

FROM CORTICAL NEURAL SPIKE TRAINS TO BEHAVIOR: MODELING AND
ANALYSIS

By

JUSTIN CORT SANCHEZ

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2004

Copyright 2004

by

Justin Cort Sanchez

This dissertation is dedicated to my father. His sense of practicality and his love of biology has inspired me to become the Biomedical Engineer that I am today.

ACKNOWLEDGMENTS

The path to achieving a doctorate is like a rollercoaster ride. Some days you are up and some days you are down, and other times you just feel like screaming. All along the ride though, my wife Karen has been strapped in the seat right next to me. Never in my life has a person brought such a sense of calmness and balance. Not only does she make everything better, but she always brings out the best in me.

Operating this rollercoaster ride were my committee members. They made sure that I was safely strapped in and prepared to ride. When I started my graduate studies on Brain-Machine interfaces, I knew nothing about signal processing. With their infinite patience and guidance, I was given the opportunity to grow as a person and as a researcher. The chair of my committee, Dr. Jose Principe, inspired me to think big and sent me all over the world to meet the leaders in the field. I hope that one day I will be able to give back the gifts and opportunities that my committee members have so graciously given me.

I would like to thank my family for giving me the financial and emotional support to ride this rollercoaster. They have always believed in me. They have always supported me. They have sacrificed themselves for me. They taught me that when everything is said and done, your family will always be there for you.

Along for the ride were Deniz, Ken, and Yadu who cheered me on along the way. There is a saying that you are only as good as the people around you and I believe that all of them made me a better person both in terms of friendship and signal processing. I will

never forget discussions we had during our 14-hour ride through the Canadian Rockies. I thank them for all of their help and insight over the years. I also cannot forget Phil who was involved in this research from the beginning, Scott Morrison for the long hours with the DSP, and Shalom for always adding a bit of humor to all of the smoke and mirrors.

Last but not least I need to thank Brian Whitten. Throughout our 12-year friendship in Tampa and Gainesville we always thought that we would end up as rock stars. We had some great times getting away from work and playing shows at all of the smoky bars in Florida. In the end though, he became a Veterinarian and I became a Biomedical Engineer; how unlikely!

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	xi
ABSTRACT	xv
CHAPTER	
1 INTRODUCTION	1
Historical Overview of BMI Modeling Approaches	3
Foundations of Neuronal Recordings	6
Characteristics of Neuronal Activity	7
Local Neuronal Correlations	10
2 DUKE BRAIN-MACHINE INTERFACE PARADIGM	12
Neuronal Recording Methodology	12
Behavioral Experiments	16
3 MODELING PROBLEM	19
What Mapping Does the Model Have to Find?	19
Signal Processing Approaches to Modeling	20
White Box	20
Gray Box	20
Population vector algorithm	21
Todorov's mechanistic model	22
Implementation of "gray box" models	24
Black Box	27
Finite impulse response filter	28
Time-delay neural network	29
Recurrent multilayer perceptron	31
Development and Testing of BMI Models	32
Reaching Task Performance	32
Topology and training complexity comparisons	33

	Regularization, weight decay, and cross validation	38
	Performance metrics	40
	Test set performance	42
	Cursor Control Task	48
	Discussion	52
4	RMLP MODEL GENERALIZATION	55
	Motivation for Studying the RMLP	55
	Motivation for Quantifying Model Generalization	58
	Multi-Session Model Generalization	59
	Multi-Task Model Generalization	65
	Discussion	70
5	ANALYSIS OF THE NEURAL TO MOTOR REPRESENTATION SPACE CONSTRUCTED BY THE RMLP	73
	Introduction	73
	Understanding the RMLP Mapping – Network Organization	74
	Understanding the Mapping	74
	Input Layer Weights	77
	Output Layer Weights	80
	Cursor Control Mapping	83
	Discussion	85
6	INTERPRETING CORTICAL CONTRIBUTIONS THROUGH TRAINED BMI MODELS	89
	Introduction	89
	Cortices Involved in Hand Reaching	90
	Belle’s Cortical Contributions	90
	Carmen’s Cortical Contributions	92
	Cortices Involved in Cursor Tracking	95
	Discussion	97
7	ASCERTAINING THE IMPORTANCE OF NEURONS	100
	Introduction	100
	Assumptions for Ranking the Importance of a Neuron	102
	Sensitivity Analysis for Reaching Tasks	103
	Cellular Importance for Cursor Control Tasks	111
	Model-Dependent – Sensitivity Analysis	111
	Model-Independent – Cellular Tuning Analysis	113
	Relative Ranking of Neurons	118
	Model Performance with a Subset of Sensitive Cells	119
	Implications of Sensitivity Analysis for Model Generalization	121
	Discussion	123

8	CONCLUSIONS AND FUTURE DIRECTIONS	128
	Summary of Contributions	131
	Difficulties Encountered in BMI Research.....	131
	Future Directions	133
	Final Thoughts	134
APPENDIX		
A	TRAINING THE RMLP	135
	Introduction.....	135
	Topology and Trajectory Learning.....	135
	Monte Carlo Simulations.....	137
B	EVALUATION OF THE RMLP MODELING PERFORMANCE USING SPIKE-SORTED AND NON-SPIKE-SORTED NEURONAL FIRING PATTERNS.....	140
	Introduction.....	140
	Data Preparation	140
	Simulations	141
	Discussion and Conclusion.....	144
C	MODEL EXCITATION WITH RANDOM INPUTS.....	145
	LIST OF REFERENCES.....	148
	BIOGRAPHICAL SKETCH	155

LIST OF TABLES

<u>Table</u>	<u>page</u>
1-1. Neuronal activity for a 25-minute recording session.....	10
2-1. Assignment of electrode arrays to cortical regions for owl monkeys	14
2-2. Assignment of electrode arrays to cortical regions for Rhesus monkeys.....	14
3-1. Model parameters	37
3-2. Model computational complexity.....	38
3-3. Reaching task testing CC and SER (Belle)	45
3-4. Reaching task testing CC and SER (Carmen)	47
3-5. Reaching task testing CC and SER (Ivy).....	50
3-6. Reaching task testing CC and SER (Aurora).....	50
4-1. Scenario 1: Significant decreases in correlation between sessions	64
4-2. Scenario 2: Significant increases in correlation between sessions	64
4-3. Multi-task testing CC and SER values	68
4-4. Significance of CC compared to FIR filter.....	68
6-1. Summary of cortical assignments.....	92
7-1. The 10 top ranked cells.....	118
7-2. Test set correlation coefficients using the full ensemble.....	120
7-3. Test set correlation coefficients using 10 most important neurons	120
A-1. RMLP performance as a function of the number of hidden PEs.....	136
A-2. Average testing correlation coefficients as a function of trajectory length.....	137
A-3. Training performance for 100 Monte Carlo simulations.....	139

A-4. Best performing network	139
B-1. Testing SER and CC values	142
C-1. Model performance using random vs. real neuronal activity	146

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1. Conceptual drawing of BMI components.....	2
1-2. The spike-binning process. A) Cellular potentials, B) A spike train, C) Bin count for a single cell, D) An ensemble of bin counts.....	8
1-3. Time-varying statistics of neuronal recordings for two behaviors.....	9
1-4. Local neuronal correlations time-synchronized with hand position and velocity.....	11
2-1. Research components of the Duke-DARPA Brain-Machine Interface Project.....	13
2-2. Chronic implant distributed over six distinct cortical areas of a Rhesus monkey.....	15
2-3. Reaching-task experimental setup and a representative 3-D hand trajectory.....	17
2-4. Using a joystick the monkey controlled the cursor to intersect the target (Task 2) and to grasp a virtual object by applying a gripping force indicated by the rings (Task 3).	18
3-1. Kalman filter block diagram.....	25
3-2. FIR filter topology. Each neuronal input s_N contains a tap-delay line with l taps.....	29
3-3. Time-delay neural network topology	30
3-4. Fully connected, state recurrent neural network.....	31
3-5. Reaching movement trajectory.....	42
3-6. Testing performance for a three reaching movements (Belle).....	44
3-7. Reaching task testing CEM (Belle)	45
3-8. Testing performance for a three reaching movements (Carmen).....	46
3-9. Reaching task testing CEM (Carmen).....	48
3-10. Testing performance for a three reaching movements (Ivy)	49

3-11. Testing performance for a three reaching movements (Aurora)	50
3-12. Reaching task testing CEM (Ivy)	51
3-13. Reaching task testing CEM (Aurora)	51
4-1. Two scenarios for data preparation	60
4-2. Scenario 1: Testing correlation coefficients for HP, HV, and GF	61
4-3. Scenario 2: Testing correlation coefficients for HP, HV, and GF	62
4-4. Multi-task model training trajectories	66
4-5. CEM curves for linear and nonlinear models trained on a multi-task	69
4-6. Multi-task testing trajectories centered upon a transition between the tasks	69
5-1. Pre-activity and Activity in a RMLP with one hidden PE	75
5-2. Operating points on hidden layer nonlinearity	76
5-3. Input layer decomposition into $\mathbf{W}_1\mathbf{s}(t)$ (solid) and $W_f y_1(t-1)$ (dashed)	77
5-4. Norm of the input vector (104 neurons)	79
5-5. Angle between $\mathbf{s}(t)$ and \mathbf{W}_1 . Direction cosines for successive input vectors $\mathbf{s}(t)$ and $\mathbf{s}(t-1)$	79
5-6. Selection of neurons contributing the most to input vector rotation	80
5-7. Output layer weight vector direction for one PE	82
5-8. Movement trajectory with superimposed output weight vectors (solid) and principal components (dashed). This view is in the direction of PC3	82
5-9. RMLP network decomposition for the cursor control task	84
6-1. One movement segmented into rest/food, food/mouth, and mouth/rest motions.....	91
6-2. FIR filter (Aurora): Testing output X, Y, and Z trajectories (bold) for one desired movement (light) from fifteen Wiener filters trained with neuronal firing counts from all combinations of four cortical areas	93
6-3. RMLP (Aurora): Testing output X, Y, and Z trajectories (bold) for one desired movement (light) from fifteen RMLPs trained with neuronal firing counts from all combinations of four cortical areas	94

6-4. RMLP (Carmen): - Testing output X, Y, and Z trajectories (bold) for one desired movement (light) from three RMLPs trained with neuronal firing counts from all combinations of two cortical areas.....	94
6-5. RMLP (Aurora): - Testing outputs (o – markers) and desired positions (x – markers) for six models trained with each separate cortical input. Testing (X, Y) correlation coefficients are provided in the title of each subplot	96
6-6. RMLP (Ivy): - Testing outputs (o – markers) and desired positions (x – markers) for four models trained with each separate cortical input. Testing (X, Y) correlation coefficients are provided in the title of each subplot	97
7-1. Sensitivity at time t for a typical neuron as a function of Δ	105
7-2. RMLP time-varying sensitivity. A) X, Y, Z desired trajectories for three similar movements. B) Neuronal firing counts summed (at each time bin) over 104 neurons. C) Sensitivity (averaged over 104 neurons) for three coordinate directions	106
7-3. Reaching task neuronal sensitivities sorted from minimum to maximum for a movement. The ten highest sensitivities are labeled with the corresponding neuron	108
7-4 Testing outputs for RMLP models trained with subsets of neurons. A,B,C) X, Y, and Z trajectories (bold) for one movement (light) from three RMLPs trained with the highest, intermediate, and lowest sensitivity neurons. D) CEM decreases as sensitive neurons are dropped	109
7-5. Belle’s neuronal firing counts from the ten highest and lowest sensitivity neurons time-synchronized with the trajectory of one reaching movement	110
7-6. Sensitivity based neuronal ranking for hand position and velocity for two sessions using a RMLP. The cortical areas corresponding to the ten highest ranking HP, HV and GF neurons are given by the colormap.....	112
7-7. Neuronal ranking based upon the depth of the tuning for each cell. The cortical areas corresponding to the most sharply tuned cells is given by the colormap	116
7-8. A-D) Cellular tuning curves for hand direction and gripping force using a model independent ranking method, E-F) Tuning curves for two representative cells from plots B) and D)	117
7-9. Scatter plot of neuron ranking for tuning depth and sensitivity analysis	119
7-10. Model performance as a function of the number of cells utilized. Cells were removed from the analysis either by randomly dropping cells (neuron dropping – ND) or by removing the least sensitive cell in an ordered list (computed from a sensitivity analysis - SA).....	122
A-1. RMLP learning curve. MSE (upper curve) Crossvalidation (lower curve).....	137

A-2. Training MSE curves for 100 Monte Carlo simulations	138
B-1. Signal to error ratio between the actual and estimated hand coordinates for SS and NSS data	142
B-2. Peaks of hand trajectory (Z-coordinate)	143
B-3. Estimation errors for six peaks. Targets are represented by an x at the origins. The average error (mm) in each direction is displayed on the respective axis.....	143
C-1. Performance measures for five Monte-Carlo simulations using neuronal and random data	147

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

FROM CORTICAL NEURAL SPIKE TRAINS TO BEHAVIOR:
MODELING AND ANALYSIS

By

Justin Cort Sanchez

May 2004

Chair: Jose C. Principe

Major Department: Biomedical Engineering

Brain machine interface (BMI) design can be achieved by training linear and nonlinear models with simultaneously recorded cortical neural activity and goal directed behavior. Real-time implementation of this technology requires reliable and accurate signal processing models that produce small error variance in the estimated kinematic trajectories. In this dissertation, the mapping performance and generalization of a recurrent multilayer perceptron (RMLP) is compared with standard linear and nonlinear signal processing models for two species of primates and two behavioral tasks. Each modeling approach is shown to have strengths and weaknesses that are compared experimentally. The RMLP approach shows very accurate peak amplitude estimations with small error variance using a parsimonious model topology. To validate and advance the state-of-the-art of this BMI modeling design, it is necessary to understand how the proposed model represents the neural-to-motor mappings. The RMLP is analyzed here and an interpretation of the neural-to-motor solution of this network is built by tracing the

signals through the topology using signal processing concepts. We then propose the use of optimized BMI models for analyzing neural activity to assess the role of and importance of individual neurons and cortical areas in generating the performed movement. It is further shown that by pruning the initial ensemble of neural inputs with the ranked importance of cells, a reduced set of cells can be found that exceed the BMI performance levels of the full ensemble.

CHAPTER 1 INTRODUCTION

Throughout a lifetime, it is not often that one has the opportunity to be a part of a revolution. However, when the time arises, it is often an unexpected, challenging, and turbulent time because details about the future remain unknown. For example, history has been faced with political, ethical, and scientific revolutions in which nations were born, lives were lost, and lifestyles were changed. Each of these events bestowed on the revolutionists an opportunity to change their lives by learning about the core of their beliefs, which enabled them to expand their horizons. Acting upon the new opportunities, the activists embraced the new perspectives which, undoubtedly made the future more clear.

Presently we are in the midst of a technological revolution. Our lives are being transformed by world-wide high-speed communications and digital technology that allows for instant gratification in the exchange of ideas and experiences. Interaction with digital computers is the means through which this revolution is taking place. As time passes and our scientific abilities develop, it remains unknown how deeply cultures will embrace this technology to express their will, and share their ideas and experiences. We must ask ourselves how and when will the line between man and machine blur or even vanish. How can we prepare for this merger? What are the scientific, ethical, and engineering challenges that must be overcome for such a change? What opportunities should be seized now, that will shape our future?

Recently, several landmark experimental paradigms have begun to blur the line between man and machine by showing the feasibility of using neuroprosthetic devices to restore motor function and control in individuals who are “locked in” or who have lost the ability to control the movement of their limbs [1-19]. In these experiments, researchers seek to both rehabilitate and augment the performance of neural-motor systems using Brain-Machine Interfaces (BMIs) that directly transfer the intent of the individual (as collected from the brain cortex) into control commands for prosthetic limbs and computers. Brain Machine Interface research has been motivated by the need to help the more than 200,000 individuals in the U.S. suffering from a wide variety of neurological disorders that include spinal cord injury and diseases of the peripheral nervous system [20]. While the symptoms and causes of these disabilities are diverse, one characteristic is common in many of these neurologic conditions; normal functioning of the brain remains intact. If the brain is spared from injury and control signals can be extracted, the BMI problem becomes one of finding *optimal* signal processing techniques to efficiently and accurately convert these signals into operative control commands.

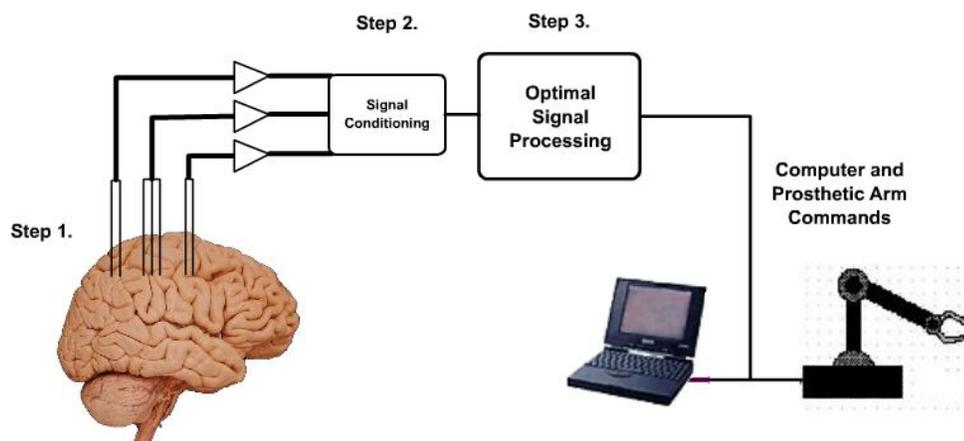


Figure 1-1. Conceptual drawing of BMI components

A conceptual drawing of a BMI is depicted in (Fig. 1-1) where neural activity from hundreds of cells is recorded (step 1), conditioned (step 2), and translated (step 3) directly into hand position (HP), hand velocity (HV), and hand gripping force (GF) of a prosthetic arm or cursor control for a computer. Our study focused on step 3 of the diagram where optimal signal processing techniques were used to find the functional relationship between neuronal activity and behavior. From an optimal signal processing viewpoint, BMI modeling in step 3 is a challenging task because of several factors: the intrinsic partial access to the motor cortex information due to the subsampling of the neural activity, the unknown aspects of neural coding, the huge dimensionality of the problem, and the need for real-time signal processing algorithms. The problem is further complicated by a need for good generalization in nonstationary environments that depends on model topologies, fitting criteria, and training algorithms. Finally, reconstruction accuracy must be assessed, since it is linked to the choice of linear vs. nonlinear and feedforward vs. feedback models.

Since the basic biological and engineering challenges associated with optimal signal processing for BMI experiments require a highly interdisciplinary knowledge base involving neuroscience, electrical and computer engineering, and biomechanics, the BMI modeling problem is introduced in several steps. First, an overview of the pioneering modeling approaches gives the reader a deeper understanding of what has been accomplished in this area of research. Second, the reader is familiarized with characteristics of the neural recordings used in signal processing methods.

Historical Overview of BMI Modeling Approaches

The foundations of BMI research were laid in the early 1980s by E. M. Schmidt [1] who was interested in finding out if it was possible to use neural recordings from the

motor cortex of a primate to control external devices. In this pioneering work, Schmidt measured how well primates could be conditioned to modulate the firing patterns of *single* cortical cells using a series of eight target lamps, each symbolizing a cellular *firing rate* that the primate was required to produce. The study confirmed that a primate could intend to match the target firing rates and also estimated the information transfer rate in the neural recordings to be half that of using the intact motor system as the output. With this result, Schmidt proposed that engineered interfaces could be designed to use modulations of neural firing rates as control signals.

Shortly after Schmidt published his results, Georgopoulos and Schwartz presented a theory for neural *population* coding of hand kinematics as well as a method for reconstructing hand trajectories, called the population vector algorithm (PVA) [7]. Using center out reaching tasks, Georgopoulos proposed that each cell in the *motor cortex* has a “preferred hand direction” for which it fires maximally and the distribution of cellular firing over a range of movement directions could be characterized by a simple cosine function [5]. In this theory, arm movements were shown to be constructed by a population “voting” process among the cells; each cell makes a vectoral contribution to the overall movement in its preferred direction with magnitude proportional to the cell’s average firing rate [21].

Schmidt’s proof of concept and Georgopoulos’ BMI application to reaching tasks spawned a variety of studies implementing “out of the box” signal processing modeling approaches. One of the most notable studies by Chapin et al. [3] showed that a *Jordan style* recurrent neural network could be used to translate the neural activity (21 to 46 neurons) of rats trained to obtain water by pressing a lever with a paw. The usefulness of

this BMI was shown when the animals routinely stopped physically moving their limbs to obtain the water reward. Also in the neural network class, Lin et al. [22] used a self-organizing map (SOM) that clustered neurons with similar firing patterns which then indicated movement directions for a spiral drawing task. Borrowing from control theory, Kalaska and Scott [23] proposed the use of forward and inverse control architectures for reaching movements. Also during this period other researchers presented interpretations of population coding which included a probability-based population coding from Sanger [24] and muscle-based cellular tuning from Mussa-Ivaldi [25].

Almost 20 years after Schmidt and Georgopoulos' initial experiments, Nicolelis and colleagues [19] presented the next major advancement in BMIs by demonstrating a real (nonportable) neuroprosthetic device in which the neuronal activity of a primate was used to control a robotic arm. This research group hypothesized that the information needed for the BMI is distributed across several cortices and therefore neuronal activity was collected from 100 cells in *multiple* cortical areas (premotor, primary motor, and posterior parietal) while the primate performed a 3-D feeding (reaching) task. Linear and nonlinear signal processing techniques including a frequency domain Wiener filter (WF) and a time-delay neural network (TDNN) were used to estimate hand position. Trajectory estimates were then transferred via the internet to a local robot and a robot located at another university.

In parallel with the the work of Nicolelis, Donoghue and colleagues [16] presented a contrasting view of BMIs by showing that a 2-D computer cursor control task could be achieved using only a few neurons (between 7 and 30) located only in the primary motor cortex of a primate. The Wiener filter signal processing methodology was again

implemented here, however this paradigm was *closed-loop* since the primate received instant visual feedback from the cursor position output from the WF. The novelty of their experiment was the primate's opportunity to incorporate the signal processing model into its motor processing.

The final BMI approach presented is from the Andersen research group [26] who showed that the end point of hand reaching can be estimated using a Bayesian probabilistic method. Neural recordings were taken from the Parietal Reach Region (PRR) since this region is believed to encode the planning and target of hand motor tasks. Using this hypothesis they devised a paradigm where a primate was cued to move its hand to a rectangular grid of target locations presented on a computer screen. The neural-to-motor translation involves computing the likelihood of neural activity, given a particular target. While this technique has been shown to accurately predict the end point of hand reaching, it differs from the forementioned techniques by not accounting for the hand trajectory.

Foundations of Neuronal Recordings

One of the most important steps in implementing an optimal signal processing technique for *any* application is data analysis. Optimality in the signal processing technique implies that the *a priori* information about the statistics of the data match the *a priori* information used in designing the signal processing technique [27]. In the case of BMIs, the statistical properties of the neural recordings and the analysis of neural ensemble data are not fully understood. Hence, our lack of information means that the neural-motor translation is not guaranteed to be optimum. Despite this reality, by developing new neuronal data-analysis techniques, the match between neural recordings

and BMI design can be improved [28, 29]. For this reason, it is important for the reader to be familiar with the characteristics of neural recordings that would be encountered.

Characteristics of Neuronal Activity

The process of extracting signals from the motor, premotor, and parietal cortices of a behaving animal involves the implantation of subdural microwire electrode arrays into the brain tissue (usually layer V) [19]. At this point, the reader should be aware that scope of sampling of current BMI studies involves tens to hundreds of cortical cells recorded from tissue that is estimated to contain 10^{11} neurons, 10^{14} synapses, and 10^{10} cortical circuits [30]. Each microwire measures the potentials (*action potentials*) resulting from ionic current exchanges across the membranes of neurons locally surrounding the electrode. Typical cellular potentials (Fig. 1-2A), have magnitudes ranging from hundreds of microvolts to tens of millivolts, and time durations of tens to a couple of milliseconds [29]. Since action potentials are so short in duration, it is common to treat them as point processes where the continuous voltage waveform is converted into a series of *timestamps* indicating the instance in time when the spike occurred. Using the timestamps, a series of pulses or *spikes* (zeros or ones) can be used to visualize the activity of each neuron; this time-series (Fig. 1-2B) is referred to as a *spike train*. The spike trains of neural ensembles have several properties including sparsity, nonstationarity, and are noncontinuous valued. While the statistical properties of neural recordings can vary depending on the sample area, the animal, and the behavior paradigm, in general, spike trains can be estimated to have a Poisson distribution [28]. To reduce the sparsity in neuronal recordings, a method of binning is used to count the

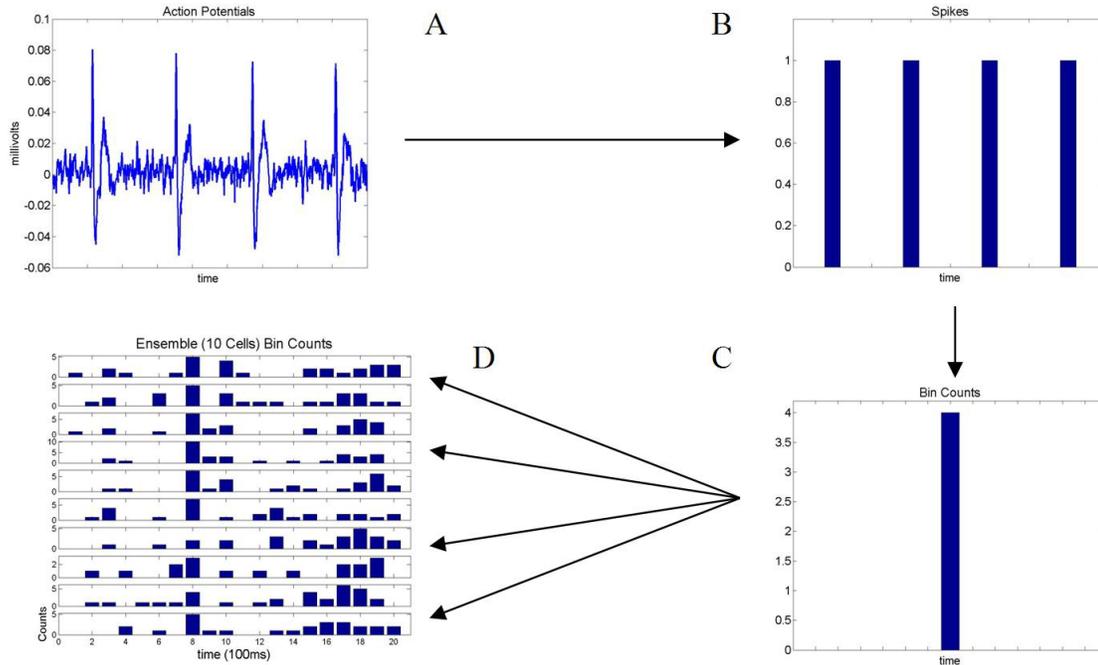


Figure 1-2. The spike-binning process. A) Cellular potentials, B) A spike train, C) Bin count for a single cell, D) An ensemble of bin counts.

number of spikes in 100 ms non-overlapping windows as shown in Fig. 1-2C. This method greatly reduces the number of zeros in the digitized time-series and also provides a time to amplitude conversion of the firing events. Even with the binning procedure, the data remains extremely sparse. To assess the degree of sparsity and nonstationarity in BMI data, we used observations from a 25-minute BMI experiment from the Nicolelis Lab in Duke University (Table 1-1). The table shows that the percentage of zeros can be as high as 80%. Next, an ensemble average of the firing rate per minute is computed using nonoverlapping 1 minute windows averaged across all cells. The ensemble of cells used in this analysis primarily contains low firing rates given by the small ensemble average. Additionally we can see time variability of the 1- minute ensemble average given by the associated standard deviation.

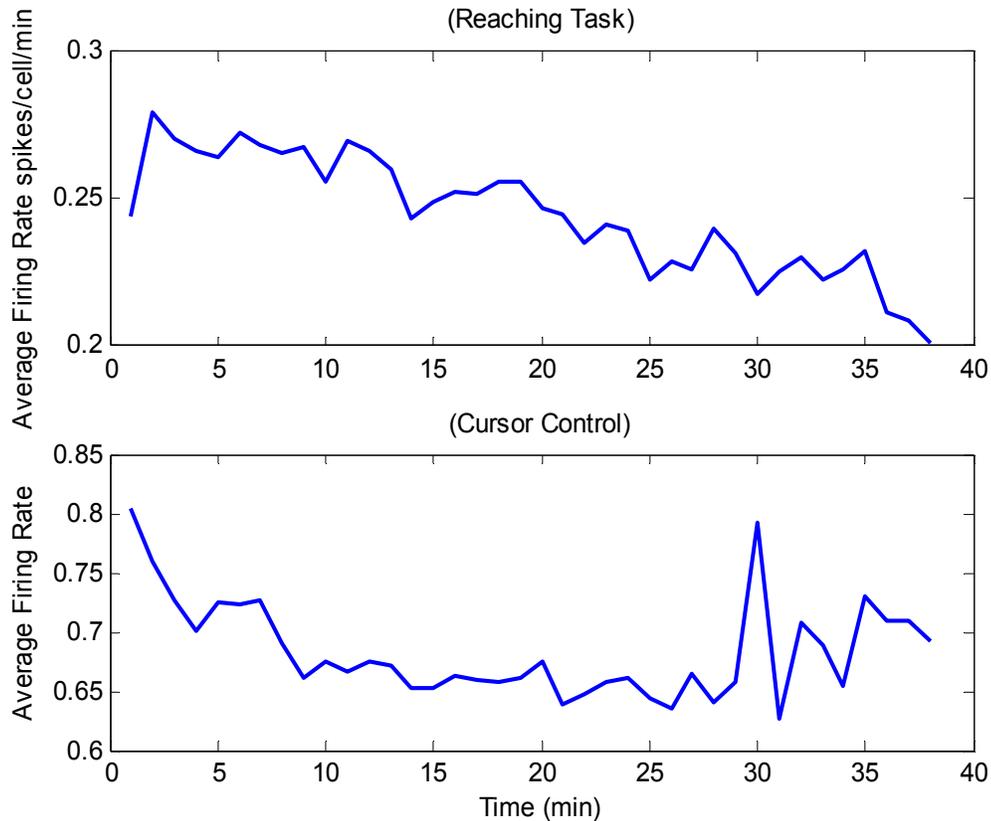


Figure 1-3. Time-varying statistics of neuronal recordings for two behaviors

The important message of this analysis is that standard optimal signal processing techniques (Wiener filters, neural network, etc.) were not designed for data that is nonstationary and discrete valued. In Figure 1-3, the average firing rate of the ensemble (computed in nonoverlapping 60 sec windows) is tracked for a 38-minute session. From minute to minute, the mean value in the firing rate can change drastically depending on the movement being performed. Ideally we would like our optimal signal processing techniques to capture the changes observed in Fig. 1-3. However, the reader should be aware that in the environment of this dataset, applying any of the “out of the box” signal processing techniques means that the neural to motor mapping is not optimal. More importantly, any performance evaluations and model interpretations drawn by the

experimenter can be directly linked and biased by the mismatch between data and model type.

Table 1-1. Neuronal activity for a 25-minute recording session

	Percentage of Zeros	Average Firing Rate spikes/cell/minute
3-D Reaching Task (104 cells)	86	0.25±0.03
2-D Cursor Control Task (185 cells)	60	0.69±0.02

Local Neuronal Correlations

Another method to describe the nonstationary nature of the data is by computing local neuronal correlations. In this scheme, we attempt to observe information contained in cell assemblies, or subsets of neurons in each cortical area that mutually excite each other [31]. We attempt to extract useful information in the local bursting activity of cells in the ensemble by analyzing the local correlations among the cells. To identify and extract this local activity, the cross-correlation function in Eq. 1-1 was used to quantify synchronized firing among cells in the ensemble. In this case, small sliding (overlapping) windows of data are defined by $\mathbf{A}(t)$ which is a matrix containing L delayed versions of firing patterns from 180 cells. At each time tick t in the simulation, the cross-correlation between neurons i and j for all delays l is computed. Since we are interested in picking only the most strongly synchronized bursting neurons in the local window, we simply average over the delays in Eq. 1-2. We define \bar{C}_j as a vector representing how well correlated the activity of neuron j is with the rest of the neuronal ensemble. Next, a single 180x1 vector at each time tick t is obtained in Eq. 1-3 by summing cellular assembly cross-correlations only within each sampled cortical area, M1, PMd, SMA, and S1.

$$C(t)_{ijl} = E[\mathbf{A}(t)_i \mathbf{A}(t-l)_j] \quad (1-1)$$

$$\bar{C}(t)_{ij} = \frac{1}{L} \sum_{l=1}^L |C(t)_{ijl}| \quad (1-2)$$

$$\tilde{C}(t)_j = \sum_{i=cortex} \bar{C}(t)_{ij} \quad (1-3)$$

The neural correlation measure in Eq. 1-3 was then used as a real-time marker of neural activity corresponding to segments of movements of a hand trajectory. In Fig. 1-4, highly correlated neuronal activity is shown to be time-varying in synchrony with changes in movement direction as indicated by the vertical dashed line. Figure 1-4 shows that the firing rates contained in the data is highly variable even for similar movements. Using “off the shelf” signal processing modeling approaches, it is difficult to capture this local variability in time since many of the models use inner product operations that average the ensemble’s activity.

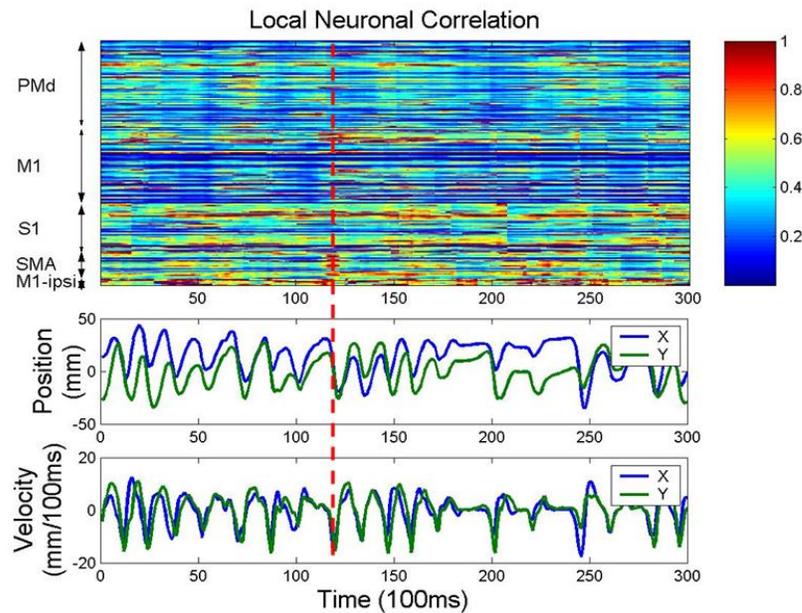


Figure 1-4. Local neuronal correlations time-synchronized with hand position and velocity

CHAPTER 2 DUKE BRAIN-MACHINE INTERFACE PARADIGM

The ultimate goal for modeling and analyzing the relationship from cortical neuronal spike trains to behavior is to develop components of a real closed-loop BMI system. The envisioned multicomponent, multi-university experimental paradigm was developed for primates by Miguel Nicolelis at Duke University (Fig. 2-1). This novel research paradigm seeks to collect neuronal activity from multiple cortices and translate the encoded information into commands for a robotic arm that the primate can see and react to. The hope is that visual feedback and somatosensory stimulation from the movement of the robotic arm will allow the primate to incorporate the mechanical device into its own cognitive space. It remains unknown how neural control of a mechanical device will impact the normal neurophysiology of the primate. Modeling and analysis of neuronal and behavioral recordings bridges the gap between component 1 and component 5 of the diagram, and provides an intersection for studying motor neurophysiology. Here we provide a description of the neuronal recording techniques and behavioral paradigms. Data collected from these experiments serves as inputs and desired responses for the models in the remainder of this dissertation.

Neuronal Recording Methodology

Data for the following experiments were generated in the primate laboratory at Duke University and neuronal recordings were collected from four adult, female primates: 2 owl monkeys (*Aotus trivirgatus*) Belle and Carmen and 2 Rhesus monkeys (*Macaca mulatta*) Aurora and Ivy. All four animals were instrumented with several high-

density microelectrode arrays, each consisting of up to 128 microwires (30 to 50 μm in diameter, spaced 300 μm apart), distributed in a 16 X 8 matrix. Each recording site occupies a total area of 15.7 mm^2 (5.6 \times 2.8 mm) and is capable of recording up to four single cells from each microwire for a total of 512 neurons (4 \times 128).

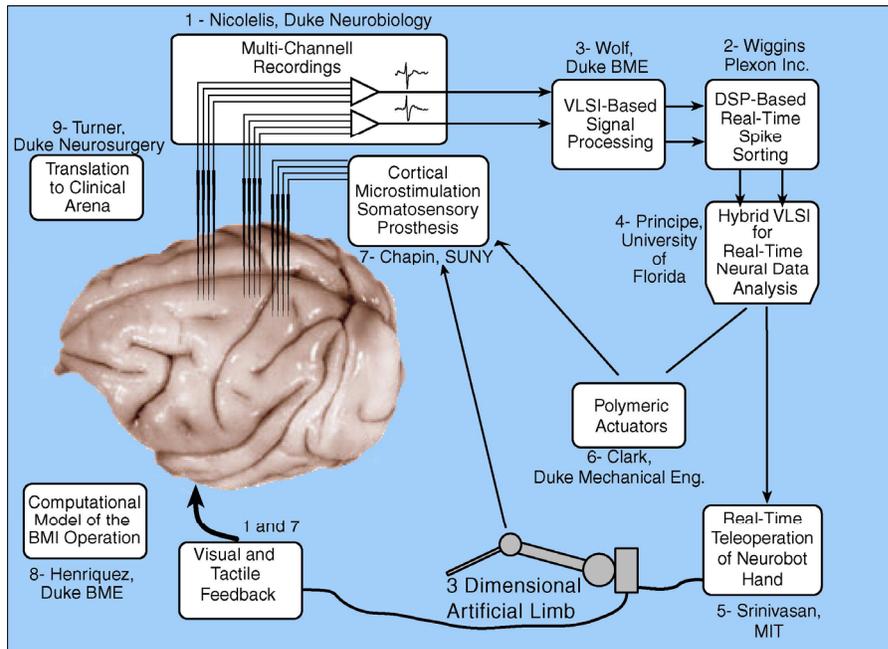


Figure 2-1. Research components of the Duke-DARPA Brain-Machine Interface Project

Through a series of small craniotomies, electrode arrays were implanted stereotaxically in seven cortical neural structures that are involved in controlling fine arm and hand movements. The following cortical areas and arm/hand neural structures were targeted using neuroanatomical atlases and intraoperative stimulation: the anterior parietal cortex (areas 3a, 1, 2-5); and in area 7a of the posterior parietal cortex (this area receives both visual and tactile inputs and projects to the premotor cortex); primary motor (MI) cortex, the dorsolateral premotor areas (PMd); the supplementary motor area (SMA); and the primary somatosensory cortex (S1) [19, 32]. Assignment of electrode arrays and the sampled cortical area for each primate are shown in Tables 2-1 and 2-2. The number of cells contained in each session's (S1, S2) recording is also included.

