# INVESTIGATING THE NEURAL BASIS OF LEARNING USING BRAIN-COMPUTER INTERFACES

by

**Patrick Thomas Sadtler**

BS, Purdue University, 2009

Submitted to the Graduate Faculty of
the Swanson School of Engineering in partial fulfillment
of the requirements for the degree of

**Doctor of Philosophy**

University of Pittsburgh

2014

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Patrick Thomas Sadtler

It was defended on

November 21, 2014

and approved by

Aaron Batista, Ph.D., Assistant Professor, Department of Bioengineering

Byron Yu, Ph.D., Assistant Professor, Departments of Electrical and Computer Engineering and Biomedical Engineering, Carnegie Mellon University

Zhi-Hong Mao, Ph.D., Associate Professor, Departments of Electrical and Computer Engineering and Bioengineering

Peter Strick, Ph.D., Distinguished Professor and Chair, Department of Neurobiology, Professor, Department of Psychiatry

Doug Weber, Ph.D., Associate Professor, Departments of Bioengineering and Physical Medicine and Rehabilitation

Dissertation Director: Aaron Batista, Ph.D., Assistant Professor, Department of Bioengineering

# INVESTIGATING THE NEURAL BASIS OF LEARNING USING BRAIN-COMPUTER INTERFACES

Patrick Thomas Sadtler, PhD

University of Pittsburgh, 2014

Learning a new skill requires one to produce new patterns of activity among networks of neurons. This applies not only to physical skills, such as learning to play a new sport, but also to abstract skills, such as learning to play chess. An abstract skill that we can use to study the neural mechanisms of learning, in general, is controlling a brain-computer interface (BCI). BCIs were conceived of as assistive devices to help people with paralysis, limb-loss, or other neurological disorder, but they have also proven effective as tools to study the neural basis of sensory-motor control and learning.

We tested the ability of subjects to generate neural activity patterns required to control arbitrary BCI decoders. We found that the subjects could more easily learn to control the decoder when they could use existing patterns of neural activity than when they needed to generate new patterns. We also analyzed the way in which subjects adapted their neural activity during learning. We found that neural activity adapts in a way that is consistent with the learning-related performance improvements and that the trial-to-trial variability of neural activity decreased as performance improved.

We tested how specific properties of BCI decoders, which translate neural activity into movements of the effector, influence the ability to learn to control a BCI by incorporating dimensionality reduction into a Kalman filter and assessing how performance related to the number of latent dimensions. We found that the subjects could use a standard Kalman filter just as well as a Kalman filter that incorporates dimensionality reduction. However, as the dimensionality of the model increased, performance improved up to an asymptotic level. Lastly, we tested whether increasing the difficulty of a task would lead the subjects to learn to demonstrate better BCI performance. We implemented an instructed path task that required the animals to move a cursor along re-defined paths, and we found that this task motivated one monkey to improve his performance. In all, these studies help to uncover what contributes to BCI control, and they help pave the way for transitioning BCIs from the lab to the clinic.

## TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# PREFACE

The following chapters and appendices of this dissertation have been published, are currently under review, or are in preparation for publication.

Chapter 3:

**Sadtler PT, Quick KM, Golub MD, Chase SM, Ryu SI, Tyler-Kabara EC, Yu BM and Batista AP**. Neural constraints on learning. *Nature* 512:423-426. 2014.

Chapter 4: in preparation.

Chapter 6:

**Sadtler PT, Ryu SI, Tyler-Kabara EC, Yu BM & Batista AP.** Brain-computer interface control along instructed paths. *J Neural Eng.* in review.

**Sadtler PT, Ryu SI, Yu BM & Batista AP.** High-performance neural prosthetic control along instructed paths. *Proc 5th International Conf IEEE EMBS Neural Engineering*. 601-604. 2011.

Appendix C:

**Perel S, Sadtler PT, Godlove JM, Ryu SI, Wang W, Batista AP & Chase SM.** Direction and Speed Tuning of Motor-Cortex Multi-Unit Acivity and Local Field Potentials During Reaching Movements. *Proc 35th Annual Conf IEEE EMBS*. 299-302. 2014.

**Todorova S, Sadtler P, Batista A, Chase S & Ventura V.** To sort or not to sort: the impact of spike-sorting on neural decoding performance. *J Neural Eng.* 11(5):056005. 2014.

***Acknowledgments*** I'd like to first thank my advisors, Dr. Aaron Batista and Dr. Byron Yu. Our relationship has come to feel more that of friends and co-workers rather than as advisor/advisee. Working with them for the past five years has been a great experience, and I hope that I can take away some of their best qualities.

Aaron taught me a lot about neuroscience and also about how to be a good leader. His enthusiasm and excitement for discovering how the brain works was infectious. I never doubted that Aaron would support my individual goals and aspirations.

Byron has a very strong attention to detail. He taught me how to thoroughly analyze a problem and how to think carefully about interpreting result in light of the methods being used. Despite having numerous collaborations, Byron was always able to devote time and effort to each one.

I'd also like to thank the members of the Batista lab. Working with and alongside you has been fun and rewarding. Kristin Quick, Berook Alemayehu, and I joined the lab in the same year. I've enjoyed our 'science journey' together. I'd like to especially thank Kristin, Sharlene Flesher, and Kara Bocan for help with collecting some of the data presented in this dissertation. I cannot think Jocelyn Stafura and Melissa James enough. Our lab would not be as successful as it is without the care that you provided to the monkeys. I look forward to hearing about future Batista lab accomplishments!

I'd like to thank the members of the Yu lab, Dr. Steve Chase, and the members of his lab. I've learned so much from our collaborations and countless Friday afternoon meetings. I could not have accomplished this work without your valuable input.

Lastly, I'd like to thank my family for all of the support they have given to me. They have been instrumental in helping me to be successful.

# 1.0 GENERAL INTRODUCTION

Learning is an essential part of life. It is a process that occurs everyday whether or not we are aware of it. In young children, learning is often tied to development. Children first learn to crawl and then to walk and eventually to run. As development continues, children learn other activities, such as riding a bike or swimming. As we get older, people can learn to specialize their physical skills, for example, to play one sport or another. In addition, intellectual learning occurs often. Young children learn the alphabet and learn to count. As we progress, we learn a multitude of subjects in school. Often, we learn in-depth knowledge in one area of interest as a career focus. In addition to physical and intellectual learning, other forms of learning are also common. For example, we might learn to associate the scent of a particular type of flower with a favorite tropical vacation.

The question of how learning occurs in the brain has been studied from many different aspects, ranging from the systems level (Merzenich et al., 1984; Pons et al., 1991; Ungerleider et al., 2002; Orban de Xivry et al., 2012) to single neurons (Li et al., 2001; Rokni et al., 2007; Picard et al., 2013), and even to specific cellular processes (Buonomano and Merzenich, 1998; Jackson et al., 2006; Holtmaat and Svoboda, 2009; Kuhlman et al., 2014; McKenzie et al., 2014). In this work, I address learning from the aspect of populations of neurons. Evidence suggests that this might be the level of organization from which behavior, sensation, and other cognitive processes arise. For example, limb trajectories (Wu et al., 2006; Yu et al., 2007) and visual information (Graf et al., 2011) can be accurately reconstructed by analyzing the activity of populations of neurons.

A seemingly unrelated topic in neuroscience is brain-computer interfaces (BCIs). BCIs are primarily being developed to provide people who are paralyzed or suffer from limb-loss or other neurological disorder with a higher quality of life. Some BCIs are designed to allow the user control a cursor on a computer, while others are designed to allow the user to control a robotic limb. In Chapter 3, I describe how we use a BCI to study the ability of subjects to learn. In Chapter 4, I describe how we have begun to uncover the specific neural mechanisms of learning from a neural population point of view. In Chapters 5 and 6, we tested ways in which we could adjust the BCI framework – both the details of the mapping from neural activity to cursor movement and the task the subjects were undertaking – in order to improve learning and controllability in a BCI setting.

In the following sections, I first provide an overview of the types of learning that are commonly studied in neuroscience (Section 1.1) as well as the different levels of the neurophysiological hierarchy from which learning has been studied (Section 1.2). I then focus on the properties of neurons in primary motor cortex, as that is the area of focus in this dissertation (Section 1.3). Finally, I describe why we use a BCI in order to study sensory-motor learning (Section 1.4).

## 1.1 TYPES OF LEARNING

In neuroscience, learning is often categorized into the broad categories of sensory learning, perceptual learning, and motor learning. These types of learning are not independent — they blend together and overlap. The songbird provides a good model system for studying perceptual learning (e.g., Fee and Scharff (2010); Brainard and Doupe (2002)). Songbirds are known for their abilities to generate highly repeatable songs and to replicate the songs of other birds within their species, which might be similar to a person learning to speak a language by hearing others use it. These studies have begun to uncover neural mechanisms related to the generation of sequential behaviors.

Perceptual learning is closely linked to sensory learning. Perceptual learning is described as the ability to discriminate different stimuli (Law and Gold, 2008). In one study, Newsome et al. (1989) presented monkeys with a visual display showing dots moving in random directions, with varying amounts of correlation. The monkeys were supposed to learn to identify the direction of correlated movement. The researchers recorded from area MT during these experiments, and they found that the reliability and sensitivity of individual neurons was on par or exceeded that of the monkeys. In another set of experiments, Romo et al. (2004) recorded from ventral premotor cortex (PMv) of monkeys while they stimulated the monkeys' fingertips with vibrations of different frequencies. The task required the monkeys to learn to identify which of two consecutive stimulations had a higher frequency, and the researchers found that single neurons in PMv carried information related to the frequency of each stimulus and the difference between frequencies. Studying humans, Pascual-Leone and Torres (1993) analyzed the size of the hand representations in somatosensory cortex in subjects who had learned to read Braille. They found that the cortical representation of the digit used to read was substantially larger than the representations of the digits not used for reading.

Finally, studies of motor learning typically analyze the way in which subjects move their arms in altered environments. Extensive research has been conducted studying how humans adapt to perturbations, which often take the form of visuomotor rotations (e.g., Mazzoni and Krakauer (2006); Herzfeld et al. (2014)) (during which the movement direction of a cursor is offset from the movement direction of one's limb by a constant angle) or force-fields (e.g., Lackner and Dizio (1994); Izawa et al. (2008)) (during which a force is applied to the movement of one's arm). In more complicated experimental paradigms, the type of perturbation changes rapidly across trials (e.g., Kording and Wolpert (2004); Herzfeld et al. (2014)).

These and other studies have allowed researchers to propose control systems that satisfy certain computational strategies for movement generation (Shadmehr and Krakauer, 2008). Various aspects of the control systems – for example, error integration, sensory estimation (i.e., forward internal models), and movement prediction (i.e., inverse internal models) – have been hypothesized to occur in different brain structures, but neurological evidence for this has been sparse and often relied on lesion studies. For example, in Nowak et al. (2007), subjects held on to a cup while a metal ball was dropped into the cup. The subject with the cerebellar lesion and the healthy subjects could react to the additional weight in the cup. However, only healthy subjects could predict (presumably the result of a forward model) the additional weight when they dropped it themselves. The cerebellar patient reacted in the same way as if someone else had dropped the ball into the cup. From this, the authors inferred that the cerebellum is involved in the formation of internal models.

These behavioral studies of motor learning set the stage for investigations of the changes in neural circuitry that underlie learning. To that end, some researchers have studied motor learning in monkeys. Many of these studies require the monkeys to move the arm under different types of perturbations (e.g., Li et al. (2001); Rokni et al. (2007); Mandelblat-Cerf et al. (2011)) or to reach in a specific overtrained sequence (e.g., Desmurget and Turner (2010); Picard et al. (2013)).

Despite the extent to which different types of learning have been studied, few studies have examined the *constraints* on learning. We know that constraints on learning must exist. Otherwise, for example, it would not take as long as it does to progress from learning simple arithmetic to learning calculus. The existing studies on the constraints of learning focus on language learning. For example, Saffran (2002) showed that learners can discover structure in linguistic input better when predictive relationships link elements within phrases. In another study, Newport (1990) suggested that there are cognitive elements related to maturation that constrain language learning. In this work, we focus on motor learning with BCIs. One open question regarding learning is what is the neural basis for the constraints? Are the neurons simply unable to fire in the necessary patterns required for learning? We address these questions in the work presented here.

## 1.2 NEUROPHYSIOLOGICAL HIERARCHY AND LEARNING

From a neurophysiological perspective, learning has been studied from a variety of hierarchical levels, ranging from the systems to the single cell and even to the level of the synapse. Below, I highlight several studies to provide an overview of what has been discovered at each level. I focus on changes in neural activity that have been observed at the cortical level, but changes have also been observed in deeper brain structures (e.g., Hikosaka et al. (1999); Brasted and Wise (2004); Yin et al. (2009)).

From a systems-level perspective, Nudo et al. (1996) analyzed the cortical representations in motor cortex as monkeys learned two different skills. One skill required the monkeys to make fine movements with their digits. The other skill required monkeys to repeatedly pronate and supinate their forearm. They found that for the first skill, the cortical representation of the digits expanded, while for the second skill, the cortical representation of the forearm expanded. This implies that the broad organization of motor cortex is plastic and use-dependent.

In agreement, Ungerleider et al. (2002) showed that the primary motor cortex (M1) in humans slowly reorganizes over the course of weeks when subjects learn to generate sequential finger movements. The reorganization of M1 followed changes that were more dynamic and swift in brain areas such as the cerebellum, striatum, and other motor-related cortical regions. Another study found that corticospinal excitability changes in M1 when a subject encounters a sudden change in environment rather than a gradual change, which suggests that repetition might be required for plasticity (Orban de Xivry et al., 2012). These human studies were conducted using functional magnetic resonance imaging (fMRI) and transcranial magnetic stimulation (TMS), respectively. fMRI detects changes in brain activity with low temporal and spatial resolution, and the effects of TMS are measured by recording motor evoked potentials in the muscles. Neither of these methods directly assess the activity of neurons within the brain.

Another way in which the neurophysiological components of learning have been addressed at a systems level is by studying patients who have certain neural diseases or disorders. For example, Role et al. (2000) and Smith and Shadmehr (2005) studied the ability of patients with Huntington's disease (HD) or cerebellar ataxia to make arm movements in an environment with altered dynamics. They found that HD patients adapted differently than cerebellar patients to different types of perturbations. From this, they concluded that *within* trial and *between* trial error correction mechanisms have different neural substrates.

The neural effects of learning have been studied on a single-neuron level. Li et al. (2001) trained animals to make reaching movements in the presence of an externally-applied force-field. They studied the firing properties of individual neurons in M1 relative to the direction of movement of the arm. They found a variety of categorical changes in the preferred directions of the neurons. Some neurons changed preferred direction when the force-field was introduced and reverted back to the original preferred direction when the force-field was removed. Other neurons only changed preferred direction when the force-field was removed, and another group of neurons changed preferred direction when the force-field was introduced but did not revert back when the force-field was removed. Subsequent studies have examined whether these changes in preferred direction are due to random fluctuations in M1 or whether they are task-induced. Rokni et al. (2007) showed that a component of the preferred direction changes were due to random drift of tuning properties in M1 and that another component was due to the force-field. In striking contrast, Stevenson et al. (2011) showed that, after accounting for noise, neural tuning curves are relatively stable when the dynamics of the environment are unchanged. This is in agreement with an earlier finding from Chestek et al. (2007).

Other research has examined the effects of long-term learning on individual neurons. Picard et al. (2013) found that following extended practice, there is a reduction in metabolic activity for internally-generated movements compared to visually-guided movements. Surprisingly, the single-neuron firing rates remained the same across tasks. This points to a change in synaptic efficacy as the neural instantiation of learning.

The effects of learning have been studied on an even smaller scale: the specific cellular processes that underlie changes in firing properties. The study of this topic dates back to the work of Kandel and Spencer (1968) and Bliss and Gardner-Medwin (1973). Here, I highlight some of the recent findings. In order for a neuron to respond differently to the same stimulus, plasticity must occur within the relevant network of neurons. In general, the plasticity of neurons follow Hebbian rules, which say that a synapse between two neurons will be strengthened as they fire together (Hebb, 1949). Hebbian plasticity is often studied as long-term potentiation (LTP) in a slice preparation. Buonomano and Merzenich (1998) were able to induce LTP in excitatory neurons in rat auditory cortex slices using a paired stimuluation protocol. Similar examples of LTP have been shown in excitatory neurons other cortical areas, such as somatosensory cortex (Crair and Malenka, 1995; Markram and Tsodyks, 1996), motor cortex (Rioult-Pedotti et al., 2000), and visual cortex

(Kirkwood and Bear, 1994; Frégnac et al., 1994) as well as in inhibitory neurons in rat visual cortex (Komatsu, 1994).

In addition to strengthening existing synapses, new synapses might form. A neuron might develop new dendritic spines which would allow it to receive input from other neurons, or the axons of other neurons might grow to make new synapses with the neuron of interest (Holtmaat and Svoboda, 2009). In both cases, the neuron would develop new firing properties in response to a stimulus. Studies in mice have shown that new dendritic spines are developed and maintained across days as the subjects learn a whisking task (Kuhlman et al., 2014). McKenzie et al. (2014) examined whether blocking the ability neurons to generate new connections would impair the ability of mice to learn a motor skill. Indeed, they found that the mice which could not form new connections between the neurons were unable to learn the task.

In summary, many brain areas and neural mechanisms are involved in skill learning and adaptation. Evidence suggests that the brain areas involved in learning are not localized to one specific region of the brain (such as the cortex, the cerebellum, or deeper structures), but the areas are widely distributed. Unifying theories have begun to understand how all hierarchical levels of changes related to learning are related (Buonomano and Merzenich, 1998). However, there is a substantial gap in the literature regarding the spatial scales at which learning has been studied. We know that neurons are interconnected in dense networks, but so far, most studies of learning have examined neural activity at the synapse or single-neuron level, or at the other extreme, the systems level. Few studies have examined the simultaneous activity of many neurons within a population and how that activity might change with learning. This might be the level from which behavior arises. In the work presented here, I will study learning from the perspective of populations of neurons within M1. It is my expectation and hope that the observations we made in M1 will apply more generally to other cortical areas that subserve other cognitive, sensory, and motor functions.

## 1.3 NEUROPHYSIOLOGY OF PRIMARY MOTOR CORTEX

The most general function of M1 is to execute movements. It is known that many neurons within M1 make direct synapses with neurons in the spinal cord (Dum and Strick, 2002; Rathelot and Strick, 2009). The spinal neurons then make direct connections to the muscles at the neuromuscular junctions. Thus, neurons in M1 provide the most direct link between the brain and the muscles. Incidentally, other brain areas, such as premotor cortex and supplementary motor area, also make direction connections with the spinal cord, but the greatest proportion of corticospinal connections originate in M1. In addition to projecting to the spinal cord, M1 also outputs to other brain areas, such as pons and lobules IV - VI of the cerebellum (Kelly and Strick, 2003).

The firing properties of M1 neurons have been extensively studied during overtrained behavior (i.e., after extensive learning). In many of these studies, the experimenters recorded the activity of individual neurons while monkeys moved their arms. The goals of these studies were to discover how M1 controlled the movements. The activity of neurons in M1 has been related to many different movement parameters. Some studies found that their discharge rates were directly related to the direction of movement (e.g., Georgopoulos et al. (1982)). This implies that the dynamics of the limbs themselves and the neurons in the spinal cord control other aspects of the reaches, such as muscle contraction, arm speed, arm acceleration, etc. Other studies of M1 found that the discharges rate of the neurons were related to endpoint (i.e., hand) acceleration (e.g., Flament and Hore (1988)). Yet other studies have found that neurons in M1 are related to joint kinematics such as angular velocity (Reina et al., 2001). Further studies have related neurons in M1 to dynamic properties of movement, such as EMG activity (Cherian et al., 2011) and joint torque (Kurtzer et al., 2005).

Other work has examined neural activity while subjects make the same hand movement with different wrist orientations, with the goal of determining whether neurons in motor cortex represent movement parameters such as muscle contraction or movement direction. Kakei et al. (1999) found that a substantial proportion of neurons encoded muscle contractions, while another proportion encoded movement direction. In a related study, Oby et al. (2013) found most motor neurons to encode muscle contractions. These results confirm each other in that both find a population of M1 neurons that encode muscle activity; the differences between them could be due to task design, precise sites of recording, and the analysis epoch.

Advances in our understanding of the function of M1 will continue to emerge from refined task designs and single-neuron recordings. In addition, some research has begun to examine the properties of populations of neurons within M1. The ability to record from many neurons simultaneously has been enabled by the development of multi-electrode recording arrays (Nordhausen et al., 1996). Recording from many neurons simultaneously has also presented challenges. Researchers have needed to develop new analysis techniques in order to take advantage of the simultaneous recordings. Simply performing conventional single-neuron analyses on that data would likely yield only limited new insights beyond what single-electrode recordings can already provide. New multineuronal analyses mainly involve dimensionality reduction, which attempts to extract meaningful signals from noisy high-dimensional data (Cunningham and Yu, 2014).

Churchland et al. (2012) proposed that instead of M1 neurons controlling specific kinematic (e.g., position, velocity, or acceleration) or kinetic (e.g., muscle force) parameters, the population of neurons together might control a more abstract dynamical process. They developed a novel dimensionality reduction technique ('jPCA') which captures rotational structure within the neural population activity. They found that this structure was present across a wide range of movement conditions and that it accounted for the variety of single-neuron responses that they observed. In another study, Kaufman et al. (2014) also used dimensionality reduction to study the activity of populations of motor cortex neurons. The goal of this study was to understand neural activity during movement preparation. They had observed that firing rates of individual neurons change while a monkey holds his arm still, and they wondered why neural activity was not constant while muscle activity was constant. They found that certain low-dimensional projections of the neural activity led to muscle contraction whereas other low-dimensional projections did not. Together, these two studies suggest that in order to study new insights into the neural underpinnings of learning, we should study the simultaneous activity of many neurons.

## 1.4 USING BRAIN-COMPUTER INTERFACES TO STUDY LEARNING

To analyze learning from the standpoint of populations of neurons, I employed the use of BCIs (Figure 1.1). BCIs are designed to translate neural activity into the action of an assistive device, such as a prosthetic limb or a computer cursor. Researchers have shown that monkeys can use BCIs to continuously control a computer cursor (Carmena et al., 2003; Jarosiewicz et al., 2008; Suminski et al., 2010; Gilja et al., 2012; Rouse et al., 2013; Sadtler et al., 2014), to make discrete selections similar to typing on a keyboard (Santhanam et al., 2006), and to control robotic limbs (Velliste et al., 2008). Recent work has shown that human subjects are also able to control computer cursors (Hochberg et al., 2006; Wang et al., 2013) and robotic limbs (Collinger et al., 2012; Hochberg et al., 2012).

In addition to providing quality of life benefits, BCIs provide a useful tool for studying the function of a healthy brain. Consider the movements of one's arm. We know that M1 is highly active during arm movements, as are many other such as premotor cortex (Kaufman et al., 2010), the posterior-parietal cortex (Della-Maggiore et al., 2004), the cerebellum (Shadmehr and Krakauer, 2008), and more. Many neurons, including those in the primary motor cortex, the premotor cortex, and the supplementary motor area, have been shown to project to the spinal cord where they synapse with motoneurons (Dum and Strick, 2002). It is currently not possible to record from all of the neurons that project to the spinal cord and cause the movement, nor is it currently possible to know the exact mapping from neural activity to arm movements.

In contrast, consider a BCI. Only the neurons that the experimenters directly record affect the effector's movement. The experimenters can specify the mapping from neural activity to cursor movement, making it possible to precisely study how changes in neural activity affect the effector's action. Thus, a BCIs provide a simplified motor control system, which is an ideal platform for studying sensory-motor learning.

Some existing studies have used BCIs to study learning, albeit from a single-neuron point of view. Jarosiewicz et al. (2008) studied the ways in which individual neurons changed their firing rates when the mapping between neural activity and cursor movements changed. They proposed three strategies that the monkeys might have employed to overcome the perturbation: aiming for a virtual target instead of the presented target, changing the tuning properties of individual neurons, or remapping the activity of the population. They found evidence to support all types of changes. In another study, Ganguly and Carmena (2009) tested the ability of a monkey to rapidly switch between using two different decoders. They demonstrated that a monkey was able to use a random decoder. Similarly, Hwang et al. (2013) trained monkeys to differentially modulate two neurons in order to select targets. They found that the combined activity across the two neurons was consistent with natural patterns those neurons typically displayed. Clancy et al. (2014) found that mice were even able to modulate individual neurons in order to control an auditory BCI.

Another example of using a BCI to study principles of motor control from a population perspective was conducted by Golub et al. (2013). Their study was based on the fact that as a subject moves a BCI cursor, the cursor often does not move directly to the target. They assumed that the subjects intended to move the cursor directly to the target, and using that assumption, they were able to extract a numerical representation of the subject's internal model (i.e., the monkey's idea of the mapping between neural activity and cursor movements). They showed that if the monkey had been controlling the cursor with its internal model, then it would have made fewer errors than he actually did. These studies together indicate that a new direction in understanding the neural basis of learning is emerging. My work follows this lead.

## 1.5 RESEARCH OBJECTIVES AND OUTLINE

The overall theme of the research presented in this dissertation is to study the neural mechanisms of learning with the underlying goal of improving BCI systems. In Chapter 3, I assess whether there are constraints within a population of neurons that prevent or limit learning. It has been hypothesized that BCI users might be able to learn any mapping between neural activity and effector movement and BCI systems would improved by using the arbitrary mappings since technicians would no longer need to spend time calibrating parameters. In Chapter 4, I begin to uncover the neural strategies that monkeys use when learning to con-

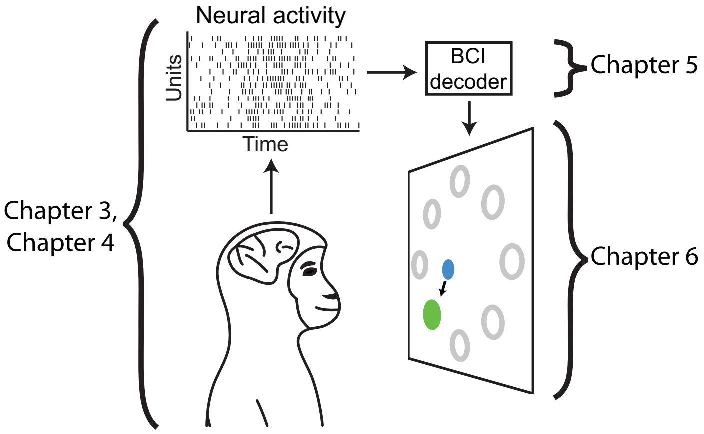

Figure 1.1: Diagram of a brain-computer interface. In a BCI, neural activity is recorded from the brain, and a BCI decoder translates neural activity into the movement of an effector (a cursor in this case). Chapters 3 and 4 focus on neural activity during BCI control. Chapter 5 focuses on the mapping from neural activity to cursor movements, and Chapter 6 focuses on the task the subject is completing.

trol a BCI. By understanding the way in which they adapt their neural activity patterns, we might be able to exploit the adaptation to improve BCIs. In Chapter 5, I assess whether we could design BCI decoders that would enable the subject to learn to control a cursor with more precision, and in Chapter 6, I investigate whether the task requirements will influence a subject to learn to show better BCI performance. Appendix B describes software tools I developed to enable my research, and Appedix C describes research projects on which I collaborated with the lead investigator. To begin, Chapter 2 describes experimental methods that are common to Chapters 3 - 6.

# 2.0 GENERAL METHODS

In this chapter, I will describe methods that are used throughout the experiments in Chapters 3 - 6. Section 2.1 describes the recording methods that were common to most experiments. For the BCI experiments, we used a decoder similar to the Kalman filter (Wu et al., 2006). Section 2.2 describes the Kalman filter and the way we derived its parameters. All animal handling procedures were approved by the University of Pittsburgh Institutional Animal Care and Use Committee.

## 2.1 ELECTROPHYSIOLOGY AND BEHAVIORAL DATA RECORDING

Data in Chapters 3 and 6 were recorded from two male Rhesus macaques (*Macaca mulatta*), monkey J and monkey L. Only monkey J was studied in Chapters 4 and 5. Both monkeys were implanted with a 96-channel microelectrode array in the proximal arm region of primary motor cortex (M1). Monkey J was implanted in the right hemisphere, and monkey L was implanted in the left hemisphere.

For all experiments, the monkeys sat in a custom primate chair. Visual stimuli were presented in a 3-D virtual reality (VR) environment. All stimuli were presented in the same fronto-parallel plane. For all tasks (details of which are included in the relevant chapters), the monkeys controlled the position of a cursor that appeared in the VR environment. The monkeys controlled the cursor either with neural activity (‘brain control’) or with the position of their hand contralateral to the implanted hemisphere (‘arm control’). We monitored the monkeys’ hand position using LEDs placed on the hand (Phasespace, Inc.). For brain control sessions, neural spike counts were directly mapped to cursor kinematics (either velocity or position, velocity, and acceleration). For hand control sessions, the position of the hand was directly mapped to cursor position.

## 2.2 BRAIN-COMPUTER INTERFACE DECODING

For all brain control sessions, we decoded cursor kinematics from neural activity using a linear mapping

$$\hat{\mathbf{x}}_t = \mathrm{M}_1\hat{\mathbf{x}}_{t-1} + \mathrm{M}_2\mathbf{u}_t + \mathrm{M}_0 \tag{2.1}$$

where $\hat{\mathbf{x}}_t \in \mathbb{R}^{r\times 1}$ is the cursor kinematics and $\mathbf{u}_t \in \mathbb{R}^{q\times 1}$ is a vector of z-scored activity for each neural unit in the time bin $t$ (45 ms or 60 ms). Each unit was z-scored separately. We defined one neural unit as corresponding to all of the threshold crossings recorded on one electrode. We set the threshold at 3.0 or 3.5 times the root-mean-square value for each electrode. For each session, the value was consistent across all electrodes. The number of units $q$ was typically around 90 each session. At the beginning of each session, we determined which electrodes to use for decoding. We did not use an electrode if its signal did not appear to be neural in origin or if it was electrically shorted to another electrode. $\mathrm{M}_1 \in \mathbb{R}^{r\times r}$ defines the contribution of the cursor kinematics at time $t-1$ to the kinematics at time $t$, and $\mathrm{M}_2 \in \mathbb{R}^{r\times q}$ defines the contribution of the neural activity recorded at time $t$ to the cursor kinematics at time $t$. $\mathrm{M}_0 \in \mathbb{R}^{r\times 1}$ accounts for constant offsets.

For some sessions, we derived the values of $\mathrm{M}_0$, $\mathrm{M}_1$, and $\mathrm{M}_2$ from the standard Kalman filter (Wu et al., 2006). For other sessions, we derived the values from a modified version of the Kalman filter (Sadtler et al., 2014). Details of the derivations are given in Section 2.2.1.

The number of kinematic parameters, $r$, was 6 for some sessions and 2 for other sessions. When $r = 6$, the elements of $\hat{\mathbf{x}}_t$ corresponded to horizontal and vertical position, velocity, and acceleration of the cursor. At each timestep, the elements of $\hat{\mathbf{x}}_t$ corresponding to position determined the cursor's position on the screen. When $r = 2$, the elements of $\hat{\mathbf{x}}_t$ corresponded to horizontal and vertical velocity, and we integrated the velocity across timesteps to define the cursor position on the screen.

### 2.2.1 Derivation of decoding parameters

I first describe the derivation of $\mathrm{M}_2$, $\mathrm{M}_1$, and $\mathrm{M}_0$ from the standard Kalman filter. Using the data from the calibration blocks (described in detail in each chapter), we fit the parameters of the Kalman filter using maximum-likelihood estimation (Wu et al., 2006). Then, we used those parameters to decode cursor kinematics from neural activity during closed-loop BCI control.

