A Normative, Spiking Network Model for Neuro-motor Control and Movement Disorder

BY

STEVEN PENNY
B.S., University of Notre Dame, Notre Dame, Indiana, 2016
M.S., University of Illinois at Chicago, Chicago, Illinois, 2019

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the University of Illinois at Chicago, 2020

Chicago, Illinois

Committee Members:

Max Berniker, Advisor
Pranav Bhounsule, Advisor and Committee Chair
James Patton, Bioengineering
Miloš Žefran, Electrical and Computer Engineering
Myunghee Kim, Mechanical Engineering
Brian Ziebart, Computer Science

ACKNOWLEDGMENTS

First, I want to thank my incredible wife, Ina, for supporting me in every way possible. She sacrificed a great deal of time and effort, so I could achieve my academic dreams. Without her, this would certainly not have been possible. Thank you Ed and Deb for raising such an amazing woman.

I want to thank my parents, Steve and Leslee, for always supporting my passions. But more importantly, for teaching me to love to learn, that all knowledge is good knowledge, and that failure leads to growth if you seek it.

I want to thank my labmates, Yekta, Nat, and Ali, for tolerating my dad jokes and always keeping our lab a fun, yet scholarly environment.

I want to thank my advisor, Dr. Max Berniker, for being the smartest and most critical thinker I know. He has been a great mentor and friend over my years pursuing this degree. His work and knowledge directly inspired my dissertation, and I could not have done this without him.

Finally, I would like to thank God for guiding me on this path. I am so blessed for this opportunity, the people I have met along the way, and the future it has created for me and my family.

Thank you all.

CONTRIBUTION OF AUTHORS

Introduction: Steven Penny wrote the section and Max Berniker advised and edited, with some paragraphs adapted or reused from previous work of the author:

#1. "Planned Straight or Biased to Be So? The Influence of Visual Feedback on Reaching Movements" Natarajan, V., Penny, S., Berniker, M., Journal of Motor Behavior (2019)

#2. "A normative approach to neuromotor control," Berniker, M., Penny, S. *Biological Cybernetics* (2018).

Methods: Steven Penny wrote the section and Max Berniker advised and edited, with 2.1-2.4 being adapted from author work #1.

Results: Steven Penny wrote the section and the code used to generate results and figures. Max Berniker advised and edited, with 3.1 being adapted from author work #1. Training and simulation algorithm for 3.1 was written by Max Berniker.

Discussion: Steven Penny wrote the section and Max Berniker advised and edited.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

What are the objectives, control architectures and encoded information that govern our everyday volitional movements? For decades, behavioral experimentalists have carefully analyzed human reaching movements under a multitude of conditions and stimuli to understand the hidden control objectives of the brain. Similarly, electrophysiologists have recorded neural firing patterns during reaching movements to understand what individual neurons encode, how they convey information as a population and how they influence the downstream muscle activity. Oftentimes, these results are studied in parallel rather than in conjunction. With our normative approach, we allow hypotheses to prescribe the rules of constructing mathematical models of neuromotor control. These initial hypotheses of the brain's control architecture have led to scientifically grounded predictions of motor behavior and neural firing patterns. To implement this method, the limb was modeled as a dynamical system that receives its control input from a collection of probabilistically spiking artificial neural networks. These networks are trained to implement hypothesized control schemes in order to move the limb to the target state. The output motor behavior and spiking patterns can then be compared with the observations found in prior studies. Using this method, we have trained models of neural "controllers" that exhibit commonly found phenomena in motor and neuron behavior such as: asymmetric and bell-shaped velocity profiles, directionally tuned firing rates, population vectors, and low dimensional dynamics. We extend this approach to other features of control (state estimators and forward models) to predict neural firing patterns for these features. Additionally, these models are used to test how neuron failure can lead to frequently seen motor behavior pathologies. Here, we show that a normative approach to the study of motor behavior may help accelerate research in this field and bridge the gap between behaviorists and electrophysiologists.

INTRODUCTION

1.1 Motivation

Motor control is fundamentally about the interaction of our neurons and our body. That is, how do the coupled dynamics of our neurons and musculo-skeletal apparatus give rise to intelligent motor behavior? Unfortunately, a detailed analysis of these interactions has proven unwieldy. While relatively simple models of our limbs and muscles are adequate for describing reaching movements, the collective behavior of neurons is far more complex. The number of neurons involved in motor control and the complexity associated with describing their nonlinear stochastic interactions, renders their analysis particularly difficult. In part because of this difficulty, a holistic neuromotor approach has been avoided in favor of research that examines motor behavior and neural activity largely independently. This parallelized approach has been productive for decades, but lacking a precise description for how neural activity and motor behaviors are coupled, the results are often necessarily correlational.

1.2 Reaching Experiments as a Tool for Studying Motor Objectives

Reaching studies are broadly used as a tool to examine the motor system and the control of movement. In general, these studies have found that the hand follows a straight path with bell-shaped velocity profile during both unperturbed reaches (Flash & Hogan, 1985; Morasso, 1981) and those made with perturbed visual feedback (Danziger & Mussa-Ivaldi, 2012; Flanagan & Rao, 1995; Krakauer et al., 2000; Wolpert et al., 1995). When given adequate practice, subjects also make straight reaches when interacting with perturbing loads, such as a force field (Berniker et al., 2014; Berniker et al., 2014; Shadmehr & Mussa-Ivaldi, 1994; Uno et al., 1989), a Coriolis force (Dizio & Lackner, 1995), or an external mass (Krakauer et al., 1999; Sainburg et al., 1999). The

fact that reaches are straight under these very different circumstances seems to imply that the objective of the motor system is to move the hand to a target in a minimum jerk, straight line.

Despite this evidence, there are several reasons to believe that the objective of the motor system is in fact to move the hand efficiently. Studies of human locomotion find that legs are controlled for efficient movement (Selinger et al., 2015). In fact, the movements of many animals are consistent with mechanical and metabolic efficiency (Hatze & Buys, 1977; Rayner, 1993). The wealth of data suggests movements, including reaches, should be efficient. Reaching models that minimize metabolic costs predict specific reaches that travel along curved paths (Kistemaker et al., 2014; Rosenbaum et al., 1995). Minimizing other mechanical precursors of metabolic costs, such as joint torque rates, energy, and work, also predict curved reaches (Alexander, 1997; Farshchiansadegh et al., 2016; Nakano et al., 1999; Uno et al., 1989). Indeed, everyday movements outside the laboratory setting are often curved (Ramanathan et al., 2000). These results are counter to a decades long effort to provide evidence that the hand is controlled to move straight. Studies on both sides of this debate have utilized cleverly designed tasks to reveal novel motor behaviors, but largely lack a causal link between the neural activity and the witnessed behaviors.

Additionally, studies have found that altering the feedback provided during reaching tasks can influence motor objectives. For example, subjects will move in the opposite direction of a visual perturbation of the hand's position, such that the displayed cursor follows a straight path (Wolpert et al., 1995). This is also true when the visual feedback is nonlinearly perturbed (Flanagan & Rao, 1995). Even when a cursor's trajectory is defined through a complex association of finger orientation and posture, subjects alter their movement to force the cursor along a straight trajectory (Danziger & Mussa-Ivaldi, 2012). Feedback has also been shown to have a large effect on movements during object manipulation experiments, where the type or amount of information

provided can change how subjects interact with their environment (Ahmed et al., 2008; Farshchiansadegh et al., 2016; Kluzik et al., 2008; Vaidyanathan et al., 2019; Zahed and Berniker, 2018). It is evident, then, that the brain uses feedback information concerning the state of the hand or limb to govern motor behavior, but again there is a general lack of a causal link to explain these observed phenomena.

1.3 Neural Recordings Reveal the Encoded Information for Reaches

Complementary to behavioral studies, electrophysiological results have led to discoveries in how neurons encode and relay information during motor control. One famous study found that individual neurons had an increased spike activity, or preference, to reaching movements in specific directions (Georgopoulos et al., 1982). Others have found neuron preferences for variables such as target location, initial movement direction and amount of reach curvature (Hocherman and Wise, 1991). Finally, during force field adaptation, where subjects must alter their limb dynamics in response to forces in order to maintain a straight movement, recordings have shown that only some neurons change their preferred directions into the direction of the field (Padoa-Schioppa et al., 2004). Therefore, certain neurons may encode limb dynamics objectives while others represent kinematic, or trajectory planning, objectives. Therefore, at the individual neuron level, neurons seem to encode specific values that are pertinent to the reaching movement. More broadly, the neural population as a whole also seems to encode important information related to the reach. It was discovered that the weighted sum of the directional preferences of neurons in the motor cortex pointed in the intended movement direction (Georgopoulos et al., 1986). Furthermore, it was found that low dimensional projections of the neural populations' firing rate in the primary motor and dorsal premotor cortex may contain hidden dynamic properties (Churchland et al., 2012). These studies seem to indicate that the encodings of variables related to control are split amongst many

cells and that they must work together to generate an accurate movement.

1.4 Interpretations of Control Functions in the Brain

How do these patterns of neural activity lead to robust control of the upper limb? One commonly supported hypothesis is that the population of neurons implements optimal control (Harris, 1998; Pandy and Zajac, 1991; Uno et al., 1989; Hoff et al., 1984; Hogan, 1984; Dingwell et al, 2004). Optimal controllers require an optimal estimate of the system's state variables and feedback gains which convert errors in the state to motor commands (Stengel, 1994). An optimal state estimate uses a copy of the motor command to predict its next state, and often integrates information from sensors to make a more informed prediction. The feedback gains, on the other hand, are tuned to optimize a motor objective. These properties of optimal controllers have distinct parallels with known functionalities and observations of the brain (Wolpert et al., 1995; Todorov and Jordan, 2002; Pearson, 1995).

While it is difficult to locate where these control processes might cluster in the brain, some useful information has been gathered from neural recordings. Optimal control involves estimating the system state and computing a motor command. It is expected, then, that regions of the brain controlling the limb should have correlations with the movement of the hand. This association has manifested in several brain regions such as: primary motor cortex (Georgeopolus et al., 1982), parietal area 5 (Kalaska et al., 1983), the primary somatosensory cortex (Cohen et al., 1994), the dorsal premotor cortex, or PMd (Fu et al., 1995; Caminiti et al., 1991), the cerebellum (Fortier et al., 1993), and muscle afferents (Jones et al., 2001; Ribot-Ciscar 2003). If the brain does implement optimal control, then the operations involved are likely to be distributed over these brain regions. Recordings have also shown that the brain does not act as a simple feedback controller. Sensory

feedback information can have synaptic delays up to half the duration of a reach (Marsden et al., 1978) and motor commands exhibit rapid changes well before the feedback is processed (Gordon and Ghez, 1987). Hence, it has been widely adopted that a forward model of the limb must be involved in estimating the state variables. These findings were used to guide the hypotheses formed for this manuscript.

Lastly, regional functionality in the brain can be interpreted from analyzing the movement behavior of patients with localized damage. Damage to the cerebellum leads to distinct motor pathologies. Cerebellar tremor has been categorized as an intention tremor, meaning that it worsens as a target is approached (Bhidayasiri et al., 2005). Often it presents without a tremor at rest or when holding a posture (differentiating it from parkinsonian tremor). These movements are often shown to have increased trajectory variance (Becker et al., 1991), larger overshoots, and curvature (Deuschl et al., 2000). Additionally, it has been found that these characteristics worsen when vision is removed (Day et al., 1998), indicating that the cerebellum may play a role in sensory integration to estimate the system state (Rondi-Reig et al., 2014). Motor disorders also present in victims of stroke. Depending on the stroke location, many different characteristics can develop: weakness on the affected side (hemiparesis), parkinson-like tremor and sudden involuntary jerking (Handley et al., 2009; Siniscalchi et al., 2012). These movement pathologies lead to increases in initial angular error (Beer et al., 2000; Kisiel-Sajewicz et al., 2011). These distinct movement characteristics were used to compare with "damaged" versions of the hypothesized models (see sections 2.9 and 3.4).

1.5 Modern Models of Neuro-motor Control and Large Probabilistic Models

A computational description of the causal relationship between neurons and the motor apparatus could verify and validate our current best theories. Recent work in this direction has

described how neural activity may give rise to motor behaviors. Rate-based models, which substitute a time averaged value, or rate of firing, for actual spikes, have been used to describe a number of empirical and theoretical findings (Laje and Buonomano, 2013; Sussillo and Abbot, 2009; Mante et al., 2013; Sussillo and Barak, 2013). A model that describes how neurons interact when exchanging continuously varying information, however, may not correctly describe the interactions of neurons that share binary information. This is true both for a deterministic setting, and for more realistic stochastic settings. Models of deterministically spiking neurons, on the other hand, can precisely describe the nonlinear interactions of a large number of neurons (Bourdoukan et al., 2012; Deneve et al., 2017; Thalmeier et al., 2016). Yet, the validity and use of this approach under noisy settings is also uncertain. Ultimately, to examine the relationship between neurons and movement we will require models that couple the two through stochastic spiking events (e.g. De Wolf et al., 2016). Here, we use a normative approach to prescribe hypotheses concerning the control architecture in the brain.

Due to the stochastic nature of our spiking networks, we must calculate the exact probability of the output neurons to train the model. As more hidden neurons or hidden layers are added the permutations of hidden unit activity grows exponentially, which makes this computation intractable (Neal, 1990). Our initial models use a continuous valued input, which kept the size of our networks small. To push the model to be more biologically equivalent, we decided to remove all continuous values and encode them into sets of binary inputs. This binary encoding of relevant variables (state and command) required more nodes in our networks to represent the same information. Therefore, an approximation to the exact probability was required. We adopted the mean field learning algorithm to approximate the output probabilities with a tight lower bound (Saul et al., 1996; Saul and Jordan, 1998). This method, borrowed from statistical mechanics, uses

the mean field equations to approximate the statistics of each hidden unit using their Markov Blanket (Pearl, 1988). These estimates can then be used as approximate means of the hidden unit probabilities instead of calculating them directly. While this method adds additional learned parameters (twice the number of nodes), it significantly reduces the time to calculate the visible unit probabilities (from $O(2^h)$ to $O(h)$). Therefore, for networks of larger size (with more than 20 hidden units), this method makes calculations much more feasible.

Frequently, computational explanations of electrophysiological data rely on artificial neural network models that communicate via firing rate or deterministic spikes. Yet neither can accurately describe the interactions of biological neurons that issue spikes, probabilistically. Here, we take a normative approach. Instead of constructing models to describe what is found experimentally, we first propose testable hypotheses which guide the construction of the model. In analyzing the neural firing patterns of these artificial networks, we compared model predictions to results that are typically observed in the primate brain during volitional movement: relatively straight hand paths, positively-skewed and bell shaped velocity profiles, neural patterns of cosine tuning curves, population vectors that correlate with reaching directions, low-dimensional dynamics, and changes in neuron's tuning curves after force field adaptation. Additionally, these models can be damaged to simulate trauma or disease, which forms a causal mechanism for motor disorder starting with neural activity and ending with physical pathologies.

1.6 Research Objectives

Over the past few decades, reaching studies have uncovered the underlying neural activity that is fundamental to volitional movements. This thesis serves as a bridge between two mostly

separate approaches: research concerning the physical characteristics of movement and the neural activity during a movement. More specifically, the goal of this thesis was to model the brain's precise control of a limb and test if it can provide a causal explanation from the neural activity to the physical characteristics that they generate. To accomplish this, two stages of hypotheses were formed about the neural control architecture that formed the brain model:

Hypothesis 1: The brain uses LQR control to govern reaching movements.

Hypothesis 2: The brain uses LQR control, forward model prediction, sensory integration to govern reaching movements and an efficient binary neural code to pass information.

The author has published two peer-reviewed articles that contributed either indirectly helped form Hypothesis 1 or directly provided and analyzed the results from that hypothesis:

#1. "Planned Straight or Biased to Be So? The Influence of Visual Feedback on Reaching Movements" Natarajan, V., Penny, S., Berniker, M., Journal of Motor Behavior (2019)

#2. "A normative approach to neuromotor control," Berniker, M., Penny, S. *Biological Cybernetics* (2018).

The author plans to publish the findings from Hypothesis 2 in a peer-reviewed journal as well.

The approach to evaluating these two hypotheses were largely similar. In the next chapter (**Chapter 2: Methods)**, the approach for Hypothesis 1 utilized the methodology from a subsection of the chapter (sections **2.1-2.4**, and **2.8**). This approach was built upon for Hypothesis 2, which used techniques from all sections outlined in this chapter. The following chapter (**Chapter 3: Results)** outlines the findings from these two hypothesis. Section **3.1** reports the findings from Hypothesis 1 and features text and images from the previously published article, Berniker and

Penny, 2018, see above. Sections **3.2-3.4** are all unpublished work that reports findings from Hypothesis 2. The last chapter (**Chapter 4: Discussion**), will summarize the findings of both hypotheses and provide an in-depth discussion of the results.

METHODS

Many computational explanations of electrophysiological data often rely on artificial neural network models that communicate via firing rate or deterministic spikes. Yet neither can accurately describe the interactions of biological neurons that issue spikes, probabilistically. Here, we take a normative approach by designing a control architecture of probabilistic spiking networks to implement LQR control, forward model and Kalman filter for a model limb. By analyzing the neural firing patterns of these artificial networks we look for typical results that are observed in the primate brain during volitional movement: relatively straight hand paths, positively-skewed and bell shaped velocity profiles, neural patterns of cosine tuning curves, population vectors that correlate with reaching directions, low-dimensional dynamics, and changes in neuron's tuning curves after force field adaptation.

If there is consistency between our artificial networks and experimental results, then that would support the hypothesis that the brain (likely in the primary motor cortex) uses LQR control during volitional movements.

2.1 General Experimental Protocol

Rather than attempting to precisely model neurons and the motor apparatus, with a normative approach our focus is on organizing hypotheses that can explain how these two systems interact and what testable predictions can be made. Therefore, we model the salient features that describe how the motor apparatus is coupled to neurons. Here, we suggest the motor apparatus is approximately a point mass, driven by muscles that are activated through a network of neurons, which spike stochastically. Changes in the limb's state are then fed back to these neurons in the form of an error signal so that the network can update its commands to the muscles (Figure 1). Our

hypothesis is that the network of neurons is implementing an LQR controller, a commonly supported proposition for both upper and lower limbs (Kuo, 1995; Scott, 2004; Todorov, 2004).

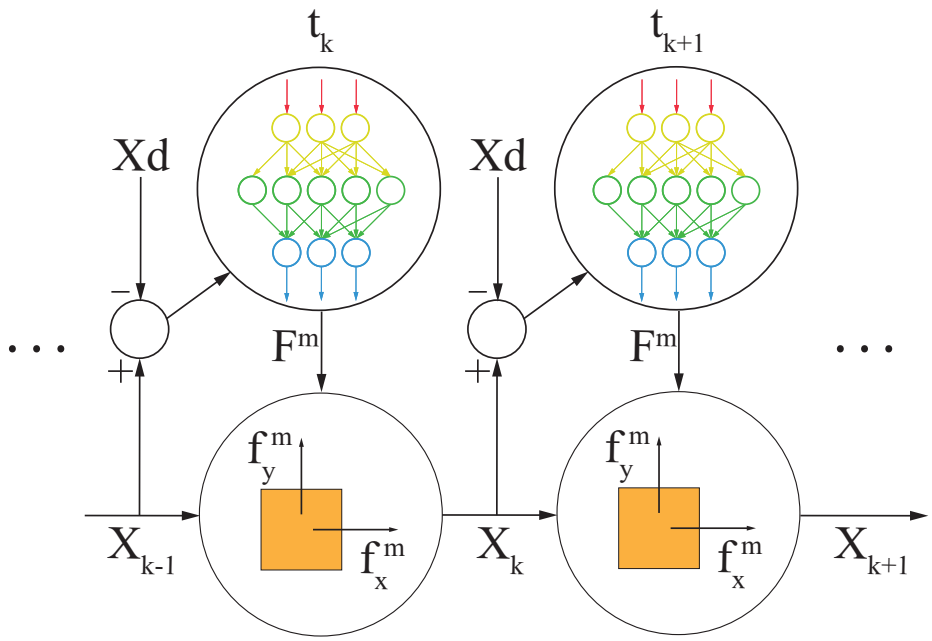

Figure 1. The generative model for the neuromotor system. A feed-forward network of probabilistically firing neurons is used to generate muscle forces. These forces accelerate the limb model, changing its current state. The updated state error is fed back to the network, and a new muscle force is sampled.

2.2 Musculo-skeletal Model

To examine point-to-point reaches, the limb is modeled as a point mass that moves in a plane. For the x-axis we have,

$$m\dot{v}_x = -b\dot{x} + f_x^m \qquad (1)$$

The y-component is modeled identically, where m = 1kg, and b = 0.1Ns/m. Analogous to real muscles, muscle force, f_x^m, is modeled as a linear sum of activated motor units,

$$f_x^m = \sum_i^n \alpha_i \, o_i^x \qquad (2a)$$

where α_i is the force of the i^{th} motor unit and o_i is the corresponding drive to that unit, modeled as a spike (zero or one) from the network. In order to "pull" in both the positive and negative directions, half the α were positive, and half negative. The muscle force values for each direction were defined such that each motor unit's value was twice that of the previous unit. Additionally, the sum of the forces equaled the maximum force provided. The force value for the smallest motor unit (X) could be calculated using the following equation:

$$X(\sum_{i=1}^{n/2} 2^{i-1}) = F_{max} \qquad (2b)$$

where n is the total number of motor units and F_{max} is the maximum desired Force from the population.

Our hypothesis is that control of the limb is approximating a Linear Quadratic Regulator, driving the limb to a desired location in the workspace.

$$L_{lqr} = \int_0^\infty \frac{1}{2}(\Delta X^T Q \Delta X + U^T R U) dt \qquad (3)$$

where $X=[x,y,v_x,v_y]^T$ and $U=F^m= [f_x^m, f_y^m]^T$. The resulting optimal control signal can be found from the algebraic Riccati equation, $U_{lqr} = -K\Delta X$. It is important to note, that the optimal feedback control law is smooth, continuous, and deterministic, whereas our network can only output discrete force values (according the sum in equation 2a), and does so stochastically.

