uc3m | Universidad Carlos III de Madrid

University Degree in Biomedical Engineering
2018-2019

Bachelor Thesis

Comparison of BMI decoders based on neural population activity and local field potentials

---

Cecilia Gallego Carracedo

Advisors

Juan Álvaro Gallego Abella

Juan José Vaquero López

Leganés, 2019

ACKNOWLEDGMENTS

First of all, I want to thank Juan Gallego for teaching me how to keep it short and simple. You guidance enlightened my project and my life.

I also want to thank the group of Neural and Cognitive Engineering for taking me in during this project. Especially to Julio and to our beloved leader, Eduardo.

To my university professors, for teaching me so much during my bachelor's degree, especially to Javier Pascau for his dedication, and to Juan José Vaquero for providing me with my first real world experience as an intern and for overseeing this project.

To my university friends, Tractores, for sharing this four years with me. And to my lifelong friends, B******, for their support and understanding during my reclusion.

Last but not least, to my family, for their unconditional love, support and patience after hearing for a thousand times what a neural manifold is.

ABSTRACT

Motor cortex is the main output from the brain to control the muscles. Motor cortical activity contains rich information about movement features. This information can be read out at different levels: from the activity of single neurons, to local field potentials (LFP) that reflect features in the electric potentials of groups of neighboring neurons. Researchers have leveraged this observation to develop brain-machine interfaces (BMIs), systems that "decode" movement parameters from the neural activity and use them to control external devices, such as computer cursors or robots, or even allow the subject to control their own paralyzed limb.

Current intracortical BMIs take as inputs "control signals" from the individual activity of neural populations or from LFPs. Both types of decoders yield quite accurate performance when tested online, however it is not clear what type of movement-related features each of these two modalities contains and which are shared among them.

The goal of this project is to understand what information useful for movement decoding is common across neural population activity and LFPs. To this end, I built decoders based on neural population activity and LFPs. I also used recent conceptual developments that assume that neural computations are based on population-wide activity patterns rather than on independently modulated single units. My analysis showed that the performance of these three types of decoders was quite similar, with LFP inputs providing lightly worst predictions. However, LFP-based decoders were more robust against input channel lost, a common challenge to BMIs. Finally, I began to explore the relationship between neural population dynamics and the time course of the LFPs, identifying an intriguing, previously unreported relationship between the two. These results set the basis for future comparisons of decoder inputs and, hold potential to enable more robust BMIs.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

1. INTRODUCTION

1.1. Motivation

Nowadays, there is no fully effective cure for many neurological diseases and injuries, including spinal cord injury, stroke or amyotrophic lateral sclerosis. However, as technology develops, new solutions for these problems are starting to appear. Brain-machine interfaces (BMI) that "connect" the brain to an external device hold great promise to compensate for movement disabilities. Their potential applications vary from controlling a computer cursor or a robotic limb [1], [2], to even restoring the patient's control of their paralyzed muscles with functional electrical stimulation (FES) [3].

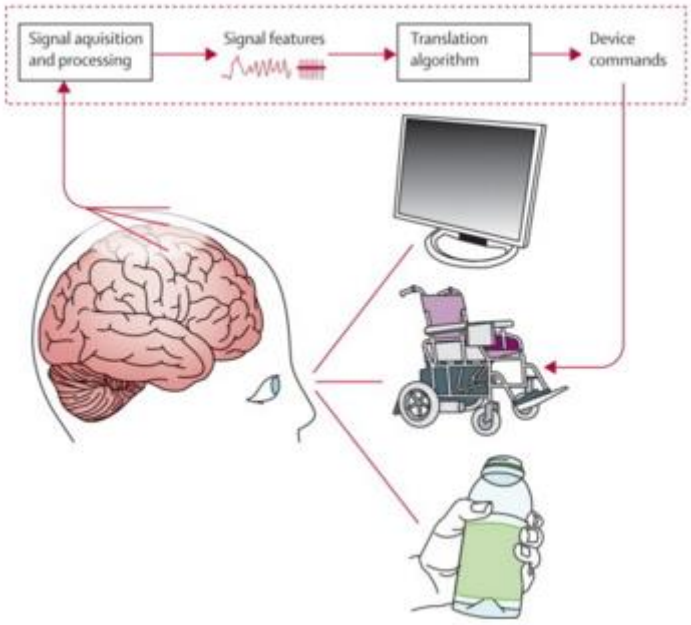

Figure 1.1. BMI operation scheme [1]

However, since BMI performance is still limited, these applications are hardly achievable. Current BMIs face at least two main challenges. First, we need better technology that records from more locations in the brain with high temporal and spatial resolution, and that are stable over stable periods of time. Second, if we are to restore normal function after neurological injury or disease, we need to further our current

understanding of the brain in health and disease. This project lies at the intersection of these two challenges: it aims to improve the robustness of current BMIs by improving our understanding of different type of neural activity patterns that can be recorded from the brain.

1.2. Movement-related information in cortex

The central nervous system consists of the brain and the spinal cord. It is so-called because it integrates afferent information, coming from sensors that inform about the state of the body, and commands movement through efferent pathways [4]. The combination of afferent and efferent signals is critical for the neural control of movement. (Figure 1.2).

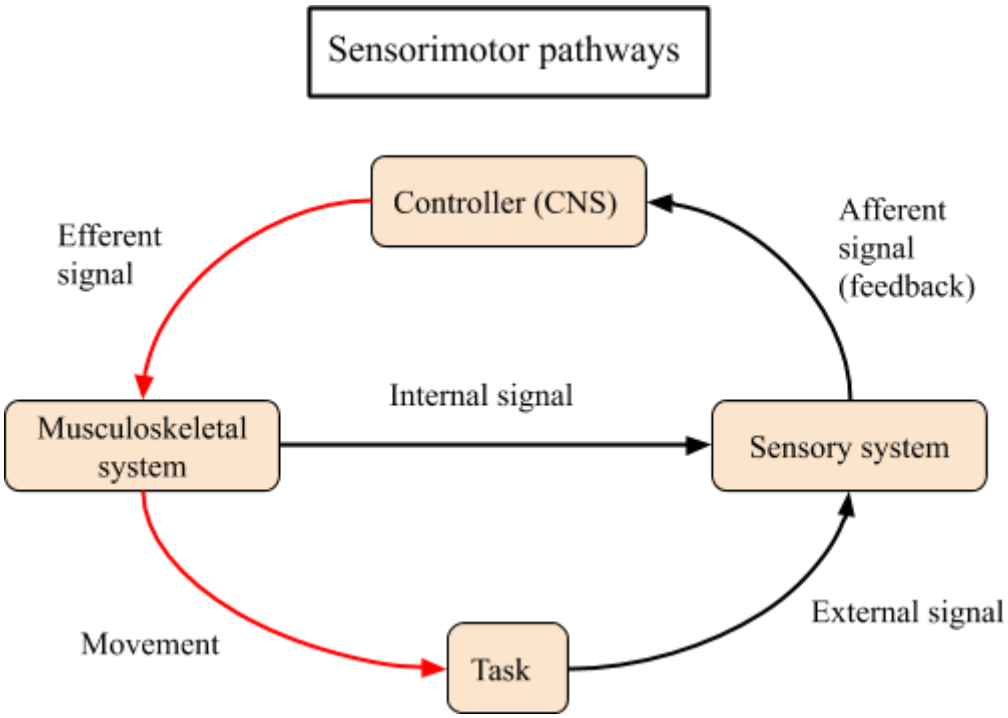

Figure 1.2. Human movement control scheme. Red arrows represent the paths included in a typical brain-machine interface, although musculoskeletal system is substituted by a decoder.

Although both structures controlled a huge variety of body function, spinal cord is specially related to reflexes while brain is in charge of voluntary movement [4]. Since the project is focused on voluntary movement during a trained activity, only the brain will be relevant.

The brain is spatially organized into a number of regions that are primarily involved in specific functions or processes [5]. Since this project focuses on BMIs to restore movement, I will focus on primary motor cortex, the main output from cortex to the spinal motoneurons that make the muscles control [6]. The primary motor cortex (M1) has a number of interesting properties, for example, it is somatotopically organized, meaning that different regions are more implicated in controlling different parts of the body[7].

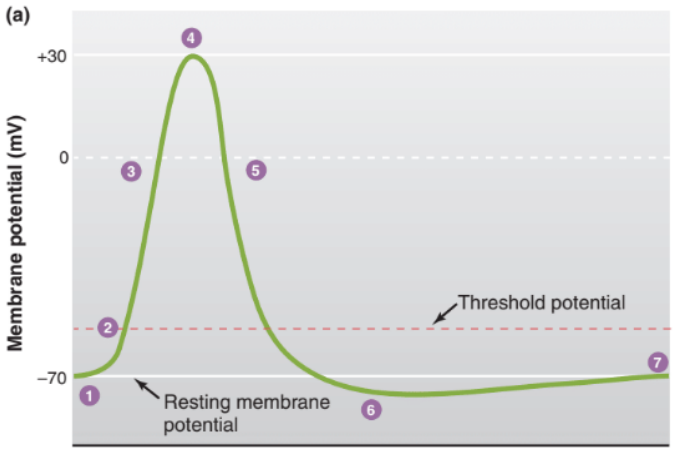

Figure 1.3. Typical action potential representation. Neural membrane potential is represented as a function of time [4]

The nervous system is composed of neural (i.e., neurons) and glial cells. In mammals, neural information exchange is mediated by action potentials fired by neurons. Action potentials are caused by an ion exchange in the neuron's membrane which generates a potential change from the -70 mV, resting potential, to +30 mV. An action potential has the characteristic shape shown in Figure 1.3. Note that the action potential waveform is invariant to the synaptic inputs that the neuron receives; instead, strong inputs lead to higher firing frequency (i.e. more action potentials per unit time) [4]. To digitalize the neuron activation implies detecting when the neuron is firing an action potential and adding a 1 to an otherwise 0 recorded vector. In other words, neuronal activity is interpreted as a binary system that can be either activated or deactivated

1.3. From single neurons to neural manifolds

The activity of individual neurons in the motor cortex –and the entire brain– can be recorded by inserting electrodes that detect the occurrence of action potentials. The most commonly accepted view is that information is encoded as the firing frequency of individual neurons, although alternative theories exist, including one that posits that the precise timing of each action potential is critical [8].

Focusing on motor cortex, for decades, neuroscientists sought to understand what variables M1 neurons encoded [9] [10]. As experiments became more sophisticated, it became apparent that there M1 does not merely encode movement parameters in a robust context-independent manner.

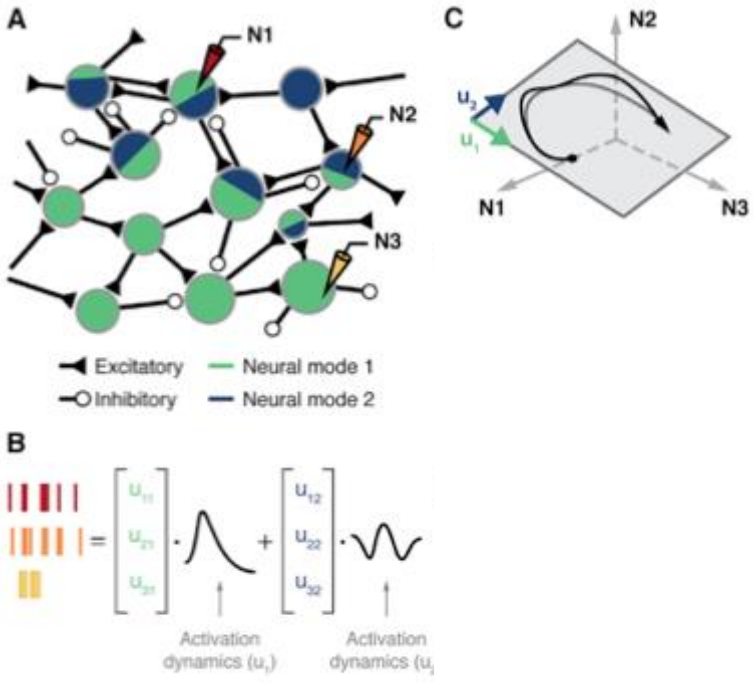

Figure 1.4. Neural manifold hypothesis. **A**: The neural population of two neural modes (green and blue). **B:** The relation between neurons N1, N2 and N3 firing rates and the two obtained neural modes (u_1 and u_2). **C** illustrates the linear manifold (grey) formed by the neural nodes u_1 and u_2. Activation dynamics in **B** correspond to the black curve in **C**. [11]

An interesting observation is that even though primate motor cortex is made up of millions of neurons, movement can be well described based on the activity of only a few tens of hundreds of neurons. This observation implies that neural activity is "low dimensional," and that the activity of different neurons should be related to each other. Many believe that this low-dimensionality, which follows from the correlations across

neurons, arises from the “constraints” imposed by the thousands of connections that cortical neurons make onto other neurons [11] (Figure 1.4A).

Based on these ideas an increasing number of scientists are exploring the idea that neural computation is based on population-wide activity patterns rather than on the activity of individual neurons [12] [13]. Each covariation pattern has been called a “neural mode” and, the combination of all the neural modes defines a neural manifold (Figure 1.4C), which includes all possible activity patterns of the recorded neural population. The time-varying activation of a neural mode is often referred to as a “latent variable” (Figure 1.4B).

Mathematically, this neural manifold is a low-dimensional surface within a high dimensional space in which each axis is the activity of one recorded neuron. The neural manifold can be estimated using dimensionality reduction methods, such as principal component analysis (PCA), factor analysis (FA), or local linear embedding [14]. All these methods find patterns underlying the neural mypopulation activity.

The experiment that perhaps provides the strongest evidence for the neural manifold capturing some aspect of neural connectivity was performed by Sadtler et. Al. [13]. The authors built a mathematical mapping from the latent variables to the hand movements (a BMI “decoder,” as discussed below). After the monkeys had performed a few hundreds of trials using this mapping to move the cursor with their thoughts, the authors applied one of two types of perturbation to the decoder. In the first type of perturbation, they rotated this intuitive decoder within the manifold, whereas in the second type of perturbation, they rotated the decoder so it lied outside the manifold (Figure 1.5C). The monkeys only had to adapt to one of these two perturbations in each experimental session. Interestingly, for the within manifold perturbations, the monkeys learned quite easily how to modulate the neural activity, achieving performance levels comparable to those observed when they used the intuitive decoders after approximately 10 trials (Figure 1.5A). In stark contrast, the monkeys were not able to generate the new activity patterns required by the outside manifold perturbations to complete the task. This study thus provides strong evidence that the manifold may indeed capture some intrinsic aspect of the neural circuitry, as it is not possible to learn new activity patterns that may require the formation of new synapses.

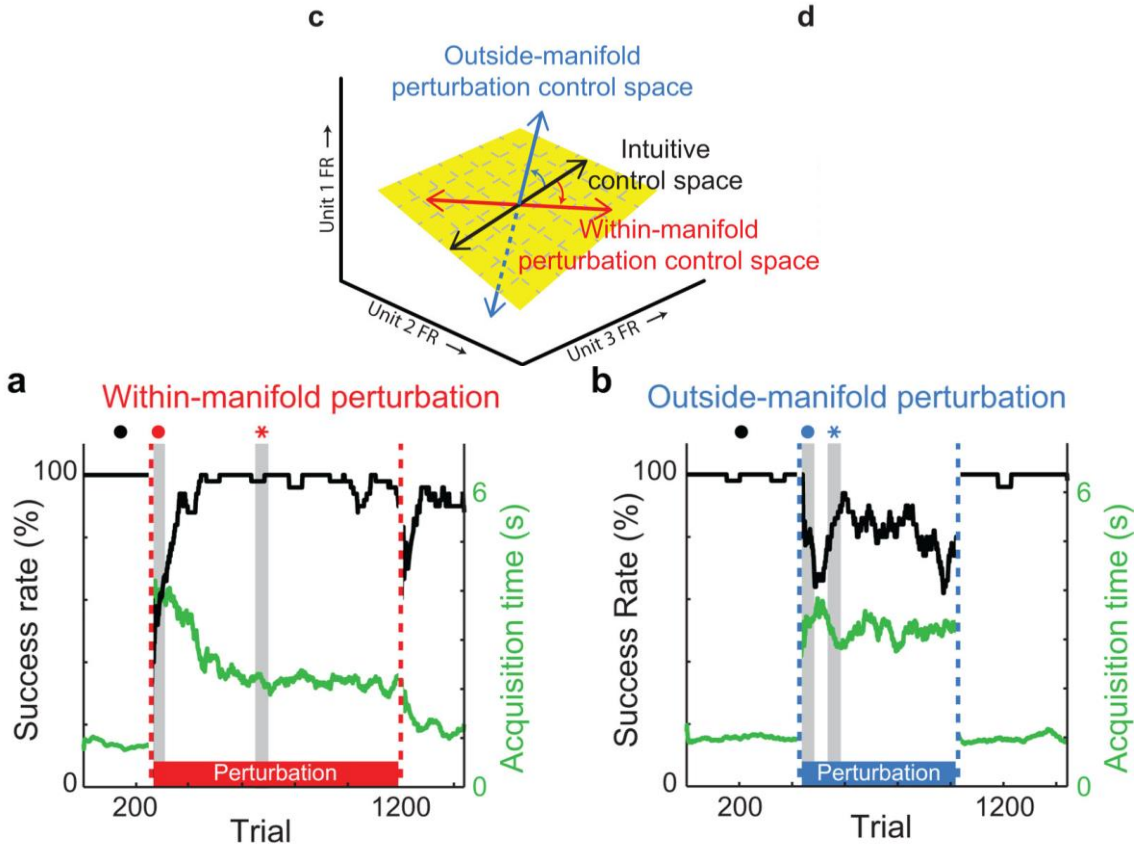

Figure 1.5. **C:** Simplified representation of a three neuron's manifold and the within and outside perturbations. **A** and **B:** monkey's performance after each type of perturbation in the decoder.[13]

1.4. Local field potentials

Although our most detailed characterization of "neural processing" comes from reading out the activity of many neurons simultaneously, current technologies present two limitations that hamper their use in BMIs. First, chronically implanted electrodes often record from different neurons on different days; for example, some estimate that 50 % of the neurons recorded with a state of the art 100-channel microelectrode array change after one week. Second, electrodes eventually lose the ability to record single neurons due to body reaction [9].

Local field potentials (LFP), extracellular changes in voltage arising from transient imbalances in ions around the recording electrode, it can often be detected when the activity of single neurons cannot be resolved. This robustness makes them a potentially appealing input signal for BMIs. It must be noted, however, that although it is commonly accepted that LFPs reflect shared synaptic inputs to neural populations [10], their origin is still controversial [11].