At each timestep, we first computed a one-step prediction $\hat{\mathbf{x}}_t^-$

$$\hat{\mathbf{x}}_t^- = A\hat{\mathbf{x}}_{t-1} + \mathbf{b} \tag{2.2}$$

where $A \in \mathbb{R}^{r\times r}$, and $\mathbf{b} \in \mathbb{R}^{r\times 1}$. Then, the one-step prediction was updated using the neural activity $\mathbf{u}_t$ to obtain the cursor kinematics $\hat{\mathbf{x}}_t$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + K(\mathbf{u}_t - C\hat{\mathbf{x}}_t^- - \mathbf{d}) \tag{2.3}$$

where $C \in \mathbb{R}^{q\times r}$, and $\mathbf{d} \in \mathbb{R}^{q\times 1}$. $K \in \mathbb{R}^{r\times q}$ is the Kalman gain (Wu et al., 2006) converged after many timesteps. Because the spike counts were z-scored, $\mathbf{d} = \mathbf{0}$. Combining equations 2.2 and 2.3 yields

$$\mathrm{M}_2 = K \tag{2.4}$$

$$\mathrm{M}_1 = A - KCA \tag{2.5}$$

$$\mathrm{M}_0 = b - KCb \tag{2.6}$$

Now I describe the modified Kalman filter used in some sessions (Sadtler et al., 2014). We first reduced the dimensionality of the neural activity from the calibration blocks to a set of low-dimensional latent factors using factor analysis (Everitt, 1984). At each timestep, we estimated the posterior mean of latent factors $\hat{\mathbf{z}}_t$.

$$\hat{\mathbf{z}}_t = \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\mathbf{u}_t \tag{2.7}$$

where $\mathbf{u}_t$ is defined above, and $\mathbf{z} \in \mathbb{R}^{p \times 1}$ contains the $p$ latent factors. Each latent factor can be understood as a well-modulated pattern of neural activity across the recorded units. The parameters $\Lambda \in \mathbb{R}^{q \times p}$ and $\Psi \in \mathbb{R}^{q \times q}$ were determined using the expectation-maximization algorithm (Dempster et al., 1977). Because $\mathbf{u}_t$ has a mean of $\mathbf{0}$ due to z-scoring, we do not need to explicitly subtract the mean in Equation 2.7. We related $\hat{\mathbf{z}}_t$ to the cursor kinematics to fit the Kalman filter parameters using the calibration data. This is in contrast to Equations 2.2 - 2.4, where the Kalman filter directly related the neural activity $\mathbf{u}_t$ to cursor kinematics. During closed-loop control, we used this modified Kalman filter which can be expressed in the form of Equation 2.1, where

$$\mathrm{M}_2 = K\Lambda^T(\Lambda\Lambda^T + \Psi)^{-1} \tag{2.8}$$

while $\mathrm{M}_1$ and $\mathrm{M}_0$ remain the same (but the dimensionalities of the matrices $K$ and $C$ are now based on $p$ rather than $q$). The specific decoding parameters (i.e., the kinematic elements in $\mathbf{x}_t$, whether standard or modified version of the Kalman filter was used, and the values of $p$ and $r$) are defined for each experiment in the relevant chapters.

# 3.0 NEURAL CONSTRAINTS ON LEARNING

*Figures and text in this chapter have been reproduced with permission from (Sadtler et al., 2014).*

In this chapter, I describe my research analyzing the neural constraints on learning. From a BCI standpoint, it is important to understand these constraints. BCI users (both in the clinic and in the lab) are not accustomed to using a BCI. The user will need to learn to use the BCI, and if we understand the constraints on learning, then we can design BCIs that avoid these constraints.

## 3.1 INTRODUCTION

All types of learning require the brain to generate new patterns of neural activity. We know that some skills and knowledge are harder to obtain than others, and we wondered whether this was a result of some neural activity patterns being more difficult to generate than others. We can visualize neural activity patterns in a high-dimensional space where each axis corresponds to the firing rate of one neuron (an example with three neurons is shown in Figure 3.1). Each point within this space indicates the firing rates of all neurons in one small time bin. When we record neural activity over a long period of time, we can analyze the extent to which the points fill the 'neural space'. Studies have shown that neural activity is not evenly distributed throughout the space (Mazor and Laurent, 2005; Luczak et al., 2009; Churchland et al., 2012; Mante et al., 2013; Rigotti et al., 2013; Cunningham and Yu, 2014). We term the space in which neural activity typically lies the 'intrinsic manifold'. The axes of the intrinsic manifold then define patterns of co-modulation – that is, the way in which neurons change their firing rates together.

In this study, we used a BCI to study the ability of monkeys to generate arbitrary patterns of neural activity. Using a BCI has advantages over arm control: 1) In a BCI, we record all of the neural activity that directly controls the effector (in this case, a cursor). However, in arm control, many more neurons than can be recorded contribute to the movement of limbs. 2) With a BCI, we can specify the mapping between neural activity and cursor movement. In this case, we specified a linear mapping between neural activity and cursor velocity. With arm control, the mapping from neural activity to muscle contraction is unknown. We utilized the ability to specify the mapping in order to request the animals to generate specific neural activity patterns.

BCIs can be visualized in the same neural space as the neural activity (Figure 3.1). Linear BCIs (which we use in this work) appear as a 'control space' in the neural space. Neural activity is mapped onto the control space in each time bin. The specific location where the neural activity maps onto the control space determines the kinematics (velocity in this case) of the effector. In the figure, the control space is shown as a line. In our actual experiments, the control space was a place representing the horizontal and vertical velocity of the cursor.

Other studies have shown that BCI learning can be extensive, which raises the intriguing possibility that any BCI mapping from neural activity to cursor movement can be learned (and hence, any neural activity pattern can be generated). In this study, we presented subjects with perturbed BCI mappings and observed whether or not they could learn to regain control of the cursor. Some of the perturbations reoriented the control space within the intrinsic manifold. We call these 'within-manifold perturbations'. Other perturbations reoriented the control plane so that it was outside of the intrinsic manifold. We call these 'outside-manifold perturbations'.

In order to regain control of the cursor under a within-manifold perturbation, the monkeys would need to combine their previously generated co-modulation patterns in new ways. In order to regain control of an outside-manifold perturbation, the monkeys would need to generate new co-modulation patterns among the neurons. We hypothesized that it would

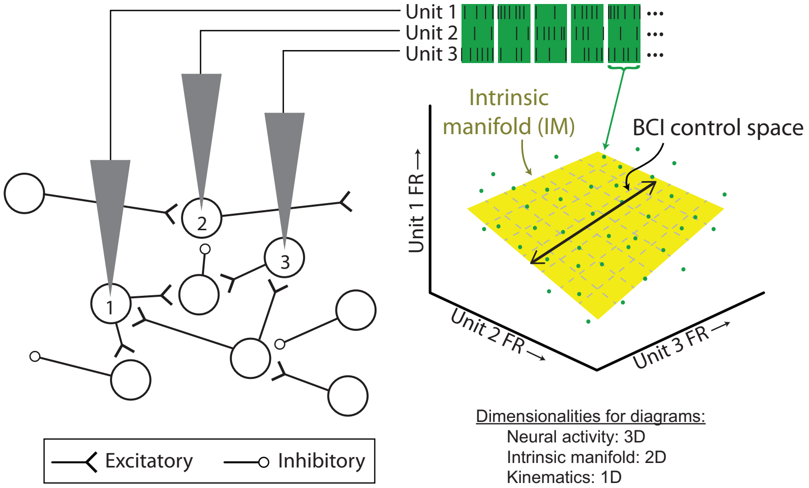

Figure 3.1: Conceptual illustration of neural space and intrinsic manifold. Spikes from three neurons within a larger network are recorded. The firing rate (FR) on each electrode in a short time epoch defines a point (green dots) in the neural space. The intrinsic manifold (yellow plane) captures the prominent patterns of co-modulation. Neural activity maps to the BCI control space (black line) to specify cursor velocity.

be easier for the monkeys to learn to control the cursor under a within-manifold perturbation than under an outside-manifold perturbation because the intrinsic manifold, which represents the underlying network, constrains the activity patterns that can be generated.

## 3.2 METHODS AND DATA ANALYSIS

For this study, both monkey J and monkey L were trained to move a cursor on a computer screen using a BCI. At the beginning of each day, we performed a calibration procedure during which we determined the parameters of an 'intuitive mapping'. The monkeys then used the intuitive mapping to control the cursor on a center-out task (baseline block). The intuitive mapping was easy for the monkeys to use with high-proficiency. After a few hundred trials with the intuitive mapping, we switched to a perturbed mapping. The monkeys controlled the cursor with perturbed mapping for several hundred trials (perturbation block) before we switched back to the intuitive mapping (washout block). In the following sections, I describe the center-out task (Section 3.2.1), how we identified the intrinsic manifold and intuitive mapping (Section 3.2.2), how we perturbed the mappings (Section 3.2.3), and how we analyzed the data (Section 3.2.4).

### 3.2.1 Center-out task

The same closed-loop BCI center-out task was used during the baseline, perturbation, and washout blocks (Figure 3.2). At the beginning of each trial, a cursor appeared in the center of the workspace, and a target appeared in the periphery. The target was chosen pseudo-randomly from eight targets evenly spaced around the center of the workspace. The cursor remained in the center for 300 ms. Following the 'freeze period', the velocity of the cursor was controlled by the monkey's neural activity through the BCI mapping. The monkey had 7.5 s to move the cursor to the peripheral target, which remained visible. If the monkey was successful, he received a water reward. If the monkey was unsuccessful, the trial was followed by a 1.5 s penalty during which no stimuli were presented to the monkey.

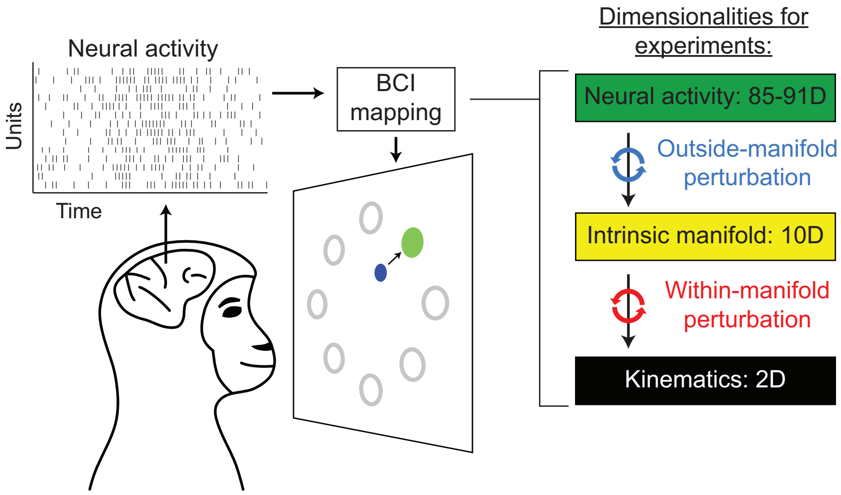

Figure 3.2: Closed-loop brain-computer interface control. Monkeys move the BCI cursor (blue circle) to acquire targets (green circle) by modulating their neural activity. The BCI mapping consisted of first mapping the population neural activity to the intrinsic manifold using factor analysis, then from the intrinsic manifold to cursor kinematics using a Kalman filter. This two-step procedure allowed us to perform outside-manifold perturbations (blue arrows) and within-manifold perturbations (red arrows). D, dimensions.

### 3.2.2 Intrinsic manifold and intuitive mapping

We estimated the intrinsic manifold and determined the parameters of the intuitive mappings from data recorded during the calibration block. The calibration procedures were slightly different for the two monkeys.

*Monkey J* For monkey J, we used two different methods – one per day – to calibrate the parameters of the intuitive mapping. For the first method, the monkey used the intuitive mapping from the previous day to control the cursor for 80 trials. This procedure relied on the neural signals being relatively stable on each electrode from day to day. This was the case on most days since the array had been implanted for 18 months prior to data collection.

We implemented the second calibration method because we were concerned that biases might be carried and exaggerated from one day to the next due to the reliance on the previous day's intuitive mapping. For this second method, the monkey observed a cursor automatically complete the center-out task for 80 trials. At the beginning of each 'observation' trial, the cursor appeared in the center of the workspace and remained there for 300 ms. Then the cursor moved with a constant velocity (0.15 m $s^{-1}$) to the peripheral target. For both methods for monkey J, we used neural activity from 300 ms after the start of each trial until the cursor acquired the peripheral target.

*Monkey L* We implemented a different calibration procedure for monkey L. Because the electrode array had been implanted more recently for monkey L than for monkey J, the neural signals were not as stable from day to day. Thus, we were unable to use monkey J's first method for monkey L. Additionally, the observation-based procedure was not as effective at generating an intuitive mapping the monkey could use well. Therefore, we developed a hybrid procedure combining both passive observation and closed-loop BCI control, similar to Velliste et al. (2008).

For the first 16 trials (two to each target), the monkey observed the cursor move to the target ('observation task'). We then fit the parameters of an intuitive mapping from the recorded neural activity and observed cursor velocity on those 16 trials. For the next 8 trials, the monkey used the newly calibrated mapping to control the cursor in closed-loop. On those trials, we restricted cursor movement so that it moved in a straight line towards (or away from) the target (i.e., thee perpendicular component of velocity was scaled by a factor of 0). We then calibrated another mapping from those trials. The monkey use that newly calibrated mapping to control the cursor with a perpendicular factor of 0.125. We repeated that procedure until the monkey was in full control of the cursor (i.e., a perpendicular factor of 1). We calibrated the intuitive mapping from 80 trials during which the monkey had full or partial control of the cursor.

We used the dimensionality reduction technique factor analysis to identify the intrinsic manifold from data recorded in the calibration block. See Section 2.2.1 for details on factor analysis. We used factor analysis to fit a 10-dimensional latent model of the data. We defined the intrinsic manifold as the column space of $\Lambda$ (Equation 2.7).

The intuitive mappings were modified Kalman filters, as described in Section 2.2.1. The factor analysis model was the same as the model used to identify the intrinsic manifold. Therefore, we can think of this intuitive mapping as first mapping neural activity onto the intrinsic manifold and then decoding cursor kinematics from the neural activity in the manifold. For this experiment, the kinematic states in the mapping included horizontal and vertical cursor velocity (i.e., $r = 2$).

### 3.2.3 Perturbed mappings

The perturbed mappings were modified versions of the intuitive mapping. Within-manifold perturbations altered the relationship between the low-dimensional factors and the kinematics of the cursor. As indicated in Figure 3.2 by the red arrows, the within-manifold perturbations were implemented by permuting the elements of $\hat{\mathbf{z}}_t$ before being passed to the Kalman filter. This perturbation did not alter the relationship between neural units and dimensions of the intrinsic manifold. Rather, it altered the relationship between the intrinsic manifold and the velocity of the cursor. Geometrically, this perturbation would appear as the red line in Figure 3.3. Note that the control space for a within-manifold perturbation lies within the intrinsic manifold but is different from the intuitive control space.

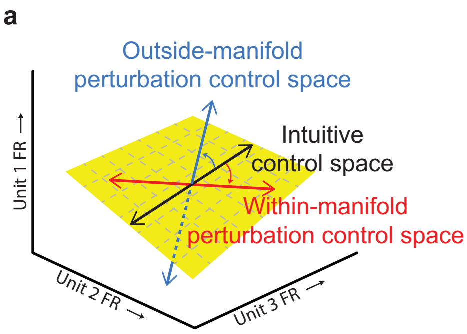

Figure 3.3: Control spaces for perturbed mappings in the neural space. (a) Within-manifold perturbation control space (red arrow) was rotated away from the intuitive mapping control space but remained within the intrinsic manifold. Outside-manifold perturbation control space (blue arrow) was rotated away from the intuitive mapping and off of the intrinsic manifold. (b) Neural activity (green dot) elicits different cursor velocities under different mappings. Arrow colors as in a.

The following equations describe within-manifold perturbations:

$$\hat{\mathbf{x}}_t = \mathrm{M}_1 \hat{\mathbf{x}}_{t-1} + \mathrm{M}_{2,\mathrm{WM}} \mathbf{u}_t \tag{3.1}$$

$$\mathrm{M}_{2,\mathrm{WM}} = K \eta_{WM} \Sigma_z \beta \tag{3.2}$$

where $\eta_{OM}$ is a $q \times q$ permutation matrix defining the outside-manifold perturbation (i.e., the outside-manifold perturbation matrix). In other words, $\eta_{WM} \Sigma_z \beta \mathbf{u}_t$ is a permuted version of $\Sigma_z \beta \mathbf{u}_t$.

Outside-manifold perturbations altered the relationship between the neural units and the dimensions of the intrinsic manifold. As shown in Figure 3.2 by the blue arrows, the outside-manifold perturbations were implemented by permuting the elements of $\hat{\mathbf{u}}_t$ before being passed to factor analysis. These perturbations required the monkeys to form new co-modulations among the neural units in order to be able to control the cursor proficiently.

Geometrically, these perturbations would appear as the blue line in Figure 3.3. This control space is outside of the intrinsic manifold.

The following equations describe outside-manifold perturbations:

$$\hat{\mathbf{x}}_t = \mathrm{M}_1 \hat{\mathbf{x}}_{t-1} + \mathrm{M}_{2,\mathrm{OM}} \mathbf{u}_t \tag{3.3}$$

$$\mathrm{M}_{2,\mathrm{OM}} = K \Sigma_z \beta \eta_{OM} \tag{3.4}$$

where $\eta_{OM}$ is a $q \times q$ permutation matrix defining the outside-manifold perturbation (i.e., the outside-manifold perturbation matrix). In other words, $\eta_{OM}\mathbf{u}_t$ is a permuted version of $\mathbf{u}_t$.

**3.2.3.1 Choosing a perturbed mapping** In each session, we used data from the first 200 trials (monkey J) or 150 trials (monkey L) to determine the perturbation matrix for that session. To do so, we used three steps: 1) We defined a set of candidate perturbations, 2) we predicted the open-loop cursor velocities for each candidate perturbation, and 3) we selected one candidate perturbation based on the open-loop velocities. The steps are detailed in the following paragraphs. We aimed to select a perturbation that would not be too difficult so that the monkey would not be able to control the cursor with it but would also not be too easy that the monkey's performance was not impacted by the perturbed mapping.

For monkey J, we often alternated perturbation types across consecutive days. For monkey L, we determined which type of perturbation we would use each day prior to the first experiment. That order was set randomly, and we did this in order to avoid a detectable pattern of perturbation types.

The following describes the first step in choosing a perturbed mapping: defining the candidate perturbations. For within-manifold perturbations, $\eta_{WM}$ is a $10 \times 10$ permutation matrix. The total number of possible $\eta_{WM}$ is 10 factorial (3,628,800). We considered all of these candidate within-manifold perturbations.

For outside-manifold perturbations, $\eta_{OM}$ is a $q \times q$ permutation matrix, where $q$ is the number of neural units. For a population of 90 neural units, there are 90 factorial ($> 10^{100}$)

possible values of $\eta_{OM}$. Due to computational constraints, we were unable to consider every possible $\eta_{OM}$ as a candidate perturbation. We used slightly different procedures to determine the candidate outside-manifold perturbations for the two monkeys.

The procedure we used for monkey J is as follows. We permuted the neural units independently. We chose to permute only the neural units with the largest modulation depths (mean number of units permuted: $39 \pm 18$). Permuting the units with larger modulation depths impacted the monkey's ability to proficiently control the cursor more than would permuting units with smaller modulation depths. For each session, we randomly chose 6 million $\eta_{OM}$ that permuted only the specified units. This formed the set of candidate outside-manifold perturbations.

The procedure we used for monkey L is as follows. To motivate it, note that for monkey J, the two perturbation types altered the intuitive mapping control space within a different number of dimensions of the neural space. Within-manifold perturbations were confined to 10 dimensions of the neural space, but outside-manifold perturbations were confined to $N$ dimensions of the neural space (where $N$ is the number of permuted units; 39 on average). Thus, the dimensionality of the search space for the perturbed mappings was larger for the outside-manifold perturbations than for the within-manifold perturbations. We recognized that this difference may have affected the monkey's ability to learn outside-manifold perturbations. For monkey L, we equalized the size of the search space for the two perturbation types. We did this by constraining $\eta_{OM}$ so that the number of possible $\eta_{OM}$ was equal to the number of candidate within-manifold perturbations. We then considered all $\eta_{OM}$ to be candidate outside-manifold perturbations. To construct outside-manifold perturbations, we assigned each neural unit to one of eleven groups. The first 10 groups had an equal number of neural units. The eleventh group had the remaining neural units. We specifically put the neural units with the lowest modulation depths in the eleventh group. The $10m$ (where $m$ is the number of neural units per group) neural units with the highest modulation depths were randomly assigned to the first 10 groups. We created outside-manifold perturbations by permuting the first 10 groups, keeping all the neural units within a group together. Thus, the number of possible $\eta_{OM}$ is 10 factorial, all of which were considered as candidate outside-manifold perturbations.

We attempted to keep these groupings as constant as possible across days. On some days, one electrode would become unusable (relative to the previous day) as evident from the threshold crossing waveforms. When this occurred, we kept all of groupings fixed that did not involve that electrode. If an electrode in one of the first ten groups became unusable, we would substitute it with a neural unit from the eleventh group.

The following describes the second step in choosing a perturbed mapping: estimating the open-loop velocities of each candidate perturbation. The open-loop velocity measurement captures how the neural activity updates the velocity of the cursor from the previous time step, whereas the closed-loop decoder (Equation 2.1) includes contributions from the decoded velocity at the previous time step ($\mathrm{M}_1\hat{\mathbf{x}}_{t-1}$) and from the neural activity at the current time step ($\mathrm{M}_2\mathbf{u}_t$). To compute the open-loop velocity, we first computed the average z-scored spike counts of every neural unit in the first 200 (monkey J) or 150 (monkey L) trials of the baseline block. We binned the spike counts from 300 ms to 1300 ms (monkey J) or 1100 ms (monkey L) after the beginning of each trial, and then averaged the spike counts for all trials to the same target. Together, these comprised 8 spike count vectors (one per target). For each of the spike count vectors, we computed the open-loop velocity for the candidate perturbations:

$$\mathbf{x}^{i}_{\mathrm{OL}} = \mathrm{M}_{2,\mathrm{P}}\mathbf{u}^{i}_{B} \tag{3.5}$$

where $\mathbf{u}^{i}_{B}$ is the mean z-scored spike count vector for the $i^{\mathrm{th}}$ target. $\mathrm{M}_{2,\mathrm{P}}$ is $\mathrm{M}_{2,\mathrm{WM}}$ for within-manifold perturbations and $\mathrm{M}_{2,\mathrm{OM}}$ for outside-manifold perturbations.

The following describes the third step in choosing a perturbation: selecting a candidate perturbation. For each candidate perturbation, we compared the open-loop velocities under the perturbed mapping to the open-loop velocities under the intuitive mapping on a per target basis. We needed the velocities to be dissimilar (to induce learning) but not so different that the animal could not control the cursor. We measured the angles between the 2D open-loop velocity vectors. We also measured the magnitude of the open-loop velocity for the perturbed mapping. For each session, we defined a range of angles (average minimum of range across sessions: $19.7° \pm 7.0°$; average maximum of range across sessions: $44.4° \pm 8.9°$) and a range of velocity magnitudes (average minimum of range across sessions: $0.7\,\mathrm{mm\ s^{-1}} \pm$

0.4 mm $s^{-1}$; average maximum of range across sessions: 5.5 mm $s^{-1}$ ± 4.0 mm $s^{-1}$). Note that when the monkey controlled the cursor in closed-loop (Equation 2.1), the cursor speeds were much greater than these ranges of open-loop velocities. This is because $\mathrm{M}_1$ was nearly an identity matrix for our experiments. Thus, the term $\mathrm{M}_1\mathbf{x}_{t-1}$ is expected to be larger than the term $\mathrm{M}_2\mathbf{u}_t$. We found all candidate perturbations for which the angles and magnitudes for all targets were within the designated ranges. From the candidate perturbations that remained after applying these criteria, we arbitrarily chose one to use as the perturbation for that session.

### 3.2.4 Data anlaysis

**3.2.4.1 Amount of learning** This section corresponds to Figure 3.5a. For each session, we computed the amount of learning during perturbation blocks as a single scalar value that incorporated both changes in success rate (percent of trials on which the peripheral target was acquired successfully) and target acquisition time. We sought to use a metric that captured how much the monkeys' performance improved throughout the perturbation block relative to how much it was impaired at the beginning of the perturbation block. Having a single value for each session allowed us to more easily compare learning across sessions and to relate the amount of learning to a variety of properties of each perturbation (Figure 3.8). We also analyzed each performance criterion individually for each monkey without any normalization, and we saw consistent differences in learnability. Thus, our results do not rely on the precise form of our learning metric, but the form provides a convenient summary metric.

Because success rate and target acquisition time are expressed in different units, we first normalized each metric. We found the mean and standard deviation of the success rates and target acquisition times across all non-overlapping 50-trial bins in the baseline, perturbation, and washout blocks for each monkey. We then z-scored the success rates and target acquisition times separately for each monkey. Figure 3.5a shows normalized performance projected onto veridical units.

For each session, we computed the average z-scored success rate and the average z-scored target acquisition time across all bins in the baseline block.

$$P_\mathrm{B} = \begin{bmatrix} s_\mathrm{B} \\ a_\mathrm{B} \end{bmatrix} \tag{3.6}$$

where $P_\mathrm{B}$ is the performance, $s_\mathrm{B}$ is the average normalized success rate, and $a_\mathrm{B}$ is the average normalized acquisition time during the baseline block (monkey J: $386.9 \pm 82.5$ trials; monkey L: $292.1 \pm 43.5$ trials).

We also computed the normalized success rates and acquisition times for all bins in the perturbation blocks.

$$P_\mathrm{p}(j) = \begin{bmatrix} s_\mathrm{p}(j) \\ a_\mathrm{p}(j) \end{bmatrix} \tag{3.7}$$

where $P_\mathrm{P}(j)$ is the performance, $s_\mathrm{p}(j)$ is the normalized success rate, and $a_\mathrm{p}(j)$ is the average normalized acquisition time during the $j^\mathrm{th}$ 50-trial bin of the perturbation block.

Empirically, we observed that the monkeys' performance during the perturbation blocks did not exceed the performance during the baseline blocks. Therefore, we define a maximum learning vector ($\overrightarrow{L}_\mathrm{max}$) as a vector that extends from the performance in the first bin with the perturbed mapping to the point corresponding to baseline performance (Figure 3.5a).

$$\overrightarrow{L}_\mathrm{max} = P_\mathrm{B} - P_\mathrm{P}(1) \tag{3.8}$$

The length of this vector is the initial performance impairment because it describes the drop in performance that resulted when we switched from the baseline block to the perturbation block. For each bin ($j$) within the perturbation blocks, we defined a raw learning vector ($\overrightarrow{L}_\mathrm{raw}(j)$). This vector extended from the point corresponding to initial performance during the perturbation block to the point corresponding to performance during each bin.

$$\overrightarrow{L}_\mathrm{raw}(j) = P_\mathrm{P}(j) - P_\mathrm{P}(1) \tag{3.9}$$

We projected the raw learning vectors onto the maximum learning vector. These were termed the projected learning vectors ($\overrightarrow{L}_\mathrm{proj}(j)$).

$$\overrightarrow{L}_{\text{proj}}(j) = \left( \overrightarrow{L}_{\text{raw}}(j) \cdot \frac{\overrightarrow{L}_{\text{max}}}{\left\| \overrightarrow{L}_{\text{max}} \right\|} \right) \left( \frac{\overrightarrow{L}_{\text{max}}}{\left\| \overrightarrow{L}_{\text{max}} \right\|} \right) \tag{3.10}$$

The lengths of the projected learning vectors relative to the lengths of the maximum learning vectors define the amount of learning in each 50-trial bin ($L_{\text{bin}}(j)$).

$$L_{\text{bin}}(j) = \frac{\left\| \overrightarrow{L}_{\text{proj}}(j) \right\|}{\left\| \overrightarrow{L}_{\text{max}} \right\|} \tag{3.11}$$

An amount of learning of 0 indicates that the monkey did not improve performance, and a value of 1 indicates that the monkey fully improved (up to the level during the baseline block). For each session, we computed the amount of learning for all bins, and we selected the largest one as the amount of learning for that session.

$$L_{\text{session}} = \max_{j} \left( L_{\text{bin}}(j) \right) \tag{3.12}$$

Figure 3.5a shows the raw learning vectors for one bin in each of two sessions (thick blue and red lines), along with the projected learning vector (thin red line) and the maximum learning vector (dashed gray line) for one of those sessions.

**3.2.4.2 Principal angles between intuitive and perturbed control spaces** This section corresponds to Figure 3.8b and Figure 3.10b. The control spaces for the intuitive and perturbed BCI mappings in our experiments were spanned by the rows of $M_2$ for the intuitive mapping, $M_{2,\text{WM}}$ for within-manifold perturbations, and $M_{2,\text{OM}}$ for outside-manifold perturbations. Because we z-scored spike counts in advance, the control spaces for each day intersected at the origin of the neural space. The two principal angles (Gilja et al., 2012) between the intuitive and perturbed control spaces defined the maximum and minimum angles of separation between the control spaces.

**3.2.4.3 Required preferred direction changes** This section corresponds to Figure 3.8c and Figure 3.10c. One way in which learning is manifested is by changes in preferred directions of individual neurons (Li et al., 2001; Jarosiewicz et al., 2008). For each session, we sought to compute the required changes in preferred direction for each neural unit that would lead to proficient control of the cursor under the perturbed mapping. One possibility would be to examine the columns of $M_2$ and $M_{2,P}$. Each column can be thought of as representing the pushing direction and pushing magnitude of one unit (i.e., the contribution of each neural unit to the velocity of the cursor). We could simply estimate the required change in preferred direction by measuring the change in pushing directions for each unit between the intuitive and perturbed mappings. However, this method is not suitable for the following reason. For outside-manifold perturbations for monkey J, we permuted only a subset of the neural units. As a result, the columns of $M_{2,OM}$ corresponding to the non-permuted units were the same as in $M_2$ By estimating the required changed in preferred direction as the difference in directional components of $M_2$ and $M_{2,OM}$, we would be implicitly assuming that the monkey is capable of identifying which units we perturbed and changing only their preferred directions, which appears to be difficult to achieve in the timeframe of a few hours (Jarosiewicz et al., 2008). Therefore, we sought a more biologically-plausible method of computing the required preferred direction changes.

Using a minimal set of assumptions, we computed the firing rates that each unit should show under one particular learning strategy. Then, we computed the preferred direction of each unit using those firing rates and compared them to the preferred directions during the baseline block. The following were the assumptions used to compute the firing rates:

1. We assumed the monkeys would intend to move the cursor to each target at the same velocity it exhibited under the intuitive mapping. Fitts' Law predicts that movement speed depends on movement amplitude and target size (Fitts, 1954).
2. The firing rates for the perturbed mapping should be as close as possible to the firing rates we recorded when the monkeys used the intuitive mapping. This keeps the predicted firing rates within a physiological range and implies a plausible exploration strategy in neural space.

We used the following procedure to compute the required preferred direction changes. First, we found the average normalized spike count vector $\mathbf{u}_B^i$ across timepoints (300 ms 1000 ms after the start of the trial) and all trials to each target ($i$) during the baseline blocks. We minimized the Euclidean distance between $\mathbf{u}_{\mathrm{B}}^{i}$ and $\mathbf{u}_{\mathrm{P}}^{i}$, the normalized spike count vector for the perturbed mapping (assumption 2), subject to $\mathrm{M}_2\mathbf{u}_{\mathrm{B}}^{i} = \mathrm{M}_{2,\mathrm{P}}u_{\mathrm{P}}^{i}$ (assumption 1). $\mathrm{M}_2\mathbf{u}_{\mathrm{B}}^{i}$ (the open-loop velocity for the intuitive mapping) is known from the baseline block.

For a given perturbed mapping (with $\mathrm{M}_{2,\mathrm{P}}$), we sought to find $\mathbf{u}_{\mathrm{P}}^{i}$ that would lead to the same open-loop velocity, which has a closed-form solution:

$$\mathbf{u}_{\mathrm{p}}^{i} = u_{\mathrm{B}}^{i} + \mathrm{M}_{2,\mathrm{P}}^{\mathrm{T}} \left( M_{2,\mathrm{P}}\mathrm{M}_{2,\mathrm{P}}^{\mathrm{T}} \right)^{-1} \left( \mathrm{M}_2 - \mathrm{M}_{2,\mathrm{P}} \right) \mathbf{u}_{\mathrm{B}}^{i} \tag{3.13}$$

For each neural unit $(k)$, we computed its preferred direction $\theta_{\mathrm{B}}(k)$ with the intuitive mapping by fitting a standard cosine tuning model.

$$\mathbf{u}_{\mathrm{B}}^{i}(k) = m_k \cdot \cos\left(\theta_i - \theta_{\mathrm{B}}(k)\right) + b_k \tag{3.14}$$

where $\mathbf{u}_{\mathrm{B}}^{i}(k)$ is the $k^{\mathrm{th}}$ element of $\mathbf{u}_{\mathrm{B}}^{i}$, $m_k$ is the depth of modulation, $b_k$ is the model offset of unit $k$, and $\theta_i$ is the direction of the $i^{th}$ target. We also computed the preferred direction of each unit for the perturbed mapping ($\theta_{\mathrm{P}}(k)$) in the same way. Figure 3.8 shows histograms of

$$|\theta_{\mathrm{P}}(k) - \theta_{\mathrm{B}}(k)| \tag{3.15}$$

averaged across all units for each session.

**3.2.4.4 Estimation of intrinsic dimensionality** This section accompanies Figure 3.11a. During all experiments, we identified a 10-dimensional IM (i.e., 10 factors). Offline, we confirmed this was a reasonable choice by estimating the intrinsic dimensionality of the data recorded in each calibration block. For each day, we performed a standard model selection procedure to compare FA models with dimensionalities ranging from 2 to 30. For each candidate dimensionality, we used 4-fold cross-validation. For each fold, we estimated the FA model parameters using 75% of the calibration data. We then computed the likelihood of the remaining 25% of the calibration data with the FA model. For each dimensionality, we averaged the likelihoods across all folds. Each day's intrinsic dimensionality was defined as the dimensionality corresponding to the largest cross-validated data likelihood of the calibration data for that day.

