of the hidden intent of the monkey, given a set of observed variables (neural firing rates). Other estimators have since been developed which better account for some properties of real neurons, such as non-uniform PD distributions, Poisson noise, and non-cosine-tuned neurons (Brockwell et al. 2004; Koyama et al. 2010; Wu et al. 2006). If we are interested in using

neuroprosthetics as a medical device, these schemes are all hobbled by the need to convert the raw signals being recorded from the subject's cortex into a series of spike times associated with individual neurons. This process, spike sorting, is a time-consuming manual system which is not plausible in a device intended to be used by patients in their own homes. Converting the time-varying voltage signal on a microelectrode into a series of spike times requires two steps: setting a threshold which detects waveforms that may be action potentials, and then sorting these waveforms into categories corresponding to one or more neurons and noise. The first step is straightforward to automate: a threshold can be set based on the standard deviation of the voltage signal, or it can be adjusted to target a specific number of waveforms per second, or we can simply set a single level across all channels. Circuits have been developed that can fit on the back of an implanted Utah array and perform threshold detection (Harrison et al. 2009). The second step, spike sorting, is more difficult to automate. A scheme based on modeling the waveform distribution as a mixture of Gaussians has been implemented (Sahani 1999; Santhanam et al. 2004) but it requires that the implanted signal processing circuitry either export all detected waveforms on all channels, or implement a spike-sorting algorithm on-chip. This increases the complexity of a device which needs to be very compact and low-power, and we will find that it is possible to avoid the need for spike sorting entirely.

Ventura (2008) demonstrated that a detected action potential could be assigned to a neuron of origin probabilistically, by assessing the likelihood that a particular neuron would have fired at that time given its tuning function and the current estimated intended direction of movement. Pseudo-sorting the spikes in this manner offers a way around the obstacle of

manual spike sorting in a clinical setting. However, if we consider that we are ultimately interested only in the hidden intent and not in the individual neurons, there is more direct solution. We can re-pose the neuroprosthetic problem by considering the intended movement direction as a latent variable. The population vector and other similar algorithms extract this latent variable by inferring it from the firing rates of individual neurons. However, it is possible to skip the intermediate step of neurons and proceed directly from a relatively unprocessed signal to the intended direction of movement. We set a threshold on each channel using a global rule that does not require manual intervention. We used a uniform voltage level across all channels that was chosen to be as unselective as possible without overloading the DSP system (Fraser et al. 2009). A similar scheme has since been described which used a threshold based on the root-mean-squared power of the signal on each electrode (Gilja et al. 2010). The theory of these schemes is that the threshold-crossing count is an observable variable which, like individual neurons, is linearly related to the latent variables of the monkey's intended movement direction. The threshold-crossing counts reflect the activity of identifiable neurons together with background activity consisting of the combined electrical fluctuations of many neurons. A neuroprosthetic system based on this scheme is the subject of Chapter 2.0 , "Control of a brain-computer interface without spike sorting."

### 1.3 USING THE REDUNDANT CHARACTERISTICS OF FIRING RATES TO TRACK NEURONS CHRONICALLY

The most obvious finding when latent variable models are applied to neural data is that the firing rates of a population of neurons, when smoothed at long timescales, contain considerable redundancy (Yu et al. 2009). It implies that there is a low-dimensional representation that explains a large portion of the firing rates of many single neurons, at least in the context of a specific task. This redundancy is expressed in the cross-correlograms between pairs of neurons as large triangular peaks at zero-offset, which are thought to reflect the mass action of many common inputs to the two neurons (Moore et al. 1970). The latent variable concept can be thought of as a mathematical formulation of a neural network that is dominated by the mass action of many neurons. If the latent variable model is consistent in a given population from day to day, then these cross-correlograms will also be consistent. That opens up the possibility that we might use these cross-correlograms as an identifying feature to recognize when the same neuron has been recorded in multiple sessions separated by one or more days.

The notion of recording from the same neurons chronically is not new (for example, Schmidt et al. 1976). However, all methods for labeling the same neurons over multiple session have either been subjective (Chestek et al. 2007; Ganguly and Carmena 2009; Greenberg and Wilson 2004; Jackson and Fetz 2007; Schmidt et al. 1976; Williams et al. 1999) or have been based on weak classifiers that are subject to severe error rates (Dickey et al. 2009; Tolias et al. 2007). The fundamental problem is that the average shape of the action potential from one sorted unit provides only a limited amount of information. Cross-

correlograms between neurons, if they are consistent from day to day, represent a powerful potential source of information because they are so numerous: for  $n$  neurons there are  $n \cdot (n-1)/2$  cross-correlograms. The supposition that long-timescale features of cross-correlograms might be consistent is borne out in the data. It represents an extremely useful technical consequence of the latent variable concept.

In Chapter 3.0 , “Recording from the same neurons chronically in motor cortex” we take advantage of the ability to track neurons chronically to address a fundamental question: are the firing properties of neurons in motor cortex stable over time? There is a divergence of opinion in the literature as to whether the tuning characteristics of neurons are more or less fixed, or whether they evolve continuously as part of a dynamic network. Li et al. (2001) show examples of primary motor cortex neurons which seem to maintain a changed preferred direction after macaques performed a practiced visuomotor rotation task, but they are only able to record these neurons for a single session and thus it is unknown whether these changes remain the next day. Ganguly and Carmena (2009) recorded primary motor cortex neurons chronically and identified a subset which they deemed stable across periods of up to 19 days. They showed examples of cells with stable tuning to arm movement direction across the study period. In the context of a brain-computer interface paradigm where movement of a cursor was directed by brain activity, they showed that the tuning of neurons to movement direction evolved and then stabilized as performance in the task improved. Chestek et al. (2007) recorded neurons in primary motor cortex for single sessions, as well as some neurons which they deemed stable across two sessions. They found small variations in tuning to arm movement which were potentially explained by a similar level of variation in the movement

itself. Greenberg and Wilson (2004) recorded premotor cortex neurons and identified a small number of cells that seemed stable across periods of up to 9 days. They found that most neurons were consistently tuned to the abstract characteristics of a delayed-response task, but a minority changed their tuning across sessions. With the ability to track the same neurons over many weeks, we were able to demonstrate that there are small changes in PD from day to day, but these small changes do not add up into large changes over time.

#### **1.4 COMPLEXITY OF TUNING FUNCTIONS IN A RICH MOVEMENT TASK**

Early studies of the activity of motor cortex neurons in awake behaving monkeys focused on very simple tasks, such as squeezing a force sensor, flexing the wrist, or pulling a lever (Evarts and Tanji 1976; Fetz and Cheney 1980; Smith et al. 1975). Investigators focused on the corticospinal tract and the motor cortex's role as a driver of muscle activity. When motor cortex neurons were recorded during a more complex two-dimensional reaching task, they were found to be broadly tuned to the direction of movement, a comparatively abstract parameter (Georgopoulos et al. 1982). However, there is heavy correlation between EMGs, the direction of movement, and kinetic and kinematic parameters. This has led to an ongoing controversy over what variables are and are not encoded in motor cortex. Depending on the details of the behavioral paradigm, different features such as muscle activity or direction of movement can appear to be dominant. A recent study (Churchland and Shenoy 2007) provided a somewhat radical perspective on this issue. Churchland and Shenoy recorded single neurons in the primary motor cortex and dorsal premotor cortex of macaques

performing a center-out reaching task with multiple target directions and distances. They trained the macaques to an unusually high degree, so that their movements would be extremely consistent across repetitions. They also discarded any trials where the movement was unusual, and they recorded individual neurons for many more trials than were necessary simply to get estimates of tuning functions. Because of these factors, they were able to make highly accurate estimates of how a neuron's firing rate would evolve over time during a specific type of reach. They compared these firing rate profiles to EMGs and a number of kinematic parameters, and found that these parameters explained only part of the neurons' firing rates. Furthermore, they were able to show that it was mathematically impossible for any small set of variables to explain the details of all the firing rate profiles. Those authors suggest an unusual conclusion: perhaps these neurons do not actually encode any abstractly meaningful parameters.

We sought to examine the same issues using a 26-target three-dimensional reaching task. Where Churchland and Shenoy studied the temporal details of firing rates, we studied the spatial details. Additionally, because we used fixed electrode arrays, we recorded many low-firing-rate neurons that would likely have been missed when electrodes are introduced daily and an investigator searches for neurons to record while advancing the electrode. We found that individual neurons were sensitive to both the direction of movement and the phase of the task. The distribution of preferred directions in 3D space was highly non-uniform, with a bias towards the sagittal plane, and the distribution of preferred phases was dispersed across all phases of the task, including before and after movement occurred. These tuning functions were not accurately described by kinematic parameters. In an effort to

summarize the meaning of population activity, we reduced the many simultaneously recorded neurons into a small set 6-15 latent variables and performed information theoretic analysis. We determined that activity was influenced by both the direction and target of movement throughout all phases of the task, with a modest tendency for more directional information around the period of peak movement speed.

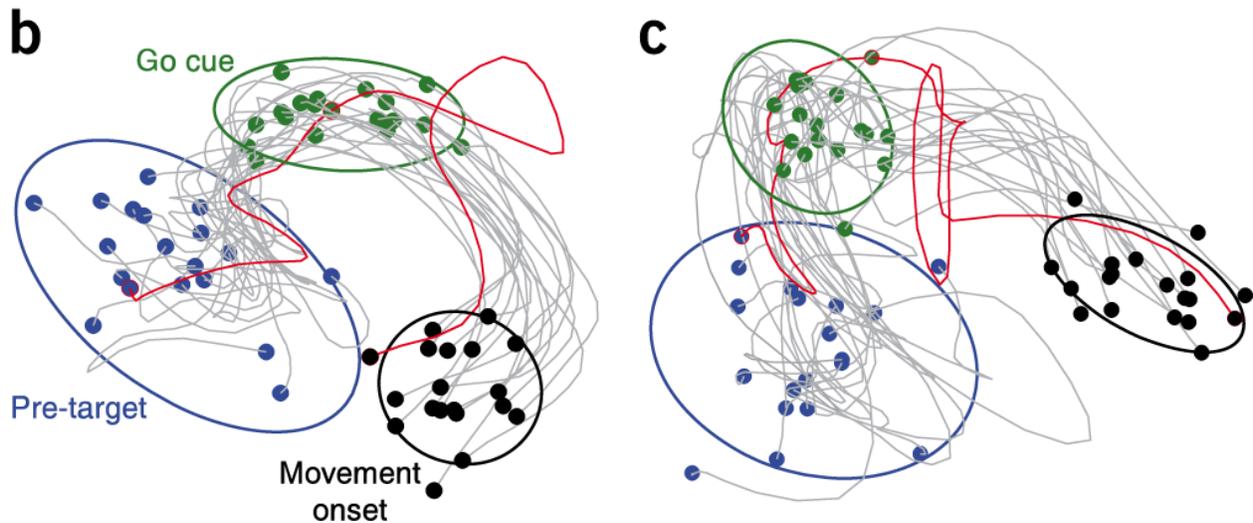
## **1.5 USING LATENT VARIABLES TO CHARACTERIZE BRAIN FUNCTION**

The latent variable concept is not only useful for motivating technical advances. Latent variable models are often used in system identification, where the dynamical system driving a set of observed variables is modeled automatically using machine learning techniques (Shi and MacGregor 2000). Latent variables are applicable whenever the observed variables are driven by a simpler underlying system. The brain is obviously too complex a system to be subjected to true system identification when only a few hundred neurons are recorded, but there may be aspects of brain function that are implemented by a large number of neurons but fundamentally reflect a simple low-dimensional system. Under these circumstances, it would be feasible to record enough neurons participating in the same low-dimensional system to identify some dimensions automatically.

The representation of direction of movement in motor cortex seems to be an example of this type of system. Direction of intended arm movement explains a meaningful fraction of the variance in the firing rate of neurons in motor cortex even though it only has 2 or 3 degrees of freedom. The population vector extracts a directional signal from this population;

it can be thought of as a way to extract latent variables (directional components of movement) where we have specified an explicit model for the  $C$  matrix in Equation 1 (the cosine tuning model, which can be formulated as a linear operation). The fact that the population vector works means there is a directional signal in this population, at least in specific contexts. But it doesn't exclude the possibility that other signals are also present. We need a different method to extract those other signals. Instead of specifying the  $C$  matrix based on an explicit model of how the neurons are tuned, we can use machine learning methods to discover  $C$  and  $X$  in simultaneously recorded data.

In a prior example of this approach, Yu et al. (2009) developed a method called Gaussian process factor analysis (GPFA) to extract a latent variable representation from a population of 61 simultaneously recorded neurons in motor cortex. In Churchland et al. (2010) this method was applied to population activity in dorsal premotor cortex during a delay-period movement task. After using GPFA to extract the matrix of latent variables  $X$  (from Equation 1,  $p$  latent variables  $\times T$  timepoints), they considered each column of  $X$  as a position in  $p$ -dimensional space, and showed that that this position followed a characteristic trajectory each time the monkey made a movement (Figure 3).



**Figure 3.** From Churchland et al. (2010). A low-dimensional latent variable representation of neural population activity follows a characteristic trajectory in each trial of a delay-period movement task. Single trials with unusually long reaction times also have unusual trajectories in this space (red traces). (b) and (c) represent different days of recording. Adapted by permission from Macmillan Publishers Ltd: Nature Neuroscience (Churchland et al. 2010), © 2010.

The activation of latent variables always moved to a specific position in  $p$ -dimensional space during the pre-movement period. When it was outside of that region when the go cue was given, the monkey had longer reaction times, indicating that the activations of latent variables had to be at specific values before the monkey could proceed with movement. This paper was primarily about the decline in variability associated with the presentation of a stimulus, but it was a significant application of the latent variable concept because it showed that it was possible to extract a low-dimensional set of variables that seemed to be meaningful. However, it was limited to a single day's recordings, and the possible meaning of

individual latent variables was not explored. That is the issue at the center of Chapter 4.0. We take advantage of the cyclic nature of an ellipse-drawing task to selectively extract latent variables that are sinusoidal. This approach produces a small set of latent variables consisting of simple features that can be studied directly. These features seem to correspond to the horizontal and vertical direction of movement, and an error signal that anticipates the need for corrective movement.

## **1.6 DEVELOPING A PLATFORM FOR WIDESPREAD SIMULTANEOUS RECORDING**

The Utah array (Maynard 1997) has been invaluable for providing reliable simultaneous recordings of many neurons. However, it is subject to some significant limitations. One Utah array can only record in one brain area, and the total number of simultaneously implanted arrays is limited by our ability to place multiple connectors on a single monkey's head. Due to the short length of its electrodes (1.0-1.5mm), the Utah array cannot access brain areas that are located in the sulci of the cortex, and it cannot access deep brain areas at all. To address some of these issues, we developed a new platform for simultaneous recording of many neurons across many brain areas, with much longer, individually adjustable electrodes. Unlike the Utah array, which is implanted under the bone with a lead wire extending to a head-mounted connector, our platform is based on recording chamber technology. In order to achieve our goals, it was necessary to significantly improve the performance of existing recording chambers, which are limited in size by the need to anchor the edges with dental acrylic, and which are prone to catastrophically detach from the skull in some monkeys.

While our efforts are incomplete, we were able to make significant progress in the improvement of this technology. Recording chambers with vastly improved performance were developed and used in four monkeys without any failures. An array of 256 individually moveable electrodes was developed based on a new, highly compact and cheap to manufacture linear motion mechanism. It can be implanted into an existing recording chamber without additional surgery. Full-scale prototypes were constructed and successfully recorded neural activity. The details of this system are described in Chapter 6.0 .

## 2.0 CONTROL OF A BRAIN-COMPUTER INTERFACE WITHOUT SPIKE SORTING<sup>1</sup>

In this chapter, our application of the latent variable concept to neural data was at an early stage. By thinking about neuroprosthetics in terms of extracting hidden signals, we were able to skip an intermediate step of identifying individual neurons and create a simpler scheme for neuroprosthesis. It is a purely technical work; its purpose is to bring brain-computer interfaces a step closer to being useable in the real world.

### 2.1 ABSTRACT

Two rhesus monkeys were trained to move a cursor using neural activity recorded with silicon arrays of 96 microelectrodes implanted in the primary motor cortex. We have developed a method to extract movement information from the recorded single and multi-unit activity in the absence of spike sorting. By setting a single threshold across all channels and fitting the resultant events with a spline tuning function, a control signal was extracted from this population using a Bayesian particle-filter extraction algorithm. The animals achieved high-

---

<sup>1</sup> This chapter was published as **Fraser GW, Chase SM, Whitford A, and Schwartz AB**. Control of a brain-computer interface without spike sorting. *J Neural Eng* 6: 055004, 2009.

quality control comparable to the performance of decoding schemes based on sorted spikes. Our results suggest that even the simplest signal processing is sufficient for high-quality neuroprosthetic control.

## 2.2 INTRODUCTION

This paper demonstrates an alternate strategy for processing signals from intracortical electrodes for prosthetic control. Current brain-computer interfaces based on intracortical electrode arrays use extracellular action potentials processed with spike-sorting strategies to generate a control signal (Hochberg et al. 2006; Musallam et al. 2004; Santhanam et al. 2006; Taylor et al. 2002; Velliste et al. 2008; Wessberg et al. 2000). A threshold voltage is set for a particular electrode and each time the voltage potential exceeds that threshold, a waveform snippet of the time-varying potential is recorded. These waveforms are divided into one or more categories, in the hopes of identifying individual cells or separating out cell-related activity. During a typical neuroprosthetic control experiment, a human operator will view the ongoing waveforms for a brief period on each channel at the beginning of each day's session and attempt to identify distinct waveforms by setting the spike-sorting parameters of a digital signal processing device. The need for human intervention is an obstacle to bringing neural prosthetics from the lab to the clinic, as is the complex digital signal processing employed in current brain-computer interfaces.

A separate class of brain-computer interfaces (BCIs) use low-frequency signals from external electrodes (EEGs, for example Wolpaw et al. 1991), surface macroelectrodes (ECoGs,

for example Leuthardt et al. 2004), or intracortical microelectrodes (LFPs, for example Mehring et al. 2003, Pesaran et al. 2002). These techniques and those based on spikes activity can be ordered according to their spatial resolution, with EEGs at one end and spike activity at the other. The BCIs with the greatest number of degrees of freedom and accuracy have been operated with spike activity (Taylor et al. 2002; Velliste et al. 2008; Wessberg et al. 2000).

Recent papers have suggested other schemes for extracting a control signal from intracortical recordings. Stark and Abeles (2007) showed that multiunit activity, reflected in the power between 300-6000 Hz, is a good predictor of upcoming hand movement in macaques. Ventura (2008), using simulated data, extracted movement intent from mixtures of tuned units, with an accuracy comparable to traditional spike-sorting approaches. Both of these papers suggest that there is information to be found in relatively unprocessed signals from microelectrodes in the motor cortex. In the current study, we report on the feasibility of operating a brain-computer interface without spike sorting with recordings from two monkeys chronically implanted with microelectrode arrays.

## **2.3 METHODS**

### **2.3.1 Chronic microelectrode implant**

Monkey A, a male rhesus macaque, was implanted in January of 2007 with a single 96-channel array (Blackrock Microsystems; Maynard et al. 1997) on the convexity of the motor cortex next to the central sulcus, with the lateral edge of the array ~2mm medial to the genu

of the arcuate sulcus. The recording sessions reported here were done in June of 2008. Monkey C, also a male rhesus macaque, was implanted in February of 2009 in the same location. The reported sessions were recorded 10 weeks later. Monkey A's array gave as many as 130 distinguishable cells and multineuron combinations during the period of best recordings. At the time of this experiment, June 2008, there were 1-2 clear single units, 1-6 probable single units, ~40 fair neurons/multineuron combinations, and 15-30 poor, multineuron traces on a typical day of recording. Monkey C's array gave ~75 fair neurons/multineuron combinations and ~5 probable single units on the day of this experiment.

### **2.3.2 Behavioral task**

Prior to the implant, the monkey was trained to do a center-out arm movement task in a virtual environment. It sat in front of a stereoscopic computer monitor (DTI, Rochester NY) with one arm gently restrained, the other free to move with an infrared marker taped slightly distal to the wrist so that its position could be monitored with a motion capture system (Northern Digital, Waterloo ON, Canada). The position of the infrared marker drove the position of a green cursor displayed on the screen in a virtual environment. Blue spheres were displayed as targets at various points in the three-dimensional environment and the monkey was rewarded with a droplet of water for making and holding contact with the targets. In each case, the monkey first made contact with a central target and touched the target for a required time of 200—300 ms, selected randomly from a uniform distribution. A peripheral target was then randomly selected from a queue of remaining targets. In the

experiments described in this paper, we used a set of 16 targets, their centers arranged evenly in an circle on the plane of the monitor with a radius of 85 mm for monkey A and 79 mm for monkey C. The monkey moved to the peripheral target within a limited time period—1600 ms in this study—and held contact for a required time of 0-200 ms. The cursor and the target had radii of 8 mm for monkey A and 9 mm for monkey C. The variable hold period was long enough that the monkey couldn't consistently succeed by moving straight through the target—it had to stop or drastically slow its movement as it approached the target. Once a target was hit successfully, it was dequeued from the remaining targets.

After the monkey was implanted and the neural signals were deemed large enough and stable enough to sort (about three weeks for both monkeys), the animal began to use the brain-computer interface. Both arms were restrained and the cursor was driven with neural activity. To decode intent, it is necessary to determine the tuning parameters of the neurons being recorded. In this study, we made a first estimate by running the task with null tuning parameters. In brain control, the cursor was placed on the central target at the beginning of each trial. Then the peripheral target was presented after the expiration of the central hold period. Initially, the monkey was unable to move the cursor with null tuning parameters and failed each trial. Nevertheless, the monkey modulated its neural activity consistently for each target, making it possible to estimate tuning parameters and begin real-time neural decoding. During this initial period where the control parameters are adapting, we enhanced the straightness of the trajectories by artificially reducing the deviation from the ideal, perfectly straight movement. At each timestep, the prediction of the decoding algorithm was decomposed as the sum of a vector straight towards the target and a vector orthogonal to it.

The orthogonal vector was then multiplied by the *deviation gain*: between 0.1 and 0.5 in this study. Adjusting the deviation gain is a highly subjective process; we ramp up the deviation gain as the model parameters are re-fit with more data and the decoding becomes more consistent. Choosing how much deviation gain to apply is a matter of balancing the need for straight trajectories with the tendency of the monkey to modulate erratically if it realizes that the trajectories come out straight regardless of the consistency of its modulation. We need straight trajectories while the decoding parameters are being fit so that we can accurately compute the tuning functions of signals. At the same time, the monkeys have shown a tendency to produce inconsistent modulation if the deviation gain is too strong for too long. The experimenter attempts to balance these priorities. Typically the deviation gain is 0.1-0.25 for the first 16 targets, 0.5 for the next set, and off thereafter. If the trajectories are still erratic we will keep it on longer. This procedure was used only in these adaptive sessions. Deviation gains were only used for a short period at the beginning of a daily experimental session for calibration purposes. Subsequent control used no deviation gain.

### **2.3.3 Signal Processing**

Signals were buffered, amplified, bandpass filtered to 250-8000 Hz, and processed using a 96-channel Plexon Multichannel Acquisition Processor (Plexon Inc, Dallas TX). The DSP system was configured so that it would register all events crossing a negative threshold in the downward direction and send their times to one of the task-management computers. In monkey A, a threshold was set across all channels at -37.5  $\mu\text{V}$ , except 10 channels where the power of the time-varying voltage was so high that it constantly saturated the A/D converter.

