UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

PREDICTING WRIST FORCES AND MUSCLE ELECTROMYOGRAMS FROM CORTICAL DATA

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

MICHAEL A. CRAIG
Norman, Oklahoma
2013

PREDICTING WRIST FORCES AND MUSCLE ELECTROMYOGRAMS FROM CORTICAL DATA

A THESIS APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY

Dr. Andrew H. Fagg

Dr. Amy McGovern

Dr. Lee Miller

Acknowledgments

This work was supported by funding from the National Institute of Neurological Disorder and Stroke (Bioengineering Research Partnership grant R01 NS048845). I would like to thank Dr. Andrew H. Fagg for his support, guidance, and mentoring that made this work possible. The data used in this thesis was provided by Dr. Lee E. Miller and Emily R. Oby at Northwestern University. Everyone in my committee is greatly appreciated: Dr. Andrew H. Fagg, Dr. Amy McGovern, and Dr. Lee E. Miller. I would like to thank everyone in the Symbiotic Computing Laboratory at the University of Oklahoma, including (but not limited to): Josh Southerland, Matt Bodenhamer, Tom Palmer, and Joohee Suh for all of their help and knowledge. My parents Daniel and Sheryl Craig, as well as my brother Josh and sister Chloe, have always been big supporters of me, and have always been an encouraging force in my undergraduate and graduate studies.

Table of Contents

List of Tables

List of Figures

Abstract

When producing movements to external stimuli, the nervous system must transform a representation of these external spatial stimulus to the muscle recruitment patterns that produce the appropriate movement. The recruitment patterns of the intervening neurons and the muscles are often described as a level of activity with respect to the extrinsic direction of movement and/or force production. The preferred direction (PD) of a muscle or neuron is the extrinsic direction corresponding to maximal activity, with recruitment activity dropping smoothly as the direction of action deviates from the preferred direction. Muscle preferred directions, among a range of factors, are affected by limb posture, due to the changes in the mechanics of muscle action with the posture changes. However, neurons in the primary motor cortex (MI), which play a role in the transformation process, exhibit a range of effects in the PD response to posture. Kakei et al. (1999) shows, in a center-out wrist movement task, that a subset of MI neurons show a PD shift as a function of wrist posture, while a substantial number of neurons do not exhibit a PD shift. Oby et al. (2012) shows in a related isometric wrist force tracking task, that these two types are drawn from a single, unimodal distribution. Nevertheless, in both studies, the average MI PD shift was substantially less than the PD shifts of the muscles used to perform the task.

I present in this thesis a recurrent neural network model capable of producing temporal muscle activation patterns in an isometric center-out wrist task. I show that by selecting model parameters that minimize force tracking error while also minimizing neural and muscle activity, the model produces both force trajectories and PD shift distributions quantitatively similar to those seen in the monkey experiments. Furthermore, when the constraints are relaxed, the model is unable to produce reliable PDs for neurons, and thus unable to produce PD shift distributions similar to monkey experiments. This implies that the constraints placed on the model are sufficient to make the transformation from extrinsic visual input to intrinsic muscle activations, and are sufficient to create representations similar to those seen in MI. The results

suggest that, although the neuron PDs are determined in part by the constraints of the input (extrinsic) and output (muscle) encodings, due to the large number of intervening neurons, the nervous system is able to make a range of choices for the tuning properties of individual neurons.

Chapter 1

Introduction

The Primary Motor Cortex (MI) region of the brain participates in the planning and execution of movements. These movements are often performed as a reaction to some external stimuli, such as visual stimuli. In order to complete a movement, the central nervous system must transform the stimuli into muscle recruitment patterns that are capable of performing the appropriate movement. These recruitment patterns must take into account both the goal position in extrinsic space and the current musculoskeletal state of the limb. This musculoskeletal state includes both the position and orientation of the limb.

In order to make the transformation from external stimulus to muscle recruitment patterns, neurons in MI must modulate with some function of movement. Georgopoulos et al. (1982) showed that neurons in MI modulate with both the form and timing of the movement direction. In the experiment, Georgopoulos et al. (1982) had monkey subjects perform a center-out reaching task. The monkey held its hand at a center location and observed a visual target that appeared along a circle around the center position. The monkey then moved its hand to the visual target position; the hand

trajectory along the way and the corresponding neural activity were recorded. Georgopoulos et al. (1982) modeled the neural activation patterns as a cosine function of movement direction and a neuron-specific preferred direction. The direction by which a cell is maximally activated according to this *tuning curve* is referred to as its *Preferred Direction* (PD). Georgopoulos et al. (1986) introduced the idea of *population codes*, in which the combined activity of a set of neurons determined the direction of movement. Here, each neuron "votes" for a particular preferred direction of movement to a degree that is proportional to its level of activity. The movement direction is then a weighted sum of these individual preferred directions.

While the Georgeopoulos perspective was presented in terms of directions within an extrinsic Cartesian coordinate frame, many neurons within MI directly influence muscle activity, and in fact one expects a range of factors to influence the activity of MI neurons. For example, in addition to neurons modulating with movement direction (Georgopoulos et al., 1986; Ajemian et al., 2001), they have also been shown to modulate with limb posture (Scott and Kalaska, 1997), direction of movement force (Kalaska et al., 1989), and direction of inertial loads (Sergio and Kalaska, 1998).

There is debate as to whether MI encodes movement in an extrinsic coordinate frame or an intrinsic muscle-like coordinate frame. Georgopoulos et al. (1982) argued that neurons modulate according to an extrinsic coordinate frame. Mussa-Ivaldi (1988) given experimental design, the distinction between extrinsic and intrinsic coordinate frames is unclear.

Kakei et al. (1999) examined neural modulation in a wrist-based center-out task

as the forearm was held in one of three postures. The monkey moved a manipulandum by controlling wrist flexion/extension and radial/ulnar deviations. The monkey was required to move from a center position to one of eight targets on a circle around the center position. Through this experiment, Kakei et al. (1999) quantified how the PDs of a neuron population changed as forearm configuration changed between the pronated, midrange, and supinated postures. The change in a neuron's PD from one pronated to the supinated posture defines the neuron's *PD shift*. Kakei observed a bimodality in the PD shifts across postures. One subpopulation exhibited PD shifts close to 0° and were therefore labeled as *extrinsic*. Another subpopulation exhibited PD shifts similar to those of pulling directions of the muscles used to complete the task, and were therefore referred to as *muscle-like*.

Kakei suggested that the central nervous system computes the transform from extrinsic stimuli to intrinsic muscle activation patterns through a *serial processing scheme* in which extrinsic visual information is transformed into intermediate or muscle-like representations before being sent to spinal motorneurons. The existence of both extrinsic and muscle-like populations of neurons led Kakei et al. (1999) to the conclusion that portions of this serial transformation might occur within MI, and thus the two populations of neurons were simply neurons at different stages of the transformation process.

Oby et al. (2012) produced an isometric wrist movement experiment in which monkeys applied forces to a manipulandum in order to reach one of eight targets on a circle around a central starting position. The starting position required no forces, and each of the eight targets required the monkey to produce forces, which controlled

a cursor on a screen indicating the monkey's position in the trial. Oby found a unimodal distribution of PD shifts between the pronated and midrange forearm postures. The PD shift mean of neurons was much closer to muscles than that of an extrinsic coordinate frame.

Shah et al. (2004) challenged this serial processing idea by showing that a linear transformation exists that can form muscle activations from extrinsic neurons that are gain modulated by posture. Because extrinsic neurons were shown to be capable of commanding muscles directly, a parallel processing scheme was proposed, in which some populations of MI neurons could directly control muscles, while others formed a more intermediate representation first (Figure 1.1). In the parallel processing scheme, neurons that encode each coordinate frame are capable of receiving input directly from external stimuli and capable of directly commanding muscles.

Given that a transformation must be made from visual stimuli and forearm posture to muscle activation patterns, one important question to ask is: what types of neural representations are necessary and/or sufficient to implement this transformation? In other words, can a model be formed using information from visual stimuli and forearm posture that can reconstruct the appropriate muscle activation patterns while exhibiting neural representations similar to those observed in the central nervous system? In this thesis, I present the hypothesis that given a recurrent neural network model that is a) structured to make the transformation from external visual stimuli to intrinsic wrist muscle activations and extrinsic hand force trajectories, and b) constrained to minimize muscle and neural activation levels, the neurons in the

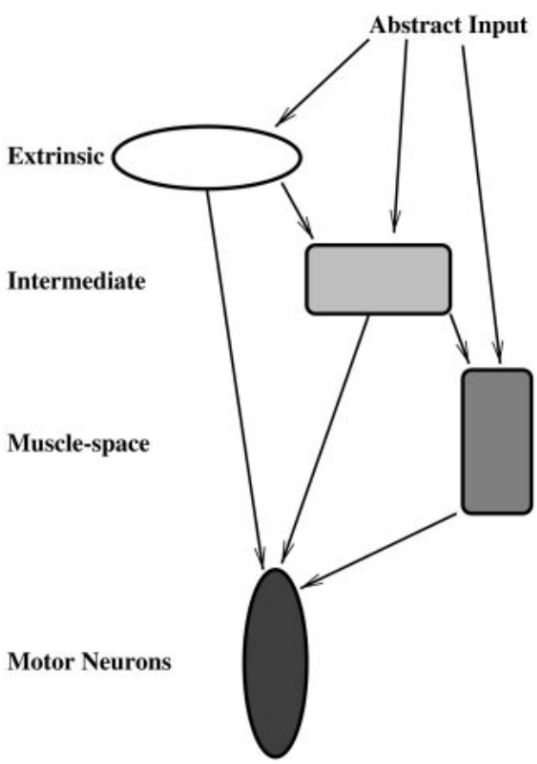

Figure 1.1: *The parallel processing scheme as proposed by Shah et al. (2004). Figure 1, used with permission.*

model will show modulation as a function of movement direction, and the distribution of PD shifts across postures will be similar to those seen in monkey experiments.

This thesis extends the model of Shah et al. (2004) by allowing the behavior of the neuron population to be learned rather than predetermined. This removes the constraint that all neurons behave extrinsically, and instead allows neural representations to be learned given the task and metabolic energy constraints. In order to accomplish the task, an input layer consisting of a population vector encoding extrinsic target direction is defined. The neuron population then becomes an adaptable hidden layer in the model. In order to capture temporal changes of a trajectory, recurrent connections among the neuron population are used. This enables the network to fully model

a trajectory of motion all the way from its origins as a visual cue to its neural representation, to the corresponding muscle activations, and finally to the extrinsic force trajectories. This thesis, I show that a model can be formed that is capable of producing force trajectories similar to those observed in the monkey experiments, and that the modeled neural representations are similar to those seen in observed neural data. Specifically, I show that the mean of the PD shifts exhibited by the model are within an extrinsic and muscle-like coordinate frame.

The following chapter introduces background information and related work. The model is discussed in detail in Chapter 3. This includes discussion of the structure of the model, as well as the equations that govern it. Results are reported in Chapter 4.

Chapter 2

Background

The central nervous system is responsible for transforming sensory information into muscle recruitment patterns in order to create goal-directed trajectories. How the central nervous system accomplishes this is a central problem in motor control research. One way to formulate this task is as a series of coordinate transformations between different representations.

Georgopoulos et al. (1982) recorded activity from cells in the primary motor cortex while a monkey subject performed a planar center-out reaching task to eight different directions spaced evenly around a central point. Analysis of the recorded neural data showed that the activity of individual neurons varied with extrinsic movement direction in a cosine-like fashion. Specifically, the average activity of a neuron followed a function of the form $y = a + b\ cos(c - \theta)$, where a, b, and c are regression coefficients, and θ is the direction of the target. The a parameter captures the baseline activity of a neuron and the b parameter captures the depth of modulation of a neuron. Georgopoulos also noted that a neuron is maximally active for a particular movement direction (denoted by c in the model), which he referred to as its

preferred direction (PD). From this model, Georgopoulos et al. (1986) was able to encode movement information by allowing each neuron to make some contribution towards the limb movement as a function of its preferred direction and its degree of activity. He formulated a *population code* in which each element in the vector is the difference between a neuron's activity at its preferred direction and the cell's current activity. Movement direction could then be computed as a function of a cell's preferred direction and its current activation.

MI neurons have also been shown to modulate with limb posture (Scott and Kalaska, 1997). This is important because it challenged Georgopoulos' assumption that neurons modulated only with respect to extrinsic movement direction. This posture-dependent modulation of neurons is evidence that MI does not encode in a purely extrinsic coordinate frame, and at least in part, can encode some intrinsic information.

Kakei et al. (1999) studied the effect of forearm posture on MI activity. Monkey subjects performed center-out wrist movements to one of eight targets spaced evenly around a center position, each 45° apart from each other. The monkey's wrist movements (flexion/extension and radial/ulnar deviation) were mapped to cursor positions on the screen, and the monkey was required to move the cursor position to the target position. The monkey performed this task for three distinct forearm postures. Figure 2.1 shows the three forearm postures used in this study. Performing the task in the three forearm postures allowed for the dissociation between movement direction with respect to wrist joints, extrinsic space, and muscle activity. Kakei observed that many MI cells were directionally tuned according to Georgopoulos' cosine tuning function.

Additionally, Kakei showed that as the forearm posture changed, the preferred direction of some of the neurons shifted. The distribution of preferred direction shifts between the pronated and supinated forearm postures as computed by Kakei et al. (1999) is shown in Figure 2.2. The distribution appeared to be bimodal, with one mode around 0 degrees, and another mode around 60-80 degrees, which is consistent with the changes in the PD of the recorded wrist muscles. This distribution was used to subdivide the neural population into sets of "extrinsic" neurons and "muscle-like" neurons. The boundaries that defined his subdivisions were determined through a qualitative analysis of the PD shift distribution. The interesting consequence of the experiment design is that it allows dissociation between three coordinate frames: one extrinsic coordinate frame representing movement direction, one intrinsic coordinate frame relating to muscles, and one intrinsic coordinate frame relating to joint angle (Kakei et al., 2003).

Kakei et al. (1999) interpreted the PD shift bimodality to be an indication that a serial processing approach was being undertaken in MI to transform a visual representation of movement to a muscle-based one. The PD shift bimodality could then be explained as two populations of neurons at different stages of the transformation process.

Another possible interpretation of the PD shift bimodality observed by Kakei et al. (1999) is that the primary motor cortex employs some form of parallel processing scheme that allows both extrinsic and intrinsic populations of neurons to control muscles directly (Shah et al., 2004; Dum and Strick, 2002). To show that the parallel processing scheme was viable, Shah et al. (2004) developed a neural network model

that activated muscles based on a population of purely extrinsic neurons. Neurons in the population did not exhibit substantial PD shifts between postures, but did exhibit depth of modulation changes between postures. Connections from muscles to wrist forces were determined from the muscle pulling directions, and were posture dependent. The connections from MI to muscles were learned.

This neural network model was able to recreate muscle activations qualitatively similar to those seen by Hoffman and Strick (1999), which involved a step-tracking center-out task identical to Kakei et al. (1999). The success of the network at producing muscle activations indicates that neurons that encode information in an extrinsic coordinate frame are able to directly control muscles. This challenged the previous assumption that cortical processing for movement is a serial process, and that instrinsically tuned neurons are necessary for controlling muscles. The results imply that a parallel processing scheme could be a plausible explanation in motor processing.

Oby et al. (2012) presented an isometric wrist task very similar to that of Kakei et al. (1999), for two different wrist postures: pronated and midrange. The change from pronated posture to the midrange posture equates to a 90° degree change in joint orientation. The PD shift distribution was found to be unimodal rather than bimodal as found by Kakei et al. (1999), with the mode falling around 50°. In particular, the mode falls in between an unchanging extrinsic coordinate frame and a coordinate frame similar to muscle activations. Furthermore, Oby et al. (2012) formed a linear model that could compute intrinsic muscle activations from cortical MI data regardless of wrist posture. This meant that given a posture-dependent model of muscle activations to hand forces, the model could accurately predict hand force from MI.

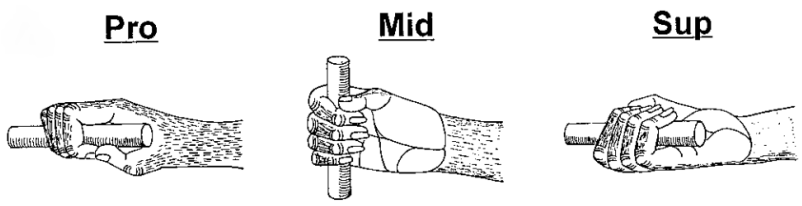

Figure 2.1: *The three wrist postures used in Kakei et al. (1999). Figure 1A, used with permission.*

Because of musculoskeletal changes with posture, a linear model could not be formed from muscle activations to extrinsic forces regardless of wrist posture, but separate posture-dependent models can. Further, a linear model could not be formed from cortical MI data directly to hand forces that is robust to changes in forearm posture. The ability to form a linear model of muscle activations regardless of posture, but not being able to form a linear model of hand forces regardless of posture implies that MI may more closely represent muscles than extrinsic forces.

2.1 Feed-forward Neural Networks

A neural network is a mathematical model inspired by the nervous system that allows for a network of connected units to communicate information between each other. Feed-forward neural models allow for function approximation of nonlinear functions. In fact, a neural network with one hidden layer and continuous outputs can theoretically approximate any continuous function, provided that the number of hidden units is sufficient (Bishop and Nasrabadi, 2006; Funahashi, 1998). The general structure of a three-layer feed-forward neural network is shown in Figure 2.3. The input units, denoted by x_i project to each hidden layer node, which in turn

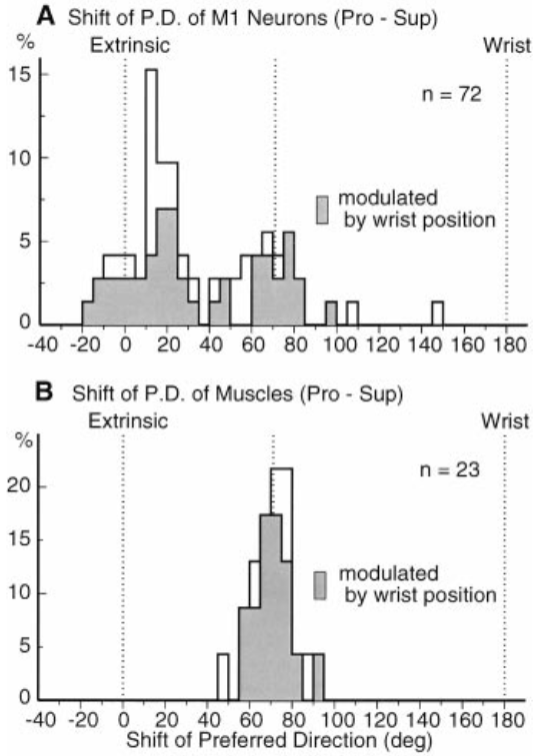

Figure 2.2: *Distribution of PD shifts of motor neurons going from Pronated to Supinated postures found by Kakei et al. (1999) (A). Distribution of PD shifts of muscle going from Pronated to Supinated postures (B). Figure 4, used with permission.*

projects onto the output units. A nonlinear function known as an *activation function* resides at each hidden unit. This activation function adds the nonlinearity that gives neural networks an advantage in terms of function approximation over simple linear models.

To produce an output from the network given an input vector $\mathbf{x}$, we can follow the network equations given by:

$$\mathbf{a} = \mathbf{W^{(1)^T}}\mathbf{x}, \tag{2.1}$$

$$\mathbf{z} = f(\mathbf{a}), \text{and} \tag{2.2}$$

$$\hat{\mathbf{y}} = g(\mathbf{W^{(2)^T}}\mathbf{z}), \tag{2.3}$$

where $\mathbf{W^{(1)}}$ is the weight matrix connecting the input units to the hidden units, $\mathbf{W^{(2)}}$ is the weight matrix connecting the hidden units to the output units, $f(\cdot)$ is a nonlinear activation function for the hidden layer, and $g(\cdot)$ is an activation function for the output layer. The activation function for the output layer need not be nonlinear, and, in fact, is often the identity function $g(x) = x$.

Backpropagation is a method often used for updating weights in a neural network (Le Cun et al., 1990). Backpropagation is accomplished by computing the error in the prediction from the observed target variable, and propagating the error backwards to the connection weights. The algorithm is implemented as a gradient descent algorithm. Thus, we define our error function to be:

$$E = \sum_{n=1}^{N} \frac{1}{2} \left(\mathbf{y}_n - \hat{\mathbf{y}}_n\right)^T \left(\mathbf{y}_n - \hat{\mathbf{y}}_n\right), \tag{2.4}$$

where $\mathbf{y}_n$ is the vector of observed values for sample n, $\hat{\mathbf{y}}_n$ is the predicted value computed through the forward propagation process given input n, and N is the number of samples. We then compute the gradient of the error with respect to the connection

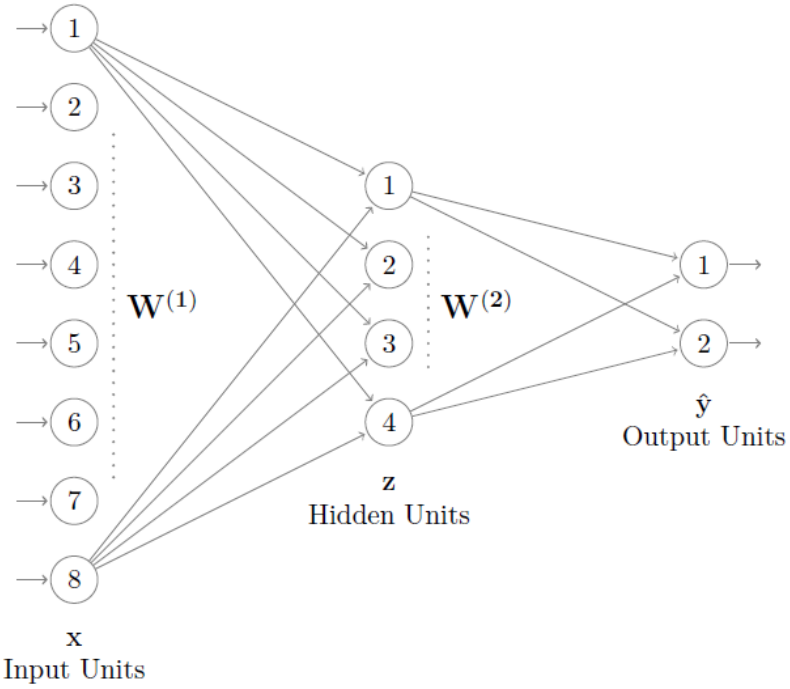

Figure 2.3: *The structure of a typical neural network. For clarity, 8 input units, 4 hidden units, and 2 output units are shown. The* $\mathbf{W^{(1)}}$ *and* $\mathbf{W^{(2)}}$ *matrices govern the behavior of the network. Not shown are the activation functions which occur at the hidden units.*

matrices. Given Equations 2.2 and 2.3, the gradients can be computed using the chain rule as such:

$$\begin{aligned}
\frac{\partial E_i}{\partial \mathbf{W^{(2)}}} &= \frac{\partial E_i}{\partial \hat{\mathbf{y}}_i}\frac{\partial \hat{\mathbf{y}}_i}{\partial \mathbf{W^{(2)}}} \\
&= -\mathbf{z}_i \left[(\mathbf{y}_i - \hat{\mathbf{y}}_i) g'(\hat{\mathbf{y}}_i)\right]^{\mathbf{T}}, \text{and} \qquad (2.5) \\
\frac{\partial E_i}{\partial \mathbf{W^{(1)}}} &= \frac{\partial E}{\partial \hat{\mathbf{y}}_i}\frac{\partial \hat{\mathbf{y}}_i}{\partial \mathbf{z}}\frac{\partial \mathbf{z}}{\partial \mathbf{a}}\frac{\partial \mathbf{a}}{\partial \mathbf{W^{(1)}}} \\
&= -\mathbf{x}_i \left[(\mathbf{y}_i - \hat{\mathbf{y}}_i) g'(\hat{y})\right]^{\mathbf{T}} \mathbf{W^{(2)^T}} \mathbf{F}, \qquad (2.6)
\end{aligned}$$

where $\mathbf{F}$ is a diagonal matrix where the diagonal entries form the vector $f'(\mathbf{W^{(1)^T}x})$.

The weights are then updated according to the following rule:

$$\mathbf{W}^{(\mathbf{1})}_{\mathbf{new}} \leftarrow \mathbf{W}^{(\mathbf{1})} - \eta \frac{\partial E}{\partial \mathbf{W}^{(\mathbf{1})}}, \text{ and}$$
$$\mathbf{W}^{(\mathbf{2})}_{\mathbf{new}} \leftarrow \mathbf{W}^{(\mathbf{2})} - \eta \frac{\partial E}{\partial \mathbf{W}^{(\mathbf{2})}},$$

where η is a learning rate parameter which controls how quickly and coarsely learning occurs.

2.2 Recurrent Neural Networks and Elman Networks

Recurrent neural networks are a special kind of neural network that are capable of modeling dynamical systems, that is, systems in which state is dependent on previous states as well as the immediate inputs (Elman, 1990). This ordering is often considered to be temporal in nature, and therefore can be used to model a temporally-dependent state.

A neural network can be made recurrent by adding cycles into the graph. An internal unit receives information from the external network inputs, as well as from other units within the network. From this, it is clear that the network is able to use information from the activations from states in the past in order to compute the desired function.

One possible implementation of a recurrent neural network is the *Elman network* (Elman, 1990), which assumes discrete time. The general structure of the Elman network is shown in Figure 2.4. The temporal dependence is implemented

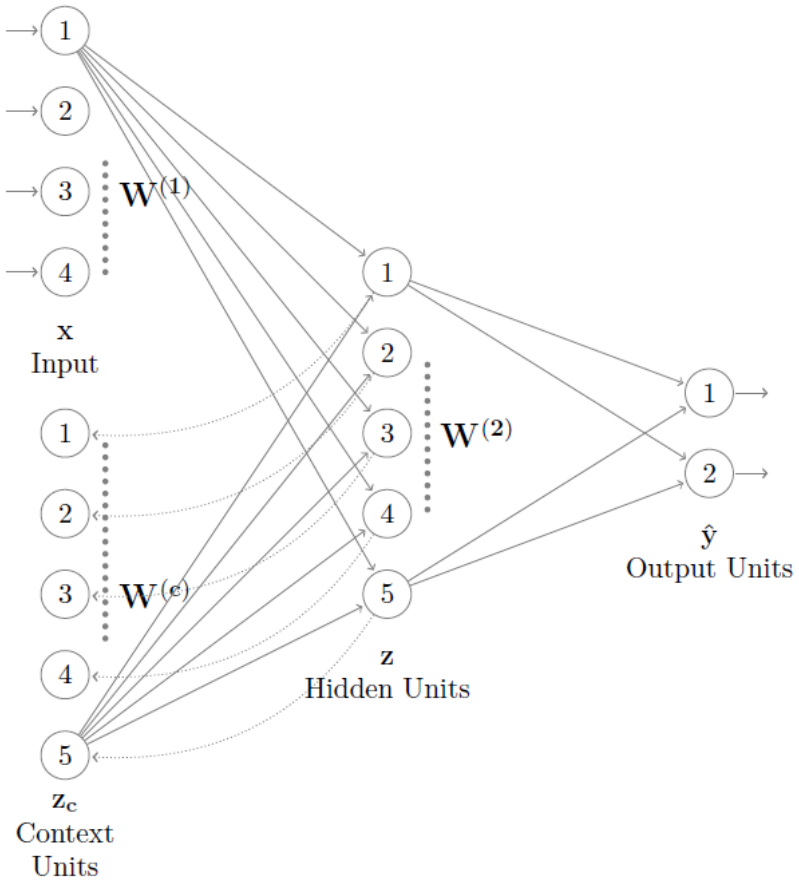

Figure 2.4: *The structure of a typical Elman network. For clarity, this network contains 4 input units, 5 hidden units, and 2 output units are shown. The* $\mathbf{W^{(1)}}$, $\mathbf{W^{(2)}}$, *and* $\mathbf{W^{(c)}}$ *matrices govern the behavior of the network. Not shown are the activation functions that are used at the hidden units.*

through the use of *context nodes*, which hold the state of the hidden layer from the previous time step. The network equations are as follows:

$$\mathbf{a}(t) = \mathbf{W^{(1)^T}}\mathbf{x}(t) + \mathbf{W^{(c)^T}}\mathbf{z^c}(t), \tag{2.7}$$

$$\mathbf{z}(t) = f(\mathbf{a}(t)), \tag{2.8}$$

$$\hat{\mathbf{y}}(t) = \mathbf{W^{(2)^T}}\mathbf{z}(t), \text{and} \tag{2.9}$$

$$\mathbf{z^{(c)}} = \mathbf{z}(t-1). \tag{2.10}$$

Note the difference between Equation 2.1 and Equation 2.7. The Elman network makes use of the context nodes $\mathbf{z^c}(t)$ when computing the $\mathbf{a}$ vector. A new weight matrix is introduced, $\mathbf{W^{(c)}}$. This matrix determines the strength of the recurrent connections. The other striking difference between recurrent neural network calculations and traditional neural network calculations is that time is now incorporated. Forward propagating an input is now dependent on the input at the previous time. At time $t = 0$, the context node activations are often set to 0.5 (Pham and Liu, 1996).