**3.2.4.5 Measuring the cumulative shared variance explained** This section corresponds to Figure 3.11d. Factor analysis partitions the sample covariance of the population activity (cov($\mathbf{u}$)) into a shared component ($\Lambda\Lambda^{\mathrm{T}}$) and an independent component ($\psi$). In offline analyses, we sought to characterize the amount of shared variance along orthogonal directions within the intrinsic manifold (akin to measuring the lengths of the major and minor axes of an ellipse). These shared variance values are given by the eigenvalues of $\Lambda\Lambda^{\mathrm{T}}$, which can be ordered from largest to smallest. Each eigenvalue corresponds to an 'orthonormalized latent dimension', which refers to identifying orthonormal axes that span the intrinsic manifold. Each orthonormalized dimension is a linear combination of the original 10 dimensions. The cumulative shared variance curve is thus informative of how 'oblong' the shared variance is within the manifold, and it can be compared across days. By definition, the cumulative shared variance explained reaches 100% using all 10 dimensions, and none of the independent variance ($\psi$) is explained by those latent dimensions.

## 3.3 RESULTS

We trained two monkeys to control a cursor using a BCI on a center-out task. In each session, the monkeys first controlled the cursor with the intuitive mapping, which was easy for the monkey to use. Then, we perturbed the mapping, and the monkey attempted to learn to use this perturbed decoder over the next 400 - 600 trials. Afterwards, we switched back to the intuitive mapping.

### 3.3.1 Success rate, acquisition time, and learning

Figure 3.4 shows the performance from two representative sessions – one of each perturbation type. Our performance metrics were success rate (defined as the percent of trials for which

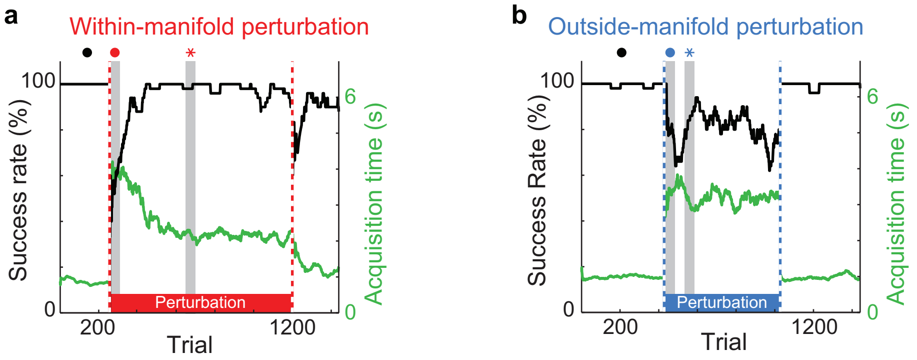

Figure 3.4: Performance on representative sessions. Black trace, success rate; green trace, target acquisition time. Dashed vertical lines indicate when the BCI mapping changed. Gray vertical bands represent the 50-trial bins used to determine initial (red and blue dots) and best (red and blue asterisks) performance with the perturbed mapping. (a) Within-manifold perturbation, monkey J. (b) Outside-manifold perturbation, monkey L.

the monkey successfully acquired the peripheral targets) and acquisition time (defined as the amount of time required for the monkey to move the cursor to the target). Figure 3.4a shows a within-manifold perturbation session. The success rate (black line) is high, and the acquisition time (green line) is low during the baseline block (left of the first dashed line). When the perturbed mapping is introduced (just after the first dashed line), the success rate drops, and the acquisition time increases. Over time, the monkey was able to learn to control the cursor proficiently, as indicated by the increase in success rate and decrease in acquisition time. At the second dashed line, we reintroduced the intuitive mapping for the washout block. Figure 3.4b shows an outside-manifold perturbation session. At the beginning of the perturbation block, performance decreased, but the monkey did not improve performance with time. The lack of improvement indicates that the monkey did not learn to control the cursor proficiently.

We desired a compact way to measure the amount of learning on each session. To that end, we developed a vectorial approach which incorporates changes in success rate and acquisition time in a single number for each session. Figure 3.5 shows histograms of

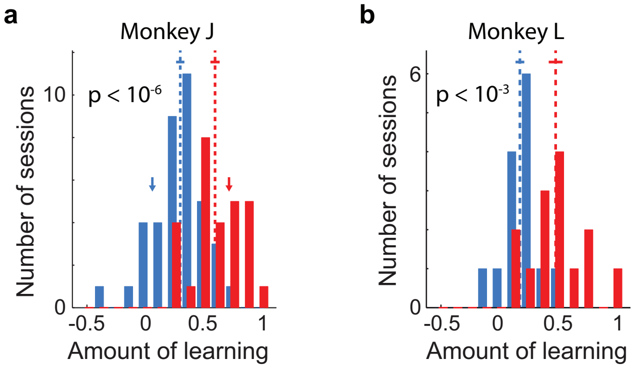

Figure 3.5: Greater learning for within-manifold perturbations than outside-manifold perturbations. Learning is significantly greater for within-manifold perturbations (red, n = 28 (monkey J, panel a), 14 (monkey J, panel b)) than for outside-manifold perturbations (blue, n = 39 (monkey J), 15 (monkeyL)). Arrows indicate the sessions shown in Figure 3.4. Dashed lines: means of distributions. Solid lines: mean ± SEM. p-values: two-sided t-tests.

the amounts of learning for monkey J (panel a) and monkey L (panel b). An amount of learning of 1 indicates complete learning of the relationship between neural activity and kinematics, and a value of 0 indicates no learning. Both monkeys demonstrated significantly more learning for within-manifold perturbations than for outside-manifold perturbations. This indicates that the monkeys could more easily combine existing co-modulation patterns in new ways than they could generate new co-modulation patterns among the neurons.

Two additional lines of evidence show that within-manifold perturbations were more learnable than outside-manifold perturbations. First, perturbation types differed in their aftereffects (Figure 3.6). After a lengthy exposure to the perturbed mapping, we again presented the intuitive mapping (following the second dashed vertical line in Figure 3.4). We measured the aftereffect as the size of the performance impairment at the beginning of the washout block in the same way that we measured the performance impairment at the beginning of the perturbation block. A larger aftereffect indicates more learning had occurred in response to the perturbation (Shadmehr et al., 2010). Following within-manifold perturbations, performance was impaired briefly, indicating that learning had occurred. Following outside-manifold perturbations, performance was not impaired, which is consistent with little, if any, learning.

Second, the difference in learnability between the two types of perturbation was present from the earliest sessions, and over the course of the study the monkeys did not improve at learning (Figure 3.7). It might have been that, over the course of weeks and months, the animals improved at learning to use perturbed mappings, either one type or both types together. This did not occur. Within-manifold perturbations showed more learning than outside-manifold perturbations across the duration of experiments. Animals did not get better at learning to use either type of perturbation separately.

### 3.3.2 Alternative explanations of learning

Taken together, the amount of learning and after-effect show that the intrinsic manifold was a reliable predictor of the learnability of a BCI mapping: new BCI mappings that were within the intrinsic manifold were more learnable than those outside of it. We considered five alternative explanations for the difference in learnability. First, we considered whether mappings that were more difficult to use at first might be more difficult to learn. We ensured that the initial performance impairments were equivalent for the two perturbation types (Figure 3.8a).

Second, we posited that the animals must search through neural space for the new control space following the perturbation. If the control spaces for one type of perturbation tended to be farther from the intuitive control space, then they might be harder to find, and thus, learning would be reduced. We ensured that the angles between the intuitive and perturbed control spaces did not differ between the two perturbation types (Figure 3.8b). Incidentally, Figure 3.8b also shows that the perturbations were not pure workspace rotations because in that case, the angles between control spaces would have been zero.

Third, we considered how much of an impact the perturbations exerted on the tuning of neural units. Learning is manifested (at least in part) as changes in the preferred direction (PD) of individual neurons (Li et al., 2001; Jarosiewicz et al., 2008). If learning one type of perturbation required larger changes in PDs, then those perturbations might be harder to

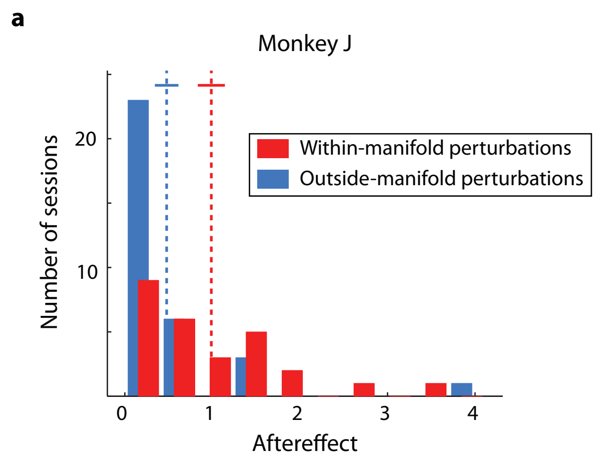

Figure 3.6: Aftereffect during washout blocks. For monkey J (left), the aftereffect was significantly larger for within-manifold perturbations (red) than for outside-manifold perturbations (blue) (Wilcoxon rank-sum test, p ¡ 10-3). For monkey L (right), the trend is in the same direction as monkey J, but the effect did not achieve significance (Wilcoxon rank-sum test, $p > 0.05$). Number of within-manifold perturbations: $n = 27$ (monkey J), 14 (monkey L); outside-manifold perturbations: $n = 33$ (monkey J), 15 (monkey L).

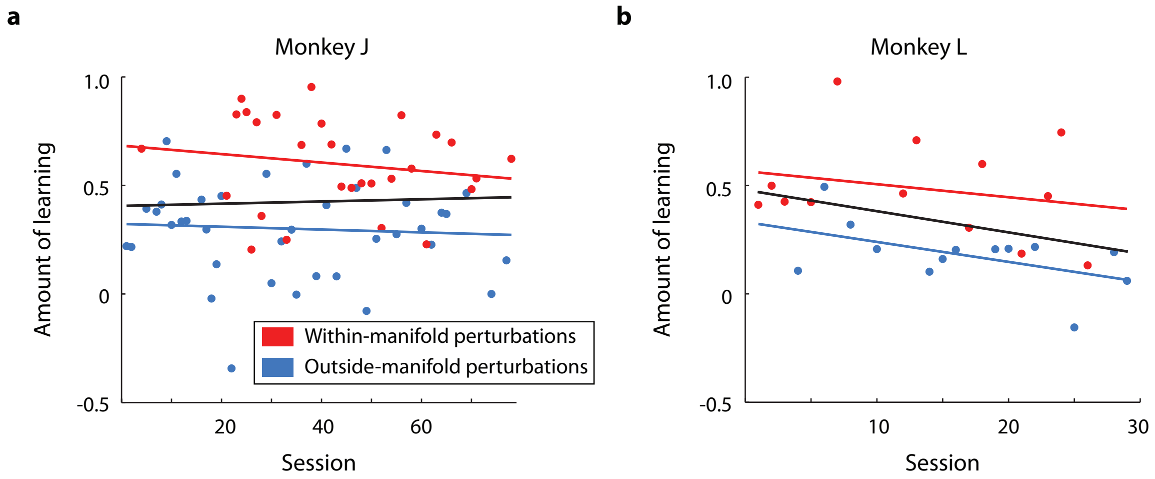

Figure 3.7: Learning across sessions. Animals (monkey J, panel a; monkey L, panel b) did not improve at learning either type of perturbation across sessions. Slopes of regression lines (black) are not significantly different from 0 (F-text, $p > 0.05$).

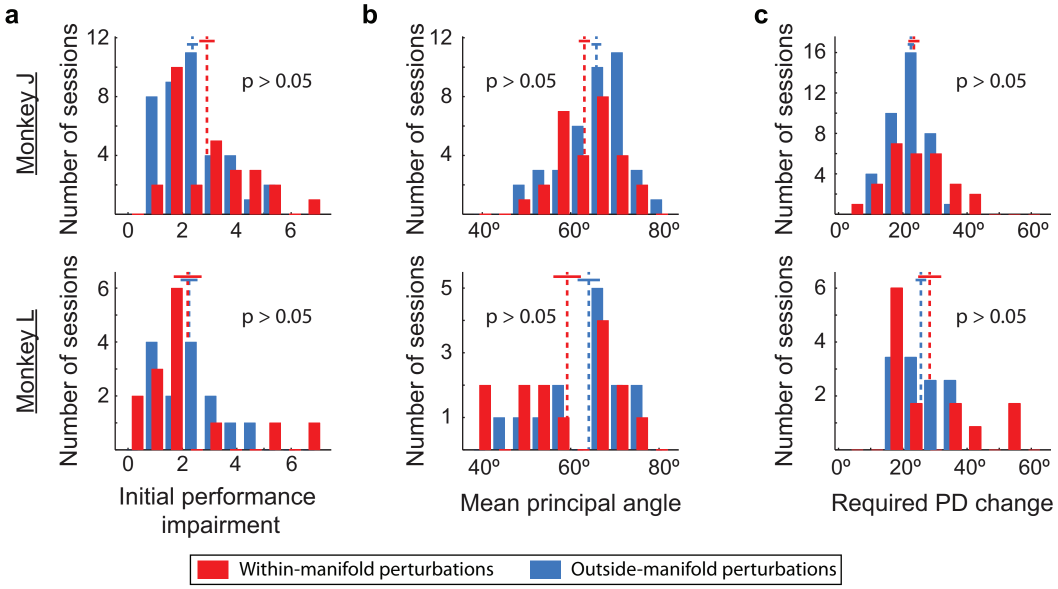

Figure 3.8: Alternative explanations do not explain difference in learnability between perturbation types. (a) Performance impairment immediately following within-manifold perturbations (red) and outside-manifold perturbations (blue). Dashed lines: means of distributions. Solid lines: mean ± SEM. (b) Mean principal angles between intuitive and perturbed mappings. (c) Mean required change in preferred direction (PD) for individual neural units. All panels: p-values are for two-sided t-tests; same number of sessions as in Figure 3.5.

learn. We predicted the changes in PDs that would be required to learn each perturbation while minimizing changes in firing rates. We ensured that learning the two perturbation types required comparable PD changes (Figure 3.8c).

Fourth, for one monkey (L), we ensured that the sizes of the search spaces for finding a strategy to proficiently control the cursor were the same for both perturbation types (see Section 3.2.3.1).

Fifth, we considered how much the monkeys moved their hands while controlling the BCI cursor and if this impacted learning. We loosely restrained the monkey's arms and found that the monkeys only minimally moved their hands, well below the limit of the restraints. Furthermore, hand movements were comparable and nearly non-existent for both perturbation types (Figure 3.9). The hand speeds were much smaller than typical hand speeds on an arm control task. We conclude from these analyses that the parsimonious explanation for BCI learning is whether or not the new control space is within the IM.

These alternative explanations did reveal interesting secondary aspects of the data: the explanations partially explained within-category differences in learnability, albeit in an idiosyncratic manner between the two monkeys. Figure 3.8 showed that the properties of the perturbed mappings (other than whether their control spaces were within or outside the IM) could not account for differences in learning between the two types of perturbation. However, as is evident in Figure 3.5, within each type of perturbation, there was a range in the amount of learning, including some outside-manifold perturbations that were learnable (Jarosiewicz et al., 2008; Ganguly and Carmena, 2009). In Figure 3.10, we examined whether learning within each perturbation type could be accounted for by considering other properties of the perturbed mapping. We regressed the amount of learning within each perturbation type against the various properties we considered in Figure 3.8. Figure 3.10a shows the initial performance impairment could explain a portion of the variability of learning within both classes of perturbation for monkey J, who showed more learning on sessions when the initial performance impairment was larger. For monkey L, the initial performance impairment could account for a portion of the within-class variation in learning only for outside-manifold perturbations; this monkey showed less learning when the initial performance impairment was larger. We speculate that monkey J was motivated by more difficult perturbations while

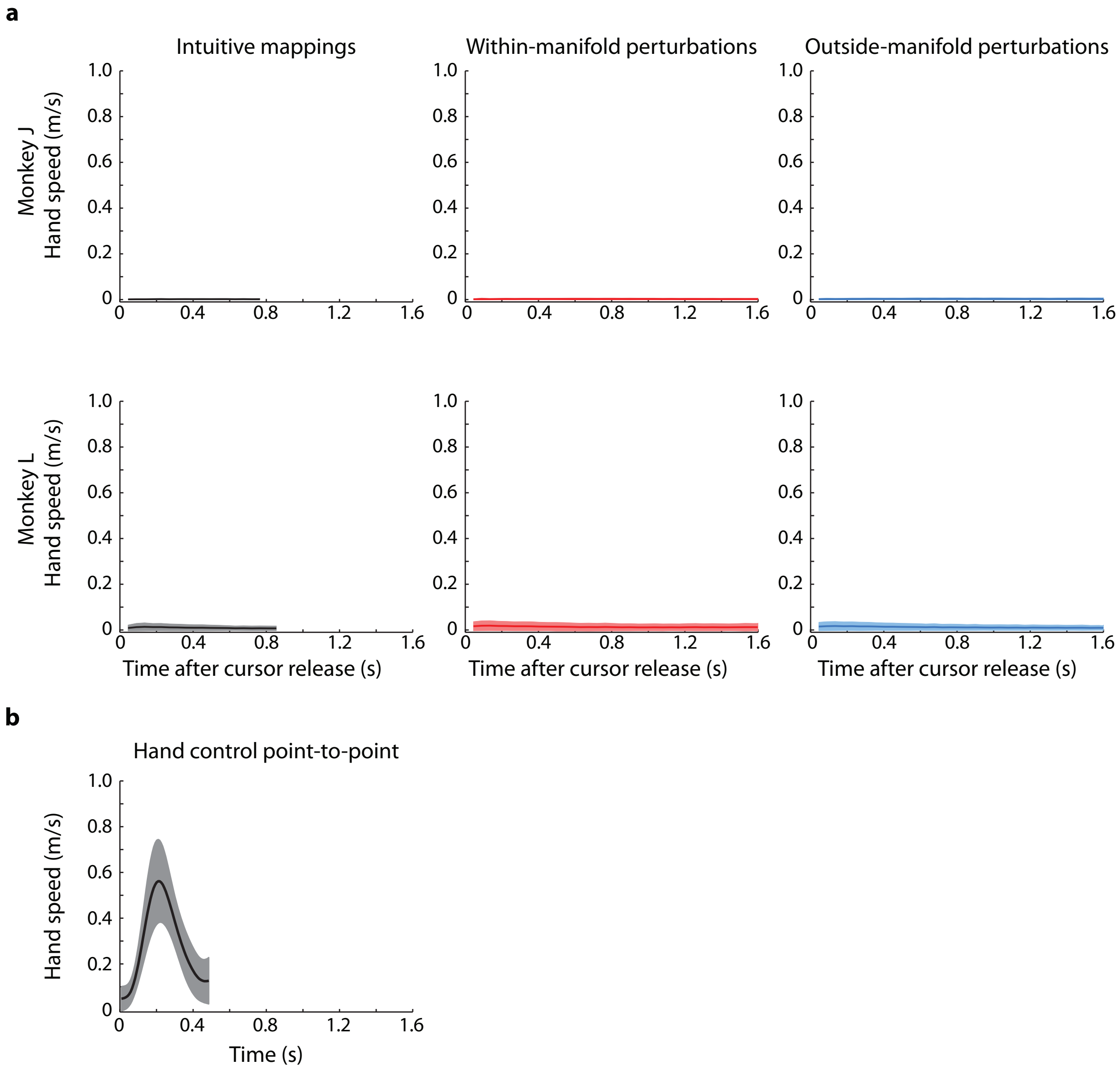

Figure 3.9: Hand movements during brain-computer interface control. (a) Average hand speeds across all trials in all sessions for the baseline blocks (left column), within-manifold perturbation blocks (middle column), and outside-manifold perturbation blocks (right column) for monkey J (top row) and monkey L (bottom row). (b) Average hand speed during a typical point-to-point reaching task (monkey L). Thus, the hand movements for the BCI tasks are substantially smaller than for the reaching task.

monkey L could be frustrated by more difficult perturbations. Figure 3.10b shows that the mean principal angles between control planes were related to learning within each class of perturbation for monkey L only. Larger mean principal angles between the control planes led to less learning. Figure 3.10c shows that the required PD changes were not related to learning for either type of perturbation for both monkeys. This makes the important point that we were unable to account for the amount of learning by studying each neural unit individually.

### 3.3.3 Properties of the intrinsic manifold

A key step in these experiments is the identification of an IM using dimensionality reduction (Cunningham and Yu, 2014). Although our estimate of the IM can depend on several methodological factors (such as the richness of the behavioral task, the size of the training set, the number of neurons, the dimensionality reduction method, and the criterion for assessing dimensionality), the critical property of an IM is that it captures the prominent patterns of co-modulation among the recorded neurons, which presumably reflect underlying network constraints. For consistency, we estimated a linear, 10-dimensional IM each day. *Post hoc*, we considered whether our choice of 10 dimensions had been appropriate. We estimated the intrinsic dimensionality of the neural activity for each day (Figure 3.11a). The average dimensionality was about 10 for both monkeys (Figure 3.11a), and this was true for each monkey as well (Figure 3.12a). Even though the estimated dimensionalities ranged from 4 to 16, the selection of 10 dimensions still provided a model that was nearly as good as the best model (Figure 3.11b-c). Because the top few dimensions captured the majority of the co-modulation among the neural units (Figure 3.11d), we likely could have selected a different dimensionality within the range of near-optimal dimensionalities and still attained similar results. We note that we cannot make claims about the 'true' dimensionality of M1 in part because it likely depends on considerations such as the behaviors the animal is performing and perhaps its level of skill.

The key to the success of this experiment was capturing the prominent patterns by which the neural units covary. As shown in Figure 3.11d, the top several dimensions capture the

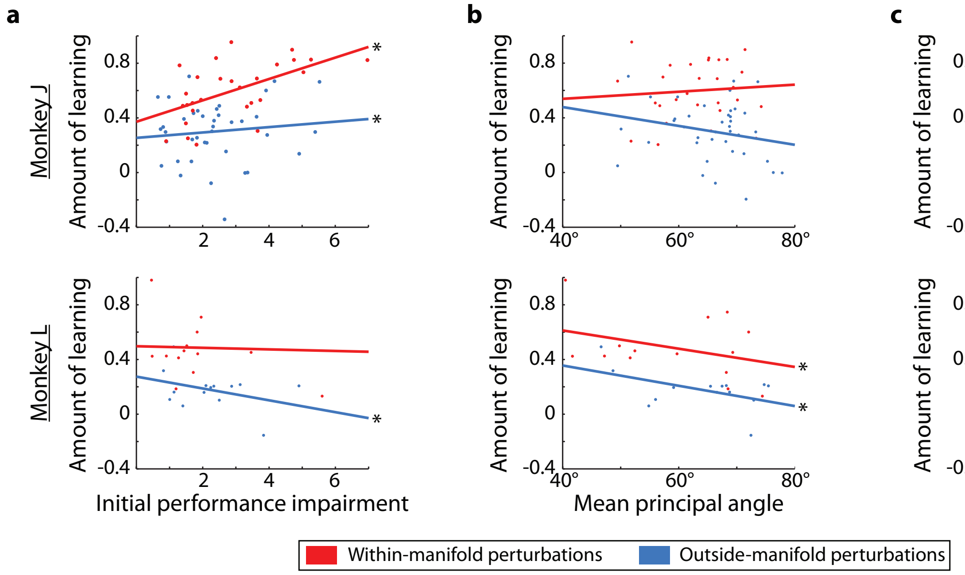

Figure 3.10: Accounting for within-class differences in learning. (a) Relation between amount of learning and initial impairment in performance for monkey J (top) and monkey L (bottom). Each point corresponds to one session. Lines are linear regressions for the within-manifold perturbations and outside-manifold perturbations. *: slope significantly different than 0 (F-test, $p < 0.05$). (b) Relation between amount of learning and mean principal angles between control spaces for perturbed and intuitive mappings. (c) Relation between amount of learning and mean required PD change. Same number of sessions as in Figure 3.5.

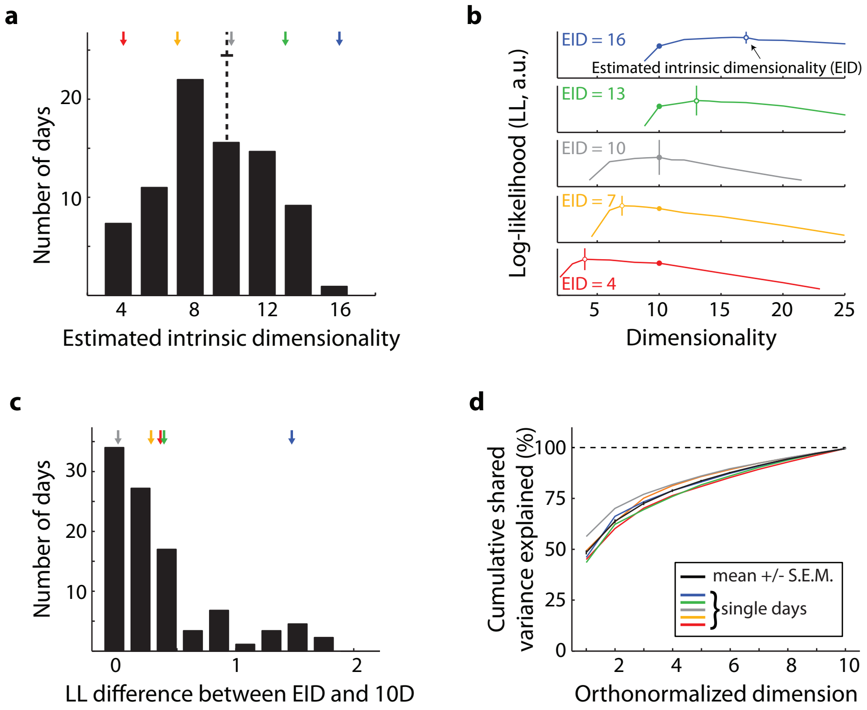

Figure 3.11: Properties of the intrinsic manifold. (a) EID across all days and both monkeys (mean ± SEM: 9.81 ± 0.31). (b) Cross-validated log-likelihoods (LL) of the population activity for different days. The peaks (open circles) indicate the estimated intrinsic dimensionality (EID). Vertical bars indicate the standard error of LL, computed across 4 cross-validation folds. We always used a 10-dimensional IM for the experiments (closed circles). (c) Difference between the LL for the 10-dimensional model and the EID model. Units are the number of standard errors of LL for the EID model. For 89% (78/88) of the days, the LL for the 10-dimensional model was within 1 standard error of the EID model. All sessions were less than 2 standard errors away. (d) Cumulative shared variance explained by the 10-dimensional IM used during the experiment. Colored curves correspond to the experimental days shown in panel a. The black curve shows the mean ± SEM across all days (n = 88; monkey J: 58, monkey L: 30).

majority of the shared variance. Thus, we believe that our main results are robust to the precise number of dimensions used during the experiment. Namely, the effects would have been similar as long as we had identified at least a small handful of dimensions. Given the relative simplicity of the BCI and observation tasks, our estimated intrinsic dimensionality is likely an underestimate. For example, a richer task may have revealed a larger set of co-modulation patterns that the circuit is capable of expressing, and Figure 3.12b shows that the estimated intrinsic dimensionality increases as the number of data points used in the computation increases. Even so, our results suggest that the IM estimated in the present study already captures some of the key constraints imposed by the underlying neural circuitry. The likely underestimate of intrinsic dimensionality may explain why a few ostensible outside-manifold perturbations were readily learnable (cf. Figure 3.5).

It is worth noting that improperly estimating the intrinsic dimensionality would only have weakened the main result. If we had overestimated the dimensionality, then some of the ostensible within-manifold perturbations would actually have been outside-manifold perturbations. In this case, the amount of learning would tend to be erroneously low for nominal within-manifold perturbations. If we had underestimated the dimensionality, then some of the ostensible outside-manifold perturbations would actually have been within-manifold perturbations. In this case, the amount of learning would tend to be erroneously high for outside-manifold perturbations. Both types of estimation error would have decreased the measured difference in the amount of learning between within-manifold perturbation and outside-manifold perturbations.

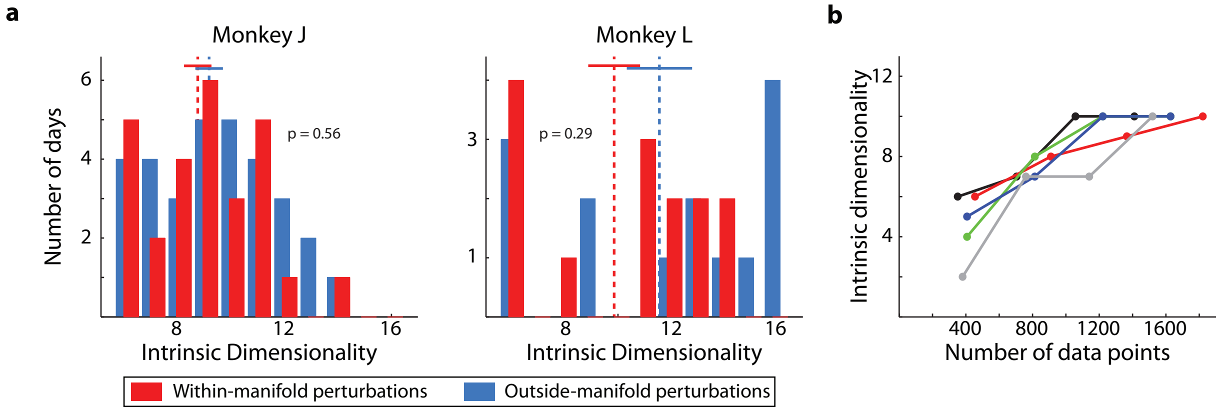

Figure 3.12: Offline analyses of intrinsic manifold properties. (a) The intrinsic dimensionalities for all sessions for monkey J (left) and monkey L (right). For both monkeys, the intrinsic dimensionalities were not significantly different between days when we performed within-manifold perturbations and days when we performed outside-manifold perturbations (t-test, $p > 0.05$). Dashed lines: means of distributions. Solid lines: mean ± SEM. (b) Relation between intrinsic dimensionality and the number of data points used to compute intrinsic dimensionality. For each of 5 days (one curve per day), we computed the intrinsic dimensionality using 25%, 50%, 75%, and 100% of the total number of data points recorded during the calibration block. As the number of data points increased, our estimate of the intrinsic dimensionality increased in a saturating manner.

It is also important to note that we perturbed the raw factors as opposed to the orthonormalized factors. The othonormalized factors are ordered such that the first factor accounts for the most shared variance (Figure 3.11d), but the raw factors are not ordered in any manner. Figure 3.13 highlights this difference. The raw factors have approximately the same depths of modulation, and the directions of the 'preferred' and 'anti-preferred' axes are seemingly arbitrary across dimensions. On the other hand, the modulation depths of the orthonormalized factors decreases from the first dimension to the last dimension, and the preferred axes of the first two dimensions are often orthogonal.

### 3.3.4 Per-target learning

The analyses described so far have considered learning for all targets in a session. We next examined whether the mechanisms of learning affect each target differently. We found that we could predict the extent to which BCI control to individual targets is impacted by the perturbation. For each target, there exists a unique point on the control space that would move the cursor straight to it at a given speed. We measured the angle in neural space between each targets point on the control spaces for the intuitive and perturbed mappings. That is, for each target $i$, we computed the angle between $\mathbf{u}_{\mathrm{B}}^{i}$ $\mathbf{u}_{\mathrm{P}}^{i}$. These angles were good predictors of the target-by-target performance decrement induced by the perturbations (Figure 3.14a-b, left column). Thus, our geometrical approach has sufficient resolving power to provide local predictions of performance. After exposure to within-manifold perturbations, performance improved more for the targets for which the angle between control spaces was large than for targets for which the angle between control spaces was small (Figure 3.14a, right column). Consistent with the results shown in Fig. 2, performance to individual targets improved very little under outside-manifold perturbations (Figure 3.14b, right column). Thus, when learning did occur, it acted preferentially on the aspects of behavior that were most impacted by the perturbation.