2.3 Spiking Network Model

In rough approximation neurons appear to transmit information through discrete spikes, in a probabilistic fashion, based on the spikes they receive from other neurons. These discrete events render the relationship between incoming and outgoing spikes nonlinear, and particularly challenging for a large number of neurons. However, with two important simplifications we can obtain a model with an analytic solution for each neuron's probability distribution.

First, we assume each neuron's probability of emitting a spike is only a function of the current incoming spikes, and not the history of incoming or outgoing spikes; that is, our neurons have no dynamics. Second, we assume the network only has feed-forward connections. While this limits the potential distributions the network can represent, this significantly simplifies the required calculations (Jordan et al., 1999). With these two assumptions our network of spiking neurons can be modeled as a sigmoidal belief network (Neal, 1990) with a single hidden layer of neurons or $H = [h_1, h_2, ..., h_m]$ and an output layer, $O = [o_1 , o_2 , ..., o_n]$. Parameterized by the hidden and output layer weights, $\{W_1, B_1\}$ and $\{W_2, B_2\}$ respectively, the probability of any neuron firing is given by,

$$p(h_i = 1|\Delta X) = \sigma(\sum_j w_{ij}^1 \Delta x_i + b_i^1) \quad (4)$$

$$p(o_i = 1|H) = \sigma(\sum_j w_{ij}^2 h_j + b_i^2) \quad (5)$$

where $\sigma(\cdot)$ is the standard sigmoidal activation function. We can compute the exact probability that any output neuron in the network spikes by marginalizing over all the $2m = M$ permutations of the hidden neurons.

$$p(o_i = 1|\Delta X) = \sum_v^M p(o_i = 1|H^v)p(H^v|\Delta X) \quad (6)$$

where

$$p(H|\Delta X) = \prod_i^m p(h_i|\Delta X) \quad (7)$$

For all of the forthcoming trained networks, we designed the initial weights to achieve an approximate solution (see Results), and then fine-tuned the network by optimizing a cost function that penalizes the network's expected deviations from the desired probability (see Appendix).

2.4 Simulated Experiments and Data Analysis

We can sample muscle forces from the network by providing an input, ΔX, then drawing spikes from the hidden layer according to its probabilities (equation 4), and then drawing spikes from the output layer according to its probabilities (equation 5) and computing F (equation 2a). With a sample force, the limb's dynamics could be discretely updated to the next time step (Δt = 5ms), and the process repeated. Point-to-point reaches were simulated by providing zero input for 250 milliseconds, and then switching X_d to the desired state value and integrating the equations forward for another second. This resulted in time-varying trajectories for the limb, along with a spike train history for each neuron.

With our simulated data we performed conventional statistical analyses, including computing peristimulus time histograms (PSTH), cosine tuning curves, and population vectors. PSTHs were created using a 25 millisecond time bin, aligning spike data at time t=0, and averaging across trials. Cosine tuning curves were found using a least squares fit of the parameters A, B and C from the equation $y = A + B \sin(\theta) + C \cos(\theta)$, where y is the firing rate of a neuron for a specific reach angle, θ, and the resulting preferred direction $\theta p = \tan^{-1}(B/C)$. Using these tuning parameters, we

computed the population vectors as the vector sum of each neuron's preferred direction that is scaled by its firing rate. In addition, we performed a dynamical systems analysis as described in (Churchland et al., 2012), which searches the neural firing rate data for a low-dimensional subspace that exhibits oscillatory behavior (representing the dynamics of the limb).

2.5 Encoding Continuous Information into Binary Values

The first half of this study analyzed a network approximating LQR control that relied on a noiseless, continuous error signal. While this provided interesting initial results, this does not mimic what is found biologically. To create more biomimetic artificial networks, we encoded all inputs and outputs throughout the control architecture into binary values. This binary representation of state and command will be referred to as state bits and command (or muscle) bits, respectively. With this encoding scheme, only discretized signals could be communicated between the networks and the plant. Since neurons communicate via spikes, we considered this a necessary change to our model.

The human sensorimotor system has been crafted by many years of optimization through evolution; therefore, it is reasonable to hypothesize that the brain encodes and conveys information in an efficient manner. This suggestion has led us to construct a binary encoding scheme that, when given a number of bits, can encode the maximum discrete values. This encoding scheme is non-linear. Each bit requires the analog signal value and each of the previous bits' values (see Figure 2A). Specifically, the inequality for each bit can be represented as:

$$B_n\colon\ X - R_{min} \geq \frac{R}{2^n} + \sum_{m=1}^{m=n-1} \frac{B_m}{2} \qquad (8)$$

where R is the range of continuous values to be encoded and R_{min} is the minimum of that set. If this inequality is true, then B_n takes a value of 1, otherwise 0. In Figure 2B, the provided example details how an analog value was encoded into bits and what discrete value those bits represent. In the example shown, 3 bits can encode 2^3 = 8 discrete values. Other encoding schemes have taken an approach where each bit corresponds to a location in space, such that only one bit can take a positive value at a time (Makin et al., 2013; Makin et al., 2015). This simpler encoding is intuitive and can be represented as a system of linear inequalities. Therefore, it can be encoded and decoded with linear operations. Yet, it quickly becomes impractical to use when higher precision is desired. For each variable, it can only represent *n* discrete values, where *n* is the number of bits. Our encoding method, where each additional bit doubles the precision, can represent 2^n discrete values. Additionally, the digital-to-analog decoding of these bits is affine. Therefore, to be consistent with the efficient inclinations of the brain, the more efficient encoding scheme was adopted.

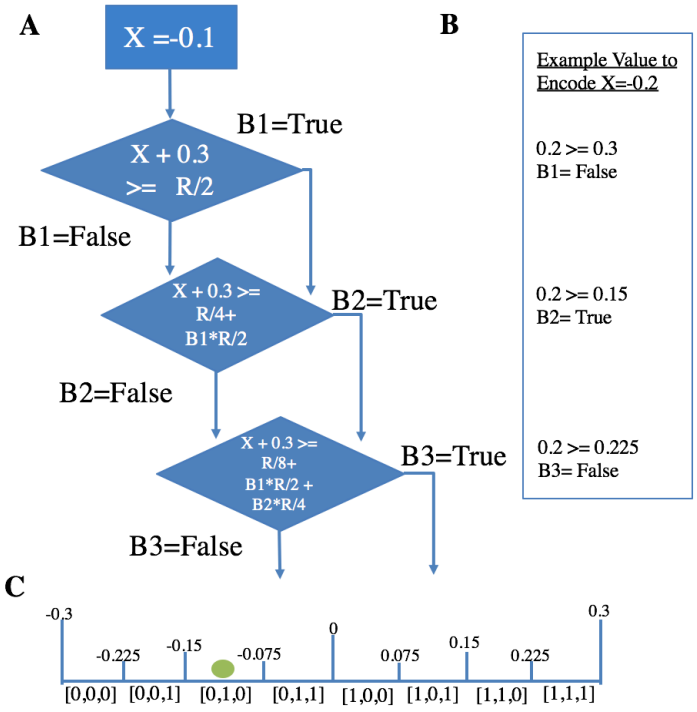

Figure 2 A) Flow diagram of our nonlinear process for encoding continuous values into binary, where R is the range of values to be encoded (0.6 in the example). If the inequality value was true or false then that encoding bit took a value of 1 or 0, respectively. **B)** An example value of -0.2 is used to show the encoded value for 3 bits. **C)** The discrete bins that the encoded values represent. For 3 bits, this encoding scheme can represent 2^3, or 8, discrete values.

For all networks that incorporated a binary encoding of state and command, the state and command were encoded into 15 and 10 binary bits, respectively. These variables could take a range of discrete values between [-0.3 0.3] m, [-1.5, 1.5] m/s, and [-10 10] N, for the position, velocity and command, respectively. These ranges were selected by simulating 100 random reaches over the 0.6x0.6 m^2 workspace and looking at the most extreme velocity and command values.

The number of bits in the encodings were chosen specifically. The state encoding should be precise enough such that one discrete time step of the smallest possible command can push the limb into a different encoded state. This prevents the state from becoming "stuck" at one encoded value, despite a given command. For example, with 10 muscle bits and a maximum force of 10N, the smallest force (F_{Xm} and F_{Ym}) generated in either direction could be calculated using Equation

2b:

$F_{Xm}(\sum_{i=1}^{5} 2^{i-1}) = 10N$ ➔ $31F_{Xm}$=10N ➔ F_{Xm}=0.323 N

A single discrete time step of this force can generate a change in position using the dynamical equations of the point mass system with damping:

$X^{+}=A_dX+B_dU$, where $X=\begin{bmatrix} X \\ Y \\ V_x \\ V_y \end{bmatrix}$, $U=\begin{bmatrix} F_x \\ F_y \end{bmatrix}$, $A_d=\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1-\frac{b}{m}*\Delta t & 0 \\ 0 & 0 & 0 & 1-\frac{b}{m}*\Delta t \end{bmatrix}$, $Bd=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\Delta t}{m} & 0 \\ 0 & \frac{\Delta t}{m} \end{bmatrix}$

As is evident from the dynamical equations above, the command does directly affect a change in position but only through generating a change in velocity. Using these equations, one can find that a single discrete command of 0.323N can generate a change in velocity of 0.0032 m/s. In turn, this velocity after one discrete time step creates a 3.2e-5 m change in position. Therefore, we can use this value to find the minimum precision of our network. Since each additional bit in our encoding scheme doubles the precision of the encoding, we can find the number of bits needed with the following equation:

$\frac{R}{2^{\#bits}} < Minimum\ change$, which for the position state encoding can be solved below:

$\frac{0.6}{2^{\#bits}} < 3.2e^{-5}$, which gives a minimum number of bits of 14.18.

Therefore, all of the state values in this text were encoded with 15 bits, which means our states can take 2^{15} = 32,768 different values.

As mention above, if less than this calculated number of bits is used, it is possible for a

command to be given and the encoding of position to not change. In these simulations, the plant operates in continuous time, but the state estimator and controller, both spiking networks, communicate with binary encodings of discrete values. Near the target, it is possible for the controller to issue a small, nonzero command that does not change the state estimate encoded value. Therefore, the plant experiences drift as it is continuously pushed away from the target. A few examples of this analysis can be seen in Figure 3. Here, one can see that for only 11 bits of encoding there is a significant drift in the plant after reaching the target (Figure 3A). Adding more bits, but less than the calculated minimum number, the drift remains but is significantly reduced (Figure 3B). Using our calculated minimum number of bits (Figure 3C), accurate control of the plant is achieved without any drift. This analysis reinforces the concept of using our calculated number of bits for the encoding.

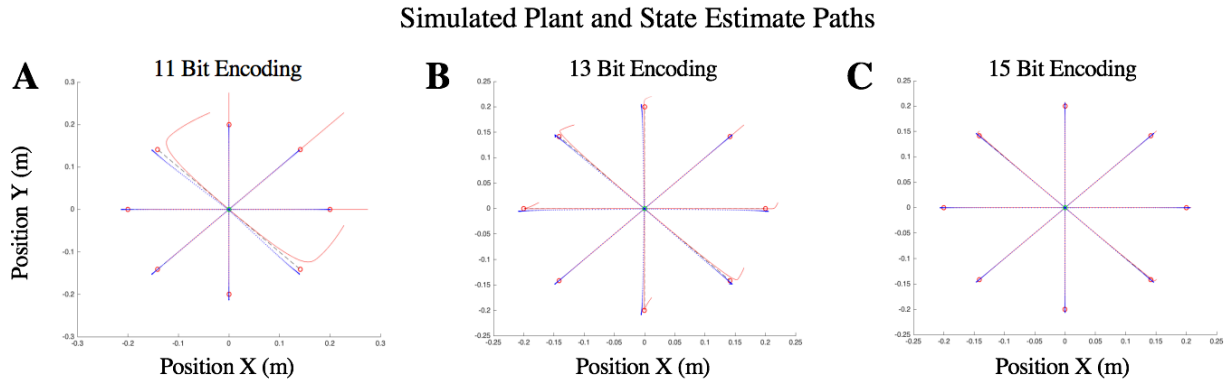

Figure 3) All plots display the effect of using a discrete approximation of the state (blue trace) to compute the error signal for the controller. For lower resolution encodings **A)** and **B)**, the plant experiences drift at the end of some reaches. A 15-bit encoding **C)** was calculated to be the smallest number of bits needed to achieve accurate performance.

2.6 Applying Mean Field Approximation for Probabilistic Networks

Using binary encoded variables in place of continuous valued variables leads to much larger networks; the state inputs alone have increased 15 times in size. Therefore, calculating the exact probability in the traditional manner becomes unwieldy, as it requires summing over 2^n

permutations of conditional probabilities, where *n* is the number of hidden nodes. To combat this issue, we implemented a variational method for calculating the distribution known as the mean field approximation (Saul et al., 1996). To cover in broad terms, the mean field approximation uses an approximate, tractable distribution (Q(H|V)) to approximate the actual, intractable distribution (P(H|V)). Q is a function of latent variables, which are loosely related to the mean and standard deviation of the probability of a node given a specific input. The learning rule for updating the weights were calculated by minimizing the KL divergence between Q(H|V) and P(H|V), while simultaneously updating the latent variables to make Q a better approximation to P (see Appendix for equations). An in depth discussion of using the mean field approximation in the context of probabilistic networks was beyond the scope of this manuscript, but is expertly addressed in previous literature (Saul et al., 1996; Jordan et al., 1998; Saul and Jordan, 1999).

The mean field method was used in conjunction with Adam, a gradient based optimizer with momentum, (Kingma and Ba, 2015) to iteratively updated the network weights. The standard Adam parameters were used with a base learning rate (alpha) of 0.05 and decayed such that alpha= 0.05/(1+epoch). One novel method that we implemented to quicken learning was to also decay the mean output label. The mean field approximation assumes the visible node mean output is known. One advantage to this method is that it allows for flexibility in choosing a non-binary mean output value (similar to label smoothing in traditional neural network architectures, see Muller et al, 2019). We used a decaying mu value that started at .25 and .75 and decayed to 0 and 1, respectively, with increasing number of epochs. All of these processes helped increase the stability of gradients and to quickly find more optimal minima.

The mean field method worked the best for our problem, but another widely used algorithm, one-step contrastive divergence (CD1) algorithm (Hinton, 2002), was also attempted.

While greatly reducing the computational time, CD1 simplifies the learning rule by collecting one sample of the networks activity and using it to calculate the gradient. This works very well for non-spiking networks as the hidden nodes can communicate average values between 0 and 1. In our spiking model, however, every node only communicates in binary. For our problem, the gradient from CD1 was too noisy to use for learning and was not worth the decrease in computation time. Another attempted method was to sample just to find the mean field latent variables. On average, it took ~ 30 samples to achieve gradients that were within the same noise level as the mean field equations. Taking this many samples, however, did not reduce the computation time and was not used because of this. Finally, a mean field method variant was attempted (Jordan et al., 1998), where there is an assumption of normally distributed activities in the hidden nodes. For networks with nodes that have many inputs, it can be a very reasonable approximation and dramatically reduce computation time. However, we found that the gradient from this method was also too noisy to learn our problem. Based on these findings, the standard mean field method was considered the best choice to train our spiking networks.

2.7 Implementing State Estimation with Networks

With the encoding scheme proposed here, training a network to preform accurate state estimation requires more input and output nodes than to train a network to act as a controller. The controller required 4, 15-bit state inputs and 2, 10-bit command outputs, for a total of 80 input/output (I/O) bits. On the other hand, the forward model required 4 state inputs, 2 command inputs and 4 state outputs and the Kalman filter required 8 state inputs and 4 state outputs (140 and 180 I/O bits, respectively). As the network sizes grew for state estimation, so did the importance of computation time and the approximate methods mentioned above. The mean field method with Adam optimizer was able to provide decent network performance for the task, but the best results

were found when we provided initial guess for the weights.

To solve for initial weights, networks with binary encoded state values as outputs were assumed to have a depth of n+1 and n fully connected feedforward layers (input layer is exempt). This means that skip connections, weights connecting two nodes that are two or more layers removed, were allowed (although training could reduce them to a negligible value). These assumptions provided a solid framework for solving the weights by hand. To do so, we started with the discrete equations governing each control task. For the forward model, that is:

$$X^{+} = A_d X + B_d U$$

To decode a bitwise encoded version of X and U signals, the affine transformation $D2A_x$ and linear transformation $D2A_u$ were applied. These matrices are functions of the range of encoded values and number of bits:

$$D2A_x = \begin{bmatrix} R_x/2^1 & R_x/2^2 & R_x/2^3 & \dots & R_x/2^n \\ R_{vx}/2^1 & R_{vx}/2^2 & R_{vx}/2^3 & \dots & R_{vx}/2^n \\ R_y/2^1 & R_y/2^2 & R_y/2^3 & \dots & R_y/2^n \\ R_{vy}/2^1 & R_{vy}/2^2 & R_{vy}/2^3 & \dots & R_{vy}/2^n \end{bmatrix}$$, where *n* is number of state bits for encoding.

$$X_{Bias} = \begin{bmatrix} -R_x min \\ -R_{vx} min \\ -R_y min \\ -R_{vy} min \end{bmatrix}$$, where R_{min} is the minimum value to encode for that state

$$D2A_u = \begin{bmatrix} -2^0 F_X & -2^1 F_X & \dots & -2^{\frac{n-1}{2}} F_X & , & 2^0 F_X & 2^1 F_X & \dots & 2^{\frac{n-1}{2}} F_X \\ -2^0 F_Y & -2^1 F_Y & \dots & -2^{\frac{n-1}{2}} F_Y & , & 2^0 F_Y & 2^1 F_Y & \dots & 2^{\frac{n-1}{2}} F_Y \end{bmatrix}$$

where F_X and F_Y are the minimum value force as found in Equation 2b. Using these matrices, X and U can be rewritten as:

$X = D2A_xBX + X_{Bias}$; $U = D2A_uBU$

where BX and BU are the binary encoding of X and U, respectively. The original dynamical equation can be substituted in with these variables:

$$X^+ = A_dD2A_xBX + B_dD2A_uBU + A_dX_{Bias} \quad (9)$$

It is evident from this equation that a discrete output X^+ is a linear function of binary encoded inputs BX and BU. Therefore, if we did not restrict our networks to only outputting binary encoded values then a simple two-layer feedforward network would suffice to learn this problem. However, due to this restriction a deep network is required for the nonlinear binary encoding of X^+.

Each bit in the encoding (see Figure 2A) requires its own equation, which relies on the discretized value X^+ and all preceding bits, or equations, in the encoding. This idea motivated the assumption of a fully connected output layer. We can use the encoding equations to compute an initial guess for the weights and biases of the fully connected layer. Using the same example as in Figure 2A, the second equation (X^++0.3 >= R/4 + B1*R/2) can be written in the form:

$X^+*(1) + B1*(R/2) + (0.3 + R/4) >= 0$

Here, it can be seen that the weights from X^+ and the first encoding bit (B1) would be 1 and R/2. Similarly, the bias on the second output node would be 0.3 + R/4 (in addition to the bias from X+ in Equation 9). Since we used sigmoidal activation functions, these weights only provided outputs that were on the correct side of 0.5. In other words, if the desired output was 0 or 1 then an output of 0.3 or 0.7 were likely. With this initial guess for weights, frequent bit flips leading to noisy state trajectories, but these errors subsided after several epochs of training.

A similar method was applied to solve the weights of the Kalman Filter network as it too relies on D2Ax to decode the incoming signals and the output is the binary encoded state. One additional caveat, however, was that we assumed that the covariance matrices had stabilized (Kalman, 1960). Therefore, this is not a model of learning under novel task or stimulus. The focus here is on generating results similar to what would be seen in practiced reaches.

2.8 Training the Spiking Network Models

The networks were all separately trained on uniformly distributed data over the input bits. To accomplish this, we generated 1000-10,000 pseudorandom vectors of length equal to the input size of the network (i.e. length of 60 for the controller). To treat these values as samples from a Bernoulli distribution, a threshold was applied at a probability of $p=0.5$. These binary values were used as the training inputs for the network. To calculate the binary output associated with each input training vector, the input was first decoded to a discrete value. Then, this value was used as if it were the input to the equation that the network approximates. For example, the controller network approximates $U = -K\,\Delta X$. Therefore, the randomly generated binary vector was decoded to give a discrete version of ΔX and then used to calculate U. This output value was then encoded back to a binary vector to be used in training. For larger sample sizes (>500 sampled vectors), training data was also uniformly distributed over the decoded input and output values. It is important to note that the training data for the controller and forward model contained no noise besides the small loss of precision associated with encoding scheme. The Kalman filter, however, was trained with a noisy sensory observation whose covariance was equivalent to the covariance of the forward model prediction. The Kalman filter, then, was intended to learn to equally weigh the two signals, although after training it more heavily weighed the forward model prediction.

Once the three networks (controller, forward model, and Kalman filter) were trained and producing reaches resembling LQR control trajectories, we injured this "healthy" network to analyze its reaching and neural behavior under these "diseased" conditions. There were three methods that we sought to injure the model: lower the spike probability of a node, increase the spike probability of a node, silence a node (representing cell death). This was rather straightforward to do since all of the software for training and simulations was handwritten by the author. For this document, terms such as "neuron #X was killed" will be used to represent the silencing of a particular node.

2.9 Evaluating Reaching Characteristics

To evaluate the simulated reaching behaviors, several metrics were utilized. When testing the model's capability of generating a reliable state estimate, the mean squared error (MSE) was calculated. In this manuscript, the state estimate error referred to the error in both position and velocity compared to the trajectory generated by LQR control. In general, the values of velocity were larger than those of position, so errors were also larger on average. This reinforced a harsher penalty for errors in velocity, which have a larger effect on the overall trajectory.