2.2.1 Training Elman Networks

Training Elman networks is similar to training traditional neural networks, and involves errors being backpropagated through the network and weights being updated in a gradient descent manner. Because of the recurrent connections, however, a different algorithm is used, known as *dynamic backpropagation*, where a dynamic trace of the error gradient with respect to the recurrent connections is kept (Pham and Liu, 1996). First, we must define our error function:

$$E = \sum_{t=1}^{T} E(t), \text{ and}$$

$$E(t) = \frac{1}{2} \left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right)^{\mathbf{T}} \left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right), \tag{2.11}$$

where $\hat{\mathbf{y}}(t)$ is the predicted output at time t and $\mathbf{y}(t)$ is the target output at time t. To adapt the weights in the network, the derivative of the error with respect to each of the weight matrices must be calculated. For the weight matrix $\mathbf{W^{(2)}}$, we have:

$$\frac{\partial E(t)}{\partial \mathbf{W^{(2)}}} = -(\mathbf{y}(t) - \hat{\mathbf{y}}(t)) \frac{\partial \mathbf{y}(t)}{\partial \mathbf{W^{(2)}}}$$
$$= -\mathbf{z}(t) \left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right)^{\mathbf{T}}.$$

Using the chain rule, we can derive $\frac{\partial E(t)}{\partial \mathbf{W^{(1)}}}$ as:

$$\frac{\partial E(t)}{\partial \mathbf{W}^{(1)}} = -\frac{\partial E(t)}{\partial \mathbf{y}(t)} \frac{\partial \mathbf{y}(t)}{\partial \mathbf{z}(t)} \frac{\partial \mathbf{z}(t)}{\partial \mathbf{a}(t)} \frac{\partial \mathbf{a}(t)}{\partial \mathbf{W}^{(1)}} \tag{2.12}$$
$$= -\mathbf{x} \left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right)^{\mathbf{T}} \mathbf{W}^{\mathbf{(2)^T}} \mathbf{F} \tag{2.13}$$

where $\mathbf{F}$ is a diagonal matrix where the diagonal entries form the vector $f'(\mathbf{z}(t))$. Finally, we can compute the gradient of the context unit weights $\frac{\partial E(t)}{\partial \mathbf{W^{(c)}}}$ as:

$$\frac{\partial E(t)}{\partial \mathbf{W}^{(c)}} = -\frac{\partial E(t)}{\partial \mathbf{y}(t)} \frac{\partial \mathbf{y}(t)}{\partial \mathbf{z}(t)} \frac{\partial \mathbf{z}(t)}{\partial \mathbf{W^{(c)}}} \tag{2.14}$$
$$= -\mathbf{W^{(2)}} \left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right) \frac{\partial \mathbf{z}(t)}{\partial \mathbf{W^{(c)}}}, \text{and} \tag{2.15}$$
$$\frac{\partial \mathbf{z}(t)}{\partial \mathbf{W^{(c)}}} = \mathbf{z^c}(t) \frac{\partial \mathbf{z_c}(t)}{\partial \mathbf{W^{(c)}}}. \tag{2.16}$$

Note that $\mathbf{z^c}(t) = \mathbf{z}(t-1)$ is a function of $\mathbf{W^{(c)}}$, so by using the product rule on Equation 2.7, $\frac{\partial \mathbf{z}(t)}{\partial \mathbf{W^{(c)}}}$ can be expanded as:

$$\frac{\partial \mathbf{z}(t)}{\partial \mathbf{W^{(c)}}} = f'(\mathbf{a}(t)) \left[\mathbf{x}(t-1) + \mathbf{W^{(c)}} \frac{\partial \mathbf{z}(t-1)}{\partial \mathbf{W^{(c)}}}\right], \text{where:} \tag{2.17}$$

$$\frac{\partial \mathbf{z}(t-1)}{\partial \mathbf{W}^{(\mathbf{c})}} = \mathbf{W}^{(\mathbf{c})^{\mathbf{T}}} f'(\mathbf{a}(t-2)) \frac{\partial \mathbf{z}(t-2)}{\partial \mathbf{W}^{(\mathbf{c})}}. \tag{2.18}$$

Using this algorithm, information about previous states can be used in the computation of later states. From a practical standpoint, a problem that arises is the *vanishing gradient problem*, in which information at times far into the past are diminished, because these gradients can become quite small. While this may pose problems for situations in which events far back in history can play an important role at the current time, for the purposes of trajectory recreations, this problem does not pose a threat.

2.3 Isometric Wrist Experiment

This thesis is based on data from a male Rhesus macaque monkey performing an isometric wrist task (Oby et al., 2012). The monkey performed a center-out wrist task by applying forces to a manipulandum. The forces were recorded by a torque sensor that recorded forces in an extrinsic space. The monkey performed the center-out task in two wrist postures, pronated and midrange (Figure 2.5).

By applying forces, the monkey was able to move a cursor on a screen visible to the monkey. The task was to move the cursor from a central position to one of eight visual targets on the screen, where each target was distanced equally from the center and separated from one another by 45°. A visual cue indicating the goal position was shown to the monkey at the beginning of the trial. The monkey was required to stay

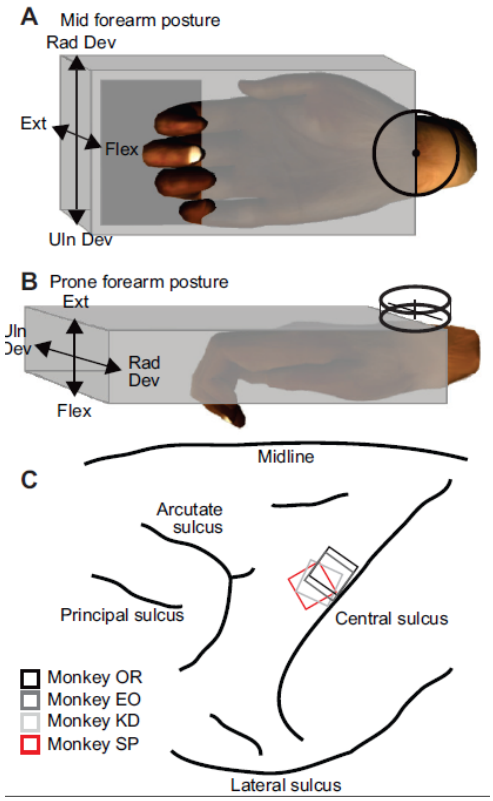

Figure 2.5: *Hand set up and recording array locations for the Oby et al. (2012) experiment. A torque sensor was mounted on top of the box at the point of rotation. The monkey's hand was placed in the box that gave the ability to flex fingers if necessary. The box was rotated to account for the midrange and pronated postures. C, the locations within MI where the recording arrays were placed. Figure 1, used with permission.*

at the center position for 0.5 seconds after the visual cue was presented, then was required to produce forces in order to move to the target position and hold at that for 0.5 seconds. A trial was considered successful only if the monkey was able to complete the task within a five second time limit. Successful trials were rewarded with juice. The monkey completed multiple trials for each target direction in both the midrange and pronated postures.

The monkey was trained on the task for two months prior to data collection. After training, a 10x10 microelectrode array was implanted into the primary motor cortex.

Locations of the implant are shown in Figure 2.5C. The electrode array was placed in the primary motor cortex on the side contralateral to the wrist that was used to complete the task.

In addition to cortical implants, electrodes were implanted into the muscles used to actuate the wrist. These included finger muscles flexor digitorum profundus (FDP), extensor digitorum communis (EDC), and flexor digitorum superficialis (FDS), as well as wrist muscles flexor carpi radialis (FCR), extensor carpi ulnaris (ECU), flexor carpi ulnaris (FCU), and extensor carpi radialis (ECR).

The details of the surgery and the instruments used to record both the neural and muscle signals are discussed in (Oby et al., 2012). The muscle signals were amplified and sampled at 2000Hz. These signals were then low-pass filtered and subsampled at 20Hz. Muscle activations and wrist forces were sampled at 20Hz, and neural spike counts were binned at 20Hz, yielding a count of spikes within each bin. The resulting dataset consists of a time series of a population of neurons, a time series of a population of wrist muscle EMG signals, and a time series of wrist forces, with each time step representing 50 milliseconds of a recorded session.

Chapter 3

Elman Network Model

Shah et al. (2004) showed that modeled posture-modulated extrinsic neurons can be used to directly drive muscle activation level. This model did not account for the time-dependent variation of neural behavior and muscle activations during movement and did not explain why a range of PD shifts is observed. The goal of this modeling effort is to address these issues.

In order to produce extrinsic force trajectories, the model presented in this thesis must be capable of driving muscles in a time-dependent manner given a purely extrinsic representation of an isometric goal and the wrist posture. My aim is to solve for a set of recurrent network parameters that implement this time-dependent transformation. Given successful solutions, we can then examine the response properties of individual neurons as they participate in this task.

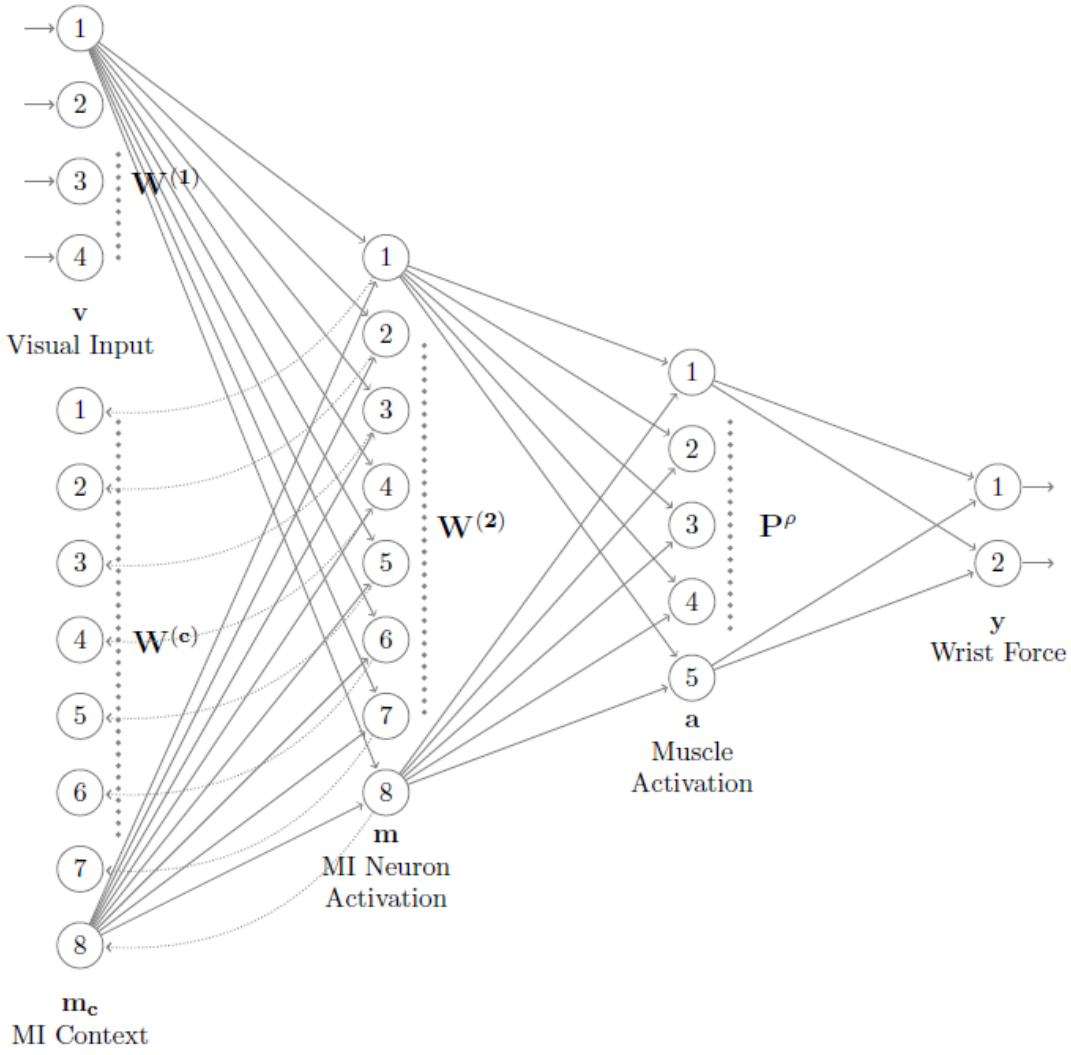

Figure 3.1: *The structure of the network used in this thesis. Only four inputs, eight MI neurons, and five muscles are shown for illustration purposes. The visual input vector* $\mathbf{v}$ *serves as the input to the network. The* $\mathbf{m}$ *vector serves to model the population of MI neurons, and the* $\mathbf{a}$ *vector models the activations of the wrist muscles. The MI units have a recurrent connection, which is indicated by the dotted lines, to the context units, which are then projected back to the MI units as input during the next timestep. Each unit is fully connected to every other unit in the next layer. The vertical dots represent connections not shown for clarity. The* $\mathbf{W^{(1)}}$, $\mathbf{W^{(c)}}$, *and* $\mathbf{W^{(2)}}$ *matrices are all adaptable, whereas the* $\mathbf{P}^{\rho}$ *matrix is not.*

3.1 Model Overview

The architecture of the proposed model is shown in Figure 3.1. This model closely follows the architecture of the Elman network shown in Chapter 2, with an additional layer that computes the transformation of muscle activations to extrinsic force space. As with the traditional Elman network, the recurrence is formulated as a pseudo-feedforward architecture through the use of the context units $\mathbf{m_c}$. The $\mathbf{W^{(c)}}$ matrix governs the behavior of the hidden units with respect to the context units. The $\mathbf{W^{(1)}}$, $\mathbf{W^{(c)}}$, and $\mathbf{W^{(2)}}$ connection matrices are all adaptable, but the $\mathbf{P}^{\rho}$ matrix is pre-computed and not adaptable.

3.1.1 A Population Code for Movement Goal

The input vector $\mathbf{v}$ to the network is defined by a simulated population vector (Georgopoulos et al., 1986) of visual neurons. More specifically, the input vector $\mathbf{v}$ is defined as:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \\ \rho \end{bmatrix},$$

where $v_1, v_2, ..., v_n$ are the activations of visual cells $1, ..., n$ respectively at the given target position, and ρ is the wrist posture. Specifically, ρ is a two element vector defined as:

$$\rho = \begin{cases} \begin{bmatrix} 1 \\ \\ 0 \end{bmatrix} & \text{for pronated posture, and} \\ \\ \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} & \text{for midrange posture.} \end{cases} \tag{3.1}$$

The visual input population vector simulates n cells with preferred directions evenly distributed around the circle (from $-\pi$ to $+\pi$). They are simulated as Gaussians of the form:

$$v_i = e^{\left(\frac{PD_i - \theta}{\sigma}\right)^2},$$

where PD_i is the preferred direction of visual cell i, θ is the target direction of the trial, and σ is a constant defined here to be one.

Since the input specification only accounts for target direction and wrist posture, there are only 16 different tasks. Further, because the aim is to create a model capable of recreating trajectories in all of these directions and orientations, both the training and test sets must be comprised of all 16 possible inputs. Thus, there is no clear distinction between a training and test set, and, in fact, no distinction between any

two trials in the same direction and wrist orientation.

To alleviate this problem, *signal dependent noise* is introduced to the visual input vector. Signal dependent noise has been shown to be present in saccadic eye movements and goal-directed arm movements (Harris and Wolpert, 1998). The idea is to add a normally distributed noise value whose variance changes with the magnitude of the signal. Thus, a signal s_i can add signal dependent noise as follows:

$$s_i^{new} = s_i + \epsilon,$$

where ϵ is a normally distributed value with a mean of 0 and a standard deviation of the form αs_i. The amount of noise in the signal is governed by α. The effects of this signal dependent noise for various choices of α are shown in Figure 3.2. Training with input vectors subject to signal-dependent noise allows for the model to become robust to variation in the input representation.

3.2 Model Dynamics

The network dynamics of the network at a time t are as follows:

$$\mathbf{z}(t) = \mathbf{W^{(1)^T}}\mathbf{v} + \mathbf{W^{(c)^T}}\mathbf{m_c}(t), \text{ and} \tag{3.2}$$

$$\mathbf{m}(t) = f(\mathbf{z}(t)), \tag{3.3}$$

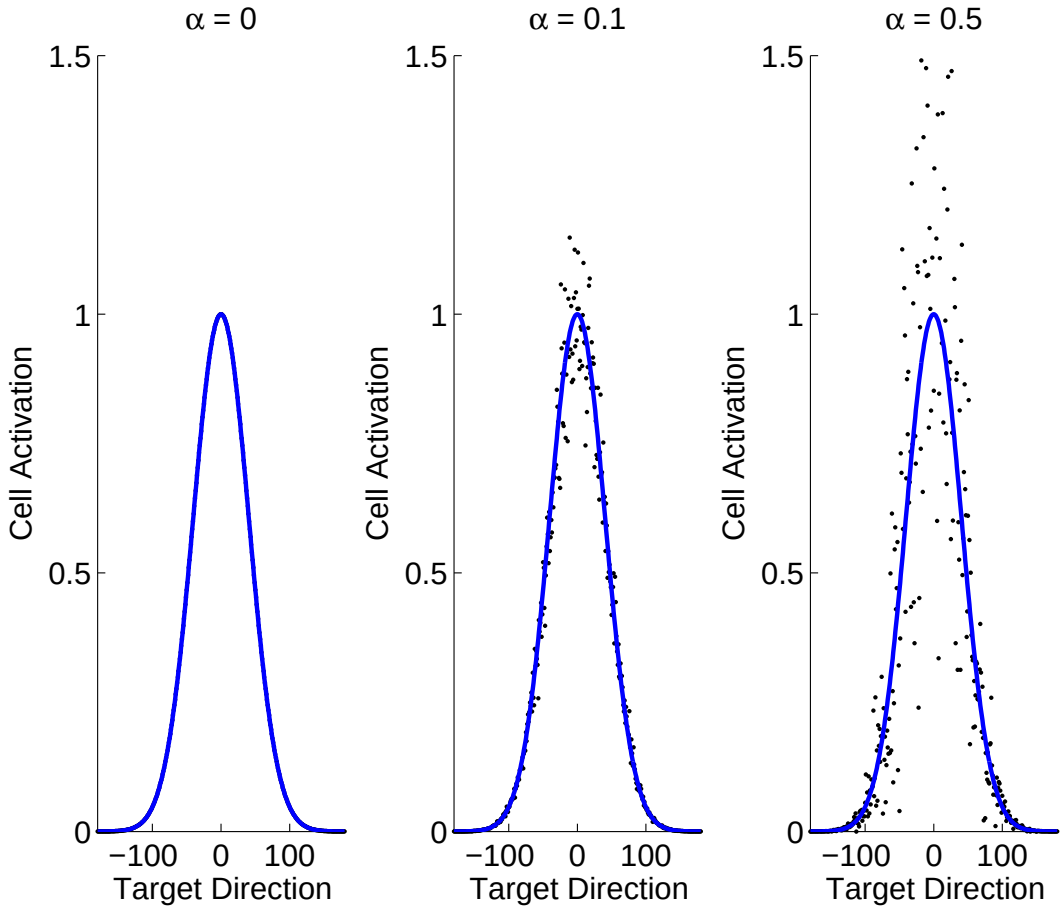

Figure 3.2: *Visual cell activations for noise parameter* $\alpha = 0, 0.1,$ *and* 0.5. *The black circles represent the activations of a sample visual cell with preferred direction at* $\theta = 0°$ *for target directions sampled on the range* $[-180°, 180°]$. *The blue line represents the activation of the neuron with no noise added.*

where the $\mathbf{z}(t)$ vector represents the weighted sum of inputs to the MI units. This becomes the input of the activation function $f(\cdot)$. It should be noted that this function computes some transform for each element in the vector $\mathbf{z}(t)$ independently. It is shown here as a function of the vector $\mathbf{z}(t)$ for simplicity.

The activation function must be chosen that models the expected range of values of the units it transforms. MI activation levels need to be restricted to non-negative

values, because spike rates are non negative. Because multiple linear transformations is itself a linear transformation, defining our activation function to be linear ($f(x) = ax + b$ for some a and b) would remove the usefulness of the hidden layer and would be equivalent to a linear model without the hidden layer. Therefore, some non-linearity in the activation function is desired. In order to prevent the recurrent connections from resulting in a run-away growth in activation level, the activation function must be sub-linear over most of its positive range. Finally, because the derivative of the activation function is needed in the computation of the gradients, the activation function needs to be differentiable over the range of acceptable inputs.

With these constraints in mind, the activation function used at the MI layer of this model is given by:

$$f(x) = ln\left(1 + \frac{e^{\kappa x}}{\kappa}\right)^{\beta}, \tag{3.4}$$

where $0 < \beta < 1$ and $\kappa > 0$. The behavior of this function is shown in Figure 3.3. Note that:

$$\lim_{x \to +\infty} f(x) = x^{\beta}, \text{and} \tag{3.5}$$

$$\lim_{x \to -\infty} f(x) = 0. \tag{3.6}$$

For the purposes of this model, $\kappa = 5$ and $\beta = 0.5$. In general, κ determines how "sharply" the activation function tends to x^{β} for $x > 0$, and β determines how slowly $f(x)$ increases for $x > 0$, relative to $f(x) = x$.

Note that Equation 3.4 is both differentiable and always non-negative. Thus,

these two constraints are satisfied. The third constraint is used to ensure that recurrent connections do not achieve large values throughout the trial. Note that the function presented accomplishes this by requiring inputs to be larger than outputs, for $x \gg 0$. To illustrate this, consider $\beta = 1$ and $x \gg 0$. From Equation 3.5, it is clear that

$$f(x+\gamma) \approx (x+\gamma)^1 = f(x) + \gamma.$$

However, if $0 < \beta < 1$, then for $x \gg 0$

$$f(x+\gamma) \approx (x+\gamma)^\beta < f(x) + \gamma.$$

Thus, to achieve large activation levels, even larger inputs must be provided. This discourages large activations unless they are needed.

The remaining network dynamics equations are:

$$\mathbf{m_c}(t) = \begin{cases} \mathbf{m}(t-1) & \text{for } t > 0, \\ \mathbf{d} & \text{for } t = 0, \end{cases} \tag{3.7}$$

$$\mathbf{a}(t) = \mathbf{W^{(2)^T}}\mathbf{m}(t) \text{ and} \tag{3.8}$$

$$\hat{\mathbf{y}}(t) = \mathbf{P}^{\rho^\mathbf{T}}\mathbf{a}(t), \tag{3.9}$$

where $\mathbf{d}$ is a vector in which all elements $d_j = 0.5$. The context units are defined as the MI activations at the previous time. Note that the node activations are all dependent on time t, however the connection matrices are not. This is a design choice of the network to allow batch updating in the backpropagation algorithm.

$\mathbf{P}^\rho$ is defined as the ridge regression solution from muscle activations in wrist

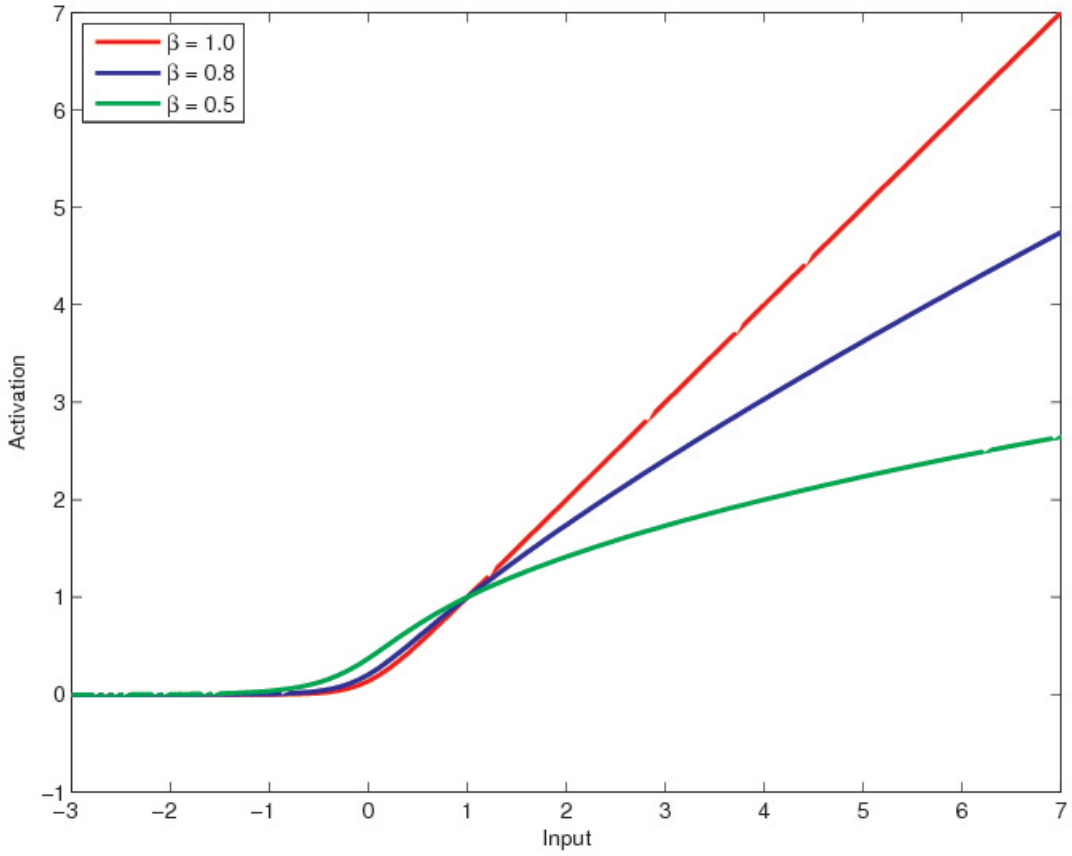

Figure 3.3: *The activation function used in this architecture for various choices of* β.

posture ρ to wrist forces in forearm posture ρ. These muscle activations and wrist forces are pulled from the data of Oby et al. (2012). Muscle activations $\mathbf{M}$ are transformed into hand force $\mathbf{F}$ by:

$$\mathbf{F} = \mathbf{P}^{\rho}\mathbf{M}.$$

Ridge regression is a form of linear regression in which regularization is included. $\mathbf{P}^{\rho}$ is the solution to this linear regression formulation and is of the form:

$$\mathbf{P}^{\rho} = \mathbf{F}\mathbf{M}^{\mathbf{T}}\left(\mathbf{M}\mathbf{M}^{\mathbf{T}} + \mathbf{\Lambda}\mathbf{\Lambda}^{\mathbf{T}}\right)^{-1}, \tag{3.10}$$

where $\mathbf{M}$ is a column sampled matrix of muscle activations, $\mathbf{F}$ is a column sampled matrix of hand force values, and $\mathbf{\Lambda}$ is the identity matrix scaled to some scalar λ. The addition of the $\mathbf{\Lambda}$ matrix allows for regularization in the solution.