LFPs are often analyzed by calculating their power in a series of frequency bands, which are traditionally associated with different brain processes. These bands usually are: delta (0 - 5 Hz), theta (5 - 8 Hz), alpha (8 - 15 Hz), beta (15 - 30 Hz), and gamma (>30 Hz), with the latter often divided into low (30 – 100 Hz) and high gamma frequencies (> 100 Hz).

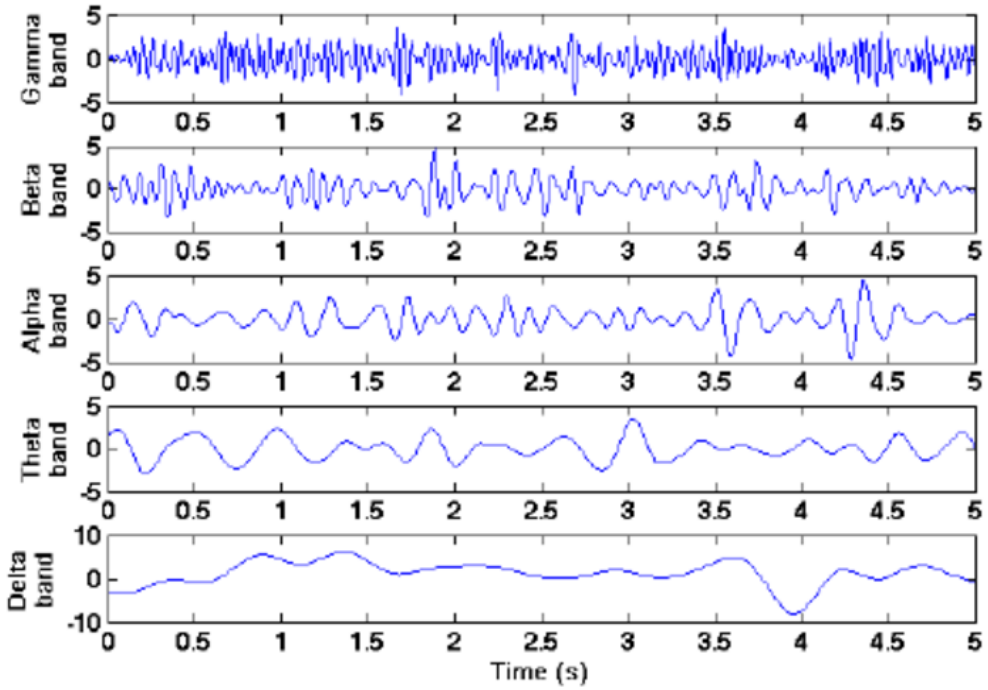

Figure 1.5. LFP visualization at different frequency bands. [12]

LFPs in the motor cortex are modulated by a number of movement parameters, including target location [10] or how animals contract their muscles [13]. Several groups have built BMIs based on LFPs rather than the activity of neural populations, which achieve often comparable performance [14].

In this project, I will further this comparison of neural signals as BMI inputs.

1.5. Brain-machine interfaces: State of the art, and looming challenges

A brain-machine interface (BMI) typically estimates or “decodes” movement-related information in the neural activity and uses it as a control signal to command an external device [18] or electrical stimulation of specific pathways [19]. Historically, Eb Fetz unknowingly set the foundation for BMIs in 1969, when he demonstrated that monkeys

could voluntarily increase or decrease the firing rate of a cortical neuron to get a reward[20]. Besides their appeal as scientific tool, BMIs hold great promise to restore movement to patients with severe motor disabilities, effectively bypassing their impaired injuries.

A brain-machine interface finds the contribution of each neural parameter to the movement. Once the contribution has been determined, the BMI can "translate" the neural activity to movement.

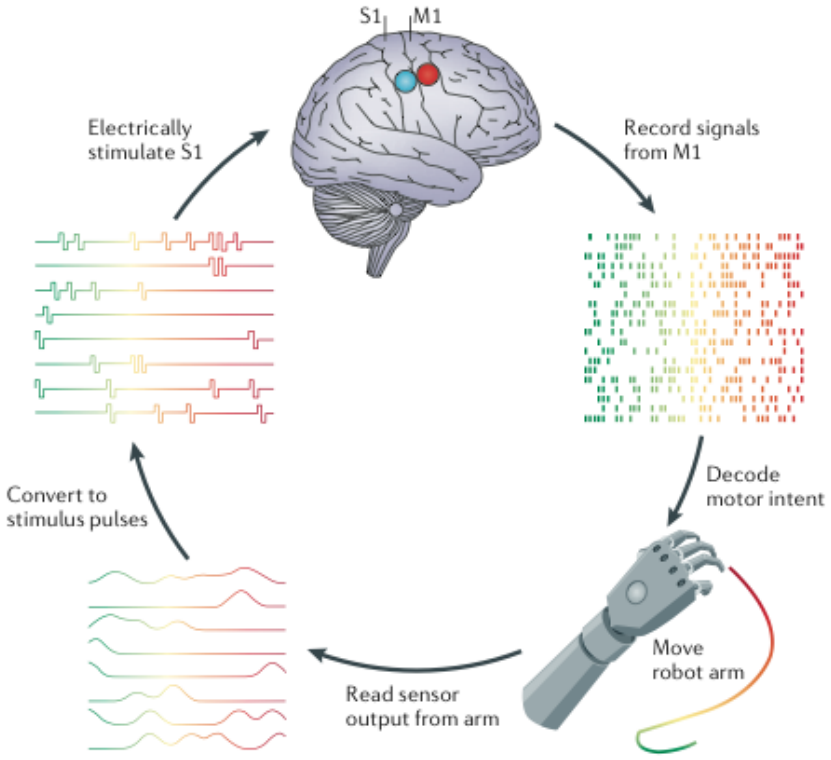

Figure 1.7. Ideal bidirectional BMI for a robotic arm control, that is, that considers both sensory and motor information to control de device, as the natural movement control [21]

BMIs read out neural information using either intracortical electrodes [21], electrodes sitting on the surface of the brain [1] or electrodes placed on the scalp (electroencephalogram; EEG). EEG offers a simple, inexpensive, and non-invasive way of recording neural activity; however its resolution is spatially limited due to the filtering properties of the scalp and the soft tissues. Besides, inverse modeling of the sources that underlie the recorded activity is extremely challenging. Despite these limitations, several groups have made quite impressive demonstrations of EEG-based BMIs [22]. In contrast, intracortical electrodes are an invasive, yet effective technique to measure localized neuronal activity with high spatial and temporal resolution [23].

They have been used for over two decades in rodents, monkeys, and in an increasing number of human patients, to record the activity from tens to hundreds of neurons [24], [25].

Most intracortical BMIs (from now on, simply BMIs) are time-invariant linear model fitted using least squares [15], [24], [26], [27], although some non-linear BMI decoders have also been developed [28]. Among these non-linear decoders, some recent studies leveraged modern machine learning techniques to improve decoder performance [29]. The improvement provided by using non-linear decoders seems to be largely dependent on the specific application.

Regarding the "output" signals of the decoders, many BMI predict kinematic variables, typically the velocity of the hand [11], [16], [24], although there is an increasing number of BMIs that decode muscle activity [27], [30]. These types of decoders rely on the notion that M1 neural activity is most strongly related to muscle activation than to movement kinematics. Selection of the appropriate output is primarily based on the application, e.g., it seems more intuitive to move a computer cursor based on kinematic outputs, and to control the stimulation of a paralyzed limb with muscle-like control signals.

Despite their successes, current intracortical BMIs do face important challenges. Foremost, they suffer the important drawback that recording quality decreases over time due to foreign body reaction and gliosis [31]. The time course of this degradation typically happens as follows: the number of resolvable neurons declines over time, until it is only possible to record LFPs. Moreover, for reasons yet unknown, chronically implanted intracortical electrodes typically record from different neurons on different days.

Thus, although neural population-based decoders are the most accurate type of decoders [32], this change in recorded neurons or "neural turnover" poses a formidable challenge for bringing them into the clinic: as decoder inputs change, the BMI feels different, which forces users to learn how to wield a new tool. The drastic decline in BMI performance due to neural turnover is illustrated by the green trace in Figure 1.8.

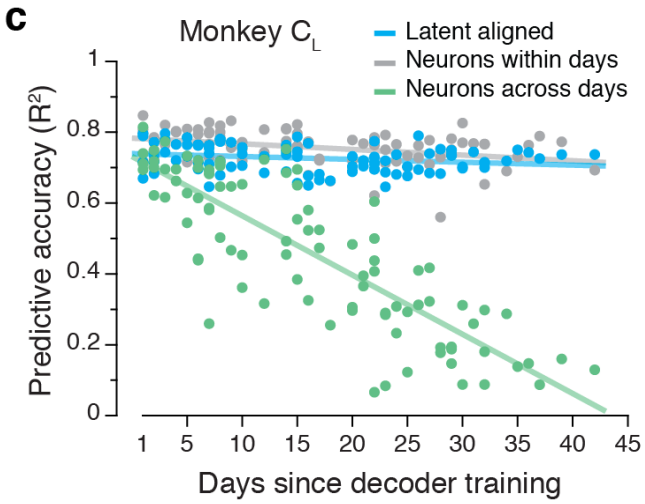

Figure 1.8. Performance of decoders across days. Green line represents the spike based decoder trained just on the first day, gray line is the spike based decoder trained each day, and blue line is the latent variable based decoder trained the first day. [26]

In a recent breakthrough, a group of researchers has been able to capitalize on the latent variable to build BMI decoders that are stable over months or even years [26] (blue trace in Figure 1.8). Using similar ideas, another group has been able to stitch neural activity across days to build better decoders [33].

An alternative approach to achieve robust BMI performance is to circumvent neural turnover by using LFPs rather than neural population activity as decoder inputs (Figure 1.9). It must be noted that with well-functioning electrode arrays, LFP-based decoders typically perform worse than neural population-based decoders [32]. Several recent studies have focused on increasing LFP decoding quality [15], [27]. For example, Jhuang et al. used high frequency gamma LFP activity for decoding, based on the assumption that these higher frequencies correlate with the activity of single neurons [16]. Others have combined LFPs and neural population firings to build hybrid decoders: Stavinsky et al. combined the low frequency LFP component with neural population spiking to compute hybrid decoders, which performed slightly better compared to decoders based neural population activity along [34].

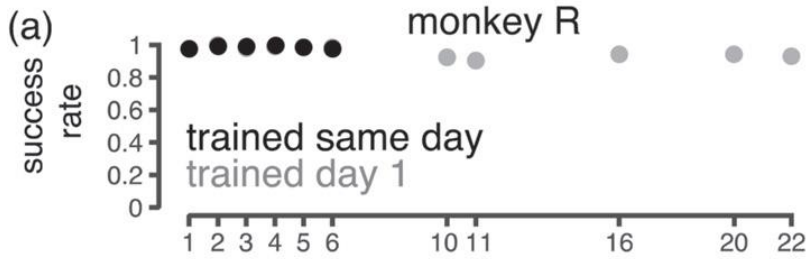

Figure 1.9. Performance of LFP based decoders across days. Decoders were trained on day 1 (gray dots) and on each day (black dots). [34]

In this project, I developed and studied BMI decoders based on neural population activity, latent variables, and LFPs, all recorded using intracortical electrodes [35]. I took a biomimetic approach, building decoders that mapped neural activity onto movement kinematics. Using multivariate analysis techniques, I showed for the first time why LFP-based decoders yield quite reliable prediction of movement kinematics: because their dynamics are similar to those of the latent variables. This observation opens the door for new developments, including stabilizing the dynamics of LFP-based decoders to achieve several-year robust control in the absence of single cell recordings.

2. METHODS

2.1. Subjects and experimental protocol

I used neural and behavioral data recorded from a monkey (macaca mulatta) at Northwestern University (Chicago, IL, USA). All surgical and experimental procedures were approved by Northwestern's Animal Care and Use Committee.

The monkey was seated in a primate chair and had to control a computer cursor with a manipulandum, performing an instructed-delay two dimensional center-out reaching task. In this task, the monkey had to move a computer cursor from the center target to an outer target located in circle of 8 cm radius. The targets were circles of 2 cm radius located at eight different angles with respect to the horizontal: 0, π/4, π/2, 3π/4, π, 5π/4, 3π/2, 7π/4, 2π.

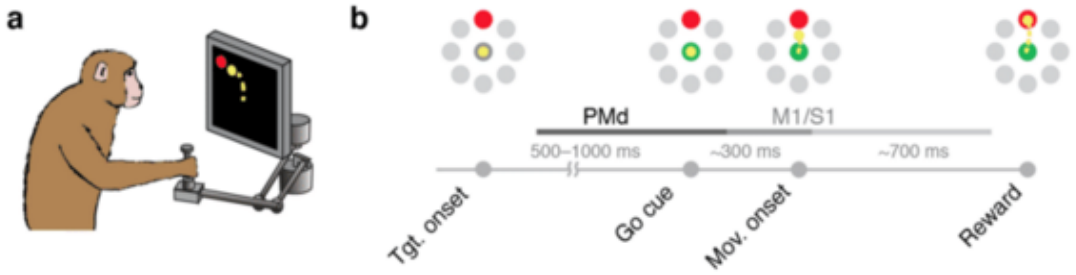

Figure 2.1. Scheme of the experimental protocol. **a** shows the set up and **b** a representation of the task to target π/2, including approximate times and labels of important times. [36]

In each session, the monkey performed several hundreds of trials. The trials were structured as follows: first, the monkey had to hold the manipulandum in the center target between 0.5 and 1.5 seconds for the target to appear; then, once one of the eight possible targets was illuminated, the monkey had to wait for an auditory go cue during a variable delay period of 0.5 - 1.5 ms. After this cue, the monkey moved the manipulandum to the outer target, where he had to hold the cursor for 0.5 ms to obtain a reward. After that, the monkey had to return the manipulandum to the center position and wait for a new trial to begin. For more information about the subject or the behavioral task, see Refs.[36], [26].

2.2. Behavioral and neural recordings

Hand movement was recorded by encoders in the manipulandum at a sampling frequency of 1 kHz. From this measurements, hand kinematics were obtained (i.e. position, velocity and acceleration at x and y directions).

Neural recordings were simultaneously obtained using a 96-channel chronic microelectrode array ("Utah Array," Blackrock Microsystems, Salt Lake City, UT) implanted in the arm area of primary motor cortex (M1). Array location was identified intraoperatively using micro stimulation of the cortical surface. This array provided two types of output.

Neural activity was simultaneously recorded (30 kHz) using a standard acquisition system (Cerebus, Blackrock Microsystems, Salt Lake City, UT). The recordings on each channel were digitized, band-pass filtered (250–5000 Hz), and then converted to spike times based on threshold crossings. The threshold was selected according to the root mean squared (RMS) activity in each channel (-5.5*RMS). An expert user performed offline neuron sorting, i.e. identified putative single neurons in each electrode, using off-the-shelf software (Offline Sorter v3, Plexon, Inc, Dallas, TX).

At the same time, the local field potential (LFP) in each microelectrode channel was recorded at a sampling frequency of 2 kHz. For more information regarding data acquisition, see[36], [26].

2.3. Data preprocessing

All neural (i.e. spikes, local field potentials) and behavioral signals (i.e. hand position and velocity) were synchronized and binned into 10 ms bins. However, time bins were further down sampled to 20 ms bins, unless said otherwise.

The recording time began before even the target is on, and ended almost a seconds after the trial had finished. Since I was interested in comparing different forms of movement decoding, I analyzed data from the go cue, to the end of the trial. Among the 286 trials used, a variety of duration is found, since monkey's velocity and holding times vary from trial to trial. However, in this project I trimmed all the trials to the shortest one, which was 1.25 seconds. Finally, al the unified trials were time concatenated and a final dataset of K bins of duration is obtained, were K is the number of time bins per trial (125 bins) times the number of trials (286 trials).

Furthermore, I smoothed these time-varying firing rates using a Gaussian kernel:

$$Kernel\ (t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-t^2}{2\sigma^2}} \quad (1)$$

Where t is the time position of kernel samples and σ is the standard deviation, in this case was, 0.05.

An example of this smoothing is shown in Figure 2.2.

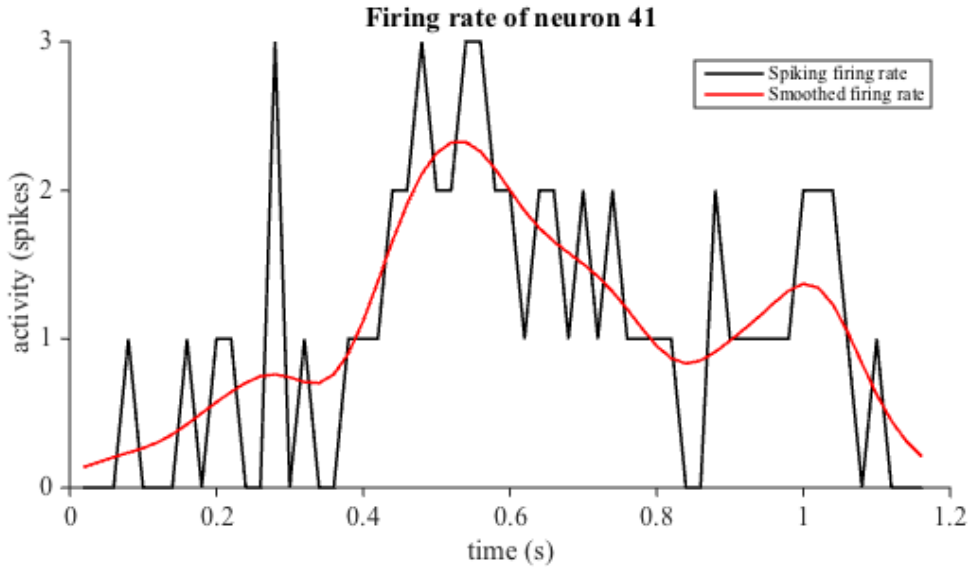

Figure 2.2. Firing rate of neuron 41 before (black) and after (red) smoothing with a Gaussian kernel of 0.05 standard deviation. Time window goes from the go cue to the end of the trial. Data has been down sampled to 50 ms per time bin to get more than one spike per time bin.

The smoothing of the spike signal was useful for further analysis of the firing rates and it also acted as a low pass filter to reduce noise.

The LFP signals were first de-referenced by computing their common average, and then detrended to eliminate any low frequency drifts. Power line interference was eliminated using a notch filter (zero-phase Butterworth, 2nd order, f_c = 60 Hz). After this pre-processing, I computed the LFP power in the following frequency bands, obtained using zero-phase filters (2nd order Butterworth): 0-5 Hz, 5-15 Hz, 15-30 Hz, 30-50 Hz, 50-100 Hz, 100 - 200 Hz, and 200 - 400 Hz. Amplitude was obtained using Short Time Fourier Transform with a 2000 samples Hamming window.

2.4. Neural activity and movement analysis

I used multiple analyses to study the relationship between the neural activity and the kinematic parameters recorded.