Notice that the number of cells in each session can either increase or decrease. For Belle, Aurora, and Ivy, the time interval between sessions was 1 day; but in the case of Carmen, only one experiment could be obtained. Figure 2-2 shows a fully instrumented primate that has microwire arrays implanted into three cortical areas on each hemisphere.

Table 2-1. Assignment of electrode arrays to cortical regions for owl monkeys

Belle (right handed)				Carmen (right handed)	
Area 1	Area 2	Area 3	Area 4	Area 2	Area 3
Posterior Parietal (PP-contra)	Primary Motor (M1-contra)	Dorsal Premotor (PMd-contra)	Primary Motor & Dorsal Premotor (M1/PMd-ipsi)	Primary Motor (M1-contra)	Dorsal Premotor (PMd-contra)
S1-33 cells S2-29 cells	S1-21 cells S2-19 cells	S1-27 cells S2-23 cells	S1-23 cells S2-20 cells	S1-37 cells	S1-17 cells

Table 2-2. Assignment of electrode arrays to cortical regions for Rhesus monkeys

Aurora (left handed)					Ivy (left handed)		
Area 2	Area 4	Area 5	Area 6	Area 7	Area 1	Area 2	Area 6
Primary Motor (M1-contra)	Dorsal Pre-motor (PMd-contra)	Somato-sensory (S1-contra)	Supp. Motor Associative (SMA-contra)	Primary Motor (M1-ipsi)	Posterior Parietal (PP-contra)	Primary Motor (M1-contra)	Supp. Motor Associative (SMA-contra)
S1-56 cells S2-57 cells	S1-64 cells S2-66 cells	S1-39 cells S2-38 cells	S1-19 cells S2-19 cells	S1-5 cells S2-5 cells	S1-49 cells S2-50 cells	S1-90 cells S2-97 cells	S1-53 cells S2-58 cells

On recovering from the surgical procedure, the primates are treated with a daily regiment of antibiotics. Neuronal recordings begin 15 days after surgery. Multineuron recording hardware and software, developed by Plexon Inc. (Dallas, TX), is used in the experimental setup. With Plexon's Multi-channel Many Neuron Acquisition Processor (MNAP), the neuronal action potentials are amplified and band pass filtered (500 Hz to 5

KHz), and later digitized using a sampling rate of 30 KHz. From the raw electrode voltage potentials, cells are sorted, and single spikes are discriminated using a principle component algorithm and a pair of time-voltage windows per unit. Particular neurons are sorted by the BMI experimenter, who adjusts the sizes and positions of the time-voltage boxes in an attempt to gather features of single cells. The firing times of all sorted spikes are transferred to the hard-disk of a controller PC. Neuronal firing times are then binned (added) in nonoverlapping windows of 100 ms (and are the values directly used as inputs to the models used in this dissertation). Along with the firing times, the MNAP software keeps records of the action potential waveforms for all sorted cells, so the single neurons can be identified and compared over the span of several recording sessions.

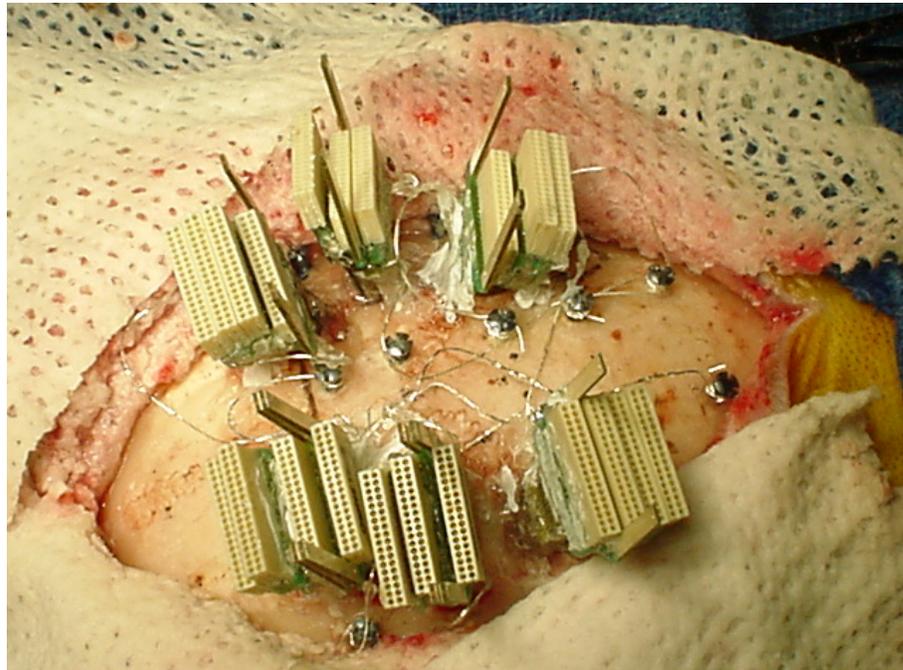


Figure 2-2. Chronic implant distributed over six distinct cortical areas of a Rhesus monkey

Behavioral Experiments

Three separate behavioral tasks were used in this study. In the first experiment, the firing times of single neurons were recorded while the primates (Belle and Carmen) performed a 3-D reaching task that involved a right-handed reach to food then placing the food in the mouth. The primates are seated in a chair (Fig 2-3) and are cued to reach to food in one of four positions. The primate's hand position, used as the models' desired signal, was recorded (with a time shared clock) and digitized with 200 Hz sampling rate. To take into account the primate's reaction time, the spike trains were delayed by 0.230¹ seconds with respect to the hand position. This delay was chosen based on loose neurophysiologic reasoning, and should be subject to optimization in future studies. Neuronal and positional data were collected during two independent sessions on 2 consecutive days, from the same primate performing the same reaching task. The task and recording procedure were repeated for the second primate in two independent sessions over 2 consecutive days for each of the tasks. In this experiment, since reaching movements are sparsely embedded in a background of resting movements we used a large set of 20,000 samples was used for model training; and 3,000 samples for model testing.

The second task, a cursor control task, involved the presentation of a randomly placed target (large disk) on a computer monitor in front of the monkey (Aurora and Ivy). The monkey used a hand-held manipulandum (joystick) to move the cursor (smaller circle) so that it intersects the target (Fig. 2-4). On intersecting the target with the cursor, the monkey received a juice reward. While the monkey performed the motor task, the

¹ For the purpose of this dissertation, all results are consistent on an interval of 0.030 to 0.430 seconds. While optimization of the delay for a particular animal and motor task can be the subject of another study, we have observed performance to decrease with increasing the delay.

hand position and velocity for each coordinate direction (HP_x , HP_y and HV_x , HV_y) were recorded in real time (1000 Hz) along with the corresponding neural activity.

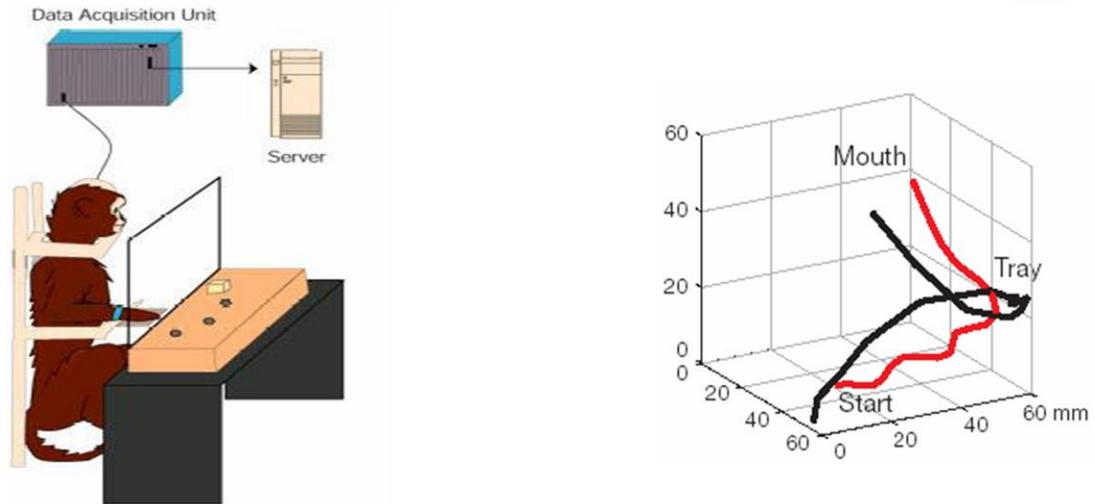


Figure 2-3. Reaching-task experimental setup and a representative 3-D hand trajectory

In the third task, the monkey was presented with the cursor in the center of the screen and two concentric rings. The diameter difference of these two circles instructed the amount of gripping force the animal had to produce. Gripping force (GF) was measured by a pressure transducer located on the joystick. The size of the cursor grew as the monkey gripped the joystick, providing continuous visual feedback of the amount of gripping force. Force instruction changed every trial while the position of the joystick was fixed. Hand kinematic parameters, were digitally low-pass filtered and downsampled to 10 Hz. Both the neural recordings and behavior were time aligned² and used directly as inputs and desired signals for each model. During each recording session, a sample data set was chosen consisting of 8,000 data points. This data set was segmented into two exclusive parts: 5,000 samples for model training and 3,000 samples for model testing.

² We assume that the memory structures in each model can account for the delay between neural activity and the generation of the corresponding behavior.

Results in [32] showed that models trained with approximately 10 minutes of data produced the best fit.

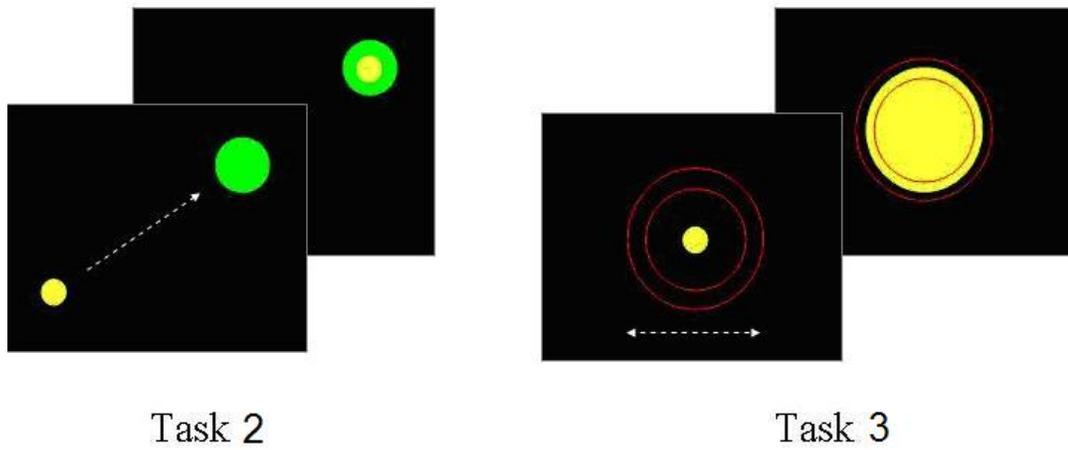


Figure 2-4. Using a joystick the monkey controlled the cursor to intersect the target (Task 2) and to grasp a virtual object by applying a gripping force indicated by the rings (Task 3).

CHAPTER 3 MODELING PROBLEM

What Mapping Does the Model Have to Find?

The models implemented in BMIs must learn to interpret neuronal activity and accurately translate it into motor commands. By analyzing recordings of neural activity collected simultaneously with behavior the aim is to find a *functional* relationship between neural activity and the kinematic variables. An important question here is how to choose the class of functions and model topologies that best match the data and that are sufficiently powerful to create a mapping from neuronal activity to a variety of behaviors. As a guide, prior knowledge about the nervous system can be used to help develop this relationship. Since the experimental paradigm involves measuring only two variables (neural firing and behavior) we are directed to a general class of Input-Output (I/O) models that have the ability to create a representation space between two time-series. Within this class, several candidate models are available. Based on the amount of neurophysiologic information known about the system, an appropriate model can be chosen. Three types of I/O models based on the amount of prior knowledge exist in the literature [33]:

- **White Box.** The model is perfectly known and built from physical insight and observations.
- **Gray Box.** Some physical insight to the model is known but other parameters need to be determined from the data
- **Black Box.** No physical insight is available or used to choose the model. The chosen model is known to be robust in a variety of applications.

Our choice of white, gray or black box is dependent upon our ability to access and measure signals at various levels of the motor system as well as the computational cost of implementing the model in our current computing hardware.

Signal Processing Approaches to Modeling

White Box

The first modeling approach, “white box,” would require the highest level of physiologic detail. Starting with behavior and trace back, the system comprises muscles, peripheral nerves, the spinal cord, and ultimately the brain. This is a daunting task of system modeling due to the complexity, interconnectivity, and dimensionality of the involved neural structures. Model implementation would require the parameterization of a complete motor system [34] that includes the cortex, cerebellum, basal ganglia, thalamus, corticospinal tracts, and motor units. Since all of the details of each component/subcomponent of the described motor system remain unknown and are the subject of study for many neurophysiologic research groups around the world it is not presently feasible to implement white-box BMIs. Even if were possible to parameterize the system to some high level of detail, the task of implementing the system in our state-of-the-art computers and digital signal processors (DSPs) would be a cumbersome task.

Gray Box

Next, the “gray box” model requires a reduced level of physical insight. In the “gray box” approach, one could take a particularly important feature of the motor nervous system, incorporate this knowledge into the model, and then use data to determine the rest of the unknown parameters. Two examples of “gray box” models can be found in the BMI literature. First, one of the most common examples is Georgopoulos’ population vector algorithm (PVA) [7]. Using observations that cortical neuronal firing rates were

dependent on the direction of arm movement, a model was formulated to incorporate the weighted sum of the neuronal firing rates. The weights of the model are then determined from the neural and behavioral recordings. A second example is given by Todorov who extended the PVA by observing multiple correlations of M1 firing with movement position, velocity, acceleration, force exerted on an object, visual target position, movement preparation, and joint configuration [5, 6, 10, 12, 14, 16, 18, 19, 35-38]. With these observations, Todorov proposed a minimal, linear model that relates the delayed firings in M1 to the sum of many mechanistic variables (position, velocity, acceleration and force of the hand) [39]. Todorov's mechanistic variables are incorporated into signal processing methodologies through a general class of generative models [4, 40]. Using knowledge about the relationship between arm kinematics and neural activity, the states (preferably the feature space of Todorov) of linear or nonlinear dynamical systems can be assigned. This methodology is supported by a well known training procedure developed by Kalman [41]. Since the formulation of generative models is recursive in nature, it is believed that the model is well suited for learning about motor systems because the states are all intrinsically related in time.

Population vector algorithm

The first model discussed is the population vector algorithm which assumes that a cell's firing rate is a function of the velocity vector associated with the movement performed by the individual. The PVA model is given by Eq. (3-1)

$$s_n(\mathbf{V}) = b_0^n + b_x^n v_x + b_y^n v_y + b_z^n v_z = \mathbf{B} \cdot \mathbf{V} = |\mathbf{B}| |\mathbf{V}| \cos \theta \quad (3-1)$$

where the firing rate s for neuron n is a weighted ($b_{x,y,z}^n$) sum of the vectoral components ($v_{x,y,z}$) of the unit velocity vector \mathbf{V} of the hand plus mean firing rate b_0^n . The relationship

in Eq. (3-1) is the inner product between the velocity vector of the movement and the weight vector for each neuron. The inner product (i.e spiking rate) of this relationship becomes maximum when the weight vector \mathbf{B} is collinear with the velocity vector \mathbf{V} . At this point, the weight vector \mathbf{B} can be thought as the cells preferred direction for firing since it indicates the direction for which the neuron's activity will be maximum. The weights b^n can be determined by multiple regression techniques [7]. Each neuron makes a vectoral contribution w in the direction of \mathbf{P}_i with magnitude given in Eq. (3-2). The resulting population vector or movement is given by Eq. (3-3) where the reconstructed movement at time t is simply the sum of each neurons preferred direction weighted by the firing rate.

$$w_n(\mathbf{V}, t) = s_n(\mathbf{V}) - b_0^n \quad (3-2)$$

$$\mathbf{P}(\mathbf{V}, t) = \sum_{n=1}^N w_n(\mathbf{V}, t) \frac{\mathbf{B}_n}{\|\mathbf{B}_n\|} \quad (3-3)$$

Todorov's mechanistic model

An extension to the PVA has been proposed by Todorov [39] who considered multiple correlations of M1 firing with movement velocity and acceleration, position, force exerted on an object, visual target position, movement preparation, and joint configuration [5, 6, 10, 12, 14, 16, 18, 19, 35-38]. Todorov proposed an alternative hypothesis stating that neural correlates with kinematic variables are epiphenomena of muscle activation stimulated by neural activation. Using studies showing that M1 that contains multiple, overlapping representations of arm muscles and forms dense corticospinal projections to the spinal cord and is involved with the triggering of motor programs and modulation of spinal reflexes [30], Todorov proposed a minimal, linear

model that relates the delayed firings in M1 to the *sum* of mechanistic variables (position, velocity, acceleration and force of the hand). Todorov's model takes the form

$$\mathbf{U}\mathbf{s}(t - \Delta) = \mathbf{F}^{-1}\mathbf{GF}(t) + m\mathbf{HA}(t) + b\mathbf{HV}(t) + k\mathbf{HP}(t) \quad (3-4)$$

where the neural population vector \mathbf{U} is scaled by the neural activity $\mathbf{s}(t)$ and related to the scaled kinematic properties of gripping force $\mathbf{GF}(t)$, hand acceleration $\mathbf{HA}(t)$, velocity $\mathbf{HV}(t)$, and position $\mathbf{HP}(t)$ ³. From the BMI experimental setup, a spatial sampling (in the hundred of neurons) of the input $\mathbf{s}(t)$ and the hand position, velocity, and acceleration are collected synchronously, therefore the problem is one of finding the appropriate constants using a system identification framework [27]. Todorov's model in Eq. (3-4) assumes a first order force production model and a local linear approximation to multi-joint kinematics that may be too restrictive for BMIs. The mechanistic model for neural control of motor activity given in Eq. (3-4) involves a dynamical system where the output variables, position, velocity, acceleration, and force, of the motor system are driven by an high dimensional input signal that is comprised of delayed ensemble neural activity [39]. In this interpretation of Eq. (3-4), the neural activity can be viewed as the *cause* of the changes in the mechanical variables and the system will be performing the decoding. In an alternative interpretation of equation Eq. (3-4), one can regard the neural activity as a distributed *representation* of the mechanical activity, and the system will be performing generative modeling. Next a more general state space model implementation, the Kalman Filter, will be presented. This filter corresponds to the *representation* interpretation of Todorov's model for neural control.

³ The mechanistic model reduces to the PVA if the force, acceleration, and position terms are removed.

Implementation of “gray box” models

The Kalman formulation attempts to estimate the state, $\mathbf{x}(t)$, of a linear dynamical system (Fig. 3-1). Here, for BMI applications define that the hand position, velocity, acceleration, and the neuronal spike counts are governed by a linear dynamical equation. In this model, the state vector is defined in Eq. (3-5)

$$\mathbf{x}(t) = [\mathbf{HP}(t) \quad \mathbf{HV}(t) \quad \mathbf{HA}(t) \quad \mathbf{f}_{1\dots N}(t)]^T \quad (3-5)$$

where \mathbf{HP} , \mathbf{HV} , and \mathbf{HA} are the hand position, velocity, and acceleration vectors⁴, respectively. The spike counts of N neurons are also included in the state vector as $\mathbf{f}_1, \dots, \mathbf{f}_N$. The Kalman formulation consists of a generative model for the data specified by the linear dynamic equation for the state in Eq. (3-6)

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{w}(t) \quad (3-6)$$

where $\mathbf{w}(t)$ is assumed to be a zero-mean noise term with covariance \mathbf{W} . The output mapping (from state to spike trains) for this BMI linear system is simply

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{v}(t) \quad (3-7)$$

where $\mathbf{v}(t)$ is the zero mean measurement noise with covariance \mathbf{V} and \mathbf{y} is a vector consisting of the neuron firing patterns binned in non-overlapping windows. In this specific formulation, the output-mapping matrix is $\mathbf{C} = \begin{bmatrix} \mathbf{0}_{N \times 9} & \mathbf{I}_{N \times N} \end{bmatrix}$ and the output noise is zero, i.e. $\mathbf{V} = \mathbf{0}$. This recursive state estimation using hand position, velocity, and acceleration is potentially more powerful than the linear filter mapping just neural activity to position. This advantage comes at the cost of long training set requirements since the model contains many parameters to be optimized.

⁴ The state vector is of dimension $9+N$; each kinematic variable contains an x, y, and z component plus the dimensionality of the neural ensemble.

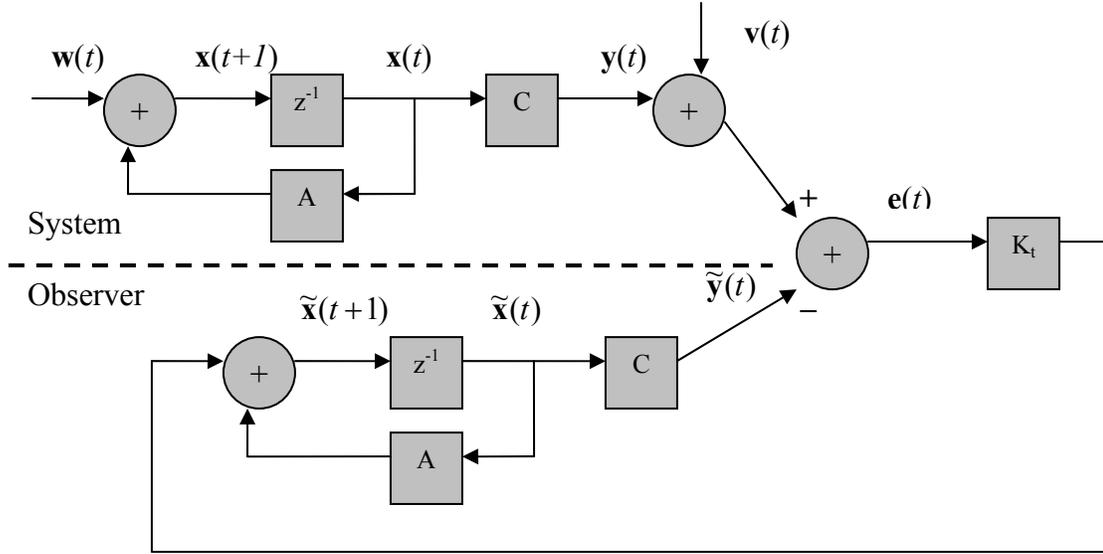


Figure 3-1. Kalman filter block diagram

Suppose there are L training samples of $\mathbf{x}(t)$ and $\mathbf{y}(t)$, and the model parameters \mathbf{A} and \mathbf{W} are determined using least squares. The optimization problem to be solved is given by Eq. (3-8).

$$\mathbf{A} = \arg \min_{\mathbf{A}} \sum_{t=1}^{L-1} \|\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t)\|^2 \quad (3-8)$$

The solution to this optimization problem is found to be Eq. (3-9)

$$\mathbf{A} = \mathbf{X}_1 \mathbf{X}_0^T (\mathbf{X}_1 \mathbf{X}_1^T)^{-1} \quad (3-9)$$

where the matrices are defined as $\mathbf{X}_0 = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_{L-1}]$, $\mathbf{X}_1 = [\mathbf{x}_2 \ \cdots \ \mathbf{x}_L]$. The estimate of the covariance matrix \mathbf{W} can then be obtained using Eq. (3-10).

$$\mathbf{W} = (\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0)(\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0)^T / (L-1) \quad (3-10)$$

Once the system parameters are determined using least squares on the training data, the model obtained (\mathbf{A} , \mathbf{C} , \mathbf{W}) can be used in the Kalman filter to generate estimates of hand positions from neuronal firing measurements. Essentially, the model proposed here assumes a linear dynamical relationship between current and future trajectory states and

spike counts. Since the Kalman filter formulation requires a reference output from the model, the spike counts are assigned to the output, as they are the only available signals.

The Kalman filter is an adaptive implementation of the Luenberger observer where the observer gains are optimized to minimize the state estimation error variance. In real-time operation, the Kalman gain matrix K Eq. (3-12), is updated using the projection of the error covariance in Eq. (3-11) and the error covariance update in Eq. (3-14). During model testing, the Kalman gain correction is a powerful method for decreasing estimation error. The state in Eq. (3-13) is updated by adjusting the current state value by the error multiplied with the Kalman gain.

$$\mathbf{P}^-(t+1) = \mathbf{A}\mathbf{P}(t)\mathbf{A}^T + \mathbf{W} \quad (3-11)$$

$$\mathbf{K}(t+1) = \mathbf{P}^-(t+1)\mathbf{C}^T (\mathbf{C}\mathbf{P}^-(t+1)\mathbf{C}^T)^{-1} \quad (3-12)$$

$$\tilde{\mathbf{x}}(t+1) = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{K}(t+1)(\mathbf{Y}(t+1) - \mathbf{C}\mathbf{A}\tilde{\mathbf{x}}(t)) \quad (3-13)$$

$$\mathbf{P}(t+1) = (\mathbf{I} - \mathbf{K}(t+1)\mathbf{C})\mathbf{P}^-(t+1) \quad (3-14)$$

Using measured position, velocity, and acceleration as states and neuronal firing counts as model outputs within this recursive framework, this approach may seem to be the best state-of-the-art method available to understand the encoding and decoding between neural activity and hand kinematics. Unfortunately, for BMIs this particular formulation is faced with problems of parameter estimation. The generative model is required to find the mapping from the low-dimensional kinematic parameter state space to the high-dimensional output space of neuronal firing patterns (100+ dimensions). Estimating model parameters from the collapsed space to the high-dimensional neural can be difficult and yield multiple solutions. For this modeling approach, our use of physiologic knowledge in the framework of the model actually complicates the mapping

process. As an alternative, one could disregard any knowledge about the system being modeled and use a strictly data-driven methodology to build the model.

Other BMI research groups have studied and extended the Kalman formulation presented here for neural decoding [42-44]. A multiple model implementation or “switching Kalman filter” has been proposed by Black et al. [43]. Additionally, Particle filtering has been applied to both linear and nonlinear state equations and Gaussian and Poisson models [42, 44]. While these approaches to the BMI mapping problem are novel and interesting, they have produced similar performance results to the standard Kalman formulation.

Black Box

The last I/O model presented is the “black box” model for BMIs. In this case, it is assumed that no physical insight is available for the model. The foundations of this type of time series modeling were laid by Norbert Wiener for applications of gun control during World War II [45]. While military gun control applications may not seem to have a natural connection to BMIs, Wiener provided the tools for building models that correlate a wide variety of inputs (in our case neuronal firing rates) and outputs (hand/arm movements). While this Wiener filter is topographically similar to the PVA algorithm, it is interesting to note that it was developed more than thirty years before Georgopoulos was developing his linear model relating neuronal activity to arm movement direction.

The three input-output modeling abstractions have gained a large support from the scientific community and are also a well established methodology in signal processing and control theory for system identification [27]. Engineers have applied these models for many years to a wide variety of applications and have proven that the method produces

viable phenomenological descriptions when properly applied [46, 47]. One of the advantages of the technique is that it quickly finds, with relatively simple algorithmic techniques, optimal mappings (in the sense of minimum error power) between different time series using a nonparametric approach (i.e. without requiring a specific model for the time series generation). These advantages have to be counter weighted by the abstract (nonstructural) level of the modeling and the many difficulties of the method, such as determining what a reasonable fit, a model order, and a topology is to appropriately represent the relationships among the input and desired response time series.

Finite impulse response filter

The first black box model that we will discuss assumes that there exists a linear mapping between the desired hand kinematics and neuronal firing counts. In this model, the delayed versions of the firing counts, $s(t-l)$, are the bases that construct the output signal. Figure 3-2 shows the topology of the multiple output Wiener filter (WF) where the output y_j is a weighted linear combination of the 10 most recent values⁵ of neuronal inputs s given in Eq. (3.15) [27]. Here y_j can be defined to be any of the single coordinate directions of the kinematic variables HP, HV, HA, or GF. The model parameters are updated using either the optimal linear Least Squares (LS) solution (Wiener Solution) or the LMS algorithm which utilizes stochastic gradient descent [27]. The Wiener solution is given by Eq. (3-16), where R and P_j are the autocorrelation and cross correlation functions, respectively, and d_j is one of the following: hand trajectory, velocity, or gripping force⁶. Using the iterative LMS the algorithm in Eq. (3-17), the model

⁵ In our studies we have observed neuronal activity correlated with behavior for up to 10 lags.

⁶ Each neuronal input and desired trajectory for the WF was preprocessed to have a mean value of zero.

parameters are updated incrementally using η , the constant learning rate, and the error $e(t)=d_x(t)-y_x(t)$.

$$y_j(t) = \mathbf{W}_j \mathbf{s}(t) \quad (3-15)$$

$$\mathbf{W}_j = \mathbf{R}^{-1} \mathbf{P}_j = E(\mathbf{s}^T \mathbf{s})^{-1} E(\mathbf{s}^T \mathbf{d}_j) \quad (\text{Wiener}) \quad (3-16)$$

$$\mathbf{W}_j(t+1) = \mathbf{W}_j(t) + \eta e_j(t) \mathbf{s}(t) \quad (\text{LMS}) \quad (3-17)$$

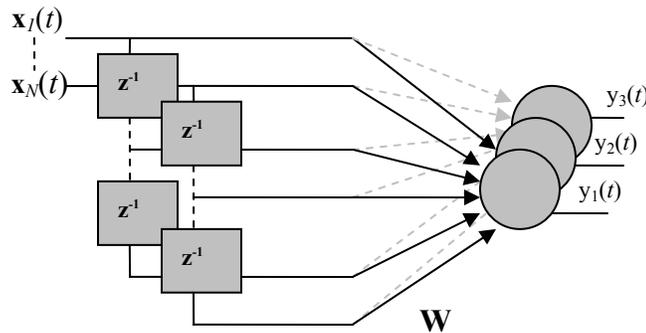


Figure 3-2. FIR filter topology. Each neuronal input \mathbf{s}_N contains a tap-delay line with l taps

Linear filters trained with mean square error (MSE) provide the best linear estimate of the mapping between neural firing patterns and hand position. Even though the solution is guaranteed to converge to the global optimum, the model assumes that the relationship between neural activity and hand position is linear which may not be the case. Furthermore, for large input spaces, including memory in the input introduces many extra degrees of freedom to the model hindering generalization capabilities.

Time-delay neural network

Spatio-temporal nonlinear mappings of neuronal firing patterns to hand position can be constructed using Time-Delay Neural Networks (TDNN). This topology (Fig. 3-3) is more powerful than the linear FIR filter because each of the hidden PEs outputs can be thought of as a nonlinear adaptive basis of the output space utilized to project the high

dimensional data. Then these projections can be linearly combined to form the outputs that will predict the desired hand movements. This architecture consists of a tap delay line memory structure at the input in which past neuronal firing patterns in time can be stored, followed by a nonlinearity. The output of the first hidden layer of the network can be described with the relation $\mathbf{y}_1(t) = f(\mathbf{W}_1 \mathbf{s}(t))$ where $f(\cdot)$ is the hyperbolic tangent nonlinearity $(\tanh(\beta x))^7$. The input vector \mathbf{s} includes l most recent spike counts from N input neurons. In this model the delayed versions of the firing counts, $\mathbf{s}(t-l)$, are the bases that construct the output of the *hidden layer*. The number of delays in the topology should be set that there is significant coupling between the input and desired signal. The output layer of the network produces the hand trajectory $\mathbf{y}_2(t)$ using a linear combination of the hidden states and is given by $\mathbf{y}_2(t) = \mathbf{W}_2 \mathbf{y}_1(t)$. The weights $(\mathbf{W}_1, \mathbf{W}_2)$ of this network can be trained using static backpropagation⁸ with mean square error (MSE) as the learning criterion.

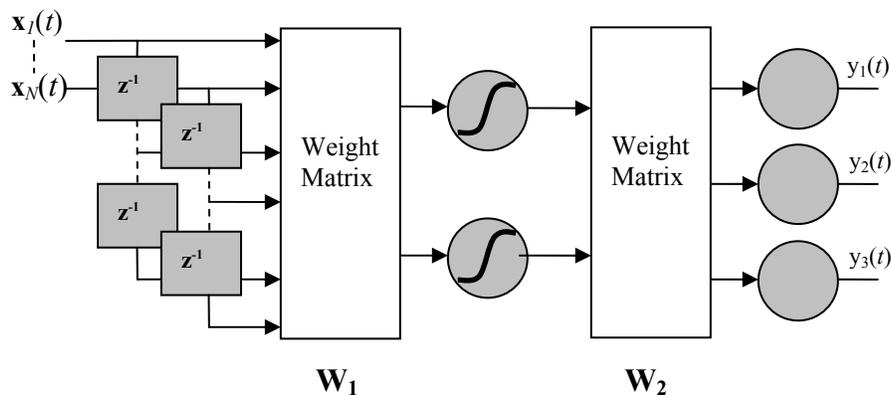


Figure 3-3. Time-delay neural network topology

⁷ The logistic function is another common nonlinearity used in neural networks.

⁸ Backpropagation is a simple application of the chainrule which propagates the gradients through the topology.

While the nonlinear nature of the TDNN may seem as an attractive choice for BMIs, putting memory at the input of this topology presents a difficulty in training and model generalization. Adding memory to the high dimensional neural input introduces many free parameters to train. For example, if a neural ensemble contains 100 neurons with ten delays of memory and the TDNN topology contains five hidden processing elements (PEs), 5000 free parameters are introduced in the input layer alone. Large datasets and slow learning rates are required to avoid overfitting. Untrained weights can also add variance to the testing performance thus decreasing accuracy.

Recurrent multilayer perceptron

The final black box BMI model discussed is potentially the most powerful because it not only contains a nonlinearity but it includes dynamics through the use of feedback. The recurrent multilayer perceptron (RMLP) architecture in Fig. 3-4 consists of an input layer with N neuronal input channels, a fully connected hidden layer of nonlinear processing elements (PEs), (in this case tanh), and an output layer of linear PEs.

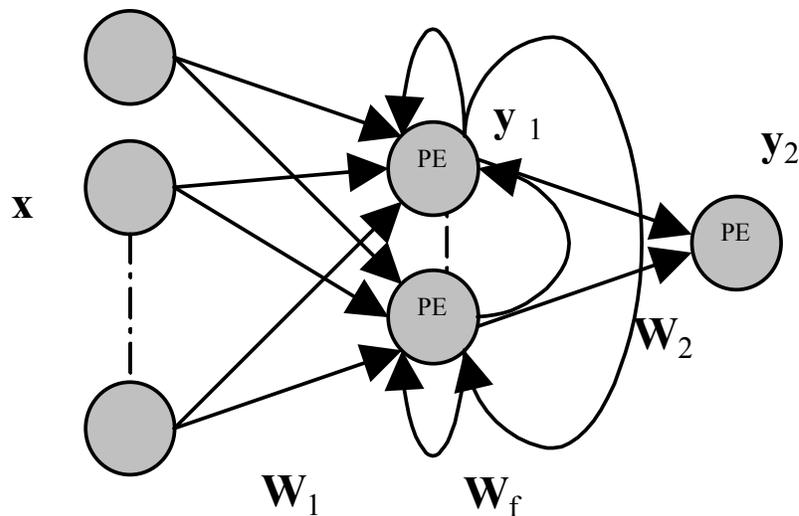


Figure 3-4. Fully connected, state recurrent neural network

Each hidden layer PE is connected to every other hidden PE using a unit time delay. In the input layer equation of Eq. (3-18), the state produced at the output of the first hidden layer is a nonlinear function of a weighted combination (including a bias) of the current input and the previous state. The feedback of the state allows for continuous representations on multiple timescales, and effectively implements a short-term memory mechanism. Here, $f(\cdot)$ is a sigmoid nonlinearity (in this case \tanh), and the weight matrices \mathbf{W}_1 , \mathbf{W}_2 , and \mathbf{W}_f , as well as the bias vectors \mathbf{b}_1 and \mathbf{b}_2 are again trained using synchronized neural activity and hand position data. Each of the hidden PEs outputs can be thought of as a nonlinear adaptive basis of the output space utilized to project the high dimensional data. These projections are then linearly combined to form the outputs of the RMLP that will predict the desired hand movements as shown in Eq. (3-19). One of the disadvantages of the RMLP when compared with the Kalman filter is that there is no known closed form solution to estimate the matrices \mathbf{W}_f , \mathbf{W}_1 and \mathbf{W}_2 in the model; therefore, gradient descent learning is used. The RMLP can be trained (see appendix A, B and C for details) with backpropagation through time (BPTT) or real-time recurrent learning (RTRL) [48]. Later, in Chapter 4, the training formulation differences between the Kalman filter and the RMLP will be compared and contrasted.

$$\mathbf{y}_1(t) = f(\mathbf{W}_1\mathbf{x}(t) + \mathbf{W}_f\mathbf{y}_1(t-1) + \mathbf{b}_1) \quad (3-18)$$

$$\mathbf{y}_2(t) = \mathbf{W}_2\mathbf{y}_1(t) + \mathbf{b}_2 \quad (3-19)$$

Development and Testing of BMI Models

Reaching Task Performance

Preliminary BMI modeling studies were focused on comparing the performance of linear, generative, nonlinear, feedforward, and dynamical models for the hand reaching

motor task. The four models studied included the FIR-Wiener filter, TDNN (the nonlinear extension to the Wiener filter), Kalman filter, and RMLP. Since each of these models employs very different principles and has different mapping power, it is expected that they will perform differently; however, the extent to which they differ remains unknown. Here, a comparison of gray and black box BMI models for a hand reaching task will be provided.

Topology and training complexity comparisons

One of the most difficult aspects of modeling for BMIs is the dimensionality of the neuronal input. Because of this large dimensionality, even the simplest models contain topologies with thousands of free parameters. Moreover, the BMI model is often trying to approximate relatively simple trajectories resembling sine waves which practically can be approximated with only two free parameters. Immediately we are faced with avoiding overfitting the data. Large dimensionality also has an impact on the computational complexity of the model which can require thousands more multiplications, divisions, and function evaluations. This is especially a problem if we wish to implement the model in low-power portable digital signal processors (DSP). Here we will assess each of the four BMI models in terms of their number of free parameters and computational complexity.

Model overfitting is often described in terms of prediction risk (PR) which is the expected performance of a topology when predicting new trajectories not encountered during training [49]. Several estimates of the PR for linear models have been proposed in the literature [50-53]. A simple way to develop a formulation for the prediction risk is to assume the quadratic form in Eq (3-20) where e is the training error for a model with Φ parameters and N training samples. In this quadratic formulation, we can consider an

optimal number of parameters, Φ^{Opt} , that minimizes the PR. We wish to estimate how the prediction risk will vary with Φ which can be given by a simple Taylor series expansion of Eq. (3-20) around Φ^{Opt} as performed in [53]. Manipulation of the Taylor expansion will yield the general form in Eq. (3-21). Other formulations for the prediction risk include the generalized cross-validation (GCV) and Akaike's final prediction error (FPE) given in Eqs. (3-22) and (3-23). The important characteristic of Eqs (3-21 to 3-23) is that they all involve the interplay of the number of model parameters to the number of training samples. In general, the prediction risk increases as the number of model parameters increases.

$$PR = E[e^2(\Phi_N)] \quad (3-20)$$

$$PR \approx e^2 \left(1 + \frac{\Phi}{N} \right) \quad (3-21)$$

$$GCV = \frac{e^2}{\left(1 - \frac{\Phi}{N} \right)^2} \quad (3-22)$$

$$FPE = e^2 \left(\frac{1 + \frac{\Phi}{N}}{1 - \frac{\Phi}{N}} \right) \quad (3-23)$$

The formulations for the prediction risk presented here have been extended to nonlinear models [54]. While the estimation of the prediction risk for linear models is rather straight-forward, in the case nonlinear models the formulation is complicated by two factors. First, the nonlinear formulation involves computing effective number of parameters (a number that differs from the true number of parameter in the model) which is nontrivial to estimate since it depends on the amount of model bias, model

nonlinearity, and amount of regularization used in training [54]. Second, the formulation involves computing the noise covariance matrix of the desired signal; another parameter that is nontrivial to compute especially in the context of BMI hand trajectories.