3.3 Model Training

The training algorithm is derived from that used for typical Elman networks. The typical squared error is given by Equation 2.11. In order to avoid solutions with overly large neuron or muscle activations, I add two cost terms to Equation (2.11). These costs serve to prefer solutions with minimal activations, and will be known as the *metabolic minimization constraint* and the *muscle minimization constraint*, respectively (Fagg et al., 2002). Thus, the error function at time t becomes:

$$E(t) = \frac{1}{2}\left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right)^{\mathbf{T}}\left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right) + \frac{\lambda_1}{2}\mathbf{m}(t)^{\mathbf{T}}\mathbf{m}(t) + \frac{\lambda_2}{2}\mathbf{a}(t)^{\mathbf{T}}\mathbf{a}(t), \quad (3.11)$$

where λ_1 is a regularization parameter determining the relative cost of neural activation, and λ_2 is a regularization parameter determining the relative cost of muscle activation. These are manually tuned values, and their tuning will be discussed in Chapter 4.

3.3.1 Error Gradients

The gradient calculations generally follow the form outlined in Chapter 2 for Elman networks. The additional costs add to the complexity, however. The transformation from muscle activations, $\mathbf{a}$, to wrist forces, $\mathbf{y}$, also changes the gradient calculations. Thus, $\frac{\partial E(t)}{\partial \mathbf{a}(t)}$ can be written as

$$\begin{aligned}\frac{\partial E(t)}{\partial \mathbf{a}(t)} &= \frac{\partial}{\partial \mathbf{a}(t)}\left(\frac{1}{2}\left(\mathbf{y}(t)-\mathbf{P}^{\rho^{\mathbf{T}}}\mathbf{a}(t)\right)^{T}\left(\mathbf{y}(t)-\mathbf{P}^{\rho^{\mathbf{T}}}\mathbf{a}(t)\right)\right) \\ &+ \frac{\partial}{\partial \mathbf{a}(t)}\left(\frac{\lambda_1}{2}\mathbf{m}(t)^{T}\mathbf{m}(t)\right)+\frac{\partial}{\partial \mathbf{a}(t)}\left(\frac{\lambda_2}{2}\mathbf{a}(t)^{T}\mathbf{a}(t)\right) \\ &= -\mathbf{P}^{\rho}(\mathbf{y}(t)-\hat{\mathbf{y}}(t))+\lambda_2\mathbf{a}(t).\end{aligned} \tag{3.12}$$

Since the metabolic cost term is not a function of $\mathbf{a}$, the gradient of that term goes to zero. In order to compute the error gradient with respect to the metabolic cost, we must compute the direct gradient of $\frac{\partial E(t)}{\partial \mathbf{m}(t)}$, which is given by:

$$\begin{aligned}\frac{\partial E(t)}{\partial \mathbf{m}(t)} &= \frac{\partial}{\partial \mathbf{m}(t)}\left[\frac{1}{2}\left(\mathbf{y}(t)-\mathbf{P}^{\rho^{\mathbf{T}}}\mathbf{W}^{(\mathbf{2})^{\mathbf{T}}}\mathbf{m}(t)\right)^{T}\left(\mathbf{y}(t)-\mathbf{P}^{\rho^{\mathbf{T}}}\mathbf{W}^{(\mathbf{2})^{\mathbf{T}}}\mathbf{m}(t)\right)\right] \\ &+ \frac{\partial}{\partial \mathbf{m}(t)}\left[\frac{\lambda_1}{2}\left(\mathbf{m}(t)^{T}\mathbf{m}(t)\right)+\frac{\lambda_2}{2}\left(\mathbf{a}(t)^{\mathbf{T}}\mathbf{a}(t)\right)\right].\end{aligned}$$

Substituting equations 4.3 and 4.4:

$$\frac{\partial E(t)}{\partial \mathbf{m}(t)} = -\mathbf{W}^{(\mathbf{2})}\mathbf{P}^{\rho}(\mathbf{y}(t)-\hat{\mathbf{y}}(t))+\lambda_1\mathbf{m}(t)+\lambda_2\mathbf{W}^{(\mathbf{2})}\mathbf{a}(t).$$

Note that $\mathbf{m}$ is a function of both costs, since its activation affects the computation of both the $\mathbf{m}$ and $\mathbf{a}$ vectors.

With these gradients, the error gradients with respect to the connection parameters are a straightforward application of the chain rule. Note that since $\mathbf{P}^{\rho}$ is unchanging, there is no need to compute the erorr gradient with respect to it. The error gradient with respect to the other connection matrices are given by:

$$\begin{aligned}
\frac{\partial E(t)}{\partial \mathbf{W^{(2)}}} &= \frac{\partial E(t)}{\partial \mathbf{a}(t)} \frac{\partial \mathbf{a}(t)}{\partial \mathbf{W^{(2)}}} \\
&= \frac{\partial E(t)}{\partial \mathbf{a}(t)} \mathbf{m}(t), \\
\frac{\partial E(t)}{\partial \mathbf{W^{(1)}}} &= \frac{\partial E(t)}{\partial \mathbf{m}(t)} \frac{\partial \mathbf{m}(t)}{\partial \mathbf{z}(t)} \frac{\partial \mathbf{z}(t)}{\partial \mathbf{W^{(1)}}} \\
&= \mathbf{v}(t) \frac{\partial E(t)}{\partial \mathbf{m}(t)} f'(\mathbf{z}(t)), \\
\frac{\partial E(t)}{\partial \mathbf{W^{(c)}}} &= \frac{\partial E(t)}{\partial \mathbf{m}(t)} \frac{\partial \mathbf{m}(t)}{\partial \mathbf{z}(t)} \frac{\partial \mathbf{z}(t)}{\partial \mathbf{W^{(c)}}} \\
&= \frac{\partial E(t)}{\partial \mathbf{m}(t)} f'(\mathbf{z}(t)) \left[\mathbf{m}(t-1) + \mathbf{W^{(c)}} \frac{\partial \mathbf{m}(t-1)}{\partial \mathbf{W^{(c)}}} \right],
\end{aligned}$$

where:

$$\frac{\partial \mathbf{m}(t-1)}{\partial \mathbf{W^{(c)}}} = \mathbf{W^{(c)}} f'(\mathbf{z}(t-1)) \frac{\partial \mathbf{m}(t-2)}{\partial \mathbf{W^{(c)}}}.$$

The error gradient with respect to the $\mathbf{W^{(c)}}$ matrix is computed in a similar fashion to those computed in Section 2.2.1. Given these gradients, the weight matrices are then updated by:

$$\mathbf{W^{(2)}}^{new} \leftarrow \mathbf{W^{(2)}} - \eta \frac{\partial E(t)}{\partial \mathbf{W^{(2)}}},$$
$$\mathbf{W^{(1)}}^{new} \leftarrow \mathbf{W^{(1)}} - \eta \frac{\partial E(t)}{\partial \mathbf{W^{(1)}}}, \text{and}$$
$$\mathbf{W^{(c)}}^{new} \leftarrow \mathbf{W^{(c)}} - \eta \frac{\partial E(t)}{\partial \mathbf{W^{(c)}}},$$

where $0 < \eta \leq 1$ is the learning rate parameter.

3.4 Model Evaluation

In order to quantify how well the model performs, we compute the *Fraction of Variance Accounted For* (FVAF) of its predictions. The FVAF is defined as:

$$FVAF = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where y_i is an observed scalar value and $\hat{y}_i$ is a predicted scalar value. Note that the FVAF computation must be done for each dimension separately. The resulting FVAF metric can fall into one of four cases: $FVAF < 0$, $FVAF = 0$, $0 < FVAF < 1$, and $FVAF = 1$. If $FVAF = 1$, then $y_i = \hat{y}_i, \ \forall i$. Thus, in this case, the model makes perfect predictions. If $FVAF = 0$, then:

$$1 = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \Rightarrow \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - \bar{y})^2,$$

meaning that the sum of prediction errors is exactly equal to the sum of the distance to the mean. This implies that the model prediction is no better or worse than simply predicting the mean. From this, it is clear that if $0 < FVAF < 1$, the model predicts some of the variance present in the data, and if $FVAF < 0$, the model inserts variance not found in the data. The model presented learns force trajectories in a two-dimensional space, and therefore has two $FVAF$ measures associated with it: one for the X dimension, and one for the Y dimension. This is done so as to preserve any differences in performance between the two dimensions.

3.5 Stability Analysis

In order to determine the reliability of cosine tuning curves to neurons and muscles, a stability analysis is performed on the tuning curves. This is to ensure that changes in PDs are a result of actual changes rather than changes in the tuning curve estimation (Stevenson et al., 2011). To perform stability analysis, I use *bootstrap resampling*, in which I form many models from a random subset of the data, and I observe the differences in the model (Stevenson et al., 2011). To determine the stability of the models, the changes in PD between models were determined. I followed Stevenson et al. (2011) and defined a cutoff value of 25°. If the PD changes by more than 25° among the models, the neuron or muscle was considered unstable and not included in the neural population analysis.

3.6 Aggregate Evaluation

Producing one model for a given data set is insufficient for generalization across new data. If the number of available data points is limited, *N-fold cross validation* can be used (Kohavi et al., 1995), in which the dataset is split up into N folds, and N models are created with $N - 2$ folds for training, 1 fold for testing, and 1 fold for parameter validation. With N distinct models, N performance measured can be obtained, giving information about the average performance as well as the variation of the performance.

I create tuples for the training and test data sets by drawing trajectories from the data set used in Oby et al. (2012). Specifically, a trial from each direction and wrist posture is drawn from a uniformally random distribution. These trials are then transformed into minimum jerk trajectories based on the starting and ending points of the trial. Visual input vectors are then created that encode the target direction of each trajectory. Visual variation is added to the visual input vectors to allow for the model to be robust to various input variations. These visual input vectors were paired with their respective trajectories.

The procedure for creating aggregate results is then to train N models, each with differently seeded random noise, thus achieving N solutions. These N solutions are then evaluated in terms of their consistency with one another. The models are then evaluated against the observed values.

Chapter 4

Results

The results of Kakei et al. (1999) and Oby et al. (2012) show that MI neurons demonstrate a movement direction tuning in wrist movement tasks involving wrist flexion/extension and radial/ulnar deviation. Furthermore, they show that the directional tuning changes with wrist posture, but that change in tuning can differ dramatically from one neuron to the next, with the mean change in directional tuning falling between zero and that which is exhibited by the muscles.

I hypothesize that constraining the network parameter choice so that hand force trajectories are accurate, while neural and muscle activation levels are minimal, will lead to a structure of neural activation patterns in the modeled neuron population that is similar to those observed in MI. Specifically:

1. The cells will be directionally tuned with a distinct preferred direction
2. The PDs of the neurons will change with respect to time and wrist posture
3. The mean PD shift will be significantly larger than 0° and distinctly different from that of muscles.

4.1 Parameter Selection

The cost function parameters determine the relative balance of force trajectory accuracy and activation levels of the neurons and muscles. I first analyze the results of the model with a specific set of parameter choices. Recall from the discussion of the model that two constraints were present: one that aimed to minimize the neuron activation patterns and one that aimed to minimize the muscle activation patterns. These parameters are referred to mathematically as λ_1 and λ_2, respectively.

For the analysis that follows, the regularization parameters λ_1 and λ_2 were chosen to be $\lambda_1 = \frac{0.05}{N}$, and $\lambda_2 = \frac{0.08}{M}$, where N and M are normalizing terms. The normalizers N and M were chosen to be the size of the population multiplied by the average activity of each member of the population in the observed dataset. Specifically, $N = 1 \times 107$, where each of the 107 neurons had an average activity of 1, and $M = 12 \times 150$, where 12 muscles were in the population with an average activation of 150.

In addition to these constraints of the model, the visual target representation was varied to ensure a robust response to small changes in the visual target input. The strength of these variation signals are characterized by the visual variation parameter α, which was chosen to be $\alpha = 0.1$. This value was chosen because it imposed some amount of variation to the inputs of the model while not causing the variation to distort the signal in a detrimental way. Full discussion of the effects of the minimization constraints as well as the visual variation parameter are explained in Section 4.7.

4.2 Model Performance

We first consider the ability of the model to recreate hand force trajectories and the muscle activation patterns used to form these trajectories, given target direction and wrist posture. Predictions were evaluated using Fraction of Variance Accounted For (FVAF).

4.3 Model Training

Training and test sets comprised of three trials in each of the eight directions for each of the two wrist postures. These trials were randomly sampled from the dataset recorded by Oby et al. (2012). The trials in the training and test set were sampled such that they were distinct. In order to remove unnecessary jerks in the trajectory, these trials were transformed to become minimum jerk trajectories, which also served to normalize the trial lengths. The inputs for these trials were a visual encoding of the endpoint of the trajectory.

Each model was trained until at least one of three criteria was met: 1) zero performance improvements were achieved in the last 50 training epochs, 2) a maximum of 20,000 training epochs was reached, or 3) the mean absolute value of the error gradients is less than 1×10^{-5}. In practice, this equated to about 10,000 training epochs. A total of 20 models were trained for the same conditions, each with different random seeds for weight initialization and input variation. The performance was assessed for

each of the 20 models and an aggregate set of modeled neurons and muscles was formed by combining the populations from each of the 20 models.

4.4 Hand Force Trajectories

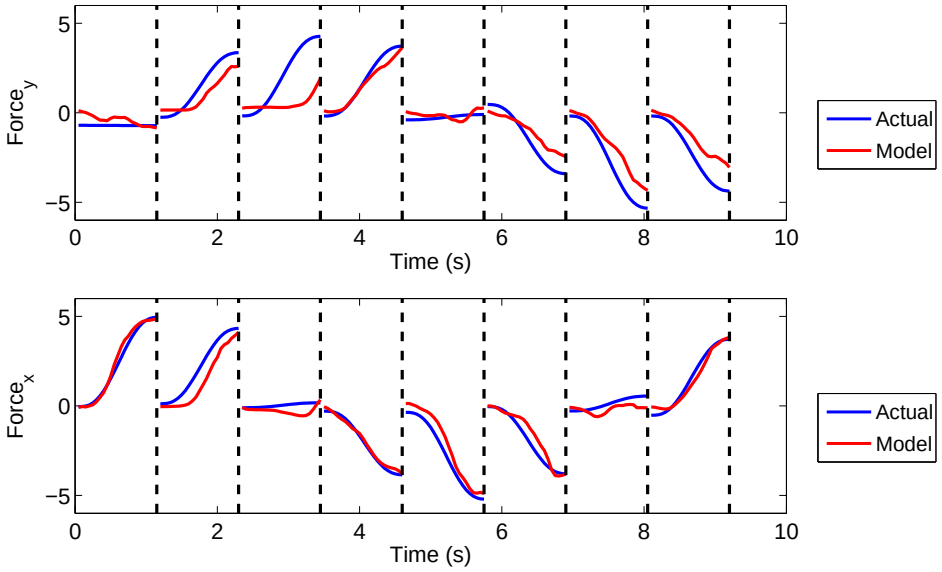

Figure 4.1: *Time series of select force reconstructions. Vertical dotted lines represent the beginning of a new trial.*

Reconstructions as a function of time for each target direction for a single model are shown in Figure 4.1. The figure shows that the model is capable of reconstructing force trajectories with good accuracy in both the midrange and pronated postures. The model is capable of using both the target direction and forearm configuration information to create the trajectory in the correct direction.

Figure 4.2 shows the mean and standard deviation of the hand force reconstruction performance exhibited by the model. This figure shows the aggregate performance of 20 runs. The bar represents the mean of these performances, and the

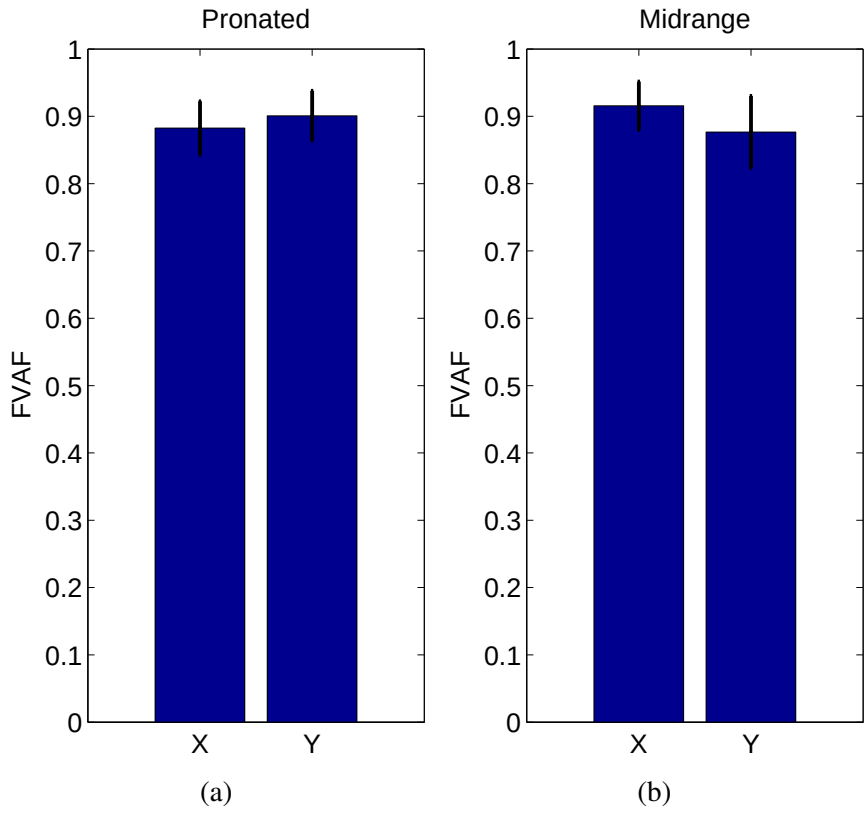

Figure 4.2: *FVAF values for the hand force reconstructions. The bars correspond to the average FVAF among a set of runs, and the lines above the bars correspond to the standard deviations among the different runs. The midrange posture is shown on the left, and the pronated posture is shown on the right.*

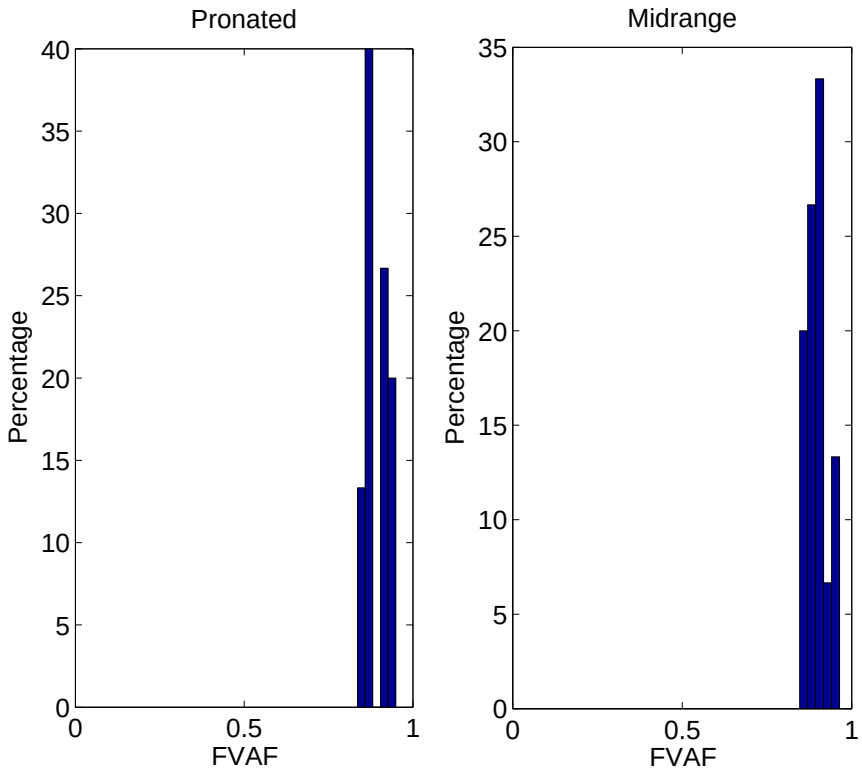

Figure 4.3: *Distribution of force FVAF values for the models. The midrange posture is shown on the left and the pronated posture is shown on the right.*

line above the bar represents the standard deviation of these performances. As can be seen, the model produces hand force trajectories with high accuracy. Figure 4.3 shows the distribution of FVAF prediction measures among the models. The midrange posture is shown on the left and the pronated posture is shown on the right. Performance measures were averaged among the X and Y dimensions, causing one performance value for each model. Performance is consistent from model to model, and all models form solutions with FVAF values greater than 0.8. The relationship of X-Force FVAF to Y-Force FVAF is shown in Figure 4.4. This figure shows a consistent relationship when predicting forces in each of the directions, with only a few outliers.

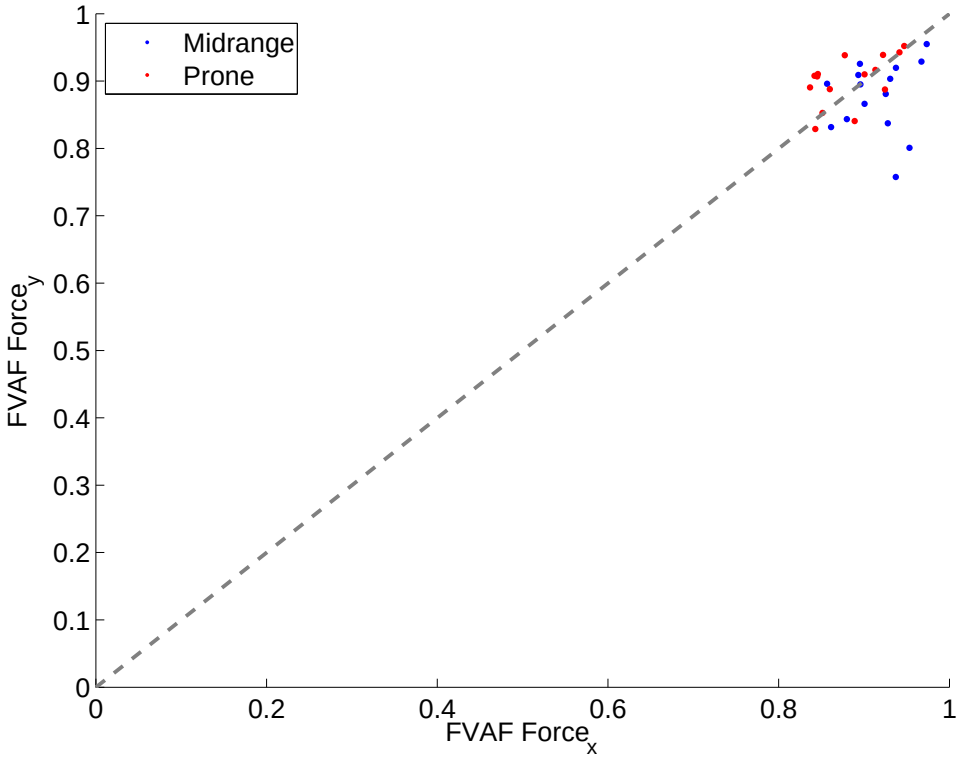

Figure 4.4: *Scatter plot of force FVAF for the X and Y dimensions. The indentity function $f(x) = x$ is shown for comparison.*

Because the precise path and timing of hand force does not affect success/failure in the wrist tracking task, I also examine the hand force produced at the end of the trial with respect to the hand force target. Each trial contains a 0.5 second hold time at the end in which the modeled subject holds the force at the target. The prediction error at this hold time shows the ability of the model to reach the target position during the trial. I use the *root mean squared error* of the trajectory at the hold time to define the endpoint error. Root mean squared error is defined as:

$$RMS = \sqrt{\frac{\sum_{t=1}^{T}(y(t) - \hat{y}(t))^2}{T}},$$

where N is the number of observed hand force values during the hold period, $y(t)$ is the true force values at time t during the hold period and $\hat{y}(t)$ is the predicted value at time t during the hold period. The distance from the starting location to the center of the target is five units, and each target is a 2x2 square. Thus, an error $0 < r < 5$ indicate a trajectory that models the general direction of the true trajectory, but falls short in terms of distance. A boxplot of the force FVAF measures for the various models averaged over the X and Y dimensions is shown in Figure 4.5. In order to visualize this error as it relates to the task, the mean endpoint predictions are shown in force space in Figure 4.6. The endpoint predictions are somewhat better in the pronated posture than in the midrange posture. The predictions generally undershoot the goal postures. This effect is likely due to the neuron and muscle terms in the cost function. This is discussed further in Section 4.7.

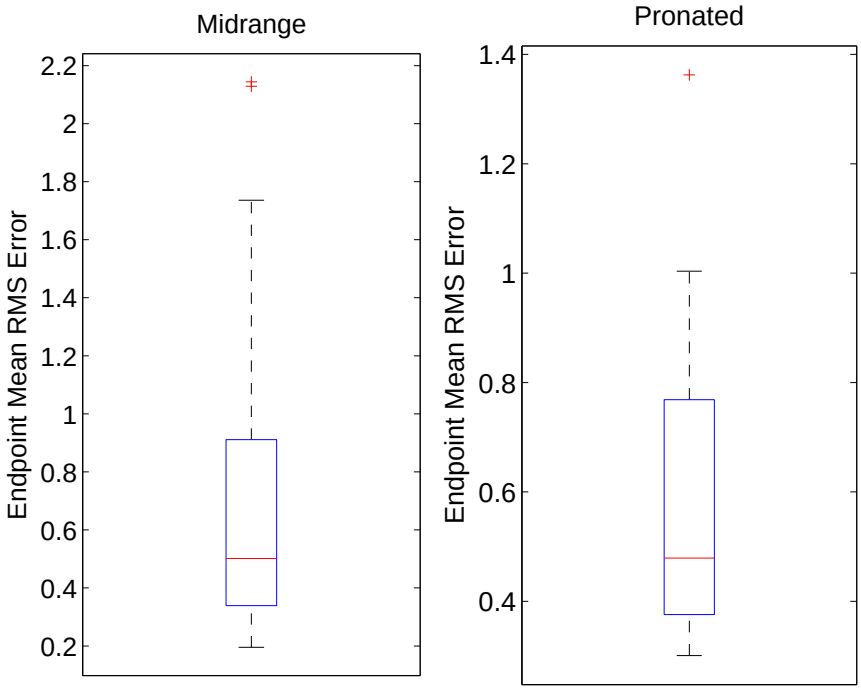

Figure 4.5: *Boxplot of RMS error of endpoint prediction.*

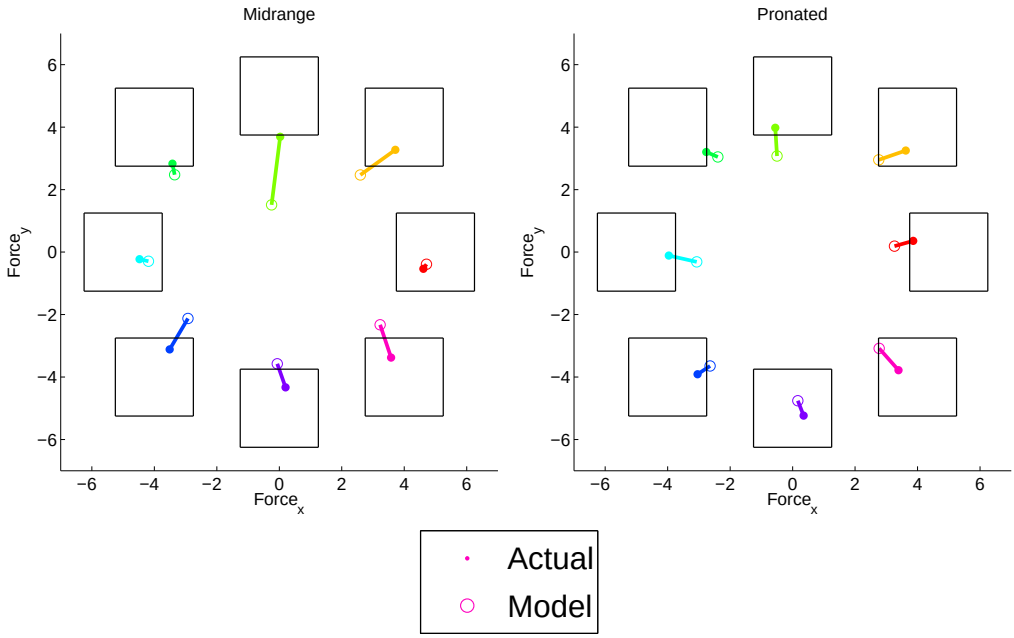

Figure 4.6: *Predictions of endpoint in force space. Lines are drawn between the predicted endpoints and actual endpoints. The midrange posture is shown on the left and the pronated posture is shown on the right.*

4.5 Neuron Behavior

In order for the model to produce hand force trajectories in time, it is necessary for the MI neurons to modulate their activity as a function of both time within the trial and the target direction. In order to describe the tuning properties of a neuron at a particular time in a trial, I fit a tuning curve of the form $f(t, \theta) = a(t) + b(t)\ cos(\theta_{PD}(t) - \theta)$, where $a(t)$ is the baseline activity of the cell at time t, b is the depth of modulation at time t, θ_{PD} is the preferred direction of the cell at time t, and θ is the movement direction. Note that at a given time, a neuron may or may not exhibit clear directional tuning. The stability of a neuron is determined through stability analysis as described in Chapter 3. Figure 4.7 shows the activity of a typical neuron as a function of target direction and time. Figures 4.7C and 4.7D show the same information as a heatmap, so that changes in time are more discernable. As explained in Chapter 2, this θ_{PD} parameter defines the preferred direction of the neuron.