Correlation matrices

I computed the correlation across firing rate profiles of individual neurons, which is defined by the following equation:

$$Corr\big(x(t), y(t)\big) = \frac{\frac{1}{T}\sum_{t=1}^{T}(x(t)-\bar{x})(y(t)-\bar{y})}{\sqrt{\left(\frac{1}{T}\sum_{t=1}^{T}(x(t)-\bar{x})^2\right)\left(\frac{1}{T}\sum_{t=1}^{T}(y(t)-\bar{y})^2\right)}} \quad (2)$$

Where *x(t)* and *y(t)* correspond to the signals to correlate at each time *t*; and *T* refers to the total timing of both signals. This analysis required that both signals have the same length.

Results of these correlations were presented as a symmetric matrix in which each position indicated the correlation of the row signal with the column one (Figure 3.7). Thus, the diagonal of this matrix was always 1 (i.e. maximum correlation) because it corresponded to the correlation of a signal with itself.

Tuning curves analysis

In this project, I performed and analysis originally done by Georgopoulos AP, 1982 [37], which consisted in computing the average activity of each neuron during the movement time, to obtain a single value indicating how much activity has the neuron had. This was done separately to each target, so at the end I had one single value, representing the mean activity aiming that target. For a reaching task, this function often takes a sinusoidal form.

To get a numerical value of the results I performed a sinusoidal fitting with the first term of a Fourier series as follows:

$$y(\alpha) = a_0 + a_1 \sin(w\, x(\alpha)) + b_1 \cos(w\, x(\alpha)) \quad (3)$$

Where x is the aimed angle, y is the resulted neural activity, and a_o a_1 b_1 and w are the coefficients computed by the fitting.
The r-square value of the fitting will determine how tuned the neuron was.

2.5. Decoding

I used standard methods to decode hand velocity in the x and y directions. These decoders are Multiple Input Single Output systems (MISO).
The algorithm used to decode hand velocity from brain activity was the same for all input types: firing rates, latent variables, and LFPs. This decoder is a model that linearly predicts hand velocity with the specific neural input, as follows:

$$y_j(k) = \sum_{n=1}^{N} x_n(k) \cdot w_n \tag{4}$$

Where $y_j(k)$ represents the velocities at $j = x$ and $j = y$ at each discrete time bin k up to the total of time bins (K); $x_n(k)$ represents the n-th input variable (neurons, latent variables or frequency bands) at k, with a total of N input variables; and w_n represents the contribution of the n-th input variable to the output velocity.
In matrix form, the previous equation is expressed as:

$$\mathbf{Y} = \mathbf{XW} \tag{5}$$

Where **Y** is a K by 2 matrix, with K being the total number of time bins; **X** is a matrix of K by N, being N the total number of input variables; and **W** is a matrix of dimensions N by 2 that indicates the contribution of each input variable to the output.
Notice that matrices **Y** and **W** have two columns because I am computing two MISO decoders, one for the x velocity component and other one for the y component.

Addition of history bins

Motor cortical activity maximally correlates with movement parameters with a few tens of milliseconds delay, from 50 ms to 100 ms on average according to Morrow and Miller [30]. Thus, hand velocity is likely to be better predicted if the model considers not only neural activity during the present time bin, but also activity in the previous bins. Accordingly, I extended the previous model to include past neural activity ("history") at the input. This was done by concatenating delayed versions of the inputs as additional variables (Figure 2.3).

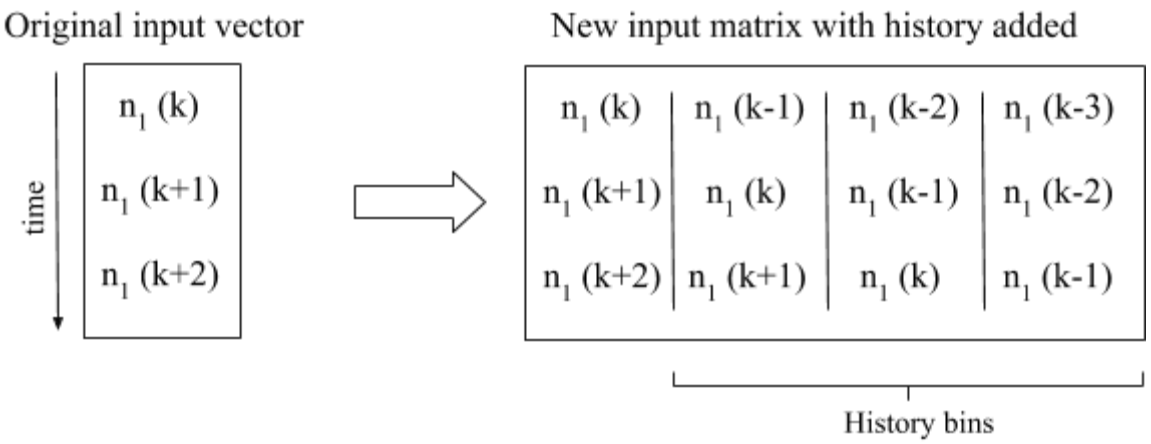

Figure 2.3. Scheme showing the input matrix transformation when history bins are added. 3 history bins has been added to a matrix of one neural input (n_i) in three times (k, k+1, k+2).

However, this is a simplified model, in reality I am adding up to 5 history bins (corresponding to 100 ms) to the total number of neural inputs. This new decoder with history will be obtained as:

$$y_j(k) = \sum_{n=1}^{N} \sum_{b=0}^{B} x_n(k-b) \cdot w_n \qquad (6)$$

Variables in this formula are already explained in Eq.4., except for *b* which is the specific bin of history added up to a total of *B* history bins. Thus, the new input matrix, **X**, had dimensions T times Nx(B+1).

Static nonlinearity

When fitting a linear model to decode movement variables, there is often a nonlinear estimation error [38]. Thus, I also explored whether decoder performance could be improved by adding a static nonlinearity to its output. This is known as Wiener cascade model [39] and operates as follows:

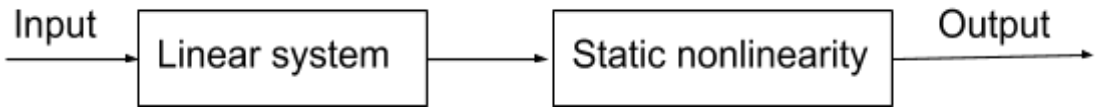

Figure 2.4: Wiener cascade box diagram

The static nonlinearity was computed by fitting a polynomial that captured the relationship between the actual data and the output of the linear decoder. Polynomial fitting was performed using least squares error minimization. Since velocity signals take positive and negative values, odd order polynomials are likely to yield better fits. The fitted polynomial takes the following form:

$$\widehat{y'} = \sum_{i=0}^{p} a_i \hat{y}^i \tag{7}$$

Where i is the polynomial order, a_i are the polynomial coefficients from the fitting, p is the polynomial order, $\hat{y}$ is the estimated velocity obtained directly from the linear model, and $\hat{y}'$ is the estimated velocity after applying the static nonlinearity to $\hat{y}$.

2.6. Latent variables

As explained in the Introduction, recent models propose that cortical processing is best described based on dominant covariance patterns across individual neurons than based on the activity of the individual neurons themselves ([40], [41]). These patterns are often called latent variables [5].

To obtain these latent variables, I applied principal component analysis (PCA) to the neural population firing rates, following standard methods [11][14][42]. PCA is a linear dimensionality reduction method that identifies orthogonal directions in the input data that explain most variance. These vectors, often called principal directions, are sorted based on the amount of variance they explain. Then, the data set is projected to these new directions, obtaining the so called principal components. Since projections onto the last principal directions often explain little variance, they can be removed. This way, a huge dimensional data set can be reduced, maintaining the majority of the data variance.
Consider de neural data matrix **X** with dimensions *K* by *N*, where *N* is the number of recorded neurons and *K* is the number of time bins. Thus, applying PCA implies finding the *N* principal directions. To do so, I used Singular Value Decomposition (SVD) which diagonalizes de covariance matrix as follows:

$$\mathbf{C} = \mathbf{V}\,\mathbf{S}\,\mathbf{V}^{\mathbf{T}} \tag{8}$$

Where **C** is the covariance matrix defined as $\mathbf{C} = \mathbf{X}^{\mathbf{T}}\mathbf{X}$, **V** is an *N* by *N* matrix containing the eigenvectors, and **S** is a diagonal matrix containing the *N* eigenvalues. The eigenvectors obtained are to the principal directions.
Projections of the data set are the new input variables for our linear system, which can be obtained by a matrix multiplication:

$$\mathbf{X'_d} = \mathbf{X}\mathbf{V_d} \tag{9}$$

Where $\mathbf{V_d}$ is a *N* by *d* matrix containing the top *d* principal directions, **X** is aforementioned data matrix, and $\mathbf{X'_d}$ is a *K* by *d* matrix containing the data projected to *d* principal directions, i.e. the latent variables.
The parameter *d* determines the dimensionality of the neural manifold and number of latent variables.
Figure 2.5. shows a simplified example of Principal Component Analysis over a data set. In this case only two neurons activity is considered, creating a two dimensional space, although I am actually using the neural activity of more than 80 neurons, which defines a huge multi-dimensional space. Principal component 1 is the one including the

more variance of the dataset so, ideally, one could reduce the dimensionality of this dataset by projecting all the points exclusively into the first principal component, eliminating the second one, thus, converting the two dimensional data into a one dimensional space. Equivalently, in this project case, I converted a more than 80 dimensional data into a lower dimensional space, depending on the number of latent variables considered.

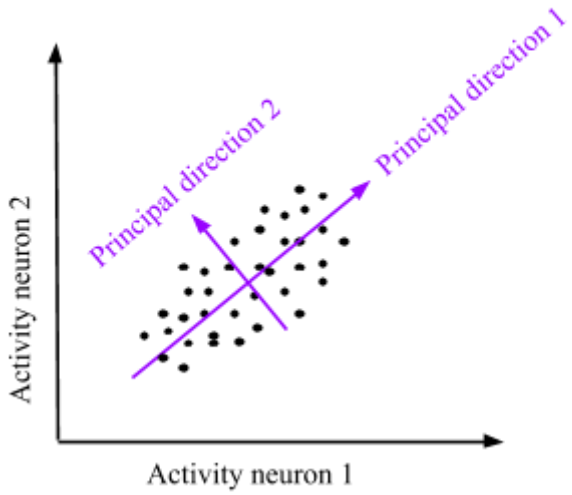

Figure 2.5: Illustration of PCA algorithm in a simplified 2 dimensional dataset (black dots). The original variables (activity of neuron 1 and 2) is changed by the principal direction or neural modes obtained from PCA (purple axis).

2.7. Decoder validation

In this project, I computed several decoders changing the input variables and the parameters previously explained (history bins, static nonlinearity, smoothing of the input signal, etc). Decoders were validated using leave one out multifold cross-validation, maintaining the parameters of the model through the cross-validation. I separated the data into two sets: a training set and a testing one, with the purpose of validating a model computed with the training data, by using testing data only. This tests decoder performance in a more realistic situation, and makes over fitting less likely. The test set was chosen randomly from all the recorded trials and corresponds to the 10% of the total number. The remaining trials were used as training sets.

Assessing decoder performance

I calculated the similarity between actual and decoded hand velocity based on their variance accounted for (VAF).

VAF will return two numbers corresponding to the similarity of the estimated and actual velocities in the x and y directions. VAF is defined as:

$$VAF = 1 - \frac{SSE}{SS} = 1 - \frac{\sum_{k=1}^{K}(y(k) - y'(k))^2}{\sum_{k=1}^{K}(y(k) - \bar{y})^2} \quad (10)$$

Where *SSE* is the sum of the squared difference between the actual and predicted signals and *SS* is the sum of squares of the actual signal; *y* is the mean value of the actual velocity, *y(k)* and *y'(k)* are the respectively actual and predicted velocity values at each time bin, and *K* is the total number of time bins.

The values obtained in each fold of the multifold cross-validation were either averaged, to get a single value indicating the performance, or represented as a histogram, which gave a good representation of the results distribution.

2.8. Statistics

When comparing the performance of different types of decoders, I assessed whether their distributions of cross-validated VAF metrics were statistically different using a Wilcoxon rank sum test.

I used this test with the desired histogram (a distribution) and a null distribution obtained with random observations. This null distribution was obtained differently depending on the histogram I am testing. This procedure is commonly known as Bootstrapping.

Wilcoxon rank sum test, checks the null hypothesis over two distributions. The hypothesis is that both distributions are similar, so the rejection of this null hypothesis means that the two distributions are significantly different.

A 0.1% significance level was set, which means that it will reject the null hypothesis if p-value < 0.001.

Throughout this report, results are reported as mean ± standard deviation, when available.

2.9. Simulating channel loss

The goal of this analysis was to simulate electrode failure by removing input signals to the decoder. For the decoders based on neural population firing rates, I progressively removed the input neurons in steps of 10 % of the total number of neurons. Similarly, for the decoders based on LFPs, I progressively removed 10 % of the input channels (i.e., all seven frequency bands for those channels). Finally, for the decoders based on latent variables, I removed the corresponding percentage of neurons and, re-computed the latent variables based on the remaining population. For each input signal type, I repeated this procedure 5 times and report the mean decoder accuracy.

2.10. Relationship between local field potentials and latent variables

To investigate a potential association between LFPs and latent variables, I used canonical correlation analysis (CCA). In brief, CCA finds linear transformations that applied to each of two sets of signals, make them maximally correlated. Thus, in the context of this project, it compares the dynamics of the LFPs and the latent variables.
Consider the LFP matrix **A,** that has dimensions K by N_1, where K is the total number time bins per trial times the number of trials, and N_1 is the number of frequency bands included times the number of recorded channels. Also consider the latent variable matrix **B**, had dimensions K by N_2, where N_2 was the number of latent variables included. CCA performs a QR decomposition of the previous matrices and obtain the new matrices $\mathbf{Q_A}$ and $\mathbf{Q_B}$. Then the inner product matrix is constructed an a singular value decomposition is performed [43]:

$$\mathbf{Q_A}^{\mathbf{T}}\mathbf{Q_B} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^{\mathbf{T}} \tag{11}$$

Where **S** was a diagonal matrix with $\min(N_1,N_2)$ diagonal elements, which contains the correlation coefficients between the columns of $\mathbf{Q_A}$ and $\mathbf{Q_B}$. The elements of S, which are sorted by value as they are computed, using singular value decomposition, quantify the similarity between the LFP and latent variable dynamics.

3. RESULTS

3.1. Behavioral analysis

From the total of 330 trials, I only analyzed the 286 in which the monkey successfully completed the required movement. The mean reaction time (time to move after the go cue) was 270 ± 60 ms (mean ± s.d.), and the time it took the monkey to complete the task was, on average, 880 ± 90 ms. Thus, the monkey's behavior was very consistent across trials (Figure 3.1). Furthermore, the time course of hand velocity and position was very similar across all the trials to a specific target, as exemplified in Figure 3.2. Figure 3.3. shows the mean hand velocity to each target.

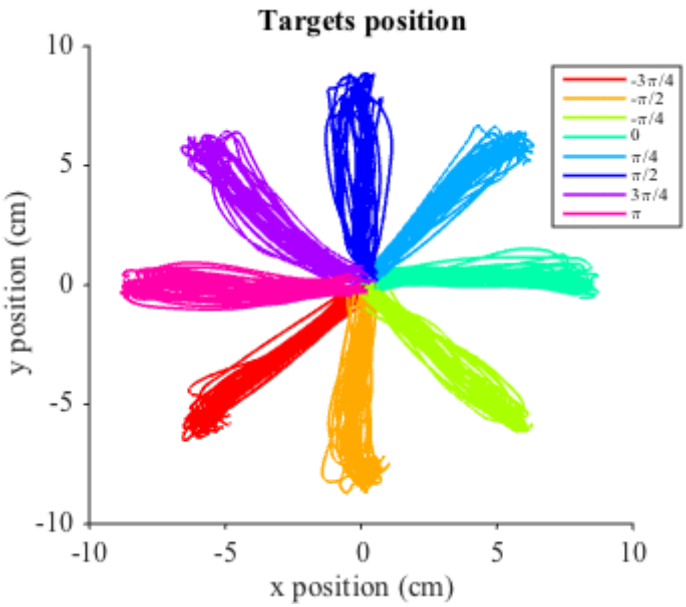

Figure 3.1. Hand trajectories of the subject during all the trials of one session, represented as the position of the manipulandum in the y axis vs the position in that moment in the x axis. Trajectories are color coded by target proximity.

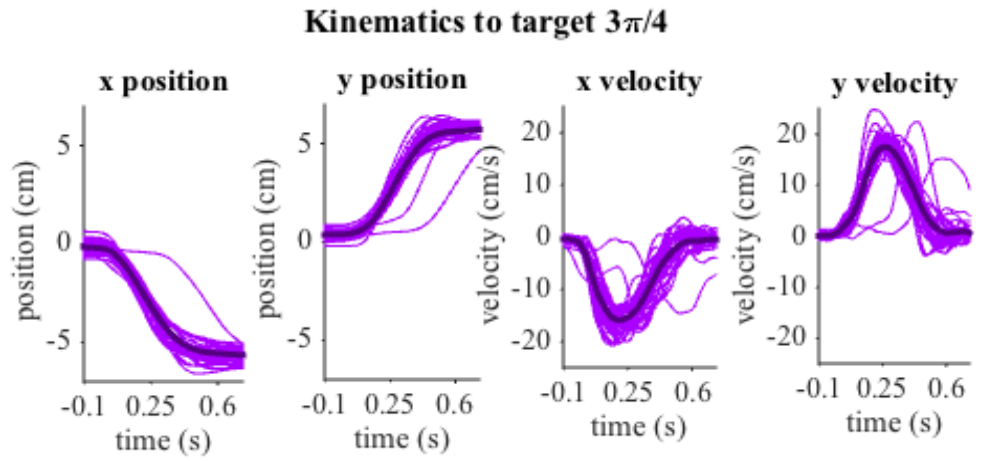

Figure 3.2. All the hand positions and velocities recorded towards target 3π/4 in both directions, x and y. Data shown in window from 100 ms before movement onset to the end of the trial. The individual trials are represented as thin color lines; the average of the trials is represented as a dark thick line. Color is chosen following the previous palette.