For the reaching task dataset, all of the models utilize 104 neuronal inputs as shown in Table 3-1. The first encounter with an explosion in the number of free parameters occurs for both the Wiener filter and TDNN since they contain a 10 tap-delay line at the input. Immediately the number of inputs is multiplied by 10. The TDNN topology has the greatest number of free parameters, 5215, of the feedforward topologies because the neuronal tap-delay memory structure is also multiplied by the 5 hidden processing elements following the input. The Wiener filter, which does not contain any hidden processing elements, contains 3120 free parameters. In the case of the Kalman filter, which is the largest topology, the number of parameters explodes due to the size of the **A** and **C** matrices since they both contain the square of the dimensionality of the 104 neuronal inputs. Finally, the RMLP topology is the most frugal since it moves its memory structure to the hidden layer through the use of feedback yielding a total of 560 free parameters.

To quantify how the number of free parameters effects model training time, a Pentium 4 class computer with 512 MB DDR RAM, the software package NeuroSolutions for the neural networks [55], and Matlab for computing the Kalman and Wiener solution were used to train the models. The training times of all four topologies is given in Table 3-1. For the Wiener filter, the computation of the inverse of a 1040x1040 autocorrelation matrix took 47 seconds in Matlab which is optimized for matrix computations. For the neural networks, the complete set of data is presented to the

learning algorithm in several iterations called epochs. In NeuroSolutions, whose programming is based in C, 20010 samples were presented 130 and 1000 times in 22 min 15 sec. and 6 min. 35 sec. for the TDNN and RMLP respectively [55]. The TDNN was trained with backpropagation and the RMLP was trained with backpropagation through time (BPTT) [48] with a trajectory of 30 samples and learning rates of 0.01, 0.01, and 0.001 for the input, feedback, and output layers respectively. Momentum learning was also implemented with a rate of 0.7. One hundred Monte Carlo simulations with different initial conditions were conducted of neuronal data to improve the chances of obtaining the global optimum. Of all the Monte Carlo simulations, the network with the smallest error achieved a MSE of 0.0203 ± 0.0009 . A small training standard deviation indicates the network repeatedly achieved the same level of performance. Neural network training was stopped using the method of cross-validation (batch size of 1000 pts.), to maximize the generalization of the network [56]. The Kalman proved to be the slowest to train since the update of the Kalman gain requires several matrix multiplies and divisions. In these simulations, the number of epochs chosen was based upon performance in a 1000 sample cross-validation set which will be discussed in the next section. To maximize generalization during training, ridge regression, weight decay, and slow learning rates were also implemented.

The number of free parameters is also related to the computational complexity of each model given in Table 3-2. The number of multiplies, adds, and function evaluations describe how demanding the topology is for producing an output. The computational complexity especially becomes critical when implementing the model in a low-power portable DSP, which is the intended outcome for BMI applications. In Table 3 define N_θ ,

Table 3-1. Model parameters

	Wiener Filter	TDNN	Kalman	RMLP
Training Time	47 sec.	22 min. 15 sec.	2 min. 43 sec.	6 min. 35 sec.
Number of Epochs	1	130	1	1000
Crossvalidation	N/A	1000pts.	N/A	1000pts.
Number of Inputs	104	104	104	104
Number of Tap-Delays	10	10	N/A	N/A
Number of Hidden PEs	N/A	5	113 (states)	5
Number of Outputs	3	3	9	3
Number of Adapted Weights	3120	5215	12073	560
Regularization	0.1 (RR)	1e-5 (WD)	N/A	1e-5 (WD)
Learning Rates	N/A	1e-4 (Input) 1e-5 (Output)	N/A	1e-2 (Input) 1e-2 (Feedback) 1e-3 (Output)

t , d , and N_I to be the number of inputs, tap-delays, number of output, and number of hidden PEs, respectively. In this case, only the number of multiplications and function evaluations are presented since the number of adds is essentially identical to the number of multiplications. Again it can be seen that demanding models contain memory in the neural input layer. With the addition of each neuronal input the computational complexity of the Wiener filter increases by 10 and the TDNN by 50. The Kalman filter is the most computationally complex ($O((N_0 + 9)^3)$) since both the state transition and output matrix contain dimensionality of the neuronal input. For the neural networks, the number of function evaluations is not as demanding since they contain only five for both the TDNN and RMLP. Comparing the neural network training times, also exemplifies the computational complexity of each topology; the TDNN (the most computationally complex) requires the most training time and allows only a hundred presentations of the

training data. As a rule of thumb to overcome these difficulties, BMI architectures should avoid the use of memory structures at the input.

Table 3-2. Model computational complexity

	Multiplications	Function Evaluations
Wiener Filter	$N_0 \times t \times d$	N/A
TDNN	$N_0 \times t \times N_1 + N_1 \times d$	N1
Kalman	$O((N_0 + 9)^3)$	N/A
RMLP	$N_0 \times N_1 + N_1 \times d + N_1 \times N_1$	N1

Regularization, weight decay, and cross validation

The primary goal in BMI modeling experiments is to produce the best estimates of HP, HV, and GF from neuronal activity that has not been used to train the model. This testing performance describes the generalization ability of the models. To achieve good generalization for a given problem, the two first considerations to be addressed are the choice of model topology and training algorithm. These choices are especially important in the design of BMIs because performance is dependent upon how well the model deals with the large dimensionality of the input as well as how the model generalizes in nonstationary environments. The generalization of the model can be explained in terms of the bias-variance dilemma of machine learning [57], which is related to the number of free parameters of a model. The MIMO structure of BMIs built for the data presented here were shown to have as few as several hundred to as many as several thousand free parameters. On one extreme if the model does not contain enough parameters, there are too few degrees of freedom to fit the function to be estimated which results in bias errors. On the other extreme, models with too many degrees of freedom tend to overfit the function to be estimated. In terms of BMIs, models tend to err on the latter because of the large dimensionality of the input. We discussed earlier that BMI model overfitting is

especially a problem in topologies where memory is implemented in the neural input layer. With each new delay element, the number of free parameters will scale with the number of input neurons as in the FIR filter and TDNN network.

To handle the bias-variance dilemma we would like to effectively eliminate extraneous model parameters or reduce the value of Φ in Eqs. (3-21 to 3-23). One could use the traditional Akaike or BIC criteria; however, the MIMO structure of BMIs excludes these approaches [58]. As a second option, during model training regularization techniques could be implemented that attempt to reduce the value of unimportant weights to zero and effectively prune the size of the model topology [59].

In BMI experiments we are not only faced with regularization issues but we must also consider ill-conditioned model solutions that result from the use of finite datasets. For example, computation of the optimal solution for the linear Wiener filter involves inverting a poorly conditioned input correlation matrix that results from sparse neural firing data that is highly variable. One method of dealing with this problem is to use the pseudoinverse. However, since we are interested in both conditioning and regularization we chose to use ridge regression (RR) [60] where an identity matrix is multiplied by a white noise variance and is added to the correlation matrix. The criterion function of RR is given by,

$$J(\mathbf{w}) = E[\|\mathbf{e}\|^2] + \delta\|\mathbf{w}\|^2 \quad (3-24)$$

where \mathbf{w} are the weights, \mathbf{e} is the model error, and the additional term $\delta\|\mathbf{w}\|^2$ smoothes the cost function. The choice of the amount of regularization (δ) plays an important role in the generalization performance and for larger deltas performance can suffer because SNR is sacrificed for smaller condition numbers. It has been proposed by Larsen et al. that δ

can be optimized by minimizing the generalization error with respect to δ [47]. For other model topologies such as the TDNN, RMLP, and the LMS update for the FIR, weight decay (WD) regularization is an on-line method of RR to minimize the criterion function in Eq. (3-24) using the stochastic gradient, updating the weights by Eq. (3-25).

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta \nabla(J) - \delta \mathbf{w}(n) \quad (3-25)$$

Both RR and weight decay can be viewed as the implementations of a Bayesian approach to complexity control in supervised learning using a zero-mean Gaussian prior [61].

A second method that can be used to maximize the generalization of a BMI model is called cross-validation. Developments in learning theory have shown that during model training there is a point of maximum generalization after which model performance on unseen data will begin to deteriorate [56]. After this point, the model is said to be *overtrained*. To circumvent this problem, a cross-validation set can be used to indicate an early stopping point in the training procedure. To implement this method, the training data is divided into a training set and a cross-validation set. Periodically during model training, the cross-validation set is used to test the performance of the model. When the error in the validation set begins to increase, the training should be stopped.

Performance metrics

The most common metric for reporting results in BMI is through the correlation coefficient (CC) computed between the desired trajectory and the model output. While the CC is the gold standard in BMI research, it is not free from biases. The CC is a good measure of linearity between two signals; however, in the testing of our BMI models, we often encountered signals that were linearly related but also contained constant biases or spatial errors (in millimeters) incurred in the tasks. In [12] we proposed an additional

metric to quantify the accuracy of the mapping by computing the signal to error ratio also between the actual and estimated hand trajectories. The SER (square of the desired signal divided by the square of the estimation error) gives a measure of the accuracy of estimated position in terms of the error variance. High SERs are desired since they are produced when the estimated output error variance is small.

Throughout our BMI modeling studies we also observed that CC and SER do not relate directly to errors in the physical world. In a BMI application, we are interested in quantifying the distance (e.g. in millimeters) between the target position and the BMI output. In order to evaluate the performance of a BMI we propose a more specific figure of merit that emphasizes the accuracy of the reach portion of this movement, which we call the Cumulative Error Metric (CEM). CEM was inspired by the receiver operating characteristics (ROC) a hallmark in detection theory. CEM is defined as the probability of finding errors less than a given size (in millimeters) along the trajectory.

To use this metric, plot the probability of finding a network output within a 3-D radius around the desired data point, defined by

$$CEM = P[\|\mathbf{e}\|_2 \leq r] \quad (3-26)$$

where $\mathbf{e} = \mathbf{d} - \mathbf{y}$. It is therefore very easy to visually assess the quality of an algorithm in terms of maximum error (the right extent of the curve) and how probable are large errors (the closer the CEM is to the left top corner of the plot the better). This plot contains a lot of information, since it tells how accurate the BMI is throughout the test set or simply in the movement trajectory (very much like a receiver operating characteristic used in detection). It should also be noted that similar curves can have large sample-by-sample deviations, when for instance one curve is delayed with respect to the other. Therefore

CEM is judged as a sensitive metric when a delay between the spike data and the hand positions exists.

Test set performance

The performance of the four BMI models was evaluated in terms of CC, SER, and CEM. Each of the models is approximating a trajectory as illustrated in Fig. 3-5. The reaching task which consists of a reach to food and subsequent reach to the mouth is embedded in periods where the animal's hand is at rest as shown by the flat trajectories to the left and right of the movement. Since we are interested in how the models perform in each mode of the movement we present CC, SER, and CEM curves both for movement and rest periods. The performance metrics are also computed using a sliding window of 40 samples (4 seconds) so that an estimate of the standard deviation could be quantified. The window length of 40 was selected because each movement spans about 4 seconds.

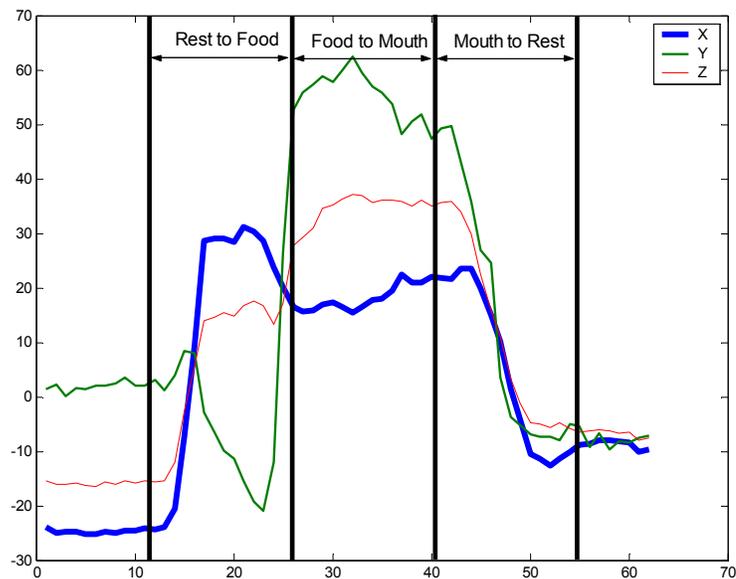


Figure 3-5. Reaching movement trajectory

In testing, all model parameters were fixed and 3,000 consecutive bins (300 secs) of novel neuronal data were fed into the models to predict new hand trajectories. Fig. 3-6 shows the output (bold traces) of the four topologies for three desired reaching movements (thin traces). While only three movements are presented for simplicity, it can be shown that during the short period of observation (5 min), there is no noticeable degradation of the model fitting across time. From the plots it can be seen that qualitatively all three topologies do a fair job at capturing the reach to the food and the initial reach to the mouth. However, the FIR filter, TDNN and Kalman filter cannot maintain the peak values of HP at the mouth position. Additionally the FIR filter and Kalman have smooth transitions between the food and mouth while the RMLP and TDNN sharply change their positions in this region. The most “noisy” trajectories were produced by the FIR and Kalman while the TDNN produce sharp changes in trajectory during the resting periods. The RMLP generated the smoothest trajectories during resting and maintained the peak values compared to the other models.

The reaching task testing metrics are presented in Table 3-3. It can be seen in the table that the CC can give a misleading perspective of performance since all the models produce approximately the same values. Nevertheless, the Kolmogorov-Smirnov (K-S) for a p value of 0.05 is used to compare the correlation coefficients with the simplest model, the FIR filter. The TDNN, Kalman and RMLP all produced CC values that were significantly different than the FIR filter and the CC values itself can be used to gauge if the difference is significantly better. The RMLP had the highest SER indicating that on average it produced smaller errors. All four models have poor resting CC values which

can be attributed to the output variability in the trajectory (i.e. there is not a strong linear relationship between the output and desired trajectories)

Overall, the probability of finding small errors is highest for the RMLP as shown in the CEM curves of Fig. 3-7. While the RMLP follows in 2nd place for a high probability of generating small errors, it is overcome by the FIR which performs better for larger errors. The Kalman filter has the worst overall CEM curve. In the movements of the reaching task though, the Kalman always outperforms the FIR and for moderate and large errors it outperforms the TDNN and RMLP. The RMLP outperforms the TDNN in the movements as corroborated with the trajectories (Fig. 3-6).

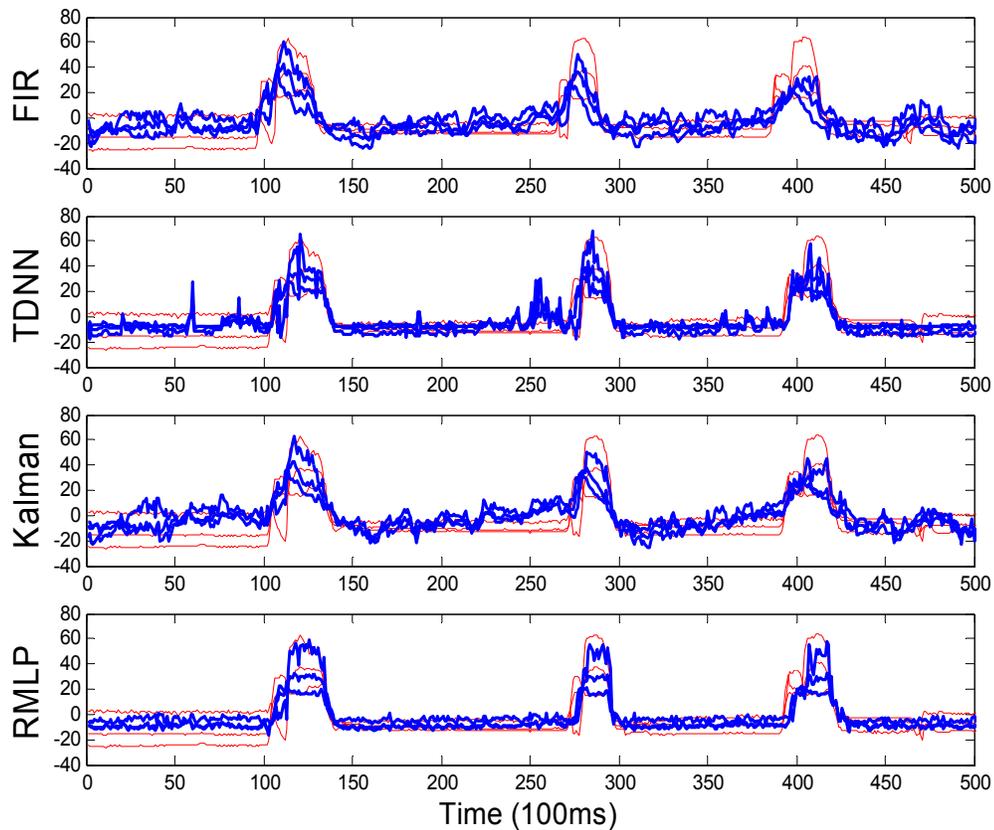


Figure 3-6. Testing performance for a three reaching movements (Belle)

Table 3-3. Reaching task testing CC and SER (Belle)

	Linear Model (FIR)	TDNN	Kalman Filter	RMLP
Correlation Coefficient (movement)	0.83±0.09	0.80±0.17	0.83±0.11	0.84±0.15
CC K-S Test (movement)	0	1	1	1
SER (dB) (movement)	5.97±1.31	5.45±2.52	5.30±2.07	6.93±2.55
Correlation Coefficient (rest)	0.10±0.29	0.04±0.25	0.03±0.26	0.06±0.25
CC K-S Test (rest)	0	1	1	1
SER (dB) (rest)	3.16±2.69	3.00±5.55	0.93±3.69	6.99±3.95

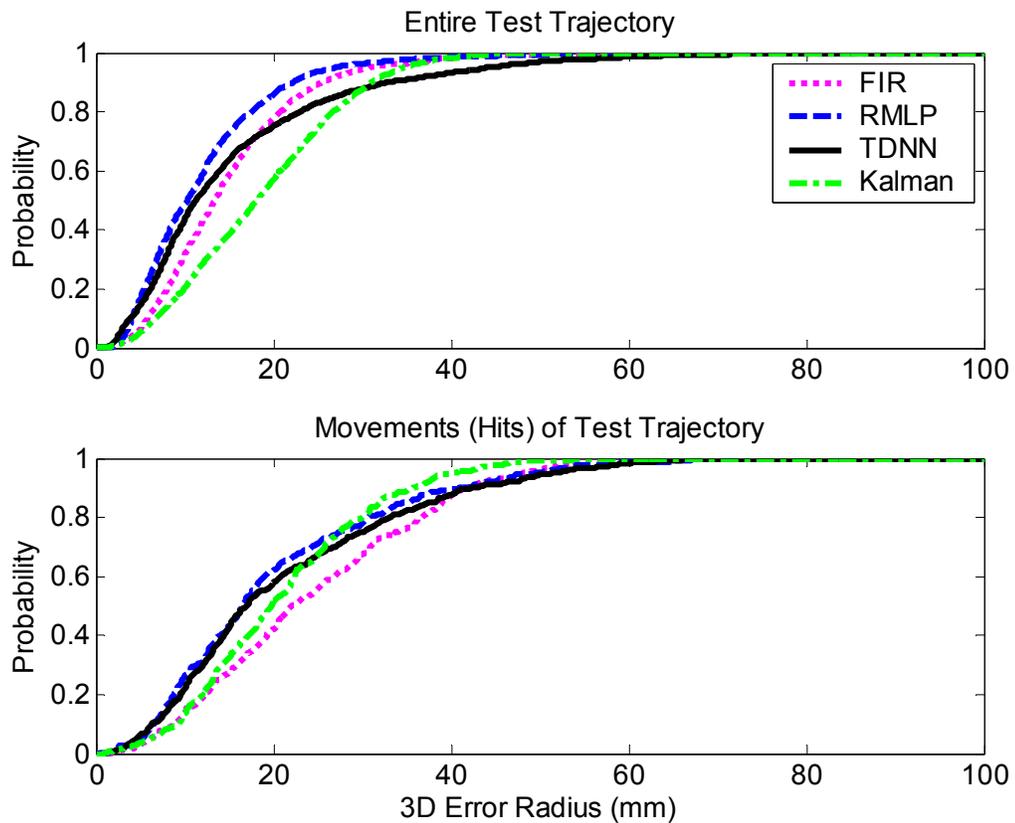


Figure 3-7. Reaching task testing CEM (Belle)

The training of the four models was repeated for the second owl monkey named Carmen. In this experiment, the model topologies (FIR, TDNN, Kalman, and RMLP) and the type of reaching movement were held as controls. The only variable was the neuronal recordings extracted from the behaving primate. In this case, only half the number of cells was (54 neurons) collected compared to the previous experiment. The reduction in the number of inputs resulted from not sampling the PP and M1/PMd ipsi cortices (see Table. 2-1).

In Fig. (3-8) the output trajectories from the four models are presented as bold traces. Qualitatively we see an immediate decrease in performance. All of the generated trajectories are much noisier and miss capturing peaks of the movements. Specifically all four models cannot produce an increase in any of the coordinated during the reach to the food.

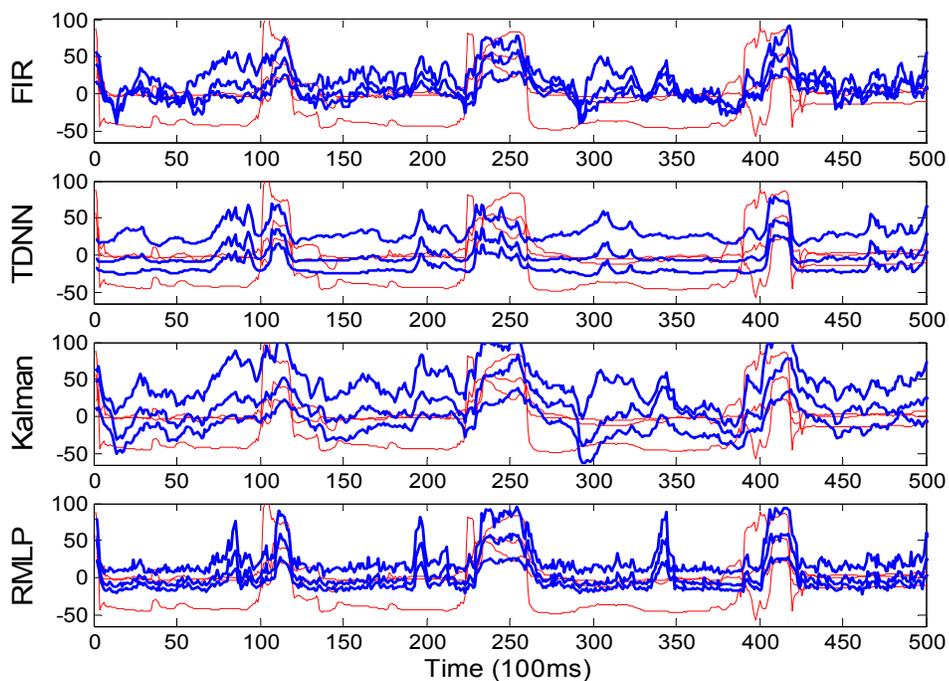


Figure 3-8. Testing performance for a three reaching movements (Carmen)

Quantification of this reduction in performance is given in Table 3-4 where the CC values showed a 30% drop. The worst performance during the resting periods was produced by the Kalman filter which produced a SER of -3.11 dB. The relative trends in performance were again maintained; the Kalman and FIR produced noisy trajectories, the TDNN produced sharp spikes, and the RMLP and Kalman maintained the peak values. The CEM curves (Fig. 3-9) show again that the RMLP outperforms all other models for small errors but for large errors the RMLP is comparable with the FIR.

At this point for a reaching BMI task, the expressive power of each of the four models has been demonstrated. Depending upon the use of nonlinearity, dynamics, or tap-delay memory structures different levels of performance can be obtained in terms of trajectory smoothness, and reconstruction of peak values. While some of these differences may be subtle, we observed the largest change in performance when switching between neuronal inputs. The use of a reduced number of neurons and cortical areas sampled had a dramatic effect on model performance even though the topologies and task remained fixed.

Table 3-4. Reaching task testing CC and SER (Carmen)

	Linear Model (FIR)	TDNN	Kalman Filter	RMLP
Correlation Coefficient (movement)	0.64±0.24	0.42±0.38	0.63±0.28	0.65±0.24
CC K-S Test (movement)	0	1	0	0
SER (dB) (movement)	4.82±3.29	4.21±3.75	3.46±3.80	5.04±3.67
Correlation Coefficient (rest)	0.03±0.28	0.18±0.32	0.02±0.30	0.04±0.27
CC K-S Test (rest)	0	1	1	1
SER (dB) (rest)	-1.03±3.16	0.22±3.32	-3.11±3.72	0.22±3.51

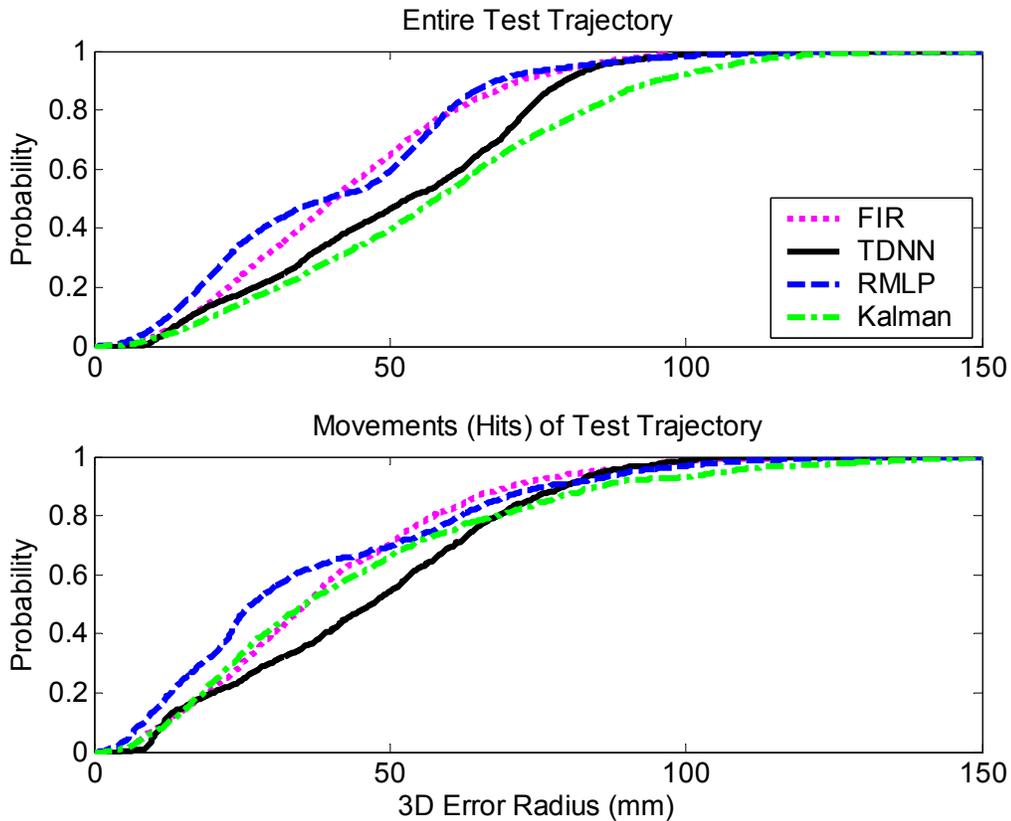


Figure 3-9. Reaching task testing CEM (Carmen)

Cursor Control Task

In light of the results produced in the reach task experiments, BMI modeling was extended to the cursor control task again for two primates (in this case the Rhesus monkeys have a more sophisticated nervous system). In experiments for Ivy and Aurora, the results were not as dramatic but the relative features in the model outputs were the same. Again the FIR had difficulties maintaining the peaks of the trajectories (see Fig. 3-10 Y-Coordinate) and the TDNN had sharp changes in the trajectories (see Fig. 3-11 Y-Coordinate). Both the Kalman and RMLP produce the best reconstructions of the trajectories in terms of smoothness and capturing the peaks of the movements. In Tables 3-5 and 3-6 this result is corroborated with the high CC and SER values. Overall, the

TDNN, Kalman, and RMLP all produce CC values that are significantly better than the FIR filter as indicated by the K-S test. In both tables we present the X and Y coordinates separately since there is a noticeable difference in the level of performance for each coordinate. This discrepancy can be attributed to the differences in the nature of the movements (i.e. in each experiment one coordinate primarily contains either a low or high frequency component). Performance may be dependent upon the model's ability to cope with both characteristics. Another possible explanation is that the cursor control experimental paradigm involves a coordinate transformation for the primate. Manipulandum movements are made in a plane parallel to the floor while the target cursor is being placed on a computer monitor whose screen is parallel to the walls. Depending on how the experimenters at Duke University defined the coordinate axes, one coordinate is the same in both planes while the other is rotated 90°.

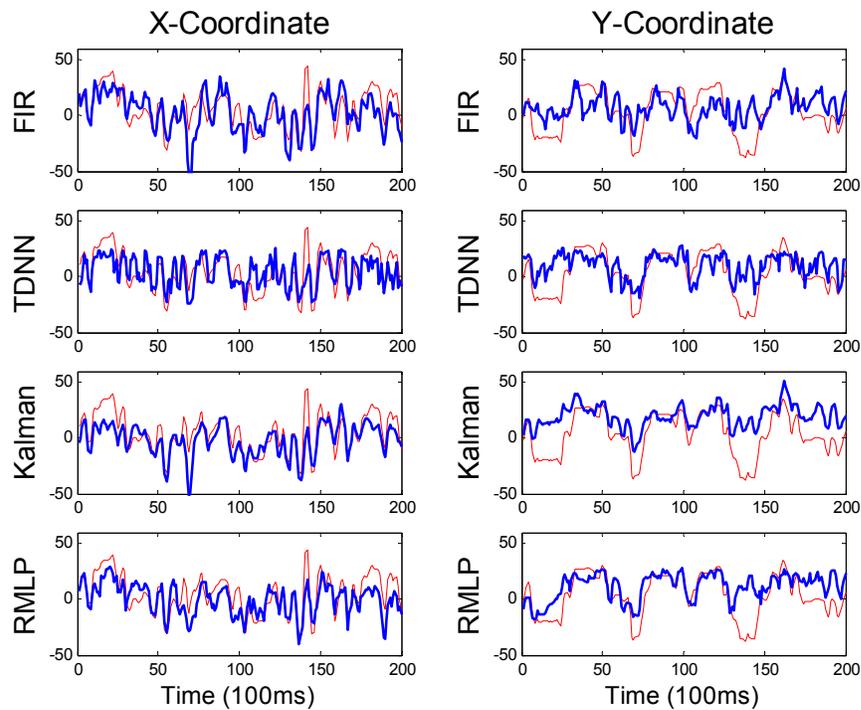


Figure 3-10. Testing performance for a three reaching movements (Ivy)

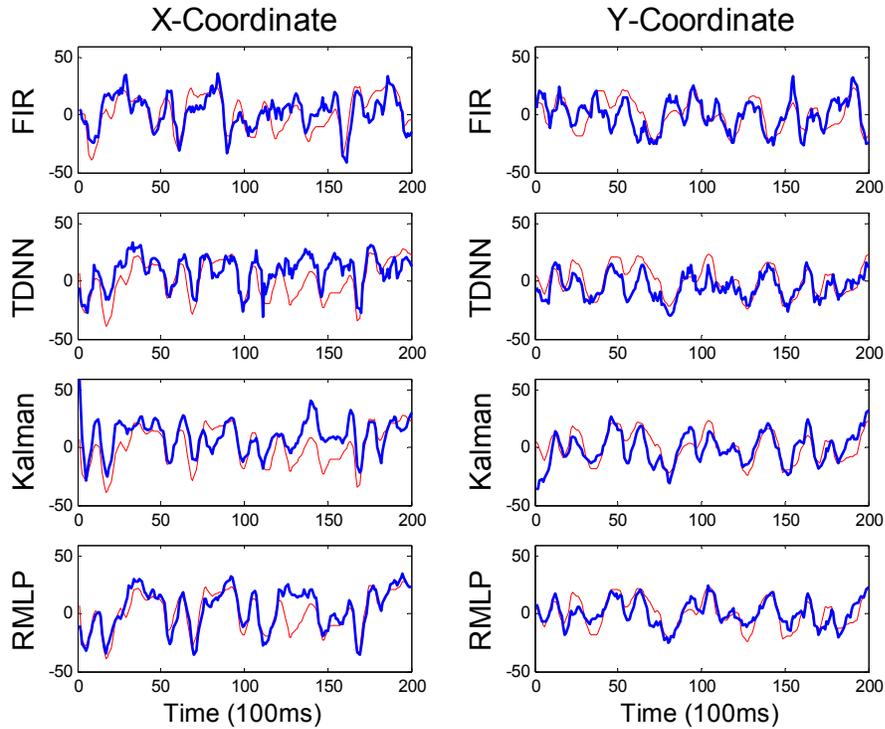


Figure 3-11. Testing performance for a three reaching movements (Aurora)

Table 3-5. Reaching task testing CC and SER (Ivy)

		FIR	TDNN	Kalman	RMLP
Correlation Coefficient	X	0.64±0.16	0.58±0.12	0.66±0.18	0.65±0.17
	Y	0.40±0.24	0.47±0.23	0.58±0.23	0.46±0.31
CC K-S Test	X	0	1	1	0
	Y	0	1	1	1
SER (dB)	X	1.50±2.29	1.35±1.42	1.97±2.10	1.58±2.62
	Y	0.10±2.09	0.99±2.52	0.08±3.76	0.62±3.93

Table 3-6. Reaching task testing CC and SER (Aurora)

		FIR	TDNN	Kalman	RMLP
Correlation Coefficient	X	0.65±0.21	0.69±0.17	0.62±0.26	0.66±0.24
	Y	0.77±0.15	0.79±0.11	0.82±0.11	0.79±0.16
CC K-S Test	X	0	1	1	1
	Y	0	1	1	1
SER (dB)	X	1.34±2.45	16.11±2.18	15.65±2.33	16.99±2.48
	Y	3.24±2.70	13.80±1.97	14.56±2.63	14.30±2.80

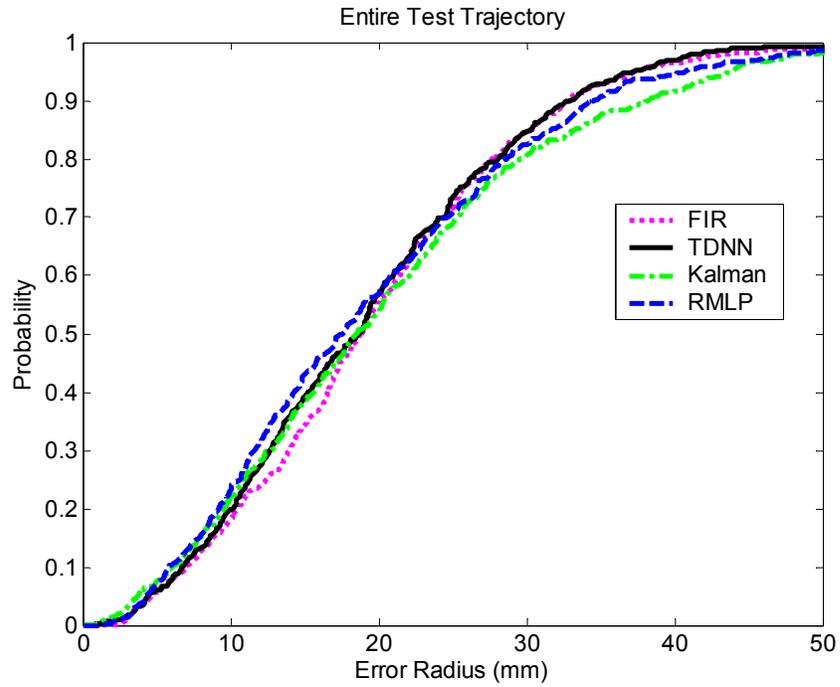


Figure 3-12. Reaching task testing CEM (Ivy)

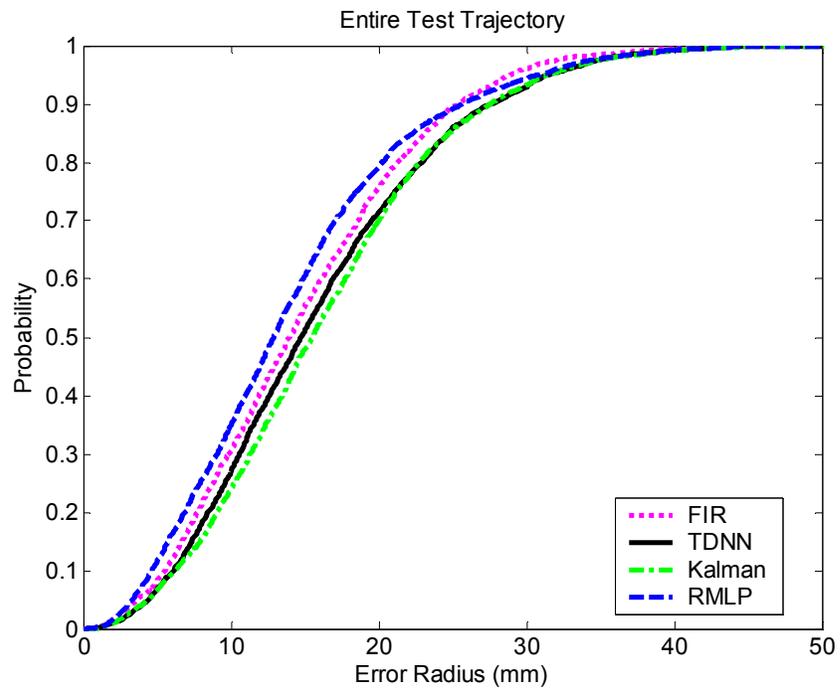


Figure 3-13. Reaching task testing CEM (Aurora)

In terms in CEM, for both sessions (Figs. 3-13 and 3-14) all four models do not deviate much from each other indicating that the performance is basically the same for the models. For a given probability, the maximum difference in error radius was four millimeters. While the variation in each model's CEM small, the slope of the curves for each primate varied greatly. For Ivy the probability increased 0.03 percent per millimeter increase in radius while Aurora doubled that amount for 0.06 percent pre millimeter. Again we attribute the increase in performance to differences in the cells from the of sampled cortices. For ivy only three cortices (M1, SMA, and PP) were used while Aurora's data included (M1, S1, SMA, PMd, and M1ipsi). Further analysis of the contributions of each cortex will be presented later in this dissertation.