Figure 4.7 shows the activation levels and cosine tuning curves of a single neuron through time. The color of the heatmaps shows the activation levels as time and target direction changes. The other panels show the activation levels as a function of target direction and time, and their associated cosine tuning curves. Color in these panels represents time through the trial. This figure not only shows the difference in behavior of a neuron across postures, but also the behavior of a neuron changes through time. The PD at a particular time can be read as the peak of the tuning curve at that time. A trial starts with red for the activations and tuning curves, by the middle

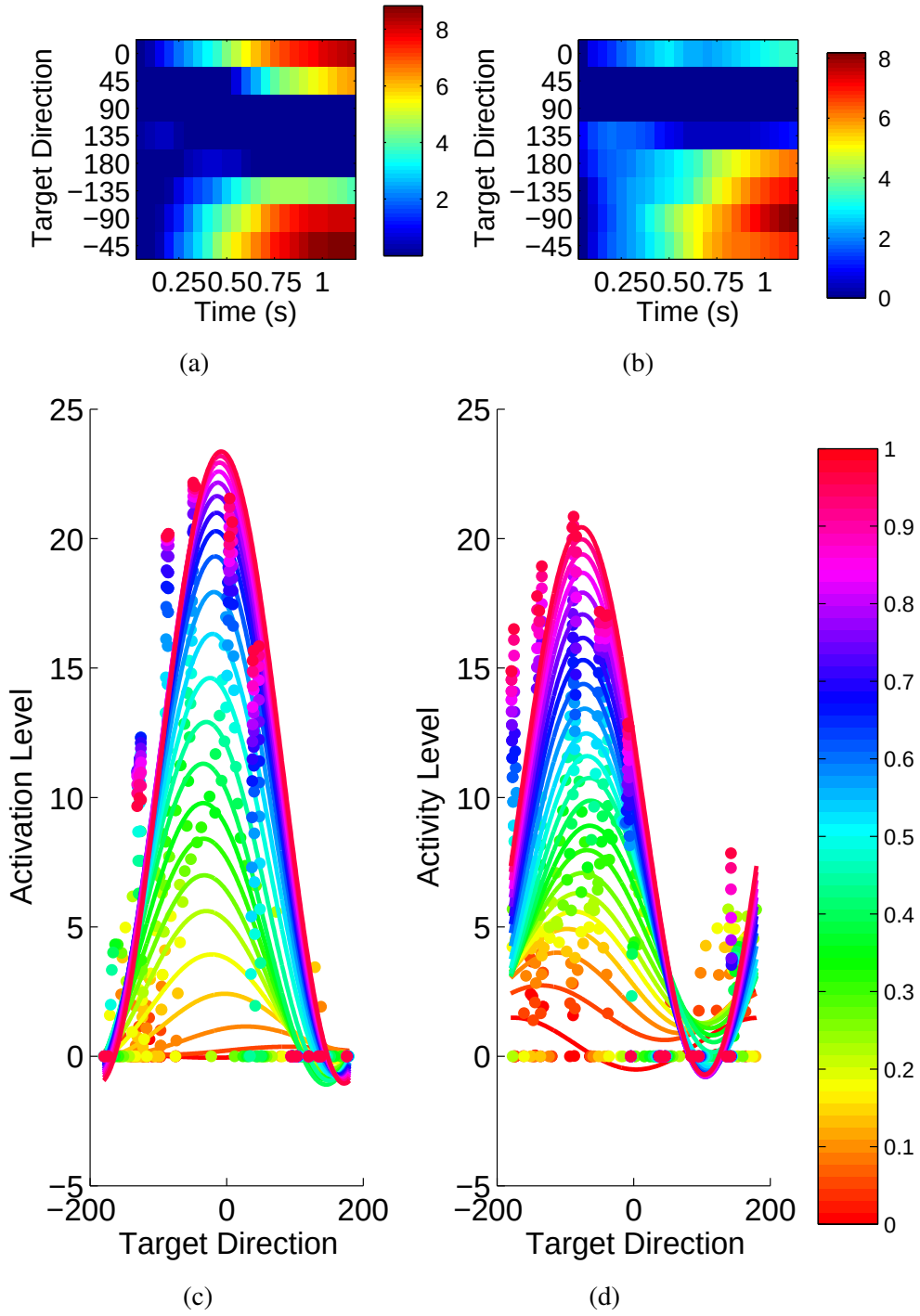

Figure 4.7: *Neuron activity (indicated with color) as a function of time and target direction in the midrange (a) and pronated postures (b). Activation level of the neuron as a function of target direction and the corresponding cosine tuning function for the midrange (c) and pronated (d) postures. The activation level and tuning function is shown for different times during the trial, with time being encoded by color.*

of the trial the activations and the associated cosine tuning curves are blue, and the activations and tuning curves are magenta at the end of the trial. Changes in shape of the tuning curves as color changes show a change in neural behavior through time.

There are many different types of neurons observed in the modeled population. Figure 4.8 shows a neuron in which there is a significant depth of modulation change across the two postures. The neuron distinguishes between postures by being more active in the pronated posture than the midrange posture. The change in PD between the two postures is only $30°$, but the depth of modulation for the midrange posture is twice that of the pronated posture. This is reflective of the "modulated" neurons that were observed by Kakei et al. (1999). Figure 4.9 shows a neuron in which there is little difference between the midrange and pronated postures with respect to both PD and depth of modulation. Such neurons were somewhat rare, and often changes in wrist posture were accompanied by changes in PD, depth of modulation, or both.

Some neurons change their behavior through the trial in a particular posture. For example, Figure 4.10 shows a neuron in which there is a substantial PD change as the trial progresses in the midrange posture, but not such a significant PD change in the pronated posture.

Many neurons in the modeled population were not active until later in the trial. Such a neuron is shown in Figure 4.11. These neurons show no activity during the beginning of the trial, and begin to modulate as the trajectory reaches the target. This behavior could be an indication that a particular neuron encodes forces that fall above a certain threshold.

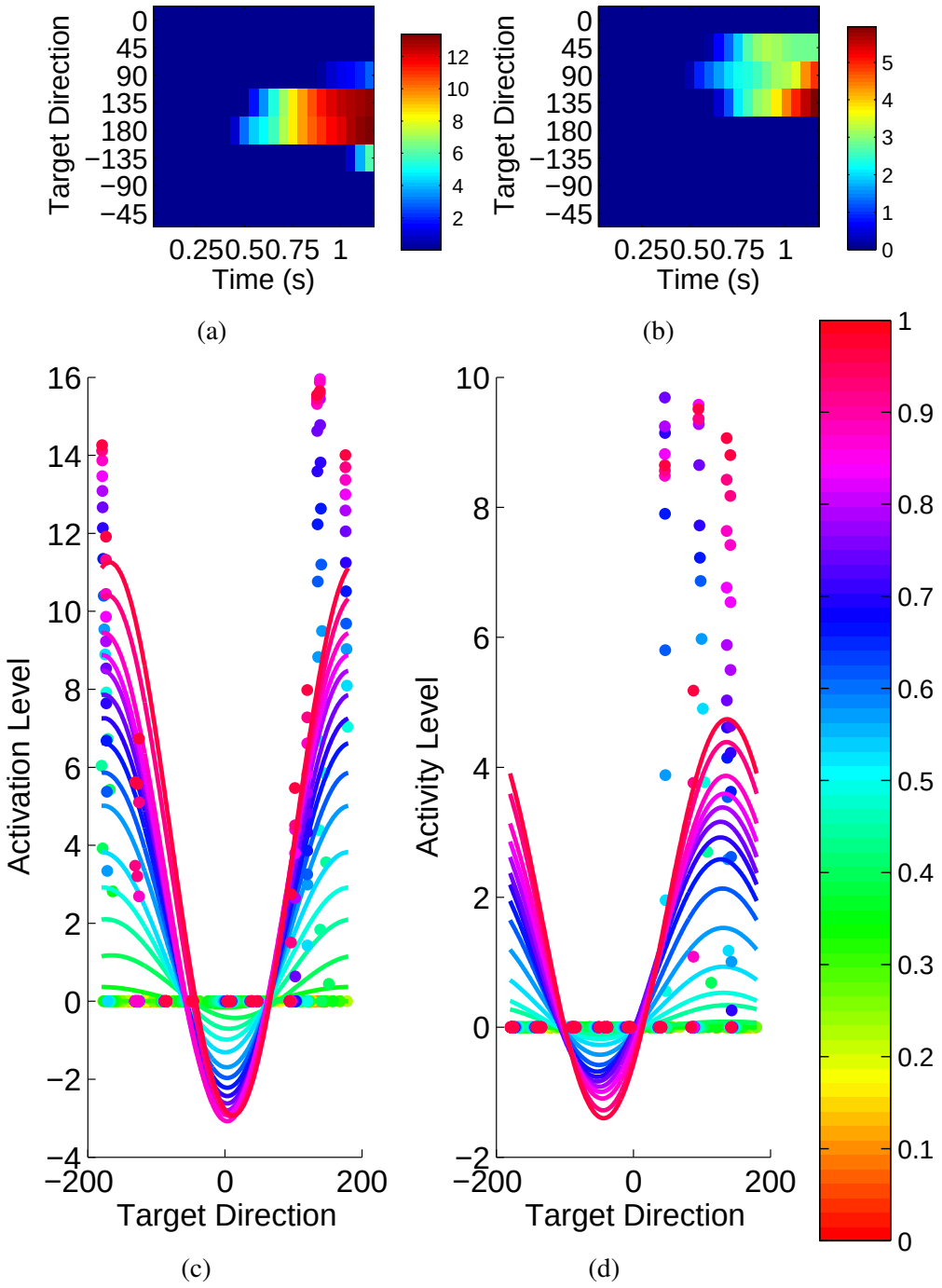

Figure 4.8: *Neuron activity (indicated with color) as a function of time and target direction in the midrange (a) and pronated postures (b). Activation level of the neuron as a function of target direction and the corresponding cosine tuning function for the midrange (c) and pronated (d) postures. The activation level and tuning function is shown for different times during the trial, with time being encoded by color.*

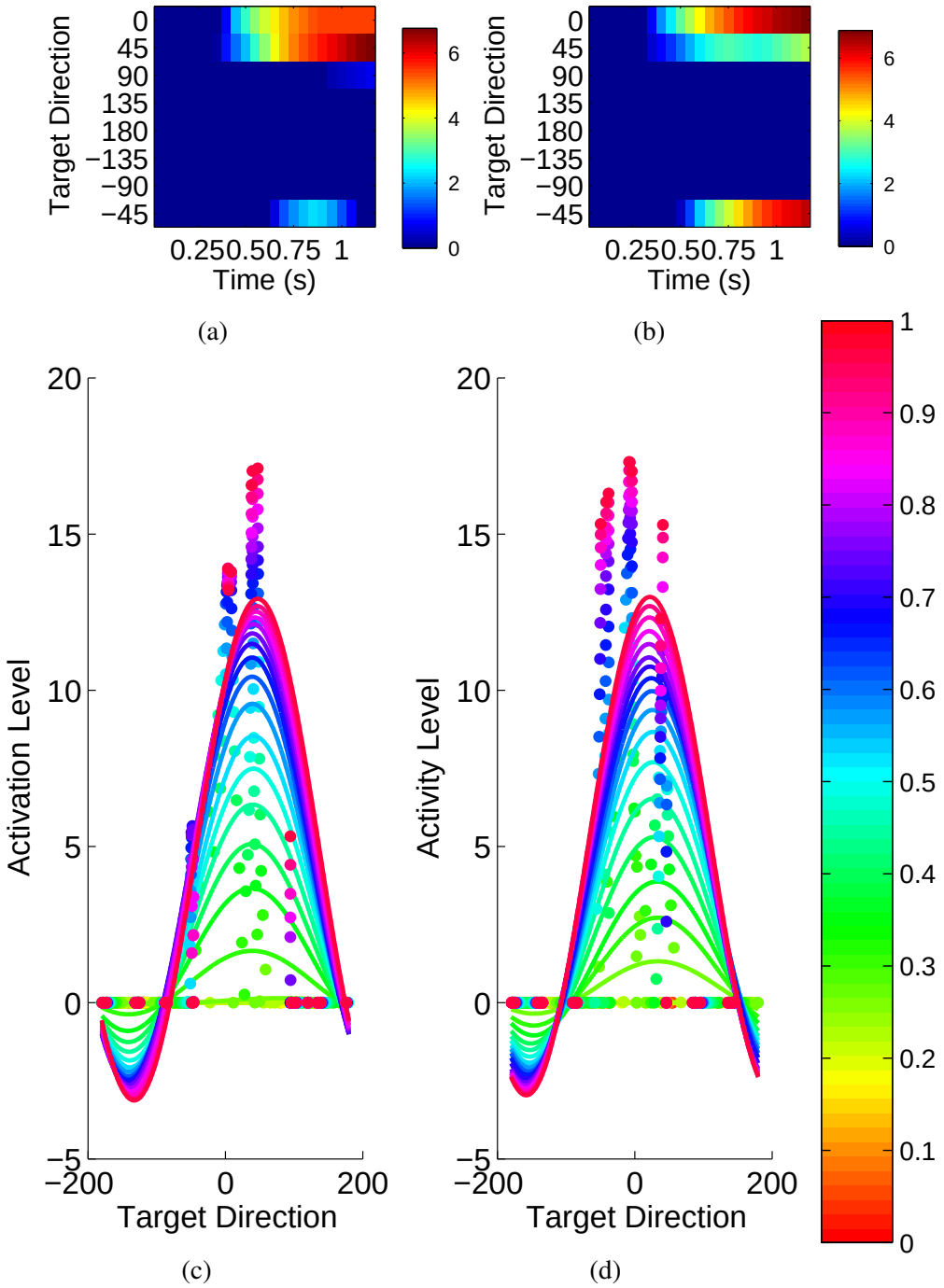

Figure 4.9: *Neuron activity (indicated with color) as a function of time and target direction in the midrange (a) and pronated postures (b). Activation level of the neuron as a function of target direction and the corresponding cosine tuning function for the midrange (c) and pronated (d) postures. The activation level and tuning function is shown for different times during the trial, with time being encoded by color.*

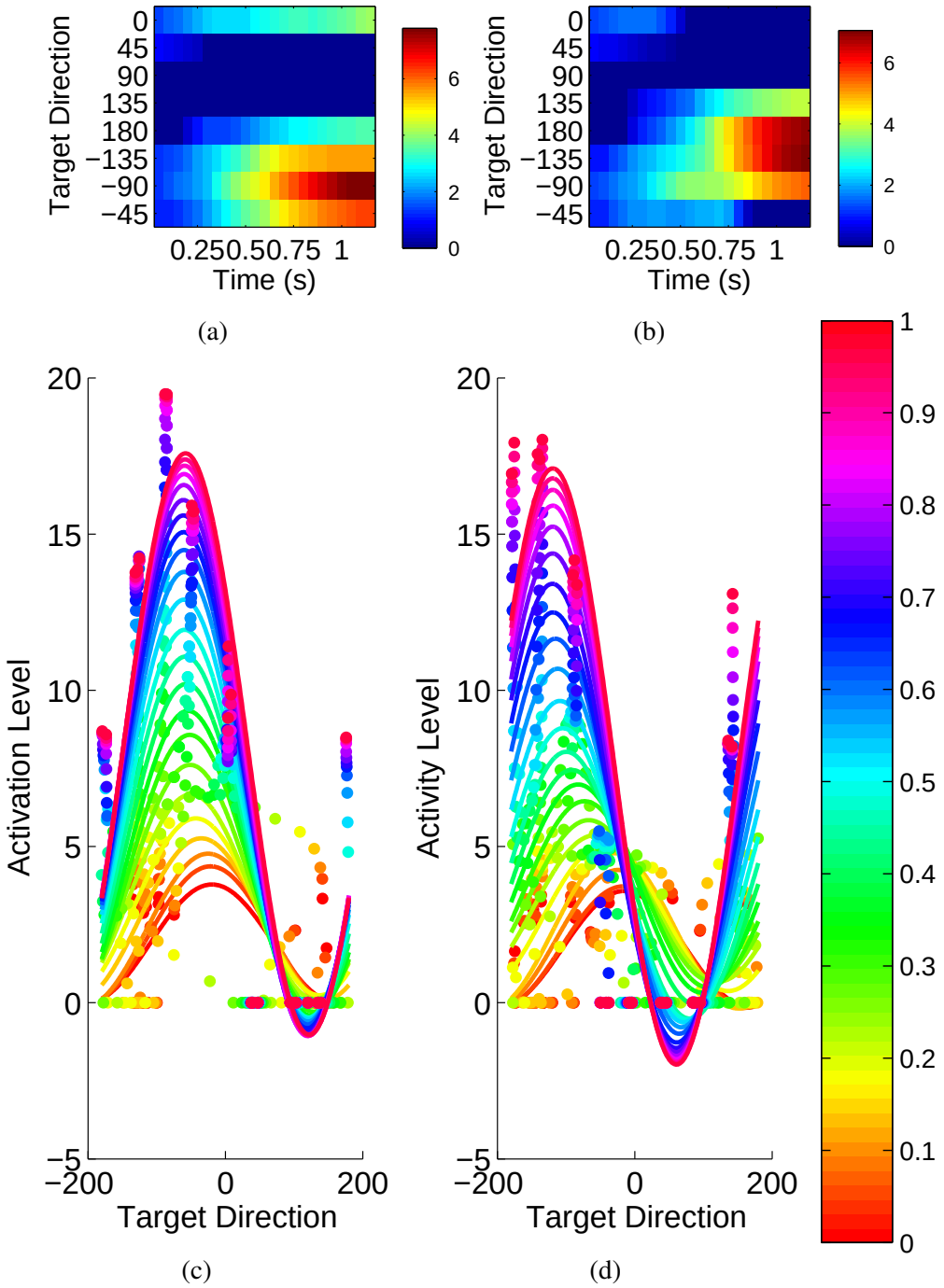

Figure 4.10: *Neuron activity (indicated with color) as a function of time and target direction in the midrange (a) and pronated postures (b). Activation level of the neuron as a function of target direction and the corresponding cosine tuning function for the midrange (c) and pronated (d) postures. The activation level and tuning function is shown for different times during the trial, with time being encoded by color.*

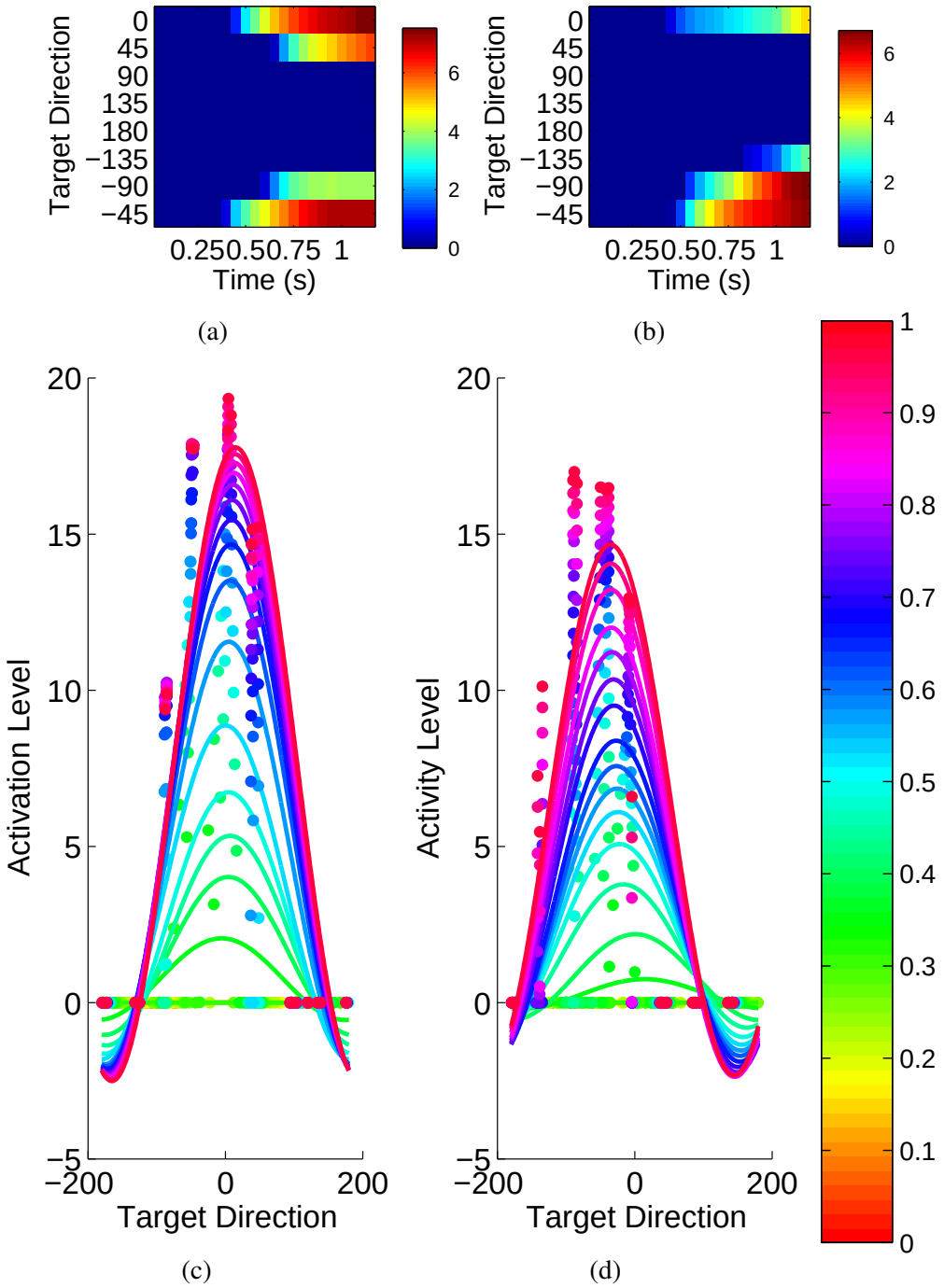

Figure 4.11: *Neuron activity (indicated with color) as a function of time and target direction in the midrange (a) and pronated postures (b). Activation level of the neuron as a function of target direction and the corresponding cosine tuning function for the midrange (c) and pronated (d) postures. The activation level and tuning function is shown for different times during the trial, with time being encoded by color.*

Finally, some neurons show little or no activity throughout the trial. This is shown in Figure 4.12. No activity is shown at the beginning of a trial, and very little activity is shown at the end of a trial. Figure 4.13 shows a neuron which is active in the pronated posture but not active in the midrange posture. Neural behaviors such as this can occur because of the excess degrees of freedom that are available in the network. A large population of neurons (107 in this case) is recruited to control a much smaller group of muscles (12 in this case), which are used to drive forces in a much lower dimensional space (2).

Individual neurons in the modeled population exhibit a variety of behaviors. While many neurons show activity in both postures and have definite PDs and depths of modulation, other neurons show little or no activity in one posture or both. Many neurons in the modeled population show low activation levels at the beginning of the trial, with some neurons showing activations only late in the trial and others showing activations earlier. Because of the excess degrees freedom in the neural population, individual neurons can behave in many different ways without loss of task performance. This can explain the many different behaviors previously shown. While the neural population of my model encodes a constrained task, each neuron need not consistently contribute to this task. Specifically, each neuron need not encode all target directions or both forearm postures. For this reason, we see the behaviors previously shown, including neurons that are tuned for one forearm posture and not the other, neurons that modulate their PD with forearm posture, neurons that change their depth of modulation with forearm posture, and neurons that show little or no activity throughout the trial.