On those channels the gain was reduced until the clipping was under control, effectively putting the thresholds in the (-)40-80 uV range. In monkey C, the signals had higher amplitude and the threshold was set at -60 uV for all channels. In both animals the threshold was chosen to be as low as possible without severely overloading the capacity of the DSP system on too many channels. There was sufficient variation in the signals to cross the threshold on 95/96 channels in monkey A, and 78/96 channels in monkey C. It should be noted that because of the difference in the age of the implanted arrays (10 weeks in monkey C versus 1.5 years in monkey A), the signals are very different. In the very old array of monkey A, the amplitude of the sortable action potentials are much closer to the amplitude of the background noise. In the very young array of monkey C, the action potentials stand out much more strongly from the background, but there are often more active neurons observable on a given channel. Hence the threshold had to be set higher in monkey C to avoid capturing so much activity that the capacity of the DSP system would be overloaded. This higher threshold resulted in 18 channels that observed no threshold-crossings. For the same reason monkey C showed less of a difference in baseline rate for sorted versus unsorted sessions, as evidenced in Figure 8. It should be noted that in both monkeys the resulting signals were not comparable to single-unit recordings; the waveforms picked up by the threshold included large amounts of background activity and noise on nearly every channel where the signal crossed threshold at all. This is evidenced by the fact that threshold-crossings showed consistently higher baseline rates and modulation depths than units sorted on the same channels (Figure 8).

### 2.3.4 Decoding algorithm

We used a Bayesian Monte-Carlo estimation algorithm very similar to the one described by Brockwell and colleagues (2004). This algorithm, called a particle filter, solves the problem of decoding neural intent by offering a firing-rate model for each recorded unit, then estimating the movement intent most consistent with this forward model *and* the recent history of movement. The particle filter can be intuitively understood in terms of a hypothesis space of movement intents that the monkey might have at the present timestep. The particle filter uses modulated control signals: a neuron whose firing rate increases for a particular movement direction, or in this study, a time-varying voltage signal which crosses a threshold more or less often depending on the current intended movement direction. A single modulating signal gives the particle filter information about which regions of the hypothesis space are consistent with its current level of activity. One signal alone will leave ambiguity: a high-firing neuron indicates movement in its preferred direction, or perhaps movement somewhat off its preferred direction at a higher speed. These signals are noisy, so one signal may simply be wrong at the present moment. The particle filter searches the hypothesis space at several hundred points—particles—for the region that is most likely given all the current activity levels. The distribution of particles in one timestep is generated from their position in the previous timestep, which enforces our assumption that movement in the current timestep is similar to movement in the previous timestep. This distribution is biased according to the likelihood of the hypothesis that each particle had found. Thus the particles form a cloud that follows the most likely region of the hypothesis space from one timestep to another. This process is diagrammed in Figure 4. Using a particle filter as an extraction

algorithm is simply a matter of identifying a modulated signal and specifying a tuning function for it. Brockwell and colleagues used an exp(cosine) firing rate model where each sorted unit had a baseline rate, a single preferred direction, and an adjustable-width tuning function. Our new approach is based on a firing rate model that makes fewer assumptions about the shape of the tuning function.

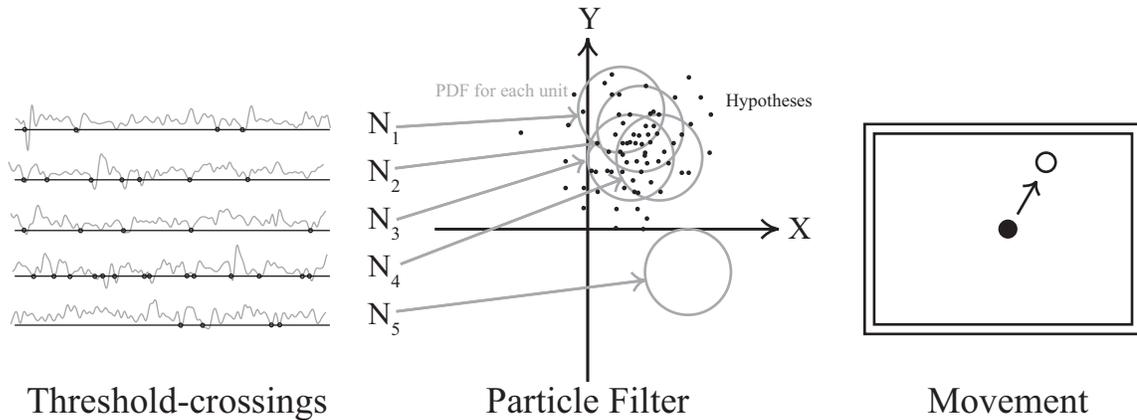
$$\lambda_i = b_{i,0} + s \sum_{j=1}^8 w_j \cdot f(\theta \cdot 4/\pi - j) \quad (1)$$

Here,  $\lambda_i$  is the total rate of threshold-crossing on channel  $i$ ,  $b_{i,0}$  is the baseline firing rate of that channel;  $s$  is the speed of movement; and  $\theta$  is the angle of movement in the X-Y plane.  $f$  is a cubic spline basis function which is being shifted and stretched by the  $\theta \cdot 4/\pi - i$  formula. It has the following mathematical definition:

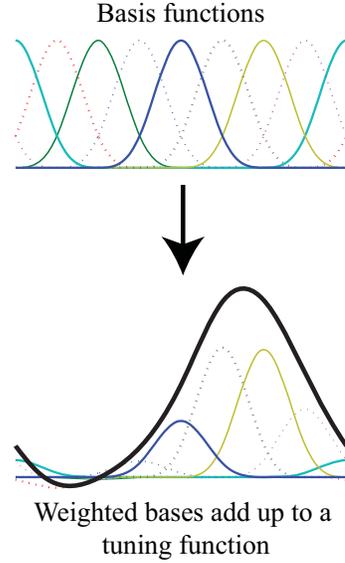
$$f(x) = \left\langle \begin{array}{l|l} x < -2 \mid 2 < x & 0 \\ -2 \leq x < -1 \mid 1 < x \leq 2 & (2 - |x|)^3 \\ -1 \leq x \leq 1 & 1 + 3(1 - |x|) + 3(1 - |x|)^2 - 3(1 - |x|)^3 \end{array} \right\rangle$$

$f$  is a bell-shaped function spanning the range [-2,2]. It represents a cubic spline basis function; by shifting and spacing these basis functions at intervals of 1 and adding them up, we get a set of basis functions that span the entire circle of movement directions.  $w_i$  is the weight of a particular spline basis function, fit by a regression model. The effect of the summation portion of the firing model is to fit a smoothing spline to firing rate as a function of the intended movement direction in the X-Y plane. This process is diagrammed in Figure 5. Our model effectively states that firing rate is equal to a baseline rate plus an 8-knot spline function in polar coordinates that expands and contracts according to the speed of movement. The  $w_i$  parameters which determine the shape of the spline function are fit using

linear regression. Our decoder is written in Matlab (Mathworks, Natick MA) and uses its GLM library to fit the model with an identity link function.



**Figure 4.** Extracellular activity-related signals are recorded from motor cortex activity ( $N_1$ - $N_5$ ). The number of threshold-crossings is counted on each channel for each 33 ms interval (left panel) and fed into a particle filter extraction algorithm that compares bin counts to the known tuning functions of those channels. Each bin count points to a probable region of the hypothesis space of movement trajectories (middle panel). The particle filter maintains a set of hypotheses about movement (middle panel, small dots). At each timepoint these particles are moved randomly by a distance drawn from a Gaussian distribution, then resampled according to the probabilities indicated by the counts  $N_1$ - $N_5$ . In this manner the particle cloud is dragged around by the probable regions from the  $N_1$ - $N_5$  counts. An outlier bin count which disagrees with the rest of the population (above,  $N_5$ ) will have little influence over the particle cloud. To generate an estimate of the global movement intent we simply take the mean of all the particles.



**Figure 5.** A tuning function is fit as a weighted sum of spline basis functions. The tuning function of one channel is shown as the thick black line. It is fitted as a sum of shifted bell-shaped basis functions (top), each of which spans a  $\pi/4$  radian section of movement directions. The basis functions are scaled (bottom) and added to produce the fitted function.

The empirical firing rate of cell  $i$  is taken to be poisson-distributed with mean  $\lambda_i$ . Thus the probability of observing a specific bin count  $n_i$  given a firing rate  $\lambda_i$  from equation 1 is given by the equation:

$$P(n_i|\lambda_i) = \text{poisspdf}(n_i, \lambda_i \cdot \Delta t)$$

where *poisspdf* gives the poisson probability density function with mean  $\lambda_i \cdot \Delta t$  evaluated at  $n_i$  ( $\Delta t$  is the width of the bin). The particle filter also incorporates an assumption about the way velocity changes over time. In both Brockwell *et al.* and this paper, it is assumed that the velocity at one timepoint is related to the previous timepoint according to a Gaussian distribution. The standard deviation of this distribution depends on the length of the time

step and the assumptions of the experimenter. In a brain-control experiment it should correspond to the distance to the target and the movement duration. Here, we assume that velocity changes with a standard deviation of 67 mm/s over one second. The prior probability of the x-component velocity at the current timestep,  $\bar{x}$ , given the x-component of velocity at the previous timestep,  $x$ , is a function of the length of the timestep  $\Delta t$  and the standard deviation of movement per second  $\sigma$ :

$$P(\bar{x}|x) = \text{normpdf}(\bar{x}, x, \sqrt{\Delta t} \cdot \sigma) \quad (2)$$

where *normpdf* describes the normal distribution probability density function for the distribution with mean  $x$  and standard deviation  $\sqrt{\Delta t} \cdot \sigma$ , evaluated at  $\bar{x}$ . The problem of decoding an intended movement direction from the threshold-crossing activity is then a matter of maximizing the probability distribution defined by equations (1) and (2):

$$P(\bar{x}, \bar{y}|x, y, n_1, \dots, n_m) \sim P(\bar{x}|x) \cdot P(\bar{y}|y) \cdot P(n_1|\bar{x}, \bar{y}) \cdot \dots \cdot P(n_m|\bar{x}, \bar{y}) \quad (3)$$

$x$  and  $y$  are set to 0 at the beginning of each trial (during the central hold period);  $n_1, \dots, n_m$  are the observed bin counts for the  $m$  channels being used. The absolute probability of the left-hand-side of equation 3 is scaled by additional terms but since we are only interested in the relative maximum we can leave them out. The particular filter uses a set of particles—points in the X-Y velocity coordinate space—to estimate the maximum of this distribution. Each time a new set of bin counts arrives, the particles are moved probabilistically. First, each particle makes a random movement whose destination is chosen from a normal distribution with mean at the previous location of the particle and standard deviation of  $\sqrt{\Delta t} \cdot \sigma$ . This corresponds to the first two terms of the right-hand-side of equation 3. Second, a probability is calculated for each particle equal to the remaining terms of equation 3. The entire

population of particles is then resampled from itself, with replacement, according to these probabilities. This means that high-probability particles are more likely to reappear after resampling, and that some particles will end up being represented multiple times in the new population.

These steps are constructed so that the probability distribution of the position of a single particle is exactly equal to the distribution being estimated. The entire population of particles acts as a proxy for the distribution being estimated. We generate a single estimate of movement for real-time brain control by taking the mean of the entire population of particles. The accuracy of decoding increases with the number of particles, but so does the computational complexity. We used 400 particles, which we found to be the highest number where the computation could be completed reliably in the time between bin counts. With a simulated population we found that the quality of control did not become noticeably worse until the number of particles dropped below 50.

The described decoding algorithm has a number of parameters (the  $b_0$  and  $w_i$  terms in the model) that must be fit from the actual tuning of the neurons. In the reported unsorted neural control session for monkey A, we used a previous day's parameters as the initial parameters for decoding. At the beginning of the day we ran an adaptive session where we re-fit the  $w_i$  and  $b_0$  terms. We gave 4 sets of 16 targets, re-fitting the model to the cumulative set of data for that day after each set. For the purposes of linear regression, the intended movement was taken as the idealized vector from the center of the workspace to the peripheral target. For monkey C we set the initial parameters to 0, which means that for the first 16 targets the cursor did not move. Nevertheless the monkey looked at the monitor and

modulated its neural activity sufficiently to get a set of parameters to move during the remainder of the adaption period.

The sorted session for monkey C was done in exactly the same way as the unsorted session, with parameters initially set to 0. The sorted session for monkey A was done somewhat differently; parameters were initially 0 but the extraction algorithm being used was a variation on the Bayesian inference decoder where Laplace's method of integration is used in place of the Monte Carlo particles of the particle filter. We have found the accuracy of the two methods is similar; the main difference is that the Laplacian integration method is more computationally efficient.

## **2.4 RESULTS**

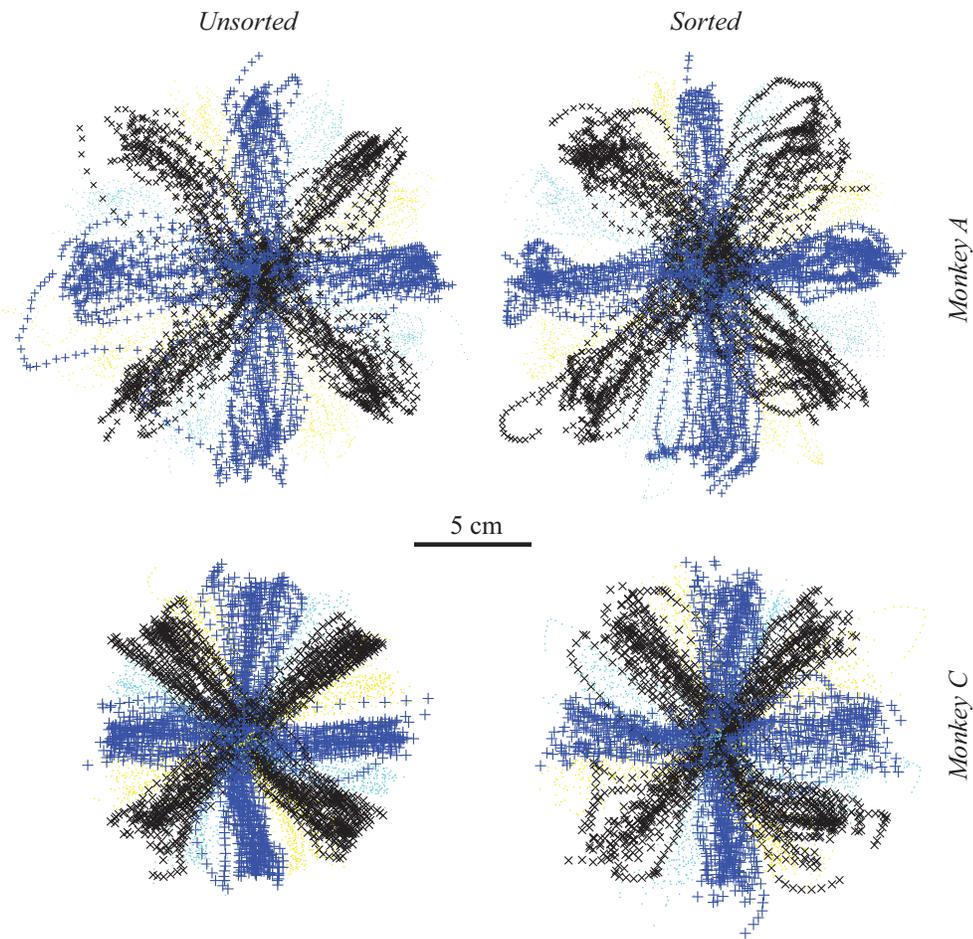
Four datasets are reported here, unsorted and sorted for monkeys A and C. In all cases cursor movement was confined to the two-dimensional plane of the monitor, though the VR system displayed objects in three dimensions. Monkey C did the unsorted control session first, followed by an 80 minute break, followed by the sorted control session. It had less than half its daily water quota in the first session so its motivation level was still high during the second session, assessed by the fact that it engaged in the task continuously for the entirety of both sessions. It used exactly the same decoding algorithm for both sessions, except that the extraction algorithm was being fed counts of threshold-crossings in the first session and counts of sorted spikes in the second. Monkey A did the unsorted session two weeks after the sorted session. The sorted session contained two breaks where the decoding algorithm was

seamlessly switched to population vector algorithm (Taylor et al. 2002) for 80 and 112 trials. These trials are excluded. This particular sorted session was chosen for comparison because it was also done in the middle of the week when the monkey's motivation level tends to be highest, it used a Bayesian decoding algorithm very similar to the particle filter, and it was contemporaneous with the other session.

There is inherent variation in the quality of brain-control decoding from one session to the next, even when they are performed on the same day. When the parameters are initially set to zero and then fit from a limited set of movements during the adaptation period, the accuracy of the fit depends on the way that the animal modulates its activity during those particular trials. Therefore, it is not possible to make an exacting comparison between the quality of control in the sorted versus unsorted session. We can only evaluate whether the unsorted scheme works approximately as well as a decoder based on sorted units. The primary metrics we have for the quality of control are our subjective impression of Figure 6; the success rate of the animal; the speed of movement; and the straightness of the trajectories.

**Table 1.** Decoder performance.

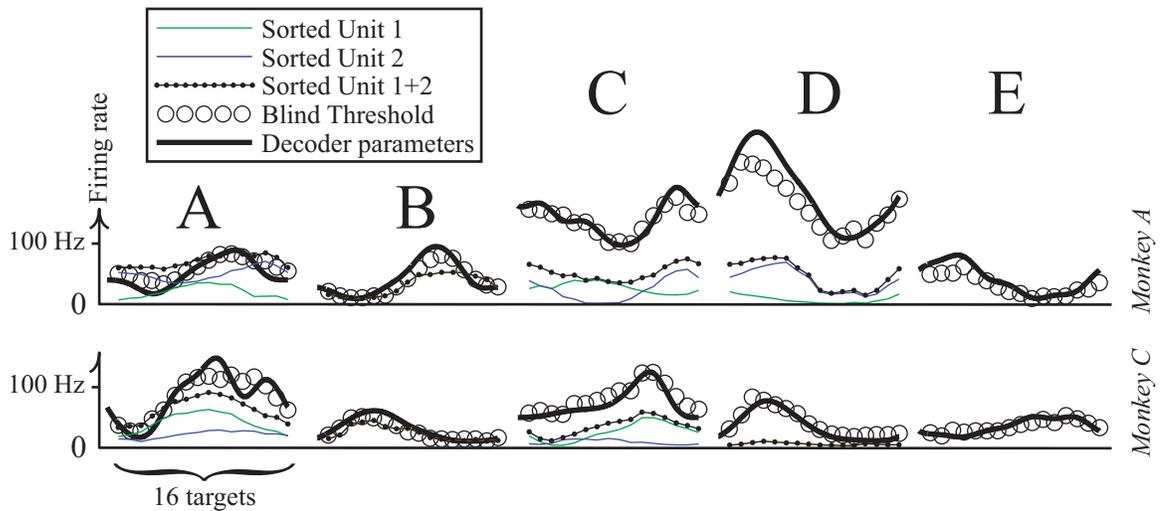
	Monkey A		Monkey C	
	Unsorted	Sorted	Unsorted	Sorted
Success	78%	93%	96%	84%
Rate				
Speed	815 ms	932	807 ms	735
		ms		ms
Straightness	1.12	1.1	1.11	1.17



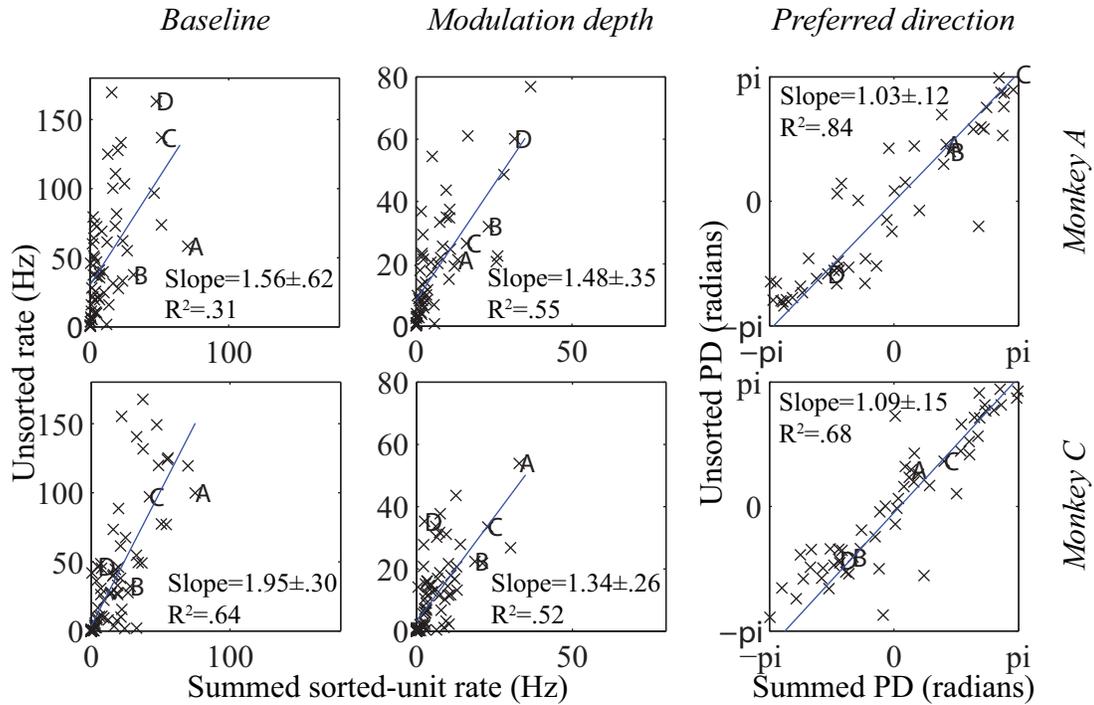
**Figure 6.** Superimposed movement trajectories as the monkey does the center-out task for 16 targets using unsorted threshold-crossing signals (left) or sorted activity (right). This plot shows position samples at 30 Hz and incorporates all successful trials on a single day between the start of the central hold period and the end of the peripheral hold period. The colors and symbols are varied according to the target direction in the current trial. The upper-left plot contains 384 trials, the upper right shows 233; lower left 480; lower right 512. The trials for each condition were done in one contiguous block, except that in the monkey A / sorted condition there were two breaks in the trials where the decoder was switched to population vector algorithm (Taylor et al. 2002) for 80 and 112 trials (these trials are excluded). Success rates were 78%, 93%, 96%, and 84% for A/unsorted, A/sorted, C/unsorted, C/sorted. Because of the vagaries of monkey motivation and the fact that no two adaptation sessions are the same, it is not possible to make an exacting comparison of performance here. We can only observe that both algorithms produce qualitatively good control.

Success rate is simply the proportion of trials where the animal succeeds according to the central hold, movement time, and peripheral hold criteria defined in the methods section. Speed is the time between the peripheral target being presented and the cursor contacting it. Straightness is the total path length of the cursor from when it left contact with the no longer visible central target to when it made contact with the peripheral target, divided by the length of a perfectly straight line between the endpoints of the same path. The above table gives the median speed and straightness for all successful trials.

We can treat the threshold-crossings on a particular channel as though it were a neuron and define a tuning function for it—an estimate of what the threshold-crossing rate will be when the monkey moves in a particular direction at a particular speed. The  $w_i$  parameters of the particle filter decoder describe such a tuning function; Figure 7 compares the assumptions of the decoder with the empirical per-target firing rates observed in the control session. The thick black lines and the black circles in Figure 7 show the comparison between the model's spline fit and the later activity of the channels. The channels shown are selected from the better-tuned half of what we recorded, but are representative in terms of the model's fit and the lack of multimodality in the tuning functions. The particular examples shown in Figure 7 are indicated on the scatterplots in Figure 8.