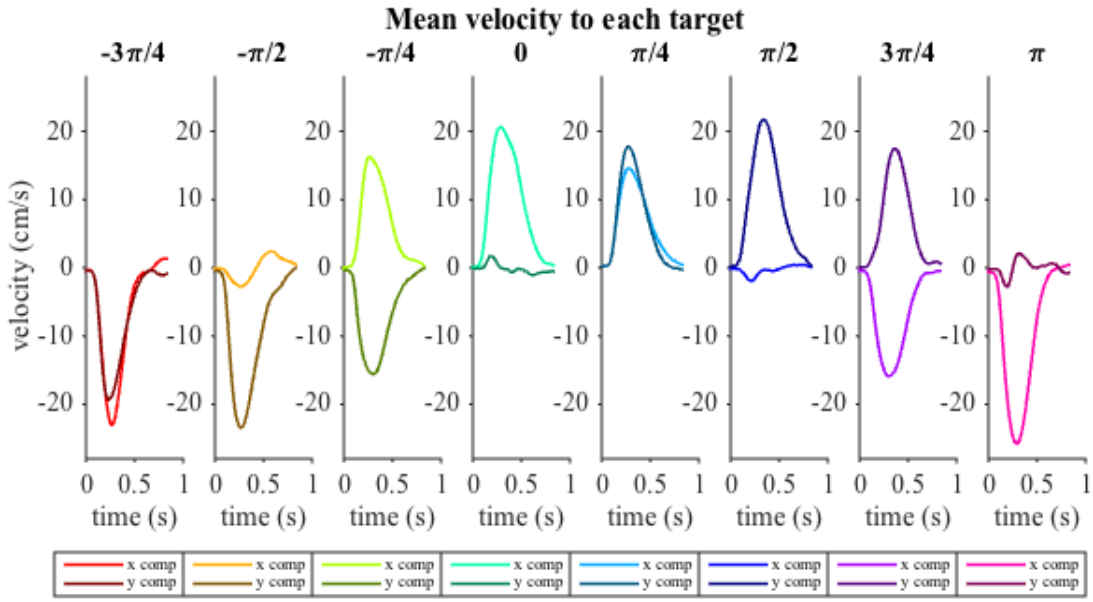

Figure 3.3. Mean hand velocity to each target, from 100 ms before go cue to the end of the trial. Light line corresponds to the x component of the velocity while dark line corresponds to the y component. The targets are colored coded by proximity as in Figure 3.1.

3.2. Single neural activity

3.2.1. Analysis of the neural activity and movement relationship

As a first step to build decoders for movement prediction, I examined the activity of each neuron of the registered population and they relation with the corresponding movement. Figure 3.4. shows the main characteristics of the neural population.

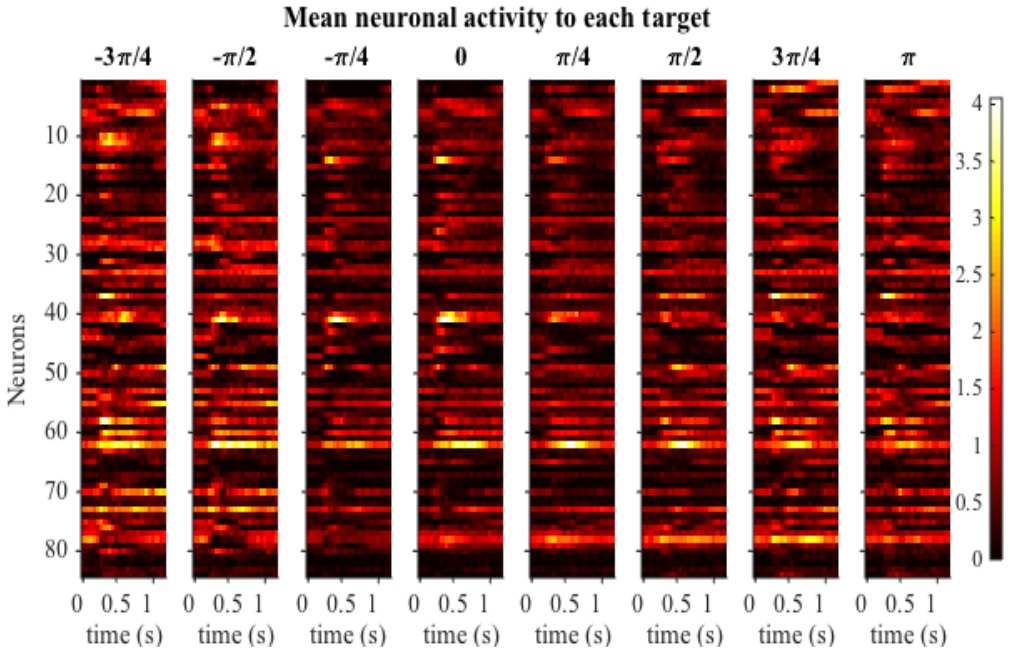

Figure 3.4. Mean activity of each neuron when the monkey reached to each target. Each subplot corresponds to the different targets. In each subplot, each row represents a different neuron (of a total of 84). Data are represented from the go cue until the end of the trial. The number of action potentials in each time bin is indicated in the colorbar to the right.

When looking at the activity to different targets, there are variations in activity for each target. A neuron tends to modulate its firing rate when the monkey reaches to different targets. Moreover, its firing rate profile tends to be similar when the monkey repeatedly performs the same movement. This is exemplified in Figure 3.5: this neuron's activity pattern is very consistent across reaches to the same target, but different across the different targets.

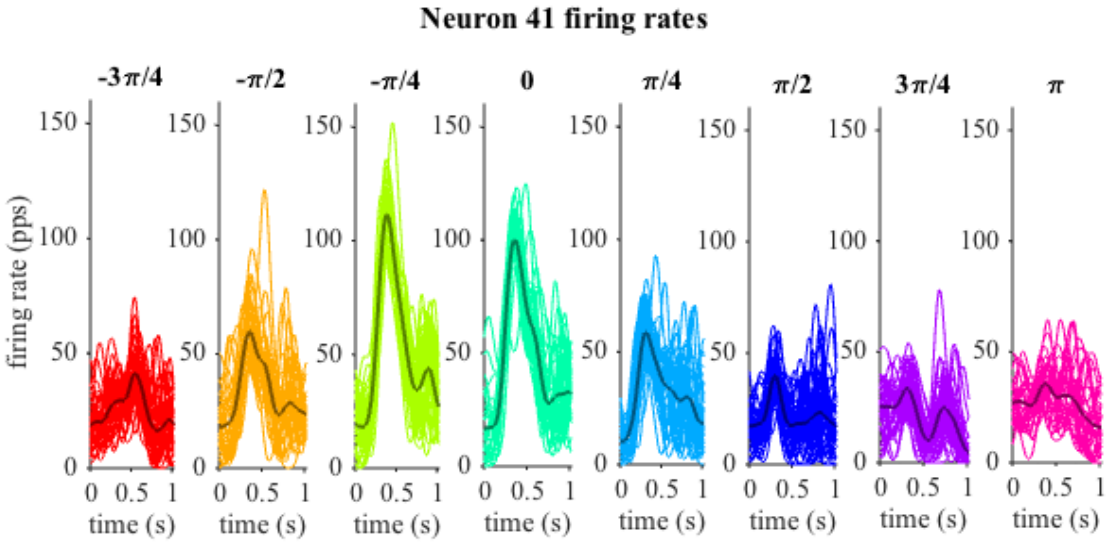

Figure 3.5. Neural activity patterns during firing rates of neuron 41 in all the trials ordered by target location. Data is shown from go cue to trial end. Each colored trace represents one trial; the average neural activity for each target is also represented as a dark line. Firing rate is measured in pulses per second (pps), and targets are color coded according to proximity.

Given the consistency of the neural activity patterns to each target, firing rate profiles were averaged across trials to the same target. Figure 3.6 shows some representative examples of these averaged firing rates. These examples show the complexity of neural activity patterns: some neurons are clearly target-modulated (Figure 3.6A and 3.6B) while others not (Figure 3.6C and 3.6D). They also illustrate that the firing rate of motor cortical neurons can vary very dramatically across cells, e.g. Neuron 41 fires at a >100 Hz for some targets, whereas the other example neurons do not go above 15 pps.

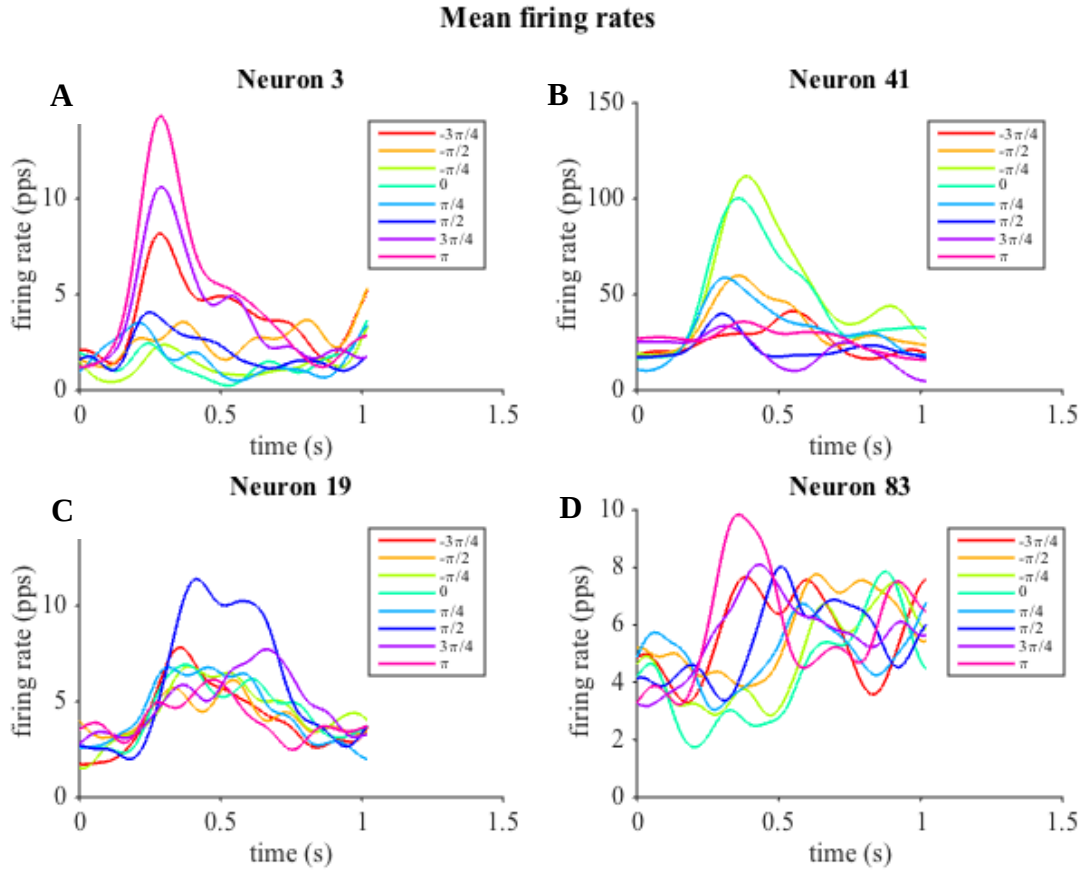

Figure 3.6. Mean neural activity of neuron 3 (**A**), 41 (**B**), as examples of target-modulated neurons, and neurons 19 (**C**) and 83 (**D**), as non target-modulated neurons, during reaches to the eight different targets. Data shown in a window from the go cue to trial end.

To quantify the consistency of a neuron's activity when the monkey reached to the same target, I computed the correlation between its firing rates across all pairs of trials to the same target. Figure 3.5A shows the correlation between the firing rate profiles of a neuron with highly consistent activity across all combinations of reaches to the same target.

The large correlation between firing rate profiles across trials to the same target relates to the large correlation between hand kinematics across those same trials (Figure 3.7B and 3.7C). In fact, the kinematics correlation for those trials with low firing rate correlations, were also low, as evidenced by the linear relationship between firing rate and kinematic correlations (Fig. 3.7D**)**.

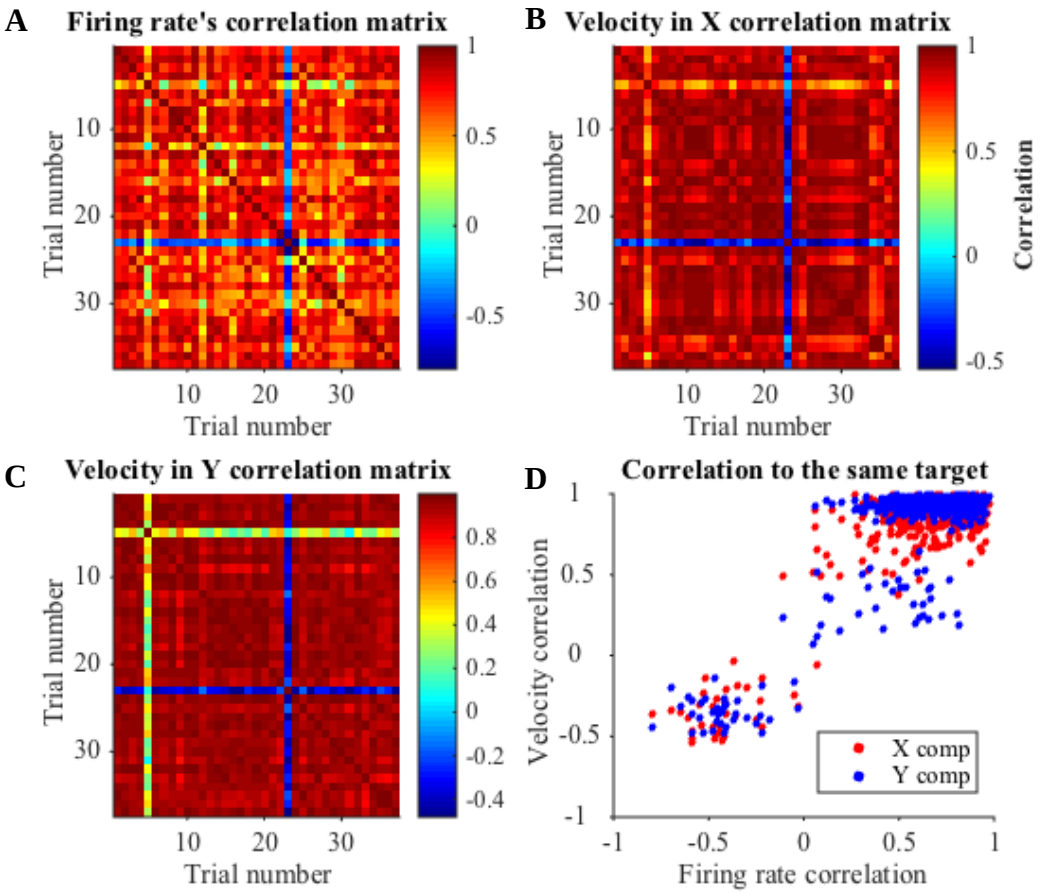

Figure 3.7. **A:** Correlation matrix of the neuron 2 firing rates of all the trials aiming target $3\pi/4$. **B** and **C:** Correlation matrices of velocities in x and y directions (respectively) between all the trials aiming target $3\pi/4$. **D:** Relationship between correlation coefficients of x (red) and y (blue) velocity components with neural activity correlation.

How can we interpret those apparently high neural correlations? To obtain a lower bound for these inter-trial correlations, I performed the same analysis but comparing firing rate profiles across trials to different, randomly selected targets. Figure 3.8**A** shows one example correlation matrix when comparing trials to different targets. Notably, this "null distribution" of neural correlations is very different to the distribution of correlations to trials to the same target (Fig. 3.8**B**; Wilcoxon rank sum test, $p \sim 0$). Consequently, for this example target and neuron, neural activity is significantly more consistent than when reaching to any other target.

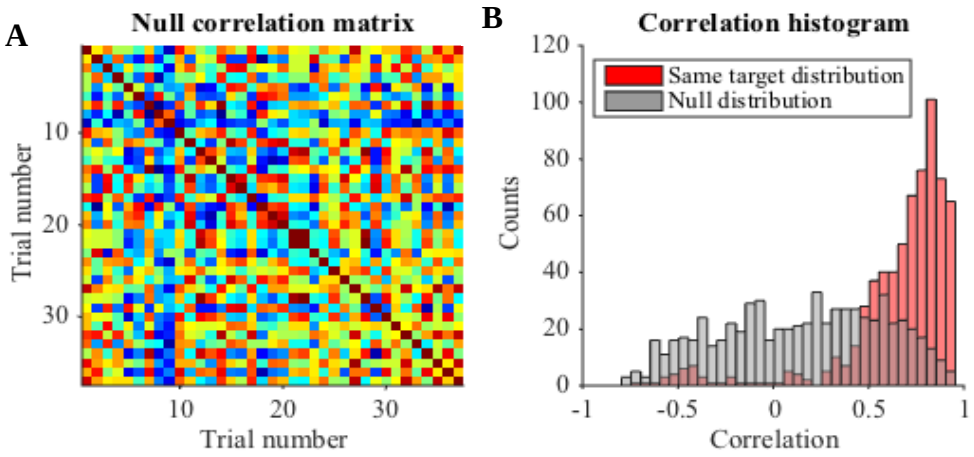

Figure 3.8. **A**: Correlation matrix of neuron 2 firing rate profiles to random targets. **B**: Histogram of the correlation coefficients of neuron 2 to the same target (Figure 3.7**A**) and different target (**A**) represented in red and gray respectively.

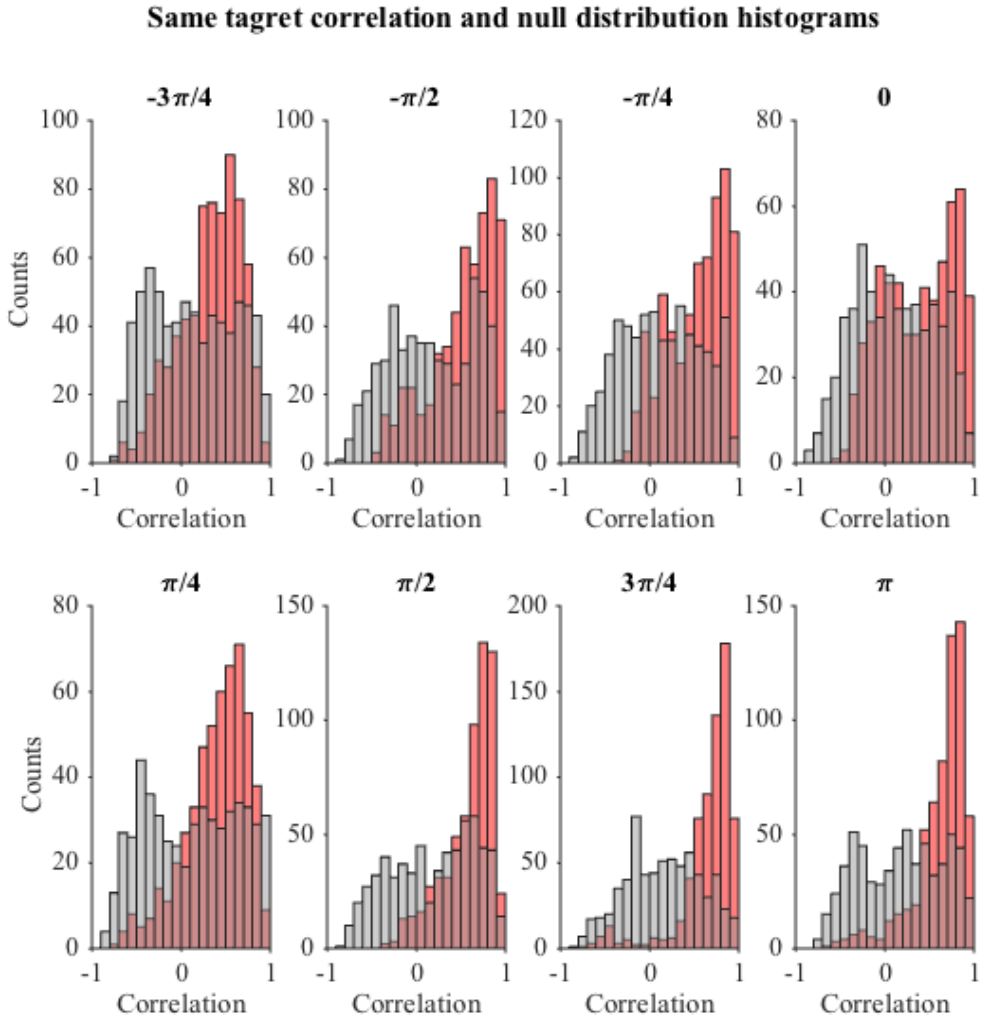

Figure 3.9. Histograms ordered by target containing the correlation coefficients distributions between firing rates to the same target (red) and to random ones (gray). This is a generalized case from Figure 3.8**B**.