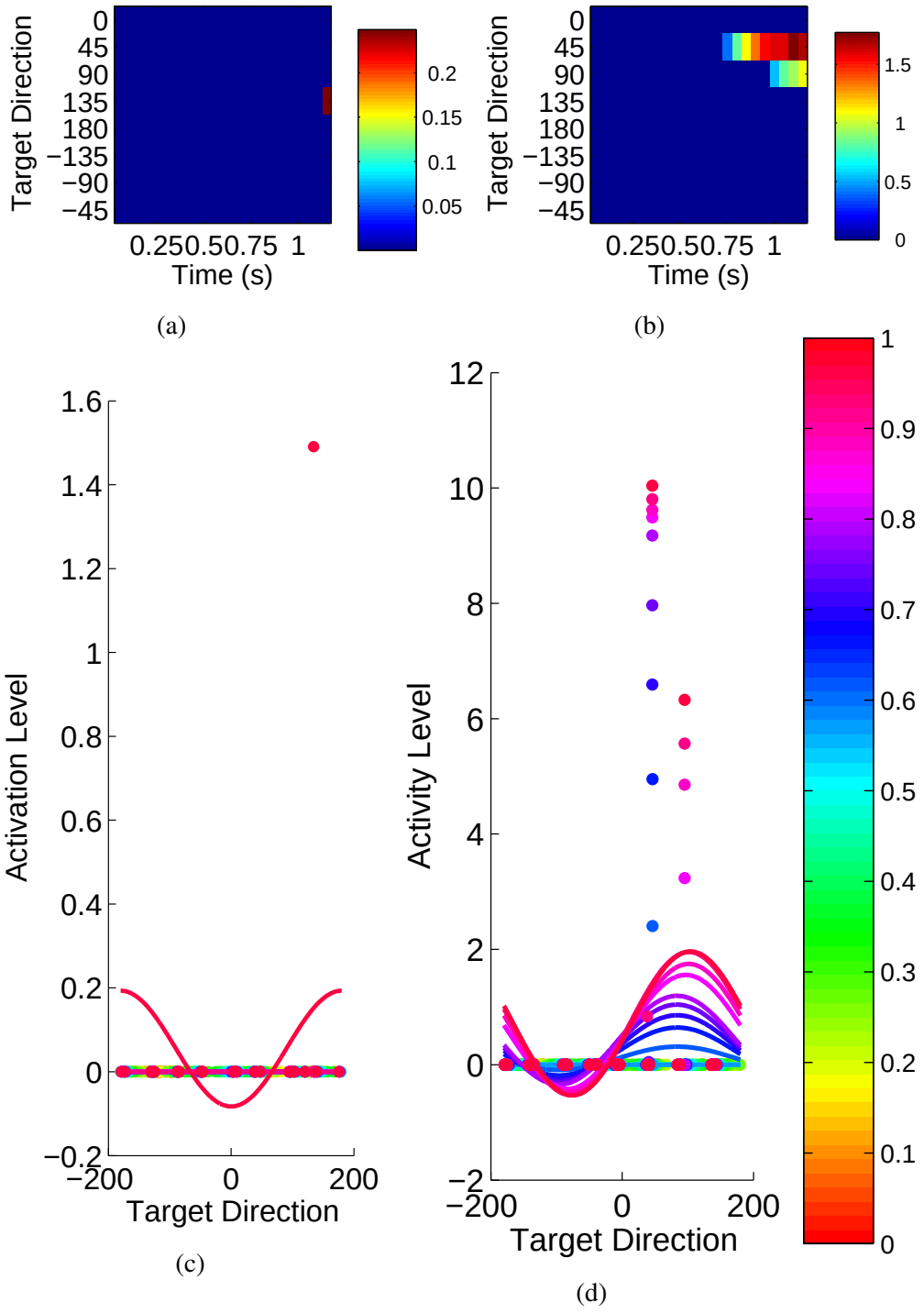

Figure 4.12: *Neuron activity (indicated with color) as a function of time and target direction in the midrange (a) and pronated postures (b). Activation level of the neuron as a function of target direction and the corresponding cosine tuning function for the midrange (c) and pronated (d) postures. The activation level and tuning function is shown for different times during the trial, with time being encoded by color.*

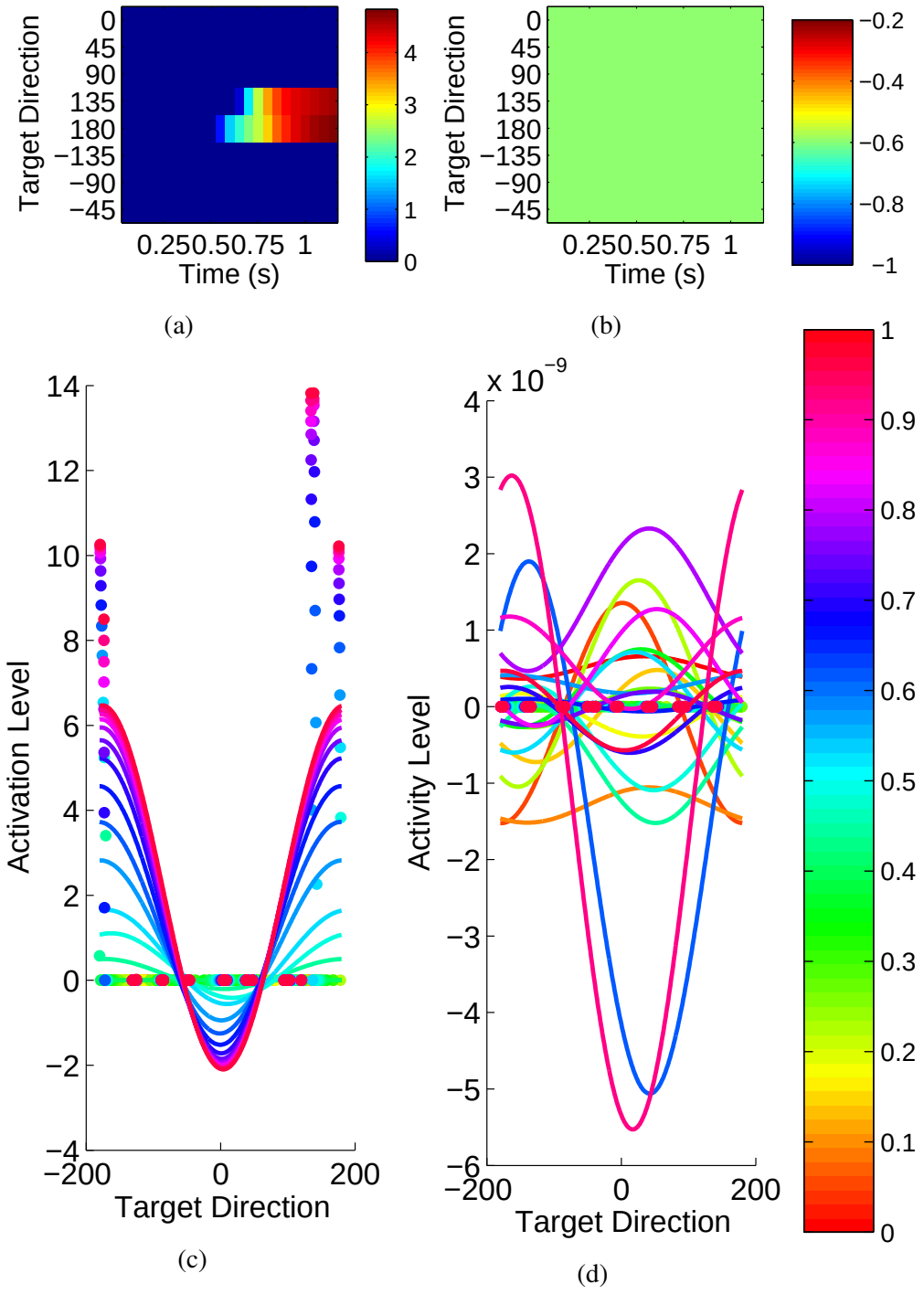

Figure 4.13: *Neuron activity (indicated with color) as a function of time and target direction in the midrange (a) and pronated postures (b). Activation level of the neuron as a function of target direction and the corresponding cosine tuning function for the midrange (c) and pronated (d) postures. The activation level and tuning function is shown for different times during the trial, with time being encoded by color.*

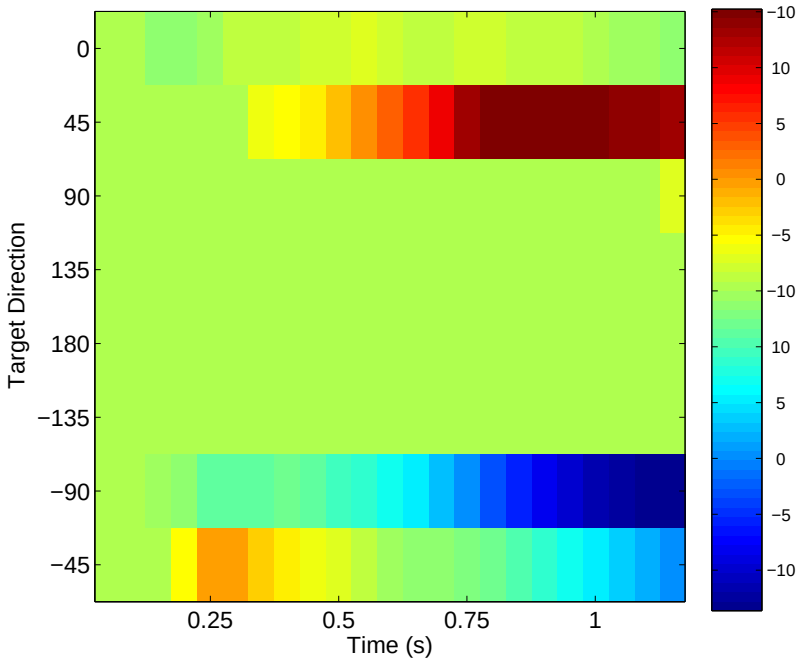

Figure 4.14: *Heatmap representing the change behavior between postures of a neuron for a given time and target direction. Color represents the change in activation between the two postures.*

As previously seen, many neurons change behavior as the forearm posture changes. This change in behavior across forearm postures is a function of time and target direction. A visualization of the changes in behavior as a function of time and target direction for a neuron is shown in Figure 4.16. This particular neuron has depth of modulation changes and a smaller PD shift. This figure corresponds to the neuron depicted in Figure 4.8. In Figure 4.16, color represents the difference in activity between the pronated and midrange postures. Thus, a cell represented as zero corresponds to no change in activity level, whereas non-zero values represent some amount of change in behavior. The figure shows that much of the depth of modulation changes occur towards the end of the trial. This is likely because of the increased activity that neurons exhibit towards the end of the trial.

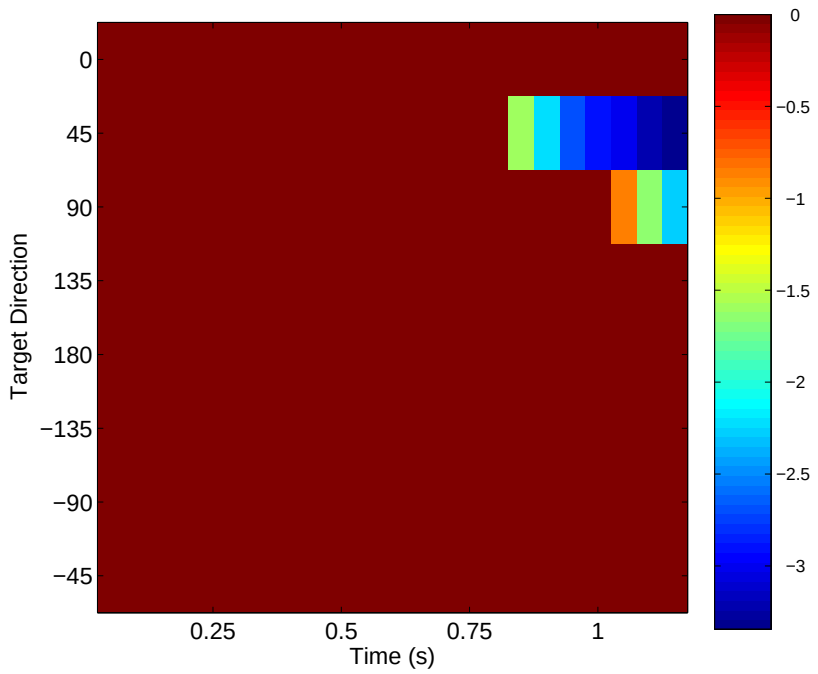

Figure 4.15: *Heatmap representing the change behavior between postures of a neuron for a given time and target direction. Color represents the change in activation between the two postures.*

Similar analysis can be done for the other types of neurons. Figure 4.14 shows a neuron that changes its PD by $45°$ between postures. Neurons that exhibit little activity throughout the trials often exhibit little change between postures as well. One such neuron is shown in Figure 4.15. There were some small posture-dependent changes at the end of the trial, but these changes were concentrated among certain times and target directions.

The distribution of neurons active or not active in one or both postures for a single model is shown in Table 4.1. The majority of neurons are active in both postures, and there is no bias towards one posture over another.

Changes in the PD of an individual neuron can explain the origin of the PD shift for that neuron. Some neurons end the trial with a PD shift in the opposite direction

	Not Active in Midrange	Active in Midrange
Not Active in Pronated	10%	11%
Active in Pronated	14%	65%

Table 4.1: *Distribution of neurons active in both of the forearm postures.*

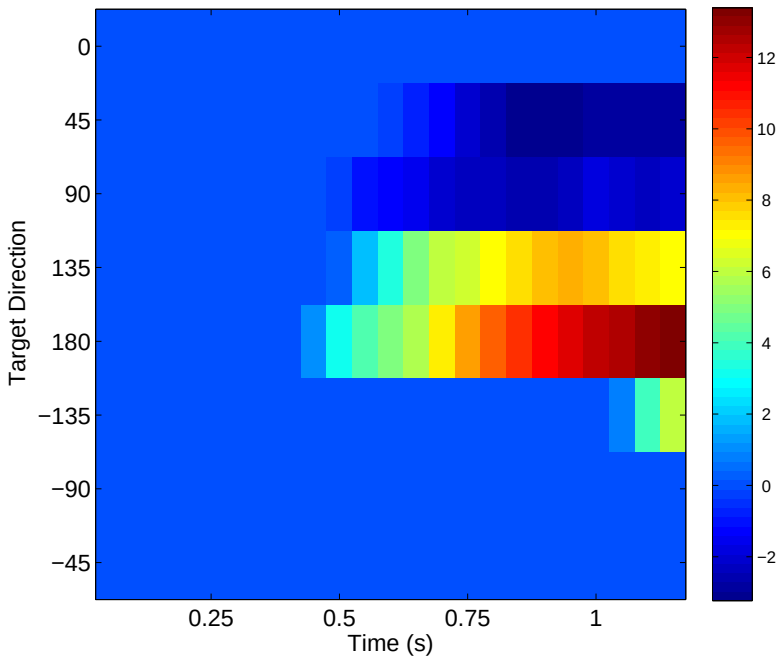

Figure 4.16: *Heatmap representing the change behavior between postures of a neuron for a given time and target direction. Color represents the change in activation between the two postures.*

as the PD shift it exhibited at the beginning of the trial. This is shown in Figure 4.17. The neuron begins the trial with a PD shift of about $-50°$, and ends with a PD shift of about $50°$. As mentioned earlier, many neurons do not exhibit large activations at the beginning of a trial. Thus, their preferred directions are not as reliable, and do not convey as much information as when the neuron exhibits larger depth of modulations. Because of this, it is possible that the change in PD shift sign that the neuron in Figure 4.17 exhibits is insignificant.

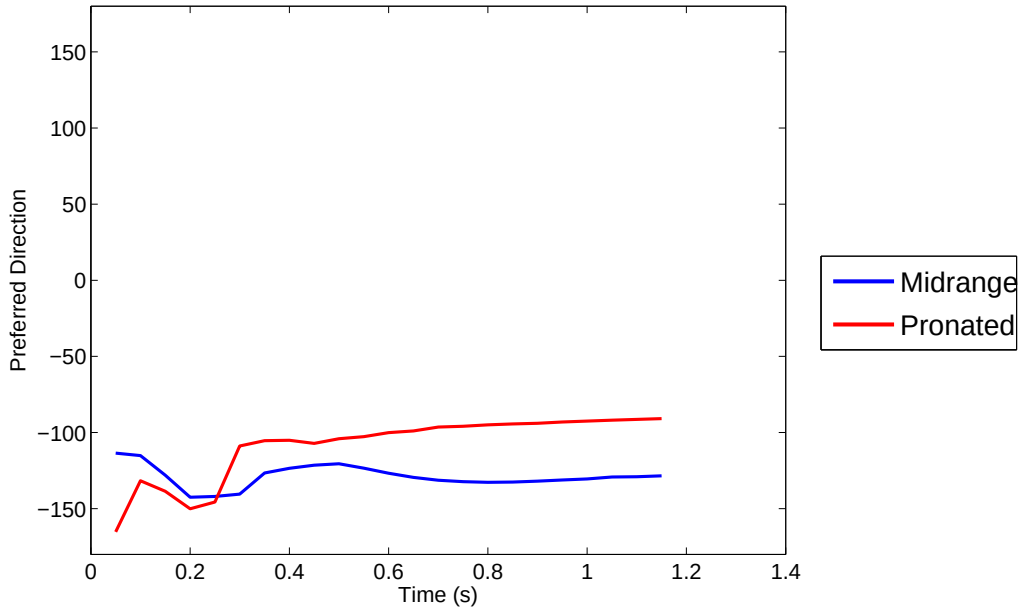

Figure 4.17: *The change of PD throughout the trial in both postures for a neuron. The blue line represents the midrange posture, and the red line represents the pronated posture.*

Another subpopulation of neurons start their activity with no PD shift, and form their PD shift as the trial progresses. This is shown in Figure 4.18. This neuron does not show activity at the start of the trial, and therefore has no reliable PD for which to analyze. When the neuron begins to show activity, the PD shift is small, and then becomes larger as the trial progresses. With this particular neuron, the origins of the PD shift arise from the change in preferred direction in the pronated posture. The preferred direction of the pronated posture does not change substantially through the trial, whereas it does change substantially in the midrange posture. This neuron is somewhat rare in that respect, as most neurons see a PD change in both postures, which creates the PD shift.

Because the context units of the network are initialized homogeneously, the only direction-specific input to the MI neurons is visual, and hence, extrinsic. For most

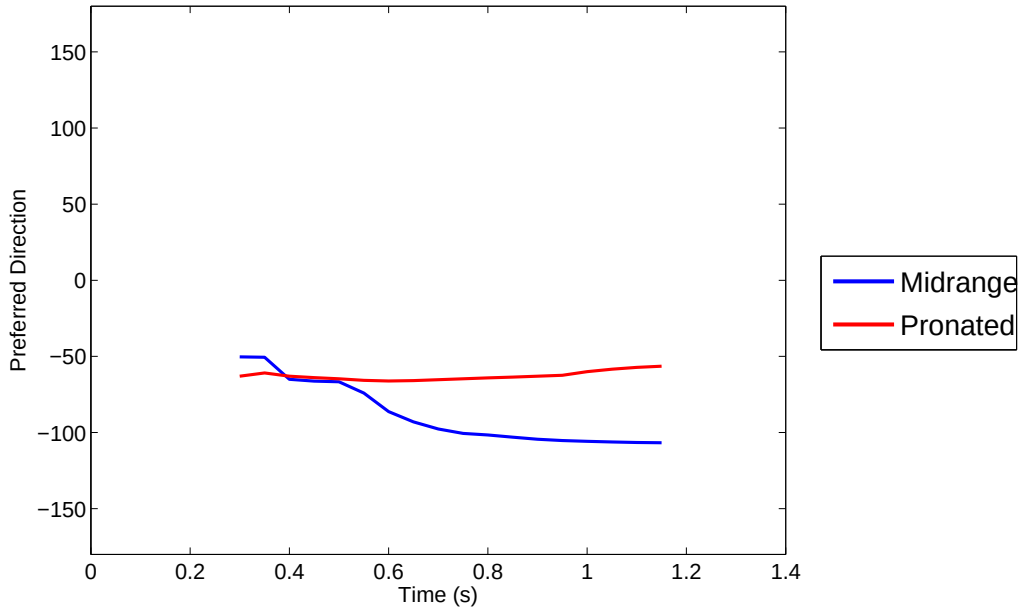

Figure 4.18: *The change of PD throughout the trial in both postures for a neuron. The blue line represents the midrange posture, and the red line represents the pronated posture.*

neurons, this manifests itself as largely an extrinsic encoding early in the trial. Only through the recurrence and the posture-specific input do the neurons generally begin to exhibit a directional tuning that is posture-specific. These PD shifts are driven by the requirement of the network to recruit muscles in a posture-dependent manner in order to produce the correct forces in task space.

Some neurons become well-tuned at different points in the trial, depending on forearm posture. This is shown in Figure 4.19. This neuron is well-tuned throughout the trial in the midrange posture, but only becomes active in the pronated posture after about 0.5 seconds. This difference in time of initial activation among postures is common in the neuron population, and does not favor one posture over another. Because the central nervous system has excess degrees of freedom to compute the transformations, each individual neuron need not encode information in the same way

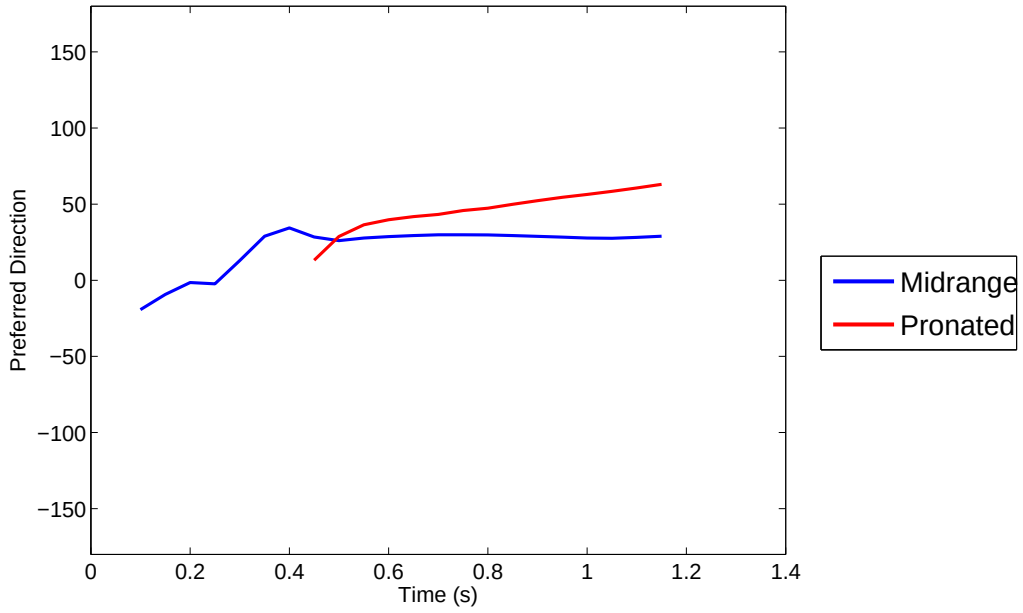

Figure 4.19: *The change of PD throughout the trial in both postures for a neuron. The blue line represents the midrange posture, and the red line represents the pronated posture.*

across the two postures. While these recruitment schemes may seem inconsistent on a neuron-by-neuron basis, it is the recruitment of the entire population of neurons that drives the transformation.

Analyzing the dynamics of the neural popuation as a whole can give insight into how the model computes the transformation. A histogram showing PD shifts computed from observed MI data recorded from monkeys is shown in Figure 4.20. For this distribution, PDs were computed using activations during the hold period of a trial (the last 0.5 seconds of a trial), and only stbale neurons (as determined through stability analysis) were included. The distribution has a mean of $18.9°$. This mean is distinctly between an extrinsic coordinate frame (which would exhibit a PD shift mean of about $0°$), and a muscle coordinate frame (which would exhibit a PD shift mean of about $70°$).

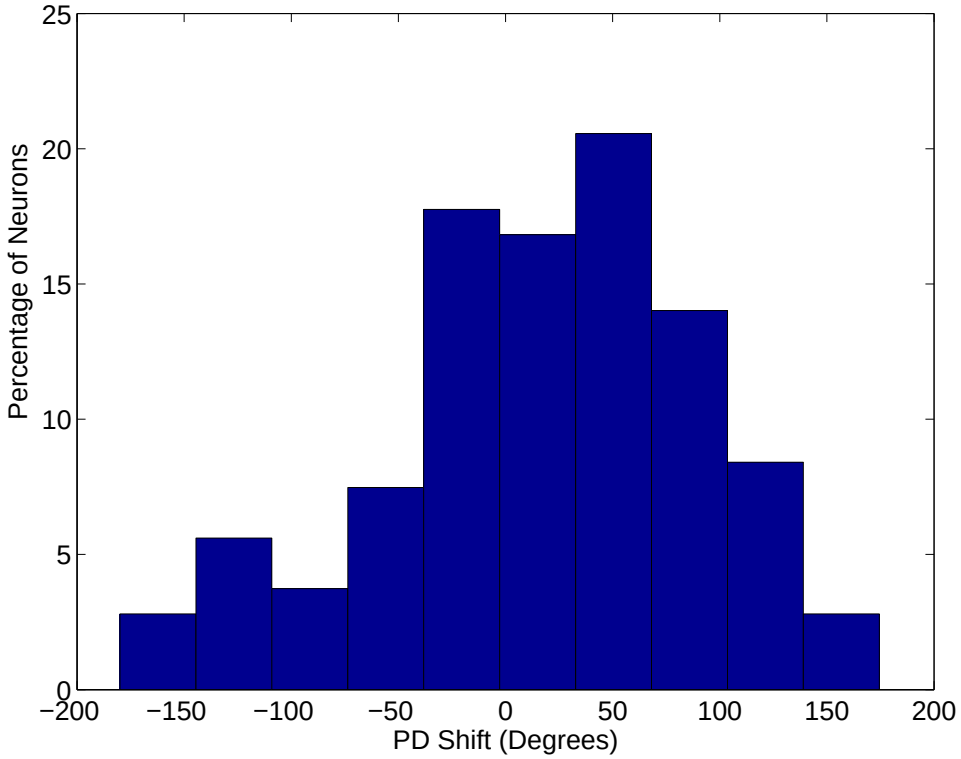

Figure 4.20: *Histogram of PD shifts exhibited by MI neurons recorded by Oby et al. (2012) during the hold period of the wrist task.*

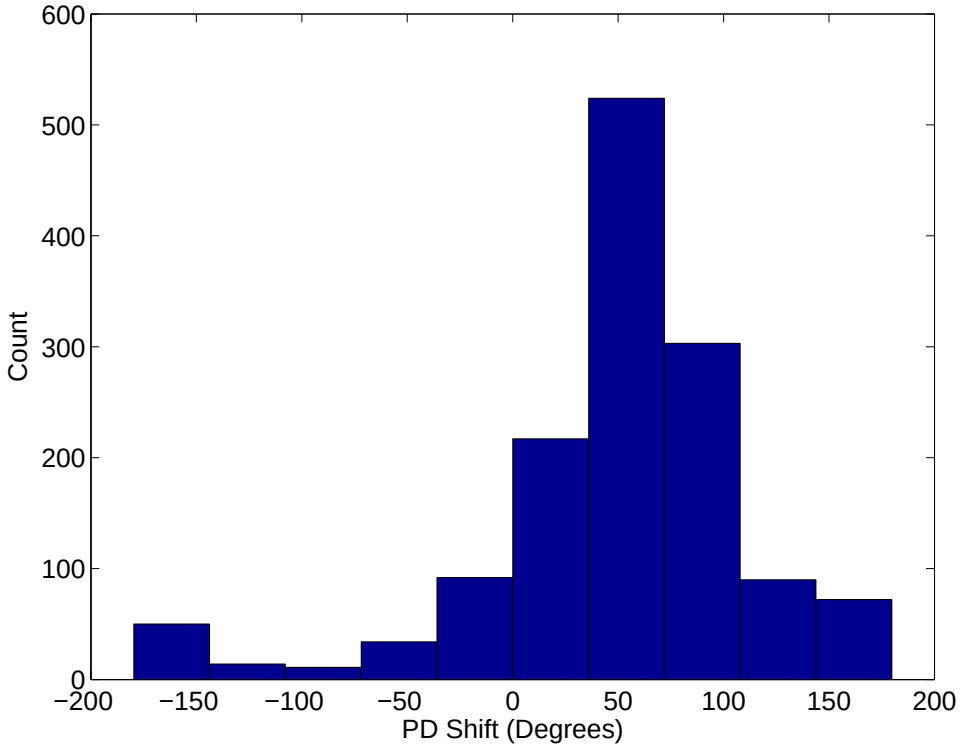

Figure 4.21: *Histogram of PD shifts exhibited by modeled MI neurons during the hold period of the wrist task.*

The PD shift distribution computed from the aggregate population of 20 runs of the model is shown in Figure 4.21. The PDs were again computed using activations during the hold period of a trial, and only stable neurons were included. The mean of the distribution equals $52.2°$, which, similar to the observed population, falls between an extrinsic coordinate frame and a muscle-like coordinate frame.

Aside from preferred direction, the depth of modulation of a neuron can change across postures. Figure 4.22 shows a scatter plot of changes of depth of modulation vs. PD shift for the aggregate modeled population. There appears to be some structure in this figure, in which neurons with PD shifts around $0° - 100°$ exhibit greater variance in their depth of modulations than other neurons. By binning each neuron according to its preferred direction, with a bin size of $30°$, several distributions of

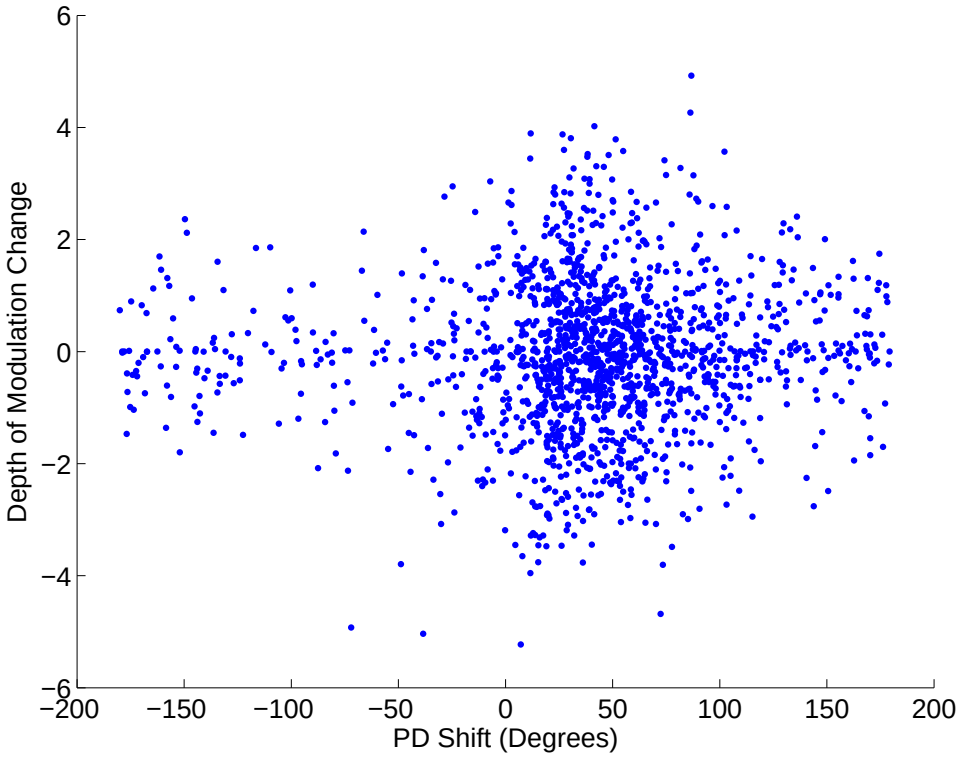

Figure 4.22: *Scatterplot of PD shift versus changes in depth of modulation for the modeled neurons.*

depth of modulation changes were formed. These distributions were compared with each other via the Kolmogorow-Smirnov test in order to determine any differences between distributions. There is a statistically significant increase in the variance of depth of modulation changes among neurons that have PD shifts between 0° and 30°. This suggests that neurons that have a more "extrinsic" PD shift have a wider range of depth of modulation changes.