Discussion

With each trained model, we evaluated the performance with standard metrics such as correlation coefficients (CC) and compared them with new metrics which included signal to error ratio (SER), and the cumulative error metric (CEM). This study showed that each model's performance can vary depending on the task, number of cells, sampled cortices, species, and individual. For the reaching movements, the nonlinear RMLP performed significantly better than other linear models. However, for the cursor control task all models performed similarly in terms of the performance metrics used. Examples of the differences were demonstrated in the smoothness of the trajectories as well as the ability to capture the peaks of the movements. Variability in these results indicates a need to further study model performance in both more controlled neuronal/behavioral datasets that include a variety of motor tasks, trajectories, and dynamic ranges.

In BMI experiments, we are seeking movement trajectories that are similar real hand movements. In particular, we desire accurate smooth trajectories. In the experiments

presented here, the nonlinear, dynamical model (RMLP) produced the most realistic trajectories compared to the FIR, TDNN and Kalman filter. The ability of this model to perform better may result from the use of saturating nonlinearities in the large movement space as well as the ability to use neuronal inputs at multiple timescales through the use of feedback in the hidden layer. In all models studied though, performance (to different degrees) was also affected by the order of the model (i.e. the number of free parameters). This was especially true for models that implemented memory structures in the input space such as the linear models, and TDNN. The RMLP can overcome this issue by reducing the number of free parameters by shifting the memory structure to the hidden layer through the use of feedback.

With the performance results reported in these experiments, we can now discuss practical considerations when building BMIs. By far the easiest model to implement is the Wiener filter. With its quick computation time and straightforward linear mathematical theory it is clearly an attractive choice for BMIs. We can also explain its function in terms of simple weighted sums of delayed versions of the ensemble neuronal firing (i.e. it is correlating neuronal activity with HP). However from the trajectories in Figs. 3-6, 3-8, 3-10, and 3-11, the output is noisy and does not accurately capture the details of the movement. We can first attribute these errors to the solution obtained from inverting a poorly conditioned autocorrelation matrix and second to the number of free parameters in the model topology. While we may think that by adding nonlinearity to the Wiener filter topology as in the TDNN we can obtain a more powerful tool, we found out that the large increase in the number of free parameters overshadowed the increase in performance. We have found that training the TDNN is slow and tedious and subject to

getting trapped in local minima. Moving to a Kalman based training procedure we thought that the online Kalman gain parameter improve performance; however this technique suffered from parameter estimation issues. Knowledge of these training and performance issues leads us to the RMLP. With the choice of moving the memory structure to the hidden layer, we immediately gain a reduction in the number of free parameters. This change is not without a cost since the BPTT training algorithm is more difficult to implement than for example the Wiener solution. Nevertheless, using a combination of dynamics and nonlinearity in the hidden layer also allowed the model to accurately capture the quick transitions in the movement as well as maintain the peak hand positions at the mouth. Capturing these positions resulted in larger values in the correlation coefficient, and SER. While the RMLP was able to outperform the other three topologies, it is not free from error; the output is still extremely noisy for applications of real BMIs (imagine trying to grasp a glass of water). The search for the right modeling tools and techniques to overcome the errors presented here are the subject of future research for optimal signal processing for brain machine interfaces.

CHAPTER 4 RMLP MODEL GENERALIZATION

Motivation for Studying the RMLP

In the previous chapter, the performance of four models (grey and black box) was compared for BMI experiments involving reaching and cursor tracking tasks. The goal was to compare and contrast how model topologies and training methodologies effect trajectory reconstruction for BMIs. Evaluation of model performance using CC, SER, and CEM indicated that the RMLP is a better choice compared to the other models. Here we continue evaluation of the RMLP by discussing generalization or the ability of the RMLP to continuously produce accurate estimates of the desired trajectory over long testing sets of novel data. First we would like to discuss in further detail the list below indicating why the RMLP is an appropriate choice for BMI applications.

- RMLP topology has the desired biological plausibility needed for BMI design
- The use of nonlinearity and dynamics gives the RMLP a powerful approximating capability
- RMLP produced equivalent or better CC, SER, and CEM with the fewest number of model parameters

While the RMLP may first appear to be an “off the shelf” black box model, comparison to Todorov’s mechanistic model reveals that it has the desired biological plausibility needed for BMI design. To illustrate the plausibility we propose a general (possibly nonlinear) state space model implementation that corresponds to the *representation* interpretation of Todorov’s model for neural control:

$$\begin{aligned} \mathbf{s}(t) &= \mathbf{g}(\mathbf{s}(t-1)) + \mathbf{w}(t-1) && \text{(State dynamics)} \\ \mathbf{x}(t) &= \mathbf{h}(\mathbf{s}(t)) + \mathbf{v}(t) && \text{(Output mapping)} \end{aligned} \quad (4-1)$$

In Eq. (4-1), the state vector $\mathbf{s}(t)$ can include the mechanic variables position, velocity, acceleration, and force. A state-space, linear filter from which the Kalman is one example is a special case of (4-1) when $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$ are linear. For BMIs, the output vector $\mathbf{x}(t)$ must consist of the neuronal activity since in testing mode (after optimization of the models $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$ is done) the BMI will operate using only the neuronal activity. The modeling errors and measurement noise can be summarized by the two noise terms $\mathbf{w}(t)$ and $\mathbf{v}(t)$. In comparison, the RMLP approach for the BMI involves the *cause* interpretation of Todorov's model. Differences in the *representation* and *cause* interpretations result in very different training procedures for both topologies. Referring Eq. (3-19) the model output \mathbf{y}_2 consists only of the hand position; however, the RMLP must learn to build an efficient internal dynamical representation of the other mechanical variables (velocity, acceleration and force) through the use of feedback. In fact, in the RMLP model, we can regard the hidden state vector (\mathbf{y}_1 in Eq. 3-18) as the RMLP representation of these mechanical variables driven by the neural activity in the input (\mathbf{x}). Hence, the dynamical nature of Todorov's model is implemented through the nonlinear feedback in the RMLP. The output layer is responsible for extracting the position information from the representation in \mathbf{y}_1 using a linear combination. An interesting analogy exists between the output layer weight matrix \mathbf{W}_2 (Eq. 3-19) in the RMLP and the matrix \mathbf{U} (Eq. 3-4) in Todorov's model. This analogy stems from the fact that each column of \mathbf{U} represents a direction in the space spanning the mixture of mechanical variables to which the corresponding individual neuron is cosine-tuned, which is a natural

consequence of the inner product. Similarly, each column of \mathbf{W}_2 represents a direction in the space of hand position to which a nonlinear mixture of neuronal activity is tuned.

While the state and output equations of the RMLP and Kalman filter can be written in a similar form, the training of the two formulations is very different. On one hand, neural networks simply adjust model parameters by minimizing a cost function, typically the mean-square error (MSE) between the output of the model and the desired trajectory. On the other hand, the Kalman formulation assumes a probabilistic model of the temporal kinematics. Also included in the Kalman formulation is an estimation of the uncertainty in the state and output estimates. It may seem that the Kalman formulation is a more mathematically principled and structured approach; however the performance of the model is subject to the assumptions imposed by the probabilistic model. It is often difficult to corroborate the assumptions (Gaussian firing statistics, linear kinematics) with the unknown aspects of neural coding and motor systems. Again the Kalman formulation also suffers from issues of parameter estimation which can be overcome by using backpropagation as in the RMLP BPTT formulation.

Compared to the other models (FIR, TDNN) the RMLP is the only topology capable of creating Todorov's internal position, velocity, and acceleration representations that may be necessary for BMI applications. The expressive power of the RMLP is greater because it can utilize the dynamics created by feedback in the hidden layer to create these states. The other topologies are far more restrictive. For example, the FIR limits the neural to motor mapping to a linear feedforward system. Since there is still no proof that the neural to motor mapping is linear, the TDNN takes this a step further by using nonlinearity; however, it is still feedforward. Since BMI model generalization is

related to the complexity of the specific task at hand, we wish to equip the topology with features that give it the best chance of succeeding in the task. Therefore, we believe the use of both nonlinearity and dynamics are the most appropriate.

Lastly, the RMLP produced equivalent or better CC, SER, and CEM values with the fewest number of model parameters. One aspect and explanation of this performance level is that the RMLP generalized better during the testing. As discussed earlier, the ability of a model to generalize is directly related to the complexity or the number of free parameters in the topology. The RMLP effectively reduces the number of parameters by moving the memory structure away from the neuronal input which has large dimensionality. This topology is capable of achieving an equivalent memory depth in samples with fewer parameters. The use of a low complexity smooth function enables the topology to avoid overfitting the data.

Motivation for Quantifying Model Generalization

There are two aspects of model generalization that we deem will be important for the success of real BMIs. The ultimate vision is that the BMI could be operated by a paralyzed patient who through some training procedure could use the neuroprosthetic device for long periods of time and for a wide variety of movements. If BMI models are not reliable in this sense, then the user would have to continuously retrain the system which would place a discontinuity in the freedom to operate the device. Currently BMI models have only been tested in limited datasets involving only one session and one type of movement (either reaching *or* cursor control). Here we will investigate model performance for both multi-session and multi-task datasets.

We have shown that the RMLP generalizes well over short testing datasets (5 min.). In the introduction of this dissertation, it is claimed that the modeling problem is

complicated by the nonstationary nature of the neuronal recordings. At this point, it remains unclear if the nonstationarity is a result of a change in the underlying properties of individual cells or if the ensemble activity itself is evolving as a function of the attention state of the primate. If we refer to the literature, many BMI groups have found experimentally that the model mappings were updated frequently (i.e. every 10 min.) to achieve a high level of performance [16, 18, 19]. The consensus is nonspecific and states without quantification that the models need to be updated because the ensemble of recorded neurons is changing its firing properties and directional tuning frequently. This statement implies that the fixed model parameters are no longer relevant for producing accurate trajectory reconstructions.

Multi-Session Model Generalization

The scheme used to test the multi-session model generalization involves the concatenation of datasets (neuronal and behavioral recordings) from several recording sessions spanning several days. The objective is to train a model with data from session 1 and continue testing into session 2. Two scenarios for testing model generalization are presented in Fig. 4-1. Since only two recording sessions from Duke University included the necessary information to concatenate the datasets, the second scenario involves reversing the order of the sessions. Optimally the concatenation of several consecutive sessions is desirable since it would mimic a real BMI situation; however, the data was not available.

Over the span of several recording sessions it was shown in Tables 2-1 and 2-2 that the number of cells in the ensemble can vary. In order to test a fixed model in another recording session, the experimenter must be careful that the same training neurons are assigned to their respective weights in the testing set. Without aligning cells

to weights that they were specifically trained for, the model is not *guaranteed* to produce testing movements that are in the same class of training movements. To prevent this problem Duke University saves the spike templates used for spike sorting the cells during the data collection process. In other recording sessions, the templates can be compared and similar cells in both sessions can be indexed. In the case of Aurora’s BMI dataset used in these experiments, only 180 cells were common in the two sessions (183 S-1, 185 S-2) and they were aligned in sequence in the data concatenation process.

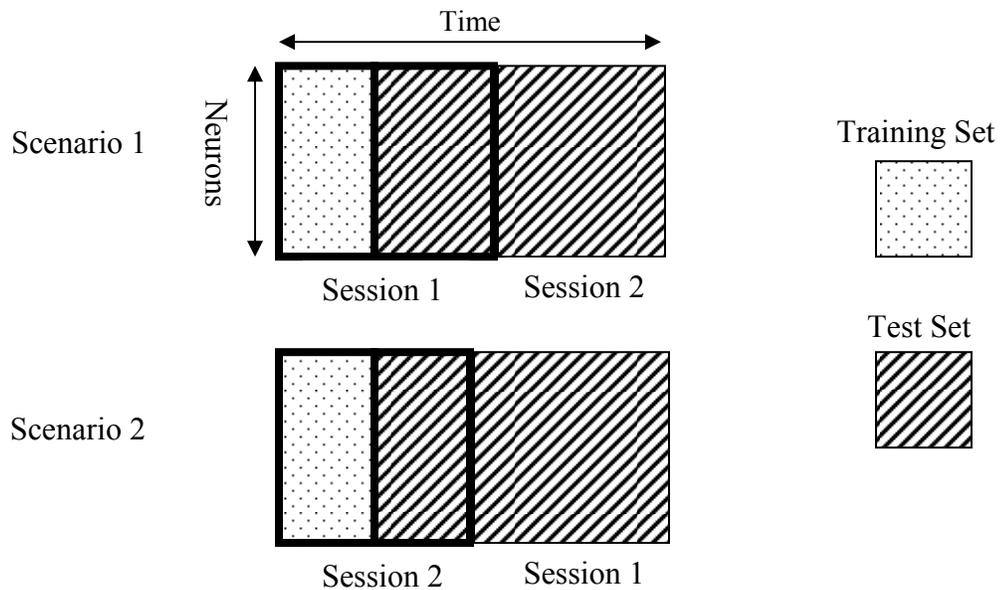


Figure 4-1. Two scenarios for data preparation

To test the RMLP generalization, the model was first trained with either the first 5000 samples from the dataset corresponding to scenario 1 or 2. A RMLP with topology identical to those used in the cursor tracking experiments of Chapter 2 was employed in the training of each scenario. Upon training completion, the model parameters were fixed and novel neuronal data samples were run through the networks. A minimum of 44 min. and a maximum of 85.5 minutes of testing data were used in the experiments. The

duration between sessions ranged between one day for HP and HV while GF had a two day time-lapse.

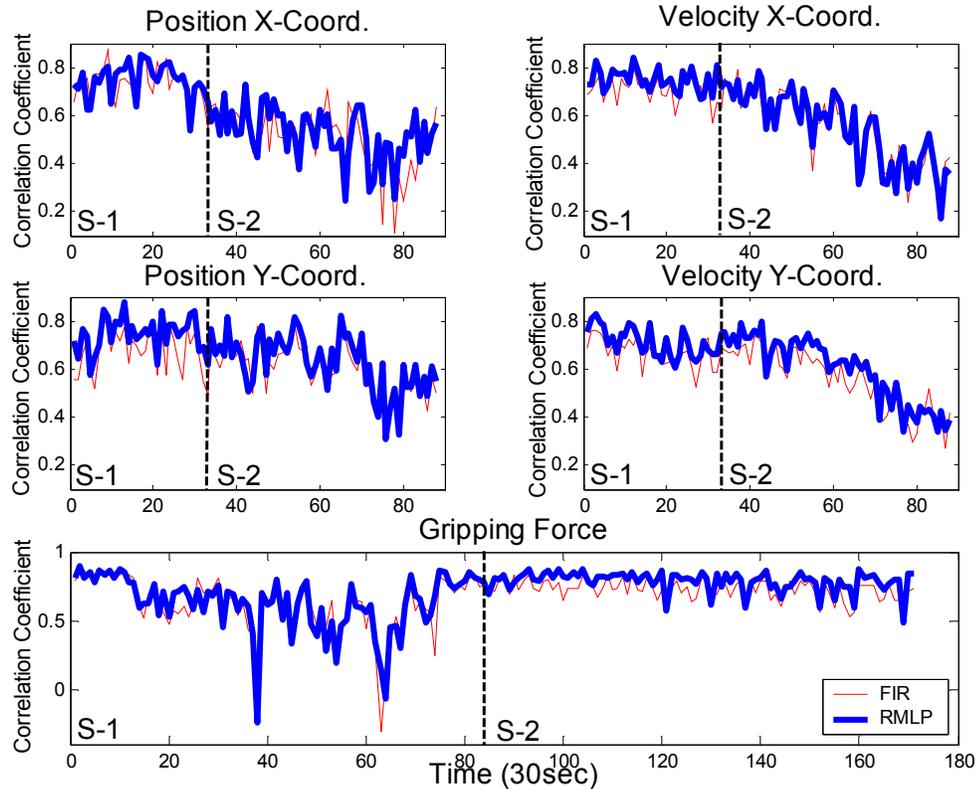


Figure 4-2. Scenario 1: Testing correlation coefficients for HP, HV, and GF

The testing CC values, computed in 30 second non-overlapping windows, are presented for both scenarios in Figs. 4-2 and 4-3. The selected window size represents a compromise between time resolution and the need for enough data samples to compute the correlation. For comparison, the experiment described above was repeated for the simplest BMI model, the FIR filter (10 tap-delay), which will serve as a control. The most obvious trend in the curves is that they are highly variable and this observation is corroborated with the large standard deviation reported in Tables 3-3 through 3-6. We can also see that both the RMLP and FIR filter fail at producing high values for the same

time windows. Lastly, depending upon the scenario, the correlation curves show either a downward trend (Fig 4-2 HP, HV, Fig 4-3 GF) or an upward trend (Fig 4-3 HP, HV, Fig 4-2 GF).

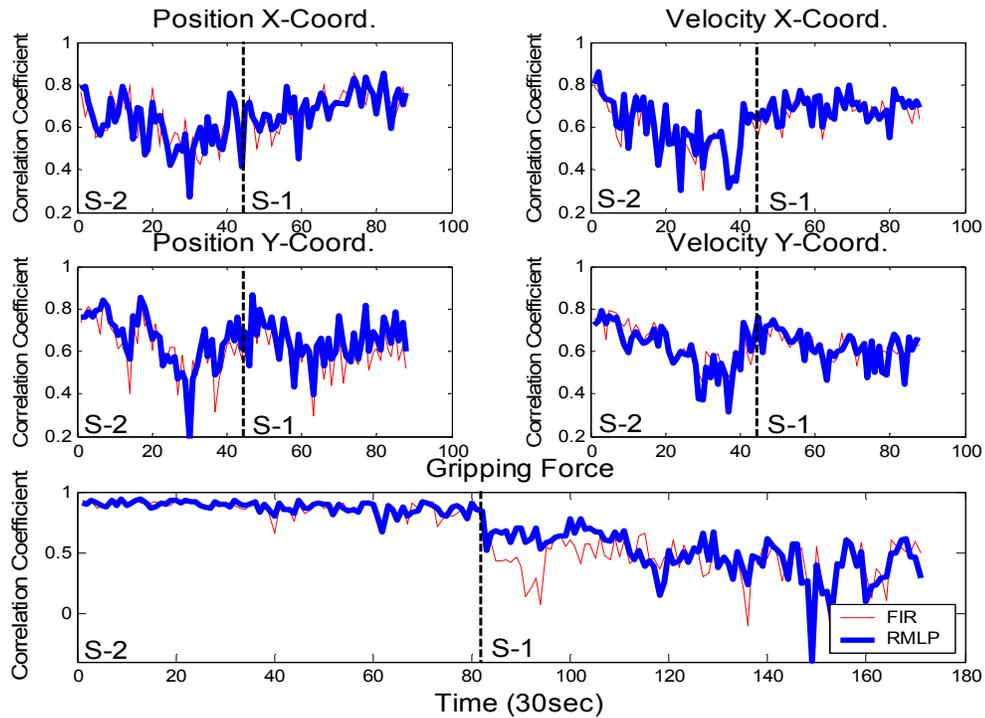


Figure 4-3. Scenario 2: Testing correlation coefficients for HP, HV, and GF

The trends in the correlation curves are quantified for each scenario in Tables 4-1 and 4-2. Here the Kolmogorov-Smirnov (K-S) and two-sample t-test for a p value of 0.05 are used to compare the correlation coefficients from each session. The K-S test is a “goodness of fit” test that measures if two independent samples (for BMIs this is the correlation in S-1 and S-2) are drawn from the same underlying continuous population. Since we observe increases and decreases in the mean the t-test will also be used to determine if the two independent samples come from distributions with equal means. Included in both tables are the mean values of the correlation for each session which can be used to gauge if significant changes in correlation either increased or decreased.

In S-1, the HP and HV datasets failed the K-S test indicating the correlations in each session belong to separate populations. Additionally, the average correlation values (highlighted in the table) from S-2 were all significantly lower than in S-1. This result would seem to indicate that both the RMLP and FIR models generalize well up to a point and then the performance slowly degrades. It should be noted here that between model training and testing, performance will always degrade. However, the amount of degradation is dependent upon the particular segment used in training because of the nonstationary nature of the data. Additionally this result does not guarantee that performance will not increase in other signal segments depending upon the conditions in the data. In the case of the GF, a significant increase in the observed correlations occurred. In Table 4-2 samples 15 through 75, after inspection of the model output and desired GFs, the statistical properties of the desired signal were observed to change significantly. Differences in performance between the sessions could be attributed to differences in the dynamic range and frequency of the gripping forces that could not be captured by the trained model.

In scenario 2 an equivalent or significant increase in HP and HV testing performance was observed as time progressed. Significant values are highlighted in Table 4-3. Most CC values achieved on average a 17% increase in correlation in S-2. This result which is contrary to that which is reported in the BMI literature quantifies that both linear and nonlinear models can generalize for testing periods close to an hour in length even when the generation of neuronal activity is discontinued for a period of one or two days (1 day HP, HV; 2 days GF). Again we also observed a significant change in the GF performance but this time performance degraded on average to a low value of 0.45.

During testing the dynamic range and frequency changed to values not encountered while training and may again serve as an explanation for the poor performance.

Table 4-1. Scenario 1: Significant decreases in correlation between sessions

			HP	HV	GF
FIR	X	K-S	1	1	1
		T-Test	1	1	1
		Mean S-1	0.74	0.71	0.62
		Mean S-2	0.51	0.54	0.72
	Y	K-S	1	1	
		T-Test	1	1	
		Mean S-1	0.68	0.67	
		Mean S-2	0.60	0.55	
RMLP	X	K-S	1	1	1
		T-Test	1	1	1
		Mean S-1	0.75	0.75	0.63
		Mean S-2	0.53	0.55	0.79
	Y	K-S	1	1	
		T-Test	1	1	
		Mean S-1	0.75	0.71	
		Mean S-2	0.63	0.60	

Table 4-2. Scenario 2: Significant increases in correlation between sessions

			HP	HV	GF
FIR	X	K-S	1	1	1
		T-Test	1	1	1
		Mean S-1	0.60	0.57	0.87
		Mean S-2	0.70	0.68	0.43
	Y	K-S	0	0	
		T-Test	0	0	
		Mean S-1	0.63	0.64	
		Mean S-2	0.60	0.62	
RMLP	X	K-S	1	1	1
		T-Test	1	1	1
		Mean S-1	0.60	0.59	0.88
		Mean S-2	0.70	0.69	0.48
	Y	K-S	0	0	
		T-Test	0	0	
		Mean S-1	0.65	0.61	
		Mean S-2	0.66	0.63	

Multi-Task Model Generalization

The second component of BMI model generalization involves testing the RMLP model on a multi-task movement. To stress the generalization capability of the model we require data that fully utilizes the 3-D working space of the primate. In animal experiments, it is often difficult to obtaining these datasets because BMI experimental paradigms involving primates require carefully planned behavioral tasks that are motivational, nonthreatening, and require simple skill sets for the primate. Even with extensive planning, the primates must be trained for several months prior to neuroprosthetic implantation and behavioral experimentation. For these reasons, the datasets that have been provided by Duke University often only involve single-task behaviors collected from several species of primates. Here data from a single primate performing multiple tasks in a single session was not available; therefore, we devised an experimental paradigm that could demonstrate multi-task model generalization using existing datasets.

The experimental paradigm involves concatenating datasets from the behavioral experiments of Belle (hand reaching) and Aurora (cursor control). The idea is to mix segments of 3-D hand reaching with 2-D cursor control movements thus forcing the trained network topology to learn changes in movement frequency and dynamic range. Obviously since the dimensionality (both neuronal and positional) of the two datasets differs, several constraints were artificially imposed.

First, the desired trajectory for all movements included X, Y, and Z coordinates; therefore, segments containing 2-D movements were zeroed in the Z direction. An example of the desired signal used in model training presented in Fig. 4-4 which shows how the cursor control movement (samples 1000-4000) with small amplitude is

interleaved with the large amplitude reaching movement (samples 4000-7000). The entire training dataset consists of 14000 samples with alternating patterns of the two behaviors.

The second constraint imposed involves the neuronal inputs and is highly artificial since the resulting dataset concatenates neurons from different cortices of two different species (owl and Rhesus monkeys). In this experiment we make the strong assumption that assigning different types of cells to each model weight is not a significant problem. Only the number of cells used from each recording session is preserved in the concatenation process. The firing patterns of 104 cells were selected using a “cellular importance measure” discussed later in Chapter 7 of this dissertation. As with the desired dataset, the firing patterns from the 104 cells were alternated in synchrony with their respective movements in this multi-task paradigm.

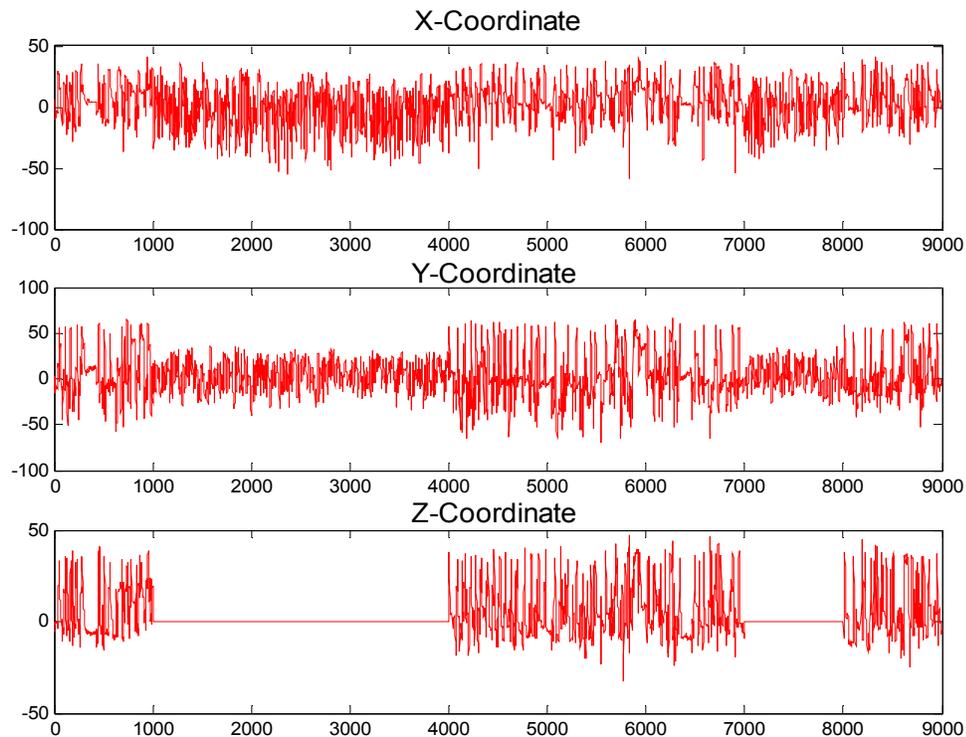


Figure 4-4. Multi-task model training trajectories

The RMLP topology trained using this dataset was modified to contain fifteen hidden PEs which will give the model extra degrees of freedom to account for the variability encountered in the movements. In the separate behavioral experiments the training BPTT trajectory length was chosen to match the number of samples of each movement. For our datasets 40 samples and 50 samples were chosen for the cursor tracking and hand reaching movements, respectively. In the multi-task experiment, the trajectory was chosen to match the largest movement and in this case it is the cursor tracking task with 50 samples.

Given the strong assumptions about the data in this experiment we are surprised by the high performance of the linear and nonlinear models. While the RMLP has been shown to be a universal mapper in \mathcal{R}^N it is not obvious that the linear FIR filter would be able to reconstruct the two trajectories in testing. The FIR CC values were all significantly lower than the RMLP as shown by the K-S test in Table 4-4. One of the most attractive features of the RMLP is its ability to zero the Z coordinate in the 2-D movements. The RMLP produced the smallest mean square error during these segments as shown in the last column of Table 4-3. Differences in performance is also confirmed further by the variance in the CEM curves of Fig. 4-5. Since the variance in the curves is small this result shows that for this dataset a topology with a sufficient number of degrees of freedom and training examples it is possible to learn *simultaneously* the functional mapping from two species of primates and two movements.

Next we would like to push the performance of the two topologies up to a point where they may fail in producing a good mapping in this multi-task. As discussed earlier the expressive power of the RMLP is much greater than the FIR because of the use of

nonlinear recurrent PEs in the hidden layer. We believe that the power of these elements will become much more important when the size of the topology becomes greatly reduced. This may occur when recordings of the neuronal ensemble contain only a few *important* (see Chapter 7) cells. To test this hypothesis, both topologies were retrained using only a subset of 10 important cells. Again in Table 4-3 we present the testing CC and SER values which show an overall decrease in both model's performance. However, we observed a much greater average drop in performance by the FIR filter. In this case, the CC values for the FIR decreased by 12% while the RMLP had only a 7% drop. In Fig. 4-6 we can observe qualitative differences in the performance by plotting the testing X, Y, and Z model outputs (bold) along with the desired signal centered upon a transition in the type of movement (reaching vs. cursor). In the figure, the X-coordinate of the FIR tends to average the outputs for both reaching and cursor control while the RMLP is capable of distinguishing between the two. We can additionally observe that the FIR continues to have problems producing flat outputs in the Z-coordinate.

Table 4-3. Multi-task testing CC and SER values

	Correlation Coefficient/SER (dB)			MSE Z dir
	X	Y	Z	
FIR	0.53±0.20/1.40±1.19	0.59±0.18/1.57±1.40	0.61±0.20/0.68±1.07	13.76
RMLP	0.60±0.18/1.60±1.40	0.64±0.19/1.88±1.56	0.69±0.23/1.06±1.96	1.39
FIR subset	0.26±0.26/1.05±0.71	0.51±0.22/1.39±1.03	0.59±0.24/0.61±0.87	8.96
RMLP subset	0.52±0.21/1.38±1.01	0.53±0.25/1.61±1.21	0.67±0.27/1.06±1.94	1.09

Table 4-4. Significance of CC compared to FIR filter

	K-S Test		
	X	Y	Z
FIR	0	0	0
RMLP	1	1	1
FIR subset	1	1	1
RMLP subset	1	1	1

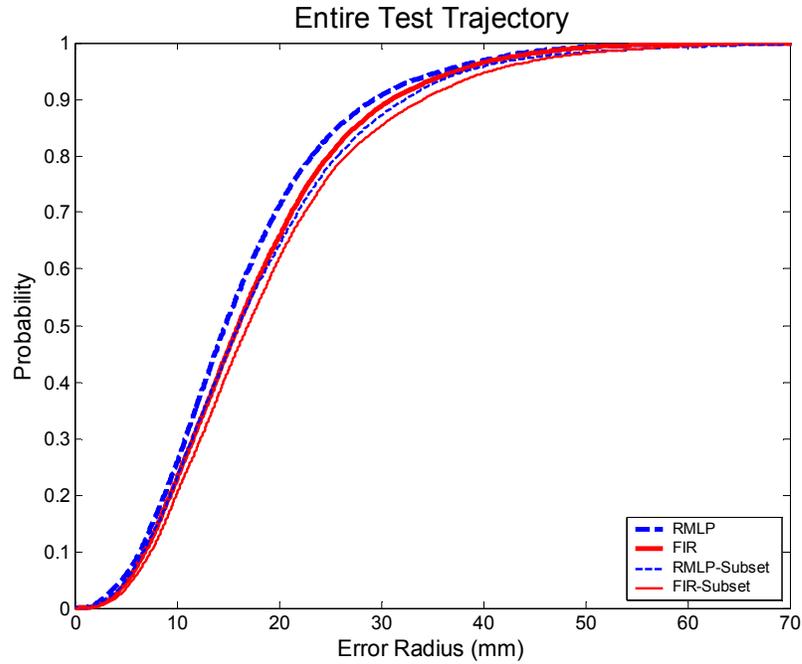


Figure 4-5. CEM curves for linear and nonlinear models trained on a multi-task

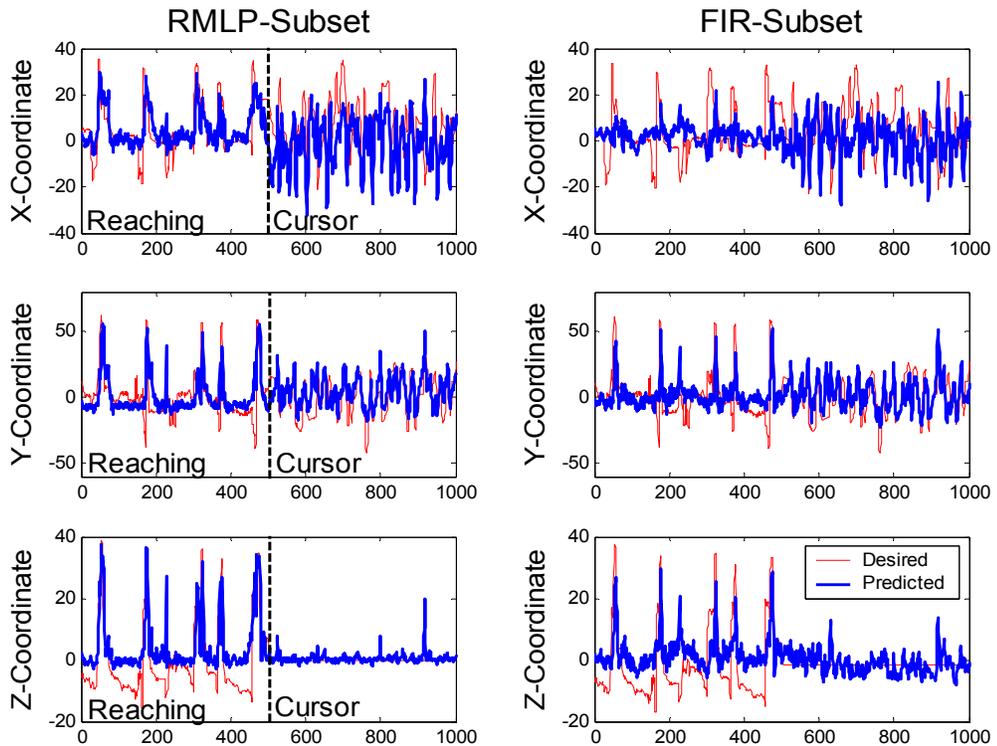


Figure 4-6. Multi-task testing trajectories centered upon a transition between the tasks

Discussion

During training, any model topology with a sufficient number of parameters can produce an output with zero error especially when the desired signal is simple. Models of this type that produce zero training error have *overfit* or *memorized* the training data and will not perform well when faced with novel testing datasets. In BMI modeling experiments, the chance of overfitting the data is especially high because the large dimensionality of the neuronal input leads to topologies with hundred or even thousands of parameters that are attempting to estimate trajectories that resemble simple sine waves. These undesirable qualities bring a sense of brittleness to BMI models and suggest that they will not perform well in real BMI applications where fixed models will be required to produce accurate trajectories for a wide variety of movements and long periods of time.

The generalization of our most promising network, the RMLP, was tested for both the longest duration dataset in our archive and for combined movement/species tasks. The results of our experiments are predicative of precepts that are well known in adaptive filter and neural network theory that involve properties of both the training data and model topology.

In terms of training data, all model topologies assume that the input and desired signals have stationary statistics. The correlation plots in Fig. 1-4 clearly show that this is not the case and local neuronal activity sometimes fires in synchrony with peaks in movement and sometimes it doesn't. Consequently the models selected in these experiments learn to average cellular responses (a result of inner product filtering operations in all topologies). Time instances where this averaging is not appropriate is reflected in the dips in correlation of Figs. 4-2 and 4-3. We also observed that dips in the

correlation occurred when the models were trained on data that was not representative of the data encountered during testing.

Qualitative observations about model generalization regarding the number of training observations, number of model parameters, distribution function of the desired signal, and final training error have been well described by Vapnik's principle of Empirical Risk Minimization and the V-C dimension (Vapnik-Chervonekis dimension). Vapnik's theory provides a mathematical foundation for solving the problem of choosing model/data design parameters for best generalization. In practice though the process of designing the model for a specific problem/dataset involves a compromise between available data, model expressive power, and computational complexity. In our experiments on the multi-task data set, we found that large linear and nonlinear models (that include the full ensemble of cells) performed well in testing. From a signal processing point of view, this result is obvious because the process of optimally orienting a hyperplane becomes much simpler because the size of the neuronal input naturally maps it to a high dimensional space. Moreover we trained both linear and nonlinear models with observations of reaching and cursor tracking covering the entire testing distribution function. In this case, the problem of regression may be solved by a simple linear system. Mapping to high-dimensional spaces and then performing linear regression is one of the fundamental principles of Support Vector regression.

However, when the input was pruned to include only ten cells, the size of the space of both models reduced significantly but the performance of the RMLP remained high while the FIR reduced greatly. This result is showing that compared to the FIR the RMLP is capable of a better compromise of representing different patterns as can be expected.

The use of multiple PEs in the hidden layer gives the RMLP the ability to nonlinearly segment the space of the movement and assign neuronal activity to each of the segments with the input layer weights. The FIR is limited to a simple linear transformation directly from the neuronal activity to the movement coordinate system. In the next chapter we will look in depth at how the RMLP finds the functional mapping by tracing the signals through the topology.

In summary, we have shown for two BMI models (linear feedforward, and nonlinear feedback) trained with MSE that testing performance may not be as brittle as expected. Depending upon the training dataset, it is possible (in open loop experiments) that model performance can significantly increase even when the animal was disconnected from the model for up to two days. Moreover, in high dimensional spaces both linear and nonlinear models have the expressive power to simultaneously learn the functional mapping between cortical activity and behavior for two species of animals and two very distinct hand movements. In environments where dimensionality is a premium, the nonlinear feedback model (RMLP) proved to be more powerful in producing high performance. To achieve high performance in real BMI applications the results of these experiments advocate the use of training sets that incorporate diverse training movements coupled with a parsimonious model with broad expressive power.

CHAPTER 5
ANALYSIS OF THE NEURAL TO MOTOR REPRESENTATION SPACE
CONSTRUCTED BY THE RMLP

Introduction

In order to advance the state-of-the-art in BMI design and gain confidence in the models obtained, it is critical to understand how they exploit information in the neural recordings. Moreover, comprehending the models' solution may help discover features in the neural recordings that were previously unrecognizable. With an explanation of how the network addressed these simple tasks, we can guide the development of our next generation BMI models, which will face more complicated motor control problems.

It is relatively simple to analyze linear BMI models because they are well understood and the mathematical theory of least squares is well-developed [27, 46, 47]; however, we are interested in analyzing our best performing neural network, the RMLP in a non-autonomous task. It is a common belief in the neural network and the machine-learning community that neural networks are "black boxes" and interpreting the function of nonlinear dynamical networks is extremely difficult [62, 63], the exception being the autonomous Hopfield network with its fixed point dynamics. Here we will apply both signal processing constructs and dynamics to make sense of the representations constructed by the topology, weights, and activations of the model.

We hypothesize that a trained RMLP can be used to discover how changes in the neural input can be used to create changes in behavior. By tracing the signals through the topology, visualizing projections of the input and intermediate states, and mathematically

understanding the dynamics of the system we will show the mechanisms by which real neuronal activity is transformed into hand trajectories by the RMLP. From a neurophysiologic point of view, we are using a signal processing modeling framework as a tool to tell us about the mapping from neural activity to behavior. We will demonstrate that for BMI experiments the choice of a parsimonious model, the use of the desired response in a model-based framework, and the appropriate use of signal processing theory provide sufficient evidence to present constructs that agree with basic principles of neural control.