Figure 3.9. replicates this result for all the targets, and shows that even though there are some differences in the consistency of neural activity across repetitions, neural activity was always significantly consistent (Wilcoxon rank sum test, $p < 0.001$). Thus, for this example neuron, the firing rates are highly correlated when the monkey is reaching to the same target, independently of which one.

I then repeated this analysis for all recorded neurons, adding up all the inter-trial correlation and null distributions shown in Figure 3.9 and computing their means. These comparisons are shown in Figure 3.10. For all neurons (individual dots), the firing rate was more consistent when the monkey reached to the same target than expected by chance.

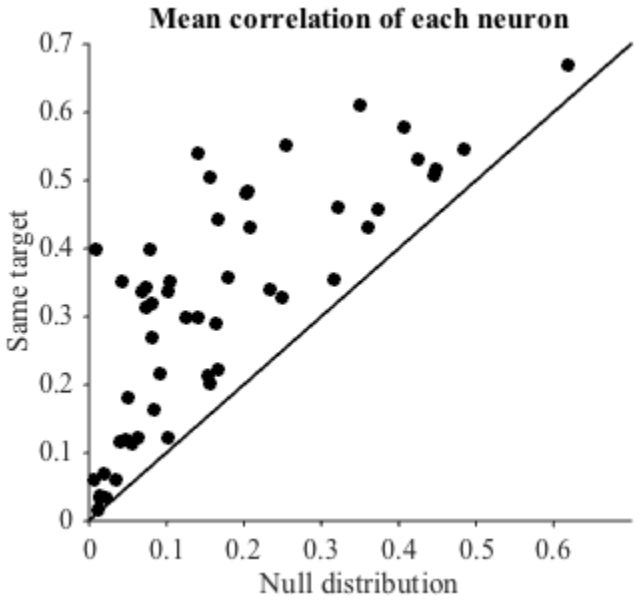

Figure 3.10. Average between the correlation coefficients of firing rates to the same target and to random ones. Each dot represents a neuron. Average correlation is obtained by using the correlation distributions (Figure 3.9) and computing its average value. mean. The black line indicates the limit at which the same-target correlation outvalues the random-target one.

Tuning curve analysis

I studied how neural activity varies as a function of target direction using the classical concept of a "tuning curve." In brief, a tuning curve represents how the activity of a neuron relates to the location of the targets (see Methods for further details). Figure 3.11. shows the activity patterns of two example neurons (top) and their tuning curves (bottom). These are the best and worst cosine-fitted neurons in the entire population.

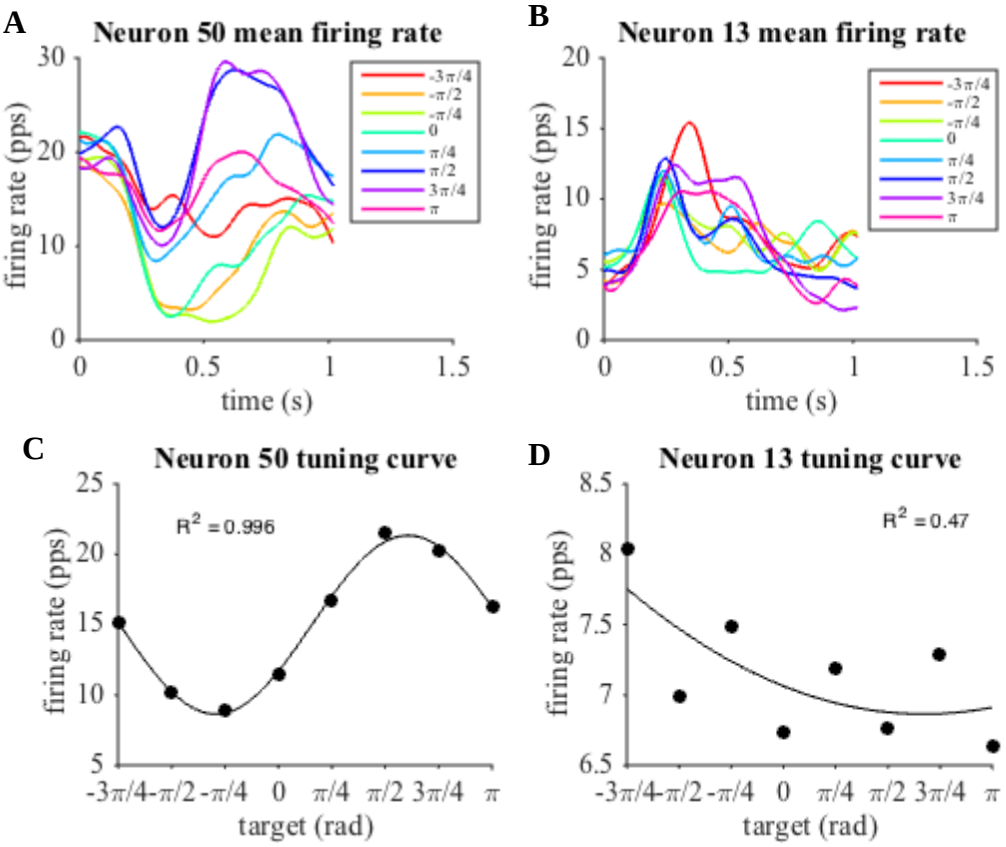

Figure 3.11. Tuning curve of two representative neurons. **A**: Firing rate profiles to each target of the most tuned neuron (number 50). **B**: mean firing rate profiles to each target of the less tuned neuron (number 13). **C,D**: Sinusoidal fit of the mean activity to each target indicating the R^2 of the fitting. Mean activity is obtained by averaging the each target curve in **A** and **B**.

Overall, the activity of the majority of neurons (60 out of 84) was well described by a sinusoidal tuning curve ($R^2 > 0.8$). This implies that most recorded neurons contain information of the spatial location of the targets.

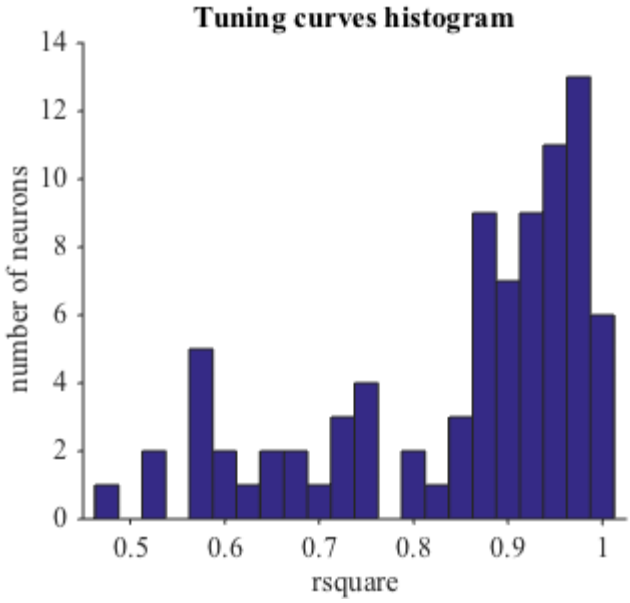

Figure 3.12. Distribution of sinusoidal tuning fit quality (R^2) across all recorded neurons.

3.2.2. Decoders based on neural population activity

Given that the activity of individual neurons has movement-related information, I next built linear decoders that took as inputs this neural population activity [44] (see Methods).

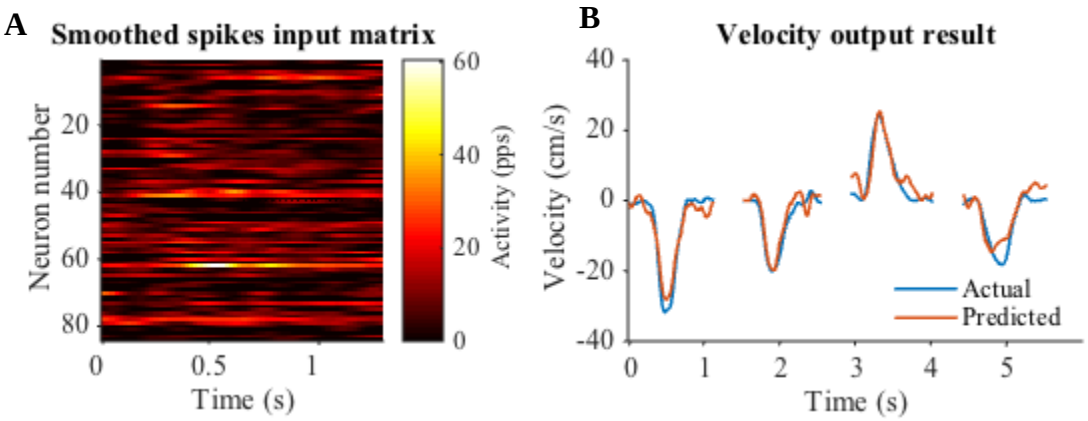

Figure 3.13. **A**: Example of one trial input matrix of smoothed firing rates from all the 84 neurons. Data is trimmed from the go cue to the end of the trial. **B**: Example of x velocity component predicted by the model (orange), together with the actual velocity at that time (blue). The VAF from this prediction was of 0.9.

The output of the decoder will be the predicted velocity of the monkey's hand, which is compared with the actual velocity to estimate the decoder's performance. I am generating two independent decoders, one for each velocity component; hence, all the results will be duplicate. Figure 3.13 shows an example of decoder inputs for one trial (A) and its outputs for four trials (B). This decoder took the binned firing rates of all recorded neurons as inputs, including the current neural activity as well as activity up to 5 bins into the past (see Methods). Comparison of the predicted velocity to the actual velocity showcases that the decoder predicts hand velocity with great accuracy.

The next sections explore how different decoder parameters impact its performance. For all the analyses I used multi-fold cross-validation to avoid overfitting and ensure that the decoders will generalize to different testing sets.

Moderate smoothing of the input signal improves decoder accuracy

As explained in the Methods, smoothing the firing rates is useful for reducing Poisson noise, but it also eliminates data variance, which might negatively impact decoder performance. I thus studied how moderate smoothing with a Gaussian kernel (s.d. 0.05), influences the performance of a decoder with 5 bins of history into the past.

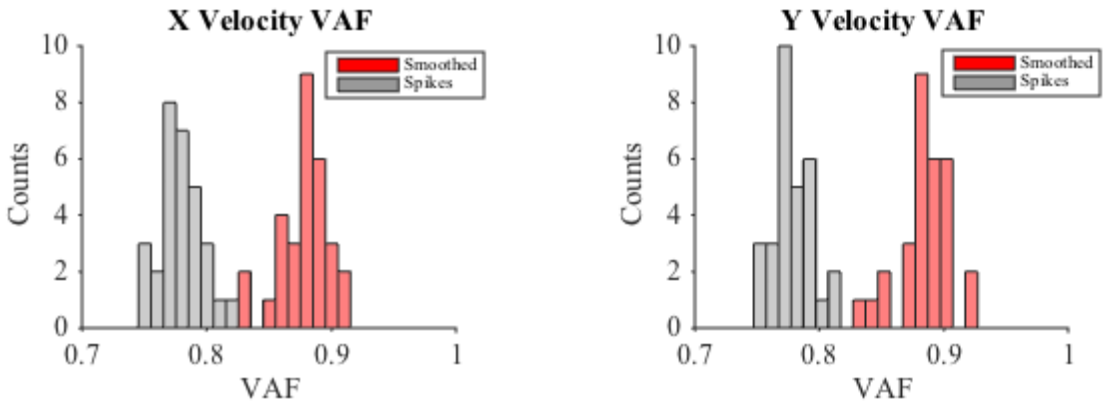

Figure 3.14. Histograms of VAF between predicted and actual velocities at x and y components, obtained by a 30-folds cross validation. Red distributions correspond to decoders trained with smoothed firing rates, while gray distributions correspond to decoders trained with spikes.

Figure 3.14. shows how this slight smoothing did significantly (Wilcoxon rank sum test; p-value < 0.001) improve velocity decoding of both x and y components (mean VAF increases from 0.75 to 0.88 in x component and form 0.78 to 0.88 in y component).

Addition of history to the input data improves decoder accuracy

The decoders used in previous analysis not only took as inputs the population firing rates at the current time, but also in the previous bins, up to 100 ms into the past (see Methods). Here, I explored how decoder accuracy depends on how many bins into the past I included as inputs to the decoder. Adding past neural activity is motivated by observations that neural activity is maximally correlated with movement after adding a 50-100 ms delay. [30] (see Methods).

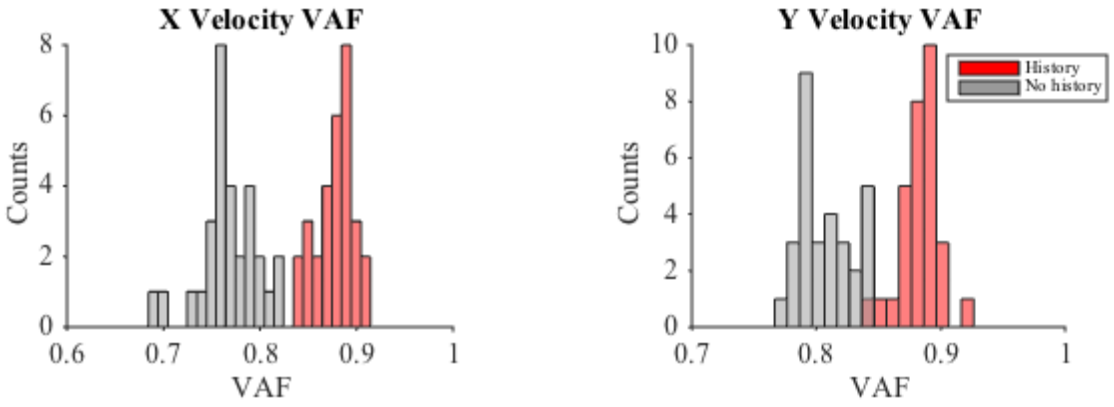

Figure 3.15. Histograms of VAF between predicted and actual velocities at x and y components, obtained by a 30-folds cross validation. Red distributions correspond to decoders trained with history, while gray distributions correspond to decoders without history.

Figure 3.15 shows how decoder accuracy changes when using history bins. To get this figure I took 5 time bins for each velocity time bin, which corresponds to 100 ms considering the time bin size used (20 ms). Decoder performance improved significantly (Wilcoxon rank sum test; $P < 0.001$) when adding history bins.

Figure 3.16 summarizes the effect of the number of history bins on decoder performance. For the x velocity component decoder performance likely decreases for more than 6 history bins because it overfits due to its increasing number of parameters. For the y velocity component, decoder performance plateaued around 4 history bins (Figure 3.16B). Given that decoder performs best for an average of 5 history bins, I will use this number for the remainder of the project.

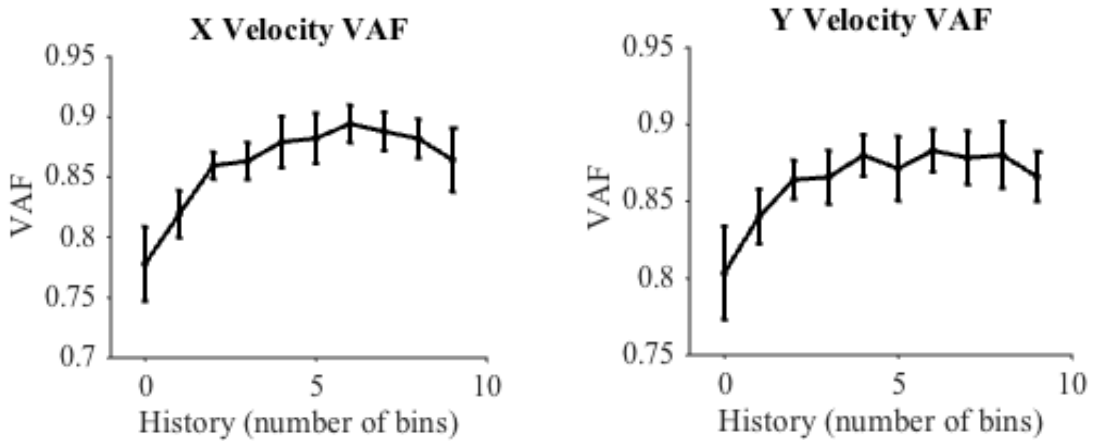

Figure 3.16. VAF between predicted and actual velocities at x and y components, obtained by a 10-folds cross validation using different history duration. Bars in each point show the standard deviation of the cross validation with each time of history.

Application of static nonlinearity at the decoder output

Previous studies have shown that adding a static nonlinearity at the output of a linear decoder may improve decoder performance [38]. Here I explored whether this was the case for the velocity predictions in this dataset (see Methods).