**Figure 7.** 5 examples of 16-target tuning functions for neurons, sums of neurons, and unsorted threshold-crossings. Each of these 5 examples shows various tuning functions from a single channel. The Y-axis shows the firing rate of a pseudo-unit during the movement period of the task. The X-axis gives the angle from the start point in the center of the monitor to the peripheral target. The multiple plotted lines illustrate how one or two identifiable neurons and background activity add up into the signal we observe with the blind thresholding we used. The green and blue lines are the tuning functions of sorted units. The connected-circles line is the sum of these units. The open black circles show the tuning function of the blind-threshold pseudo-unit used in unsorted control. The thick black line shows the fitted spline function that the particle filter decoder is using, which is fit from a separate dataset at the beginning of the recording session. In these 5 example channels we can see the different relationships between the identifiable units on a channel and the signal you get when you set a blind threshold. Unsorted activity is sometimes well-explained as the sum of units on that channel (A,B); most often it has the same shape as the sum of units but a higher baseline rate and modulation depth (C,D); and sometimes a channel without sortable activity will show strong modulation in the hash (E).



**Figure 8.** The summed activity of all the units on each channel was fit as a linear model of the X- and Y-components of target direction plus a constant. The same was done with the unsorted-threshold crossings. This fit is equivalent to a cosine-tuning function with a baseline rate, a modulation depth, and a preferred direction. Baseline rate is the mean firing rate across all targets. Modulation depth is the height of the cosine tuning function, approximately the difference between the most-active target and the least-active target. Preferred direction is the movement direction which, according to the tuning function, would elicit maximum firing from the unit. The above plots show the relationship between each parameter of this model when it is fit with the unsorted threshold-crossing data, on the Y-axis, versus the sum of the sorted units recorded on the same channels, on the X-axis. Each channel gives a single point on the plot; the channels from Figure 7 are indicated with letters. A linear fit is detailed on each plot. Unsorted control shows higher baseline rates, higher modulation depths, and nearly the same preferred directions.

We would also like to make a quantitative comparison of the tuning characteristics of channels and neurons between unsorted and sorted sessions. In the unsorted session, we fit a cosine function to the 16-target mean threshold-crossing rates. In the sorted session, we fit a cosine function in the same manner, except that we used the combined activity of the sorted neurons in place of the threshold-crossing event. Monkey A had 60 channels with sorted units where such a comparison could be made. Monkey C had 62. Monkey A had 10 channels with two sorted units and monkey C had 20; in these cases the activity of the two units were simply merged together for the purpose of fitting a cosine. In this small dataset, the mean difference in preferred directions for two units on the same channel was 80 degrees. The model for the cosine fit is:

$$n_{i,j} = \beta_{i,0} + \beta_{i,x}x_j + \beta_{i,y}y_j$$

$n_{i,j}$  is the mean firing rate for unit  $i$ , target  $j$ ;  $\beta_{i,0}$  is the baseline firing rate for neuron  $i$ ;  $\langle x_j, y_j \rangle$  is the vector to target  $j$ ; the angle of  $\langle \beta_{i,x}, \beta_{i,y} \rangle$  is the preferred direction of unit  $i$  and the length of  $\langle \beta_{i,x}, \beta_{i,y} \rangle$  is the modulation depth of unit  $i$ . The baseline rates, modulation depths and preferred directions are compared between sorted and unsorted conditions in Figure 8. In both animals the baseline rates and modulation depths were higher in the unsorted condition, and the preferred directions were similar between unsorted threshold crossings and the combined sorted units recorded later on the same channels.

## 2.5 DISCUSSION

We have demonstrated that good neural control can be achieved without conventional spike sorting or careful setting of thresholds. There is intrinsic variability in the quality of control from one session to the next, so it is not possible to make an exact quantitative comparison between sessions in these data. An experimental paradigm that better controls for the quality of adaptation data and the motivation level of the animal is clearly an avenue for future research. This initial finding has demonstrated that unsorted signals can be substituted for conventional ones without a dramatic, obvious drop in the quality of control.

We chose an 8-knot spline to model the tuning function because we expected mixed neuron signals to create complicated, sometimes multimodal tuning functions. We were surprised to find that on virtually all channels, the tuning function of the unsorted threshold-crossings was roughly uni-modal, in spite of the fact that where two units were recorded on the same channel their preferred directions did not tend to be similar. Since the tuning functions of the unsorted signals are not multi-modal, one may reasonably ask why population vector algorithm does not work well with these signals. We did attempt to use population vector algorithm on these signals, but found the quality of control so poor that we could not collect enough data to report. We speculate that the challenging aspect of unsorted signals is not multi-modality but the background noise that is introduced. Statistical extraction algorithms like the particle filter have the advantage of being able to recognize when a channel is an outlier in the present timestep, and to effectively reduce its contribution to the inference of intent. The particle filter examines the space of hypothetical velocities and

asks the question: what is the probability of observing these firing rates at various points in the hypothesis space of velocities? If a channel is momentarily inundated with background noise it will point to a region of the hypothesis space that is inconsistent with the rest of the channels, and it will have little weight toward the decoding of intent. The population vector algorithm gives each signal equal weight in the computation of intent, even if that signal is highly inconsistent with the majority of the population. We have demonstrated that with a good choice of extraction algorithm, a simple global threshold specification can be used in place of the spike-sorting schemes of conventional brain computer interfaces. These results have an immediate relevance for designs of self-contained spike processing circuits in the next generation of neural prosthetic devices. Without the need to set parameters of signal processing, it is possible to make an effective neural prosthetic system without operator intervention.

Perhaps more importantly, the kind of extraction algorithm demonstrated in this paper is arguably better suited to the indistinct patterns of multiunit activity that are typical of long-term chronic multielectrode recordings. These signals are composed of summated mixtures of signals from a number of sources rather than action potentials of individual neurons. They are subject to high levels of baseline noise and their tuning functions are harder to predict. We have shown that a well-chosen extraction algorithm can contend with these issues and provide a good control signal. While we used the same signal processing equipment that has been employed for years in neural prosthetics, we used it in a way that simulated a much simpler system. Our results show that a probe with fixed thresholds and one-way telemetry could be used for effective prosthetic control. It is easier to imagine a

turn-key clinical system for neural prosthetics that is based on the threshold-crossing counter used here. The elimination of elaborate signal processing regimes is an important step towards putting neural prosthetics into the real world.

### 3.0 RECORDING FROM THE SAME NEURONS CHRONICALLY IN MOTOR CORTEX<sup>2</sup>

In this chapter, we take advantage of the shared variance that is the defining characteristic of systems driven by latent variables. This shared variance manifests itself in cross-correlograms, and in the process of studying these cross-correlograms it became clear that they would be an excellent metric for identifying the same neuron in many different recording sessions. This solves a longstanding problem in chronic neural recordings: tracking the same neurons over long periods. Previous work on this topic has been based on subjective metrics, or quantitative metrics that are unreliable. In a population of simultaneously recorded neurons, there is an enormous amount of information about neuron identity in the cross-correlograms due to the many pairs of neurons that can be examined. This level of information is superior even to waveform shape as an identifier of the same neuron in separate recording sessions. By combining it with several other simple metrics of identity we are able to produce a definitive classifier.

---

<sup>2</sup> The following chapter has been submitted to the Journal of Neurophysiology as **Fraser and Schwartz**, Recording from the same neurons chronically in motor cortex. This chapter reflects changes made in response to reviewer comments.

### 3.1 ABSTRACT

Two rhesus monkeys were implanted with silicon arrays of 96 microelectrodes. Neural activity was recorded periodically over a period of weeks to months. We have developed a method to determine whether single units in two separate recording sessions represent the same neuron. Pairwise cross-correlograms, the autocorrelogram, waveform shape, and mean firing rate were used together as identifying features of a neuron. When two units recorded on separate days were compared using these features, their similarity score tended to be either high, indicating two recordings from the same neuron, or low, indicating different neurons. While these metrics are individually weak, together they produce a strong classifier. Some neurons were recorded for >100 days. These monkeys performed a center-out reaching task, and we found that the firing properties of chronically recorded neurons were stable over time.

## 3.2 INTRODUCTION

Neurobiologists who do chronic extracellular recordings frequently observe similar activity recorded on the same electrode from day to day. Occasionally a single neuron will have some unusual characteristic—a distinctive waveform, or some unusual and obvious firing property—that makes it clear that this same neuron is present in multiple sessions. The possibility that some neurons may be represented multiple times in a series of recording sessions creates a problem and an opportunity. Separately recorded neurons may not actually represent independent sources of data, so statistical tests which assume each unit is an independent sample may not be valid. But if the same neuron could be identified as such across multiple sessions, it would be possible to combine data and thereby estimate that neuron's firing properties with greater confidence. A sufficiently accurate metric of identity would allow all the recordings from a long series of sessions to be considered as a single population of neurons, with each identified unit contributing to the population for some portion of time.

A number of authors have attempted to identify the same neurons across recording sessions in a systematic way. The identification problem amounts to deciding, for each comparison between a sorted unit in one session and a sorted unit in another session, whether they represent the same neuron. Some authors have taken a qualitative approach, looking at waveform and sometimes inter-spike interval distribution information to identify

examples with very stable characteristics (Chestek et al. 2007; Ganguly and Carmena 2009; Greenberg and Wilson 2004; Jackson and Fetz 2007; Schmidt et al. 1976; Williams et al. 1999). A few have developed classifiers that identify stable neurons systematically (Dickey et al. 2009; Tolia et al. 2007), but these methods are subject to severe trade-offs between false negatives and false positives when the classifier is unreliable.

We have developed a new metric of unit identity using pairwise cross-correlograms between neurons in a simultaneously recorded population. It provides unit identification information comparable to that based on wave shape. Combining this metric with wave shape, autocorrelation shape, and mean firing rate, we are able to clearly identify whether two separately recorded units represent the same or different underlying neurons. We followed the identities of neurons across multiple sessions, in some cases for over 100 days.

The ability to track a large number of neurons across sessions allows us to address a fundamental question: how much do the tuning characteristics of neurons vary from day to day? There is a divergence of opinion in the literature as to whether the tuning characteristics of neurons are more or less fixed (Chestek et al. 2007; Ganguly and Carmena 2009; Greenberg and Wilson 2004), or whether they evolve continuously as part of a dynamic network that is only stable at the ensemble level (Carmena et al. 2005; Li et al. 2001; Rokni et al. 2007). We use our classifier to follow the same neurons over periods of weeks to months, and find that the tuning of neurons to the direction of movement is stable over time.

### 3.3 METHODS

#### 3.3.1 Chronic microelectrode implant

Two male rhesus macaques were implanted with 96-channel microelectrode arrays (Blackrock Microsystems; Maynard et al. 1997). Monkey C was implanted in February of 2009 with a single array on the convexity of the motor cortex next to the central sulcus, with the lateral edge of the array ~2mm medial to the genu of the arcuate sulcus. The recordings reported here were done in March—April of 2010, and consist of 6 sessions, recorded once a week on a day when the monkey did center-out movement tasks. Monkey F was implanted in April of 2009 with two arrays. One array was implanted in the same location as monkey C's array, targeting the primary motor cortex arm area. The other array was implanted further anterior and lateral, directly adjacent to the genu of the arcuate sulcus. This array was intended to target ventral premotor cortex. The recordings reported here were done in May 2009-March 2010. They consist of 40 sessions spread irregularly over that period. All animal procedures were approved by the institutional care and use committee of the University of Pittsburgh.

All activity was sorted off-line using OfflineSorter (Plexon). OfflineSorter allows a variety of features to be used to sort; we used principal component distributions, peak/valley voltage, and voltage at specific timepoints. We used different features depending on the particular arrangement of waveforms on a given channel/day, and we only sorted units that

were sufficiently distinct from noise and from each other. We identified 32—106 neurons per session from the combined activity of both arrays in monkey F, and 14—22 neurons per session from the single array of monkey C.

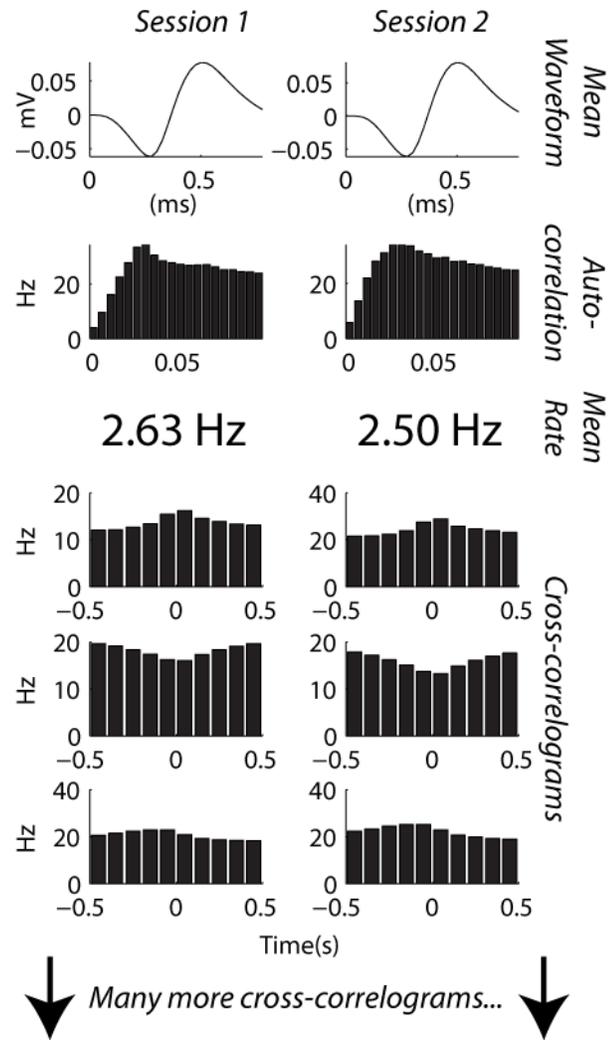
### **3.3.2 Behavioral task**

Prior to implantation, each monkey was trained to do a center-out reaching task in a 3D virtual environment. They viewed a stereoscopic monitor (Dimension Technologies) which displayed a target sphere in its center and a cursor sphere which tracked the movement of an infrared marker (Northern Digital) taped to the back of the monkey's hand. To receive a water reward, the monkey had to complete a center-out movement. First, it had to move the cursor sphere to contact a central target for a required period randomly selected in each trial from 400-600 ms (monkey F) or 200-600 ms (monkey C). The target sphere would then be moved to a peripheral location selected randomly from a queue of 26 locations spread evenly in a sphere with radius 66 mm (monkey F) or 83 mm (monkey C). The monkey would then have to contact the peripheral target for 400-600 ms (monkey F) or 200-300 ms (monkey C). A failed trial resulted in the target being requeued. Monkey F also performed out-center trials, where the order of targets was reversed.

### **3.3.3 Tracking the same neurons**

An implementation of this algorithm has been posted to MATLAB Central as "Tracking neurons over multiple days", id # 30113.

Let us consider the problem of determining whether a particular sorted unit in session 1 represents the same neuron as another sorted unit in session 2, one or more days later. In this data, we need only consider cases where the two units in question were recorded on the same electrode. This is because the inter-electrode spacing on the Utah array is large (400  $\mu\text{m}$ ) so it is unlikely that one neuron will be recognized on two different channels. If the two units do represent the same neuron, there will be several indicators in the data that we can quantify. We expect that the mean waveform shape, the autocorrelation function, the mean firing rate, and the cross-correlograms with other neurons will be similar. An example of these parameters for the same neuron in two recording sessions is shown in Figure 9. We quantify the similarity of the wave shape in the same manner as Jackson and Fetz (2007), as the peak value of the cross-correlogram between the average waveform shape in session 1 versus session 2. This allows for changes in the overall size of the waveform and slight shifts in the time domain, which are common. The resulting coefficient is Fisher transformed (the arc tangent of the hypotenuse function) to make it more normally distributed.



**Figure 9.** The same neuron has been detected in two recording sessions 3 days apart. There are various indicators that this has happened. The cross correlograms are between the target neuron and various other neurons that are present in both sessions. In order to line them up as we did above, we need to already know which neurons survived from session 1 to session 2. This problem is solved by an iterative procedure as described in Methods/Tracking.

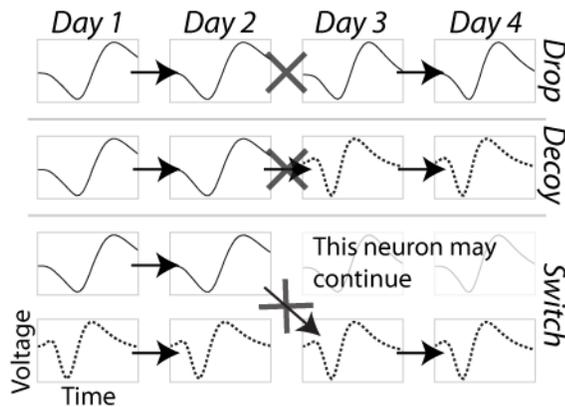
We estimate the autocorrelation function from 0 to 100 ms by binning at 5 ms resolution, exactly as is shown in Figure 9. That gives us a 20-point vector for each session. To quantify the similarity of these vectors, we take the Pearson correlation coefficient between them. Again we Fisher transform to make the distribution more normal. The similarity of the mean firing rates is computed simply as the difference between the log of the mean rates. We take the log because mean firing rates follow an approximately log-normal distribution.

Because there are many neurons simultaneously recorded, there are many pairwise cross-correlograms. Those shown in Figure 9 were chosen because they illustrate the strongest features for that neuron. The correlograms are computed for a range of  $\pm 0.5$  s at 100 ms resolution. We found this range captured the largest and most consistent features of the cross-correlograms, which tended to be positive and negative triangular bumps with lags near zero. The time-resolution represents a trade-off between capturing finer features of the correlogram and computation time. These macroscopic features reflect common inputs rather than synaptic connections between neurons. In order to summarize all the cross-correlograms as a single metric, we first take Pearson correlations between presumed identical cross-correlograms in the same manner as we do for the autocorrelation functions. That means we are comparing one of the cross-correlograms on the left in Figure 9 to the one immediately to its right, resulting in a single number for each pair. We Fisher transform those numbers and take the mean, resulting in a combined pairwise cross-correlogram similarity score.

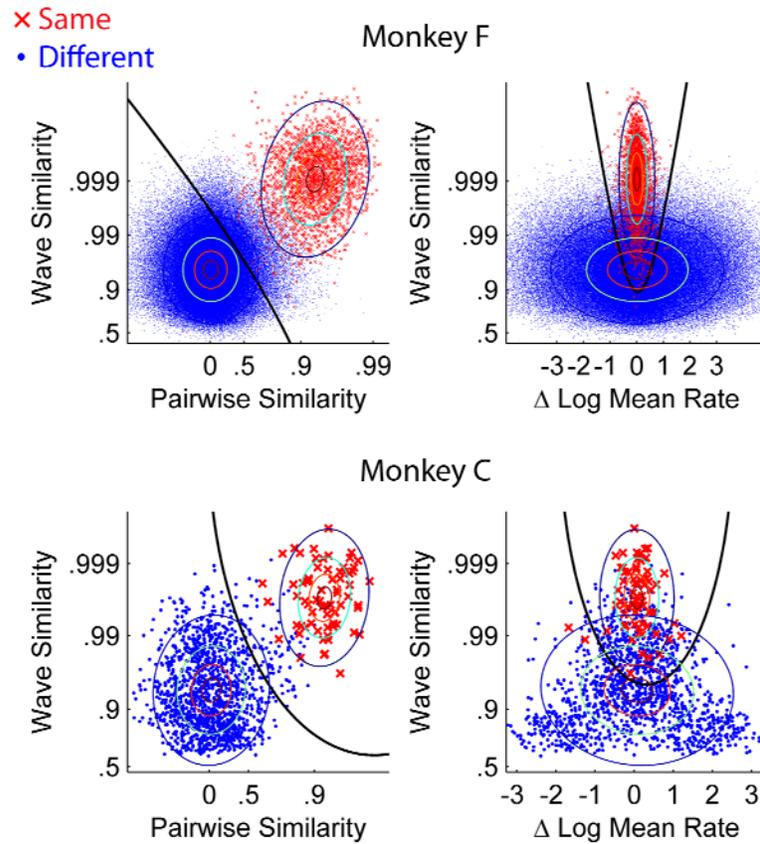
These four scores are combined with a quadratic classifier that computes an optimal decision boundary under the assumption that the underlying data can be modeled as a

mixture of multivariate Gaussians. Ordinarily a training data set is used to fit these Gaussians. In our case, we have a great deal of known different-neuron data (comparisons across different channels, which can't be the same neuron). However we do not have any known same-neuron data. Therefore we used partially supervised expectation-maximization to fit a mixture of Gaussians model (Come et al. 2009; Lanquillon 2000). Our dataset includes many points with known labels (cross-channel comparisons) while the remainder has mixed labels (within-channel, subsequent-day comparisons that might be the same neuron). Because the different-neuron dataset is so large, it essentially dictates the shape of one of the Gaussians, and the other Gaussian converges very quickly onto a second cluster of points that lie away from the different-neuron distribution, putatively corresponding to the same-neuron comparisons. The decision boundary of the classifier is calibrated to produce a 5% error rate in the known different-neuron distribution. This may seem high but it is mitigated by the types of potential errors that occur in real recordings. Figure 10 illustrates the types of possible errors, which depend on how many neurons are present on a channel and how they emerge and disappear over time. Drop errors occur when the same neuron on two days is classified as two different neurons. Switch errors occur when the classifier switches the labels between two neurons but at least one of them continues between days. This error happens rarely because the classifier will always label the session 2 neurons according to which session 1 neuron they fit best, so to produce a switch it essentially has to produce two errors simultaneously. Decoy errors occur when one neuron disappears at the same time another appears, and they are classified as the same neuron. The 5% error rate that we use as a target applies only to instances where it is possible to make a decoy error, which are inherently

unlikely. The location of the decision boundary and the contours of the two Gaussians are shown in Figure 11, and the parameters of the Gaussians are summarized in Table 2.



**Figure 10.** Types of errors we can make while trying to label the same neurons over multiple sessions. Drop errors occur when the target neuron continues but the classifier fails to positively identify it across one of the gaps (indicated by the X). The original label is then a false negative until the target neuron actually disappears. Decoy errors occur when the target neuron disappears at the same time as a new neuron appears, and the classifier mistakenly labels the new neuron as being the same as the old one. The label is then a false positive until the new neuron disappears. Switch errors occur when a distractor neuron is present simultaneously with the target, and the classifier mistakenly switches the label to the distractor. Switch errors are the least likely because they essentially require the classifier to simultaneously make a drop and a decoy error.



**Figure 11.** Similarity score distributions for the real data. Each point represents a comparison between two units on two different days. We computed four similarity scores as described in Methods/Tracking. These plots show projections of two scores at a time. Points are labeled according to whether they were classified as the same neuron. The same neuron / different neuron Gaussians estimated from the data are shown as contour plots. A two-dimensional slice of the decision boundary that is used by the classifier is shown as a black line. The contours correspond to 25%, 50%, 75% and 95% of the distribution. There are 6 unique combinations that could be shown; we chose the pairwise/wave scatterplot because they are the two most informative features, and we chose the mean/wave scatterplot because it illustrates the unique characteristics of the change-in-mean-rate feature.

When we use cross-correlograms to assess the identity of many neurons across two sessions, changes in labeling across days need to be considered. For instance, two neurons sorted on channel 1 might be labeled “unit 1a” and “unit 1b.” On the next day, their labels may be exchanged by the investigator doing spike sorting, or unit 1a may have disappeared and unit 1b is now labeled 1a. If we then wish to assess whether some other unit, for example unit 2a, is the same in session 1 and session 2, there is a problem with the cross-correlogram similarity metric. The cross-correlogram between unit 2a and unit 1a in session 1 will be different than in session 2, even if unit 2a is actually still the same neuron. We solved this problem with an iterative procedure, making an initial assumption that wherever the unit labels are the same between session 1 and session 2, they represent the same neuron. We then used our 4-score classifier to identify which units putatively corresponded to the same neurons from session to session. This set of identities was then used to relabel all the units and classification was performed again, under the assumption that the number of labeling errors will be reduced with each iteration. If this assumption is correct, the labeling will tend to converge, which it does after a few iterations.