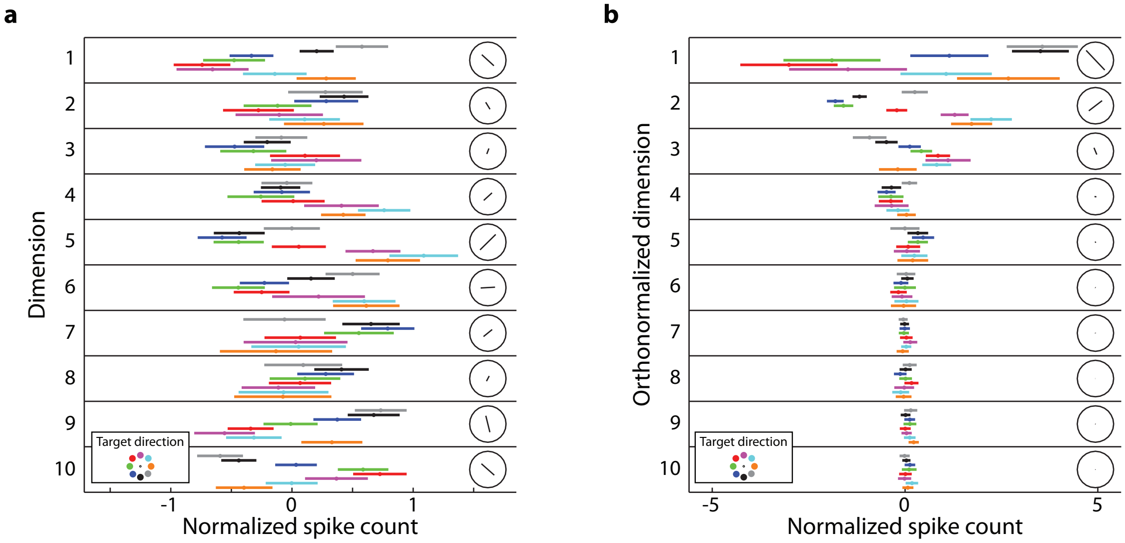

Figure 3.13: Additional offline analyses of intrinsic manifold properties. (a) Tuning of the raw factors. These plots exhibit the factors that were shuffled during within-manifold perturbations. We show for one typical day the average factors $\hat{\mathbf{z}}$ corresponding to the 10 dimensions of the IM over a time interval of 700 ms beginning 300 ms after the start of every trial. Within each row, the colored bars indicate the mean ± standard deviation of the factors for each target. The line in each circular inset indicates the axis of 'preferred' and 'anti-preferred' directions of the factor. The length of the axis indicates the relative depth of modulation. The tuning is along an axis (rather than in a single direction) because the sign of a given factor is arbitrary. (b) Tuning of the orthonormalized factors. Same session and plotting format as a. The orthonormalized dimensions are ordered by the amount of shared variance explained, which can be seen by the variance of the factors across all targets. Note that the axes of greatest variation are separated by approximately 90° for orthonormalized dimensions 1 and 2. This property was typical across days.

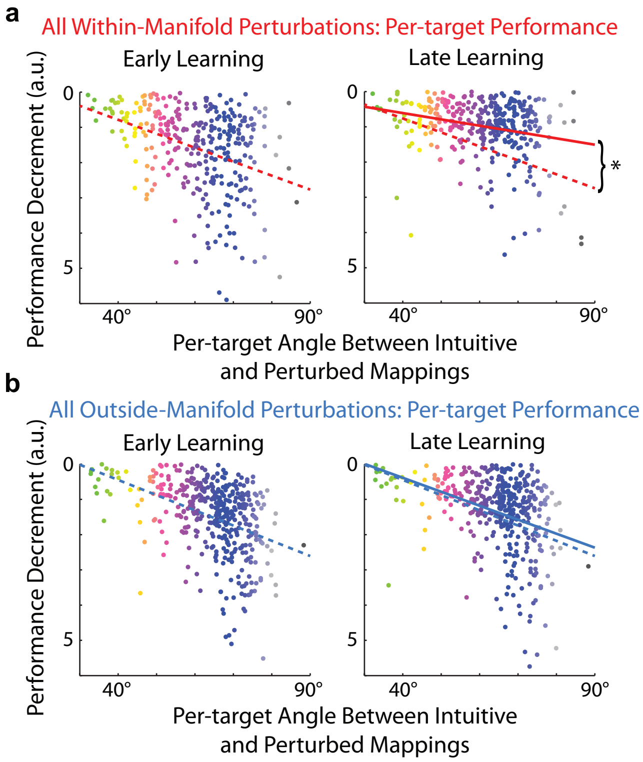

Figure 3.14: Per-target learning strategies. (a) Within-manifold perturbations across all targets and sessions. The horizontal axis is the angle between the intuitive and perturbed control spaces for each target. The vertical axis is the performance with the perturbed mappings relative to performance with the intuitive mapping. Left panel: each point is average performance for the first 5 trials to each target after the perturbation was introduced. Dashed line: linear regression. Right panel: each point is average performance for the last 5 trials to each target during the perturbation block. Solid line: linear regression. Dashed line: regression line from the early learning data. n indicates the number of data points. (If a monkey did not attempt any trials to a given target during a 5-trial window, those data were excluded.) Performance improves more for targets that are more impaired by the perturbation. *: difference in slopes is significant (t-test, $p < 0.01$). (b) Outside-manifold perturbations across all targets and sessions. Performance decrement is again related to the angle for each target, but performance does not improve with experience. The slopes of all linear regressions are significantly different than 0 (F-test, $p < 0.01$). Data shown are from both monkeys.

## 3.4 DISCUSSION

Sensory-motor learning likely encompasses a variety of neural mechanisms, operating at diverse timescales and levels of organization. We posit that learning a within-manifold perturbation harnesses the fast-timescale learning mechanisms that underlie adaptation (Salinas, 2004), whereas learning an outside-manifold perturbation engages the neural mechanisms required for skill learning (Rioult-Pedotti et al., 2000; Picard et al., 2013). This suggests that learning outside-manifold perturbations could benefit from multi-day exposure (Ganguly and Carmena, 2009; Peters et al., 2014). Such learning might require the IM to expand or change orientation.

Other studies have employed dimensionality-reduction techniques to interpret how networks of neurons encode information (Mazor and Laurent, 2005; Luczak et al., 2009; Churchland et al., 2012; Mante et al., 2013; Rigotti et al., 2013; Cunningham and Yu, 2014) and change their activity during learning (Paz et al., 2005; Durstewitz et al., 2010). Our findings strengthen those discoveries by showing that low-dimensional projections of neural data are not only visualization tools  they can reveal causal constraints on the activity expressed by networks of neurons. Our study also indicates that the low-dimensional patterns present among a population of neurons may better reflect the elemental units of volitional control than do individual neurons.

Related to the development of BCIs, we showed that not all mappings between neural activity and cursor movement are equally learnable. This indicates that in the clinic BCI mappings should be designed by taking into account the properties of the network of neurons being used to control the effector. It is still an open question as to whether subjects would have been able to learn to control the outside-manifold perturbations if they had been given more time, although recent evidence suggests that might be the case (Ganguly and Carmena, 2009). While it is unlikely that a clinician would intentionally give a perturbed mapping to a patient, this work is related to the complexity of mappings that a subject can learn to control. Suppose a patient is learning to use a BCI. One likely scenario would be for the clinician to gradually add degrees of freedom to the BCI (Collinger et al., 2012). The number of degrees of freedom that a subject can learn to control is likely bounded by the intrinsic dimensionality of the IM.

In summary, a BCI paradigm enabled us to reveal neural constraints on learning. The principles we observed may govern other forms of learning (Braun et al., 2010; Gu et al., 2011; Ingvalson et al., 2011; Jeanne et al., 2013; Park et al., 2014) and perhaps even cognitive processes. For example, combinatorial creativity (Boden, 1998), which involves re-combining cognitive elements in new ways, might involve the generation of new neural activity patterns that are within the IM of relevant brain areas. Transformational creativity, which creates new cognitive elements, may result from generating neural activity patterns outside of the relevant IM. More broadly, our results help to provide a neural explanation for the balance we possess between adaptability and persistence in our actions and thoughts (Ajemian et al., 2013).

# 4.0 NEURAL STRATEGIES OF LEARNING

In this chapter, I describe my research analyzing, from a neural population point of view, the specific ways in which monkeys learn to use a BCI. Chapter 3 showed that not all BCI mappings are equally learnable, but how does the monkey adapt his neural activity while attempting to learn? If we understand the neural mechanisms of learning, then perhaps we can design BCI systems that take advantage of the mechanisms.

## 4.1 INTRODUCTION

Learning is a complex process that ultimately results in the generation of new neural activity patterns in response to a stimulus. The behavioral aspects of learning have been extensively studied in humans. Some studies have analyzed the way in which people adapt their movements in perturbed environments (e.g., Lackner and Dizio (1994); Mazzoni and Krakauer (2006); Izawa et al. (2008); Herzfeld et al. (2014)). Other studies have compared the way in which people who have neurological disorders adapt to novel environments relative to the way in which healthy individuals adapt (e.g., Role et al. (2000); Smith and Shadmehr (2005); Nowak et al. (2007)).

The neural aspects of learning have been less well examined. Learning has been characterized at the systems level (e.g., Ungerleider et al. (2002); Orban de Xivry et al. (2012)), the single-neuron level (e.g., Li et al. (2001); Rokni et al. (2007); Picard et al. (2013), and the cellular process level (e.g., Jackson et al. (2006); Holtmaat and Svoboda (2009); Kuhlman et al. (2014); McKenzie et al. (2014)), but few studies have analyzed the neural population aspects of learning.

Jarosiewicz et al. (2008) trained monkeys to control a BCI cursor using a population-vector algorithm decoder. Midway through each session, they rotated the assigned preferred directions of a portion of the neurons, and they analyzed the way the monkeys regained proficient control of the cursor. They proposed three strategies for the way a monkey might adapt his neural activity, and they found evidence for all strategies by analyzing the activity of neurons individually. Ganguly and Carmena (2009) trained monkeys to move a BCI cursor using a linear filter. They shuffled the weights of the decoder, and they showed that across days, the monkey's performance with the shuffled decoder improved. They showed that as the monkey's performance improved, the consistency of neural tuning also increased.

In Chapter 3, we analyzed the performance of monkeys as they moved a BCI cursor using perturbed mappings between neural activity and cursor velocity. Here, we extend those analyses by considering the 'neural strategies' one monkey employed to learn to control the cursor with the perturbed mappings. How does a monkey change his neural activity patterns when he first encounters the perturbation? Does the consistency of neural activity patterns from trial-to-trial reflect the monkey's performance? Zacksenhouse et al. (2007) compared the variance of neural activity between arm control and brain control, and they found that variance was higher during brain control. They also found that the variance decreases across days as the monkeys improved performance.

In addition to analyzing the total covariance of neural activity, we projected the neural activity patterns onto two subspaces within the full neural space to enable a decomposition of the covariance into a signal component and a noise component. We hypothesized that the covariance of the neural activity would increase when we introduce the perturbed decoders, but the increase would not be equal in all dimensions of the neural space Figure 3.1. Lastly, we generated neural trajectories (Yu et al., 2009; Ames et al., 2014) to analyze the trial-to-trial fluctuations in neural activity. We hypothesized that the trial-to-trial fluctuations would decrease as the monkey's performance improves. This is consistent with Zacksenhouse et al. (2007), but we anticipated a decrease in variance within each day whereas they observed it across days.

## 4.2 METHODS AND DATA ANALYSIS

### 4.2.1 Neurophysiology and behavioral recording

The data analyzed in this chapter are from monkey J in Chapter 3. Briefly, the monkey was trained to control a computer cursor on a center-out task using a BCI. At the beginning of each session, the monkey used an 'intuitive mapping', which was a decoder that allowed him to move the cursor proficiently. Midway through each session, we perturbed the intuitive mapping. There were two types of perturbations, and we chose one on each session. To learn to control the cursor under a 'within-manifold perturbations', the monkey could continue to use the same co-modulation patterns among the recorded neurons, but he needed to learn the new relationship between co-modulation patterns and cursor kinematics. To learn to control the cursor under an 'outside-manifold perturbation', the monkey needed to generate new co-modulation patterns among the recorded neurons. We mapped the neural activity to cursor kinematics using the linear relationship in Equation 2.1, repeated here

$$\hat{\mathbf{x}}_t = \mathrm{M}_1\hat{\mathbf{x}}_{t-1} + \mathrm{M}_2\mathbf{u}_t + \mathrm{M}_0$$

To implement a perturbation, we manipulated $\mathrm{M}_2$.

### 4.2.2 Neural activity projections

Each session, we recorded the neural activity from approximately 90 electrodes. We can represent the activity on all electrodes in a high-dimensional space where each axis corresponds to the firing rate on one electrode (Figure 3.1). In this work, we project the neural activity onto different two-dimensional surfaces in the high-dimensional space. The first 2D space we consider is the 'control plane'. Neural activity projected onto this plane drives the cursor. Said another way, the control space is the row space of $\mathrm{M}_2$ for the perturbed mapping. Hereafter, this space is referred to as the 'control plane'. Neural activity projected into the null space of the control plane (i.e., the $(n-2)$-dimensional space orthogonal to the row space of $\mathrm{M}_2$, where $n$ is the number of electrodes) does not drive the cursor. The second 2D space we consider is the 'null plane'. The null plane is the 2D plane within the null space that captures the most neural activity covariance. By definition, the null plane is orthogonal to the control plane.

The following procedure describes how we identify the null plane. First, we find an orthonormal basis of $\mathrm{M}_2$. We call this basis $\mathrm{M}_{2,\mathrm{orth}}$, and the row space of $\mathrm{M}_{2,\mathrm{orth}}$ is the same as the row space of $\mathrm{M}_2$. Next, we find an orthonormal basis of the $(n-2)$-dimensional space orthogonal to $\mathrm{M}_{2,\mathrm{orth}}$. We call this space $\mathrm{M}_{2,\mathrm{null}}$, and the size of $\mathrm{M}_{2,\mathrm{null}}$ is $(n-2) \times n$. Next, we project all of the neural activity recorded during the perturbation block into the null space.

$$\mathbf{u}_{\mathrm{null}} = \mathrm{M}_{2,\mathrm{null}}\mathbf{u} \tag{4.1}$$

We then use factor analysis to find the two dimensions of $\mathrm{M}_{2,\mathrm{null}}$ that capture the most covariance of $\mathbf{u}_{\mathrm{null}}$. Factor analysis is described in more detail in Section 2.2.1. The matrix $\beta_{\mathrm{null}}$ defines the mapping between the $(n-2)$-dimensional space and the null plane.

$$\beta_{\mathrm{null}} = \Lambda_{\mathrm{null}}^T(\Lambda_{\mathrm{null}}\Lambda_{\mathrm{null}}^T + \Psi_{\mathrm{null}})^{-1} \tag{4.2}$$

where $\Lambda_{\mathrm{null}}$ and $\Psi_{\mathrm{null}}$ are fit using expectation-maximization. We then define the mapping from the $n$-dimensional neural space to the null plane

$$\mathrm{M}_{\mathrm{null}} = \beta_{\mathrm{null}}\mathrm{M}_{2,\mathrm{null}}^T \tag{4.3}$$

where the row space of $\mathrm{M}_{\mathrm{null}}$ is the null plane.

Once the planes were identified, we projected neural activity onto the control plane

$$\mathbf{u}_{\mathrm{control}} = \mathrm{M}_{2,\mathrm{orth}}\mathbf{u} \tag{4.4}$$

and onto the null plane

$$\mathbf{u}_{\mathrm{null}} = \mathrm{M}_{\mathrm{null}}\mathbf{u} \tag{4.5}$$

for further analysis. Figure 4.1a,b show an example of finding the null plane within a high-dimensional neural space. In the figure, the control plane and the null plane are shown as lines rather than planes for visualization. Figure 4.1c,d show histograms of the neural activity projected onto the control plane and null plane.

### 4.2.3 Decomposition of covariance

We utilize the law of total covariance to understand the spread of neural activity in the planes defined above. We wanted to identify how much of the total covariance of the neural activity was due to the monkey driving the cursor to different targets and how much was due to variability when driving the cursor to a given target. Figure 4.2 provides an intuitive explanation. Suppose we record from two neurons, and we plot their firing rates as the monkey moves the cursor to eight targets (panel a). Panel b shows representations of the covariances. The size of the outer black ring represents the total covariance of all neural activity for all targets. The size of dashed black ring indicates the covariance across targets (i.e., the covariance of the means of each colored cluster in panel a). The size of the colored rings indicate the covariance of each cluster of neural activity. The law of total covariance states that

$$\operatorname{cov}(\mathbf{u}') = \operatorname{E}(\operatorname{cov}(\mathbf{u}' \mid i)) + \operatorname{cov}(\operatorname{E}(\mathbf{u}' \mid i)) \tag{4.6}$$

In this application, $\mathbf{u}'$ are the two-dimensional spike counts (i.e., $\mathbf{u}_{\text{control}}$ or $\mathbf{u}_{\text{null}}$), and $i$ is the identity of the target. $\operatorname{cov}(\mathbf{u}')$ is the total covariance. $\operatorname{E}(\operatorname{cov}(\mathbf{u}' \mid i))$ is the covariance within targets, and $\operatorname{cov}(\operatorname{E}(\mathbf{u}' \mid i))$ is the covariance across targets. Those three terms are $2 \times 2$ covariance matrices. For simplicity, we extracted geometric mean of the eigenvalues of each covariance matrix. These geometric means are plotted in Figures 4.4 - 4.6.

### 4.2.4 Neural trajectories and trial-to-trial variability

The decomposition of covariance analyzes covariance across many trials, but it ignores trial-to-trial variations in neural activity. To assess trial-to-trial variability, we began by creating ‘neural trajectories’ for each trial. A neural trajectory describes how neural activity evolves in the neural space during a trial. We extracted neural trajectories from the normalized spike counts using a similar method to how we extract cursor positions from the spike counts. The

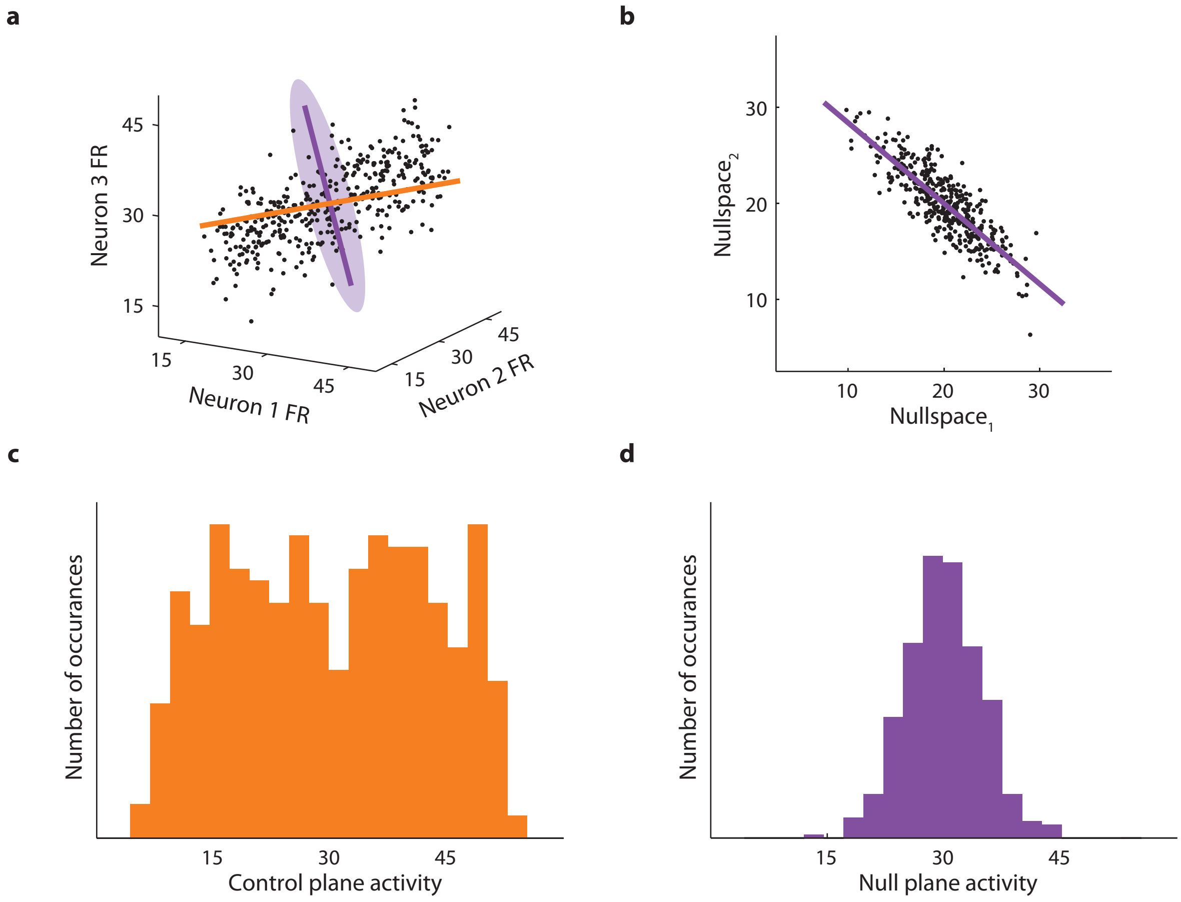

Figure 4.1: Determination of null plane. (a) Neural activity (black dots) is plotted in a three-dimensional neural space. Each axis represents the firing rate of one neuron. The control plane is depicted by the orange line. The purple shaded space represents the null space of the control plane, and the purple line represents the null plane. (b) Neural activity projected onto the two dimensions of the null space. The null plane (purple line) is the dimension in the null space with the greatest covariance in the neural activity. (c) Histogram of neural activity projected onto the control plane. (d) Histogram of neural activity projected onto null plane.

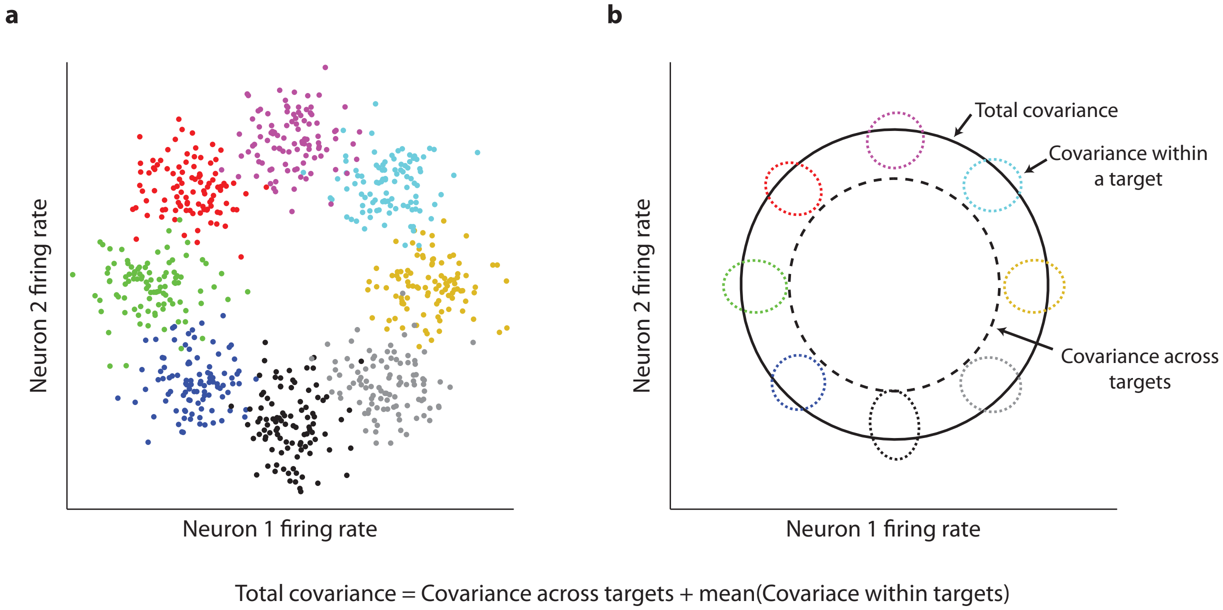

Figure 4.2: Decomposition of covariance. (a) Simulated firing rates (dots) from two neurons as a monkey moves a cursor to eight targets. The color of the dots indicate the target to which the monkey was moving the cursor. (b) Decomposition of covariance of neural activity in a. Solid black ring, total covariance. Dashed black ring, covariance across targets. Colored dotted rings, covariance within each target.

decoder (i.e., Equation 2.1) essentially smooths the spikes counts across time to extract cursor velocity. Then, it extracts cursor positions by inegrating the velocity across time. To create a neural trajectory, we first smoothed the spike counts by passing them through a Gaussian kernal (standard deviation of 225 ms). Then, we integrated the smoothed spike counts across time.

$$\mathbf{p}_0 = \mathbf{0} \tag{4.7}$$

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \tilde{\mathbf{u}}_t \tag{4.8}$$

where $\mathbf{p}$ is the neural trajectory, and $\tilde{\mathbf{u}}_t$ are the smoothed spike counts at time $t$. Figure 4.3 shows two example neural trajectories in blue and green. To assess the variability between two trials, we measured the Euclidean distance points in the neural trajectories at corresponding times, indicated by the gray lines, and we defined the variability as the average length of those lines. For this analysis, we measured the trial-to-trial variability for trials to each target separately. For example, suppose the monkey was presented with targets in the following order: $[0°, 45°, 90°, 45°, 90°, 0°...]$. The trial-to-trial variability on trial 4 would be determined by comparing the trajectories from trials 2 and 4. The trial-to-trial variability on trial 5 would be determined by comparing the trajectories from trials 3 and 5, and the trial-to-trial variability on trial 6 would be determined by comparing the trajectories from trials 1 and 6.

## 4.3 RESULTS

We trained a monkey to control a cursor using a BCI in a center-out task. During each session, we changed the mapping that translated neural activity into cursor velocity. To learn the new mappings, the monkey had to adapt the neural activity patterns in one of two ways. He either had to learn a new association between the natural co-modulations among the neurons and the kinematics of the cursor (within-manifold perturbations), or he had to

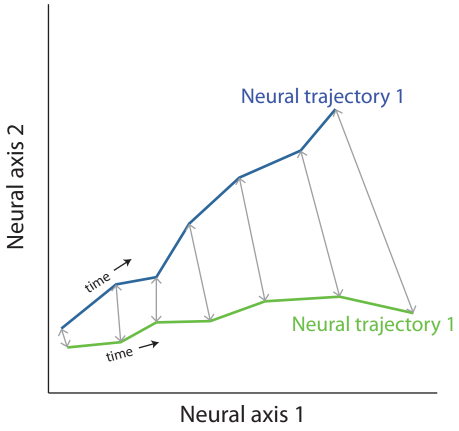

Figure 4.3: Neural trajectories and trial-to-trial variability. Neural trajectories from two trials (blue and green lines) in a two-dimensional neural space. The horizontal axis represents the integrated firing rates from neuron 1, and the vertical axis represents the integrated firing rates for neuron 2. The trial-to-trial variability was computed as the average distances between points in the trajectories across time (gray lines).

learn to generate new co-modulations among the neurons (outside-manifold perturbations). In Chapter 3, we assessed how well monkeys were able to do this, and we found that they could more easily learn to re-associate neural activity patterns with kinematics than they could learn to generate new co-modulations among the neurons. In this chapter, we assess the 'neural strategies' that one monkey used to learn the new mappings.

### 4.3.1 Total variance of neural activity

For each session, we measured the covariance of the neural activity in the full neural space during the last 30 trials of the baseline block and the first 30 trials of the perturbation block. Figure 4.4 shows these covariance values. The data are plotted as covariance in the first 30 perturbation trials against covariance in the last 30 baseline trials to enable three comparisons between the perturbation types: 1) along the horizontal axis, we compare the covariances during baseline, 2) along the vertical axis, we compare the covariances during perturbation, and 3) along the diagonal, we compare the change in covariance from baseline to perturbation.

The total covariance in the last 30 baseline trials was not significantly different between the two perturbation types, as indicated by the marginal histograms along the horizontal axis. This is reassuring as it helps to confirm that differences we observe between the perturbation types are not due to systematic differences in neural activity between datasets. The total covariance in the first 30 trials of the perturbation was also not significantly different between the two types of perturbations, as indicated by the marginal histograms along the vertical axis. For both perturbation types, the monkey tended to increase the total covariance when the perturbation was applied, but covariance increased by similar amounts for the two types of perturbations. This is indicated by the marginal histograms along the diagonal.

### 4.3.2 Decomposition of neural activity covariance

To further assess how the monkey learns to change his neural activity patterns to control the cursor under the perturbed mapping, we decomposed the total covariance into covariance across targets and covariance within targets. We can think of the covariance across targets as the 'signal' component of the total covariance and the covariance across targets as the 'noise' component.

Because the dimensionality of the neural space is approximately 90D and there are only eight target angles, the covariance across targets (as we have defined it using the geometric mean of the eigenvalues) would be 0. Therefore, we projected the neural activity onto two planes of interest and then performed the decomposition of covariance. The first plane of interest is the control plane of the perturbed BCI mapping. Neural activity in this plane controls the cursor velocity. Neural activity in dimensions orthogonal to this plane do not affect the cursor velocity. The second plane of interest is the null plane. The null plane is orthogonal to the control plane, and the null plane accounts for the most neural activity covariance in the null space of the control plane.

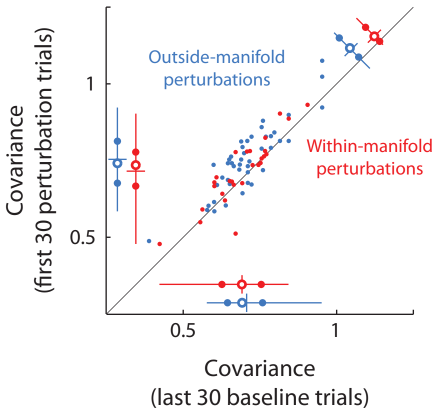

Figure 4.4: Covariance in the full neural space. Each dot represents one session. Total covariance in the full neural space during the first 10 perturbation trials is plotted against total covariance in the full neural space during the last 30 baseline trials. Line segments along each axis and the diagonal represent the marginal histograms for baseline (horizontal axis), perturbation (vertical axis), and the difference between perturbation and baseline (diagonal axis). The length of the line segments indicate the inner 90% of the data. Tick marks indicate the average value. Solid and hollow dots indicate the inner-quartile boundaries. Data above the diagonal indicate greater covariance during the first 30 perturbation trials, and data below the diagonal indicate greater variance during the last 30 baseline trials.

**4.3.2.1 Control planes** The plots in Figure 4.5a shows the decomposition of covariance for neural activity projected onto the control plane for the last 30 trials of baseline and the first 30 trials of perturbation. It is important to note that the control plane onto which we projected the neural activity is the control plane of the perturbed mapping, not the intuitive mapping. The left panel assesses the total variance in these projections (i.e., the size of the solid black ellipse in Figure 4.2b). As we expected, before and during the perturbation, there is more covariance in the neural activity projected onto the control plane for within-manifold perturbations than for outside-manifold perturbations. Recall from Chapter 3 that the intrinsic manifold is the 10-dimensional space within the full neural space that accounts for the most covariance in the neural activity. By definition, the control plane for a within-manifold perturbation lies within the intrinsic manifold, and the control plane for an outside-manifold perturbations lies outside of the intrinsic manifold. Therefore, we expect, at least for the baseline data, that there should be more variance in the control plane for within-manifold perturbations than for outside-manifold perturbations.

Despite the difference in baseline values, we can still compare the change in variance from baseline the perturbation. The histogram along the diagonal in Figure 4.5a shows that total variance increased for both types of perturbations but by a statistically similar amount. The left panel of Figure 4.5b shows that the change in total variance was not related to the amount of learning the monkey demonstrated.