Understanding the RMLP Mapping – Network Organization

To elucidate how RMLP maps the firing neuronal patterns to hand position we examine the representation space constructed by the trained network. We interpret the output of the hidden layer PEs as the bases, and the output layer weights as the coordinates of the projection. For *tanh* nonlinearities, the bases are ridge functions [48], and the input weight vector defines the direction in the large input space where the ridge is oriented. To motivate the analysis, we consider two networks with hidden layers that included five (best performing network), and one (simplest RMLP architecture) processing element (PE). Analysis of the RMLP organization will first be performed for the reaching task since the simple, repeated movement contains landmarks (see Fig. 3-5) which we can refer to. After this analysis is performed, we will compare the organization to the random movements in the cursor control task.

Understanding the Mapping

We examine the performance of the first layer of the simplest RMLP network by plotting the neuronal projections before entering (pre-activity) and after leaving (activity)

the hidden layer in Fig. 5-1. We see that the PE is bursting during the reaching instants⁹. This observation offers a great simplification to the RMLP analysis because in the single PE network, W_f reduces to a scalar. Linearizing the input layer equation in (3-18) around zero (point of maximum slope) we can unfold the recursive equations of the RMLP over time as:

$$y_1(t) = f'(0) \mathbf{W}_1 \left(\mathbf{s}(t) + f'(0) \left(W_f \mathbf{s}(t-1) + W_f^2 \mathbf{s}(t-2) + \dots + W_f^n \mathbf{s}(t-n) \right) \right) \quad (5-1)$$

We see that $y_1(t)$ is constructed by exponentially weighting the past neuronal activity by W_f . We have observed that W_f settles to a value that makes this operating point locally unstable ($f(0)$ is an *unstable attractor*). For example, when the slope of the *tanh* function at this point is 0.5, W_f settles to 2.1, yielding a pole of the linearized system at $0.5 \times 2.1 = 1.05$. This locally unstable behavior of the RMLP is the reason of bursting, and segments very accurately the movements from the arm at rest.

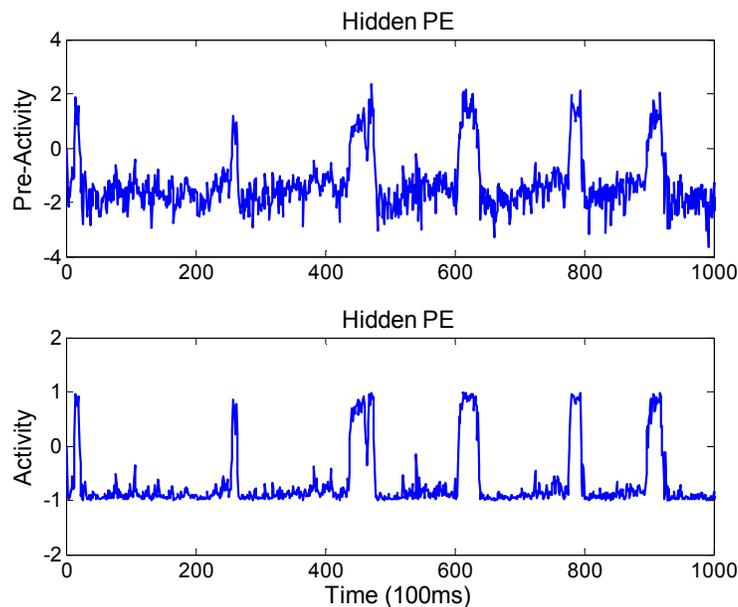


Figure 5-1. Pre-activity and Activity in a RMLP with one hidden PE

⁹ In multiple PE cases only one PE exhibits bursting, which is another reason for studying this network.

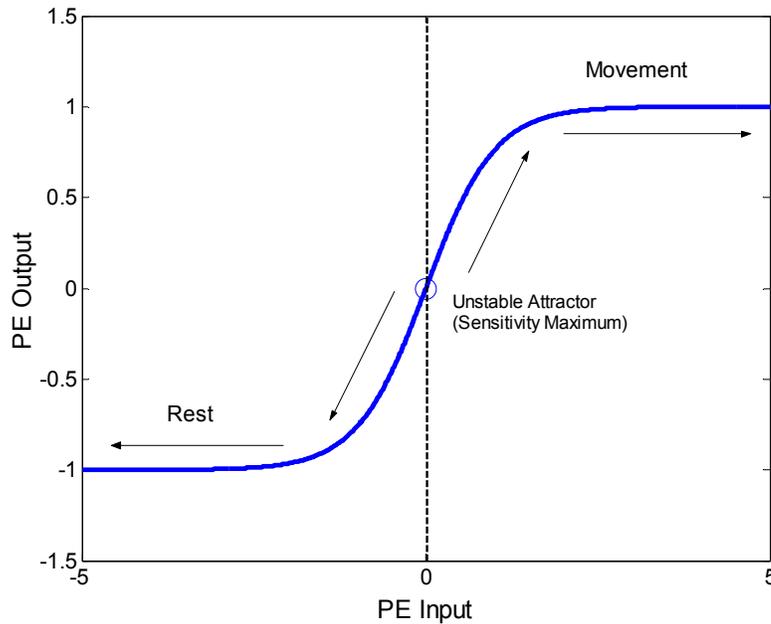


Figure 5-2. Operating points on hidden layer nonlinearity

Observing Fig. 5-2, the two saturation regions of the PE are obviously stable ($f'(x) \cong 0$) and the system state will tend to converge to one of them. The operating point of the hidden PE is controlled by $\mathbf{W}_1\mathbf{s}(t) + \mathbf{W}_f\mathbf{y}_1(t-1)$. When this value is slightly negative, the operating point moves to the lower negative region of the nonlinearity (labeled Rest in Fig. 5-2). Otherwise, the operating point moves to the upper positive region (Movement).

In order to understand how the projected time-varying neural activity vector and feedback interact to produce the bursting behavior during movement observed in Fig. 5-1, we decompose the first layer of the network into its two components $\mathbf{W}_1\mathbf{s}(t)$ and $\mathbf{W}_f\mathbf{y}_1(t-1)$, and plot each of these components in Fig. 5-3 during one movement preceded by a resting period. Notice that when the arm is at rest, both $\mathbf{W}_1\mathbf{s}(t)$ and $\mathbf{W}_f\mathbf{y}_1(t-1)$ are negative, so according to our analysis, the PE is saturated in the negative region. In the figure, we see that every time $\mathbf{W}_1\mathbf{s}(t)$ approaches zero (e.g. at $t=40$) we obtain a rapid

increase in $W_f y_1(t+1)$ (e.g. $t=41$), because the operating point approaches the unstable region of the hidden PE, and the PE goes briefly out of negative saturation. We further observe that in order for the feedback to remain positive, $W_1 s(t)$ must be sustained at a positive value for some samples (e.g. $t=110-130$), and then the feedback kicks-in, amplifies this minute change, and the operating point of the hidden PE goes to the positive saturation, smoothing out the changes in the input. This condition corresponds to the movement part of the trajectory. Therefore, we can think of the feedback loop as a switch that is triggered when the projection of the input $W_1 s(t)$ approaches zero and becomes positive.

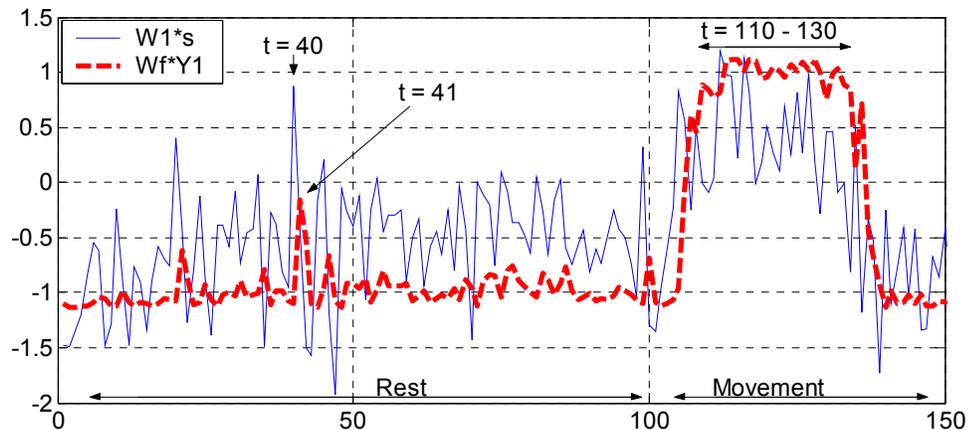


Figure 5-3. Input layer decomposition into $W_1 s(t)$ (solid) and $W_f y_1(t-1)$ (dashed)

Input Layer Weights

The input neural activity vector controls the RMLP operation by maintaining the PE input around $W_1 s(t) \sim 0$, which is achieved by placing through training the W_1 vector perpendicular to the input. This seems an odd choice, since this is the unstable attractor of the dynamics. What does this solution tell in terms of neural activity? To answer this question, we have to do some further signal processing analysis. Remember that the neural activity is a long vector with 104 entries (neural channels). As we can see from

Fig. 5-3, the inner product with \mathbf{W}_1 is very noisy, reflecting the highly irregular nature of the neural firings.

We first computed the norm of the weight vector over time as in Fig. 5-4. As we can see the norm of $\mathbf{s}(t)$ is mostly constant, meaning that on average the number of spikes per unit time (100 msec) is basically constant. Therefore, we conclude that what differentiates the neural firings during a movement must be a slight rotation of the vector that makes $\mathbf{W}_1\mathbf{s}(t)$ barely positive. We plot in Fig. 5-5 subplot 1 the angle at successive time ticks (100 msec) between inputs and the \mathbf{W}_1 vector to corroborate our reasoning. Notice that immediately preceding and during movement the angle between the input and \mathbf{W}_1 vectors becomes slightly less than 90 degrees as expected. The sign change is a result of a slight rotation (92° to 88°) of the neural activity vector initiating and during movement. The components of this vector are the neuronal firing counts, therefore, either all components change or just a few change. In order to find out which case applies here, we computed the directional cosine change for all the 104 neurons during movement¹⁰. Figure 5-6 shows a plot of the neurons and their average weighted firing during a rest and movement segment of data. We can observe that neurons 4, 5, 7, 22, 26, 29, 38, 45, 93, and 94 can be considered the ones that affect the rotation of the neural vector the most. Figure 5-5 subplot 2 shows the directional cosine for successive inputs of the subspace spanned by the most important 10 neurons, and the directional cosine of the rest of the neural population consisting of 94 neurons. Basically there is no change of the rotation of the neural activity in the 94 dimensional space, while in the space of the 10 highest neurons the directional cosines change appreciably.

¹⁰ Ten neurons with the highest weighted average firing rate over a rest/movement window of 150 samples are selected. The weights are the first layer weights of RMLP corresponding to each neuron.

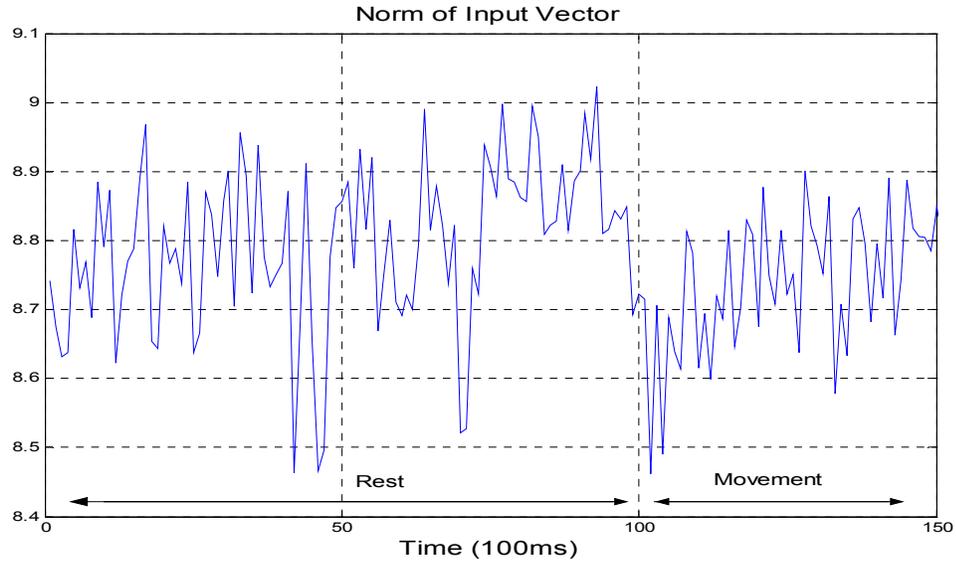


Figure 5-4. Norm of the input vector (104 neurons)

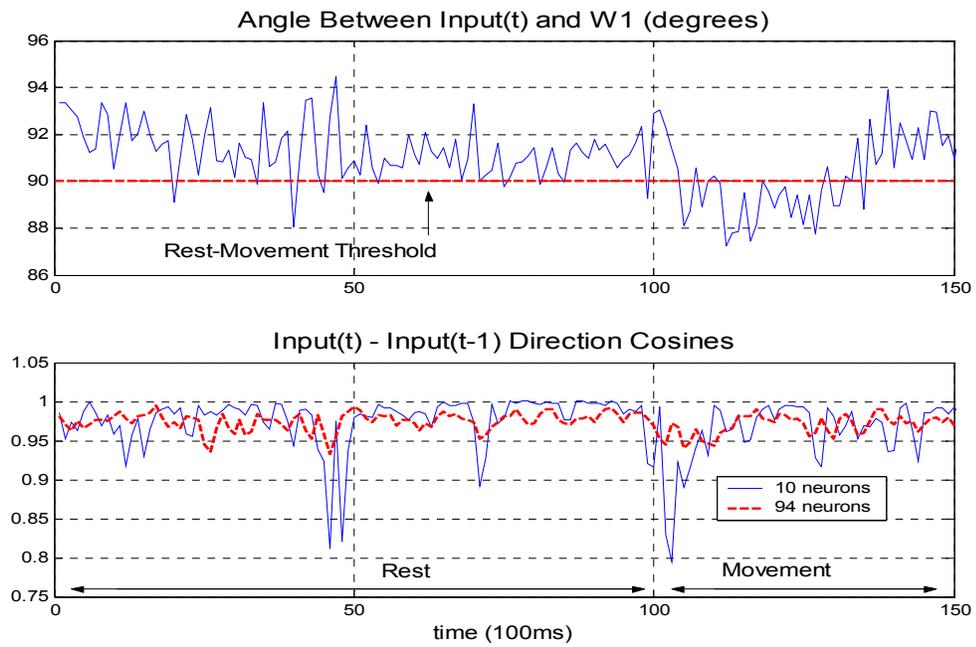


Figure 5-5. Angle between $s(t)$ and W_1 . Direction cosines for successive input vectors $s(t)$ and $s(t-1)$

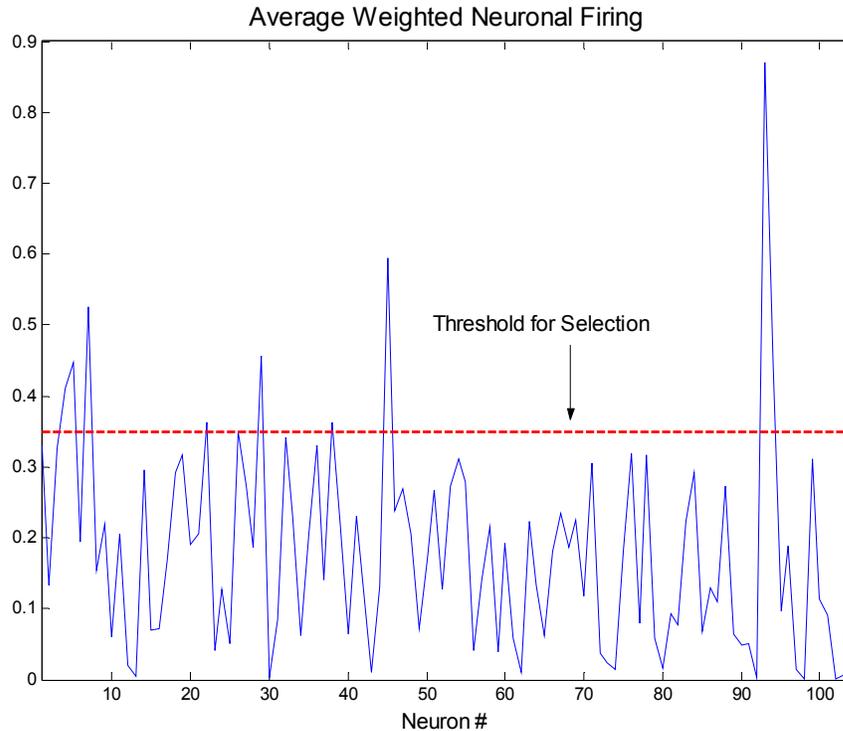


Figure 5-6. Selection of neurons contributing the most to input vector rotation

Output Layer Weights

In this single hidden PE case, the second (output) layer consists of a 3x1 weight vector to predict 3-D hand position. The network output is always a scaled version of this vector, where the time-varying scale factor is given by the hidden PE output. Hence, this simplified single hidden PE network can only generate movements along a 1-D space (line).

In light of this 1-D result, we proceed with our best performing network, which contains 5 hidden PEs¹¹. The weight connections for one of the hidden PE directions are shown in bold in Fig. 5-7. Each hidden PE direction contributes to the X, Y, Z output through the weights W_X , W_Y , and W_Z .

¹¹ For multiple PEs, one eigenvalue of \mathbf{W}_f leads to an unstable pole, so the results extend.

Let us relate the output layer weight vectors to the reaching movement. A plot of the hand trajectory in 3-D space along with the weight vectors associated with each hidden PE as shown in Fig. 5-8. As a first qualitative observation, PE #1 seems to point to the mouth of the primate while PEs #4, 2 and 5 point to the food. We also observe PE #3 to be pointing from the food to the mouth. To quantify these observations, we plot in Fig. 9 the Principal Components (PC) of the movement (the view is in the direction of the third PC) and compute the angle (in degrees) between the output weight vectors and the principal components. The first two PCs point in the directions of maximum variance (rest-food and rest-mouth) of the movement while the third PC vector captures little variance. The corresponding eigenvalues are 568, 207, 30 indicating that the first two PCs capture 70% and 26% of the movement variance, while the third PC only captures 4%. Since the trajectory of reaching for food and putting it in the mouth essentially lies in a plane, the first two PCs define a minimal spanning set for this movement. PC1 points toward the mouth while PC2 points toward the food. The third PC is orthogonal to this plane. Investigating the angles between the PE weights and the PCs show that while the directions of PE#1, PE#3 approach PC1 (with angle differences of 28° and 34° respectively), the directions of PE#2 and PE#5 approach PC2 (23° and 20°). Finally, PE#4 aligns itself equally distant from PC1 and PC2 (46° and 44°). This indicates that the network output weight vectors organize so that each specializes for either reach to food or reach to mouth. All PEs form large angles with PC3 indicating that the network exploits the fact that the movements lie in a plane.

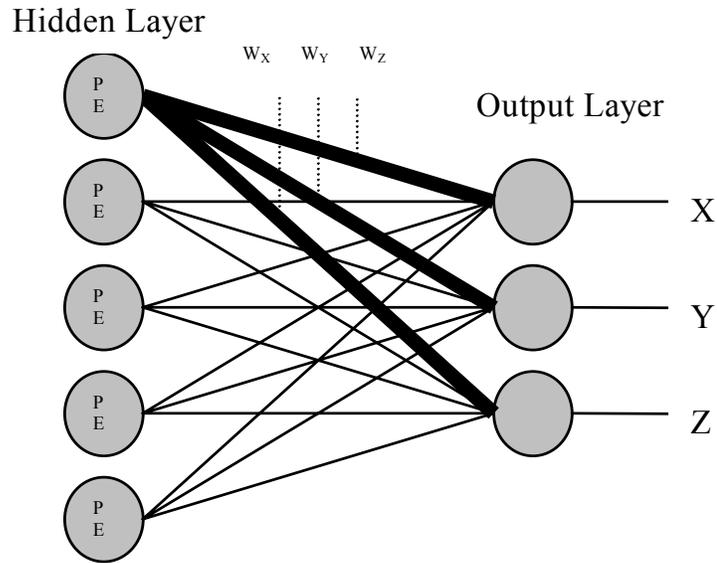


Figure 5-7. Output layer weight vector direction for one PE

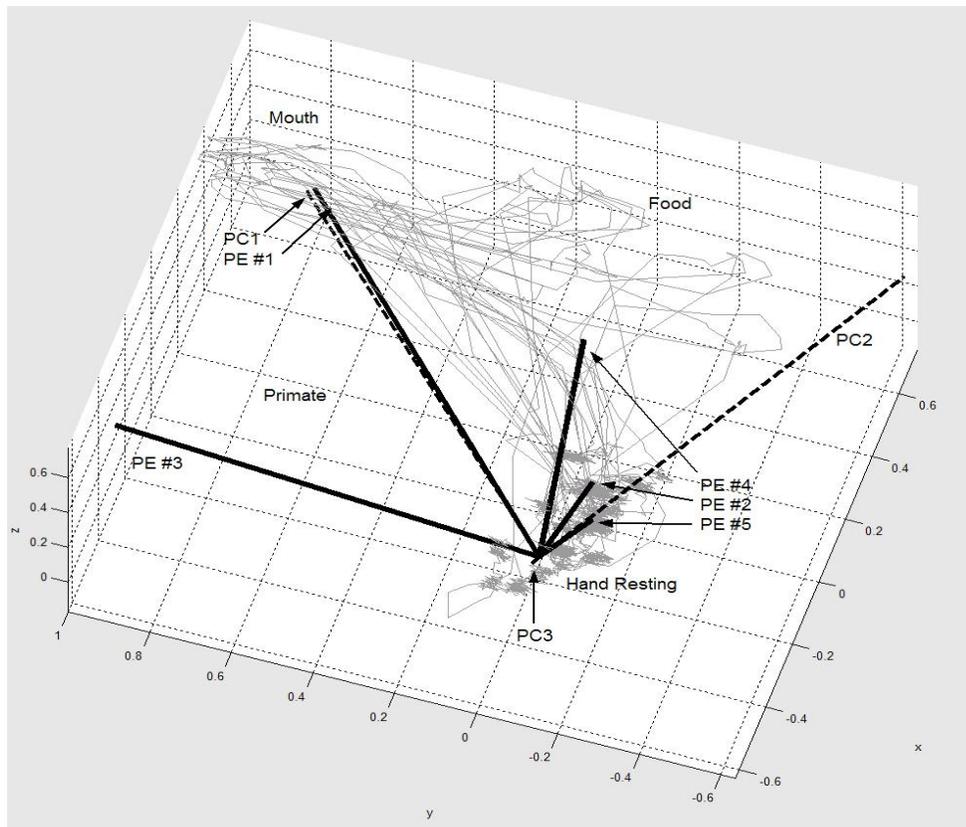


Figure 5-8. Movement trajectory with superimposed output weight vectors (solid) and principal components (dashed). This view is in the direction of PC3

Cursor Control Mapping

One of the most interesting aspects of the RMLP organization is that through the training procedure the input layer weights were adapted so that they are orthogonal to the neuronal input vector. In the reaching task, we suspected that this orientation of the weights combined with unstable dynamics were a special solution for the situation of a reaching task requiring a quick burst in the change of position. Until the trained RMLP for the cursor control task was analyzed, it remained unknown how the network would solve the mapping problem for smooth trajectories.

The same analysis techniques used for the reaching task are presented in Fig. 5-9 for the cursor control task. Here, 150 samples of the smooth trajectory are analyzed and for simplicity the decomposition is performed on a RMLP trained with one hidden PE¹². In the first subplot, we see that again the angle between neuronal input vector and the input layer weights are roughly orthogonal. At certain time instances, the angle decreases to less than 90° and immediately following this crossing a large increase in the weighted feedback results (see Fig. 5-9 subplot 2 dashed line)¹³. Again we see that the feedback is triggered by a rotation of the neuronal input vector. The projection of the neuronal input vector, shown in subplot 2 (solid line), remains negative for all negative hand positions (Fig. 5-9 subplots 4 and 5). Only when the projection of the input approaches zero does the feedback kick in to amplify the signal and saturate the nonlinearity shown in subplot 3. Recall that for the reaching task that the negative tail of the nonlinearity was assigned

¹² This network achieved testing correlation coefficients of 0.40 and 0.71 for X and Y coordinates, respectively.

¹³ This trend repeated itself for the multi-PE network. The weights for each PE were oriented such that the angle formed with the input was either slightly greater or less than 90°. These differences are a matter of a sign change in the weights.

to resting movements and the positive tail was assigned to reaches; however, in this cursor control task the negative tail of the nonlinearity is assigned to all negative positions while the positive tail is assigned to positive positions.

Analysis of the output layer vectors of the trained RMLP network revealed that again they basically span the 2-D space of the cursor movements. Since there are no observable landmarks in this random movement analysis of the PE alignment is not as clear.

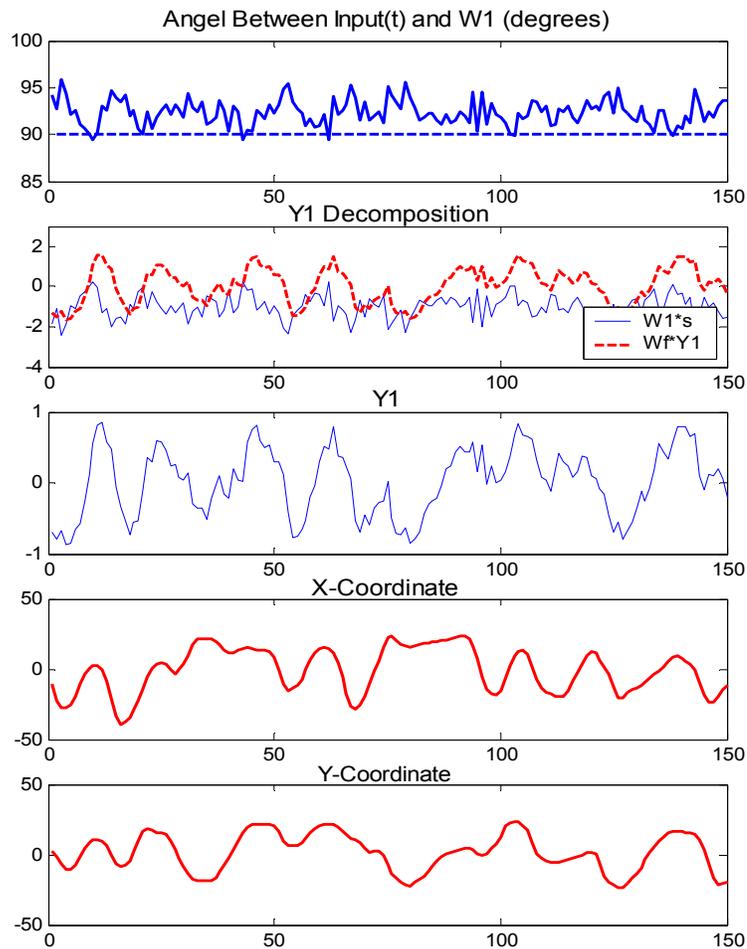


Figure 5-9. RMLP network decomposition for the cursor control task

Discussion

In BMI research, the primary goal is to efficiently decode neural data to construct mappings from spike trains to hand movements. We have elected a state recurrent neural network to implement this mapping, because RMLPs are parsimonious nonlinear models that are very effective and efficient for finding timing relationships between input and desired time series.

After analyzing how the RMLP finds the mapping from neural activity vectors to hand positions, we can fully explain and comprehend why this neural network performs better than the linear models for reaching tasks. The RMLP has the advantage of sensing slight changes in neural activity and switching to a temporarily unstable state where large reaching amplitudes are required. Unlike the linear models, the RMLP has the ability to modify its outputs in a more powerful way by using the saturating nonlinearity and its internal dynamics. The projection of the neural input is used to indicate which direction in the output space to move and for how long. Once these decisions are made, the feedback takes over to amplify and smooth the input projection.

The solution of placing the input layer weights orthogonal to the input vector has ties to the theory of anti-hebbian learning [46, 48]. In this style of network learning, weight values are reduced so that the projection of the input reduces to a point (i.e. the input is decorrelated or orthogonal to the weights). This result occurs when the data important for the mapping exists in a subspace of the input dimensionality [48]. In terms of the neuronal input used in these BMI experiments, we observed that the vector of cells is rotating only a few degrees in a high dimensional space. This result can come about if only a few cells in the ensemble are modulating their activity (i.e. the data important for

the mapping exists in a subspace). Later we will present methodologies to identify important cells that create this subspace.

Very seldom, are we able to apply artificial neural networks in the setting where they were initially proposed, e.g. as abstractions of brain function. Here we have an opportunity since the RMLP inputs are neuronal firing counts. The hidden layer PEs and weights represent the nonlinear conversion between the microscopic spike counts and the population wave density as proposed by Freeman [64] and Grossberg [65]. Therefore, we can interpret the hidden layer activity as *mesoscopic motor cortex variables*. The organization of this first layer creates highly sensitive mesoscopic detectors from distributed microscopic population vectors. These neural population vectors are noisy, due to the variability of firing, and individual contributions are swamped by many other neurons. However, if the weight vector is organized to be perpendicular to the mean population vector, small changes in just a few components of the ensemble activity are sufficient to produce changes in the mesoscopic variables. The RMLP architecture is specially tuned to exploit these minute changes because it has state feedback, and since the linearized system is unstable, it is able to amplify them. The RMLP hidden PEs are additionally analogous to the individual neurons in Georgopoulos' Population Vector Algorithm [7], but the big difference now is that the modeling is done at the mesoscopic level, where projected distributed neural activity, instead of single neurons, are modulating the activities of the PEs in the hidden layer.

From the earlier discussion of Todorov's mechanistic model and the RMLP architecture, we can expect that the output layer weights code for the directions in space spanning the movement directions. This hypothesis is experimentally verified where we

show that the output weights are aligned primarily with the PCA components of the trajectory in 3D space. Alternatively, we could use the Population Vector Algorithm (PVA) proposed by Georgopoulos and Schwartz who also use an inner product rule [7]. The PVA states that *each* neuron in the motor cortex has a “preferred direction” for which it fires maximally. By determining the preferred direction for each neuron, a movement can be reconstructed by taking a weighted sum of these preferred directions. The RMLP hidden PEs in this framework are analogous to the individual neurons in the PVA, but the big difference now is that the modeling is done at the mesoscopic level, where projected distributed neural activity, instead of single neurons, are modulating the activities of the PEs in the hidden layer. From our experience with the data, Todorov’s model seems to capture more the reality of the situation. In fact only a few neurons seem to be tuned to the direction of movement. Therefore the tuning curves in the RMLP are an epiphenomenon of the inner product rule of vectors when one goes from the microlevel (neuron) to the mesoscopic level (distributed representations). Of course it is unknown if the neurons that do not show firings correlated with the reaching task are tuned to other directions.

We and other groups have shown that the neural to motor activity mapping task, although not easy, can be accomplished with reasonable accuracy using an input-output system identification approach by training neural networks. In this chapter, we hope to overcome the perception that neural networks are incomprehensible black boxes that defeat any explanation of how they reach the input-output map. We argue that they are engineering solutions that give us a rich understanding about motor cortex neurophysiology and organization. We have shown that the RMLP inner workings can be

understood from first principles, using signal processing concepts of projection operators and network linearization. RMLPs are therefore a viable alternative not only to achieve accurate decoding with little training data, but also to help improve our understanding of the spatio-temporal relationships of neural population vectors.

CHAPTER 6 INTERPRETING CORTICAL CONTRIBUTIONS THROUGH TRAINED BMI MODELS

Introduction

A natural next step in BMI development is to hypothesize that these models can be studied in a principled way to extract neurophysiologic trends in the neural recordings. By analyzing the model parameters in a signal-processing context, we can extract relationships between cortical regions and the desired behavior. Note that this approach to neuronal analysis contrasts with traditional neuroscience methods that implement data-driven reasoning under extremely well controlled experimental paradigms. However, we feel that this methodology is not prepared to attack the study of interactions between large populations of neurons used in BMI experiments. Consequently, we propose the use of signal processing constructs to deduce interpretations of neural activity.

The success of neural analysis through models depends on having well trained models that encode the mapping from neural activity to hand kinematics. It has been shown in previous chapters that testing correlation coefficients have an average value of 0.8. We must also be aware that the model could bias the interpretation due to abstract level of modeling, and difficulties in determining a reasonable fit, model order, and topology. For these reasons, the validity of this approach must be tested before continuing this type of analysis. By investigating how the choice of the model topology affects the interpretation of neural activity, we can test and validate the aptness of this approach.

With these considerations, we first compared interpretations of neural activity through two different models: linear feedforward FIR filter and nonlinear dynamic RMLP. Again the FIR will serve as a control since it is extensively used in the BMI literature [16, 19] and the RMLP produced high levels of performance with a parsimonious topology [8, 9, 12, 14]. In this chapter, both models will be trained to predict the hand trajectory of the behaving primate using various combinations of cortical activity from the primary motor, premotor, supplementary motor, somatosensory, and posterior parietal cortices. This investigation will show how each model identifies which cortical regions are involved with the production of the hand trajectory.

Cortices Involved in Hand Reaching

Belle's Cortical Contributions

We are interested in the cortical regions that contribute to the triad of movements defined in Fig. 6-1. Each reaching movement is segmented to three reaches: rest/food, food/mouth, and mouth/rest. By training both the FIR and RMLP using combinations of neurons from different cortical areas, and observing the network outputs, we can build a set of constructs to compare with established neurophysiologic principles. Both models are trained using multichannel neuronal firing times from up to 104 cells and hand trajectories. Recall in Table 2-1 that for the owl monkeys that the neural recordings are collected from four cortical regions (posterior parietal (PP) – Area 1, primary motor (M1) – Area 2, premotor dorsal (PMd) – Area 3, and (M1/PMd-ipsi) – Area 4).

Using approximately 33mins of Aurora's neural activity¹⁴ from all combinations of the base set of cortical regions, we trained 15 FIRs and RMLPs. For example, the first

¹⁴ The results presented here are only for the first recording session; however, the results are consistent among all trials.

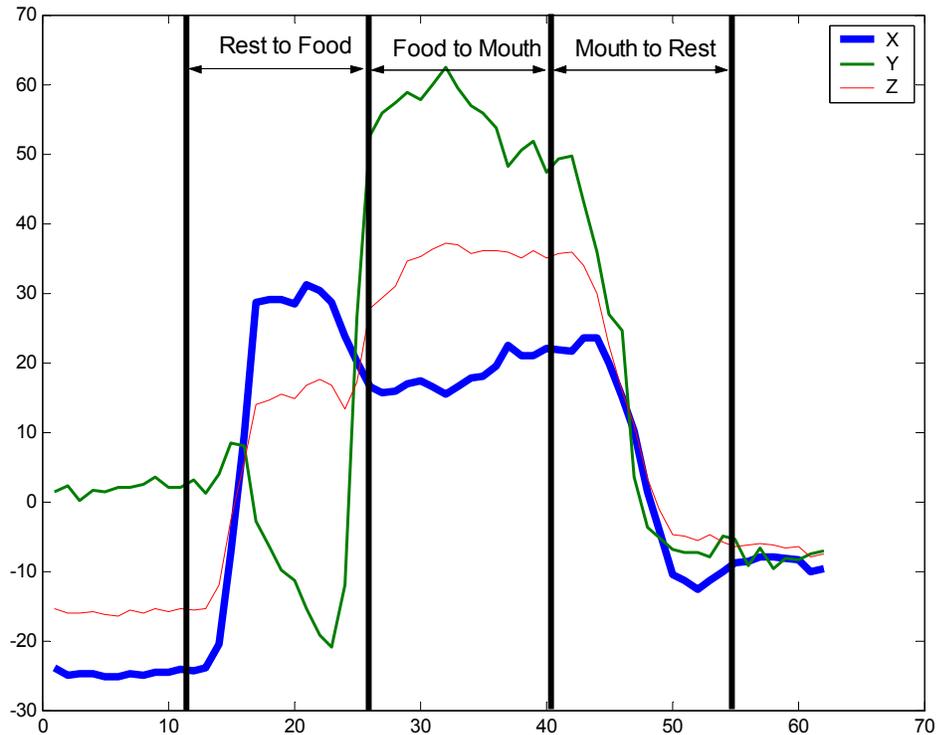


Figure 6-1. One movement segmented into rest/food, food/mouth, and mouth/rest motions

network was trained only with activity from PP while the second network was trained with only M1 and the fifth network was trained with a combination of PP and M1. After training, the weights of each topology were fixed and 5mins of novel data were presented to produce hand position predictions. The X, Y, Z network outputs (bold) and the actual hand coordinates for one sample movement are plotted in Figs. 6-2 and 6-3 for each network. For clarity, a brief summary of the assignment of areas to cortices is provided in Table 6-1. Even though the RMLP outperforms the FIR in capturing the trajectory peaks (subplots Area 1234), we observe that the trends of both topologies are consistent. For example in both topologies, Area 1 captured rest/food, but showed a poor fit in food/mouth. Area 2 does not display any correlation to this desired trajectory even though

neuronal firing in this region is nonzero. Sharp changes in the model output appear in movement transitions for the network trained with Area 3. Area 4 accurately captures the food/mouth and mouth/rest regions, but misses the beginning of movement. Both the WF and RMLP display the following trends in the hand trajectory reconstruction:

- Area 1 is necessary to capture rest/food.
- Area 2 is not crucial in trajectory reconstruction.
- Area 3 relates to sharp transitions in trajectory.
- Area 4 is necessary to capture food/mouth.
- Combining multiple areas (e.g., 1, 3, 4) reduce magnitude of fluctuations in trajectory predictions (more so for the RMLP).

Table 6-1. Summary of cortical assignments

Area	Cortex
1	PP contralateral
2	M1 contralateral
3	PMd contralateral
4	M1/PMd ipsilateral
5	S1 contralateral
6	SMA contralateral
7	M1 ipsilateral

Carmen's Cortical Contributions

In the previous example, it was shown that the qualitative interpretations of neural activity are independent of the model topology. Even with two distinct topologies (linear-feedforward vs nonlinear-feedback) were used, the same cortices were contributing to the same parts of the movement trajectory. With this corroboration of results, we will continue the model based analysis of cortical activity through only one network, the RMLP. The procedure of training separate models for each cortex and combination of

cortices was repeated on the second owl monkey, Carmen. Testing trajectories are presented in Fig. 6-4 and the following trends were observed:

- All networks were unable to capture well the reach rest/food
- The network trained with area 2 was able to capture the reach food/mouth
- The network trained with areas 2 and 3 has a better fit during the movement than the network trained without area 3

In this experimental setup an interesting observation is that the lack of neuronal activity from the PP resulted again in trajectories that do not capture the reach from rest to the food. The second observation that M1 cells produced food/mouth trajectories corroborates the results obtained with Belle.