### **3.3.4 Synthetic data**

Since we could not test our algorithm with data where the identity of neurons across days is known, we constructed a synthetic dataset by using the actual data to capture the variability in these metrics across separate recording sessions. There are four similarity scores: pairwise cross-correlation, wave shape, autocorrelation similarity, and mean firing rate. We

computed a synthetic true-positive dataset for one score at a time, by ignoring that score and using only the other three to classify the entire data set. The points classified as same-neuron can then be used as a synthetic same-neuron distribution for the score we left out. To avoid introducing a lot of errors, we defined a conservative 3-score boundary that would drop 25% of the points classified as positives by the full 4-score classifier. To understand how this works, let us consider a simplified example where we only have the two scores shown in the upper-left panel of Figure 11. We are going to use just the pairwise similarity (x-axis) to classify a set of points as positive. That means we will draw a vertical line and classify everything to the right of it as positive. We can then use these points as a true-positive dataset for the wave similarity score.

By using this technique four times we create four different pools of synthetic true-positive scores. We then recombine a random value from each pool to create artificial data points. With these synthetic true-positive points and the known-negative distribution (neurons on separate channels), we have a complete data set where the ground truth is known. This technique for creating synthetic data creates a specific kind of bias due to the fact that there is some correlation between the different scores, as can be seen in the slight tilt of the red cloud in Figure 11. When we use 3 scores to generate a known same-neuron distribution for the 4<sup>th</sup>, we throw away 25% of the positive category that were worst with respect to the 3 scores. Even though we didn't consider the 4<sup>th</sup> score in deciding what to throw away, because of the presence of correlation we end up with a slightly non-representative set of points with respect to the 4<sup>th</sup> score. On average, this technique biases the distribution of 4<sup>th</sup> metric 0.1 standard deviations upward.

### 3.3.5 Long-term accuracy

In order to extrapolate the various error rates to performance in a long series of recordings, we need to know how many neurons disappear from our electrodes each day, how likely we are to record  $n$  neurons on the same channel, and how often the decoy error scenario occurs (Figure 10). These parameters were estimated using the labels produced by the full 4-score classifier. It is not necessary that these labels be exactly right, only that they have the same general characteristics as an actual set of recordings. We modeled the turnover rate (the proportion of the population replaced daily) as an exponential decay that became smaller as a neuron was recorded for a longer duration, which corresponded to tendency in the data for a core group of stable neurons to persist from day to day, coexisting with another group of more marginal neurons which turned over frequently. The estimates for monkeys F and C are turnover rate (for the average gap between sessions): 15%/35%, additional neurons per channel: 0.4/0.4, percent of neurons ending in a decoy-prone situation: 11%/10%. We used these parameters along with the drop, decoy and switch rates from synthetic data to extrapolate the performance of the classifier over time. In the Results section we use the terms “false negative” and “false positive” in the context of long-term data, defining the false positive rate as the proportion of *labels* that exist at a given time which are on the wrong neuron. The false negative rate was defined as the number of labels no longer in existence even though their target neuron still is, divided by the number of labels currently in existence.

### 3.3.6 Preferred directions

We estimated the preferred direction of each neuron on each day it was recorded by fitting a linear model:

$$\lambda = \beta_0 + \beta_x x + \beta_y y + \beta_z z$$

where  $\lambda$  is the firing rate of the cell,  $x$ ,  $y$ , and  $z$  are the target direction and the  $\beta$  terms are the parameters of the model. The preferred direction (PD) of the cell is the vector  $\langle \beta_x, \beta_y, \beta_z \rangle$ . We generated a measurement error distribution for this cell by bootstrap resampling the residuals of the fit on a per-trial basis. This method will incorporate the variability in firing rates that is caused by variability in kinematics. In this model, nonlinearity in the tuning function is considered part of the noise term, so the model will tend to slightly overestimate the amount of measurement error. We will need this measurement error cloud when we make two observations of the same neuron's PD, so that we can associate a level of uncertainty with our estimate of the angle between observations. Since these PDs are three-dimensional, the effect of measurement error is somewhat complicated. If two PDs are 90 degrees apart the measurement error is as likely to make the angle smaller as bigger. If they are more than 90 degrees apart, measurement error will tend to bring the observations closer together. We solve this problem by calculating a "pure measurement error" distribution for each comparison, rotating the measurement error cloud of PD 2 so that its mean matches the mean of the PD 1 cloud and taking repeated samples from each distribution to compute the angle between them. In Results, we will test two null hypotheses about the evolution of PDs over time: that they are unchanging, or that they change in a random walk. To test the no-

change hypothesis, for each case where a neuron was observed twice, we computed a quantile for the observed change in PD indicating where it lies in the appropriate measurement error distribution. If the true PD is unchanging, these quantiles should be uniformly distributed. To test the random-walk hypothesis, we assumed that the real change in PD could be modeled as a step each day in a random direction with a Gaussian-distributed step size. The step size was estimated using comparisons between adjacent sessions (only available in monkey F), taking into consideration the fact that observed PD change equals real change plus measurement error. We then generated a random-walk distribution numerically and added the appropriate measurement error distribution to it. Again, we compared the observed change in PD to the numerically generated distribution and computed a quantile. The uniformity of these quantiles was assessed with a K-S test.

## **3.4 RESULTS**

### **3.4.1 Classification accuracy**

Figure 11 shows the converged similarity score distributions, the shape of the Gaussians that are fit to them, and the decision boundary of the classifier. The fact that the distribution of points is clearly divided into a high-similarity cluster and a low-similarity cluster indicates that this approach is generally working. The accuracy of our algorithm is assessed in several ways, each of which is subject to different kinds of bias. The simplest approach is to generate a dataset where the ground truth is known by splitting each recording session in

half and comparing the two halves as though they were separate sessions. This results in a high-similarity cluster from comparisons between the same neuron in the first and second half of the data, and a low-similarity cluster from comparisons between different neurons. Compared with multi-day data, the high-similarity cluster is likely to be more tightly distributed and further from the low-similarity cluster, because the similarity metrics we are using are likely to change less between the first and second half of a single session than across the interval between sessions. For this dataset we changed the initial conditions of the iterative identification procedure by randomizing the unit labels, so that the classifier was not initialized with the correct answer. Testing our classification algorithm using split session data gave 0 errors in monkey C and a .005% overall error rate in monkey F.

Without knowing the ground truth, there are some ways to estimate the error rates in the real data.

1. Estimate the decoy error rate using comparisons across separate electrodes, which can't be the same neuron.
2. Estimate the drop rate by modeling the data as a mixture of Gaussians.
3. Estimate the drop rate with synthetic data.

We use method 1 to set the classification boundary with a target decoy error rate of 5%. Where we set the classification boundary amounts to a trade-off between drop errors and switch or decoy errors. A 5% target for the decoy rate heavily favors the drop rate, which ends up  $\ll 1\%$ . The decoy error rate in a real dataset will be the product of 5% and the rate at which the decoy error scenario occurs (see Figure 10), which is rare.

For method 2, we used the Gaussian models shown in Figure 11. For every same-channel comparison classified as a negative, we estimated the probability that it was actually an unusually inconsistent single neuron using the density of the same-neuron and different-neuron Gaussians. By taking the mean of these probabilities we estimated the overall drop rate (Table 3).

For method 3, we constructed a synthetic data set as described in Methods, applied our classifier, and calculated the drop rate (Table 3). The accuracy results are broadly similar to method 2. Pairwise cross-correlograms are the most important metric, followed by waveform, autocorrelation, then mean rate. We then extrapolated the synthetic data error rates to generate the long-term false negative and false positive scores shown in Figure 12 (Methods/Long-term accuracy). For comparison, we implemented a similar algorithm that uses the same waveshape score as ours and a similar autocorrelation score (Dickey et al. 2009). The long-term false positive rate of the Dickey et al. algorithm was different than they reported because of differences in the way we tested the classifiers. Dickey et al. assessed long-term false positives by constructing a synthetic data set where each day a neuron from a different channel was used. In order to get a false positive after  $n$  days, the classifier would need to make  $n$  errors in a row. We assessed long-term error rates by estimating the various error rates for single comparisons (Table 3), then estimating how often various error scenarios would occur (Figure 10). Using this approach we find error rates tend to increase over longer periods of recording. Figure 12 shows performance for both a conservative threshold, which minimizes false positives, and an aggressive threshold, which minimizes false negatives. The aggressive threshold is similar to the one used in Dickey et al., targeting a ~25% false positive

rate. The conservative threshold targets 5%. We used the conservative threshold for all classification in the remainder of this paper.

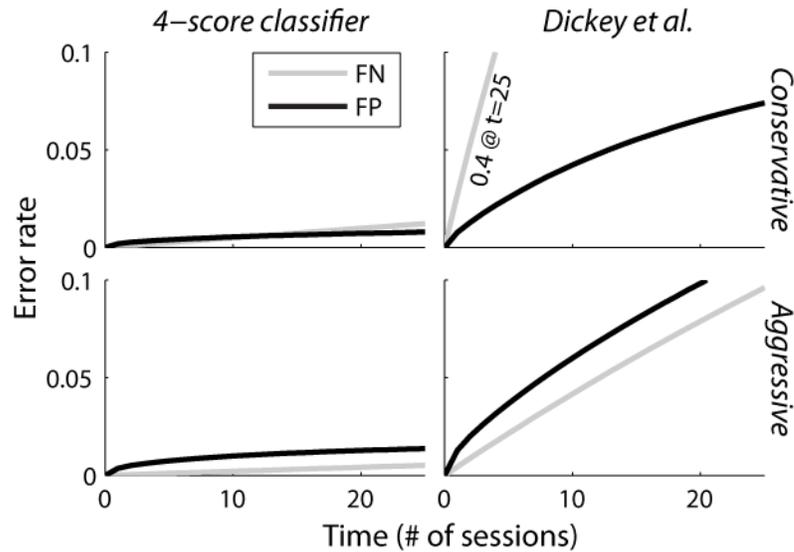
Our classification algorithm identified 760 unique neurons in monkey F and 35 in monkey C. The lengths of observation for these neurons are shown in Figure 13. Most neurons were recorded for less than 30 days but some in monkey F were recorded for over 100.

<b>Monkey F</b>					
<b>Same-neuron (red) cluster</b>					
	<i>Pair</i>	<i>Wave</i>	<i>Auto</i>	<i>Mean</i>	<i>Mean</i>
	1.70	3.85	2.72	0.00	
<i>Pairwise X-Corr</i>	0.19	0.06	0.19	0.00	
<i>Waveform</i>		0.52	0.05	0.00	<i>Cov</i>
<i>Autocorrelation</i>			0.72	0.00	
<i>Mean Rate</i>				0.08	
<b>Different-neuron (blue) cluster</b>					
	<i>Pair</i>	<i>Wave</i>	<i>Auto</i>	<i>Mean</i>	<i>Mean</i>
	0.01	1.90	1.06	0.02	
<i>Pairwise X-Corr</i>	0.15	0.00	0.01	0.00	
<i>Waveform</i>		0.36	0.05	0.00	<i>Cov</i>
<i>Autocorrelation</i>			0.67	0.00	
<i>Mean Rate</i>				2.74	
<b>Monkey C</b>					
<b>Same-neuron (red) cluster</b>					
	<i>Pair</i>	<i>Wave</i>	<i>Auto</i>	<i>Mean</i>	<i>Mean</i>
	1.61	3.26	2.50	0.07	
<i>Pairwise X-Corr</i>	0.10	0.01	0.11	-0.02	
<i>Waveform</i>		0.25	0.03	-0.01	<i>Cov</i>
<i>Autocorrelation</i>			0.72	-0.02	
<i>Mean Rate</i>				0.19	
<b>Different-neuron (blue) cluster</b>					
	<i>Pair</i>	<i>Wave</i>	<i>Auto</i>	<i>Mean</i>	<i>Mean</i>
	0.03	1.78	1.26	0.08	
<i>Pairwise X-Corr</i>	0.17	0.01	0.01	-0.01	
<i>Waveform</i>		0.40	0.08	-0.04	<i>Cov</i>
<i>Autocorrelation</i>			0.32	-0.01	
<i>Mean Rate</i>				1.57	

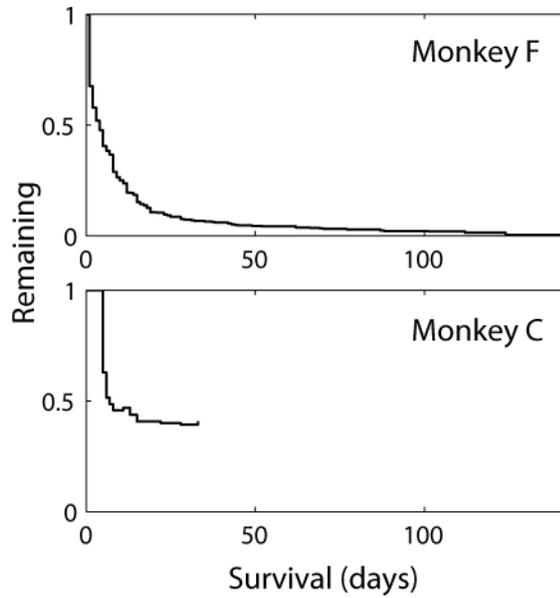
**Table 2.** Means and covariances of the Gaussians fit to the same-neuron and different-neuron distributions, shown as contour plots in Figure 11.

<b>Gaussian Model</b>						
	<i>Pair</i>	<i>Wave</i>	<i>Auto</i>	<i>Mean</i>	<i>Not</i>	
<i>Pairwise X-Corr</i>	0.04	0.01	0.04	0.02	0.12	<b>Monkey F</b>
<i>Waveform</i>		0.22	0.13	0.16	0.02	
<i>Autocorrelation</i>			0.40	0.37	<.01	
<i>Mean Rate</i>				0.59	<.01	
<i>All</i>	<b>&lt;.01</b>					
<i>Pairwise X-Corr</i>	<.01	<.01	0.02	<.01	0.16	<b>Monkey C</b>
<i>Waveform</i>		0.31	0.13	0.24	<.01	
<i>Autocorrelation</i>			0.35	0.35	<.01	
<i>Mean Rate</i>				0.53	<.01	
<i>All</i>	<b>&lt;.01</b>					
<b>Synthetic Data</b>						
	<i>Pair</i>	<i>Wave</i>	<i>Auto</i>	<i>Mean</i>	<i>Not</i>	
<i>Pairwise X-Corr</i>	0.01	<.01	0.01	<.01	0.02	<b>Monkey F</b>
<i>Waveform</i>		0.09	0.03	0.04	0.00	
<i>Autocorrelation</i>			0.23	0.20	<.01	
<i>Mean Rate</i>				0.68	<.01	
<i>All</i>	<b>&lt;.01</b>					
<i>Pairwise X-Corr</i>	<.01	<.01	<.01	0.01	0.08	<b>Monkey C</b>
<i>Waveform</i>		0.23	0.07	0.15	<.01	
<i>Autocorrelation</i>			0.31	0.31	<.01	
<i>Mean Rate</i>				0.73	<.01	
<i>All</i>	<b>&lt;.01</b>					

**Table 3.** Drop rate tables. Each entry indicates the drop rate for a quadratic classifier based on one or more scores. We assessed drop rate by modeling the data as a mixture of Gaussians (method 2 in the text, top two tables) or using synthetic data (method 3 in the text, bottom two tables). Performance is shown for each metric of identity, each combination of two, and for the full classifier based on all four combined. Single-metric performance is on the diagonal. Two-metric performance is indicated by the combination of row and column labels. The “Not” column indicates the performance of a classifier with the 3 scores other than the row label. The decoy rate was always 5% (method 1 in the text) due to the way we set the classification boundary.



**Figure 12.** Long-term accuracy in a synthetic dataset for our method and a similar method with 2 classifiers (Dickey et al. 2009). Here we define accuracy in terms of whether after  $x$  days a label is still correct. False negatives (FN) are cases where the label is gone but the target neuron is still around. False positives (FP) are cases where the label is on the wrong neuron, whether or not the original target is still present. In the top row, the classification boundary was set conservatively, targeting a 5% decoy rate. In the bottom row the classification boundary was set aggressively as described in Dickey et al. (2009). The figures on the right have a significantly higher false positive rate because they experience a non-negligible number of switch errors, which accumulate rapidly. False positive errors tend to be more damaging in studies that examine the properties of neurons over time, but a high false negative rate will be inefficient because many neurons will not be tracked as long as they could be.

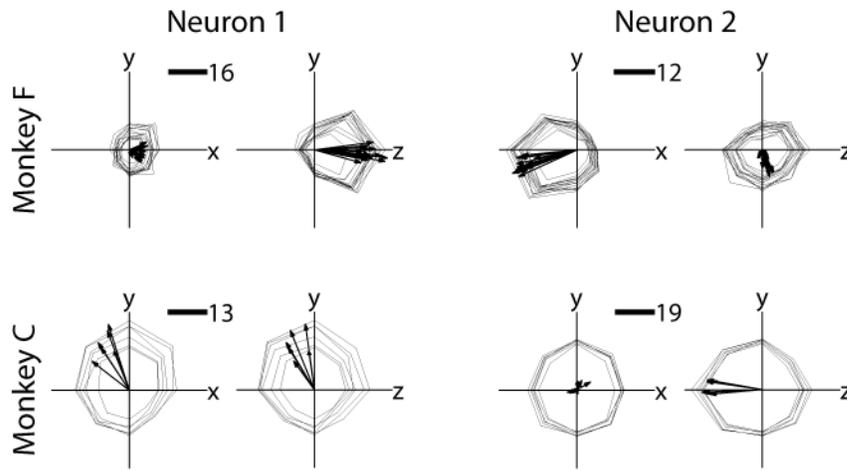


**Figure 13.** Observation lengths for recorded neurons. X-axis indicates recording duration for a single neuron. Y-axis indicates the number of neurons that were recorded that long, divided by the number of neurons that could have been recorded that long. We identified 760 unique neurons in monkey F out of 2892 sorted units recorded over 40 sessions. We identified 35 unique neurons in monkey C out of 104 sorted units recorded over 6 sessions. Monkey C had a smaller but more stable population.

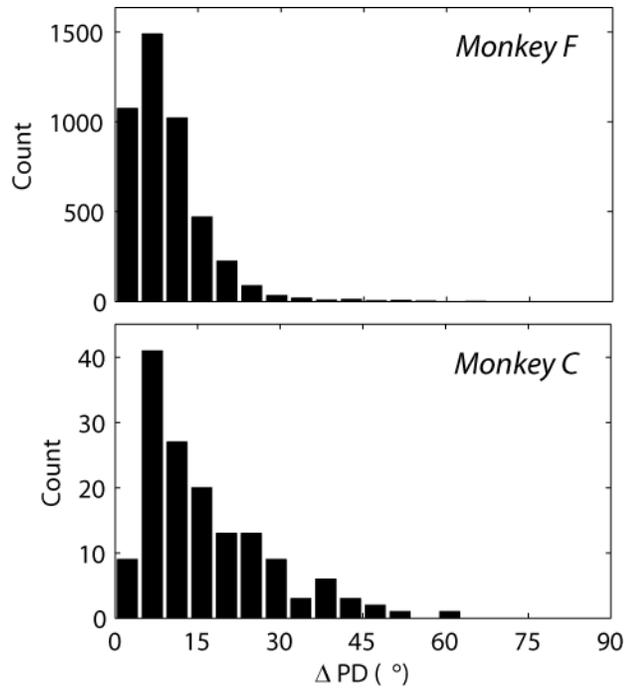
### 3.4.2 Tuning Parameters

With the same neurons identified over a long-term dataset, we can evaluate the stability of these cells' directional tuning over time. Examples of tuning profiles for neurons that were tracked for a particularly long time are shown in Figure 14. These examples show low variability in their tuning function between sessions. For each day that each neuron was

recorded, we fitted a cosine tuning model describing a linear relationship between the direction of movement and the firing rate of one cell (see Methods/Preferred directions). Figure 15 shows histograms for the PD variability between two or more sessions, for all recorded cells. It was necessary to exclude units with weak or inconsistent modulation because changes in their PDs reflect more measurement error than real change. We assessed measurement error by bootstrapping the residuals of the cosine tuning fit (Methods/Preferred directions) and excluded all comparisons where the uncertainty in our estimate of PD change was greater than  $10^\circ$ . This excluded 49% of the population in monkey F, and 17% of the population in monkey C. The excluded set is based on the measurement error, not change in PD, so we are not limiting the potential for real variation in the PD across sessions; a neuron with a strong preferred direction in one session could have an equally strong but altered preferred direction in the next session.



**Figure 14.** Directional tuning profile across multiple sessions, two neurons from each monkey. The mean firing rate for each target is displayed in the direction of that target. There are 26 targets in 3D space; here we see x-y and z-y slices. Firing rate profiles from each single neuron are rendered simultaneously for all sessions where that cell was recorded. Arrows indicate preferred directions (PDs) from model fit. Scale bars indicate number of spikes per reach. The neurons shown are the first two neurons from each monkey that were recorded for at least 14 days with a mean preferred direction measurement error of  $< 5^\circ$ . Low measurement error does not necessarily limit the amount of variation in PD across sessions.



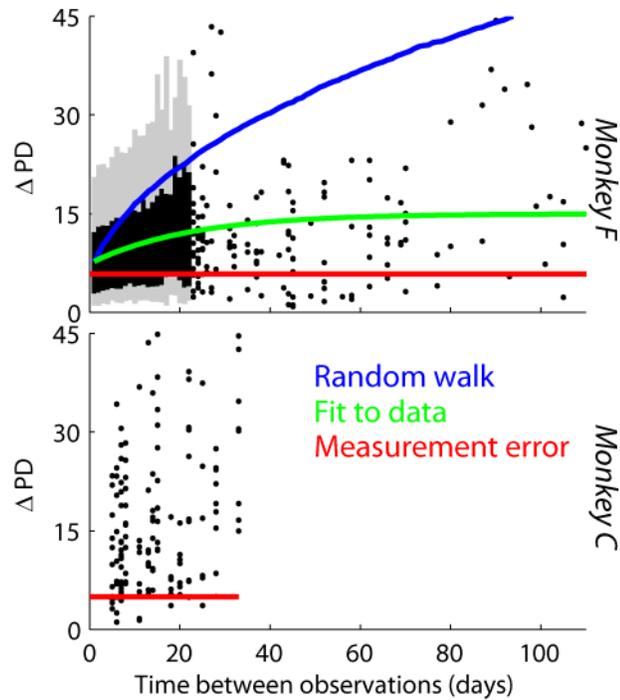
**Figure 15.** Observed differences in PD of single neurons across sessions. PD is the vector indicating the target which would theoretically elicit the maximal firing rate from the neuron according to a cosine-tuning model fitted from the data. X-axis indicates the absolute difference in PD. Y-axis indicates histogram bin counts. The histogram includes the angle between every possible pair of two observations of the same neuron's PD. Comparisons are only included where the uncertainty in our estimate is  $< 10^\circ$  according to the bootstrap distributions described in Methods/Preferred directions. This does not constrain the potential variation in PD across sessions. Note that in three dimensions, two PD vectors chosen at random are much more likely to have a angle difference of around  $90^\circ$  than around  $0^\circ$ .

Figure 15 shows that PD variability is low, generally  $< 30^\circ$ . In assessing these changes in PD, we consider three hypotheses:

1. The PD is static, and all variation is due to measurement error.
2. The PD experiences slight real variation, which accumulates over time to create a random walk.
3. The PD experiences slight real variation which accumulates, but it is tethered to an underlying intrinsic value which does not change.

Hypothesis 1: the PD is static. If the true change in PD is always 0, then the observed changes should follow the distribution of measurement error that we computed by bootstrapping the residuals of the cosine tuning fit. We can compare each observed change in PD to the measurement error distribution described in Methods and compute a quantile. If the observed changes were due only to measurement error, these quantiles would be uniformly distributed. Using a K-S test, we rejected this hypothesis ( $p < .01$ ).