The middle panel of Figure 4.5a compares the covariance across targets for the neural activity projected onto the control plane (i.e., the size of the black dashed ellipse in Figure 4.2b). There were no statistically significant differences between the two types of perturbations in this projection. Interestingly, the change in covariance across targets from baseline to perturbation was related to the amount of learning the monkey demonstrated (middle panel, Figure 4.5b). As the change in covariance increases, the amount of learning the monkey demonstrates also increases. This effect approached significance ($p = 0.08$). An increase in covariance across targets essentially means that the neural activity across targets is becoming better separated. One way to picture this is as if the colored ellipses in Figure 4.2b moved away from each other. That would indicate less overlap in the neural activity patterns

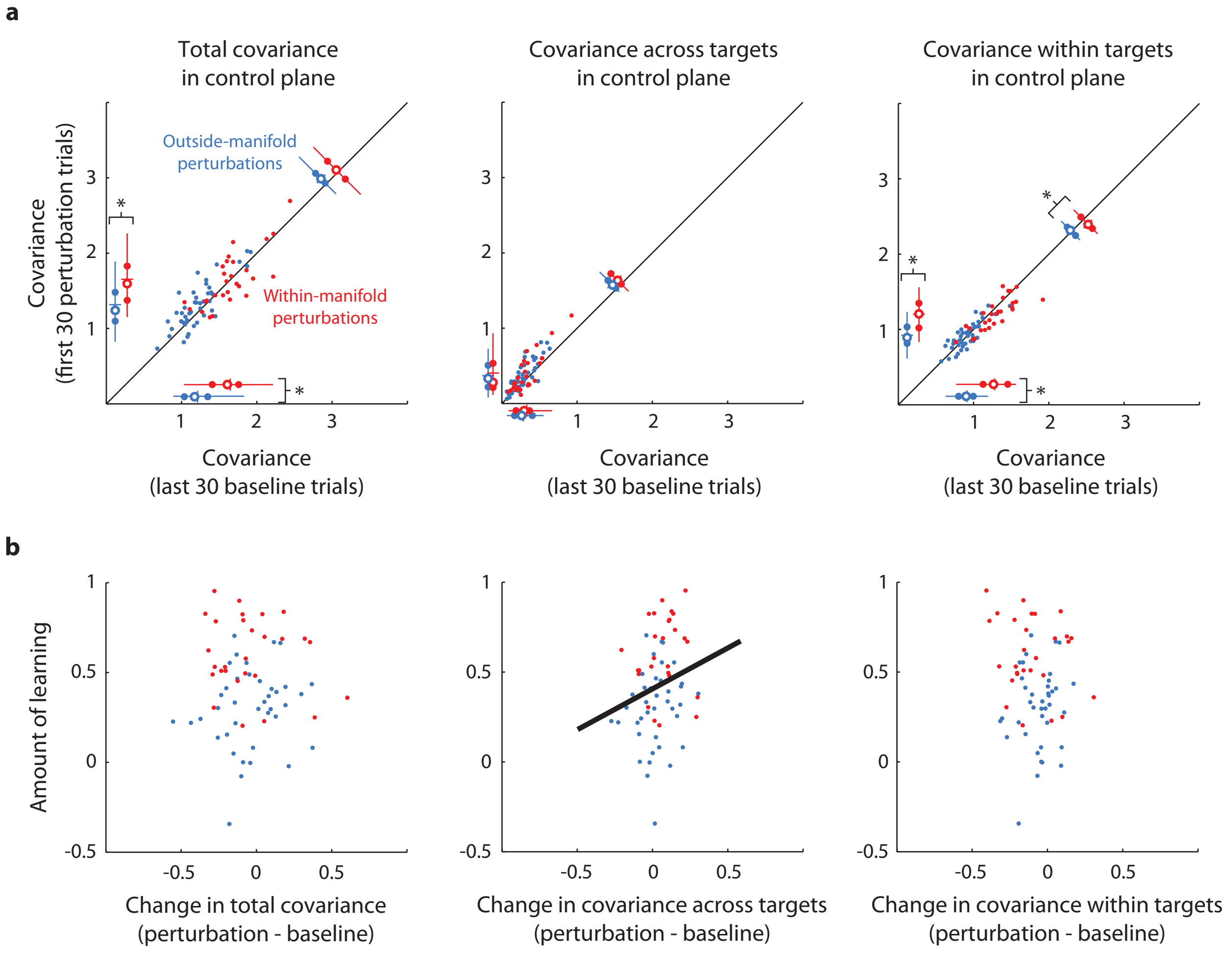

Figure 4.5: Covariance in the control plane. (a) Decomposition of neural covariance projected onto the control plane. Left panel, total covariance. Same plotting format as Figure 4.4. *, significant difference between distributions (t-test, $p < 0.05$). (b) Relation between amount of learning and change in covariance of neural activity projected onto the control plane. For total covariance (left panel) and covariance within targets (right panel), there was no significant relationship between learning and change in covariance. For covariance across targets (middle panel), the relationship approached significance ($p = 0.08$).

that are generated as the monkey drives the cursor to each target – resulting in more direct cursor trajectories and higher proficiency.

The right panel of Figure 4.5a shows the covariance within targets for neural activity projected onto the control plane (i.e., the sizes of the colored ellipses in Figure 4.2b). We see that there was significantly more covariance within targets for within-manifold perturbations before and after the perturbation was introduced. This accounts for the difference in total variance seen in panel a. Interestingly, the covariance within targets decreases for within-manifold perturbations while it slightly increases for outside-manifold perturbations. This is also consistent with our intuition of how the monkey learns improves his performance. As the clusters become smaller in the control plane, that indicates velocity is more consistent (i.e., the cursor takes 'straighter' trajectories to the targets.

The change in covariance across targets was not related to the amount of learning the monkey demonstrated (right panel, Figure 4.5b). Decreasing the variance within targets can be visualized as decreasing the sizes of the clusters of colored points in Figure 4.2a. By decreasing the size of those clusters, the cursor velocity is likely to be more similar from trial to trial, which is indicative of better performance.

**4.3.2.2 Null planes** The plots in Figure 4.6 shows the covariance of the data projected onto the null plane. The data are presented in the same format as Figure 4.5. As indicated by the marginal histograms along the horizontal and vertical axes in Figure 4.6a, there were no significant differences between the two types of perturbation for total covariance, covariance across targets, or covariance within targets for baseline and perturbation neural activity projected onto the null plane.

However, there were significant differences in the change in covariance from baseline to perturbation. Total covariance in the null plane increased for both types of perturbations, but it increased significantly more for within-manifold perturbations than for outside-manifold perturbations (left panel). This was driven mostly by the change in covariance across targets, where the change is significantly larger for outside-manifold perturbations than for within-manifold perturbations (middle panel). Covariance within targets tended to decrease for both types of perturbation (right panel). Both the change in total covariance and the

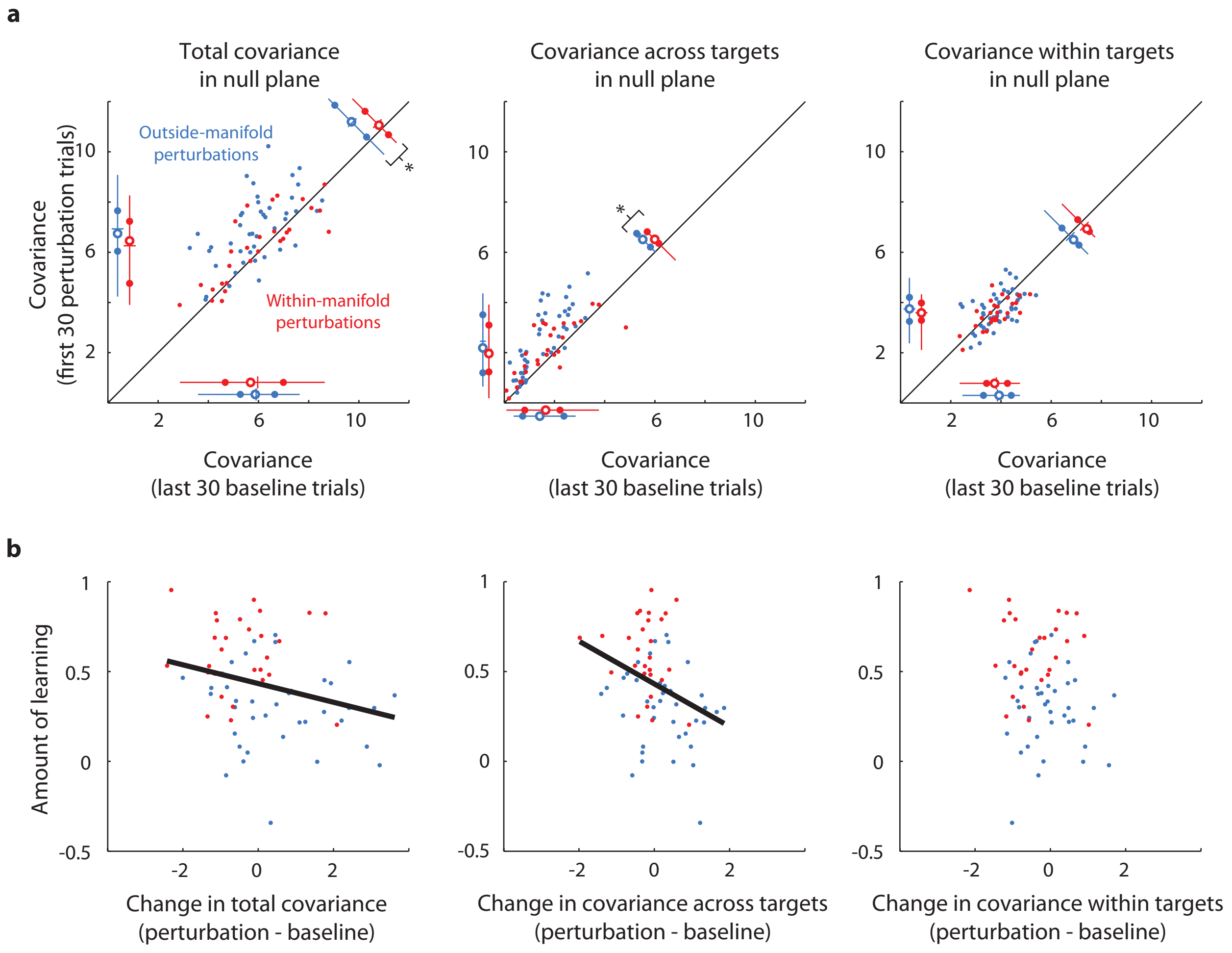

Figure 4.6: Covariance in the null plane. Same plotting format as Figure 4.5. (a) Total covariance and covariance across targets increased more for outside-manifold perturbations than for within-manifold perturbations. (b) There were significant relationships between the amount of learning and the change in total covariance and between the amount of learning and the change in covariance across targets.

change in covariance across targets were related to the amount of learning demonstrated by the monkey (Figure 4.6b). As the monkey increased covariance, the amount of learning tended to decrease.

This is consistent with learning in that as the monkeys 'search' through the neural space to find the neural activity patterns that appropriately move the cursor, neural covariance will increase. For outside-manifold perturbations, the monkeys increases covariance in the null plane, and by definition, neural activity in the null plane does not affect the cursor kinematics. Increasing the variance in the null plane more for outside-manifold perturbations than for within-manifold perturbations may not be the specific reason why the monkeys were better able to learn within-manifold perturbations, but it is consistent with less learning for outside-manifold perturbations.

Considering Figures 4.4, 4.5, and 4.6 together, we can begin to understand how the monkey changes his neural activity during learning. For both perturbation types, the total covariance in the full neural space increases by similar amounts. The total covariance also increases by similar amounts within the control plane. From observation, we know that the monkey was able to move the cursor faster for within-manifold perturbations than for outside-manifold perturbations. Because of this, the errors in the cursor trajectories were amplified for the within-manifold perturbations, and it was easier for the monkey to learn to make the appropriate corrective actions on those sessions. That is why the monkey was able to decrease the covariance within targets in the null plane for within-manifold perturbations. As the monkey continued to find an appropriate strategy for the outside-manifold perturbations, it ended up increasing variance in the null plane, which did not lead to an improvement in the cursor trajectories.

### 4.3.3 Trial-to-trial variability

The analyses in Section 4.3.2 address how the aggregate neural activity for a group of 30 trials relates to the aggregate activity for another group of 30 trials. The analyses do not take into account the changes that happen from one trial to the next. To measure the trial-to-trial variability, we generated neural trajectories for every trial. We then measured the Euclidean distance in the neural space between trajectories consecutive trials to the same target. We performed this analysis only on the first 100 trials during perturbation because we are most interested in how variability changes in the first stages of learning. Figure 4.7 relates the distance between trajectories to the monkey's performance. The horizontal axis shows the moving average success rate. At each trial, we computed the success rate (i.e., percent of successful trials for the current and the previous nine trials). There is a doubly-linear relationship between trial-to-trial variability and performance. When performance is low (i.e., success rate less than 50%), the trial-to-trial variability is high and fairly constant (as indicated by $L_1$). As performance increases from 50% success rate to 100%, the trial-to-trial variability decreases (as indicated by $L_2$).

When the perturbation is first introduced, the monkeys' performance drops (Figure 3.8). Because of this, the monkeys will attempt to activate their neurons in a new pattern to find a pattern that proficiently moves the cursor. The monkey will continue to try new patterns until performance improves. High trial-to-trial variability is indicative of attempting to use different neural activity patterns to control the cursor. Perhaps there is a physiological limit to the variability that can occur from one trial to the next. This might explain why $L_1$ is flat. As a more tangible example of trial-to-trial variability decreasing with learning, consider someone learning to play golf. The novice will continue to adjust different variables such as hand grip position, club face angle, and swing speed until he learns to hit the ball proficiently. At that point, the golfer will use a consistent swing strategy, thus reducing the 'swing-to-swing' variability.

## 4.4 DISCUSSION

In this work, we trained a monkey to control a cursor on a 2D center-out task using a BCI. We applied perturbations to the decoders, and we observed how well the monkeys learned to generate the appropriate neural activity patterns to proficiently move the cursor under the perturbation.

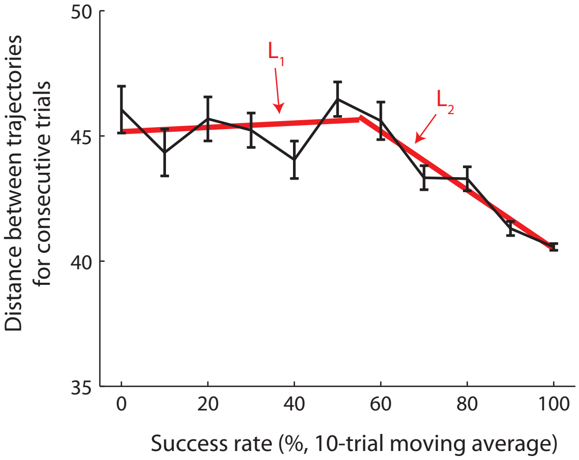

Figure 4.7: Brain-computer interface performance and trial-to-trial variability. There was a doubly-linear relationship between trial-to-trial variability (as measured by Euclidean distance between neural trajectories) and success rate. When performance was low, trial-to-trial variability was high. As performance increased, trial-to-trial variability decreases. Error bars represent standard error.

We found that neural covariance in the full neural space increased when we introduced the perturbed mappings (Figure 4.4). Other research has shown that muscular coactivation (i.e., the simultaneous contraction of antagonistic muscles) occurs when subjects are instructed to make movements in an environment with altered dynamics (Granata and Marras, 2000; Milner, 2002; Darainy and Ostry, 2008; Huang and Ahmed, 2014). Perhaps the effect we observed is a neural instantiation of muscular coactivation. The dynamics of the environment changed when we perturbed the decoders, and in order to compensate for the perturbation, the monkeys coactivated all of the neurons. Thus, the covariance of the neural activity in the full neural space increased.

One interesting finding from these results is that the covariance within targets is greater than the covariance across targets (Figure 4.5a and Figure 4.6a). This is a bit surprising given that the subjects were able to proficiently control the cursor in many of the sessions with the perturbed mappings. One would have expected that the variability within targets would need to be relatively small in order for the subject to have proficient control of the cursor. Perhaps this effect was a result of the task design. We used eight targets that were equally spaced in the workspace, but there were no constraints on how the monkeys were required to move the cursor to the target. They could move the cursor in a straight line or in a more circuitous trajectory. Those two cursor trajectories would have resulted from quite different patterns of neural activity. If we had instead placed tight constraints on how the monkey moved the cursor (see Chapter 6), then there might have been less within target covariance.

An open question that remains to be answered is what specifically the monkeys are doing to change the variance of their neural activity. Previously, we proposed that they might be coactivating their neurons as a result of the altered dynamics of the perturbed decoders. Another possibility is that the monkeys might first attempt to use neural strategies that allowed them to control previous decoder perturbations. For example, if on one day, the monkey learns to control the perturbed decoder well, then on the next day, the monkey might attempt to use that same strategy when he encounters the perturbation. By fully understanding how a monkey adapts his neural activity when he encounters altered dynamics in a BCI context, then we might be able to understand the learning strategies that people use in contexts such as perceptual learning, motor learning, and sensory learning.

# 5.0 INCORPORATION OF FACTOR ANALYSIS INTO A KALMAN FILTER

In this chapter, I describe my research incorporating dimensionality reduction into a BCI decoder. We used this decoder in the work described in Chapters 3 and 4. The incorporation of dimensionality reduction allowed us to study learning within and outside of the intrinsic manifold, but that was not our original motivation. We wanted to test whether the incorporation of dimensionality reduction would allow monkeys to learn to control a BCI better than when dimensionality reduction is not incorporated.

## 5.1 INTRODUCTION

Many previous studies have investigated methods for improving neural prosthetic decoding algorithms so that the subject can control the effector with more precision. Bishop et al. (2014) introduced a method for automatically adapting decoding parameters for discrete decoding. In offline analyses, they showed that their algorithm led to improved decoding performance across many sessions relative to using the same parameters over the same duration. Orsborn et al. (2014) introduced a similar method for continuous decoding. They found that the closed-loop decoder adaptation algorithm led to better cursor control than if they had not adapted the parameters.

In other work, Wu et al. (2006) found that the Kalman filter outperformed the population vector algorithm and linear filter in offline reconstructions of reach trajectories. The Kalman filter's advantage likely arose from it being a two-stage decoder. The Kalman filter first estimates the effector's kinematics from the kinematics at the previous timestep. Then, it updates the estimate using neural activity. The other decoders directly estimate kinematics from neural activity at each timestep. Wu and colleagues also investigated how the number of decoded kinematic parameters (e.g., position, velocity, acceleration, etc.) impacted decoding performance. They found the lowest reconstruction error when they simultaneously decoded the position, velocity, and acceleration of movement. Li et al. (2009) used an unscented Kalman filter for BCI decoding. They found that this decoder led to better offline arm trajectory reconstruction and better online control of the cursor than both the standard Kalman filter and the Wiener filter. Gilja et al. (2012) implemented modifications to the standard Kalman filter algorithm and to the way it is calibrated that also led to improved performance relative to the standard Kalman filter.

The work presented in this chapter is philosophically aligned with those efforts. We incorporate dimensionality reduction into the standard Kalman filter. Dimensionality reduction techniques are ways of describing high-dimensional data with a low-dimensional set of latent factors (Cunningham and Yu, 2014). Dimensionality reduction has been used in diverse fields such as face recognition (Turk and Pentland, 1991) and human gait analysis (Omlor and Giese, 2011). In neuroscience, dimensionality reduction has been used to study working memory (Machens et al., 2010), movement generation (Churchland et al., 2012), olfaction (Mazor and Laurent, 2005), decision-making (Briggman et al., 2005), and other neural processes.

Two common dimensionality reduction techniques are principal component analysis (PCA) (Pearson, 1901) and factor analysis (FA) (Everitt, 1984). PCA identifies latent dimensions that capture the most *variance* of the data, while FA identifies latent dimensions that capture the most *covariance* of the data. This difference can be seen in Figure 5.1. Neuron spiking is well-modeled by a Poisson process. That is, the variance in the number of spikes generated in small time bins is equal to the mean number of spikes. In this figure, neurons 1 and 2 are correlated – when neuron 1 increases its firing rate, neuron 2 also increases its firing rate. Neuron 1 has a much larger firing rate than neuron 2. We can describe the firing of these two neurons with one latent dimension. The orange line is the latent dimension identified by PCA, and the purple line is the latent dimension identified by FA. The latent identified by FA takes into account the larger variance in neuron 1 and identifies a direction that is closer to the true relationship from which the data were generated (gray line).

In this work, we incorporated FA into a Kalman filter. We hypothesized that the neural spiking activity is a noisy representation of high-level control signals and that FA would be able to extract those control signals from spikes. By driving the decoder with latent activity instead of spiking activity, the subject might have better control of the cursor, relative to a decoder that does not incorporate dimensionality reduction.

Other work has incorporated dimensionality reduction with BCI decoding. Santhanam et al. (2009) incorporated factor analysis into offline discrete decoding. They found the FA could effectively model trial-to-trial variability to improve target classification. Kao et al. (2013) incorporated principal components analysis into an online decoder in order to improve the robustness of the decoder. They showed that when a neuron was 'lost', decoder accuracy was higher when PCA was included in the decoder than when PCA was not included. The work that we present here is the first we are aware of that incorporates FA into a closed-loop (i.e., online) Kalman filter. We investigated how decoding performance varied with the number of latent dimensions included in the FA model.

## 5.2 METHODS AND DATA ANALYSIS

### 5.2.1 Subjects and brain-computer interface task

For this experiment, one monkey (J) was trained to perform the instructed path task (see Chapter 6) with straight paths, single-inflection paths, and double inflections paths with a BCI cursor. Briefly, at the beginning of each trial, a start target appeared on the screen, and the monkey moved the cursor moved the cursor to that target using a BCI. Once the start target had been acquired, an instructed path appeared on the screen. The path went from the start target to the location of the end target. The monkey moved the cursor through the path, and when the cursor was 75% of the way through the path, the end target appeared. When the monkey acquired the end target with the cursor, a water reward was given. For all

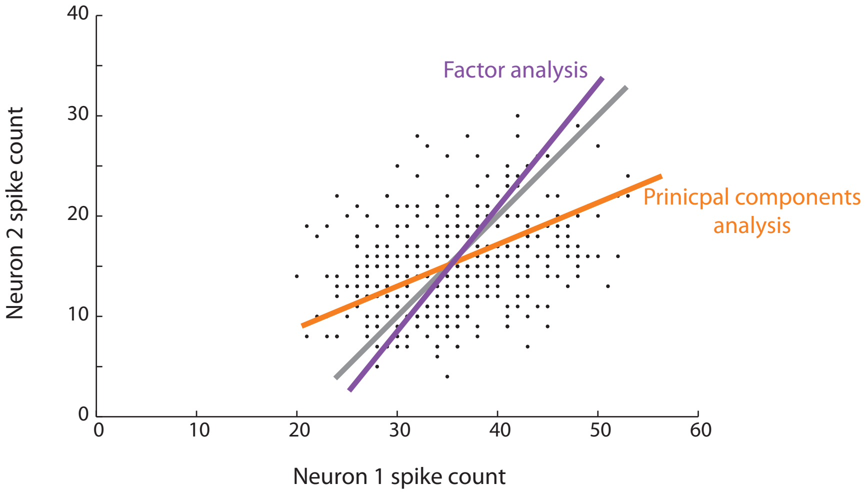

Figure 5.1: Dimensionality reduction example. The firing rates for two neurons are positively correlated, and neuron 1 has larger firing rates than neuron 2. Spike counts for these neurons were generated assuming a Poisson distribution. The orange line indicates the latent direction obtained through principal component analysis. The purple line indicates the latent direction obtained through factor analysis. The gray line indicates the true relationship from which the spike counts were drawn.

sessions in this chapter, the path tolerance radius was 50 mm, and the straight-line distance from the start target to the end target was 225 mm. We used the instructed path task here because we wanted to make the task more difficult. This would make it more realistic than conventional center-out reaching, and also give us more evidence for a performance differential among the decode algorithms we tried.

There were 8 possible combinations of start and end targets, chosen pseudorandomly. For each pair of targets, there was one possible straight instructed path trajectory, and there were two possible single-inflection and double-inflection trajectories. That is, the inflected trajectories could either be inflected clockwise or counter-clockwise.

During each session, the monkey was presented with only one type of instructed trajectory. All sessions with straight instructed trajectories occurred first, followed by sessions with single-inflection paths, and followed by sessions with double-inflection paths. Each session was divided into three blocks. The monkey used the same decoder for the first and third experimental blocks and a different decoder for the second experimental block (400 trials per block). Portions of the data in Chapter 6 are analyzed in this chapter. Additional data is included here that is not included in Chapter 6.

### 5.2.2 Brain-computer interface decoders

The details of the incorporation of factor analysis into a Kalman filter are described in Section 2.2.1. Briefly, we first reduced the dimensionality of the spike counts to $p$ latent dimensions. We then related the latent dimensions to cursor kinematics using a Kalman filter. In this work, we decoded six kinematics elements: position, velocity, and acceleration in horizontal and vertical coordinates. That is, $r = 6$. The location of the cursor on the screen was determined from the position elements in the kinematic vector.

In this work, we tested the ability of a monkey to control BCI decoders with different dimensionalities in the FA model. For each instructed path type, we tested decoders with the following FA dimensionalities: 6, 10, 15, and 25. We also compared performance with the modified Kalman filter to performance with the standard Kalman filter (i.e., full neural

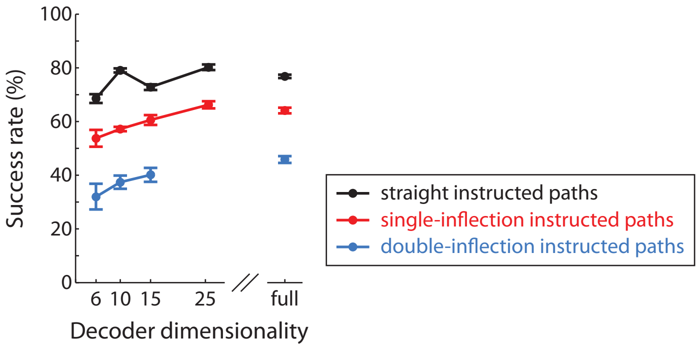

Figure 5.2: Decoder dimensionality and success rate. Success rates for all trials for a given task type and latent dimensionality are shown by the dots. Error bars indicate the 95% confidence interval as indicated by the Bernoulli process. 'Full' decoder dimensionality indicates the standard Kalman filter without FA. Within each task type and across task types, each distribution of success rates was significantly different than all other distributions.

dimensionality). Within each session, we tested two different decoders. Across sessions for each type of instructed trajectory, we tested all combinations of decoder dimensionalities.

We calibrated the parameters of these decoders by having the monkey perform the instructed path task under hand control with multiple path types: single-inflection, double-inflection, triple-inflection, and u-shaped (see Section 6.3.5).

## 5.3 RESULTS

We trained one monkey to perform the instructed path task with three path types using a BCI. We used BCI decoders that incorporated factor analysis with different latent dimensionalities in order to identify the dimensionality at which performance was highest. Figure 5.2 shows the success rates (i.e., percentage of successful trials) for each path type and decoder latent dimensionality. Success rates and their corresponding error bars were computed using the same methods as described in Section 6.2.3.

In general, for each path type, success rates increase as the latent dimensionality increases. For straight instructed paths, success rate is maximum at 25 latent dimensions, but the peak success rate is not substantially different from success rates with 10 latent dimensions or the standard Kalman filter (i.e., ‘full’ dimensionality). There is a small dip is the success rate curve at 15 latent dimensions. This is in contrast to the curves for single-inflection and double-inflection paths, which did not show a dip at 15 dimensions.

The single-inflection instructed path success rate curve peaked at 25 latent dimensions, and the peak was similar to the success rate with the standard Kalman filter. The double inflection instructed path success rate peaked with the standard Kalman filter.

Overall, the success rates were highest for straight instructed paths, followed by single inflection instructed paths and double-inflection instructed paths. This trend is not surprising and is in agreement with success rate decreasing as task difficulty increases. The added constraints of the inflected paths likely increased the task difficulty.

## 5.4 DISCUSSION

We found that BCI decoding performance increased as the number of latent dimensions in the factor analysis model is increased. The peak success rate with the Kalman filter incorporating factor analysis is not substantially different than the success rates with the standard Kalman filter. This does not support our hypothesis, which was that there would be a latent dimensionality at which performance was highest.

We incorporated factor analysis into the Kalman filter because we thought factor analysis would extract ‘latent drivers’ of neural activity. That is, we thought there were low-level underlying processes within the brain that drove the spiking of the neurons. One potential reason for why there is not a latent dimensionality with peak performance is that the standard Kalman filter sufficiently extracts those latent drivers. The Kalman filter finds a set of weights that relate the spiking activity of the neurons to the kinematics of the cursor. If a neuron is not well-modulated due to a latent driver, then it is likely to also not be well correlated with cursor kinematics. For those reasons, both the Kalman filter that incorporates FA and the standard Kalman filter would give that neuron a low weight.

While we did not identify a latent dimensionality that led to to peak performance, low latent dimensionalities led to worse performance than with higher latent dimensionalities or with a standard Kalman filter. This is likely due to the FA models with low dimensionality removing too much of the neural signal that was needed to control the cursor. For example, it might have been the case that latent dimensions 1 - 5 controlled the horizontal velocity of the cursor more than the vertical velocity, whereas dimensions 6 - 10 controlled the vertical velocity more than the horizontal velocity. While this is an extreme example, it illustrates how including too few latent dimensions could restrict the monkey's ability to control the BCI cursor.

This work is related to the earliest demonstrations of BCI control. Fetz (1969) and Fetz and Finocchio (1971) trained monkeys to modulate the activity of one and two neurons independently. Instead of controlling a computer cursor, these monkeys controlled voltage traces on an oscilloscope. It is slightly contradictory that monkeys could modulate two neurons two control two degrees of freedom, but in our study, the monkey needed approximately 10 latent factors to control cursor with two degrees of freedom. This suggests that the dimensionality reduction step we incorporated into the decoder does not accurately identify the dimensions of control within the brain.

In this set of experiments, the monkey controlled the cursor in two dimensions (i.e., horizontal and vertical on the screen). However, many BCI systems include more dimensions in the effectors. For example, some BCIs control robotic limbs with many degrees of freedom (Velliste et al., 2008; Collinger et al., 2012; Hochberg et al., 2012), and some BCIs control cursors in three dimensions (Jarosiewicz et al., 2008). Perhaps the number of latent dimensions required to achieve the same performance as with the standard Kalman filter would be higher than we found here.

# 6.0 BRAIN-COMPUTER INTERFACE CONTROL ALONG INSTRUCTED PATHS

*Figures and text in this chapter have been submitted as* **Sadtler PT, Ryu SI, Tyler-Kabara EC, Yu BM, Batista AP.** Brain-computer interface control along instructed paths. *J Neural Eng. A previous version of this work was presented in (Sadtler et al., 2011).*

Previously, we examined whether there are constraints on learning at the neural level (Chapter 3), what neural adaptation mechanisms occur during learning (Chapter 4), and whether we can get better performance by incorporating FA into a decoder (Chapter 5). In this chapter, we wondered whether we could elicit better BCI performance by changing the task structure. It is important to have a solid understanding of all of the factors that affect the way in which a subject learns to control a BCI.

## 6.1 INTRODUCTION

BCIs aim to assist paralyzed people and amputees by translating neural activity into movement of a computer cursor or prosthetic limb. Researchers have demonstrated the potential of these systems in the laboratory using animal subjects. Monkeys have been trained to control computer cursors (Serruya et al., 2002; Taylor et al., 2002; Mulliken et al., 2008; Gilja et al., 2012), robotic limbs (Carmena et al., 2003; Velliste et al., 2008), electrical stimulation of their own muscles (Ethier et al., 2012), and electrical stimulation of their own spinal cord (Zimmermann and Jackson, 2014) with BCIs. Recently, human subjects have also controlled computer cursors (Kim et al., 2008; Schalk et al., 2008; Wang et al., 2013) and robotic limbs (Hochberg et al., 2012; Collinger et al., 2012) via BCI. In addition to providing quality-of-life benefits, BCIs can also be used to study the neural mechanisms of motor control and learning in healthy individuals (Jarosiewicz et al., 2008; Ganguly and Carmena, 2009; Golub et al., 2013; Sadtler et al., 2014).

The most common tasks used in BCI studies are *point-to-point* tasks, such as center-out (Taylor et al., 2002; Jarosiewicz et al., 2008; Mulliken et al., 2008; Kim et al., 2008; Schalk et al., 2008; Chase et al., 2009; Ganguly and Carmena, 2009; Gilja et al., 2012; Wang et al., 2013; Sadtler et al., 2014). In a point-to-point task, a subject is instructed to move the effector (i.e., computer cursor or robotic limb) from one location to another, and there are no constraints on the effector's trajectory. Consider the three cursor trajectories for a point-to-point task shown in Figure 6.1a. Although the trajectories are quite different from each other, all three trials would be deemed successful. However, since trajectory 1 most closely resembles the movement of the hand in a point-to-point task under arm control, it may most closely reflect the user's intention.

Common performance metrics for point-to-point tasks are success rate (i.e., the percentage of successful trials) and target acquisition time (i.e., the duration of time for the effector to move from one point to another) (Thompson et al., 2014). Other performance metrics evaluate the trajectory of the cursor, measuring properties such as movement variability (across trials) and straightness (within a trial) (Kim et al., 2008). To accelerate the development of BCIs from the lab to the clinic, we need more realistic tasks and richer performance metrics than point-to-point tasks can provide. Specifically, we propose the following three goals for BCI system design.

First, BCI tasks should be designed to extend the performance limits of current systems. One way in which this can be accomplished is by increasing the difficulty of a task in a parametric manner until the subject is no longer able to complete it successfully. With a point-to-point task, one can 1) reduce the amount of time allowed for driving the effector to the target, 2) increase the duration of time during which the subject must keep the effector on a target (the 'hold time'), 3) increase the distance between the targets, or 4) decrease the size of the targets. We sought to develop a task that allows for finer control of difficulty.

Second, we need to have a good understanding of the subject's intentions against which to evaluate the performance of a BCI. In a point-to-point task where there are few constraints

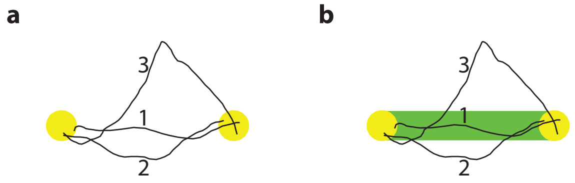

Figure 6.1: Raising performance standards in brain-computer interface paradigms. Cursor trajectories (black) from three trials in two tasks are shown. (a) Point-to-point task. All three trials would be successful since the cursor cursor moves from one target (yellow) to the other. (b) Instructed path task. The subject is required to keep the cursor within the path (green) while moving it from one target to the other. Only trial 1 would be successful in this task.

on the effector's trajectory, it is difficult to know whether or not any particular trajectory accurately reflects the subject's intent. The subject could intentionally drive the effector in an indirect path to the target (such as trajectory 3 in Figure 6.1a) — rather than drive the effector in a straight path (such as trajectory 1 in Figure 6.1a) — and still succeed at the task. We would not know whether the subject intended the indirect trajectory, or whether the subject intended a direct trajectory but could not control the cursor well enough to drive it in a straight line.