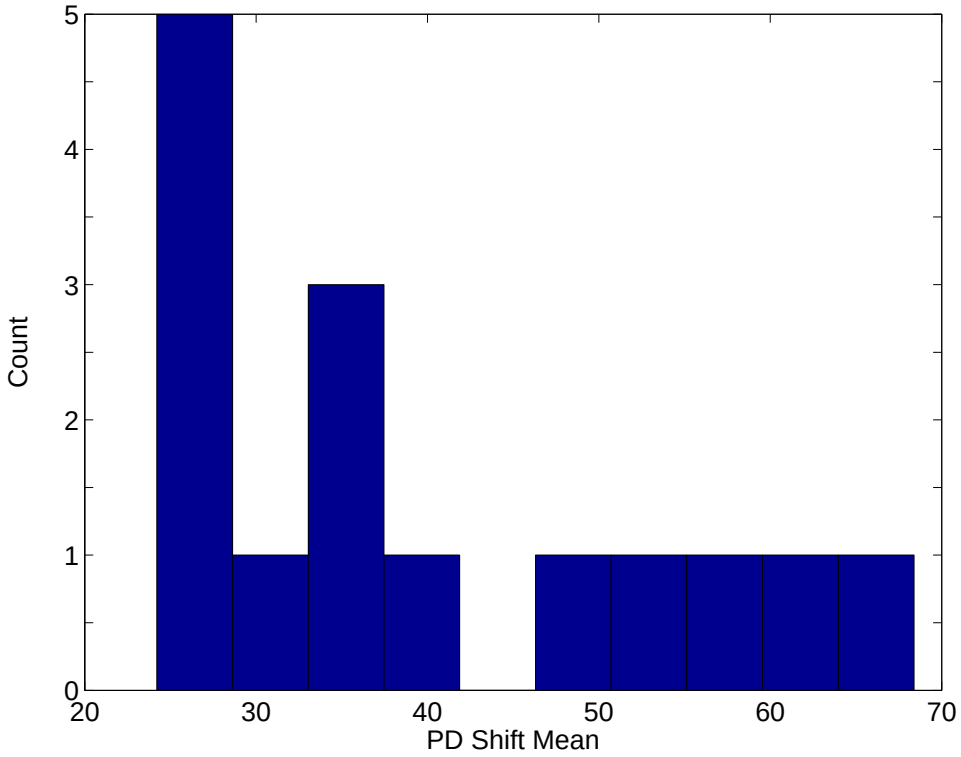

Figure 4.23: *The distribution of PD shift means of the models learned.*

Figure 4.23 shows the distribution of the endpoint PD shift means from the various models learned. Some models reach PD shift means of almost 70°, but most models have PD shift means in the range of $25^\circ - 40^\circ$. It is interesting to note that all of the models have PD shift means within 0° and 70°. This is consistent with the findings of Oby et al. (2012), in which the PD shift mean fell between an extrinsic

coordinate frame ($0°$) and PD shift mean of muscles ($70°$). Oby et al. (2012) noticed different PD shift means from different monkeys, ranging from $34°$ to $63°$, which is a similar range shown in Figure 4.23.

One of the interesting analyses that can be done is observing how PDs and PD shifts change through time. Quantifying PD shift changes through time is achieved by computing the mean of the PD shift distribution at each point throughout the trial. The PD shift mean as it changes through time for the observed MI data is shown in Figure 4.24. Note that the mean does not change much. From this figure, the wide distribution is clearly visible through the standard deviation bars. The lack of a substantial increase in PD shift through time can indicate that neurons in MI are actively tuned for the duration of the trial. This may be because time zero corresponds to the onset of movement, and by this point much of the motor planning would have already occurred, and thus many neurons may be mostly tuned by the time the trial begins.

Changes in PD shift mean through time for the modeled neuron population are shown in Figure 4.25. In both the observed and modeled data, the PD shift mean starts around $0°$ at the beginning of the trial, and ends with a PD shift mean that is equal to the mean of the PD shift computed through the hold period. The modeled MI population has substantially less variation among these PD shift means as the trial progresses.

As mentioned, modeled neurons tend to have smaller activations at the beginning of the trial. Because of this, changes in their activations at the beginning of the trial across postures are small, too. Thus, they have small depth of modulation changes and PD shifts at the beginning of the trial. As the trial progresses, neurons exhibit

larger activations.

The monkey MI neurons do not show a lower level of activation at time zero than other points in the trial. This is due to the fact that time zero represents the initiation of movement. In addition, MI is likely involved in the monkey needing to hold its arm in a specific configuration during the task. In the model, movement planning and execution occur at the same time, and therefore the neurons in the modeled population do not show large activations at the beginning of the trial.

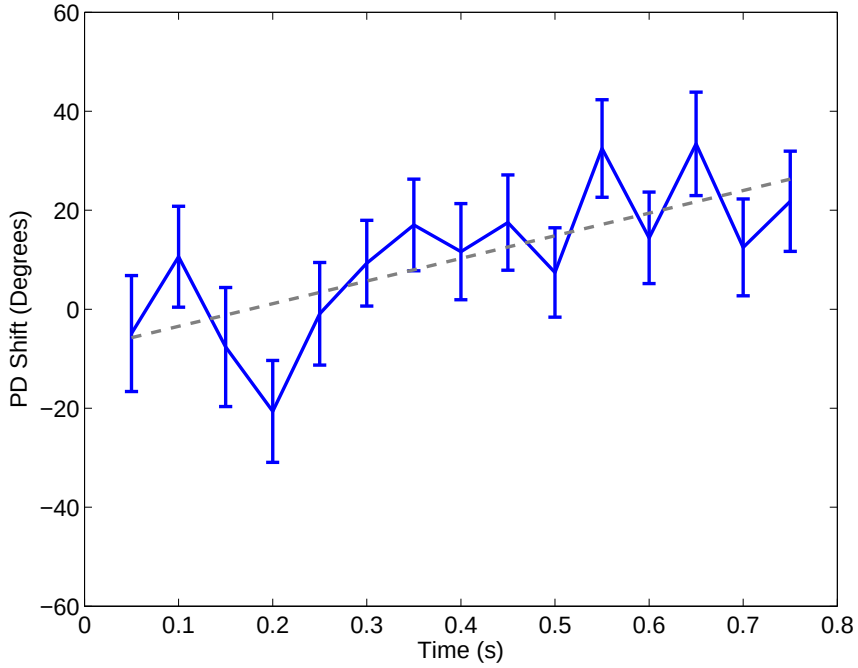

Figure 4.24: *PD Shift Mean as it changes through time for observed MI monkey data. The bars show the standard error in the PD shift distribution at a given time.*

PD changes through time for an entire population of neurons for a model is shown in Figure 4.26. In the figure, only stable PDs are shown. This is the reason that some lines in the figure seem to start later in the trial, or are visible for a short time before disappearing, only to reappear later. The neurons that exhibit this behavior are stable at some points along the trial and unstable at other points. A neuron might be unstable

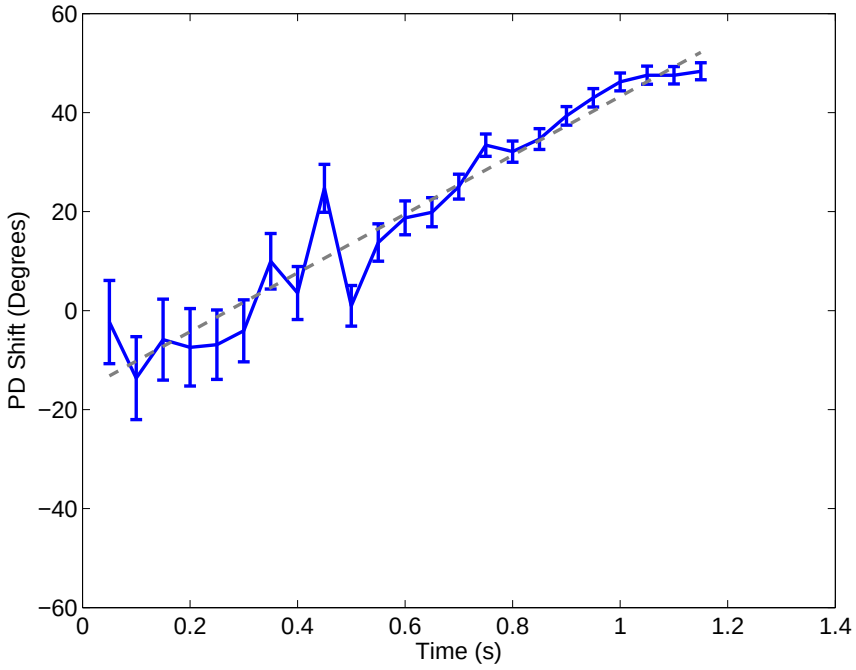

Figure 4.25: *PD Shift Mean as it changes through time for modeled MI data. The bars show the standard error in the PD shift distribution at a given time.*

if it has little or no activity, does not exhibit cosine-like behavior, or does not encode movement direction. Figure 4.26 shows that the distribution of PDs for the neural population at a given forearm posture is largely uniform. There are PDs that occur at many points around the circle. Figure 4.26 also shows that PDs are generally more reliable at the end of the trial than at the beginning of the trial.

Because individual neurons may have differing contributions to muscle activations, it is possible that neurons which behave in a more muscle-like manner are recruited more heavily than other neurons. I hypothesize that a diverse population of neurons are capable of commanding muscles, and therefore this recruitment bias will not be present in the model. A scatter plot of the PD shift of neurons versus the connection strength to muscles for a model is shown in Figure 4.27. There is no clear bias in muscle recruitment as a function of neuron PD shift. This indicates that

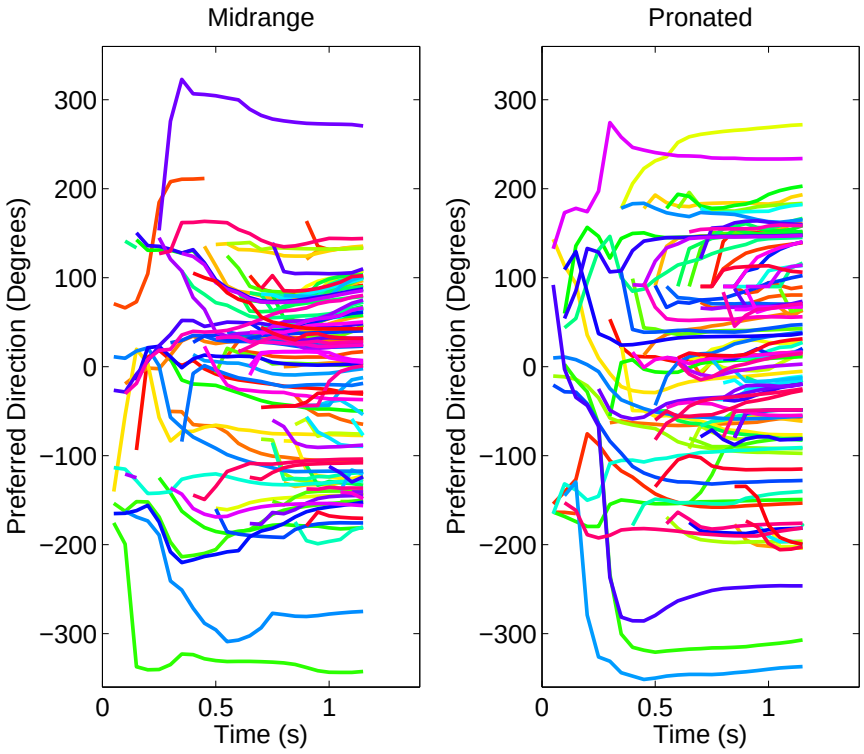

Figure 4.26: *Changes in neuron PD for an entire population of neurons in a single model.*

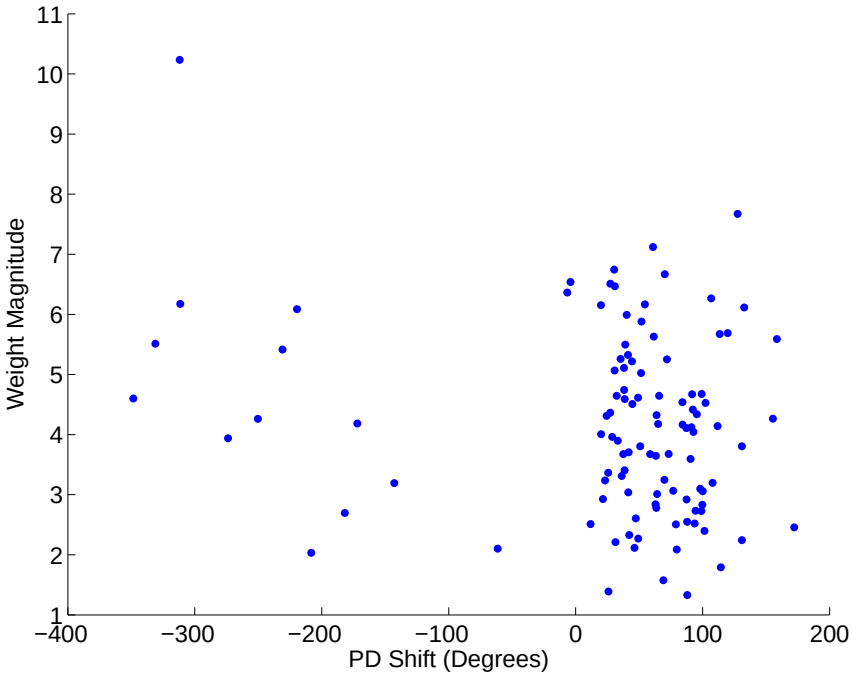

Figure 4.27: *Scatter plot of neuron PD shift versus contribution strength in commanding muscles. PD shift is measured in radians.*

neurons can command muscles in a non-trivial manner even when their PD shifts do not shift like muscles.

4.6 Muscle Analysis

Similar to individual neurons, an individual muscle's behavior can be analyzed. Muscles activiate maximally for a preferred direction, and thus can be fit to cosine tuning curves. The PD changes throughout a trial can then be computed in a similar fashion to neurons.

Most muscles have small PD shifts early in the trial and form a stable PD shift as the trial progresses. This is shown in Figure 4.31 and Figure 4.28, and the muscles have a PD shift of about $60°$ and $80°$, respectively.

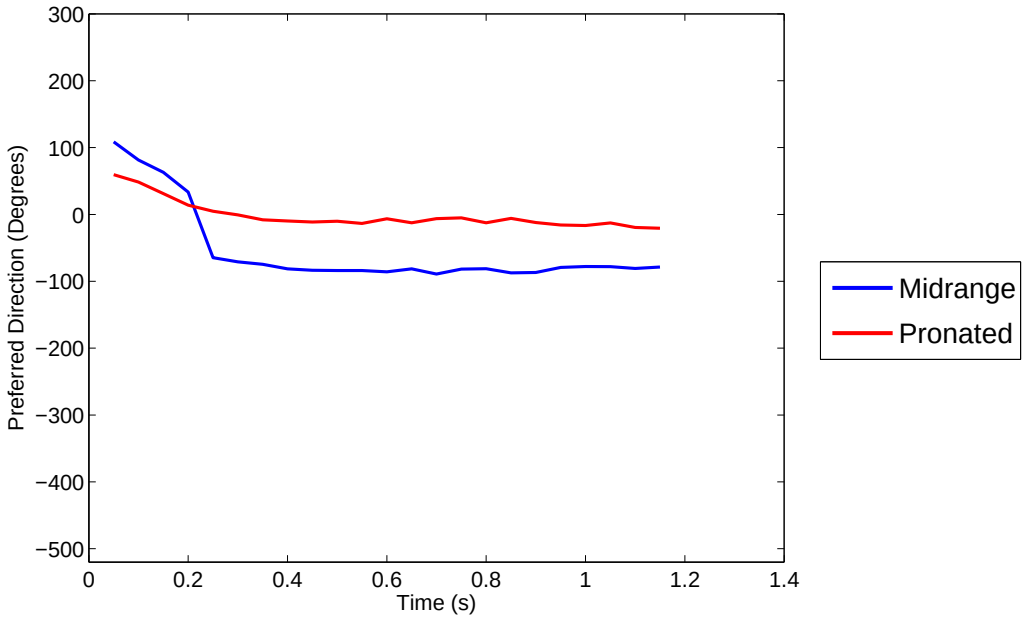

Figure 4.28: *PD change of a muscle through time. The midrange posture is shown in blue and the pronated posture is shown in red.*

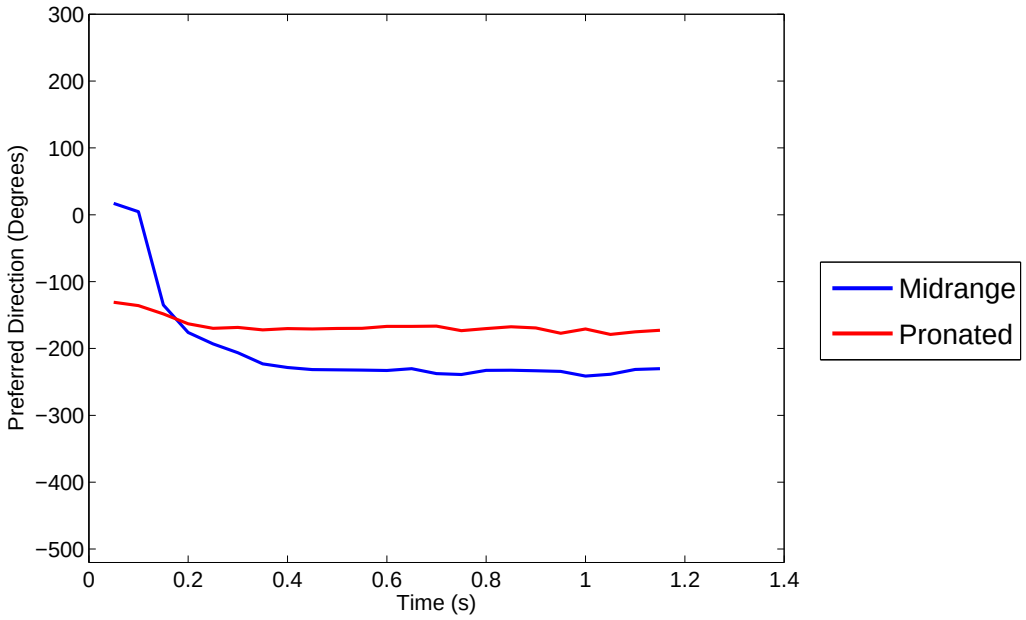

Figure 4.29: *PD change of a muscle through time. The midrange posture is shown in blue and the pronated posture is shown in red.*

Some muscles have substantial PD changes at the beginning of the trial for a given posture, as shown in Figure 4.29. The muscle exhibits large PD changes in the midrange posture at the beginning of the trial, and stabilizes by the end of the trial. Similar to neurons, this large PD change at the beginning of the trial could be a product of the small activations that muscles exhibit at the beginning of the trial, which can lead to poor cosine function fit. Small activation levels can cause PDs to be less reliable than for larger activation levels. Thus, changes in the PD of a muscle at the beginning of the trial may be insignificant.

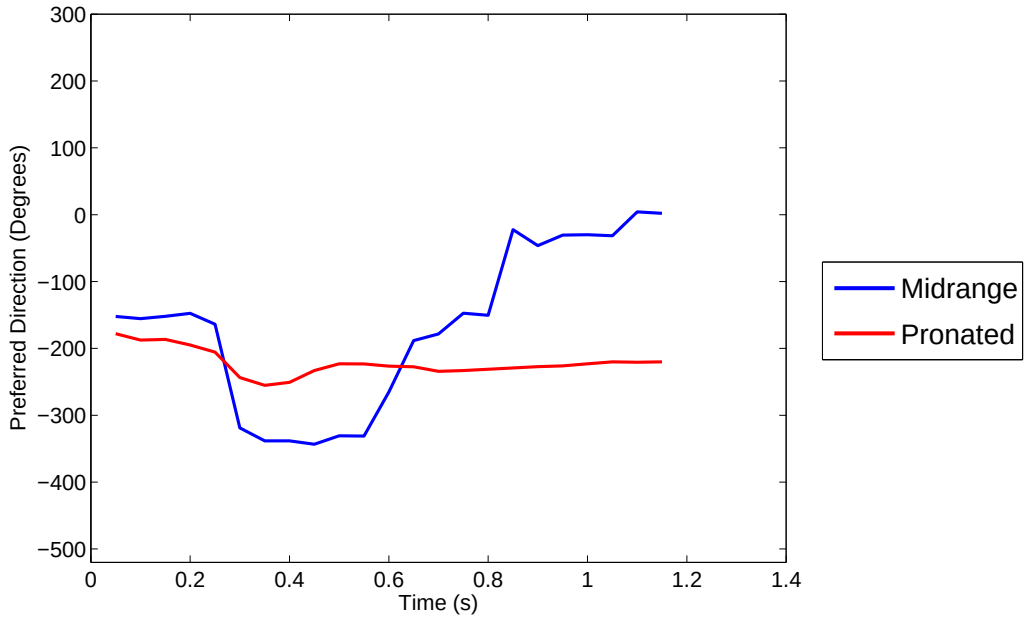

Figure 4.30: *PD change of a muscle through time. The midrange posture is shown in blue and the pronated posture is shown in red.*

Some muscles are generally more active in one posture over another. This causes more unreliable PDs to be formed, and therefore the muscle appears to change its PD greatly for one posture. One such muscle is shown in Figure 4.30. Similar to

neurons, the excess degrees of freedom that muscles have in computing force trajectories in two-dimensional space can allow some muscles to be less active in a particular forearm posture.

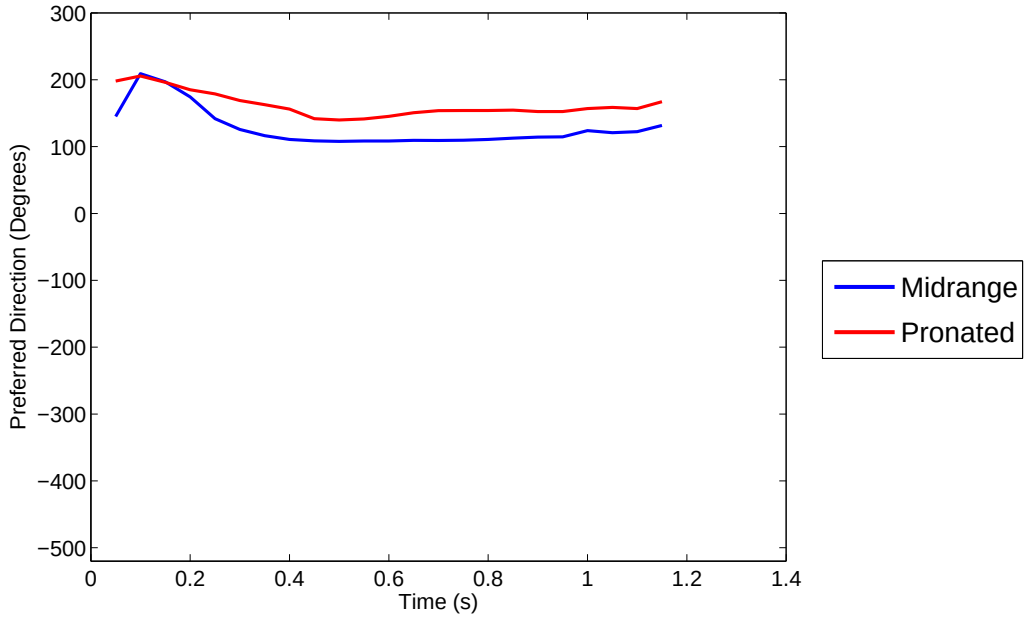

Figure 4.31: *PD change of a muscle through time. The midrange posture is shown in blue and the pronated posture is shown in red.*

A small number of muscles exhibit very small PD shifts as shown in Figure 4.32. While the muscle shown in the figure begins the trial with a substantial PD shift, the PDs stabilize by the middle of the trial and the muscle ends the trial with essentially no PD shift.

Figure 4.33 shows the PD changes over time for the entire muscle population. Muscles exhibit PDs at all points around the circle. Most muscles have an unchanging PD by about the midpoint of the trial. Figure 4.34 shows the PD shift for all muscles. Many muscles exhibit a PD shift of around $70°$, some by forming a PD shift one way

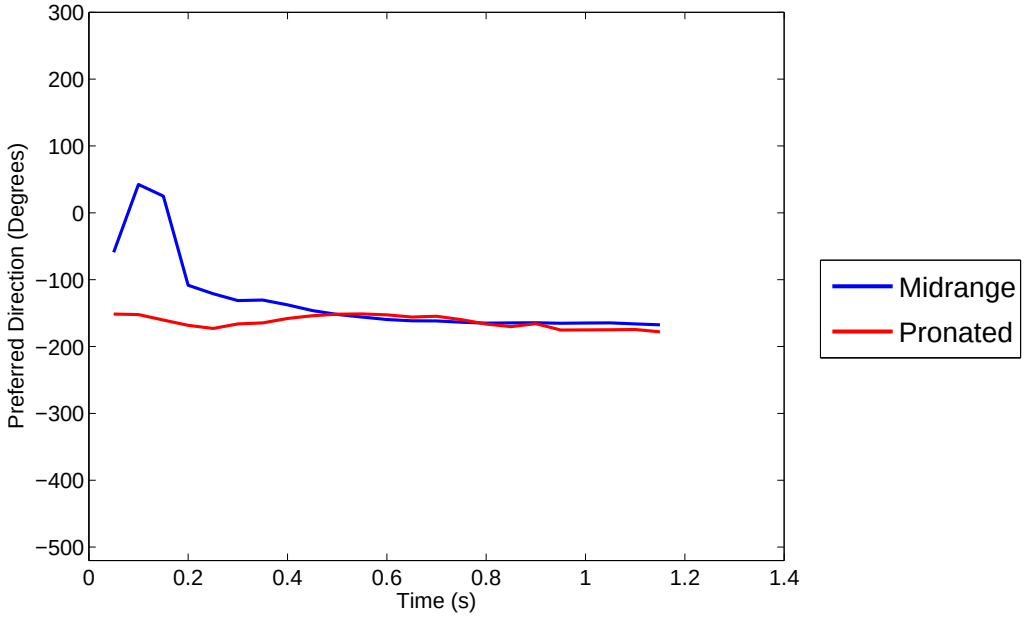

Figure 4.32: *PD change of a muscle through time. The midrange posture is shown in blue and the pronated posture is shown in red.*

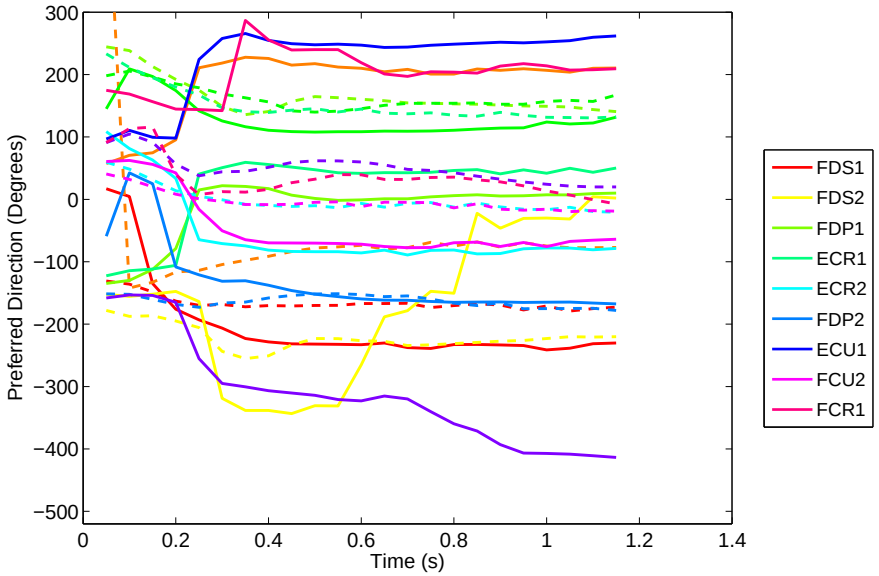

Figure 4.33: *PD change of all muscles through time. The midrange posture is shown with the solid line and the pronated posture is shown with the dotted line.*

around the circle, and other muscles by forming a PD shift the other way around the circle (corresponding to PD shifts of about $290°$.