The start and end of a movement was important to define for accurate comparisons across simulations. The movement onset was defined as the first moment that the velocity reaches $1/8^{th}$ of its maximum value during the reach. Usually, this was between 0.01-0.03 seconds after the signal to move was given. The end of the movement was defined as the first point after the start time, where the velocity falls below $1/8^{th}$ of its maximum value. It is important to note that for all reaches that do not reach this threshold an ending time of 2 seconds was assigned (the total time of the simulation). The movement time of the reach was then the end time subtracted by the time

of movement onset.

Other reaching metrics were calculated similarly to what was used in the literature. The reach overshoot was calculated as the point on the trajectory furthest from the target in the direction parallel to the straight line trajectory. This value was normalized such that reaches past the target had values > 1 and ones that did not make it to target had values < 1. The initial angular error was defined as the angle between the line connecting the position at movement onset and peak velocity and the straight line trajectory. A positive error referred to deviations in the counterclockwise direction.

The path length was calculated as the cumulative sum of distance between two points on the trajectory normalized by the straight distance between those two points. The two points used to calculate path length differed depending on the experiment that was being compared. Unless otherwise specified, the path length was calculated between position at movement onset and end of movement. The acceleration phase path length was calculated between position at movement onset and at peak velocity. In a similar way, the deceleration phase path length was calculated between the position at peak velocity and at end of movement.

All comparisons between our models used two-sample Student t-test to determine significant differences *(p<0.05)*. For some studies used as comparisons, however, only mean and standard deviations of a metric were provided. Therefore, the raw data was not available to perform standard statistical tests. In these cases, a Kolmogorov-Smirnov (KS) test was used to determine whether our simulated data could have come from the distribution reported in the study. To accomplish this, we first normalized our data (of length N) to a standard normal distribution using the approximate mean and variance reported in the study. For path length, it was normalized

similarly but to a standard half-normal distribution (values could not be below 1). Then, *N* data points were sampled from a standard normal distribution and compared to our normalized data with a two-sample KS Test. For small sample sizes, the test can vary in its outcome. Therefore, we conducted 1000 iterations and report the number of times the null-hypothesis was not able to be rejected.

RESULTS

3.1 A Spiking Network Controller with Exact, Continuous State Information

The following results reflect the examination of a network that implements the LQR control law, $U = -K\,\Delta X$. The state, X, was assumed to be known exactly and continuously valued. While this cannot accurately portray how the brain receives information, it proved to be a useful proof of concept with its own interesting results. The results in this section have been taken from published work by the author.

We designed a network to accurately encode the linear LQR control law, $U = -K\Delta X$, while limited to discrete force values (15 for the x and y-axis each, see Methods). Note that an ideal function would achieve this by mapping arbitrary inputs to the nearest discrete muscle force value. Thus, the input space would effectively be categorized into *k* linearly separable regions if there were *k* discrete muscle forces. This is precisely what the hidden layer of a sigmoidal network does; regions of the input space separated by the probability, p = 0.5 of eliciting a spike (the line obeying $\sum w_{ij}^{1}\Delta \mathrm{x_j} + b_j^1 = 0$), are effectively decision boundaries, determining whether a spike is likely or not. In other words, the hidden layer acts like a common flash analog-to-digital converter (Walden, 1999). As such, the hidden layer requires *k-1* neurons to precisely encode the *k* discrete input space regions (see Figure 4).

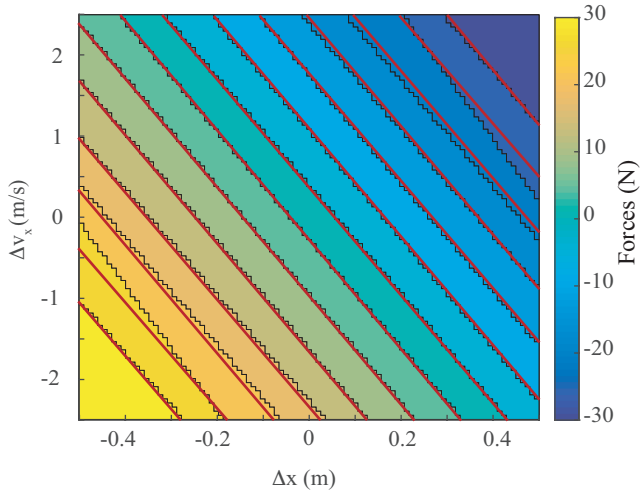

Figure 4) To design the network weights the input space (shown here as x and v_X) is broken into linearly separable regions. The ideal decision boundary for the 15 force levels of an LQR controller are shown in red. These boundaries are then used to initialize the hidden neuron's weights such that the boundaries align with the hidden neuron's probability of p = 0.5. After training, these boundaries are altered (black lines).

Note, that since our muscle units contain identical positive and negative values, for *n* output units the net- work can only generate $k = 2^{n/2+1}-1$ unique forces, not 2^n. Recognizing this, the hidden layer's weights can be chosen to create these decision boundaries by setting $W^1 = K$, and B^1 to the values of 14 linearly spaced forces between -30 and 30 N (see Figure 4). Therefore, in total the network had 40 neurons: 28 in the hidden layer, and 12 in the output layer.

Determining the output layer's weights was similarly straight forward. Each of the 15 discrete force values is generated from a set of output spikes. These force values also correspond to inputs from a unique region of the input, with an associated expected hidden layer activity. The output weights were initially chosen to map these expected hidden layer spikes to values that would increase the probability of the appropriate forces.

To explain, let H^i be the expected hidden neuron values corresponding to the i^{th} discrete muscle force value, and μ^{i*} be the desired expected output for this same force value. $\mu^{i*} = E[O^i] = \sigma(W^2H^i$

$+B^2)$ is composed of terms approximately zero and one, so the argument must correspond to large positive and negative numbers (which we arbitrarily chose as ±3). Therefore, for each of the discrete force values, and associated hidden and output values, we assume $W^2H^i + B^2 = 6\mu^{i*} + 3I_n$, and solve a linear set of equations for W^2 and B^2. With these designed initial weights, the network approximately represented the LQR controller. To fine-tune these weights, we performed gradient descent on the expected error in the output probabilities (see Methods, Appendix).

With the network complete, muscle forces, obtained by sampling over a probabilistically firing group of neurons, are noisy and may vary from the desired force (Figure 5). However, we can analytically compute the expected value and variance of a muscle force conditioned on the current input. For unperturbed movements, f_x^m is only a function of deviations in x and v_x, so we can depict the expected muscle force, the expected error in muscle forces and its variance as a surface (Figure 5B, C & D). Although the network can only generate 15 discrete force values over a range of 60 Newtons, the expected error in f_x^m averaged over the workspace was 0.45 ± 0.72 Newtons (mean and standard error of data in Figure 5C), while the analytically derived standard deviation in the expected error, averaged over the workspace was 3.14 ± 0.23 Newtons (mean and standard error of data in Figure 5D).

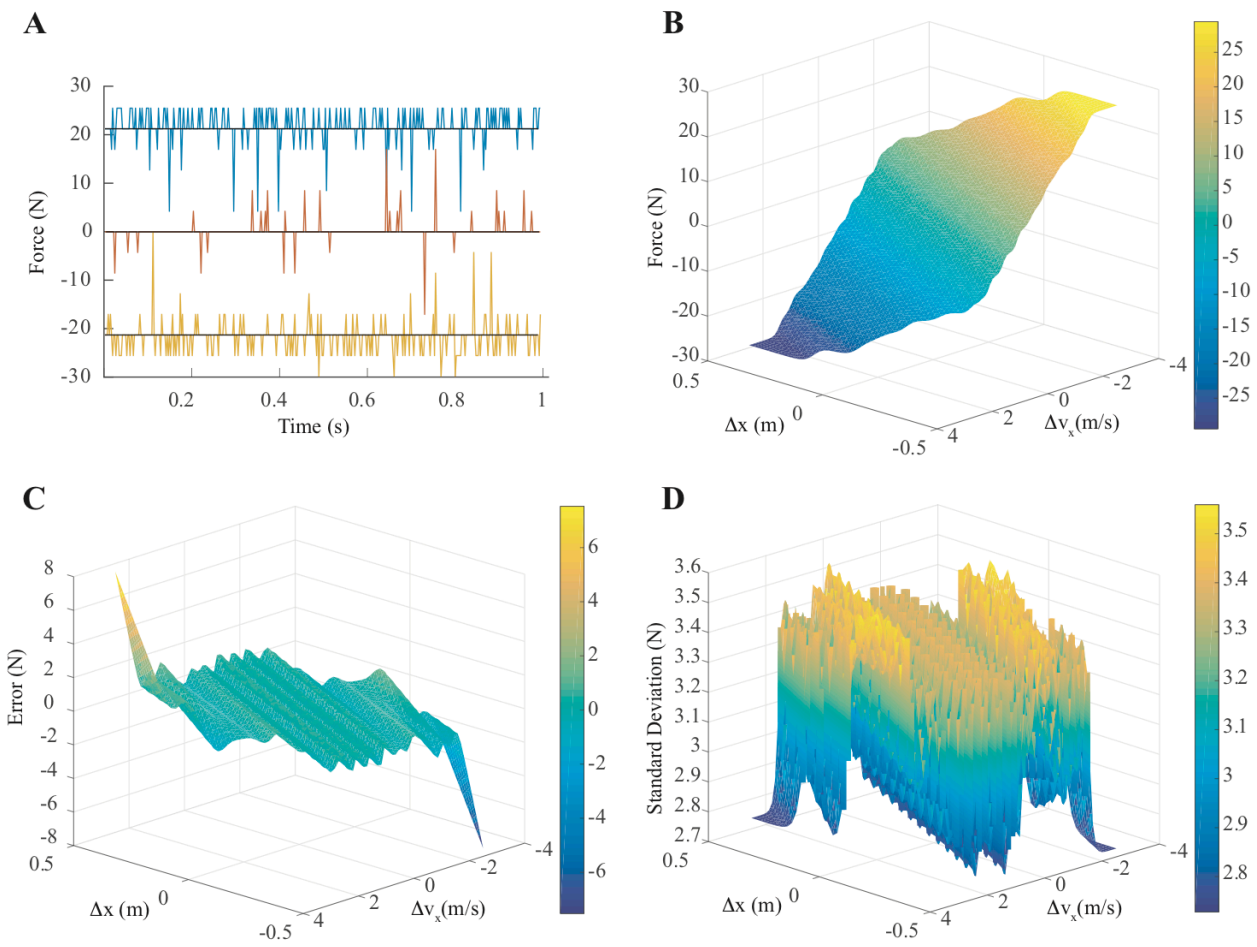

Figure 5 A) Sample forces obtained from the network. The input is held constant for 1 second and muscle forces are sampled every 5ms. The desired output forces are 21.4, 0 and -21.4 Newtons (black lines). **B)** The expected forces the network generates over the workspace in the x-component (y-component is identical). **C)**, **D)** The expected error between the network and an LQR controller and standard deviation in muscle forces can also be computed analytically across the limb's entire workspace.

3.1.1 Neuromotor Model Dynamics

To simulate movement, the limb's state is integrated forward in time discretely, at each time step getting a randomly sampled muscle force from the network using the previous time step's state information. To illustrate, 20 random reaches were simulated (Figure 6A). As was expected, reaches exhibit realistic characteristics: straight paths with approximately bell-shaped velocity

profiles (Figure 6B). As with experimental data, speed profiles are positively skewed (Mutha and Sainburg, 2007; Corcos et al., 1989; Nagasaki, 1989). Although movement times are not specified, they generally reached their target within approximately 0.8 seconds. In addition to generating movement, the neuromotor model also generates spike trains, predicting a variety of patterns (Figure 6C). For a given reach, some neurons are largely silent, others are tonically active and still others display phasic activity.

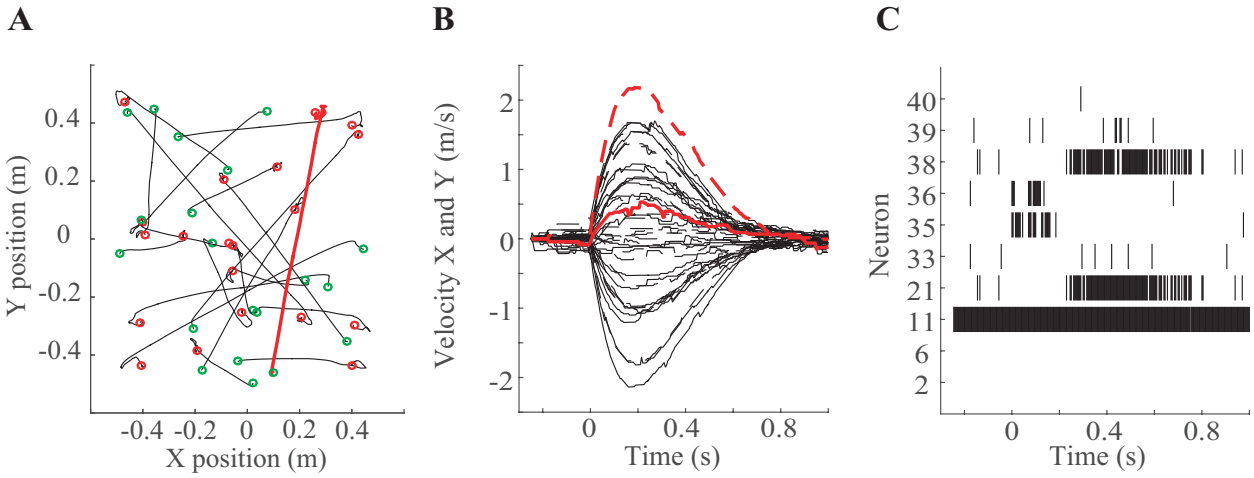

Figure 6 A) Randomly generated reaches with the neuromotor control model. Each reach starts at a green circle and ends at a red circle. In red, one such example reach is shown. **B)** The corresponding velocities of these reaches exhibit roughly bell-shaped profiles. The x- and y- components are shown in solid and dashed lines, respectively. In red are the velocity profiles for the example reach. **C)** For each simulated reach there is an associated history of neural activity. A raster plots is shown for 10 illustrative neurons for the example reach in red. Neuron indices are arbitrary labels in the network with indices 1-28 corresponding to hidden units and indices 29-40 corresponding to output, muscle units.

3.1.2 Virtual Experiment and Analysis

To model a typical electrophysiological study, e.g. (Georgopoulos et al., 1982 ;Li et al., 2001; Churchland et al.,2012), we simulated center-out point-to-point reaches to eight evenly spaced targets, 50 times each for a total of 400 trials (see Figure 7A, B). With this data we obtained

average spike rates and other commonly computed neural statistics. As is seen experimentally, during these center-out reaches, half of our neurons were largely silent, and did not have firing rates significantly greater than zero (Georgopoulos, 1982). Of those neurons that were active, most had phasic activity with clear signs for a preferred reach direction (Figure 7C-H). We can quantify this preference with a cosine tuning curve. Of the 16 neurons that were active, 11 showed significant tuning ($R^2 > 0.5$, see Figure 7C, D).

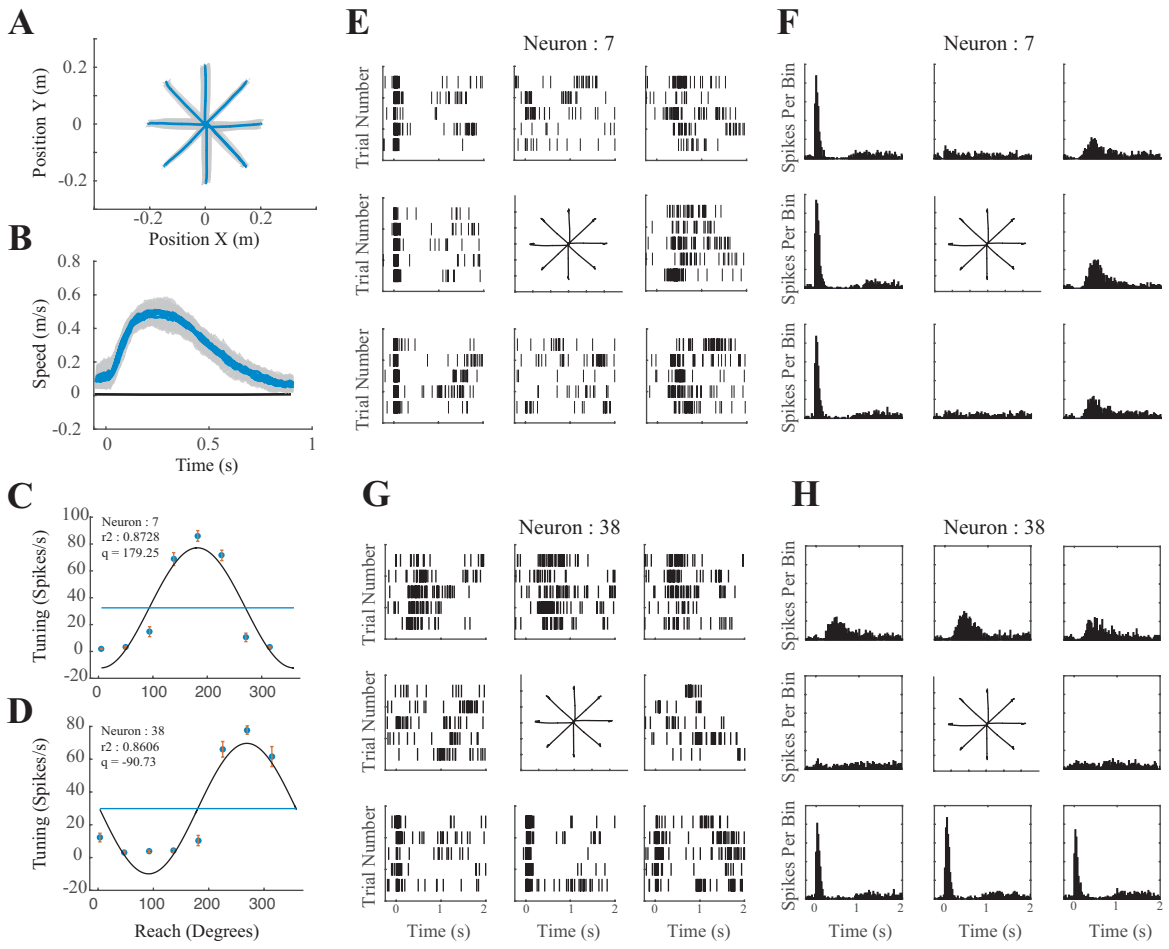

Figure 7 A), B) Average paths and speed profiles for the simulated 8 center-out reach directions. **C), D)** Two neuron's tuning curves reveal a strong correlation with reach direction. **E), G),** Raster plots for the same two neurons are displayed for each of the 8 reach directions. They reveal phasic activity that initially increases during leftwards and downwards reaches respectively**. F), H)** PSTH's for the same neurons.

We then used this cosine tuning to construct population vectors (Georgopoulos et al., 1986), and we found that these population vectors point in the direction of movement. Throughout the duration of the reach, the population vector for all eight directions consistently points towards the desired target and the direction of movement (Figure 8). The error between the mean population vector and reaching direction was $0.53^{o}\pm2.5^{o}$.

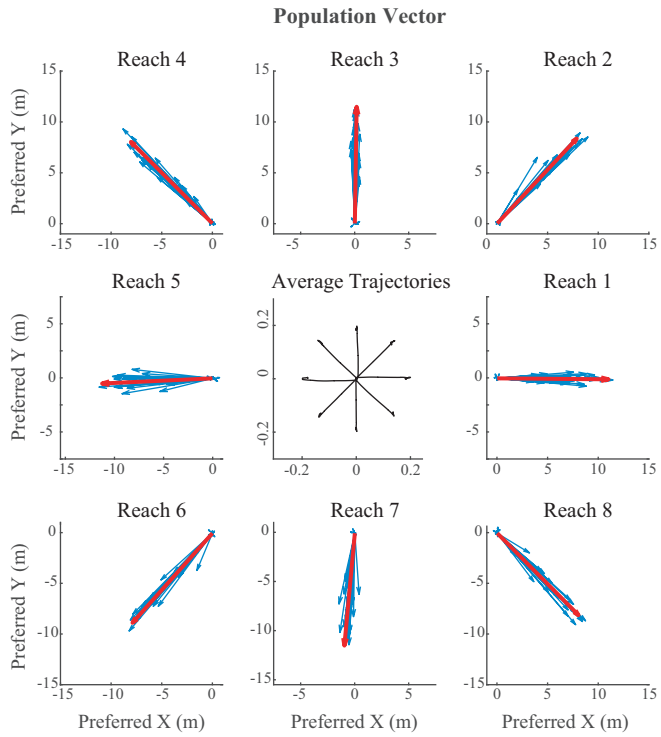

Figure 8) Population vectors (red arrows) for each reach are displayed along with their individual neuron components (blue arrows).

Do neural populations encode limb dynamics in their firing patterns? Performing the identical analysis as done in Churchland et al., we too find oscillatory trajectories in low dimensional projections of firing rate (jPCA) for center out reaches (Figure 9A) as well as 100 random reaches (Figure 9B). Thus the simulated data of our normative neuro-motor model is consistent with the hypothesis that neurons are encoding the dynamics of movement. Furthermore,

we re-trained our network to make reaches in a force field. The original network weights are re-trained to implement an LQR feedback controller for a velocity-dependent clockwise curl field Overall the average change in preferred direction was 25.94^{o}, thus rotating into the direction of the field. Interestingly, we find that while some neurons' preferred directions have changed, others have not (Li et al, 2001; Padoa-Schioppa et al., 2004).