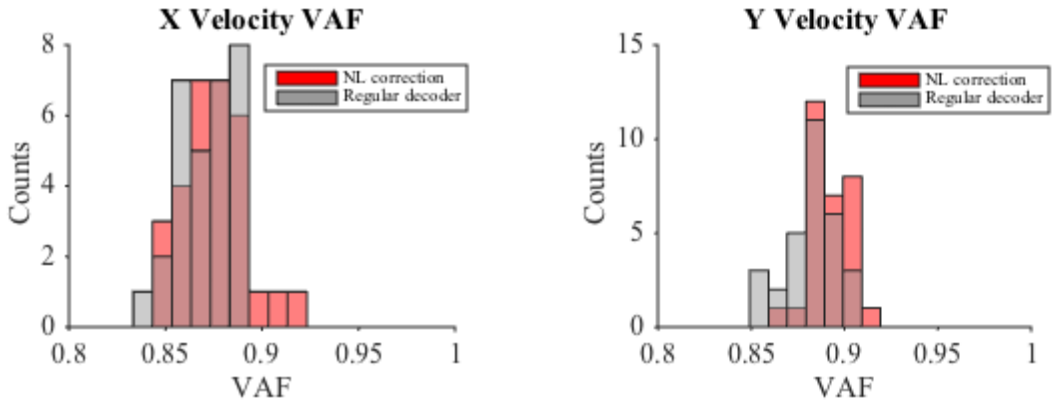

Figure 3.17. Histograms of VAF between predicted and actual velocities at x and y components, obtained by a 30-folds cross validation. Red distributions correspond to decoders with static nonlinearity, while gray distributions correspond to decoders without it.

Figure 3.17 compares decoder performance obtained after a 30 folds cross correlation with a static nonlinearity using a 3rd order polynomial. The polynomial order was selected because it gave the best results. Adding this nonlinearity to the decoder did not significantly improve decoder performance (Wilcoxon rank sum test; p-values: 0.78 and 0.67 respectively).

3.3. Latent variables

3.3.1. Describing neural population activity using latent

As mentioned in the Introduction, a recent theoretical framework proposes that neural computation is based on population-wide latent variables, rather than on the independent modulation of individual neurons [11]. Here, I obtained the latent variables using PCA, the most common dimensionality reduction method [12] (see Methods). PCA finds the covariance patterns from the data and uses them as new directions. The new directions are sorted by the variance they explained, so one can easily eliminate the ones that barely explain variance.

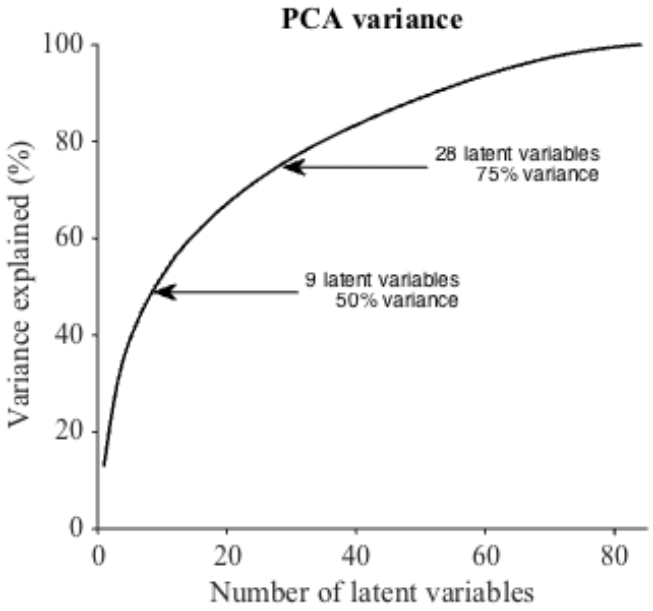

Figure 3.18. Percentage of variance explained by the dataset as a function of the number of principal directions considered. Arrows indicate the relevant points (approximately 50% and 75% of variance explained) together with the number of latent variables considered.

Figure 3.18 shows how neural population activity can be reduced to a small number of population-wide activity patterns. For example, nine of these population-wide activity patterns or latent variables explain as much as 50% of the total neural variance, and 28, 75%. This number is quite small compared to the 84 neurons included in the population.

3.3.2. Relationship between latent variables and movement

Since the latent variables obtained with PCA are linear combinations of the neural population firing rates (Methods), they should also include movement-related information. Figure 3.19 shows three example latent variables and how they are modulated by target location. Note that, although the latent variable activity is still measured in pps (pulse per second) since it is the linear combination of the activity of all the neurons; I use arbitrary units because it does not have conceptual sense.

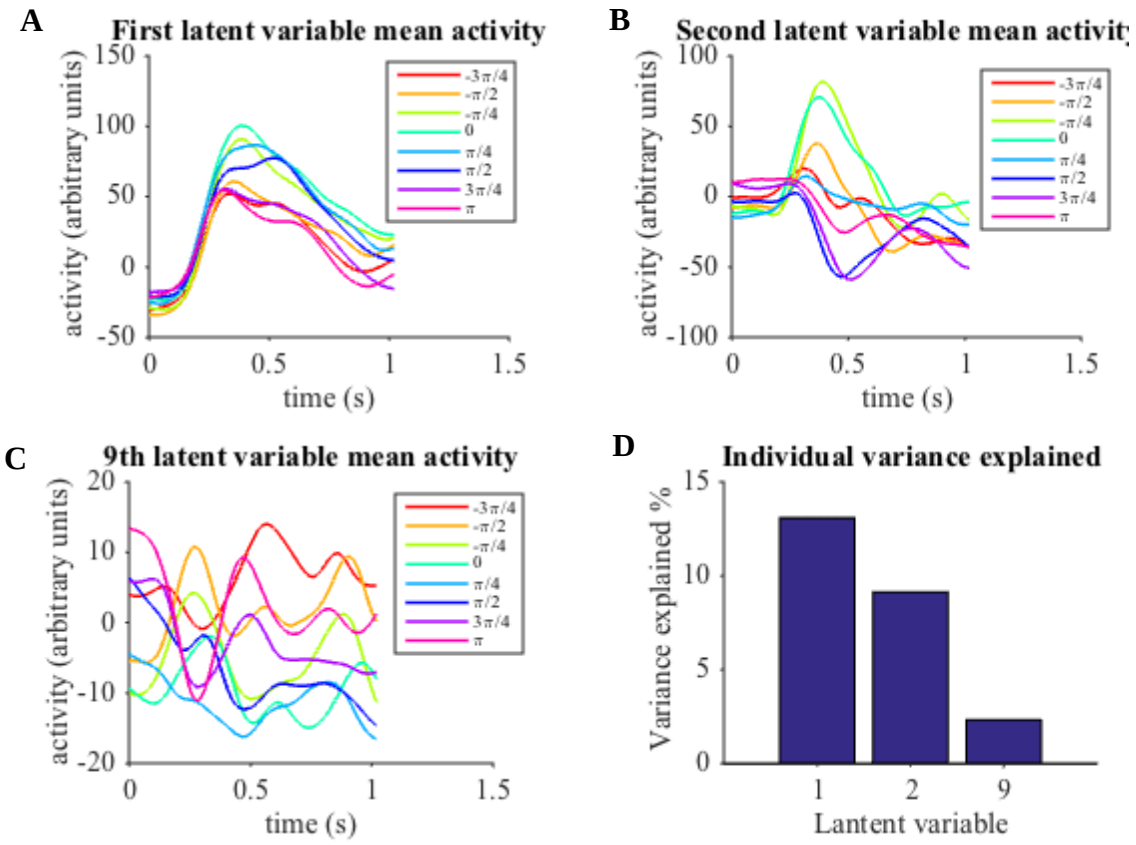

Figure 3.19. **A**, **B** and **C:** Mean activity in the first, second and 9th latent variables (respectively) for each aimed target. Data is shown in a window from the go cue to trial end. **D:** Additional figure showing the individual percentage of variance explained of each latent variable.

The first latent variable does not present significant tuning to target location, although it describes ~13 % of the total neural variance (Figure 1.19D). Instead, this latent variable has a steep rise at movement onset, consistent with previous reports that the dominant latent variable reflects the timing of movement [45]. In contrast, the second latent variable has dynamics that clearly depend on the location of the target. Finally, latent variable 9 is only slightly tuned.

The observed movement-related tuning of the latent variables suggests that they can be used to predict movement, as shown in previous studies. [12]

3.3.3. Latent variable-based decoders

Here I examined the performance of decoders based on latent variables, rather than the population firing rates as done in Section 3.2. However, I maintain the linear algorithm for decoders with 5 history bins from the neural activity section.

Given that only 9 latent variables explain 50% of the total neural variance, I first built decoders that took these 9 latent variables as inputs (see Methods). Figure 3.20 compares the performance of these decoders with that of decoders based on the neural population spiking. Interestingly, even though the latent variable-based decoders have fewer inputs than the population firing based-decoders (9 vs. 84), their performance is quite similar, albeit statistically different (Wilcoxon rank sum test; $P < 0.001$). Mean VAF form the distributions (both for x and y components) differs in 0.04, which implies only a decrease of 5% in performance.

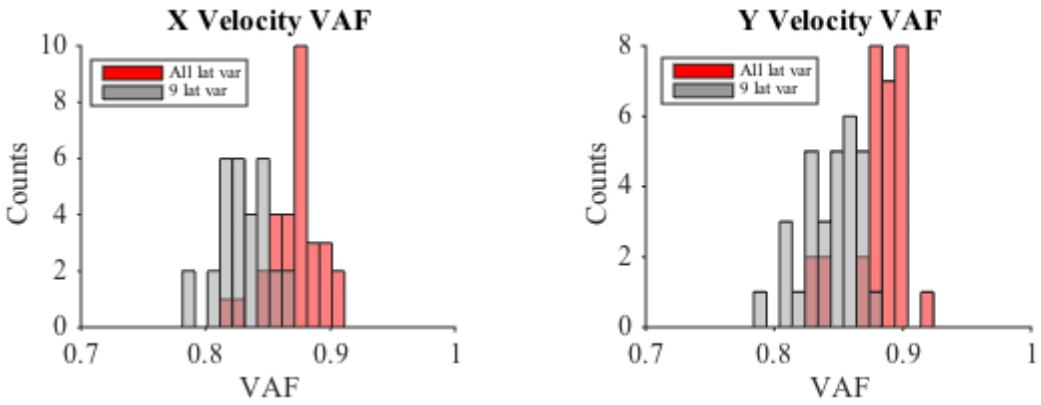

Figure 3.20. Histograms of VAF between predicted and actual velocities at x and y components, obtained by a 30-folds cross validation. Red distributions correspond to decoders with all the latent variables, while gray distributions correspond to decoders with just 9 of them.

To study to what extent the number latent variable inputs influence prediction accuracy, I computed the mean VAF for decoders from a 10 folds cross validation based on an increasing number of latent variables (Figure 3.21).

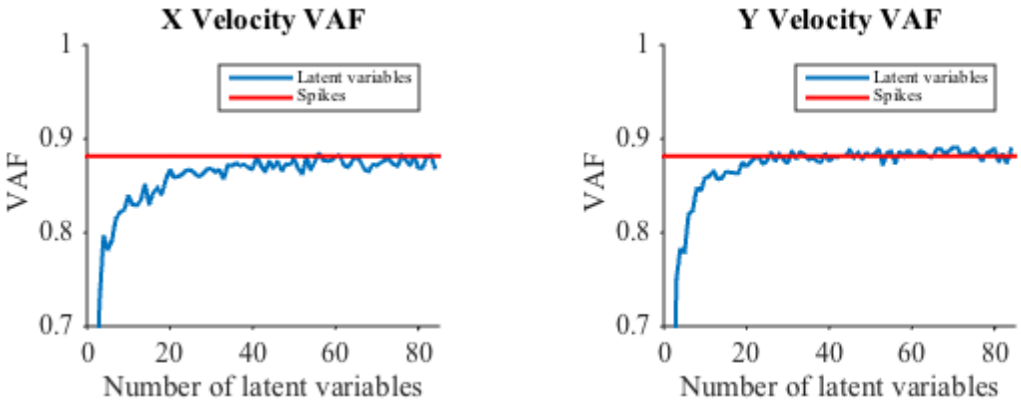

Figure 3.21. VAF between predicted and actual velocities at x and y components, obtained by a 10-folds cross validation from decoders with latent variables as input parameters (blue) or firing rates (red).

Decoder performance improves rapidly for the first few latent variables and plateaus around ~20 latent variables, when decoding performance becomes similar to that of the decoder based neural population firing rates.

3.4. Local Field Potentials

3.4.1. LFP and movement relationship

In the previous sections, I have shown how neural population firing rates and latent variables yield accurate predictions of hand kinematics. Here, I compare their performance with that of decoders based on intracortical LFPs, the voltage fluctuations detected by each implanted electrode.

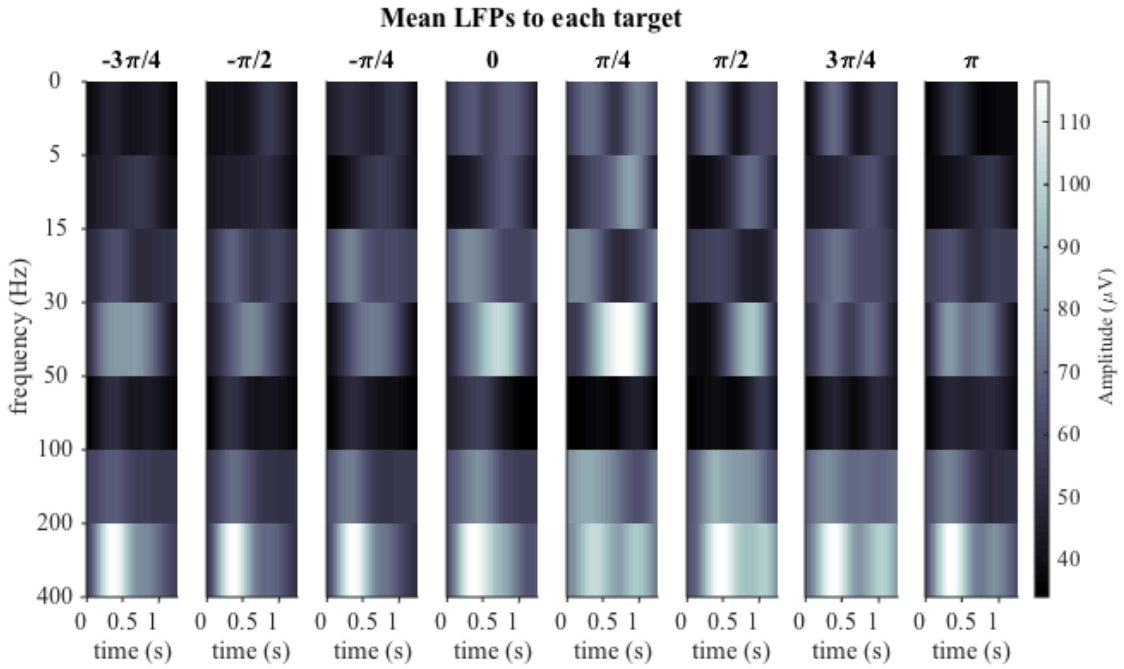

Figure 3.22. Mean LFP amplitude at each frequency bands for channel 3 when reaching each target, from go cue to the end of the trial. LFP amplitude in each frequency band is obtained from STFT of the signal in that specific band.

Figure 3.22. shows, for one representative electrode, how the mean LFP power at different frequency bands (see Methods) changes across targets. For some frequency bands, such as the 30-50 Hz band, we observe modulation by target location.

First, I studied whether they present the same modulation in all the trials when reaching for the same target. To do so, I computed the correlation of the same target trials in each bands at the same channels. Similar to Figure 3.5. The correlation matrices are represented in Figure 3.23 and Figure 3.24.

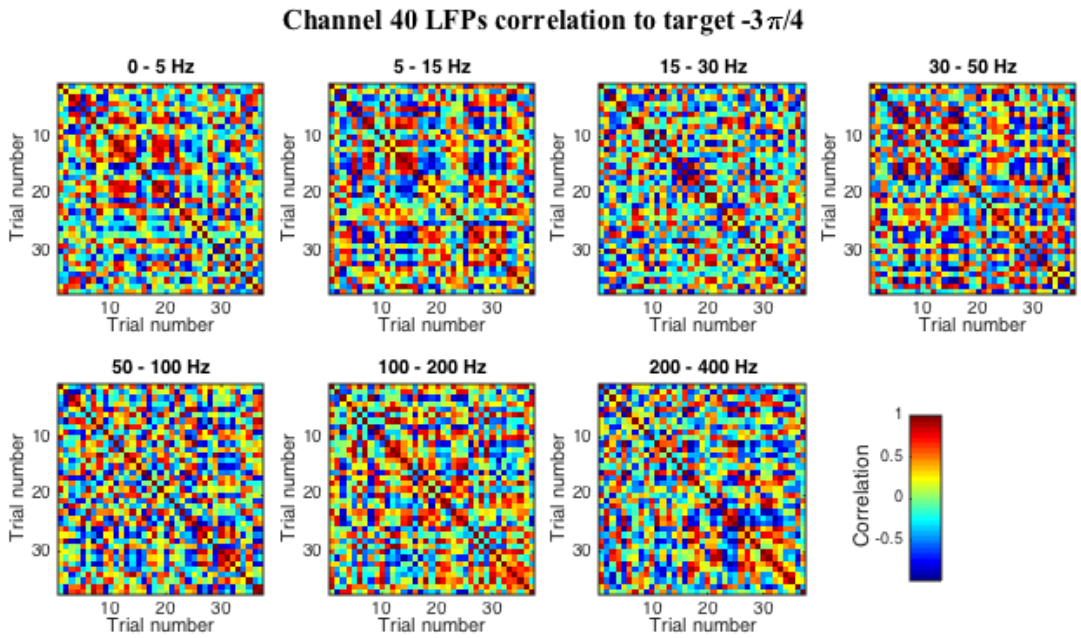

Figure 3.23. Correlation matrices in channel 40 for all the trials which target direction was -3π/4, as an example of uncorrelated channel when reaching the same target. Each correlation matrix corresponds to a frequency band indicated in the title.