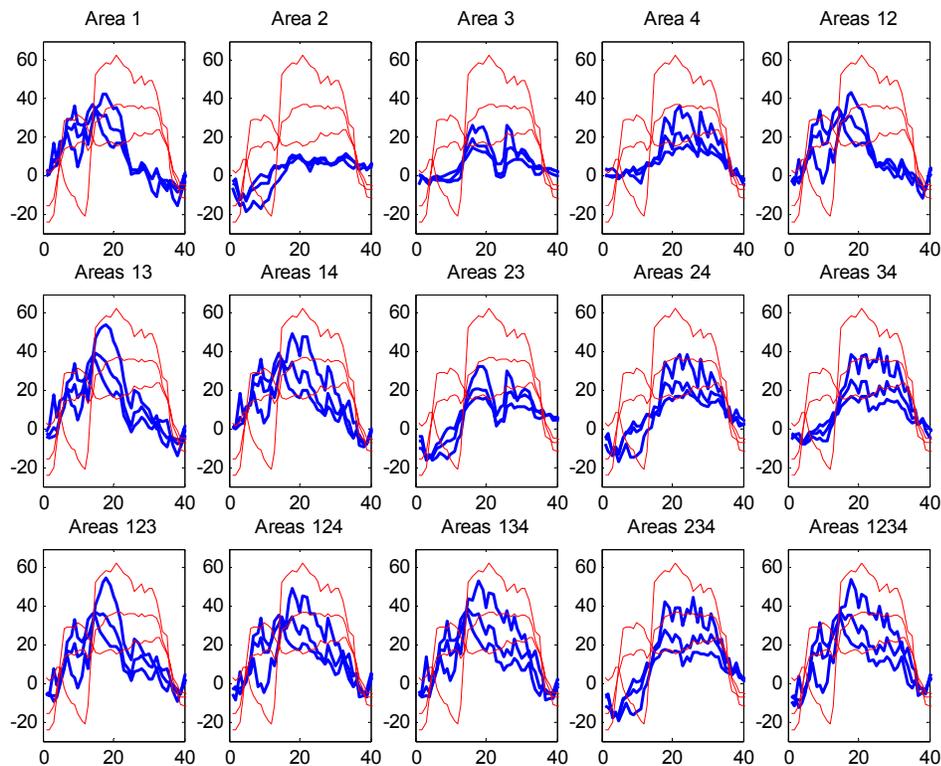


Figure 6-2. FIR filter (Aurora): Testing output X, Y, and Z trajectories (bold) for one desired movement (light) from fifteen Wiener filters trained with neuronal firing counts from all combinations of four cortical areas

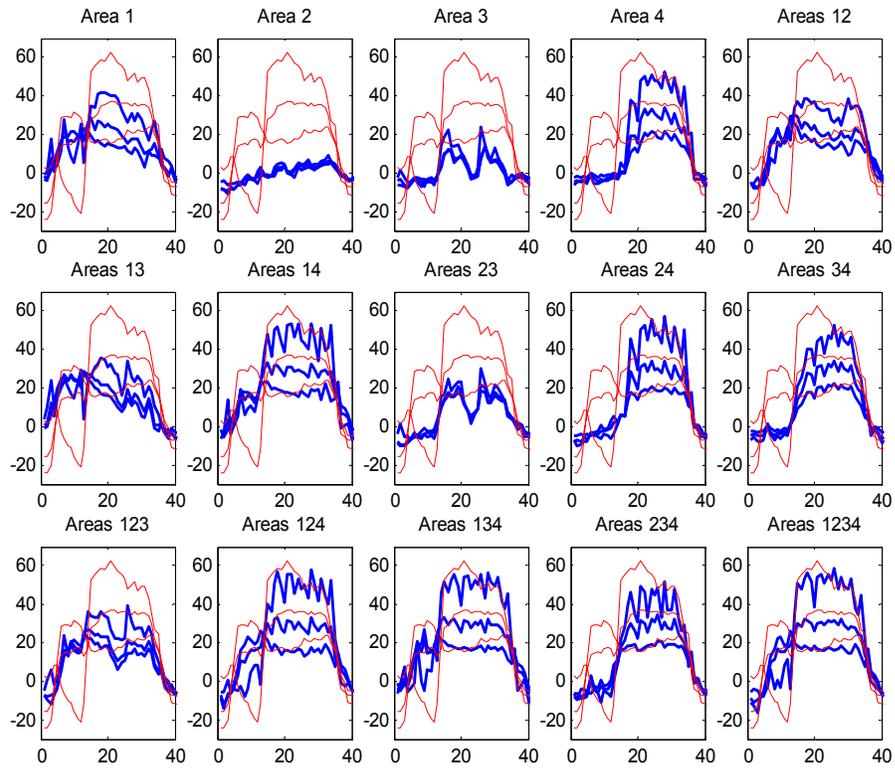


Figure 6-3. RMLP (Aurora): Testing output X, Y, and Z trajectories (bold) for one desired movement (light) from fifteen RMLPs trained with neuronal firing counts from all combinations of four cortical areas

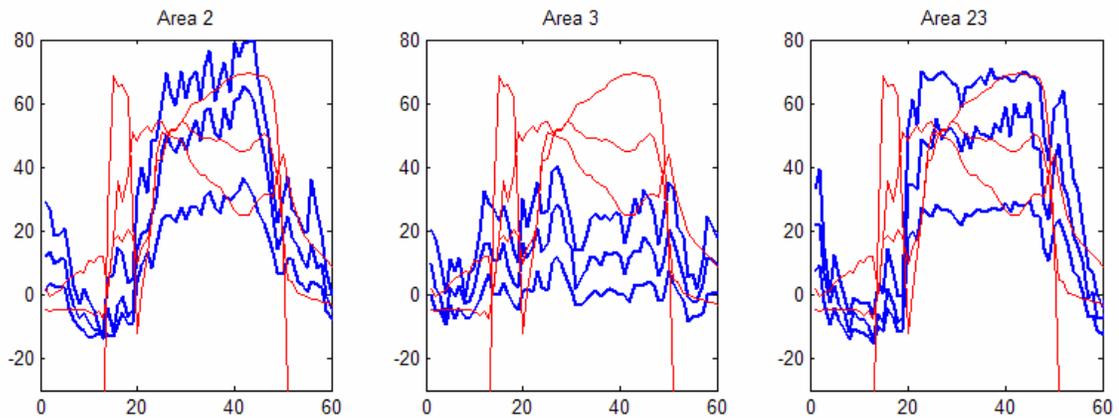


Figure 6-4. RMLP (Carmen): - Testing output X, Y, and Z trajectories (bold) for one desired movement (light) from three RMLPs trained with neuronal firing counts from all combinations of two cortical areas

Cortices Involved in Cursor Tracking

The simplicity of this procedure to infer principles about the role of cortices through trained input/output models was a motivating factor for repeating the experiments on a different species of primate and a different motor task. In the reaching task, we made qualitative assessments about the cortical contributions with respect to well defined landmarks in the movement trajectory. In cursor control, landmarks are not clearly defined because the cursor used for tracking is randomly placed in the workspace. To quantify the contributions, the testing correlation coefficients will be used and high correlations will indicate a large contribution. The correlation values are supplemented with scatter plots of the desired trajectory (x - markers) and the model outputs (o - markers) (Figs. 6-5 and 6-6). We again had the opportunity assess the cortical contributions of two individual primates (Rhesus monkeys) but in this case we sampled from additional cortices that included S1 and SMA (see Table 2-2 and 6-1).

The first Rhesus monkey, Aurora, generated neuronal activity from M1 that contributed strongly to the overall reconstruction of the trajectory. Correlation coefficient values from this area were equivalent to the model trained with all sampled cortices. To a lesser extent, Ivy's M1 cortex in Fig. 6-6 was also involved but it is overshadowed by the CC values of the PP. This lower value of correlation is surprising since compared to all the other primates Ivy has the largest number of M1 cells sampled (nearly 100). This result shows the degree of uncertainty in the recording process; even though M1 cells can be shown to produce model outputs that are highly correlated with the desired signal the cells sampled are not guaranteed to generate high levels of performance. In the ipsilateral M1 cortex, Belle's data contributed strongly while Aurora's cortex in area 7 of Fig. 6-5 produces outputs that were essential zero. For SMA in both Aurora and Ivy, the model

outputs were tightly clustered around zero and the correlation coefficients were small indicating a small contribution. Interestingly, the PP model in Ivy's dataset produced two dense clusters which may correspond to positions where movements were initiated however it is difficult to quantify this interpretation because of the random design of the experimental paradigm. Additionally, Aurora's S1 model produced a narrow band of outputs which may correspond to positions where neuronal activity increased due to the somatosensory input of the manipulandum pressing against her left hand. The result is speculative and again suggests the need to design a well controlled experiment to unequivocally verify this result.

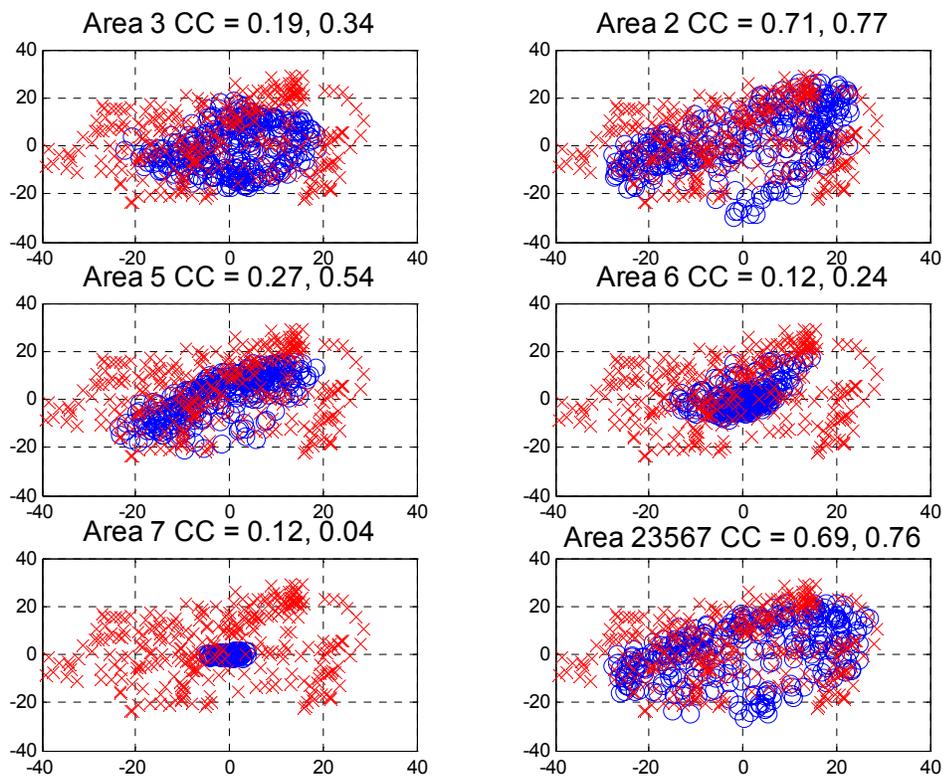


Figure 6-5. RMLP (Aurora): - Testing outputs (o – markers) and desired positions (x – markers) for six models trained with each separate cortical input. Testing (X, Y) correlation coefficients are provided in the title of each subplot

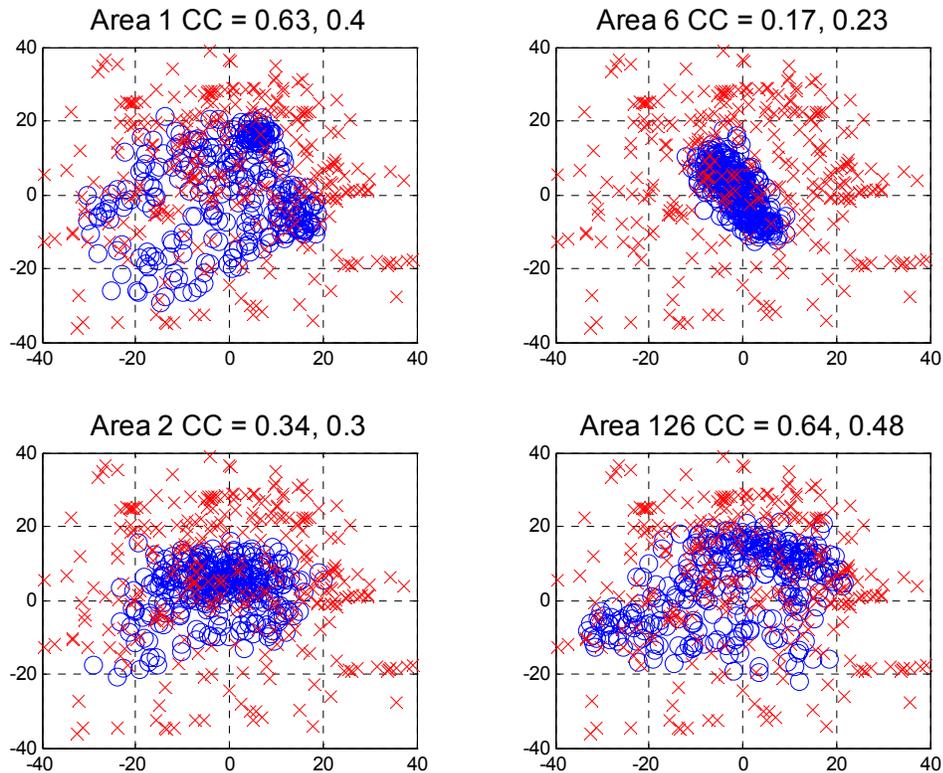


Figure 6-6. RMLP (Ivy): - Testing outputs (o – markers) and desired positions (x – markers) for four models trained with each separate cortical input. Testing (X, Y) correlation coefficients are provided in the title of each subplot

Discussion

We investigated the possibility of analyzing neural data from a BMI design perspective by considering the relative contributions of individual cortical regions and single neurons to the construction of hand trajectories through optimally trained models. It is encouraging that the qualitative interpretations of neural activity are independent of the model topology, even with two distinct topologies (linear-feedforward vs nonlinear-feedback). This builds confidence in our conclusions about neurophysiology drawn from signal processing techniques. The interpretations obtained by the model analyses corroborate the view of broad tuning of the motor cortex, that is, the spatio-temporal

encoding of the motor information is such that only a minute population of 100+ neurons is enough to enable a relatively precise mapping spike counts to hand movements. But the broad tuning seems to be limited to local organization. It seems that different cortical areas are required to track the different parts of the reaching task. For Belle, the PP was controlling the reach to the food and the M1 was controlling the reach from food to mouth. To a lesser extent, the PMd was controlling the transitions in the movement. For Aurora and Ivy the PP and M1 were again shown to be critical for producing high testing CCs. This indicates that the electrodes should be placed strategically throughout the motor cortex to capture vital information. If electrodes are not placed in a cortical region important for a part of the movement, the trajectory cannot be reconstructed well.

This cortical analysis motivates appropriate questions about the roles of cortical regions in voluntary movements so that we can compare our observations with experimental neuroscience. Using well controlled experiments, the posterior parietal cortex has been associated with motor imagery [66], visual/tactile manipulation of objects [67], and spatial coordinate transformations [2]. In our alternative analysis approach, we repeatedly identified the PP as the active area during the rest/food reach, which is a task that may involve all of the mentioned PP associations. This could be an additional example confirming the role of PP. A growing body of research has presented a relaxed view of motor hemispheric specializations (i.e. each hemisphere controls the contralateral *and* ipsilateral limbs) [68-70]. The results given by our model analysis are in accordance with this relaxed idea of specialization and can be inferred from the result that the right M1 cortex (of Belle) was the primary area that the RMLP used to construct the right hand

reach from mouth to rest. At the very least, this method of model analysis can be used as an additional tool to extract relationships between neural activity and behavior.

Our last comment is on the scalability of this analysis approach. So far, we only utilized data for a reaching and cursor control task. However we have shown that the best models primary involved the firing patterns from M1 however the contributions are not necessarily guaranteed and do not scale with the number of cells (as in the case of Ivy). Qualitatively the importance of areas in the two tasks remained consistent and we anticipate that the global view of cortical contributions will remain for other movements. For a given random subsampling of cortical neurons though, a limited repertoire of reconstructable movements may result depending upon the amount redundancy in the specific region.

CHAPTER 7 ASCERTAINING THE IMPORTANCE OF NEURONS

Introduction

We discussed earlier that model overfitting is a significant problem when hundreds of neurons are used as model inputs. We showed that the introduction of extra degrees of freedom not related to the mapping can result in poor generalization, especially in topologies where tap-delay memory structures are implemented in the neural input layer. The problem occurs with each additional memory delay element scales the number of free parameters by the number of input neurons. This explosion in the number of free parameters also puts a computational burden on computing an optimal solution especially when the goal is to implement the BMI in low-power, portable hardware.

As a first step in reducing the number of free parameters, more parsimonious models such as the recurrent multilayer perceptron (RMLP) [12, 14, 71] were studied. The topology of this model can significantly reduce the number of free parameters by implementing feedback memory structures in hidden network layers. Secondly, during model training, regularization techniques were also implemented [59] that attempt to reduce the value of unimportant weights to zero and effectively prune the size of the model topology. However, this approach is strictly a statistical modeling technique that requires lots of data and computation, it is not trivial to use, and does not necessarily provide information about importance of neurons. Finally, the number of inputs given to the models could be manually pruned, but it is difficult to know how it will affect BMI performance.

For these reasons, we develop here techniques and compare tools that can be used to assess the utility of recorded neural ensembles in an adaptive modeling framework. Since our ultimate goal for BMIs is to design the most accurate reconstructions of hand kinematics from cortical activity using adaptive signal processing techniques, it seems natural to equate here neural importance to sensitivity of inputs to model fitting quality. Moreover, these measures should be compared with the available neurophysiologic knowledge, with the hope that we can understand better the data, enhance our methodologies and ultimately the performance of BMIs. Therefore, the importance of neurons will be ascertained using two techniques:

- Sensitivity analysis through trained linear and nonlinear models.
- Cellular directional tuning analysis.

Given a set of data, we would like to evaluate how well these two methodologies are able to find important neurons for building BMI models. Secondly, we would like to use this information to tackle the model generalization issues encountered in BMIs. The goals of the study are formulated in the following questions:

- Can our methods automatically indicate important cells for the prediction of the kinematic variables in the tasks studied?
- In this model-based framework, can better BMIs be built using a subset of important cells?
- Is the model-independent technique a good indicator of cellular importance, and how is it related to sensitivity through the linear model?

It is well known that neurons vary in their involvement in a given task [72].

However, quantifying neuronal involvement for BMI applications is still an ongoing area of research. This is where BMI modeling is an asset, because once trained, the model implicitly contains the information of how cells contribute for the mapping. The difficulty is that the assessment is in principle dependent upon the type of model chosen

to predict the kinematic variables and its performance (model-dependent). We will first compare the linear FIR model and nonlinear RMLP and contrast the ranking of neural importance with the tuning characteristics of each neuron established directly from the data (hence we call this method model-independent). Our second question quantifies the change in performance when only a small subset of cells is used to build the BMI. In principle one could think that any subset of cells will perform worse than the whole ensemble, but due to the poor generalization of large models, performance may be in fact better in a test set with a reduced number of important cells. Of course, this also makes BMIs more dependent upon the stability over time of these cells, and in the long run we have shown that performance can either worsen or improve. The final question seeks to validate ranking of neurons obtained through trained models with directional tuning analysis, which has been used to model the role of motor cortex neurons. First, by picking cells that are highly tuned to the prediction of kinematic variables, BMI performance will be tested. Secondly, the tuning features of the important cells automatically chosen by the models will be analyzed. The ultimate goal is to improve understanding of how cells encode kinematic parameters so that better “gray-box” models can be built using the underlying information in neural recordings.

Assumptions for Ranking the Importance of a Neuron

We would like to obtain an automatic measure of each cell’s contribution to encoding motor parameters for a given task, which we call the cell importance. For this reason, a structured approach is taken to ascertaining the importance of neurons with the two methods described above. Our methodological choices however are not free from assumptions. First, the methods assume stationarity in the data. A *snapshot* of neural activity is taken and importance is ascertained without addressing time variability in the

recordings, which is a shortcoming. Second, in spite of the highly interconnected nature of neural structures, both the cellular tuning is computed independently for each individual neuron. With this independence assumption, it is difficult to quantify the importance of pairs, triples, etc. of cells. In contrast, the model sensitivity analysis considers covariations in firing rate among groups of cells in the neural ensemble, but depends on the type of model utilized. Third, the cellular tuning technique only considers the instantaneous neural activity while the modeling techniques include memory structures (tap-delay-lines). Finally, each technique for ascertaining importance focuses on different neuronal firing features. In the case of directional tuning, the local variations of the firing rate are of interest, while in the model sensitivity variations in firing rate over the analysis interval are quantified.

Sensitivity Analysis for Reaching Tasks

With the weights of the trained linear and nonlinear networks, we have a tool with which we can identify the neurons that affect the output the most. A sensitivity analysis, using the Jacobian of the output vector with respect to the input vector, tells how each neuron's spike counts affect the output given the data of the training set. Since the model topology can affect the interpretation of sensitivity we will first examine sensitivities through the FIR filter which has served as the "control model" throughout this dissertation. The procedure for deriving the sensitivity for a feedforward topology is an application of the chain rule [73]. For the case of the FIR filter, differentiating the output with respect to the input (see Eq. 3-15) yields directly a sensitivity with respect to each neuronal input i in Eq. (7-1).

$$\frac{\partial \mathbf{y}_j}{\partial \mathbf{s}_i} = \mathbf{W}_{10(i-1)+1:10(i-1)+10,j} \quad (7-1)$$

Hence, a neuron's importance can be determined by simply reading the corresponding weight value¹⁵ in the trained model, if the input data for every channel is power normalized. Since for neural data this is not the case, the neuron importance is estimated in the vector Wiener filter by multiplying the absolute value of a neuron's sensitivity with the standard deviation of its firing computed over the dataset¹⁶ as in Eq. (7-2). To obtain a scalar sensitivity value for each neuron, the weight values are also averaged over the ten tap-delays and three output dimensions.

$$Sensitivity_i = \sigma_i \frac{1}{2} \sum_{j=1}^3 \frac{1}{10} \sum_{k=1}^{10} |\mathbf{W}_{10(i-1)+k,j}| \quad (7-2)$$

The procedure for deriving the sensitivity for a feedforward multilayer perceptron (MLP) also discussed in [73] is again a simple application of the chain rule through the layers of the network topology as in Eq. (7-3):

$$\frac{\partial y_2(t)}{\partial s(t)} = \frac{\partial y_2(t)}{\partial y_1(t)} \frac{\partial y_1(t)}{\partial s(t)} \quad (7-3)$$

Since our RMLP model displays dependencies over time that results from feedback in the hidden layer, we must modify the procedure presented in [73]. Starting at each time t we compute the sensitivities in Eq. (7-3) as well as the product of sensitivities clocked back in time. For example, using the RMLP feedforward equations (see Eq. 3-18 and 3-19) we can compute at $t = 0$ the chain rule shown in Eq. (7-4). \mathbf{D}_t is the derivative of the hidden layer nonlinearity evaluated at the operating point shown in Eq. (7-5). Notice at $t = 0$ there are no dependencies on y_1 . If we clock back one cycle, we must now include the

¹⁵ In this analysis we consider the absolute values of the weights averaged over the two output dimensions and the *ten* tap-delays per neuron.

¹⁶ By multiplying the model weights by the firing standard deviation we have modified the standard definition of sensitivity; however, for the remainder of this paper we will refer to this quantity as the model sensitivity.

dependencies introduced by the feedback, which is shown in Eq. (7-6). At each clock cycle back in time, we simply multiply an additional $\mathbf{W}_f^T \mathbf{D}_{t-i}$. The general form of the sensitivity calculation is shown in Eq. (7-7). Experimentally we determined that the effect of an input decays to zero over a window of 20 samples (Fig. 7-1). At each time t the sensitivity of the output with respect to the input is represented as the sum of the sensitivities over the 20-sample window.

$$\frac{\partial \mathbf{y}_2(t)}{\partial \mathbf{s}(t)} = \mathbf{W}_2^T \mathbf{D}_t \mathbf{W}_1^T \quad (7-4)$$

$$\mathbf{D} = \begin{bmatrix} f'(z_1^1) & 0 & \dots & 0 \\ 0 & f'(z_1^2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & f'(z_1^n) \end{bmatrix} \quad (7-5)$$

$$\frac{\partial \mathbf{y}_2(t)}{\partial \mathbf{s}(t-1)} = \mathbf{W}_2^T \mathbf{D}_t \mathbf{W}_f^T \mathbf{D}_{t-1} \mathbf{W}_1^T \quad (7-6)$$

$$\frac{\partial \mathbf{y}_2(t)}{\partial \mathbf{s}(t-\Delta)} = \mathbf{W}_2^T \mathbf{D}_t \left(\prod_{i=1}^{\Delta} \mathbf{W}_f^T \mathbf{D}_{t-i} \right) \mathbf{W}_1^T \quad (7-7)$$

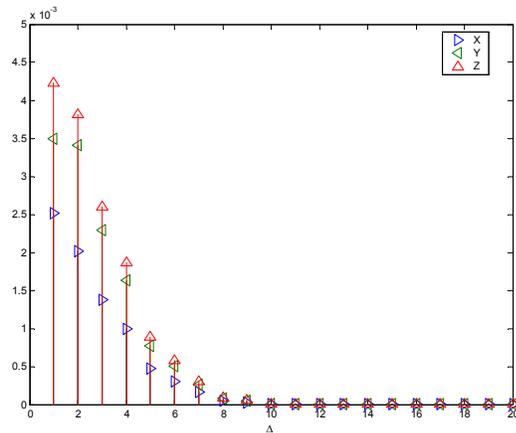


Figure 7-1. Sensitivity at time t for a typical neuron as a function of Δ

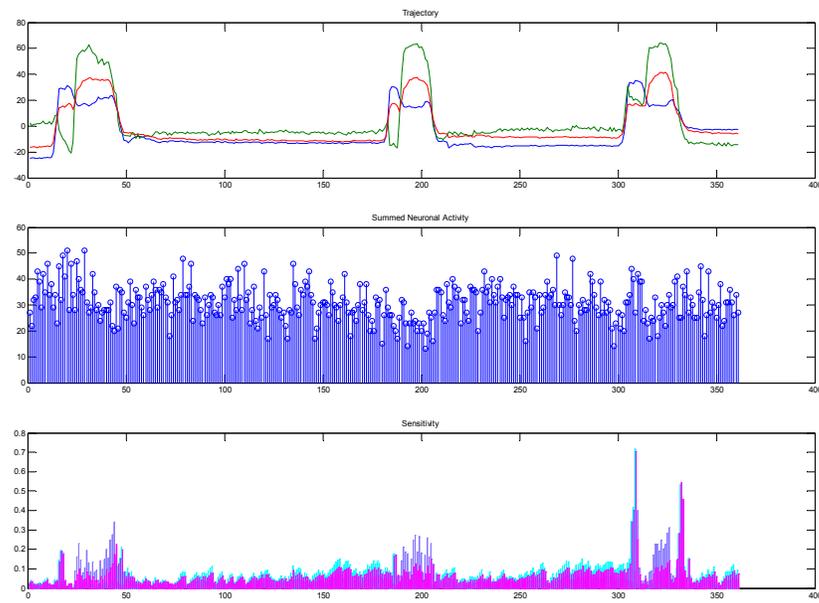


Figure 7-2. RMLP time-varying sensitivity. A) X, Y, Z desired trajectories for three similar movements. B) Neuronal firing counts summed (at each time bin) over 104 neurons. C) Sensitivity (averaged over 104 neurons) for three coordinate directions

Compared to the FIR model which produces a static measure of sensitivity, the RMLP produces a time varying sensitivity that we will now use to analyze three similar movements from the testing trajectory; specifically reaching movements shown in the testing plots of Fig. 3-6. We should note that the first movement is the same reaching movement used in the analysis of cortical contributions for the reaching task. The three movements chosen are presented in Fig. 7-2. Notice that movements are similar except that movement 3 does not have a decrease in the y-coordinate. To visualize trends in how the input affects the output, we plot the neuronal activity (input) along with the computed sensitivity (averaged over 104 neurons and 20 delays) in Fig. 7-2 B-C. We take first a macroscopic view of both the neuronal activity and sensitivity by summing over 104 neurons at each time bin. Comparing Fig. 7-2 A with Fig. 7-2 B we see that it is difficult

to visually extract features that relate neuronal firing counts to hand movement. Despite this fact, a few trends arise:

- Neuronal activity consists of a time-varying firing rate around some mean firing rate.
- Movements 1 and 3 seem to show increased neuronal activity at the beginning and end of a movement while 2 does not.
- All three movements contain a decrease in neuronal activity during the peak of the movement.

With these three trends in mind, we include now the sensitivity plot estimated for

the RMLP in Fig. 7-2 C. We can observe the following:

- In general, the network becomes more sensitive during all three movements.
- Sensitivity is large when velocity is high.
- The sensitivity shows large, sharp values at the beginning and end of the movement and a plateau during the peak of the movement.

From sensitivity peaks during the movements in Fig. 7-2 C, we ascertain that the sensitivity analysis is a requirement for relating neuronal activity to behavior. Without the model based sensitivity, finding relationships in the raw data is difficult. Now that we have a mesoscopic view (from the cortical contribution analysis and this overview of sensitivity analysis) of how the ensemble of sampled neuronal activity affects the output of the network, we change our focus and use the model based sensitivity to “zoom in” on the important individual neurons. We are interested in learning why the individual neurons in a given cortical region are necessary for constructing the network output movement. Using the sensitivity analysis we can select neurons that are most closely associated with the reaching movements (i.e. neurons that elicit large perturbations in the output with small perturbations in firing).

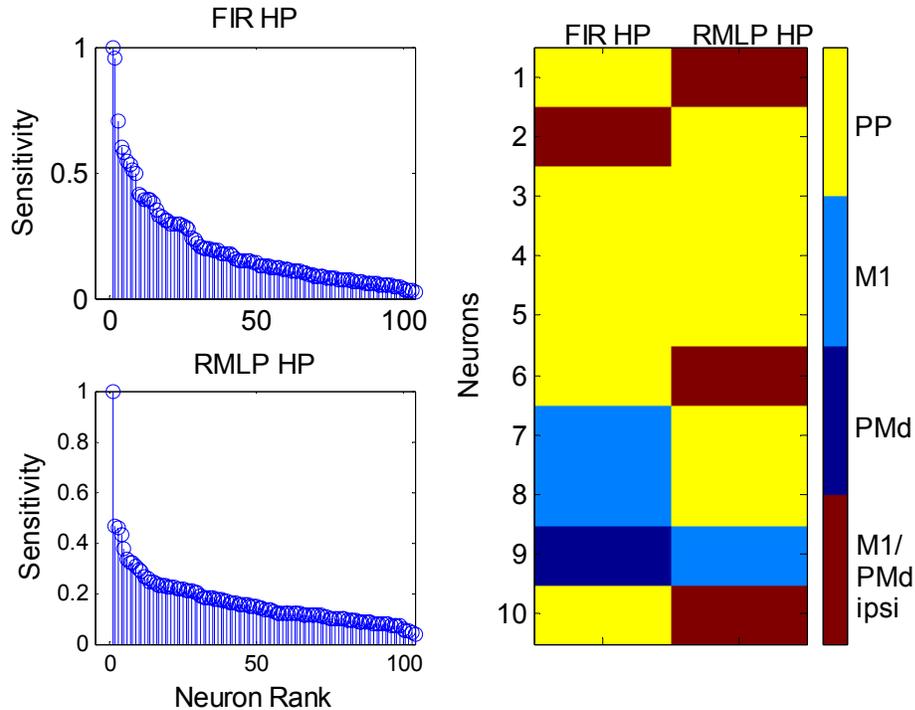


Figure 7-3. Reaching task neuronal sensitivities sorted from minimum to maximum for a movement. The ten highest sensitivities are labeled with the corresponding neuron

Neurons for a particular movement are selected for the RMLP by choosing the maximum of a sorted list of neuronal sensitivities computed by averaging the time-varying sensitivity in Eq. 7-7 over the interval of a movement and all three coordinate directions. For comparison, the neurons from the FIR filter are sorted directly using Eq. 7-2. The sorted ensemble neuronal sensitivities for both the FIR and RMLP models (Fig. 7-3) shows an initially sharp decrease from maximum indicating that only a few neurons are required for outlining the movement reconstruction. Of the 104 neurons from Belle used in this study, the ten highest sensitivity neurons for a given movement are presented in the image plot in Fig. 7-3 and are color coded by cortical area. The highest sensitivity neurons are primarily distributed over cortical areas 1 and 4. This finding that few neurons in regions 2 and 3 are sensitive corroborates with the results from the multiple

RMLPs trained with neural data from all combinations of cortical regions. Of the 104 neurons, 7 of the 10 highest-ranking neurons are common for the WF and RMLP. By experimentally verifying the effect of computing the sensitivities through both WF and RMLP topologies, we have found that sensitivity based selection of neurons is not heavily dependent upon the model topology, even with two distinct topologies (linear-feedforward vs nonlinear-feedback) are utilized.

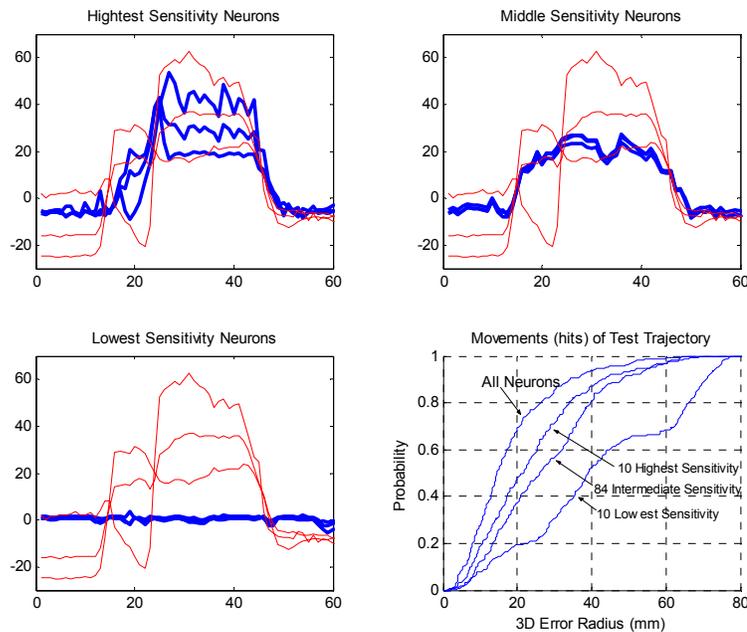


Figure 7-4 Testing outputs for RMLP models trained with subsets of neurons. A,B,C) X, Y, and Z trajectories (bold) for one movement (light) from three RMLPs trained with the highest, intermediate, and lowest sensitivity neurons. D) CEM decreases as sensitive neurons are dropped

As a further check of the sensitivity importance measure, we trained three identical RMLP networks of 5 hidden PEs, one with the ten highest sensitivity neurons, one with the eighty-four intermediate sensitivity neurons, and one with the ten lowest sensitivity neurons. A plot of the network outputs in Fig. 7-4 A, B, and C shows that the reduced set of highest sensitivity neurons do a good job at capturing the peaks of the movement while the lowest sensitivity neurons are unable to discover the mapping from spike trains to

hand movements. The remaining intermediate sensitivity neurons do contain information about the movement but it is not enough to capture the full detail of the trajectory, but improve the overall trajectory fitting. Using the CEM for BMI performance in movements, we see that the curves approach the bisector of the space as sensitive neurons are dropped from the mapping indicating a decrease in performance. This result verifies that the method of RMLP model based sensitivity analysis for BMIs is able to produce a graded list of neurons involved in the reconstructing a hand trajectory

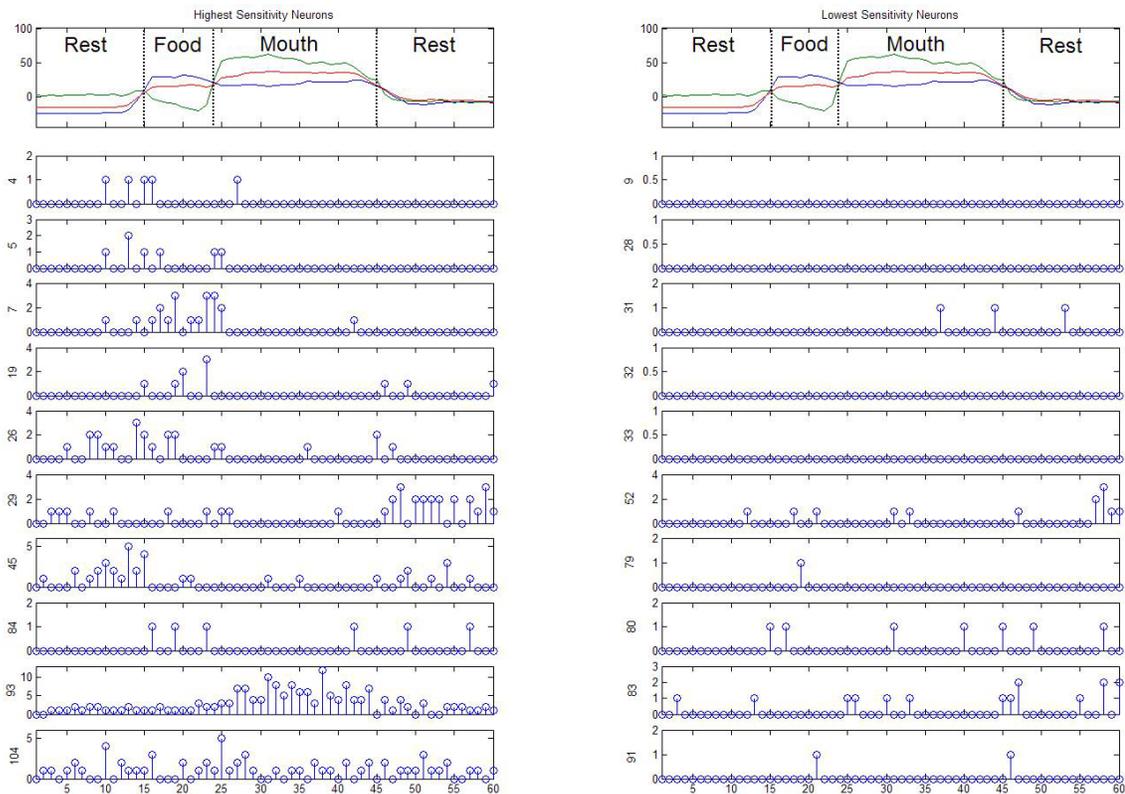


Figure 7-5. Belle's neuronal firing counts from the ten highest and lowest sensitivity neurons time-synchronized with the trajectory of one reaching movement

We now need to address why the neurons with the highest sensitivity are important for the production of the output movement. The answer is quite simple and can be obtained by plotting the binned firing counts for highest and lowest sensitivity neurons in the vicinity of a reaching movement (Fig. 7-5). In the top subplot of each column, we

have included the first reaching movement as in the above simulations. For the highest sensitivity neurons, we can see some strong correlations between the neuronal firing and the rest/food, food/mouth, and mouth/rest movements. Neurons 4, 5, 7, 19, 26 (Area 1 – Belle, S1) display increased firing during the rest/food movement and are zero otherwise. Neurons 93 and 104 (Area 4 - Belle, S1) show a firing increase during the food/mouth movement. Neuron 45 (Area 2 - Belle, S1) does not correlate with the movement directly but seems to fire right before the movement indicating that it may be involved in movement initiation. Neuron 84 (Area 4 - Belle, S1) fires only single pulses during the transitions of the movement. In the firing patterns of these neurons, we see the effect of the decay of the time varying sensitivity at time t which can be influenced by samples 2 seconds in the past.

All of the lowest sensitivity neurons primarily contain single firings, which do not display any measurable correlations to the movement. The data from these neurons is relatively sparse when compared to the highest sensitivity neurons indicating that the network is not using them for the mapping.