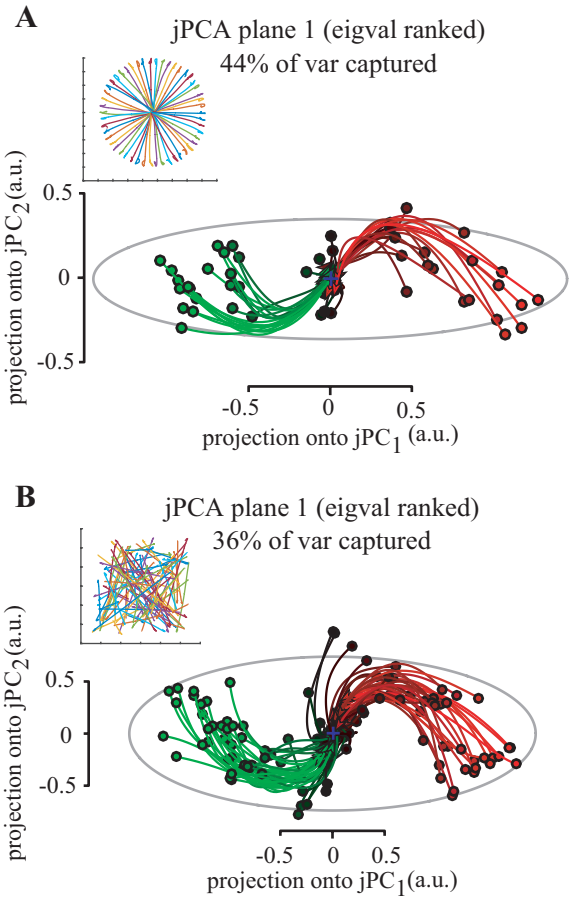

Figure 9) A dynamical systems analysis of average firing rates identical to that presented in [15]. Even though center out **A)** and random **B)** reaches are not cyclic a low-dimensional projection of their activity appears to be.

3.1.3 Normative Explanation

With our normative approach, we can uncover the "black box" of the network and analyze when, and why each neuron fires to understand the population as a whole. First we note that despite the evidence, our neurons are not encoding any of the experimentally motivated descriptions described above (i.e. tuning curves and limb dynamics). These features are in fact epiphenomena. The neurons collectively encode the linear mapping from state errors to motor units that results in the LQR controller.

As explained above, the neurons of the output layer use the spikes from the hidden layer to probabilistically command appropriate motor units and generate forces. Much like tuning curves, we can visualize these probabilities by looking at how they change as we vary errors in position and velocity (Fig 10). While the hidden neurons only depend on state errors, we can compute the probability of an output neuron spiking conditioned on state errors by integrating out their dependence on the hidden states (Figure 10A-C). From these plots we see how movement across the workspace results in activity that is correlated with reach direction and velocity. For example, some neurons fire with higher probability when the hand is to the left of the target (Figure 10D), some to the right (Figure 10E), whereas some are silent during these movements and only fire during larger speeds and positional errors (Figure 10F).

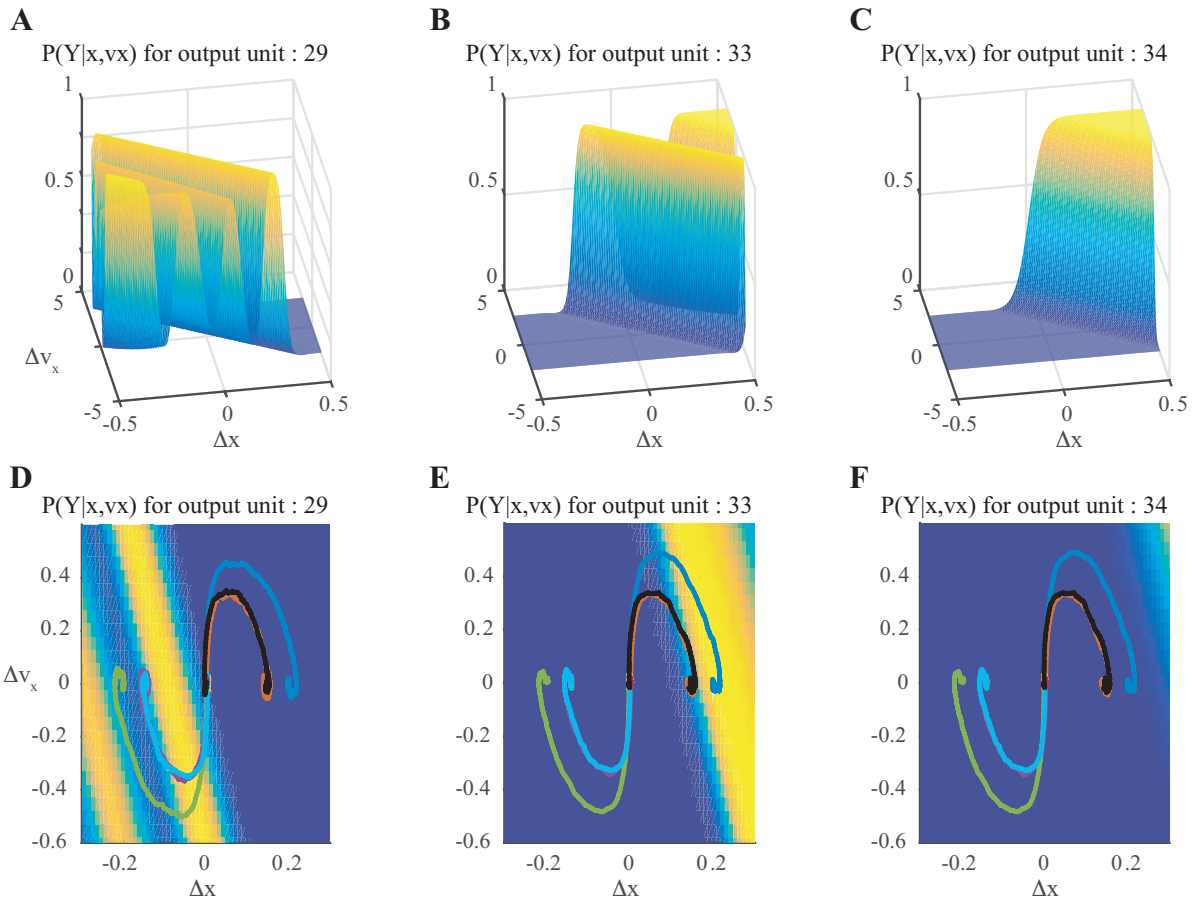

Figure 10 A), B), C) The probability of firing conditioned on errors in x and v_x for three of the twelve output neurons. The non-trivial shape is the result of designing the neurons to activate motor units that precisely approximate the LQR control signal. **D), E), F)**, The x and v_x components of the average center out reaches projected onto the same distributions shows how the neurons correlate with reach directions to the right and left or remain silent, respectively.

Finally, with regard to the oscillatory components of the networks firing rates, we see that this behavior is due to the dynamics of the limb itself. Not only does the network not encode dynamics, it has no dynamics itself. The firing rate trajectories are a consequence of the coupling of the limb with the network. In particular, the velocity of the limb beginning and ending at zero, gives rise to a subspace that appears oscillatory.

3.2 A Spiking, Network Controller with Binary Encoded State Input

By using binary encoded inputs, our goal was to approach a more realistic model. Since neurons can only convey information in spikes, it seemed impractical to use continuously valued state inputs for the controller. We will use similar analyses, here, to compare the physical and neuronal properties of the model with these new inputs.

3.2.1 Virtual Experiment: Limb Dynamics

As expected, providing a network with a binary encoding of the state rather than continuous values led to similar, but noisier results in both the physical and neural components of the model. Using the same simulated experimental setup as before (8 center-out reaches), we found that reaches were relatively straight and had bell-shaped velocity profiles (Figure 11A, B) with movement times of 0.87±0.08 s. Furthermore, the simulated hand's path curvature can be quantified with initial angles and path length (See Methods). These metrics displayed small values of curvature, which were 0.12±1.61° and 1.07±0.03, respectively. As before, reaches also had bell-shaped velocity profiles with positive skew. Due to the binary encoding, there was now a noticeable overshoot of 1.09±0.01 (or 9% overshoot). Small overshoot is common in healthy subjects, but it is often exacerbated in those with movement abnormalities. Therefore, this metric will be used to compare models and human experiments going forward.

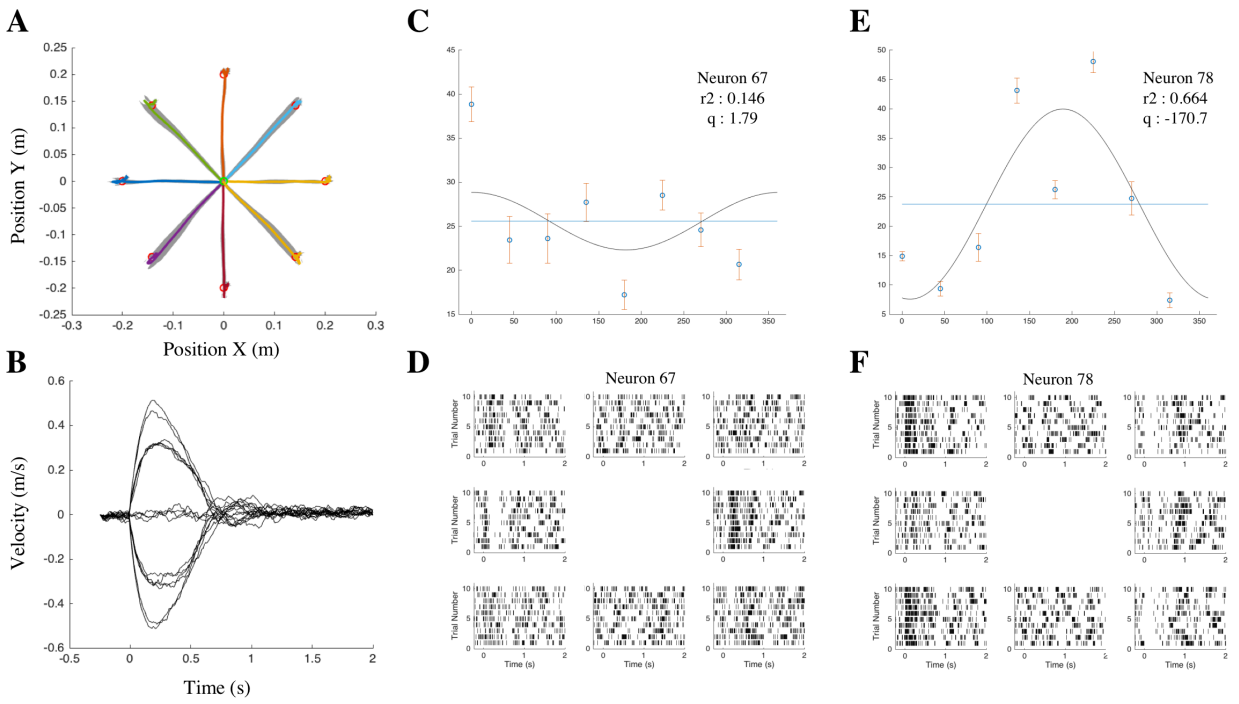

Figure 11 A), **B)** Average paths with SEM (shaded region) and velocity profiles for the simulated 8 center-out reach directions with binary encoded input. **C)**, **E)** Two neuron's tuning curves show preferences for certain reaches with a weak to average cosine fit. **D)**, **F)**, Raster plots for the same two neurons are displayed for each of the 8 reach directions. They reveal a strong preference for movements to the right (D) and for the leftward diagonal movements (F)**.**

3.2.2 Virtual Experiment: Neural Properties

While the physical characteristics of movement were largely similar, the electrophysiological results displayed new differences with binary encoded inputs. As before, many neurons exhibited directional tuning. Of the 126 active neurons, 66 of them showed significant cosine tuning ($R^2 > 0.5$). In contrast, some neurons adopted more unique tuning preferences, such as only increasing activity for one of the reaches (Figure 11C, D) or to only diagonal reaches in one direction (Figure 11E, F). With continuous state input, most tuning profiles varied smoothly with reach angle. The binary encoded inputs, however, are nonlinear. It is intuitive then, that non-smooth tuning profiles developed as a result of these new inputs.

We can study the network further on a population level. Here, cosine shaped tuning curves were once again used to construct population vectors. We found that the population vector pointed in the direction of the intended movement (Figure 12) with an average deviation of -0.20±6.89°. As the standard deviation is roughly three times larger, it is evident that the binary encoding adds noise to this analysis. Interestingly, when analyzing the low dimensional oscillations, we noticed projections that qualitatively appeared more similar to those found in the literature (Figure 13, Churchland et al. 2012). In the previous results, the trajectories started far from the origin and rotated towards it, yet in the new results and in the literature these trajectories start at the origin, oscillate away and then back to the center. Additionally, the percentage variance explained in the projection was closer to what was found in the literature (8% difference compared to the 16%).

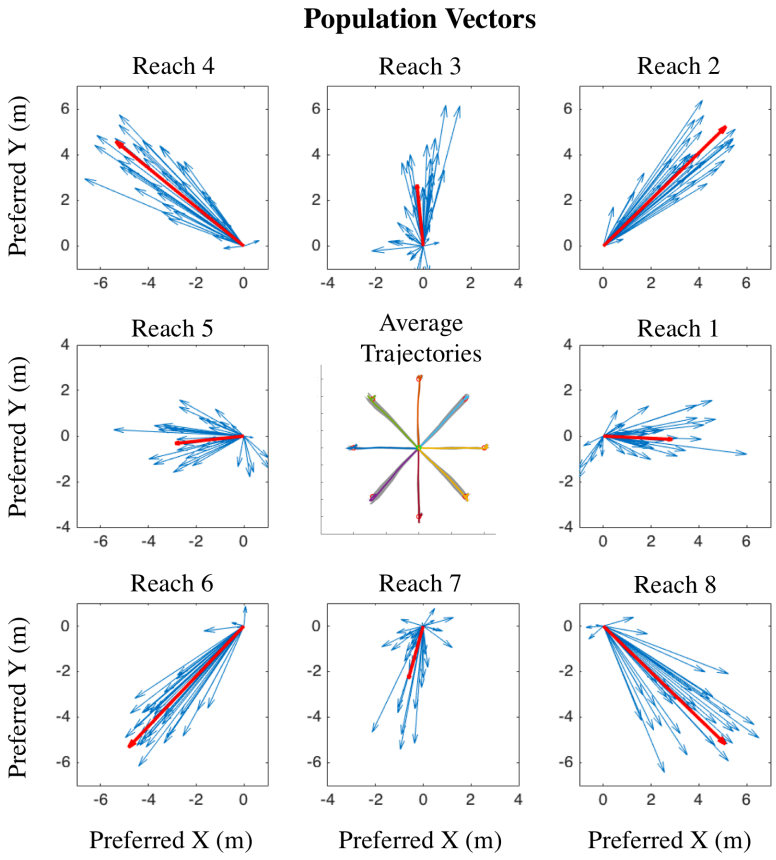

Figure 12) Population vectors (red arrows) for the controller network with binary encoded inputs are displayed along with their individual neuron components (blue arrows).

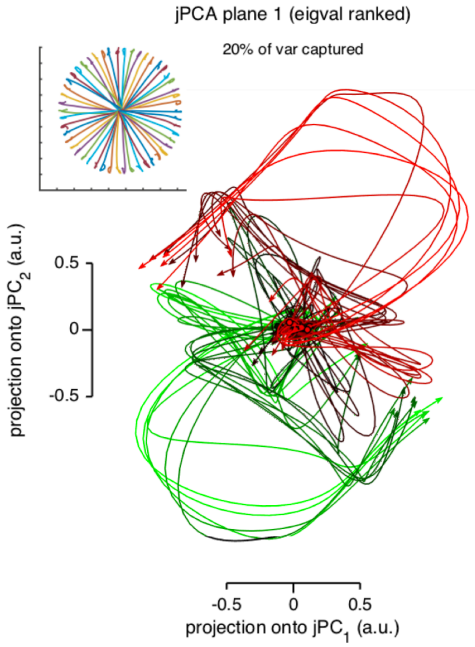

Figure 13) A dynamical systems analysis of average firing rates during 50 center out reaches. The trajectories display oscillatory behavior with clustered preparatory states (colored circles at origin).

3.3 A Spiking, Network Controller and State Estimator

In reality, the brain cannot intuitively know the state of the hand. It must get this information from either a noisy observation of the state (vision or proprioception) or through a well-tuned forward model prediction. A biological forward model prediction would rely on the previous state estimate and the efference copy of the motor command given to the muscles. To more appropriately simulate this in our model, two additional networks were trained to implement state estimation. One network used state and command information to approximate a forward model prediction of the next state. Another, acting as a Kalman filter, combined the state prediction

with a noisy observation of the state. It was imperative that the estimated state was highly precise to exhibit smooth control of the plant with a spiking controller network. Therefore, several stages of tests were utilized to locate the noise in the system and train these networks to produce viable state estimates. The mean squared error (MSE) was used to compare the model estimated state trajectories to those produced from an LQR controller with exact state information.

3.3.1 Virtual Experiment: Limb Dynamics

The first test was to generate an accurate state estimate with what will be referred to as a One-Step prediction (see Figure 14A for control architecture and trajectories). Here, the state is known exactly and the command is generated from this exact state. The forward model used these two values to make a prediction at each time step. The total MSE for these 8 trajectories was 2.47, the lowest value of the four tests. Similar to previous results the reaches were straight (angular deviation: 0.49±1.97°, path curvature: 1.10±0.03, movement time: 0.92±0.03s) the velocity profiles were bell-shaped, and contained small overshoots 1.09±0.02. Therefore, the network passed this first test, but further analysis was needed to assess how small errors in the network integrate over time.

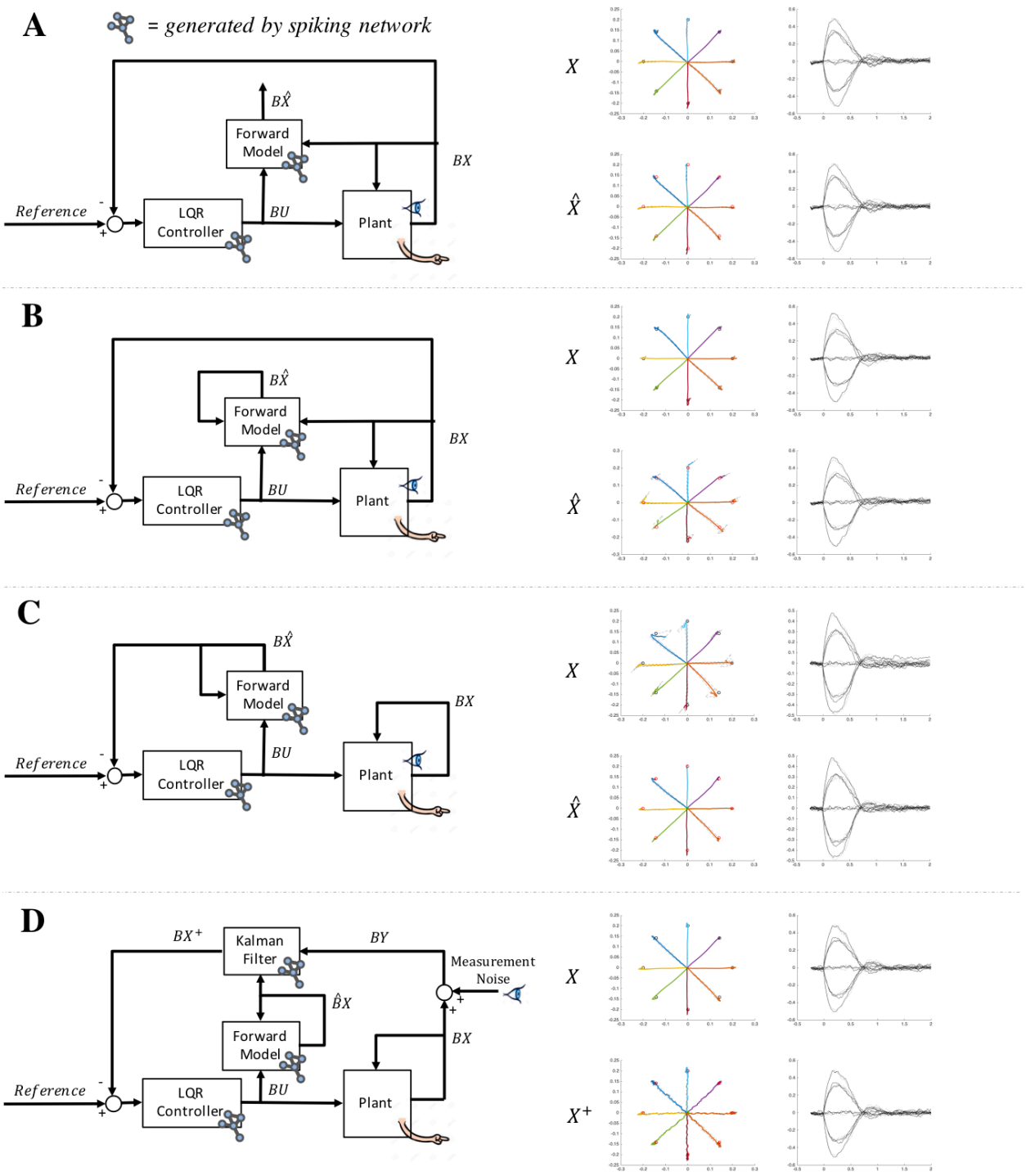

Figure 14) For all plots, the control architecture (left) generated a plant state trajectory (right, X) and a state estimate (right, $\hat{X}$ or X^{+}) with SEM (dashed line). For network inputs and outputs, these variables are binary encoded as BX, B$\hat{X}$ and BX^{+}. The state estimate produced by the networks were evaluated with a one-step prediction **A)**, feedback control **B)**, feedforward control **C)**, and feedforward control with Kalman Filter **D)**.