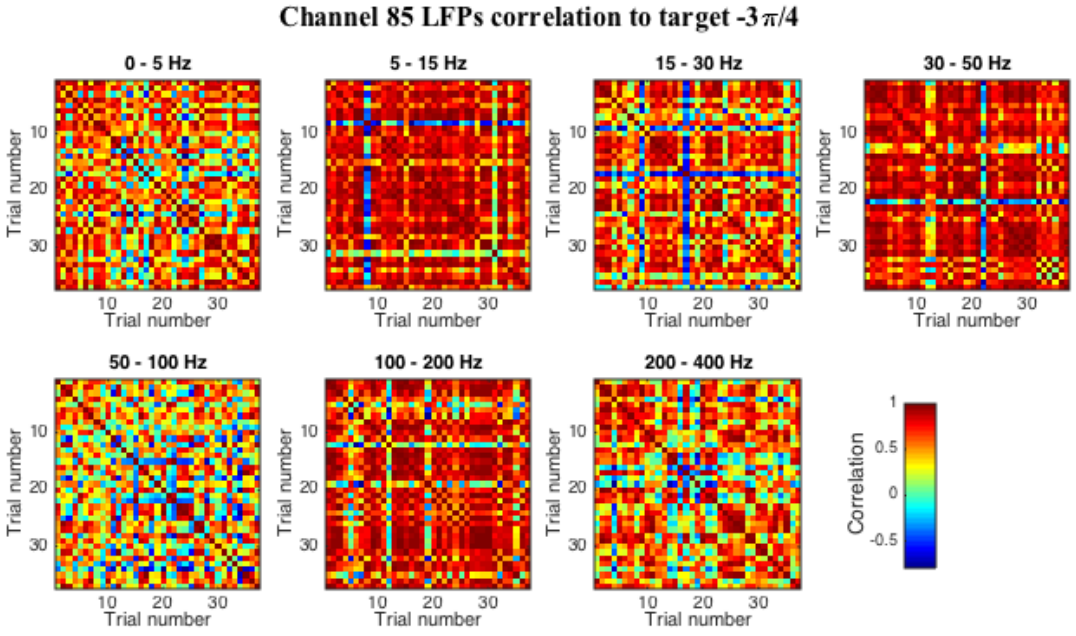

Figure 3.24. Correlation matrices in channel 85 for all the trials which target direction was -3π/4, as an example of correlated channel when reaching the same target. Each correlation matrix corresponds to a frequency band indicated in the title.

For some channels, such as the one exemplified in Figure 3.23, there were large inter-trial changes in LFP amplitude in all frequency bands, which led to low inter-trial correlations. For others, LFP amplitude was very consistent across reaches to the same target (Figure 3.24). I excluded the channels with low inter-trial correlations for the decoder analysis, as they would decrease decoder performance.

Then, I check whether the modulation by target becomes more clearly when representing the LFP amplitude in a single frequency band. Figure 3.25 shows the LFP amplitude of two different channels at the delta band (0 - 5 Hz). The activity in both example channels is modulated by target location even though their amplitudes are quite different.

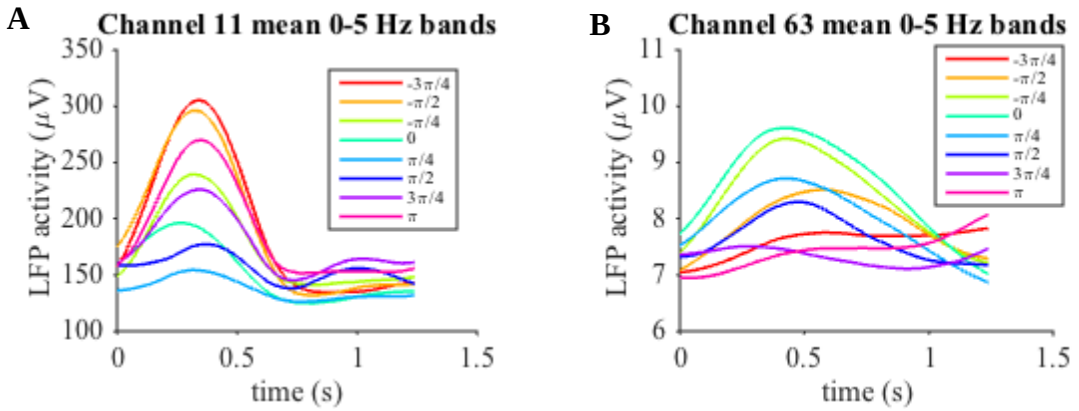

Figure 3.25. **A, B:** Mean LFP amplitude in 0 - 5 Hz bands between all the trials reaching the eight different targets of channels 11 and 63 (respectively). Data is shown in a window from the go cue to trial end.

To quantify movement tuning across all electrodes and frequency bands, I performed the same tuning curve analysis as for the firing rates (see Methods, and Figure 3.11). Figure 3.26 summarizes the quality of fit of these tuning curves for the different LFP frequencies The mean R^2 among the frequency bands varied from 0.7 ± 0.19 to 0.75 ± 0.16, indicating that there is significant tuning at all frequencies. However, this average tuning is ~15 % lower than for the neural firing rates (see Figure 3.12).

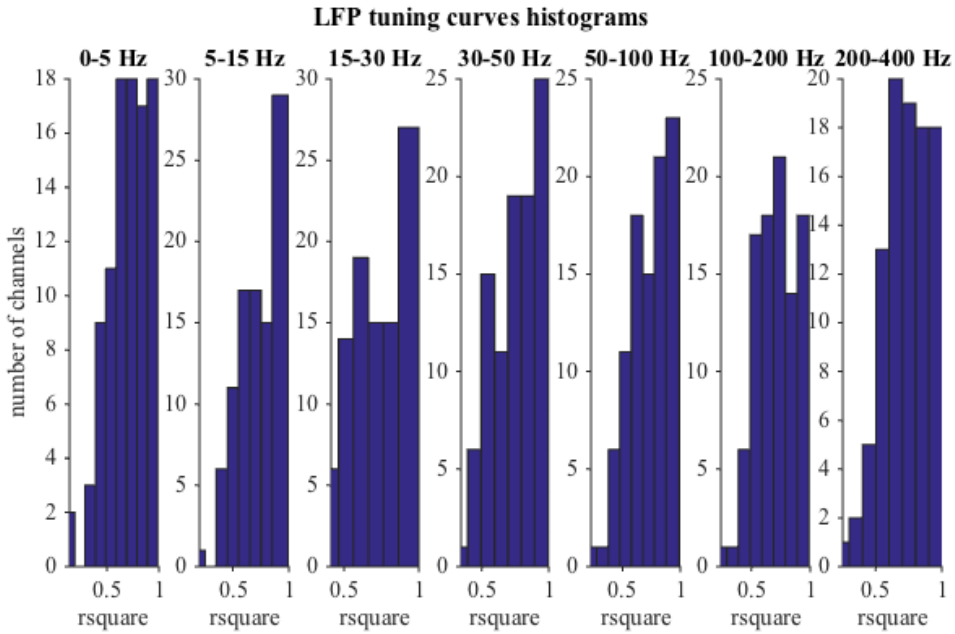

Figure 3.26. R-squared coefficients of the tuning curve's sinusoidal fitting. Tuning curves where obtained separately for each frequency band for all the channels. Amplitudes were averaged from go cue to the end of the trial

3.4.2. LFPs-based decoders

After showing that LFP activity has movement-related information, I built velocity decoders based on the LFP amplitude at different frequency bands. These decoders had a similar structure as the population firing rate- and latent variable-based decoders. As mentioned above, I only considered electrodes with consistent inter-trial activity

As in the previous decoders, to maximize the performance, the following decoders will be computed using 100 ms of history.

Decoder performance typically depended on the LFP frequency band chosen for the inputs. For example, Figure 3.27. shows two example decoders; in this case high-frequency bands yielded more accurate predictions than low frequency bands, a difference that was only significant for the X axis (Wilcoxon rank sum test, $p < 0.001$; for the Y axis, Wilcoxon rank sum test, $p = 0.97$).

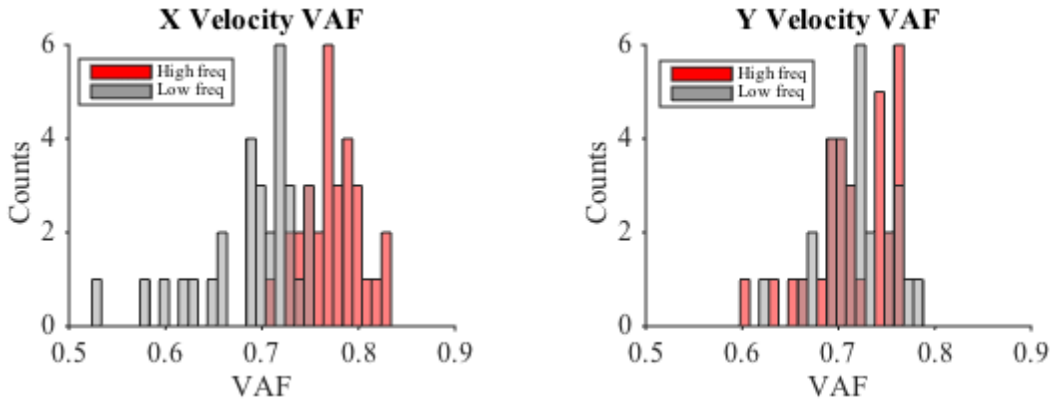

Figure 3.27. Histograms of VAF between predicted and actual velocities at x and y components, obtained by a 30-folds cross validation. Red distributions correspond to decoders using frequencies 30 - 400 Hz, while gray distributions correspond to decoders using frequencies 0 - 30 Hz.

To further study the different LFP frequency bands, I computed decoders using only the time-varying amplitude at one frequency band as inputs (Figure 3.28). All these single-band decoders yielded worse average predictions than decoders that took all frequency bands as inputs, probably because of the drastically decrease of the number of inputs (only 1 out of 7 total input variables are used). Decoders based on low gamma frequencies (30-50 and 50-100 Hz) were the most accurate among them.

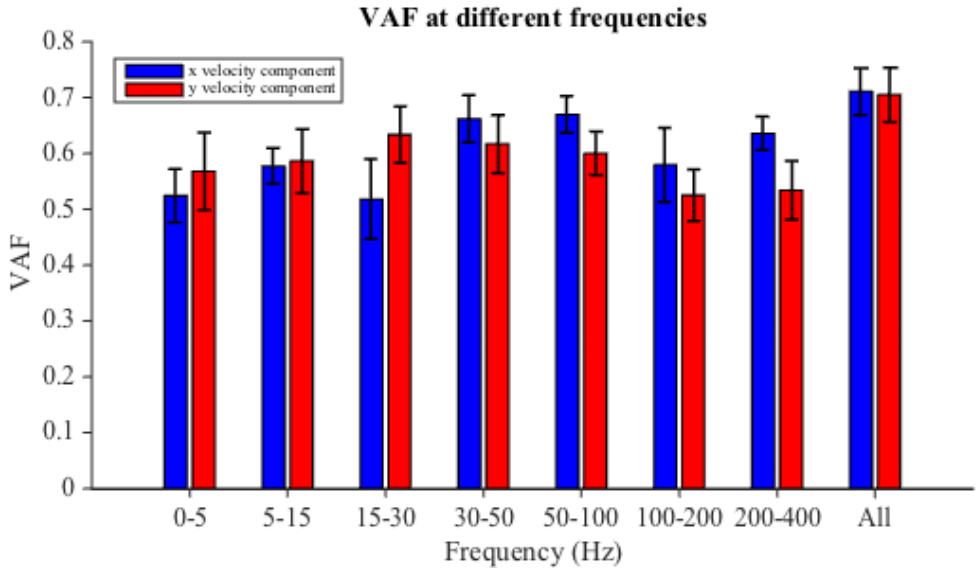

Figure 3.28. Mean VAF of x and y velocity components obtained by a 10-folds cross validation using individually each frequency band, and all the frequency bands at the end. Standard deviation after the cross validation is indicated as error bars.

3.5. Comparison of decoder inputs

3.5.1. LFP and population firing rates comparison as decoder inputs

When comparing LFP-based decoders and population firing rate-based decoders, the latter yielded significantly better performance ($P<0.001$, Wilcoxon rank sum test). However, LFPs are still useful signals for velocity decoding, since their mean VAF is ~0.7.

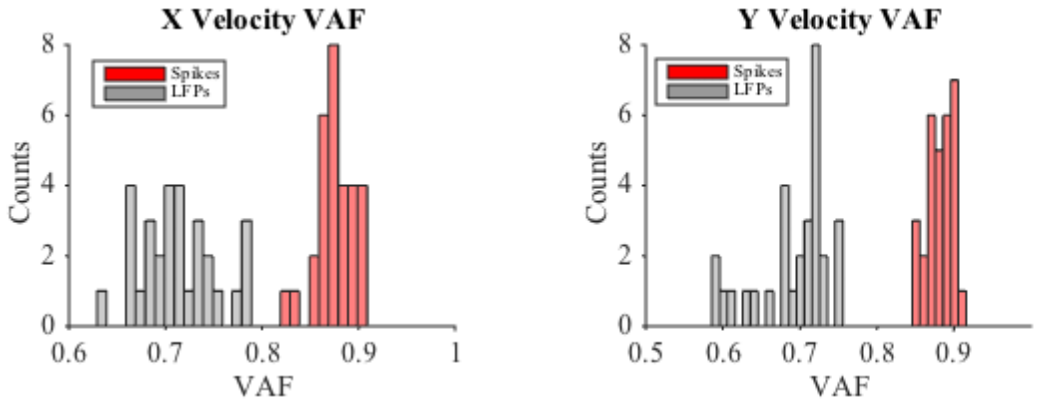

Figure 3.29. Histograms of VAF between predicted and actual velocities at x and y components, obtained by a 30-folds cross validation. Red distributions correspond to decoders using firing rates, while gray distributions correspond to decoders using LFPs.

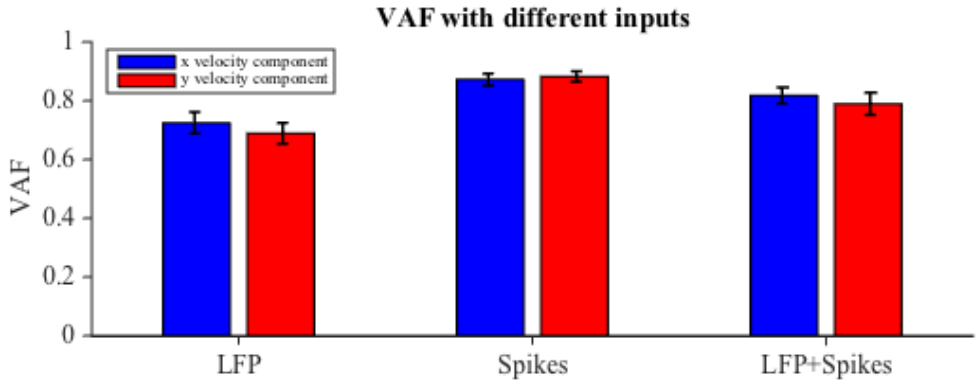

Figure 3.30. Mean VAF of x and y velocity components obtained by a 10-folds cross validation using either LFP, spikes or both as input for the decoder. Standard deviation after the cross validation is indicated as error bars.

To further investigate the relative predictive power of LFPs and population firing rates, I built hybrid decoders that combined both types of inputs [32]. LFP's frequency bands and neuron's firing rates are used as inputs. The addition of LFP inputs to the good-performing population firing rate-based decoder decreases significantly its performance (Wilcoxon rank sum test; p-value < 0.001). This results are consistent with prior studies in hybrid decoders [32], [34].

3.5.2. Decoder robustness as function of the number of inputs

Making BCI decoders that are robust to degradation of input signals is a key challenge in the field [21]. Thus, here I studied which among LFP-based, neural population firing rates-based, or latent variable-based decoders is more robust to input channel loss. To this end, I gradually decreased the number of input variables of the decoders and assessed their performance (see Methods).

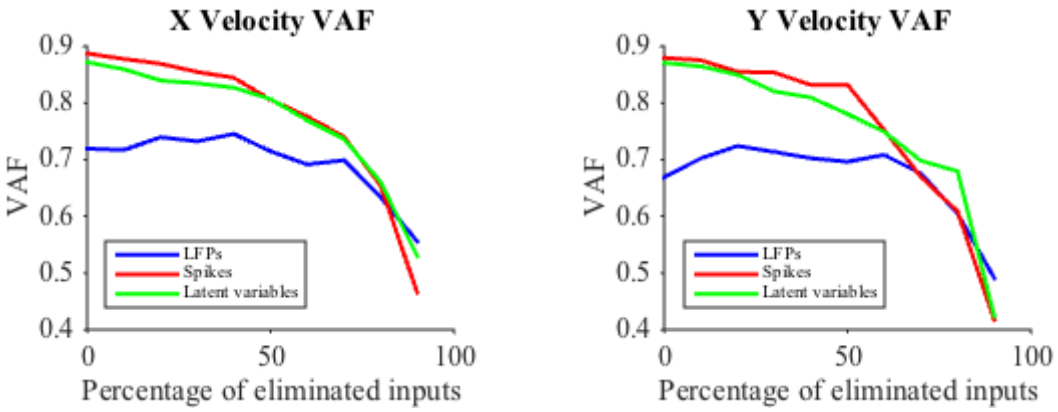

Figure 3.31. VAF between predicted and actual velocities at x and y components, obtained by a 5-folds cross validation at 5 different input combinations, as the number of inputs is reduced. Inputs considered where LFPs (Blue), neuron's firing rates (Red) and 10 latent variables (Green).

LFP-based decoders are more robust against channel loss: they maintain the same accuracy even after removing ~50% of the channels. However, their performance is the worst among the three types of decoders I studied. Both the accuracy and robustness of decoders based neural population firing rates and latent-variables are similar although only 10 latent variables are used.

3.5.3. Comparison of LFPs and Latent variables

By definition, the latent variables and the neural population firing rates are linear combinations of each other (see Methods). But what is their relationship with LFPs? Here I examined for the first time the instantaneous relationship between the time course of the latent variables and the LFPs. To this end, I used Canonical Correlation Analysis (CCA), a method that compares point by point two sets of time-varying signals (see Methods) [43]. Using CCA, I studied the similarity (i.e, pairwise correlations) between the dynamics of the latent variables and the LFPs at different bands.

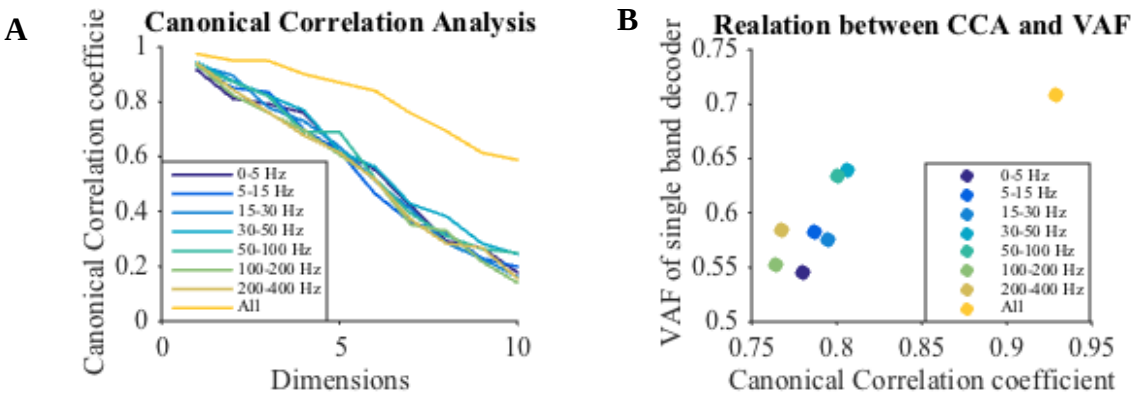

Figure 3.32. **A**: Coefficients obtained from CCA between LFP bands (independently and all together) and the 10 first latent variables. **B**: Relationship between the averages of the 5 first canonical correlation coefficients and the VAF of the decoders of each frequency band separately (Figure 3.28)

CCA shows that there is a strong correlation between the LFP activity pooled across frequency bands and the latent variables (mean correlation of the first 5 coefficients > 0.9 in Figure 3.32B). In comparison, all individual LFP frequency bands show a much lower correlation (Figure 3.32A). Interestingly, there is an association between the predictive power of the LFP inputs and their similarity of the activity at that band with the latent variables (Figure 3.32B). This suggests an intriguing relationship between the activity in specific LFP bands and the neural population dynamics represented by the latent variables.