Cellular Importance for Cursor Control Tasks

Model-Dependent – Sensitivity Analysis

We extend our analysis methodology for selecting important neurons for constructing the BMI mapping to the cursor control task. This time, however, the analysis will include all the kinematic variables available, HP, HV, and GF from Aurora's dataset. For each variable, separate RMLPs were trained and the sensitivity was computed as in Eq. 7-7. For the 2-D cursor task sensitivities were averaged over the X/Y coordinates and an arbitrary 3000 sample window since no clear marker of movement/non-movement could be discriminated. The sorted position, velocity, and gripping force neuronal

sensitivities for each session's ensemble of neurons is plotted in Fig. 7-6 A and 7-6 B. Like Belle's cellular analysis, an initially sharp decrease from maximum indicates that there are a few neurons that are significantly more sensitive than the rest of the ensemble. For HP and HV, the most important neurons are again mostly located in the primary motor cortex (M1). For the GF we can also see that initially during first session multiple brain areas are contributing (30% PMd, 10% SMA, 10% S1) but many of these cells are replaced during Session two with cells from M1.

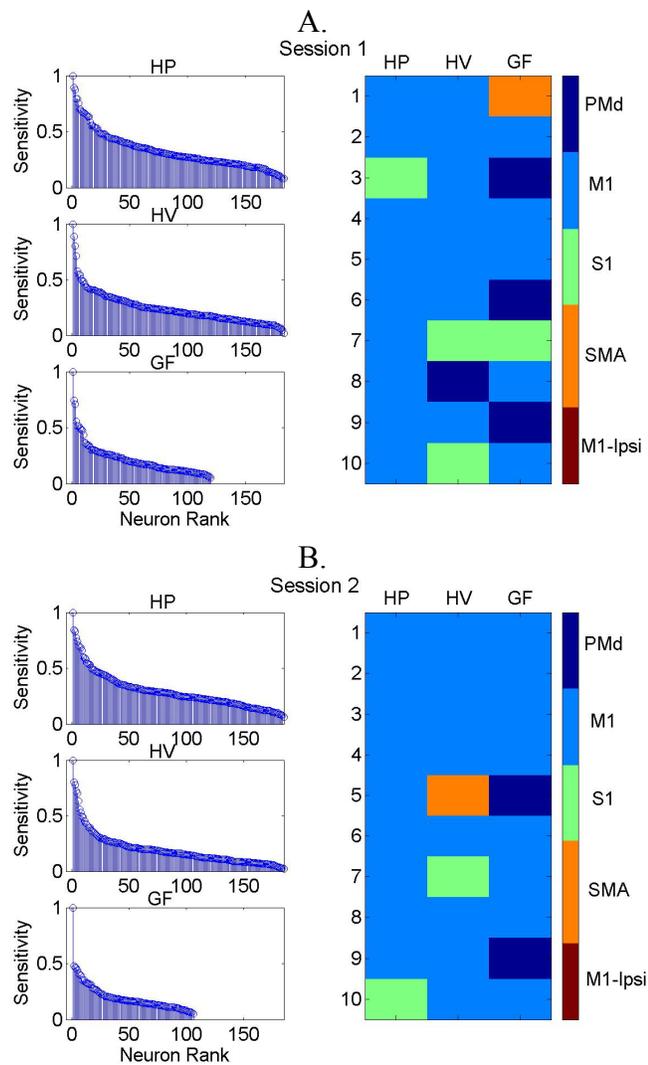


Figure 7-6. Sensitivity based neuronal ranking for hand position and velocity for two sessions using a RMLP. The cortical areas corresponding to the ten highest ranking HP, HV and GF neurons are given by the colormap

Model-Independent – Cellular Tuning Analysis

The second method for ascertaining the importance of neurons involves computing the tuning or preferred direction [5] of each cell in the ensemble. Tuning curves convey the expected value of a probability density function indicating the average firing a cell will exhibit given a particular movement direction. In this approach, the data is analyzed in the neural space as compared to using an input-output model that optimizes a set of parameters using a statistical learning criterion to map the neural inputs and desired kinematics.

It has been shown in the literature that a variety of hand trajectories can be reconstructed by simply weighting and summing the vectors indicating preferred directions of cells in an ensemble of neurons [6, 7, 15, 18]. Neurophysiologic evidence suggests that the direction of hand movements is encoded in cells that have cosine shaped tuning curves. Our metric for ranking will be the tuning depth of a cell's tuning curve. This quantity is defined as the difference between the maximum and minimum values in the cellular tuning. For an impartial comparison of the cellular tuning depth, the tuning curves are normalized by the standard deviation of the firing rate. This measurement relates how "peaky" the tuning curve is for each cell and is an indicator of how well modulated the cell's firing rate is to the kinematic parameter of interest.

To use this measurement, the hand direction tuning curves and gripping force histograms must first be computed for each cell. Since this investigation of kinematic parameters exists in different dimensional vector spaces (HP, HV in 2D, GF in 1D spaces), the differences in the computation are described here. In the multidimensional case, hand direction is determined by using the desired hand position from which the corresponding velocity vector (magnitude and direction) between successive points in the

trajectory is computed. The quantity that is commonly used in BMI experiments is the hand movement direction measured as an angle between 0 and 360 degrees [10]. Since cellular tuning can produce properties where the average of angle 0° and 360° is not 180° (this results from the wrap-around effect of the measurements), the mean of each cell's hand direction tuning is computed using circular statistics as in (7-8)¹⁷ [74],

$$\text{circular mean} = \arg\left(\sum_N r_N e^{i\theta_N}\right) \quad (7-8)$$

where r_N is the cell's average firing rate for angle θ_N where N is from 1 to 360.

In the 1-D case of force, the vector space is restricted to the magnitude of the vector. Typical force histograms are ramp functions that saturate since cells tend to increase their activity proportionally to greater force demands up to the maximum firing rate, so the range is still finite [75]. Moreover, since GF curves are not subject to the wrap-around effect, (7-8) is disregarded and the maximum value of the firing rate curve of each cell is used to determine the preferred force the cell is tuned to. While the force curves obtained in this analysis do not fit the classical definition of “tuning”, for simplicity we will use the term tuning depth for both GF and HV for the remainder of this study. Despite these differences, the depth of tuning of each cell can still be used as a measure of information content provided by the cells¹⁸. Since computing the tuning depth from finite data is noisy, the curves are smoothed by using a coarse resolution¹⁹ to

¹⁷ In computing the circular mean, we used the four quadrant inverse tangent.

¹⁸ We have observed that the tuning of a cell can vary as a function of the delay between the generation of the neural activity and the physical movement of the hand. Since we are interested in tuning depth, we are looking for cells that have the smallest variance in their tuning function. After computing tuning variance across all cells for delays up to 1 second, we found that the sharpest tuning occurs at the 0th delay or the instantaneous time alignment. This was the delay chosen for the remaining sections of this study.

¹⁹ For hand direction, 45 degree bins were chosen. In the case of force, the dynamic range of the measurement was divided into 10 nonoverlapping bins.

count the number of times that each cell fires. The sum of neuronal firings is also normalized for each angle/force by the total number of times that the hand visits the angle/force. Essentially with this method the cellular mean firing rate for each angle/force is computed.

In Fig. 7-7 A and 7-7 B (left subplots) a plot of the ranked list of cells is given for hand direction and gripping force based upon the normalized depth of their tuning. From the plots of both sessions, we can see that the histograms have a knee between 10 and 40 (depending on the kinematic variable) suggesting that the cells can be divided into two groups according to tuning depth.

One can expect that for model building, these highly tuned cells will be more relevant. But tuning depth is not the only parameter of interest for a BMI since the input must also supply sufficient information to reconstruct *all* of the HPs, HV, and GFs desired trajectories, that is, the set of deeply tuned neural vectors must create a basis that spans the space (i.e. angles 0-36, 37-72, ...) of the desired kinematic variable. Therefore each cell was ranked by the tuning mean angle and the corresponding depth of tuning was used to select cells to cover the space. For simplicity and robustness, the hand direction space is divided into 10 equally spaced bins, and in each bin, the cell that exhibits the deepest tuning is selected.

The brain areas corresponding to the deepest tuned cells that span the kinematic space are presented in the right subplots of Fig. 7-7. Notice the different brain areas that contribute deepest tuned cells for GF and HV. In the first session, the top cells for hand direction are primarily from M1 while gripping force involves not only M1, but also S1, and PMd. In session two, hand direction is still dominated by M1, but two of the top PMd

cells for gripping force are replaced with M1 cells. In both figures, it is interesting to observe that S1, a somatosensory cortical area, was ranked highly in the cell ordering for motor tasks.

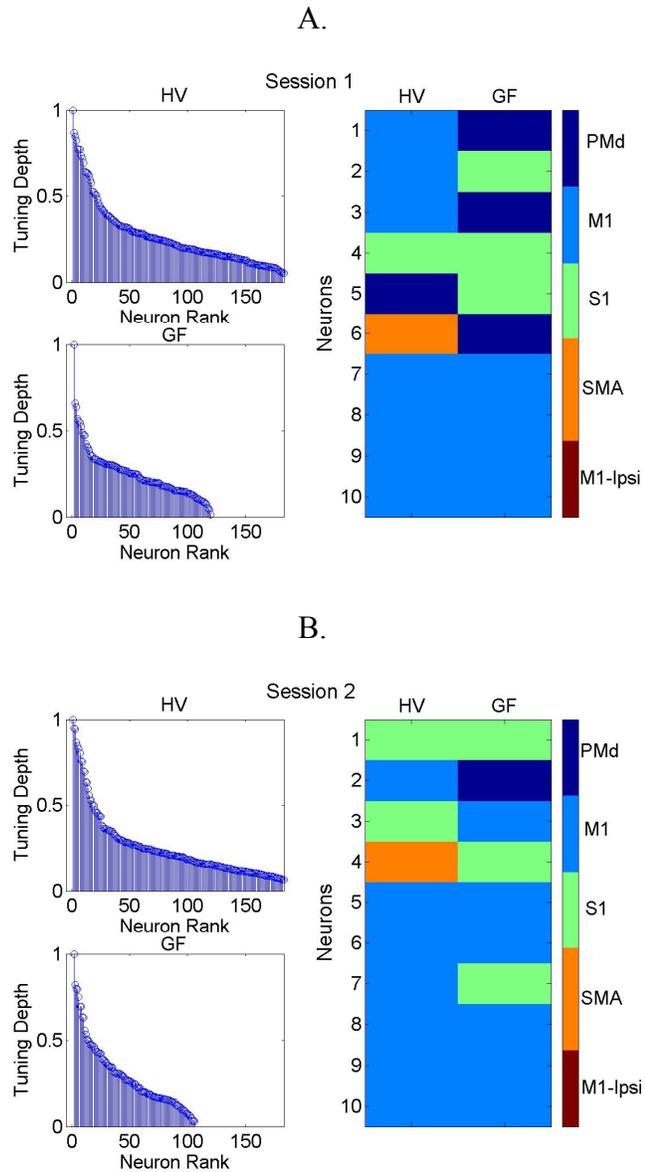


Figure 7-7. Neuronal ranking based upon the depth of the tuning for each cell. The cortical areas corresponding to the most sharply tuned cells is given by the colormap

The corresponding tuning curves for the top ten cells are shown in Fig. 7-8. Here the maximum and minimum values for each tuning curve are represented by white and

black respectively. In both sessions, for hand direction and gripping force, we could find cells that span the kinematic variables (i.e. have their maximum values approximately on the diagonal). Fig. 7-8 also shows the differences in the tuning between hand direction and gripping force where in (E) a representative neuron's tuning curve shows a bell-shape while in (F) the neuron exhibits a (left to right) graded increase in the tuning curve from smaller to maximum force. The latter organization is consistent with a threshold of firing arranged by force level.

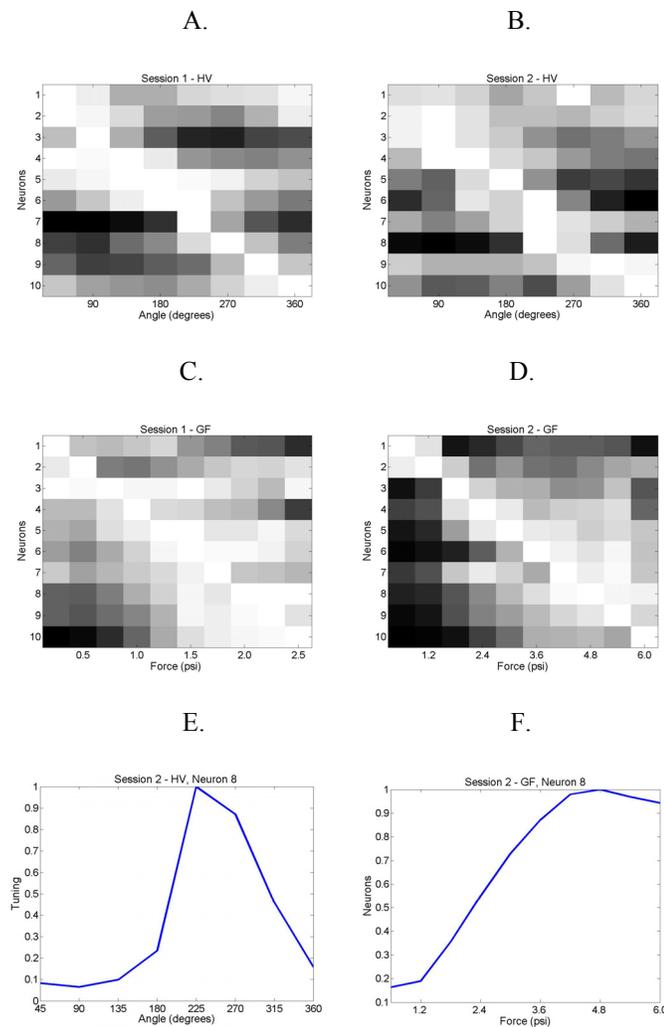


Figure 7-8. A-D) Cellular tuning curves for hand direction and gripping force using a model independent ranking method, E-F) Tuning curves for two representative cells from plots B) and D)

Relative Ranking of Neurons

The two approaches utilized to ascertain the importance of cells for reconstructing hand kinematics lead to very similar distributions of cell rankings, and consistently point to similar brain areas for the important neurons. We now quantify in more detail the relative ranking of cells given by each method.

The pair wise ordering of cells from the two methods is compared by plotting the normalized²⁰ rank value of each cell as in Fig 7-9. Ideally, if both methods contained the same order in ranking, all points would lie on the line $y = x$. In these figures, we did observe differences in the rankings produced by the three methods. Principal component analysis (PCA) revealed that the eigenvector corresponding to the largest eigenvalue roughly aligns with the diagonal. Additionally, the ratio of first and second eigenvalues was large for each scatter plot (Sensitivity vs. Tuning Depth – 4.40). This can be interpreted as meaning that the methods tend to pick the same top ranked cells. Table 7-1 shows in bold the common cells in the group of the 10 top ranked cells.

Table 7-1. The 10 top ranked cells

HP	HV		GF	
<i>Sens.</i>	<i>Sens.</i>	<i>TD</i>	<i>Sens.</i>	<i>TD</i>
99*	110*	126*	41*	82*
80	72	67	44	20
69	80	149	62	41
107	114	167	53	76
104	167	106	61	49
68	99	104	56	57
81	149	76	57	71
110	85	80	72	53
84	106	69	27	44
149	84	72	14	56

*Neurons are listed from most important to least important

²⁰ The ranking values for each method were normalized by the maximum.

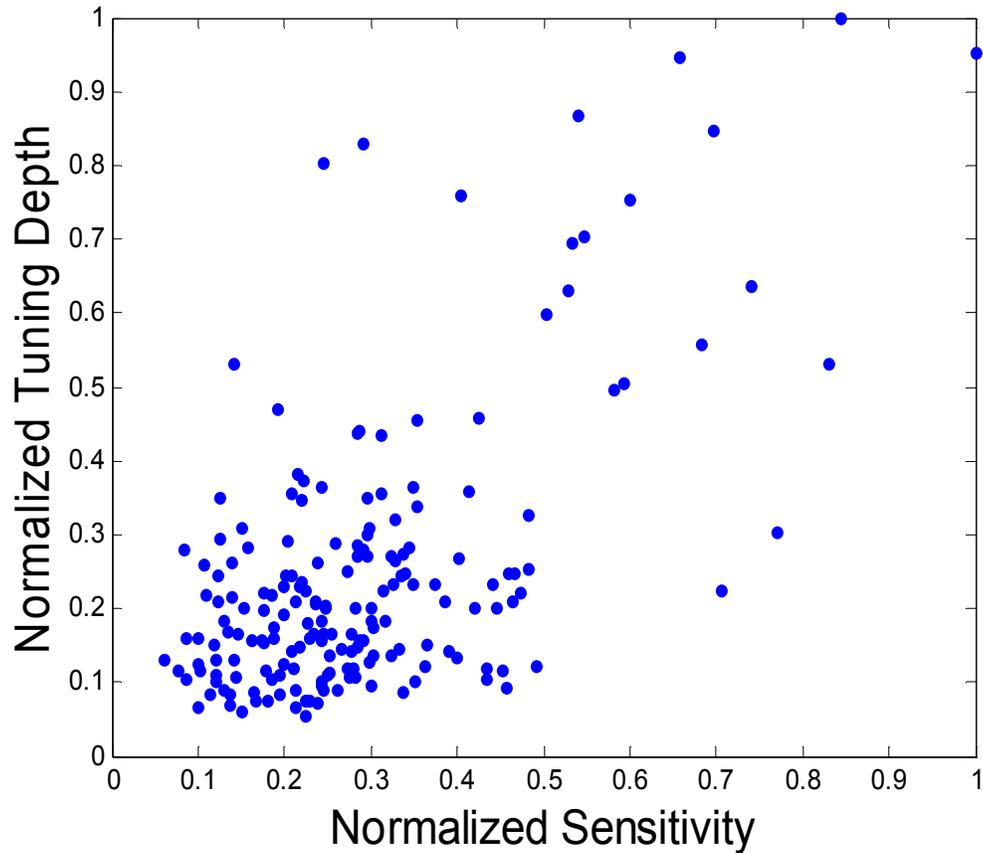


Figure 7-9. Scatter plot of neuron ranking for tuning depth and sensitivity analysis

Model Performance with a Subset of Sensitive Cells

For uniformity with the tuning curve method, only the ten most important cells were picked from each ranking and then used as model inputs. As in all cursor control modeling experiments, the same 8,000 data points were chosen for evaluating the models (5000pts training, 3000pts testing).

The performance of each model was evaluated by computing the correlation coefficient between the model output and the desired kinematic trajectory. In Table 7-2, the average and best test set correlation coefficients for both sessions, coordinates (X, Y), and kinematic variables (HP, HV, and GF) are presented when all of Aurora's 185 neurons are used in the modeling.

Table 7-2. Test set correlation coefficients using the full ensemble

	CC	HP		HV		GF
		X	Y	X	Y	n/a
Session 1	Average	0.75	0.74	0.76	0.76	0.87
		± 0.21	± 0.19	± 0.16	± 0.16	± 0.12
Session 2	Average	0.66	0.79	0.73	0.77	0.92
		± 0.24	± 0.16	± 0.18	± 0.13	± 0.06

Table 7-3. Test set correlation coefficients using 10 most important neurons

			HP		HV		GF
			X	Y	X	Y	n/a
Session 1	Model-Independent	Average	0.65	0.57	0.70	0.69	0.86
			± 0.27	± 0.30	± 0.20	± 0.18	± 0.14
	K-S test	1	1	1	1	0	
	RMLP Sensitivity	Average	0.67	0.60	0.64	0.67	0.85
		± 0.26	± 0.28	± 0.18	± 0.17	± 0.14	
Session 2	Model-Independent	Average	0.68	0.67	0.67	0.69	0.94
			± 0.21	± 0.25	± 0.21	± 0.17	± 0.06
	K-S test	0	1	1	1	1*	
	RMLP Sensitivity	Average	0.69	0.74	0.68	0.75±	0.93
		± 0.23	± 0.20	± 0.24	± 0.14	± 0.05	
	K-S test	1*	1	1	1	1*	

* Significantly better in subset

Table 7-3 shows the average and best CC for the models trained with the 10 most important cells picked by each of the criteria. Even with a gross subsampling (10 cells out of 185) of the ensemble, these models produced correlation coefficients close to our best performing models which were trained on the full ensemble. This observation provides

confidence in the methodologies for selecting cellular importance. Statistically, a significant decrease in performance was observed in several of the kinematic variable predictions when the Kolmogorov-Smirnov (K-S) significance test ($p = 0.05$) was applied to 40 sampled windowed correlations between the ensemble and subsets (Table 3, 1 means significantly different).

Implications of Sensitivity Analysis for Model Generalization

In our sensitivity analyses, 10 cells were chosen to match the model-independent space-spanning cell ranking. However, the optimal number of cells required for the maximum model performance remains unknown. Previously, Wessberg et al. have shown through neuron dropping analysis that increasing the size of the neural ensemble as the inputs to a FIR filter will lead to the best HP test set CC values [19]. In this paradigm, the FIR filter was initially trained with the entire neural population and test set CC values were computed. A single neuron was then *randomly* removed from the population, the model was retrained, and the new CC values were computed. This process was then repeated until a single cell remained. We have reproduced this analysis for HP using the data from Session 2 of our neural/behavioral recordings and the corresponding curves for both the x and y coordinates are shown in red (ND) in Fig. 7-10²¹.

The neuron dropping analysis was repeated but now removing the neuron that is *least sensitive* according to the vector WF sensitivity. The CC in the test set obtained with the modified neuron dropping analysis in blue (SA) for the x and y coordinates is presented in Fig. 10. In this figure, one can see that using the ten most sensitive cells high CC values are quickly achieved. As more cells are added, the performance increases

²¹ Neuron dropping curves obtained are the average CC values from 30 Monte Carlo simulations. This method is consistent with that used by Wessberg, et al.

further, peaking above the performance of the full ensemble once 41 cells have been added as inputs. Beyond 41 cells, a *decrease in model performance* especially in the x-coordinate can be observed. The optimal number of cells (41) was determined by taking the maximum of the average CC over the two coordinate directions and yielded CC of 0.70/0.77 for x/y HP and 0.72/0.78 for x/y HV.

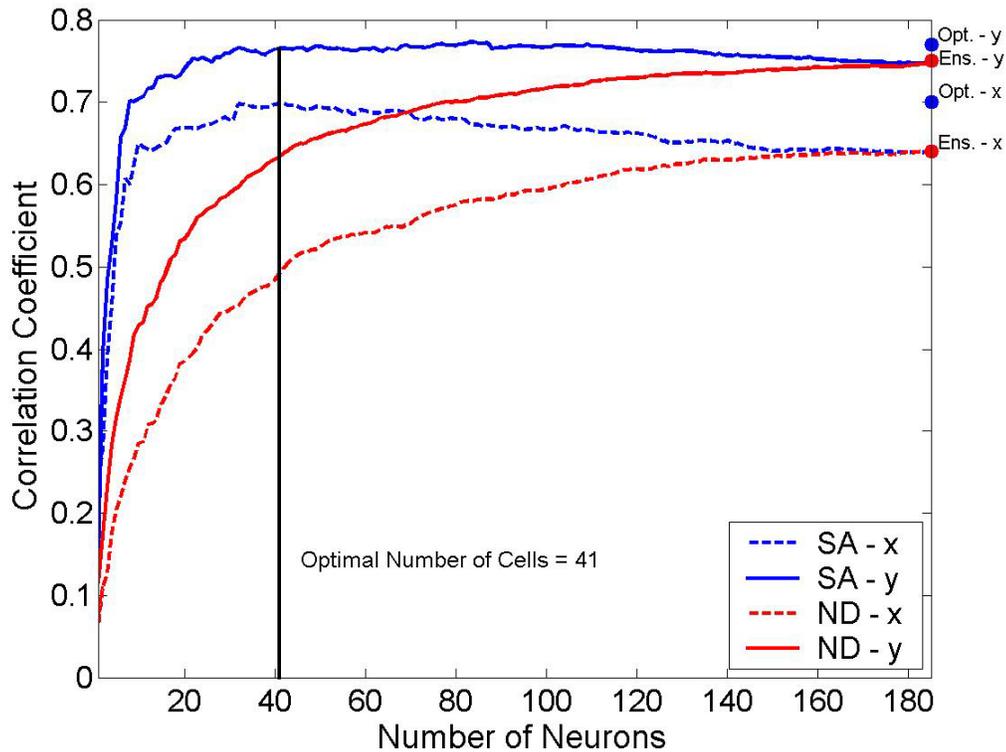


Figure 7-10. Model performance as a function of the number of cells utilized. Cells were removed from the analysis either by randomly dropping cells (neuron dropping – ND) or by removing the least sensitive cell in an ordered list (computed from a sensitivity analysis - SA)

Referring back to all of the sensitivity curves presented thus far, the threshold of peak performance for the ranked list of cells appears where the slope of the sensitivity curves change. The curves in blue resemble the generalization curves reflecting the bias variance dilemma of model fitting [57]. Using this sensitivity based neuron dropping analysis, a similar behavior is obtained by pruning the input from cells that were less

important. Using the 41 most sensitive cells the FIR filter obtained a 6% and 5% increase in CC on the test set for HP and HV, w.r.t. the FIR filter with 185 neurons. By repeating this analysis technique using rankings from tuning depth, we have observed differences in the number of cells needed to achieve the maximum average correlation (Tuning Depth max avg CC=0.74 for 81 cells). Once again, the ranking by sensitivity finds a smaller cell subset that generalizes well.

Discussion

We have chosen to ascertain the importance of neurons in BMIs for two reasons:

- To build better engineered BMIs systems
- To better understand neural activity

A reduction in the number of free parameters without affecting performance leads directly to BMI systems that require less power, bandwidth, and computational demands. Solving these challenges will bring us one step closer to real, portable BMIs. Analysis of the most important neurons in both the model-independent and model-dependent methods has opened up directions to better understand how neural activity can be effectively used for BMI design. Having now evaluated and verified the three different methods, the questions posed in the introduction will now be addressed.

Can our methods automatically indicate important cells for the prediction of the kinematic variables in the tasks studied?

Since the work of Wessberg *et al*, it is known that the performance of linear models for BMIs improves as more neurons are recorded, but that the performance improvement does not increase linearly with the number of neurons. Motivated by their work, this analysis takes a closer look at the neuron importance for modeling. For the sample set of 104 and 185 neurons collected from multiple cortical areas of a owl monkey and rhesus

macaque, we have shown using two techniques that these methods can automatically choose important cells for reconstructing the kinematic parameters of HP, HV, and GF. The similarity in the shape of the ranking curves regardless of the methodology leads us to believe that there may be an underlying principle governing the kinematic contribution of randomly selected cells in the motor cortex. To quantify the shape similarity, we fit two exponentials to the ranking curves and found that all three method's (for the HP, HV, and GF) ranking curves can be explained ($R^2 = 0.99$) by a function of the form $y = Ae^{Bx} + Ce^{Dx}$ where the values of A, B, C, and D ranged respectively from 0.3–0.6, -0.1–(-0.2), 0.5–0.8, and -0.01–(-0.02).

RMLP models trained with only the 10 most important cells were able to achieve high performance compared the full model (see Fig. 7-4). In Table 7-3 it was verified that BMIs performance is statistically inferior to the full ensemble for some of the kinematic variables and sessions. Both the figure and table shows that BMI performance with such a large reduction in the number of inputs is inferior to the full ensemble, but from an implementation point of view, this slight reduction in performance is coupled with huge savings in number of training parameters that simplifies the real-time implementations.

One must remember that this reduced number of cells makes BMIs more sensitive to the instability of neuronal firings over time. Recent studies are showing that the activity of individual neurons and cortical areas used in BMI experiments can vary considerably from day to day [32], therefore the variance over time of the importance of neurons must be quantified in future studies. Moreover, this neuron selection requires long records, and it is done off-line from trained models or tuning curves. It is therefore not clear how practical the neuron selection techniques will be to the surgery stage. For

these reasons, we advocate the use of a higher sampling of the cortical activity to help improve this ratio until other models are proposed that take advantage of the information potentially contained in the spike trains and not exploited by linear models.

In this model-based framework, can better BMIs be built using a subset of important cells?

This question is rooted in our goal to build models for BMIs that generalize optimally. The problem is the well-known bias-variance dilemma of machine learning [57], and it is related to the number of free parameters of a model. The MIMO structure of BMIs, with thousands of free parameters, makes this problem particularly critical, and excludes the traditional Akaike or BIC criteria [58]. The previous sensitivity analysis is an attractive alternative because it ranks the importance of neurons; however, it does not give any indication on where to start excluding neurons for better generalization. Inspired by the neuron dropping methodology, but excluding the least sensitive of the neurons instead of a random one, we were able to find a set of 41 important neurons that performs better than the full set of neurons (Figure 7-10). It is very interesting that such a small change in the model dropping methodology can provide an answer to this important question for BMI design. However, it should be noted that the other ranking methodology found different sets of neurons for best generalization. Therefore, it remains to be seen if alternative methods (possibly combining regularization theory) are able to find smaller combinations of neurons that improve performance even further. At this point of the research, we suggest the use of sensitivity analysis as an indicator of neurophysiologic cellular importance combined with a neuron dropping scheme for each modeling task so that an estimate of the appropriate model order can be obtained.

Is the model-independent technique a good indicator of cellular importance, and how is it related to sensitivity through the linear model?

The previous two questions were successfully answered by sensitivity analysis through the model, but it should not be forgotten that model-based sensitivity is biased by the type of model chosen, its performance level and by noise in the data. Therefore it is important to pursue a model independent approach to establish the importance of neurons. We hypothesized from a combined neurophysiologic and modeling point of view that highly modulated neurons spanning the space of the kinematic parameter of interest should be chosen. Intuitively these constraints make sense for the following reasons:

- If a cell is modulated to the kinematic parameter, the adaptive filter will be able to correlate this activity with the behavior through training. Otherwise, neural firing works as a broadband excitation that is not necessarily related to better performance.
- If a group of cells are highly modulated to only a part of the space, the adaptive filter may not be able to reconstruct data points in other parts of the space.

After choosing cells based upon these considerations, the hypothesis was validated by achieving performance levels that are akin to using the model-based techniques of choosing cells (see Table 7-3). Therefore, we conclude that the two methods pick many common cells as their top 10, and this is what matters for model performance, since the RMLP is not sensitive to the ordering of the cells. However, as can be seen in Table 7-1 (and Fig 7-9) the cells picked by the three methods do not exactly coincide. We are in fact surprised that many cells coincide when recalling the different assumptions behind each method: sensitivity is computed by averaging cell firings over the training interval, while tuning relates to instantaneous cell firings. We can imagine many cases where a cell is important for a given metric but not important in another. Therefore, this

coincidence in cell importance is either a feature of the spatio-temporal organization of the motor cortex, or is a byproduct of the model and correlation coefficients to implement and assess performance in the BMI model. Further research is needed to fully understand this important issue.

A final comment goes to the overall performance of the BMI system built from adaptive models. Although it is impressive that an optimum nonlinear system is able to identify the complex relations between spike trains in the motor cortex and hand movements/gripping force, a correlation coefficient of ~ 0.8 may not be sufficient for real world applications of BMIs. Therefore, further research is necessary to understand what is limiting the performance of this class of adaptive linear and nonlinear systems.

Another issue relates to the unrealistic assumption of stationarity in neural firings over two sessions that is used to derive results presented in this study about neuron sensitivity and tuning curves. In future studies, it will be necessary to assess the time variability of the neuronal rankings and determine its effect on model generalization.

CHAPTER 8 CONCLUSIONS AND FUTURE DIRECTIONS

Our introduction and journey into the study of brain-machine interfaces began as a “mystical” proposal. We were faced with spike trains and adaptive systems that could “magically” find a functional relationship between neuronal activity and behavior. Adding to the lack of understanding was a wealth of literature referring to neuronal activity as a type of “secret code” that researchers often try to crack. While these mystical, magical, and secret perceptions of BMIs are not very scientific they do contribute to the allure of BMIs. This allure has contributed to the mass media (radio, magazines, television, scientific journals, and fictional works) coverage of BMI research [76-79]. The fundamental reality behind the allure is that we do not fully understand the functioning of the brain and in this case of the motor system. On a much more grand scale, we are motivated to pursue this line of research because we have potential to learn a little more about ourselves and how we come to use our goal directed behaviors. Often, the results of our intent are a product of locomotion which is a function that defines our livelihood. This ability is often taken for granted since most of us don’t even think about how we move through the day. Only when this functionality is taken away from us, as in the case of paraplegics/quadruplegics, do we fully come to appreciate the importance of the freedom to move. The prospect of the ability to give this functionality back to an individual has been a motivation for many researchers to develop BMIs. Scientifically, the idea of being able to engineer a man-made system that interprets an individual’s intentions and expresses them as machine commands is a great feat. Again though, we are

faced with blurring the line between man and machine by developing interfaces that will allow man to express its will in new and more powerful ways.

In this dissertation, we addressed some of the fundamental engineering challenges involved with making BMIs a reality. Starting from a signal processing modeling approach, it was shown that there are several “off the shelf” models that are capable of finding the neural to motor mapping with good results (average CC of ~ 0.8 depending upon the animal/task). For the datasets studied here, the difference in performance between linear and nonlinear models was not great. Reasons for this result could include the limitations of the types/simplicity of the tasks, time averaging of global models, or, the coarse binning preprocessing of the data. Nevertheless, nonlinear models are attractive for BMI modeling because they can bring smoothness to the trajectories by saturating nonlinearities, reduce the number free parameters with feedback in hidden layers, and create a diversity of kinematic outputs by using multiple PEs in the hidden layer.

While many of the models studied in this dissertation are simple function approximators or generative models, we found the RMLP neural network to have biologic plausibility. If generation of neuronal activity is thought of as the precursor to muscle activation, a simple linear relationship between neuronal activity and the kinematic variables position, velocity, and acceleration can be constructed (see Todorov). Using the RMLP equations, we showed that the RMLP is the nonlinear extension to this hypothesis. Moreover the use of feedback in the hidden state of the topology allows us to reconstruct hidden variables of HV, HA, or even GF. Compared to the FIR topology the RMLP has the powerful ability to simultaneously reconstruct multiple kinematic

parameters using the interplay of the recurrent PEs in the hidden layer and the multiple PEs in the output layer. To further test the universal approximating ability and broad expressive power of the trained RMLP topology, we tested generalization on several movement tasks over the span of several days of recording. We have shown that it is relatively simple to train (linear or nonlinear) topologies when the number of degrees of freedom is large. However when the dimensionality of the models are greatly reduced, linear topologies exhibit a reduction in performance while the nonlinear RMLP maintained its performance.

The RMLP implemented here was also able to quantify cortical neuronal interactions. Moreover, we provide a method to extract information about motor neuronal interactions from the trained RMLP. The analysis began by tracing the signals through the topology. By showing that the RMLP is not an “incomprehensible black box” we have determined that the RMLP maps neural activity to hand position using the mesoscopic activity of the neural ensemble and a limited set of preferred directions that span the space of the hand trajectory; a result that adds to its biologic plausibility. Additionally, we found that the BPTT training procedure employed on the high dimensional neuronal input set the model weights in the null space of the input; a result not unlike anti-Hebbian learning. The second part of the analysis showed that with the trained model we can specify which cortical regions are involved with the production of the motor task as well as use a model based sensitivity analysis to construct an ordered list of neurons most related to a reaching task. This simple signal processing approach of analyzing neuronal recordings yielded constructs similar to those found in the labor intensive approaches found in neurophysiology laboratories. Being able to identify and

understand the importance of cells in the ensemble additionally allowed us to improve the performance of the BMI models.

Summary of Contributions

Here a brief list is presented which summarizes of the contributions found in this dissertation:

- We have shown that recurrent neural networks are not incomprehensible black boxes by providing an explanation of the neural to motor mapping.
- Proposed new performance metrics for BMIs
- Developed a methodology to ascertain the importance of cells in the neuronal ensemble and showed how to it can be used to improve BMI model performance.
- Quantified model generalization for multi-task and multi-session datasets.
- Developed a methodology to quantify cortical contributions to motor tasks. Analyzed and compared the contributions to those found in the functional neurophysiology literature.

Difficulties Encountered in BMI Research

In the BMI modeling studies presented in this dissertation, several difficulties were encountered that could be affecting model performance and analysis. While these difficulties have been identified throughout this work, many of the topics are the subject of future work.

In the first stage of any modeling experiment presented here, the neuronal data was preprocessed using a binning procedure. Binning is a simple procedure that produces time-to-amplitude conversions in the data that correlate well with the desired trajectory. While the choice of bin size in the literature can range from 25ms to 300 ms, it is a parameter that has yet to be optimized. Moreover, since our most basic models are simple correlators, the bin size should be set so that it is easy for them to detect changes in neuronal amplitude that are synchronized with behavior. Additionally, the bin size

selected for an experiment may affect performance because it could mask nonlinearities important for performance due to the temporal averaging.

Related to the temporal structure is the nonstationary nature of the data. In all of our studies, we have assumed that the contributions of cells in the ensemble are fixed over time windows of minutes to days. While this assumption may seem strong, the models continued to produce high testing correlations over long periods of time. However depending upon the attention of the animal, type of trajectory, and the learning involved with using a BMI the statistics of the firing patterns of the sampled cells may change. A long-term BMI study is required that principally quantifies neuronal firing patterns using generative models.

We were somewhat surprised that we did not see any large difference in performance between linear and nonlinear models. Of course nonlinear models have many desirable characteristics needed for BMI design (trajectory smoothness, parsimonious architectures, and broad expressive power). A possibility for the similarity in performance may stem from the fact that we may not have enough diversity in the data that requires a nonlinear mapping. As BMI research continues, we should seek out more diverse behavioral paradigms that try to address this question.

In terms of regularization, we have attempted to select important neurons from the ensemble by studying information contained in the trained model. It would be beneficial to BMI design to develop a principled regularization procedure that works on the input data alone. At this point in our understanding of BMI design we should concentrate on outliers in the distribution of firing patterns produced by the cells.

One of the easiest explanations of why BMI model performance is limited to CC values of 0.8 is that the sampling of the motor areas may be too coarse to achieve better performance. This hypothesis is reasonable since we are only sampling hundreds of neurons from cortices that contain millions of networked cells. In this case, the problem is not one of algorithms and requires collaboration with neurophysiologists so at the surgical stage more cells whose activity is related to behavior can be sampled.

At this juncture, one of the most troubling aspects of BMI research is the difficulty in training models when the subject is a paraplegic. Traditionally we use supervised learning techniques to train the models using neuronal activity and the known physical desired kinematic variables. In the case of a paralyzed individual, we do not have access the desired movements; we only have access to imagined trajectories. How this reality will affect current BMI modeling techniques remains unknown. However, for BMI design to be useful for disabled individual, this training difficulty must be overcome.

Future Directions

By definition, a Biomedical Engineer with a specialization in Neural Engineering requires both experience in signal processing/modeling as well as experience in the techniques of neural recording. While my education at the Computational NeuroEngineering Laboratory has provided the essential foundations and tools necessary to analyze multi-dimensional time-series, it has relied heavily upon other researchers to provide the neurophysiological recordings necessary studying BMI modeling. My desire to answer more of the unknown functional neurophysiological aspects of BMIs will require that I am involved in the surgical, behavioral, experimental paradigm design of BMI studies. Without this knowledge and experience, I am only half of the Biomedical Engineer that I could be and the types of problems addressed in my studies are limited by

the datasets provided by other researchers. It is my goal to continue applying signal processing analysis methodologies to new behavioral and neuronal recordings in the setting of an electrophysiology laboratory.