To better understand error integration in the model, a feedback controller was used to control the limb. Here, the forward model ran in parallel, not affecting the plant. It used its own prediction as its next state input (see Figure 14B for control architecture and trajectories). In general, these reaches move straight to the target (angular deviation: -0.29±2.39°, path curvature: 1.06±0.02, movement time: 0.86±0.04 s), but the estimated state drifts away from the target after a short time (Figure 14B, $\hat{X}$ plot). These reaches had an MSE of 4.26, which was the highest value. Again, slight overshoots were observed in these reaches (1.09±0.02). The next step was to test models ability to generate command purely on the feedforward signal.

To create a feedforward prediction, the control law is calculated from the error in the state estimate. Therefore, no sensory estimation was used to control the plant (see Figure 14C for control architecture and trajectories). Here, the state estimate converges to the target, but the plant drifts after reaching the target (see Methods for analysis). Despite the individual reaches drifting from the target, the average of each reach was relatively straight. Therefore, the MSE on the average reach trajectories were lower than the previous test 3.16. The reaches had an increase in overshoot (1.16±0.06), longer movement times (0.94±0.09 s), and more variable initial angular deviation (1.12±3.28°). There was also a large increase in path length (1.26±0.35), which was clearly due to the drift in the plant not necessarily a curving reach. It is important to note that these trajectories can reach the target with feedforward control alone. It was apparent, however, that a feedback signal was needed for the precise control needed to keep the limb from drifting off target.

In this final test, a network that approximated a Kalman filter combined the forward model prediction with a noisy observation of the state to make the next state prediction (see Figure 14D for architecture and trajectories). As expected, the addition of a noisy sensory observation keeps the plant from drifting off the target, and the MSE further reduced to 3.07. The average trajectories were again straight (angular deviation: -0.04±1.64°, path curvature: 1.08±0.04, movement time: 0.89±0.05 s) with positively skewed velocity profiles. The overshoot reduced from the previous test to 1.10±0.03. These results indicated that three spiking networks implementing LQR control, a forward model, and a Kalman filter can generate realistic and accurate reaching movements. This model was considered the model of "healthy" reaching movements, or Model H, and all comparisons to patients with movement pathologies start with this model as the base.

3.3.2 Virtual Experiment: Neural Properties

In the controller network, the input was an encoded state error signal and the output was a binary encoded command. The inputs and outputs of the forward model and Kalman Filter were also combinations of encoded state and encoded command. Therefore, these networks performed computations on similar variables and the spiking activity should reflect this. It was not expected, then, that the neural properties would differ strongly among the three networks.

For the forward model, half of the active neurons (58 of 136) exhibited cosine shaped tuning profile ($R^2 > 0.5$). As before, some of these neurons had very strict cutoffs for phasic activity (closer to a square wave, see Figure 15A. B), while others appeared to transition smoothly (Figure 15C, D). These tuning profiles were used to calculate a population vector that pointed within 0.99±9.16° of the direction of movement (Figure 15E). It is important to note that this population vector is slightly noisier than what was reported on previous networks. This population

vector had the integrated noise of three spiking network models, whereas the previous population vector only contained noise from the controller network. The Kalman filter's neural activation properties were nearly identical to that of the forward model (roughly half of the active neurons were tuned and population vector was within 0.12±8.97° of the direction of movement). For more detailed results, the representative neural analysis for the Kalman filter network can be found in Figure 16. Overall, no differences were observed between the forward model and Kalman filter networks using these neural analyses.

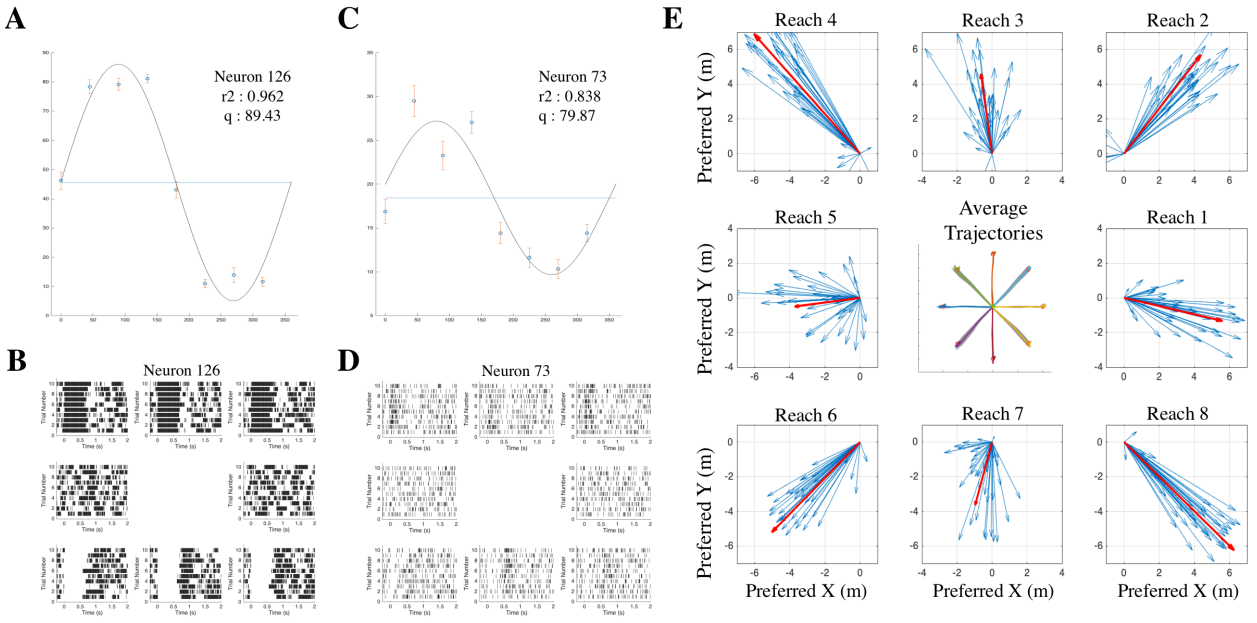

Figure 15) Simulated electrophysiological results of the forward model network during the 8 center-out reaches task, similar to Figures 11 and 12. **A)**, **C)** Two neuron's tuning curves show sharp and smooth tuning profiles. **B)**, **D)**, Raster plots for the same two neurons are displayed for each of the 8 reach directions. **E)** Population vectors for the 8-center out reaches.

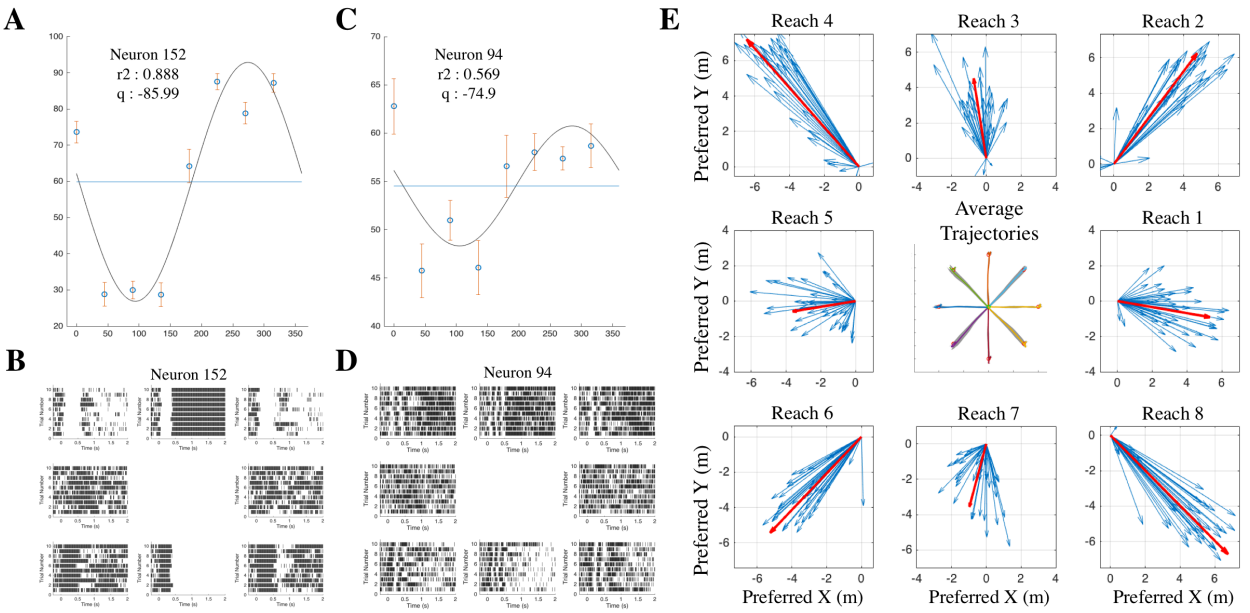

Figure 16) Simulated electrophysiological results of the Kalman filter network during the 8 center-out reaches task. **A)**, **C)** Two neuron's tuning curves also show sharp and smooth tuning profiles as in Figure 15. **B)**, **D)**, Raster plots for the same two neurons are displayed for each of the 8 reach directions. **E)** Population vectors for the 8-center out reaches.

When looking at the low dimensional projections, some interesting differences were observed among the three networks. The controller network trajectories initially start at the origin, disperse outward from it, and then circle back to the center (Figure 17A). On the other hand, the forward model and Kalman filter networks had preparatory activity that was more dispersed (Figure 17B, C) before the oscillations began. A similar phenomenon was observed in Churchland et al. when recording from the dorsal premotor and primary motor cortex. While only qualitative in nature, a comparison with our model provides evidence for hypotheses of what computations occur in these brain regions. For the primary motor cortex, it has been suggested that the activity in this region is directly related to the control of muscles during the execution of movement (Georgopoulos et al., 1986). Therefore, little preparatory activity in this region should be expected before movement onset, regardless of the distance, direction, or forces related to the reach. The

starting points for the neural trajectories in the low dimensional projections should then be clustered together. In our network controller, this phenomenon is observed similarly (Figure 17A). The premotor cortex, however, has long been associated with movement preplanning (Kaufman et al., 2010; Messier and Kalaska, 2000; Churchland et al., 2006). Across random reaches, the preplanning of movement would be unique for the different movement directions, magnitudes, etc. It is intuitive, then, that the preparatory activity of the neurons would not be similar among several unique reaches. Here, the forward model and Kalman filter networks display similar preparatory activity characteristics (Figure 17B, C). Therefore, the controller network seems to resemble a different region of the brain than the forward model and Kalman filter networks.

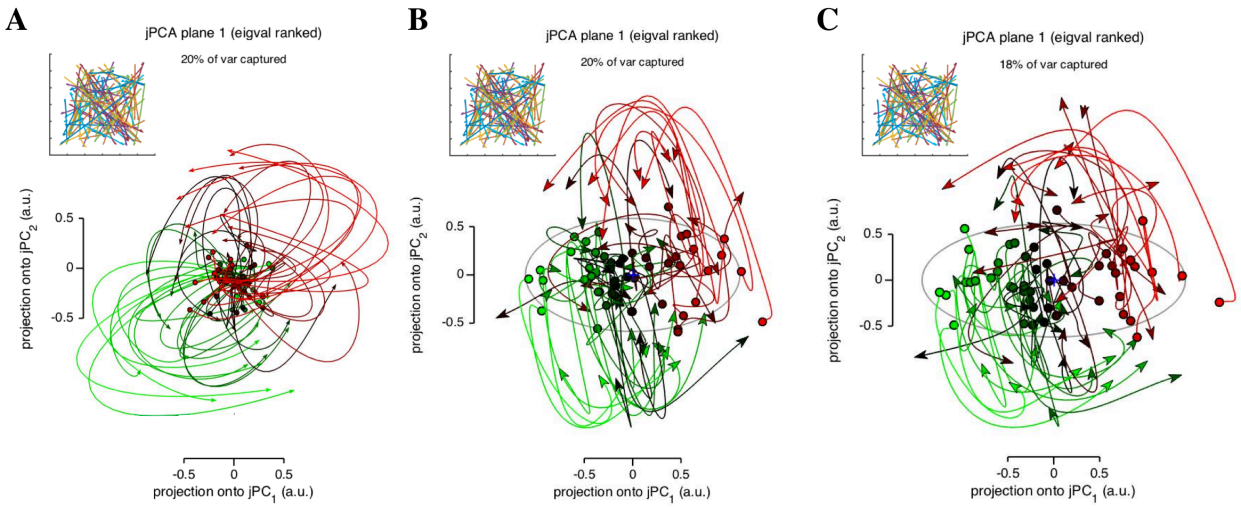

Figure 17) A dynamical systems analysis of average firing rates during 50 random reaches for the controller **A),** forward model **B),** and Kalman filter **C)** networks. As before, all three exhibited low dimensional oscillations. Importantly, the preparatory activity (circles) for the controller networks is clustered at the origin, but spread along the first principle axis for the other two networks (as seen in PMd recordings in Churchland et al. 2012).

A normative explanation for these differences can then be derived from this model. The controller network computes an appropriate muscle activity given a state error signal. Therefore, there would be relatively little preparatory activity before the movement onset, leading to the clustered initial points of the jPCA trajectories. Conversely, the preparatory activity of the forward

model and Kalman filter networks are computing future values of state. Since these starting values are unique for different random reaches, the preparatory activity is spread out along the axes of the low dimensional projection. Therefore, this model provides causal evidence that the M1 and PMd regions of the brain could compute the command and state estimate (or estimated state error), respectively.

3.4 Normative Models for Movement Disorder

The previous evidence showed that a collection of three networks performing approximate computations of LQR control, forward model, and Kalman filter can reasonably mimic the physical and neural characteristics of healthy, volitional movements. To generate abnormal movements, neurons in these networks were systematically "killed". The source of damage in these artificial networks may provide insight into the causes of biological movement abnormalities and disorders. Mathematically, killing a neuron was accomplished by setting the probability of a neuron firing to zero. Initially, neurons were killed in a methodical manner. All of the networks' outputs were binary encodings of a state or command. The decoding matrices (See Methods) provided clear indications of how each binary node impacted the discrete value that was being encoded. Therefore, it was possible to choose nodes that were likely to cause the most severe effect.

Starting with the forward model, we killed neurons of high and low impact on the encoded value of position and velocity (Figure 18). When high-impact encoding neurons were killed (Figure 18A), oscillations of the hand's path are observed in the damaged direction. Killing lower impact neurons (Figure 18B) had little effect on the trajectory. It is important to note that some neurons were very important to accurately encode a state value. When these neurons were killed,

some reaches were unstable (Figure 18C). For the network with high-impact damage, a significant increase in path length, overshoot and initial angular error was observed ($p<1e^{-6}$, $p<1e^{-4}$, and $p<1e^{-10}$, respectively). Hence, unique trajectories could be achieved by killing high-impact neurons.

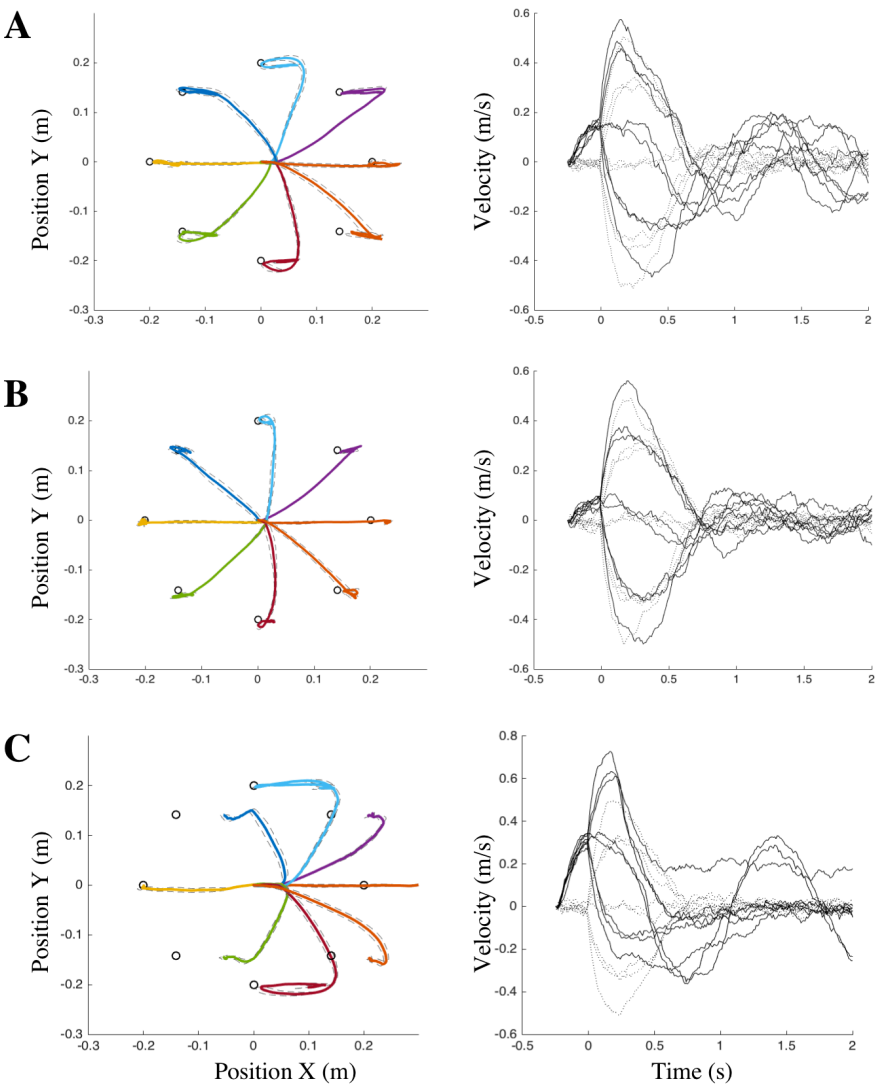

Figure 18) A) Hand path trajectories and velocity profiles for a model with a high-impact neuron killed from the forward model network. Similar plots are shown for a killed low-impact neuron **B)** and a neuron that when killed led to the control of a reach (moving right) to become unstable **C)**.

A similar process was used to generate abnormal reaching behavior for the Kalman Filter network. The output was also an encoding of state, so the results were largely expected to be

similar to that of the forward model. Qualitatively, damaging neurons in the sensory observation (Figure 19A) encoding appeared to have less of an effect than damaging an equivalent neuron involved with the forward model output (Figure 19B). For sensory observation damage, the trajectories demonstrated a significant increase in path length ($p<1e^{-4}$) compared to Model H, but no difference could be found in percent overshoot or angular error ($p=0.33$ and $p=0.056$, respectively). This result indicated that the trajectories deviated at the end of some reaches, but before the target was reached. This type of error was not seen in the simulations with forward model damage (compare with Figure 18A). When the neurons representing the forward model prediction in the Kalman filter were killed, the paths are similar to Figure 18A. All metrics were significantly increased compared to Model H (overshoot: $p<1e^{-6}$, initial angular error: $p<1e^{-10}$, path length: $p<1e^{-10}$, and movement time $p<1e^{-10}$). A comparison could also be drawn between the model in Figure 19A and 19B. Killing the neurons associated with the forward model prediction (19B) produced a significant increase in all metrics compared to killing those associated with sensory feedback information (overshoot: $p=0.02$, initial angular error: $p<1e^{-8}$, path length: $p<1e^{-4}$, and movement time $p<1e^{-4}$). Therefore, distinctly different trajectories were generated depending on the type of state information that was damaged.

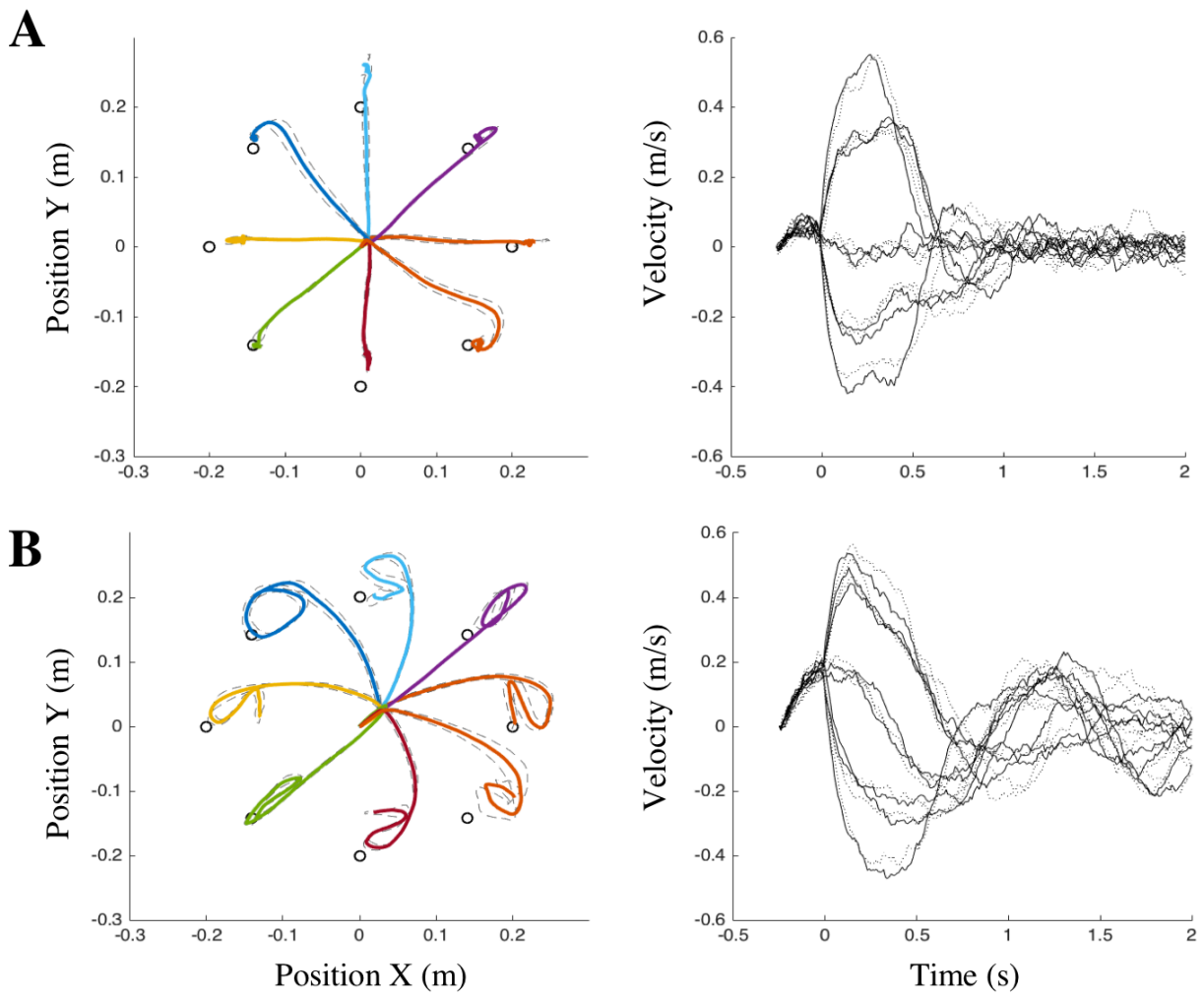

Figure 19) A) Hand path trajectories and velocity profiles for a model with killed high-impact neurons that were involved with the sensory feedback information in the Kalman filter. Similar plots for killing the equivalent neurons that contributed to forward model information **B)**. Killing equivalent neurons related to the forward model signal in the Kalman filter had a larger effect on reaching accuracy and curvature.