Hypotheses 2: the PD experiences real variation that accumulates over time. We extrapolated a series of distributions of PD changes from the one-day changes by assuming that PD change represented a random walk plus measurement error, as described in Methods. This distribution was tested against the data with a K-S test in the same way we assessed hypothesis 1, and again it is a bad fit ( $p < .01$ ). This rejects hypothesis 2.



**Figure 16.** The relationship between observation interval (x axis) and difference in PD (y axis). This is the same data as Figure 15, scattered out over time. Each point indicates that two observations were made of the PD of the same neuron  $x$  days apart, and the angle between the two observations was  $y$  degrees. For monkey F, we summarized dense regions of the scatter plot with a gray line indicating the 95% of distribution and a black line indicating 66%. Red line shows the expected average difference in PD if all change is caused by measurement error. Blue line indicates a hypothetical trend assuming that the changes in PD accumulate over time (see Methods/Preferred directions for details). Green line shows a simple nonlinear function fit to the data,  $y = b_1 - b_2 \cdot \exp(b_3 \cdot x)$ . While there is a limited amount of accumulation, it is significantly below the distribution associated with the blue line, indicating there is an intrinsic unchanging preferred direction for each cell.

Figure 16 shows a scatterplot of the relationship between observation interval and PD difference. The expected-value lines associated with hypotheses 1 and 2 are shown and are above the mean of the data, visually confirming the results of the K-S test. This leaves us to conclude that while there is real variation in PDs, they are tethered to underlying intrinsic PDs.

### 3.5 DISCUSSION

This report takes a series of extracellularly recorded populations and attempts to identify in every case whether an earlier session / later session pair represents the same neuron. Most past work on this topic has attempted to identify a minority of stable cells that can be reliably tracked, leaving the rest of the population in the category of “uncertain”. The four features we used are individually weak classifiers, but because they represent independent sources of information they can be combined into a strong classifier. For example, mean firing rates can indicate that two units are definitely *not* the same neuron, but they can never give high confidence that they *are* the same. The mean firing rate of a single neuron tends to be consistent from day to day, but the expected difference in mean rate between two different neurons is also zero (Figure 11). Thus if two units on two days have a very different mean rate, then they are almost certainly different neurons; but if they have a similar mean rate, we cannot be certain that they are the same neuron. By itself, mean firing rate would be an inaccurate way identify neurons, but when combined with other metrics it contributes useful information. With four different metrics of similarity, we can produce a

very strong classifier that can follow neuron identity throughout entire population, not just the largest units with the most consistent characteristics.

Estimating identity for the entire dataset not only allows us to take full advantage of the data we have collected, it will allow us to treat the entire chronic series as a single dataset for the purpose of network analysis. The analysis of multi-observation data, where different subsets of a network are observed in overlapping intervals of time, has been a topic of growing interest (Lambiotte et al. 2009; Mucha et al. 2010). Setting aside these future directions, the most obvious immediate application of a unit identification algorithm is to determine whether the firing properties of neurons change over time. Other authors have speculated on the possible role of tuning changes, especially preferred direction changes, in the underlying motor control algorithms of the brain (Carmena et al. 2005; Rokni et al. 2007). It has been observed before that changes in PD across two adjacent sessions tend to be small (Chestek et al. 2007). Our results confirm these small changes and demonstrate that over a long series, they do not accumulate into large changes. Instead, the PDs of these neurons are tethered to an unchanging intrinsic value.

Unobserved kinematic parameters may account for some or all of the variability in PDs that we see. Since there are no buttons or manipulandum that involve the hand in our task, and the reaches are performed and tracked in three-dimensional space, the main unobserved kinematic parameters are subtle changes in wrist posture and the way the monkey sits each day. Chestek et al. (2007) showed that the variability in PD within a single day was at least partially attributable to subtle changes in kinematics. It is likely that such subtle changes may account for some of the PD variability in our data.

The arrays used in these experiments are physically able to record the same neurons for long periods. Even though the shape of waveforms will change from day to day (especially in magnitude), we have shown that it is possible to identify the same neurons reliably. Applying this technique to other types of arrays that are less physically stable might produce different results. Also, the performance of Utah arrays in this respect is not completely consistent. It has been our experience that the kind of long-term stability we identified in these data usually emerges after an array has been implanted for several months. One reason for recording instability is physical motion of the array during accelerations of the monkey's head (Santhanam et al. 2007). Over long periods of implantation Utah arrays accumulate scar tissue, especially at the surface of the cortex (Rousche and Normann 1998). This scar may serve to physically stabilize the array. Our monkey C had an older array (12-13 months versus 1-11 months) and more stable recordings. If tracking the same neurons over long periods is an important aspect of an experiment, it may be prudent to plan data collection for such experiments several months after electrode implantation.

The monkeys in this dataset performed straightforward arm movement tasks in the data we have analyzed. The sessions used for unit identification and center-out analysis represent only part of the experiments that were conducted over the time period they span. On other days the monkeys performed different tasks, but none were specifically designed to elicit changes in preferred direction. Clearly we would like to know whether an experimental paradigm designed to produce changes in preferred direction (Jarosiewicz et al. 2008; Li et al. 2001) might produce a long-term trend when applied repeatedly to the same neurons. That issue will have to await future experiments.

### **3.6 ACKNOWLEDGEMENTS**

Thanks to Steve Chase, Angus McMorland, Jeong-Woo Sohn, and Andrew Whitford for data collection and comments on the manuscript. Supported by NIH Grant 5R01NS047356.

## **4.0 COMPLEXITY OF TUNING FUNCTIONS IN A RICH MOVEMENT TASK**

In this chapter, we re-examine a simple reaching task that has long been studied in terms of coordinate frames. Using a finely spaced array of targets in 3-D space, we are able to characterize the tuning of neurons with higher accuracy than is possible with only a few targets or only two dimensions of movement. We find that individual neuron's tuning functions are not well described by simple models. We reduce the population to a low-dimensional set of latent variables and analyze the activations of those variables using information theory, and find that all the parameters of this task influence activity throughout the movement. Along with similar recent research, particularly the work of Churchland and Shenoy (2007), this indicates that coordinate frames may be an overly simplistic metaphor for the meaning of neural activity in motor areas.

### **4.1 INTRODUCTION**

The pioneering work of Georgopoulos and colleagues (1982) demonstrated that broad directional tuning describes the activity of many neurons in primary motor cortex during center-out reaching tasks on a two-dimensional plane. This finding has been extended to out-center movements (Georgopoulos 1985), movements in three dimensions (Schwartz et al.

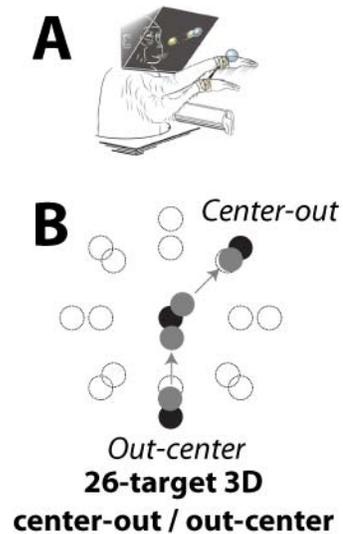
1988) and has been used to develop the population vector (Georgopoulos et al. 1988), an algorithm that extracts intended movement direction from the activity of many neurons with different preferred directions. It has also been shown that some variations of the original center-out reaching task produce discrepancies between neural activity and a strict interpretation of the directional tuning model (review: Scott 2008). In order to more clearly characterize directional tuning of neurons in primary motor and premotor cortex neurons during a free reaching task, we used a fine grid of 26 targets in three-dimensional space and incorporated both center-out and out-center movements.

## **4.2 METHODS**

### **4.2.1 Behavioral task**

Two rhesus macaques (monkeys F and T) performed reaching tasks in a virtual environment that has been described previously (Schwartz et al. 2004). The monkey sat in front of a 45° angled mirror that reflected the image of a stereoscopic monitor (Dimension Technologies). The position of the monkey's hand was tracked with an infrared marker (Northern Digital) and rendered as a spherical cursor in a 3D environment. As diagrammed in Figure 17, they performed 3D center-out/out-center with 26 targets arranged approximately evenly on a 66 mm (monkey F) or 75 mm (monkey T) sphere. In a single successful center-out trial, a target would be presented at the origin and the monkey would make contact for a required hold period. The central target would then be removed and a random peripheral

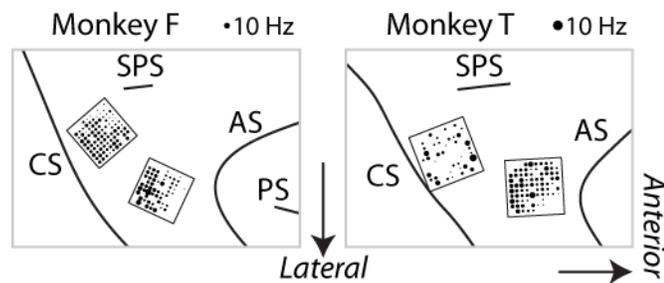
target would be presented, which the monkey would also have to contact and hold. The hold periods were in the range of 150-600 ms and were partially random to prevent the monkey from making guesses. In out-center trials the order of target presentation was reversed.



**Figure 17. A:** The monkey sits in front of a 45° angled mirror which reflects the image of a stereoscopic monitor. A 3D virtual environment is rendered on the monitor. The position of the monkey's hand is tracked with an infrared marker and represented in the virtual environment by a cursor. **B:** The 26-target center-out/out-center task at 15% scale. We use 26 peripheral target locations arranged approximately evenly on a sphere. In each trial a random location is selected. Half the trials are center-out trials where the monkey must contact a target sphere at the origin, then at the peripheral location; half the trials are out-center trials where the order is reversed.

## 4.2.2 Neural recording

These macaques were implanted with two 96-electrode arrays (Blackrock Microsystems) in accordance with the rules of the Institutional Animal Care and Use Committee of the University of Pittsburgh. The position of the arrays were chosen to target the arm representation of the primary motor cortex (M1) and ventral premotor cortex (PMv), also known as area F4 (Luppino and Rizzolatti 2000). The positions of the arrays were reconstructed from photos taken at the time of implantation and are shown in Figure 25. Spike waveforms were recorded using a digital signal processor (Tucker Davis Technologies), and well-isolated single units were sorted offline using OfflineSorter (Plexon).



**Figure 18.** Location of two arrays in monkeys F and T, reconstructed from surgical photos. Each Utah array has 96 channels which are represented by the black circles. The area of each black circle indicates the difference in average firing rate between the best and worst target in the center-out / out-center task for neurons on that channel. CS: central sulcus, SPS: superior precentral sulcus, AS: arcuate sulcus, PS: principal sulcus.

### 4.2.3 Sampling

Before any further analysis, the firing rate and kinematic data were smoothed, time-rescaled and sampled. The spike times were converted to a continuous function using a 400 ms wide cosine-shaped filter. Kinematics were smoothed with a 5 Hz lowpass filter. We then defined a region of interest within each trial, starting when the monkey's hand was at the center and the peripheral target was presented, and ending when the monkey made contact with the peripheral target. We rescaled time in each trial so that 10 evenly spaced points would cover this period, with an additional 5 points at the same spacing before and after the period of interest. The average period was 0.495/0.631 seconds for monkeys F/T.

### 4.2.4 Tuning parameters

For Figure 20, we computed for each neuron the average firing rate versus time for each of the 52 types of movement (26 targets  $\times$  2 directions) and identified the time and target direction which elicited maximal firing. For Figure 21, we computed preferred direction (PD) using linear regression with firing rate as a function of target direction (not hand kinematics):  $\lambda = b_0 + b_x t_x + b_y t_y + b_z t_z$ . Modulation depth (MD) is the length of the vector  $\langle b_x, b_y, b_z \rangle$ . Target direction,  $\langle t_x, t_y, t_z \rangle$ , is defined as the difference between the target position and the starting position.

#### 4.2.5 Latent variable analysis

We used factor analysis, a standard dimensionality-reduction technique, to reduce each population of simultaneously recorded neurons into a single set of latent variables. Factor analysis is very similar to the more common principal component analysis, except that it allows different observed variables to have different variances. In neural data, factor analysis is less likely to be dominated by the most active neurons in the population. Similar to PCA, factor analysis may generate a set of latent variables that do not each have an easily describable identity. Instead each latent variable produced by factor analysis may be a mixture of features.

The number of putative latent variables must be specified before performing factor analysis. Every day, we performed factor analysis with 1 to 15 factors, and assessed the quality of these models by reconstructing the firing rates of the neurons from the factor model in the manner of Yu et al. (2009). Their method cross-validates in two ways. We used 4/5 of the data to derive a factor model, and then used that model to reconstruct the remaining 1/5. Within that 1/5, we would compute the activation of the factors using all but the neuron we were reconstructing, so that a neuron would not contribute to its own reconstruction. The  $R^2$  for reconstruction across all neurons varied from 0.21-0.47 from day to day with a median of 0.4. For each day we chose the model that produced the best reconstruction (6-15 latent variables).

#### 4.2.6 Information theoretic analysis

We used information theoretic analysis to quantify the relationship between latent variable activation, direction of movement, and target position. This type of analysis is useful because it can capture nonlinear relationships between variables. Information is usually quantified in categorical variables, however the concept is extended to continuous variables by differential entropy. We quantified differential entropy by assuming that each latent variable was independent and Gaussian distributed, which gives a closed-form solution. This assumption is a better approximation for latent variables than it is for individual neurons. We are interested in three quantities:

1.  $I(A;D|T)$ : the mutual information between activation of the latent variables and direction of movement, conditioned on the target position.
2.  $I(A;T|D)$ : the mutual information between activation and target position, conditioned on direction.
3.  $I(A;D;T)$ : the co-information (Bell 2003) between the three variables

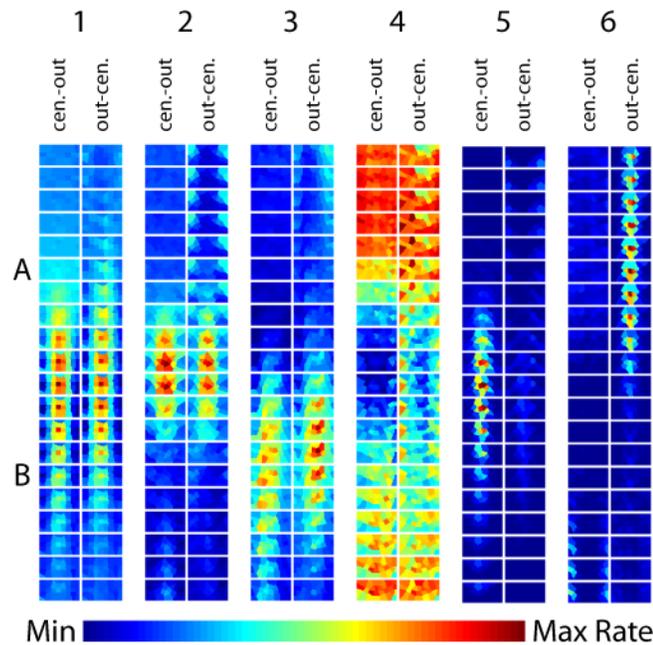
The first two are straightforward;  $I(A;D|T)$  indicates how much information D gives us about A, given that we already know T. This value is computed using out-center data, where the target is always the same.  $I(A;T|D)$  indicates how much information T gives us about A, given that we already know D. We computed this value by comparing activity between each center-out movement direction and the co-linear out-center movement, which has the same direction but a different target position. Co-information is less intuitive. It indicates how much redundancy or synergy is present between the three variables. A negative value of co-

information indicates that there is redundancy, meaning that some of what we learn about A from D we can also learn from T. A positive value of co-information indicates there is something about A we can only learn by knowing both A and T. An example of this phenomenon is XOR function. Only by knowing both inputs can we know the output; individually they give no information.

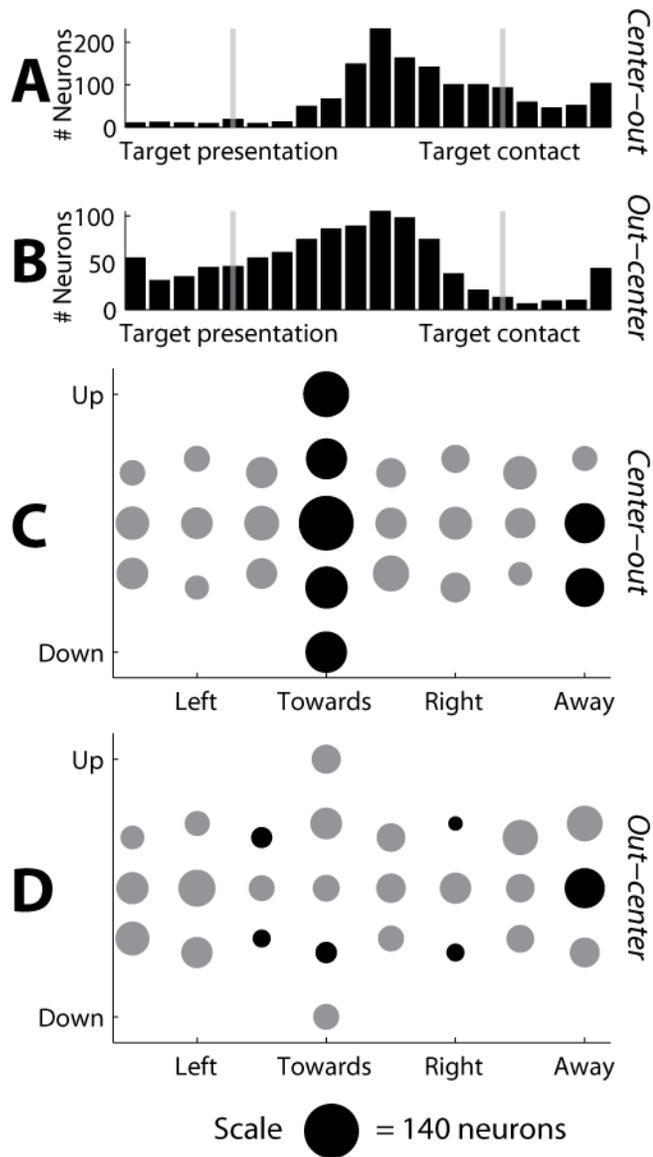
In order to compute  $I(A;D|T)$ ,  $I(A;T|D)$  and  $I(A;D;T)$  we need a way to measure the entropy of A conditioned on knowing T, D, or both (the conditional entropy). T and D are categorical variables, so we simply subtracted the within-category means of A to produce a conditional distribution (with n-fold cross-validation). This procedure is analogous to an ANOVA, and makes the same assumptions. The difference is that we interpret the within-category reduction of variance in terms of information sharing.

### **4.3 RESULTS**

We recorded 2,095 single units over 25 sessions on two arrays in monkey F, and 339 single units over 5 sessions in monkey T. Many of these units represent multiple observations of the same neuron in separate sessions, but with the exception of Figure 19 they are treated separately. The locations of the arrays are illustrated in Figure 18, as well as the tendency for neurons in particular regions of the array to modulate during center-out movements.



**Figure 19.** Tuning of individual neurons in the 3D center-out/out-center task. 6 neurons from monkey F are presented. For each neuron we are looking at a 40 snapshots of the tuning of the cell at a single moment in time, in the manner of a filmstrip. Each snapshot corresponds to one of the tiny images, which is a picture of the firing rate as a function of the azimuth (horizontal axis of the image) and elevation (vertical axis of the image) of the direction of movement. The direction eliciting the highest firing rate variance is always at the center of the image. A and B indicate the time of target presentation and target contact. The average time between them is 0.495 s. Neuron 1 shows classical velocity tuning. Neuron 2 is tuned to direction of movement but is active only early in the movement. Neuron 3 is active only late in the movement. Neuron 4 decreases activity during movement in any direction. Neuron 5 is directionally tuned, but only in the center-out trials. Neuron 6 is directionally tuned before the onset of movement, exclusively in the out-center trials. In the complete population, seemingly every conceivable combination of temporal and directional tuning can be observed. The minimum and maximum rates for the neurons shown are [0.85,27], [1.0,12], [2.1,11], [1.0,4.3], [0.01,6.6], [0,2.1]. We used the neuron-tracking algorithm described in Chapter 3.0 to combine data across multiple sessions. Neurons 1 and 5 are from the PMv array; the rest are from the M1 array.

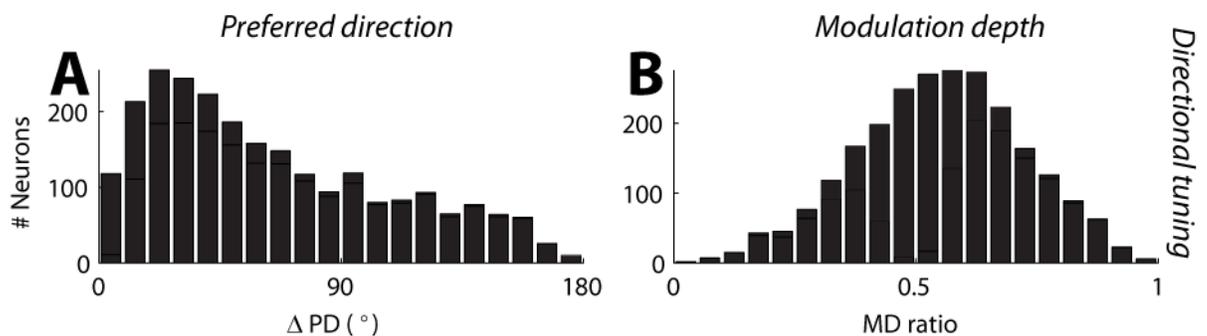


**Figure 20.** We computed the time and target direction that elicited maximum firing for each recorded neuron. A & B show simple histograms with preferred time on the x-axis. C & D show two dimensional histograms with azimuth on the x-axis and elevation on the y-axis. There are 26 unique directions in center-out and out-center; each is represented by a spot. The area of the spot is proportional to the number of neurons that fired maximally for that target. The spots are colored black if this number of neurons is statistically greater or fewer than expected at  $\alpha=0.05$  (Bonferroni corrected). There is a strong tendency for neurons to be tuned for movements directly towards or away from the body in the center-out task (C). Maximal rates tend to occur later

in the center-out task (A) and earlier in the out-center task (B). The directional bias is present in both the M1 and PMv populations when they are analyzed separately.

Having 26 targets with both outward and inward movement in this task generates a rich dataset for evaluating the tuning of individual neurons. Figure 19 shows a few examples of the many types of tuning we observed. Most neurons exhibit directional tuning at some point in the task consistent with cosine tuning (Georgopoulos et al. 1982). Most of these also had preferences for specific phases of movement or for outward versus inward movement. These preferences do not divide into obvious categories. Neurons were found which fire before, during, and after the movement. Figure 20 shows histograms of the number of neurons that fired maximally for a particular time or target direction. There were individual cells that might be considered pre-movement, early, late, hold-period, etc., but the population does not divide into clearly identifiable groups. There were two exceptions to that rule: the vast majority of neurons that fired maximally late in the movement were center-out neurons, while early-maximum neurons are nearly always out-center neurons. Also, there was a clear bias in the distribution of PDs for directions in the sagittal plane. This trend does not correlate with any obvious biomechanical parameters. For instance, the stiffness of the arm is greatest for movements directly towards or away from the shoulder (Gomi and Kawato 1996; Mussa-Ivaldi et al. 1985). This would explain the overrepresentation of preferred directions pointing towards or away from the body, but not the other 6 directions in the sagittal plane. Furthermore, the tendency for stiffness to be greatest for movements towards the shoulder is stronger as the hand gets further away from the body (Mussa-Ivaldi et al. 1985). But there is no tendency for PDs to be clustered around the distal starting positions in the out-center task. Another plausible explanation of the biased PD distribution is that our

array was simply in the elbow representation of the motor cortex. The elbow tends to be more involved in saggital plane movements in three-dimensional free reaching tasks, where the arm is oriented with the elbow pointing down. However the biased distribution of preferred directions is equally strong on both arrays, which are far apart compared to the map of the body on the motor cortex. Gentilucci et al. (1988) reported a tendency for PMv neurons to be tuned to movements up, down, and towards the body, which seemed to relate to sensory receptive fields in the same locations. Movements in the saggital plane are common in reaching and self-feeding behaviors; it is possible that the biased distribution of preferred directions is a reflection of the importance of these behaviors.