4. DISCUSSION

In this project, I studied the relationship between motor cortical LFPs and latent variables, two types of neural signals known to include movement-related information. After replicating results in previous reports demonstrating that these two types of signals can be successfully used as BMI inputs, I showed that LFPs are slightly more robust to electrode loss than neural-based signals are. Finally, I leveraged a novel approach based on canonical correlation analysis to reveal that the dynamics of the LFPs and the latent variables are strongly related to each other. This intriguing unreported finding may enable the development of more robust BMI decoders to restore movement to neurological patients.

4.1. Neural population-based and latent variable-based decoders

The innovative approach explored in this project was to try and relate the dynamics of the LFPs with those of the latent variables that capture the activity of neural populations in the same brain area. Before investigating this relationship, I tested the predictive power of the different neural signals, mostly replicating recent results in the scientific literature. I found that neural population-based decoders provide the best accuracy. Latent variable-based decoders reached the same performance when using only 20 neural modes, suggesting that the activity in higher (lower variance) neural modes do not contribute with new movement-related information for this task. Furthermore, decoders using only 10 latent variables as inputs achieved >80% maximum performance, and even outperformed the quite accurate predictions obtained with LFP-based decoders.

4.2. Local field potential-based decoders

A comparison of the predictive power of LFP oscillations in different frequency bands showed that decoding performance changes slightly across bands. I obtained the best results for the low gamma bands: 30-50 Hz and 50-100 Hz, which contradicts some prior studies, where the best performance was obtained using high gamma LFP activity: 50-300 Hz [34], 200-300 Hz [27]. A third study reported the 200-400 Hz band as the best one for decoding [32], although their decoders were not accurate as mine —note

that they studied a different, more complex task. Note that this strong preference for higher frequencies does not hold when doing online BMI control using LFPs: K. So et al reported that different subjects have different "preferred LFP bands," which led to better BMI performance than the others [15].

In agreement with previous comparative studies, my LFP-based decoders performed worse than decoders based on neural population activity [27], [32] or latent variables [26]. The appeal of LFP-based decoders is their intrinsically longer stability over time [34], a topic that was out of the scope of this project. I did however simulate loss of recording electrodes by progressively eliminate input signals, i.e. neurons or LFPs channels. Interestingly, LFP-based decoders were the most robust against input channel lost. Latent variable-based decoders also outperformed neural population-based decoders, likely because movement-related information is largely population-wide [43]. This was the case even though, my neural population-based decoders were more robust than what is typically reported in prior studies [27], [34], indicating that this dataset contained a particularly "movement-tuned" neural population.

4.3. Relationship between local field potentials and latent variables

To begin exploring what movement-related information is shared between LFPs and neural population activity, I studied how the dynamics of the latent variables and the LFPs at different frequency bands related to each other. I found that the dynamics of the combined LFP activity including all the frequency bands was strongly correlated with the latent variables within the neural manifold. Individual frequency also showed some correlation with the latent variables, although it was significantly lower for all the combined bands. Intriguingly, there seems to be a linear relationship between the predictive power of each LFP frequency band and its similarity with the latent variables (Figure 3.31).

4.4. Limitations and Future Work

Although my analyses reproduced results in several published studies, and even suggested a previously unreported relationship between the dynamics of the LFPs and the latent variables, this study has several shortcomings. First, I had to discard some LFP signals as they were largely inconsistent across trials when the monkey reached to the same target, contrary to the activity of individual neurons or many LFP channels. Moreover, I built my decoders based on the peri-movement periods (i.e. from the go cue to the end of each trial), not the whole recording times. Discarding the inter-trial epochs, when linear decoders often yield some small false positives, tends to artificially increase the amount of VAF. Perhaps for this reason, the addition of a static non-linearity did not improve decoder performance significantly. Finally, I only analyzed one dataset, although I did cross-validate all the analyses to ensure the generalizability of my results. Importantly, all these shortcomings can be potentially solved in my future work.

In future work, it would be interesting to replicate these analyses to investigate the relationship between the dynamics of the LFPs and the latent variables during movement preparation, the interval when the monkey plans the movement before executing it. I have access to several datasets that include simultaneous recordings from dorsal motor cortex (PMd), a "higher" cortical area that integrates sensory and visual information and has been greatly implicated in motor planning [26], during this same task. Another interesting analysis would be to investigate the stability of the LFPs over time, by testing the performance of a decoder trained on "Day 1" on subsequent days. This analysis could be enriched by studying whether the relationship between the LFPs and the latent variables is also stable across days.

4.5. Conclusions

I have used a novel analytical approach to report that the inputs to LFP-based decoders and neural population-based decoders have quite similar dynamics. Follow up analyses may further our understanding of the relationship between these two types of neural signals. Combining the present results with recent developments in the neural manifold framework may enable the development of decoders that lead to robust BMI performance over unprecedented periods of time.

5. SOCIO-ECONOMIC IMPACT

5.1. Research impact

This project do not constitute and important economic gain, since no monetary benefit is coming from it. However, it does imply an important research advance on the neurological field. The task of understanding brain functionality is one of the most difficult challenges for this new century. This individual project constitutes the first step of a long period of research in which actual neurological hypothesis will be proven right, or rejected, generating new ones.

The relevance of the project is not restricted to the understanding of brain functionality for the sake of knowledge. Completely understanding the brain would give brain machine interfaces the proper methodology to follow to satisfactory substitute the natural movement control for people with motor disabilities. Moreover, understanding how the brain learns to develop new tasks could be the key for restoring lost brain functionality after a neurological accident (i.e. stroke), or even a new born disease (i.e. Amyotrophic lateral sclerosis (ALS)).

5.2. Budget

The project costs are divided in two main groups: human resources and materials.

Human resources costs			
Concept	**Cost/hour**	**Working hours**	**Total cost**
Student	20 €	600 h	12,000 €
Supervisor	40 €	180 h	7,200 €
		Subtotal	19,200 €

No lab technician was needed for this project. The total student hours are estimated based on the working schedule and the extra hours for writing the bachelor thesis. Estimation of supervisor hours is based on a 30 % of supervised time, including time for revisions and reunions.

Material costs			
Concept	**Initial cost**	**Dedication**	**Proportional cost**
Laptop	1500 €	8 months	87.5 €
Computer	3500 €	2 months	58 €
Matlab license	500 €/year	8 months	291.7 €
Office 2010	100 €/year	8 months	58.3 €
		Subtotal	495.5 €

Both computer and laptop expenses are estimated by a 10 year lifetime.

The animal care budget was not included in these estimations since the data was recorded long time ago and all their expenses were covered by the Northwestern University (Chicago, IL, USA).

No industrial benefit comes from this project, but a general cost of 16% (of the material costs) and 21% taxes should be included. Thus, the final budget is:

Material costs	
Type of cost	**Cost**
Human resources	19,200 €
Materials	495.50 €
General costs	74.48 €
Total without taxes	19,769.98 €
FINAL (Total + taxes)	23,921.68 €

6. LEGAL FRAMEWORK

Regarding the software, the totality of the project was developed using Matlab with a student license [46]. It was not required a general license since the implemented code is exclusive for research purposes, thus, will not be commercialized.

All the Matlab data was organized in a structure that comprises many functions and classes developed by the research group previously. The codes used for this project remain as intellectual property of M. G. Perich, J. A. Gallego, and R. H. Chowdhury, although they could be used in future researches.

Regarding the project itself, I used neural and behavioral data recorded from a monkey (macaca mulatta) at Northwestern University (Chicago, IL, USA) in 2016. All surgical and experimental procedures were approved by Northwestern's Animal Care and Use Committee [47].

7. BIBLIOGRAPHY

[1] J. J. Daly and J. R. Wolpaw, “Brain–computer interfaces in neurological rehabilitation,” *Lancet Neurol.*, vol. 7, no. 11, pp. 1032–1043, Nov. 2008.

[2] A. Ramos-Murguialday *et al.*, “Brain–machine interface in chronic stroke rehabilitation: A controlled study,” *Ann. Neurol.*, vol. 74, no. 1, pp. 100–108, 2013.

[3] N. Mrachacz-Kersting *et al.*, “Efficient neuroplasticity induction in chronic stroke patients by an associative brain-computer interface,” *J. Neurophysiol.*, vol. 115, no. 3, pp. 1410–1421, Mar. 2016.

[4] E. Widmaier, H. Raff, and K. Strang, *Vander’s Human Physiology: The Mechanisms of Body Function, 13th Edition*, 13 edition. New York: McGraw-Hill Education, 2013.

[5] D. J. Felleman and D. C. Van Essen, “Distributed hierarchical processing in the primate cerebral cortex,” *Cereb. Cortex N. Y. N 1991*, vol. 1, no. 1, pp. 1–47, Feb. 1991.

[6] R. Shadmehr and S. P. Wise, *The computational neurobiology of reaching and pointing: A foundation for motor learning*. Cambridge, MA, US: MIT Press, 2005.

[7] F. H. Martini, R. B. Tallitsch, and J. L. Nath, *Human Anatomy*, Edición: 9. NY, NY: Pearson, 2017.

[8] K. H. Srivastava *et al.*, “Motor control by precisely timed spike patterns,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 114, no. 5, pp. 1171–1176, Jan. 2017.

[9] W. T. Thach, “Correlation of neural discharge with pattern and force of muscular activity, joint position, and direction of intended next movement in motor cortex and cerebellum,” *J. Neurophysiol.*, vol. 41, no. 3, pp. 654–676, May 1978.

[10] M. M. Morrow, L. R. Jordan, and L. E. Miller, “Direct comparison of the task-dependent discharge of M1 in hand space and muscle space,” *J. Neurophysiol.*, vol. 97, no. 2, pp. 1786–1798, Feb. 2007.

[11] J. A. Gallego, M. G. Perich, L. E. Miller, and S. A. Solla, “Neural Manifolds for the Control of Movement,” *Neuron*, vol. 94, no. 5, pp. 978–984, Jun. 2017.

[12] C. Pandarinath *et al.*, “Latent Factors and Dynamics in Motor Cortex and Their Application to Brain–Machine Interfaces,” *J. Neurosci.*, vol. 38, no. 44, pp. 9390–9401, Oct. 2018.

[13] P. T. Sadtler *et al.*, “Neural constraints on learning,” *Nature*, vol. 512, no. 7515, pp. 423–426, Aug. 2014.

[14] J. P. Cunningham and B. M. Yu, “Dimensionality reduction for large-scale neural recordings,” *Nat. Neurosci.*, vol. 17, no. 11, pp. 1500–1509, Nov. 2014.

[15] K. So, S. Dangi, A. L. Orsborn, M. C. Gastpar, and J. M. Carmena, “Subject-specific modulation of local field potential spectral power during brain–machine interface control in primates,” *J. Neural Eng.*, vol. 11, no. 2, p. 026002, Apr. 2014.

[16] J. Zhuang, W. Truccolo, C. Vargas-Irwin, and J. P. Donoghue, “Decoding 3-D Reach and Grasp Kinematics from High-Frequency Local Field Potentials in Primate Primary Motor Cortex,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 7, pp. 1774–1784, Jul. 2010.

[17] Department of Electrical and Electronic Engineering, Assiut University, Assiut, Egypt, M. Abo-Zahhad, S. M. Ahmed, and S. N. Abbas, “A New EEG Acquisition Protocol for Biometric Identification Using Eye Blinking Signals,” *Int. J. Intell. Syst. Appl.*, vol. 7, no. 6, pp. 48–54, May 2015.

[18] J. L. Collinger *et al.*, “High-performance neuroprosthetic control by an individual with tetraplegia,” *The Lancet*, vol. 381, no. 9866, pp. 557–564, Feb. 2013.

[19] A. B. Ajiboye *et al.*, “Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration,” *The Lancet*, vol. 389, no. 10081, pp. 1821–1830, May 2017.

[20] E. E. Fetz, “Operant Conditioning of Cortical Unit Activity,” *Science*, vol. 163, no. 3870, pp. 955–958, Feb. 1969.

[21] S. J. Bensmaia and L. E. Miller, “Restoring sensorimotor function through intracortical interfaces: progress and looming challenges,” *Nat. Rev. Neurosci.*, vol. 15, no. 5, pp. 313–

325, May 2014.
[22] J. R. Wolpaw *et al.*, “Independent home use of a brain-computer interface by people with amyotrophic lateral sclerosis,” *Neurology*, vol. 91, no. 3, pp. e258–e267, Jul. 2018.
[23] J. del R. Millán and J. M. Carmena, “Invasive or noninvasive: understanding brain-machine interface technology,” *IEEE Eng. Med. Biol. Mag. Q. Mag. Eng. Med. Biol. Soc.*, vol. 29, no. 1, pp. 16–22, Feb. 2010.
[24] W. Truccolo, G. M. Friehs, J. P. Donoghue, and L. R. Hochberg, “Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia,” *J. Neurosci. Off. J. Soc. Neurosci.*, vol. 28, no. 5, pp. 1163–1178, Jan. 2008.
[25] L. R. Hochberg *et al.*, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia,” *Nature*, vol. 442, no. 7099, pp. 164–171, Jul. 2006.
[26] J. A. Gallego, M. G. Perich, R. H. Chowdhury, S. A. Solla, and L. E. Miller, “A stable, long-term cortical signature underlying consistent behavior,” *bioRxiv*, p. 447441, Nov. 2018.
[27] R. D. Flint, C. Ethier, E. R. Oby, L. E. Miller, and M. W. Slutzky, “Local field potentials allow accurate decoding of muscle activity,” *J. Neurophysiol.*, vol. 108, no. 1, pp. 18–24, Jul. 2012.
[28] S.-P. Kim *et al.*, “A comparison of optimal MIMO linear and nonlinear models for brain–machine interfaces,” *J. Neural Eng.*, vol. 3, no. 2, pp. 145–161, May 2006.
[29] D. Sussillo *et al.*, “A recurrent neural network for closed-loop intracortical brain–machine interface decoders,” *J. Neural Eng.*, vol. 9, no. 2, p. 026027, Apr. 2012.
[30] M. M. Morrow and L. E. Miller, “Prediction of Muscle Activity by Populations of Sequentially Recorded Primary Motor Cortex Neurons,” *J. Neurophysiol.*, vol. 89, no. 4, pp. 2279–2288, Apr. 2003.
[31] A. S. Dickey, A. Suminski, Y. Amit, and N. G. Hatsopoulos, “Single-unit stability using chronically implanted multielectrode arrays,” *J. Neurophysiol.*, vol. 102, no. 2, pp. 1331–1339, Aug. 2009.
[32] A. K. Bansal, W. Truccolo, C. E. Vargas-Irwin, and J. P. Donoghue, “Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials,” *J. Neurophysiol.*, vol. 107, no. 5, pp. 1337–1355, Mar. 2012.
[33] D. Sussillo, R. Jozefowicz, L. F. Abbott, and C. Pandarinath, “LFADS - Latent Factor Analysis via Dynamical Systems,” *ArXiv160806315 Cs Q-Bio Stat*, Aug. 2016.
[34] S. D. Stavisky, J. C. Kao, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, “A high performing brain–machine interface driven by low-frequency local field potentials alone and together with spikes,” *J. Neural Eng.*, vol. 12, no. 3, p. 036009, Jun. 2015.
[35] J. M. Carmena *et al.*, “Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates,” *PLOS Biol.*, vol. 1, no. 2, p. e42, Oct. 2003.
[36] M. G. Perich, J. A. Gallego, and L. E. Miller, “A Neural Population Mechanism for Rapid Learning,” *Neuron*, vol. 100, no. 4, pp. 964-976.e7, Nov. 2018.
[37] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, “On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex,” *J. Neurosci. Off. J. Soc. Neurosci.*, vol. 2, no. 11, pp. 1527–1537, Nov. 1982.
[38] E. A. Pohlmeyer, S. A. Solla, E. J. Perreault, and L. E. Miller, “Prediction of upper limb muscle activity from motor cortical discharge during reaching,” *J. Neural Eng.*, vol. 4, no. 4, pp. 369–379, Dec. 2007.
[39] I. W. Hunter and M. J. Korenberg, “The identification of nonlinear biological systems: Wiener and Hammerstein cascade models,” *Biol. Cybern.*, vol. 55, no. 2, pp. 135–144, Nov. 1986.
[40] P. Gao and S. Ganguli, “On simplicity and complexity in the brave new world of large-scale neuroscience,” *Curr. Opin. Neurobiol.*, vol. 32, pp. 148–155, Jun. 2015.
[41] K. V. Shenoy, M. Sahani, and M. M. Churchland, “Cortical Control of Arm Movements: A Dynamical Systems Perspective,” *Annu. Rev. Neurosci.*, vol. 36, no. 1, pp. 337–359, 2013.
[42] M. Stopfer, V. Jayaraman, and G. Laurent, “Intensity versus Identity Coding in an

Olfactory System," *Neuron*, vol. 39, no. 6, pp. 991–1004, Sep. 2003.
[43] J. A. Gallego, M. G. Perich, S. N. Naufel, C. Ethier, S. A. Solla, and L. E. Miller, "Cortical population activity within a preserved neural manifold underlies multiple motor behaviors," *Nat. Commun.*, vol. 9, no. 1, p. 4233, Oct. 2018.
[44] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain–computer interface," *Nature*, vol. 442, no. 7099, p. 195, Jul. 2006.
[45] M. T. Kaufman, J. S. Seely, D. Sussillo, S. I. Ryu, K. V. Shenoy, and M. M. Churchland, "The Largest Response Component in the Motor Cortex Reflects Movement Timing but Not Movement Type," *eNeuro*, vol. 3, no. 4, p. ENEURO.0085-16.2016, Jul. 2016.
[46] "MATLAB and Simulink for Student Use - Math software for engineering and science students." [Online]. Available: https://www.mathworks.com/academia/student_version.html. [Accessed: 13-Jun-2019].
[47] "Institutional Animal Care and Use Committee (IACUC) - Office for Research - Northwestern University." [Online]. Available: https://iacuc.northwestern.edu/. [Accessed: 13-Jun-2019].