Finally, the impact of neuron death on the controller network was also analyzed. One intuitive way of affecting the network would be to kill output nodes, which are directly responsible for muscle activity. By destroying this communication, the muscle units would not fire when commanded by the controller. By killing neurons connected to small force producing muscle units (Model C1), jerky, imprecise movements arose due to the lack of fine control of the limb near the target (Figure 20A). These movements had a significant increase in curvature, compared to Model

H, based on path length and initial angle ($p<5e^{-4}$ and $p<2e^{-6}$, respectively). The model's inability recruit lower force producing muscles to slow movements caused a drastic increase in overshoot and movement time ($p<1e^{-10}$ and $p<1e^{-10}$, respectively). Killing neurons connected to the larger force producing muscle units (Model C2) also led to much slower movements compared to the healthy model ($p<1e^{-10}$). Many of the reaches did not finish given the time constraint of the simulation (Figure 20B). Interestingly, Model C2 (Figure 20B) had no difference in initial angle and even a significant decrease in path length ($p=0.02$), indicating that these reaches were less curved than the healthy model. Damaging the command to the motor units, therefore, slowed movements, but demonstrated unique results depending on the force production of the damaged units.

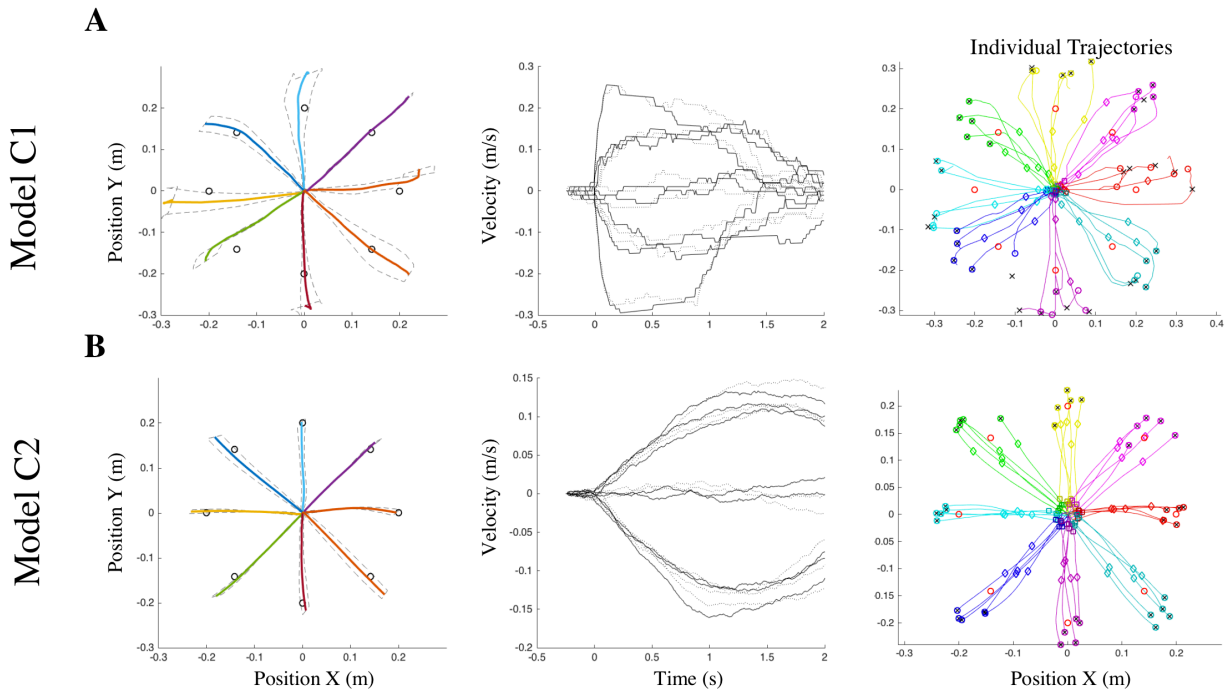

Figure 20) Hand path trajectories and velocity profiles for a model with killed neurons in the controller network. **A)** All neurons were killed except for those associated with muscle units contributing large forces. **B)** All neurons were killed except for those associated with muscle units contributing weak forces. The reaches in **B)** did not all come to a stop by the end of the 2 second simulation.

3.4.1 Randomized Cell Death in Networks

This normative approach of killing neurons provided key insights into how damage to each network impacts the physical characteristics of movement. To visualize the integrated effect of cell death across many networks, simulations were conducted with randomly killed neurons. Both the number of neurons (maximum of 10% of total neuron population) and the indices of the neurons to be killed were randomly chosen. While some simulations (<20%) lead to completely unstable trajectories, many of them produced interesting limb paths with different kinds of tremor-like movements. Some representative trajectories were selected to exemplify this diverse set of trajectories (Figure 21). On occasion, some disease simulations displayed asymmetric hand paths or velocity profiles, an unexpected result considering these movements are governed by point mass dynamics. It is important to note that each simulation is unique in its variance of movement, severity of overshoot, curvature, movement time, or type of tremor-like behavior. These characteristics were used to make quantitative comparisons to the trajectories of patients with movement disorders.

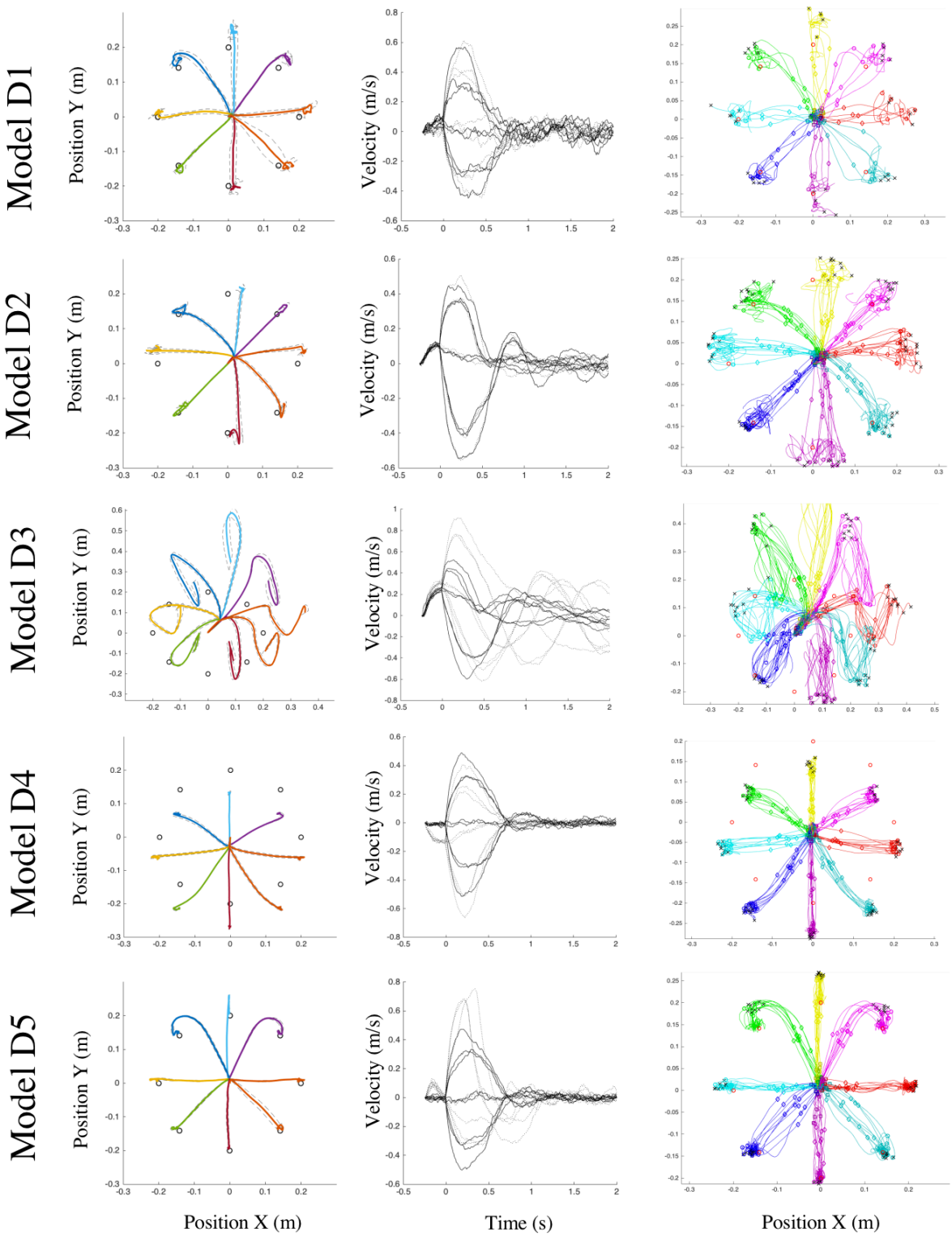

Figure 21) Average hand path trajectories (left), velocity profiles (middle) and individual hand path trajectories (right) for models with randomly killed neurons. The representative models were selected to display a diverse set of movement characteristics.

3.4.2 Simulating Movement Disorders

One common movement disorder, known as cerebellar tremor, is commonly found in patients with lesions in the upper brainstem, thalamus, or cerebellum. Common movement side effects of this disorder are slow kinetic tremors that worsen as a target is approached, or terminal tremor (Choi, 2016). This often leads to large overshoots at the end of the reach. These tremors are rarely accompanied by a resting or postural tremor. To compare with biological results, Model D1 from Figure 21 was selected as a representative model for a cerebellar tremor. Only neurons from the Kalman Filter network were damaged in Model D1, and all of them were involved with predicting the next position state. Qualitatively, this model displayed the characteristic intention tremor and overshoot, but importantly did not oscillate at the end of the reach (rest tremor). One study of reaching movements in a patient with Purkinje cell death found substantial increase in the variance in initial angle error compared to healthy subjects (Becker et al., 1991). The study reported the standard deviation of angle errors of ~2.8° for healthy subjects (approximated from Figure 3 in the article) and ~16.7° for the patient with cerebellum damage. Using the same method of computation as described in the article, we found that the trajectories of the healthy model and Model D1 had a standard deviation of 5.7° and 14.9°, respectively. No statistical comparisons were made, as only one patient was studied in this article, but the increase in standard deviation was similar enough to justify the comparison.

To further comprehend cerebellar tremor, Model A was compared with another study that analyzed the path length of patients' trajectories (Deuschl et al., 2000). These patients displayed tremor behavior due to either cerebellar dysfunction or one of two types of essential tremor. It is important to note that path length was calculated differently in this study and was only over the

deceleration phase of the movement. Additionally, the patients made 50 cm, three-dimensional reaches, whereas the simulations were 20cm, two-dimensional reaches. To compare exactly, path lengths were calculated over the distance of deceleration (see Deuschl et al., 2000) and scaled by 2.5 to simulate the longer reach. Using this scaled metric, Model H and Model D1 had an average path length of 5.9±8.6mm and 27.6±41.4mm, respectively. In the study, the path length for controls and Cerebellar disease patients were 5.4±1.3mm and 26.5±33.9mm. A student t-test comparing the simulated path lengths of Model H and D1 to that of the average control and cerebellar disease patient could not find statistical difference ($p=0.095$ and $p=0.796$, respectively). To further test whether the simulated metrics could have come from the distribution reported in the study, a two-sample Kolmogorov-Smirnov test (K-S test). Of 1000 samples from a half-normal distribution, 54% of them were not statistically different from Model D1 simulated data. A similar test found that all simulations of the control and patient distributions were statistically different from Model H simulations. Therefore, the simulated control group had a significantly larger standard deviation in path length compared to the control group in the study, but no significant difference was found between that of Model D1 and the patient group. These results provide evidence that damaging the integration of state signals leads to trajectories similar to that seen in patients with cerebellar disease; thus, supporting the hypothesis that one job of the cerebellum may be the integration of state information.

Victims of stroke also have abnormal movement behaviors. Depending on the epicenter of the stroke, many different characteristics can develop: weakness on the affected side (hemiparesis), parkinson-like tremor, and sudden involuntary jerking (Handley et al., 2009**)**. One study analyzed reaching behavior for patients who were one year removed from a stoke incident and had hemiparesis (Beer et al., 2000). When reaching to one of 16 center-out targets, a systematic angular

error was found that depended on the target direction (Figures 2, 4 in Beer et al.). To compare with these trajectories, Model D2 was selected. Model D2 had neuron damage related to the forward model prediction of position and the Kalman filter output of velocity. As in the Beer et al. experiment, 16 center-out reaches were simulated for a healthy and damaged model (Figure 22A, B). Qualitatively, the degree of curvature appears similar and there are clear differences of angular error across the target directions. For a statistical comparison, the angular error from Model H and Model D2 simulations were displayed as presented in Beer et al. (Figure 22C). The study found 7 of the 16 reaches had a significant increase in angular error for the patient's limb with hemiparesis. For the simulated movements, the angular error of Model D2 was significantly larger than Model H for 9 of the 16 movements, where Model H is used to simulate the unaffected limb. Interestingly, the disease model simulations and patient study both had the largest clockwise deviation between the 67.5°-225° targets. This was not expected as the simulations are generated from the dynamics of a point mass system and does not account for nonlinearities of the true limb. Therefore, simulated damage to networks calculating both the forward model prediction and integration of state signals lead to trajectories that are similar to stroke victims with hemiparesis.

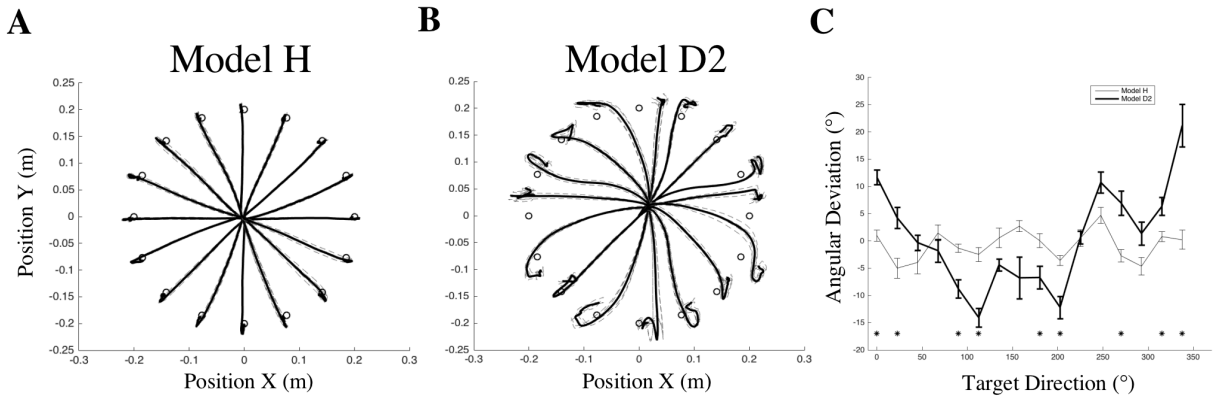

Figure 22 A) Average hand path trajectories for the healthy model, which for comparison to Beer et al. is assumed to be the patient arm not affected by stroke. **B)** Model D2 was selected from Figure 21 to compare with the patient's stroke-affected arm. **C)** The simulated angular errors over 16 targets, where a positive value corresponds with a counterclockwise error.

DISCUSSION

Here we've argued for a holistic neuro-motor approach to examining the motor system, combining known features of both limb biomechanics and the neurons that drive it. Taking a normative approach, we have designed a probabilistic network of spiking neurons to control movement of the limb. By design, the resulting reaches were largely consistent with known kinematic features of human and non-human primate movements. And while not designed as such, the electrophysiological data was also consistent with many statistical correlations observed experimentally. Furthermore, disrupting the neural activity led to abnormal reach trajectories with characteristics similar to patients with movement disorders. Importantly, we can analyze the causal mechanisms of our model to explain these phenomena.

In order for our networks to communicate in spikes, we needed to assume that an efficient encoding scheme existed to store information in binary digits. The encoding scheme chosen for this study was somewhat arbitrary. The binary tags associated with the discrete bins can be randomly rearranged to create many unique, yet viable mappings from bits to discrete values. Nevertheless, this encoding scheme was chosen for its capacity to represent the most discrete states per number of bits and could be decoded with a linear transformation. These characteristics allowed for quick conversion from a binary output to discrete values and an easy derivation for good initial weight values. One negative aspect of this encoding was that a single bit flip could lead to drastically different discrete value representations. For example, the first bit in the encoding essentially determined if the encoded value was a positive or negative number. These bit flips were mostly benign when close to zero, but for large values could lead to extreme differences. Despite this potential flaw, our spiking networks were able to learn to minimize the effect of bit flips on their output values. There are ways to avoid this bit flip issue. One could simply grid up the

workspace and assign one region of space to each neuron as in (Makin et al., 2013). However, to get the same resolution as reported in this study, it would require over 32,000 neurons to encode a value compared to the 15 used here. One way the brain could compensate for these bit flips would be to have many copies of the same network running in parallel with a grouping of neurons used to average over all the copies. Then, when a random bit flip does occur, it is diluted in the average and reduces the downstream impact.

Our incorporation of sensory feedback could be altered to approach a more biologically similar model. We assumed that the sensory observation was simply the true value of the state plus Gaussian noise with variance equal to the variance in the forward model prediction. While this appeared to be an intuitive choice, there was no biological inspiration behind the decision. Additionally, the Kalman filter network uses the relative variance of the signals to weigh each signal's contribution to the output. Therefore, the feedforward prediction and feedback command were weighed evenly. If the variance of the feedback signal were to change, then the Kalman filter network would need to be retrained on data from the new feedback signal. To test how the feedback noise could affect the system we trained networks on sensory observation with up to a 10-fold increase in variance. Even in this extreme case, the network produced reliable state estimates that were only slightly different from the base model (see Appendix F2). Finally, the issue of feedback delay was completely ignored in this study. Sensory information does not arrive immediately and usually must jump multiple synapses before reaching the cortex. It has been suggested that uncoordinated movements or tremors are caused by the brain's failure to handle this delay (Dovzhenok and Rubchinsky, 2012). To handle large time delays, a network approximating a Smith Predictor could be utilized (Smith, 1957; Ingimundarson and Hagglund, 2000). Any future work on this project will address this issue robustly.

It is important to state that the author of this manuscript was not trained to classify certain tremors or reaching pathologies. In general, this classification process is often performed by a highly trained physician and is not based on quantitative measurements. For example, the Movement Disorder Society's Unified Parkinson's Disease Rating Scale is simply a series 4-point scales based on the severity of oscillations during a series of tasks (Goetz et al., 2008). Therefore, when it is mentioned in this manuscript that a certain model exhibited characteristics of a tremor, this is merely a best guess given limited experience. Going forward, a trained physician could classify the simulated movements for more reliable comparisons to the motor diseases presented.

In this study, each model was trained separately based on data generated from our hypothesized control architecture. The training data for the networks used a perfect encoding of state error and command, which would not be available to a naïve brain. As we learn the dynamics of our limb or when the dynamics change for various reasons, errors from an inaccurate controller would be propagated to the forward model and errors in state estimation would be propagated back to the controller. Therefore, this model for motor control should not also be seen as a model for motor learning. This work assumed that reaches have been well practiced and that the network connectivity is mostly stationary. To train the networks in a more biological fashion, one could use the discrepancy between the forward model prediction and sensory feedback as a learning signal to train all the networks simultaneously. This hypothesis is popular among the motor learning community (Miall and Wolpert, 1996), and would be interesting to study within our normative model framework. In general, to study motor learning, some manner of concurrent network training should be adopted.

For this manuscript, all of the networks were examined using point mass dynamics. While the hand can be considered a point in space to be controlled, in reality it is the rotation of the lower

and upper limb that directs the location of the end effector. The debate over whether motor neurons encode extrinsic (hand-based) or intrinsic (limb-based) variables has been extensive (Georgopoulos, 1991, Mussa-Ivaldi, 1988; Todorov, 2000; Reina et al., 2001; Soechting & Flanders, 1992; Scott & Kalaska, 1997). Preliminary data of studying a network implementing LQR control of a 2-link arm has shown unique neural predictions based on the input during training (intrinsic vs. extrinsic, see Appendix F3). Using this methodology, the two hypotheses could be tested for their physical and neural similarity to biological data.