Third, BCIs should allow the user to demonstrate rich kinematics of the effector, resembling those of the arm in everyday life. A point-to-point task typically explores only a limited subset of kinematics (that is, the position and velocity of the effector). For example, consider a standard center-out task. For that task, if a subject has proficient control of the effector, position and velocity will be highly correlated (Morasso, 1981; Paninski et al., 2004). Ideally, we would like the subject to show diverse combinations of position and velocity while proficiently controlling the effector.

Previous BCI studies have proposed extensions to the center-out task to expand its capacities, such as requiring subjects to move a BCI cursor around visual barriers (Gilja et al., 2012) or to sequentially-placed targets (i.e., 'pinball task') (Serruya et al., 2002; Carmena et al., 2003; Hochberg et al., 2006; Wu et al., 2006; Suminski et al., 2010). Both of these tasks push the limits of the subject's control and increase the richness of kinematics relative to a center-out task, although it can be difficult to estimate the subject's true intentions in these tasks for the same reason as shown in Figure 6.1a. Another extension requires subjects to track a continuously moving target (i.e., 'pursuit tracking') (Hochberg et al., 2006; Wu et al., 2006). This extension helps to meet the aforementioned three goals and tests aspects of BCI control that are complementary to the work presented here.

Similar extensions have been applied to arm control tasks. For example, via-points (Hocherman and Wise, 1991) (i.e., requiring subjects to reach through a sequence of predefined targets), a pinball task (Suminski et al., 2010), and tasks with visual barriers (Churchland et al., 2012; Pearce and Moran, 2012) have been used. Other studies have used tasks where the target changes mid-reach (Georgopoulos et al., 1983; Archambault et al., 2009). Schwartz and colleagues introduced tasks with instructed paths wherein the subjects traced spirals and ellipses with their hands (Schwartz, 1994; Schwartz et al., 2004). All of these extensions increase the difficulty of the task and provide richer kinematics.

We introduce the *instructed path task* for BCI (Figure 6.2). This task specifies the path along which the subject should drive the cursor on each trial. It is particularly beneficial for BCI studies because it adds to the number of ways in which we can increase the difficulty of a task, which could lead to improved BCI performance. For example, in Figure 6.1b, the subject must keep the cursor on the green path. In order to maintain the same proficiency as with the point-to-point task, the subject must improve its performance as trajectories like 2 and 3 would be deemed unsuccessful. Additionally, the instructed path constrains the possibilities for the subject's true intentions. That is, we know the subject intends to move the cursor on the path instead of along any other unspecified route. This task can also be used to increase the richness of kinematics a subject exhibits.

Here, we present findings in two monkeys who performed the instructed path task under BCI control. We compare performance to a conventional point-to-point task under BCI control. We also compare performance in both tasks under arm control.

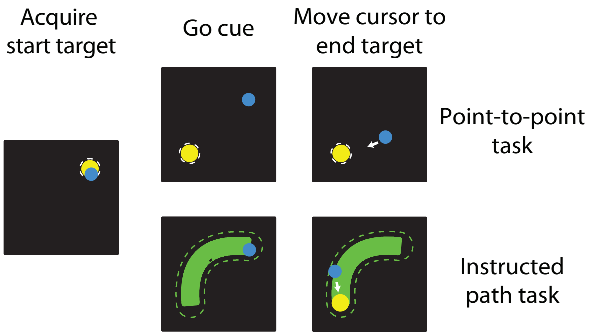

Figure 6.2: Instructed path task description. At the beginning of each trial, the monkey moved the cursor (blue circle) to acquire the start target (yellow circle). The go cue (appearance of the end target the for the point-to-point task, appearance of the path for the instructed path task) was then given, and the monkey was required to move the cursor to the end target (point-to-point task) or through the path (green; instructed path task). White dashed circle, tolerance zone around the target; green dashed line, tolerance zone around the instructed path.

## 6.2 METHODS AND DATA ANALYSIS

### 6.2.1 Subjects and tasks

Both monkey L and monkey J were trained the monkeys to perform two tasks under brain control and arm control: the instructed path task and, for comparison, a point-to-point task. Both tasks are depicted in Figure 6.2. Sections 6.2.1.1 and Sections 6.2.1.2 describe the tasks in general. The specific details of the tasks varied from session to session. Appendix A provides detailed descriptions of the tasks, and Table A1 outlines the parameters used in each session.

**6.2.1.1 Point-to-point task** The point-to-point task began with the appearance of a 'start target' (Figure 6.2a, top). The monkey was required to move the cursor to this target. For most sessions, there was no hold time requirement at the start target before the 'end target' appeared. For monkey J arm control, there was a hold time on the start target that was randomly chosen from the set {200, 450, 600, 750, 900, 1500} ms. The appearance of the end target signaled the 'go cue' to the monkey, at which time he was required to move the cursor to the end target. Once the cursor acquired the end target, the animal received a liquid reward. There was no hold requirement on the end target. The monkeys usually moved the cursor swiftly to the end target; if they did not do so, then after 5 s (monkey J) or 7.5 s (monkey L), the trial aborted, and a 1.5 s penalty period ensued.

**6.2.1.2 Instructed path task** The instructed path task also began with the appearance of a start target (Figure 6.2a, bottom). The start target hold time requirements were the same as for the point-to-point task. A path, which connected the start target to the location of the end target, appeared on the screen and signaled the go cue. The monkey was required to move the cursor through the path toward the end target. Once the cursor was 75% of the way through the path, the end target appeared (the path remained visible). When the monkey acquired the end target with the cursor, a liquid reward was administered. If the center of the cursor left the tolerance region around the path or if the monkey did not acquire the end target within the time limit (same limits as the point-to-point task), then the trial aborted and was followed by a timeout. For both monkeys, we used instructed paths of two types: straight paths and paths with one inflection. For monkey J, we also used paths that were U-shaped, had two inflections, or had three inflections. Hereafter, the 'target distance' refers to the straight-line distance between targets, regardless of the presence or type of the path. The 'path length' of the instructed path refers to the distance between targets along the centerline of the path.

### 6.2.2 Control modalities

The monkeys controlled the position of the cursor using one of two modalities during each session: arm control or brain control.

**6.2.2.1 Arm control** For sessions during which a monkey controlled the cursor with his arm, the position of the cursor was determined from the position of an LED on the monkey's hand. Only the horizontal and vertical components of the arm movements moved the cursor; arm movements toward or away from the monkey's body were not rendered on screen.

**6.2.2.2 Brain control** For some sessions, the number of kinematic parameters, $r$, was 6. In those sessions, the elements of $\hat{\mathbf{x}}_t$ corresponded to horizontal and vertical position, velocity, and acceleration of the cursor. At each timestep, the elements of $\hat{\mathbf{x}}_t$ corresponding to position determined the cursor's position on the screen. For other sessions, $r = 2$. The elements of $\hat{\mathbf{x}}_t$ corresponded to horizontal and vertical velocity. We integrated the velocity across timesteps to define the cursor position on the screen.

**6.2.2.3 Decoder calibration** Each brain control session began with a calibration block. We used the data recorded during that block to calibrate the parameters of the decoders.

*Monkey L* The calibration procedure for monkey L was similar to a method used by Chase et al. (2009). The procedure began with a modified version of the point-to-point task. For the first 16 trials of the calibration block, the monkey observed a cursor move with a constant velocity from the start target to the end target (i.e., the 'observation task'). We calibrated the parameters of an initial decoder from the neural activity and cursor kinematics on these trials (Tkach et al., 2008). For the next 8 trials, the monkey controlled the cursor with the decoder calibrated on the 16 observation trials. The cursor was placed on the start target at the beginning of each trial. We calibrated a decoder from these 8 trials, and the monkey controlled the cursor with that decoder for 8 more trials. We then calibrated a decoder from all 16 closed-loop trials. We repeated this process until the monkey had completed 48 closed-loop trials. The decoder that the monkey used for the remainder of the session was calibrated from data recorded on all 48 trials.

*Monkey J* Monkey J performed two different tasks for calibration, one per session. For some sessions, we used the instructed path task under arm control to calibrate the BCI decoder. The monkey completed 2 repetitions of 64 unique combinations of path shape and target location. We used four path shapes during calibration: single-inflection, double-inflection, triple-inflection, and U-shaped. We calibrated the decoding parameters using the arm kinematics and the recorded neural activity.

For other sessions for monkey J, the calibration block consisted entirely of the observation task. The monkey observed the cursor move for 80 trials, and we calibrated the parameters of the decoder from the observed cursor kinematics and recorded neural activity. The specific calibration methods and decoder formats are given in Table A1.

### 6.2.3 Quantification of performance

We quantified performance using several metrics. Two of the metrics (success rate and target acquisition time) are standard for BCI studies. Success rate is computed as the fraction of trials successfully completed, and acquisition time is the amount of time required by the monkey to drive the cursor to the target on successful trials. We use two other metrics (virtual success rate and histograms of kinematics) to highlight the benefits of the instructed path task.

**6.2.3.1 Virtual success rate** For a given set of point-to-point cursor trajectories, we asked how well the subject would have performed on a more difficult task. To assess this, we applied a 'virtual' path tolerance using a hypothetical straight instructed path and computed the percentage of trials that would have been successful if a path with that tolerance had been present. We termed this the virtual success rate.

**6.2.3.2 Significance testing** For all measures of success rate and virtual success rate, we combined each monkey's performance across many sessions. To compare the success rates of two different experimental conditions, we could not simply apply a t-test because the data are binary (i.e., each trial is deemed a success or failure). Instead, we adopted a Bayesian approach whereby each success or failure is considered to be the result of a coin toss, whose probability of heads (i.e., success) is $q$ (i.e., a Bernoulli process) (MacKay, 2003;

Santhanam et al., 2009). Since $q$ is unknown (in fact, it is what we desire to estimate), we set a uniform prior distribution on $q$ between 0 and 1. Having observed the distribution of successes and failures, we can then compute the posterior distribution on $q$, which is a beta distribution. The figures showing success rate or virtual success rate in Section 6.3 indicate the mean and the 95% confidence interval of the posterior distribution on $q$. We computed a separate posterior beta distribution for each experimental condition. To assess whether the success rate of one experimental condition ($q_1$) was significantly greater than that of another condition ($q_2$), we evaluated whether $P(q_1 > q_2)$ was larger than 0.95.

For acquisition time and normalized acquisition time, we also combined performance across sessions. Because those metrics are continuous-valued, we could compare their distributions directly. The histograms were not normally distributed (Kolmogorov-Smirnov test, $p = 0.05$), so we compared distributions using Wilcoxon rank-sum test with a significance threshold of $p = 0.05$.

## 6.3 RESULTS

We trained two monkeys to control the position of a cursor either using their neural activity ('brain control') or their hand ('arm control'). For both control modalities, there were two tasks. The point-to-point task required the monkeys to move the cursor from one target to another, and there were no constraints on the cursor's trajectory. The instructed path task required the monkeys to move the cursor along a specified path. We used two different shapes of instructed paths for both monkeys: straight and single-inflection.

Figure 6.3a shows representative cursor trajectories for both control modalities for the point-to-point task for monkey L. The cursor largely moves directly from the start target to the end target for brain control and arm control, but the movements are less direct for brain control. These cursor trajectories are qualitatively similar to those shown in other studies (e.g., Chase et al. (2009), Suminski et al. (2010), and Gilja et al. (2012)). The point-to-point cursor trajectories for monkey J to those for monkey L are similar (Figure 6.3b).

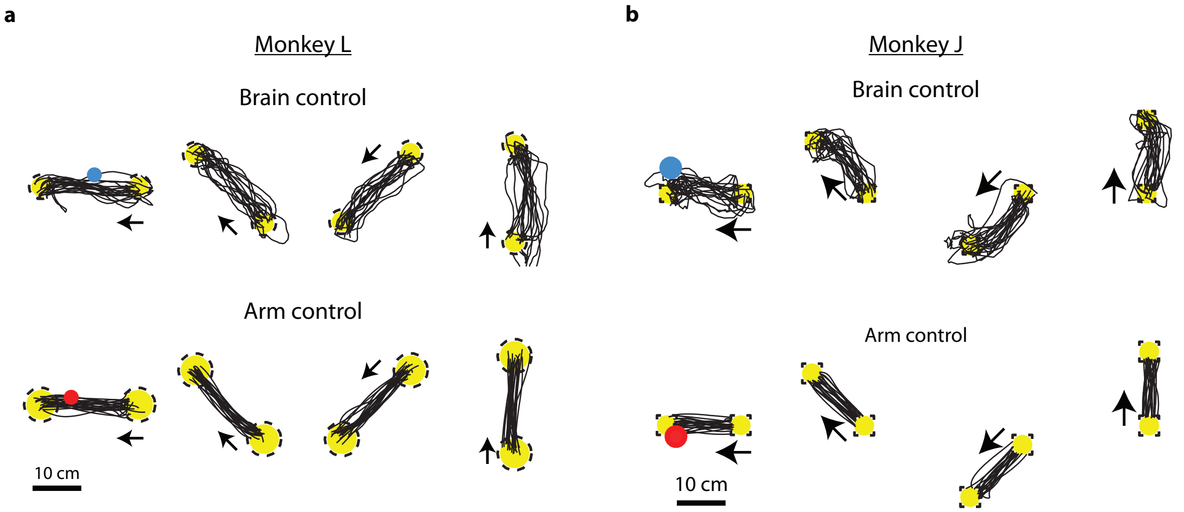

Figure 6.3: Representative point-to-point task cursor trajectories. Cursor trajectories for 4 of the 8 target configurations for the point-to-point task under brain control (top) and arm control (bottom). Yellow circles, start and end targets; arrows, direction of cursor movement (i.e., from start target to end target); dashed lines, target tolerance zone; blue circle, brain-controlled cursor; red circle, arm-controlled cursor; black lines, cursor trajectories. (a) Cursor trajectories from monkey L. (b) Cursor trajectories from monkey J.

We implemented the instructed path task push the limits of BCI systems beyond what point-to-point tasks can provide. Figure 6.4 shows representative cursor trajectories for the instructed path task for both control modalities (brain control and arm control) and both shapes of instructed paths (straight and single-inflection) for monkey L. In the point-to-point task, the cursor was allowed to take a circuitous route to arrive at the target. The instructed path task is more challenging in that the cursor needs to follow a prescribed path.

Figure 6.4a highlights monkey L's strategy for the curved trajectories. Presumably, this animal attempted to acquire the end target as quickly as possible by driving the cursor closer to the inside edge of the inflection. The instructed path cursor trajectories for monkey J are similar to those for monkey L, except that for both control modalities, the single-inflection cursor trajectories are not as close to the inside edge of the inflection as for monkey L (Figure 6.4b).

### 6.3.1 Success rate

To quantify performance, we measured the success rates for the different tasks. Success rates for monkey L are shown in Figure 6.5a. The brain-control success rates are nearly as large as the success rates for arm control (point-to-point task: 7.9 percentage point difference; straight instructed paths: 16.9 percentage point difference; single-inflection instructed paths: 23.5 percentage point difference).

We expected that as we increased the task difficulty, the success rate would decrease. The straight instructed path is more difficult than the point-to-point task because of the added constraint to keep the cursor within a small tolerance zone around the instructed path. Furthermore, the single-inflection instructed path is more difficult than the straight instructed path because of its curvature. Thus, we would expect that the success rates would be highest for point-to-point, lower for straight instructed paths, and lowest for single-inflection instructed paths. This is indeed what we found. We also found that the difference in success rates between arm control and brain control increased as the as the difficulty of the task increased. This shows that more challenging tasks can reveal deficiencies in BCI

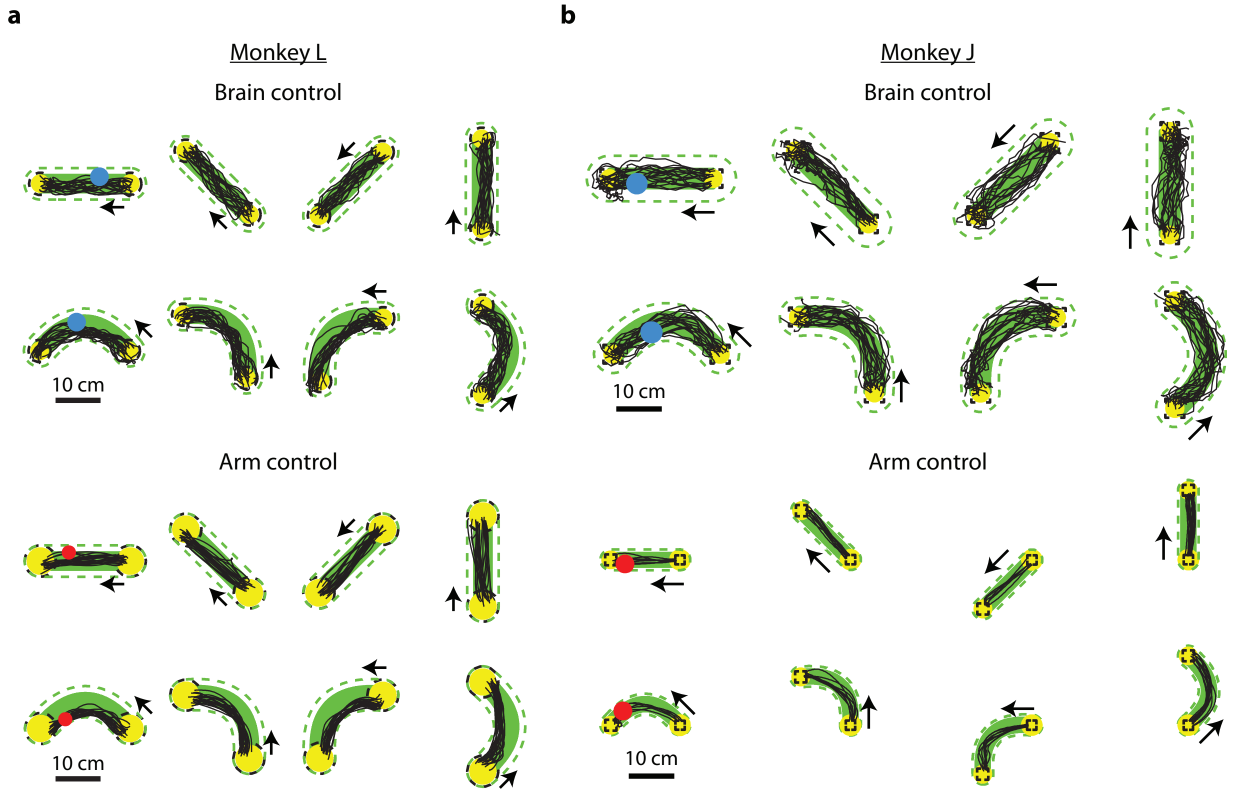

Figure 6.4: Representative instructed path task cursor trajectories. Brain control (top) and arm control (bottom) cursor trajectories for 4 of the 8 straight path configurations and 4 of the 16 single-inflection path configurations. Same format as Figure 6.3. Green dashed line: path tolerance zone. (a) Cursor trajectories from monkey L. (b) Cursor trajectories from monkey J.

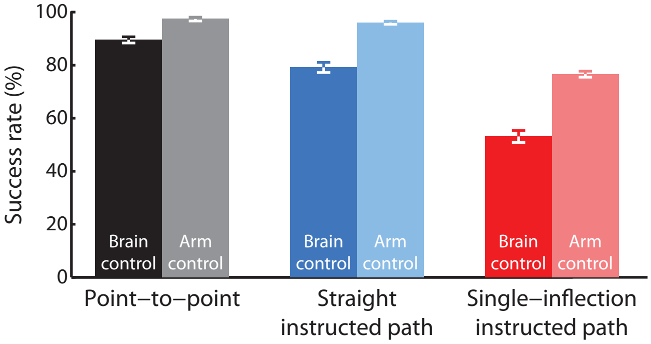

Figure 6.5: Success rate comparison: monkey L. Each bar shows the percentage of successful trials for a given task and control modality across all sessions. Error bars are the 95% confidence interval defined by the Bernoulli process. The success rates of all pairs of conditions shown in the figure are significantly different at the 95% level. Point-to-point: brain control, $n = 2660$ trials; arm control, $n = 1859$. Straight instructed paths: brain control, $n = 1723$; arm control, $n = 4259$. Single-inflection instructed paths: brain control, $n = 1889$; arm control, $n = 5344$. Target-to-target distance is 20 cm for all tasks. Path tolerance radius is 3.5 cm for the instructed path tasks.

control that are not evident using a point-to-point task. The instructed path task requires finer control over the kinematics of the cursor than the point-to-point task.

Monkey J showed similar trends in success rates (Figure 6.6). For brain control, success rates were larger as the tolerance radius increased, with the highest success rate for the point-to-point task. The success rates for straight paths with 5 cm tolerance radius were not significantly different from each other. All other brain control success rates are significantly different from each other. For a given tolerance radius, we would expect the success rate to be greater for the straight instructed paths than for the single-inflection instructed paths. Here, the single-inflection success rate is between the straight path success rates due to differences in tolerance radius. For arm control, success rates were similar for all three task types. We believe this similarity was due to monkey J being motivated to perform better by the increased difficulty of the instructed path task.

### 6.3.2 Acquisition time

We measured the amount of time required to move the cursor from the start target to the end target. The acquisition times for monkey L are shown in Figure 6.7a. The median acquisition times were around 1 s for the brain control sessions, and the median arm control acquisition times were approximately 2 to 3 times shorter (point-to-point: 3.65 times shorter; straight instructed paths: 2.78 times shorter; single-inflection instructed paths: 1.87 times shorter). Note that the brain control cursor speed can be affected by choices made by the experimenter during calibration. For example, increasing the observed cursor speed during calibration can increase the speed of the cursor during closed-loop control at the expense of lower success rates (data not shown). This is a manifestation of the classic speed-accuracy trade-off (Fitts, 1954). We chose observed cursor speeds so that the monkeys could achieve reasonable success rates on both the point-to-point and instructed path tasks, and for each monkey, the observed speed was constant for all sessions (monkey L: 17.5 cm/s, monkey J: 15 cm/s).

When comparing the acquisition times across tasks, we need to take into account the path length. For a given pair of start and end targets, a single-inflection path is 1.2 times longer

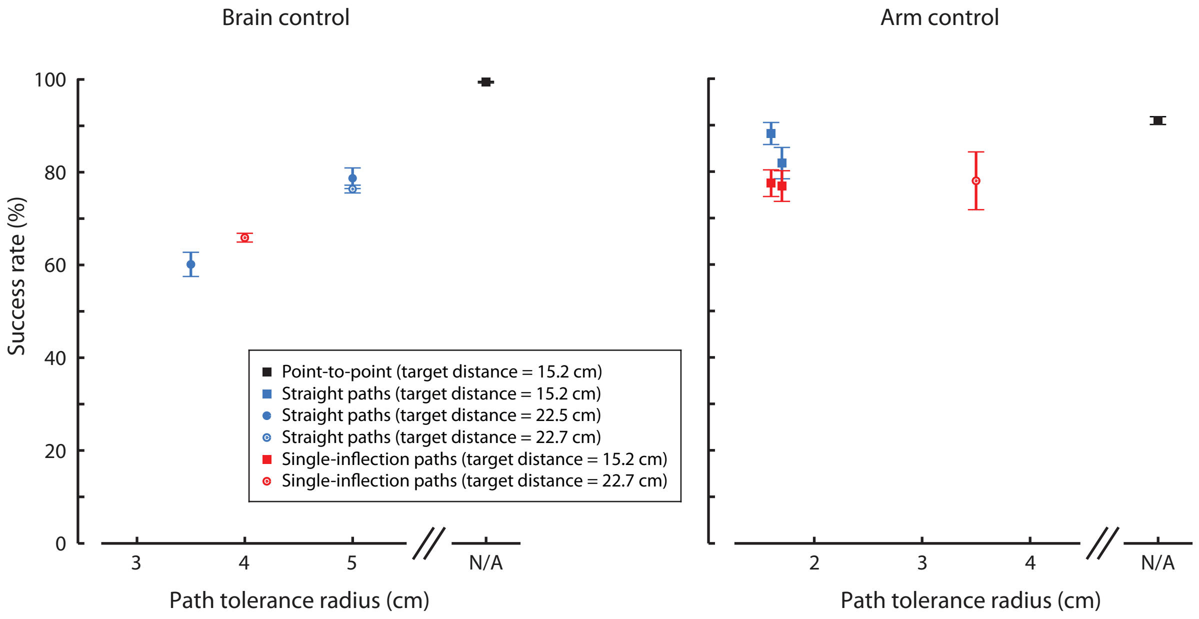

Figure 6.6: Success rate comparison: monkey J. Success rates for monkey J are presented in a different format than monkey L since there were a variety of path tolerance radii and target distances for that monkey. Each data point represents the total success rate for trials from all sessions with the given task type, tolerance radius, and target distance. Error bars represent the 95% confidence interval based on the Bernoulli process. (a) Success rate versus instructed path tolerance radius for brain control. Point-to-point: $n = 8670$ trials; Straight instructed paths, tolerance radius = 3.5 cm: $n = 1353$; Straight instructed paths, tolerance radius = 5 cm, $n = 9826$; Single-inflection instructed paths: $n = 9605$. (b) Success rate versus instructed path tolerance radius for arm control. Point-to-point: $n = 4491$ trials; Straight instructed paths, tolerance radius = 1.6 cm: $n = 713$; Straight instructed paths, tolerance radius = 1.7 cm: $n = 505$; Single-inflection instructed paths, tolerance radius = 1.6 cm: $n = 810$; Single-inflection instructed paths, tolerance radius = 1.7 cm: $n = 622$; Single-inflection instructed paths, tolerance radius = 3.5 cm: $n = 168$. All arm control single-inflection instructed path success rates are not significantly different from each other. Arm control success rate with straight instructed path and tolerance radius 1.7 cm was not significantly different from arm control success rate with single-inflection instructed path and tolerance radius 3.5 cm. All other arm control success rate comparisons were significantly different.

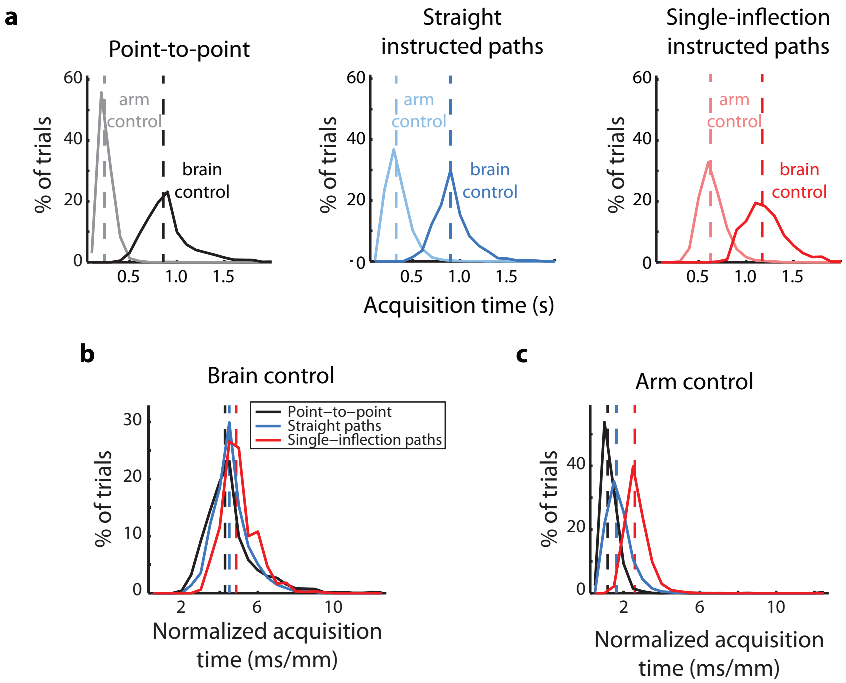

Figure 6.7: Acquisition times for monkey L. (a) Histograms of veridical acquisition times for the brain control and arm control tasks. Dashed lines: medians of distributions. Standard error across all trials are too small to be visible. (b) Histogram of normalized acquisition times for brain control. (c) Histogram of normalized acquisition times for arm control. Point-to-point: brain control, $n = 2363$ trials; arm control, $n = 1811$. Straight instructed paths: brain control, $n = 1363$; arm control, $n = 4089$. Single-inflection instructed paths: brain control, $n = 1001$; arm control, $n = 4093$. Acquisition times were computed only for successful trials. In each panel, all acquisition times (and normalized acquisition times) are significantly different from each other ($p < 0.05$, Wilcoxon rank-sum test).

than a straight path, and therefore, we expect the acquisition times to be longer for those trials. This is the case in both our brain control and arm control results. To enable further comparison across tasks, we normalized the acquisition times by the distance between start and end targets (point-to-point task) or the distance along the instructed path (instructed path task). Figures 6.7b,c and show the normalized acquisition times for brain control and arm control, respectively.

The differences between the normalized acquisition times for brain control are smaller than they are for arm control. This suggests that subjects are less able to modulate the speed of their movement under brain control than they can in arm control, as has been reported previously (Kim et al., 2008; Golub et al., 2014).

We found that the acquisition times (veridical and normalized) were larger for the straight instructed paths than for the point-to-point task for both control modalities, despite having the same distance between the targets. The additional constraints of the path may have influenced the monkeys to attempt to move the cursor more slowly in order to stay within its bounds.

For both brain control and arm control, the longest normalized acquisition times were for the single-inflection instructed paths. For arm control, this indicates that the monkey moved his arm slower when making a curved trajectory, as has been reported previously (Lacquaniti et al., 1983). This is likely because the monkey needed to change the momentum of his arm, which required him to make slower movements. For brain control, perhaps the monkey attempted to modulate cursor speed to emulate arm control.

For monkey J (Figure 6.8), it is more difficult to make these comparisons because we used different target distances and path tolerances. At first glance, the trends for monkey J look different from those for monkey, in that it appears he performed better with curved paths than with straight paths. However, most of these differences can be explained by differences in target distance and path tolerances used across experiments for monkey J. We did note that monkey J appeared to be more motivated by more difficult tasks (such as curved paths) than monkey L.

Because the point-to-point target distance is substantially shorter than that for the straight instructed path trials, it is expected that the normalized acquisition time is longer for the point-to-point trials. This is because peak cursor speed increases as distance between targets increases (Churchland et al., 2006). Interestingly, the normalized acquisition time for single-inflection instructed paths was lower than that for straight paths. This is likely due to monkey J being motivated by the increased difficulty.

Similar to monkey L, normalized acquisition times increase as task difficulty increases for a given target distance (point-to-point < straight instructed paths < single-inflection instructed paths). Normalized acquisition time decreased for single-inflection paths as the target distance and tolerance radius increased.

### 6.3.3 Virtual success rate

For brain control point-to-point tasks, there can be substantial variability in the cursor trajectory across trials. We wondered how much of that variability is inevitable due to stochastic spiking of neurons and how much of it the subjects could control, albeit with perhaps greater effort. To address this, we asked whether the subjects decreased the variability of the cursor movements in the face of the more demanding instructed path task. We directly compared the performance on the instructed path task with straight paths to the performance on the point-to-point task. The path length for the instructed path task is the same as the target distance for the point-to-point task, which enabled a direct comparison. *Post hoc*, we measured the percentage of point-to-point task trials that would have been successful for the path tolerance radius applied online during the instructed path trials. Figure 6.1a showed three example cursor trajectories for the point-to-point task. Trial 1 would have been deemed successful in this analysis, but trials 2 and 3 would have been deemed unsuccessful, since the cursor left the tolerance region.