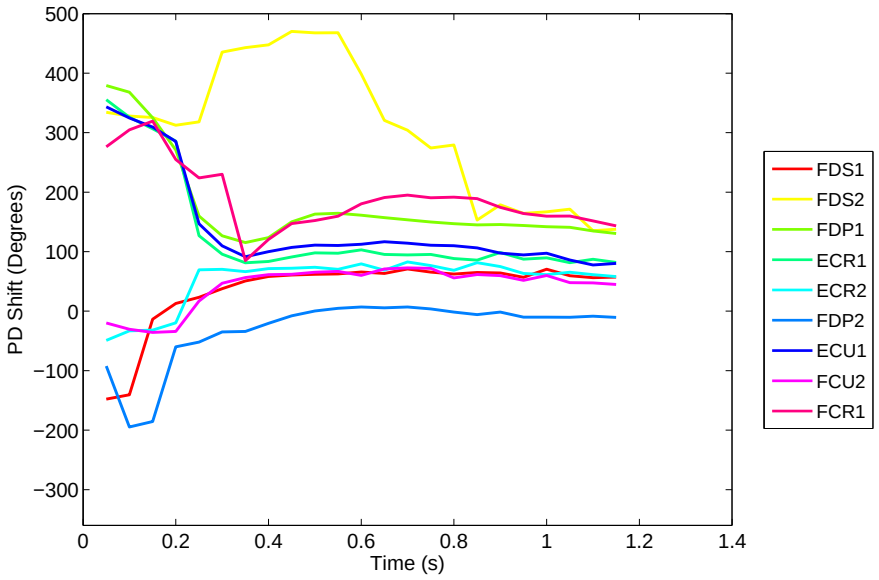

Figure 4.34: *PD shift of all muscles through time.*

Like neurons, muscles compute the transform using an excess of degrees of freedom. Because of this, individual muscles need not contribute to forces in the same way. However, because the vector of contribution for each muscle is held constant (via the $\mathbf{P}^{\rho}$ matrix), and because the size of the muscle population is smaller than the size of the neural population, individual muscles are somewhat more constrained that individual neurons. Therefore, whereas in the neuron population there were a non-trivial number of neurons that showed little or no activity throughout the trial, muscles generally do show activity through the trial.

Similar to the neuron population analysis, we can observe how the PD shift changes through time for the muscle population. The muscle PD shift changes for the observed data is shown in Figure 4.35, and the PD shift changes for the modeled

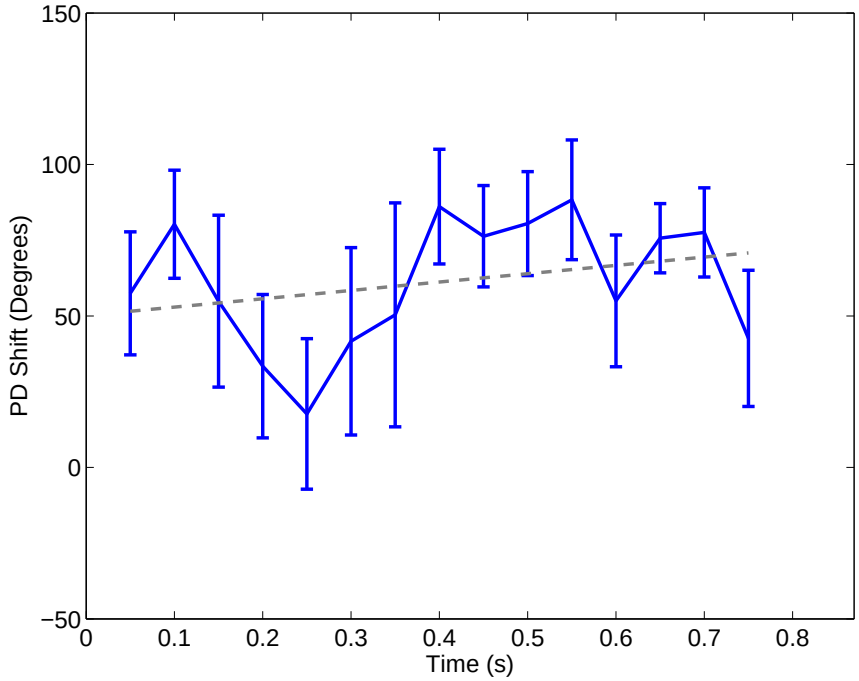

Figure 4.35: *Mean and standard error of PD Shift over time for the observed muscle population.*

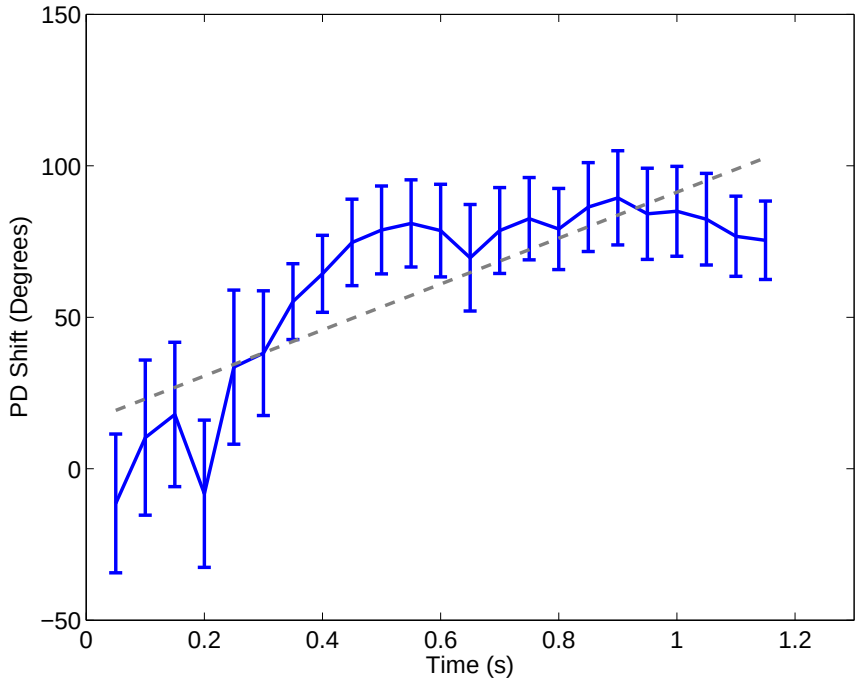

Figure 4.36: *Mean and standard error of PD Shift over time for the modeled muscle population.*

muscle population is shown in Figure 4.36. The modeled muscles change their PD shifts somewhat differently than the observed muscles. The observed muscles have PD shifts that hover around $30^\circ - 70^\circ$ throughout the trial. The modeled muscles start at PD shift changes around -10° and then move towards the PD shifts observed in actual muscles, ending the trial with PD shifts around 70°. This difference could be a product of sampling. Only stable muscles are shown in Figure 4.35 and Figure 4.36.

Because the size of the muscle population is small, the number of muscles included in the figures are small, which could cause unreliable means. In fact, muscles in the modeled population exhibit small depths of modulation at the beginning of the trial, whereas muscles in the monkey experiments do not show small depths of modulation at the beginning of the trial. These small depth of modulations at the beginning of the trial for the modeled population cause PD shifts that start close to 0° at the beginning of the trial. As the trial continues and the depth of modulations increase, PDs become more reliable, and the muscles exhibit PD shifts of about 70°.

The reason that the observed muscles exhibit larger depths of modulation at the beginning of the trial than the modeled population is likely due to natural fluctuations that occur in a real system. The model is capable of producing exactly zero muscle activations at the start of the trial to keep the force at the center, whereas the monkey is unlikely to produce exactly zero muscle activations at the beginning of the trial. Because some muscle activations are produced at the beginning of the trial in the monkey trials, the muscles in the monkey trials have a more clearly defined PD at the beginning of the trial than the modeled muscles.

4.7 Sensitivity Analysis

As mentioned previously, terms were part of the error function in order to prefer solutions with minimal MI and muscle activations. This was done as a means of avoiding solutions that produced overly large and unrealistic neural representations and muscle activations. The specific value of these parameters used in previous analysis were manually chosen so as to provide a good solution.

Increasing the cost of metabolic and muscle activation levels adds additional constraints on the model and prevents certain solutions. Increasing input variation makes the inputs to the model more unreliable. Therefore, intuitively, increasing both the regularization parameters and the visual variation parameter would result in poorer performance in terms of trajectory prediction.

The mean model performance as the metabolic minimization constraint changes is shown in Figure 4.37. Each point is the mean performance of a set of models for which λ_1 is set to a specific value. As λ_1 continues to increase, performance drops off. To illustrate of what happens when the performance drops, a time series for a selected group of trajectories and their predictions from the test set for $\lambda_1 = \frac{1.0}{N}$ is shown in Figure 4.39. The model is able to learn some trial directions well, but not others. In fact, some trials do not have predictions that vary from the center position.

This performance can be compared with the performance of $\lambda_1 = \frac{0.05}{N}$, which is the λ_1 for which, among the values tested, performance is maximized. The time series predictions, for the same group of trajectories shown previously, for $\lambda_1 = \frac{0.05}{N}$ is shown in Figure 4.38. The model is capable of capturing the trajectory in each of

the various directions.

The reason for large λ_1 values forcing predictions close to the center position for some target directions and not others may lie in the neural recruitment process. Large λ_1 values impose great restrictions on neural activity, often forcing many neurons to have little or no activity. The few neurons that show activity are recruited more heavily for certain target directions than others. This could cause those remaining target directions to recruit from a set of neurons with little or no activity, resulting in little to no force.

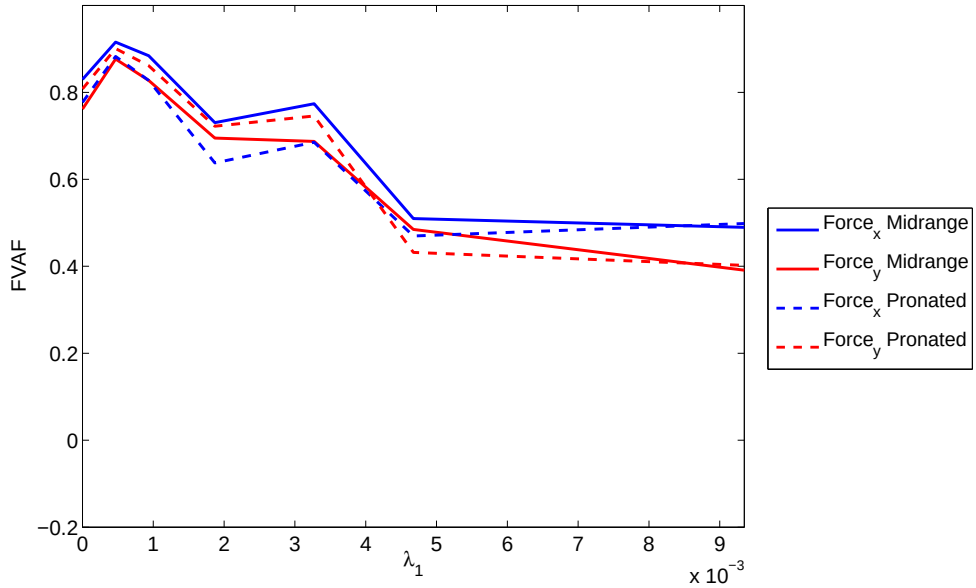

Figure 4.37: *Model trajectory prediction performance measured by FVAF as the metabolic minimization constraint changes. The horizontal axis coresponds to* λ_1. *Visual variation parameter* α *is held constant to* $\alpha = 0.1$. *Muscle minimization constraint* λ_2 *is held constant to* $\lambda_2 = \frac{0.08}{M}$.

The effect of λ_1 on endpoint prediction is shown in Figure 4.40. Similar to Figure 4.37, endpoint prediction error drops as small amounts of regularization is introduced. After a certain point, the prediction errors increase as λ_1 increases. To better understand what this means, Figure 4.41 shows the endpoint predictions where

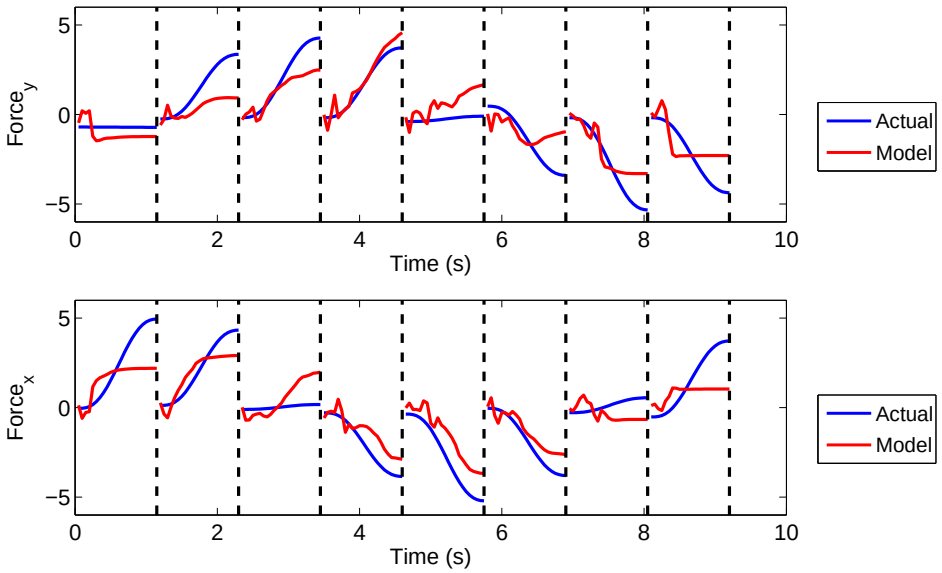

Figure 4.38: *Time series of select force reconstructions for* $\lambda_1 = 0$.

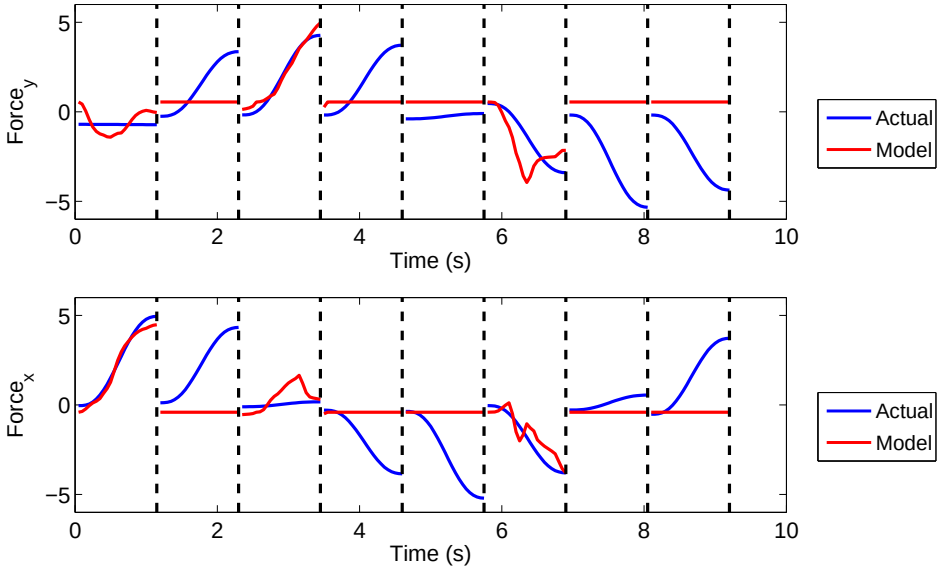

Figure 4.39: *Time series of select force reconstructions for* $\lambda_1 = \frac{1.0}{N}$.

$\lambda_1 = \frac{1.0}{N}$ in force space. While some trial directions have quite small endpoint errors, other directions have large endpoint errors. Most prediction errors are caused by the model undershooting the target, that is, the model tending towards the center position. As discussed earlier, this could be caused by neurons with little or no activity.

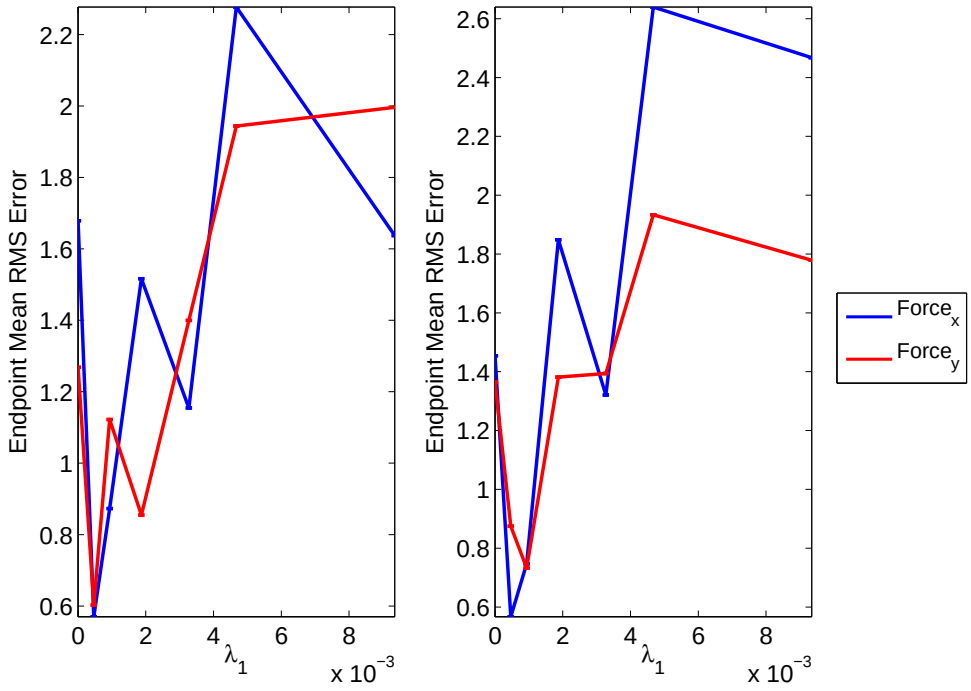

Figure 4.40: *Endpoint prediction errors while λ_1 varies. The horizontal axis measures λ_1. Muscle minimization was held constant at $\lambda_2 = \frac{0.08}{M}$, and the visual variation parameter was held constant at $\alpha = 0.1$.*

On the other end of the performance spectrum, $\lambda_1 = \frac{0.05}{N}$ produces low endpoint errors. The endpoint predictions under these conditions in force space is shown in Figure 4.42. While the errors are much smaller than those of $\lambda_1 = \frac{1.0}{N}$, the source of the error still appears to be undershooting. There is some regularization, albeit small, which can cause some fraction of the neural population to be non-active, resulting in predictions tending towards zero.

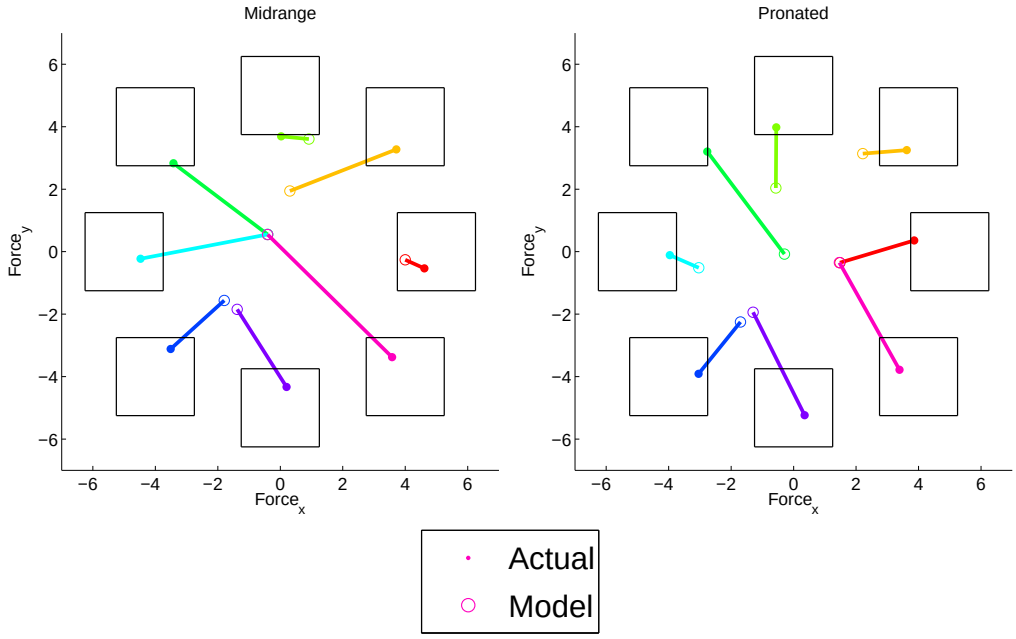

Figure 4.41: *Endpoint prediction errors with* $\lambda_1 = \frac{1.0}{N}$ *in force space. Lines are drawn between the predicted endpoints and their corresponding observed endpoints. The left graph shows the midrange posture and the right graph shows the pronated posture.*

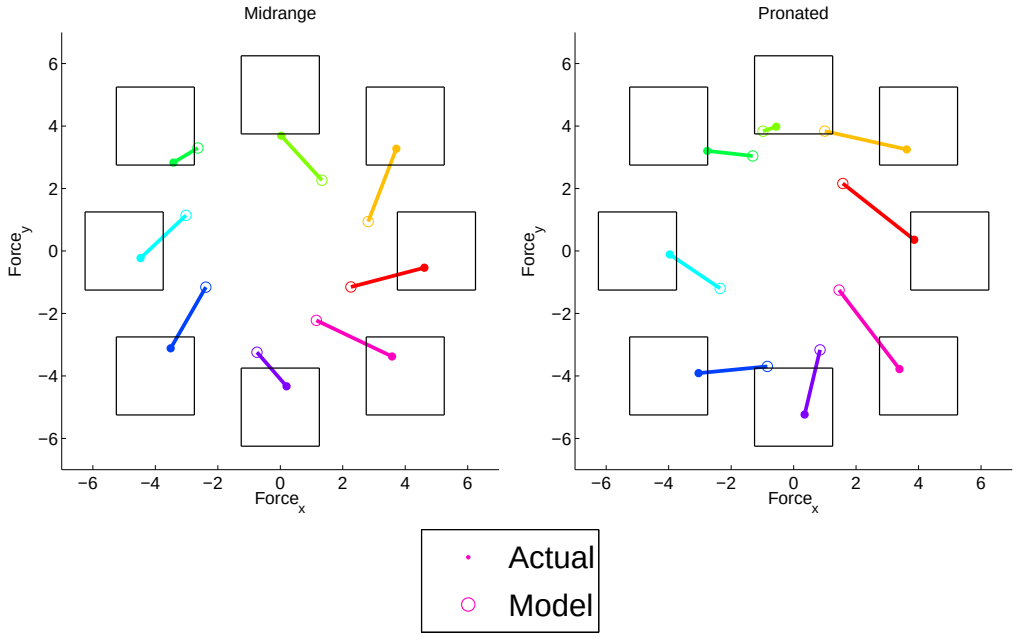

Figure 4.42: *Endpoint prediction errors with* $\lambda_1 = \frac{0.05}{N}$ *in force space. Lines are drawn between the predicted endpoints and their corresponding observed endpoints. The left graph shows the midrange posture and the right graph shows the pronated posture.*

In addition to model performance, the model parameters can also affect neural behavior. In particular, what affect does λ_1 have on the model's ability to form well-tuned neurons? The tuning curves are defined to be a cosine of the form $f(\theta) = a + b\ cos(\theta_{PD} - \theta)$, where a is the baseline activation, b is the depth of modulation, and θ_{PD} is the preferred direction of the neuron. To measure the ability to form tuning curves, we measure the root mean squared error values of a best-fit cosine function against a neuron's activations. This error is then divided by the depth of modulation of the neuron, in order normalize the error across various ranges of modulation. Dividing by the depth of modulation prevents neurons with large activations from overshadowing neurons with smaller activations. Thus, the *cosine error* $C = \frac{RMS\ Error}{b}$. Thus, a number representing the mean "cosine error" of a population is computed. This cosine error as a function of the metabolic minimization constraint λ_1 is shown in Figure 4.43. Similar to previous analysis, this mean cosine error is averaged over all models by which λ_1 is the specified value.

As the metabolic minimization criterion λ_1 is increased, the mean cosine error decreases. This implies that the metabolic minimization criterion results in neurons that are better tuned than without this cost. By increasing λ_1, the model is forced to reduce neural activation levels except when needed to compute the task. This results in neurons that are well modulated to certain target directions.

The effects of the muscle minimization criterion on model trajectory prediction performance are shown in Figure 4.44. The muscle minimization criterion does not have as much effect on modle performance as does the metabolic minimization criterion.

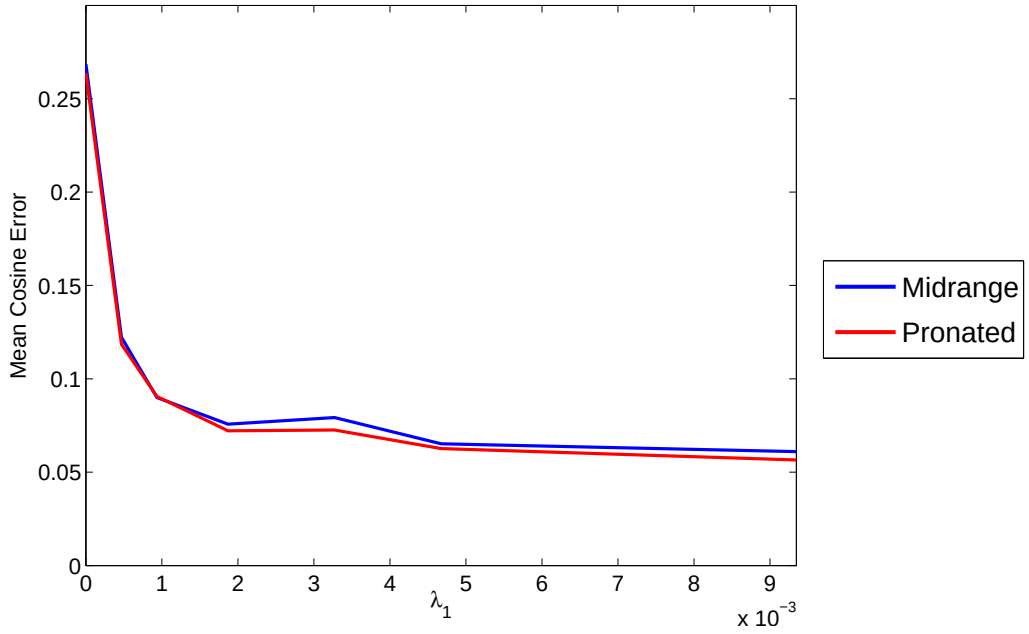

Figure 4.43: *Mean error of the cosine tuning curve to the neuron activations for a range of λ_1 choices.*

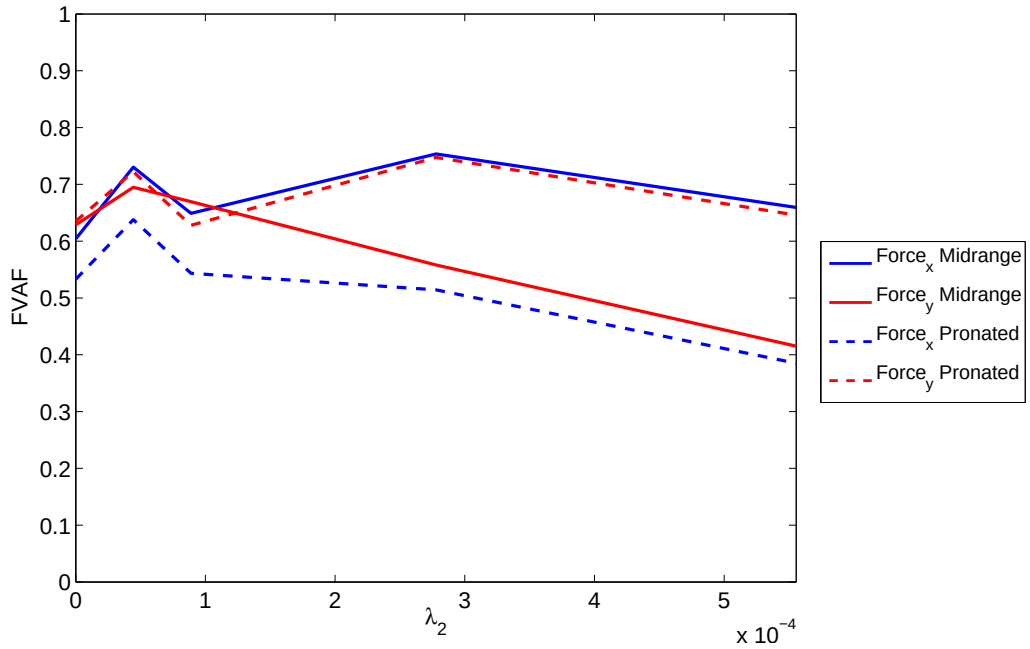

Figure 4.44: *Model trajectory prediction performance measured by FVAF as the muscle minimization constraint changes. The horizontal axis coresponds to λ_2. Visual variation parameter α is held constant to $\alpha = 0.1$. Metabolic minimization constraint λ_1 is held constant to $\lambda_1 = \frac{0.05}{N}$.*

Muscle effort cost also affects endpoint prediction errors. Effects of muscle minimization parameter λ_2 on endpoint prediction error is shown in Figure 4.45. Introducing muscle minimization does not have a clear affect on endpoint prediction performance. It is interesting to note, however, that when no muscle cost was introduced, the model was unable to perform well in the midrange posture. To illustrate this, Figure 4.46 shows the endpoint error in force space for $\lambda_2 = 0$, and Figure 4.47 shows the endpoint error in force space for $\lambda_2 = \frac{1.0}{M}$. When $\lambda_2 = 0$, the models are unable to generalize across postures, and loses predictive capabilities for the midrange posture. The model is able to learn well on the pronated posture, however. When $\lambda_2 = \frac{1.0}{M}$, performance suffers in the pronated posture, however performance gains are observable in the midrange posture such that the model performs approximately equally well in both postures. The introduction of a muscle minimization constraint appears to improve generalization across postures. This could be because generalization occurs with fewer nonzero muscle activations, which the muscle minimization constraint aims to provide.