The idea that the brain uses a structured control architecture for movement is not novel. For many years, reaching movements have been studied under a variety of complex conditions to understand the control processes governing the upper limb. The novelty, here, lies in the normative method used to study this complex problem. By training artificial, spiking networks to implement unique and specific control functions, we have proposed truly falsifiable hypotheses. These hypotheses concern the existence of these control functions, where in the brain they could be clustered, and how damage to those areas are manifested in the physical reach. Evidence was provided that a network implementing LQR control alone mirrored the physical (straight reaches with bell-shaped velocity profiles) and neural (directionally tuned activity, population vectors, and oscillatory jPCA activity) characteristics commonly observed in reaching experiments. These features were largely maintained with the addition of spiking networks simulating a forward model and Kalman filter. Additionally, it was found that networks computing state related values produced simulated jPCA trajectories resembling those recorded from the premotor cortex, rather than the primary motor cortex. Finally, damaging these networks in unique ways generated trajectories similar to those found in patients with cerebellar disease and stroke, providing a hint to the causal link driving these impairments. These findings not only assist in our understanding

of volitional reaching movements, but also provide a solid framework for studying other control hypotheses executed by the brain.

REFERENCES

Ahmed, A. A., Wolpert, D. M., & Flanagan, J. R. (2008). Flexible representations of dynamics are used in object manipulation. Curr Biol, 18(10), 763-768. doi:10.1016/j.cub.2008.04.061

Alexander, R. M. (1997). A minimum energy cost hypothesis for human arm trajectories. Biol Cybern, 76(2), 97-105. doi:10.1007/s004220050324

Amirikian, B., & Georgopoulos, A. P. (2000). Directional tuning profiles of motor cortical cells. Neurosci Res, 36(1), 73-79. doi:10.1016/s0168-0102(99)00112-1

Atkeson, C. G., & Hollerbach, J. M. (1985). Kinematic features of unrestrained vertical arm movements. J Neurosci, 5(9), 2318-2330.

Becker, W. J., Morrice, B. L., Clark, A. W., & Lee, R. G. (1991). Multi-joint reaching movements and eye-hand tracking in cerebellar incoordination: investigation of a patient with complete loss of Purkinje cells. Can J Neurol Sci, 18(4), 476-487. doi:10.1017/s0317167100032194

Beer, R. F., Dewald, J. P., & Rymer, W. Z. (2000). Deficits in the coordination of multijoint arm movements in patients with hemiparesis: evidence for disturbed control of limb dynamics. Exp Brain Res, 131(3), 305-319. doi:10.1007/s002219900275

Berniker, M., Franklin, D. W., Flanagan, J. R., Wolpert, D. M., & Kording, K. (2014). Motor learning of novel dynamics is not represented in a single global coordinate system: evaluation of mixed coordinate representations and local learning. J Neurophysiol, 111(6), 1165-1182. doi:10.1152/jn.00493.2013

Berniker, M., Jarc, A., Bizzi, E., & Tresch, M. C. (2009). Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics. Proc Natl Acad Sci U S A, 106(18), 7601-7606. doi:10.1073/pnas.0901512106

Berniker, M., & Kording, K. (2008). Estimating the sources of motor errors for adaptation and generalization. Nat Neurosci, 11(12), 1454-1461. doi:10.1038/nn.2229

Berniker, M., & Kording, K. P. (2015). Deep networks for motor control functions. Front Comput Neurosci, 9, 32. doi:10.3389/fncom.2015.00032

Berniker, M., Mirzaei, H., & Kording, K. P. (2014). The effects of training breadth on motor generalization. J Neurophysiol, 112(11), 2791-2798. doi:10.1152/jn.00615.2013

Berniker, M., & Penny, S. (2019). A normative approach to neuromotor control. Biol Cybern, 113(1-2), 83-92. doi:10.1007/s00422-018-0777-7

Bhidayasiri, R. (2005). Differential diagnosis of common tremor syndromes. Postgrad Med J, 81(962), 756-762. doi:10.1136/pgmj.2005.032979

Bourdoukan, R., Barrett, D., Den`eve, S., and Machens, C.K., (2012). "Learning optimal spike-based representations," Ad- vances in neural information processing systems, pp. 2285-2293.

Caminiti, R., Johnson, P. B., Galli, C., Ferraina, S., & Burnod, Y. (1991). Making arm movements within different parts of space: the premotor and motor cortical representation of a coordinate system for reaching to visual targets. J Neurosci, 11(5), 1182-1197.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., & Shenoy, K. V. (2012). Neural population dynamics during reaching. Nature, 487(7405), 51-56. doi:10.1038/nature11129

Churchland, M. M., Santhanam, G., & Shenoy, K. V. (2006). Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach. J Neurophysiol, 96(6), 3130-3146. doi:10.1152/jn.00307.2006

Cohen, D. A., Prud'homme, M. J., & Kalaska, J. F. (1994). Tactile activity in primate primary somatosensory cortex during active arm movements: correlation with receptive field properties. J Neurophysiol, 71(1), 161-172. doi:10.1152/jn.1994.71.1.161

Cohen, J. D., Forman, S. D., Braver, T. S., Casey, B. J., Servan-Schreiber, D., & Noll, D. C. (1994). Activation of the prefrontal cortex in a nonspatial working memory task with functional MRI. Hum Brain Mapp, 1(4), 293-304. doi:10.1002/hbm.460010407

Corcos, D. M., Gottlieb, G. L., & Agarwal, G. C. (1989). Organizing principles for single-joint movements. II. A speed-sensitive strategy. J Neurophysiol, 62(2), 358-368. doi:10.1152/jn.1989.62.2.358

Danziger, Z., & Mussa-Ivaldi, F. A. (2012). The influence of visual motion on motor learning. J Neurosci, 32(29), 9859-9869. doi:10.1523/JNEUROSCI.5528-11.2012

Day, B. L., Thompson, P. D., Harding, A. E., & Marsden, C. D. (1998). Influence of vision on upper limb reaching movements in patients with cerebellar ataxia. Brain, 121 (Pt 2), 357-372. doi:10.1093/brain/121.2.357

Deneve, S., Alemi, A., & Bourdoukan, R. (2017). The Brain as an Efficient and Robust Adaptive Learner. Neuron, 94(5), 969-977. doi:10.1016/j.neuron.2017.05.016

Deneve, S., Duhamel, J. R., & Pouget, A. (2007). Optimal sensorimotor integration in recurrent cortical networks: a neural implementation of Kalman filters. J Neurosci, 27(21), 5744-5756. doi:10.1523/JNEUROSCI.3985-06.2007

Deuschl, G., Wenzelburger, R., Loffler, K., Raethjen, J., & Stolze, H. (2000). Essential tremor and cerebellar dysfunction clinical and kinematic analysis of intention tremor. Brain, 123 (Pt 8), 1568-1580. doi:10.1093/brain/123.8.1568

DeWolf, T., Stewart, T. C., Slotine, J. J., & Eliasmith, C. (2016). A spiking neural model of adaptive arm control. Proc Biol Sci, 283(1843). doi:10.1098/rspb.2016.2134

Dingwell, J. B., Mah, C. D., & Mussa-Ivaldi, F. A. (2004). Experimentally confirmed mathematical model for human control of a non-rigid object. J Neurophysiol, 91(3), 1158-1170. doi:10.1152/jn.00704.2003

Dizio, P., & Lackner, J. R. (1995). Motor adaptation to Coriolis force perturbations of reaching movements: endpoint but not trajectory adaptation transfers to the nonexposed arm. J Neurophysiol, 74(4), 1787-1792.

Dovzhenok, A., & Rubchinsky, L. L. (2012). On the origin of tremor in Parkinson's disease. PLoS One, 7(7), e41598. doi:10.1371/journal.pone.0041598

Farshchiansadegh, A., Melendez-Calderon, A., Ranganathan, R., Murphey, T. D., & Mussa-Ivaldi, F. A. (2016). Sensory Agreement Guides Kinetic Energy Optimization of Arm Movements during Object Manipulation. PLoS Comput Biol, 12(4), e1004861. doi:10.1371/journal.pcbi.1004861

Flanagan, J. R., & Rao, A. K. (1995). Trajectory adaptation to a nonlinear visuomotor transformation: evidence of motion planning in visually perceived space. J Neurophysiol, 74(5), 2174-2178.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. J Neurosci, 5(7), 1688-1703.
Fortier, P. A., Kalaska, J. F., & Smith, A. M. (1989). Cerebellar neuronal activity related to whole-arm reaching movements in the monkey. J Neurophysiol, 62(1), 198-211. doi:10.1152/jn.1989.62.1.198
Fortier, P. A., Smith, A. M., & Kalaska, J. F. (1993). Comparison of cerebellar and motor cortex activity during reaching: directional tuning and response variability. J Neurophysiol, 69(4), 1136-1149. doi:10.1152/jn.1993.69.4.1136
Fu, Q. G., Flament, D., Coltz, J. D., & Ebner, T. J. (1995). Temporal encoding of movement kinematics in the discharge of primate primary motor and premotor neurons. J Neurophysiol, 73(2), 836-854. doi:10.1152/jn.1995.73.2.836
Georgopoulos, A. P. (1991). Higher order motor control. Annu Rev Neurosci, 14, 361-377. doi:10.1146/annurev.ne.14.030191.002045
Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. J Neurosci, 2(11), 1527-1537.
Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1983). Interruption of motor cortical discharge subserving aimed arm movements. Exp Brain Res, 49(3), 327-340. doi:10.1007/BF00238775
Georgopoulos, A. P., Schwartz, A. B., & Kettner, R. E. (1986). Neuronal population coding of movement direction. Science, 233(4771), 1416-1419. doi:10.1126/science.3749885
Goetz, C. G., Tilley, B. C., Shaftman, S. R., Stebbins, G. T., Fahn, S., Martinez-Martin, P., . . . Movement Disorder Society, U. R. T. F. (2008). Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): scale presentation and clinimetric testing results. Mov Disord, 23(15), 2129-2170. doi:10.1002/mds.22340
Gordon, J., & Ghez, C. (1987). Trajectory control in targeted force impulses. III. Compensatory adjustments for initial errors. Exp Brain Res, 67(2), 253-269. doi:10.1007/BF00248547
Haith, A. M., & Krakauer, J. W. (2013). Model-based and model-free mechanisms of human motor learning. Adv Exp Med Biol, 782, 1-21. doi:10.1007/978-1-4614-5465-6_1
Handley, A., Medcalf, P., Hellier, K., & Dutta, D. (2009). Movement disorders after stroke. Age Ageing, 38(3), 260-266. doi:10.1093/ageing/afp020
Harris, C. M. (1998). On the optimal control of behaviour: a stochastic perspective. J Neurosci Methods, 83(1), 73-88. doi:10.1016/s0165-0270(98)00063-6
Hatze, H., & Buys, J. D. (1977). Energy-optimal controls in the mammalian neuromuscular system. Biol Cybern, 27(1), 9-20.
Hinton, Geoffrey E. "Training products of experts by minimizing contrastive divergence." *Neural computation* 14.8 (2002): 1771-1800.
Hocherman, S., & Wise, S. P. (1991). Effects of hand movement path on motor cortical activity in awake, behaving rhesus monkeys. Exp Brain Res, 83(2), 285-302.
Hoff, B., & Arbib, M. A. (1993). Models of Trajectory Formation and Temporal Interaction of Reach and Grasp. J Mot Behav, 25(3), 175-192. doi:10.1080/00222895.1993.9942048
Hogan, N. (1984). An organizing principle for a class of voluntary movements. J Neurosci, 4(11), 2745-2754.

Hu, Z., Hao, M., Xu, S., Xiao, Q., & Lan, N. (2019). Evaluation of tremor interference with control of voluntary reaching movements in patients with Parkinson's disease. J Neuroeng Rehabil, 16(1), 38. doi:10.1186/s12984-019-0505-0
Huang, H. J., & Ahmed, A. A. (2014). Reductions in muscle coactivation and metabolic cost during visuomotor adaptation. J Neurophysiol, 112(9), 2264-2274. doi:10.1152/jn.00014.2014
Huang, H. J., Kram, R., & Ahmed, A. A. (2012). Reduction of metabolic cost during motor learning of arm reaching dynamics. J Neurosci, 32(6), 2182-2190. doi:10.1523/JNEUROSCI.4003-11.2012
Ingimundarson, A., & Hägglund, T. (2000). Robust automatic tuning of an industrial PI controller for dead-time systems. *IFAC Proceedings Volumes*, *33*(4), 141-146.
Ingram, J. N., & Wolpert, D. M. (2011). Naturalistic approaches to sensorimotor control. Prog Brain Res, 191, 3-29. doi:10.1016/B978-0-444-53752-2.00016-3
Jones, K. E., Wessberg, J., & Vallbo, A. B. (2001). Directional tuning of human forearm muscle afferents during voluntary wrist movements. J Physiol, 536(Pt 2), 635-647. doi:10.1111/j.1469-7793.2001.0635c.xd
Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183-233.
Kalaska, J. F., Caminiti, R., & Georgopoulos, A. P. (1983). Cortical mechanisms related to the direction of two-dimensional arm movements: relations in parietal area 5 and comparison with motor cortex. Exp Brain Res, 51(2), 247-260. doi:10.1007/BF00237200
Kalaska, J. F., Scott, S. H., Cisek, P., & Sergio, L. E. (1997). Cortical control of reaching movements. Curr Opin Neurobiol, 7(6), 849-859. doi:10.1016/s0959-4388(97)80146-8
Kalman, Rudolf Emil. "Contributions to the theory of optimal control." *Bol. soc. mat. mexicana* 5.2 (1960): 102-119.
Kang, T., He, J., & Tillery, S. I. (2005). Determining natural arm configuration along a reaching trajectory. Exp Brain Res, 167(3), 352-361. doi:10.1007/s00221-005-0039-5
Kaufman, M. T., Churchland, M. M., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., & Shenoy, K. V. (2010). Roles of monkey premotor neuron classes in movement preparation and execution. J Neurophysiol, 104(2), 799-810. doi:10.1152/jn.00231.2009
Kingma, D. P., and Ba J. Adam. "A method for stochastic optimization. cornell university library." *arXiv preprint arXiv:1412.6980* (2017).
Kisiel-Sajewicz, K., Fang, Y., Hrovat, K., Yue, G. H., Siemionow, V., Sun, C. K., . . . Daly, J. J. (2011). Weakening of synergist muscle coupling during reaching movement in stroke patients. Neurorehabil Neural Repair, 25(4), 359-368. doi:10.1177/1545968310388665
Kistemaker, D. A., Wong, J. D., & Gribble, P. L. (2010). The central nervous system does not minimize energy cost in arm movements. J Neurophysiol, 104(6), 2985-2994. doi:10.1152/jn.00483.2010
Kistemaker, D. A., Wong, J. D., & Gribble, P. L. (2014). The cost of moving optimally: kinematic path selection. J Neurophysiol, 112(8), 1815-1824. doi:10.1152/jn.00291.2014
Kluzik, J., Diedrichsen, J., Shadmehr, R., & Bastian, A. J. (2008). Reach adaptation: what determines whether we learn an internal model of the tool or adapt the model of our arm? J Neurophysiol, 100(3), 1455-1464. doi:10.1152/jn.90334.2008
Krakauer, J. W., Ghilardi, M. F., & Ghez, C. (1999). Independent learning of internal models for kinematic and dynamic control of reaching. Nat Neurosci, 2(11), 1026-1031. doi:10.1038/14826

Krakauer, J. W., Pine, Z. M., Ghilardi, M. F., & Ghez, C. (2000). Learning of visuomotor transformations for vectorial planning of reaching trajectories. J Neurosci, 20(23), 8916-8924.
Kuo, A. D. (1995). An optimal control model for analyzing human postural balance. IEEE Trans Biomed Eng, 42(1), 87-101. doi:10.1109/10.362914
Lackner, J. R., & Dizio, P. (1994). Rapid adaptation to Coriolis force perturbations of arm trajectory. J Neurophysiol, 72(1), 299-313. doi:10.1152/jn.1994.72.1.299
Laje, R., & Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. Nat Neurosci, 16(7), 925-933. doi:10.1038/nn.3405
Le, Quoc V., "Building high-level features using large scale unsupervised learning," In Acoustics, Speech and Sig- nal Processing (ICASSP) IEEE International conference on, (2012).
Li, C. S., Padoa-Schioppa, C., & Bizzi, E. (2001). Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field. Neuron, 30(2), 593-607. doi:10.1016/s0896-6273(01)00301-4
Makin, A. D., Rampone, G., Pecchinenda, A., & Bertamini, M. (2013). Electrophysiological responses to visuospatial regularity. Psychophysiology, 50(10), 1045-1055. doi:10.1111/psyp.12082
Makin, J. G., Dichter, B. K., & Sabes, P. N. (2015). Learning to Estimate Dynamical State with Probabilistic Population Codes. PLoS Comput Biol, 11(11), e1004554. doi:10.1371/journal.pcbi.1004554
Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature, 503(7474), 78-84. doi:10.1038/nature12742
Marsden, C. D., Merton, P. A., Morton, H. B., & Adam, J. E. (1978). Feedback control of voluntary movements in man. Electroencephalogr Clin Neurophysiol Suppl(34), 507-510.
Mathew, A., & Cook, M. (1990). The control of reaching movements by young infants. Child Dev, 61(4), 1238-1257.
Messier, J., & Kalaska, J. F. (2000). Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task. J Neurophysiol, 84(1), 152-165. doi:10.1152/jn.2000.84.1.152
Micci Barreca, D., & Guenther, F. H. (2001). A modeling study of potential sources of curvature in human reaching movements. J Mot Behav, 33(4), 387-400. doi:10.1080/00222890109601922
Moll, L., & Kuypers, H. G. (1977). Premotor cortical ablations in monkeys: contralateral changes in visually guided reaching behavior. Science, 198(4314), 317-319.
Morasso, P. (1981). Spatial control of arm movements. Exp Brain Res, 42(2), 223-227.
Moreno-Bote, R. (2014). Poisson-like spiking in circuits with probabilistic synapses. PLoS Comput Biol, 10(7), e1003522. doi:10.1371/journal.pcbi.1003522
Mussa-Ivaldi, F. A. (1988). Do neurons in the motor cortex encode movement direction? An alternative hypothesis. Neurosci Lett, 91(1), 106-111. doi:10.1016/0304-3940(88)90257-1
Mutha, P. K., & Sainburg, R. L. (2007). Control of velocity and position in single joint movements. Hum Mov Sci, 26(6), 808-823. doi:10.1016/j.humov.2007.06.001
Nagasaki, H. (1989). Asymmetric velocity and acceleration profiles of human arm movements. Exp Brain Res, 74(2), 319-326. doi:10.1007/BF00248865

Nakano, E., Imamizu, H., Osu, R., Uno, Y., Gomi, H., Yoshioka, T., & Kawato, M. (1999). Quantitative examinations of internal representations for arm trajectory planning: minimum commanded torque change model. J Neurophysiol, 81(5), 2140-2155. doi:10.1152/jn.1999.81.5.2140

Neal, R. M. (1990). Learning stochastic feedforward networks. Department of Computer Science, University of Toronto, 64, 1283.

Oldfield, R. C. (1971). The assessment and analysis of handedness: the Edinburgh inventory. Neuropsychologia, 9(1), 97-113.

Padoa-Schioppa, C., Li, C. S., & Bizzi, E. (2004). Neuronal activity in the supplementary motor area of monkeys adapting to a new dynamic environment. J Neurophysiol, 91(1), 449-473. doi:10.1152/jn.00876.2002

Pandy, M. G., & Zajac, F. E. (1991). Optimal muscular coordination strategies for jumping. J Biomech, 24(1), 1-10. doi:10.1016/0021-9290(91)90321-d

Pearce, T. M., & Moran, D. W. (2012). Strategy-dependent encoding of planned arm movements in the dorsal premotor cortex. Science, 337(6097), 984-988. doi:10.1126/science.1220642

Pearl, J. (1988). Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier.

Pearson, K. G. (1995). Proprioceptive regulation of locomotion. Curr Opin Neurobiol, 5(6), 786-791. doi:10.1016/0959-4388(95)80107-3

Ramanathan, R., Eberhardt, S. P., Rahman, T., Sample, W., Seliktar, R., & Alexander, M. (2000). Analysis of arm trajectories of everyday tasks for the development of an upper-limb orthosis. IEEE Trans Rehabil Eng, 8(1), 60-70.

Rayner, J. M. (1993). On aerodynamics and the energetics of vertebrate flapping flight. Cont. Math, 141, 351-400.

Reina, G. A., Moran, D. W., & Schwartz, A. B. (2001). On the relationship between joint angular velocity and motor cortical discharge during reaching. J Neurophysiol, 85(6), 2576-2589. doi:10.1152/jn.2001.85.6.2576

Ribot-Ciscar, E., Butler, J. E., & Thomas, C. K. (2003). Facilitation of triceps brachii muscle contraction by tendon vibration after chronic cervical spinal cord injury. J Appl Physiol (1985), 94(6), 2358-2367. doi:10.1152/japplphysiol.00894.2002

Rondi-Reig, L., Paradis, A. L., Lefort, J. M., Babayan, B. M., & Tobin, C. (2014). How the cerebellum may monitor sensory information for spatial representation. Front Syst Neurosci, 8, 205. doi:10.3389/fnsys.2014.00205

Rosenbaum, D. A., Loukopoulos, L. D., Meulenbroek, R. G., Vaughan, J., & Engelbrecht, S. E. (1995). Planning reaches by evaluating stored postures. Psychol Rev, 102(1), 28-67.

Sainburg, R. L., Ghez, C., & Kalakanis, D. (1999). Intersegmental dynamics are controlled by sequential anticipatory, error correction, and postural mechanisms. J Neurophysiol, 81(3), 1045-1056.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J., "Learning internal representations by error propagation," (No. ICS-8506) California Univ San Diego La Jolla Inst for Cognitive Science, (1985).