Figure 6.9 compares monkey L's point-to-point virtual success rates to the actual straight path success rates for both brain control and arm control. The fraction of trials that were successful for brain control was significantly greater for the instructed path task than the fraction that would have been successful for the point-to-point task. This indicates that the presence of the instructed path did indeed influence this monkey to reduce the cursor variability. For arm control, the virtual success rate for the point-to-point task are not

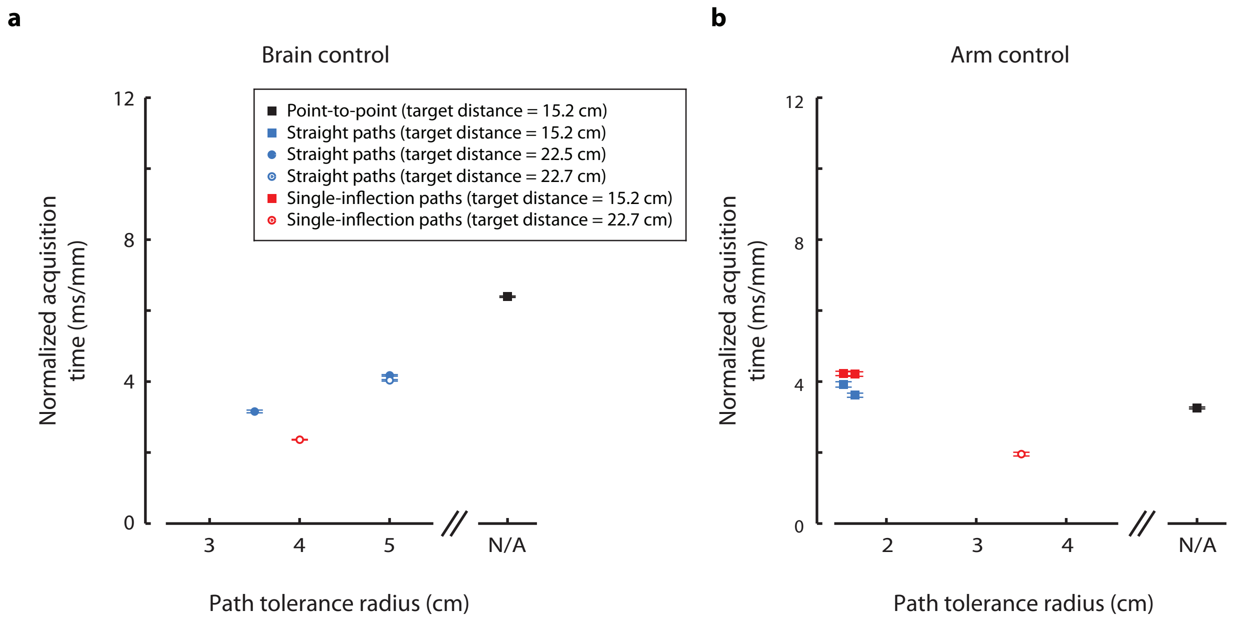

Figure 6.8: Normalized acquisition times for monkey J. Each data point represents the average normalized acquisition time for successful trials from all sessions with the given task type, tolerance radius, and distance between start and end target. Error bars represent mean +/- standard error. (a) Brain control normalized acquisition times. Normalized acquisition times for straight instructed path (target distance = 22.5 cm) and straight instructed path (target distance = 22.7 cm) are not statistically different ($p < 0.05$, Wilcoxon rank-sum test). All other comparisons are significantly different. Point-to-point: $n = 8623$ trials; Straight instructed paths, tolerance radius = 3.5 cm, target distance = 22.5 cm: $n = 813$; Straight instructed paths, tolerance radius = 5 cm, target distance = 22.5 cm: $n = 6690$; Straight instructed paths, tolerance radius = 5 cm, target distance = 22.7 cm: $n = 839$; Single-inflection instructed paths: $n = 6631$. (b) Arm control normalized acquisition times. All normalized acquisition time comparisons were significantly different ($p < 0.05$, Wilcoxon rank-sum test) except for the following pairs: straight instructed paths (radius = 1.6 cm) and straight instructed paths (radius = 1.7 cm), straight instructed paths (radius = 1.6 cm). Point-to-point: $n = 4087$ trials; Straight instructed paths, tolerance radius = 1.6 cm: $n = 628$; Straight instructed paths, tolerance radius = 1.7 cm: $n = 478$; Straight instructed paths, tolerance radius = 3.5 cm: $n = 131$; Single-inflection instructed paths, tolerance radius = 1.6 cm: $n = 629$; Single-inflection instructed paths, tolerance radius = 1.7 cm: $n = 412$.

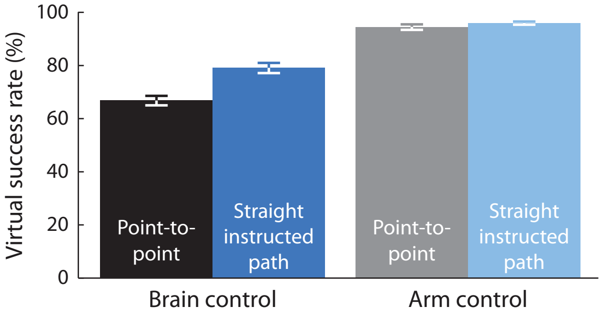

Figure 6.9: Virtual success rates for monkey L. Virtual success rates for brain control and arm control. Each bar shows the percentage of trials (mean and 95% confidence interval based on the Bernoulli process) which would have succeeded for a given tolerance radius for brain control and arm control. Virtual success rates for brain control point-to-point and straight paths are significantly different. Target-to-target distance: 20 cm. Point-to-point: brain control, $n = 2660$ trials; arm control, $n = 1859$. Straight instructed paths: brain control, $n = 1723$; arm control, $n = 4259$.

significantly different than the success rate for the straight instructed paths. This indicates that for arm movements, the presence of the path did not influence the monkey to tighten his movement variability (Figure 6.3b), although it is possible that an even smaller tolerance radius might have done so. We infer from this that it is relatively easier to reach with consistency than it is to move a BCI cursor with consistency. However, with the proper incentives, BCI movement variability can be volitionally reduced.

In Figure 6.10, we extend this analysis. Instead of computing the virtual success rates only at the path tolerance radius applied online, we computed the virtual success rates at a range of path tolerance radii. The results are consistent across a broad range of virtual path tolerance radii: BCI performance would have dropped if the path tolerance radius had been smaller, but arm control performance would be far less affected. We were not able to conduct the virtual success rate analysis for monkey J because there were not enough trials with matched target distances and path tolerance radii.

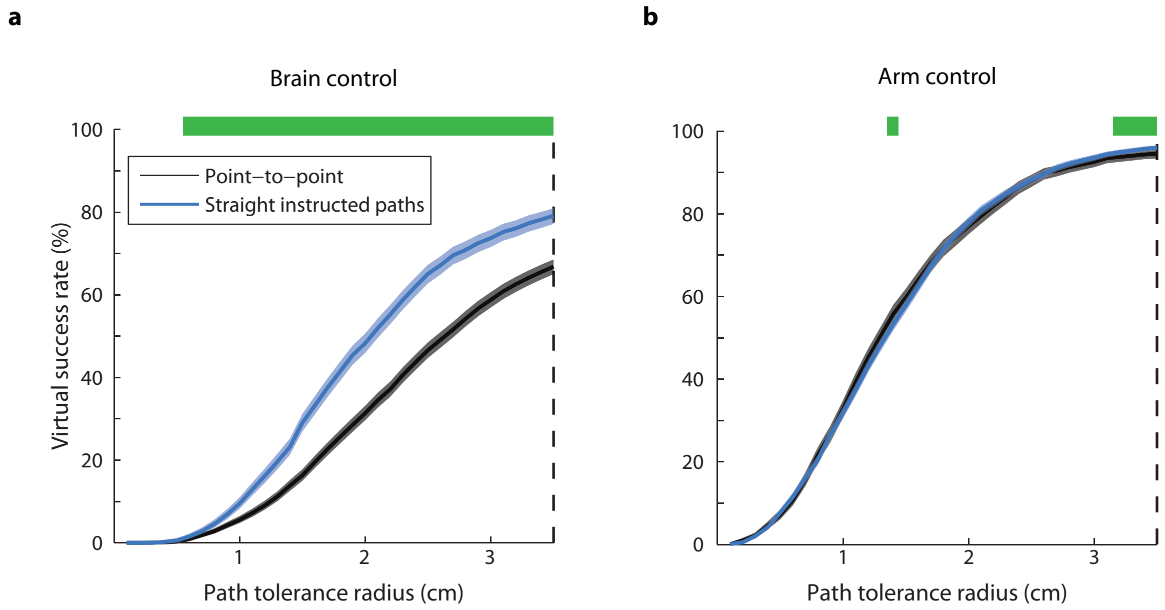

Figure 6.10: Extended virtual success rates for monkey L. (a) Virtual success rates across a wide range of virtual path tolerance radii for brain control. The line is the mean virtual success rate, and the band is the 95% confidence interval based on the Bernoulli process. Dashed line, tolerance radius that was applied online. Green bar, tolerance radii for which the virtual success rate for straight instructed path trials was significantly different than that for point-to-point trials. Target-to-target distance: 20 cm. Point-to-point, $n$ = 2660 trials; Straight instructed paths, $n$ = 1723. (b) Virtual success rates for arm control. Same plotting format as panel a. Point-to-point, $n$ = 1859; Straight instructed paths, $n$ = 4259.

### 6.3.4 Richness of kinematics

A benefit we propose of the instructed path task is that it will allow us to extract richer volitionally-controlled kinematics of the movement. To demonstrate this, we measured the instantaneous position and velocity of the cursor at every timestep and generated 2D histograms of the kinematic elements. These histograms for arm control are shown in Figure 6.11a. As expected, the kinematics are similar for the point-to-point task (first column) and the straight instructed path task (second column). However, single-inflection instructed paths (third column) provide a different set of kinematics. Combining the two types of instructed paths (fourth column) provides a larger range of kinematics than the point-to-point task or either type of instructed path alone. By designing additional instructed paths, one could continue to increase the richness of the kinematics. This could be helpful, for example, in basic-science characterizations of neural function in M1 (Paninski et al., 2004).

For brain control , the kinematic richness in velocity is similar across the different task types (Figure 6.11b). This is consistent with previous studies showing that there is limited speed modulation in BCI control (Kim et al., 2008; Golub et al., 2014), highlighting one target for improvement in future BCI systems. Note that because we conducted this analysis only on successful trials, the kinematic distribution in position is, by definition, different for straight path and single-inflection paths.

### 6.3.5 More challenging instructed paths

We trained monkey J to perform double-inflection instructed paths with both brain control and arm control, U-shaped instructed paths with arm control, and triple-inflection paths with arm control. Figure 6.12 shows cursor trajectories for representative trials with these types of instructed paths.

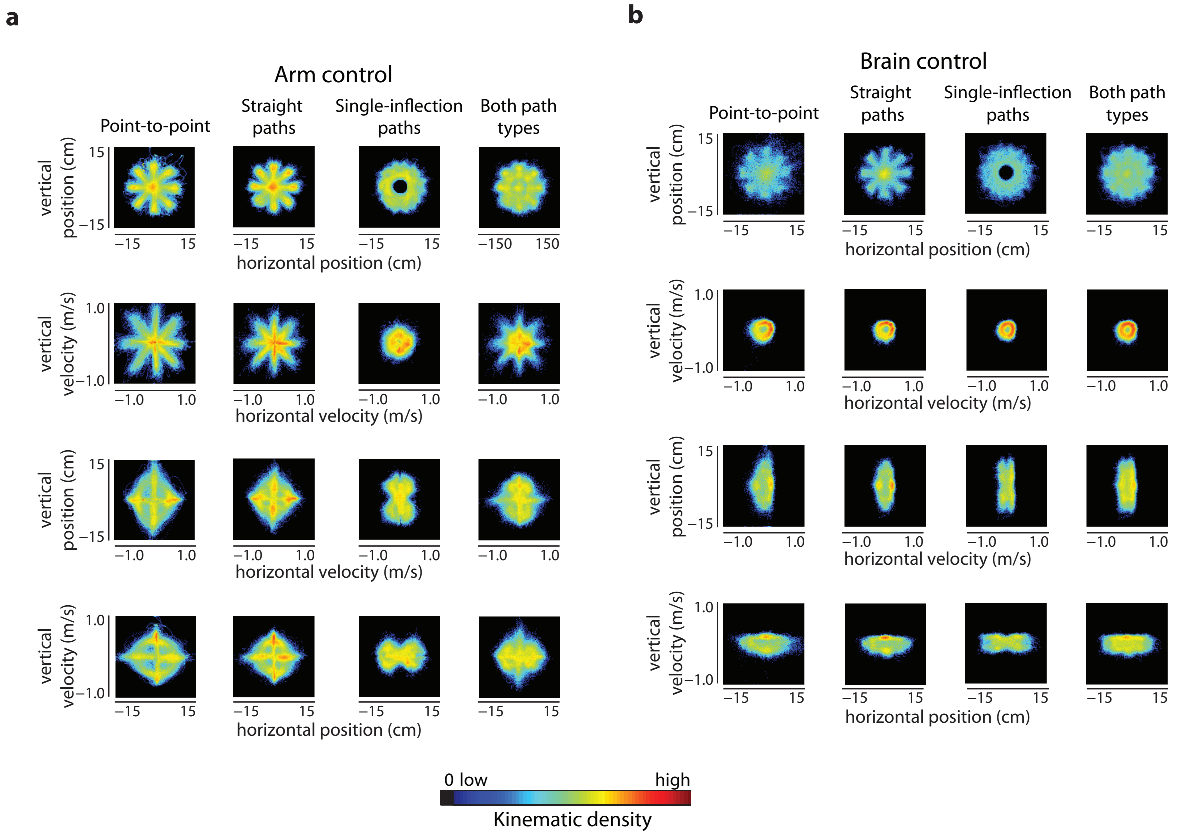

Figure 6.11: Instructed path kinematic histograms. Kinematic histograms for different task types for monkey L arm control. (a) Four pairwise combinations of kinematic distributions for monkey L for the arm control point-to-point task (first column), straight instructed paths (second column), single-inflection instructed paths (third column), and both instructed path tasks combined (fourth column). Number of data points in each panel: 102,064. Each data point corresponds to one time bin in one trial. (b) Four pairwise combinations of kinematic histograms for monkey L brain control. Same format as a.

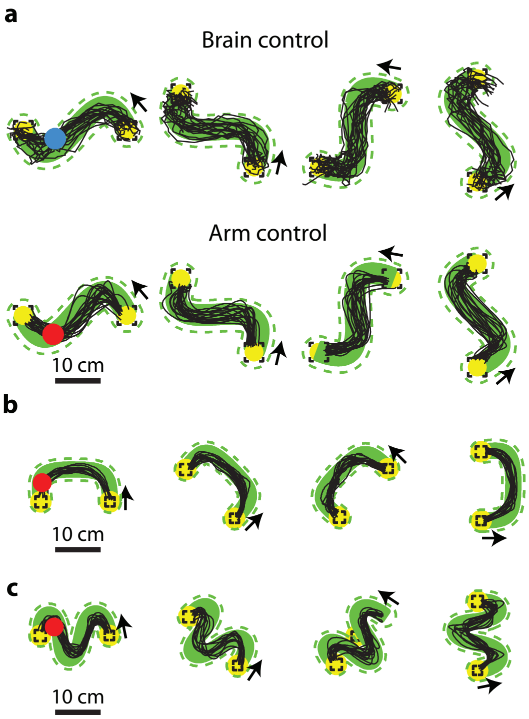

Figure 6.12: Cursor trajectories from additional instructed path types for monkey J. (a) Double-inflection instructed paths. Top row, brain control. Bottom row, arm control. Target distance: 22.5 cm. (b) U-shaped instructed paths with arm control. Target distance: 15.2 cm. (c) Triple-inflection paths with arm control. Target distance: 15.2 cm. Same plotting format as Figure 6.4.

## 6.4 DISCUSSION

Here, we demonstrated for the first time that monkeys can steer a BCI cursor along instructed paths. The instructed path task provides experimenters with more ways in which to change task difficulty compared to a standard point-to-point task. It also allows us to know a BCI user's motor intentions more precisely, and it provides an environment for eliciting rich kinematics (similar to those achieved naturally with the arm) with a BCI. As revealed by our virtual success rate analysis, the instructed path task led to better (i.e., more accurate and consistent) BCI control than in a traditional point-to-point task.

BCIs have shown promise for improving the quality of life for paralyzed individuals, but BCIs cannot currently duplicate the richness and deftness of natural reaches. Recent work is closing the gap between BCI control and natural reaches (e.g., Gilja et al. (2012) and Golub et al. (2014)), but further progress will be on many fronts. Among them is the need for more challenging tasks. The instructed path task is an important step in that direction. Other task alternatives to the point-to-point paradigm have been used. Especially relevant to our work is the pursuit tracking task (Hochberg et al., 2006; Wu et al., 2006), in which the subject must track a continuously-moving target with the BCI cursor. Both pursuit-tracking and instructed paths have some real-world relevance. In everyday life, we often make tracking movements (such as turning a steering wheel) and movements where the trajectory can be planned before the movement begins (such as reaching for a glass of water on a cluttered table). A pursuit tracking task has the advantage of allowing us to directly study the extent to which BCI users can effectively control cursor speed. In the instructed path task, speed was manipulated only indirectly by changing the curvature of the path. An advantage of the instructed path task is that the entire movement can be per-planned, and the neural activity during the plan stage could be decoded to improve BCI control (Yu et al., 2007).

In this work, we focused on virtual BCI effectors since we can fully specify their dynamics (Gowda et al., 2014). Some researchers are developing BCIs to control robotic limbs (Carmena et al., 2003; Velliste et al., 2008; Hochberg et al., 2012; Collinger et al., 2012) and to restore function to the individual's own arm (Ethier et al., 2012; Zimmermann and Jackson, 2014) for reaching and grasping. The instructed path task could extend naturally to three dimensions to help raise the performance standard in those studies as well by prescribing the path through which the robotic limb or arm should reach.

An interesting secondary outcome of these experiments is that the two animals we studied showed individual differences (Pearce and Moran, 2012). Although both could complete the main tasks about equally well, one (monkey J) may have been more motivated by the challenging tasks, and as a result, we were able to train him to follow more complex paths. Another difference between the monkeys is the type of calibration procedure that we used. In our initial experiments with monkey L (data not included here), we attempted to calibrate decoders using only the observation based procedure like we did for monkey J. Doing so, we were unable to calibrate decoders that the monkey could use, and as a result, we switched to the gradual calibration procedure. We can speculate that monkey L's cognitive processing of the observation task was different from that of monkey J, but we cannot rule out the possibility that neurons in monkey L's motor cortex other than the ones we recorded may have been activated by the observation procedure. Nevertheless, the differences between the two monkeys highlight the need to tailor BCI systems to suit each user.

In order for BCIs to be more useful, we also need to develop BCI systems that can exhibit all the rich kinematic diversity of real reaches and tasks for which kinematic diversity is essential for proficient performance. Other important targets for continued improvement in BCIs include decoding algorithms and recording technologies. Recent innovations in decoding algorithms have raised BCI performance considerably (e.g., Li et al. (2011), Gilja et al. (2012), and Orsborn et al. (2012)), and increasing the number of recorded neurons might further improve performance (Schwarz et al., 2014). Combining innovative tasks with algorithmic and technological advances is likely to accelerate the development of BCIs, enabling the devices to transition from the lab to the clinic more quickly.

# 7.0 CONCLUSION

## 7.1 SUMMARY

The general theme of the work presented in this dissertation has been to understand how learning occurs within the brain and more specifically how we can utilize learning to improve brain-computer interface performance. One of the main goals of BCI technology is to improve the quality of life for people who have limb-loss or people afflicted with paralysis or other neurological disorder. For a BCI to be truly effective, it should be easy for the person to use, and it should not require much intervention from a third party. To that end, it is important that we understand the neurological processes that underlie usage of a BCI. Are people able to use a BCI with an arbitrary mapping between neural activity and effector control? What are the processes that underlie learning in a BCI context? How can we design BCI mappings that best exploit the subject's innate learning strategies? In order to develop BCIs for the clinic, much testing has been done and needs to continue to be done with animal subjects. It is important that we motivate the animals to demonstrate their best possible performance. This will help us to find weaknesses in the current implementation of BCIs. Can we design tasks that will motivate animals to demonstrate their best possible performance?

In Chapter 3, we tested the extent to which a subject could learn to use an arbitrary mapping between neural activity and cursor movement. We began each session by identifying the 'intrinsic manifold', which is the low-dimensional space within the high-dimensional neural space that captures the most prominent patterns of co-modulation among the recorded neurons (Figure 3.1). The monkeys then controlled the BCI cursor using an intuitive mapping that resided within the intrinsic manifold. In the middle of each session, we perturbed the mapping and observed whether the monkeys could learn to regain control of the cursor.

On some sessions, we perturbed the mapping within the manifold. That is, we reoriented the control space of the mapping so that it was different from the intuitive mapping but remained within the intrinsic manifold. On other sessions, we reoriented the control space of the mapping so that it was no longer within the intrinsic manifold. We found that the monkeys were better able to learn to regain control of the cursor on sessions when the control space was perturbed within the intrinsic manifold. This indicates that the monkeys could easily learn to recombine their natural neural activity patterns in new ways but that it was more difficult for them to learn to generate new co-modulation patterns among the neurons.

In Chapter 4, we investigated the means by which the animals learned to control novel decoders. We found that the biggest changes in neural activity occur just after we switch from one BCI mapping to another. By projecting neural activity onto two planes of interest (the control plane and the null plane), we could decompose variability into its signal (i.e., variance across targets) and noise (i.e., variance within targets) components. Consistent with our intuition, as the signal increased in the control plane, the monkeys showed greater learning. Interestingly, the converse was also true. As signal increased in the null plane, learning decreased. In addition to analyzing neural activity as two-dimensional projections of the full neural space, we also analyze neural trajectories in the full neural space. We found that as the the trial-to-trial variability of the neural trajectories decreased, the monkey's performance increased.

In Chapter 5, we tested ways in which we could design decoders in order to elicit the best possible performance from animal subjects. Specifically, we incorporated the dimensionality reduction technique factor analysis into a Kalman filter. A standard Kalman filter maps neural activity to the effector kinematics (Wu et al., 2006). In our modified Kalman filter, we first reduced the dimensionality of neural activity to a set of latent factors, and then we mapped the latent factors directly to the kinematics of the cursor. We hypothesized that the dimensionality reduction step would extract latent dimensions that captured underlying high-level control signals in the neural activity that gave rise to the recorded spike counts. We were interested in finding the number of latent dimensions that would lead to the best performance. As we increased the number of latent dimensions in the FA model, we found that the monkey's performance also increased. However, performance with the standard

Kalman filter was not substantially different from the peak performance with the modified Kalman filter. This indicates that if there are low-dimensional control signals present in the neural activity, the standard Kalman filter does a sufficient job of identifying those to move the cursor.

Finally, in Chapter 6, we explored whether we could motivate monkeys to demonstrate better performance with a BCI by changing the requirements of the task. We used the 'instructed path task' for the first time in a BCI setting. In the instructed path task, the monkeys were required to move the cursor along a path that was displayed to them. For both monkeys, we tested their performance on straight instructed paths and single-inflection paths, while one of the monkeys was able to control a cursor on even more complicated paths. To enable comparison, we also trained both monkeys on a point-to-point task that was akin to center-out. We compared each monkey's performance on the point-to-point task to his performance on the straight instructed path task. We found that the presence of the instructed path motivated one of the monkeys to demonstrate better performance relative to the point-to-point task. The other monkey demonstrated the same level of performance in both tasks. From this, we concluded that there is a need to tailor BCIs to be specific to each individual.

## 7.2 FINAL THOUGHTS AND FUTURE DIRECTIONS

In some of this work, we utilized dimensionality reduction in order to gain insights into the biological processes in the brain. In Chapter 3, we used factor analysis to identify an intrinsic manifold, which captured the prominent patterns of co-modulation among the neurons, and we found that it was difficult for the neurons to develop new co-modulation patterns within a short period of time. In Chapter 5, we incorporated factor analysis into a Kalman filter to try to find the latent dimensionality that would lead to best BCI performance. Both of these studies are essentially asking a similar question about the brain. That is, what is the intrinsic dimensionality within a small network of neurons? Or stated another way, how many true degrees of freedom are in a small network of neurons? While the intrinsic dimensionality likely depends on a number of parameters, such as the monkey's skill level, the task the monkey is doing, and the dimensionality reduction technique being used among others, we can still ask what impact the intrinsic dimensionality has on BCI control. In the work described in this dissertation, the monkeys only controlled a BCI cursor in two dimensions. Is the number of degrees of freedom of effector control limited by the intrinsic dimensionality of the network of neurons? Other studies have shown that monkeys and humans can control robotic limbs that have five (Velliste et al., 2008) or seven (Collinger et al., 2012; Hochberg et al., 2012) degrees of freedom. This leads us to wonder whether the maximum number of degrees of freedom in an effector is capped by the intrinsic dimensionality of the neurons being recorded.

Along those same lines, we might wonder whether the intrinsic dimensionality of a network of neurons is flexible. Can we train a subject to reliably show more co-modulation patterns, or is the intrinsic dimensionality somehow bounded or defined by the connections within the underlying network? In Chapter 3, we showed that within a short amount of time, it was more difficult for the monkeys to generate co-modulation patterns that were outside of the intrinsic manifold. But what would it take for subjects to be able to do so? One hypothesis is that the subjects just need more time. This agrees with a previous study which demonstrated a monkey learning to control a decoder with scrambled parameters across many days (Ganguly and Carmena, 2009). Another hypothesis is that we might need to guide the subjects away from their intrinsic manifold. For example, to use nomenclature from Chapter 3, if we gradually rotated the control surface of the mapping from the intuitive mapping's control surface to a control surface that is outside of the intrinsic manifold, then the subject might more easily learn to generate co-modulation patterns that are outside of the intrinsic manifold. If we did this, would the intrinsic dimensionality of the neurons increase since the neural activity is exploring new co-modulation patterns? Or would the intrinsic dimensionality remain the same since some previously explored co-modulation patterns are no longer being explored? From a practical standpoint, would this accelerate learning? Explorations of this issue, which build directly from my research, are currently underway in the lab.

This work has led me to wonder how a subject actually learns to control a BCI. Surely a subject cannot identify the neurons that are being recorded and used to control the effector. In fact, other work (Ganguly et al., 2011) has shown that neurons which do not directly control the effector are modulated as the subject moves the effector. As a subject initially learns to control a BCI, do they begin by using the same co-modulation patterns that they use when controlling their own arm? Fully understanding the differences between arm control and BCI control will be key in transitioning BCIs from the lab to the clinic. For example, we know that millions of cortical neurons are involved in controlling the actual arm. However, tens to hundreds of neurons are sufficient for controlling a robotic limb. Why are so many neurons involved in arm control? The native arm is more complex than the robotic limbs that have been used, but that does not fully account for the difference in the number of required neurons. Perhaps the brain is highly redundant so that it can overcome injuries which cause some neurons to die. Or perhaps a larger number of neurons are required in order to elicit smooth movements. We know that neurons are noisy – their spiking noise is often modeled as a Poisson process. By using the combined activity of millions of neurons, does the noise become averaged out which could lead to smooth movements? If we increase the number of neurons that we use for BCI control (Schwarz et al., 2014), even if that doesn't allow one to control more degrees of freedom, it might enable smoother movements of the effector due to the spiking noise being averaged out across neurons.

Understanding the neural processes that underlie BCI control and BCI learning will be key to continuing the development of BCIs from the lab to the clinic. As technological advances continue, we are able to record from more and more neurons simultaneously. This has the potential to revolutionize the field of neural prostheses by enabling swift and accurate control of a BCI. However, as we harness the benefits of the increase in data, it is important to also not lose sight of other important factors that might influence BCI control.

# APPENDIX A

# TASK SPECIFICS FOR POINT-TO-POINT AND INSTRUCTED PATH TASKS

This appendix describes the specific task details for the data in Chapter 6. For both the instructed path task and the point-to-point task, the targets were defined by two parameters: the visible radius and the tolerance radius. The visible radius is the radius of the target as it appears on the screen. The tolerance radius defines the tolerance region around the target. In order to 'acquire' a target, the center of the cursor needed to be within the tolerance region around the target. Visually, this sometimes corresponded to the edge of the cursor just touching the edge of the target, and sometimes the cursor being a few pixels within the target. All targets appeared as yellow circles on the screen. The start and end target visible radii ranged from 1.4 to 2.5 cm, and the tolerance radii ranged from 1 to 3.5 cm. The cursor radius ranged from 1 to 1.8 cm. The cursor appeared as a blue circle for brain control trials and as a red circle for arm control trials.

For some sessions, the start target appeared in the center of the workspace. For those trials, the end target was selected from 8 possible locations uniformly spaced on a circle around the start target. The distance from the start target to the end target was 15.2 cm. For other sessions, the start target appeared in 1 of 8 locations uniformly spaced on a circle around the center of the workspace. For those trials, the end target was located on the circle diametrically opposite the start target. The distance from the start target to the end target was 20 cm. For some brain control point-to-point sessions for monkey J, a start target did not appear. At the start of each of those trials, the cursor was placed in the center of the workspace at the same time as the end target appeared.

The instructed paths had a path tolerance radius. This radius defined the maximum distance the center of the cursor could be from any point along the center-line of the path. The path tolerance ranged from 1.6 to 5 cm. The width of the instructed path as it appeared on the screen was 8% of the straight-line distance from the start target to the end target. The path tolerance was chosen to be larger than the width, but otherwise independent of the width. For a given pair of start and end targets, there were two possible orientations of a single-inflection path (i.e., clockwise and counter-clockwise inflections). The specific target and task parameters that we used for each trial are listed in Table A1.

Table A1: Task types and decoding parameters for every session. Sessions are listed in chronological order. All distances and radii are in cm. [1]a = arm control, b = brain control. [2]pp = point-to-point task, ip = instructed path task. [3]This is the straight-line distance from the center of the start target to the center of the end target. [4]start target display radius, end target display radius. [5]start target tolerance radius, end target tolerance radius, path tolerance radius (if applicable); numbers in parentheses indicate different radii in a session. [6]m = modified Kalman filter, s = standard Kalman filter; number in parentheses is $p$; number after comma is $r$. If multiple decoders were used in a session, the decoder parameters are separated by a semi-colon. [7]a = arm control (monkey J, method 1), o = observation (monkey J, method 2), g = gradual procedure (monkey L).

| Monkey | Control type[1] | Task type[2] | Target distance[3] | Cursor radius | Target visible radius[4] | Target tolerance radius[5] | Decoder format[6] | Decoder calibration[7] | Number of Sessions |
|---|---|---|---|---|---|---|---|---|---|
| L | a | ip | 20 | 1 | 2.5, 2.5 | 3.5, 3.5, 3.5 | - | - | 10 |
| L | b | ip | 20 | 1 | 1.5, 1.5 | 2.5, 2.5, 3.5 | m(10), 2 | g | 10 |
| L | a | pp | 20 | 1 | 2.5, 2.5 | 3.5, 3.5, - | - | - | 7 |
| J | b | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 2, 3.5 | s, 6 | a | 5 |
| J | b | pp | 15.2 | 1.8 | -,2 | -, 2, - | m(10), 2 | o | 5 |
| L | b | pp | 20 | 1 | 1.5, 1.5 | 2.5, 2.5, - | m(10), 2 | g | 5 |
| J | a | ip | 15.2 | 1.5 | 1.8, .1 | 1.2, 1.2, (1.6, 1.7) | - | - | 4 |
| J | b | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 2, 5 | m(25), 6 | a | 4 |
| J | b | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 4, 4 | m(25), 6 | a | 4 |
| J | a | ip | 15.2 | 1.5 | 1.8, 1.4 | 1, 1.2, 1.7 | - | - | 3 |
| J | a | pp | 15.2 | 1.5 | 2, 1.5 | 2, 2, - | - | - | 3 |
| J | a | ip | 15.2 | 1.5 | 1.8, 1.4 | 1, 1.2, (1.6, 1.7) | - | - | 2 |
| J | a | ip | 15.2 | 1.5 | 1.8, 18 | 1, 1.2, (1.6, 1.7) | - | - | 2 |
| J | b | ip | 22.5 | 1.8 | 1.5, .1 | 2, 2, 5 | s, 6 | a | 2 |
| J | b | ip | 22.5 | 1.8 | 1.5, 1.5 | 2, 2, 5 | s, 6; m(25), 6 | a | 2 |
| J | b | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 4, 4 | s, 6 | a | 2 |
| J | b | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 2, 3.5 | m(25), 6 | a | 2 |
| J | a | ip | 15.2 | 1.5 | 1.8, 1.4 | 1, 1.2, (1.6, 1.7) | - | - | 1 |
| J | a | pp | 15.2 | 1.8 | 2, 1.5 | 2, 1.5, - | - | - | 1 |
| J | a | ip | 15.2 | 1.5 | 1.8, 1.4 | 1, 1.2, 1.6 | - | - | 1 |
| J | b | ip | 22.7 | 1.8 | 1.5, .1 | 2, 2, 3.5 | a |  | 1 |
| J | b | pp | 15.2 | 1.8 | 1.5, 1.5 | 1.7, 1.7, - | m(1), 6 | a | 1 |
| J | a | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 2, 3.5 | - | - | 1 |
| J | a | pp | 15.2 | 1.8 | 1.5, 1.5 | 2, 2, - | - | - | 1 |
| J | b | ip | 22.5 | 1.8 | 1.5, 1.5 | 2, 2, 5 | s, 6 | a | 1 |
| J | b | ip | 22.5 | 1.8 | 1.5, .1 | 2, 2, 5 | s, 6; m(25), 6 | a | 1 |
| J | b | ip | 22.5 | 1.8 | 1.5, 1.5 | 2, 2, 3.5 | s, 6; m(25), 6 | a | 1 |
| J | b | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 2, 4 | m(25), 6 | a | 1 |
| J | b | ip | 22.7 | 1.8 | 1.5, 1.5 | 2, 2, 3.5 | s, 6; m(25), 6 | a | 1 |

# APPENDIX B

# SOFTWARE TOOLS

I developed two software tools that enabled my own research projects as well as other projects in the lab. Below, I provide a brief description of the tools I developed.

## B.1 EXPERIMENTAL CONTROL SOFTWARE

When I joined the lab in August 2009, former students Dan Bacher and Chris Ayers along with Joe McFerron had developed the hardware and software for our experimental stations. As our experiments progressed and became more complicated, I wrote the software to enable many new features. Among all of the features I added, the following are the most prominent.