**Figure 21.** Changes in PD and MD between center-out and out-center.  $\Delta PD$  is the angle between the PD in center-out and the PD in out-center. The MD ratio is defined as  $MD_{OUT}/(MD_{OUT}+MD_{IN})$ . 0 indicates only out-center modulation; 1 indicates only center-out modulation. There is a modest but statistically significant tendency for MDs to be higher for center-out than out-center.

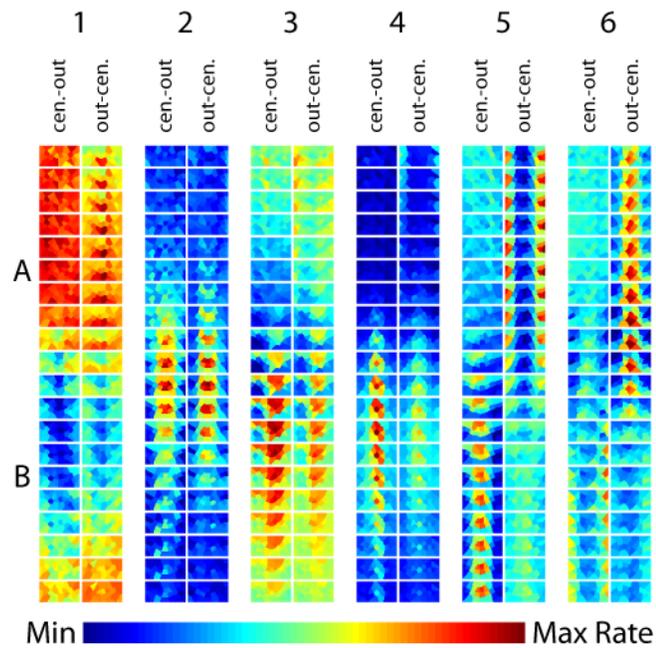
Most neurons have different PDs and MDs in center-out versus out-center (Figure 21), but there is a lack of modes in the distribution. There is a tendency to have similar PDs for

outward and inward movements, but counterexamples abound. MDs also showed wide variation between inward and outward movements, with a modest overall tendency to be higher for outward.

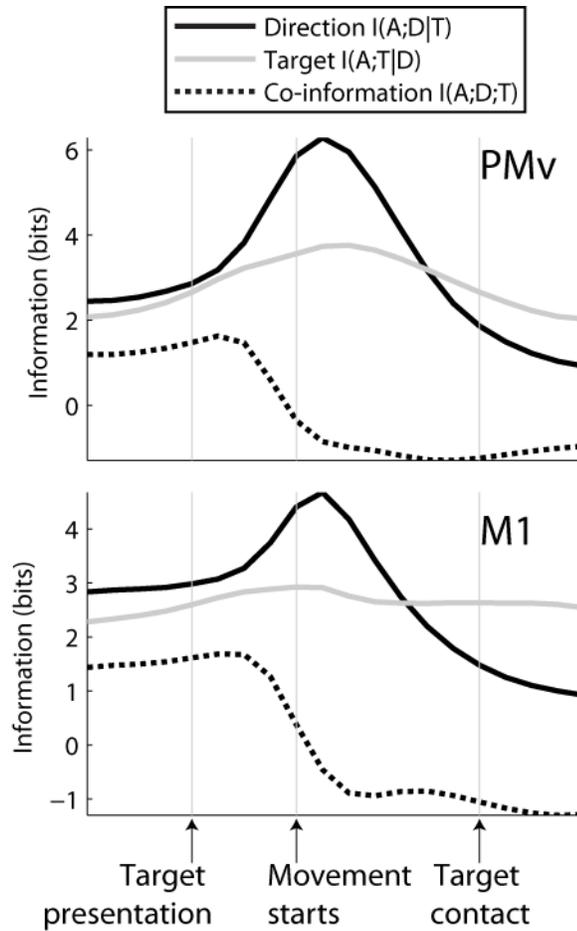
In order to understand what information is contained in this population, we reduced the simultaneous firing rates to a smaller set of latent variables and applied information theoretic analysis. Latent variables are more amenable to this type of analysis because they are estimated using sums of many neurons, and therefore are more normally distributed than individual neurons. We treated the latent variable activation as a multivariate Gaussian and calculated three information metrics as described in Methods. Figure 23 shows how early in the task, latent variable activations contained somewhat more information about the direction of the upcoming movement, while late in the task there was more information about the target position. Neither direction nor target can be said to be the dominant signal. The co-information was initially positive, but dropped quickly to negative values around the time the monkey started to move its hand. This implies that during the hold period, population activity encoded specific *combinations* of direction and target of the upcoming movement. After movement begins, the population apparently switches to a simpler encoding scheme where direction and endpoint are encoded simultaneously but without the earlier synergistic interaction between variables.

To test whether the complex tuning functions of these neurons might be accounted for by combinations of kinematic parameters, we fit a model that predicted instantaneous firing rate using position, velocity, acceleration, speed and curvature. Kinematic models achieved a median  $R^2$  of 0.18 as compared to 0.4 for the latent variable model. The kinematic

model has a similar order (11 kinematic parameters versus 6-15 latent variables) but only describes part of the shared variance that is captured by the latent variable model.



**Figure 22.** Tuning of latent variables. Format is the same as Figure 19. We observe phasic variables (1), directionally tuned variables (2-4), and position tuned variables (5-6). We also see many latent variables that are mixtures of features not easily described. A weighted sum of latent variables can produce a wide variety of features.



**Figure 23.** Conditional mutual information between the latent variable model's low-dimensional representation of population activity (A) and movement direction (D) versus target position (T). A high value of  $I(A;D|T)$  means that population activity differs between movements that have different directions but the same target (out-center movements). A high value of  $I(A;T|D)$  means that population activity differs between movements with the same direction but different targets (center-out and out-center movements that start in different locations but go in the same direction). Co-information indicates how the three variables interact. A negative value of co-information indicates that the information about activity we get from direction is partially redundant with the information we get from target position. A positive value of co-information indicates that direction and target together give us information about activity that is impossible to learn from them separately. Information values were calculated separately for 20 timepoints, so that we can see how the information content of population activity evolves over the course of the movement.

## 4.4 DISCUSSION

The 26-target center-out/out-center task gave us an extensive set of conditions in a pure reaching task with no hand movement relating to buttons or a manipulandum. We were confronted by a set of idiosyncratic neurons whose tuning functions changed, seemingly in response to every variable in the task, and which were not explained by any simple unifying principle. A neuron that might have seemed simply directionally tuned when observed only in the center-out task is suddenly inexplicable when we add the out-center version and observe a 90° shift. The fact that many neurons in motor and premotor cortex have a complex directional and temporal relationships to movement has been demonstrated before, most systematically by Churchland and Shenoy (2007). The effectiveness of the population vector (Georgopoulos et al. 1988) at reconstructing movement is sometimes overly strictly interpreted to mean that individual neurons must be velocity-tuned. But the population vector only demonstrates that the directional parameter is embedded in population activity and can be extracted from an average of many neurons. It does not exclude the possibility that other parameters may also be present.

The tuning functions of individual neurons and latent variables seemed to be a combination of directional and phasic elements. We performed information-theoretic analysis to determine how much the basic parameters of the task—direction and target position—influenced population activity at different phases of the task. The low-dimensional representations in latent variable models are especially amenable to information theoretic analysis because they are less noisy, less correlated, and more normally distributed than the individual neurons they are computed from. We found that there was a similar level of

information about direction and target position, though direction was more dominant around the time of movement initiation while target was more dominant towards the end of the trial. The evolution of co-information over time implies that before the onset of movement, the population codes specific combinations of direction and target, but once the movement starts it switches to a more straightforward scheme. Co-information is a sometimes-counterintuitive quantity that can be hard to interpret. In the context of this task, the positive value of co-information before movement probably reflects the fact that there were many neurons that were directionally tuned during hold A (the initial hold period), but only during the out-center task where the monkey knew the upcoming direction (for example cell 6 in Figure 19).

The approach of thinking about the population as transmitting information is particularly illuminating when we consider the preferred-time histograms in Figure 20. There are virtually no cells that fire maximally during hold A in center-out, and very few that fire maximally during hold B in out-center. Center-out hold A and out-center hold B both occur at the origin of the workspace, but there is nothing particularly special about that point in space with respect to the monkey's body. But these events do share an important characteristic in the context of our task: they are low points of information content. The origin is the most common position of the cursor, so at a purely kinematic level this is the least surprising and therefore lowest-information position for a monkey that has spent a lot of time doing this task. Center-out hold A and out-center hold B are also both times when upcoming movement is unknown. The lack of information about the future means there is

less for the population to encode. Thus, it makes sense that there is a lack of neurons encoding anything at this point in time.

There are two primary differences between this study and past work on this topic. First, target directions were spaced finely in three-dimensional space, making it more likely that we would accurately characterize neurons with narrow tuning functions. Second, because we used fixed arrays and sorted all activity off-line, we included many neurons with low firing rates and a modest response to the task. Under these circumstances we identified a dramatically biased distribution of preferred directions. We also found that many neurons were tuned to a specific phase of the task, including phases where no movement was occurring. These results do not lend themselves to a simple interpretation in terms of coordinate frames. In an effort to summarize the modulation of the population, we applied information theoretic analysis and showed that neural firing rates were sensitive to both the direction of movement and the target position throughout all the phases of movement. These findings suggest that the firing rates of many neurons may not represent abstractly meaningful parameters.

## 5.0 LATENT VARIABLES REVEAL AN ERROR SIGNAL IN PREMOTOR AND MOTOR CORTEX<sup>3</sup>

In this chapter, we are able to study latent variables directly. We take advantage of an unusual task, in which a monkey makes circular arm movements for several revolutions. Because of the cyclic nature of this task, many plausible candidates for latent variables will produce a sinusoidal signal, whatever their exact nature or lag relationship to movement. We are able to use a carefully tuned canonical correlation-based procedure to specifically extract signals that are sinusoidal at the frequency of movement. Because of the selectivity of this approach, we identify a small set of latent variables that can be studied directly. This avoids the endemic problem of this type of analysis, in which the extracted signals are mixtures of unrelated features and therefore difficult to interpret.

---

<sup>3</sup> This chapter has been prepared for publication, to be submitted shortly as **Fraser and Schwartz**, Latent variables reveal an error signal in premotor and motor cortex.

## 5.1 ABSTRACT

Some time-varying signals may be represented by the brain in a weak but widely distributed manner, such that they are difficult to detect in individual neurons. In order to study such signals, we used dimensionality reduction techniques to extract latent variables embedded in populations of 50-100 simultaneously recorded neurons in the primary motor and premotor cortex of macaque monkeys doing a circular drawing task. We identified two signals well-known to exist in these neurons, the horizontal and vertical directions of movement. We also identified a third signal that anticipated positional error of the monkey's movements. This signal is a very robust and widely distributed feature of activity in these areas. It represents a neural correlate of the future-predicting components in theoretical models of movement control.

## 5.2 INTRODUCTION

To better understand how brain coordinates the movement of the body, we would like to characterize the signals embedded in neural activity during movement. In traditional electrophysiology, this is accomplished by recording neurons individually. Parameters such as muscle activation, joint kinematics and endpoint are chosen based on prior knowledge of brain function and compared to the firing rates of individual neurons. The population vector (Georgopoulos et al. 1988) is an alternative to this basic scheme in which the firing rates of many separately recorded neurons are combined using vector sums to produce a decoded

representation of an internal parameter, such as intended movement direction. This method summarizes the activity of many neurons, but still requires the investigator to specify the parameters encoded by the population.

With simultaneous recording of many neurons, it is possible to use dimensionality reduction techniques to extract directly from the data the *latent variables* that many neurons have in common. Latent variables are a mathematical concept, in which many observed variables, such as neural firing rates, can be partially explained by a smaller number of hidden variables containing shared variance that many of the observed variables have in common. An example of the use of latent variable analysis to study neural firing rates is Yu et al. (2009), where the investigators were able to extract a low-dimensional representation which exhibited characteristic patterns of activation during each trial. This type of analysis is less biased by prior beliefs, and it is capable of revealing signals that are inaccessible to single-neuron recording. It is possible for a set of neurons to encode signals in a way that is difficult to understand when they are viewed one at a time. If individual neurons encode many parameters simultaneously, and only a small minority of these parameters are well-characterized, it will be extremely difficult to relate individual activity patterns to behavioral events. This is especially true in studies of motor behavior. When studying sensory responses in an anesthetized animal, an experimenter can isolate a single parameter and manipulate it consistently for many repetitions to determine a neuron's response. But with awake behavior, the animal will tend to produce complex movements and behave differently from trial to trial.

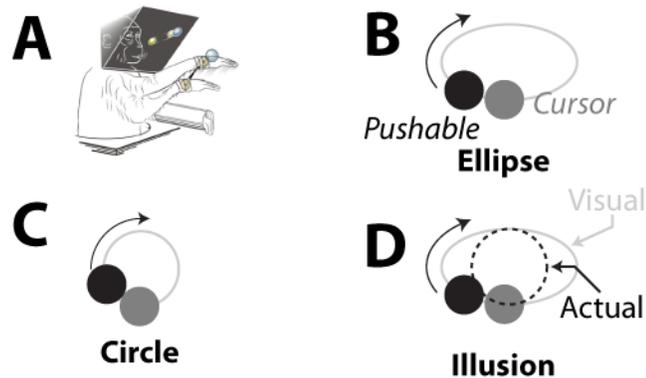
Traditional electrophysiology analysis often involves a search for neurons which significantly increase or decrease their firing rates in response to an event in the experiment. Alternatively, the firing rates of all recorded neurons may be summed so that an overall increase can be detected. These approaches may not reveal the full extent of the signals embedded in the population for two reasons. First, one set of neurons may increase firing rate while another set simultaneously decreases firing rate, as is the case with cosine tuning (Georgopoulos et al. 1982). The net result is that there is no change in the summed firing rate. Second, the firing rates of individual neurons may be generated by many factors simultaneously. Some of these factors may be distributed across the population, so that they cannot be easily identified in any one neuron.

With simultaneous recording of a small population of 50-100 neurons, we can address these issues using dimensionality reduction techniques to extract latent variables directly from the data. One of the latent variables we identified was a widely distributed signal that seems to be related to error correction. Modulation of firing rates during corrective movement has been identified in these brain areas before (Flament et al. 1992; Wise et al. 1998), but we were able to study the error signal more directly than has previously been possible.

## 5.3 METHODS

### 5.3.1 Behavioral task

Two rhesus macaques (monkeys F and T) performed reaching tasks in a virtual environment, described previously (Schwartz et al. 2004). The monkey sat in front of a 45° angled mirror that reflected the image of a stereoscopic monitor (Dimension Technologies). The position of the monkey's hand was tracked with an infrared marker (Northern Digital) and rendered as a spherical cursor in a 3D environment. As diagrammed in Figure 24, these monkeys performed a drawing task in which they pushed a virtual sphere along a circular or elliptical template for 6 cycles. In some trials, we introduced an illusion effect, in which we distorted the relationship between the horizontal position of the hand and the position of the cursor in the virtual environment so that the monkey was drawing a circle in the real world but seeing an ellipse in the virtual world. In these trials the first two cycles would be performed normally, and then the distortion would build gradually in cycles 3 and 4. In cycles 5 and 6 the distortion would be completely in effect, so a circle drawn in reality would appear as an ellipse on the screen.

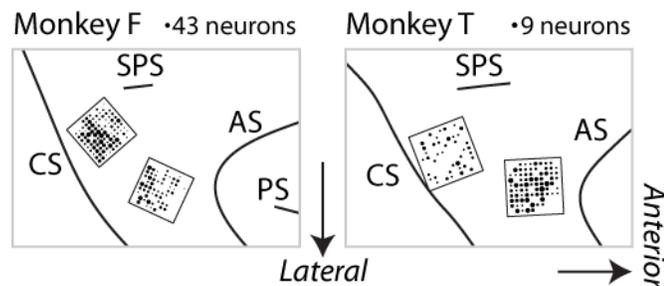


**Figure 24.** Task objects at 15% scale (monkey not to scale). **A:** The monkey sits in front of a 45° angled mirror which reflects the image of a stereoscopic monitor. A 3D virtual environment is rendered on the monitor. The position of the monkey’s hand is tracked with an infrared marker and represented in the virtual environment by a cursor. **B:** Ellipse drawing task. The monkey must push a sphere around an elliptical template six times. **C:** Circle drawing, also six cycles. **D:** Illusion drawing. The monkey starts out drawing an ellipse for two cycles. Then a distortion is introduced so that while it is drawing a circle in the real world, it is drawing an ellipse on the screen. The distortion is introduced gradually during cycles 3 and 4, so that by cycles 5 and 6 the monkey is drawing a circle but seeing an ellipse as illustrated in panel D.

### 5.3.2 Neural recording

These macaques were implanted with two 96-electrode arrays (Blackrock Microsystems) in accordance with the rules of the Institutional Animal Care and Use Committee of the University of Pittsburgh. The position of the arrays were chosen to target the arm representation of the primary motor cortex (M1) and ventral premotor cortex (PMv), also known as area F4 (Luppino and Rizzolatti 2000). The positions of the arrays were

reconstructed from photos taken at the time of implantation and are shown in Figure 25. Spike waveforms were recorded using digital signal processors (Plexon Inc; Tucker Davis Technologies), and well-isolated single units were sorted offline using OfflineSorter (Plexon).



**Figure 25.** Location of two arrays in monkeys F and T, reconstructed from surgical photos. Each Utah array has 96 channels which are represented by the black circles. The area of each black circle indicates the number of neurons that were recorded on that channel. An example circle is shown above each panel. CS: central sulcus, SPS: superior precentral sulcus, AS: arcuate sulcus, PS: principal sulcus.

### 5.3.3 Sampling

Before any further analysis, the firing rates of neurons and the position and velocity of the monkey's hand were smoothed, time-rescaled and sampled. The spike times were converted to a continuous function by convolving the point process (a series of Dirac delta functions at the spike times) with a 400 ms wide cosine-shaped filter. Position and velocity,

which were captured at 60 Hz, were smoothed with a 5 Hz lowpass filter. Drawing trials were divided into cycles, with the cycle edges defined by the cursor crossing the  $x=0$  plane. The first half of the first cycle and the second half of the last (6<sup>th</sup>) cycle were discarded because they were inconsistent across trials. The remaining five cycles were divided into 20 equally spaced timepoints. Because the speed of movement was somewhat variable, the 20 samples were closer together in some cycles than others. The mean cycle length was 1.21/1.00 s for monkeys F/T. 20 sample times were added to the beginning and end of the series of 100 (5 cycles  $\times$  20 samples) using the spacing from the first and last cycles.

#### **5.3.4 Illusion response scores**

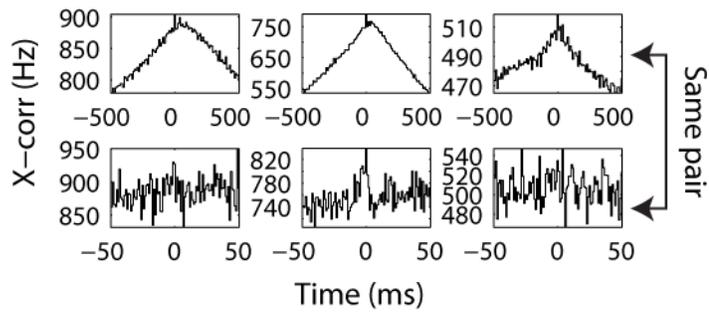
We quantified the response of each neuron to the illusion in a series of steps. First, we detrended the neuron's average firing rate by removing the mean and the average linear trend within a condition (circle, ellipse, or illusion). Second, we identified the cyclic component of the neuron's firing by fitting it with a sine function at the frequency of the circular movement of the hand. The points where the sine function crossed zero while increasing defined the edges of the firing-rate cycle. The five central cycles were identified for the ellipse and illusion conditions. Third, we quantified the modulation in each cycle by summing up the absolute value of firing rate within the cycle. Since the mean and trend were removed in step 1, the absolute value indicates modulation in either direction. Illusion modulation for each cycle was normalized by dividing by the corresponding non-illusion ellipse cycle's modulation, in order to remove any trends that reflect progress through the task rather than the illusion. Finally, we divided the amount of modulation in the last two

cycles by the first two, resulting in a measurement of the impact of the illusion on modulation.

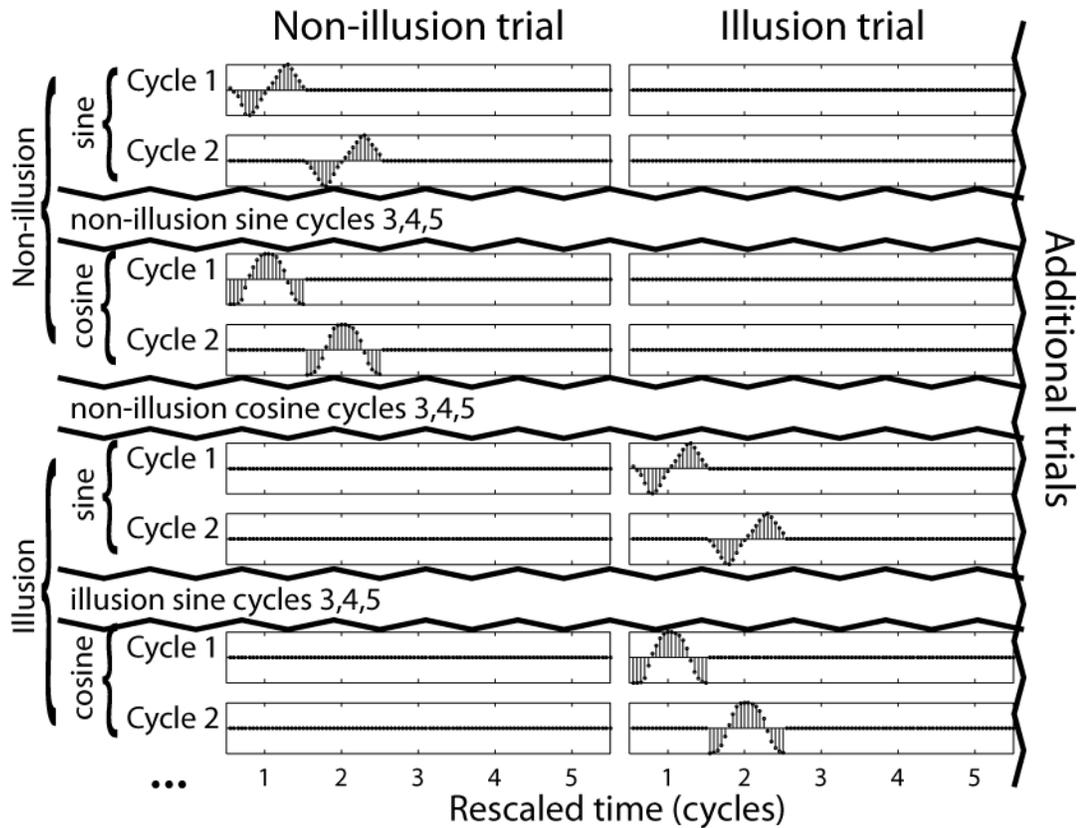
### **5.3.5 Latent variable analysis**

We would like to take the simultaneous activity of all the neurons recorded on one day and transform it into a set of latent variables. Broad-timescale correlation of the type shown in Figure 26 was ubiquitous in these data, indicating that there was some shared variance in the simultaneous activity of these neurons which can be exploited by dimensionality reduction. The drawing task had two unique features that allowed us to estimate latent variables with a specialized technique. First, the drawing movement was cyclic, so many of the interesting signals will appear sinusoidal. Second, there was a manipulation during the illusion trials that may alter firing rates. These two features were built into our estimation of latent variables using canonical correlation. Canonical correlation is a statistical technique that identifies signals that two sets of observed variables have in common (Hotelling 1936). In our case, one set of variables was the firing rates of neurons. The other set of variables was a processed version of the kinematics. Five cycles from each drawing trial were sampled at 100 points as described earlier. At each point, the sine and cosine of the movement direction (vertical and horizontal component) was calculated. These values were inserted into a matrix, illustrated in Figure 27, with each column representing one of the 100 timepoints  $\cdot$  [number of trials] elements. Each row contained the sine or cosine values for only one cycle from each trial, with zeros elsewhere. Each row was also selective for illusion or non-illusion, so it would stay at zero for the other type of trials. There were 10 (5 cycles  $\times$  2

components/cycle) rows for the non-illusion trials and 10 rows for the illusion trials. The entire matrix was then  $20 \times 100n$  where  $n$  is the number of trials. For any particular column, there would be only two non-zero entries corresponding to the sine and cosine of movement direction. This matrix and the matrix of neural firing rates at the same time points were analyzed with canonical correlation to generate a set of latent variables that the two inputs had in common. Canonical correlation outputs a weight matrix, which describes the relationships between latent variables and neural firing rates. This weight matrix was used to estimate the latent variable activations from the observed firing rates. We used 5-fold cross-validation to avoid creating artificial correlations, using 4/5 of the data to create the weight matrix, and then reconstructing the left-out 1/5. Our canonical correlation based approach differs from traditional dimensionality-reduction algorithms (for example Yu et al. 2009) in that the processed kinematics (Figure 27) impose assumptions about the nature of the latent variables: they must have a sinusoidal shape, which may vary from cycle to cycle and between normal and illusion trials.