Sainburg, R. L., Poizner, H., & Ghez, C. (1993). Loss of proprioception produces deficits in interjoint coordination. J Neurophysiol, 70(5), 2136-2147. doi:10.1152/jn.1993.70.5.2136

Sarlegna, F. R., & Sainburg, R. L. (2009). The roles of vision and proprioception in the planning of reaching movements. Adv Exp Med Biol, 629, 317-335. doi:10.1007/978-0-387-77064-2_16

Saul, L. K., Jaakkola, T., & Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. Journal of artificial intelligence research, 4, 61-76.
Saul, L., & Jordan, M. (1998). A mean field learning algorithm for unsupervised neural networks. In Learning in graphical models (pp. 541-554). Springer, Dordrecht.
Scheidt, R. A., Conditt, M. A., Secco, E. L., & Mussa-Ivaldi, F. A. (2005). Interaction of visual and proprioceptive feedback during adaptation of human reaching movements. J Neurophysiol, 93(6), 3200-3213. doi:10.1152/jn.00947.2004
Scholz, J. P., Schoner, G., & Latash, M. L. (2000). Identifying the control structure of multijoint coordination during pistol shooting. Exp Brain Res, 135(3), 382-404.
Scott, S. H., & Kalaska, J. F. (1997). Reaching movements with similar hand paths but different arm orientations. I. Activity of individual cells in motor cortex. J Neurophysiol, 77(2), 826-852. doi:10.1152/jn.1997.77.2.826
Scott, S.H., "Optimal feedback control and the neural basis of volitional motor control," Nature Reviews Neuroscience, vol. 5, pp. 532, (2004).
Selinger, J. C., O'Connor, S. M., Wong, J. D., & Donelan, J. M. (2015). Humans Can Continuously Optimize Energetic Cost during Walking. Curr Biol, 25(18), 2452-2456. doi:10.1016/j.cub.2015.08.016
Shadmehr, R., & Mussa-Ivaldi, F. A. (1994). Adaptive representation of dynamics during learning of a motor task. J Neurosci, 14(5 Pt 2), 3208-3224.
Siniscalchi, A., Gallelli, L., Labate, A., Malferrari, G., Palleria, C., & Sarro, G. D. (2012). Post-stroke Movement Disorders: Clinical Manifestations and Pharmacological Management. Curr Neuropharmacol, 10(3), 254-262. doi:10.2174/157015912803217341
Smith, A. M., Dugas, C., Fortier, P., Kalaska, J., & Picard, N. (1993). Comparing cerebellar and motor cortical activity in reaching and grasping. Can J Neurol Sci, 20 Suppl 3, S53-61.
Smith, O.J.M. (1957) "Closed control of loops with dead time." Chemical Engineer-ing Progress, 53, May, pp.217-219
Soechting, J. F., & Flanders, M. (1992). Moving in three-dimensional space: frames of reference, vectors, and coordinate systems. Annu Rev Neurosci, 15, 167-191. doi:10.1146/annurev.ne.15.030192.001123
Stengel, R. F. Optimal Control and Estimation (Dover, New York, 1994)
Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. Neuron, 63(4), 544-557. doi:10.1016/j.neuron.2009.07.018
Sussillo, D., & Barak, O. (2013). Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. Neural Comput, 25(3), 626-649. doi:10.1162/NECO_a_00409
Thalmeier, D., Uhlmann, M., Kappen, H. J., & Memmesheimer, R. M. (2016). Learning Universal Computations with Spikes. PLoS Comput Biol, 12(6), e1004895. doi:10.1371/journal.pcbi.1004895
Todorov, E. (2000). Direct cortical control of muscle activation in voluntary arm movements: a model. Nat Neurosci, 3(4), 391-398. doi:10.1038/73964
Todorov, E. (2004). Optimality principles in sensorimotor control. Nat Neurosci, 7(9), 907-915. doi:10.1038/nn1309
Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. Nat Neurosci, 5(11), 1226-1235. doi:10.1038/nn963
Tucker, V. A. (1970). Energetic cost of locomotion in animals. Comp Biochem Physiol, 34(4), 841-846.

Uno, Y., Kawato, M., & Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. Minimum torque-change model. Biol Cybern, 61(2), 89-101.
Vaidyanathan, N., Penny, S., & Berniker, M. (2020). Planned Straight or Biased to Be So? The Influence of Visual Feedback on Reaching Movements. J Mot Behav, 52(2), 236-248. doi:10.1080/00222895.2019.1609409
Walden, R. H., "Analog-to-digital converter survey and analysis," IEEE Journal on selected areas in communications, vol. 17.4, pp. 539-550 (1999).
Wei, K., Yan, X., Kong, G., Yin, C., Zhang, F., Wang, Q., & Kording, K. P. (2014). Computer use changes generalization of movement learning. Curr Biol, 24(1), 82-85. doi:10.1016/j.cub.2013.11.012
Wolpert, D. M., Ghahramani, Z., & Jordan, M. I. (1994). Perceptual distortion contributes to the curvature of human reaching movements. Exp Brain Res, 98(1), 153-156.
Wolpert, D. M., Ghahramani, Z., & Jordan, M. I. (1995). Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study. Exp Brain Res, 103(3), 460-470.
Wolpert, D. M., & Miall, R. C. (1996). Forward Models for Physiological Motor Control. Neural Netw, 9(8), 1265-1279. doi:10.1016/s0893-6080(96)00035-4

APPENDIX

A. Mean Field Equations

By minimizing the KL divergence between a tractable distribution, Q, and the desired, intractable distribution, P, the cost function becomes:

$$L_v = \sum_{ij} J_{ij}\mu_i\mu_j + \sum_i h_i\mu_i - \sum_i \xi_i \left(\sum_j J_{ij_i}\mu_j + h_i \right) - \sum_i \ln \langle e_i^{-\xi_i z_i} + e_i^{(1-\xi_i)z_i} \rangle - \sum_i \mu_i \ln\mu_i + (1-\mu_i)\ln(1-\mu_i)$$

where J_{ij} is the connective weight from node j to node i and h_i is the bias on node i. Latent parameters μ and ξ are calculated for each training sample through concurrent updates of the mean field equation and solving for a minimum ξ through one dimensional search. The mean field equation can be found below:

$$\mu_i = \sigma(h_i + \sum_j J_{ij}\mu_j + J_{ji}(\mu_j - \xi_j) + K_{ji}$$

where

$$K_{ij} = -\frac{\partial}{\partial \mu_j} \langle e^{-\xi_i z_i} + e^{(1-\xi_i)z_i} \rangle$$

The following gradients were used in traditional backpropagation algorithms to find weights and biases that minimize the cost function:

$$\frac{\partial L_v}{\partial J_{ij}} = -(\xi_i - \mu_i)\mu_j + \frac{(1-\phi_i)\xi_i\mu_j e^{-\xi_i J_{ij}}}{1-\mu_j + \mu_j e^{-\xi_i J_{ij}}} - \frac{\phi_i(1-\xi_i)\mu_j e^{(1-\xi_i)J_{ij}}}{1-\mu_j + \mu_j e^{(1-\xi_i)J_{ij}}}$$

$$\frac{\partial L_v}{\partial h_i} = \mu_i - \phi_i$$

where

$$\phi_i = \frac{\langle e^{(1-\xi_i)z_i} \rangle}{\langle e^{-\xi_i z_i} + e^{(1-\xi_i)z_i} \rangle}$$

These equations, taken from Saul et al., 1996, were used exactly as written in the reference.

B. Gradient on Expected Error Output (From Berniker and Penny, 2018)

The cost function used in section 3.1 was the error in the expected output probability:

$$L_k = \frac{1}{2n}(\mu(\Delta X^k) - \bar{O}^k)^T(\mu(\Delta X^k) - \bar{O}^k)$$

The gradient for the cost function can be represented as follows

$$L = \sum_k^N L_k + \lambda \sum_{i,j,p,q} \left[(w_{ij}^1)^2 + (w_{pq}^2)^2\right]$$

$$\begin{aligned} L_k &= \frac{1}{2n}(\mu(\Delta X_k) - \bar{O}_k)^T(\mu(\Delta X_k) - \bar{O}_k) \\ &= \frac{1}{2n}\sum_l^n(\mu_l(\Delta X_k) - \bar{o}_l)^2 \end{aligned}$$

where

$$\begin{aligned} \bar{o}_l &= p(o_l = 1|\Delta X_k) \\ &= \sum_v^M p(o_l = 1|H^v)\prod_i^m p(h_i|\Delta X_k) \end{aligned}$$

then we find the following

$$\begin{aligned} \frac{\partial L_k}{\partial W_{pq}^1} = \frac{-1}{n}\sum_l^n[\mu_l(\Delta X_k) - \bar{o}_l]\Big(\sum_v^M p(o_l = 1|h^v)... \\ \prod_{j\neq p}^m p(h_j^v|x)dp(h_p^v|x)x_q\Big) \end{aligned}$$

$$\begin{aligned} \frac{\partial L_k}{\partial B_{pq}^1} = \frac{-1}{n}\sum_l^n[\mu_l(\Delta X_k) - \bar{o}_l]\Big(\sum_v^M p(o_l = 1|h^v)... \\ \prod_{j\neq p}^m p(h_j^v|x)dp(h_p^v|x)\Big) \end{aligned}$$

$$\frac{\partial L_k}{\partial W^2_{pq}} = \frac{-1}{n}[\mu_p(\Delta X_k) - \bar{o}_p)]\Big(\sum_v^M dp(o_p = 1|h^v)... \prod_j^m p(h^v_j|x)h^v_q\Big)$$

$$\frac{\partial L_k}{\partial B^2_{pq}} = \frac{-1}{n}[\mu_p(\Delta X_k) - \bar{o}_p)]\Big(\sum_v^M dp(o_p = 1|h^v)... \prod_j^m p(h^v_j|x)\Big)$$

where *p* and *dp* are the sigmoid function and its derivative, respectively.

C. Supplementary Figures

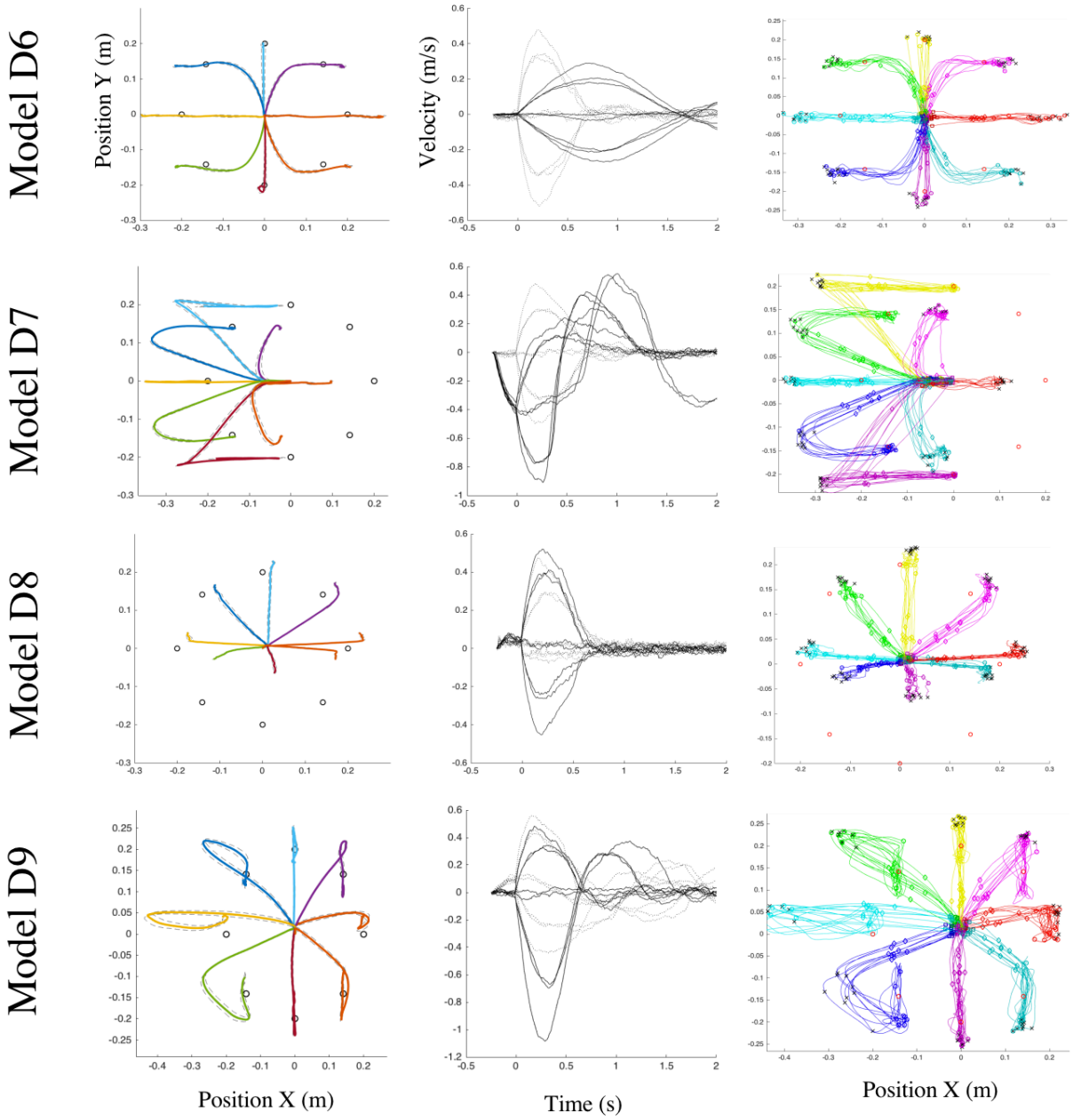

Figure F1) Similar to Figure 21, but with more unique trajectories. It should be noted that Model D8 has high-frequency tremor-like movements towards the body, but not away from the body. This will be explored in the future.

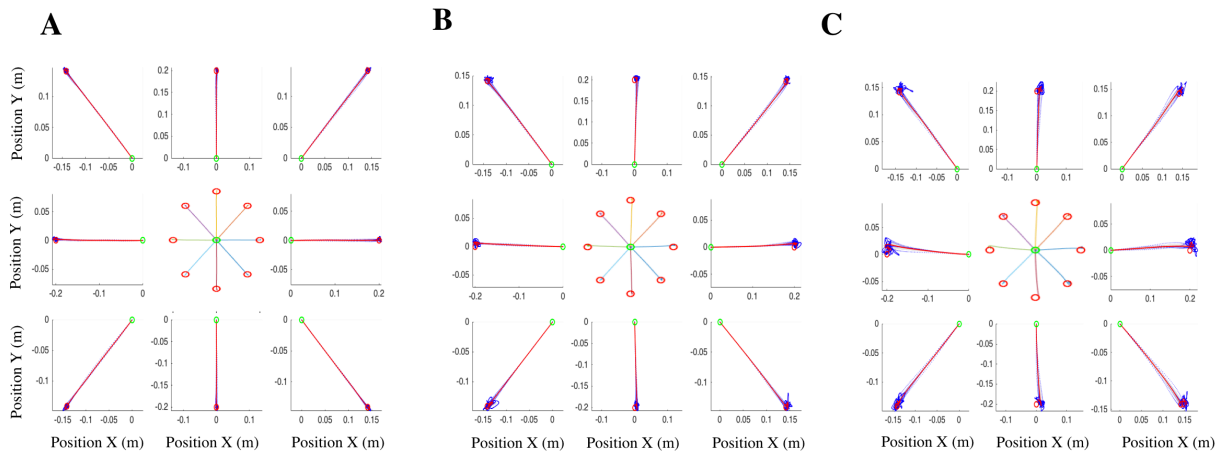

Figure F2) An analysis of the effect on observation noise on the system. To highlight the effect of noise on the state estimate, an LQR controller was used, not a network approximating LQR. Simulations are shown where the variance of the sensory observation noise was equal to that of the forward model **A)** 5 times the forward model **B)** and 10 times the forward model **C)**. As the observation noise increased the trajectories became more variant but still reached the target.

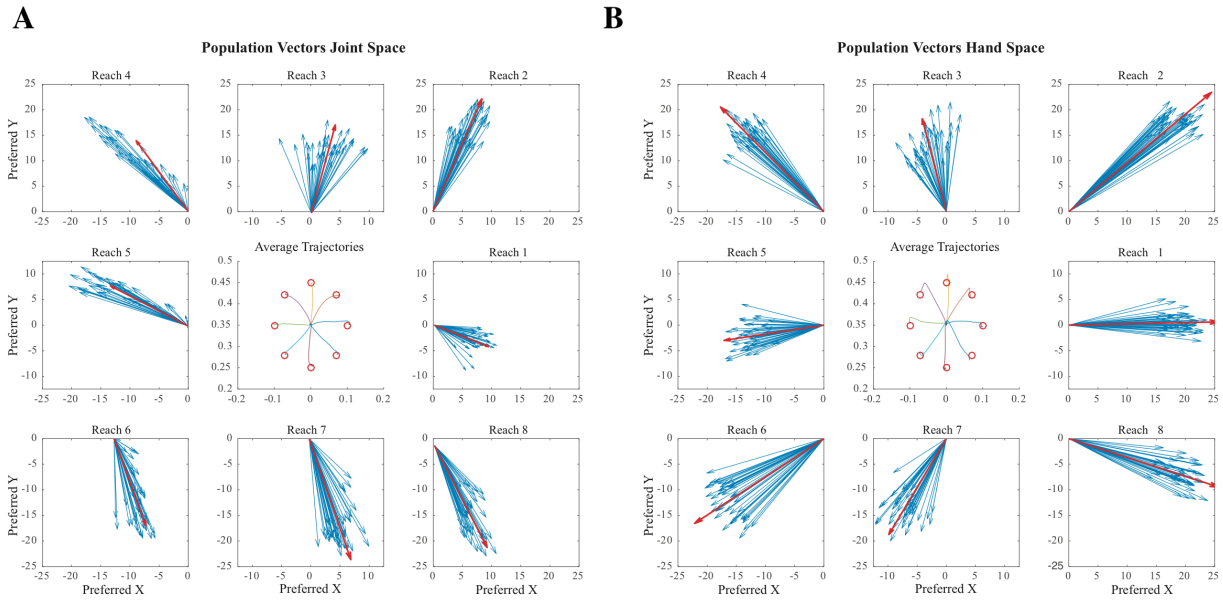

Figure F3) The population vectors for a spiking network trained to approximate an LQR controller for the dynamics of a two link arm. This was trained on two different inputs: **A)** intrinsic inputs (joint, or rotational, variables) and **B)** extrinsic inputs (hand, or translational, variables). It can be seen that distinct predictions were generated with the two different inputs.

D. Permissions

Biological Cybernetics (Publisher: Springer Nature):
https://www.springer.com/gp/rights-permissions/obtaining-permissions/882

Author reuse

Please check the Copyright Transfer Statement (CTS) or Licence to Publish (LTP) that you have signed with Springer Nature to find further information about the reuse of your content.

Authors have the right to reuse their article's Version of Record, in whole or in part, in their own thesis. Additionally, they may reproduce and make available their thesis, including Springer Nature content, as required by their awarding academic institution. Authors must properly cite the published article in their thesis according to current citation standards.
Material from: 'AUTHOR, TITLE, JOURNAL TITLE, published [YEAR], [publisher - as it appears on our copyright page]'

If you are any doubt about whether your intended re-use is covered, please contact journalpermissions@springernature.com for confirmation.

Journal of Motor Behavior (Publisher: Taylor & Francis)
https://authorservices.taylorandfrancis.com/copyright-and-you/

Copyright assignment

In our standard author contract, you transfer – or "assign" – copyright to us as the owner and publisher of the journal (or, in the case of a society-owned journal, to that learned society).

Assigning the copyright enables us to:

- Effectively manage, publish and make your work available to the academic community and beyond.
- Act as stewards of your work as it appears in the scholarly record.
- Handle reuse requests on your behalf.
- Take action when appropriate where your article has been infringed or plagiarized.
- Increase visibility of your work through third parties.

After assigning copyright, you will still retain the right to:

- Be credited as the author of the article.
- Make printed copies of your article to use for a lecture or class that you are leading on a non-commercial basis.
- Share your article using your free eprints with friends, colleagues and influential people you would like to read your work.
- Include your article Author's Original Manuscript (AOM) or Accepted Manuscript(AM), depending on the embargo period in your thesis or dissertation. The Version of Record cannot be used. For more information about manuscript versions and how you can use them, please see our guide to sharing your work.

VITA

NAME: Steven Penny

EDUCATION:

University of Illinois at Chicago: Ph.D in Mechanical Engineering
Chicago, IL (2016-2020)

University of Illinois at Chicago: M.S. in Bioengineering
Chicago, IL (2016-2019)

University of Notre Dame: B.S. in Chemical Engineering
South Bend, IN (2012-2016)

TEACHING EXPERIENCE

University of Illinois at Chicago: Teaching Assistant (2017-2019)
Introduction to Dynamics, Dynamic Systems and Controls, and Introduction to Vibration Theory

RESEARCH EXPERIENCE

University of Illinois at Chicago: Research Assistant
(2016-2020)

Research Fellow – University of Notre Dame Center for Nano Science and Technology
(2014)

Undergraduate Research Assistant – University of Notre Dame
(2013-2016)

WORK EXPERIENCE

Senior Data Scientist - CCC Information Services
(2020-present)

Data Science Intern - CCC Information Services
(2019-2020)

Device Development Intern – Regeneron Pharmaceuticals Inc.
(2015-2016

HONORS

Tau Beta Pi – Indiana Gamma Chapter – Elected in November 2015

Fr. Edward "Monk" Malloy Award – Awarded to an undergraduate Freshman who exhibits exceptional leadership.

Undergraduate Dean's List – Every semester between Spring 2013 - Spring 2016

PUBLICATIONS

KS, VK, ZD, Steven Penny, & GT, 'Synthetic Capillaries to Control Blood Flow.' *Nature: Scientific Reports*, 2016.

Max Berniker & Steven Penny, 'A normative approach to neuromotor control.' *Biological Cybernetics*, 2018.

NV, Steven Penny, & MB, 'The influence of visual feedback on reaching movements.' *Journal of Motor Behavior*, 2019.