- **Online decoding:** I added a component to the software that allowed us to translate spikes as they are being recorded in real-time into cursor kinematics. I designed this feature so that many types of decoders (e.g., Kalman filter, population vector algorithm, Wiener filter, etc.) can easily be added.
- **Auto-monkey cursor:** For some of the studies in this dissertation, we calibrated the parameters of the decoders using data that was recorded while the monkeys observed a cursor automatically complete a center-out task. I implemented the feature that allowed us to automatically move a cursor, and wrote the code so that the experimenter can customize the velocity profile of the cursor.

- **Multiple targets per state:** Previously, the code was written such that we could only define one target per task state. This is sufficient for most studies, but some require multiple targets per state. For example, former student Jason Godlove used a task where he cued a monkey to change the direction in which he was reaching. If the monkey failed to do so, he would acquire the 'failure target', but if he succeeded, then he acquired the 'success target'. Each target led to a different subsequent state. I added the ability to define multiple targets in each state.
- **Cumulative hold times:** In some tasks, we require the subjects to hold the cursor within the bounds of a target for a duration of time. This is difficult for some of the monkeys to learn to do, so I added features that might help these monkeys learn to hold a cursor on a target. The software allows the experimenter to specify whether the cursor can leave and re-enter the hold target without failing the task. The experimenter can specify whether the hold time resets each time the target enters the hold target or whether the hold time cumulates across each period of time when the cursor is in the target.
- **Online analysis of performance:** When conducting experiments with a monkey, it is important to constantly monitor how well he performing at a task. If the monkey is performing well, then the experimenter might want to increase the difficulty. If the monkey is not performing well, then the experimenter might want to relax the difficulty. I added a feature to the software that allows the experimenter to monitor the subject's level of performance. This feature allows the experimenter to customize the metrics that they monitor. For example, I used this tool to monitor the monkeys' success rates and target acquisition times along with measures of the preferred directions of the activity on each electrode.
- **Batch processing of data:** In order to analyze the data that we record, we must first translate the data from binary files to a format that is readable in Matlab. I added a feature to our data translator that allows us to multiple datasets in one batch. This saves a significant amount of time because the experimenter doesn't have to manually translate each dataset separately.

- **Online processing of data:** To calibrate a decoder, we first collect a set of data. Then we translate and process the data to extract decoder parameters. This process can take a few minutes, during which time the monkey is not working. The monkey might become frustrated and not work as well for the remainder of the session. I implemented a feature in the data translator that allows us to translate data while are collecting it, rather than having to wait until after it is collected. This reduces the amount of downtime for the monkey.

## B.2 MATLAB-BASED SPIKE SORTER

One common requirement among electrophysiological studies is to spike sort the threshold crossing waveforms recorded on each electrode. One does this in order to sort the waveforms based on the neuron that generated them. This is a cumbersome and time-consuming process. Originally, we used spike sorting software developed by TDT. This software made the spike sorting process quite slow mainly because switching from one electrode to another often took a minute or more. I wrote a Matlab program to speed up the process. The increases in speed are due to the following features of the program.

1. When the spike sorter is first opened and a dataset is selected, the program extracts all of the waveforms from each electrode on all trials. A separate binary file is saved that contains the waveforms for each electrode. Matlab is able to read and write binary files very quickly, so switching from one electrode to another is not bogged down by saving and opening files. One drawback of this method is that it takes a substantial amount of time (a few minutes) to initially process the data to create the binary files. However, this is only required once per dataset.
2. Another increase in speed is due to the way in which spikes are plotting on the screen. During the course of an experiment, it is not uncommon to record 100,000 or more threshold crossing events on a single electrode. Plotting that many waveforms can take a long time. However, the resolution at which we record the waveform voltages is much greater than what we can visualize on a computer screen. Before plotting the waveforms, the

voltages are rounded so that the resolution of the voltages is the same as the resolution that we can visualize on a screen. By doing this, many of the waveforms overlapped. Considering all waveforms, the program extracts the unique pairs of voltages from consecutive samples in the waveforms. As a result of this, many fewer data points need to be plotted on the screen than if all data points from all waveforms were plotted. Thus, plotting does not take nearly as long.

The spike sorting program was written to analyze data specific to our data structures. However, the program could easily be extended to work with other data structures.

# APPENDIX C

# COLLABORATIVE WORK

In addition to the work described in the main chapters, I have also done work that contributed to two other publications. The first of these studies was led by Sagi Perel, and it compared multi-unit activity and local field potentials (LFPs) during an arm control task (Perel et al., 2013). The second study was led by Sonia Todorova, and it analyzes various spike sorting methods by comparing the quality of offline decoding performed with each method's resultant spikes (Todorova et al., 2014). Below, I provide a summary of both studies.

## C.1 COMPARING MULTI-UNIT ACTIVITY TO LFP

Previous studies have proposed using different types of neural signals to control BCIs. Single-unit activity – where the neural signals represent the action potential times of individual neurons, multi-unit activity – where the neural signals represent the times when combined activity of a few to many neurons crosses a user-defined threshold, and LFPs – where the neural signals represent the time-varying voltage within a small region of motor cortex – have all been considered for BCIs. Studies have examined the tuning properties of individual types of neural signals, but their results were contradictory. In this study, we compared the tuning of the different types of neural signals, which were all derived from the same source signal.

We found that multi-unit activity encodes arm movement direction and velocity better than low-frequency and high-frequency LFPs. High-frequency LFPs encode speed better than multi-unit activity, and low-frequency LFPs encode speed even better than high-frequency LFP. Furthermore, while multi-unit activity was more tuned to direction than LFP, multi-unit activity was more tuned fro speed than it was for direction. For this study, encoding was determined by regressing the neural activity – multi-unit activity or LFP recorded on one electrode – against the movement kinematics of the arm. My contributions to the work were to train the two monkeys and to collect the data as well as to assist in revising the manuscript.

## C.2 COMPARING DIFFERENT METHODS OF SPIKE SORTING

There are many different ways to process single-unit activity for use with brain-computer interfaces. We tested many of the different processing methods for their ability to accurately reconstruct arm velocity. The spike sorting methods compared in this study were: 1) unsorted – a single voltage threshold was applied to the raw signal on each electrode, 2) expert-sorted – an expert manually classifies threshold crossing waveforms according to their source neuron, 3) expert+hash – same as 2 but with the hash retained as an extra unit on each electrode, 4) amplitude-sort – waveforms amplitudes are clustered using a mixture of Gaussian distributions, 4) split-sort – waveforms separated into equal-sized clusters on each electrode, and 5) joint distribution of spike counts and waveform amplitudes. The last method is a novel methods proposed by this work.

We found that discarding the noise recorded on each electrode decreases decoding quality while spike-sorting improves decoding efficiency. Furthermore, decoding using the novel method was not always superior to the other methods. My contributions to this work were to collect data from one of the monkeys, to spike sort that data, and to help with manuscript preparation.

# BIBLIOGRAPHY

**Ajemian R, D'Ausilio A, Moorman H and Bizzi E**. A theory for how sensorimotor skills are learned and retained in noisy and nonstationary neural circuits. *Proc Natl Acad Sci U S A* 110(52): E5078–87. 2013.

**Ames K, Ryu S and Shenoy K**. Neural Dynamics of Reaching following Incorrect or Absent Motor Preparation. *Neuron* 81(2): 438–451. 2014.

**Archambault PS, Caminiti R and Battaglia-Mayer A**. Cortical mechanisms for online control of hand movement trajectory: the role of the posterior parietal cortex. *Cerebral Cortex* 19(12): 2848–64. 2009.

**Bishop W, Chestek CC, Gilja V, Nuyujukian P, Foster JD, Ryu SI, Shenoy KV and Yu BM**. Self-recalibrating classifiers for intracortical brain-computer interfaces. *J Neural Eng* 11(2): 026001. 2014.

**Bliss T and Gardner-Medwin A**. Long-lasting potentiation of synaptic transmission in the dentate area of the unanaesthetized rabbit following stimulation of the perforant path. *J Physiol* (232): 357–374. 1973.

**Boden MA**. Creativity and Artif. Intell. *Artif Intell* 103: 347–356. 1998.

**Brainard MS and Doupe AJ**. What songbirds teach us about learning. *Nature* 417(May): 351–360. 2002.

**Brasted P and Wise S**. Comparison of learning related Neuronal activity in the dorsal premotor cortex and striatum. *European J Neurosci* 19: 721–740. 2004.

**Braun D, Mehring C and Wolpert D**. Structure learning in action. *Behavioural Brain Res* 206(2): 157–65. 2010.

**Briggman KL, Abarbanel HDI and Kristan WB**. Optical imaging of Neuronal populations during decision-making. *Science* 307(5711): 896–901. 2005.

**Buonomano DV and Merzenich MM**. Cortical plasticity: from synapses to maps. *Annu Rev Neurosci* 21: 149–86. 1998.

**Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, Dimitrov DF, Patil PG, Henriquez CS and Nicolelis MaL**. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol* 13: 0000042. 2003.

**Chase SM, Schwartz AB and Kass RE**. Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Networks* 22: 1203–1312. 2009.

**Cherian A, Krucoff MO and Miller LE**. Motor cortical prediction of EMG: evidence that a kinetic brain-machine interface may be robust across altered movement dynamics. *J Neurophysiol* 106(2): 564–575. 2011.

**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(40): 10742–10750. 2007.

**Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, Ryu SI and Shenoy KV**. Neural population dynamics during reaching. *Nature* 487(7405): 51–56. 2012.

**Churchland MM, Santhanam G and Shenoy KV**. Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach. *J Neurophysiol* 96(6): 3130–3146. 2006.

**Clancy KB, Koralek AC, Costa RM, Feldman DE and Carmena JM**. Volitional modulation of optically recorded calcium signals during neuroprosthetic learning. *Nat Neurosci* 17(6): 807–809. 2014.

**Collinger JL, Wodlinger B, Downey JE, Wang W, Tyler-Kabara EC, Weber DJ, Mcmorland AJC, Velliste M, Boninger ML and Schwartz AB**. High-performance neuroprosthetic control by an individual with tetraplegia. *Lancet* 6736(12). 2012.

**Crair M and Malenka R**. A critical period for long-term potentiation at thalamocortical synapses. *Nature* 375: 325–328. 1995.

**Cunningham JP and Yu BM**. Dimensionality reduction for large-scale neural recordings. *Nat Neurosci* . 2014.

**Darainy M and Ostry DJ**. Muscle cocontraction following dynamics learning. *Exp Brain Res* 190(2): 153–163. 2008.

**Della-Maggiore V, Malfait N, Ostry DJ and Paus T**. Stimulation of the posterior parietal cortex interferes with arm trajectory adjustments during the learning of new dynamics. *J Neurosci* 24(44): 9971–9976. 2004.

**Dempster AP, Laird NM and Rubin DB**. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J R Stat Soc* 39. 1977.

**Desmurget M and Turner RS**. Motor sequences and the basal ganglia: kinematics, not habits. *J Neurosci* 30(22): 7685–7690. 2010.

**Dum RP and Strick PL**. Motor areas in the frontal lobe of the primate. *Physiology & Behavior* 77(4-5): 677–682. 2002.

**Durstewitz D, Vittoz NM, Floresco SB and Seamans JK**. Abrupt transitions between prefrontal neural ensemble states accompany behavioral transitions during rule learning. *Neuron* 66(3): 438–48. 2010.

**Ethier C, Oby ER, Bauman MJ and Miller LE**. Restoration of grasp following paralysis through brain-controlled stimulation of muscles. *Nature* pages 1–4. 2012.

**Everitt BS**. *An Introduction to Latent Variable Models*. Chapman & Hall, London. 1984.

**Fee MS and Scharff C**. The songbird as a model for the Generation and learning of complex sequential behaviors. *ILAR J* 51(4): 362–77. 2010.

**Fetz EE**. Operant Conditioning of Cortical Unit Activity. *Science* 163(3870): 955–958. 1969.

**Fetz EE and Finocchio DV**. Operant conditioning of specific patterns of neural & muscle activity. *Science* 174: 431–435. 1971.

**Fitts PM**. The information capacity of the human motor system in controlling the amplitude of movement. *J Exp Psychol-Gen* 121(3): 262–9. 1954.

**Flament D and Hore J**. Relations of motor cortex neural discharge to kinematics of passive and active elbow movements in the monkey. *J Neurophysiol* 60(4): 1268–1284. 1988.

**Frégnac Y, Burke J, Smith D and Friedlander M**. Temporal covariance of pre-and postsynaptic activity regulates functional connectivity in the visual cortex. *J Neurophysiol* 71(4): 1403–1421. 1994.

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

**Ganguly K, Dimitrov DF, Wallis JD and Carmena JM**. Reversible large-scale modification of cortical networks during neuroprosthetic control. *Nat Neurosci* 14(5): 662–667. 2011.

**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(11): 1527–1537. 1982.

**Georgopoulos AP, Kalaska JF, Caminiti R and Massey JT**. Interruption of motor cortical discharge subserving aimed arm movements. *Exp Brain Res* pages 327–340. 1983.

**Gilja V, Nuyujukian P, Chestek Ca, Cunningham JP, Yu BM, Fan JM, Churchland MM, Kaufman MT, Kao JC, Ryu SI and Shenoy KV**. A high-performance neural prosthesis enabled by control algorithm design. *Nat Neurosci* 15(12): 1–49. 2012.

**Golub MD, Chase SM and Yu BM**. Learning an InteRNAl Dynamics Model from Control Demonstration. *Proceedings of the 30th InteRNAtional Conference on Machine Learning* 28: 606–614. 2013.

**Golub MD, Yu BM, Schwartz AB and Chase SM**. Motor cortical control of movement speed with implications for brain-machine interface control. *J Neurophysiol* 112(April): 411–429. 2014.

**Gowda S, Orsborn AL, Overduin Sa, Moorman HG and Carmena JM**. Designing dynamical properties of brain-machine interfaces to optimize task-specific performance. *IEEE T Neur Sys Reh* 22(5): 911–20. 2014.

**Graf ABa, Kohn A, Jazayeri M and Movshon JA**. Decoding the activity of Neuronal populations in macaque primary visual cortex. *Nat Neurosci* 14(2): 239–245. 2011.

**Granata KP and Marras WS**. CostBenefit of Muscle Cocontraction in Protecting Against Spinal Instability. *Spine* 25(11): 1398–1404. 2000.

**Gu Y, Liu S, Fetsch CR, Yang Y, Fok S, Sunkara A, DeAngelis GC and Angelaki DE**. Perceptual learning reduces interNeuronal correlations in macaque visual cortex. *Neuron* 71(4): 750–61. 2011.

**Hebb DO**. *The Organization of Behavior.* John Wiley & Sons. 1949.

**Herzfeld DJ, Vaswani Pa, Marko M and Shadmehr R**. A memory of errors in sensorimotor learning. *Science* 345: 1349–1353. 2014.

**Hikosaka O, Nakahara H, Rand M and Sakai K**. Parallel neural networks for learning sequential procedures. *Trends Neurosci* 2236(99): 464–471. 1999.

**Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, Vogel J, Haddadin S, Liu J, Cash SS, van der Smagt P and Donoghue JP**. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* 485(7398): 372–375. 2012.

**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(7099): 164–171. 2006.

**Hocherman S and Wise SP**. Effects of hand movement path on motor cortical activity in awake, behaving rhesus monkeys. *Exp Brain Res* 83(2): 285–302. 1991.

**Holtmaat A and Svoboda K**. Experience-dependent structural synaptic plasticity in the mammalian brain. *Nat Rev Neurosci* 10(9): 647–58. 2009.

**Huang HJ and Ahmed Aa**. Reductions in muscle coactivation and metabolic cost during visuomotor adaptation. *J Neurophysiol* 112(9): 2264–2274. 2014.

**Hwang EJ, Bailey PM and Andersen RA**. Volitional Control of Neural Activity Relies on the Natural Motor Repertoire. *Curr Biol* 23(5): 1–9. 2013.

**Ingvalson EM, Holt LL and McClelland JL**. Can native Japanese listeners learn to differentiate /rl/ on the basis of F3 onset frequency? *Biling-Lang Cogn* 15(02): 255–274. 2011.

**Izawa J, Rane T, Donchin O and Shadmehr R**. Motor adaptation as a process of reoptimization. *J Neurosci* 28(11): 2883–2891. 2008.

**Jackson A, Mavoori J and Fetz EE**. Long-term motor cortex plasticity induced by an electronic neural implant. *Nature* 444(7115): 56–60. 2006.

**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(49): 19486–19491. 2008.

**Jeanne JM, Sharpee TO and Gentner TQ**. Associative learning enhances population coding by inverting interNeuronal correlation patterns. *Neuron* 78(2): 352–63. 2013.

**Kakei S, Hoffman DS and Strick PL**. Muscle and movement representations in the primary motor cortex. *Science* 285(5436): 2136–2139. 1999.

**Kandel E and Spencer W**. Cellular neurophysiological approaches in the study of learning. *Physiological Reviews* 48(1): 65–135. 1968.

**Kao JC, Nuyujukian P, Stavisky S, Ryu SI, Ganguli S and Shenoy KV**. Investigating the role of firing-rate normalization and dimensionality reduction in brain-machine interface robustness. *35th Annual InteRNAtion Converence of the IEEE EMBS* pages 293–298. 2013.

**Kaufman MT, Churchland MM, Ryu SI and Shenoy KV**. Cortical activity in the null space: permitting preparation without movement. *Nat Neurosci* (August 2013). 2014.

**Kaufman MT, Churchland MM, Santhanam G, Yu BM, Afshar A, Ryu SI and Shenoy KV**. Roles of monkey premotor Neuron classes in movement preparation and execution. *J Neurophysiol* 104(2): 799–810. 2010.

**Kelly R and Strick P**. Cerebellar loops with motor cortex and prefrontal cortex of a nonhuman primate. *The J Neurosci* 23(23): 8432–8444. 2003.

**Kim SP, Simeral JD, Hochberg LR, Donoghue JP and Black MJ**. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *J Neural Eng* 5: 455–476. 2008.

**Kirkwood A and Bear M**. Hebbian synapses in visual cortex. *The J Neurosci* 14(3): 1634–1645. 1994.

**Komatsu Y**. Age-Dependent Long-Term Potentiation Transmission in Rat Visual Cortex. *J Neurosci* 14(11): 6488–6499. 1994.

**Kording KP and Wolpert DM**. Bayesian integration in sensorimotor learning. *Nature* 427(6971): 244–247. 2004.

**Kuhlman SJ, O'Connor DH, Fox K and Svoboda K**. Structural Plasticity within the Barrel Cortex during Initial Phases of Whisker-Dependent Learning. *J Neurosci* 34(17): 6078–6083. 2014.

**Kurtzer I, Herter TM and Scott SH**. Random change in cortical load representation suggests distinct control of posture and movement. *Nat Neurosci* 8(4): 498–504. 2005.

**Lackner JR and Dizio P**. Rapid adaptation to Coriolis force perturbations of arm trajectory. *J Neurophysiol* 72(1): 299–313. 1994.

**Lacquaniti F, Terzuolo C and Viviani P**. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychol* 54: 115–130. 1983.

**Law CT and Gold JI**. Neural correlates of perceptual learning in a sensory-motor, but not a sensory, cortical area. *Nat Neurosci* 11(4): 505–513. 2008.

**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(2): 593–607. 2001.

**Li Z, Lebedev MA and Nicolelis MAL**. Adaptive Decoding for Brain-Machine Interfaces Through Bayesian Parameter Updates. *Neural Comput* 23: 3162–3204. 2011.

**Li Z, O'Doherty JE, Hanson TL, Lebedev Ma, Henriquez CS and Nicolelis MaL**. Unscented Kalman filter for brain-machine interfaces. *PloS One* 4(7): e6243. 2009.

**Luczak A, Barthó P and Harris KD**. Spontaneous events outline the realm of possible sensory responses in neocortical populations. *Neuron* 62(3): 413–25. 2009.

**Machens CK, Romo R and Brody CD**. Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex. *J Neurosci* 30(1): 350–360. 2010.

**MacKay DJC**. Bayesian Inference and Sampling Theory. In *Information Theory, Inference and Learning Algorithms*, chapter 37, pages 457–466. Cambridge University Press, Cambridge. 2003.

**Mandelblat-Cerf Y, Novick I, Paz R, Link Y, Freeman S and Vaadia E**. The Neuronal basis of long-term sensorimotor learning. *J Neurosci* 31(1): 300–313. 2011.

**Mante V, Sussillo D, Shenoy KV and Newsome WT**. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* 503(7474): 78–84. 2013.

**Markram H and Tsodyks M**. Redistribution of synaptic efficacy between neocortical pyramidal Neurons. *Nature* 382: 807–810. 1996.

**Mazor O and Laurent G**. Transient dynamics versus fixed points in odor representations by locust antennal lobe projection Neurons. *Neuron* 48(4): 661–73. 2005.

**Mazzoni P and Krakauer JW**. An implicit plan overrides an explicit strategy during visuomotor adaptation. *J Neurosci* 26(14): 3642–3645. 2006.

**McKenzie Ia, Ohayon D, Li H, Paes de Faria J, Emery B, Tohyama K and Richardson WD**. Motor skill learning requires active central myelination. *Science* 346(6207): 318–322. 2014.

**Merzenich MM, Nelson RJ, Stryker MP, Cynader MS, Schoppmann a and Zook JM**. Somatosensory cortical map changes following digit amputation in adult monkeys. *J Comp Neurol* 224(4): 591–605. 1984.

**Milner TE**. Adaptation to destabilizing dynamics by means of muscle cocontraction. *Exp Brain Res* 143(4): 406–416. 2002.

**Morasso P**. Spatial control of arm movements. *Exp Brain Res* 42: 223–227. 1981.

**Mulliken GH, Musallam S and Andersen RA**. Decoding trajectories from posterior parietal cortex ensembles. *J Neurosci* 28(48): 12913–12926. 2008.

**Newport E**. Maturational constraints on language learning. *Cognitive Sci* 14(1): 11–28. 1990.

**Newsome W, Britten K and Movshon J**. Neuronal correlates of a perceptual decision. *Nature* 341: 52–54. 1989.

**Nordhausen CT, Maynard EM and Normann RA**. Single unit recording capabilities of a 100 microelectrode array. *Brain Res* 726: 129–140. 1996.

**Nowak Da, Timmann D and Hermsdörfer J**. Dexterity in cerebellar aGenesis. *Neuropsychologia* 45(4): 696–703. 2007.

**Nudo RJ, Milliken GW, Jenkins WM and Merzenich MM**. Use-Dependent Alterations of Movement Representations Cortex of Adult Squirrel Monkeys. *J Neurosci* 16(2): 785–807. 1996.

**Oby ER, Ethier C and Miller LE**. Movement representation in the primary motor cortex and its contribution to Generalizable EMG predictions. *J Neurophysiol* 109(3): 666–678. 2013.

**Omlor L and Giese MA**. Anechoic Blind Source Separation Using Wigner Marginals. *J Mach Learn Res* 12: 1111–1148. 2011.

**Orban de Xivry JJ, Ahmadi-Pajouh MA, Harran MD, Salimpour Y and Shadmehr R**. Changes in corticospinal excitability during reach adaptation in force fields. *J Neurophysiol* (October 2012): 124–136. 2012.

**Orsborn AL, Dangi S, Moorman HG and Carmena JM**. Closed-loop decoder adaptation on intermediate time-scales facilitates rapid BMI performance improvements independent of decoder initialization conditions. *IEEE T Neur Sys Reh* 20(4): 468–77. 2012.

**Orsborn AL, Moorman HG, Overduin Sa, Shanechi MM, Dimitrov DF and Carmena JM**. Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron* 82(6): 1380–93. 2014.

**Paninski L, Fellows MR, Hatsopoulos NG and Donoghue JP**. Spatiotemporal Tuning of Motor Cortical Neurons for Hand Position and Velocity. *J Neurophysiol* 91: 515–532. 2004.

**Park DC, Lodi-Smith J, Drew L, Haber S, Hebrank A, Bischof GN and Aamodt W**. The impact of sustained engagement on cognitive function in older adults: the Synapse Project. *Psychological Science* 25(1): 103–12. 2014.

**Pascual-Leone A and Torres F**. Plasticity of the sensorimotor cortex representation of the reading finger in Braille readers. *Brain* 116(1): 39–52. 1993.

**Paz R, Natan C, Boraud T, Bergman H and Vaadia E**. Emerging patterns of Neuronal responses in supplementary and primary motor areas during sensorimotor adaptation. *J Neurosci* 25(47): 10941–10951. 2005.

**Pearce TM and Moran DW**. Strategy-Dependent Encoding of Planned Arm Movements in the Dorsal Premotor Cortex. *Science* (July). 2012.

**Pearson K**. On lines and planes of closest fit to systems of points in space. *Philos Mag* 2: 559–572. 1901.

**Perel S, Sadtler PT, Godlove JM, Ryu SI, Wang W, Batista AP and Chase SM**. Direction and speed tuning of motor-cortex multi-unit activity and local field potentials during reaching movements. *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual InteRNAtional Conference of the IEEE* pages 299–302. 2013.

**Peters AJ, Chen SX and Komiyama T**. Emergence of reproducible spatiotemporal activity during motor learning. *Nature* 510(7504): 263–267. 2014.

**Picard N, Matsuzaka Y and Strick PL**. Extended practice of a motor skill is associated with reduced metabolic activity in M1. *Nat Neurosci* 16(9). 2013.

**Pons T, Garraghty P, Ommaya A, Kaas J, Tuab E and Mishkin M**. Massive cortical reorganization after sensory deafferentation in adult macaques. *Science* 252(5014): 1857–1860. 1991.

**Rathelot JA and Strick PL**. Subdivisions of primary motor cortex based on cortico-motoNeuronal cells. *Proc Natl Acad Sci U S A* 106(3): 918–923. 2009.

**Reina GA, Moran DW, Schwartz AB, Snider J, Lee D, Harrington DL, Poizner H, Velliste M, Kennedy SD, Whitford AS, Sohn Jw and Mcmorland AJC**. On the Relationship Between Joint Angular Velocity and Motor Cortical Discharge During Reaching. *J Neurophysiol* 85: 2576–2589. 2001.

**Rigotti M, Barak O, Warden MR, Wang XJ, Daw ND, Miller EK and Fusi S**. The importance of mixed selectivity in complex cognitive tasks. *Nature* 497(7451): 585–590. 2013.

**Rioult-Pedotti MS, Friedman D and Donoghue JP**. Learning-Induced LTP in Neocortex. *Science* 290(5491): 533. 2000.

**Rokni U, Richardson AG, Bizzi E and Seung H**. Motor learning with unstable neural representations. *Neuron* 54(4): 653–66. 2007.

**Role BL, Smith MA, Brandt J and Shadmehr R**. Motor disorder in Huntington ' s disease begins as a dysfunction in error feedback control. *Nature* 403: 544–549. 2000.

**Romo R, Hernández A and Zainos A**. Neuronal correlates of a perceptual decision in ventral premotor cortex. *Neuron* 41: 165–173. 2004.

**Rouse AG, Williams JJ, Wheeler JJ and Moran DW**. Cortical adaptation to a chronic micro-electrocorticographic brain computer interface. *J Neurosci* 33(4): 1326–1330. 2013.

**Sadtler PT, Quick KM, Golub MD, Chase SM, Ryu SI, Tyler-Kabara EC, Yu BM and Batista AP**. Neural constraints on learning. *Nature* 512: 423–426. 2014.

**Sadtler PT, Ryu SI, Yu BM and Batista AP**. High-performance neural prosthetic control along instructed paths. In *Proceedings of the 5th InteRNAtional IEEE EMBS Conference on Neural Engineering*, pages 601–604. Cancun. 2011.

**Saffran J**. Constraints on Statistical Language Learning. *J Mem Lang* 47(1): 172–196. 2002.

**Salinas E**. Fast remapping of sensory stimuli onto motor actions on the basis of contextual modulation. *J Neurosci* 24(5): 1113–1118. 2004.

**Santhanam G, Ryu SI, Yu BM, Afshar A and Shenoy KV**. A high-performance brain-computer interface. *Nature* 442(7099): 195–198. 2006.

**Santhanam G, Yu BM, Gilja V, Ryu SI, Afshar A, Sahani M and Shenoy KV**. Factor-Analysis Methods for Higher-Performance Neural Prostheses. *J Neurophysiol* 102: 1315–1330. 2009.

**Schalk G, Miller KJ, Anderson NR, Wilson Ja, Smyth MD, Ojemann JG, Moran DW, Wolpaw JR and Leuthardt EC**. Two-dimensional movement control using electrocorticographic signals in humans. *J Neural Eng* 5(1): 75–84. 2008.

**Schwartz AB**. Direct Cortical Representation of Drawing. *Science* 265(5171): 540–542. 1994.

**Schwartz AB, Moran DW and Reina GA**. Differential representation of perception and action in the frontal cortex. *Science* 303(5656): 380–383. 2004.

**Schwarz Da, Lebedev Ma, Hanson TL, Dimitrov DF, Lehew G, Meloy J, Rajangam S, Subramanian V, Ifft PJ, Li Z, Ramakrishnan A, Tate A, Zhuang KZ and Nicolelis MaL**. Chronic, wireless recordings of large-scale brain activity in freely moving rhesus monkeys. *Nat Methods* 11(6): 670–676. 2014.

**Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR and Donoghue JP**. Instant neural control of a movement signal. *Nature* 416(6877): 141–2. 2002.

**Shadmehr R and Krakauer JW**. A computational neuroanatomy for motor control. *Exp Brain Res* 185(3): 359–381. 2008.

**Shadmehr R, Smith Ma and Krakauer JW**. Error correction, sensory prediction, and adaptation in motor control. *Annu Rev Neurosci* 33: 89–108. 2010.

**Smith Ma and Shadmehr R**. Intact ability to learn inteRNAl models of arm dynamics in Huntington's disease but not cerebellar deGeneration. *J Neurophysiol* 93(5): 2809–2821. 2005.

**Stevenson IH, Cherian A, London BM, Sachs Na, Lindberg E, Reimer J, Slutzky MW, Hatsopoulos NG, Miller LE and Kording KP**. Statistical assessment of the stability of neural movement representations. *J Neurophysiol* 106(2): 764–774. 2011.

**Suminski AJ, Tkach DC, Fagg AH and Hatsopoulos NG**. Incorporating Feedback from Multiple Sensory Modalities Enhances Brain  Machine Interface Control. *J Neurosci* 30(50): 16777–16787. 2010.

**Taylor DM, Tillery SIH and Schwartz AB**. Direct cortical control of 3D neuroprosthetic devices. *Science* 296(5574): 1829–32. 2002.

**Thompson DE, Quitadamo LR, Mainardi L, Laghari KUR, Gao S, Kindermans PJ, Simeral JD, Fazel-Rezai R, Matteucci M, Falk TH, Bianchi L, Chestek Ca and Huggins JE**. Performance measurement for brain-computer or brain-machine interfaces: a tutorial. *J Neural Eng* 11(3): 035001. 2014.

**Tkach D, Reimer J and Hatsopoulos NG**. Observation-based learning for brain-machine interfaces. *Curr Opin Neurobiol* 18(6): 589–94. 2008.

**Todorova S, Sadtler P, Batista A, Chase S and Ventura V**. To sort or not to sort: the impact of spike-sorting on neural decoding performance. *J Neural Eng* 11(5): 056005. 2014.

**Turk M and Pentland A**. Eigenfaces for Reconition. *J Cognitive Neurosci* 3(1): 71–86. 1991.

**Ungerleider LG, Doyon J and Karni A**. Imaging Brain Plasticity during Motor Skill Learning. *Neurobiol Learn Mem* 78(3): 553–564. 2002.

**Velliste M, Perel S, Spalding MC, Whitford AS and Schwartz AB**. Cortical control of a prosthetic arm for self-feeding. *Nature* 453(7198): 1098–1101. 2008.

**Wang W, Collinger JL, Degenhart AD, Tyler-Kabara EC, Schwartz AB, Moran DW, Weber DJ, Wodlinger B, Vinjamuri RK, Ashmore RC, Kelly JW and Boninger ML**. An Electrocorticographic Brain Interface in an Individual with Tetraplegia. *PloS One* 8(2): e55344. 2013.

**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(1): 80–118. 2006.

**Yin HH, Mulcare SP, Hilário MRF, Clouse E, Holloway T, Davis MI, Hansson AC, Lovinger DM and Costa RM**. Dynamic reorganization of striatal circuits during the acquisition and consolidation of a skill. *Nat Neurosci* 12(3): 333–341. 2009.

**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.

**Yu BM, Kemere CT, Santhanam G, Afshar A, Ryu SI, Meng TH, Sahani M and Shenoy KV**. Mixture of Trajectory Models for Neural Decoding of Goal-Directed Movements. *J Neurophysiol* 97: 3763–3780. 2007.

**Zacksenhouse M, Lebedev Ma, Carmena JM, O'Doherty JE, Henriquez CS and Nicolelis MaL**. Cortical modulations increase in early sessions with brain-machine interface. *PloS One* 2(7): e619. 2007.

**Zimmermann JB and Jackson A**. Closed-loop control of spinal cord stimulation to restore hand function after paralysis. *Front Neurosci* 8(May): 1–8. 2014.