**Figure 26.** Cross-correlograms (unnormalized) showing broad-timescale correlation, which is ubiquitous in this data. Top row, long-timescale cross correlograms. Bottom row, short-timescale versions of the same cross-correlograms. The strong features at long timescales are sometimes accompanied by short timescale features.



**Figure 27.** Illustration of the processed kinematics that are used as one of the inputs to canonical correlation. Rows correspond to variables, columns to observations. Each row contains the sine or cosine of movement direction for a specific cycle of drawing, for either illusion or non-illusion trials. Jagged lines indicate that some rows are not shown. When canonical correlation is performed using this matrix and the matrix of simultaneous neural firing rates, it will identify signals that both have in common. The common signals can be any linear combination of the rows shown above.

### 5.3.6 Computing the kinematic error signal in drawing

One of the latent variables consistently identified in the drawing task looked like an error signal. In the context of this task we formulated error as the disparity between the

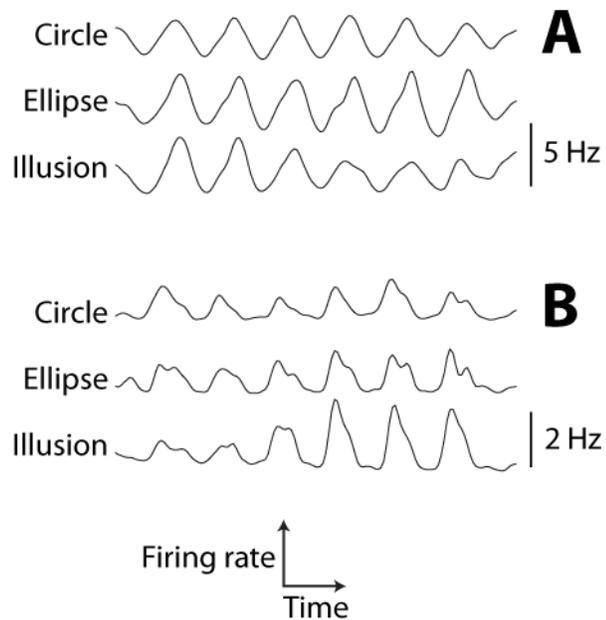
current position of the arm and the average elliptical trajectory across all trials. To calculate positional error, we first constructed an idealized trajectory by taking the mean of all trials within a specific condition (for instance, ellipse cycle 5). Then, at each instant in time, we identified the point on this idealized figure closest to the monkey's hand. We subtracted the x-position of the monkey's hand from the x-position of the nearest point on the figure (the figure was interpolated at high resolution to avoid aliasing). We always used the visual position to create the ideal trajectory, but compared it to the actual position of the monkey's hand, so the error signal will be most obvious during late cycles of the illusion trials. During non-illusion trials, it will be zero on average, because the ideal trajectory is constructed from the average trajectory. But it will modulate during individual trials, because the monkey's trajectory varies from trial to trial.

## 5.4 RESULTS

We recorded 2,095 neurons over 25 sessions on two arrays in monkey F, and 339 neurons over 5 sessions in monkey T. The number of neurons recorded at each electrode is shown in Figure 25.

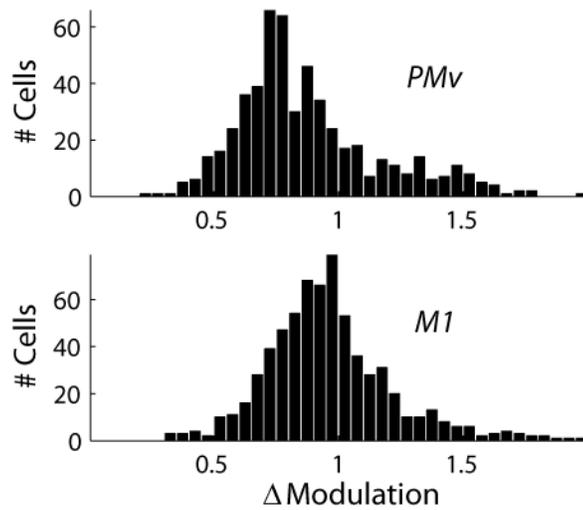
During the illusion trials, a dissociation was created between the actual movement of the monkey's hand and the movement displayed on the screen. In the last two cycles, the monkey makes circular movements with its hand while elliptical movements appear on the screen. In both M1 and PMv we observed a wide range of responses to the illusion, including cells that dramatically diminish or enhance their response during the illusion (Figure 28). We

analyzed the increase or decrease in modulation for all recorded cells and found that the dominant phenomenon is decreased modulation, especially in the part of PMv where we recorded (Figure 29). However, there was also a minority of neurons that increased their modulation in response to the illusion.

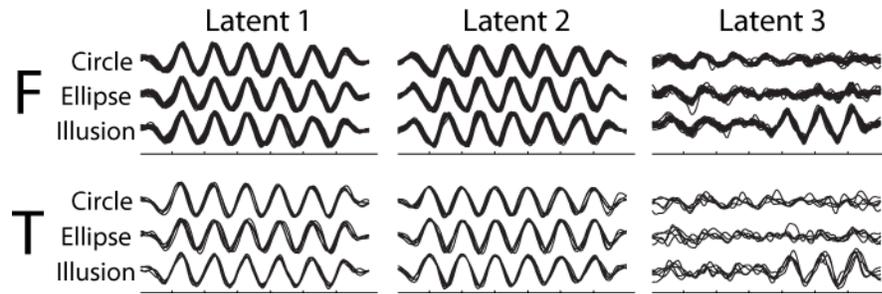


**Figure 28.** Average firing rate profile of two neurons during the circle, ellipse, and illusion versions of the drawing task. **A:** This cell is anti-illusion. It diminishes its modulation during the illusion, below what is expected from the actual (circular) movement. **B:** This cell is pro-illusion. It increases its firing rate during the illusion, even though the movement is getting smaller.

Individual neurons had complex cyclic responses during the drawing task. These responses changed shape as well as magnitude in the last two cycles of the illusion. Nonetheless, it is possible that there are simpler signals embedded in these complex responses. The canonical-correlation procedure described in Methods consistently identified the same three latent variables in every session of drawing task, in both monkeys (Figure 30). The first two correspond closely to horizontal and vertical movement direction. The third variable emerged in the late cycles of the illusion, corresponding to the time when a disparity was created between the actual movement of the monkey's hand and the apparent movement on the monitor. Within individual trials, we compared the latent variables to the horizontal error of the monkey's trajectory (Methods) and found that latent variable 3 was tightly correlated with the difference between actual and ideal position approximately 150 ms in the future (Figure 31). During non-illusion trials, latent 3 appears inconsistently from trial to trial because the monkey makes different errors each time. Therefore, it averages out and is not visible in Figure 30. However, we can compare the instantaneous value of latent 3 to the instantaneous value of horizontal error. The source of horizontal error is different in illusion versus non-illusion trials. In illusion trials, error is created by the experiment, while in non-illusion circle and ellipse trials, it comes from small errors in the monkey's movements.



**Figure 29.** Distributions of illusion response scores. A neuron with a score of 1 has the same amplitude of modulation in the last two cycles of the illusion as the first two.  $< 1$  means modulation decreases (A in the previous figure),  $> 1$  means modulation increases (B in the previous figure). PMv neurons are more likely to be diminished by the illusion, but both populations exhibit a range of responses. The details of calculating these scores are described in Methods.



**Figure 30.** We consistently identify the same 3 latent variables in the drawing task. The above plots show the activation of these variables, calculated independently for each session, overlaid on one another. They were extracted from the neural firing rates using the weights that result from canonical correlation. Latent variables 1 and 2 correspond closely to the horizontal and vertical component of movement direction, while latent 3 coincides with the discrepancy between actual and visual movement that appears in the late cycles of the illusion condition. The x-axis of each of the 6 plots indicates rescaled time, as described in Methods. Tick marks correspond to boundaries between cycles. Activations are unitless.

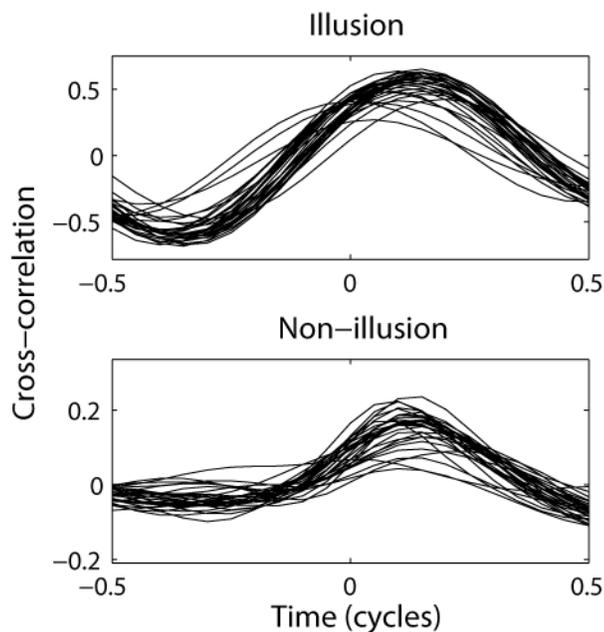


Figure 31: Latent 3 is correlated with an error signal. Shown are cross correlograms between latent 3 activation and the horizontal error signal that is described in Methods. Correlograms were calculated separately for each session and are shown overlaid. The relationship between latent 3 and future x-error holds for both the illusion trials (top), where error is created by the experiment, and non-illusion trials (bottom), where error comes from the monkey. The x-axis represents rescaled time; the peak in the cross-correlograms corresponds to ~150 ms.

## 5.5 DISCUSSION

We were able to leverage some of the special characteristics of our task to extract meaningful latent variables. Movement and neural activity in the drawing task was cyclic, so we assumed that many of the interesting signals would follow that cycle, and constrained our search to latent variables that resembled sinusoids. The same three latent variables were found in every session. Latent variables 1 and 2 correspond closely to the horizontal and vertical direction of movement, consistent with the previously identified representation of direction in motor cortex (Georgopoulos et al. 1982). Latent variable 3 is more novel. Previous studies have shown that corrective movement influences the activity of neurons in motor and premotor cortex (Flament et al. 1993), but the precise nature of this signal was unknown because it was usually weakly represented in neurons that encoded other parameters simultaneously. By directly extracting this signal, we have been able to show that it is correlated with positional error ~150 ms in the future, whether that “error” is the result of a visuomotor distortion or the trial-to-trial variability of the monkey. It may seem odd that a neural signal would be correlated with future positional error, since positional error is the

cause rather than the result of corrective movement. However, the idea of predicting future error for the purpose of making online adjustments to ongoing movement is common in control theory (Miall et al. 1993; Todorov and Jordan 2002).

Traditional thinking about motor and premotor cortex has localized error-correction primarily in PMd (Gomez et al. 2000; Wise et al. 1998). Most of the neurons we recorded responded to the illusion with modest decreases in modulation (Figure 29). But this does not necessarily mean that error-correction is not a part of their function. Consider two ways a signal might be embedded in a set of neurons. In the simpler case, some percentage of the population is driven primarily by the signal of interest. Studying this signal is simply a matter of finding those neurons. In the more difficult case, the signal of interest may weakly drive a large percentage of the population, with other unrelated signals present in the same neurons. In this case there may not be any individual neurons that clearly show the signal of interest. The only way to observe such a signal is to record the entire population and demix the latent variables. Latent 3 seems to be an example of this type of encoding. The fact that we observe such a consistent representation of the error signal implies the existence of a separate channel within the direction-encoding neurons, which anticipates the future need for error correction. This signal influenced neurons that were widely distributed across our recording electrodes; its distribution was approximately the same as the distribution of recorded neurons in Figure 25.

It has previously been shown that in this task, some neurons follow the apparent movement on the monitor while others follow the actual movement of the monkey's hand (Schwartz et al. 2004). A hypothetical population of neurons tuned to movement velocity and

position in both visual and motor coordinates would contain signals like latent 3, due to the difference between the motor and visual neurons late in the illusion. However, they will not replicate the relationship between latent 3 and the small errors in non-illusion trials, because there is no difference between the motor and visual signals in non-illusion trials. Furthermore, the exercise of constructing a population of neurons tuned to specific features of movement and then showing they can replicate aspects of the results misses the point of doing latent variable analysis. We can hypothesize many different encoding schemes that would produce differential firing rate modulation in the late cycles of the illusion, and thereby replicate the main feature of latent 3. The purpose of the analysis in this paper is to avoid specifying an encoding scheme *a priori*. Instead we extract the important signals directly from the data and then study them.

We do impose some assumptions by using the processed kinematics matrix (Figure 27) as one of the inputs to canonical correlation. The sines and cosines in the rows of this matrix biased the canonical correlation technique to extract latent variables which are cyclic. Because of the circular nature of the drawing task this is a very broad assumption; many different hypothetical encoding schemes will produce a cyclic signal in this task. Indeed, the error signal we identified in latent 3 is compatible with multiple interpretations. It might reflect a disparity between the monkeys expected perception and reality, or an upcoming corrective movement, or simply a computational element of the brain that has no intuitively meaningful description. We have no way of knowing whether this signal is being used by the brain for any function; it could be a faint effect of a stronger error signal located in another area. We can only say that there is a consistent signal in these brain areas that is correlated

with upcoming positional error. The most telling finding is that in non-illusion trials, when the monkey deviates from its average trajectory the activation of latent 3 follows those deviations.

Latent variable analysis is based on the idea that correlation across recorded neurons is generated by factors that drive each neuron in a common manner. This common drive may be too weak to observe clearly in individual firing rates. By looking at a population recorded simultaneously, these effects become evident. Furthermore, because they are extracted directly from the data, they do not require *a priori* assumptions about the identity of an encoded variable. The directional components in latent 1 and 2 were expected, but the discovery of the error-related signal in latent 3 was surprising. The fact that corrective movement influences neural activity in motor and premotor cortex has been demonstrated before, but it has not been possible to directly observe an error signal because of the way it is embedded amongst stronger signals in individual neurons. Simultaneous recordings of populations, and the new analysis techniques they enable, have revealed a previously inaccessible signal.

## **6.0 A PLATFORM FOR WIDESPREAD SIMULTANEOUS RECORDING**

In this chapter, we describe a recording chamber and an array of 256 individually moveable electrodes that can be used to recording the activity of neurons across many brain areas. The recording chamber has been used in four monkeys over the last two years without a single failure. While the electrode array has not yet been used to record in a monkey performing a behavioral task, it has successfully recorded neural activity in full-scale prototypes.

### **6.1 A LEGGED TITANIUM RECORDING CHAMBER**

Recording neural activity in trained macaques requires permanent access to the brain. This involves creating a hole in the skull and then somehow protecting that hole from infection. Most existing schemes use a recording chamber—a metal cylinder, usually affixed to the skull with dental acrylic. The need to place a significant quantity of dental acrylic around the edge limits the size of the recording chamber. Also, this method of attachment is prone to failure, as the dental acrylic tends to gradually detach from the underlying skull. While individual monkeys will sometimes retain their recording chambers for years, this failure mode cannot be completely eliminated.

To allow a larger portion of the brain to be accessed in a single chamber, and to eliminate the detachment failure mode, we developed a recording chamber that included legs that could be screwed directly into the skull (Figure 32).

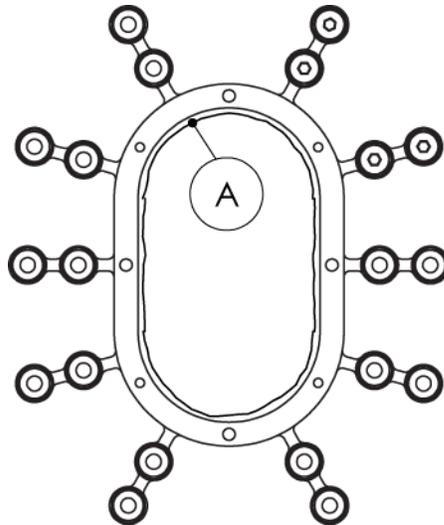


**Figure 32:** Legged recording chamber.

This recording chamber was milled out of a single piece of titanium using a CNC machine. It consisted of two parts: an outer rim and an inner sleeve. The outer rim would be implanted completely under the skin for a minimum of two months. During this time the monkey would be trained and the legs of the recording chamber would gradually become engulfed in bone, due to the osteoconductive property of titanium. This period of bone

integration lasted between two months and two years. We found that longer periods were better, and that if they were over a year the bone formed a nearly complete seal with the edge of the outer rim.

Following the period of bone integration, an incision would be made along the centerline of the pre-implanted outer rim of the chamber. We found that making an incision and displacing the skin to the side produced a healthier long-term skin interface than removing an oval of tissue. While the excess skin would be wrinkly immediately after surgery, the skin tended to retract somewhat in the weeks following, so an initially wrinkly skin edge would end up in the ideal position. The legs of the chamber would always remain buried underneath the skin and muscle tissue. An oval-shaped craniotomy would be cut through the skull in the center of the outer rim. A shelf of bone would be left extending  $\sim 1/32$ " towards the center of the recording chamber (Figure 33). We devised a simple tool to cut this shelf precisely: a standard burr bit with a piece of Teflon tubing fitted tightly around it. The Teflon tubing served both as a stand-off, to keep the edge of the bit  $1/32$ " from the inner face of chamber, and as a depth stop, preventing the bit from cutting too far through the skull and into the brain.

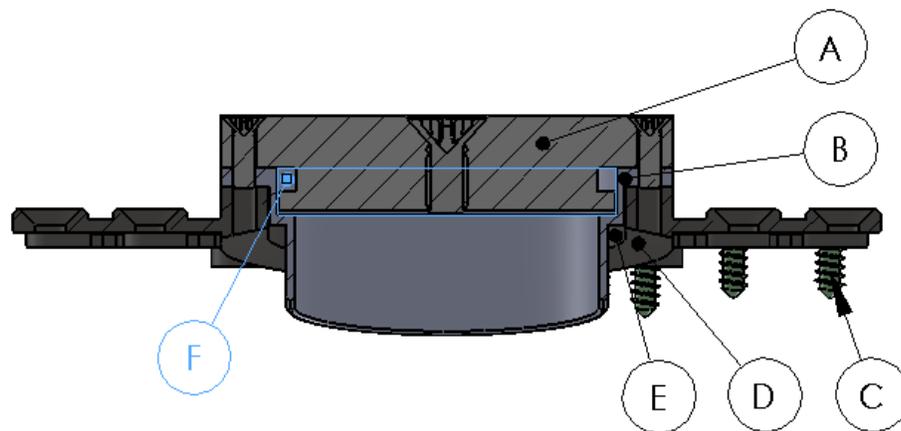


**Figure 33.** A craniotomy is cut in the center of the recording chamber, leaving a 1/32" bone shelf (A) extending inwards from the pre-implanted fixture.

After the craniotomy was cut, we would slide the inner sleeve into the recording chamber. The bottom edge of the inner sleeve needs to match the shape of the surface of the dura. We found the best way to accomplish this was simply to manufacture the inner sleeve with excessive depth and a .02" inner wall that we would then cut with a large burr during the surgery. By repeatedly inserting the sleeve and then adjusting the fit, we could produce an excellent match to the surface of the dura in about half an hour.

The inner sleeve must seal against the cut edge of the bone to prevent infection of the chamber. We experimented with several ways to accomplish this, but we found the best technique was to use a small amount of dental acrylic. We first placed an oval shaped piece of latex glove on the surface of the dura and tucked its edges under the cut edge of the bone.

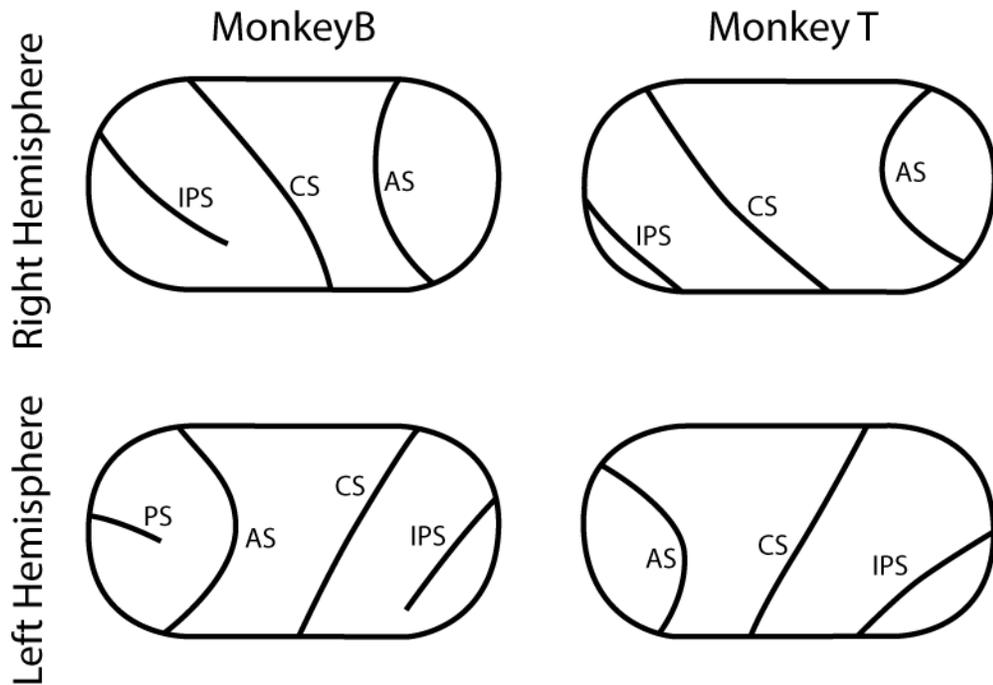
We then mixed dental acrylic, and when it reached the viscosity of honey we poured an excess directly on the cut edge of bone. It spilled onto the latex glove protecting the dura, and the inner sleeve was firmly inserted all the way into the outer rim (Figure 34). This would displace some of the dental acrylic into the center of the recording chamber. We would immediately pour saline solution into the chamber to absorb heat from the curing acrylic. At this time we would also insert the four flathead screws that attach the inner sleeve to the outer chamber. After about 30 seconds the dental acrylic would achieve the consistency of bubble gum and we would pull out the piece of latex glove protecting the dura, sweeping away the excess dental acrylic with it.