Muscle minimization can also affect the ability to form tuning curves for neurons. While neural activations are not affected by muscle activations in the forward propagation of the network model, during training of the model, connection weights to neurons can be updated so as to compensate for any restrictions placed on the muscle activations. The mean cosine error as a function of the muscle minimization parameter is shown in Figure 4.48.

The muscle minimization constraint does not greatly affect the cosine error of the modeled neurons. Since muscle activations only affect neural activations indirectly

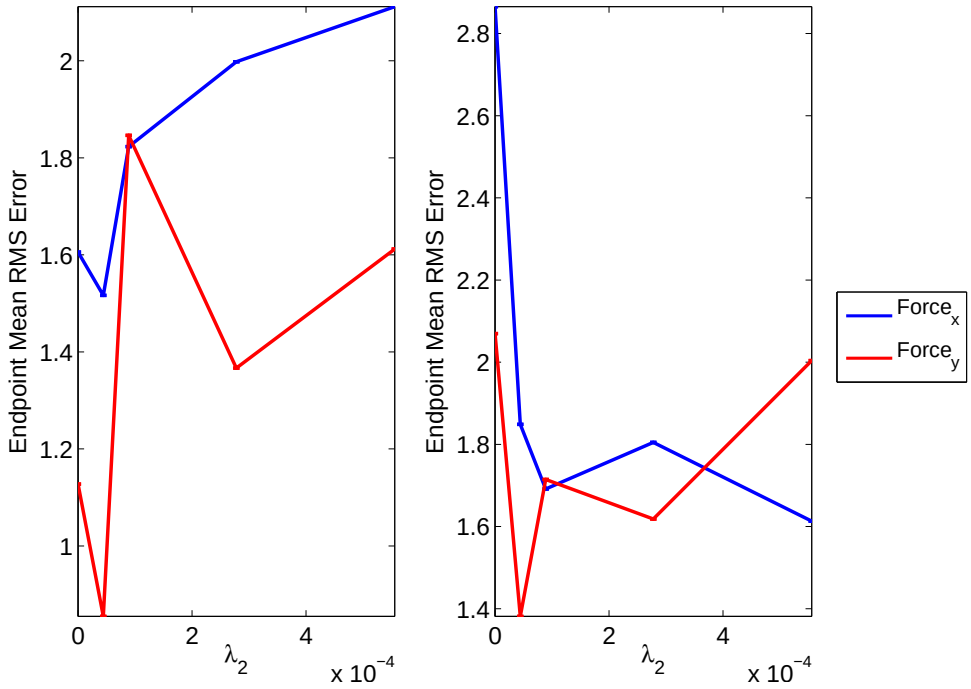

Figure 4.45: *Endpoint prediction errors. The horizontal axis measures λ_2. Metabolic minimization was held constant at $\lambda_1 = \frac{0.05}{N}$, and the visual variation parameter was held constant at $\alpha = 0.1$.*

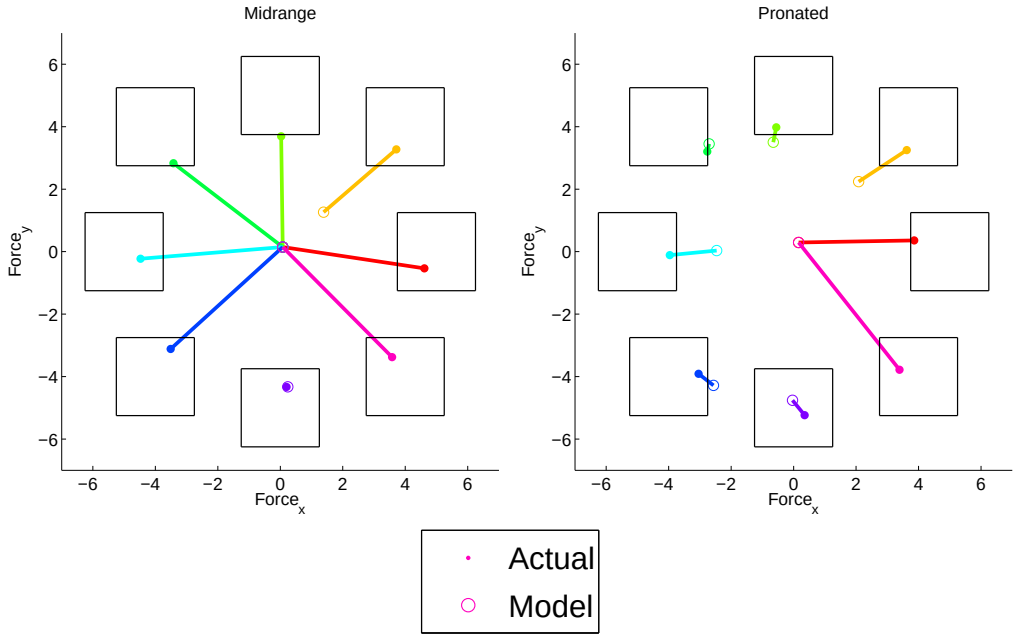

Figure 4.46: *Endpoint prediction errors with $\lambda_2 = 0$ in force space. Lines are drawn between the predicted endpoints and their corresponding observed endpoints. The left graph shows the midrange posture and the right graph shows the pronated posture.*

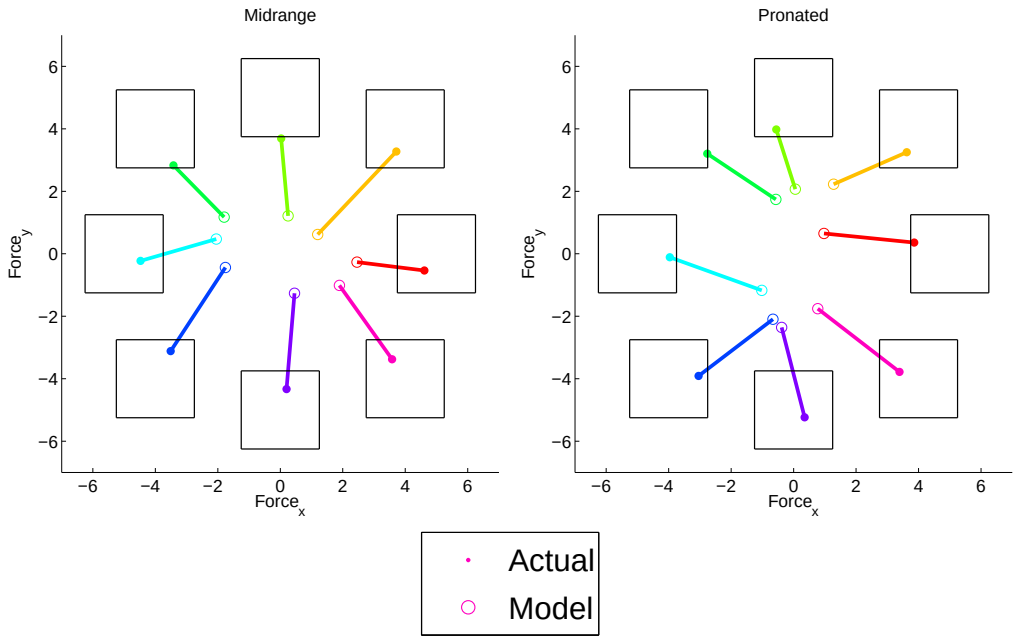

Figure 4.47: *Endpoint prediction errors with* $\lambda_2 = \frac{1.0}{M}$ *in force space. Lines are drawn between the predicted endpoints and their corresponding observed endpoints. The left graph shows the midrange posture and the right graph shows the pronated posture.*

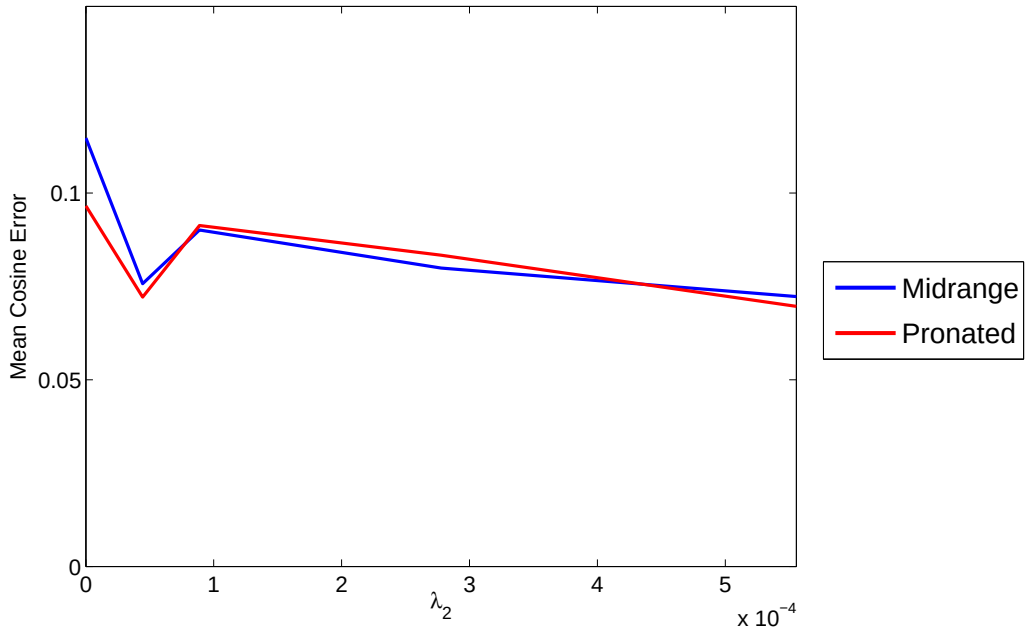

Figure 4.48: *Mean error of the cosine tuning curve to the neuron activations for various* λ_2 *values.*

through the training process, this result is not surprising. Further, as the muscle minimization constraint increases, the cosine error decreases slightly.

Another hyperparameter in the model is the amount of visual variation present. In a highly dynamic system such as the central nervous system, which is capable of performing many tasks at once, some amount of variation in the encoding of information is expected. Visual variation in the model can simulate variation that would present in a real system (Harris and Wolpert, 1998).

The effects of the visual variation parameter on model trajectory prediction performance is shown in Figure 4.49. Increasing levels of visual variation detrimentally affects model performance. Large α values for the variation parameter correspond to very large variation levels. For example, from Equation **??**, a variation level of $\alpha = 1$ corresponds to variation levels equaling that of the original signal. Concretely, this means that given a signal value $s_i = a$, the resulting value with a variation parameter of $\alpha = 1$ is $s_i = a \pm a$. This would obviously be an unreliable signal and therefore be difficult to form a reliable model. So, as expected, model performance under this condition is poor. No visual variation provides a perfectly reliable signal, and therefore we expect the model to perform well under theses conditions. Figure 4.49 shows that the model indeed performs well with no input variation, and in fact performs best when no noise is present. Such a system is unrealistic, as visual variation is often present in a real system.

Endpoint prediction errors are also affected by the visual variation parameter. Effects of the visual variation parameter α on endpoint prediction error are shown in Figure 4.50. Increasing α results in an increase in the endpoint RMS error. Large

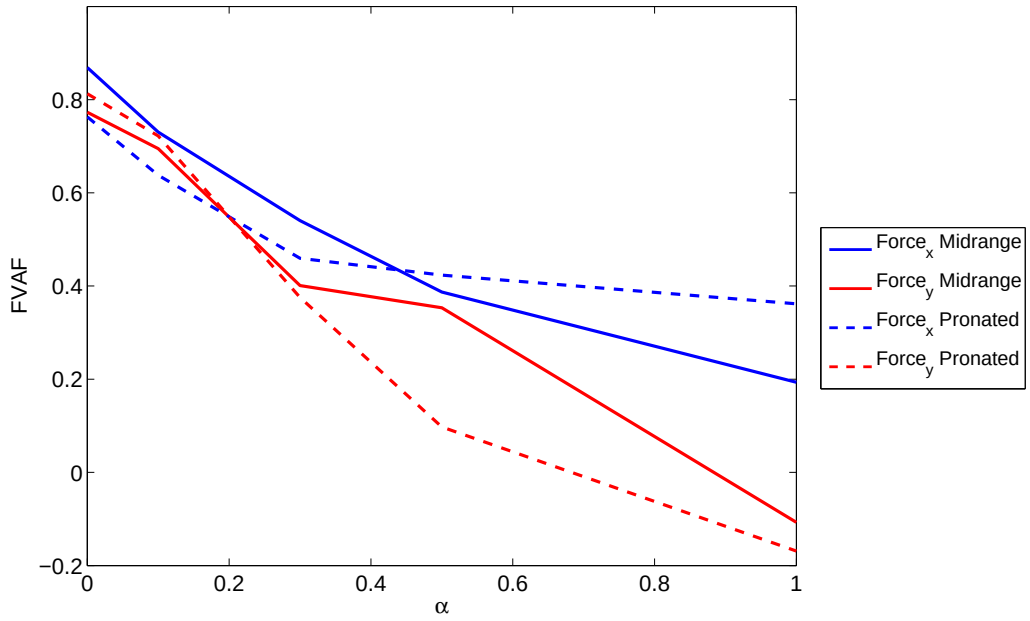

Figure 4.49: *Model trajectory prediction performance measured by FVAF as the visual variation parameter changes. The horizontal axis coresponds to the visual variation parameter*α*. Metabolic minimization constraint* λ_1 *is held constant to* $\lambda_1 = \frac{0.05}{N}$*. Muscle minimization constraint* λ_2 *is held constant to* $\lambda_2 = \frac{0.08}{M}$*.*

variation values produced large endpoint errors in both the midrange and pronated postures.

The visual variation parameter can also affect the ability to form tuning curves for neurons. The mean cosine error as a function of visual variation parameter is shown in Figure 4.51. The introduction of visual variation increases the cosine error of the modeled neuron population. This is intuitive, because adding variation to the input would lead to variation in the tuning. The increase in cosine tuning errors are fairly modest as compared with the metabolic minimization criterion.

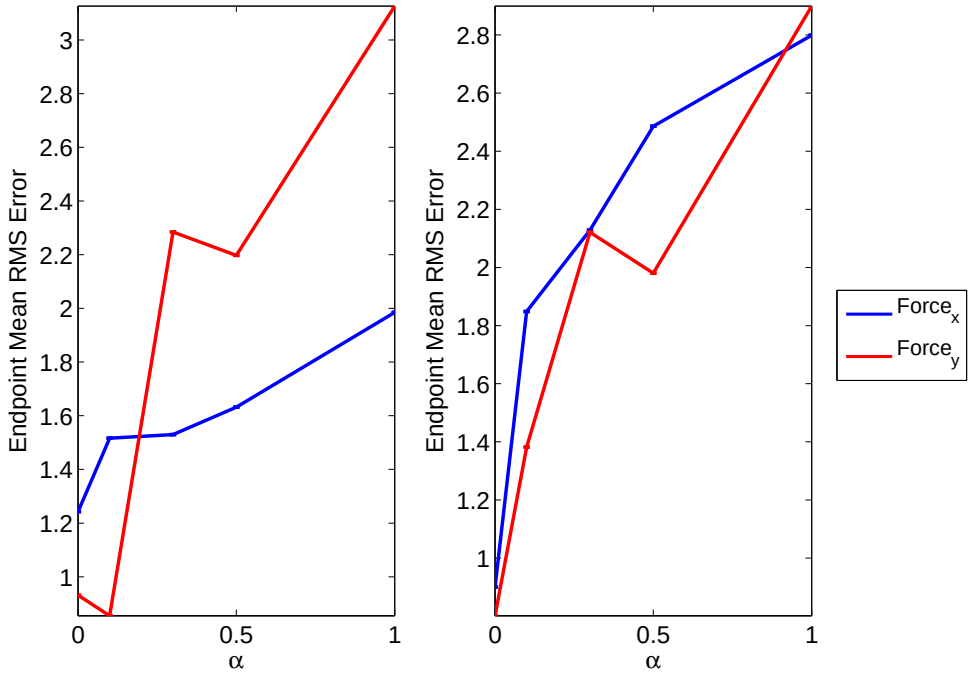

Figure 4.50: *Endpoint prediction errors. Muscle minimization was held constant at* $\lambda_2 = \frac{0.08}{M}$*, and the metabolic minimization paramete was held constant at* $\lambda = \frac{0.05}{N}$.

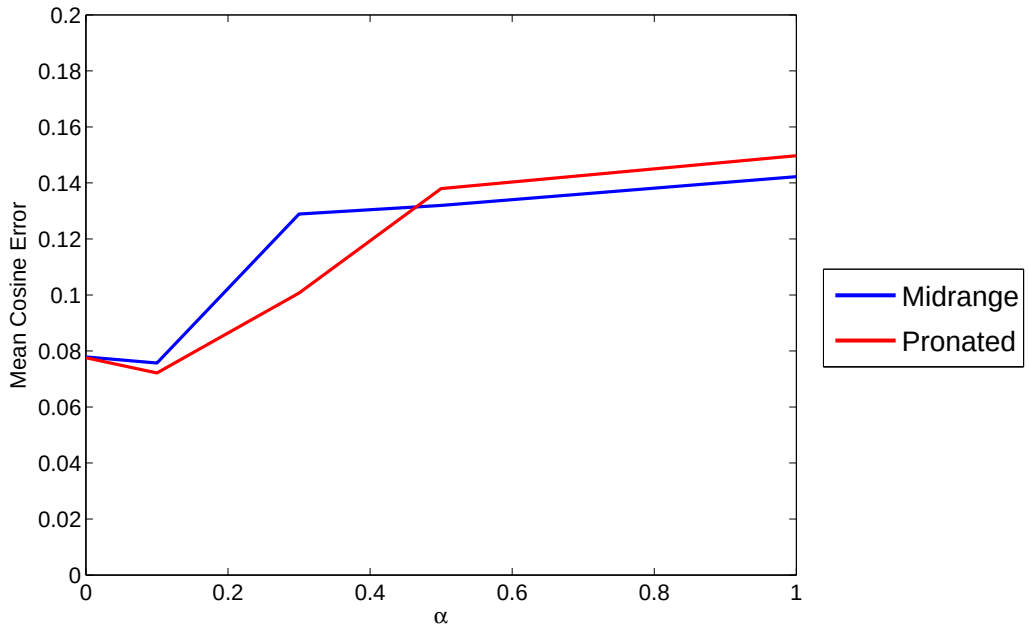

Figure 4.51: *Mean error of the cosine tuning curve to the neuron activations for various* α *values.*

Chapter 5

Conclusion

In this thesis, I proposed a new model of MI neuron recruitment in an isometric hand force task whose aim was to explain the task-dependent neural activation patterns that have been observed in monkey MI. The model is structured as a recurrent neural model that transforms an extrinsic encoding of a target force in a two-dimensional task space as well as an input describing forearm posture into intrinsic muscle activation patterns. Modeled MI neurons directly drove muscles, and the muscles in turn imposed forces at the hand in a posture-dependent manner. The model parameters were selected using a gradient descent approach that sought to minimize force trajectory errors while imposing minimization constraints on the muscle and neural activation patterns. I showed that the model is capable of making the transformation from extrinsic visual input to intrinsic muscle activations, and is able to do so with changes to wrist posture. Through an analysis of the model, I further show that the behavior of the neurons and muscles in the model is similar in structure to those seen in the central nervous system. Specifically, I show that the modeled neurons are directionally tuned, and the mean of the preferred direction shift between

postures is very similar to the mean observed in monkey experiments.

The experimental results show that the minimization constraints placed on the model, as well as the implicit constraints of the model itself (the model is tuned to perform an isometric center-out wrist task) affect both the ability of the model to perform the task and the structure of the neural behavior. The model achieves high quality force trajectory generation with cosine-like tuning properties of the MI cells. Furthermore, the metabolic energy cost is key to the cosine-like shapes. Without this cost, the modeled MI neurons were free to respond in much more arbitrary ways.

This work shows that even if the input and output representations of MI were well defined (extrinsic and muscle-like, respectively), the intermediate representation as implemented by MI need not be one that follows a well-defined pattern on a neuron-by-neuron basis (Fetz, 1992). Specifically, the change of preferred direction of a neuron with a change in posture need not shift in a manner that directly reflects the associated change in pulling direction by the muscles. Instead, it is the aggregate effect of the set of neurons that is key. Because there are so many more neural degrees of freedom than muscular, the nervous system has the freedom to make arbitrary, but constrained, choices for how individual neurons are recruited.

The PD shifts observed in the model varied dramatically from neuron to neuron, with some showing almost no shift, others showing muscle-like shifts, and others showing shifts in between zero and muscle-like shifts. The mean of this PD shift distribution is similar to what is seen in monkey MI. These means fall substantially short of the PD shift that is exhibited by the muscles. How do neurons that exhibit

these "muted" PD shifts properly drive muscles? One possible factor is the posture-dependent change in the depth of modulation that many of the modeled neurons exhibit. In fact, Shah et al. (2004) showed that a representation with **no** PD shift, but with a posture-dependent change in the depth of modulation was sufficient to enable a high-performing linear transformation from neuron to muscle. While the MI behavior is more general in the current model, many neurons exhibit changes in their depth of modulation, a factor that the model can take advantage of in selecting an appropriate set of connection parameters. I hypothesize that if the modeled neurons were constrained to minimize these changes in the depth of modulation, then PD shifts would have to be more dramatic to accommodate for this degree of freedom loss.

Much future work can be done with the proposed model. For example, while the model was able to recreate the PD shift distribution mean quite well, an interesting question: why the PD shift distribution created by the central nervous system is distinctively wider than that created by the model. One could begin to answer this question by asking what purpose do neurons in the distribution tail serve in computing the various transformations. A hypothesis would be that these neurons in fact do not directly serve to compute the various transformations. The neurons in MI can perform many different tasks, whereas the model presented in this thesis was constrained to computing the specific isometric center-out task. It is quite possible that this implicit constraint of forcing all neurons to participate in task completion may yield the narrower PD shift distribution.

One key difference in the modeled task and that of Oby et al. (2012) is the fact

that no instruction period is included in the modeled experiment. Instead, the target stimulus is presented at the same time as movement is to be initiated. This fact could explain the gradual increase in neural activity from the beginning of the trial and the PD shifts of many of the neurons and muscles that start near zero degrees. If an explicit delay were inserted between the presentation of the target stimulus and the go signal, I expect a congruence in the PD shifts during the movement period between the model and monkey data.

Finally, the model demonstrates differences in MI activation patterns across different postures, which manifest themselves in PD shifts, changes in the depth of modulation, and changes in timing. Likewise, should the model be expected to perform one of several different tasks, I hypothesize that the MI neurons will also show task-dependent changes in their tuning properties in each of these same dimensions. Separating these tasks out in the central nervous system is difficult, and activations that don't correspond to a particular task are usually considered to be noise. This "noise" may be the main source of the differences in representation between my model and the central nervous system.

Bibliography

Ajemian, R., Bullock, D., and Grossberg, S. (2001). A model of movement coordinates in the motor cortex: Posture-dependent changes in the gain and direction of single cell tuning curves. *Cerebral Cortex*, 11:1124–1135.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Springer, New York, NY.

Dum, R. P. and Strick, P. L. (2002). Motor areas in the frontal lobe of the primate. *Physiology & Behavior*, 77:677–682.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

Fagg, A. H., Shah, A., and Barto, A. G. (2002). A computational model of muscle recruitment for wrist movements. *Journal of Neurophysiology*, 88(6):3348–3358.

Fetz, E. E. (1992). Are movement parameters recognizably coded in the activity of single neurons? *Behavioral and Brain Sciences*, 15(04):679–690.

Funahashi, K.-i. (1998). Multilayer neural networks and bayes decision theory. *Neural Networks*, 11(2):209–213.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., and Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2:1527–1537.

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

Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determins motor planning. *Nature*, 394:780–784.

Hoffman, D. S. and Strick, P. L. (1999). Step-tracking movements of the wrist. IV. Muscle activity associated with movements in different directions. *Journal of Neurophysiology*, 81(1):319–333.

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

Kakei, S., Hoffman, D. S., and Strick, P. L. (2003). Sensorimotor transformations in cortical motor areas. *Neuroscience Research*, 46:1–10.

Kalaska, J. F., Cohen, D. A. D., Hyde, M. L., and Prud'Homme, M. (1989). A comparison of movement direction-related versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task. *The Journal of Neuroscience*, 9:2080–2102.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1137–1145.

Le Cun, B. B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer.

Mussa-Ivaldi, F. (1988). Do neurons in the motor cortex encode movement direction? an alternative hypothesis. *Neuroscience Letters*, 91(1):106–111.

Oby, E. R., Ethier, C., and Miller, L. E. (2012). Movement representation in primary motor cortex and its contribution to generalizable emg predictions. *Journal of Neurophysiology*, 109:666–678.

Pham, D. and Liu, X. (1996). Training of Elman networks and dynamic system modelling. *International Journal of Systems Science*, 27(2):221–226.

Scott, S. H. and Kalaska, J. F. (1997). Reaching movements with similar hand paths but different arm orientations .1. activity of individual cells in motor cortex. *Journal of Neurophysiology*, 77(2):826–852.

Sergio, L. E. and Kalaska, J. F. (1998). Changes in the temporal pattern of primary motor cortex activity in a directional isometric force versus limb movement task. *Journal of Neurophysiology*, 80(3):1577–1583.

Shah, A., Fagg, A. H., and Barto, A. G. (2004). Cortical involvement in the recruitment of wrist muscles. *Journal of neurophysiology*, 91(6):2445–2456.

Stevenson, I. H., Cherian, A., London, B. M., Sachs, N. A., Lindberg, E., Reimer, J., Slutzky, M. W., Hatsopoulos, N. G., Miller, L. E., and Kording, K. P. (2011). Statistical assessment of the stability of neural movement representations. *Journal of Neurophysiology*, 106:764–774.