Final Thoughts

Brain-Machine Interface design is an ambitious direction of research. The unknown aspects of neurophysiology and goal-directed behavior combined with the engineering challenges of bringing the technology to reality make the research extremely difficult. Only with the cooperation of researchers from many disciplines will the final product be a robust, efficient, and practical system. While the prospects of being the first to design a fully functional BMI for humans is extremely lucrative, it can bring a sense of destructive competition among researchers that actually hinders progress. Additionally, with each new technological and media development, the promises and expectations for BMIs can become inflated. By setting such a high bar of performance, the possibility of being able to deliver the promises becomes less probable. While my desires may seem idealized, in the future I hope that I can participate in *constructive* competition with other BMI researchers. As a BMI research community, the exchange of ideas can spawn new ambitions that drive researchers to strive for goals based in the reality of technological limitations. In this setting, I believe it is possible to protect intellectual property while maintaining an open relationship with other researchers so that ideas from many aspects of BMI research can feed off each other in the pursuit of advancing to new milestones, new interfaces, and new lifestyles for ourselves and for the community.

APPENDIX A TRAINING THE RMLP

Introduction

The performance of RMLP input-output models in BMI experiments is dependent upon the choice of the network topology, learning rules, and initial conditions. Control of these parameters involves choosing an appropriate number of processing elements, learning rates, and the training stopping point. The seemingly endless number of possible network parameter combinations is practically difficult to deal with and often casts a shadow upon the power of these networks. Up to this point, there is no principled way to simultaneously adjust all of the available network settings to promote the best optimization of the network weights. Therefore, a heuristic exploration of these settings is necessary to evaluate the performance.

Topology and Trajectory Learning

The first consideration in any neural network implementation is the choice of the number of processing elements. Since, the RMLP topology studied here always consisted of only a single hidden layer of tanh PEs the design question becomes one of how many PEs are required to solve the neural to motor mapping. The first approach to this problem was a brute force scan of the performance across a range of PEs. as shown in Table A-1. It can be seen for a reaching task that the topology with 5 PEs produced the highest testing correlation coefficients. While the brute force approach can immediately direct the choice of topology it does not explain why the number is appropriate for a given problem. We later came to find out (in Chapter 5, Fig. 5-8) that the number of hidden PEs should

be chosen to span the space of the desired trajectory. In the context of BMIs, this knowledge can help to avoid the computational and time costs of the brute force approach.

Table A-1. RMLP performance as a function of the number of hidden PEs

	1 PE	5 PE	10 PE	20 PE
Average Testing Correlation Coefficient	0.7099	0.7510	0.7316	0.6310

The RMLP presented in this dissertation was trained with backpropagation through time (BPTT) [48, 80] using the NeuroSolutions software package [55]. Training was stopped using the method of cross-validation (batch size of 1000 pts.) (Fig. A-1) to maximize the generalization of the network [56]. The BPTT training procedure involves unfolding the recurrent network into an equivalent feedforward topology over a fixed interval, or trajectory. For our BMI applications, we are interested in learning the dynamics of the trajectories of hand movement; therefore, for each task, the trajectory was chosen to match a complete movement. For the Belle's reaching task, this length was on average 30 samples. In Table A-2 this selection of trajectory was compared again with a brute force scanning of the testing performance as a function of the trajectory length (samples per exemplar). Included in the BPTT algorithm is the option to update the weights in online (every trajectory), semi-batch, or batch mode. Choosing to update the network weights in semi-batch mode can protect against noisy stochastic gradients that can cause the network to become unstable by averaging the gradients over several trajectories. Table A-2 also provides the average testing correlation coefficient as a function of the frequency of updates. We can see that a range of good choices (15-30 s/e and 5-15 e/u) exist for the trajectory length and update rule.

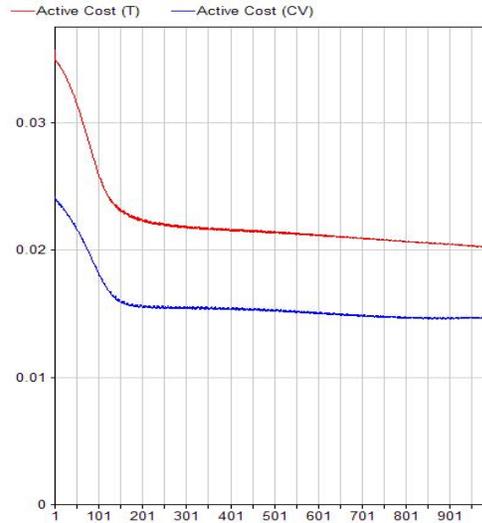


Figure A-1. RMLP learning curve. MSE (upper curve) Crossvalidation (lower curve)

Table A-2. Average testing correlation coefficients as a function of trajectory length

	5 samples/exemplar	15 s/e	30 s/e	60 s/e
1 exemplar/update	0.6514	0.7387	0.7302	0.7135
5 e/u	0.6458	0.7006	0.7263	0.7091
15 e/u	0.6544	0.7389	0.6680	0.6855
30 e/u	0.6807	0.7482	0.7116	0.6781
60 e/u	0.6457	0.6383	0.6772	0.6624

Monte Carlo Simulations

To test the effect of the initial condition on model performance, one hundred Monte Carlo simulations with different initial conditions were conducted with 20,010 consecutive bins (2,001 secs) of neuronal data to improve the chances of obtaining the global optimum. In Fig. A-2, the training MSE is presented for all simulations and it can be seen that all initial conditions reach the approximately the same solution. The greatest effect of the initial condition can be seen in the time it takes for each model to converge. In Fig. A-3, the average and standard deviations of the curves in Fig. A-2 is presented. This figure is characterized by a large standard deviation in the initial epochs resulting from the initial condition and a small standard deviation in the final epochs signifying

that the models are converging to the same solution. These relationships are quantified in Table A-3 where the average final MSE had a value of 0.0203 ± 0.0009 . Again, a small training standard deviation indicates the networks repeatedly achieved the same level of performance. Of all the Monte Carlo simulations, the network with the smallest error achieved a MSE of 0.0186 (Table A-3).

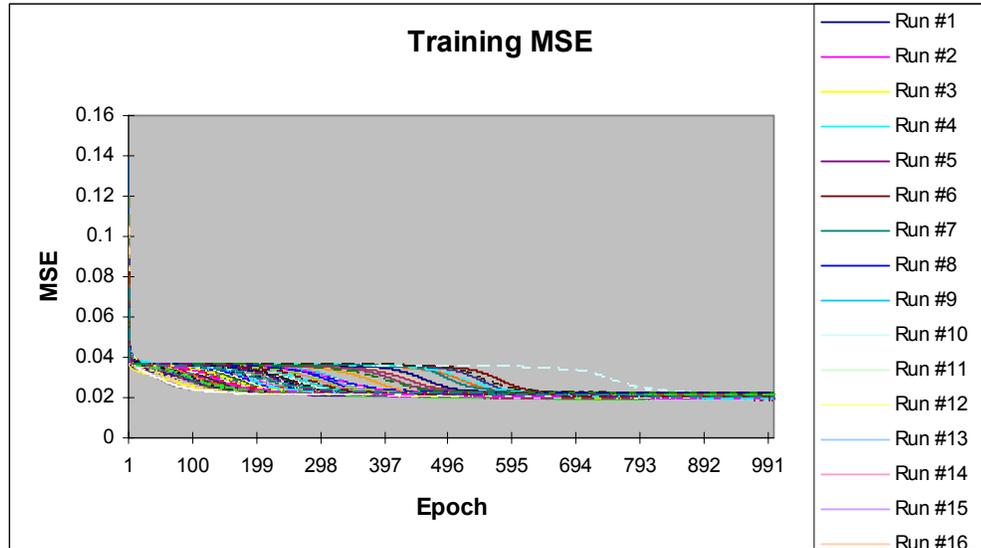


Figure A-2. Training MSE curves for 100 Monte Carlo simulations

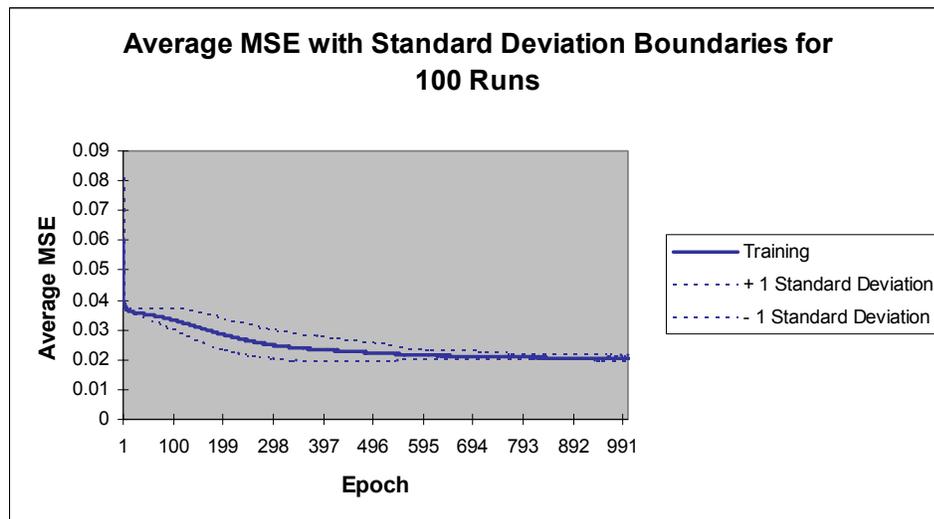


Figure A-3. Standard deviation in training MSE for 100 Monte Carlo simulations

Table A-3. Training performance for 100 Monte Carlo simulations

All Runs	Training Minimum	Training Standard Deviation
Average of Minimum MSEs	0.020328853	0.000923483
Average of Final MSEs	0.020340851	0.000920456

Table A-4. Best performing network

Best Network	Training
Run #	85
Epoch #	999
Minimum MSE	0.018601749
Final MSE	0.018616276

APPENDIX B

EVALUATION OF THE RMLP MODELING PERFORMANCE USING SPIKE-SORTED AND NON-SPIKE-SORTED NEURONAL FIRING PATTERNS

Introduction

Early electrophysiological techniques proposed by Adrian for investigating brain function were focused on single neurons. As researchers sought to understand how perception and behavior is encoded, the view of single neurons became interactive and involved time-dependent communication among many neurons. Methods to extract the firing information of large ensembles of cortical neurons using many electrodes have been proposed. These spike sorting (SS) methods are computationally intensive and involve human discretion [81]. From an adaptive signal processing point of view, separating individual neurons may not provide a significant advantage since input-output (I/O) models compute weighted sums of the neuronal data. Spike-sorting separates individual neurons per electrode while I/O models collapse the data. In this report we will evaluate the utility of spike sorting by training and testing a RMLP using spike-sorted and non-spike-sorted (NSS) neuronal firing patterns.

Data Preparation

Spike-sorting techniques yielded neuronal spike firing times from 104 neurons collected using 64 microwire electrodes. The neuronal firings were binned (added) in non-overlapping windows of 100 ms, which represents the local firing rate for a neuron. To reconstruct the NSS neuronal firing patterns, a sum of the firing rates across the neurons associated with each of the 64 electrodes was computed. For example, if

electrode 1 has neuron A and B associated with it, the NSS sorted firing rate would be the sum of the firing rates for neurons A and B at each timebin. The number of inputs is collapsed from 104 neurons to 64 electrodes. For the particular dataset used in this report, not all electrodes yielded spike trains and the data were collapsed further to 51 inputs. These spike counts from the SS and NSS data were directly used as inputs to the RMLP.

Simulations

The spike counts of each of the 104 neurons and 51 electrodes were separately used to train a RNN (104,51x5x3) with 5 nonlinear hidden PEs and 3 linear output PEs to predict the X, Y, and Z coordinates of the monkey's hand. Each exemplar of data used for the BPTT training algorithm contained a trajectory length of thirty samples (3 secs). Weight updates occurred after the presentation of 10 exemplars (30 secs). A training set of 20,010 consecutive bins (2,001 secs) of data was utilized. In testing, the network parameters were fixed and 3,000 consecutive bins (300 secs) of novel neuronal data were fed in the network to predict new hand trajectories.

To test the trained network, SER and CC values were used as metrics. The SER computed using a sliding window of 40 samples (4 seconds) for the SS and NSS RNN is shown in Fig. B-1. Windowed SER calculations, averaged over the three coordinate directions for SS and NSS, reached a value of 31.98 and 30.30 respectively. The cumulative SER averaged over all coordinate directions and the entire test set (3000 samples, 300 seconds) for SS and NSS was 1.58 and 1.52 respectively. The correlation coefficient between the desired and network output trajectories were also computed for the entire testing dataset. The SS and NSS produced correlation values 0.6857 and 0.7477 respectively. These values are presented in Table B-1. No time-dependent decay of the SNR was observed.

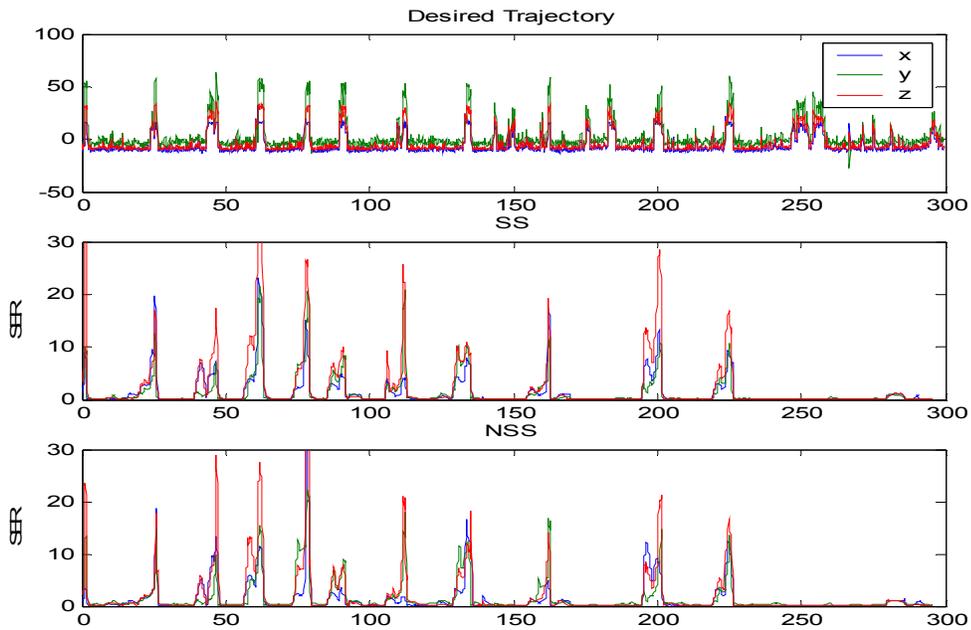


Figure B-1. Signal to error ratio between the actual and estimated hand coordinates for SS and NSS data

Table B-1. Testing SER and CC values

	Spike-Sorted	Non-Spike-Sorted
Max SER	31.98	30.30
Mean SER	1.58	1.52
Correlation Coef.	0.6857	0.7477

The peaks of the estimated hand trajectory superimposed on the actual trajectory are shown in Fig. B-2. The SS RNN estimation of the peak values does not show a significant improvement compared to the NSS RNN. The target accuracy for both datasets is further compared in Fig. B-3 which shows the target estimation errors for six peaks. In the figure, the target hand position is represented by an x located at the origin of the coordinate system. The mean of the absolute value of the error for ten points at each peak was computed. This mean error associated with each direction (x, y, z) is plotted on its respective axis. The SS and NSS estimation errors both form a tight cluster of points

around the target. The estimation errors for the NSS are slightly larger than the SS but no significant degradation in performance is observed.

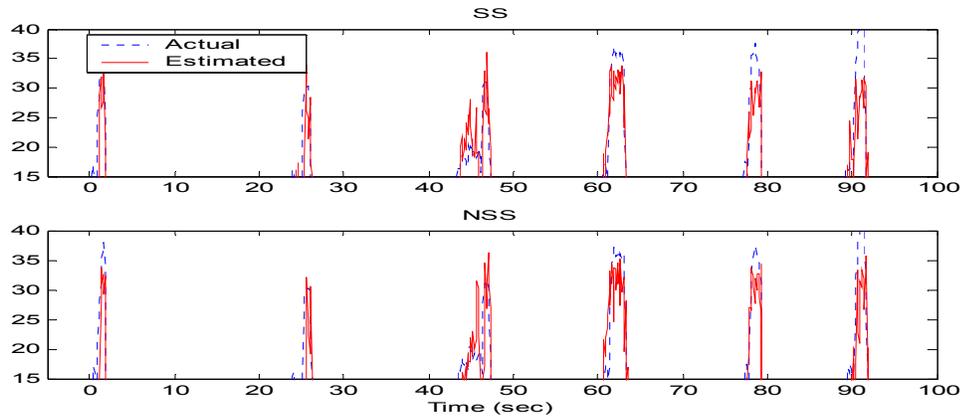


Figure B-2. Peaks of hand trajectory (Z-coordinate)

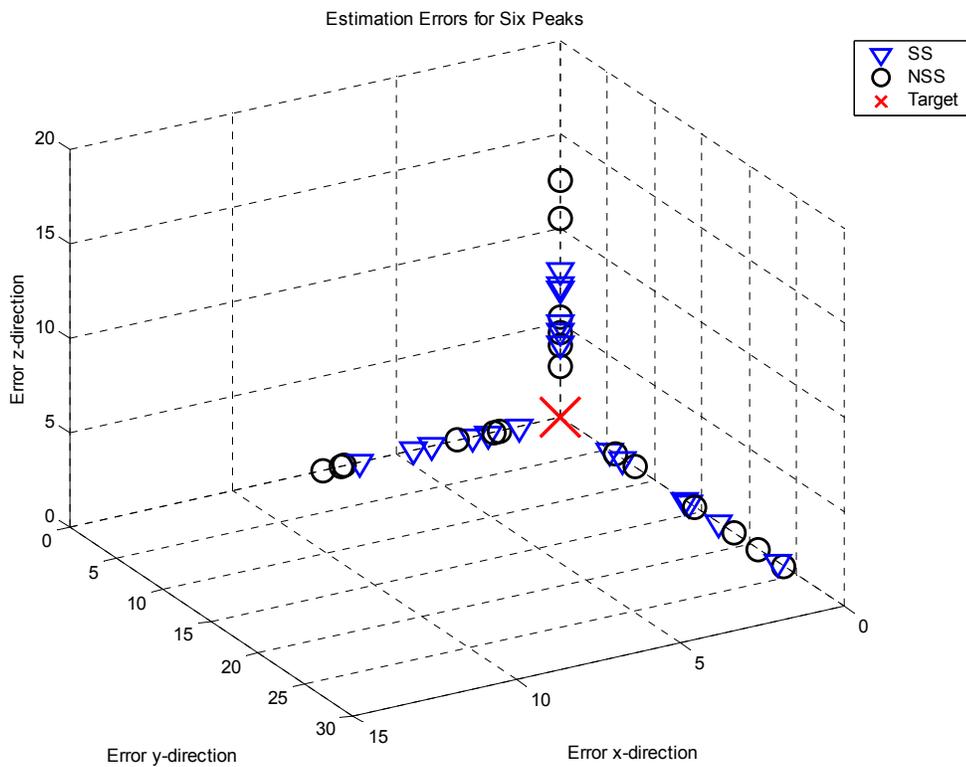


Figure B-3. Estimation errors for six peaks. Targets are represented by an x at the origins. The average error (mm) in each direction is displayed on the respective axis

Discussion and Conclusion

Current methods for estimating hand position from neuronal firing patterns involve decomposing the data to individual neurons. Neuronal spike-sorting is a computationally intensive procedure which involves human discretion and error. With the current approach, if brain-machine interfaces are to be feasible the spike sorting procedure would have to be made portable. The hardware requirements for this procedure in the laboratory involve a personal computer and a bank of DSPs which implement a modified version of principal component analysis. Included in the procedure is the matching of a pair of time-voltage boxes, defined by the experimenter, to isolate the analog waveforms which belong to an individual neuron[81]. Solving the portability and human involvement issues is a daunting task.

By sampling the firing information from a neighborhood of neurons we are still able to capture an equivalent amount of information while gaining two advantages. First, the reduction in the number of inputs reduces the number of free parameters in the network and can improve generalization. Second, the reduction of the dimensionality drastically decreases network training time.

The results presented in this analysis indicate that for the task of predicting hand trajectories from neuronal firing patterns spike-sorting is not a requirement. Removing this procedure simplifies the BMI architecture and increases the feasibility of the project.

APPENDIX C MODEL EXCITATION WITH RANDOM INPUTS

It is well known in the neural network literature that training a model with a sufficient number of degrees of freedom can track *any* trajectory when it is excited with random inputs. Naturally in BMI experiments models can have thousands of free parameters due to the dimensionality of the neuronal ensemble input and the use of tap-delay memory structures in the input. Additionally we are attempting to find the functional relationship between this activity and relatively simple trajectories that resemble sine waves. In the initial phases of our BMI modeling studies it was feared that these large models were not learning the neural to motor functional mapping.

Here we present Monte Carlo simulations of RMLP models trained with both random and neuronal inputs. The 104 neuronal inputs used in this experiment were generated from Belle who was performing the reaching task. Random inputs of dimensionality 104 were generated as follows:

- Find maximum spike rate for entire data set of real neuronal activity
- Compute PMF using histc over the range in the previous step
- Normalize the PMF by the number of timebins
- Create the CDF by summing the PMF
- Create a random vector in the range [0:1]
- Assign a firing rate determined by the CDF to each random value
- Compare the PMF of the original data to the PMF of the random data (they should be the same)
- Compare the correlation of the original data to the random data (they should be uncorrelated)

In Table C-1, we can see that the training MSE of the random input models is roughly two times that of the models trained with neuronal inputs. The final training

MSE may lead us to believe that the random input network will perform only slightly worse compared to the real network. Once the ten RMLP models were trained, the weights were fixed and 500 test samples. The testing CC values display a very different result; the testing CC values of the models with random inputs were close to zero while the real input networks produced CCs close to 0.8. In Fig. C-1, we can see that there are two distinct clusters of performance; the models trained with real neuronal recordings consistently produced lower training MSEs and higher testing correlation coefficients.

This analysis shows that indeed the RMLPs with real neuronal inputs are finding a functional mapping between the neuronal activity and behavior. Moreover, destruction of the intricate timing relations in the firing patterns will produce train model outputs that do not correlate with the desired signal.

Table C-1. Model performance using random vs. real neuronal activity

	Test Correlation X,Y,Z	Training MSE
Random 1	0.0175, 0.0440, -0.0341	0.082049
Random 2	0.0200, -0.0069, 0.0138	0.082730
Random 3	0.0888, -0.0176, 0.0237	0.082779
Random 4	0.0322, 0.0421, -0.0397	0.082576
Random 5	0.0283, 0.0066, 0.0170	0.082187
Neuronal 1	0.7010, 0.7861, 0.8207	0.041197
Neuronal 2	0.6460, 0.7762, 0.6947	0.043731
Neuronal 3	0.6853, 0.7374, 0.7537	0.044192
Neuronal 4	0.6624, 0.7416, 0.7042	0.044259
Neuronal 5	0.6691, 0.7732, 0.7055	0.044346

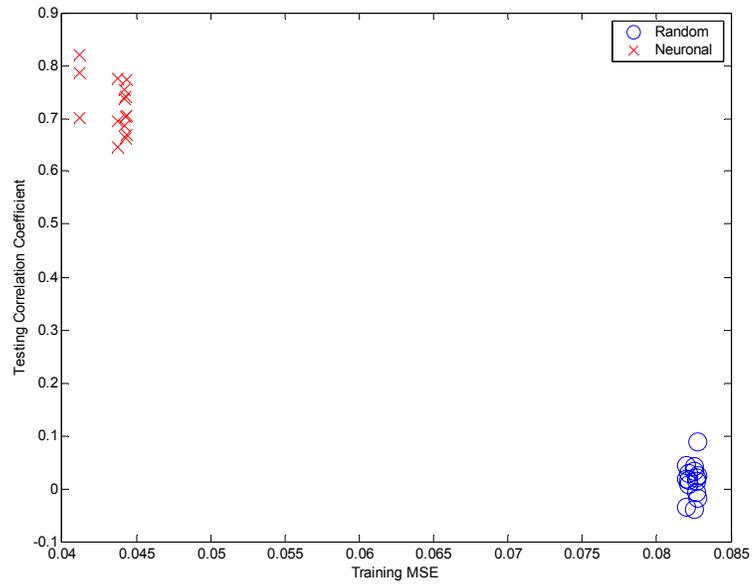


Figure C-1. Performance measures for five Monte-Carlo simulations using neuronal and random data

LIST OF REFERENCES

- [1] E. M. Schmidt, "Single neuron recording from motor cortex as a possible source of signals for control of external devices," *Ann. Biomed. Eng.*, pp. 339-349, 1980.
- [2] R. A. Andersen, L. H. Snyder, D. C. Bradley, and J. Xing, "Multimodal representation of space in the posterior parietal cortex and its use in planning movements," *Annu. Rev. Neurosci.*, vol. 20, pp. 303-330, 1997.
- [3] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neuroscience*, vol. 2, pp. 664-670, 1999.
- [4] Y. Gao, M. J. Black, E. Bienenstock, W. Wu, and J. P. Donoghue, "A quantitative comparison of linear and non-linear models of motor cortical activity for the encoding and decoding of arm motions," presented at The 1st International IEEE EMBS Conference on Neural Engineering, Capri, Italy, 2003.
- [5] A. Georgopoulos, J. Kalaska, R. Caminiti, and J. Massey, "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex.," *Journal of Neuroscience*, vol. 2, pp. 1527-1537, 1982.
- [6] A. P. Georgopoulos, J. T. Lurito, M. Petrides, A. B. Schwartz, and J. T. Massey, "Mental rotation of the neuronal population vector," *Science*, vol. 243, pp. 234-236, 1989.
- [7] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, pp. 1416-1419, 1986.
- [8] S. P. Kim, J. C. Sanchez, D. Erdogmus, Y. N. Rao, J. C. Principe, and M. Nicolelis, "Modeling the relation from motor cortical neuronal firing to hand movements using competitive linear filters and a MLP," presented at International Joint Conference on Neural Networks, Portland, Oregon, 2003.
- [9] S. P. Kim, J. C. Sanchez, D. Erdogmus, Y. N. Rao, J. C. Principe, and M. A. Nicolelis, "Divide-and-conquer approach for brain machine interfaces: nonlinear mixture of competitive linear models," *Neural Networks*, vol. 16, pp. 865-871, 2003.

- [10] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching," *Journal of Neurophysiology*, vol. 82, pp. 2676-2692, 1999.
- [11] D. W. Moran and A. B. Schwartz, "Motor cortical activity during drawing movements: population representation during spiral tracing," *Journal of Neurophysiology*, vol. 82, pp. 2693-2704, 1999.
- [12] J. C. Sanchez, D. Erdogmus, J. C. Principe, J. Wessberg, and M. Nicolelis, "A comparison between nonlinear mappings and linear state estimation to model the relation from motor cortical neuronal firing to hand movements," presented at SAB Workshop on Motor Control in Humans and Robots: on the Interplay of Real Brains and Artificial Devices, University of Edinburgh, Scotland, 2002.
- [13] J. C. Sanchez, D. Erdogmus, Y. Rao, J. C. Principe, M. Nicolelis, and J. Wessberg, "Learning the contributions of the motor, premotor, and posterior parietal cortices for hand trajectory reconstruction in a brain machine interface," presented at IEEE EMBS Neural Engineering Conference, Capri, Italy, 2003.
- [14] J. C. Sanchez, S. P. Kim, D. Erdogmus, Y. N. Rao, J. C. Principe, J. Wessberg, and M. Nicolelis, "Input-output mapping performance of linear and nonlinear models for estimating hand trajectories from cortical neuronal firing patterns," presented at International Work on Neural Networks for Signal Processing, Martigny, Switzerland, 2002.
- [15] A. B. Schwartz, D. M. Taylor, and S. I. H. Tillery, "Extraction algorithms for cortical control of arm prosthetics," *Current Opinion in Neurobiology*, vol. 11, pp. 701-708, 2001.
- [16] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Brain-machine interface: Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141-142, 2002.
- [17] K. V. Shenoy, S. A. Kureshi, D. Meeker, B. L. Gilikin, D. J. Dubowitz, A. P. Batista, C. A. Buneo, S. Cao, J. W. Burdick, and R. A. Anderson, "Toward prosthetic systems controlled by parietal cortex," *Society for Neuroscience*, vol. 25, 1999.
- [18] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, pp. 1829-1832, 2002.
- [19] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and a. Nicolelis et, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361-365, 2000.

- [20] M. A. L. Nicolelis, "Brain-machine interfaces to restore motor function and probe neural circuits," *Nature Reviews Neuroscience*, vol. 4, pp. 417-422, 2003.
- [21] A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. coding of the direction of movement by a neuronal population," *The Journal of Neuroscience: the Official Journal of the Society for Neuroscience*, vol. 8, pp. 2928-2937, 1988.
- [22] S. Lin, J. Si, and A. B. Schwartz, "Self-organization of firing activities in monkey's motor cortex: trajectory computation from spike signals.," *Neural Computation*, vol. 9, pp. 607-621, 1997.
- [23] J. F. Kalaska, S. H. Scott, P. Cisek, and L. E. Sergio, "Cortical control of reaching movements," *Current Opinion in Neurobiology*, vol. 7, pp. 849-859, 1997.
- [24] T. D. Sanger, "Probability density estimation for the interpretation of neural population codes," *J Neurophysiol*, vol. 76, pp. 2790-2793, 1996.
- [25] F. A. Mussa-Ivaldi, "Do neurons in the motor cortex encode movement directions? An alternative hypothesis," *Neuroscience Letters*, vol. 91, pp. 106-111, 1988.
- [26] K. V. Shenoy, D. Meeker, S. Cao, S. A. Kureshi, B. Pesaran, C. A. Buneo, A. P. Batista, P. P. Mitra, J. W. Burdick, and R. A. Andersen, "Neural prosthetic control signals from plan activity," *NeuroReport*, vol. 14, pp. 591-597, 2003.
- [27] S. Haykin, *Adaptive filter theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall International, 1996.
- [28] F. Rieke, *Spikes: Exploring the Neural Code*. Cambridge: MIT Press, 1996.
- [29] M. A. L. Nicolelis, *Methods for Neural Ensemble Recordings*. Boca Raton: CRC Press, 1999.
- [30] C. T. Leonard, *The neuroscience of human movement*. St. Louis: Mosby, 1998.
- [31] D. O. Hebb, *The Organization of Behavior: A neuropsychological Theory*. New York: Wiley, 1949.
- [32] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biology*, vol. 1, pp. 1-16, 2003.

- [33] L. Ljung, "Black-box models from input-output measurements," presented at IEEE Instrumentation and Measurement Technology Conference, Budapest, Hungary, 2001.
- [34] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, "Principles of Neural Science," 4 ed. New York: McGraw-Hill, 2000.
- [35] D. Flament and J. Hore, "Relations of motor cortex neural discharge to kinematics of passive and active elbow movements in the monkey," *Journal of Neurophysiology*, vol. 60, pp. 1268-1284, 1988.
- [36] J. F. Kalaska, D. A. D. Cohen, M. L. Hyde, and M. Prudhomme, "A comparison of movement direction-related versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task," *Journal of Neuroscience*, vol. 9, pp. 2080-2102, 1989.
- [37] W. T. Thach, "Correlation of neural discharge with pattern and force of muscular activity, joint position, and direction of intended next movement in motor cortex and cerebellum," *Journal of Neurophysiology*, vol. 41, pp. 654-676, 1978.
- [38] S. H. Scott and J. F. Kalaska, "Changes in motor cortex activity during reaching movements with similar hand paths but different arm postures," *Journal of Neurophysiology*, vol. 73, pp. 2563-2567, 1995.
- [39] E. Todorov, "Direct cortical control of muscle activation in voluntary arm movements: a model," *Nature Neuroscience*, vol. 3, pp. 391-398, 2000.
- [40] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, and J. P. Donoghue, "Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter," presented at SAB Workshop on Motor Control in Humans and Robots: on the Interplay of Real Brains and Artificial Devices, University of Edinburgh, Scotland, 2002.
- [41] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.
- [42] Y. Gao, M. J. Black, E. Bienenstock, and W. Wu, "A quantitative comparison of linear and non-linear models of motor cortical activity for the encoding and decoding of arm motions," presented at 1st International IEEE/EMBS Conference on Neural Engineering, Capri, Italy, 2003.
- [43] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue, "Modeling and decoding motor cortical activity using a switching Kalman filter," *IEEE Transactions on Biomedical Engineering*, vol. to appear, 2003.

- [44] A. E. Brockwell, A. L. Rojas, and R. E. Kaas, "Recursive Bayesian decoding of motor cortical signals by particle filtering," *Journal of Neurophysiology*, vol. to appear, 2003.
- [45] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*. Cambridge, MA: MIT Press, 1949.
- [46] S. Haykin, *Neural networks: a comprehensive foundation*. New York: Toronto: Macmillan; Maxwell Macmillan Canada, 1994.
- [47] G. Orr and K.-R. Müller, *Neural Networks: Tricks of the Trade*, vol. 1524. Berlin; New York: Springer, 1998.
- [48] J. C. Príncipe, N. R. Euliano, and W. C. Lefebvre, *Neural and adaptive systems: fundamentals through simulations*. New York: Wiley, 2000.
- [49] J. Moody, "Prediction Risk and Architecture Selection for Neural Networks," in *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, V. Cherkassky, J. H. Friedman, and H. Wechsler, Eds. New York: Springer-Verlag, 1994.
- [50] H. Akaike, "Statistical predictor identification," *Ann. Inst. Statist. Math.*, vol. 22, pp. 203-217, 1970.
- [51] P. Craven and G. Wahba, "Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation," *Numer. Math*, vol. 31, pp. 377-403, 1979.
- [52] G. Golub, H. Heath, and G. Wahba, "Generalized cross validation as a method for choosing a god ridge parameter," *Technometrics*, vol. 21, pp. 215-224, 1979.
- [53] T. Soderstrom and P. Stoica, *System Identification*. New York: Prentice Hall, 1989.
- [54] J. Moody, "The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems," presented at Advances in Neural Information Processing Systems, San Mateo, California, 1992.
- [55] W. C. Lefebvre, J. C. Principe, C. Fancourt, N. R. Euliano, G. Lynn, G. Geniesse, M. Allen, D. Samson, D. Wooten, and J. Gerstenberger, "NeuroSolutions," 4.20 ed. Gainesville: NeuroDimension, Inc., 1994.
- [56] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1999.

- [57] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, pp. 1-58, 1992.
- [58] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Auto. and Control*, vol. 19, pp. 716-723, 1974.
- [59] G. Wahba, *Spline Models for Observational Data*. Montpellier: Capital City Press, 1990.
- [60] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55-67, 1970.
- [61] R. Neal, *Bayesian Learning for Neural Networks*. Cambridge: Cambridge University Press, 1996.
- [62] G. Towell and J. Shavlik, "Interpretation of artificial neural networks: mapping knowledge-based neural networks into rules," presented at Advances in Neural Information Processing Systems, 1992.
- [63] M. Craven and J. Shavlik, "Visualizing learning and computation in artificial neural networks," *International Journal on Artificial Intelligence Tools*, vol. 1, pp. 399-425, 1992.
- [64] W. J. Freeman, "Mesoscopic neurodynamics: From neuron to brain," *Journal of Physiology-Paris*, vol. 94, pp. 303-322, 2000.
- [65] S. Grossberg, *Studies of Mind and Brain: Neural principles of learning, perception, development, cognition, and motor control*, vol. 70. Dordrecht, Holland: Boston, 1982.
- [66] D. J. Crammond, "Motor imagery: never in your wildest dream," *Trends in Neurosciences*, vol. 20, pp. 54-57, 1997.
- [67] Kupfermann, "Localization of Higher Cognitive and Affective Functions: The Association Cortices," in *Principles of Neural Science*, E. R. Kandel, J. H. Schwartz, and J. T. M, Eds., 3 ed. Norwalk, Conn: Appleton & Lange, 1991, pp. 823-838.
- [68] R. Chen, L. G. Cohen, and M. Hallett, "Role of the ipsilateral motor cortex in voluntary movement," *Can. J. Neurol. Sci.*, vol. 24, pp. 284-291, 1997.
- [69] P. Cisek, D. J. Crammond, and J. F. Kalaska, "Neural activity in primary motor and dorsal premotor cortex In reaching tasks with the contralateral Versus ipsilateral arm," *J Neurophysiol*, vol. 89, pp. 922-942, 2003.

- [70] J. Tanji, K. Okano, and K. C. Sato, "Neuronal activity in cortical motor areas related to ipsilateral, contralateral, and bilateral digit movements of the monkey," *J Neurophysiol*, vol. 60, pp. 325-343, 1988.
- [71] J. C. Sanchez, D. Erdogmus, Y. Rao, S. P. Kim, M. A. Nicolelis, J. Wessberg, and J. C. Principe, "Interpreting neural activity through linear and nonlinear models for brain machine interfaces," presented at Intl. Conf. of Engineering in Medicine and Biology Society, Cancun, Mexico, 2003.
- [72] E. E. Fetz, "Are movement parameters recognizably coded in the activity of single neurons," *Behavioral and Brain Sciences*, vol. 15, pp. 679-690, 1992.
- [73] L. Fu and T. Chen, "Sensitivity analysis for input vector in multilayer feedforward neural networks," presented at IEEE International Conference on Neural Networks, San Francisco, CA, 1993.
- [74] S. R. Jammalamadaka and A. SenGupta, *Topics in Circular Statistics*. River Edge, NJ: World Scientific Publishing Company, 1999.
- [75] Q. G. Fu, D. Flament, J. D. Coltz, and T. J. Ebner, "Temporal encoding of movement kinematics in the discharge of primate primary motor and premotor neurons," *Journal of Neurophysiology*, vol. 73, pp. 836-854, 1995.
- [76] J. Hamilton, "Monkeys Learn to Control Robot Limbs," in *All Things Considered*. USA: NPR, 2003.
- [77] S. Blakeslee, "Imagining thought-controlled movement for humans," in *The New York Times*, Late ed. New York, 2003, pp. 3.
- [78] A. Rudolph, "Military: brain machine could benefit millions," *Nature*, vol. 423, pp. 787, 2003.
- [79] W. Gibson, *Neuromancer*. New York: Ace Books, 1994.
- [80] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, pp. 1550-1560, 1990.
- [81] M. A. Nicolelis, A. A. Ghazanfar, B. M. Faggin, S. Votaw, and L. M. Oliveira, "Reconstructing the engram: simultaneous, multisite, many single neuron recordings," *Neuron*, vol. 18, pp. 529-537, 1997.

BIOGRAPHICAL SKETCH

Justin C. Sanchez received a B.S. with Highest Honors in Engineering Science along with a minor in Biomechanics from the University of Florida in 2000. From 1998 to 2000, he spent 3 years as a Research Assistant in the University of Florida, Department of Anesthesiology. In 2000, he joined the Department of Biomedical Engineering at the University of Florida to pursue a Ph.D. in Biomedical Signal Processing. Under the guidance of Dr. Jose C. Principe in the Computational NeuroEngineering Laboratory, his research is in the development and study of Brain Machine Interfaces (BMIs). The goal is to develop models that map the firing patterns of ensembles of cortical neurons to hand position. The models can then be used to study the physiology and function of cortical regions of the brain. His studies are funded by the Defense Advanced Research Projects Agency and are part of a joint research project among Duke, Massachusetts Institute of Technology, State University of New York, University of Florida, and Plexon.