**Figure 34.** A cross-section of the assembled recording chamber. **A:** Lid. **B:** Inner sleeve, extends down to dura. **C:** Bone screw. **D:** Outer rim. **E:** The gap between the inner sleeve, the outer rim, and the bone shelf is filled with dental acrylic. **F:** There is a groove in the lid which tightly fits a silicone o-ring that seals the chamber.

At this point the only remaining step was to place stitches at the ends of the recording chamber, where the centerline incision tended to extend somewhat beyond the edge. A purse-string suture was always placed to ensure the skin was pulled tightly against the edge of the chamber as it healed. We attached a plastic lid using four screws; the plastic cap had an internal o-ring that sealed the chamber. The center of the lid had a tapped hole with a flathead screw that could be removed to allow the volume of the chamber to equalize as the lid was pressed into place.

The legged recording chamber provides two principal advantages: its superior reliability, and the large area of the cortical surface that can be accessed. Figure 35 shows tracings of the cortical sulci in several implanted recording chambers, based on intraoperative photographs.



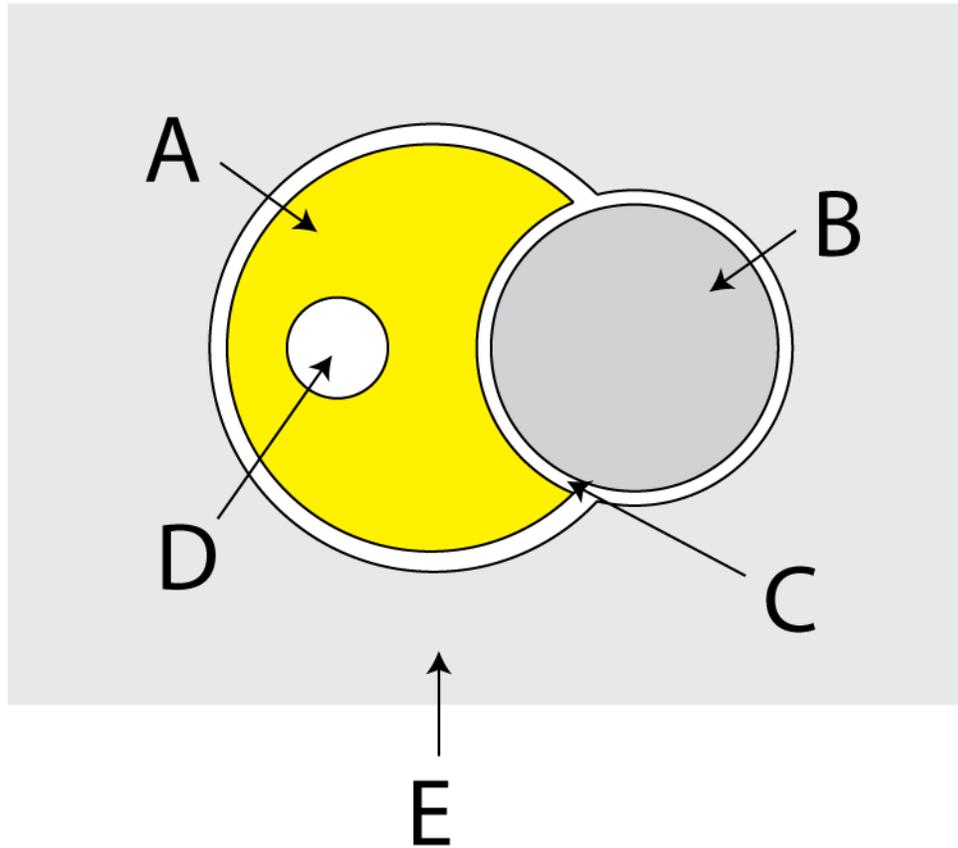
**Figure 35:** Tracings of the cortical sulci that were visible in the lumen of several legged recording chambers. The accessible surface extends from the dorsolateral prefrontal cortex to some posterior parietal areas. AS, arcuate sulcus. CS, central sulcus. IPS, intraparietal sulcus. PS, principal sulcus.

## 6.2 THE 256-ELECTRODE ARRAY

The development of the legged recording chamber provided access to a much larger area of the brain than was possible with previous designs. We sought to leverage this capability by developing an array of 256 individually moveable electrodes that could be inserted directly into the chamber. The linear motion of each electrode was actuated by turning a screw. The screw was stationary; its threads interlocked with a moveable shuttle in which the electrode was mounted. Both screw and shuttle were inserted into a pair of overlapping deep holes in a

plastic block. The deep holes served to trap the shuttle against the screw so that when the screw was turned, the shuttle would move up or down. This mechanism, diagrammed in Figure 36, was reliable and inexpensive to implement. It did require the development of some creative machining strategies. It is not possible to drill two deep holes that overlap one another with conventional drill bits. We drilled the smaller of the two holes with a standard drill, then drilled the larger hole with an extra long 3/64" end mill using a short peck operations that each make the hole slightly deeper.

The brass shuttles are moon-shaped and have a thread cut into one side of them. To produce this unusual feature, we drilled and tapped a grid of 000-120 threaded holes in a solid block of brass. We then cut the appropriate moon shape around each existing hole. The brass block was thicker than the shuttles, so this left an array of finished shuttles attached to a solid brass support. The block was then flipped over and glued with Crystalbond (Electron Microscopy Sciences) into a fixture which had a matching hole for every shuttle that was sticking out of the face of the brass block. The back side of the block was machined off, leaving the shuttles glued into the holes of the fixture. We then dissolved the Crystalbond with acetone and sonication, which served both to thoroughly dissolve the adhesive and to vibrate all the shuttles out of the fixture. It took several hours to machine an array of 320 shuttles, but the CNC machine is completely automated and ran unattended.

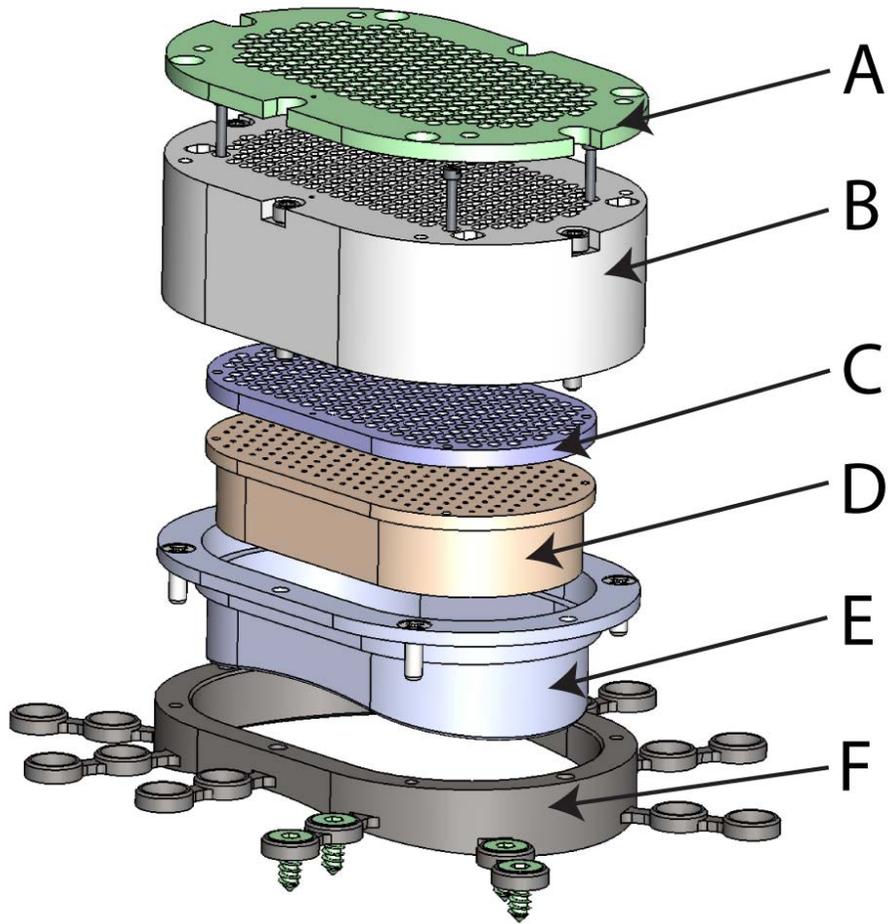


**Figure 36.** Cross-section of the linear motion mechanism used in the 256-electrode drive. **A:** Brass shuttle. **B:** Stainless steel screw. **C:** Interlocking threads between screw and shuttle. **D:** Hole for electrode to be mounted in. **E:** Plastic block (gray region) that the screw and shuttle are embedded in. The screw never moves up or down; by turning the screw, we advance or retract the shuttle with the electrode mounted in it. The screw and shuttle are held together by the surrounding plastic block.

We arranged to have electrodes manufactured by FHC at precisely the right length for the array so that they could be mounted directly into the shuttles by crimping the back end of the electrode with a manual pliers and then force-fitting into the hole in the shuttle. Several

other methods of mounting were tried but crimping was the only one sufficiently reliable to be used in an array of hundreds of electrodes.

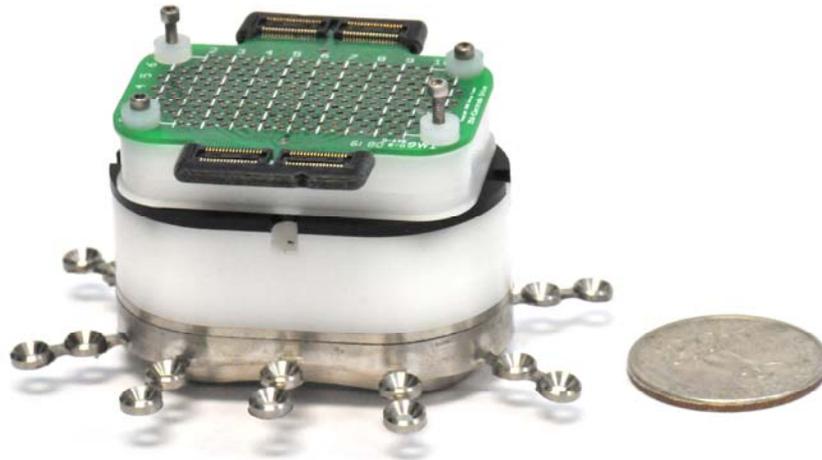
The linear motion mechanism was scaled to a 256-electrode array by simply repeating the pattern of screw/shuttle modules in a plastic block designed to fit into the recording chamber. This plastic block was surrounded by several layers designed to hold the shuttle assemblies in place and create a seal between the brain and the outside world. An exploded view of the complete assembly is diagrammed in Figure 37.



**Figure 37.** Exploded view of the 256-electrode drive and recording chamber. **A:** Push-down plate, holds the screws down. **B:** Guide block, includes 256 columns for the screw/shuttle assembly. **C:** Membrane sandwicher, traps a .005" silicone rubber membrane between C & D which is pierced by each electrode. **D:** Guide holes, includes a small hole for each electrode. **E:** Inner sleeve. **F:** Outer rim.

To record voltage signals from the electrodes, we need to make a separate electrical connection to each one. The tungsten electrode is already electrically connected to each screw via the brass shuttle. We made a spring-loaded connector that pressed a brass piston down on the head of each screw. Above each spring was another piston that pressed against

a 2-layer printed circuit board, which directed all the signals into two Tucker-Davis ZC128 connectors (Figure 38).



**Figure 38.** Assembled electrode drive with removeable connector. Includes a quarter for scale.

Several prototype electrode arrays were assembled (Figure 39) and successfully recorded neural activity. They were tested in monkeys with existing recording chambers where neural recording had already been performed using traditional semi-acute techniques where the electrodes were introduced and removed each day. Some early problems with seals have been resolved and at this time the drive is ready to be used in a full-scale experiment.



**Figure 39.** Assembled electrode drive, inverted to show fully extended electrodes.

## BIBLIOGRAPHY

**Bell AJ.** The co-information lattice. In: *ICA2003*.

**Brockwell AE, Rojas AL, and Kass RE.** Recursive bayesian decoding of motor cortical signals by particle filtering. *J Neurophysiol* 91: 1899-1907, 2004.

**Carmena JM, Lebedev MA, Henriquez CS, and Nicolelis MA.** Stable ensemble performance with single-neuron variability during reaching movements in primates. *J Neurosci* 25: 10712-10716, 2005.

**Chestek CA, Batista AP, Santhanam G, Yu BM, Afshar A, Cunningham JP, Gilja V, Ryu SI, Churchland MM, and Shenoy KV.** Single-neuron stability during repeated reaching in macaque premotor cortex. *J Neurosci* 27: 10742-10750, 2007.

**Churchland MM, and Shenoy KV.** Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *J Neurophysiol* 97: 4235-4257, 2007.

**Churchland MM, Yu BM, Cunningham JP, Sugrue LP, Cohen MR, Corrado GS, Newsome WT, Clark AM, Hosseini P, Scott BB, Bradley DC, Smith MA, Kohn A, Movshon JA, Armstrong KM, Moore T, Chang SW, Snyder LH, Lisberger SG, Priebe NJ, Finn IM, Ferster D, Ryu SI, Santhanam G, Sahani M, and Shenoy KV.** Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nat Neurosci* 13: 369-378, 2010.

**Come E, Oukhellou L, Denoeux T, and Aknin P.** Learning from partially supervised data using mixture models and belief functions. *Pattern Recogn* 42: 334-348, 2009.

**Dickey AS, Suminski A, Amit Y, and Hatsopoulos NG.** Single-unit stability using chronically implanted multielectrode arrays. *J Neurophysiol* 102: 1331-1339, 2009.

**Evarts EV, and Tanji J.** Reflex and intended responses in motor cortex pyramidal tract neurons of monkey. *J Neurophysiol* 39: 1069-1080, 1976.

**Fetz EE, and Cheney PD.** Postspike facilitation of forelimb muscle activity by primate corticomotoneuronal cells. *J Neurophysiol* 44: 751-772, 1980.

**Flament D, Fortier PA, and Fetz EE.** Response patterns and postspike effects of peripheral afferents in dorsal root ganglia of behaving monkeys. *J Neurophysiol* 67: 875-889, 1992.

**Flament D, Onstott D, Fu QG, and Ebner TJ.** Distance- and error-related discharge of cells in premotor cortex of rhesus monkeys. *Neurosci Lett* 153: 144-148, 1993.

**Fraser GW, Chase SM, Whitford A, and Schwartz AB.** Control of a brain-computer interface without spike sorting. *J Neural Eng* 6: 055004, 2009.

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

**Gentilucci M, Fogassi L, Luppino G, Matelli M, Camarda R, and Rizzolatti G.** Functional organization of inferior area 6 in the macaque monkey. I. Somatotopy and the control of proximal movements. *Exp Brain Res* 71: 475-490, 1988.

**Georgopoulos A.** Relations between two-dimensional arm movements and single-cell discharge in motor cortex and area 5: movement direction versus movement end point. *experimental brain research* 10: 175, 1985.

**Georgopoulos AP, Kalaska JF, Caminiti R, and Massey JT.** On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J Neurosci* 2: 1527-1537, 1982.

**Georgopoulos AP, Kettner RE, and Schwartz AB.** 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, 1988.

**Gilja V, Nuyujukian P, Chestek CA, Cunningham JP, Yu B, Ryu S, and Shenoy KV.** High-performance continuous neural cursor control enabled by a feedback control perspective. In: *Frontiers in Neuroscience* 2010.

**Gomez JE, Fu Q, Flament D, and Ebner TJ.** Representation of accuracy in the dorsal premotor cortex. *Eur J Neurosci* 12: 3748-3760, 2000.

**Gomi H, and Kawato.** Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science* 272: 117-120, 1996.

**Greenberg PA, and Wilson FA.** Functional stability of dorsolateral prefrontal neurons. *J Neurophysiol* 92: 1042-1055, 2004.

**Harrison RR, Kier RJ, Chestek CA, Gilja V, Nuyujukian P, Ryu S, Greger B, Solzbacher F, and Shenoy KV.** Wireless neural recording with single low-power integrated circuit. *IEEE Trans Neural Syst Rehabil Eng* 17: 322-329, 2009.

**Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, Caplan AH, Branner A, Chen D, Penn RD, and Donoghue JP.** Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* 442: 164-171, 2006.

**Hotelling H.** Relations Between Two Sets of Variates. *Biometrika* 28: 321-377, 1936.

**Jackson A, and Fetz EE.** Compact movable microwire array for long-term chronic unit recording in cerebral cortex of primates. *J Neurophysiol* 98: 3109-3118, 2007.

**Jarosiewicz B, Chase SM, Fraser GW, Velliste M, Kass RE, and Schwartz AB.** Functional network reorganization during learning in a brain-computer interface paradigm. *Proc Natl Acad Sci U S A* 105: 19486-19491, 2008.

**Koyama S, Chase SM, Whitford AS, Velliste M, Schwartz AB, and Kass RE.** Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control. *J Comput Neurosci* 29: 73-87, 2010.

**Lambiotte R, Delvenne JC, and M. B.** Laplacian Dynamics and Multiscale Modular Structure in Networks. In: *arXiv*2009.

**Lanquillon C.** Partially supervised text classification: Combining labeled and unlabeled documents using an EM-like scheme. *Lect Notes Artif Int* 1810: 229-237, 2000.

**Leuthardt EC, Schalk G, Wolpaw JR, Ojemann JG, and Moran DW.** A brain-computer interface using electrocorticographic signals in humans. *J Neural Eng* 1: 63-71, 2004.

**Li CS, Padoa-Schioppa C, and Bizzi E.** Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field. *Neuron* 30: 593-607, 2001.

**Luppino G, and Rizzolatti G.** The Organization of the Frontal Motor Cortex. *News Physiol Sci* 15: 219-224, 2000.

**Matsumura M, Chen DF, Sawaguchi T, Kubota K, and Fetz EE.** Synaptic interactions between primate precentral cortex neurons revealed by spike-triggered averaging of intracellular membrane potentials in vivo. *Journal of Neuroscience* 16: 7757-7767, 1996.

**Maynard EM, Nordhausen CT, and Normann RA.** The Utah intracortical Electrode Array: a recording structure for potential brain-computer interfaces. *Electroencephalogr Clin Neurophysiol* 102: 228-239, 1997.

**Mehring C, Rickert J, Vaadia E, Cardoso de Oliveira S, Aertsen A, and Rotter S.** Inference of hand movements from local field potentials in monkey motor cortex. *Nat Neurosci* 6: 1253-1254, 2003.

**Miall RC, Weir DJ, Wolpert DM, and Stein JF.** Is the cerebellum a smith predictor? *J Mot Behav* 25: 203-216, 1993.

**Moore GP, Segundo JP, Perkel DH, and Levitan H.** Statistical signs of synaptic interaction in neurons. *Biophys J* 10: 876-900, 1970.

**Mucha PJ, Richardson T, Macon K, Porter MA, and Onnela JP.** Community structure in time-dependent, multiscale, and multiplex networks. *Science* 328: 876-878, 2010.

**Musallam S, Corneil BD, Greger B, Scherberger H, and Andersen RA.** Cognitive control signals for neural prosthetics. *Science* 305: 258-262, 2004.

**Mussa-Ivaldi FA, Hogan N, and Bizzi E.** Neural, mechanical, and geometric factors subserving arm posture in humans. *J Neurosci* 5: 2732-2743, 1985.

**Pesaran B, Pezaris JS, Sahani M, Mitra PP, and Andersen RA.** Temporal structure in neuronal activity during working memory in macaque parietal cortex. *Nat Neurosci* 5: 805-811, 2002.

**Rokni U, Richardson AG, Bizzi E, and Seung HS.** Motor learning with unstable neural representations. *Neuron* 54: 653-666, 2007.

**Rousche PJ, and Normann RA.** Chronic recording capability of the Utah Intracortical Electrode Array in cat sensory cortex. *J Neurosci Methods* 82: 1-15, 1998.

**Sahani M.** Latent variable models for neural data analysis. Pasadena, Calif.: California Institute of Technology, 1999, p. xv, 162 leaves.

**Santhanam G, Linderman MD, Gilja V, Afshar A, Ryu SI, Meng TH, and Shenoy KV.** HermesB: a continuous neural recording system for freely behaving primates. *IEEE Trans Biomed Eng* 54: 2037-2050, 2007.

**Santhanam G, Ryu SI, Yu BM, Afshar A, and Shenoy KV.** A high-performance brain-computer interface. *Nature* 442: 195-198, 2006.

**Santhanam G, Sahani M, Ryu S, and Shenoy K.** An extensible infrastructure for fully automated spike sorting during online experiments. *Conf Proc IEEE Eng Med Biol Soc* 6: 4380-4384, 2004.

**Schmidt EM, Bak MJ, and McIntosh JS.** Long-term chronic recording from cortical neurons. *Exp Neurol* 52: 496-506, 1976.

**Schwartz AB, Kettner RE, and Georgopoulos AP.** Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement. *J Neurosci* 8: 2913-2927, 1988.

**Schwartz AB, Moran DW, and Reina GA.** Differential representation of perception and action in the frontal cortex. *Science* 303: 380-383, 2004.

**Scott SH.** Inconvenient truths about neural processing in primary motor cortex. *J Physiol* 586: 1217-1224, 2008.

**Shi RJ, and MacGregor JF.** Modeling of dynamic systems using latent variable and subspace methods. *J Chemometr* 14: 423-439, 2000.

**Smith AM, Hepp-Reymond MC, and Wyss UR.** Relation of activity in precentral cortical neurons to force and rate of force change during isometric contractions of finger muscles. *Exp Brain Res* 23: 315-332, 1975.

**Stark E, and Abeles M.** Predicting movement from multiunit activity. *J Neurosci* 27: 8387-8394, 2007.

**Taylor DM, Tillery SI, and Schwartz AB.** Direct cortical control of 3D neuroprosthetic devices. *Science* 296: 1829-1832, 2002.

**Todorov E, and Jordan MI.** Optimal feedback control as a theory of motor coordination. *Nat Neurosci* 5: 1226-1235, 2002.

**Tolias AS, Ecker AS, Siapas AG, Hoenselaar A, Keliris GA, and Logothetis NK.** Recording chronically from the same neurons in awake, behaving primates. *J Neurophysiol* 98: 3780-3790, 2007.

**Velliste M, Perel S, Spalding MC, Whitford AS, and Schwartz AB.** Cortical control of a prosthetic arm for self-feeding. *Nature* 453: 1098-1101, 2008.

**Ventura V.** Spike train decoding without spike sorting. *Neural Comput* 20: 923-963, 2008.

**Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, and Nicolelis MA.** Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* 408: 361-365, 2000.

**Williams JC, Rennaker RL, and Kipke DR.** Stability of chronic multichannel neural recordings: Implications for a long-term neural interface. *Neurocomputing* 26-7: 1069-1076, 1999.

**Wise SP, Moody SL, Blomstrom KJ, and Mitz AR.** Changes in motor cortical activity during visuomotor adaptation. *Exp Brain Res* 121: 285-299, 1998.

**Wolpaw JR, McFarland DJ, Neat GW, and Forneris CA.** An EEG-based brain-computer interface for cursor control. *Electroencephalogr Clin Neurophysiol* 78: 252-259, 1991.

**Wu W, Gao Y, Bienenstock E, Donoghue JP, and Black MJ.** Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Comput* 18: 80-118, 2006.

**Yu BM, Cunningham JP, Santhanam G, Ryu SI, Shenoy KV, and Sahani M.** Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *J Neurophysiol* 102: 614-635, 2009.